MULTI-FACETED AUTOMATIC CLASSIFICATION OF INSTITUTIONAL REPOSITORY DIGITAL OBJECTS: A CASE STUDY OF THE UNIVERSITY OF ZAMBIA

By

Robert M'sendo

A dissertation submitted to the University of Zambia in partial fulfilment of the requirements of the degree of Master of Science in Computer Science

The University of Zambia

Lusaka

2021

DECLARATION

I, Robert Musendo, declare that the work in this dissertation is original except where indicated by special reference in the text and no part of the research has been sub mitted for any other degree, diploma or academic qualification.

Name ____

Signature ______________________ Date ________________

Supervisor's Name ____________________________________

Signature ______________________ Date ________________

Supervisor's Name ____________________________________

Signature______________________ Date ________________

CERTIFICATE OF APPROVAL

This thesis of Robert M′sendo has been approved as fulfilling the requirements of

the requirements for the award of Master of Science in Computer Science by the University of Zambia.

Examiner 1′s Name ______________________________

.. Signature ____________________ Date ______________

Examiner 2′s Name ______________________________

Signature ____________________ Date ______________

Examiner 3′s Name ______________________________

Signature____________________ Date ______________

Board of Examiners Chairperson′s Name ____________________

Signature____________________ Date ______________

Supervisor′s Name ______________________________

Signature____________________ Date ______________

ABSTRACT

Institutional Repositories (IRs) provide the ability to store, manage, and disseminate intellectual products created by an institution. They provide a complementary method to the traditional system of scholarly communication, making it easier to demonstrate the scientific, social, and financial value of an institution. The potential benefit of an IR goes beyond the desire to increase an institution's profile. They also increase authors' visibility and provide users with easy access to information. Despite the hasty pace at which organizations are creating IRs and with all the potential benefits they offer, recent studies have established that the two biggest existing problems are that of digital objects having missing important metadata elements and the wrong classification of digital objects into communities. This research outlines a case study conducted at the University of Zambia (UNZA). The aim of this study was to design, develop, and implement three classification models and a prototype tool that uses the models for effective ingestion of digital objects into an IR. To achieve this, firstly, a situational analysis was conducted to appreciate the challenges of the current system being experienced in the tagging and ingestion of digital objects into the IR. Furthermore, an exploratory study was conducted in order to assess the full extent of the problem. Finally, three classification models were implemented. Experiments on classification using the developed models were conducted, and the results demonstrated the possibility of automatically classifying digital objects into an IR with an accuracy of 77 % for the collection classification model, 75 % for the document type model, and 0.005 % Hamming loss for the subject classification model. The results suggest that our proposed technique can help address the two biggest existing problems related to IRS.

Keywords: Institutional Repositories, Metadata, Digital Objects,Classification.

ACKNOWLEDGEMENTS

I would like to thank the following people for their various support during my research study; Dr. Lighton Phiri and Dr. Matumbo Nyirenda for the supervision, Dr. Jackson Phiri, for the encouragement and support. Lastly but not the least, I would also like to thank all Respondents for their participation in the baseline study. I also would to thank My children for their understanding because I didn't play with them during the time I was typing the dissertation.

Thank you

Contents

List of Figures

List of Tables

List of Abbreviations

ACM	Association for Computing Machinery
API	Application Programming Interface
BoW	Bag of words
CRISP- DM	Cross - Industry Standard Process for Data Mining
DRGS	Directorate of Research and Graduate Studies
ETD	Electronic Theses and Dissertations
FP	False Positive
FN	False Negative
HEI	Higher Education Institutions
HTML	Hyper Text Markup Language
KDD	Knowledge Discovery Database
IR	Intuitional Repository
NIPA	National Institute of Pbulic Administration
SEMMA	Sample, Explore, Modify, Model, Assess
TF	Term Frequency
TF-IDF	Term Frequency - Inverse Document Frequency
TP	True Positive
TN	True Negative
UNZA	University of Zambia

Chapter 1

Introduction

1.1 Introduction

This chapter introduces this research study. The chapter is sectioned as follows: background to the study, statement of the problem, aim of the study, research objectives, research questions, significance of the study, organization of the thesis, and lastly, the summary of the chapter.

1.2 Background

Teaching and conducting research are two of the primary responsibilities that Higher Education Institutions (HEIs) are expected to fulfill. During the course of these activities, digital content is produced, including but not limited to seminar papers, conference papers, technical reports, datasets, theses and dissertations, pre-print and post-print journal articles, images, audio, and video contents [1]. The increased utilization of computers and the internet in the process of information generation has resulted in a significant expansion of these digital materials. Institutional repositories have been implemented at academic institutions in an effort to improve the accessibility, worldwide exposure, and efficient management of digital content in these institutions.

An IR is a digital archive that is designed to capture, preserve, and make available the digital work of a community [2] HEIs set up an IR to enhance the visibility and accessibility of their research output.

The University of Zambia (UNZA) has a working IR that is constantly filled with academic work. Digital objects are put into IRs either by self-archiving [3] , in which case the authors of the article are responsible for depositing the digital object, or by a central authority, usually the library. In both cases, it is possible to misclassify digital items by putting them in the incorrect collection and, more significantly, by missing out certain descriptive terms. In the case of UNZA, this has been exacerbated by the fact that self-archiving of digital objects is non-existent, and the Library only has two individuals in charge of IR object ingestion.

A quick survey into UNZA IR showed that some of the digital objects were wrongly classified and others had missing descriptive metadata. Descriptive metadata describes the intellectual content of a digital object. A resource identifier, which uniquely identifies the object, is the most important element of descriptive metadata. Title, author, date of publication, subject, publisher, and description are examples of descriptive metadata elements. The use of descriptive elements aids in the discovery and location of digital resources. Descriptive metadata is also used to document and track the intellectual provenance of digital resources (e.g., origin, enhancement, and annotation), which is critical for certain types of research collections [4].Habukali et al [5] alluded to this problem as a result of the current IR workflow. Their investigation revealed that an IR administrator currently tags and deposits digital objects manually. Manual methods of bringing in digital objects have been shown to be tedious, time-consuming, and prone to mistakes. This means that some digital objects are wrongly categorized and others have missing metadata. In practice, misclassification of objects or objects with missing metadata leads to ineffective content searches—those that return the wrong resources or, worse, none at all, rendering them invisible to the intended user.

1.3 Statement of the Problem

The UNZA library is responsible for the ingestion/submission of digital objects into the UNZA IR. The library has continued to face challenges in the ingestion/submission process. Errors such as wrong classification of digital objects and incomplete metadata [6] of digital objects have continued to occur. In practice, wrong classification of digital objects or missing metadata results into ineffective searches for content, ones that recall the wrong resources or worse still result into no resource which makes them invisible to the intended user.

1.4 Motivation and significance of the thesis

A well-organized IR is a drive to help universities show case its scholarly output to the world. The motivation of this study is to use machine learning to achieve factors that can improve the process of ingesting and organizing the digital objects into an IR. The findings of the research can also be applied to other institutions which could be experiencing similar problems of missing metadata and wrong classification of digital objects. Furthermore, the results of this study will hopefully be useful in stimulating further research on multi-faceted automatic classification of digital into an IR and also contribute to the existing body of knowledge.

1.5 Aim of the Study

To develop and implement classification models and prototype tool that will use the classification models for effective ingestion of digital objects into the UNZA IR.

1.6 Research objectives

i. To analyse how objects are organised into the IR.

ii. To analyse how objects are tagged prior to ingestion into the IR.

iii. To implement a model for automatic classification of an IR's digital objects.

1.7 Research questions

i. How are digital objects currently organised in the UNZA IR?.

ii. How are digital objects tagged with metadata during ingestion into the UNZA IR?.

iii. What is feasible to implement a model for automatic classification of IRs digital objects?.

1.8 Research contributions

This research was aimed at improving the tagging and ingestion process of digital objects in an IR. The study was conducted using a case study of The University of Zambia's IR. The major contribution was the baseline study and building of the models and a prototype tool to use the model for effective ingestion of IRs digital objects.

1.9 Organization of the thesis

This report is divided into five chapters: Chapter 1 is the introduction to the research and it gives the overview of the research, the aim, significance and research objectives. It concludes with a summary of the chapter. Chapter 2 is a discussion on various literature that was reviewed around the subject area of Document classification and Institutional Repository Chapter 3 details the methodologies that were used in the research study. The main methods that were adopted in this research were, interviews, design and development of a model and prototype tool. Chapter 4 outlines the research findings of the baseline study and the system design and implementation. Chapter 5 presents the discussion and conclusion.

Chapter 2

Literature Review

2.1 Introduction

This chapter will focus on discussing the concepts and algorithms used and furthermore explain the possible approach to solve this problem and as well as giving an overview to the work related to this dissertation. The chapter concludes with a summary of the chapter.

2.2 Institutional Repositories

IRs are critical to development, management and leveraging enterprise wide digital content and bring greater value to an Institutions' output. A digital repository that allows academic and research institutions to store, preserve, and disseminate the scholarly work of their academics, students, and staff. Research papers, theses, dissertations, conference proceedings, technical reports, databases, and other digital assets are frequently included in institutional repositories[7]. The major objective of institutional repositories is to enhance the visibility, accessibility, and impact of the institution's research output. By making research accessible online, institutional repositories can improve the transmission of knowledge and make it easier for scholars, researchers, and the general public to discover research [8].

In addition to serving as a platform for storing and distributing research output, institutional repositories also provide teachers, students, and staff with a variety of services and functions that might support their research efforts. For instance, institutional repositories may include tools for organizing research data, measurements on research effect, and researcher collaboration.

An important advantage of institutional repositories is that they can aid in the preservation of digital materials throughout time. By providing a central location for the storage of research output, institutional repositories can contribute to the preservation and accessibility of knowledge for future generations. Institutional repositories may also include tools and services for managing metadata, which can contribute to the discoverability and accessibility of research over time.

Another essential characteristic of institutional repositories is their ability to facilitate open access to research output. Open access refers to the practice of making research freely available online, at no cost and with no access restrictions. By offering a venue for open access publishing, institutional repositories can serve to democratize access to knowledge and expand the reach of research.

Institutional repositories support the research operations of academic and research institutions significantly. By offering a platform for storing, archiving, and disseminating research output, institutional repositories can improve the visibility, accessibility, and impact of research, as well as long-term preservation and open access to knowledge.IRs are designed in such a way that they allow for the efficient and effective storing and retrieval of two fundamental features of digital objects: metadata and bitstreams [9].

2.2.1 Core functions of Institutional Repository

All IRs have core functions. Foster [10] summarized that there are six main functions of an IR;material submission, metadata application, access control, discovery support, distribution and preservation.

Material Submission

The main purpose of an IR is to preserve, manage and disseminate the intellectual products created by institutions. Prior to the management and dissemination of digital objects, an IR must have some mechanism which an author or proxy can deposit content into it. Usually material submission is done via a web form accompanied by the uploading a file if not, hard copy is deposited to the librarian where the copy is converted into a soft copy. This process is sometimes referred to as ingestion.

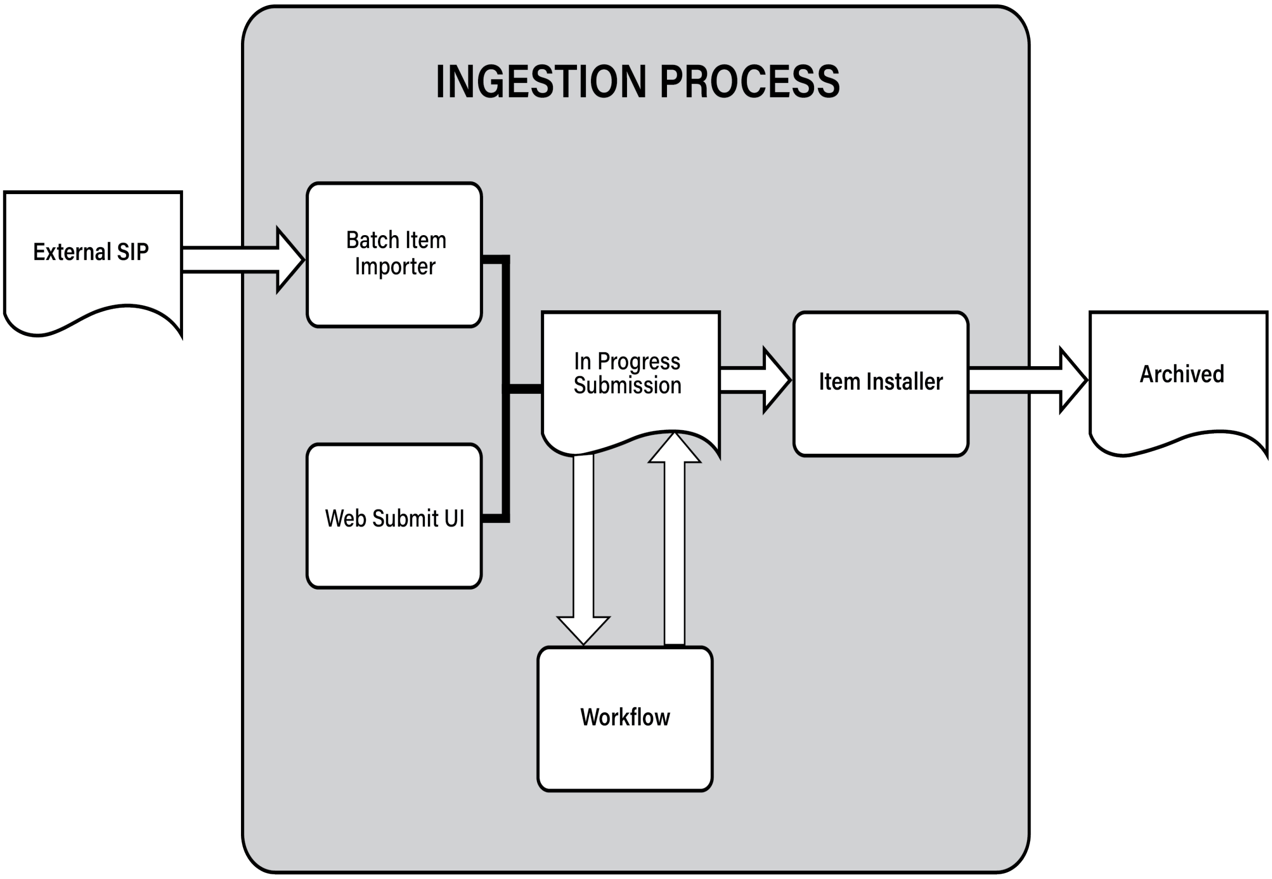


FIGURE 2.1: IR Ingestion Process

Metadata Application

Once the document has been submitted, metadata must be appended to it. In most cases a set of basic information used to identify each document, such as the title, subject and author are mandatory during the submission process. Abstracts, keywords and other descriptive metadata must be provided although some of which is optional. Administrative metadata is usually automatically supplied by the system, such as date and time of deposit.

Access Control

Access control defines the mechanism put in place in order to manage and protect access to the resources. This core function involves both authentication and authorization. IRs provide access to the digital content they maintain, generally via a web-based interface that enables users to search and view the content. With the implementation of open access policies and license agreements, IRs can also ease the distribution of content

Discovery Support

IRs provide access to the digital content they maintain, generally via a web-based interface that enables users to search and view the content. With the implementation of open access policies and license agreements, IRs can also ease the distribution of content.

Preservation

IRs are intended to assure the long-term preservation of digital content by employing storage and backup systems, migrating to new formats, and managing metadata to ensure continuing accessibility.

2.2.2 Data Mining

Data mining is the process of discovering pattern previous unknown by analyzing large data set and extracting valuable information from it for different purposes [11] It is the process of extracting useful information from large amounts of data. It achieves this by combining traditional data analysis with sophisticated algorithms to process vast amount of data. It is an interdisciplinary field merging concepts such as database system, statistics, machine learning, computing, information theory and pattern recognition. In order to interpret the business challenges into mining task, data mining requires a standard approach. As depicted in Figure 2.2

2.2.3 Knowledge Discovery Database (KDD)

The KDD is a repetitive and interactive process that consists of selection, cleaning and transformation of data extracted from not only from databases but also from other sources such as spreadsheets, data ware housing, images text, etc [11] It comprises a total number of nine steps namely Domain understanding and KDD goal, Selection and Addition, Reprocessing Transformation, Data mining, Evaluation and Implementation and Discovered Knowledge as depicted in Figure 2.3 [11] The main

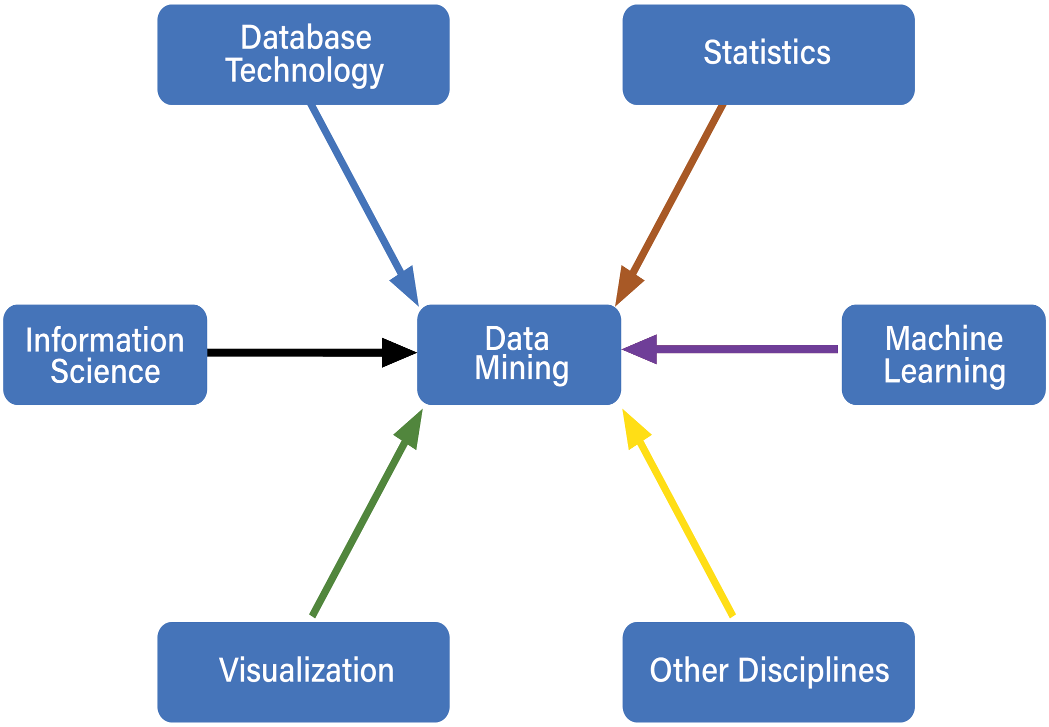


FIGURE 2.2: Data mining as interdisciplinary field

aim of KDD process is to apply to them data mining algorithm in order to discover valid, novel, potentially useful and understandable hidden pattern.

Domain Understanding and KDD Goals

Developing and understanding of the application Domain is the first phase of the KDD process.This stage is aimed at defining the goals from the customer's perspective and later used to create and comprehend about the application domain and its pre-knowledge.

Creating a Target Dataset

Creating a target dataset is the second phase which is aimed at gathering all the important data for the purposes of wanting to create a dataset. This is very imperative because data mining heavily depends on the available data in order to discover the hidden patterns.

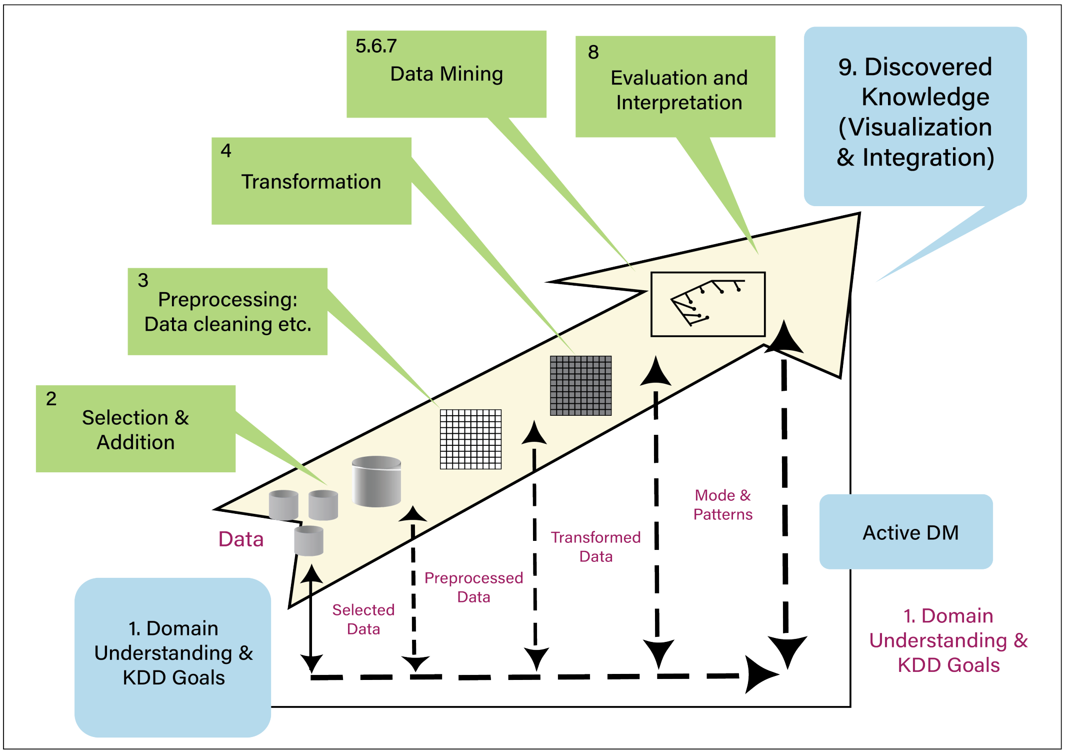

5.6.7
Data Mining
8
Evaluation and Interpretation
9. Discovered Knowledge (Visualization & Integration)
4
Transformation
3
Preprocessing: Data cleaning etc.
2
Selection & Addition
Data
Selected Data
Preprocessed Data
Transformed Data
Mode & Patterns
Active DM
1. Domain Understanding & KDD Goals
1. Domain Understanding & KDD Goals

FIGURE 2.3: Knowledge Discovery Database (KDD) Process Model

Data Cleaning and Pre-processing

The dataset created in the second phase usually contain noise which might lead to the model not working very well thus, the data cleaning and pre-processing is concentrating on the removing all outliers and also handling of missing values. At this stage sometimes it involves the application of statistical methods or data mining algorithms.

Data Transformation

Data transformation is the fifth phase of the KDD which focuses at converting the data from one form to another. This phase is very critical for the all success of the KDD project. Thus, various data reduction and transformation techniques are implemented on the targeted data.

Choosing the appropriate Data Mining task

This phase involving the process of selecting which data mining to use, among the different types,for example classification, regression or clustering. The preference is usually guided by the major goals of the KDD.

Choosing the Data Mining Algorithm Choosing the data mining algorithm is the sixth stage which involves the selecting of the appropriate algorithm to use from the variety of them. This phase also involves the process of choosing a particular method to be used for a searching pattern.

Employing the Data Mining Algorithm

This phase involves the implementation of the data mining algorithm. This stage may require to employ the algorithm several times till the desired result is obtained if not then the algorithm's parameters is fine tuned.

Evaluation

The evaluation phase of the KDD is aimed at assessing and interpreting the mined patterns against the KDD goals which were earlier on stipulated in the first phase.

Using Discovery Knowledge

This is the last phase in the KDD process were the discovered knowledge is used for different purposes. At this stage the discovered knowledge can also be integrated with other systems for further action. Actually the success of this phase determines the success of the entire KDD process.

2.3 Sample ,Explore , Modify ,Model,Assess

Sample, explore, modify, model, assess (SEMMA) is data mining methodology developed by the SAS Institute. It defines the core process of conducting data mining [12]. It offers and allows understanding, organization, development and maintenance of data mining projects. This process is made up of five steps namely sample, explore, modify, model and assess as depicted in Figure 2.4.

2.3.1 Sample

Sample is the first phase in the SEMMA process although it is optional, it concentrates on getting a portion of the data from data set in order to have a snapshot of the

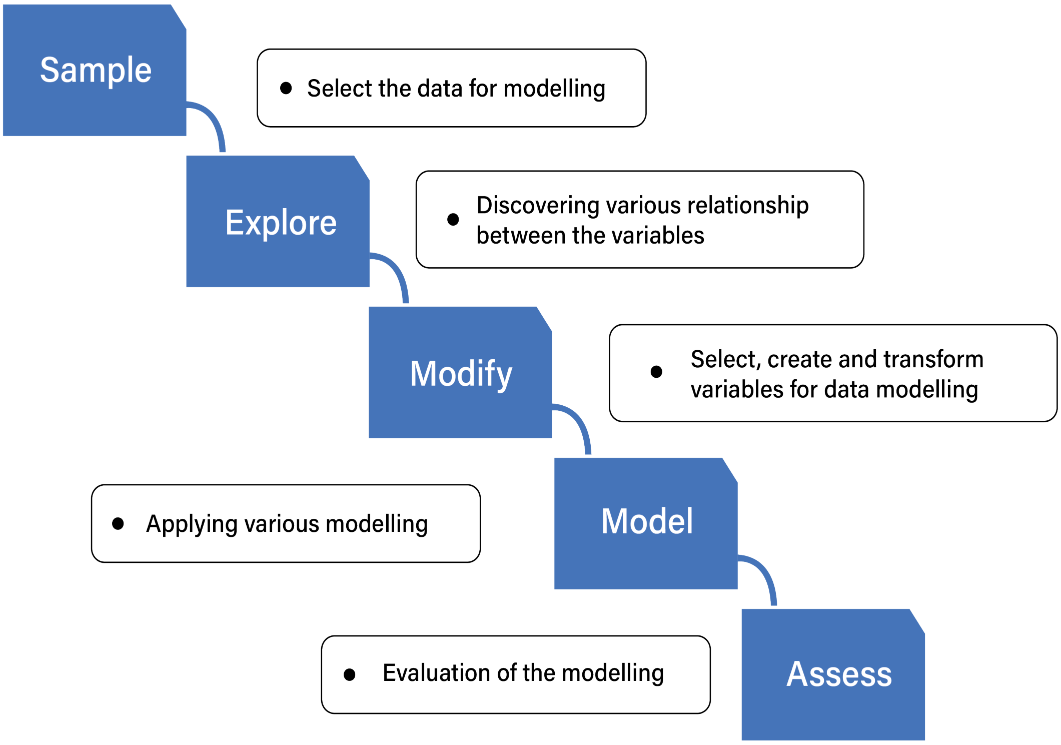

Sample
Select the data for modelling
Explore
Discovering various relationship between the variables
Modify
Select, create and transform variables for data modelling
Applying various modelling
Model
Evaluation of the modelling
Assess

FIGURE 2.4: Steps in the SEMMA Methodology

whole picture. The sample is usually small enough to easily process but it is large to provide us with all the necessary information.

2.3.2 Explore

Explore is the second phase of the SEMMA process which focuses on discovering unanticipated and anomalies in order to gain understanding and ideals as well as refining the discovery process.

2.3.3 Modify

This is the third phase of the SEMMA process which aims at manipulation of data by creating,selecting and transformation of variables for merging of data. This phase also searches for outliers and reducing the number of variables.

2.3.4 Model

Model is the fourth phase of the SEMMA process which aims at modeling the data. The application for this will automatically searches for combination of data.There are different modeling techniques are present and each technique has its own strength and is appropriate for specific situation on the data for data mining.

2.3.5 Access

This is the final phase of the SEMMA process which focuses on the accessing of the reliability and usefulness of findings and estimates the performance. A communal method mostly used to access the performance of the model is by using the data which was set aside during the sample phase. If the model has been built well then it must work for the reserved sample as well as for the sample used to construct the model.

2.4 Cross - Industry Standard Process for Data Mining

Developed in 1996 by a consortium formed by Daimler -Benz and NCR. CRISP-DM is a model that provide a structured approached to planning of data mining [13]. It is made up of six different steps or phases namely Business understanding, Data understanding, Data preparation, Modeling, Evaluation and Deployment as depicted in Figure 2.5 [14].

2.4.1 Business Understanding

This is the second stage of the CRSIP-DM process, which focuses on data gathering, quality checking, and exploring the data to get insight and form an assumption about the hidden data. This stage involves four sub-steps: the collection of initial data, the description of the data, the exploration of the data, and finally the verification of the data quality.

2.4.2 Data Understanding

This is the second stage of the CRSIP-DM process which concentrates on data gathering, quality check and exploring the data to get insight of the data to form an assumption for the hidden data. This stage involves four sub- steps, collection of initial data, the description of the data, the exploration of the data and finally the verification of the data quality.

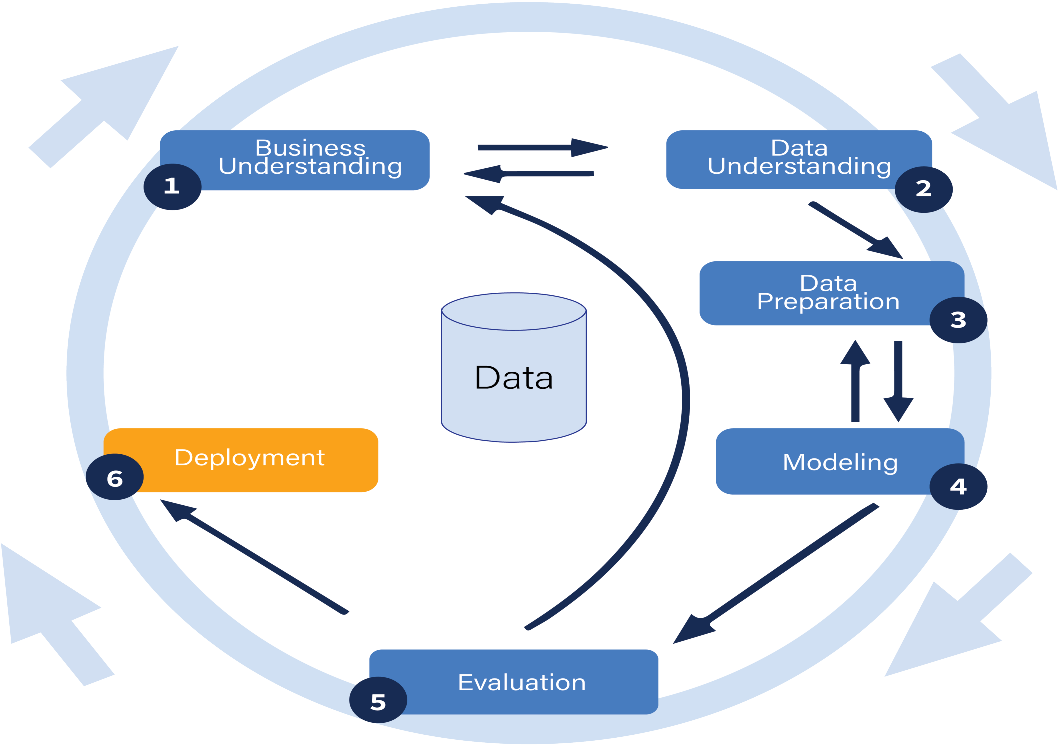


FIGURE 2.5: CRISP -DM

2.4.3 Modelling

In this phase, the data mining team selects and applies appropriate modeling techniques to the prepared data. They may use techniques such as decision trees, neural networks, or support vector machines to build models. The models are then evaluated using validation data and refined as needed.

2.4.4 Evaluation

This is the fifth stage of CRISP-DM process which aims at ascertaining if the model achieves the business objectives. It is at this point that analysis is done to critically determine if some important business issues have not been sufficiently been considered. Interpretation of the model depends upon the algorithms upon the algorithms and models are evaluated to review whether or not the objectives have been achieved.

2.4.5 Deployment

This is the final stage of the CRISP-DM process which focuses on determining use of obtained knowledge and results.

2.5 Machine Learning Techniques

Machine learning is a powerful tool that can be used in institutional repositories to improve the accessibility, searchability, and discoverability of the content stored within them. Some of the ways in which machine learning can be useful in institutional repositories include [15].

1. Automated metadata generation: Institutional repositories contain a large amount of digital content, which often lacks sufficient metadata to make it easily discoverable. Machine learning can be used to automatically generate metadata by analyzing the content of the repository and extracting relevant information such as keywords, authors, and publication dates.

2. Content recommendation: Machine learning algorithms can analyze the content of the institutional repository and identify patterns and relationships between different items. This can be used to provide personalized content recommendations to users based on their interests and preferences.

3. Text mining: Machine learning can be used to extract useful information from the full text of documents stored in the institutional repository. This can include identifying key concepts and themes, extracting relevant data and statistics, and summarizing the content in a way that makes it more accessible to users.

4. Image and video analysis: Institutional repositories often contain a large number of images and videos, which can be difficult to search and browse through manually. Machine learning can be used to analyze the content of these files and automatically identify relevant features, such as objects, faces, and landmarks, making it easier for users to find the content they are looking for.

5. Quality control: Machine learning can be used to identify and flag errors, inconsistencies, and other quality issues within the content of the institutional repository. This can help to improve the overall quality of the repository and ensure that users are accessing accurate and reliable information.

6. Document Classification : Using machine learning for document classification in an institutional repository (IR) can greatly improve the organization and discoverability of digital content stored in the repository. Here are the steps involved in using machine learning for document classification in an IR:

Using machine learning for document classification in an IR can greatly improve the discoverability of digital content stored in the repository. By automatically categorizing documents into predefined categories, users can easily search and browse the content they are interested in. This can improve the user experience and encourage more use of the repository. Additionally, machine learning can be used to identify similar documents and recommend related content to users, further improving the discoverability of the digital assets stored in the IR.

Kavakiotis et al defined machine learning as the field of study where "machines learn from experience" [16]. Thus, with help of machine learning model, the machine captures some input and thereafter produce some output. They are basically three main different types of machine learning, namely supervised learning, unsupervised learning and reinforcement learning.

2.5.1 Supervised Learning

In supervised learning, algorithms learn from the labelled data, having understood the data, the algorithm determines which label must be appended to the new data based on the pattern and associating the pattern of the unlabeled new data. A supervised algorithm takes input elements and their corresponding output in order to train the model to predict the values (predictions) of the future inputs[17] The input elements (data) whose output is predefined are called training data set and it is with this data that the model is trained to predict unknown values. Supervised Learning can be divided into two categories namely classification and Regression. Classification predicts the category the data belongs to, for example blood group, on the contrary Regression predicts a numerical value based on the previous observed data. i.e. House price prediction, height –weight prediction [16] 2.6 illustrates the workflow of the all the phases of supervised learning process.

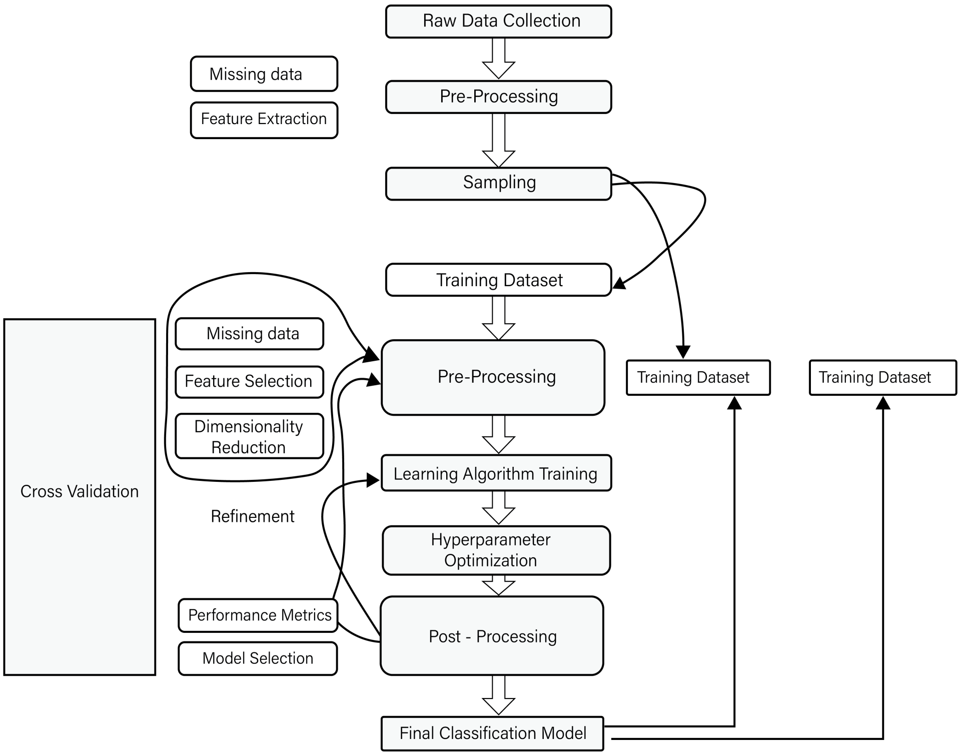


FIGURE 2.6: Workflow of Supervised Learning Process

2.5.2 Classification Algorithms

Classification is a machine learning algorithm technique used to predict the class the dependent belongs to based on one or more independent variables [18] [19]. The process starts with predicting the class given the points. The classes are often referred to as target, label or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y). The main goal is to identify which class/category the new data will belong to. In the follow section , a summary of the most popular machine learning classifiers used in this study is discussed - : Stochastic Gradient Descent, Logistic Regression, Support Vector Machine, Multi-nominal, Random Forest and Decision Tree

Logistic Regression

Logistic regression is a statistical technique used to assess and model the connection between a dependent variable that is binary or dichotomous and one or more independent variables. It is a kind of regression analysis that is frequently employed in statistical modeling to estimate the likelihood of a particular occurrence or result based on the values of the independent variables [20].

The logistic regression model is based on the logistic function, which transforms any real-valued input to a value between 0 and 1 that represents the event's occurrence probability. Given the logistic function, the model predicts the coefficients of the independent variables that maximize the likelihood of the observed data.

Logistic regression is frequently applied to issues involving binary categorization, such as predicting if a client will purchase a product, whether a patient has an illness, or whether a loan application will be granted. Nevertheless, it may also be used to multi-class classification problems by applying it to each class individually or by employing techniques such as one-versus-all or softmax regression.

Logistic regression provides a number of advantages over other classification methods, including its simplicity, interpretability, and resistance to outliers. Nevertheless, it presupposes that the connection between the dependent and independent variables is linear, which is not always the case. In circumstances when the classes are very unbalanced or the independent variables are highly connected, it may also perform poorly.

Decision Tree

A decision tree is a popular machine learning algorithm that can be used for classification and regression tasks. It partitions the data recursively into smaller subsets based on the values of the input features until a stopping criterion is met. The end result is a tree-like structure, with each node representing a decision made based on a specific input feature and each leaf node representing a class label or a numerical value for regression [21].

Support Vector Machine

Support Vector Machines (SVM) is a discriminative classification technique that derives from the Structural Risk Minimization principle from computational learning theory. The aim of SVM is to find the most optimal classification function that differentiates between units of classes in training data. With a linearly separable dataset, the most optimal classification function can be decided by constructing a hyperplane which maximizes the margin between two datasets and thus creates the largest possible distance between datasets[22].

Naïve Bayes

The Nave Bayes method is a probabilistic machine learning technique that is often used to solve classification issues. It is founded on Bayes' theorem, which asserts that the likelihood of a hypothesis (in this case, a class label) given some evidence (in this case, a collection of traits) is proportional to the probability of that evidence given the hypothesis, multiplied by the hypothesis's prior probability [23]. Given the class name, the "nave" in Nave Bayes originates from the assumption that the characteristics are independent of each other. This assumption allows us to reduce the algorithm's computations, making it more efficient and less prone to overfitting.

The technique begins by estimating the prior probability of each class label based on its frequency in the training data. The conditional probability of each feature given each class label is then calculated, again based on the frequency of that feature in the training data for each class. Eventually, it utilizes these probabilities to compute the posterior probability of each class label given the characteristics of a new instance, and the class label with the greatest probability is chosen as the projected class. Text categorization, spam filtering, sentiment analysis, and medical diagnosis are all examples of how Nave Bayes has been utilized successfully. One of its benefits is that it works well with small training sets and can handle both category and numerical data. It may, however, fail when strongly correlated characteristics are present or when the independence assumption is broken.

Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model [24] A random forest algorithm consists of many decision trees. The 'forest' generated by the random forest algorithm is trained through bagging or bootstrap aggregating. Bagging is an ensemble meta-algorithm that improves the accuracy of machine learning algorithms.

Stochastic Gradient Descent (SGD)

Stochastic gradient descent is an optimization algorithm often used in machine learning applications to find the model parameters that correspond to the best fit between predicted and actual outputs. It's an inexact but powerful technique [25]. In Stochastic Gradient Descent, a few samples are selected randomly instead of the whole data set for each iteration. In Gradient Descent, there is a term called "batch" which denotes the total number of samples from a dataset that is used for calculating the gradient for each iteration. In typical Gradient Descent optimization, like Batch Gradient Descent, the batch is taken to be the whole dataset. Although, using the whole dataset is really useful for getting to the minimal in a less noisy and less random manner, but the problem arises when our datasets gets big.

2.5.3 Multinomial

Multinomial is a term that refers to a type of probability distribution in statistics. The multinomial distribution is used to model situations where there are more than two possible outcomes or categories, and the outcome of a trial is classified into one of these categories [26]. The multinomial distribution can be thought of as a generalization of the binomial distribution, which is used to model situations where there are only two possible outcomes (success or failure). In the multinomial distribution, the probabilities of each category are represented by a vector of parameters, and the outcome of a trial is modeled as a vector of counts for each category.

The multinomial distribution is widely used in various fields, including genetics, ecology, psychology, and marketing, among others. In machine learning and natural language processing, the multinomial distribution is used to model the probability distribution over a set of discrete outcomes, such as the probability distribution over a set of words in a document.

There are several algorithms that are based on the multinomial distribution, including the Naive Bayes algorithm, which is a popular algorithm for text classification and sentiment analysis.

2.5.4 Types of Classifications

There are basically four major types of classification tasks in Machine learning: Binary Classification, Multi-class classification, Multi-Label Classification and Imbalanced Classification[27].

Binary

The goal of a binary classification task is to divide the input data into two mutually exclusive categories. Depending on the problem at hand, the training data is labeled in a binary format: true and false; positive and negative; O and 1; spam and not spam, and so on. For example, we might want to determine whether a given image represents a truck or a boat.

Multi-Class

In multi-class classification, input data is classified into more than two classes or categories. Classifying images of animals, for example, into categories such as cats, dogs, and birds.

This is a type of classification where each sample is tagged to a set of target labels (more than one class). Sometimes it can also be defined as a classification that has two or more class labels, where one or more class labels may be predicted for each example.

Multi-Label

Multi-label classification is a type of classification in which each input data point is assigned multiple labels or categories. For instance, labeling a film as having multiple genres such as comedy, romance, and action. With multi-label More than one class is available and several classes (called labels) can be assigned at once as illustrated in figured 2.8.Approaches to Multi-label Classification can be grouped into two different methods.Problem Transformation denotes the approach of transforming the multi-label into a number of easier classification problems, i.e. classifying each item with a Binary Classification where for each label a classifier decides to assign the label or not. Another possible approach is to use algorithms that were adopted to this very problem, which is being referred to as Algorithm Adaption[28].

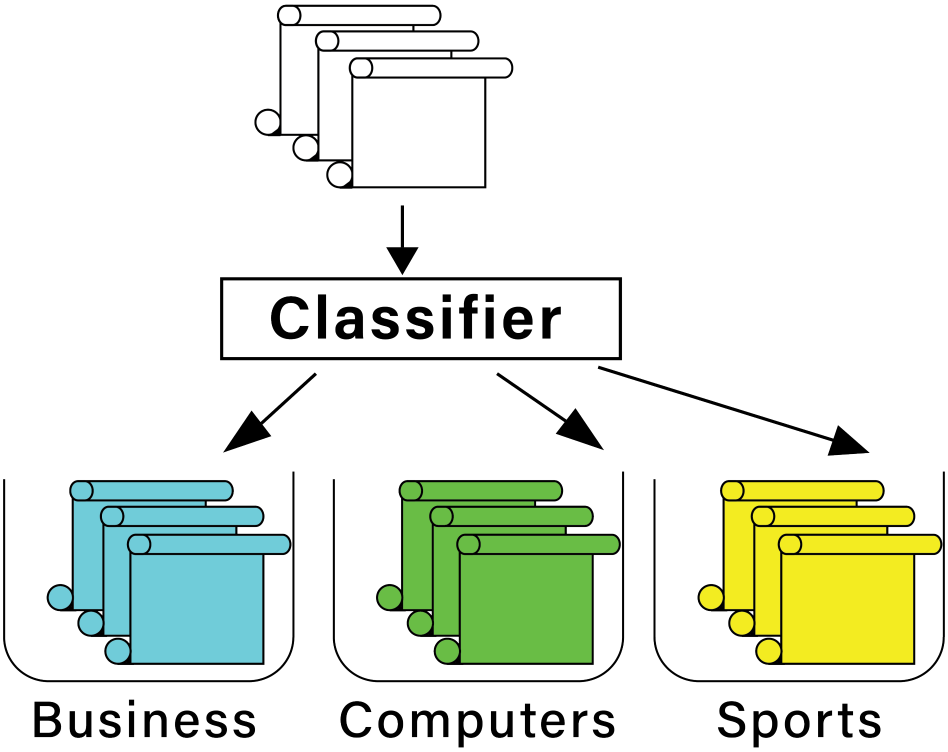


FIGURE 2.7: Multi-Label Classification

Imbalanced

Imbalanced classification deals with datasets in which the number of instances in one class is significantly greater than the number of instances in the other classes.

This is common in real-world applications where one class, such as fraud detection, is uncommon.

2.5.5 Unsupervised Learning

Unlike Supervised Learning, unsupervised learning deals with unlabeled dataset, the system itself thrives to discover the hidden pattern of the data and its associations between the data [29]. With unsupervised learning, the training data instances have no corresponding labels [16].

2.5.6 Reinforcement Learning

Reinforcement Learning (RL) can be perceived as an approach which lies between supervised and unsupervised learning. Reinforcement learning involves no supervisor and only a reward signal is used for an agent to determine if they are doing well or not [29].Time is a key component in RL where the process is sequential with delayed feedback. Each action the agent makes affects the next data it receives. The agent needs to find the "right" actions to take in different situations to achieve its overall goal. In other words, Reinforcement learning allows machines to establish automatically its behavior within a specific content to maximize its performance.

2.5.7 Deep Learning

Deep learning is a branch of machine learning that use multiple-layered artificial neural networks to learn complicated data representations. Deep learning models have reached state-of-the-art performance in a variety of applications, including picture and audio recognition, natural language processing, and game playing [30].

Deep learning relies on artificial neural networks, which are modeled after the structure and function of biological neurons. Neural networks are composed of layers of linked nodes, or neurons, which process information and acquire representations of the incoming data. There are numerous layers of neurons in a deep neural network, with each layer building on the preceding one to generate increasingly abstract and complicated data representations.

Backpropagation, a variation of stochastic gradient descent, is commonly used to train deep learning models using huge volumes of labeled data and a variant of stochastic gradient descent. The model changes its weights and biases during training to reduce the discrepancy between its predictions and the actual labels. Training a deep neural network may be computationally intensive and calls for specialized hardware such as graphics processing units (GPUs) or tensor processing units (TPUs) (TPUs).

Deep learning has revolutionized artificial intelligence and enabled substantial advancements in domains such as computer vision, natural language processing, and speech recognition. Image categorization, object identification, speech recognition, language translation, and game playing are all applications of deep learning. Deep learning is also utilized in areas such as healthcare, banking, and autonomous driving to solve complicated issues and create data-driven predictions.

2.5.8 Representation of Text

Bag of Word

The Bag of words(BoW) represents statements and sentences as multi set of words with its main concert ration on storing all the occurrences of an element while the order is ignored.The BoW is used to form a vector representing the document using the frequency count of each word in the document.This approach of document representation is called Vector Space Model(VSM).[16]

In n-gram spatial information is captured by keeping the occurrences of n words appearing concurrently in a document , thus a text is represented as set words , with no grammar and word order but keeping the multiplicity. The Bow is the most common method in document classification.

Word2vec

Word2vec is a well-known approach for producing vector representations of words from huge natural language text collections. Tomas Mikolov and his colleagues at Google created the algorithm in 2013, and it has subsequently gained widespread use in natural language processing and text mining applications [31].

Word2vec learns vector representations of words from a vast corpus of literature using a shallow neural network. The algorithm receives a huge corpus of text as input and learns to estimate the likelihood of a word occurring in the context of other terms in the corpus. The model's output is a set of vectors, one for each word in the vocabulary, that represent the meaning and connections between the words in the text.

The word2vec technique has two primary variants: continuous bag-of-words (CBOW) and skip-gram. CBOW predicts the target word based on the surrounding context words, whereas skip-gram predicts the context words based on the target word. Both types are excellent in learning high-quality vector representations of words, with the decision depending on the given task and data collection.

word2vec generates word vectors with various advantageous qualities, including the capacity to record semantic associations between words. For instance, semantically identical phrases tend to have comparable vector representations, whereas dissimilar words tend to have distinct vector representations. This makes word2vec a potent instrument for natural language processing applications including text categorization, information retrieval, and sentiment analysis.

It has been demonstrated that Word2vec's vector representations increase the performance of many natural language processing applications. Word2vec is now a commonly used tool in the field of natural language processing.

One-hot-Vector

In machine learning and natural language processing, one-hot encoding is used to encode category data as binary vectors. The representation of the one-hot vector is a binary vector of length N, where N is the number of potential categories. The vector has the value 1 at the index corresponding to the data category and 0 at all other indices.

Consider, for instance, a color dataset including the categories red, blue, and green. The one-hot vector representations of the colors red, blue, and green are [1, 0, 0], [0, 1, 0], and [0, 1], respectively.

Natural language processing applications frequently employ one-hot encoding to represent textual data. In this instance, each vocabulary word is represented as a one-hot vector, and the vectors are utilized as input for a machine learning model. One-hot encoding can also be used to represent user IDs and product categories.

One-hot encoding is a simple and efficient method for expressing categorical data, although it has significant disadvantages. One of the primary disadvantages is that the generated vectors are frequently quite high-dimensional, which can make training machine learning models computationally costly and challenging. In addition, one-hot encoding does not record connections between categories, which might be crucial for some applications [32].

Glove

GloVe (Global Vectors) is an unsupervised learning system created by Pennington, Socher, and Manning at Stanford to generate word embeddings. The program attempts to capture the semantic meaning of words by building dense vector representations of words depending on their context of occurrence [33].

GloVe generates a co-occurrence matrix from a huge corpus of text, which indicates the frequency with which each word appears in the context of every other word. The method then factors this matrix to obtain a representation of each word with fewer dimensions. The resultant word vectors can be utilized for a variety of natural language processing applications, including language modeling and sentiment analysis.

GloVe's ability to capture both global and local information about the connections between words is one of its benefits over other word embedding methods, such as word2vec. This enables it to grasp nuanced semantic links between words and perform well on a number of tasks involving natural language processing.

Term Frequency-Inverse Document Frequency

Term Frequency-Inverse Document Frequency (TF-IDF) is a method frequently used in natural language processing for determining the significance of terms in a text or corpus. It is founded on the premise that words that are significant to a text are likely to appear frequently in that document, but seldom in other papers in the corpus [34].

Multiplying the Term Frequency (TF) by the Inverse Document Frequency yields the TF-IDF (IDF). Calculated as the number of occurrences of the term divided by the total number of words in the document, the Term Frequency indicates the frequency with which a term appears in a text. Calculated as the logarithm of the total number of documents in the corpus divided by the number of documents that contain the word, the Inverse Document Frequency indicates how uncommon a phrase is across all documents in the corpus.

The TF-IDF score for a phrase within a document is then computed by multiplying the TF and the IDF. This score represents the significance of a phrase in a document compared to its significance in the corpus.

TF-IDF is frequently employed for text categorization, information retrieval, and other applications involving natural language processing. It is frequently used to translate text data into a numerical representation that may be utilized as input for machine learning models.

Tf-idf is weight is made of two words namely: Term Frequency: Which aims at measuring how frequent a term appears in a document. Documents are of different length , a term would appear many times in long document than shorter one. TF(t)=(Number of times t appears in a document)/(Total Number of terms in the document). Inverse Document Frequency:This defines how important a term is. Weighting down the frequent term and scale up the rare ones.

IDF(t)=loge(Total number of documents/Number of documents with terms t in t).

In this model , text is represented as a set of word and we applied TF-IDF method to it as well as n-gram.

2.6 Classification of Scholarly Work

Due to the increased desire by Higher Education Institutions (HEIs) to expedite classification of digital objects and also the desire for proper tagging of objects necessitated document classification to become a well-studied problem. There are various methods that can be used for extraction of features from a text document, however, the "Bag of words", binary TF or TF-IDF and rule based system are commonly used [35]. .Unfortunately classification of documents using rule based system has proven to be ineffective as research papers that contain words like references or bibliography will be wrongly classified as curriculum vitae as is the situation with Cite Seer [36] In additional ,power point slides will also be classified as research papers whenever they contain words like references or bibliography and a research paper will not be classified whenever they don't contain any of the two words.

The famous Bag of words (BoW) or TD-IDF despite being used mostly in document classification, usually lack the ability to capture all the important elements of a' document because of the wide range of topics in digital libraries or types of documents. Furthermore, BoW has two major issues, firstly it has the curse of dimensionality issue as the total dimension is the vocabulary size. Secondly, BoW representation does not consider the semantic relation between words.

Digital Libraries like Cite Seer and ArnetMire usually use document topic as categories for classification of text document. For example, "Data Mining", "Artificial intelligence"," Machine Learning"[36] In contrast to the above works, the task, we looked at was the automatic classification of documents into respective collection(faculty) and also determining the document type.

Using a combination of terms and the structure of documents (that is, the tags of XML documents) as a means of classification, Chagheri et al [37] address the issue of identifying technical documentation such as user manuals and manufacturing documents that are available in an electronic format. The problem that they are attempting to solve is locating technical documentation such as this that is already in existence.

Li and Jain [29] adopted the use of most used Bag of words to classify documents into seven classes of Yahoo newsgroup dataset. The dataset comprised documents into the following classes: International, Politics, Sports, Business, Entertainment, Health and Technology. They ignored the structure of the documents and the arrangement of the

words in their feature representation. While the study employed BoW document representation scheme for feature representation and is classified documents to specified classes, this study will use Term Frequency-Inverse Document Frequent, in addition, our work not only will it classify scholarly output types but also the subject and the collection(school) of the document to which it belongs to.

Caragea et al [38] earlier used a binary classification of documents crept from the web to determine whether they were research papers or not and the method used to achieve this was only a set of structure features. This approach was an effort to replace the rule based system currently being used by Cite Seer.The limitation to this approach is the inability to classify other digital objects like for example books, slides and other formats, thus the researchers improved on their previous works and proposed a multi class classification of documents.

Chekuri and Goldwasser [39] dealt with the problem of classifing web pages into 20 different categories. Their classification process was statistical and based on term frequency analysis. While their work is similar to our work, we are dealing with the classification of scholarly output while they dealt with web page classification. Power et al [40] proposed a combination of feature extraction and classification algorithms for documents. In their work, they suggested a simple feature extraction algorithm for development centric topic which coupled with standard classifiers yields high classification accuracy. While our work is related to their work, however, our work was classifying of Scholarly output types, subject and collection classification.

Chagheri el at [41] proposed a method document classification which aimed at classification of technical documents. They propose a method which makes use of structure elements to create a vector to classify documents into either a user manuals and product specification. While the study uses structure elements of the document, the focus of this study was aimed at classifying of Scholarly output types, subject and collection classification. Closer to our work is the work done by Caragea el at [35] which looked at the classification of text documents crawled from the web. They propose the use of structured features aimed at classifying of documents into books, slides, thesis and resume/CV. Although, there are many studies done on document classification, most of them have been classification of web page classification.

Konstantions and Kalliris [42] dealt with the multi-label classification problem of automatic detection of emotions in music, this involved the predicting multiple emotional labels for a given track. Their goal was to automatically classify a music track into multiple emotions, such as happy, sad, calm and energetic based on the acoustic feature of the track.

Runzhi et al [43] used multi-label classification to deal with the problem of multi-disease risk prediction [15].They constructed a model for prediction of multi-diseases risk relying on the big physical examination data. They acknowledged that in medical diagnosis, a symptom may be associated with various disease types.

Chalkidis et al [44] apply Extreme Multi-Label Text Classification (XMTC) in the legal domain [22] They employ neural classifiers that outperform the current multi-label state-of-the-art methods, which employ label-wise attention.

Boutell et al [45] focused on video and photography analysis .In semantic scene classification,a picture can be associated to more than on conceptual class such as a sunset and beaches at the same time. In contrast to the above works, our research will not only be classifying digital objects submitted by researchers but will include also classes that the previous works did not include as far as we know.

2.7 Types of Techniques used for ingesting digital objects into the repository

2.7.1 Batch Ingestion

This method entails uploading several digital assets in bulk to the repository. When an organization has a lot of digital items to upload, such a library of research papers or conference materials, this strategy might be helpful.

2.7.2 Manual Ingestion

This method involves uploading each digital object individually to the repository. When an institution has a limited number of digital objects or when it is essential to meticulously curate the metadata associated with each object, this strategy is useful.

2.7.3 Harvesting

With this method, digital items are automatically gathered from various external sources, such as other repositories, databases, or websites. When an institution wishes to compile material from a variety of sources into a single repository, this technique is helpful since it allows them to do so.

Automated Ingestion

This technique involves using automated tools to extract digital objects and metadata from other systems or databases. This approach is useful when an institution wants to streamline the ingestion process and reduce the amount of manual labor required.

2.8 Challenges Associated with Ingestion of Digital Objects

While UNZA has made significant efforts in depositing Objects into its IR, there are challenges that the institute is struggling with the current method its uses for tagging of the Digital Objects. Prior work [46] identified lack of use of controlled vocabulary sets as being one of the problems associated with ingestion of repository Digital

Objects. In addition, specific types of content, such as ETDs are usually deposited with missing important metadata elements associated with metadata schemes used for tagging ETDs. Incidentally, the problem associated with the quality of metadata transcends the UNZA repository; for instance Suleman [47] has highlighted the lack of use of ETD-ms. Apparently one of the critical success factors of every IR is proper tagging of metadata to the respective Digital Objects. This is attributed to the fact that metadata defines the description of the digital objects stored in the IR.The other challenge that UNZA is facing, is the issue of wrong classification of digital objects into collections and communities.

The researcher observed that the problem of having some digital objects missing critical metadata is as a result of the manual technique method which UNZA is using. Dobreva [6] observed that manual tagging of metadata affects the quality metadata because it is actioned by human being who sometimes turn to be physical or emotionally conditioned. They further observed that manual tagging is intensive and expensive. Most importantly manual tagging is prone to a lot of errors.

Manual tagging is time consuming, during the interview with the IR Manager, the researcher established that part of the process the IR Administrator goes through when classify the digital objects was reading of the abstract of each and every digital objects, and each abstract of the digital objects has maximum of 500 words. Thus, manual tagging of digital objects is not only time consuming but it's also hectic and exhausting when one has alots of documents to read. The follow are some of main challenges and gas associated with manual deposit techniques.

1. Human error: Manual deposit is prone to human error, such as incorrect metadata entry, file format errors, and inconsistent data entry. This can lead to inconsistent quality and accuracy of the metadata and digital objects, which can affect the searchability and usability of the repository

2. Duplication and inconsistency: Manual deposit can result in duplication and inconsistency of digital objects and metadata. This can happen when the same digital object is deposited multiple times, or when the same metadata is entered differently for different digital objects. This can lead to confusion and difficulty in managing the repository.

3. Time and resource-intensive: Manual deposit requires significant time and resources from both repository staff and users. This can lead to delays in the deposit process and can affect the overall efficiency of the repository.

4. Limited scalability: Manual deposit is not scalable for large-scale ingest of digital objects. It can be difficult to manage large numbers of digital objects and metadata using manual processes, which can limit the size and scope of the repository.

5. Lack of standardization: Manual deposit can result in a lack of standardization in the metadata and digital objects deposited in the repository. This can make it difficult to search and discover digital objects, and can affect the interoperability of the repository with other systems.

While manual deposit can be a useful technique for small-scale ingest of digital objects, it has several challenges and gaps that can limit its effectiveness and scalability for larger-scale ingest. It is therefore important for repositories to consider other techniques, such as automatic deposit and APIs, to ensure efficient and effective ingest of digital objects.

2.8.1 Summary

This chapter was based on a review of various literature that has been published on text document classification, technologies used in machine learning. The chapter looked at data mining technology and related works in the subject area and gives a brief reflection of the gaps that were identified in the literature that was revealed in line with the problem domain of this research. Lastly, the chapter reviews some related systems that have been developed and implemented for text document classification.

Chapter 3

Methodology

3.1 Introduction

This chapter describes the methodologies that were adopted to carry out this research. It deals with the methods and techniques that were used in order to realize the objectives, or rather the purpose, of the study. It further explains the plan and design of the experiments that were used to answer the research questions. The standard approach of the CRISP-DM process was used because we perceived this to be a pure supervised machine learning problem. The chapter deals with the following items: research purpose, research approach, research design, business understanding, data understanding, data collection and data preparation, model implementation, evaluation, deployment, ethical consideration, and limitation.

3.1.1 Research Purpose

The principle aim of the research was to develop and implement classification models for automatic classifying of digital objects into the IR.In particular ,three classification models depicted in table 3.1 were implemented. Prior work done by Phiri [46] established there was poor scholarly output in UNZA IR and one of the contributing fact was the manual methods used by Librarians to ingest digital objects into the IR.

TABLE 3.1: Metadata Classification Models

Aspect	*Metadata*	*Classification*
Publication	Structural	Multi-class
Collection	Description	Multi-class
Subject	Description	Multi-label

3.1.2 Research Approach

Feilzer [48] argues that pragmatic approach provides diverse types of data which provides the best understanding of the research problem. The pragmatic research approach typically involves a mixed-methods research design that combines quantitative and qualitative research methods, depending on the research questions and objectives. This approach emphasizes the use of multiple sources of data, including interviews, surveys, observation, and secondary data sources, to triangulate findings and ensure validity and reliability. Dudovskiy [49] suggests that either or both observable phenomena and subjective meanings can provide acceptable knowledge that is dependent upon the research question. Johnson and Onwuegbuzie [49] present mixed methods research as complementary to traditional qualitative and quantitative research, and pragmatism as offering an attractive philosophical partner for mixed methods research. They briefly reviewed the paradigm "wars" and incompatibility thesis, and they established some commonalities between quantitative and qualitative research. Creswell [50] proposed that researchers collect or analyse not only numerical data for quantitative research, but also narrative data for qualitative research in to address the research question(s). The mixed methods approach to research is regarded as an extension rather than a replacement for quantitative and qualitative approaches.

In the quest to find the solution to the research problem, the study employed a pragmatic research approach because the research involved mixing data collection methods and data analysis procedures within the research process. For this reason, data collection involved both quantitative and qualitative however distinct design was used respectively. This method approach provides a clear clarity of the research problem than either approach alone [43]. This approach is better than mono methods because to begin with, it has the ability to answer questions that other approaches fail. Furthermore, they provide stronger inferences through depth and breadth in answering complex phenomena and finally, they provide the opportunity through divergent findings for an expression of different viewpoint.

3.1.3 Research Design

The study used a case research study design. A case study design is an empirical inquiry that investigates a contemporary phenomenon with real-life context, when the boundaries between phenomenon and context are not clearly evident and in which multiple source of evidence are used. The case study has been deemed to be the most appropriate for this research as it applies a variety of methods and depends on a variety of source to investigate a problem i.e. interviews and questionnaires [50].

3.2 CRoss Industry Standard Process for Data Mining (CRISP-DM)

The research was carried out by taking advantage of the supervised machine learning approach because we viewed this as a pure classification problem. We therefore used the standard approach of the CRIS-DM [13] process which is a structured mining planning methodology as it is a well-established data mining model. The researcher followed through all the six stages of the model which were utilized as follows;

3.2.1 Business understanding

The key focus of the study was to implement classifications model for automatically classifying IR digital objects using supervised machine learning. In order to archive that, part of the work carried out aimed at analyzing how the digital objects were organized into the IR. Sub-section 4.1.1 briefly explains how the digital objects are organized into the IR.

3.2.2 Data understanding:

Digital Objects with their associated metadata and bitstreams were harvested and analyzed to gain an overview of all the elements of the digital objects.

3.2.3 Data preparation

The data preparation phase covered all the activities in the transformation and cleaning of the data to make it fit to use in the modeling phase. The missing values, noise and outliers present in the data identified during data understanding phase were removed.

3.2.4 Modelling

The main features stated in sub-section 3.1.1 were identified and used into the model implement phase.

3.2.5 Evaluation

Standard Machine estimators were used to access the effectiveness of the classification models, furthermore feature combination was evaluated to ascertain their effectiveness as explained in section 3.9

3.2.6 Deployment

Three classification models were implemented with one of them having an Application Programming Interface developed to facilitate the integration of the models with third-party tools and services.

3.3 Logical Data Model – Organization of Digital Objects in the IR

The first objective of the research was to analyze how the digital objects were organized into the repository. The main purpose of analyzing the objects was to help the researcher with an overview of how the digital object were stored into the IR, thus interviews were first used to gain the insight and purposive sampling was adopted for the selection of the participants. Purposive sampling is most preferred because it helps qualitative researchers to gain an in depth of the Phenomenon by only select people that are knowledge able in given field [51].

This was also influenced by the fact we were just looking at the subject experts hence usage of purpose sampling method. The selected approach was preferred as the research required information from the people who were responsible for tagging and ingestion of digital objects in the IR. This implies that by choosing participants purposively, it is possible to get people who have experience in the phenomenon and have hand information pertinent to the study.

3.4 Scheduled Interviews

To help answer research question 1, which was aimed at analyzing how digital objects were organized in the UNZAIR. The study used scheduled interview and a Samsung A10 phone was used to record the proceedings in order to ensure that all data was captured. The schedule was used by the researcher, who filled with actual response received during the interview. Furthermore, interview schedule was used because it increases the likehood of collecting accurate information or data, thus, allowing researchers to get more information, increase's the rate and amount of responses.

3.5 Repository Analysis

The researcher looked at the UNZA IR by navigating through the communities, collections, and individual items. This helped the researcher connect the information the researcher got from the interview about how the digital objects were organized and understand the full scope of the problem. By using the UNZA IR to look at the communities, collections, and individual items, the researcher found that some objects had missing metadata and others were wrongly categorized

3.6 Digital Objects Tagging Process

The second objective of the research was to analyse how digital objects were tagged prior to ingestion into the IR. To address objective 2, the researcher had follow up interviews with the IR Manager and the Administrator and went through the same process as explained in sub-section 3.2. The information gathered during the interview provided the researcher with all ingestion procedures that IR Manager and Administrator go through when tagging and ingesting the digital objects into the IR. In addition to the interviews, the open Archive Protocol for Metadata Harvesting was used to harvest metadata and their respective bitstreams from UNZA IR and external repositories using a python script and LibreCat Catmandu data processing toolkit [52]. External repositories are digital archives or databases that are not directly affiliated with a particular institution or organization. These repositories collect and provide access to scholarly and scientific works from a variety of sources, including academic institutions, research organizations, and individual researchers. External repositories are important resources for researchers, as they provide access to a wide range of research materials across different disciplines and geographic regions. They also help to promote open access to research, which can increase the visibility and impact of scholarly work by making it more widely available to researchers and the public. Many external repositories are free to use and provide advanced search and filtering tools to help users find relevant content quickly and easily. Examples of external repositories that was used to harvest metadata and include arXiv [53], SUNScholar (Stellenbosch University), OpenUCT(University of Cape Town), UPSpace (University of Pretoria), and UnisaIR (University of South Africa). The harvesting was done using Dublin core since our targeted digital Libraries were open source that used international standard and implemented interoperability protocols for effective storage and retrieval.

3.6.1 Harvesting of Digital Objects from Internal Repository

A total of 5500 Dublin Core encoded metadata [54] were extracted from the UNZA IR using the python script built by the researcher while adhering to the OAI-PMH standard [55].In spite of the fact that Dublin Core employs a set of fifteen components, the following were used:

1. Identifier: Used to uniquely identify the digital and link the various data sets.
2. Title: Used to extract text features used to build the classification model.

3. Description: This is typically used to encode the digital objects abstract and was thus used to extract text features used to build the classification model.

4. Type: Used for verifying the type of ETD labels for distinguishing Master's dissertations and Doctoral theses.

5. SetSpec: Used for labelling the Document in order to distinguish the faculty where the ETD was prepared from.

6. Collection – Used as a label for determining the collection to which the digital object belonged to.

3.6.2 Harvesting of Digital Objects from External Repositories

After analyzing the digital objects that were harvested, the researcher established that some of the digital objects were scanned copies and therefore it was difficult to extract data from it. This resulted into having not enough data set, for this reason, the researcher opted to harvest digital objects from external repositories. And four universities were selected from South Africa and the criteria used to select the universities was based on their scholarly output. The researcher targeted the top four universities with the highest scholarly output. Stellenbosh University, University of Cape Town, University of Pretoria and University of South Africa were identified. The researcher also harvested additional metadata from ArXiv repository because it was observed that the dataset that was created thus far did not have sufficient data for to build the subject classification model. arXiv is a repository of electronic preprints (known as e-prints) of scientific papers in the fields of mathematics, physics, astronomy, computer science, quantitative biology, statistics, and quantitative finance, which can be accessed online. The researcher identified arXiv as it uses the ACM classification system. The ACM Computer Classification System is often used to tag digital assets indexed in the ACM Digital Library4 (CCS) [56]. Data investigation was performed using Python code in association with Juptyer Note book and including the following -:

1. Statistical Analysis: Basic statistics such distribution was computed for analysis.

2. Missing value analysis: Count and percentage count of the missing values of the target and label was calculated.
3. Exploratory data analysis: Frequency distribution plots of the categories of the document types was counted.
4. Outlier analysis: Outlier analysis was performed to find out the values lying out of range.

3.7 Datasets

For comprehensive evaluation of proposed classification models, the selection of the datasets was a very critical step. Part of our study was to develop and implement three different types of classification models. The researcher, carefully selected three diversified datasets. The first one contained thesis and dissertation, conference papers, books, book chapters and Journal articles which were harvested from UNZA. The second datasets comprised of thesis and dissertations, conference papers, books, book chapters and Journal articles from UNZA, University of Pretoria, University of South Africa, University of Cape Town, and University of Stellenbosch. Finally, the third dataset contained digital objects harvested from arXiv.

3.7.1 First Dataset –UNZA Dataset

Thus, the first dataset comprised of thesis and dissertation, Conference, papers, books, book chapters and Journal articles harvested from the University of Zambia. The dataset contained 5,500 digital objects. After analyzing it, it was observed that 3,300 were thesis and dissertation and since prior work done by Phiri [46] already dealt with the problem of electronic thesis and dissertation, therefore the researcher removed all these digital objects as this study was a buildup of the work done by Phiri [46]. It was observed further that of the remaining 2,200 digital objects, 1,306 digital objects were scanned pdf from which we could not extract metadata, therefore , the objects were filtered out leaving only 894 digital objects which were later used for building the collection model. Table 3.2 depicts the final dataset distribution which was used to build the model.

TABLE 3.2: Distribution of the first dataset

Collection	*TotalNumberofDigitalObjects*
Agricultural Sciences	286
Education	310
Engineering	103
Humanities and Social Sciences	499
Institute of Distance Education	49
Law	325
Library	50
Medicine	475
Mines	49
Natural Sciences	381
University Collection	23
Veterinary Medicine	118

3.7.2 Second Dataset

The second dataset comprised of digital objects from University of Zambia, University of Pretoria, University of South Africa, University of Cape Town, University of Stellenbosch and University of Johannesburg. Table 3.3 delineate the summary of the total number of digital objects in the second dataset.

TABLE 3.3: Distribution of the first dataset

PublicationType	*TotalNumberofDigitalObjects*
Books	226
Book Chapters	227
Journal Articles	228
Conference Papers	198

3.7.3 Third dataset

The third and final dataset contained digital object harvested from arXiv, a total number of 320801 object were harvested. The digital objects contained in the dataset belong to 438 subjects. ArXiv [53] is a freely accessed highly computerized academic papers library which was initially started in 1991 and currently being maintained by Cornell University Library. It has a collection of over one millions of articles in physics, mathematics, statistics, computer science, etc. For the purposes for this experiments we only harvested 320,801 digital objects belonging to computer science only.

3.7.4 Model Implementation

The main aim of the study was to implement three classification models Collection, Publication and Subject Type Model using five main supervised machine learning algorithms for classification problem namely Logistic Regression, Support Vector Machine, Random Forest, Nearest Neighbor, Naïve Bayes and Boost.

3.8 Feature Extraction

Generally texts and documents are unstructured datasets. However, to train a model on text data, some preprocessing is required to make the data suitable for training the model. Machine learning algorithms usually take numeric data, therefore the text data have to be transformed into feature vectors [57]. The main purpose of preprocessing is to make clear the border of the respective language structure and to eliminate as much as possible the language dependent factors, tokenization, stop words removal and stemming.

Feature extraction is the process of selecting and transforming raw data into a set of meaningful features that can be used as input to machine learning algorithms or other data analysis techniques. The goal of feature extraction is to identify and extract the most important and relevant information from the data that can help to solve a particular problem [58] [59] In machine learning, features are typically represented as numerical values, and feature extraction is a crucial step in the preprocessing of data for classification, clustering, regression, and other types of analysis. Feature extraction methods can be supervised, unsupervised, or semi-supervised, depending on whether the input data is labeled or not. Figure 3.1 depicts all the commonly steps taken for feature extraction.

3.8.1 Tokenization

Tokenization is the process of decomposing a text or sentence into individual words, phrases, symbols, or other significant parts known as tokens. Tokenization is the act of breaking down a piece of text into smaller parts that may be readily evaluated, processed, or represented as input to a machine learning system [60].

Tokenization is a critical step in natural language processing and text analysis because it converts unstructured text input into structured data that a computer can evaluate and interpret. There are several tokenization approaches available, depending on the task at hand and the type of text data being processed. Among the most frequent approaches are:

Tokenization of whitespace: This is the process of dividing text depending on spaces, tabs, or line breaks.

Word tokenization is the process of separating text into distinct words.

Sentence tokenization is the process of separating text into separate sentences.

Regular expression tokenization is the process of splitting text using regular expressions based on certain patterns or rules.

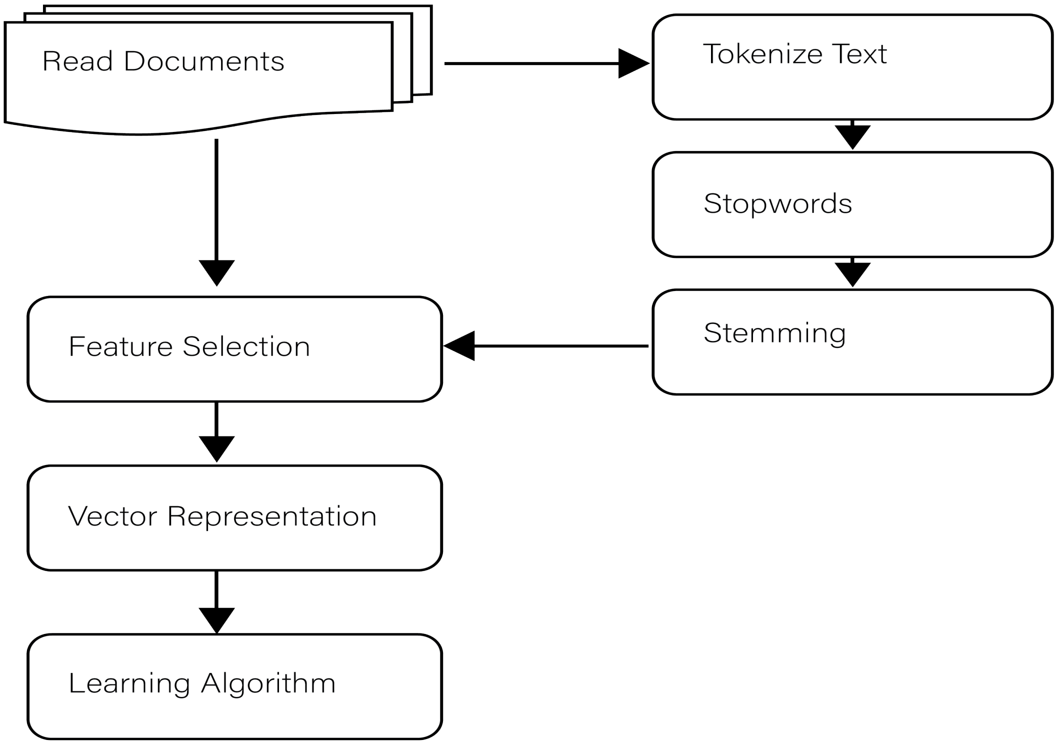


FIGURE 3.1: Document Classification Process

3.8.2 Stop word removal

When working with text classification method, removal of stop words is a common approach aimed at reducing noise in the data.Stop word removal is a frequent preprocessing step in natural language processing that entails deleting words deemed uninformative or unrelated to the study. Stop words are often high-frequency terms like "the", "and", "a", "in", and "of" that do not convey much significance on their own and are typically eliminated to minimize the dimensionality of the data and enhance the accuracy of subsequent models [61].

Typically, the method of eliminating stop words entails compiling a list of stop words and then deleting them from text data. The list of stop words can be modified dependent on the job and domain at hand. In a sentiment analysis work, for instance, negation words such as "not" and "never" may remain in the text input, but in a topic modeling assignment, domain-specific phrases may be deleted as stop words.To aid with the elimination of the stop words from all metadata parameters of the dataset, inbuilt NLTK library was because it contains list of stop words.

3.8.3 Stemming

Stemming is the process of reducing a word to its base or root form, known as a stem, by removing any suffixes or prefixes. The resulting stem may not necessarily be a real word, but it will represent the common morphological or grammatical structure of the original word. The goal of stemming is to reduce the number of unique words that need to be processed or stored in text analysis applications, while still preserving the meaning of the text [62]. For example, the words "jumping", "jumps", and "jumped" all have the same root word "jump", and stemming can reduce all of these words to the same stem "jump". Stemming is commonly used in natural language processing and text mining applications, such as search engines, document classification, and topic modeling. Some popular stemming algorithms include the Porter Stemming Algorithm, the Snowball Stemmer, and the Lancaster Stemmer.

Many Machine learning algorithms often take numeric vector as the input, but before performing any manipulation on the text, they are usually need to convert the respective document into a numeric vector. This is one of major fundamental problems in data mining which thrives to numerically represent the unstructured text document to make them mathematically computable. In this work Term Frequent (FM) and Term Frequency and Inverse Document Frequency (TFIDF) was used.

3.8.4 Term Frequent

The simplest characteristics of the text document are the words contained within it. To create numerical features from these words, word frequencies can be calculated by counting the number of occurrences of a word in the document. Combining the values for different word features in a vector gives us a feature vector that characterizes the document. Each word in the dictionary has its own place on the feature vector. This resulting feature vector is an example of sparse data, most words from the lexicon will not appear in a specific document and even if they appear, usually not multiple times. The TF does not consider the order of the words, semantics and context of a words.

3.8.5 Term Frequency and Inverse Document Frequency

While TF are a useful way to represent documents, they do not provide any information about the usage of a word in the full corpus. In Machine Learning it is often useful to know whether a particular word is a common word or used relatively much in a specific document. TF does not provide information in this regard. Applying term weighting mechanism helps to represent these types of information. TFIDF defines the importance of a term in a document [63] [64]. It contains two concepts Term and Frequent and It measures how frequently a term occurs in a document while IDF measure the importance of the word.

3.8.6 Using N Grams Features

N-gram features are a type of feature extraction technique used in natural language processing and text analysis. An n-gram is a contiguous sequence of n items from a given sample of text, such as words, letters, or even syllables. By analyzing the frequency of n-gram sequences in a text, n-gram features can provide valuable information about the structure, style, and meaning of the text [65].

N-gram features can be used in various text analysis tasks, such as language modeling, text classification, sentiment analysis, and topic modeling. For example, in language modeling, n-grams can be used to predict the likelihood of a sequence of words occurring in a given context. In text classification, n-grams can be used as features to represent the distribution of specific word or character sequences in the text. The choice of the value of n in n-grams depends on the specific task and the characteristics of the text data. Unigrams (n=1) are single words, bigrams (n=2) are pairs of words, trigrams (n=3) are three words, and so on. The higher the value of n, the more context and structure of the text is captured in the n-gram feature.

3.8.7 Feature reduction and section

Feature selection is a key phrase in data processing which aims at reducing the feature space and improve the accuracy of the classifier. Feature selection is the process of selecting relevant features which help in model construction. These techniques are required to remove unwanted data, cut the long text which provide noise to the data and irrelevant data that does not aid in improving the accuracy of the model [66]. In most cases text document often have a lot of features , it is usually important to reduce the feature space. The essence of this is to make the process of computationally easier and aids in avoiding over fitting the model because the options of the model are more limited. In document classification, often a lot of words can be removed without affecting the performance of the classifier. The stop word removal discussed earlier is a simplier way to remove some features. However, stop word removal mostly discards small portion of features and keeping only the most imperative ones. There are mainly three types of feature selection methods namely filter, wrapper and hybrid [67]. Filter method makes selection rely on data characteristics while wrapper methods uses some mining algorithm to determine the relevance instead of the immediate proprieties of the data and hybrids combine the two types. Thus , the approach the researcher used tf-idf to the reduce the number of features. In addition to feature selection, feature reduction is also a technique used in machine learning to reduce the dimensionality of the feature space. Feature reduction involves transforming the original feature space into a lower-dimensional space while preserving the most important information or structure of the data. Techniques for feature reduction include principal component analysis (PCA), linear discriminant analysis (LDA), and t-distributed stochastic neighbor embedding (t-SNE) [68].

3.9 Evaluation Metrics

The objective for evaluating with different machine learning methods is to calculate an error for a method in order to compare the methods results with other methods in a given task. The general idea behind the evaluation is to split the data into training, validation, and test set. The method is taught using the training set, validated against a validation set to get the expected error, and finally use test set to see how the method would perform in a real situation [69]. The existence of validation set is crucial due to the fact that comparing classifiers in general is useless because the performance of machine learning algorithms relies heavily on the data used. However, there are different indicators of performance other than error rates due to the fact that each task has its own objective As earlier mentioned, the main focus of the research is to implement three different types of classification model, therefore to analyze and evaluate the performance of the models we used different method as discussed below.

3.9.1 Evaluation metrics for Publication and Collection Model

To evaluate the performance of the publication and collection model, the researcher used confusion matrix ,accuracy , precision, recall and F-Measure.

3.9.2 Confusion Matrix

A confusion matrix, also called a contingency table, is a visualization of the performance of a supervised learning method. A problem with n classes, requires a confusion matrix of size n × n with the rows representing the specific actual class and the columns representing the classifiers predicted class. Basically, the confusion matrix is a table that shows the numbers of the correctly and incorrectly labeled examples [70]. The size of the matrix is the number of classes multiply by the number of classes. In a confusion matrix, TP (true positive) is the number of positives correctly identified, TN (true negative) is the number of negatives correctly identified, FP (false positive) is the number of negatives incorrectly identified as positive, and FN (false negative) is the number of positives incorrectly identified as negatives. The diagonal elements in the matrix are the instance counts of the number of correct classifications for a respective group and the off diagonal elements represent the misclassified instances. Confusion matrix visualizes different types of errors made by the classifier.

3.9.3 Classification Accuracy

This is one of the most important measure of a classifier. This measure determines the percentage of the correctly classified instances [71] The formula to calculate classification accuracy is: Accuracy = (Number of correctly classified instances / Total number of instances) x 100

3.9.4 Precision

Precision measures in form of percentage, only the correctly identified instances to a given class. It is defined using the following formula. Precision =ratio of the number of documents retrieved that are relevant to the total number of documents that are retrieved [72] The formula for precision is: Precision = True positives / (True positives + False positives)

3.9.5 Recall

Recall metrics will on the hand helped the researcher to measure the ability of models to find all correct instances per class. It is defined using the following formula :- Recall =ratio of the number of documents retrieved that are relevant to the total number of documents that are relevant [48]. The formula for recall is: Recall = True positives / (True positives + False negatives)

3.9.6 F1 Score

The F1 score is a metric used to evaluate the performance of a classification model, particularly in binary classification problems. It provides a balance between precision and recall by computing the harmonic mean of these two metrics. The F1 score ranges from 0 to 1, with a higher score indicating better performance. The formula for F1 score is: F1 score = 2 * (precision * recall) / (precision + recall) where precision and recall are the precision and recall metrics, respectively. In order to summarize the model's performance into a single metric, F1-score was used to all the model. This was archived by combination the precision and recall into a single metric using the harmonic mean [67].

3.9.7 Evaluation metrics for Subject Classification

Due to the different nature of Multi-Label Classification and standard classification problems, differing evaluation metrics have been proposed in the literature to capture Multi-Label Classification performance , thus, for the subject classification model , hamming loss ,Jaccard Distance and F1-score were the metrics used to evaluate the performance of the model.

3.9.8 Jaccard Distance

Jaccard distance evaluates the dissimilarity between two pairs by dividing the difference of the union and the intersection of two pairs with the size of the union [73].

3.9.9 Hamming Loss

Hamming loss is the fractional expression of the Hamming distance mainly used to demonstrate the distance between two arbitrary strings.The distance is usually accessed by the number of steps in a string edititing process or the number of single alternation required to transform one string into another[74].For instance , the Hamming distance between the Hamming and Hamster is 4.Hamming loss expresses the the error betweeen two strings as a ration of hamming distance to length of the expected string.

In addition to hamming loss and Jaccard distance , other standard evaluation metrics maybe adapted for use in multi-label classification including F-score.

3.10 Experiment Design

The main aim of this study was to build three models namely the Collection, Publication and Subject Type Model. Therefore, a number of experiments were performed.

3.10.1 Collection Type Classifier

A multi-class classifier was implemented by utilizing the machine learning classification, with the model implemented using the data harvested from the UNZA IR.

1. Data Preparation: As explained in section 3.7.1 , the first dataset which comprised of the data harvested from UNZA'S IR was used during experimentation of the first mutli-class model (Collection Type Classifier).The data attributes of the UNZA dataset were utilized as follows.

   - Identifier – A primary key used for uniquely identifying each record of the digital object.
   - Title – The digital objects' document title was used to extract text input features.
   - Description – The digital objects' abstract was used to extract text input features.
   - Collection – Used as a label for determining the collection to which the digital object belonged to.

2. Model Implementation: The input features for the model were extracted from the digital object publication tiles and abstract which later were transformed using CountVectorizer and TFIDFVectorizer. The model was implemented using the scikit-multilearn Python library [75] For the classification algorithms, Logistic Regression, Random Forest, Multinomial, Stochastic Gradient, Descent, Gaussian Naive Bayes and Support Vector Machine were used. Section 4.1.4 discusses experimental results which were obtained to analyze the effectiveness of the modal features and transformation strategies used.

3. Experiment Design: All the experiment were conducted using a standalone HP running on Windows 10 Pro 64-bit Intel® Core (TM) i7-10510 CPU@1.80 GHZ (8 CPUs)- 2.3GH with 8GB memory running Windows 10.

   Training and testing datasets were created using the holdout method built within the scikit-multilearn Python library, with 70 % of each dataset used for training and the remaining 30 % for testing. The performance of the single label classification were measured using the evaluation metrics discussed in section 3.9.1 – Precision , Recall, F1-measure and Accurancy. Combinations of various factors during the experiments were tried in order to access which factors yielded better results. The experiments involved the following aspects :

   - Input features—Title, Abstract and a combination of the two: Title+Abstract.
   - Text transformation techniques—CounterVectorizer and TFIDFVectorizer—used on input features and their corresponding parameters.
   - Estimators - Logistic Regression, Random Forest, Multinomial, Stochastic Gradient, Descent, Gaussian Naive Bayes and Support Vector Machine.

3.10.2 Publication Type Classifier

A multi-class classifier was implemented by utilizing the machine learning classification, with the model implemented using the data harvested from the UNZA IR and external repository.

1. Data Preparation: As explained in section 3.7.2, the second dataset which comprised of the data harvested from UNZA'S IR and external repositories was used during experimentation of the first mutli-class model (Publication Type Classifier).The data attributes of the combined dataset was utilized as follows.

   - Identifier – A primary key used for uniquely identifying each record of the digital object.
   - Description –For Journal Articles, Book chapter and Conference Paper, the researcher used the textual content of the first page while for the books

the researcher used the text content of the first sixteen pages were used as input features.

- Type – Used as a label for determining the publication type of the digital object.

2. Model Implementation: The input features for the model were extracted from the digital object publication first page for the Journal Articles, Book Chapter and Conference Paper while for the books the researcher extracted textual context from the first sixteen pages of digital objects and which later were transformed using CountVectorizer and TFIDFVectorizer. The model was implemented using the scikit-multilearn Python library [75]. For the classification algorithms, the researcher used Logistic Regression, Random Forest, Multinomial, Stochastic Gradient, Descent, Gaussian Naive Bayes and Support Vector Machine. Section 4.2.1 discusses experimental results which were obtained to analyze the effectiveness of the modal features and transformation strategies used.

3. Experiment Design: All the experiment were conducted using a standalone HP running on Windows 10 Pro 64-bit Intel® Core (TM) i7-10510 CPU@1.80 GHZ (8 CPUs)- 2.3GH with 8GB memory running Windows 10.

   Training and testing datasets were created using the holdout method built within the scikit-multilearn Python library, with 70 % of each dataset used for training and the remaining 30 % for testing. The performance of the single label classification was measured using the evaluation metrics discussed in section 3.9.1 – Precision , Recall, F1-measure and Accurancy. Combinations of various factors during the experiments were tried in order to access which factors yielded better results. The experiments involved the following aspects :

   - Input features—Title, Abstract and a combination of the two: Title+Abstract.
   - Text transformation techniques—CounterVectorizer and TFIDFVectorizer—used on input features and their corresponding parameters.
   - Estimators - Logistic Regression, Random Forest, Multinomial, Stochastic Gradient, Descent, Gaussian Naive Bayes and Support Vector Machine.

3.10.3 Multi-Label Subject Classifier

A multi-label classifier was implemented by taking advantage of machine learning classification, with the model implemented using data from an external repository.

1. Data Preparation: As mentioned in Section 3.7.3, the arXiv dataset was used during experimentation of the multi-label classification model. The data attributes of the arXiv dataset were used as follows:

    - Identifier—A primary key for uniquely identifying each of the arXiv digital objects harvested.
    - Title—The arXiv digital object publication title, used to extract text input features.
    - Description—The arXiv digital object abstract, used to extract text input features.
    - Subject—The arXiv-specific digital object subjects [76] and 1998 ACM Computing Classification System (CCS) concepts [77], used as labels.

2. Model Implementation

    The input features used for the model were extracted from the digital objects publication tiles as well as the abstract and then later transformed using CountVectorizer and TFIDFVectorizer. The model was implemented using the scikit-multilearn Python library [78]. Binary Relevance and Classifier Chains approaches to multilabel classification were used, in association with estimators - Random Forest, Naive Bayes (Multinomial) and SGDClassifier which are popularly used for text classification. Section 4.2 discusses experimental results conducted to experimentally evaluate the effectiveness of the modal features and transformations used.

3. Experiment Design: All the experiment were conducted using a standalone HP running on Windows 10 Pro 64-bit Intel® Core (TM) i7-10510 CPU@1.80 GHZ (8 CPUs)- 2.3GH with 8GB memory running Windows 10.

    Training and testing datasets were created using the holdout method built within the scikit-multilearn Python library, with 70 % of each dataset used for training and the remaining 30 % for testing. The researcher measured the performance of the multi label classification during the experiment with the metrics discussed in section 4.9 – Hamming Loss, F1-measure and Jaccard Score Similaries. In order to ascertain the combination of factors that gave better results, experimentations took into account the following aspects:

- Input features—Title, Abstract and a combination of the two: Title + Abstract.
- Text transformation techniques—CounterVectorizer and TFIDFVectorizer—used on input features and their corresponding parameters.
- Multi-label classification approaches—Binary Relevance , Classifier Chains and One versus the Rest.
- Estimators—Random Forest, Naive Bayes (Multinomial) and SDGClassifier Experimentation involved measuring evaluation metrics by varying the experiment factors and aspects mention above. Section 4.2 presents and discusses the results.

3.11 Ethical Consideration

Since the study involved accessing the records of the people, the ethical guidelines of the university was implemented throughout the study. Thus, the researcher applied for ethical clearance form DRGS' Ethical committee and the committee granted the clearance. The researcher made clear from the out set that all participants were no under any obligation to take part and were free not to and anonymity was assured. Signed consent for the participants was received and permission was requested for the interviewers to be recorded.

3.12 Limitation of the study

The main limitation of the study is the its findings cannot be generalised because the ingestion and tagging process that UNZA is currently uses is not the same tagging and ingestion process that other IR administrators use for other Institutions.

3.13 Summary

In this chapter , the materials and methods that were used in the baseline study were discussed. The methodology used for the baseline study used pragmatic approach was discussed were used. Furthermore the current ingestion process for digital objects was discussed.

Chapter 4

Results and Discussion

In this chapter, presented are the results from the experiments done, to begin with , discussed is the current ingestion of digital objects, the source of that dataset and dataset used is explained and in conclusion the results of the experiments will be explained.

4.1 Situation Analysis

4.1.1 Analysis 1: Organization of Digital Objects into IR

As explained in section 3.4, interviews with two library staff –IR Manager and his assistant was conducted. They gave an overview of how digital objects content is organized into the IR. It was established that digital content in the UNZA IR are logically organized into to logical data model as illustrated in figure 4.1. Digital content in the UNZA IR is at the highest level organized into communities. These correspond to organizational bodies in an organization like schools. A community is organized into collections of logically-related materials. For example, a department might be a collection. An item is an archival atom; that is, a grouping of content and metadata that it makes sense to archive as a single unit. This may take the form of a journal article, a dataset, or perhaps a technical report together with a dataset used in experiments described by the report. Precisely what constitutes an archival atom is largely a policy-driven decision. Each item has one Dublin Core metadata record. Other metadata might be stored in an item as a serialized bitstream, but they stored Dublin Core form for every item for interoperability and ease of discovery. The Dublin Core is usually entered by IR administrators as part of the ingestion process as they ingest the content into the IR. The content of items, and any serialized metadata, are stored in bitstreams. These are organized into bundles of closely tied bitstreams. For example, an item might contain a dataset in flat text file, and a technical report in an HTML document. The dataset text file would be stored in one bundle, and the HTML files and associated image files that make up the technical report would be grouped together in bundle. Each bitstream is linked to one bitstream format. The relationships between communities, collections, items, bundles and bitstreams may all be many-many.

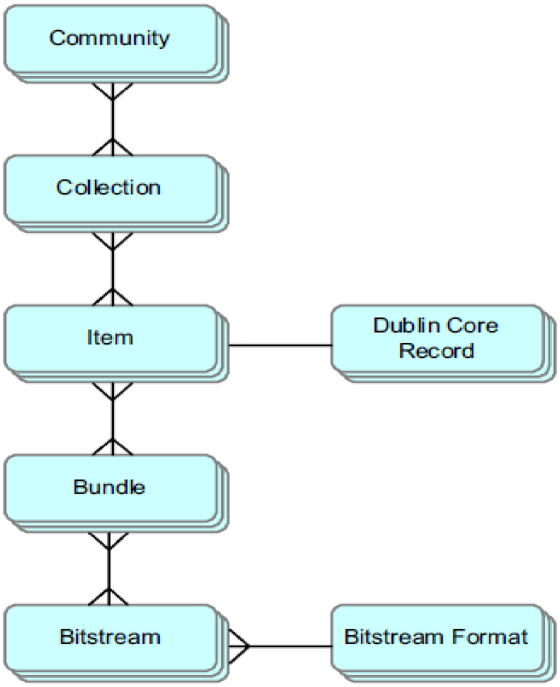
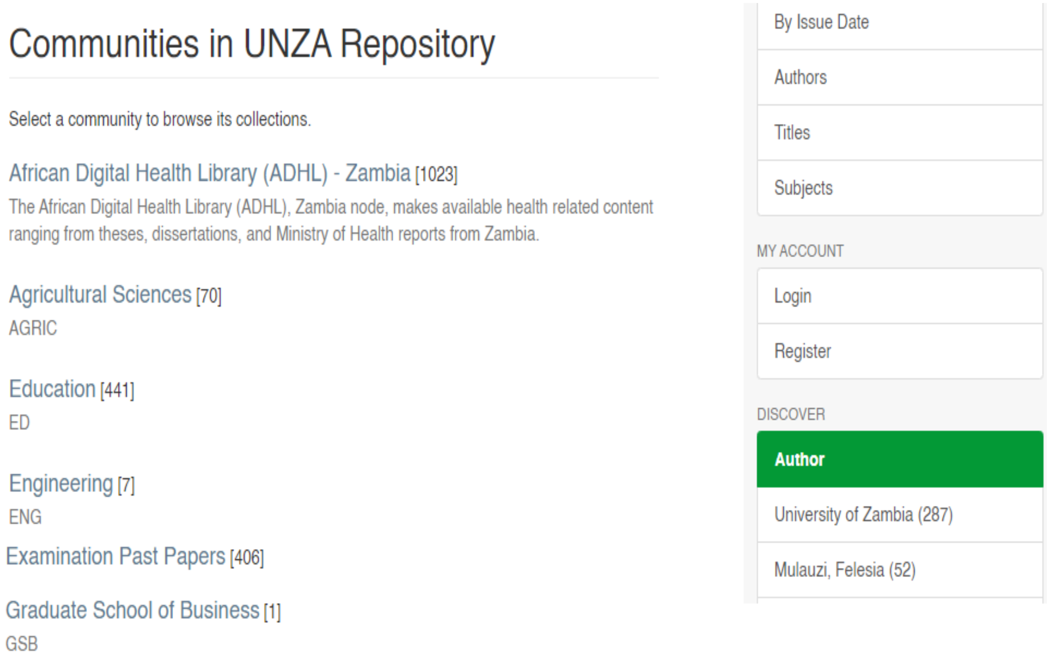


FIGURE 4.1: Hierarchical IR structure

4.1.2 Analysis 2: Ingestion Process of Digital Objects into IR

AS explained in section 3.1.5, two key staff personnel from the special collection department were interviewed in order to understand how digital objects were tagged prior to ingestion into the IR.The researcher established that the workflow of the ingestion of digital objects is categorized into two ways, one for Electronic Thesis and Dissertations(ETDs) and Non -ETDs. The non-ETDs comprised of digital objects such as journal papers, conference papers, books, book chapters, examination past papers,students report and preprints. ETDs have a distinctive workflow as depicted in figured 11 while the workflow for non-ETDs is depicted in figure 4.4.



Communities in UNZA Repository

Select a community to browse its collections.

African Digital Health Library (ADHL) - Zambia [1023]
The African Digital Health Library (ADHL), Zambia node, makes available health related content ranging from theses, dissertations, and Ministry of Health reports from Zambia.

Agricultural Sciences [70]
AGRIC

Education [441]
ED

Engineering [7]
ENG

Examination Past Papers [406]

Graduate School of Business [1]
GSB

FIGURE 4.2: UNZA's hierarchical IR structure

With the ETDs, once they have been approved (Marked) by the respective faculties/Schools, hardcopies and soft copies are send to Directorate of Research and Graduate Studies (DRGS) and in turn DRGS sent a hard and soft copies to the special collection department of the Library who are responsible for the tagging of metadata. Having done that, the soft copy of the digital copy is send to the IR department who finally upload the digital object onto the IR. On the other hand, non-ETDs, once the document is published, the document is then sent from the respective department to the special collection in the Library where the tagging of the meta data is done. Having done that, the document is send to the IR department who finally upload the document into the IR as depicted in figure 4.4.

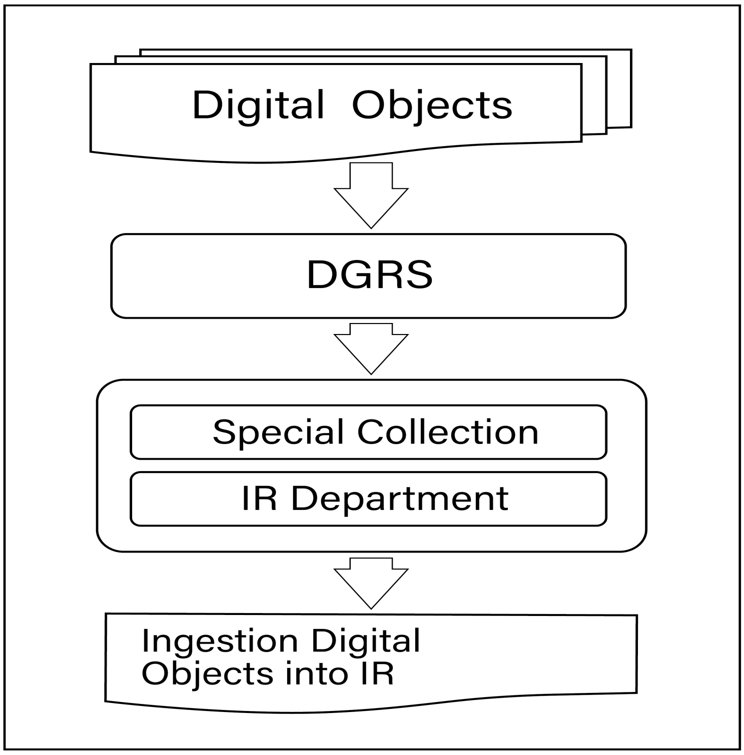


FIGURE 4.3: Ingestion Process of ETDs

4.1.3 Analysis 3: Analysis of IR

In order to understand the full content of the problem, the researcher, analyzed the digital objects in the IR by navigation through all the collections in the communities and all the digital objects in the collections. It was observed that out of the 5,500 digital objects that were stored into the IR,9 of the digital objects were wrongly misclassified as depicted in Figure 4.5, this was mainly as a results of the current manual tagging method being used. It was observed that Veterinary Medicine community had 0.1 percent of wrongly classified digital objects which apparently was the highest collection with wrong classification of digital objects. Figure 4.1.4 demonstrates the distribution of the percentage of the wrong classification of digital objects according to the respective communities.

It was observed that Veterinary Medicine community had 0.1 percent of wrongly classified digital objects which apparently was the highest collection with wrong classification of digital objects. Figure 4.1.4 demonstrates the distribution of the percentage of the wrong classification of digital objects according to the respective communities.

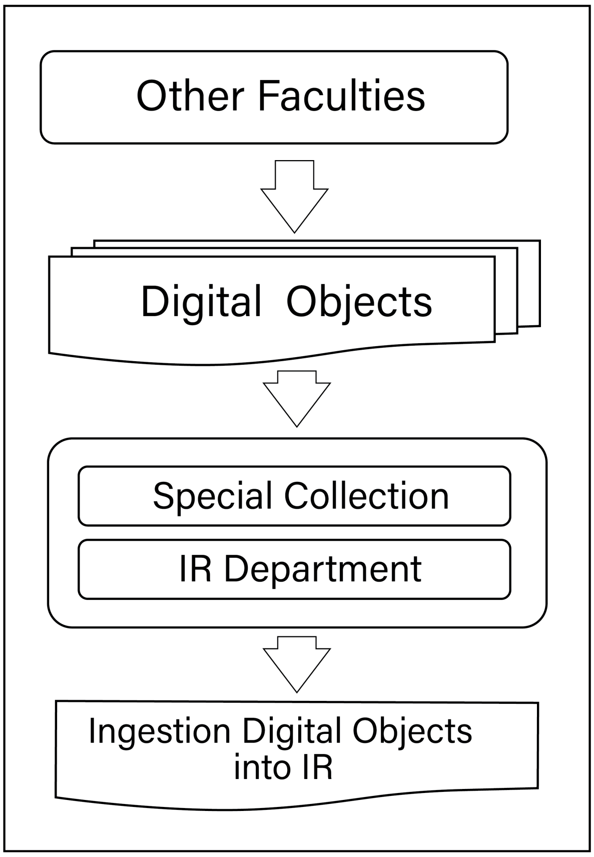


FIGURE 4.4: Ingestion Process of Non-ETDs

4.2 Model Implementation

4.2.1 Collection Type Classification

As discussed in Chapter 3, the learning algorithms implemented in the collection model classification are Stochastic Gradient Descent, Logistic Regression, Support Vector Machine, Multi-nominal, Random Forest and Decision Tree. For the respective algorithms three different features were used: Digital objects' title and Digital Objects' abstract and a combination of the title and abstract. For each input feature the researcher converted it using TfidfVectorizer and CountVectorizer.

CountVectorizer is a scikit-learn library class that converts a collection of text documents into a token count matrix. It works by tokenizing the text, or breaking it down into individual words or n-grams, and then counting the number of times each token appears in each document. The generated matrix may be fed into a machine learning algorithm to help with tasks like text categorization or clustering [78].

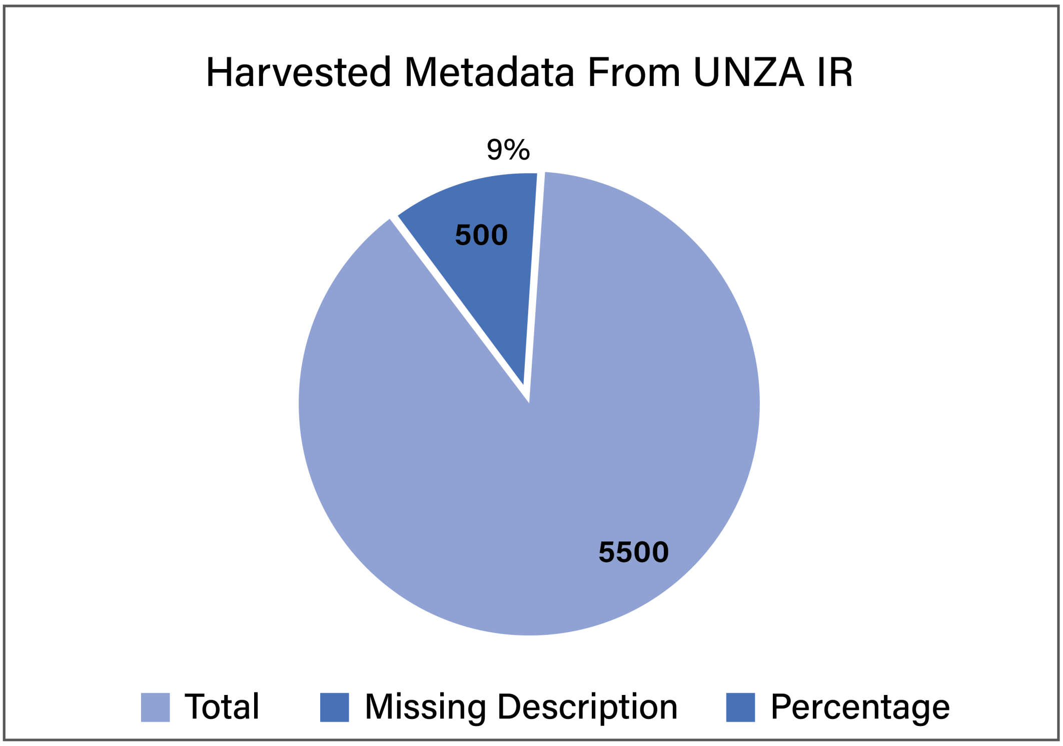


FIGURE 4.5: Missing metadata analysis in the UNZA IR

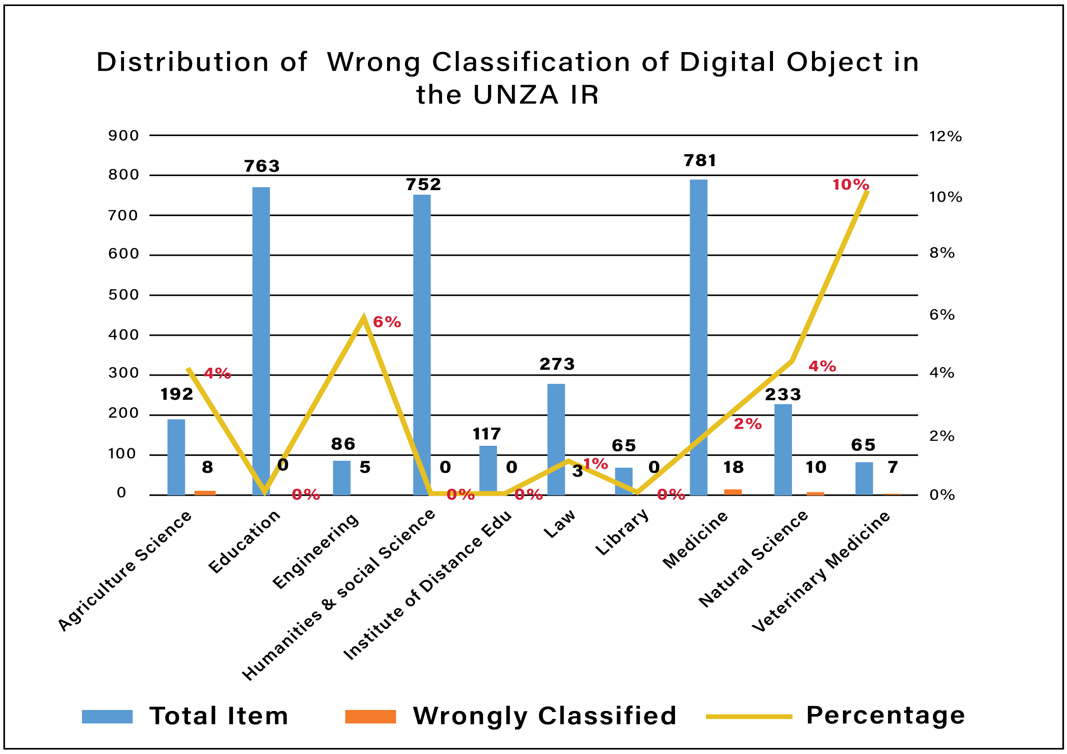


FIGURE 4.6: Distribution of wrong classification according to communities

The scikit-learn library's TfidfVectorizer class converts a collection of text documents into a matrix of term frequency-inverse document frequency (TF-IDF) characteristics. It operates by tokenizing the text and calculating the TF-IDF value for each token in each document. The TF-IDF value represents a token's relevance in a document in comparison to its importance in the corpus as a whole [79].

As anticipated, the input features converted using TFIDF-vectorizer yield better results that of countvecttorizer, this was the same in all the three input features. For each individual metadata feature, accuracy, precision, recall and f-measure was calculated for all the categories, Table 4.1 to Table 4.3 show the results obtained from the using the input feature transformed using the countvectorizer. On the contrary Table 4.4 to Table 4.6 depicts the results with the input features transformed using TDIF-Vectorizer. From the results obtained as depicted above, it was evident that the Abstract input feature obtained Accuracy Score of 0.76, Precision Score of 0.78, Recall Score of 0.67 and F1-Score of 0.71 which was better results than that of the Title feature as shown in Table 4.5. The results therefore showed that the Abstract hold a strong potential in case of single label classification.

Results for collection model using input feature transformed using Counter-Vectorizer

TABLE 4.1: Title Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.62	0.64	0.40	0.42
2	Stochastic Gradient Descent	0.67	0.70	0.56	0.60
3	Multinominal	0.58	0.52	0.34	0.35
4	Random Forest	0.62	0.67	0.50	0.55
5	Support Vector Machine	0.66	0.63	0.58	0.59
6	Decision Tree	0.47	0.48	0.43	0.44

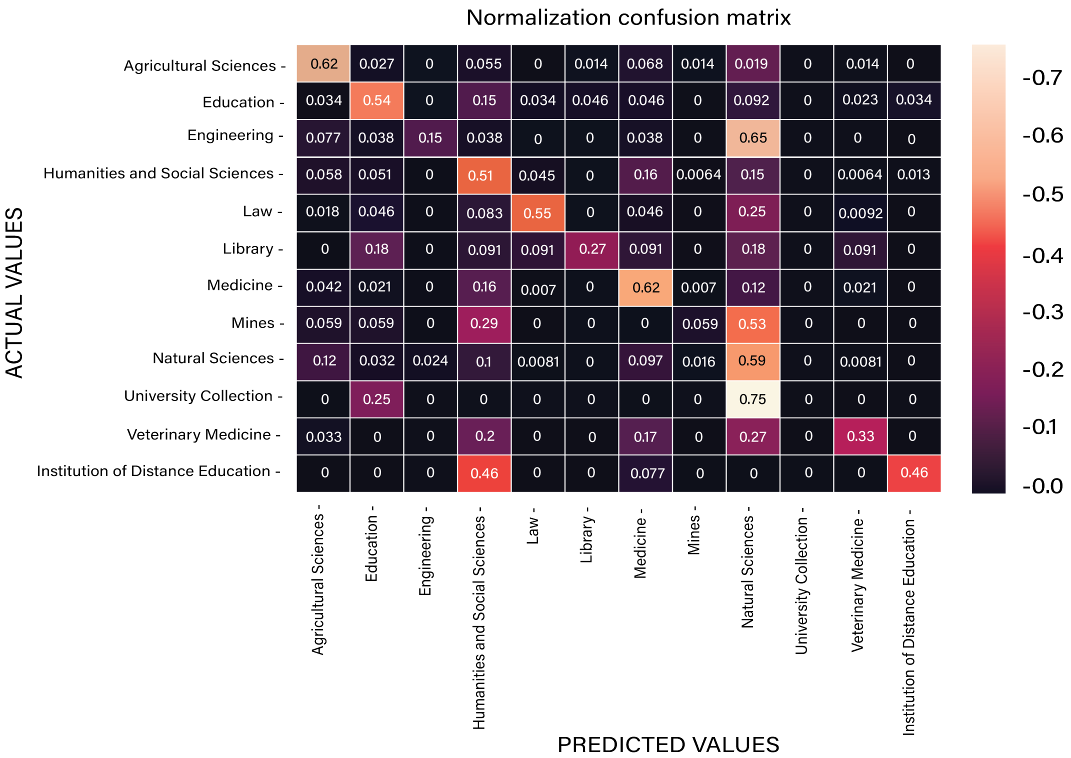
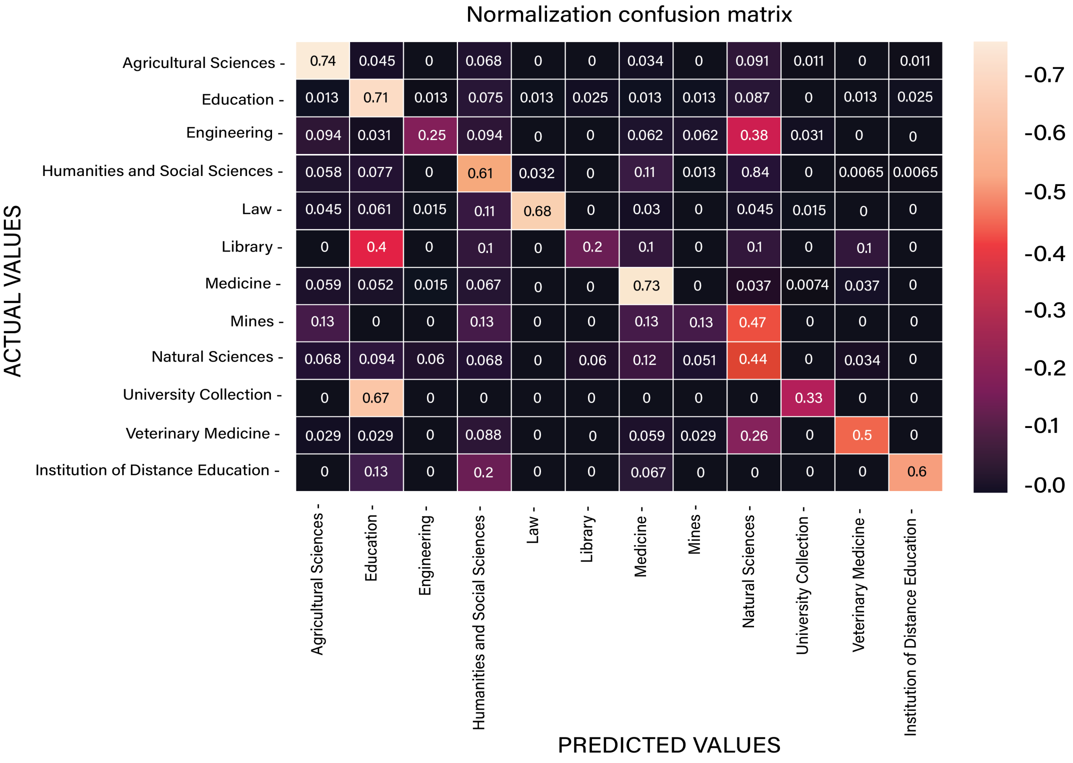


FIGURE 4.7: Confusion matrix: collection type classification - Title Feature

TABLE 4.2: Abstract Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.68	0.53	0.41	0.42
2	Stochastic Gradient Descent	0.72	0.80	0.56	0.61
3	Multinominal	0.51	0.46	0.27	0.26
4	Random Forest	0.65	0.75	0.47	0.50
5	Support Vector Machine	0.76	0.78	0.67	0.71
6	Decision Tree	0.47	0.52	0.43	0.46

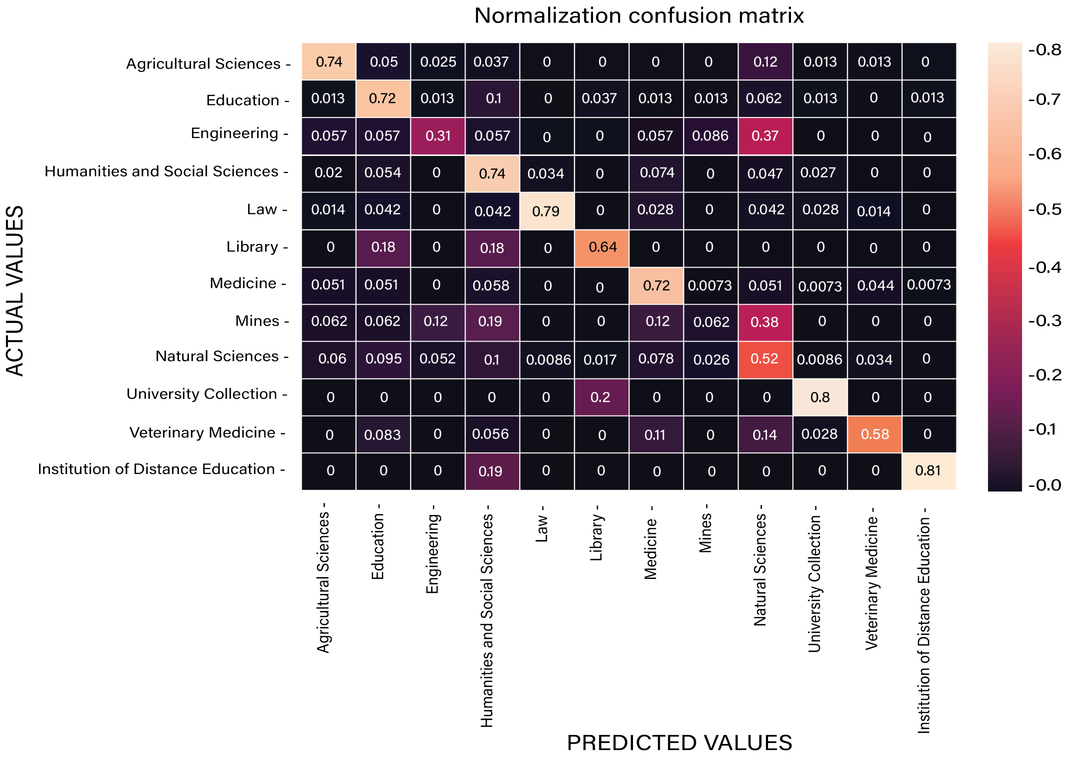


	Agricultural Sciences	Education	Engineering	Humanities and Social Sciences	Law	Library	Medicine	Mines	Natural Sciences	University Collection	Veterinary Medicine	Institution of Distance Education
Agricultural Sciences	0.74	0.045	0	0.068	0	0	0.034	0	0.091	0.011	0	0.011
Education	0.013	0.71	0.013	0.075	0.013	0.025	0.013	0.013	0.087	0	0.013	0.025
Engineering	0.094	0.031	0.25	0.094	0	0	0.062	0.062	0.38	0.031	0	0
Humanities and Social Sciences	0.058	0.077	0	0.61	0.032	0	0.11	0.013	0.84	0	0.0065	0.0065
Law	0.045	0.061	0.015	0.11	0.68	0	0.03	0	0.045	0.015	0	0
Library	0	0.4	0	0.1	0	0.2	0.1	0	0.1	0	0.1	0
Medicine	0.059	0.052	0.015	0.067	0	0	0.73	0	0.037	0.0074	0.037	0
Mines	0.13	0	0	0.13	0	0	0.13	0.13	0.47	0	0	0
Natural Sciences	0.068	0.094	0.06	0.068	0	0.06	0.12	0.051	0.44	0	0.034	0
University Collection	0	0.67	0	0	0	0	0	0	0	0.33	0	0
Veterinary Medicine	0.029	0.029	0	0.088	0	0	0.059	0.029	0.26	0	0.5	0
Institution of Distance Education	0	0.13	0	0.2	0	0	0.067	0	0	0	0	0.6

FIGURE 4.8: Confusion matrix: collection type classification - Abstract Feature

TABLE 4.3: Title and Abstract Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.68	0.69	0.44	0.46
2	Stochastic Gradient Descent	0.75	0.78	0.63	0.67
3	Multinominal	0.51	0.46	0.27	0.26
4	Random Forest	0.65	0.76	0.48	0.51
5	Support Vector Machine	0.77	0.77	0.70	0.72
6	Decision Tree	0.51	0.54	0.49	0.50



	Agricultural Sciences	Education	Engineering	Humanities and Social Sciences	Law	Library	Medicine	Mines	Natural Sciences	University Collection	Veterinary Medicine	Institution of Distance Education
Agricultural Sciences	0.74	0.05	0.025	0.037	0	0	0	0	0.12	0.013	0.013	0
Education	0.013	0.72	0.013	0.1	0	0.037	0.013	0.013	0.062	0.013	0	0.013
Engineering	0.057	0.057	0.31	0.057	0	0	0.057	0.086	0.37	0	0	0
Humanities and Social Sciences	0.02	0.054	0	0.74	0.034	0	0.074	0	0.047	0.027	0	0
Law	0.014	0.042	0	0.042	0.79	0	0.028	0	0.042	0.028	0.014	0
Library	0	0.18	0	0.18	0	0.64	0	0	0	0	0	0
Medicine	0.051	0.051	0	0.058	0	0	0.72	0.0073	0.051	0.0073	0.044	0.0073
Mines	0.062	0.062	0.12	0.19	0	0	0.12	0.062	0.38	0	0	0
Natural Sciences	0.06	0.095	0.052	0.1	0.0086	0.017	0.078	0.026	0.52	0.0086	0.034	0
University Collection	0	0	0	0	0	0.2	0	0	0	0.8	0	0
Veterinary Medicine	0	0.083	0	0.056	0	0	0.11	0	0.14	0.028	0.58	0
Institution of Distance Education	0	0	0	0.19	0	0	0	0	0	0	0	0.81

FIGURE 4.9: Confusion matrix: collection type classification - Title + Feature

Results for collection model using input title feature transformed using Tfid-Vectorizer

TABLE 4.4: Title Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.65	0.66	0.54	0.58
2	Stochastic Gradient Descent	0.65	0.61	0.56	0.58
3	Multinominal	0.66	0.72	0.53	0.57
4	Random Forest	0.61	0.67	0.52	0.57
5	Support Vector Machine	0.60	0.58	0.55	0.56
6	Decision Tree	0.60	0.57	0.54	0.55

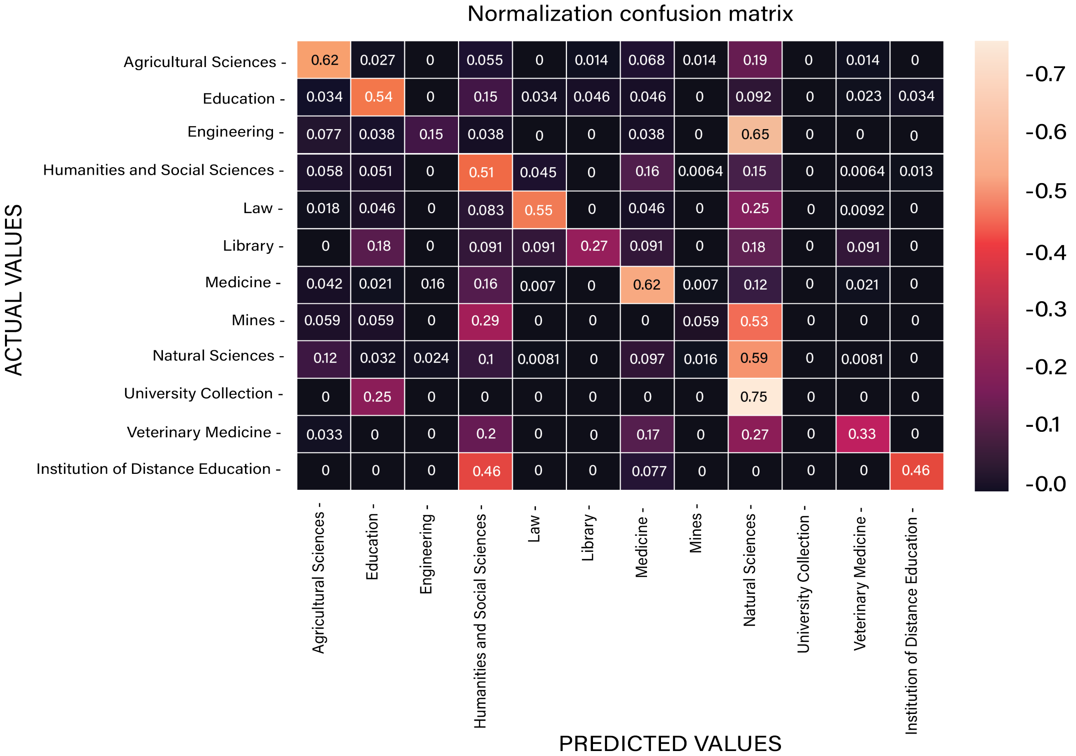
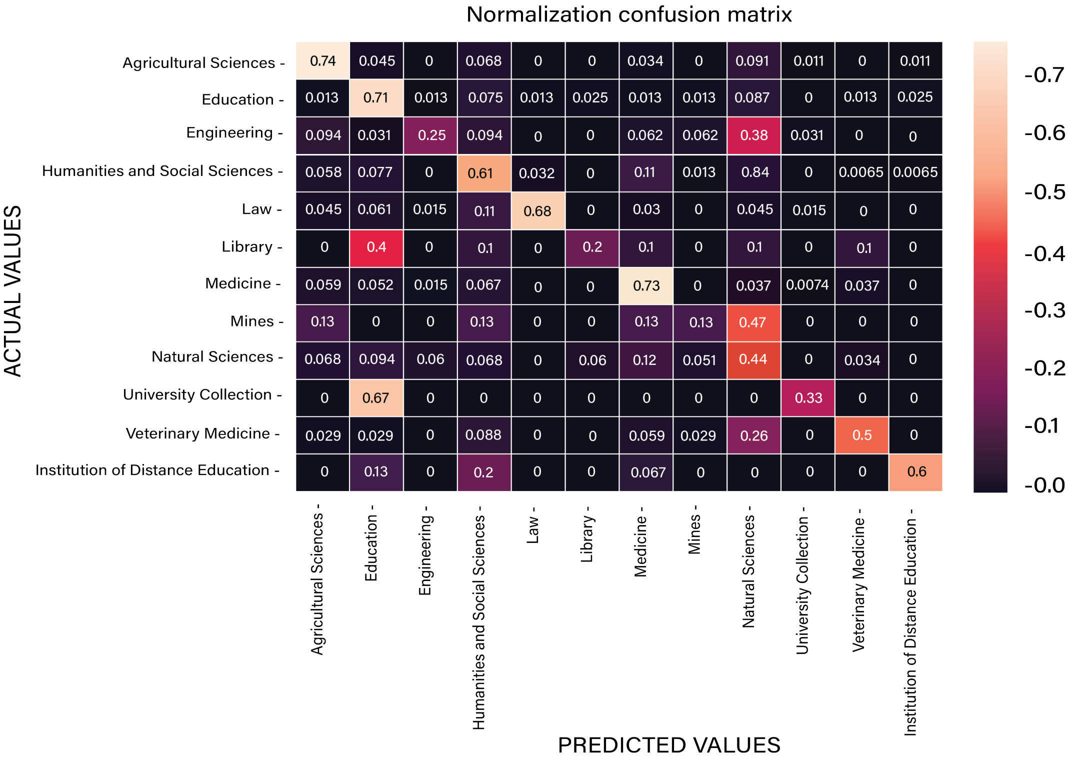


FIGURE 4.10: Confusion matrix: collection type classification - Title Feature

TABLE 4.5: Abstract Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.72	0.74	0.62	0.66
2	Stochastic Gradient Descent	0.68	0.69	0.60	0.62
3	Multinominal	0.70	0.77	0.51	0.55
4	Random Forest	0.66	0.76	0.48	0.51
5	Support Vector Machine	0.67	0.65	0.62	0.62
6	Decision Tree	0.49	0.46	0.46	0.45

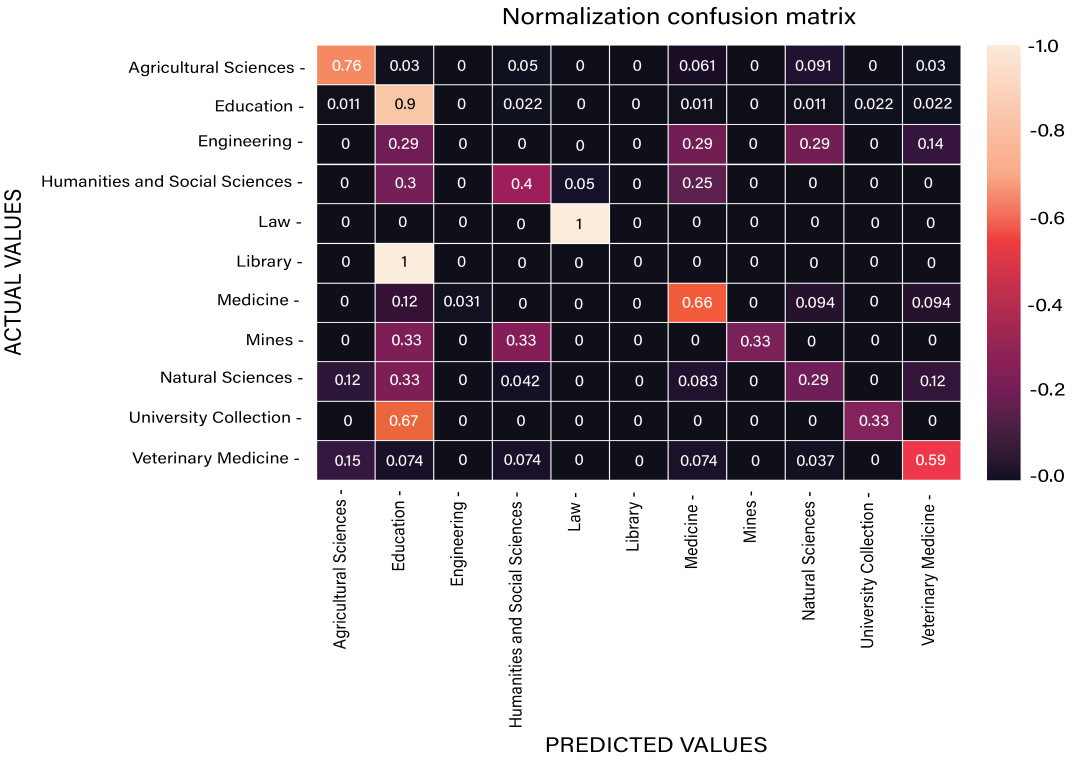


	Agricultural Sciences	Education	Engineering	Humanities and Social Sciences	Law	Library	Medicine	Mines	Natural Sciences	University Collection	Veterinary Medicine	Institution of Distance Education
Agricultural Sciences	0.74	0.045	0	0.068	0	0	0.034	0	0.091	0.011	0	0.011
Education	0.013	0.71	0.013	0.075	0.013	0.025	0.013	0.013	0.087	0	0.013	0.025
Engineering	0.094	0.031	0.25	0.094	0	0	0.062	0.062	0.38	0.031	0	0
Humanities and Social Sciences	0.058	0.077	0	0.61	0.032	0	0.11	0.013	0.84	0	0.0065	0.0065
Law	0.045	0.061	0.015	0.11	0.68	0	0.03	0	0.045	0.015	0	0
Library	0	0.4	0	0.1	0	0.2	0.1	0	0.1	0	0.1	0
Medicine	0.059	0.052	0.015	0.067	0	0	0.73	0	0.037	0.0074	0.037	0
Mines	0.13	0	0	0.13	0	0	0.13	0.13	0.47	0	0	0
Natural Sciences	0.068	0.094	0.06	0.068	0	0.06	0.12	0.051	0.44	0	0.034	0
University Collection	0	0.67	0	0	0	0	0	0	0	0.33	0	0
Veterinary Medicine	0.029	0.029	0	0.088	0	0	0.059	0.029	0.26	0	0.5	0
Institution of Distance Education	0	0.13	0	0.2	0	0	0.067	0	0	0	0	0.6

FIGURE 4.11: Confusion matrix: collection type classification - Title Feature

TABLE 4.6: Title and Abstract Feature.

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.81	0.39	0.26	0.31
2	Stochastic Gradient Descent	0.80	0.38	0.28	0.31
3	Multinominal	0.77	0.45	0.23	0.28
4	Random Forest	0.79	0.43	0.22	0.25
5	Support Vector Machine	0.80	0.39	0.27	0.30
6	Decision Tree	0.74	0.34	0.27	0.28



	Agricultural Sciences	Education	Engineering	Humanities and Social Sciences	Law	Library	Medicine	Mines	Natural Sciences	University Collection	Veterinary Medicine
Agricultural Sciences	0.76	0.03	0	0.05	0	0	0.061	0	0.091	0	0.03
Education	0.011	0.9	0	0.022	0	0	0.011	0	0.011	0.022	0.022
Engineering	0	0.29	0	0	0	0	0.29	0	0.29	0	0.14
Humanities and Social Sciences	0	0.3	0	0.4	0.05	0	0.25	0	0	0	0
Law	0	0	0	0	1	0	0	0	0	0	0
Library	0	1	0	0	0	0	0	0	0	0	0
Medicine	0	0.12	0.031	0	0	0	0.66	0	0.094	0	0.094
Mines	0	0.33	0	0.33	0	0	0	0.33	0	0	0
Natural Sciences	0.12	0.33	0	0.042	0	0	0.083	0	0.29	0	0.12
University Collection	0	0.67	0	0	0	0	0	0	0	0.33	0
Veterinary Medicine	0.15	0.074	0	0.074	0	0	0.074	0	0.037	0	0.59

FIGURE 4.12: Confusion matrix: collection type classification - Title + Abstract Feature

Abstract Feature In the double metadata parameter, the researcher combined the 'Title' and 'Abstract' feature, the combination yielded better results with Support Vector Machine outperforming the other estimators where Accuracy Score was 0.77, Precision Score was 0.77, Recall Score was 0.70 and F1 Score was 0.72 as depicted in Table 4.6. The double metadata parameter keenly results yielded better performance seeing as the combining two feature results in a more enhanced feature. The basic reason of improvement of classification is that, while it can be debated, enhanced new feature set can potentially be created , metadata elements such as keywords which in some cases provided alongside traditional ones like 'Title' and 'Abstract'.

4.2.2 Publication Type Model

In this experiments, the aim was to build a model that would classify documents according to their document type. Thus, the researcher strived to categorize documents into the following categories: Books, Book Chapters, Conference paper and Journal Articles. Like explained in section 3.7, three corpuses in this research were used. For this model, the second corpus was used which comprised of digital objects that had been harvested from the University of Zambia, Stellenbosch University, University of Pretoria, University of Cape Town and University of South Africa.

Despite harvesting 5,500 digital objects from UNZA, 12,216 from Stellenbosch, 21,068 from Cape Town, 14,499 from Pretoria and 9,729 from University of South African, after analyzing these digital objects, it was discovered that 70 percent of the them did not have bit streams but rather they only had metadata. The researcher proceed to harvest the bit streams for all the digital objects that had the files. After analysis, it was noticed that just like in the UNZA case , it was discovered that also the bitstreams harvested from these external repositories, most of them were scanned copies and it was difficult for the researcher to extract the text that was needed to use for the experiments. Despite that challenge, the researcher managed to combined all the bitstreams that had been filtered from the harvested data and extracted the needed text and built corpus that was used for the second model.

Similar to the collection classification model, the learning algorithms that was used was Stochastic Gradient Descent, Logistic Regression, Support Vector Machine, Multi-nominal, Random Forest and Decision Tree . For the respective algorithms, only one input feature was used that was extraction of the pages of the digital objects. In this regard, for the Journal Articles, book chapters and conference papers the researcher extracted text from the first two pages because the researcher was only interested in the details that was on the first and sometimes second page, while for the books the researcher, extracted text from the first sixteen pages of the books. The extracted text from the various categories was therefore used as input feature. The input feature was then converted using TfidfVectorizer and CountVectorizer.

It was observed that the input feature converted using TfidfVectorizer yield better results that of CountVectorizer. For evaluation of the performance of the model accuracy, precision, recall and f-measure was calculated , Table 4.7 shows the results obtained from the using the input feature transformed using the CountVectorizer. On the contrary Table 4.8 depicts the results with the input features transformed using TfidfVectorizer.

Among all the estimators that were used in this experiments conducted in this model, It was observed that Support Vector Machine achieved highest results with Accuracy of 0.82, average precision of 0.81, recall of 0.81 and F1 Score of 0.81 as show in Table 4.7. The second top scored was using the estimator Logistic Regression the Accuracy of 0.78, average precision of 0.77, recall of 0.76 and F1 Score of 0.77. Multinomial estimator had the least results with the accuracy of 0.69, average of precision 0.76, recall of 0.69 and 0.70 as shown in Table 4.8.

Results for collection model using input feature transformed using TfidfVectorizer.

TABLE 4.7: First 8 Pages as input feature

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.81	0.81	0.81	0.80
2	Stochastic Gradient Descent	0.79	0.78	0.79	0.78
3	Multinominal	0.69	0.76	0.69	0.70
4	Random Forest	0.80	0.81	0.79	0.79
5	Support Vector Machine	0.82	0.81	0.81	0.81
6	Decision Tree	0.73	0.72	0.72	0.72

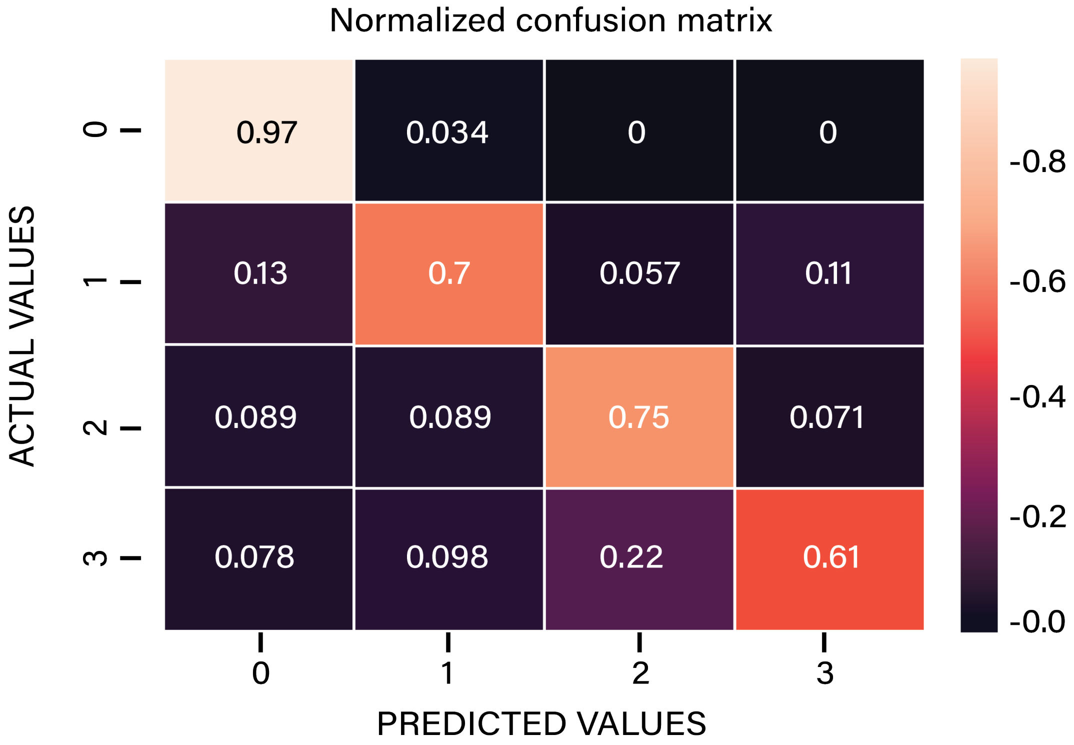
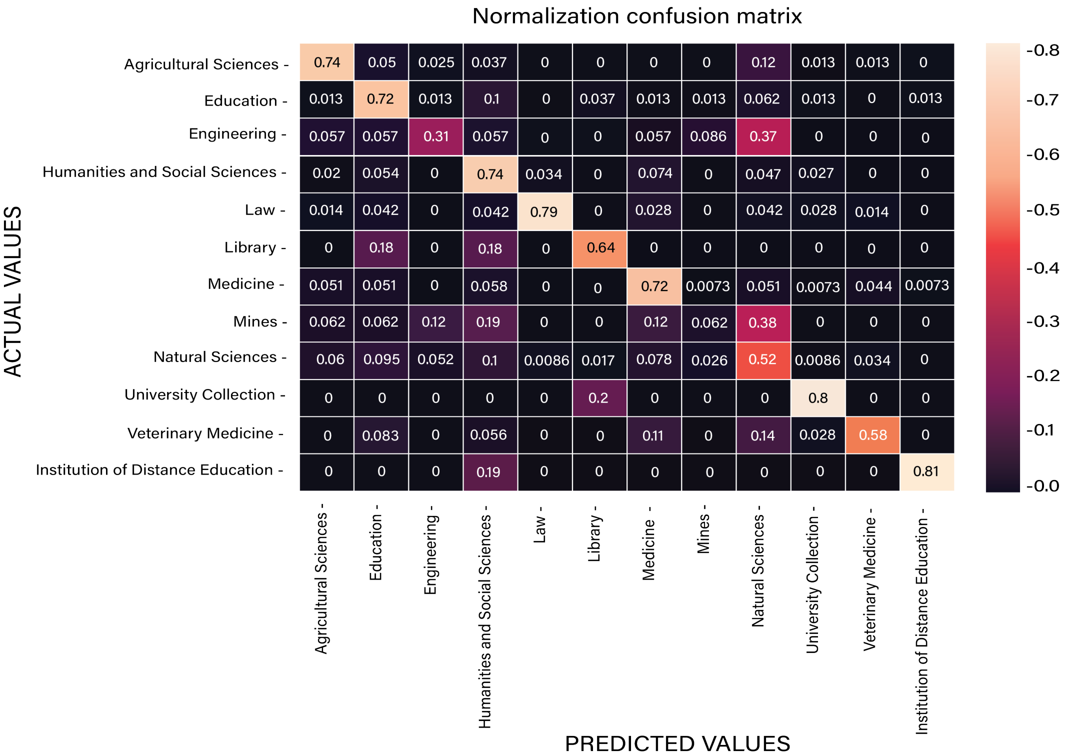


FIGURE 4.13: Confusion matrix: Publication type classification conidf

Results for collection model using input feature transformed using CountVectorizer Similarly, in the experiments conducted with input feature transformed using countvectorizer, accuracy, average precision, recall and F1 score was calculated. From the results obtained, it was noticed that Random Forest outperformed that other estimators with accuracy of 0.77, average precision of 0.78, Recall of 0.77 and F1 Score of 0.76. And the least estimator is Multinominal with lowest results with accuracy of 0.65, average precision of 0.67, recall of 0.65 and 0.65 as showed in table

TABLE 4.8: First 8 Pages as input feature

No	*Classifier*	*Accuracy*	*Precision*	*Recall*	*F1Measure*
1	Logistic Regression	0.73	0.72	0.72	0.72
2	Stochastic Gradient Descent	0.73	0.73	0.73	0.72
3	Multinominal	0.65	0.67	0.65	0.65
4	Random Forest	0.78	0.78	0.76	0.77
5	Support Vector Machine	0.66	0.69	0.63	0.63
6	Decision Tree	0.71	0.72	0.70	0.71



	Agricultural Sciences -	Education -	Engineering -	Humanities and Social Sciences -	Law -	Library -	Medicine -	Mines -	Natural Sciences -	University Collection -	Veterinary Medicine -	Institution of Distance Education -
Agricultural Sciences -	0.74	0.05	0.025	0.037	0	0	0	0	0.12	0.013	0.013	0
Education -	0.013	0.72	0.013	0.1	0	0.037	0.013	0.013	0.062	0.013	0	0.013
Engineering -	0.057	0.057	0.31	0.057	0	0	0.057	0.086	0.37	0	0	0
Humanities and Social Sciences -	0.02	0.054	0	0.74	0.034	0	0.074	0	0.047	0.027	0	0
Law -	0.014	0.042	0	0.042	0.79	0	0.028	0	0.042	0.028	0.014	0
Library -	0	0.18	0	0.18	0	0.64	0	0	0	0	0	0
Medicine -	0.051	0.051	0	0.058	0	0	0.72	0.0073	0.051	0.0073	0.044	0.0073
Mines -	0.062	0.062	0.12	0.19	0	0	0.12	0.062	0.38	0	0	0
Natural Sciences -	0.06	0.095	0.052	0.1	0.0086	0.017	0.078	0.026	0.52	0.0086	0.034	0
University Collection -	0	0	0	0	0	0.2	0	0	0	0.8	0	0
Veterinary Medicine -	0	0.083	0	0.056	0	0	0.11	0	0.14	0.028	0.58	0
Institution of Distance Education -	0	0	0	0.19	0	0	0	0	0	0	0	0.81

FIGURE 4.14: Confusion matrix: Publication type classification conidf

4.2.3 Subject Classification Model

The last model that was developed was a multi-label classification model. Multilabel classification is a supervised learning problem in which an object may be associated with multiple labels. This is different to the traditional task of multi-class or binary approach where each object is only associated with a single class label. Algorithms used to deal with multi-label problems can be classified either into problem transformation method or adaptation method [80] Problem transformation approach converts a multi-label problem into one or more single label problem on the contrary adaptation approach changes specific learning algorithms directly for multi-label. There are four data decomposition strategies that are mostly used under the problem transformation problem, namely Label Power(LP), Chain Classifier(CC), One-versus-rest(OVR) and Binary relevance(BR).

Label Power (LP) is a multi-label classification technique used in machine learning. In the categorization method known as multi-label, each instance may be simultaneously assigned to a number of different labels. LP is a technique that involves training a single model to predict all of the labels at the same time. The model is generally a multi-layer neural network. It accepts the features as input and then predicts a set of probabilities for each possible combination of labels based on those features [81].

LP is appropriate for use with huge datasets and has good processing efficiency. Also, it is able to record the dependencies that exist between the labels. This method could be helpful in circumstances in which the number of labels is somewhat high but the number of occurrences associated with each label is relatively low.

Training LP, on the other hand, may be difficult, particularly when there are a big number of labels and a huge number of possible label combinations. The process of training may call for a significant amount of data and may take a significant amount of time. In addition, LP operates under the presumption that each label operates independently of the others, which may not always be the case.

Chain Classifier (CC) is a multi-label classification technique used in machine learning. In multi-label classification, each instance can be assigned to multiple labels simultaneously. CC is a method where a chain of binary classifiers is used to predict the labels in a specific order. The output of each classifier is used as an input to the next classifier in the chain [82].

CC can handle label dependencies and is suitable for cases where the order of labels is important. For example, in a text classification task where the labels represent the topics of the text, the order of the topics might be relevant. CC can take into account the order of the topics and use it to improve the classification accuracy.

The One-versus-Rest (OVR) approach is widely used in machine learning for multi-class categorization. Each instance is classified into exactly one of a set of classes in multi-class classification.

The OVR method involves training several binary classifiers, one for each class, with each class being considered the positive class and the others the negative classes. In the testing phase, predictions are made by each classifier, and the one with the greatest confidence score is used [83].

Even if some classes have many more examples than others, OVR can still work with such an uneven dataset. Nevertheless, it does not account for interclass dependencies or correlations, which might produce less-than-ideal outcomes in some circumstances.

OVR is a popular starting point for more complex categorization systems due to its simplicity and effectiveness. It is applicable with any binary classifier and achieves optimal results based on the class distribution and the quality of the binary classifiers.

Binary Relevance (BR) is a classification algorithm for multiple labels used in machine learning. In multi-label categorization, each instance can be simultaneously assigned to numerous labels. BR is a technique in which a binary classifier is trained independently for each label. Each classifier predicts alone, without considering the other labels, whether an occurrence corresponds to a certain label [84].

BR is simple, computationally efficient, and capable of dealing with a high number of labels. Nevertheless, it does not take label dependencies into consideration. BR presupposes that each label is independent from the others, which is not always the case.

The majority of classifiers in BR will predict a negative outcome for each given case, resulting in an unbalanced dataset. Several solutions, like balanced subsampling and modifying the decision threshold, can be applied to overcome this issue.

In this work, the researcher used both problem transformation and adaptation approaches to deal with our problem. In particular, One-vs-rest(OVR), Binary Relevance and Classifier Chain decomposition strategies were the only approcahes adopted because they required less computation power requirement unlike Label Power which requires high computation power. The researcher then combined them with three estimators, namely Multinominal, SGDClassifier and Random Forest. Similar to the other two models developed, the researcher also used individual input features that is the 'title' and 'abstract' and lastly, a combined the feature ;'title' and 'abstract'.Furthermore each feature was converted each using TfidfVectorizer and CountVectorizer. It was noticed that the input features converted using TfidfVectorizer out performed that of CountVectorizer, this was observed in all the three input features. Like explain in chapter 3 for analysis and comparison of the performance of the estimators used for the multi-label classification were different but for one, Hamming Loss, Jaccard Distance and F1Score. Thus, each respective metadata feature, Hamming Loss, Jaccard Distance and F-measure was calculated for all the categories as illustrated in Table 4.9. Thus, each individual metadata feature, Hamming Loss, Jaccard Distance and F-measure was calculated for all the categories, Table 4.9 shows the results obtained from the using the input feature transformed using the CountVectorizer as well as the results with the input features transformed using TfidfVectorizer. Same trend was observed in this experiments conducted in this multi –label model, were the combination of the features of the 'Title' and 'Abstract' achieved highest results F1 Score of 0.54, hamming loss of 0.005, and Jaccard distance of 0.431 as show in Table 4.11. The reason of effectiveness of combination of features of title and abstract is the presence of keywords and in often cases keywords contains such words which is mostly used in other domains areas so their similarity increases with other categories and its predication rate have also increase. The second top scored was using the abstract feature with the F1 Score of 0.523, Hamming Loss of 0.005, and Jaccard Score of 0.413. The title feature had the least results with the F1Score of 0.09, Hamming of 0.006, and Jaccard of Distance of 0.177 as shown in

Table 4.9. Similar better performance of the combination the 'title' and 'abstract' feature was observed in all the experiments conducted in this study. Among the three decomposition strategies that was used, it was observed that classifier chain with SGDClassifer estimator outperform the other two strategies with all the three input features as shown in Table 4.9.

TABLE 4.9: Experimental Results for arXiv Subject Classes Multi-Label Classification Model.

Binary Relevance										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
2*MultinomialNB	TF	0.305	0.006	0.192	0.214	0.037	0.207	0.203	0.041	0.196
	TF-IDF	0.236	0.006	0.148	0.398	0.005	0.271	0.420	0.005	0.290
2*RandomForest	TF	0.317	0.006	0.211	0.416	0.005	0.292	0.430	0.005	0.305
	TF-IDF	0.314	0.006	0.210	0.418	0.005	0.295	0.435	0.005	0.310
2*SGDClassifier	TF	0.279	0.006	0.18	0.515	0.005	0.390	0.526	0.005	0.407
	TF-IDF	0.282	0.006	0.183	0.476	0.005	0.351	0.496	0.005	0.369
Classifier Chains										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
2*MultinomialNB	TF	0.060	0.055	0.190	0.030	0.338	0.130	0.030	0.347	0.123
	TF-IDF	0.090	0.027	0.177	0.086	0.053	0.282	0.087	0.055	0.294
2*RandomForest	TF	0.287	0.009	0.238	0.428	0.005	0.305	0.441	0.005	0.318
	TF-IDF	0.289	0.009	0.239	0.424	0.005	0.301	0.444	0.005	0.320
2*SGDClassifier	TF	0.312	0.006	0.216	0.527	0.005	0.420	0.520	0.006	0.414
	TF-IDF	0.310	0.006	0.214	0.523	0.005	0.413	**0.540**	0.005	**0.431**
One-Versus-Rest										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
2*MultinomialNB	TF	0.305	0.006	0.192	0.214	0.037	0.207	0.203	0.041	0.196
	TF-IDF	0.236	0.006	0.148	0.398	0.005	0.271	0.420	0.005	0.290
2*RandomForest	TF	0.317	0.006	0.212	0.414	0.005	0.291	0.435	0.005	0.310
	TF-IDF	0.315	0.006	0.210	0.414	0.005	0.290	0.432	0.005	0.306
2*SGDClassifier	TF	0.279	0.006	0.180	0.488	0.005	0.365	0.520	0.005	0.400
	TF-IDF	0.282	0.006	0.183	0.479	0.005	0.354	0.497	0.005	0.371

4.3 Model Deployment -Collection Classification Model

In order to demonstrate the feasibility of our proposal, the researcher deployed one model among the three models that were developed: collection classification model.The model was built using offline learning [85], and its state was saved to disk using the joblib package [86]. Deployment is the process of integrating a machine learning model into a production environment to make practical business decisions based on the data. It is only once the models have been deployed to production that they start adding value, thus making deployment a crucial step. Deployment is actually the last stage in the machine learning life cycle. The model can be deployed as Hypertext Transfer Protocol(HTTP) endpoints over a number of different environments and will usually be integrated with applications through an Application Programming Interface(API). Figure 4.15 shows the diagrammatic representation of the deployment process.

API is a collection of protocols, procedures, and tools used to develop software applications. It specifies how distinct software components should interface with one another, allowing them to exchange information and communicate [87].

APIs may be utilized to integrate disparate systems, services, and applications, allowing them to operate in concert. They are often employed in the development of online and mobile apps that rely on external data sources or services.

APIs can be secret or public. Public APIs are often available for usage by third-party developers in the creation of their own apps, whereas private APIs are used for internal reasons within a corporation or organization.

APIs can be categorized differently based on their purpose and functionality. Examples of typical API types include:

Web APIs are APIs that are accessed via HTTP requests and replies over the internet. REST (Representational State Transfer) and SOAP (Simple Object Access Protocol) APIs are examples. APIs that allow access to the features and functionalities of an operating system, including file systems, network protocols, and user interface components. Library APIs: APIs supplied by software libraries that allow application developers to utilize pre-built functions and modules. APIs are crucial building blocks for contemporary software development, allowing developers to design more robust and interconnected programs.

Collection Classification

As explained earlier, the collection classification model was the only model that was deployed in order to demonstrate the feasibility of proposal of this research. The machine learning model was deployed using the python Flask webserver [88] which was integrated with the Hyper Text Markup Language(HTML) webpage which was used to accept the abstract and the tile as the input feature and classify the document based on the classification model as shown in figure 4.16 and 4.17.

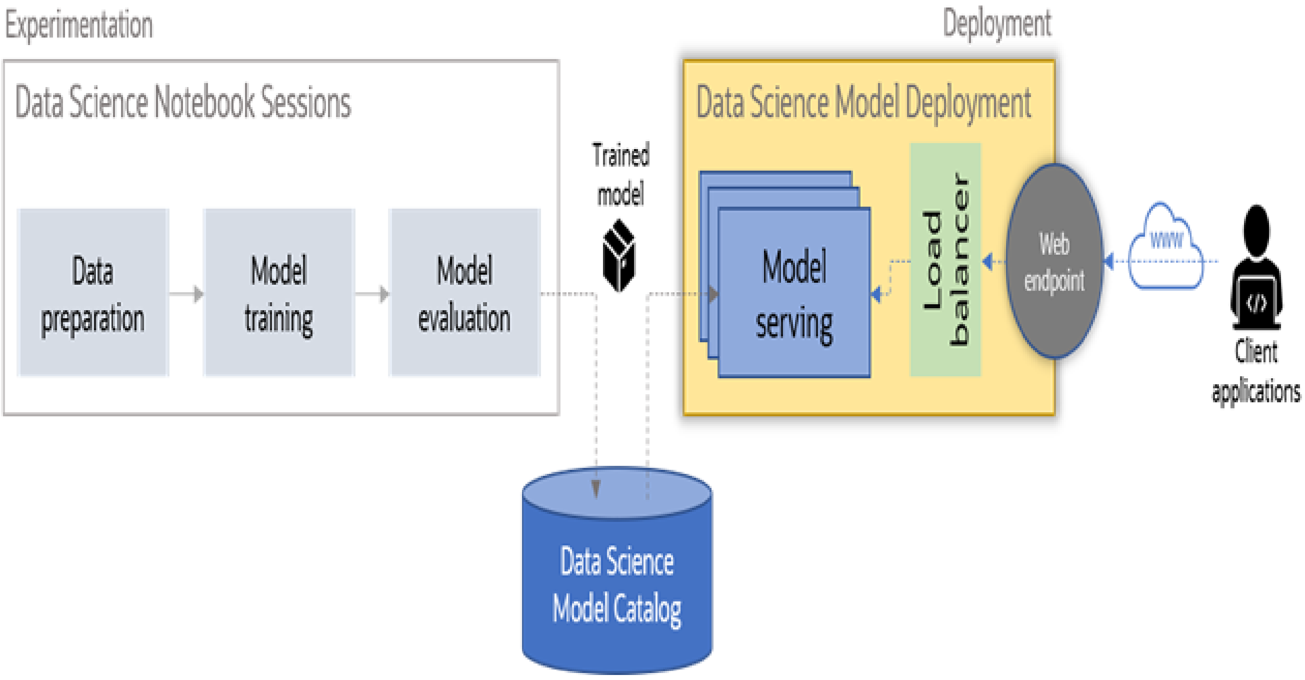
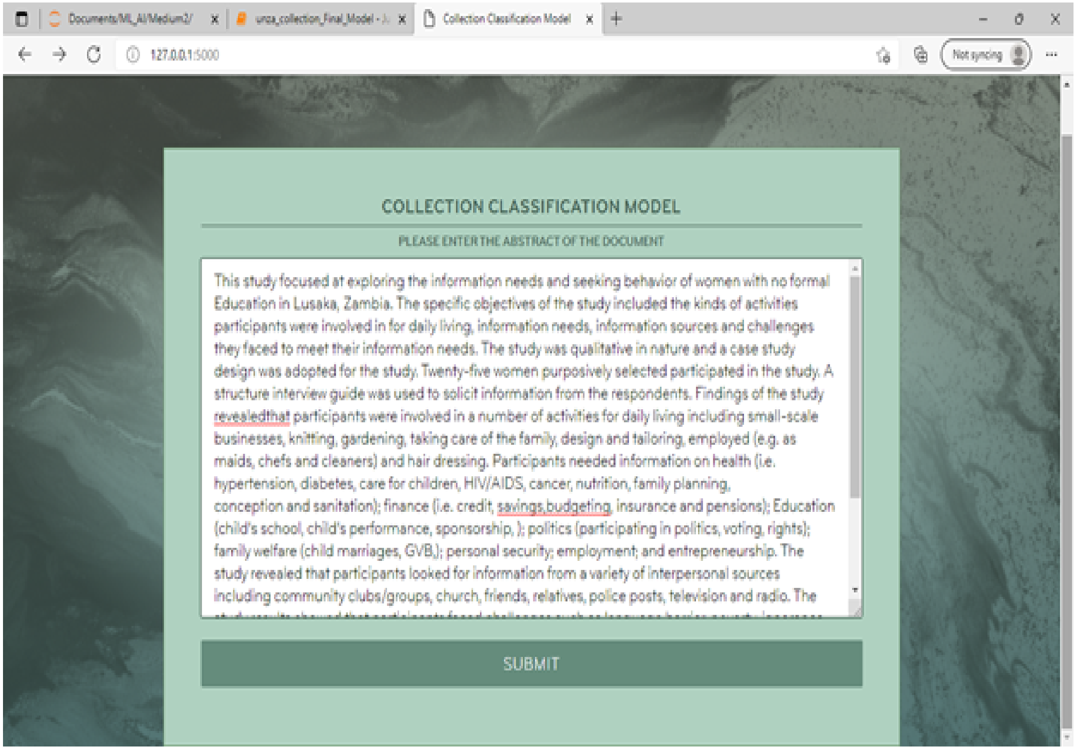


FIGURE 4.15: Deployment Process

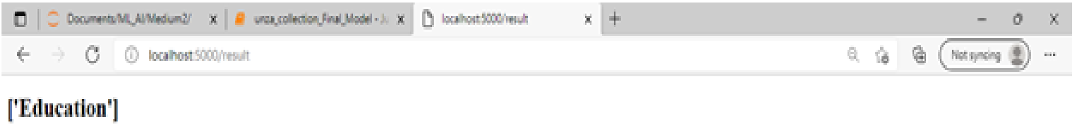

Documents/ML_AI/Medium2/
unza_collection_Final_Model - J
Collection Classification Model
127.0.0.1:5000
Not syncing

COLLECTION CLASSIFICATION MODEL

PLEASE ENTER THE ABSTRACT OF THE DOCUMENT

This study focused at exploring the information needs and seeking behavior of women with no formal Education in Lusaka, Zambia. The specific objectives of the study included the kinds of activities participants were involved in for daily living, information needs, information sources and challenges they faced to meet their information needs. The study was qualitative in nature and a case study design was adopted for the study. Twenty-five women purposively selected participated in the study. A structure interview guide was used to solicit information from the respondents. Findings of the study revealedthat participants were involved in a number of activities for daily living including small-scale businesses, knitting, gardening, taking care of the family, design and tailoring, employed (e.g. as maids, chefs and cleaners) and hair dressing. Participants needed information on health (i.e. hypertension, diabetes, care for children, HIV/AIDS, cancer, nutrition, family planning, conception and sanitation); finance (i.e. credit, savings,budgeting, insurance and pensions); Education (child's school, child's performance, sponsorship,); politics (participating in politics, voting, rights); family welfare (child marriages, GVB,); personal security; employment; and entrepreneurship. The study revealed that participants looked for information from a variety of interpersonal sources including community clubs/groups, church, friends, relatives, police posts, television and radio. The

FIGURE 4.16: Collection Input Form



```
['Education']
```

FIGURE 4.17: Collection output Form

Chapter 5

Conclusion

5.1 Conclusion and Future Work

The purpose of this research was to explore the feasibility of implementing multi-faceted automatic classification of Institutional Repository digital objects using machine learning. The rapid increase in the number of scholarly output produced has made the demand for automated classification of digital objects because there is great desire to increase the institutional profile as well as authors' visibility. Thus in order to archive that, it has become very imperative to ensure that all digital objects prior to ingestion, they have all the metadata tagged completely and furthermore, the digital objects are deposited in the correct community and collection. The inclusion of machine learning methods into the tagging and depositing process plays a vital role of ensuring the time consuming and error prone tasks are automated while human users complementing this process by countering the end results of the automation process. Based on the objectives of the study, the following conclusions were arrived at

- Two multi-class models were implemented – Publication type and Collection Models
- A multi-label model was implemented – Subject classification model.

5.1.1 Future Works and Recommendations

Some potential directions for future research on the area addressed in this dissertation are described below:

This work can be extended in the future by :

- Evaluation of proposed approaches for other digital objects like past examination papers and undergraduate research reports.
- Evaluation of proposed approaches using Label Power(LP) strategy.
- Finding ways to extract text data from scan documents to used in the publication type model.
- Evaluation of the proposed classification technique by the users so that they ascertain if it is working better than the current method.
- Use of the Application Programming Interfaces (APIs) outlined in Section 4.3 to provide efficient software tools and plugins that make use of automated classification of IR objects.

5.1.2 Real-world applications

Both the process of preparing digital objects information for ingestion into IRs prior to that procedure and the actual process of ingestion itself are activities that are time demanding and prone to mistakes such as missing important information and as well as misclassification. This is especially true for higher education institutions that lack enough resources. It is feasible to decrease the number of errors that occur during the process of preparing metadata by making use of the techniques of supervised machine learning that are explained in this work.

In addition, the amount of time spent ingesting digital objects into IRs may be able to be cut down if activities that were previously performed manually are automated. Under the suggested method, the function of employees during ingestion would basically comprise checking that the findings of automatically categorized digital objects are valid. In essence, this would be the core responsibility of the position. The verification and validation procedure may be implemented as a part of the ingestion workflow for the IR by making use of the API endpoints that are defined in the next subsection. Rather, the verification and validation might be done as an integrated service of an external service that generates an output that is readily ingestible into the repository. This would be a more efficient use of resources.

Chapter 6

Subject Classification Script

Subject Classification Script

```
mport pandas as pd
df = pd.read_csv('unza_collection_combined.csv',encoding='latin1')
df.head()
from sklearn.preprocessing import MultiLabelBinarizer
import json
subject_new=[] #declare a list
for cell in df['subject']:
    cell=cell.replace(" ", "") #remove whitespace
    cell=cell.replace("&", "& ") #add whitespace back in for ampersands
    subject_new.append(cell.split(",")) #for each genre cell, create a list of items from the original string, using a comma as a delimeter
    #add new genre column to the dataframe
df['subject_new'] = subject_new
mlb = MultiLabelBinarizer()
binary_labels=binary_labels.sort_index(axis=1)
binary_labels.head(10).T
documents = df.merge(binary_labels, how='inner', left_index=True, right_index=True)
documents= documents.drop(columns=['subject', 'description','subject_new'])
documents.tail(7)
import seaborn as sns
import matplotlib.pyplot as plt
categories = list(binary_labels.columns.values)
ax= sns.barplot(binary_labels.sum().values, categories)
plt.title("Documents for each Subject", fontsize=24)
plt.ylabel('Subject', fontsize=18)
plt.xlabel('Number of document tagged with subject', fontsize=18)
rects = ax.patches
labels = binary_labels.sum().values
plt.show()
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=10000)
xtrain, xval, ytrain, yval = train_test_split(documents['description_new'], binary_labels, test_size=0.2, random_state=9)
xtrain_tfidf = tfidf_vectorizer.fit_transform(xtrain)
xval_tfidf = tfidf_vectorizer.transform(xval)
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import accuracy_score
logreg = LogisticRegression()
logreg_classifier = OneVsRestClassifier(logreg)
logreg_classifier.fit(xtrain_tfidf, ytrain)
predictions = logreg_classifier.predict(xval_tfidf)
from sklearn.metrics import accuracy_score
print("Accuracy score for Logistic Regression:")
print(accuracy_score(yval, predictions))
from sklearn.metrics import hamming_loss
hamming_loss(yval, predictions)
from sklearn.metrics import classification_report
```

```
print(classification_report(yval, predictions, target_names=binary_labels.columns))
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import GaussianNB
classifier = BinaryRelevance(GaussianNB())
classifier.fit(xtrain_tfidf, ytrain)
predictions = classifier.predict(xval_tfidf)
print("Accuracy score for Gaussian Naive Bayes:")
print(accuracy_score(yval, predictions))
print("Individual subject predictions:")
print(classification_report(yval, predictions, target_names=binary_labels.columns))
from sklearn.metrics import hamming_loss
hamming_loss(yval, predictions)
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import MultinomialNB
classifier = BinaryRelevance(MultinomialNB())
classifier.fit(xtrain_tfidf, ytrain)
predictions = classifier.predict(xval_tfidf)
print("Accuracy score for MultinomialNB:")
print(accuracy_score(yval, predictions))
print("Individual subject predictions:")
print(classification_report(yval, predictions, target_names=binary_labels.columns))
hamming_loss(yval, predictions)
```

Chapter 7

Collection Classification Script

Collection Classification Script

```
import pandas as pd
df = pd.read_csv('unza_collection_combined.csv',encoding='latin1')
df.head()
df = df[pd.notnull(df['description'])]
df.info()
col = [ 'description','collection']
df = df[col]
df.columns
df.columns = [ 'description','collection']
df['collection_id'] = df['collection'].factorize()[0]
from io import StringIO
collection_id_df = df[['collection', 'collection_id']].drop_duplicates().sort_values('collection_id')
collection_to_id = dict(collection_id_df.values)
id_to_collection = dict(collection_id_df[['collection_id', 'collection']].values)
df['collection_id']=df['collection'].factorize()[0]
import matplotlib.pyplot as plt
fig = plt.figure(figsize=(10,8))
df.groupby('collection').description.count().plot.bar(ylim=0)
plt.show()
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf = TfidfVectorizer(sublinear_tf=True, min_df=5, norm='l2', encoding='latin-1', ngram_range=(1, 2), stop_words='english')
features = tfidf.fit_transform(df.description).toarray()
labels = df.collection_id
features.shape
from sklearn.feature_selection import chi2
import numpy as np
```

```
N = 2
for collection, collection_id in sorted(collection_to_id.items()):
  features_chi2 = chi2(features, labels == collection_id)
  indices = np.argsort(features_chi2[0])
  feature_names = np.array(tfidf.get_feature_names())[indices]
  unigrams = [v for v in feature_names if len(v.split(' ')) == 1]
  bigrams = [v for v in feature_names if len(v.split(' ')) == 2]
  print("# '{}':".format(collection))
  print("  . Most correlated unigrams:\n       . {}".format('\n       . '.join(unigrams[-N:])))
  print("  . Most correlated bigrams:\n       . {}".format('\n       . '.join(bigrams[-N:])))
  from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['description'], df['collection'], random_state = 0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = MultinomialNB().fit(X_train_tfidf, y_train)
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['description'], df['collection'], random_state = 0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
```

```
clf = MultinomialNB().fit(X_train_tfidf, y_train)
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
models = [
    RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0),
    LinearSVC(),
    MultinomialNB(),
    LogisticRegression(random_state=0),
]
CV = 5
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = []
for model in models:
  model_name = model.__class__.__name__
  accuracies = cross_val_score(model, features, labels, scoring='accuracy', cv=CV)
  for fold_idx, accuracy in enumerate(accuracies):
    entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])
import seaborn as sns
sns.boxplot(x='model_name', y='accuracy', data=cv_df)
sns.stripplot(x='model_name', y='accuracy', data=cv_df,
              size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()
cv_df.groupby('model_name').accuracy.mean()
from sklearn.model_selection import train_test_split
```

```
model = LinearSVC()

X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(features, labels, df.index, test_size=0.33, random_state=0)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

from sklearn.metrics import confusion_matrix

conf_mat = confusion_matrix(y_test, y_pred)

fig, ax = plt.subplots(figsize=(10,8))

sns.heatmap(conf_mat, annot=True, fmt='d',

            xticklabels=collection_id_df.collection.values, yticklabels=collection_id_df.collection.values)

plt.ylabel('Actual')

plt.xlabel('Predicted')

plt.show()

from IPython.display import display

for predicted in collection_id_df.collection_id:

  for actual in collection_id_df.collection_id:

    if predicted != actual and conf_mat[actual, predicted] >= 6:

      print("'{}' predicted as '{}' : {} examples.".format(id_to_collection[actual], id_to_collection[predicted], conf_mat[actual, predicted]))

      display(df.loc[indices_test[(y_test == actual) & (y_pred == predicted)]][['collection', 'description']])

      print('')

model.fit(features, labels)

from sklearn.feature_selection import chi2

N = 2

for collection, collection_id in sorted(collection_to_id.items()):

  indices = np.argsort(model.coef_[collection_id])

  feature_names = np.array(tfidf.get_feature_names())[indices]

  unigrams = [v for v in reversed(feature_names) if len(v.split(' ')) == 1][:N]

  bigrams = [v for v in reversed(feature_names) if len(v.split(' ')) == 2][:N]
  print("# '{}':".format(collection))
  print("  . Top unigrams:\n       . {}".format('\n       . '.join(unigrams)))
  print("  . Top bigrams:\n       . {}".format('\n       . '.join(bigrams)))
 from sklearn import metrics
print(metrics.classification_report(y_test, y_pred,
                                    target_names=df['collection'].unique()))
```

Chapter 8

Document Type Classification Model

DOCUMENT TYPE CLASSIFICATION SCRIPT

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_files
DATA_DIR ="./DatasetLu/"
data = load_files(DATA_DIR, encoding="utf-8", decode_error="replace")
# calculate count of each category
labels, counts = np.unique(data.target, return_counts=True)
# convert data.target_names to np array for fancy indexing
labels_str = np.array(data.target_names)[labels]
print(dict(zip(labels_str, counts)))
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target)
list(t[:80] for t in X_train[:10])
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words="english", max_features=1000, decode_error="ignore")
vectorizer.fit(X_train)
vectorizer.fit(X_train)
X_train_vectorized = vectorizer.transform(X_train)
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import cross_val_score
 # start with the classic
# with either pure counts or tfidf features
sgd = Pipeline([
    ("count vectorizer", CountVectorizer(stop_words="english", max_features=3000)),
    ("sgd", SGDClassifier(loss="modified_huber"))
```

```
   ])
sgd_tfidf = Pipeline([
       ("tfidf_vectorizer", TfidfVectorizer(stop_words="english", max_features=3000)),
       ("sgd", SGDClassifier(loss="modified_huber"))
   ])
svc = Pipeline([
       ("count_vectorizer", CountVectorizer(stop_words="english", max_features=3000)),
       ("linear svc", SVC(kernel="linear"))
   ])
svc_tfidf = Pipeline([
       ("tfidf_vectorizer", TfidfVectorizer(stop_words="english", max_features=3000)),
       ("linear svc", SVC(kernel="linear"))
   ])
all_models = [
   ("sgd", sgd),
   ("sgd_tfidf", sgd_tfidf),
   ("svc", svc),
   ("svc_tfidf", svc_tfidf),
   ]
 unsorted_scores = [(name, cross_val_score(model, X_train, y_train, cv=2).mean()) for name, model in all_models]
scores = sorted(unsorted_scores, key=lambda x: -x[1])
print(scores)
model = svc_tfidf
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Appendices

Appendix A

Interview Guidelines

Dear respondents,

The aim of this interview is to obtain information on all the procedures undertaken in ingestion of digital objects onto the University of Zambia's institutional repository. You have been selected to take part in this research project, the researchers assure you strict confidentiality and anonymity. Your area of expertise assures the researchers that you will provide the information needed.

1. What is your current position in the library?

2. What responsibilities does your position involve when it comes to handling institutional Repositories?

3. What are the different types of documents that you deposit into the Institution Repository?

4. What is the process of ingesting digital objects into the IR?

5. What criteria do you use to tag metadata to each document type?

6. How long does it take you to tag and complete the whole process of ingesting of digital objects into the Institution Repository?

7. What is the organisation of Digital Objects in the IR?

8. What are current challenges you have encountered when tagging digital Object?

9. What are current challenges you have encountered when tagging digital Object?

10. What you do you think should be done in order to overcome some of the challenges you inter counter during the process of tagging and successful ingestion of digital objects into the Institution Repository?

Appendix B

Journal Article Publication

Improved Discoverability of Digital Objects in Institutional Repositories Using Controlled Vocabularies

Bertha Chipangila†, Eric Liswaniso†, Andrew Mawila†, Philomena Mwanza†, Daisy Nawila†, Robert M'sendo‡,
Mayumbo Nyirenda‡, and Lighton Phiri†
†Department of Library and Information Science, University of Zambia, P.O. Box 32379, Lusaka, Zambia
‡Department of Computer Science, University of Zambia, P.O Box 32379, Lusaka, Zambia
Email: {13000438,15058590,15014576,15018148,15019551,20171520216}@student.unza.zm,
mayumbo.nyirenda@cs.unza.zm, lighton.phiri@unza.zm

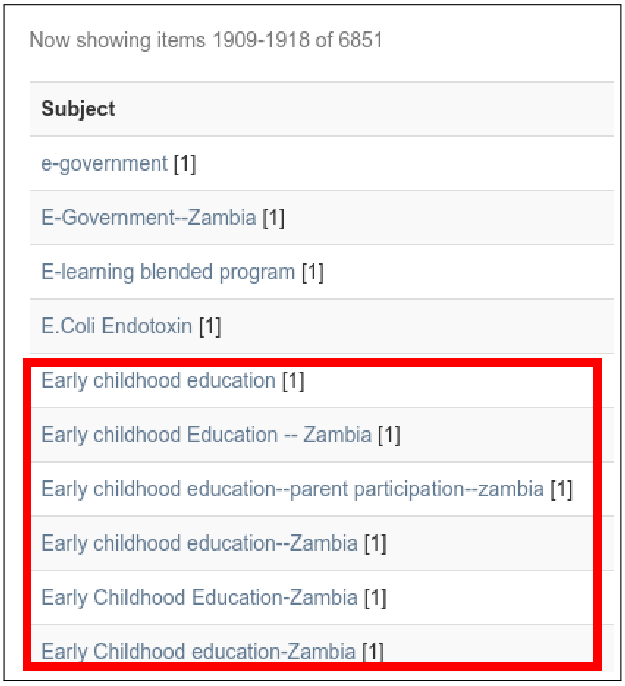
***Abstract*—Higher Education Institutions (HEIs) utilise Institutional Repositories (IRs) to electronically store and make available scholarly research output produced by faculty staff and students. With the continued increase of scholarly research output produced, accurate and comprehensive association of subject headings to digital objects, during ingestion into IRs is crucial for effective discoverability of the objects and, additionally facilitating the discovery of related content. This paper outlines a case study conducted at an HEI—The University of Zambia—in order to demonstrate the effectiveness of integrating controlled subject vocabularies during the ingestion of digital objects in to IRs. A situational analysis was conducted to understand how subject headings are associated with digital objects and to analyse subject headings associated with already ingested digital objects. In addition, an exploratory study was conducted to determine domain-specific subject headings to be integrated with the IR. Furthermore, a usability study was conducted in order to comparatively determine the usefulness of using controlled vocabularies during the ingestion of digital objects into IRs. Finally, multi-label classification experiments were carried out where digital objects were assigned with more than one class. The results of the study revealed that the majority of digital objects are currently associated with two or less subject headings ($71.2\,\%$), with a significant number of subject headings ($92.1\,\%$) being associated with a single publication. The comparative study suggests that IRs integrated with controlled vocabularies are perceived to be more usable (SUS Score = 68.9) when compared with IRs without controlled vocabularies (SUS Score = 66.2). The effectiveness of the multi-label arXiv subjects classifier demonstrates the viability of integrating automated techniques for subject classification.**

***Keywords*-Controlled Vocabularies; Digital Libraries; Document Classification; Institutional Repositories;**

I. Introduction

Institutional Repositories (IRs) are a crucial part of contemporary Higher Educational Institutions (HEIs) as they provide an avenue for making available scholarly research output produced by faculty staff and students. IRs provide a platform for capturing, preserving and facilitating access to digital work produced by a community [1].

Scholarly research output are typically stored as digital objects within the IRs, with the objects loosely comprising of digital object bitstreams and digital object metadata. Metadata, and more specifically descriptive metadata, is vital for ensuring effective discoverability of the digital objects in IRs.

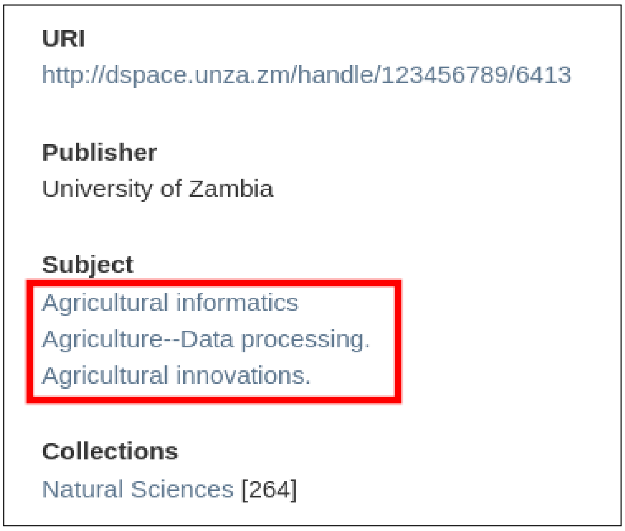


Now showing items 1909-1918 of 6851

Subject
e-government [1]
E-Government--Zambia [1]
E-learning blended program [1]
E.Coli Endotoxin [1]
Early childhood education [1]
Early childhood Education -- Zambia [1]
Early childhood education--parent participation--zambia [1]
Early childhood education--Zambia [1]
Early Childhood Education-Zambia [1]
Early Childhood education-Zambia [1]

Fig. 1: A screenshot showcasing sample subjects associated with ingested digital objects in The UNZA's IR.

The University of Zambia (UNZA) has a functional IR with scholarly output consistently deposited into the it, however, there are a number of inconsistencies associated with digital object metadata elements used to describe subject categories related to the objects. Prior work done has identified the lack of use of controlled vocabulary sets as being one of the leading causes of ineffective searching and browsing of scholarly research output in UNZA's IR [2]. In addition to the lack of use of controlled vocabularies, the lack of use of subject specific controlled vocabularies has the potential to make it difficult for end users to search and browse for domain specific content and related content. These critical anomalies are observable from UNZA's IR: Figure 1 illustrates how the extent to which subjects are inconsistently used, while Figure 2 illustrates how a digital object produced in the Department of Computer Science is associated with non domain-specific subjects.



URI
http://dspace.unza.zm/handle/123456789/6413

Publisher
University of Zambia

Subject
Agricultural informatics
Agriculture--Data processing.
Agricultural innovations.

Collections
Natural Sciences [264]

Fig. 2: A screenshot showcasing subjects associated with a sample Computer Science digital object in UNZA's IR.

This paper presents a study conducted at UNZA to investigate the effectiveness of integrating controlled subject vocabulary sets within UNZA's IR. The study comprised of three phases. First, a situational analysis was conducted to empirically determine the implications of the lack of integration of controlled vocabularies within the repository. In order to understand the potential sources of errors when preparing descriptive metadata, focus group discussions were held with Library staff that administer the IR. Secondly, interview sessions were held with faculty staff in order to identify controlled vocabularies used in their respective domains. Finally, a controlled experiment was designed to empirically determine the usability of IRs integrated with subject controlled vocabularies.

The remainder of this paper is organised as follows: Section II is a synthesis of existing literature related to this work, Section III describes the methodology associated with this work, Section IV presents and discusses the results of this study and, finally, Section V outlines concluding remarks.

II. Related Work

There is a large body of existing literature that has focused on the role of descriptive metadata in facilitating discoverability of digital objects and, the significance of using controlled vocabularies and authority control during ingestion of digital content.

A. *Digital Object Descriptive Metadata*

IR digital object metadata can be broadly categorised into into the three groups of metadata—administrative metadata, descriptive metadata and structural metadata—proposed by Riley [3]. The metadata is specified as part of an ingestion workflow, with the metadata associated to the digital object externally, as opposed to embedding it within the digital object. While all the three types of metadata are important, descriptive metadata specifically serves the purpose of facilitating the discovery of digital objects through searching and browsing services.

Arms highlights that information discovery is a complex process that can be made effective by referencing descriptive metadata about digital objects stored in repositories [4]. Similarly, Varlamis and Apostolakis emphasise the importance of labelling learning objects stored in learning object repositories in a consistent manner, in order to support indexing and discovery of the content [5]. The importance of labelling is further supported by Currier et al. who state that quality metadata, in particular, enables users to discover and retrieve digital objects in an efficient and effective manner [6].

The external metadata in IRs is encoded using internationally recognised metadata schemes, with Dublin Core [7] being the most widely integrated in popular open source IR software platforms. The digital object metadata is primarily indexed and used to facilitate searching and browsing, however, it is generally possible to activate full-text searching for text-based content.

UNZA's IR is powered by the DSpace open source repository platform. DSpace is capable of processing textual content for full-text searching, in addition to utilising metadata elements during indexing [8]. DSpace uses a default metadata registry that is derived from the 15 Dublin Core metadata elements, with the element values specified during ingestion of digital objects. One of the crucial metadata elements is the "dc.subject" element that specifies the topic associated with the resource

B. *Controlled Vocabularies*

Controlled vocabularies and authority control are popular techniques that are used to enhance access to bibliographic materials. Harpring defines controlled vocabularies as well-organised words and phrases that are used to index digital content and subsequently facilitate retrieval of the content through searching and browsing [9].

Subject headings are a form of controlled vocabularies that are used to describe topics associated to digital content, making it possible for content related content to be group together. While generic subject headings such as the Library of Congress Subject Headings (LCSH) are widely used, there are other domain specific subject headings, popular with academic databases. For instance, the Medical Subject Headings (MeSH) [10] terms are used in the medical field and the ACM Computing Classification System (CCS) [11] ontology is common used in computing disciplines.

Prior work on subject headings has mostly focused on the effectiveness of subject headings when compared with keywords. In a study aimed at comparing user tags and LCSH Rolla notes that user supplied tags can be used to enhance subject access but cannot replace the valuable role of controlled vocabularies [12]. This observation supports the results obtained by Lu et al. in a study that suggests that the existence of non-subject-related tags can improve the accessibility of collections [13].

In this work, we empirically determine the implications of sparing use of subject headings and, additionally, identify potential domain-specific subject headings that can be incorporated into IRs. Furthermore, we demonstrate the positive effect subject headings have on the overall usability of IRs.

C. *Multi-Label Classification*

Motivated by the ever increasingly vast number of digital objects and enhancement in machine learning and technology, multi-label classification has become an extensive studied problem. Multi-label context has in the recent years been researched much because of its application to a wide variety of domains. For example, Konstantions and Kalliris [14] dealt with the problem of automatic detection of emotions in music. Their work established the relation between music and emotion and further looked at multi-labelling mapping of music into emotions. Runzhi et al. used multi-label classification to deal with the problem of multi-disease risk prediction [15]. They constructed a model for prediction of multi-diseases risk relying on the big physical examination data. They acknowledged that in medical diagnosis, a symptom may be associated with various disease types. Chalkidis et al. apply Extreme Multi-Label Text Classification (XMTC) in the legal domain [16]. They employ neural classifiers that outperform the current multi-label state-of-the-art methods, which employ label-wise attention. Boutell et al. focused on video and photography analysis [17]. In semantic scene classification, a picture can be associated to more than on conceptual class such as a sunset and beaches at the same time.

In this work, a multi-label classifier is implemented using an external data source, by taking advantage of transfer learning, and subsequently applied to digital objects associated with the Computer Science field in UNZA's IR, in order to predict appropriate domain-specific subjects.

III. Methodology

The study took a mixed-methods approach involving a situational analysis (See Section III-B), an exploratory study aimed at identifying appropriate subject controlled vocabularies to integrated with the IR (see Section III-C), a usability study aimed at empirically evaluating the effect of controlled vocabularies when integrated with IRs (see Section III-D) and, implementation of a supervised machine learning multi-label classifier (see Section III-E).

A. *Datasets*

Four datasets were constructed and used to perform the situational analysis, outlined in Section III-B and, additionally, to validate the multi-label classification a model implemented as outlined in Section III-E2. Table I provides a summary of the datasets, with details outlined in Sections III-A1 to III-A4.

TABLE I: Datasets used during experimentation.

Dataset	Objects	Study
UNZA IR	7,440	Situational Analysis
CS@ UCT Archive	995	Situational Analysis · Model Validation
arXiv CoRR	328,011	Situational Analysis
NDLTD Union Catalog	7,296,562	Situational Analysis

1) Dataset #1: arXiv CoRR Dataset: The dataset used for implementing the multi-label classifier was constructed by harvesting Dublin Core [7] encoded metadata records from the arXiv Computing Research Repository (CoRR) [18]. The CoRR specific digital objects were filtered by restricting the harvesting using the OAI-PMH 'SetSpec' verb[1].

General preprocessing operations—removal of punctuations, stemming and stopword removal—were performed on the collected data. However, in addition, non-computing subjects had to be removed from dataset observations, as arXiv CoRR comprises of digital objects tagged with subjects such as Mathematics and Physics.

The constructed dataset comprises of 328,011 digital objects, ingested into CoRR between 2007 and 2021. In addition, the digital objects were tagged with combination of ACM CCS subjects, arXiv subject classes and a combination of the two subject classes. Figure 3 shows the distribution of the subject classes.

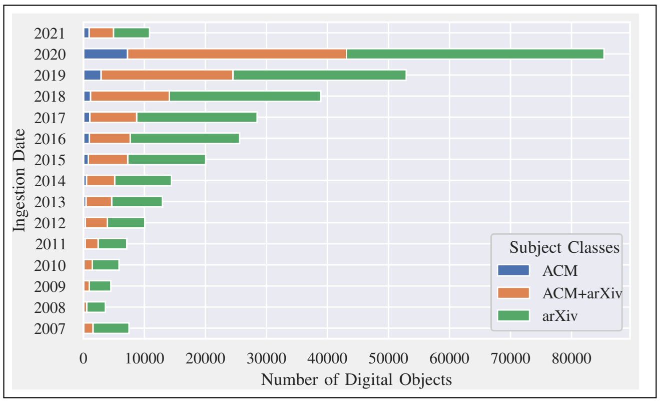


Fig. 3: Distribution of subject tags in the arXiv CoRR dataset.

2) Dataset #2: NDLTD Union Catalog Dataset: A dataset for conducting a situational analysis, outlined in Section III-B, was constructed by harvesting Dublin core encoded metadata records from the Networked Digital Library of Theses and Dissertations (NDLTD) Union Catalog [19], [20]. 7,404,617 digital object metadata were harvested, with 7,296,562 of them constituting the final dataset, after preprocessing.

3) Dataset #3: UCT CS Document Archive Dataset: A dataset for validating the model was constructed by harvesting Dublin Core encoded metadata records from a Computer Science subject repository (CS@ UCT archive) that is hosted by the Department of Computer Science at The University

[1] http://export.arxiv.org/oai2?verb=ListRecords&metadataPrefix=oai_dc&set=cs

of Cape Town [21]. UNZA's IR has very few Computer Science generated digital objects and as such, it was essential to identify an alternative IR. A total of 1,045 digital object metadata were harvested using the OAI-PMH protocol, with 995 comprising the final dataset, after applying traditional preprocessing operations to remove duplicates, stopwords, punctuations and, additionally, apply stemming. Furthermore, all digital objects with missing titles and abstracts were removed from the dataset.

Faculty and postgraduate students self-archive digital objects into the CS@ UCT archive. More importantly, however, the 2012 ACM CCS concepts are used as the primary controlled vocabulary set. Owing to the fact that the arXiv CoRR dataset described in Section III-A1 uses the 1998 ACM CCS concepts, the CS@ UCT archive dataset was used to compare the distribution of subject classes in order to demonstrate the effectiveness of the multi-label classification model implemented, as described in Section III-E2.

4) Dataset #4: UNZA IR Dataset: A dataset for conducting a situational analysis, outlined in Section III-B, was constructed by harvesting Dublin Core encoded metadata records from UNZA's IR. The 'identifier' and 'subject' Dublin Core elements were used to assess the distribution of manually subject tags. A total of 5,440 metadata records were harvested, with 4,802 constituting the final dataset after basic preprocessing.

B. Situational Analysis

1) Empirical Analysis of UNZA IR: Digital objects ingested into UNZA's IR can be broadly classified into two groups: faculty produced scholarly output—pre-print and post-print versions of peer-reviewed publications—and student produced scholarly output—Electronic Theses and Dissertations.

Ingestion of faculty produced scholarly output is not into UNZA's repository is not consistent, in part due to the lack of availability of an IR policy. However, ETDs are routinely ingested into the IR.

Digital object structural and Dublin Core encoded descriptive metadata, associated with ETDs, were thus harvested from UNZA's IR using the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH) [22]. Specifically, the ListRecord verb[2] was used in conjunction with the ListSets verb[3].

The structural metadata—Lines 7–8 in Listing 1—was required to identify the subject domains associated with the ETDs, while the descriptive metadata was required to identify subjects—Lines 17–19 in Listing 1— associated with the ETDs when ingested into the IR.

2) Empirical Analysis of Selected Portals: While the focus of this study was on UNZA's IR, in order to demonstrate the severity of the problem, a basic analysis of two additional scholarly portals was conducted. A reasonably sized Computer Science subject repository, hosted by the Department of Computer Science at The University of Cape Town was analysed in order to highlight the lack of comprehensive usage of subject controlled vocabularies. In addition, a large scale portal, the NDLTD Union Catalog was analysed in order to demonstrate the implications of lack of comprehensive use of subject controlled vocabularies on a global scale.

3) Digital Object Ingestion Workflow: In order to understand how digital objects are ingested into UNZA's IR, a focus group discussion was conducted with two Library members of staff that are tasked with preparing digital object metadata and ingestion of digital objects into the repository.

Listing 1: A sample ETD metadata record harvesting using the OAI-PMH protocol ListRecords verb.

```
1  <record>
2   <header>
3    <identifier>
4     oai:dspace.unza.zm:123456789/6413
5    </identifier>
6    <datestamp>2020-09-21T10:38:06Z</datestamp>
7    <setSpec>com_123456789_18</setSpec>
8    <setSpec>col_123456789_84</setSpec>
9   </header>
10  <metadata>
11   <oai_dc:dc>
12    <dc:title>
13     Automation of the grain purchasing Process for'
14     Zambias food reserve Agency
15    </dc:title>
16    <dc:creator>Simukanga, Alinani</dc:creator>
17    <dc:subject>Agricultural informatics</dc:subject>
18    <dc:subject>Agriculture-Data processing.</dc:subject>
19    <dc:subject>Agricultural innovations.</dc:subject>
20    <dc:description>
21     [...]
22     The aim of this work is to automate the processes of
23     FRA, FISP and the Cooperatives Society operate, with
24     a specific focus on the farmer registry and the grain
25     marketing process.
26     [...]
27    </dc:description>
28    <dc:date>2020-09-21T10:38:03Z</dc:date>
29    <dc:date>2020-09-21T10:38:03Z</dc:date>
30    <dc:date>2019</dc:date>
31    <dc:type>Thesis</dc:type>
32    <dc:identifier>
33     http://dspace.unza.zm/handle/123456789/6413
34    </dc:identifier>
35    <dc:language>en</dc:language>
36    <dc:format>application/pdf</dc:format>
37    <dc:publisher>University of Zambia</dc:publisher>
38   </oai_dc:dc>
39  </metadata>
40 </record>
```

C. Identification of Appropriate Controlled Vocabularies

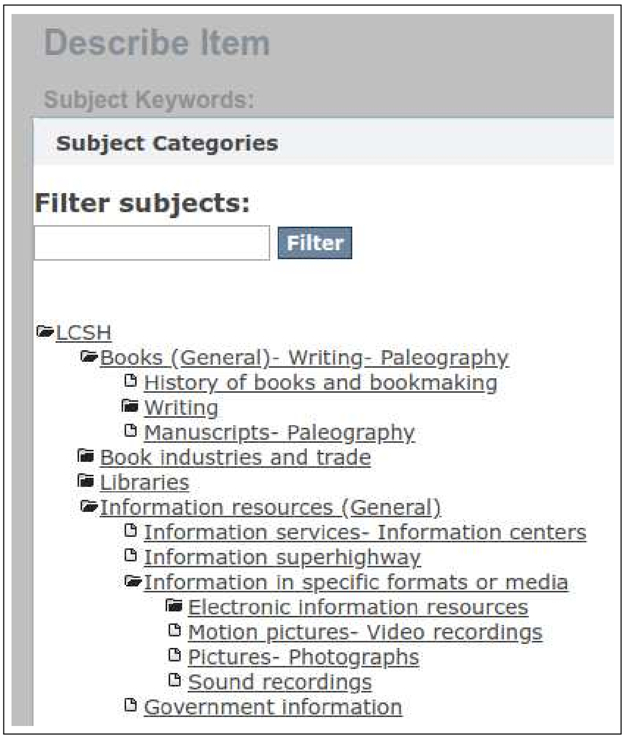
Seven faculty staff, from UNZA, were purposively sampled in order to elicit information about possible subject controlled vocabularies associated with the various disciplines at UNZA. Semi-structured face-to-face interview sessions were then conducted with each of the individual faculty staff. The interview sessions were recorded and, additionally, notes taken during the sessions.

D. Usability of IRs Integrated With Controlled Vocabularies

In order to empirically demonstrate the usability effect of integrating IRs with subject controlled vocabularies, a controlled experiment was conducted by comparing a baseline IR setup without a controlled vocabularies and a control IR integrated

[2] http://dspace.unza.zm/oai/request?verb=ListRecords&metadataPrefix=oai_dc

[3] http://dspace.unza.zm/oai/request?verb=ListSets


Describe Item

Subject Keywords:

Subject Categories

Filter subjects:

- LCSH
  - Books (General)- Writing- Paleography
    - History of books and bookmaking
    - Writing
    - Manuscripts- Paleography
  - Book industries and trade
  - Libraries
  - Information resources (General)
    - Information services- Information centers
    - Information superhighway
    - Information in specific formats or media
      - Electronic information resources
      - Motion pictures- Video recordings
      - Pictures- Photographs
      - Sound recordings
    - Government information

Fig. 4: A screenshot showing the integration of LCSH vocabulary in the intervention IR used for experimentation.

with the LCSH vocabulary set, as shown in Figure 4. Both repositories were setup using DSpace 6.x, with the intervention IR integrated with controlled vocabularies using hierarchical controlled LCSH vocabularies [23].

1) Prototype Institutional Repository Platforms: Two prototype DSpace-powered IRs were installed, setup and configured to be used to conduct the experiment. A baseline IR was setup without integrating it with controlled vocabularies, while the control IR was integrated with LCSH controlled vocabulary set.

2) Experimental Design: A within subject experiment was designed, using random experiment blocks. 50 undergraduate students were randomly sampled from the Dept. Library and Information Science at UNZA, to participate in the study. Each of the 50 participants ingested a digital object, using the two prototype IR that were setup. In each instance, participants filled out a System Usability Score (SUS) questionnaire upon successful ingestion of the digital object.

E. Multi-Label Subject Classifier

A multi-label classifier was implemented by taking advantage of transfer learning, with the model implemented using data from an external repository and, subsequently applied to new observations in UNZA's IR.

1) Data Preparation: As mentioned in Section III-A1, the arXiv CoRR dataset was used during experimentation of the multi-label classification model. The data attributes of the arXiv dataset were used as follows:

- Identifier—A unique identifier for uniquely identifying each of the arXiv digital objects harvested.
- Title—The arXiv digital object publication title, used to extract text input features.
- Description—The arXiv digital object abstract, used to extract text input features.
- Subject—The arXiv-specific digital object subjects [24] and 1998 ACM Computing Classification System (CCS) concepts [25], used as labels.

2) Model Implementation: The model features were extracted from the digital object publication titles and abstracts, with subsequent transformation of the input features done using CountVectorizer and TFIDFVectorizer. The model was implemented using the scikit-multilearn Python library [26]. Binary Relevance and Classifier Chains approaches to multi-label classification were used, in conjunction with estimators—Random Forest and Naive Bayes (Multinomial)—popularly used for text classification. Section IV-D discusses experimental results conducted to experimentally evaluate the effectiveness of the modal features and transformations used.

3) Experimental Design: All experiments were performed on a standalone LENOVO® IdeaPad 320, with an Intel® Core™ i7-8550U (CPU @ 1.80GHz), using 12 GB RAM, and running Ubuntu 18.04.3 LTS[4].

Training and testing datasets were created using the hold-out method built within the scikit-multilearn Python library, with $70\,\%$ of each dataset used for training and the remaining $30\,\%$ for testing.

It is essential to take in account multiple and contrasting metrics measures because of the additional degree of freedom that multi-label introduces and as such, the metrics used to measure the performance for multi-label classification are usually different from those used in binary and multi-class problems. Traditional multi-label classification metrics cited in literature [27]—F1 score, Jaccard Score Similarities and Hamming Loss metrics—were used to evaluate the model.

In order to determine the combination of factors that yield the best results, experimentation involved varying the following aspects:

- Input features—Title, Abstract and a combination of the two: Title+Abstract
- Text transformation techniques—CounterVectorizer and TFIDFVectorizer—used on input features and their corresponding parameters
- Multi-label classification approaches—Binary Relevance and Classifier Chains
- Estimators—Random Forest and Naive Bayes (Multinomial)

Experimentation involved measuring evaluation metrics by varying the experiment factors and aspects mention above. In addition, a validation exercise was conducted, that involved comparing the distribution manually assigned 2012 ACM CCS concepts with the 1998 ACM CCS concepts predicted by

[4] http://releases.ubuntu.com/18.04.3

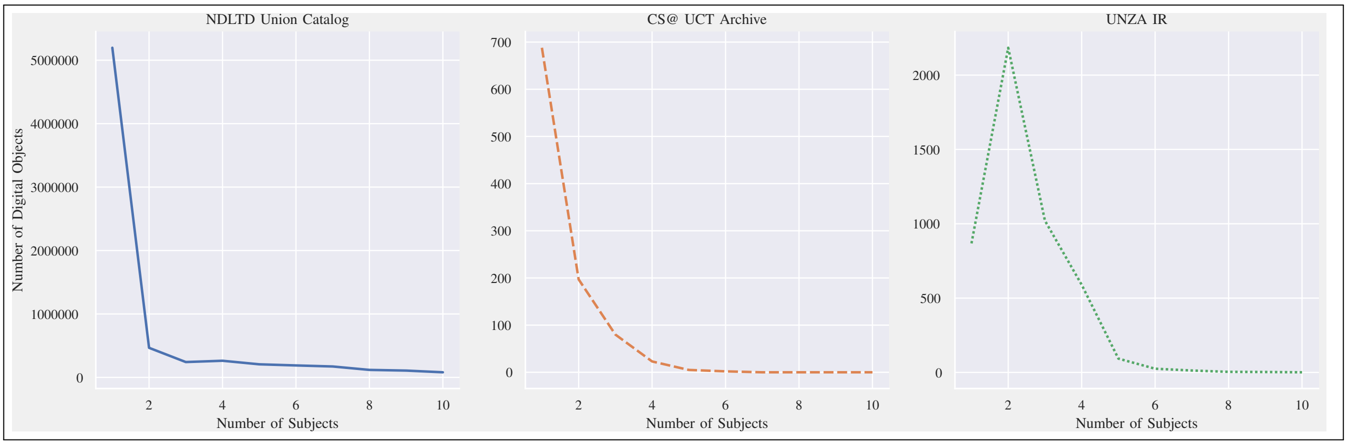


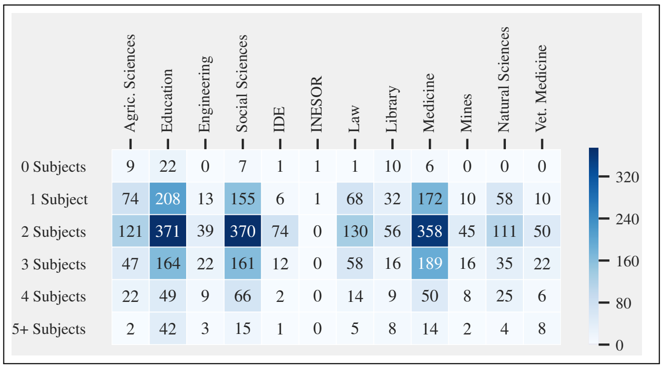
Fig. 5: Digital objects in most digital libraries are typically associated with very few subject classes.

the model, in the CS@ UCT archive dataset described in Section III-A3. Section IV-D presents and discusses the results.

IV. Results and Discussion

A. Situational Analysis

1) Analysis 1. Metadata Preparation Workflow: The focus group session was conducted with two Library staff—the IR Manager and his assistant. The Library staff highlighted that as part of the metadata preparation process, digital objects to are catalogued and subject headings copied from an online public access catalog[5]. This process presents a number of challenges as it is time consuming and error prone. Integration of the IR with appropriate subject headings would not only help address these challenges, but also ensure effective self-archiving [28] of digital objects into the repository.



	Agric. Sciences	Education	Engineering	Social Sciences	IDE	INESOR	Law	Library	Medicine	Mines	Natural Sciences	Vet. Medicine
0 Subjects	9	22	0	7	1	1	1	10	6	0	0	0
1 Subject	74	208	13	155	6	1	68	32	172	10	58	10
2 Subjects	121	371	39	370	74	0	130	56	358	45	111	50
3 Subjects	47	164	22	161	12	0	58	16	189	16	35	22
4 Subjects	22	49	9	66	2	0	14	9	50	8	25	6
5+ Subjects	2	42	3	15	1	0	5	8	14	2	4	8

Fig. 6: A heatmap showing the average number of subject/topic specified for scholarly publications for the various domains at UNZA.

2) Analysis 2. Subject Headings for UNZA IR: 5,438 digital object metadata were harvested from UNZA's IR. Scholarly output associated with the 13 faculties at UNZA were then filtered, resulting in a total of 3,638 metadata records used in this analysis. The digital object metadata records were analysed to determine the usage of subject headings.

Of the 3,638 digital objects analysed, 22.2 % were assigned a single subject heading, 47.4 % were assigned two subject headings, 24.4 % were assigned three subject headings, 7.1 % were assigned four subject headings and 2.9 % were assigned more than five subject headings. Figure 6 shows a heatmap of number of subjects assigned to scholarly research output. In the heatmap, it is evident that a significant proportion of scholarly publications, in each of the faculties, are tagged with two subjects. While the number of subjects used to classify a publication is dependent on how the contents of the publication, interviews conducted with Library staff and UNZA revealed that an internal policy requires that digital objects be associated with at least two or three subject headings. However, in an ideal case, it is desirable to associate a publication with more tags to facilitate effective discoverability of related content.

A total of 7,244 subject headings were associated with the data analysed. Of the total subject headings, 92.1 % were associated to a single publication, 6.1 % to two publications, 0.9 % to three publications, 0.1 % to four publications and 0.5 % to more than five publications. Figure 7 shows a heatmap of subject usage patterns for scholarly publications by faculty. The heatmap showcases the frequency of usage of subject headings. For instance, 1,402 subjects have been assigned to only a single publication for content ingested into "Medicine" collections, whereas only 10 subjects are associated with five or more publications. The chart indicates that lack of use of subject controlled vocabularies due to the significantly large proportion of subjects being associated with a single publication. The sparing use of subjects is also shown in Figure 1.

3) Analysis 3. Subject Class Distribution in Portals: Figure 5 show the distribution of subject classes in UNZA's IR, the CS@ UCT Archive and the NDLTD Union Catalog. A common characteristic of the three portals is that most of the digital objects are associated with less than two subjects.

[5] http://koha.unza.zm:4480

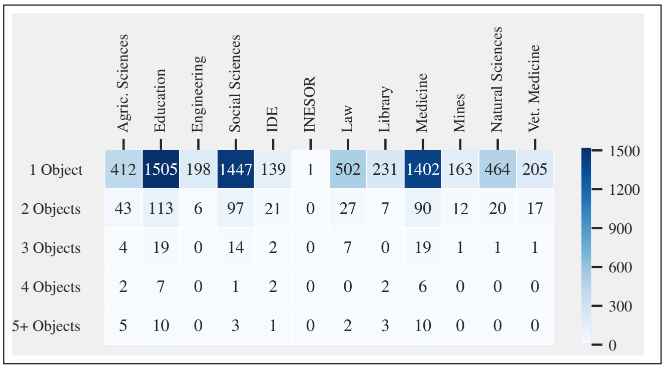


Fig. 7: A heatmap showing the number of subjects/topics associated with different thresholds of scholarly publications for the various domains at UNZA.

While the distribution is especially problematic for portals like UNZA's IR, largely due to low self-archiving practices, the picture is equally as bad for portals such as CS@ UCT Archive, where self-archiving is practised significantly; this is largely due to the fact that authors that self-archive tend not to comprehensively provide relevant subject classes to their publications. Large scale downstream services such as the NDLTD Union Catalog have worse off distributions because the content archiving in such portals is harvested from IRs that have problematic self-archiving practices.

The distributions shown in Section IV-A3 support the premise of this paper: the problem with subject classes is best addressed at the source. Incidentally, it is possible to introduce interventions that can be applied to downstream services, however, it is more effective to work with source portals.

B. Domain-Specific Subject Headings

Seven faculty staff were interviewed in order to elicit subject headings used in their various disciplines. Table II shows a summary of the major outcomes from the interview sessions. Most of the interviewees were familiar with the concept of controlled vocabularies, however, only a few were knowledgeable about the specific subject headings used in their respective domains.

While the majority of faculty staff are unaware of subject headings used in their disciplines, a question included in the interview guide required that they specify popular academic databases used in their domains. The academic databases specified can be used as a basis for identify appropriate subject headings. For instance, the the widely used ACM Computing Classification System [11] could be used to generate subject headings for scholarly research output produced by computing oriented faculties and/or departments. Using academic databases as a basis for adopting subject headings could also potentially enhance the interoperability of IRs with external downstream services that automatically harvest IR metadata.

TABLE II: Summary of results from interviews conducted as part of an exploratory study to understand subject headings used in various domains at UNZA.

Participant	Academic Databases	Subjects
FS–1	PubMed · Science Direct · Google Scholar · Mendeley	MeSH
FS–2	SCOPUS · ERIC · SCINAPSE · EBSCO HOST · PROQUEST	Not aware
FS–3	Academia.edu · Zambia Library Journals · Google Scholar · UNZA IR	Not aware
FS–4	IEEE · ELSEVIER	Not aware
FS–5	ResearchGate · Google Scholar · Academia.edu	None
FS–6	Academia.edu · Mendeley · ResearchGate · JSTOR · Google Scholar	SEARS List
FS–7	IEEE · Explorer · ResearchGate	None

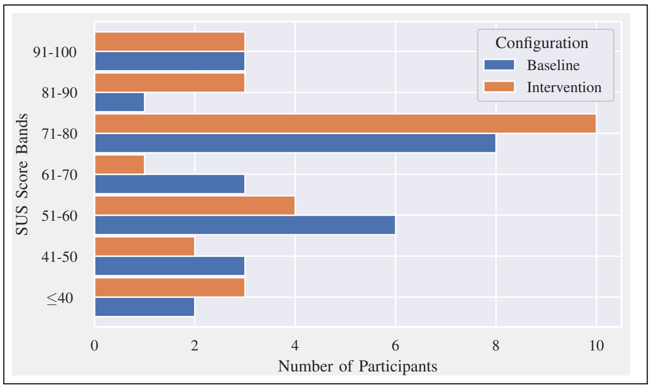


Fig. 8: SUS acceptability and adjective rating scores [29] for the baseline and intervention IRs.

C. Comparative Usability Study

1) System Usability Scale Scores: The SUS scores corresponding to responses from each participants were computed for each of the two IR platforms: baseline and intervention. The SUS scores were calculated using the standard method of that takes into account all of the 10 SUS questionnaire items [30].

The average SUS scores for the baseline and intervention IRs were 66.2 and 68.9, respectively. While both SUS scores are rated "OK" on the acceptability and adjective rating score [29], the average SUS score for the intervention is noticeably higher, as shown in Figure 8. The differences in the SUS scores is further supported by the positive responses associated with the intervention IR, outlined in Section IV-C2.

However, a paired t-test indicates no significant difference in the mean scores (p = 0.82). Furthermore, Factorial ANOVA tests conducted to determine effects of demographic factors also suggest no significant main effect as a result of "Prior Knowledge of Controlled Vocabularies" ($F_{1,48}$ = 0.041, p = 0.84), "Experience With ICTs" ($F_{2,46}$ = 1.13, p = 0.33), "Participants' Year of Study" ($F_{2,46}$ = 0.77, p = 0.47) and "Gender" ($F_{1,48}$ = 1.61, p = 0.21), *2) Participants' Comments:* Participants were also required to provide open ended comments, relative to their experiences using the two platforms. The vast majority of the comments were related to the use of controlled vocabularies and, for the most part, positive.

> *"It was easy to work around the repository with subject controlled vocabulary."* [Participant #6]

> *"The second method is more easier to work with"* [Participant #14]

> *"It was easy because you have to just click and the keywords will be provided which is less time consuming"* [Participant #18]

> *"The arrangement is well organised and kind of easy to use"* [Participant #22]

> *"I did not like typing in the subject keywords."* [Participant #26]

The participants' comments, in part, help explain the higher SUS mean score for the intervention IR, outlined in Section IV-C1.

TABLE III: Model versus Manual generated subjects.

<table>
<tr><th colspan="2">Digital Object Title</th></tr>
<tr><td colspan="2">Automation of the grain purchasing Process for Zambia's food reserve Agency</td></tr>
<tr><th colspan="2">Digital Object Abstract</th></tr>
<tr><td colspan="2">Issues of food security, post-harvest losses, lack of a national farmer database and proper grain inventory system have plagued the Ministry of Agriculture for years. The lack of requisite tools has made the management of the sector a difficult task. This has seen an increase in the number of ghost farmers benefiting from the Farmer Input Support Programme (FISP). The aim of this work is to automate the processes of FRA, FISP and the Cooperatives Society operate, with a specific focus on the farmer registry and the grain marketing process. The objectives are as follows: Map the current business processes of FISP and FRA; Develop a model of objective 1 using cloud and mobile computing technologies; Develop a system prototype that integrates farmers spatial data and mobile computing based on the model in objective 2; and integrate multi-factor authentication into the prototype in objective 3. To meet objective 1, a baseline study was conducted at the FRA depots in Chongwe and Mumbwa. The information gathered from this and from various documents provided informed the development of the model specified in objective 2. Various web technologies such as PHP, Java and PostgreSQL were employed to achieve objective 3. Multi-factor authentication was implemented as an added security feature when interfacing with the mobile application for the final objective.</td></tr>
<tr><th colspan="2">Digital Object Subjects</th></tr>
<tr><th>Manual Subjects</th><th>Model Generated Subjects</th></tr>
<tr><td>Agricultural informatics · Agriculture–Data processing · Agricultural innovations</td><td>C.2.4 · Computer Science - Artificial Intelligence · Computer Science - Computation and Language · Computer Science - General Literature · Computer Science - Human-Computer Interaction · D.2.11 · F.1.1 · H.3.4 · H.3.5 · H.5.2</td></tr>
</table>

D. ArXiv CoRR Subject Classification

Table III illustrates the results obtained when the model is applied to a sample digital object in UNZA's IR. The "Manual Subjects" column has subjects manually prepared when the digital object was being ingested into the IR, while the "Model Generated Subjects" column represents the subjects predicted by the model. The ACM CCS concepts are presented with their code for brevity, while the arXiv subjects are present with their textual descriptions.

Table IV depicts the summary of performance results in terms of F1-score, hamming loss and Jaccard Similarities accuracy for two multi-label classification approaches used, including the estimators used and data transformation technique.

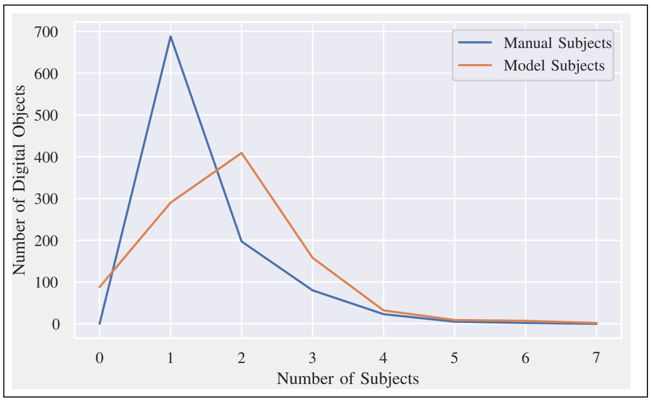


Fig. 9: Distribution of Model and Manual generated subjects.

1) Analysis 1. Transfer Learning: The implemented model was applied to digital objects in UNZA's IR, with very promising results. First, the predicted labels are specific to the domain in question: Computer Science, as opposed to the previous subjects that are manually determined by staff that ingest digital objects into the IR. More significantly, however, randomly inspected objects were noted to have been automatically associated with relevant subjects. Table III shows a comparison of manual subjects previously associated to the sample object, also shown in Figure 2, and subjects automatically predicted by the model. Only three (3) non subject-specific subjects were annually assigned to the digital object, while a total of six ACM CCS subjects and four arXiv subjects were automatically predicted using the model.

Figure 9 shows a subject classes distribution of manually assigned subjects and model generated subjects in the CS@UCT archive. It is evident from the plot that a significant proportion of digital objects only have a single subject associated with them, a common occurrence in IRs that implement self-archiving.

The automatic prediction of subjects has the obvious benefit of ensuring that a consistent subset of domain-specific subjects are associated with related digital objects. Furthermore, this technique reduces the human-centric manual processes involved in when associating metadata to digital objects, drastically reducing the time spent preparing metadata and potential errors introduced when preparing metadata.

2) Analysis 2. Input Features: As earlier mentioned in Section III-E, three input features were used during experimentation 'Title', 'Abstract' and 'Title+Abstract'. Expectantly,

TABLE IV: Experimental results for arXiv subject classes multi-label classification model.

Binary Relevance										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
MultinomialNB	TF	0.305	0.006	0.192	0.214	0.037	0.207	0.203	0.041	0.196
	TF-IDF	0.236	0.006	0.148	0.398	0.005	0.271	0.420	0.005	0.290
RandomForest	TF	0.317	0.006	0.211	0.416	0.005	0.292	0.430	0.005	0.305
	TF-IDF	0.314	0.006	0.210	0.418	0.005	0.295	0.435	0.005	0.310
SGDClassifier	TF	0.279	0.006	0.18	0.515	0.005	0.390	0.526	0.005	0.407
	TF-IDF	0.282	0.006	0.183	0.476	0.005	0.351	0.496	0.005	0.369
Classifier Chains										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
MultinomialNB	TF	0.060	0.055	0.190	0.030	0.338	0.130	0.030	0.347	0.123
	TF-IDF	0.090	0.027	0.177	0.086	0.053	0.282	0.087	0.055	0.294
RandomForest	TF	0.287	0.009	0.238	0.428	0.005	0.305	0.441	0.005	0.318
	TF-IDF	0.289	0.009	0.239	0.424	0.005	0.301	0.444	0.005	0.320
SGDClassifier	TF	0.312	0.006	0.216	0.527	0.005	0.420	0.520	0.006	0.414
	TF-IDF	0.310	0.006	0.214	0.523	0.005	0.413	**0.540**	0.005	**0.431**
One-Versus-Rest										
		Title			Abstract			Title + Abstract		
		F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score	F1 Score	Hamming Loss	Jaccard Score
MultinomialNB	TF	0.305	0.006	0.192	0.214	0.037	0.207	0.203	0.041	0.196
	TF-IDF	0.236	0.006	0.148	0.398	0.005	0.271	0.420	0.005	0.290
RandomForest	TF	0.317	0.006	0.212	0.414	0.005	0.291	0.435	0.005	0.310
	TF-IDF	0.315	0.006	0.210	0.414	0.005	0.290	0.432	0.005	0.306
SGDClassifier	TF	0.279	0.006	0.180	0.488	0.005	0.365	0.520	0.005	0.400
	TF-IDF	0.282	0.006	0.183	0.479	0.005	0.354	0.497	0.005	0.371

using a combination of titles and abstracts—Title+Abstract—results in more effective models than using the Title or Abstract features in isolation. It was also noticed that overall transforming the input features using TFIDFVectorizer result in better performing models than using CounterVectorizer. This is the case for the best performing approach and estimator, SGDClassifier using Classifier Chains, where the F1 Score was 0.540 and the Jaccard Similarities Score was 0.431. Incidentally, this is also the case for most of the other approaches and estimators

The ‘Title+Abstract‘ feature Expectantly results in better performance seeing as combining two features results in a more enriched feature. New enriched feature-sets can potentially be created by augmenting metadata elements such as ‘Keywords‘, which are sometimes provided alongside traditional ones like ‘Title‘ and ‘Abstract‘.

3) Analysis 3. Approach and Estimators: Of the three multi-label approaches used, Classifier Chains yielded the best results, with an F1 score of 0.540; Hamming Loss value of 0.005 and Jaccard Similarities Score of 0.431. The next best performing approach was One-Versus-Rest, using SGDClassifier, and finally, One-Versus-Rest using SGDClassifier. In all these instances, the ‘Title+Abstract‘ yielded the best result.

With the F1 Score results obtained, it makes logical sense that IR interfaces incorporate the automatic generation of subject classes in such a manner that they are complemented with human effort. For instance, an end-user can be presented with an interface that enable them to add and/or remove subject classes automatically generated.

V. Conclusion

This paper outlines a case study conducted to investigate the implications of integrating subject controlled vocabularies in IRs. The case study was conducted in three phases. First, a situational analysis—described in Section III-B—was conducted to understand how digital objects are tagged with subject headings. Secondly, an exploratory study—outlined in Section III-C—was conducted to determine domain specific subject headings for different faculties at UNZA. In addition, a usability study—outlined in Section III-D—was conducted to ascertain the impact on usability of IRs integrated with controlled subject vocabularies. Finally, a multi-label classification model for predicting ACM CCS and arXiv subject classes was presented. Experimental results of the classification model illustrate the potential effectiveness of automatically generating domain specific subjects.

Integrating IRs with subject controlled vocabularies has the benefit ensuring that IRs are usable and effective. More significantly, though, the digital objects are certain to be tagged with correct and comprehensive subject headings. The semi-automatic metadata generation approach proposed, where traditional human approaches are augmented with an automated approach align with techniques suggested by Tani et al. for addressing metadata quality issues [31].

The systematic process presented in this paper has the potential of making self-archiving more effective, since IRs would be integrated with pre-existing subject headings. This would ultimately complement machine learning techniques presented in prior work [32], further making the ingestion of digital objects into IRs more effective, less error and more comprehensive Beyond IRs, however, the automatic generation of subject classes can be applied to large scale portals such as the NDLTD Union Catalog [19], [20] that have been noted to experience metadata quality issues [33].

As part of future and on-going work, models are being implemented for other domains at UNZA and, additionally, there are plans to apply this technique on large-scale datasets such as the NDLTD Union Catalog [19].

References

[1] N. F. Foster and S. Gibbons, "Understanding faculty to improve content recruitment for institutional repositories," *D-Lib Magazine*, vol. 11, no. 1, 2005.

[2] L. Phiri, "Research Visibility in the Global South : Towards Increased Online Visibility of Scholarly Research Output in Zambia," in *Proceedings of the 2nd IEEE International Conference in Information and Communication Technologies (ICICT 2018)*, Lusaka, Zambia, 2018. [Online]. Available: http://dspace.unza.zm/handle/123456789/5723

[3] J. Riley, *Understanding Metadata: What is Metadata, and what is it For?* National Information Standards Organization (NISO), 2017.

[4] W. Y. Arms, "Information retrieval and descriptive metadata," in *Digital Libraries*. MIT press, 2001, ch. 10.

[5] I. Varlamis and I. Apostolakis, "The Present and Future of Standards for E-Learning Technologies," *Interdisciplinary Journal of e-Skills and Lifelong Learning*, vol. 2, pp. 59–76, 2006.

[6] C. Sarah, B. Jane, O. Rónán, and R. Ben, "Quality assurance for digital learning object repositories: issues for the metadata creation process," *ALT-J*, vol. 12, no. 1, pp. 5–20, 2004.

[7] S. L. Weibel, J. A. Kunze, C. Lagoze, and M. Wolf, "Dublin Core Metadata for Resource Discovery," 1998. [Online]. Available: http://www.hjp.at/doc/rfc/rfc2413.html

[8] DSpace, "Functional Overview - DSpace 6.x Documentation," 2018. [Online]. Available: https://wiki.lyrasis.org/display/DSDOC6x/Functional+Overview

[9] P. Harpring, "What are controlled vocabularies?" in *Introduction to controlled vocabularies: terminology for art, architecture, and other cultural works*. Getty Publications, 2010, ch. 2.

[10] I. Dhammi and S. Kumar, "Medical subject headings (MeSH) terms," *Indian Journal of Orthopaedics*, vol. 48, no. 5, p. 443, 2014.

[11] Association for Computing Machinery, "ACM Computing Classification System," 2012. [Online]. Available: https://dl.acm.org/ccs

[12] P. J. Rolla, "User Tags versus Subject Headings," *Library Resources & Technical Services*, vol. 53, no. 3, pp. 174–184, 2013.

[13] C. Lu, J. R. Park, and X. Hu, "User tags versus expert-assigned subject terms: A comparison of LibraryThing tags and Library of Congress Subject Headings," *Journal of Information Science*, vol. 36, no. 6, pp. 763–779, 2010.

[14] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music into emotions," *ISMIR 2008 - 9th International Conference on Music Information Retrieval*, pp. 325–330, 2008.

[15] R. Li, W. Liu, Y. Lin, H. Zhao, and C. Zhang, "An Ensemble Multilabel Classification for Disease Risk Prediction," *Journal of Healthcare Engineering*, vol. 2017, 2017.

[16] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in eu legislation," in *Proceedings of the Natural Legal Language Processing Workshop*, 2019, pp. 78–87.

[17] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[18] Cornell University, "Welcome to the Computing Research Repository (CoRR)," 2019. [Online]. Available: https://arxiv.org/corr

[19] The Networked Digital Library of Theses and Dissertations, "NDLTD Union Archive of ETD Metadata," [online] http://union.ndltd.org/portal, (Accessed April 1, 2021).

[20] ——, "Global ETD Search," [online] http://search.ndltd.org, (Accessed April 1, 2021).

[21] Department of Computer Science, School of IT, University of Cape Town, "UCT Computer Science Research Document Archive," [online] https://pubs.cs.uct.ac.za, (Accessed April 1, 2020).

[22] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner, "The Open Archives Initiative Protocol for Metadata Harvesting," 2002. [Online]. Available: http://www.openarchives.org/OAI/openarchivesprotocol.html

[23] DSpace. (2015) Authority Control of Metadata Values - DSpace 6.x Documentation. [Online]. Available: https://wiki.lyrasis.org/display/DSDOC6x/Authority+Control+of+Metadata+Values

[24] Cornell University, "Computer Science Subject Areas and Moderators," 2020. [Online]. Available: https://arxiv.org/corr/subjectclasses

[25] Association for Computing Machinery, "The ACM Computing Classification System (1998)," 2007. [Online]. Available: https://www.acm.org/publications/computing-classification-system/1998/ccs98

[26] P. Szymanski and T. Kajdanowicz, "Scikit-multilearn: a scikit-based python environment for performing multi-label classification," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 209–230, 2019.

[27] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007.

[28] J. Kim, "Faculty self-archiving: Motivations and barriers," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 9, pp. 1909–1922, sep 2010. [Online]. Available: http://doi.wiley.com/10.1002/asi.21336

[29] A. Bangor, P. Kortum, and J. Miller, "Determining what individual SUS scores mean: Adding an adjective rating scale," vol. 4, no. 3, pp. 114–123, May 2009. [Online]. Available: http://www.usabilityprofessionals.org/upa_publications/jus/2009may/JUS_Bangor_May2009.pdf

[30] J. Brooke, *SUS – A quick and dirty usability scale*. London: Taylor & Francis, 1996, ch. 21, pp. 189–195. [Online]. Available: http://www.usabilitynet.org/trump/documents/Suschapt.doc

[31] A. Tani, L. Candela, and D. Castelli, "Dealing with metadata quality: The legacy of digital library efforts," *Information Processing & Management*, vol. 49, no. 6, pp. 1194 – 1205, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306457313000526

[32] L. Phiri, "Automatic classification of digital objects for improved metadata quality of electronic theses and dissertations in institutional repositories," *International Journal of Metadata, Semantics and Ontologies*, vol. 14, no. 3, pp. 234–248, 2020.

[33] H. Suleman, "The NDLTD Union Catalog: Issues at a Global Scale," in *Proceedings of the 15^{th} International Symposium on Electronic Theses and Dissertations*. Universidad Peruana de Ciencias Aplicadas (UPC), 2012, [online] https://repositorioacademico.upc.edu.pe/handle/10757/622568 (Accessed April 1, 2021).

Appendix C

Subject Classification Script

Subject Classification Script

```
mport pandas as pd
df = pd.read_csv('unza_collection_combined.csv',encoding='latin1')
df.head()
from sklearn.preprocessing import MultiLabelBinarizer
import json
subject_new=[] #declare a list
for cell in df['subject']:
    cell=cell.replace(" ", "") #remove whitespace
    cell=cell.replace("&", "& ") #add whitespace back in for ampersands
    subject_new.append(cell.split(",")) #for each genre cell, create a list of items from the original string, using a comma as a delimeter
    #add new genre column to the dataframe
df['subject_new'] = subject_new
mlb = MultiLabelBinarizer()
binary_labels=binary_labels.sort_index(axis=1)
binary_labels.head(10).T
documents = df.merge(binary_labels, how='inner', left_index=True, right_index=True)
documents= documents.drop(columns=['subject', 'description','subject_new'])
documents.tail(7)
import seaborn as sns
import matplotlib.pyplot as plt
categories = list(binary_labels.columns.values)
ax= sns.barplot(binary_labels.sum().values, categories)
plt.title("Documents for each Subject", fontsize=24)
plt.ylabel('Subject', fontsize=18)
plt.xlabel('Number of document tagged with subject', fontsize=18)
rects = ax.patches
labels = binary_labels.sum().values
plt.show()
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=10000)
xtrain, xval, ytrain, yval = train_test_split(documents['description_new'], binary_labels, test_size=0.2, random_state=9)
xtrain_tfidf = tfidf_vectorizer.fit_transform(xtrain)
xval_tfidf = tfidf_vectorizer.transform(xval)
from sklearn.linear_model import LogisticRegression
from sklearn.multiclass import OneVsRestClassifier
from sklearn.metrics import accuracy_score
logreg = LogisticRegression()
logreg_classifier = OneVsRestClassifier(logreg)
logreg_classifier.fit(xtrain_tfidf, ytrain)
predictions = logreg_classifier.predict(xval_tfidf)
from sklearn.metrics import accuracy_score
print("Accuracy score for Logistic Regression:")
print(accuracy_score(yval, predictions))
from sklearn.metrics import hamming_loss
hamming_loss(yval, predictions)
from sklearn.metrics import classification_report
```

```
print(classification_report(yval, predictions, target_names=binary_labels.columns))
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import GaussianNB
classifier = BinaryRelevance(GaussianNB())
classifier.fit(xtrain_tfidf, ytrain)
predictions = classifier.predict(xval_tfidf)
print("Accuracy score for Gaussian Naive Bayes:")
print(accuracy_score(yval, predictions))
print("Individual subject predictions:")
print(classification_report(yval, predictions, target_names=binary_labels.columns))
from sklearn.metrics import hamming_loss
hamming_loss(yval, predictions)
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.naive_bayes import MultinomialNB
classifier = BinaryRelevance(MultinomialNB())
classifier.fit(xtrain_tfidf, ytrain)
predictions = classifier.predict(xval_tfidf)
print("Accuracy score for MultinomialNB:")
print(accuracy_score(yval, predictions))
print("Individual subject predictions:")
print(classification_report(yval, predictions, target_names=binary_labels.columns))
hamming_loss(yval, predictions)
```

Appendix D

Collection Classification Script

Collection Classification Script

```
import pandas as pd

df = pd.read_csv('unza_collection_combined.csv',encoding='latin1')

df.head()

df = df[pd.notnull(df['description'])]

df.info()

col = [ 'description','collection']

df = df[col]

df.columns

df.columns = [ 'description','collection']

df['collection_id'] = df['collection'].factorize()[0]

from io import StringIO

collection_id_df = df[['collection', 'collection_id']].drop_duplicates().sort_values('collection_id')

collection_to_id = dict(collection_id_df.values)

id_to_collection = dict(collection_id_df[['collection_id', 'collection']].values)

df['collection_id']=df['collection'].factorize()[0]

import matplotlib.pyplot as plt

fig = plt.figure(figsize=(10,8))

df.groupby('collection').description.count().plot.bar(ylim=0)

plt.show()

from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(sublinear_tf=True, min_df=5, norm='l2', encoding='latin-1', ngram_range=(1, 2), stop_words='english')

features = tfidf.fit_transform(df.description).toarray()

labels = df.collection_id

features.shape

from sklearn.feature_selection import chi2

import numpy as np
```

```
N = 2
for collection, collection_id in sorted(collection_to_id.items()):
  features_chi2 = chi2(features, labels == collection_id)
  indices = np.argsort(features_chi2[0])
  feature_names = np.array(tfidf.get_feature_names())[indices]
  unigrams = [v for v in feature_names if len(v.split(' ')) == 1]
  bigrams = [v for v in feature_names if len(v.split(' ')) == 2]
  print("# '{}':".format(collection))
  print("  . Most correlated unigrams:\n       . {}".format('\n       . '.join(unigrams[-N:])))
  print("  . Most correlated bigrams:\n       . {}".format('\n       . '.join(bigrams[-N:])))
  from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['description'], df['collection'], random_state = 0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
clf = MultinomialNB().fit(X_train_tfidf, y_train)
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.naive_bayes import MultinomialNB
X_train, X_test, y_train, y_test = train_test_split(df['description'], df['collection'], random_state = 0)
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
tfidf_transformer = TfidfTransformer()
X_train_tfidf = tfidf_transformer.fit_transform(X_train_counts)
```

```
clf = MultinomialNB().fit(X_train_tfidf, y_train)
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.naive_bayes import MultinomialNB
from sklearn.svm import LinearSVC
from sklearn.model_selection import cross_val_score
models = [
  RandomForestClassifier(n_estimators=200, max_depth=3, random_state=0),
  LinearSVC(),
  MultinomialNB(),
  LogisticRegression(random_state=0),
]
CV = 5
cv_df = pd.DataFrame(index=range(CV * len(models)))
entries = []
for model in models:
 model_name = model.__class__.__name__
 accuracies = cross_val_score(model, features, labels, scoring='accuracy', cv=CV)
 for fold_idx, accuracy in enumerate(accuracies):
  entries.append((model_name, fold_idx, accuracy))
cv_df = pd.DataFrame(entries, columns=['model_name', 'fold_idx', 'accuracy'])
import seaborn as sns
sns.boxplot(x='model_name', y='accuracy', data=cv_df)
sns.stripplot(x='model_name', y='accuracy', data=cv_df,
       size=8, jitter=True, edgecolor="gray", linewidth=2)
plt.show()
cv_df.groupby('model_name').accuracy.mean()
from sklearn.model_selection import train_test_split
```

```
model = LinearSVC()

X_train, X_test, y_train, y_test, indices_train, indices_test = train_test_split(features, labels, df.index, test_size=0.33, random_state=0)

model.fit(X_train, y_train)

y_pred = model.predict(X_test)

from sklearn.metrics import confusion_matrix

conf_mat = confusion_matrix(y_test, y_pred)

fig, ax = plt.subplots(figsize=(10,8))

sns.heatmap(conf_mat, annot=True, fmt='d',

            xticklabels=collection_id_df.collection.values, yticklabels=collection_id_df.collection.values)

plt.ylabel('Actual')

plt.xlabel('Predicted')

plt.show()

from IPython.display import display

for predicted in collection_id_df.collection_id:

  for actual in collection_id_df.collection_id:

    if predicted != actual and conf_mat[actual, predicted] >= 6:

      print("'{}' predicted as '{}' : {} examples.".format(id_to_collection[actual], id_to_collection[predicted], conf_mat[actual, predicted]))

      display(df.loc[indices_test[(y_test == actual) & (y_pred == predicted)]][['collection', 'description']])

      print('')

model.fit(features, labels)

from sklearn.feature_selection import chi2

N = 2

for collection, collection_id in sorted(collection_to_id.items()):

  indices = np.argsort(model.coef_[collection_id])

  feature_names = np.array(tfidf.get_feature_names())[indices]

  unigrams = [v for v in reversed(feature_names) if len(v.split(' ')) == 1][:N]

  bigrams = [v for v in reversed(feature_names) if len(v.split(' ')) == 2][:N]
  print("# '{}':".format(collection))

  print("  . Top unigrams:\n       . {}".format('\n       . '.join(unigrams)))

  print("  . Top bigrams:\n       . {}".format('\n       . '.join(bigrams)))

 from sklearn import metrics

print(metrics.classification_report(y_test, y_pred,

                                    target_names=df['collection'].unique()))
```

Appendix E

Document Type Classification Model

DOCUMENT TYPE CLASSIFICATION SCRIPT

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_files
DATA_DIR ="./DatasetLu/"
data = load_files(DATA_DIR, encoding="utf-8", decode_error="replace")
# calculate count of each category
labels, counts = np.unique(data.target, return_counts=True)
# convert data.target_names to np array for fancy indexing
labels_str = np.array(data.target_names)[labels]
print(dict(zip(labels_str, counts)))
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target)
list(t[:80] for t in X_train[:10])
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(stop_words="english", max_features=1000, decode_error="ignore")
vectorizer.fit(X_train)
vectorizer.fit(X_train)
X_train_vectorized = vectorizer.transform(X_train)
from sklearn.linear_model import SGDClassifier
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import cross_val_score
 # start with the classic
# with either pure counts or tfidf features
sgd = Pipeline([
    ("count vectorizer", CountVectorizer(stop_words="english", max_features=3000)),
    ("sgd", SGDClassifier(loss="modified_huber"))
])
sgd_tfidf = Pipeline([
    ("tfidf_vectorizer", TfidfVectorizer(stop_words="english", max_features=3000)),
    ("sgd", SGDClassifier(loss="modified_huber"))
])
svc = Pipeline([
    ("count_vectorizer", CountVectorizer(stop_words="english", max_features=3000)),
    ("linear svc", SVC(kernel="linear"))
])
svc_tfidf = Pipeline([
    ("tfidf_vectorizer", TfidfVectorizer(stop_words="english", max_features=3000)),
    ("linear svc", SVC(kernel="linear"))
])
all_models = [
("sgd", sgd),
("sgd_tfidf", sgd_tfidf),
("svc", svc),
("svc_tfidf", svc_tfidf),
]
 unsorted_scores = [(name, cross_val_score(model, X_train, y_train, cv=2).mean()) for name, model in all_models]
scores = sorted(unsorted_scores, key=lambda x: -x[1])
print(scores)
model = svc_tfidf
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

Bibliography

[1] E. S. Gbaje and M. F. Mohammed, "Long-term accessibility and re-use of institutional repository contents of some selected academic institutions in nigeria," 2017.

[2] K. Borkar and N. Dhande, "Efficient text classification of 20 newsgroup dataset using classification algorithm," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 6, pp. 1236–1240, 2017, ISSN: 2321-8169.

[3] S. Harnad, "The self-archiving initiative," *Nature*, vol. 410, no. 6832, pp. 1024–1025, 2001.

[4] A. B. Zhang and D. Gourley, *Creating Digital Collections: A Practical Guide*. Neal-Schuman Publishers, 2009.

[5] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in eu legislation," Jun. 2019.

[6] M. Dobreva, Y. Kim, and S. Ross, "Designing an Automated Prototype Tool for Preservation Quality Metadata Extraction for Ingest into Digital Repository," no. February 2015, 2008.

[7] C. A. Lynch, "Institutional Repositories: Essential Infrastructure For Scholarship In The Digital Age," *portal: Libraries and the Academy*, 2003, ISSN: 1530-7131. DOI: 10.1353/pla.2003.0039.

[8] C. A. Lynch and J. K. Lippincott, "Institutional repository strategies and implementation: A brief overview," *D-Lib Magazine*, vol. 11, no. 9, 2005. DOI: 10.1045/september2005-lippincott.

[9] W. Y. Arms, R. L. Larsen, S. Dobratz, and K. W. Smith, "Building a digital library infrastructure," *Communications of the ACM*, vol. 40, no. 2, pp. 33–41, 1997. DOI: 10.1145/253671.253674.

[10] N. F. Foster and S. Gibbons, "Understanding faculty to improve content recruitment for institutional repositories," *D-Lib Magazine*, vol. 11, no. 1, 2005, ISSN: 10829873. DOI: 10.1045/january2005-foster.

[11] M. S. M. S. Sadiku, "A Brief Introduction to Data Mining and Analysis," *eUROPEAN sCIENTIFIC jOURNAL*, vol. 11, no. 573, pp. 1–3, 2005.

[12] SAS, *Data Mining Using SAS Enterprise Miner: A Case Study Approach, Third Edition*. Cary, NC: SAS Institute Inc., 2019.

[13] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. P. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0: Step-by-step data mining guide," 2000.

[14] H. J. Watson, D. G. Grecich, C. Shearer, L. Moss, S. Adelman, K. Hammer, and S. A. Herdlein, "J OURNAL Statement of Purpose E-Business and the New Demands on Data E-Commerce Places on Data Warehousing Technology WAREHOUSING," vol. 5, no. 4, 2000.

[15] E. Alpaydin, *Introduction to Machine Learning*. Cambridge, MA: The MIT Press, 2010, ISBN: 978-0-262-01243-0.

[16] I. Kavakiotis, O. Tsave, A. Salifoglou, N. Maglaveras, I. Vlahavas, and I. Chouvarda, "Machine Learning and Data Mining Methods in Diabetes Research," *Computational and Structural Biotechnology Journal*, vol. 15, pp. 104–116, 2017, ISSN: 20010370. DOI: 10.1016/j.csbj.2016.12.005. [Online]. Available: http://dx.doi.org/10.1016/j.csbj.2016.12.005.

[17] R. Semaan, "Optimal sensor placement using machine learning," *Computers & Fluids*, vol. 159, pp. 167–176, 2017, ISSN: 0045-7930. DOI: 10.1016/J.COMPFLUID.2017.10.002. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0045793017303596.

[18] K. Kumartripathi, "Discrimination Prevention with Classification and Privacy Preservation in Data mining," *Procedia Computer Science*, vol. 79, pp. 244–253, 2016, ISSN: 18770509. DOI: 10.1016/j.procs.2016.03.032.

[19] M. A. Sicilia, "MACHINE LEARNING TECHNIQUES IN USABILITY- EVALUATION QUESTIONNAIRE SYSTEMS," 1999.

[20] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley Sons, 2013.

[21] J. Creswell, "Research design : Qualitative, quantitative, and mixed methods approaches / j.w. creswell.," Jan. 2009.

[22] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995, ISBN: 0-387-94559-8.

[23] X. Zhang, J. Li, J. Li, and G. Liu, "Investigation of naive bayes classifier for intrusion detection system," *Journal of Computational Science*, vol. 52, p. 101 404, 2021.

[24] M. Reza, S. Miri, and R. Javidan, "A Hybrid Data Mining Approach for Intrusion Detection on Imbalanced NSL-KDD Dataset," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 1–33, 2016, ISSN: 2158107X. DOI: 10.14569/ijacsa.2016.070603.

[25] L. Bottou, "Large-scale machine learning with stochastic gradient descent," *Proceedings of COMPSTAT'2010*, pp. 177–186, 2010.

[26] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[27] DataCamp, "Classification in machine learning: Basics and algorithms," *DataCamp Blog*, 2021. [Online]. Available: https://www.datacamp.com/blog/classification-machine-learning.

[28] V. Gjorgjioski, D. Kocev, and S. Džeroski, "MULTI-LABEL CLASSIFICATION WITH PCTs,"

[29] S. N. Liao, D. Zingaro, C. Alvarado, W. G. Griswold, and L. Porter, "Exploring the Value of Different Data Sources for Predicting Student Performance in Multiple CS Courses," pp. 112–118, 2019. DOI: 10.1145/3287324.3287407.

[30] I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning," 2016.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2013.

[32] C. M. Bishop, "Pattern recognition and machine learning," in *Springer*, 2006.

[33] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[34] A. Rajaraman, J. D. Ullman, and J. Leskovec, "Mining of massive datasets," *Cambridge University Press*, vol. 18, no. 3, pp. 1–101, 2011.

[35] C. Caragea, F. Bulgarov, and R. Mihalcea, "Co-training for topic classification of scholarly data," *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, vol. 2, no. September, pp. 2357–2366, 2015. DOI: 10.18653/v1/d15-1283.

[36] C. Caragea, J. Wu, K. Williams, S. D. Gollapalli, M. Khabsa, P. Teregowda, and C. L. Giles, "Automatic identification of research articles from crawled documents," *WSDM 2014 Workshop on Web-scale Classification: Classifying Big Data from the Web*, 2014.

[37] S. Chagheri, S. Calabretto, C. Roussey, C. Dumoulin, S. Chagheri, S. Calabretto, C. Roussey, C. Dumoulin, C. Roussey, and C. Dumoulin, "Document classification : Combining structure and content To cite this version : HAL Id : hal-00637665 Combining Structure and Content," 2011.

[38] C. Caragea, J. Wu, S. D. Gollapalli, and C. L. Giles, *Document Type Classification in Online Digital Libraries*, 2016. [Online]. Available: http://www.cse.unt.edu/{~}ccaragea/papers/iaai16.pdf.

[39] C. Chekuri, M. H. Goldwasser, P. Raghavan, and E. Upfal, "Web Search Using Automatic Classification," *Proceedings of the 6th International World Wide Web Conference*, no. June 1999, pp. 7–11, 1997. [Online]. Available: http://portal.acm.org/citation.cfm?id=324140.

[40] R. Power, J. Chen, T. Karthik, and L. Subramanian, "Document classification for focused topics," *AAAI Spring Symposium - Technical Report*, vol. SS-10-01, pp. 67–72, 2010.

[41] S. Chagheri, S. Calabretto, C. Roussey, and C. Dumoulin, "Document classification: Combining structure and content," *ICEIS 2011 - Proceedings of the 13th International Conference on Enterprise Information Systems*, vol. 1 DISI, no. May 2014, pp. 95–100, 2011.

[42] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music into emotions," *ISMIR 2008 - 9th International Conference on Music Information Retrieval*, no. January, pp. 325–330, 2008.

[43] J. W. CRESWELL, *RESEARCH DESIGN Qualitative, Quantitative, and Mixed Methods Approaches*. 2009, ISBN: 9781412965569. DOI: 10.2307/1523157.

[44] I. Chalkidis, M. Fergadiotis, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "Extreme multi-label legal text classification: A case study in eu legislation," *arXiv preprint arXiv:1906.01740*, 2019.

[45] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognition*, vol. 37, no. 9, pp. 1757–1771, 2004, ISSN: 00313203. DOI: 10.1016/j.patcog.2004.03.009.

[46] L. Phiri, "Research Visibility in the Global South : Towards Increased Online Visibility of Scholarly Research Output in Zambia," in *Proceedings of the 2nd IEEE International Conference in Information and Communication Technologies (ICICT 2018)*, [online] http://dspace.unza.zm/handle/123456789/5723 (Accessed 25 August 2020), Lusaka, Zambia, 2018.

[47] H. Suleman, "The NDLTD Union Catalog : Issues at a Global Scale ETD 2012 The NDLTD Union Catalog : Issues at a Global Scale," pp. 0–1, 2019.

[48] M. Y. Feilzer, "Doing mixed methods research pragmatically: Implications for the rediscovery of pragmatism as a research paradigm," *Journal of Mixed Methods Research*, vol. 4, no. 1, pp. 6–16, 2010, ISSN: 15586898. DOI: 10.1177/1558689809349691.

[49] John Dudovskiy, "The Ultimate Guide to Writing a Dissertation in Business Studies: A Step-by-Step Assistance.," *Research Methodology*, 2018.

[50] J. W. Creswell and J. D. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*. SAGE Publications, 2018.

[51] C. N. Dorsey, B. J. Powell, E. K. Proctor, and R. C. Brownson, "Purposeful sampling for qualitative data collection and analysis in mixed method implementation research," *Administration and Policy in Mental Health and Mental Health Services Research*, vol. 42, no. 5, pp. 533–544, 2015.

[52] Anonymous, "Librecat/catmandu: A comprehensive software stack for research data management, digital libraries, and scholarly communication," 2023. [Online]. Available: https://www.example.com/librecat-catmandu.

[53] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2016.

[54] S. L. Weibel, J. A. Kunze, C. Lagoze, and M. Wolf, "Dublin core metadata for resource discovery," OCLC Online Computer Library Center, RFC 2413, 1998. [Online]. Available: http://www.hjp.at/doc/rfc/rfc2413.html.

[55] C. Lagoze, H. Van de Sompel, M. Nelson, and S. Warner, "The open archives initiative protocol for metadata harvesting," Open Archives Initiative, Tech. Rep., 2002. [Online]. Available: http://www.openarchives.org/OAI/openarchivesprotocol.html (visited on 08/25/2020).

[56] N. Coulter, "Acm's computing classification system reflects changing times," *Communications of the ACM*, vol. 40, no. 12, 1997.

[57] X. Liu, Y. Chen, and X. Li, "A survey of feature extraction in image processing," *Journal of Image and Graphics*, vol. 5, no. 6, pp. 451–456, 2017.

[58] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[59] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010, ISBN: 1848829345.

[60] L. Chen, T.-Y. Li, and S. Zhang, "Short text classification based on wikipedia articles," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, 2015, pp. 435–444.

[61] G. Singh and H. Kaur, "Stop word removal in punjabi text documents," *International Journal of Computer Applications*, vol. 181, no. 8, pp. 18–22, 2018.

[62] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Upper Saddle River, NJ, USA: Prentice Hall, 2009, ISBN: 0131873210.

[63] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019. DOI: 10.3390/info10040150.

[64] A. P. Hafnan and A. Mohan, “Summary-based document classification,” in *Recent Findings in Intelligent Computing Techniques*, P. Sa, S. Bakshi, I. Hatzilygeroudis, and M. Sahoo, Eds. Singapore: Springer, 2018, pp. 153–160. DOI: 10.1007/978-981-10-8633-5_16.

[65] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” *Advances in Neural Information Processing Systems*, vol. 26, pp. 3111–3119, 2013.

[66] H. Liu, Q. Wu, L. Li, J. Li, and H.-S. Wong, “Feature selection with fuzzy entropy measures: A comparative study,” *Pattern Recognition*, vol. 43, no. 1, pp. 317–330, 2010.

[67] P. A. Flach, “Precision-recall-gain curves: Pr analysis done right,” *Machine Learning*, vol. 1, pp. 1–9, 2015.

[68] I. Jolliffe, “Principal component analysis,” *Springer International Publishing*, 2016.

[69] S. Liu, Z. Cheng, C. Li, J. Huang, Z. Yang, W. Wang, and Y. Xu, “Learning effective evaluation metrics for generative models,” *CoRR*, vol. abs/2107.11552, 2021. arXiv: 2107.11552. [Online]. Available: https://arxiv.org/abs/2107.11552.

[70] T. Fawcett, *An Introduction to ROC Analysis*. New York, NY: Springer, 2006. DOI: 10.1007/0-387-21593-4.

[71] D. M. W. Powers, “Evaluation: From precision, recall and f1 to roc, informedness, markedness & correlation,” *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2020.

[72] P. A. Flach, “Precision-Recall-Gain Curves : PR Analysis Done Right,” vol. 1, pp. 1–9,

[73] J. Wiley and S. Chapter, “Text Book : Montgomery , D . C ., Peck , E . A ., & Vining , G . G . (2015). Introduction,” vol. 1, pp. 3–6, 2015.

[74] M. L. Zhang and Z. H. Zhou, “A review on multi-label learning algorithms,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014, ISSN: 10414347. DOI: 10.1109/TKDE.2013.39.

[75] P. Szymanski and T. Kajdanowicz, “Scikit-multilearn: A scikit-based python environment for performing multi-label classification,” *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 209–230, 2019.

[76] Cornell University, *Computer Science Subject Areas and Moderators*, 2020. [Online]. Available: https://arxiv.org/corr/subjectclasses.

[77] Association for Computing Machinery, *The ACM Computing Classification System (1998)*, 2007. [Online]. Available: https://www.acm.org/publications/computing-classification-system/1998/ccs98.

[78] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html.

[79] J. D. Kelleher and B. Tierney, “Fundamentals of machine learning for predictive data analytics: Algorithms, worked examples, and case studies,” *MIT Press*, 2015.

[80] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1–13, 2007, ISSN: 15483932. DOI: 10.4018/jdwm.2007070101.

[81] ChatGPT, *Label power (lp)*, https://github.com/omarsar/chatbot-templates/blob/main/machine-learning/LabelPower.md, Accessed on: April 1, 2023, 2023.

[82] K. Dembczynski, W. Cheng, and E. Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2010, pp. 18–33.

[83] R. M. Rifkin and A. Klautau, "A defense of one-vs-all classification," in *International Conference on Machine Learning*, ACM, 2004, pp. 833–840.

[84] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2008, pp. 99–114.

[85] S. Ben-David, E. Kushilevitz, and Y. Mansour, "Online learning versus offline learning," *Machine Learning*, vol. 29, no. 1, pp. 45–63, 1997. DOI: 10.1023/A:1007465907571.

[86] J. Developers, "Joblib: Running python functions as pipeline jobs," 2008. [Online]. Available: https://joblib.readthedocs.io (visited on 08/25/2020).

[87] R. T. Fielding, "Architectural styles and the design of network-based software architectures," in *Proceedings of the 7th International Conference on Software Engineering*, IEEE, 2000, pp. 407–416.

[88] T. P. Project, *Flask | the pallets project*, https://palletsprojects.com/p/flask, 2010.