Enhancing Machine Learning Performance using Feature Engineering Techniques for Online Course Recommendation System

MSc Research Project
Data Analytics

Srivatsav Kattukottai Mani
Student ID: x18145922

School of Computing
National College of Ireland

Supervisor: Dr. Anh Duong Trinh

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Srivatsav Kattukottai Mani
Student ID:	x18145922
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr. Anh Duong Trinh
Submission Due Date:	14/08/2023
Project Title:	Enhancing Machine Learning Performance using Feature Engineering Techniques for Online Course Recommendation System
Word Count:	6042
Page Count:	23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	18th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Machine Learning Performance using Feature Engineering Techniques for Online Course Recommendation System

Srivatsav Kattukottai Mani
x18145922

Abstract

A recommendation system is a machine learning-based system that is used to recommend suggestions to users based on their previous records that provides the capabilities of decision-making to the users. Some of the major recommendation systems were developed for applications like medical search, movie search, movie reviews, course recommendations etc. Online courses provide learners with high quality and flexible online courses with no limitations regarding time and location. But, recommendation without proper features or feature engineering may lead to less effective and less chances of personalized recommended items to users. Also, feature engineering helps to make a decent recommendation to users when they are looking for new items especially when the data is sparse and unstructured such as text or images. To address these limitations, in this research, Feature engineering techniques such as Stopword removal, Stemming, Decontraction, sent to words, lemmatization and Vectorization is focussed to see how it can improve machine learning algorithms that can be fed into a real world Recommendation system to recommend online courses to everyone. There are lots of e-learning courses available on the websites like Coursera, Edx, Udacity etc. Here for the analysis we consider the publicly available Udacity dataset from the Kaggle website. The raw data is transformed to more suitable data and fed into three classification models - Support Vector Machine (SVM), K-Nearest Neighbors and Adaptive Boosting models. A comparative analysis is performed on both raw and transformed data by using a confusion matrix that provides recall, accuracy, precision, and F1 score to measure the performance of the developed models. The comparative results shows that all the models using the transformed data has shown promising improvement and proves the model developed will accurately provide relevant courses to the users tailored to their preferences. Accuracy of AdaBoost model has shown 72.5% accuracy on the transformed data compared to 37.25% on raw data.

Keywords: Recommendation System, Machine learning, Feature Engineering, SVM, KNN, AdaBoost, Stopwords, Stemming, Decontraction, sent to words, lemmatization, Vectorization.

Contents

1 Introduction 1

1.1 Research Objective . . . 2

1.1.1 Research Question . . . 2

2 Related Work 2

2.1 Course Recommendation System . . . 3

2.2 Recommendation System using different filtering techniques . . . 4

2.2.1 Recommendation using Content-Based Filtering . . . 4

2.2.2 Recommendation using Collaborative Filtering . . . 4

2.2.3 Recommendation using Hybrid Filtering . . . 5

2.3 Recommendation System using Machine learning . . . 5

2.4 Feature Engineering in NLP . . . 6

2.5 Machine learning using NLP in the field of Recommendation System . . . 7

2.6 Summary . . . 7

3 Methodology 8

3.1 Data Collection . . . 8

3.2 Data Preprocessing . . . 8

3.3 Data Transformation . . . 8

3.4 Data Mining . . . 9

3.5 Evaluation and Representation . . . 9

4 Design Specification 9

5 Implementation 11

5.1 Data Collection and Understanding . . . 11

5.2 Data Cleaning . . . 11

5.2.1 Selecting related columns and merging . . . 11

5.2.2 Checking and removing null values . . . 11

5.3 Label encoding . . . 12

5.4 Data transforming . . . 12

5.4.1 Stop words removal . . . 12

5.4.2 Removing punctuations and special characters . . . 12

5.4.3 Stemming . . . 12

5.4.4 Decontraction . . . 13

5.4.5 Sent-to-words . . . 13

5.4.6 Lemmatization . . . 13

5.4.7 Vectorization . . . 14

5.5 ML models 14
5.5.1 SVM (Support Vector Machine) 14
5.5.2 KNN (K-Nearest Neighbor) 15
5.5.3 AdaBoost (Adaptive Boosting) 15

6 Evaluation **15**
6.1 Confusion matrix 15
6.2 Accuracy 16
6.3 Precision 16
6.4 F1-score 16
6.5 Recall score 16
6.6 Results 17
6.6.1 Comparison of Accuracy before and after Feature Engineering . . 17
6.6.2 Comparison of Precision before and after Feature Engineering . . 17
6.6.3 Comparison of F1-score before and after Feature Engineering . . . 18
6.6.4 Comparison of Recall score before and after Feature Engineering . 18
6.7 Discussion 19

7 Conclusion and Future Work **20**

8 Acknowledgement **21**

1 Introduction

A recommendation system is a machine learning algorithm used to recommend relevant things to users as per their preferences and past data. The recommendation system is associated with artificial intelligence based on the big data that suggests or recommends the users the related things as per the previous record. The recommendation system initially used in 1979 called Grundy that was used to suggest books to the readers. Youtube, Netflix, and Amazon are some great examples of recommendation systems Mohamed et al. (2019). The recommendation system got more fame and was used effectively across various domains. Recommendation systems are used nowadays everywhere like restaurants, hospitals, universities, online courses, etc. All these systems use the same machine-learning algorithms and models Reddy et al. (2018). The overall working and the performance of the system depends upon the dataset processed with multiple ML techniques. The need for a well-formed dataset with various criteria and features, is a key concern of researchers to test and evaluate the models on modern applications. The publicly available dataset must be updated in time and there are numerous datasets available for experimentation, but only a few fulfill all the desired features. Udacity, EDX, Udemy and Coursera are few MOOC platforms that are presently available. A quite popular in the field of research is the Udacity dataset is used that was released recently containing diverse online courses and said to fulfill the criteria of the real-world data to overcome some issues of existing datasets.

Data Pre-processing techniques are used to clean up the unwanted words, characters or punctuations from the datasets that are not useful for the machines to interpret. So we use the Natural Language Processing(NLP) which has a lot of techniques that can be chosen for the recommendation system. The data or text pre-processing techniques deals with converting the raw data into an understandable structure where importance is given mostly to the keywords present in the text that highlight the context of the sentence or paragraph. The the processed data is used along with the machine learning algorithms to develop the desired recommendation system.

ML algorithms are widely used to develop the recommendation systems. Currently the usage of Machine Learning has been increasing in every industry. ML algorithms are classified as supervised learning (SVM, ANN, KNN, Random Forest, Bayesian Networks and Decision tree,linear regression, support vector regression etc,) unsupervised learning(K-means clustering, Hierarchical clustering, neural networks, fuzzy logics etc.) From the standpoint of machine learning, recommendation system is a typical classification problem, this machine learning-based recommendation system typically consists of three phases: (i) pre-processing, (ii) training, and (iii) detection or information retrieval.

1.1 Research Objective

The main objective of this research is to explore various Feature engineering methods suitable for text transformation for the selected Udacity courses dataset and see to what extent the transformed data perform for different machine learning models used for text classification compared to the non-transformed data.

1.1.1 Research Question

RQ1: To what extent can Feature Engineering techniques improve the performance of different Machine learning algorithms for Online Course Recommendation System?

The entire report is organized into several sections as follows:

- Extensive study on works related to Course Recommendation System using Machine learning and importance of feature engineering for NLP.
- Methodology followed in this research. Steps involved in KDD.
- Design specification.
- Implementation steps of this research.
- Evaluation of results obtained.
- Discussion and Conclusion along with few suggestions to future works.

2 Related Work

A recommendation system is used to recommend things to the users based on the previous historical data. It is a machine-learning algorithm that suggests related things to the users by following the previous history of the user (Fanca et al., 2020). It is one of the most useful systems nowadays in the field of artificial intelligence. The best example of this recommendation system in today's world is Netflix. Netflix uses the user history who used to watch movies and web series daily and then suggests related movies and series to that user. The recommendation system finds the common things in data used by the user and the upcoming data and then suggests the users accordingly. These recommendation systems are mostly used by big firms and industries because these systems generate income in a huge amounts. Recommendation system works on filtering-based algorithms that can be either collaborative filtering or content-based methods Piletskiy et al. (2020). Recommendation systems are very helpful and efficient to suggest relevant things or material to the related users. These systems also help online business stores and websites improve their engagement and business in the market.

2.1 Course Recommendation System

Recommendation system proposed by Nilashi et al. (2022) used in Massive Open Online Courses (MOOC) makes it easier for the learners to opt one from the recommended courses. This recommendation works on the previous similar data of the course learner and their profiles by using collaborative filtering. In this recommendation system LDA (Latent Dirichlet Allocation), Decision Tree, SOM (Self Organizing Map), and Fuzzy rule-based system are used for the overall recommendation function. For text mining purposes LDA is used and for rules generation, a decision tree is used. SOM is used for online courses review on different platforms and the fuzzy rule system is used for the prediction of user preferences. For the preferences prediction, the feature selection method is used. The dataset is collected from the online platform Udemy. Finally, this recommendation system is found best based on the results. Isma'il et al. (2020) implemented a machine-learning model that recommends a suitable course for secondary school students based on their marks and grades. The classification models of Linear Regression, Naïve Bayes, Support Vector Machine, K-Nearest Neighbor, and Decision Tree are used to implement this recommendation system. These classification models were found very efficient based on their accuracy which is 90%. The overall recommendation accuracy of Naïve Bayes Classifier and SVM was 99.94% whereas the K-Nearest Neighbor and Decision Tree recommended the courses with accuracy of 99.87% and 98.10%. In this study, the dataset of 8700 for three different academic sessions is collected from two universities. It has been concluded that this recommendation system is very good in prediction and can be used for the students to get the recommended courses.

Similarly Dhar and Jodder (2020) implemented a course recommendation system based on machine learning for students who passed the 10th class and want to get admission to further studies. In this system, the approach of feature selection by correlation is used to extract the related features of every academic program. Machine learning algorithms are used to compare the student's 10th class performance based on every subject and a comparative analysis has been made on the course selection. Finally, the best-selected features and models have been used for the recommendation system to recommend the courses to the students regarding their career goals. Also Pardos and Jiang (2020) designed a recommendation system based on collaborative filtering based algorithms with Recurrent Neural Network (RNN) for the recommendation of the course. The two datasets of course catalogues and the enrollment histories are used in this model. A comparative analysis of the performance of offline validation and the RNN has been performed using both datasets of course history and enrollment. It has been analyzed that using the dataset of enrollment histories has the best performance compared to others.

2.2 Recommendation System using different filtering techniques

2.2.1 Recommendation using Content-Based Filtering

The author Van Meteren (2000) proposed that a recommendation system is used to help users with their personalized suggestions based on their previous historical data. Content-based filtering is used to design a recommendation system that can recommend related or experienced things to the users. This recommendation system is efficient and dynamic as compared to other recommendation system models. A comparative analysis is performed by using a confusion matrix that contains recall, accuracy, precision, and F1 score. Yadalam et al. (2020) presented a career recommendation system based on content-based filtering. A recommendation system plays a significant role in graduates' life and guides them accordingly as per their interests, knowledge, and skills based on previous records. Similarly, the job recommendation system also plays an important role and has a high value in the career building of candidates. This study focuses on career recommendation systems by using machine learning and content-based filtering techniques and also refers to the security, reliability, and transparency features of the system. Badriyah et al. (2018) developed a web-based recommendation system by using a machine learning algorithm that works on choosing the property based on content-based filtering. Each time when the user selects any content the information will be saved in the database of the user and then based on that data the system will suggest and recommend the client similar things. This is how a recommendation system works. In this study, a web-based recommendation system is implemented that works on the choice of property produced by the users want.

2.2.2 Recommendation using Collaborative Filtering

Mike Wu et al. (2018) implemented a movie recommendation system based on machine learning by using collaborative filtering. The information and data are collected from the people or users who used to watch the movies and based on that dataset a prediction is made. In this study, collaborative filtering is used with Apache Mahout which helps to find out the best-recommended movie for the users. Gupta et al. (2020) designed a machine-learning algorithm that works to recommend movies to users based on previous knowledge and data. The dataset used is collected through the previous data of the users who watched the movies as per their interest and then machine learning algorithms are applied to that dataset by using collaborative filtering. In this study, the classification models of K-NN algorithms and collaborative filtering are used to recommend the movies and then results are compared with the content-based filtering technique. This collaborative filtering is based on the cosine similarity approach by using the K-nearest neighbour. Sahoo et al. (2019) proposed a deep learning-based health recommender system in which the food or diet is suggested as per the health conditions of the users. This system is more intelligent because it is used to analyze the patient's condition and lifestyle. In this recommendation system collaborative filtering techniques are used to check the accuracy performance. Moreover, the classification models of SVM, logistic regression, and CNN are used. The overall performance of all these classification model and the deep network are controlled. Interesting comparative study by Obeidat et al. (2019) has implemented an online course recommendation system using association rules algorithms such as SPADE and Apriori models with and without clustering the dataset and shown that clustered data has improved performance than the whole data.

2.2.3 Recommendation using Hybrid Filtering

Hybrid filtering refers to the filtering techniques that used both content-based filtering and collaborative filtering. The machine learning algorithms of the decision tree, SVM, naïve Bayes, PCA and LDA. The performance of all classification models is compared with each other and a comparative analysis is performed to check the overall performance of classification models. Walek and Fojtik (2020) implemented a machine learning algorithm with the help of hybrid filtering to recommend the best possible movie to the users based on the average movie rating given by the clients. Collaborative filtering is also used by applying the SVD algorithm, a content-based algorithm, and a fuzzy expert system. The accuracy occurred in this system is 80% and the overall performance of the model is calculated and measured by using the confusion matrix which contains the F1 score, precision, recall, and accuracy. A comparative analysis is performed for the overall accuracy measured by the model. Valdiviezo-Diaz et al. (2019) proposed a recommender system based on the hybrid filtering approach based on the naïve Bayes algorithm. In this study, collaborative filtering is also used based on historical data ratings. This study focuses on two approaches memory-based and model-based. The memory-based approach provides a less accurate recommendation whereas the model-based approach is more price and difficult. The best performance model has been analyzed by using Normalized Discounted Cumulative Gain (DCG) and the accuracy predictions are improved.

2.3 Recommendation System using Machine learning

Among Naïve Bayes, Logistic Regression, SVM, and Random Forest machine learning algorithms the Random Forest-based model gave the highest accuracy rate (95%). The machine learning model chosen relied on the learner preferences and MOOC characteristics. Assami et al. (2022) validates the ML based recommendation system in MOOC provides personalized lists to the users. The study concludes from the feature importance algorithm that MOOC features have the bigger influence, compared to the learners personal choices , on the exploration or not of a given MOOC.

The high attrition rate observed on many MOOC platforms, dropout prediction, or identifying students at risk of dropping out of a course, is an essential topic to study. Hence the institutions are investing in MOOCs platforms to monetize making them available in several languages and disciplines. Aoulad Ali et al. (2021) provides the dropout prediction as a sequence classification problem, and proposes a comparison classification model to solve the issue using machine learning algorithms like KNN, Support Vector Machines (SVM), and Logistic Regression (LR), Ada Boost and Random Forest.

A research study by Hu et al. (2021) analyzed few data to predict the final grades of the students/learners to guide them with the learning process. This inturn gives more efficient suggestions to improve the MOOC platform improving the learner satisfaction, stickiness on reducing the student failure rates. The data model developed needed data on the learning behaviors, few factors affecting the final grades for better analysis. KNN and LVQ were the machine learning algorithms used for the prediction.

2.4 Feature Engineering in NLP

It was found that text preprocessing techniques are the major contributor in increasing the accuracy of any text-based machine learning algorithm. The order in which the NLP is used varies the result. The tokenization, stopwords removal, punctuations removal, lemmatization are the widely used efficient text pre-processing techniques. Some of the popular feature extraction techniques are Bag-of-words model and TF- IDF. Most study proves that the TF-IDF is by far the best choice for highlighting the prominent features and downscaling the irrelevant features. These are predominantly used in Web Search or Search Engine tools. Depending on the use case these steps can either be enhanced or removed if unnecessary.

The author Kanwal et al. (2021) summarizes on the exploration of text based recommendation system and the critical approaches for better understanding of the models. They focused on the fundamental techniques of feature extraction, the data sets, how the models are evaluated and the performance metrics used and finally the machine learning algorithms used to effectively build a text based recommendation system. Similary Al-Hawari and Barham (2019) has studied the importance of text pre-processing for ticket classification in the field of help desk system for IT service management and proved that text processing has improved the accuracy around 20 to 30% for all the four classification models used.

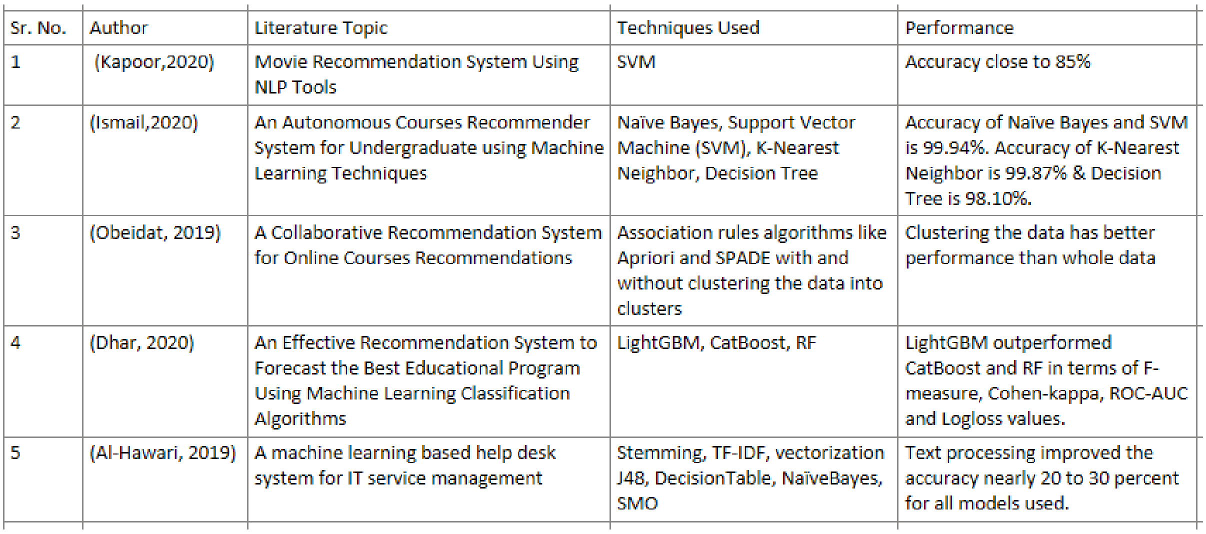
2.5 Machine learning using NLP in the field of Recommendation System

A recommendation system for healthcare is developed by Kumar et al. (2022) for assisting hospitals on providing promt medical attention to the patients based on their needs. The goal of this system's development was to give patients and healthcare professionals a collection of tools that would help them make better decisions about their health. Different machine learning algorithms were used in each case along with NLP techniques used to build the model. They also provide few insights on projection of health status, recommending therapies, hospitals etc.

A movie recommendation system is developed using NLP techniques in machine learning models by Kapoor et al. (2020). The real time dataset was used to analyse the movie reviews and movie ranking algorithm to rank the recommended movies. The developed model used SVM using NLP technique and was tested with the real time users having 65 to 70 percentage success rate based on the positive feedback received on the model developed.

2.6 Summary

A short summary of few importance literature papers were summarised in below figure 1



Sr. No.	Author	Literature Topic	Techniques Used	Performance
1	(Kapoor,2020)	Movie Recommendation System Using NLP Tools	SVM	Accuracy close to 85%
2	(Ismail,2020)	An Autonomous Courses Recommender System for Undergraduate using Machine Learning Techniques	Naïve Bayes, Support Vector Machine (SVM), K-Nearest Neighbor, Decision Tree	Accuracy of Naïve Bayes and SVM is 99.94%. Accuracy of K-Nearest Neighbor is 99.87% & Decision Tree is 98.10%.
3	(Obeidat, 2019)	A Collaborative Recommendation System for Online Courses Recommendations	Association rules algorithms like Apriori and SPADE with and without clustering the data into clusters	Clustering the data has better performance than whole data
4	(Dhar, 2020)	An Effective Recommendation System to Forecast the Best Educational Program Using Machine Learning Classification Algorithms	LightGBM, CatBoost, RF	LightGBM outperformed CatBoost and RF in terms of F-measure, Cohen-kappa, ROC-AUC and Logloss values.
5	(Al-Hawari, 2019)	A machine learning based help desk system for IT service management	Stemming, TF-IDF, vectorization J48, DecisionTable, NaïveBayes, SMO	Text processing improved the accuracy nearly 20 to 30 percent for all models used.

Figure 1: Literature review summary

3 Methodology

This research follows KDD approach as shown in the figure 2 that has sequence of steps as follows: (Azevedo and Santos, 2008) and (Shafique and Qaiser, 2014) has done a detailed study on types of methodologies and steps involved starting from suitable Data selection, pre-processing, transforming, data mining and evaluation.

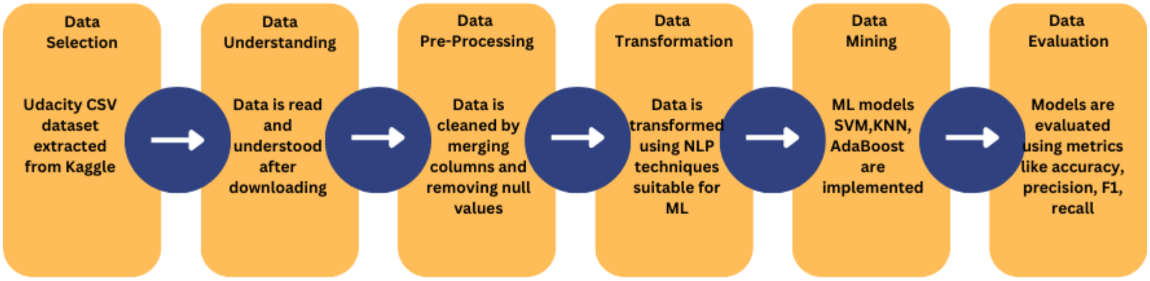

Data Selection
Udacity CSV dataset extracted from Kaggle
Data Understanding
Data is read and understood after downloading
Data Pre-Processing
Data is cleaned by merging columns and removing null values
Data Transformation
Data is transformed using NLP techniques suitable for ML
Data Mining
ML models SVM,KNN, AdaBoost are implemented
Data Evaluation
Models are evaluated using metrics like accuracy, precision, F1, recall

Figure 2: KDD methodology

3.1 Data Collection

First and foremost step of KDD approach is the Data selection and understanding. For this research, the main objective is to analyse how feature engineering can help improve the ML performances for Course recommendation and hence, the suitable and recent Udacity dataset is chosen which contains details about online courses and list of schools they fall under.

3.2 Data Preprocessing

Preprocessing the raw data in this stage involves removing any noise, inconsistencies, or missing values.

3.3 Data Transformation

Data is altered or consolidated into a appropriate format for analysis. This step is known as feature engineering where the raw data is transformed into machine understandable format. Techniques like encoding categorical variables, stop words removal, lemmatization and converting text into array of numbers using vectorizer are applied to ensure data is suitable for machine learning.

3.4 Data Mining

The core of the KDD process, data mining involves applying various algorithms and techniques to discover patterns, associations, correlations, or trends within the data. For this research, text classification models are used such as SVM, KNN and Adaptive boosting.

3.5 Evaluation and Representation

Once patterns are discovered through data mining, they need to be evaluated to determine their significance and usefulness and represented visually to stakeholders.

4 Design Specification

This entire research is done on a desktop application known as "Anaconda Navigator" which is a platform of many web-based applications such as JupyterLab, Spyder, RStudio etc. This research work is completely built using Jupyter notebook that runs on a web browser and acts a user friendly interface for writing and executing python programming language. Jupyter notebook v6.5.4 is used with Python v3.11.3 installed that runs on IPython kernel v8.12.0. Another benefits of Jupyter notebook is the simple representation of code with headers, ability to download high-resolution output graphs, all contents are consolidated in a hierarchical fashion. It is possible to execute many parallel notebooks that can simultaneously execute multiple kernels for different processing purposes. The complete notebook could then be exported in a variety of formats so that it may be shared among others with ease.

Major tasks followed in this research are as listed below:

1. Data extraction and pre-processing.

2. Performing feature engineering to transform the text data suitable for Machine learning.

3. Implementing and comparing ML models using the raw data and transformed data

4. Evaluating the model performance on the raw data vs transformed data.

Complete flow diagram of this research work is illustrated in below Figure 3. Each step shown in the flow chart below are explained in detail in section 5 of this report.

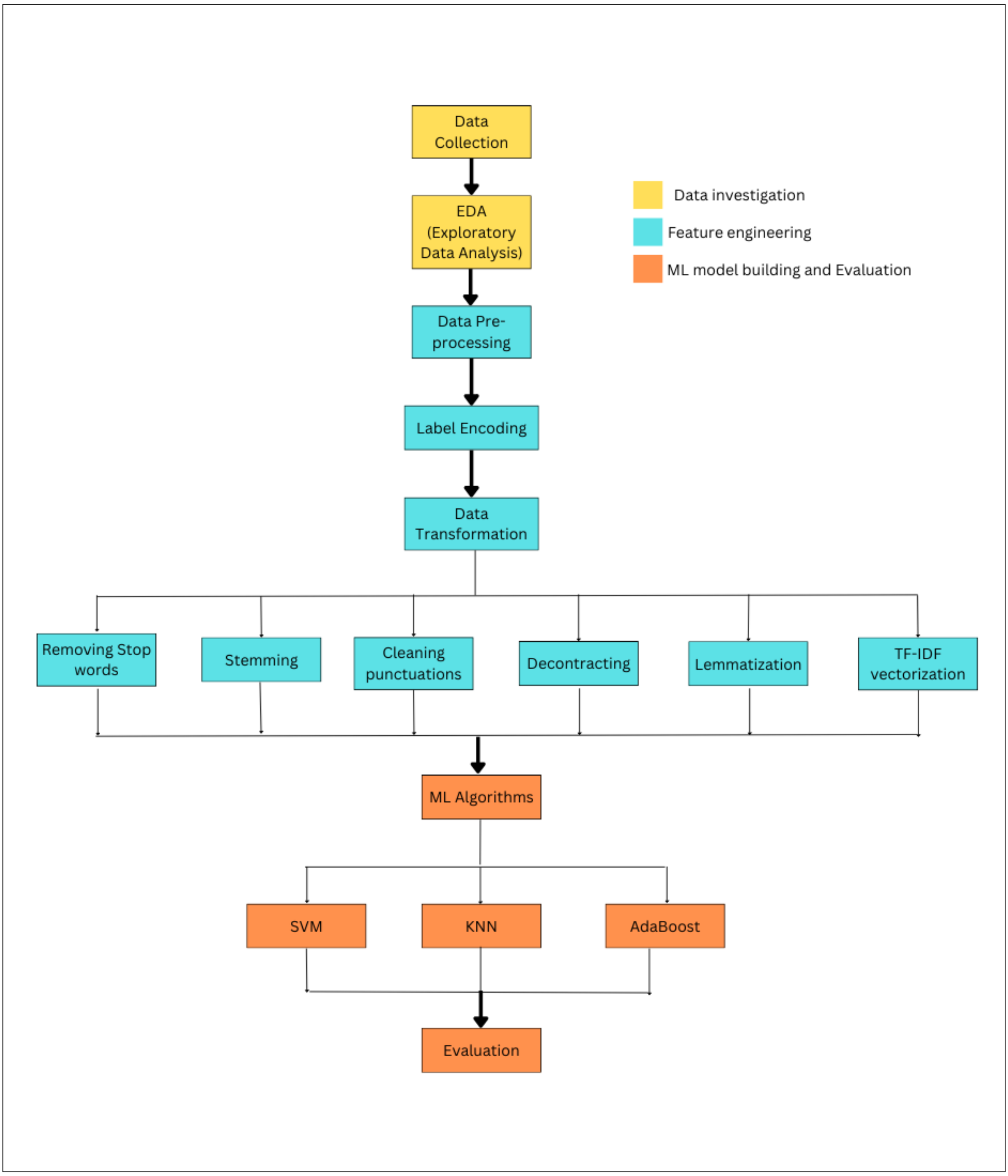


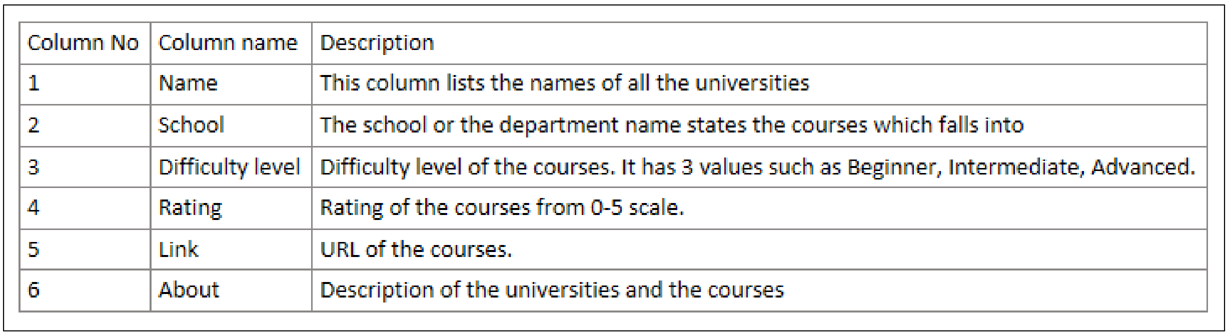
Figure 3: Project Design flow

5 Implementation

The sequence of steps followed in this research work are explained in Figure 3 describing Data investigation, Feature engineering, ML model building and evaluation.

5.1 Data Collection and Understanding

The suitable dataset for this research is taken from Kaggle which is an open source online repository that contains numerous datasets that can be utilized for machine learning researches. In this research, Udacity courses dataset (2021) is used that was scrapped by the developer as CSV format from the publicly available information on the Udacity website. The data contains 263 rows and 6 columns with information about the online courses, school or department it belongs, Difficulty level, ratings for each course, description about the courses and their respective hyperlinks. Next, the downloaded data is read and analysed. URL of the dataset:
`https://www.kaggle.com/datasets/khusheekapoor/udacity-courses-dataset-2021`
Detailed description of the dataset is shown in below Figure 4



Column No	Column name	Description
1	Name	This column lists the names of all the universities
2	School	The school or the department name states the courses which falls into
3	Difficulty level	Difficulty level of the courses. It has 3 values such as Beginner, Intermediate, Advanced.
4	Rating	Rating of the courses from 0-5 scale.
5	Link	URL of the courses.
6	About	Description of the universities and the courses

Figure 4: Dataset used in this research

5.2 Data Cleaning

5.2.1 Selecting related columns and merging

Focussing only on the required features leads the machine algorithms to better perform and time efficiently. In this research, only 3 features were focussed - name and description of the course and school. To get a precise and consolidated Course, the name and description columns are merged and school is renamed as University.

5.2.2 Checking and removing null values

In this Step, the missing values are checked and eliminated. Checking and removing null values is an essential data preprocessing step in machine learning. Null values, representing missing or undefined data, can lead to inaccurate analysis and modeling. By identifying and eliminating these values, data integrity is ensured, and the resulting dataset becomes suitable for training robust and reliable machine learning models.

5.3 Label encoding

In machine learning label encoding is used to convert categorical variables into numerical form. Each category is assigned a unique integer value, allowing algorithms to process and analyze categorical data effectively. In this research, the University is label encoded using the label encoder function so the courses can be classified to one of the labels they belong.

5.4 Data transforming

5.4.1 Stop words removal

Common terms like 'the', 'is', 'and' etc. are taken out of text data as part of a crucial preprocessing phase in NLP (Natural Language Processing). These words are common in a language yet does not add much value to the sentences overall meaning (Tabassum and Patil, 2020). The dimensionality of the data is decreased by eliminating stop words, which leads to more effective processing and enhanced model performance. As a result, processes like sentiment analysis, text categorization, and information retrieval get superior results by concentrating on more related terms. The accuracy and interpretability of NLP models are increased while the computational burden is decreased by eliminating stop words.

Example of pre and post Stop words removal:
Before Stopwords removal: 'This is a sample sentence for stopword removal'
After Stopwords removal: 'sample','sentence','stopword','removal'

5.4.2 Removing punctuations and special characters

In NLP, it's crucial to remove punctuation and special characters during the preprocessing stage. These non-alphabetic letters can amplify text data while carrying no useful information (Tabassum and Patil, 2020). The text is made cleaner by getting rid of them, which helps with precise text analysis, categorization, and language model training.

5.4.3 Stemming

The NLP (Natural Language Processing) approach of stemming entails breaking down words to their root or basic form. A popular stemming method that works with various languages is called Snowball. By deleting suffixes and prefixes to create a common base term, it helps to standardize word variants. Stemming is important because it decreases the quantity of the vocabulary, makes text processing easier, and increases the effectiveness of NLP tasks like information retrieval, and text search. Stemming improves generalization and increases the accuracy and efficacy of machine learning models used to analyze language-based data by considering word variants as a single unit.

5.4.4 Decontraction

Online platform data contains lot of contracted words that are written in short form. Decontraction in NLP is the process of disassembling text in natural language into its component parts, such as sentences, phrases, and words. It is essential for comprehending the text's structure and meaning and for facilitating other language processing activities. Language processing are made easier by decontracting, in other words expanding the contracted words for ML better understand the context and relationships between words.

Example of pre and post Decontraction:
Before Decontraction: ‘I'll see you later’
After Decontraction: ‘I will see you later’

5.4.5 Sent-to-words

In NLP, the term ’sent-to-words’ describes the process of breaking down a phrase or a sentence of text into its constituent words. In order for NLP models to comprehend the context and semantics of the text, it is essential to process and analyze text. Various language processing tasks and feature extraction in NLP applications are made possible by this transition. Python uses (sent_to_words) function for this purpose.

Example of pre and post sent_to_words:
Before sent_to_words: ‘Today is a beautiful day’
After sent_to_words: ‘today’,‘is’,‘beautiful’,‘day’

5.4.6 Lemmatization

When a word is lemmatized, it is reduced to its simplest possible form, or lemma, which is a main concept in natural language processing. Lemmatization, as opposed to stemming, takes into account the word's meaning and guarantees that appropriate words are generated. It is significant because it brings inflected words down to a common foundation, enhancing the precision of tasks involving language analysis, search, and information retrieval. By decreasing word variants and streamlining the vocabulary needed for processing and comprehending natural language text, lemmatization improves language model performance by providing more meaningful word representations, aiding in preserving linguistic consistency across text data, and providing more meaningful word representations.

5.4.7 Vectorization

In NLP, the term 'vectorization'; refers to the process of transforming text input into numerical vectors that machine learning algorithms can analyze. Machines can now successfully comprehend and analyze textual data due to this transition. In this research, the text data is vectorized using the effective TF-IDF vectorizer available in python. TF-IDF (Term frequency – Inverse document frequency) is widely used in NLP (Tabassum and Patil, 2020). Based on their frequency in the document and rarity across the full corpus, it gives words in a document numerical value. The TF-IDF measures how significant a word is in a document in relation to other instances of that word. Higher TF-IDF ratings are given to words that are often used in one text but infrequently used in another, emphasizing their importance in describing the content of that document. The resultant TF-IDF vectors provide documents a concise and insightful representation, making tasks like text categorization, clustering, and information retrieval easier to complete.
Term-Frequency (TF): Measures how frequently a word occurs in the sentence or entire documents and assigns the value.
Inverse-Document Frequency (IDF): Measure the importance of particular words instead of how often they occurs in a text. TF_IDF is the multiplication of both and represented using the formula as below:

$$tf - idf(t, d) = tf(t, d) * idf(t) \tag{1}$$

5.5 ML models

5.5.1 SVM (Support Vector Machine)

SVM is chosen in this research as it is considered as one of the most powerful machine learning algorithm for classification as per (Liu et al., 2010) and most relevant state of the art technique for text classification. SVM greatly helps to identify a best possible line or hyperplane in the high dimensional space that separates different classes of data points by calculating the distance between them. It has many advantages such as simplest structure, great optimization, time efficient etc.

5.5.2 KNN (K-Nearest Neighbor)

KNN is used in this research as it is one among simple classification algorithm as studied by (Liu et al., 2010). KNN stores the datapoints during the training phase and compares the unseen or new datapoints with the stored data at testing phase. Using various distance metrics such as Cosine distance, euclidean distance etc, KNN calculates the distance between the data points and classify them to specific K nearest points and help to predict new data using the nearest neighbors.

5.5.3 AdaBoost (Adaptive Boosting)

Adaptive Boosting also knowns as AdaBoost is considered as an important algorithm to increase the learning accuracy in ML models (Chengsheng et al., 2018). In general, AdaBoost combines several weak classifiers such as Decision Tree and combines them to form a strong classifier. Each weak classifier on the training data will be assigned a weighted score or vote and finally using multiple weak classifers are merged to form a strong classifier by taking the weighted average of them.

6 Evaluation

Evaluation is critical in ML models to assess their performance and reliability. It allows us to understand how well the model generalizes to new, unseen data and helps in comparing different algorithms to choose the best one for a specific task. Without proper evaluation, there is a risk of overfitting, where a model performs well on the training data but poorly on new data. Evaluation also aids in identifying potential biases and errors, enabling model improvement and fine-tuning for better results. skleanr-metrics from python is used to derive these mertrics.

Each metric provides specific insights into model performance, and their selection depends on the problem domain and requirements. Evaluation helps in identifying model strengths and weaknesses, guiding further improvements and ensuring ML models are suitable for real-world applications (Leo et al. 2019).

6.1 Confusion matrix

A table called a confusion matrix is used in machine learning to assess how well a classification model is performing. It presents True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) values as a comparison between expected and actual labels. The matrix offers information on the model's precision and error categories, assisting in performance evaluation and enhancement.

6.2 Accuracy

Accuracy measured in percentage indicates how accurately the model can classify the provided data. In other words, it quantifies the percentage of properly identified examples among all the occurrences in the dataset. Accuracy value ranges from 0 to 100%. It is calculated as per the below formulae. Here in this research, AdaBoost classifier has the best performance among the 3 models on the selected dataset with 72.5% accuracy on the transformed data vs 37.25% on raw data.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

6.3 Precision

The accuracy of the model's future predictions is measured by a machine learning performance metric called precision. Out of all projected positive instances (True Positive + False Positive), it calculates the proportion of positively anticipated events that really occur (True Positive). Higher precision indicates fewer false positives and better positive prediction accuracy. Precision falls between 0 to 1.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

6.4 F1-score

F1-score combines recall and accuracy. It is the harmonic mean of these two numbers, giving a fair assessment of the models effectiveness. F1 score ranges from 0 to 1.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{4}$$

6.5 Recall score

Recall score also known as Sensitivity measure is a performance statistic that assesses how well a model can identify good examples. It determines whether percentage of all real positive occurrences (True Positive + False Negative) were properly anticipated positive instances (True Positive). Higher recall indicates better positive instance identification by the model. Recall score falls between 0 to 1.

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

6.6 Results

6.6.1 Comparison of Accuracy before and after Feature Engineering

Here the accuracy of all the 3 ML models implemented in this research are compared in Fig.5 using the raw data and transformed data. It can be seen that all the 3 models has improved accuracy on the transformed data that can enhance the performance of Course recommendations and thus supports our objective that feature engineering plays a vital role for improving ML algorithms. (i) SVM model has an accuracy of 68.63% on transformed data vs 49.02% on the raw data. (ii) KNN model has 58.82% accuracy on transformed data vs 37.25% on raw data and finally AdaBoost model has 72.55% accuracy on transformed data vs 37.25% on raw data.

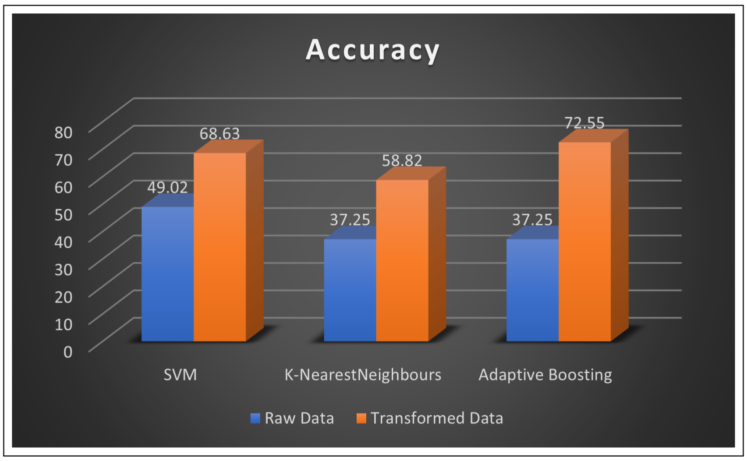


Figure 5: Comparison of Accuracy before and after Feature Engineering

6.6.2 Comparison of Precision before and after Feature Engineering

Comparing the precision values in Fig.6, it shows the transformed data using feature engineering has better precision scores on SVM and KNN models to that of raw data and closely matched precision score for AdaBoost model. (i) SVM model has the precision score of 0.62 on transformed data vs 0.58 on raw data, (ii) KNN model has 0.74 precision value on transformed data vs 0.55 on raw data and (iii) Adaboost model has 0.60 precision value on transformed data vs 0.62 score on raw data.

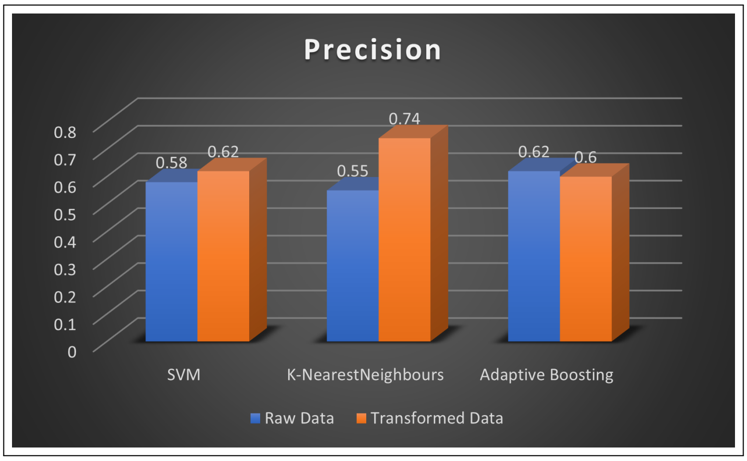


Figure 6: Comparison of Precision before and after Feature Engineering

6.6.3 Comparison of F1-score before and after Feature Engineering

Comparing the F1-scrore values in Fig.7, it shows the transformed data has better F1 scores to that of raw data on all the three ML models used in this research. (i) SVM model has 0.65 f1-score on transformed data vs 0.53 on raw data, (ii) KNN has 0.65 f1-score on transformed data vs 0.44 on raw data and (iii) AdaBoost model has an f1-score of 0.66 on transformed data vs 0.47 on raw data. It has been observed that all 3 models has pretty similar f1-score on the transformed data describing the effectiveness of these models after feature engineering techniques are utilized.

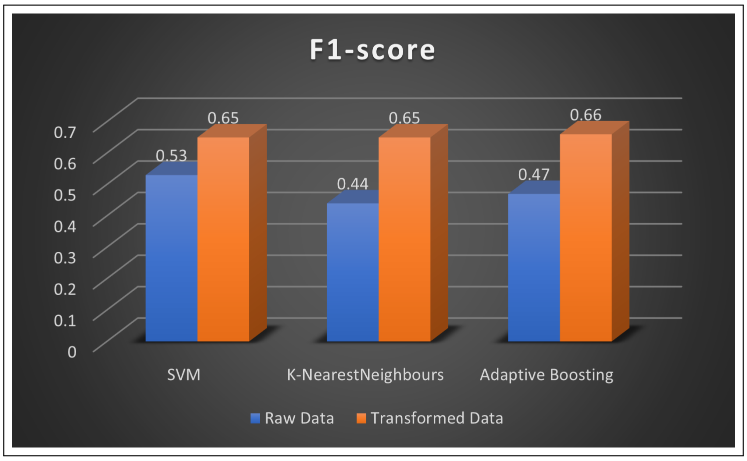


Figure 7: Comparison of F1-score before and after Feature Engineering

6.6.4 Comparison of Recall score before and after Feature Engineering

Comparing the recall values in Fig.8, it shows the transformed data has better recall scores to that of raw data. (i) SVM model has 0.69 recall value on transformed data vs 0.49 on raw data, (ii) KNN has 0.59 recall score on transformed data vs 0.37 on raw data and (iii) AdaBoost model has the highest recall score of 0.73 on transformed data vs 0.37 on raw data.

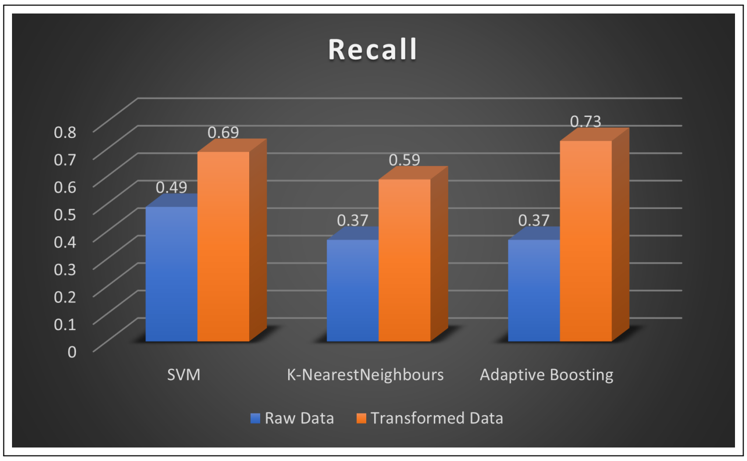
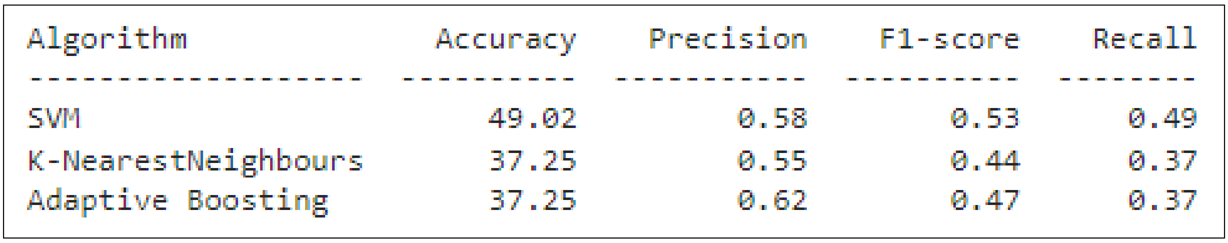


Figure 8: Comparison of Recall score before and after Feature Engineering

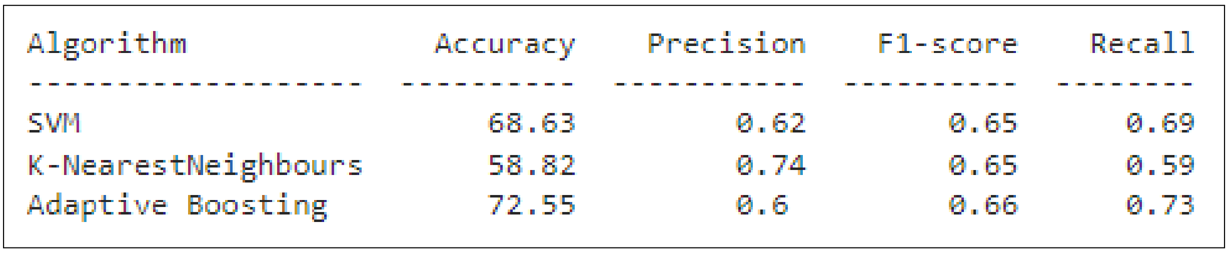
6.7 Discussion

To focus on how feature engineers plays a vital role in improving the ML performance, a comparative results were obtained on the performance of models on both raw and transformed course features as shown in below Figures 9 and 10



```
Algorithm               Accuracy    Precision    F1-score    Recall
--------------------  ----------  -----------  ----------  --------
SVM                        49.02         0.58        0.53      0.49
K-NearestNeighbours        37.25         0.55        0.44      0.37
Adaptive Boosting          37.25         0.62        0.47      0.37
```

Figure 9: Performance of Raw course feature



```
Algorithm               Accuracy    Precision    F1-score    Recall
--------------------  ----------  -----------  ----------  --------
SVM                        68.63         0.62        0.65      0.69
K-NearestNeighbours        58.82         0.74        0.65      0.59
Adaptive Boosting          72.55         0.6         0.66      0.73
```

Figure 10: Performance of Transformed course feature

From the derived results, it is clearly seen how important feature engineering can be useful and to what extent the transformed data can improve the performance of ML models selected. It is observed that among the 3 models implemented, Adaptive boosting outperforms SVM and KNN models with accuracy of 72.5% on the transformed data vs 37.25% on the raw data. SVM has shown improvement in accuracy from 49.02% to 68.63% and similarly KNN improved its accuracy from 37.25% to 58.82%. It is observed that Precision scores for Adaptive boosting is pretty close on both raw (0.62) vs transformed feature (0.60). Respectively, on the transformed feature, all the other metrics Precision, recall and F1-scores has shown decent improvement for all the models thus satisfying this research's hypothesis.

7 Conclusion and Future Work

Recommendation systems are particularly valuable in situations where there's a large volume of items or options available and users might be overwhelmed by choice. The recommendation systems leverage data and algorithms to provide tailored suggestions to users, making their interactions with digital platforms more engaging and efficient. Most of the online search would be through text, numbers, images, audio and video formats. The course recommendation system is designed to help scholars in opting suitable courses based on their academic interests, career pretensions, and previous educational history. By leveraging data analysis and machine learning techniques, these systems provide personalized suggestions to help students make informed decisions about their education. The chosen Udacity dataset is an updated one and has valuable resources for understanding online learning behavior and improving educational outcomes. They include data points from various online courses offered by Udacity, covering a wide range of universities and subjects. The developed model is analyzed by comparing the raw course feature and the transformed feature. For the datasets to be processed feature engineering for NLP is used. Tokenization, StopWords removal, punctuations removal, lemmatization are the efficiently used NLP pre-processing techniques. The TF-IDF model is the best choice for highlighting the prominent features and transforming text data to numerical form. Machine learning algorithms make mere predictions of any data be it any format. SVM, KNN and Adaboost machine learning models perform well in prediction accuracies based on the literature survey done. Table 9 and 10 shows that all the models have improved accuracies and performance metrics when treated with the transformed feature when compared to the raw feature used. Thus the results of the analytical study gives an insight on the importance of NLP and machine learning in developing a recommendation system.

Some of the future works that can be considered are as below

- Extending the research on personalized datasets with more features related to courses (university or online) or different countries.

- Extracting user behavior data and creating a customized course recommendation User Interface.
- Focusing on more advanced ML models with in-depth study on hyper parameter tuning to improve efficiency.

8 Acknowledgement

I would like to thank my professor Dr. Anh Duong Trinh for his support and guidance throughout the module that helped me in completing this research project work successfully.

References

Al-Hawari, F. and Barham, H. (2019). A machine learning based help desk system for it service management. *Journal of King Saud University - Computer and Information Sciences*, 33(6).

Aoulad Ali, H., Mohamed, C., Abdelhamid, B., and El Alami, T. (2021). Prediction mooc's for student by using machine learning methods.

Assami, S., Daoudi, N., and Ajhoun, R. (2022). Implementation of a machine learning-based mooc recommender system using learner motivation prediction. *International Journal of Engineering Pedagogy (iJEP)*, 12(5):68–85.

Azevedo, A. and Santos, M. F. (2008). Kdd, semma and crisp-dm: A parallel overview.

Badriyah, T., Azvy, S., Yuwono, W., and Syarif, I. (2018). Recommendation system for property search using content based filtering method.

Chengsheng, T., Bing, X., and Huacheng, L. (2018). The application of the adaboost algorithm in the text classification.

Dhar, J. and Jodder, A. K. (2020). An effective recommendation system to forecast the best educational program using machine learning classification algorithms.

Fanca, A., Puscasiu, A., Gota, D.-I., and Valean, H. (2020). Recommendation systems with machine learning - ieee xplore.

Gupta, M., Thakkar, A., Aashish, Gupta, V., and Rathore, D. P. S. (2020). Movie recommender system using collaborative filtering. *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*.

Hu, W., Li, Y., Liu, F., and Liu, T. (2021). Machine-learning based mooc education data analysis.

Isma'il, M., Haruna, U., Aliyu, G., Abdulmumin, I., and Adamu, S. (2020). An autonomous courses recommender system for undergraduate using machine learning techniques.

Kanwal, S., Nawaz, S., Malik, M. K., and Nawaz, Z. (2021). A review of text-based recommendation systems. *IEEE Access*, 9:31638–31661.

Kapoor, N., Vishal, S., and K. S., K. (2020). Movie recommendation system using nlp tools.

Kumar, C. S., Sirisati, R. S., Gudditti, V., Rao, K. S., and Challa, R. K. (2022). A smart recommendation system for medicine using intelligent nlp techniques.

Liu, Z., Lv, X., Liu, K., and Shi, S. (2010). Study on svm compared with the other text classification methods. *2010 Second International Workshop on Education Technology and Computer Science.*

Mike Wu, C.-S., Garg, D., and Bhandary, U. (2018). Movie recommendation system using collaborative filtering.

Mohamed, M. H., Hasan Ibrahim, M., and Khafagy, M. (2019). Recommender systems challenges and solutions survey - researchgate.

Nilashi, M., Minaei-Bidgoli, B., Alghamdi, A., Alrizq, M., Alghamdi, O., Nayer, F. K., Aljehane, N. O., Khosravi, A., and Mohd, S. (2022). Knowledge discovery for course choice decision in massive open online courses using machine learning approaches.

Obeidat, R., Duwairi, R., and Alaiad, A. (2019). A collaborative recommendation system for online courses recommendations.

Pardos, Z. A. and Jiang, W. (2020). Designing for serendipity in a university course recommendation system: Proceedings of the tenth international conference on learning analytics amp; knowledge.

Piletskiy, P., Chumachenko, D., and Meniailov, I. (2020). Development and analysis of intelligent recommendation system using machine learning approach.

Reddy, S., Nalluri, S., Kunisetti, S., Ashok, S., and Venkatesh, B. (2018). Content-based movie recommendation system using genre correlation.

Sahoo, A. K., Pradhan, C., Barik, R. K., and Dubey, H. (2019). Deepreco deep learning based health recommender system using collaborative filtering. *MDPI.*

Shafique, U. and Qaiser, H. (2014). (pdf) a comparative study of data mining process models (kdd, crisp-dm and semma).

Tabassum, A. and Patil, R. R. (2020). A survey on text pre-processing feature extraction techniques in natural language processing.

Valdiviezo-Diaz, P., Ortega, F., Cobos, E., and Lara-Cabrera, R. (2019). A collaborative filtering approach based on naïve bayes classifier. *IEEE Access*, 7:108581–108592.

Van Meteren, R. (2000). Using content-based filtering for recommendation.

Walek, B. and Fojtik, V. (2020). A hybrid recommender system for recommending relevant movies using an expert system.

Yadalam, T. V., Gowda, V. M., Kumar, V. S., Girish, D., and M., N. (2020). Career recommendation systems using content based filtering.