Bachelor of Science in Computer Science
May 2021

Movie prediction based on movie scripts using Natural Language Processing and Machine Learning Algorithms

Bhargav Chinnapottu
Govardhan Arikatla

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Bleckinge Institute of Technology in partial fulfilment of the requirements for the degree of Bachelor of Science in Computer Science. The thesis is equivalent to 10 weeks of full time studies.

The authors declare that they are the sole authors of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

Contact Information:
Author(s):
Bhargav Chinnapottu
E-mail: bhch20@student.bth.se

Govardhan Arikatla
E-mail: goar20@student.bth.se

University advisor:
Suejb Memeti, Senior Lecturer
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

Abstract

Background: Natural Language Processing (NLP) is a field in artificial intelligence which deals with the communication between humans and computers. It makes the computers understand the human language text and perform different programs with that data. NLP approaches are used to convert the text in human language to computer understandable numbers to perform the operations. NLP techniques can be used in the field of machine learning for prediction, which motivated us to proceed with our thesis of predicting the movie name based on the dataset of movie scripts, which requires natural language text preprocessing.

Objectives: The objective of this thesis is to implement a model that predicts the name of a movie using the input of random text which has a similar meaning to the text from the script data and obtain the accuracy in the prediction of the movie name.

Methods: Literature study method is used to identify the suitable algorithms that can be used for training the model and experiment method is used to find the accuracy of the model in the prediction of movie name which involves gathering dataset of movie scripts, preprocessing of the data, training the model with preprocessed data and different classification algorithms identified from the literature study, and predicting the movie name of random sentences from the developed models.

Results: Algorithms identified from the literature study include Random Forest, Logistic Regression, Naive Bayes classifier, Support Vector Machine can be used for the prediction of movie name and out of all the models, the model trained using Naive Bayes classifier and Support Vector Machine algorithm performed well in the prediction of the movie name, given the set of random sentences as input and the model trained using Random Forest Classifier has less performed compared to the remaining models.

Conclusions: Models are trained using 4 classification algorithms Random Forest, Logistic Regression, Naive Bayes classifier, Support Vector Machine identified from the literature review. For a random sample of paraphrased text inputs, all the models are tested in aim to get the appropriate movie name. Out of all the models, the model trained using Naive Bayes classifier and Support Vector Machine algorithm obtained the high accuracy.

Keywords: Natural Language Processing, Machine Learning, Classification algorithms, Movie Prediction.

Acknowledgments

We would like to thank our supervisor, Suejb Memeti for his support and encouragement for our thesis through his guidance and feedback. Finally, we would like to express our gratitude to all our friends and family.

Authors:
Bhargav Chinnapottu
Govardhan Arikatla

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the current world, movies are the best source of entertainment. Not only for entertainment but also, they are a major source of commerce, marketing, and benefits in the education. With the growth of different technologies, online streaming of movies and TV shows has become widely popular and there exist several streaming platforms like Netflix, Amazon Prime Video, and YouTube. The entire world is excited about the good streaming platforms with the best user interface and suggestion systems. Search suggestions have been a major issue that can help to seek the attention of the users, as it would be difficult for people to remember every movie name. Therefore, the suggestion or recommendation system for the movies had become a popular service over the audience in the society to get movie suggestions without remembering all the movie names.

Several works deal with classifying text by using Natural Language Processing (NLP). Some of the works include classifying the websites, books by genres or authors[33], and classifying the lyrics of the song into different genres[32]. The classification of the movies based on their summary or script involves a lot of work for the streaming platforms as they need to go through the entire movie script or summary manually. There are related works that include the model that predicts the rating of the movie based on the IMDB website using machine learning techniques[6], the model that predicts the movie genre based on the plot summaries of the movie using machine learning methods[12] and the model to predict the genre of the movie based on scripts using NLP techniques[5]. All the previous systems are developed using different artificial intelligence technologies such as machine learning classification, machine learning clustering, neural networks, and NLP techniques. As per the knowledge, the previous models deal with predicting the rating of the movie, predicting the genre of the movie, and no works related to the prediction of movie name based on movie scripts.

To predict the movie name using the human input, requires a human-computer interaction model. In this thesis to prepare a model that understands and interprets the human language to the computer, we have experimented with NLP techniques to extract the features from the human language text and classify them into different movie classes using machine learning classification algorithms.

The goal of this thesis is to identify the suitable machine learning classification algorithms that can be used for prediction and conduct the experiment to select the model with the best prediction accuracy including the collection of a dataset, preprocessing of the dataset before training the model using NLP techniques such as tokenization, stemming and prepare a bag of words model, train the model for the same dataset of movie scripts with all identified algorithms and select the model with best prediction accuracy among them.

1.1 Aims and Objectives

The thesis aims to develop a model using Natural Language Processing that gives an appropriate movie suggestion in which some plot or the scenario of the movie is given as an input to the model.

The objectives of the thesis include:

1. Gather and create a dataset with several movie names and their scripts to train the model.
2. Select the machine learning algorithms that can be used for training the model.
3. Train the model with prepared dataset after preprocessing with NLP and selected classification algorithms and validate the results of the model after testing with any plot in the movie in aim to get the appropriate movie name .
4. Analyze the results of the model with the test dataset.

1.2 Research Questions

To accomplish the aim, this thesis aims on the following research questions:

RQ1: Which classification algorithms can be used in predicting the movie name?
Motivation: The motivation for this research question is to conduct the literature study to identify the suitable classification algorithms that can be used for prediction of movie name by training the model with movie scripts data.

RQ2: How accurate the model trained using NLP in predicting the movie name?
Motivation: The motivation for this research question is to find the accuracy of the models trained in the prediction of movie name using NLP text preprocessing techniques and several classification algorithms.

1.3 Scope of the thesis

The focus of this thesis is to develop a model that predicts the name of the movie, when a random scenario from the movie is given as input. The model is developed using preprocessed dataset with the NLP techniques and classifying the script dataset using machine learning algorithms. As there is a possibility of same scenarios in more than one movie, the limitation of the developed model can be predicting the wrong movie name sometimes and we are not dealing with multi class prediction at present.

1.4 Overview

The thesis is classified into different chapters and the structure of the thesis is described below:
In chapter 2, the background of this thesis consists of information about the main field of the research and its applications in the real world. In chapter 3, the information about all the previous and related works in a similar field and the research gap that we focus on in this thesis is described. Chapter 4 describes the research method that we choose to do our research, the implementation of the experiment, and the information about the different algorithms. In chapter 5, the results from the literature study and the experiment are represented clearly. Chapter 6 deals with the analysis of the results from the experiment and a discussion about the thesis. Chapter 7 describes the conclusion of the thesis and information about the work that we would like to do in the future.

Chapter 2

Background

2.1 Machine Learning

Machine learning[3] is a subset of artificial intelligence that aims on training computers to learn from the data and develop with the knowledge of data. Machine learning applications become more efficient when they have more data to learn and improve with their use. Machine learning algorithms are used to train the model to develop patterns and correlations between the features in large datasets and to make predictions based on the knowledge of training. The different categories of machine learning algorithms are supervised learning, unsupervised learning, reinforcement learning.

2.1.1 Supervised learning

Supervised learning is a method of training the model with labeled datasets and is used to predict the test data based on the trained datasets. In supervised learning, the model is trained by feeding the model with input data and as well as output data, the model learns from the training data and after training, the algorithm predicts new data based on the learning from training. The goal of supervised learning is to develop a pattern or a procedure that predicts the new test data based on the analysis of training data that already has the class label. In supervised learning, the input labeled data acts as the reference to predict the test data correctly[18]. Supervised learning is classified into two types such as classification, regression. In classification, the algorithm predicts the class of the new test data and in regression, the algorithm predicts the real number of the new test data. Supervised learning is used in many real-world applications such as image classification, spam detection, risk assessment[25].

2.1.2 Unsupervised learning

Unsupervised learning[10] is a method of training the model using algorithms to analyze and cluster unlabeled datasets without any reference. It is like learning new things with the human brain. In unsupervised learning, only input data is provided to the model at the time of training but not the output data. Unsupervised learning aims to learn the structure of the dataset and predict the test data based on the similarities and characterize the data in a unique format. Unsupervised learning can be used, when there is no prior labeled data set for training and in more complex task processing. It requires minimum human supervision compared to supervised learning. Unsupervised learning is categorized into two types such as clustering, association. Unsupervised learning is used in many areas that include market basket analysis, pattern recognition, identifying accident-prone areas, and many business models[28].

2.1.3 Reinforcement learning

Reinforcement learning is an approach in which the machines learn by communicating with the environment. In this approach, the machine learns by performing different operations in which there will be rewards and punishment for each step. During the training of the model, there will be a reward for every appropriate action and a punishment for every inappropriate action. The main goal of reinforcement learning is to find the strategy such that it maximizes the number of rewards. In reinforcement learning, the machine works on its own without any supervision[35]. Various applications of reinforcement learning include industrial automation in robotics, development of games, marketing, and advertising[19].

2.2 Classification

Classification is the method of supervised learning that is used in the prediction of discrete data. In classification, the machine learns from the labeled data and classifies the data into different classes. Classification can be binary or multi-class classification based on the classes in the training dataset. In classification, the trained data is grouped into different target classes and the test data is predicted based on the target classes. Classification algorithms are applied in different real-world applications such as sentiment analysis, spam classification, and document classification[29].

2.3 Algorithms

The classification algorithms used for training the model in our thesis are Support Vector Machine, Naive Bayes classifier, Random Forest classifier, Logistic Regression algorithms.

2.3.1 Support Vector Machine

The Support Vector Machine (SVM) algorithm is a supervised learning algorithm that can be used for both regression and classification problems. The main aim of the Support Vector Machine algorithm is to find the best possible hyperplane that uniquely classifies all the data points plotted in the N-dimensional space. The hyperplane can be in different dimensions, which is based on the number of independent features in the dataset and the best hyperplane is chosen considering the largest margin or separation between the data classes. Support Vector Machine can be used for linear and non-linear classification problems by using different kernel functions and is efficient in high-dimensional approaches. The division of the hyperplane is represented in figure 2.1.

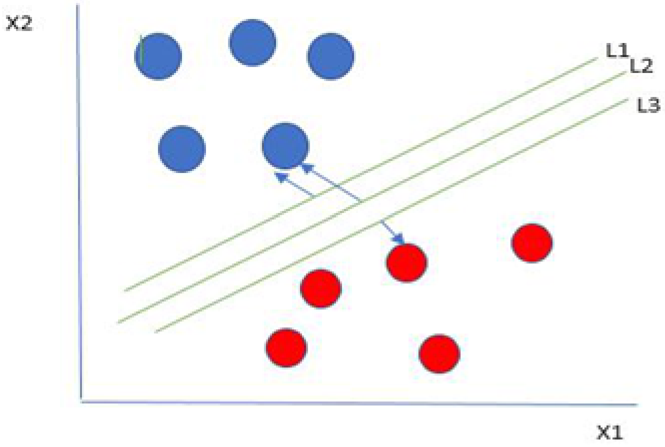


Figure 2.1: Support Vector Machine

2.3.2 Naive Bayes classifier

Naive Bayes classifier is a classification algorithm that depends on the Bayes theorem with an assumption of independence between the data points. It assumes that the presence of one data point is not correlated to the presence of another data point. There are three kinds of Naive Bayes approaches such as Gaussian, Multinomial, Bernoulli. Gaussian Naive Bayes approach is applied for the classification problems which is assumed to be a normal distribution, the Multinomial approach is applied for discrete data and Bernoulli is used for classifying, when the feature vectors are binary. Naive Bayes techniques are applied in real-world applications such as text classification, recommendation systems. Gaussian Naive Bayes distribution function is:

$$P\left(x_i y\right) = 1/\sqrt{2}\pi\sigma^2 exp(-\frac{\left(x_i - \mu_y\right)^2}{2\sigma_y^2}) where$$

$$\sigma_y and \mu_y$$

are the assumptions of likelihood.

2.3.3 Random Forest algorithm

Random Forest is one of the most frequently used machine learning approaches, which unites the output of different decision trees into a single output. It is easy to apply in classification and regression problems. This approach is used to obtain more efficient results when individual trees are not correlated with one another and larger the total of trees in the forest. It depends on the approach of ensemble learning, which is a concept of combining various classifiers to obtain a solution to a complex problem and to obtain the best accuracy. The Random Forest approach contains the random subsets of several decision trees and the final decision is the average decision of the subsets to give good prediction accuracy. The main inputs of the Random Forest algorithm are the number of trees to combine, the number of features to classify, and the size of a node that needs to specify before training the model. The tree structure of the Random Forest algorithm is represented in figure 2.2.

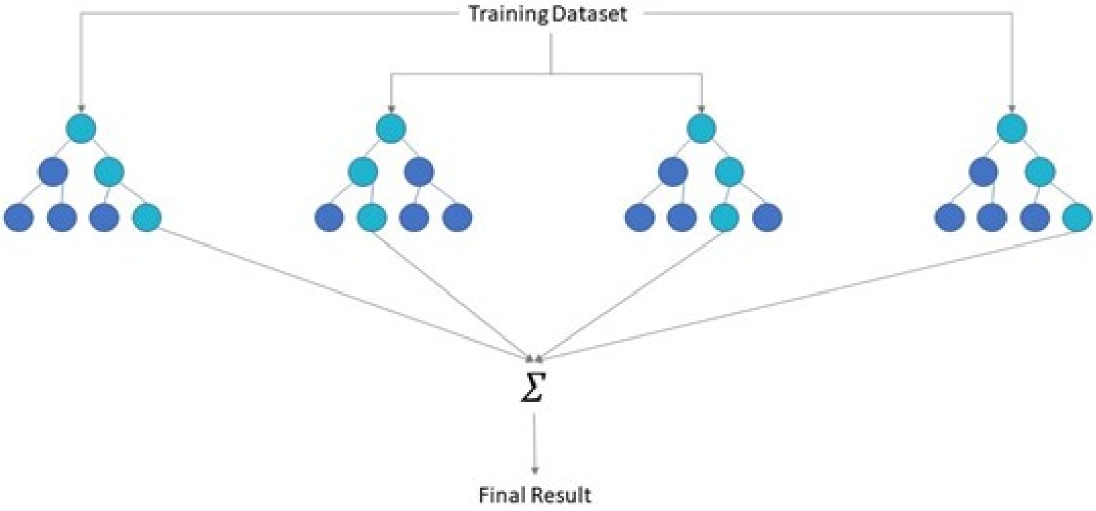


Figure 2.2: Random Forest Classifier

2.3.4 Logistic Regression

Logistic Regression algorithm is used in the classification to find the category of the dependent feature with the set of independent features. It is categorized into different types such as binary, multinomial, and ordinal logistic regression based on the dependent features. It predicts the likelihood of the test data and classifies it into one of the classes of dependent features. Only discrete data can be predicted using Logistic Regression which differs from Linear Regression, which can be used to predict the continuous data. Logistic Regression can be applied in many applications like predicting spam classification, the effect of the disease(low/high/medium) [26]. The logistic curve is represented in figure 2.3.

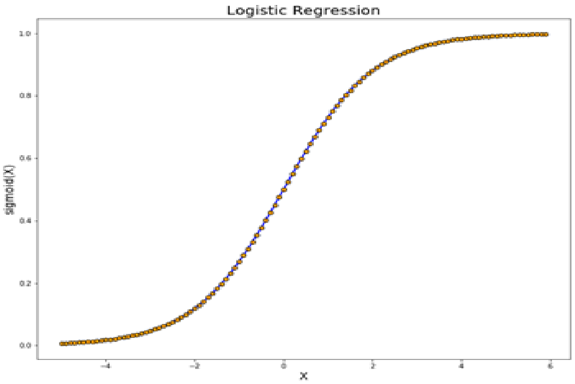


Figure 2.3: Logistic Regression

2.4 Natural Language processing

Natural Language Processing (NLP) is a branch of artificial intelligence that assists computers in understanding and explaining human language text or speech that contains a large amount of structured, semi-structured, and unstructured data. NLP defines the important parts of human language to computers and allows computers to interact with humans in their language, which involves a lot of processing. NLP can be applied to translation, search engine optimization, and filtering[7]. The available search suggestions when using the built-in search bar in most of the applications are based on NLP content categorization and topic modeling[22]. The emails are categorized due to the use of Bayesian spam filtering and a statistical model that compares the subject of the email with spam words to identify spam mails[27]. Customer feedback and reviews about any application or organization can be determined using sentiment analysis, which predicts the user's feelings about them by extracting information from various sources[40]. NLP techniques include:

Named Entity recognition

This is a popular technique of NLP that is used in information extraction. This approach takes the sentence of the text as input and identifies all the nouns present in the input text. It is widely used in applications like news content categorization, search engines to retrieve information easily[7].

Tokenization

Tokenization is the process of splitting the data into tokens like characters, words, sentences, numbers. Tokenization is used for effective storage space for the data and decreases search degree. It is applied in most information retrieval systems[21].

Stemming and lemmatization

Stemming is the process of decreasing all the words in the input text into their base or root form by removing the suffixes and making it easy for the model training. Lemmatization is used to obtain the proper vocabulary word for each word in the input text by transforming them to root form by understanding the meaning, parts of speech of the word.

Bag of Words

The bag of words model is used in text preprocessing to extract the features from the data to train the machine learning models. It counts the occurrence of each word in the sentence, and they can be represented in the form of vectors using vectorization. The bag of words model is used in several applications like email filtering, document classification[41].

Sentiment analysis

Sentiment analysis is one of the most used natural language techniques to predict the emotion or sentiment from the text. It is used to find the emotion of the text in any document, social media, news and classify them as positive, negative, or neutral. Sentiment analysis works best with the subjective data written by humans and predicts the emotion[40].

Natural Language generation

It is the process of converting the raw data into natural language text. Natural language generation techniques are used in organizations that contain a large amount of data and convert it into natural language making it easier for the machine to understand the patterns. Applications of natural language generation are document clustering, realization, content determination systems, and way-finding systems[8].

Chapter 3

Related Work

Aziz Rupawala et al. [31] have prepared a model to predict the movie genre from plot summaries by comparing various classification algorithms such as Multinomial Naive Bayes, Random Forest, Logistic Regression, SGD to obtain the best results in prediction. The authors mainly focused on selection of best suitable algorithms, comparison between different classifications and selected the algorithm which gives the maximum desired output in prediction by analyzing each model after the experiment.

Alex Blackstock et al. [5] have tried to classify the movie name based on genre using movie scripts data by developing a Logistic Regression model. In their model, they had considered features extracted from scripts using the NLP techniques. Naive Bayes classifier and Markov model classifier have used to analyze the performance metrics in their implementation. For each movie script, in the test dataset, the model predicts the possibility that the movie belongs to each genre based on extracted features and uses the k best score to predict genres, where k is a hyper-parameter.

Implementation of the model movie recommender system using machine learning algorithms like K means clustering and K nearest neighbor algorithms was performed by Rishabh Ahuja and his colleagues. In their research, they have studied different types of machine learning algorithms and after the study they got a clear picture of every algorithm and where they can be applied. The author proposed system predicts the movie based on user's preferences using different parameters[2].

Warda Ruheen Bristi et al. [6] have implemented a model that predicts the movie IMDB rating using machine learning techniques such as Bagging, Random Forest, Naive Bayes, J48, IB. There are some factors need to consider while predicting the rating of the movie, and in their research the authors considered the factors such as budget, actor, and director of the movie to predict the movie IMDB rating. Finally, they concluded with the result saying sanction and budget had a high effect on the movie success.

The research article [37] shows the implementation of a hybrid movie recommender system using sentiment analysis on spark platform to improve the accuracy of movie recommendation systems. The author states that comprehensive combination of emotions, reviews, and user preference can help to recommend the best movie. They have implemented the model with the combination of content-based filtering method and collaborative filtering to create a hybrid movie recommender system and sentiment analysis to enhance the accuracy in the prediction results.

In the research [12], the author developed a model for predicting the movie genre based on plot summaries. This article describes the implementation of several machine learning algorithms like Naive Bayes, Recurrent Neural Networks and Word2Vec + XGBoost for text classification and Probability threshold approach, K- binary transformation for genre selection to predict the movie genre based on summaries. The experiment is performed with more than 250,000 movies which concluded that the Gated Recurrent Units (GRU) neural networks with probability threshold approach reaches the best outcome on the test sample.

In the article[20], the implementation of a conversational movie search system using Conditional Random Fields is described. The authors developed a model that parses the spoken input into semantic classes using Conditional Random Fields and thereby searching for the movie in the indexed database with the help of recognized semantic classes. In their paper, authors mentioned the use of topic models in the input extension, vocabulary learning and different searching techniques for efficient searching of the database.

Vasu jain [15] has prepared a model that predicts the movie success using sentiment analysis from twitter data. They gathered data from the twitter about the several aspects that defines the movie popularity and manually labelled the training datasets into three classes as hit, flop, average. The test samples are predicted from the trained model which classified them into hit, flop, average classes. They have also considered the association between tweet time and tweet number.

In the [4] research, the authors developed a hybrid approach for the classification of emotions into happy, fear, anger, surprise, disgust, sad classes based on speech and text data. The researchers intend to analyze the speech and the consequent text of the speaker to predict the emotions of the speaker. For the implementation of the model, the researchers used NLP techniques like pos tagging, stop word removal for the feature's extraction and Support Vector Machine classifier to classify the emotions based on both speech and text. They aim to improve the efficiency of the emotion classification by considering both the audio and text feature vectors as a single feature vector which is then passed to the classifier.

From the above identified works, there are developments in the field of movie success prediction, movie recommendation systems, movie rating predictions and some related fields. In this thesis we are developing a model for the prediction of movie name based on the movie scripts, which is not implemented before.

Chapter 4

Method

Literature study and Experiment methods are chosen to answer the research questions. In the literature study, we have studied different machine learning algorithms and identified several classification algorithms that can be used to predict the movie name based on a script and learn about the NLP techniques to implement the data preprocessing before training the model. The data is gathered from the IMSDb source, which contains a movie script database and is stored in the file to prepare a dataset. The experiment is performed with the gathered dataset and algorithms identified from the literature study.

4.1 Literature Study

To identify the classification algorithms that can be used for the classification of movie script data into different movies, we choose Systematic Literature Review(SLR) [39] as our research procedure that helps to find the works in the same field and analyze them using standard procedures.

- Identified the key words like movie prediction, classification algorithms, Natural Language Processing, which are main fields of our thesis.
- By using the identified key words, searched for the sources of related works in IEEE, Google Scholar, Science Direct and the Arxiv public repositories.
- From the results obtained from the search of key words, gathered some research articles that are related to Natural Language Processing, movie recommendation or prediction works, and classification algorithms used for the prediction.
- After collecting all the related works, the inclusion and exclusion criteria is implemented to filter the important works in the field.

  Inclusion criteria:

  1. The inclusion of studies that are related to the field machine learning predictions and Natural Language Processing.
  2. Include only published articles.
  3. Include the articles in english language.
  4. Peer review of all the research articles and journals related to our work.

Exclusion criteria:

1. Consideration of reviews and editorials from unknown sources.
2. Incomplete research articles.

- After the implementation of inclusion and exclusion criteria, the suitable research works are selected.
- From these works, all the findings are reviewed, and the conclusions are presented after analysis.

4.2 Experiment

Experiment is performed using the data gathered and the algorithms identified from the literature study to predict the movie name based on the movie scripts. The experiment is performed in 5 stages.

1. Gather the data from the IMSDb source[1] using web scraping techniques and prepare a dataset.
2. Preprocess the dataset using NLP techniques.
3. Train the model with preprocessed dataset and different classification algorithms which are identified during the literature study.
4. Test the model with various test data.
5. Evaluate the results of the implemented model.

The working environment and software used for the experiment are mentioned.

4.2.1 Working Environment

For experimentation, we have used a laptop that have the following specifications. The given specifications are the updated version as of now, we have provided with version numbers and description below.

1. Windows 10 64-bit operating system.
2. AMD RYZEN 5 3550H with Radeon Vega Mobile Gfx 2.10Ghz / Intel core i5-9300H CPU 8.00 GB RAM that runs at 2667MHz.
3. Python -V 3.9.2, an open-source programming language that is dynamically programmed and supports multiple programming including functional, object-oriented programming.
4. Anaconda -V 1.7.2, an open-source environment consists of data science packages and available for windows, macOS, Linux operating systems. It is used to build and run the machine learning models.

We have used python as the programming language for the implementation and used the following python libraries:

- numPy -V 1.20.2, an open-source python package for N-dimensional arrays and numerical computations. It consists of several collection of classes which can be used to perform different mathematical operations
- pandas -V 1.0.5, an open-source python library for fast, flexible operations and good tool for data manipulation and analysis. It is used to create data frames and perform operations on data frames.
- nltk -V 3.5 is used to preprocess the data provided by humans into machine understandable language. It is the main platform that has the modules to perform human language related operations.
- scikit-learn -V 0.23.1, an open-source python tool used for simple and efficient tools for predictive analysis. It contains most of the machine learning algorithms in it.
- regex -V 2020.6.8, an open-source library used to clean the text that contains the unwanted information in it.
- matplotlib -V 3.2.2 is a comprehensive python library for creating interactive and animated visualizations in python.
- requests -V 2.24.0 is used to get URL requests, no need to do manually.
- pickle -V 0.7.5 is used for serializing and de-serializing binary protocols
- seaborn, a python library for the visualization of the data.

4.2.2 Data Collection

The dataset of movie scripts is not found online in any sources, so we have gathered the data from the source website 'https://imsdb.com/', which is a database of several movies. We have used web scraping techniques to scrap the script data from the source using python programming libraries like requests to access the URL of the source, BeautifulSoup to get data from the HTML source tags, and stored the data in the form of text files with the movie names as file names in the local system. Initially, we have tried to experiment with the dataset of 11 movie scripts. For that, we have stored the 11 movie scripts with their movie names as the filename in the local system in a way that it can be accessed at the time of training the model. There are no independent variables present in this dataset as we considered only the dependent variables like movie names to obtain the best results.

4.2.3 Data Preprocessing

Before training the model, the movie script data needs to be preprocessed as it contains a lot of unwanted data like stop words, punctuation, etc. To preprocess the data, we have used NLP techniques and regular expression libraries. Preprocessing of the data includes stop word removal, punctuation removal, tokenization, stemming, and vectorization.

- Removing Unwanted data:
  Unwanted data can be removed from the data by using regular expression library. It is used to remove unwanted spaces and unwanted data like numerical data, punctuation characters and special characters which are hard to understand by the machine.

- Tokenization:
  Tokenization of the data is the process of breaking our data into semantic units with basic meaning. It is done to break the script data into various sentences or words. We have used nltk.stem library to tokenize the script data.

- Stemming:
  Stemming is used to obtain the root word for each word in the script data, but the base meaning of every word remains same. It reduces the size of vocabulary and makes it easier for the model to understand the words. Stemming helps to train the model efficiently. we have used nltk library of python for stemming the script data.

- Vectorization:
  Machines cannot understand the human language words, but it can understand numbers. To make machine understandable, we need to convert our text words to numbers. This can be done by vectorization process. There are many ways to vectorize the text. For this model we have used Bag of Words and TF-IDF.

- Bag of Words (BoW):
  Bag of Words model is the representation of the text data in the form of vectors which are machine understandable. It calculates total number of occurrences of each word in the sentence.It represents the words and their corresponding occurrences in each sentence of the data in the form of vectors in a matrix.

- Term Frequency-Inverse Document Frequency(TF-IDF):
  This model is used to compare similarity between documents given. It helps us to assign importance for each word in the documents, TF-IDF is calculated by using the importance of Term Frequency and Inverse Document Frequency of the word in the document. There is a possibility of occurrence of an important word less in the document. To resolve this problem, TF-IDF model is used to assign the importance to each word which reduces the missing chance of important words. It calculates the term frequency and inverse document frequency of each word in the document and interprets them as vectors in the matrix.

- Data manipultion:
  The data from text file is loaded into the model using the python library sklearn.datasets. This creates two variables which contains movie scripts data in one variable and movie names as random numbers in another variable.

This preprocessed data from the above methods is used to train the model for predictions.

4.2.4 Training the model

In the first step, we have loaded the data from the dataset that contains movie scripts stored each in different text files with their movie names as filenames in the local system using sklearn.datasets library. The dataset is divided into two variables which contain movie scripts in one variable and movie names as numbers in another variable which is the dependent variable for prediction.

Regular expression operations are used to eliminate white spaces, unwanted text, special characters, and numeric characters from the scripts of the movies variable using the regex python package. After data is cleaned using a regular expression, the movies scripts variable undergoes different methods to get converted into machine understandable language, from text to vectors. Firstly, the Bag of Words model is applied to get a count of the words present in each sentence of each document within each movie scripts by using a count vectorizer. Then, text from the dataset is transformed to TF-IDF vectors using TF-IDF transformer. It transforms the text in the movie script documents to TF-IDF vectors. These vectors represent the term frequency and inverse document frequency of words present in the movie script documents and represent those vectors in the form of a matrix using the words and their importance in the document. We must select the frequency of the words in the documents while training the model. We have selected up to 3000 frequent words using the max features attribute of TfidfVectorizer.

Machine learning classification algorithms identified from the literature study such as Logistic Regression, Naive Bayes, Support Vector Machine, and Random Forest classifier are used to train the model with movie scripts data. In the next step, we have selected a classification algorithm that classifies the movie scripts into different movie classes. The same preprocessed dataset of movie scripts is used for the training with each of the identified classification algorithms. First, we have used the Logistic Regression algorithm to train the model with the given dataset. It transforms variables of the dataset to fit the model perfectly. After the Logistic Regression model, we have used the Naive Bayes algorithm, Support Vector Machine, and Random Forest classifier algorithms for transforming and fitting dataset into each model and finally predicting results for the given scenario from the movie script as input.

For using the model in the future, the model is saved the using python pickle library. We have dumped all the required information into the pickle file. For the future usage of the models, we can load the pickle file and can be used for the prediction of movie names.

4.2.5 Performance Metrics

All the machine learning models have some metrics to measure the performance of the algorithms. As our thesis deals with classification, we have considered accuracy as the performance metric.

Accuracy:
Accuracy can be defined as the total number of correct predictions of data out of total data points in the test dataset. Usually, accuracy is the most used performance metric for classification models. Using accuracy metrics, we can compare all the models with different classification algorithms and select the model with good accuracy.

$$Accuracy = \frac{Total\ number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

4.2.6 Testing the model

The movie scripts data is paraphrased, and some random sentences are taken as the text inputs for test dataset. These random sentences have been used for the prediction of the movie name using all the trained models with classification algorithms and script data. The results of the prediction for all the models are noted for all the text inputs and the results are compared. All the results are presented in the table and the prediction of each model is represented using the confusion matrix. The accuracy of each model in prediction of movie name is noted and compared to select the best algorithm among the trained models.

Chapter 5

Results

The results from the Systematic Literature Review and experiment are mentioned in this section.

5.1 Results from the Literature Study

To find the best classification algorithms that can be used for the movie name prediction, we have used Systematic Literature Review method. We have gathered the following works from the study that could help us to identify best classification algorithms to get good accuracy in prediction. The research articles and the identifications from the literature study are presented in the table 5.1.

SNo	Title of the research	Identifications
1	Prediction of diabetes disease among women using classification algorithms.	In this research, the authors used classification algorithms like **Naive Bayes, SVM, and Decision Tree** to predict the diabetes disease among pregnant women using pidd dataset and concluded that **Naive Bayes** algorithm has obtained high accuracy of 76%[34].
2	Crime category prediction using classification algorithms.	The research is about comparing two classification algorithms like **Decision Tree algorithm, Naive Bayes** algorithm for the prediction of crime category and concluded that **Decision Tree** algorithm performs well with 84% accuracy[14].

Table 5.1 – continued from previous page

SNo	Title of the research	Identifications
3	Classification algorithms for prediction of kidney disease.	This research focused on predicting the kidney disease using classification algorithms like **Naive Bayes and Support Vector Machine (SVM)** and results concluded that **SVM** algorithm has best performed in the prediction[36].
4	Performance of student prediction using data mining techniques.	This research is about the predicting the student performance using the dataset of student information using **Rule learner, a Decision Tree classifier, a Neural Network, and a K-Nearest Neighbor classifier**. The **Neural Network** model predicted with high accuracy when compared to the remaining algorithms[16].
5	Movie popularity prediction using machine learning techniques.	The researchers tried to predict the movie popularity using the IMDB movie source as the dataset and machine learning techniques. They found that the model trained using **Logistic Regression** performs well with an accuracy of 84%[17].
6	Movie success prediction using machine algorithms and comparison	This research concludes that **Random Forest classifier** predicts best when compared to remaining four algorithms **Support Vector Machine, Ada Boost, Gradient Boost, and K-Nearest Neighbor classifier** algorithms in case of movie success prediction[9].
7	Sentiment analysis of movies using Genetic and Naive Bayes algorithm.	This study shows that the hybrid approach of **Naive Bayes and Genetic** algorithm performs best for analyzing the sentiment from movie reviews[11].

Table 5.1 – continued from previous page

SNo	Title of the research	Identifications
8	Classification of movie into different genres based on audio and video features using Support Vector Machine algorithm.	They proposed a model that takes the 277 features from different audio and visual sources using self-adaptive harmony search algorithm and these features are fit into a model with **Support Vector Machine** classifier that classifies the movies into different genres. They concluded that the model obtained the accuracy of 92%[13].
9	Classification of textual reviews using sentiment analysis.	In this research, textual reviews are classified into positive, negative, neutral reviews using Bag of Words model with **Support Vector Machine classifier** and used **Fuzzy** classification algorithm to improve the efficiency in classification[23].
10	Predicting the IMDB rating of movies using ensemble learning methods and classification algorithms.	The research is done to predict the IMDB rating of the movies. For that, Hollywood movies data is taken from Wikipedia as the dataset and model is trained using algorithms like **J48, IBK, Random Forest classifier, Naive Bayes** classifier and bagging. It concluded that **Random Forest** classifier predicts the rating with accuracy of 99% [6].
11	Classification algorithms and NLP applied to narrative reports.	The researchers experimented with general purpose natural language processer to convert the human language text into machine understandable form and used different classification algorithms like **Decision Trees, Rule Generation, information retrieval, Naive Bayes** algorithms to classify about 200 chest X-ray reports[38].

Table 5.1 – continued from previous page

SNo	Title of the research	Identifications
12	Automatic classification of Russian Scientific Texts using the application of NLP approaches .	This work is done to develop a model that automatically groups the scientific abstracts and articles. They have used NLP techniques for text preprocessing and algorithms like **Support Vector Machine, Random Forest, Artificial Neural Networks, and Logistic Regression** to prepare a model. In the results, they have concluded that **Support Vector Machine** based model is the best performing model[30].
13	Analysis of music lyrics using NLP.	In this model, the researchers used **NLP** approaches to extract the text lyrics from the songs which can be used as a metadata for training another machine learning models[24].

Table 5.1: Results from Literature Study

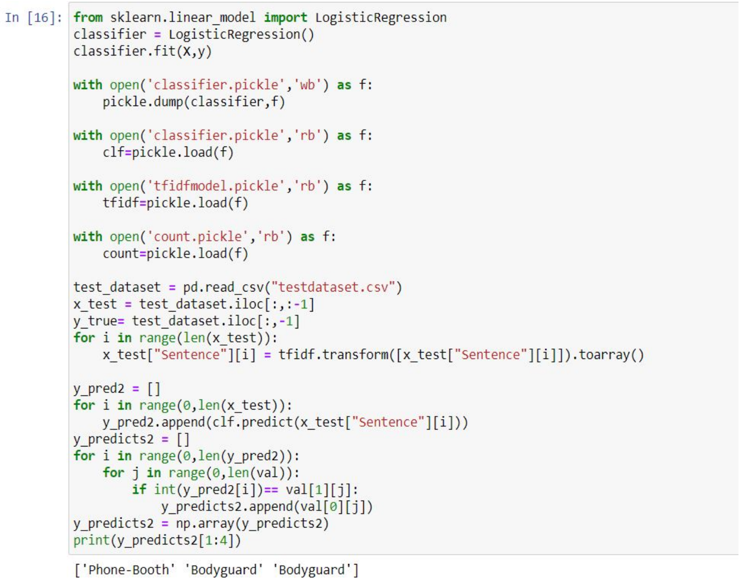
From the study of above research papers using Systematic Literature Review, we have identified that Support Vector Machine (SVM), Naive Bayes classifier, Random Forest classifier, and Logistic Regression algorithms can be used for training the model to get the movie name predictions.

5.2 Experimental Results

The results of the experiment with preprocessed dataset and the identified machine learning algorithms from literature study are presented in this section. Four algorithms Support Vector Machine (SVM), Naive Bayes classifier, Random Forest classifier, and Logistic Regression are used to train the model individually for the same dataset of movie scripts and the predictions of some random paraphrased sentences from the script are done for each model and recorded.

5.2.1 Logistic Regression Results

The model is trained with Logistic Regression algorithm and preprocessed dataset with NLP techniques in aim to predict the movie name based on the script. The configuration of Logistic Regression model and example prediction of inputs from the test dataset which contains random scenarios of the movie is shown in the code snippet figure 5.1.



```
In [16]: from sklearn.linear_model import LogisticRegression
         classifier = LogisticRegression()
         classifier.fit(X,y)

         with open('classifier.pickle','wb') as f:
             pickle.dump(classifier,f)

         with open('classifier.pickle','rb') as f:
             clf=pickle.load(f)

         with open('tfidfmodel.pickle','rb') as f:
             tfidf=pickle.load(f)

         with open('count.pickle','rb') as f:
             count=pickle.load(f)

         test_dataset = pd.read_csv("testdataset.csv")
         x_test = test_dataset.iloc[:,:-1]
         y_true= test_dataset.iloc[:,-1]
         for i in range(len(x_test)):
             x_test["Sentence"][i] = tfidf.transform([x_test["Sentence"][i]]).toarray()

         y_pred2 = []
         for i in range(0,len(x_test)):
             y_pred2.append(clf.predict(x_test["Sentence"][i]))
         y_predicts2 = []
         for i in range(0,len(y_pred2)):
             for j in range(0,len(val)):
                 if int(y_pred2[i])== val[1][j]:
                     y_predicts2.append(val[0][j])
         y_predicts2 = np.array(y_predicts2)
         print(y_predicts2[1:4])

         ['Phone-Booth' 'Bodyguard' 'Bodyguard']
```

Figure 5.1: Prediction of movie name using Logistic Regression

The predictions of random paraphrased sentences from different movie scripts using the model trained with Logistic Regression algorithm are represented along with the actual movie names in the form of confusion matrix figure 5.2

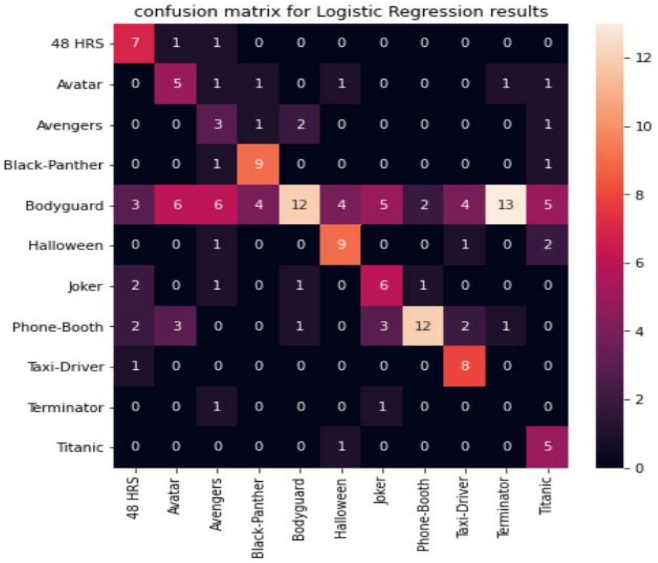
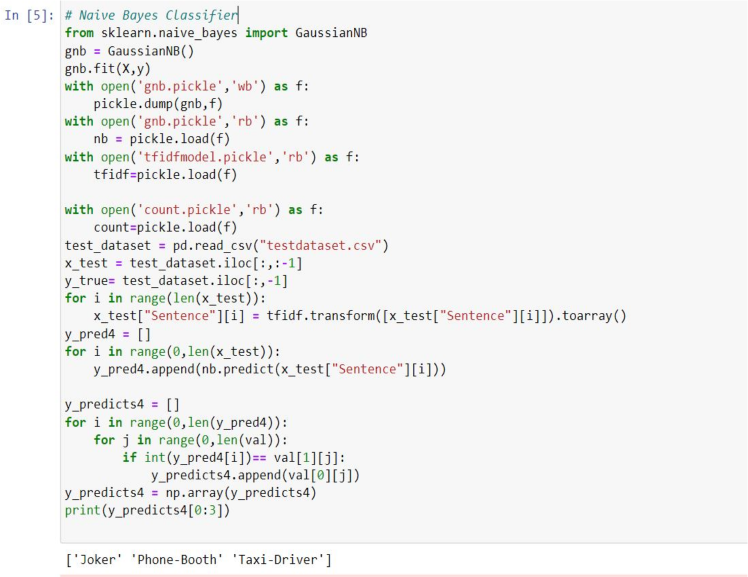


Figure 5.2: Logistic Regression confusion matrix

5.2.2 Naive Bayes results

The model is trained with Naive Bayes algorithm and preprocessed dataset with NLP techniques in aim to predict the movie name based on the script. The configuration of Naive Bayes classifier model and example prediction of inputs from the test dataset which contains random scenarios of the movie is shown in the code snippet figure 5.3.



```
In [5]: # Naive Bayes Classifier
        from sklearn.naive_bayes import GaussianNB
        gnb = GaussianNB()
        gnb.fit(X,y)
        with open('gnb.pickle','wb') as f:
            pickle.dump(gnb,f)
        with open('gnb.pickle','rb') as f:
            nb = pickle.load(f)
        with open('tfidfmodel.pickle','rb') as f:
            tfidf=pickle.load(f)

        with open('count.pickle','rb') as f:
            count=pickle.load(f)
        test_dataset = pd.read_csv("testdataset.csv")
        x_test = test_dataset.iloc[:,:-1]
        y_true= test_dataset.iloc[:,-1]
        for i in range(len(x_test)):
            x_test["Sentence"][i] = tfidf.transform([x_test["Sentence"][i]]).toarray()
        y_pred4 = []
        for i in range(0,len(x_test)):
            y_pred4.append(nb.predict(x_test["Sentence"][i]))

        y_predicts4 = []
        for i in range(0,len(y_pred4)):
            for j in range(0,len(val)):
                if int(y_pred4[i])== val[1][j]:
                    y_predicts4.append(val[0][j])
        y_predicts4 = np.array(y_predicts4)
        print(y_predicts4[0:3])

        ['Joker' 'Phone-Booth' 'Taxi-Driver']
```

Figure 5.3: Prediction of movie name using Naive Bayes

The predictions of random paraphrased sentences from different movie scripts using the model trained with Naive Bayes classifier are represented along with the actual movie names in the form of confusion matrix figure 5.4

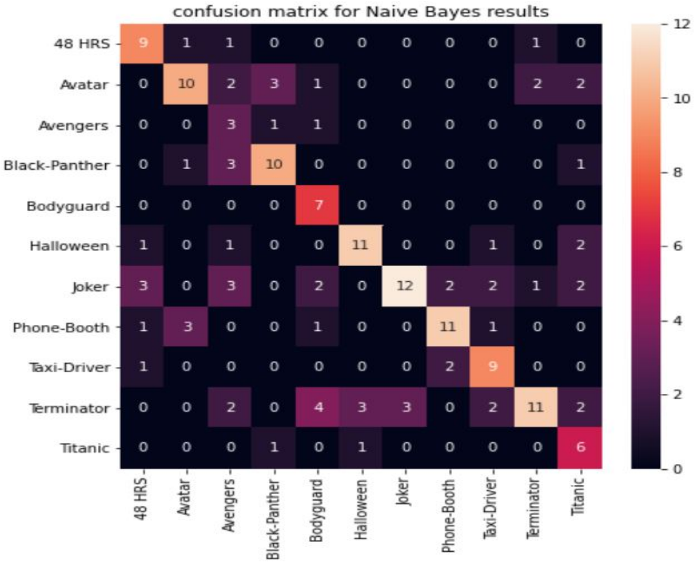
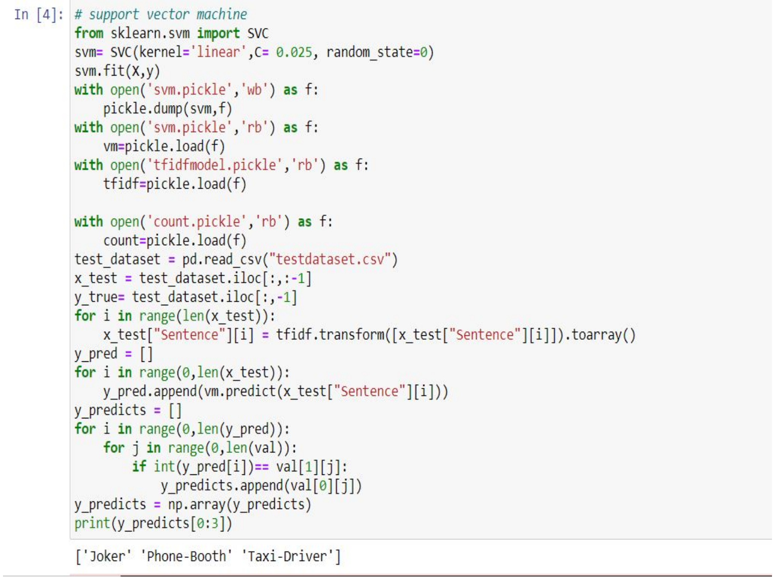


Figure 5.4: Naive Bayes confusion matrix

5.2.3 Support Vector Machine results

The model is trained with Support Vector Machine algorithm and preprocessed dataset with NLP techniques in aim to predict the movie name based on the script. The configuration of Support Vector Machine model and example prediction of inputs from the test dataset which contains random scenarios of the movie is shown in the code snippet figure 5.5.



```
In [4]: # support vector machine
        from sklearn.svm import SVC
        svm= SVC(kernel='linear',C= 0.025, random_state=0)
        svm.fit(X,y)
        with open('svm.pickle','wb') as f:
            pickle.dump(svm,f)
        with open('svm.pickle','rb') as f:
            vm=pickle.load(f)
        with open('tfidfmodel.pickle','rb') as f:
            tfidf=pickle.load(f)

        with open('count.pickle','rb') as f:
            count=pickle.load(f)
        test_dataset = pd.read_csv("testdataset.csv")
        x_test = test_dataset.iloc[:,:-1]
        y_true= test_dataset.iloc[:,-1]
        for i in range(len(x_test)):
            x_test["Sentence"][i] = tfidf.transform([x_test["Sentence"][i]]).toarray()
        y_pred = []
        for i in range(0,len(x_test)):
            y_pred.append(vm.predict(x_test["Sentence"][i]))
        y_predicts = []
        for i in range(0,len(y_pred)):
            for j in range(0,len(val)):
                if int(y_pred[i])== val[1][j]:
                    y_predicts.append(val[0][j])
        y_predicts = np.array(y_predicts)
        print(y_predicts[0:3])

        ['Joker' 'Phone-Booth' 'Taxi-Driver']
```

Figure 5.5: Prediction of movie name using Support Vector Machine

The predictions of random paraphrased sentences from different movie scripts using the model trained with Support Vector Machine algorithm are represented along with the actual movie names in the form of confusion matrix figure 5.6

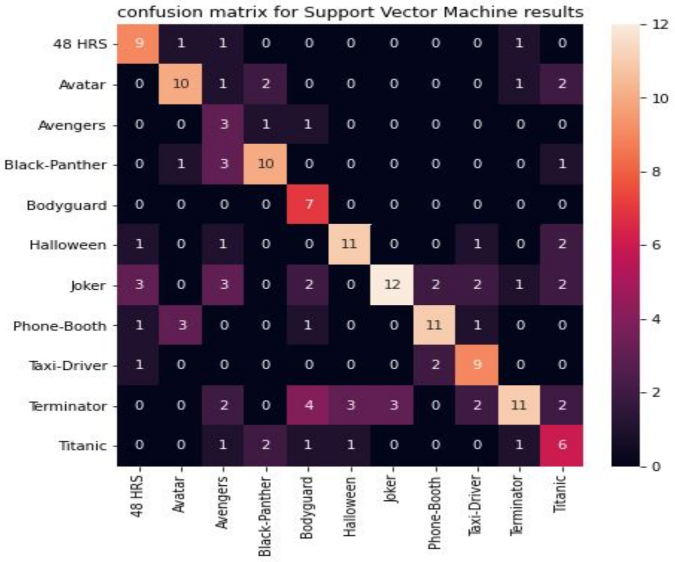
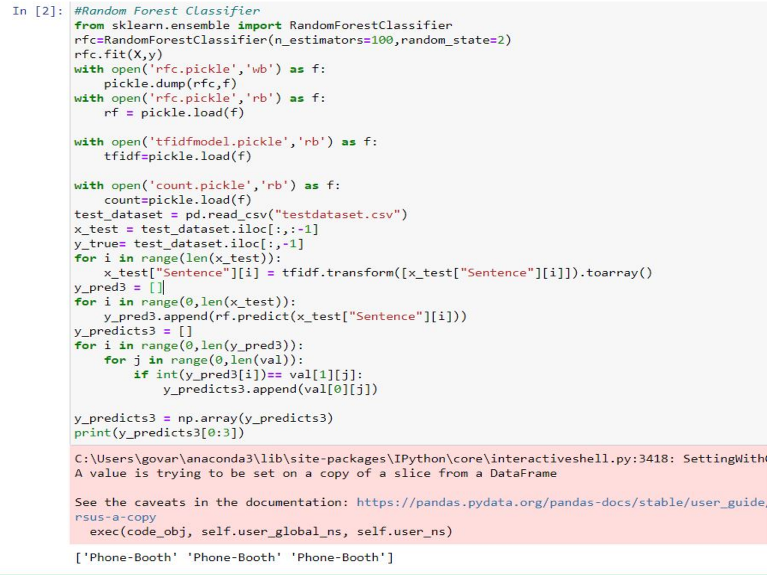


Figure 5.6: Support Vector Machine confusion matrix

5.2.4 Random Forest classifier results

The model is trained with Random Forest classifier algorithm and preprocessed dataset with NLP techniques in aim to predict the movie name based on the script. The configuration of Random Forest classifier model and example prediction of inputs from the test dataset which contains random scenarios of the movie is shown in the code snippet figure 5.7.



```
In [2]: #Random Forest Classifier
        from sklearn.ensemble import RandomForestClassifier
        rfc=RandomForestClassifier(n_estimators=100,random_state=2)
        rfc.fit(X,y)
        with open('rfc.pickle','wb') as f:
            pickle.dump(rfc,f)
        with open('rfc.pickle','rb') as f:
            rf = pickle.load(f)

        with open('tfidfmodel.pickle','rb') as f:
            tfidf=pickle.load(f)

        with open('count.pickle','rb') as f:
            count=pickle.load(f)
        test_dataset = pd.read_csv("testdataset.csv")
        x_test = test_dataset.iloc[:,:-1]
        y_true= test_dataset.iloc[:,-1]
        for i in range(len(x_test)):
            x_test["Sentence"][i] = tfidf.transform([x_test["Sentence"][i]]).toarray()
        y_pred3 = []
        for i in range(0,len(x_test)):
            y_pred3.append(rf.predict(x_test["Sentence"][i]))
        y_predicts3 = []
        for i in range(0,len(y_pred3)):
            for j in range(0,len(val)):
                if int(y_pred3[i])== val[1][j]:
                    y_predicts3.append(val[0][j])

        y_predicts3 = np.array(y_predicts3)
        print(y_predicts3[0:3])

        C:\Users\govar\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3418: SettingWith
        A value is trying to be set on a copy of a slice from a DataFrame

        See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
        rsus-a-copy
          exec(code_obj, self.user_global_ns, self.user_ns)

        ['Phone-Booth' 'Phone-Booth' 'Phone-Booth']
```

Figure 5.7: Prediction of movie name using Random Forest classifier

The predictions of random paraphrased sentences from different movie scripts using the model trained with Random Forest classifier are represented along with the actual movie names in the form of confusion matrix figure 5.8.

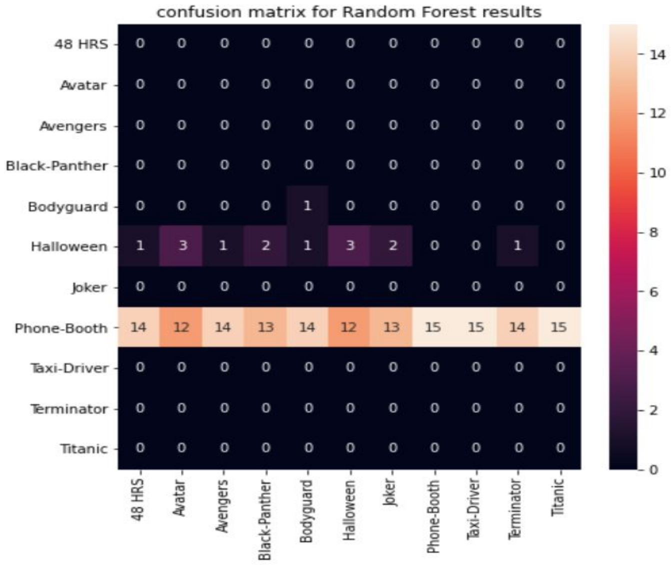


Figure 5.8: Random Forest classifier confusion matrix

5.3 Comparision of results

The predictions of some random paraphrased sentences from movie scripts using all the models trained using Logistic Regression, Support Vector Machine (SVM), Naive Bayes classifier, Random Forest classifier algorithms are compared with original movie names in the table 5.2.

Input Text	**Original Movie Name**	**Logistic Regression**	**Support Vector Machine**	**Naive Bayes**	**Random Forest**
Henry examines Frank,then sets his cloth down and walks toward the entrance where Frank is parked.	Body guard	**Body guard**	**Body guard**	**Body guard**	Phone Booth
Sarah is rushing around, attempting to meet the beginning of the dinner rush. She is 'in it,' as waitresses say.	Terminator	Body guard	**Terminator**	**Terminator**	Phone Booth
As Panther approaches, Nakia leaps to his feet, kicking the young militant's gun from his grip and grabbing him in a neck-lock.	Black Panther	**Black Panther**	**Black Panther**	**Black Panther**	Phone Booth
On the street behind him,an OLD MAN plays the piano. Both are there to draw attention to the store's big sale.	Joker	**Joker**	**Joker**	**Joker**	Phone Booth
"When it rains, the taxi driver is the boss of the city," goes the cabbie's maxim, which was proven true by the night's activity.	Taxi Driver	**Taxi Driver**	**Taxi Driver**	**Taxi Driver**	Phone Booth
Rachel erotically dances with the man, bumping and grinding and sinking to her knees. The audience applauds.	Bodyguard	Joker	Joker	Joker	Phone Booth

The disgruntled restaurateur stares at him through the glass before giving up and walking away,talking to himself as he goes up the block.	Phone Booth	**Phone Booth**	**Phone Booth**	**Phone Booth**	**Phone Booth**
Jack and Tommy sprint up the stairs with the bench and ram it into the gate with all their might.	Titanic	**Titanic**	**Titanic**	**Titanic**	Phone Booth
Jake comes to a halt,unnoticed, next to the bully. He leans in, grabs one of the man's barstool legs, and YANKS.	Avatar	**Avatar**	**Avatar**	**Avatar**	Halloween

Table 5.2: Comparison of predictions

From the results obtained from predictions, the accuracy obtained for each of the algorithms is plotted as a histogram. It is represented in figure 5.9

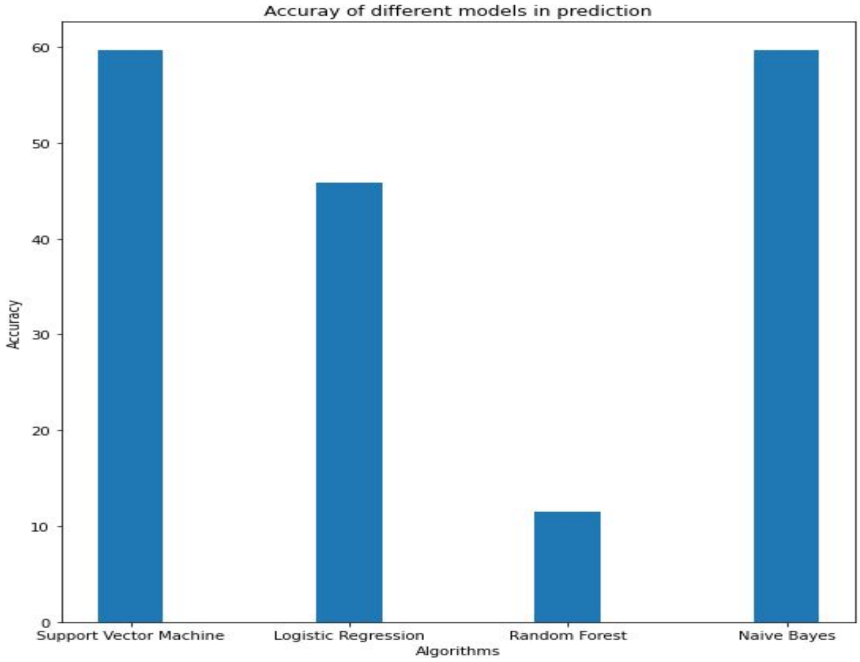


Figure 5.9: Comparison of accuracy plot

From the experiment, the accuracy obtained in prediction of movie name for each of the trained models, when several random scenarios from different movies are given as input is presented in the form of table 5.3.

Model trained using algorithm	**Accuracy**
Logistic Regression model	46 %
Naive Bayes model	59.63 %
Random Forest model	11.52 %
Support Vector Machine model	59.63 %

Table 5.3: Comparison of accuracy

Chapter 6

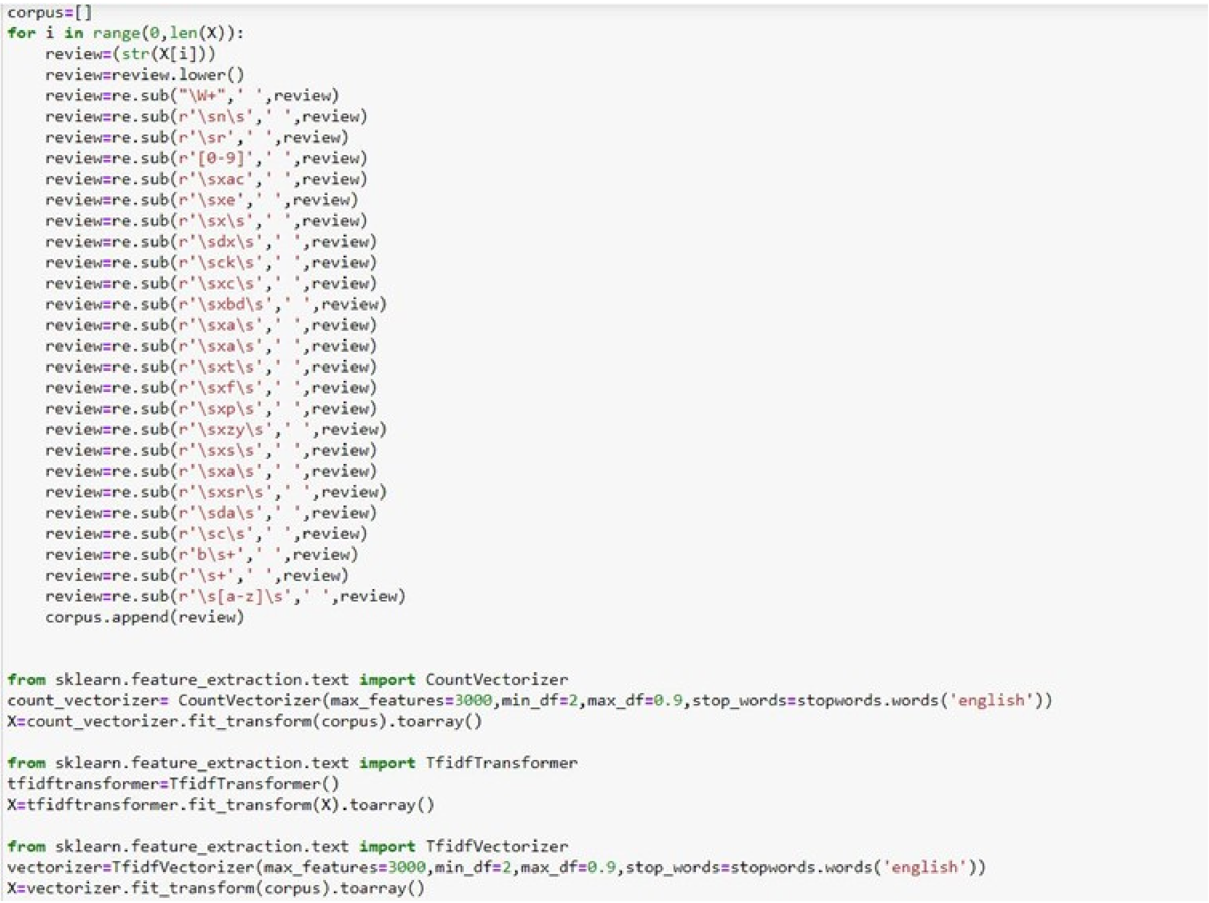
Analysis and Discussion

From table 5.1, it represents several articles, research papers from the literature study which are obtained by using the search of keywords such as Natural Language Processing, classification algorithms, movie prediction in repositories like Google Scholar, IEEE, Science Direct. All the results related to our research are noted down which includes the use of classification in different prediction models, movie-related models that use machine learning techniques, and NLP-based articles that are used in the preprocessing of data. We have observed that there are several machine algorithms such as Naive Bayes, Decision Tree, Support Vector Machine, K- Nearest Neighbor classifier, Neural Network, Logistic Regression, Random Forest, Ada Boost, Gradient Boost, and some more algorithms are mostly used machine learning algorithms in the classification models. From the study of all the conclusions of research papers, we have opted to use 4 algorithms Support Vector Machine, Random Forest, Naive Bayes classifier, and Logistic Regression in the prediction of movie name using movie scripts data to obtain good prediction accuracy.

From the table 5.2, we have got different results for all the models. However, we have got the results as expected, there is a very great chance of getting the wrong predictions in many cases because there may be the same situation or plot in one or more movie scripts. This makes the machine difficult to recognize the exact movie name and it may predict one of the movie names that have the same plot which leads to a challenge in the prediction. In such cases, multi-classification algorithms can be used which may be helpful to increase the accuracy of the prediction. From the table 5.3, it represents the performance of the different models with their accuracies made over prediction of random samples of sentences from movie scripts. It shows that model trained using Support Vector Machine and the Naive Bayes model has got the same accuracy in prediction which is the highest accuracy among the trained models and the model trained with Random Forest algorithm has got the least accuracy in prediction.

The figures 5.2, 5.4, 5.6, 5.8, represents the confusion matrix that shows the performance of each algorithm in the prediction of movie names. In the confusion matrix, the predicted movie names are represented along with their actual movie names such that the diagonal elements of the matrix corresponds to the correct predictions of each movie class and the remaining elements of each column correspond to their respective predictions. For example, in the figure 5.4, the prediction using the Naive Bayes classifier algorithm, the test data that has an actual class name as 48 HRS has given 9 correct predictions, 1 prediction as Halloween, 3 predictions as Joker, 1 prediction as Phone-Booth, and 1 prediction as Taxi-Driver movie classes.

From section 4.3.3, for transforming the natural language text of script data to vectors, we have implemented several processing techniques using NLP and regular expressions. The below code snippet is the implementation of text preprocessing which results in the removal of stop words, punctuation characters, transforming all the text into lower case characters, performing tokenization of the entire text, and using the most frequent and important 3000 words from each script to interpret them in the form of vectors using TfidfVectorizer. The implementation of preprocessing is shown in figure 6.1



```
corpus=[]
for i in range(0,len(X)):
    review=(str(X[i]))
    review=review.lower()
    review=re.sub("\W+",' ',review)
    review=re.sub(r'\sn\s',' ',review)
    review=re.sub(r'\sr',' ',review)
    review=re.sub(r'[0-9]',' ',review)
    review=re.sub(r'\sxac',' ',review)
    review=re.sub(r'\sxe',' ',review)
    review=re.sub(r'\sx\s',' ',review)
    review=re.sub(r'\sdx\s',' ',review)
    review=re.sub(r'\sck\s',' ',review)
    review=re.sub(r'\sxc\s',' ',review)
    review=re.sub(r'\sxbd\s',' ',review)
    review=re.sub(r'\sxa\s',' ',review)
    review=re.sub(r'\sxa\s',' ',review)
    review=re.sub(r'\sxt\s',' ',review)
    review=re.sub(r'\sxf\s',' ',review)
    review=re.sub(r'\sxp\s',' ',review)
    review=re.sub(r'\sxzy\s',' ',review)
    review=re.sub(r'\sxs\s',' ',review)
    review=re.sub(r'\sxa\s',' ',review)
    review=re.sub(r'\sxsr\s',' ',review)
    review=re.sub(r'\sda\s',' ',review)
    review=re.sub(r'\sc\s',' ',review)
    review=re.sub(r'b\s+',' ',review)
    review=re.sub(r'\s+',' ',review)
    review=re.sub(r'\s[a-z]\s',' ',review)
    corpus.append(review)


from sklearn.feature_extraction.text import CountVectorizer
count_vectorizer= CountVectorizer(max_features=3000,min_df=2,max_df=0.9,stop_words=stopwords.words('english'))
X=count_vectorizer.fit_transform(corpus).toarray()

from sklearn.feature_extraction.text import TfidfTransformer
tfidftransformer=TfidfTransformer()
X=tfidftransformer.fit_transform(X).toarray()

from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer=TfidfVectorizer(max_features=3000,min_df=2,max_df=0.9,stop_words=stopwords.words('english'))
X=vectorizer.fit_transform(corpus).toarray()
```

Figure 6.1: Text preprocessing

Research question 1:

Which classification algorithms can be used in predicting the movie name?
Research question 1 is answered using the research method literature review, specified in section 4.1. The results from the Systematic Literature Review process are presented in section 5.1 with a table of works and identifications that are related to our thesis field like classification algorithms, movie predictions, and Natural Language Processing. Table 5.1 illustrates the findings from the literature study which consists of the applications of classification algorithms in prediction, applications of NLP in preprocessing, and several movie recommendation systems, movie-related predictions. From the analysis of the results of the literature review, we found four algorithms Support Vector Machine, Random Forest classifier, Logistic Regression, and Naive Bayes classifier algorithms can be used for our thesis and used these algorithms in the experiment to obtain movie name predictions.

Research question 2:

How accurate the model trained using NLP in predicting the movie name?
Research question 2 is answered using the experiment method, specified in section 4.2. It was performed in the sequence of different steps starting from collection of the data, preprocessing of the data, loading of the data, training the model with different classification algorithms identified from the Systematic Literature Review, and testing with the predictions of random samples of text. The results of the experiment are specified in section 5.2, with the accuracy of different models, and the model with high prediction accuracy is selected. From the analysis of experiment results, the model trained using Naive Bayes classifier and Support Vector Machine algorithm are selected as the best models with the prediction accuracy of 59.63%.

In the case of classification algorithms, there are different algorithms used in the prediction cases in the past, and from the study of literature, we have found four algorithms that may be used in our thesis model. After the implementation, it can be stated that most of the selected algorithms have given moderate accuracy in prediction with both Support Vector Machine and Naive Bayes models are best in prediction when compared to remaining models, followed by Random Forest model as the least accuracy giving model.

From the results of the Random Forest model predictions, we have observed that the Random Forest algorithm is not performing well in the case of the movie script classification, as it is predicting the same movie name phone booth for almost all the random text inputs. These results of Random Forest predictions can occur due to the minute differences in the probability of each text that makes the classification complex, and it is classifying all those texts in the same class of phone booth.

Use of other algorithms apart from what we used in our thesis, may give more accurate or less accurate results but from the knowledge of our literature review, we had choose those algorithms, and in the future, we would like to train the model with more algorithms and compare them with the models we developed in this thesis and select the model with best prediction accuracy.

From the implementation, we have trained the models with 11 movie scripts for the present and we have got the expected results in most of the cases. The use of the dataset with more movie scripts requires much more preprocessing with natural language techniques as it may have lots of text that has the same meaning that makes it difficult for the machine to make correct predictions. Another important thing is due to the lack of time, we have not tested the model with a large set of test samples. Testing with more random text inputs may vary in accuracy and could make the model much effective in prediction. We hope that in the future, this case will get sorted.

Chapter 7

Conclusions and Future Work

In this thesis, we have implemented a model that predicts the movie name when the random scenario from the movie script is given as the input. We have used a Systematic Literature Review to identify the suitable classification algorithms for the prediction of movie names using movie scripts. As a result, we choose algorithms such as Logistic Regression, Naive Bayes classifier, Support Vector Machine, and Random Forest classifier for model implementation as there is no particular proof for choosing one algorithm. To obtain the accuracy of prediction, we have trained the model with the same set of data with movie scripts and with classification algorithms like Support Vector Machine, Random Forest classifier, Naive Bayes classifier, and Logistic Regression that are identified from the literature study. All the models are tested with random paraphrased sentences from the script and the results are noted. The accuracy of each model is calculated to compare with the remaining models. We can conclude that the model trained using Naive Bayes classifier and Support Vector Machine algorithms has shown good performance with an accuracy of 59.63% in the prediction of movie names, when compared to the remaining models.

For future work, we can use ensemble methods to obtain more accurate and use a large dataset of movie scripts and multi-classification algorithms can be used to predict all the movie names that have the same scenario in their movie scripts, when a random scenario is given as the input. In addition to that, we can use more advanced NLP techniques for text preprocessing that could help in efficient classification.

Bibliography

[1] "The Internet Movie Script Database (IMSDb)." [Online]. Available: https://imsdb.com/

[2] R. Ahuja, A. Solanki, and A. Nayyar, "Movie recommender system using K-Means clustering and K-Nearest Neighbor," in *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. IEEE, 2019, pp. 263–268.

[3] T. O. Ayodele, "Types of machine learning algorithms," *New advances in machine learning*, vol. 3, pp. 19–48, 2010, publisher: InTech.

[4] J. Bhaskar, K. Sruthi, and P. Nedungadi, "Hybrid approach for emotion classification of audio conversation based on text and speech mining," *Procedia Computer Science*, vol. 46, pp. 635–643, 2015, publisher: Elsevier.

[5] A. Blackstock and M. Spitz, "Classifying Movie Scripts by Genre with a MEMM Using NLP-Based Features," *Citeseer*, 2008, publisher: Citeseer.

[6] W. Bristi, Z. Zaman, and N. Sultana, "Predicting IMDb Rating of Movies by Machine Learning Techniques," Jul. 2019, pp. 1–5.

[7] M. J. Cafarella and O. Etzioni, "A search engine for natural language applications," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 442–452.

[8] R. Dale, S. Geldof, and J.-P. Prost, "Using natural language generation in automatic route description," *Journal of Research and practice in Information Technology*, vol. 37, no. 1, pp. 89–105, 2005.

[9] R. Dhir and A. Raj, "Movie Success Prediction using Machine Learning Algorithms and their Comparison," in *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*. IEEE, 2018, pp. 385–390.

[10] Z. Ghahramani, "Unsupervised learning," in *Summer School on Machine Learning*. Springer, 2003, pp. 72–112.

[11] M. Govindarajan, "Sentiment analysis of movie reviews using hybrid method of naive bayes and genetic algorithm," *International Journal of Advanced Computer Research*, vol. 3, no. 4, p. 139, 2013, publisher: Citeseer.

[12] Q. Hoang, "Predicting Movie Genres Based on Plot Summaries," *arXiv:1801.04813 [cs, stat]*, Jan. 2018, arXiv: 1801.04813. [Online]. Available: http://arxiv.org/abs/1801.04813

[13] Y.-F. Huang and S.-H. Wang, "Movie genre classification using svm with audio

and video features," in *International Conference on Active Media Technology.* Springer, 2012, pp. 1–10.

[14] R. Iqbal, M. A. A. Murad, A. Mustapha, P. H. S. Panahy, and N. Khanahmadliravi, "An experimental study of classification algorithms for crime prediction," *Indian Journal of Science and Technology*, vol. 6, no. 3, pp. 4219–4225, 2013, publisher: Indian Society for Education and Environment, 23(new) Neelkamal Apt, 3 d . . . .

[15] V. Jain, "Prediction of movie success using sentiment analysis of tweets," *The International Journal of Soft Computing and Software Engineering*, vol. 3, no. 3, pp. 308–313, 2013.

[16] D. Kabakchieva, "Student performance prediction by using data mining classification algorithms," *International journal of computer science and management research*, vol. 1, no. 4, pp. 686–690, 2012.

[17] M. H. Latif and H. Afzal, "Prediction of movies popularity using machine learning techniques," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 8, p. 127, 2016, publisher: International Journal of Computer Science and Network Security.

[18] E. G. Learned-Miller, "Introduction to supervised learning," *I: Department of Computer Science, University of Massachusetts*, 2014.

[19] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine learning proceedings 1994.* Elsevier, 1994, pp. 157–163.

[20] J. Liu, S. Cyphers, P. Pasupat, I. McGraw, and J. Glass, "A conversational movie search system based on conditional random fields," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[21] L. Michelbacher, "Multi-word tokenization for natural language processing," 2013.

[22] R. Mihalcea, H. Liu, and H. Lieberman, "Nlp (natural language processing) for nlp (natural language programming)," in *International Conference on Intelligent Text Processing and Computational Linguistics.* Springer, 2006, pp. 319–330.

[23] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, "Sentiment analysis and classification based on textual reviews," in *2013 international conference on Information communication and embedded systems (ICICES).* IEEE, 2013, pp. 271–276.

[24] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, no. 5, pp. 544–551, 2011, publisher: Oxford University Press.

[25] P. Navaney, G. Dubey, and A. Rana, "SMS spam filtering using supervised machine learning algorithms," in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence).* IEEE, 2018, pp. 43–48.

[26] S. Nusinovici, Y. C. Tham, M. Y. C. Yan, D. S. W. Ting, J. Li, C. Sabanayagam, T. Y. Wong, and C.-Y. Cheng, "Logistic regression was as good as machine learning for predicting major chronic diseases," *Journal of clinical epidemiology*,

vol. 122, pp. 56–69, 2020.

[27] C. Rădulescu, M. Dinsoreanu, and R. Potolea, "Identification of spam comments using natural language processing techniques," in *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*. IEEE, 2014, pp. 29–35.

[28] M. Ranzato, F. J. Huang, Y.-L. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition," in *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 2007, pp. 1–8.

[29] D. K. Renuka, T. Hamsapriya, M. R. Chakkaravarthi, and P. L. Surya, "Spam classification based on supervised learning using machine learning techniques," in *2011 International Conference on Process Automation, Control and Computing*. IEEE, 2011, pp. 1–7.

[30] A. Romanov, K. Lomotin, and E. Kozlova, "Application of Natural Language Processing Algorithms to the Task of Automatic Classification of Russian Scientific Texts," *Data Science Journal*, vol. 18, no. 1, 2019, publisher: Ubiquity Press.

[31] A. Rupawala, D. Pujara, M. Shikalgar, and E. Ukey, "Movie Genre Prediction from Plot Summaries by Comparing Various Classification Algorithms," 2020.

[32] A. Sadovsky and X. Chen, "Song genre and artist classification via supervised learning from lyrics," *Previous CS224N Final Project*, 2006.

[33] M. Santini, "Automatic identification of genre in web pages," Ph.D. dissertation, University of Brighton, 2007.

[34] D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," *Procedia computer science*, vol. 132, pp. 1578–1585, 2018, publisher: Elsevier.

[35] S. Thrun and M. L. Littman, "Reinforcement learning: an introduction," *AI Magazine*, vol. 21, no. 1, pp. 103–103, 2000, publisher: American Association for Artificial Intelligence.

[36] S. Vijayarani and S. Dhayanand, "Data mining classification algorithms for kidney disease prediction," *Int J Cybernetics Inform*, vol. 4, no. 4, pp. 13–25, 2015.

[37] Y. Wang, M. Wang, and W. Xu, "A sentiment-enhanced hybrid recommender system for movie recommendation: a big data analytics framework," *Wireless Communications and Mobile Computing*, vol. 2018, 2018, publisher: Hindawi.

[38] A. Wilcox and G. Hripcsak, "Classification algorithms applied to narrative reports." in *Proceedings of the AMIA Symposium*. American Medical Informatics Association, 1999, p. 455.

[39] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 2014, pp. 1–10.

[40] J. Yi, T. Nasukawa, R. Bunescu, and W. Niblack, "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing

techniques," in *Third IEEE international conference on data mining.* IEEE, 2003, pp. 427–434.

[41] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE transactions on fuzzy systems*, vol. 26, no. 2, pp. 794–804, 2017, publisher: IEEE.

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden