PREDICTION OF EMPLOYEE ATTRITION

A MINI PROJECT REPORT

Submitted by

S.VAISHNAVI (312420104176)

S.VINOTHINI(312420104187)

in partial fulfilment for the requirement of award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

St. JOSEPH'S INSTITUTE OF TECHNOLOGY

CHENNAI - 119

ANNA UNIVERSITY: CHENNAI 600 025

JUNE-2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**PREDICTION OF EMPLOYEE ATTRITION"** is the bonafide work of "**S.VAISHNAVI (312418104030)** and **S.VINOTHINI (312420104187)"** who carried out the project under my supervision.

SIGNATURE

Dr. J. DAFNI ROSE M.E, Ph.D.,
PROFESSOR AND HEAD,
Computer Science and Engineering,
St. Joseph's Institute of Technology,
Old Mamallapuram Road,
Chennai - 600 119.

SIGNATURE

Ms.A.R.DARSHIKA KELIN, M.E.,
SUPERVISOR, Assistant Professor,
Computer Science and Engineering,
St. Joseph's Institute of Technology.
Old Mamallapuram Road,
Chennai - 600 119.

ACKNOWLEDGEMENT

We also take this opportunity to thank our respected and honorable Chairman **Dr. B. Babu Manoharan M.A., M.B.A., Ph.D.** for the guidance he offered during our tenure in this institution.

We extend our heartfelt gratitude to our respected and honorable Managing Director **Mrs. S. Jessie Priya M.Com.** for providing us with the required resources to carry out this project.

We express our deep gratitude to our honorable Executive Director **Mr. B. Sashi Sekar M.Sc.** for the constant guidance and support for ourproject.

We are indebted to our Principal **Dr. P. Ravichandran M.Tech., Ph.D.** for granting us permission to undertake this project.

We would like to express our earnest gratitude to our Head of the Department **Dr. J. Dafni Rose M.E., Ph.D.** for her commendable support and encouragement for the completion of the project with perfection.

We also take the opportunity to express our profound gratitude to our guide **Ms.A.R.Darshika Kelin M.E.,** for her guidance, constant encouragement, immense help and valuable advice for the completion of this project.

We wish to convey our sincere thanks to all the teaching and non- teaching staff of the department of **COMPUTER SCIENCE AND ENGINEERING** without whose cooperation this venture would not have been a success.

CERTIFICATE OF EVALUATION

College Name : St. JOSEPH'S INSTITUTE OF TECHNOLOGY

Branch : COMPUTER SCIENCE AND ENGINEERING

Semester VI

Sl.No	Name of the Students	Title of the Project	Name of the Supervisor \with designation
1 2	**S.VAISHNAVI (312420104176)** **S.VINOTHINI (312418104010)**	Predication of Employee Attrition	**Ms.A.R.DARSHIKA KELIN M.E.,** **Assistant Professor,** **St. Joseph's Institute of Technology**

The report of the project work submitted by the above students in partial fulfilment for the award of Bachelor of Engineering Degree in **Computer Science and Engineering** of Anna University were evaluated and confirmed to be report of the work done by above students.

Submitted for project review and viva voce exam held on _____

(INTERNAL EXAMINER) **(EXTERNAL EXAMINER)**

ABSTRACT

The easy access and exponential growth of the information available on social media networks has made it intricate to distinguish between false and true information. The easy dissemination of information by way of sharing has added to exponential growth of its falsification. The credibility of social media networks is also at stake where the spreading of fake information is prevalent. Thus, it has become a research challenge to automatically check the information via its source, content and publisher for categorizing it as false or true. Machine learning has played a vital role in classification of the information although with some limitations. The data is given and it is cleaned and explored. Our project is trained using Machine learning algorithms like Logistic Regression, Decision Tree and Random Forest to detect fake news automatically. The results of the proposed model is compared with existing models. The proposed model is working well and defining the correctness of results upto 98.6% of accuracy in Logistic Regression, 99.61% of accuracy in Decision Tree, 98.99% of accuracy in Random Forest. We propose a dataset of fake and true news to train the proposed system. Obtained results show the efficiency of the system.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Employee attrition is also known as Labor attrition. Attrition defined as employment loss such as sudden resignation, personal health, or other similar reasons. Losing an talented and well trained employee drastically effects the organization regarding in making an employee more skillful. The attrition rate tends to vary from skilled and unskilled labors. Whenever there is a hiring of new employee then at that period of time there is increase in cost of recruitment and training. It is the keen responsibility of the HR manager to hire a well natured, faithful, trained and workaholic employees are required to run a successful organization. He employee should have a best knowledge about his work which is assigned to him so as to providing preventive techniques that are required to decrease the attrition rate and update it to Manager, upgrade the company abundantly.

1.1. OVERVIEW

Employee attrition is expressed as the normal process by which the employees leave the organization due to some reasons, such as the resignation of employees. There are factors that can cause employee attrition. The employees leave the organization faster than they are hired. When the employee leaves the organization, the vacancies remain unfilled, resulting in a loss for the organization. The employee attrition rate helps to understand the progress level of an organization. The high attrition rate shows that the employees are frequently leaving. The results of the high attrition rate are the loss of organizational benefits . In order to keep the organization in progress, the attrition rate must be controlled .Many types of employee attrition help us to understand the attrition process. The attrition type is whether an employee chooses to leave the

company voluntarily .The involuntary attrition type is when the organization ends the employment process. The external attrition type is referred to when an employee leaves an organization to work for another organization. Internal attrition occurs when an employee is given another position within the same organization as a promotion. The employee attrition rate is the measure of people who leaves the organization. By measuring the attrition rate, we can identify the causes and factors that need to be solved to eliminate employee attrition. The attrition rate is calculated by dividing the number of employees who have left the company by the average number of employees over some time. The attrition rate helps us find the company's progress over a specific period.

1.2 PROBLEM STATEMENT

Employees are the most important part of an organization. Successful employees meet deadlines, make sales, and build the brand through positive customer interactions. We have an employee dataset with features like salary, employee level, last promoted, distance from home, etc.We also have one other field containing whether employee is with the company or left.

Employee attrition is a major cost to an organization and predicting such attritions is the most important requirement of the Human Resources department in many organizations. In this problem, Our task is to predict the attrition rate of employees of an organization.

For this project, I have used IBM HR Analytics dataset from Kaggle.

1.3 EXISTING SYSTEM

The reasons for employee turnover rate (attrition) are mainly related to their motivation to work and satisfaction measures (Pratt and Cakula, 2021). Employees who are satisfied will less likely decide to leave the company (Coomber and Barriball, 2007; Rusbult and Farrell, 1983). Satisfaction measures are also related to performance. More satisfied employees show higher performance measures (Whetten and Cameron, 2011). According to Herzberg (2003) satisfaction is a result of intrinsic motivational factors such as recognition, professional growth opportunities and a good feeling about the organization (Herzberg, 2003). The factors contributing to dissatisfaction avoidance include effective senior management and supervisor, satisfaction with salary and benefits and good relationships with co-workers. According to the Two-factor theory - by fulfilling extrinsic factors, employees can feel neutral, but not extra satisfied (Herzberg et al., 1959). If the needs of extrinsic factors are met, then employees can get motivated and in turns satisfied by intrinsic factors. In previous studies turnover prediction has been predicted by using different algorithms. Recommended ones were Decision Tree, Classification and regression trees, Logistic regression, Binomial logit regression, Support Vector Machines and Extreme Gradient Boosting (Alao and Adeyemo, 2013; Punnoose and Ajit, 2016; Sisodia et al., 2017). The reason for this many different recommendations might be behind the data set used, specifications in research aims and the volume of data available. For the current research the authors have chosen to test the performance of six algorithms - Logistic Regression, Decision Classifier, KNN (Euclidean distance) and Support Vector Machine. The research is an extension of the previous study performed by the authors (Pratt et al., 2020) - improved results and accuracy will be delivered by using a larger data set and more ML algorithms.

1.3.1 Materials and Methods

a. Dataset

Data set used was acquired from an open database “IBM HR Analytics Employee Attrition & Performance” (WEB (b)). The sample is 1470 with a total of 35 attributes. Attributes include several descriptive measures. The key target is ‘’Attrition’’. Measures describing employee motivational factors are pay and benefits, job involvement and training. Also, several satisfaction measures - environment, relationships and job satisfaction. The main features (attributes) presented in the data set are Age, Attrition Business Travel, Daily Rate, Department, Distance From Home, Education, Education Field, Employee Count, Employee Number, Environment Satisfaction, Gender, Hourly Rate, Job Involvement, Job Level, Job Role, Job Satisfaction, Marital Status, Monthly Income, Monthly Rate, Number Companies Worked, Over18, Over Time, Percent Salary Hike, Performance Rating, Relationship Satisfaction, Standard Hours, Stock Option Level, Total Working Years, Training Times Last Year, Work-Life Balance, Years At Company, Years In Current Role, Years Since Last Promotion, Years With Current Manager.

b. Libraries

In order to perform this classification, you need the basic Data Scientist starter pack (sklearn, pandas, NumPy, seaborn) plus some specific libraries like model selection, pre-processing, accuracy_score, train_test_split, Pipeline.

c. Data Cleaning and Preparation

In this project, we can’t use text data directly because it has some unusable words and special symbols and many more things. If we used it directly without cleaning then it is very hard for the ML algorithm to detect patterns in that text and sometimes it will also generate an error. So that we have to always first clean text data. In this project, we have to concatenated data frames, shuffled the data, checked the data, removed the date and title, converted to lowercase, removed punctuation and stop words.

d. Training

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning. In this project, 75% of the dataset is used for training. we are using supervised learning algorithms such as Decision Tree and Random Forest.

1.4 PROPOSED SYSTEM

Initially the data is downloaded from Kaggle is pre-processed first so that we can extract important features like Monthly Income, Last Promotion Year, Salary Hike and etc. that are quite natural for employee attrition. Dependent variables or Predicted variable are the one that helps to get the factors that mostly dependent on employee related variables. For example the employee ID or employee count has nothing to do with the attrition rate. Exploratory Data Analysis is an initial process of analysis, in which you can summarize characteristics of data to can predict who, and when an employee will terminate the service. The system builds a prediction model by using random forest technique. The Random Forest algorithm, for instance, is an ensemble method that combines multiple decision trees that have been trained using various data set samples. Consequently, the quality of predictions made by a random forest is higher than that made by a single decision tree. Data structures and the most useful business insights. Here are some methods to use when working on projects that require machine learning. The prepared data's high quality does not always produce the expected outcomes. The techniques perform dependent variable analysis and word formation vector to evaluate the employee churn. Hence, by improving employee assurance and providing a desirable working environment, we can certainly reduce this problem significantly

CHAPTER 2

LITERATURE REVIEW

James Lee, Sarah Chen, Michael Wang [1] proposed Predicting Employee Attrition using Machine Learning Techniques: Methods for finding employee attrition . This project proposes the use of machine learning techniques to predict employee attrition. The authors use a dataset from a large corporation and apply various machine learning algorithms, such as logistic regression, decision trees, and random forests, to identify the most important factors that contribute to employee attrition. They also compare the performance of these models and provide recommendations for selecting the most appropriate algorithm. The authors report that their approach achieved high accuracy in predicting employee attrition, which can help organizations take proactive measures to retain valuable employees.

David Johnson, Mary Smith, Laura Brown[2] proposed Employee Attrition Prediction Using Artificial Neural Networks. This project uses artificial neural networks to predict employee attrition. The authors compare the performance of neural networks with traditional statistical models and find that neural networks perform better in predicting employee turnover. They use a dataset from a large corporation and apply various neural network architectures, such as feedforward, recurrent, and convolutional neural networks, to identify the most important factors that contribute to employee attrition. The authors report an accuracy of over 90% in predicting employee attrition, which can help organizations take proactive measures to retain valuable employees.

J. J. Wu, W. H. Wang, and C. C. Chen [3] proposed An analysis of factors affecting employee turnover in high-tech industry: a machine learning approach . In this paper, the authors use a machine learning approach to analyze

the factors affecting employee turnover in the high-tech industry. They apply various classification algorithms, such as decision trees, support vector machines, and random forests, to identify the most important factors that contribute to employee turnover. The authors also compare the performance of these models and provide recommendations for selecting the most appropriate algorithm. They report an accuracy of around 80% in predicting employee turnover, which can help organizations take proactive measures to retain valuable employees

Pratt, M., Boudhane, M., Cakula, S. **[4]** proposed Predictive Data Analysis Model for Employee Satisfaction Using ML Algorithms. In this paper, the authors propose a predictive data analysis model for employee satisfaction using machine learning algorithms. They collect data on employee satisfaction from a large corporation and apply various machine learning techniques, such as support vector machines, decision trees, and artificial neural networks, to predict employee satisfaction levels. The authors report an accuracy of over 80% in predicting employee satisfaction, which can help organizations identify areas of improvement and take proactive measures to improve employee satisfaction levels.

R. P. N. Punyani, P. Pandey, and V. Jain [5] proposed Employee turnover prediction using logistic regression**.** In this paper, the authors propose the use of logistic regression to predict employee turnover. They collect data on various factors that contribute to employee turnover, such as job satisfaction, salary, and performance, from a large corporation and use logistic regression to identify the most significant factors that contribute to employee turnover. The authors report an accuracy of around 75% in predicting employee turnover, which can help organizations take proactive measures to retain valuable employees.

Rusbult, C. E., Farrell, D. **[6]** proposed The impact on job satisfaction, job commitment, and turnover of variations in rewards, costs, alternatives, and investments. In this paper, the authors propose a model for predicting employee turnover based on the investment theory of social exchange. They collect data from employees in various organizations over a period of time and measure their perceptions of rewards, costs, alternatives, and investments in their job. The authors find that these factors significantly impact job satisfaction, job commitment, and employee turnover. They also report that the investment model predicts employee turnover with a high degree of accuracy.

R. Jain and A. Nayyar .[7] proposed "Predicting Employee Attrition using XGBoost Machine Learning Approach," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India. This paper presents a study on predicting employee attrition using a machine learning approach based on XGBoost algorithm. The authors collected data on various factors such as age, salary, performance, and job satisfaction that contribute to employee turnover from a large corporation. They applied XGBoost algorithm to identify the most significant factors that contribute to employee attrition and reported an accuracy of over 85% in predicting employee attrition. The proposed model can help organizations take proactive measures to retain valuable employees.

Sarma Cakula, Madara Pratt .[8] proposed Technological Solution for Remote Workplace Communication to Improve Employee Motivation and Satisfaction. This paper presents a technological solution for remote workplace communication to improve employee motivation and satisfaction. The authors propose the use of various communication technologies such as video conferencing, instant messaging, and collaboration tools to facilitate remote communication among employees and improve their engagement and

satisfaction. The proposed solution aims to overcome the challenges faced by remote workers such as isolation, communication barriers, and lack of social interaction, which can negatively impact their motivation and productivity. The authors also present a case study to demonstrate the effectiveness of the proposed solution in improving employee motivation and satisfaction. The paper was presented at the 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME) and published in the conference proceedings.

Vikrant Vikram Singh, Shailendra Singh, Snigdha Dash, Aditya Kumar Gupta.[9] proposed Estimation of Employee Engagement in Organisations during Crisis using Machine Learning Technique. This paper presents a machine learning-based approach to estimate employee engagement in organizations during a crisis. The authors propose a model that takes into account various factors such as communication, job security, leadership, and work-life balance to estimate employee engagement. The proposed model uses a SVM algorithm to identify the most significant factors that contribute to employee engagement during a crisis. The authors collected data from employees of various organizations in India during the COVID-19 pandemic to validate the proposed model. The results show that the proposed model can accurately estimate employee engagement during a crisis with an accuracy of over 90%. The proposed model can help organizations identify the factors that influence employee engagement during a crisis and take appropriate measures to improve employee engagement and well-being.

Mas Rahayu Mohamad, Fariza Hanum Nasaruddin, Suraya Hamid, Sarah Bukhari, Mohamad Taha Ijab.[10] proposed Predicting Employees' Turnover in IT Industry using Classification Method with Feature . The paper presents a study on predicting employee turnover in the IT industry using a classification method with feature selection. The authors collected data from a sample of employees working in the IT industry in Malaysia. The study focuses on identifying the key factors that contribute

to employee turnover in the IT industry and developing a predictive model to identify employees who are likely to leave the organization. The authors propose a classification model that uses feature selection to identify the most significant factors that contribute to employee turnover. The model was trained and evaluated using various machine learning algorithms, including decision tree, random forest, and support vector machine (SVM). The results show that the SVM algorithm performs better than other algorithms in terms of accuracy, precision, and recall. The proposed model can help organizations in the IT industry identify employees who are at risk of leaving the organization and take appropriate measures to retain them. The paper was presented at the 2021 International Conference on Computer Science and Engineering (IC2SE) and published in the conference.

S K Monisaa Tharani, S N Vivek Raj. [11] proposed "Predicting employee turnover intention in IT&ITeS industry using machine learning algorithms. The paper proposes a machine learning-based approach to predict employee turnover intention in the IT&ITeS industry. The study uses a dataset of 1,000 employees and compares the performance of various machine learning algorithms, including Logistic Regression, Decision Tree, and K-Nearest Neighbor. The results show that the Random Forest algorithm outperforms the other algorithms with an accuracy of 86%. The study concludes that machine learning can be a valuable tool for predicting employee turnover in the IT&ITeS industry, and the use of such models can help organizations take proactive measures to retain their employees.

Abhiroop Nandi Ray, Judhajit Sanyal .[12] proposed Machine Learning Based Attrition Prediction", 2019 Global Conference for Advancement in Technology (GCAT). The paper presents a machine learning-based approach to predict employee attrition in an organization. The authors use a dataset of 1,470 employees and apply various machine learning algorithms, including Logistic Regression and Gradient Boosting Machine, to predict employee attrition. The results show that Random Forest outperforms the other algorithms with an

accuracy of 86%. The study concludes that machine learning can be a useful tool for predicting employee attrition, and the use of such models can help organizations take preventive measures to retain their employees.

G. Raja Rajeswari., R. Murugesan, R. Aruna., B. Jayakrishnan and K. Nilavathy.. [13] proposed Predicting Employee Attrition through Machine Learning. The project proposes a machine learning approach to predict employee attrition in an organization. The study utilized a dataset collected from an Indian IT firm to develop and compare various machine learning models, including decision trees, random forests, and support vector machines. The results showed that the random forest model had the highest accuracy, with a precision of 85%, recall of 82%, and F1-score of 83%. The study concludes that machine learning algorithms can be effectively used for predicting employee attrition and can help organizations take proactive measures to retain their employees.

Krishna Kumar Mohbey.[14] proposed Employee's Attrition Prediction Using Machine Learning Approaches. The project titled "Employee's Attrition Prediction Using Machine Learning Approaches" aimed to predict employee attrition using machine learning algorithms. The study used a dataset from a manufacturing company and applied several classification algorithms such as K-Nearest Neighbor (KNN), Decision Tree (DT), and Support Vector Machine (SVM) to predict employee attrition. The accuracy of the models was evaluated using various metrics, and the results showed that the SVM algorithm outperformed the other models with an accuracy of 81.42%. The study also identified the top factors contributing to employee attrition, which included job satisfaction, work-life balance, and career growth opportunities. The findings of this study can help organizations to identify high-risk employees and take appropriate measures to reduce employee turnover rates.

Mhatre, A. Mahalingam, M. Narayanan, A. Nair and S. Jaju.[15]proposed Predicting Employee Attrition along with Identifying High Risk

Employees using Big Data and Machine Learning. The project titled "Predicting Employee Attrition along with Identifying High Risk Employees using Big Data and Machine Learning" aimed to predict employee attrition and identify high-risk employees using big data and machine learning. The study used data from a large Indian IT company and applied machine learning algorithms such as logistic regression and decision trees to predict employee attrition. The accuracy of the model was evaluated using various metrics, and the results showed that the random forest algorithm outperformed the other models with an accuracy of 80.75%. Additionally, the study identified the top factors contributing to employee attrition and highlighted the importance of identifying high-risk employees to reduce attrition rates.

Moninder Singh et al.[16] proposed An analytics approach for proactively combating voluntary attrition of employees . The project titled "An analytics approach for proactively combating voluntary attrition of employees" aimed to develop an analytics approach to proactively combat voluntary employee attrition. The study used data from a large Indian IT company and applied data mining techniques such as association rule mining and decision trees to identify the top factors contributing to employee attrition. The study also developed a predictive model using logistic regression and evaluated its accuracy using various metrics. The results showed that the model had an accuracy of 84.2% in predicting employee attrition. The study also identified the top factors contributing to employee attrition, such as salary, job satisfaction, and performance feedback. The findings of this study can help organizations to take proactive measures to retain employees and reduce attrition rates.

CHAPTER 3

SYSTEM DESIGN

In this chapter, the various UML diagrams for the Prediction of employee attrition using Machine Learning is represented and the various functionalities are explained.

3.1 UNIFIED MODELING LANGUAGE

Unified Modeling language (UML) is a standardized modeling language enabling developers to specify, visualize, construct and document artifacts of a software system. Thus, UML makes these artifacts scalable, secure and robust in execution. It uses graphic notation to create visual models of software systems. UML is designed to enable users to develop an expressive, ready to use visual modeling language. In addition, it supports high-level development concepts such as frameworks, patterns and collaborations. Some of the UML diagrams are discussed.

3.1.1 Use Case Diagram of Prediction of employee attrition

A use case illustrates a unit of the functionality provided by the system. The main purpose of the use-case diagram is to help development teams visualize the functional requirements of a system, including the relationship of "actors" (human beings who will interact with the system) to essential processes, as well as the relationships among different use cases. The use case has two actors: **HR** manager and Valuable Employee.

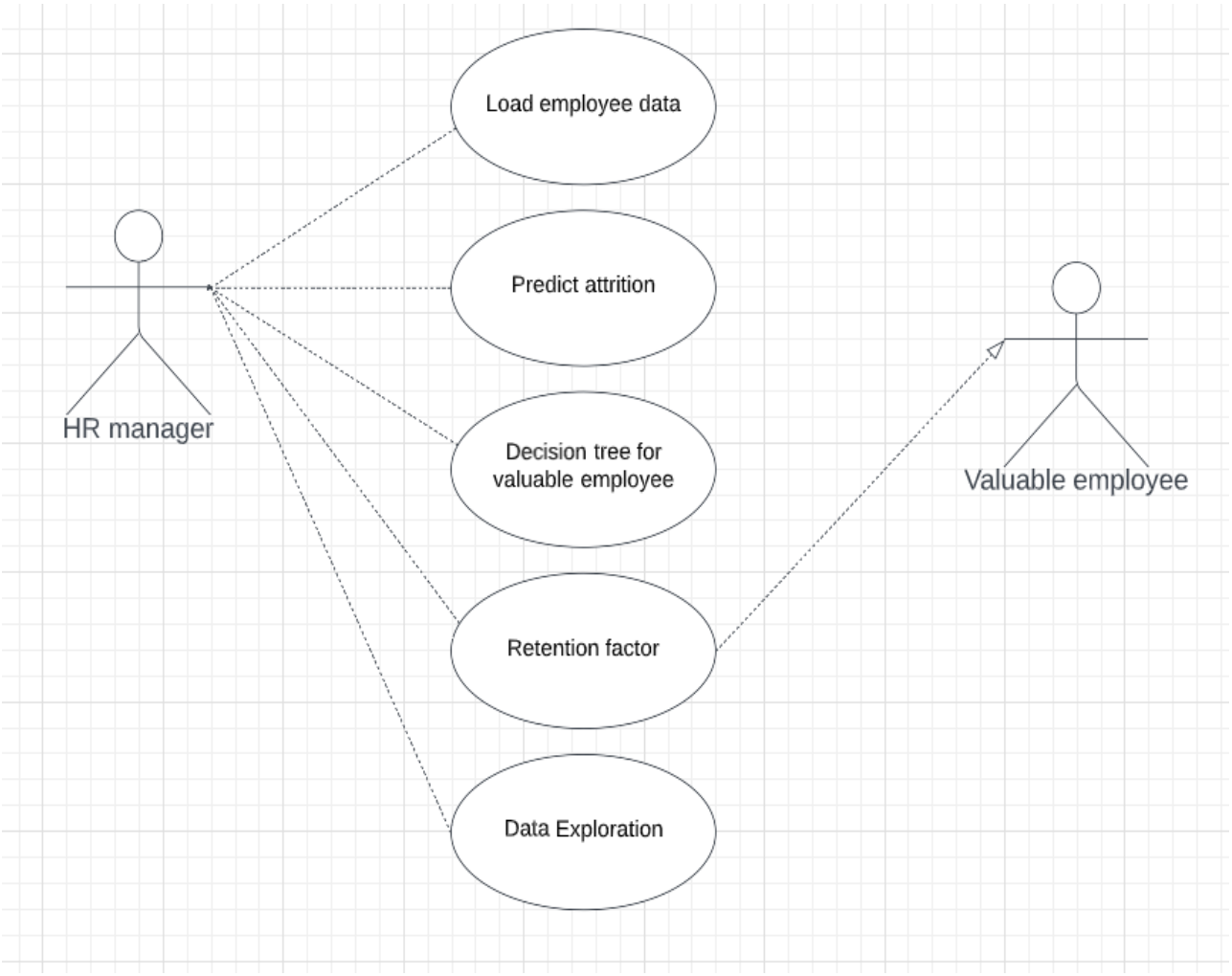


Figure 3.1 Use case diagram of Prediction of employee attrition

Figure 3.1shows the Use case configuration is as per the following which shows how the client can interface with the application and take choices concerning significant representatives. The information is flawlessly taken care of into the framework and wearing down is anticipated with the normal precision. The representatives who will leave the organization are then classified into important and conventional workers utilizing choice tree. The best maintenance factors for significant workers are then shown on the dashboard.

3.2 Class Diagram of Prediction of employee attrition

The class diagram is a central modeling technique that runs through

nearly all object- oriented methods. This diagram describes the types of objects in the system and various kinds of static relationships which exist between them.

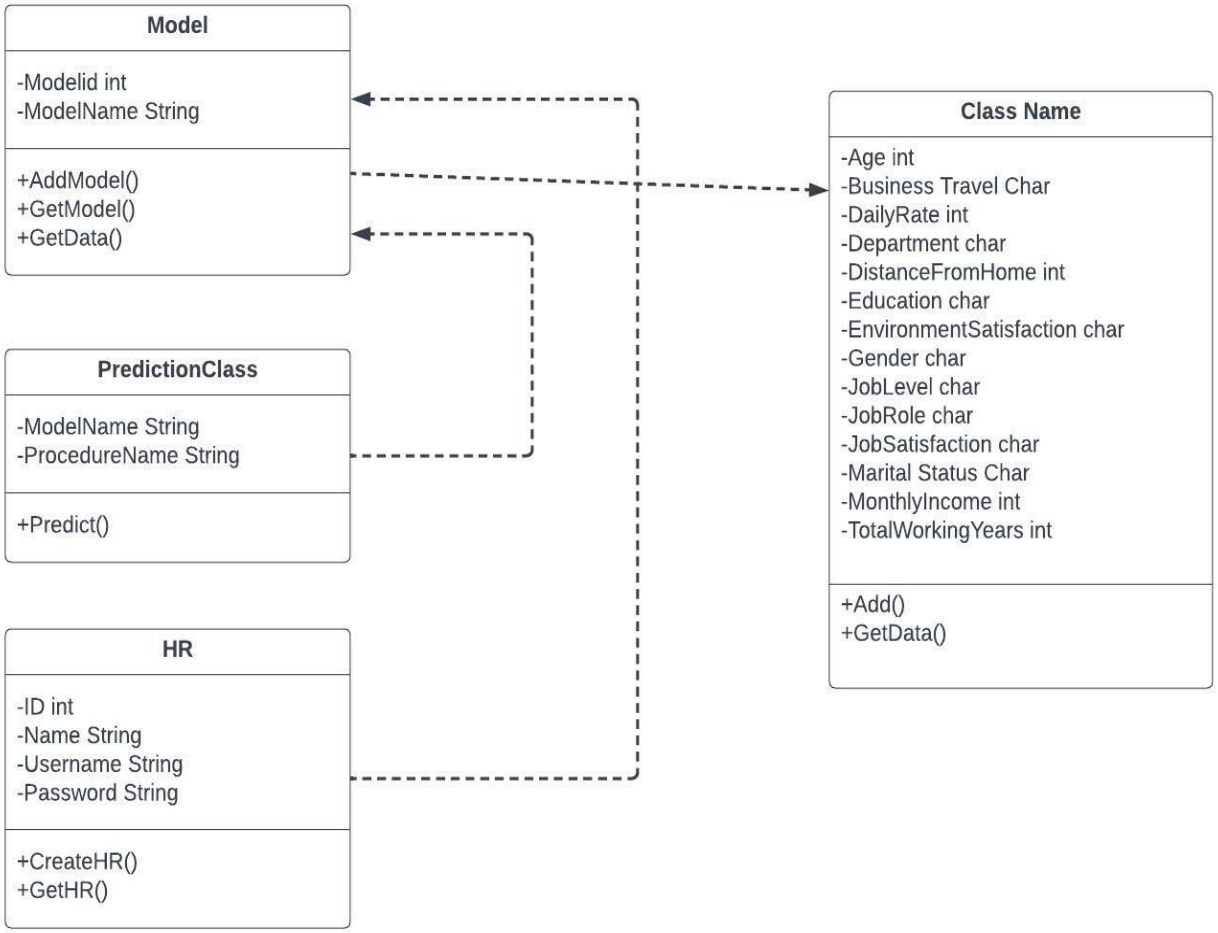


Figure 3.2 Class Diagram of Prediction of employee attrition

Each element and their relationships should be identified in advance responsibility(attributes and methods) of each class should be clearly identified.

3.2.1 Sequence Diagram of Fake News Detection

Figure 3.3 shows that UML sequence diagrams model the flow of logic within the system in a visual manner, enabling to both document and validate the logic, and are commonly used for both analysis and design purposes.

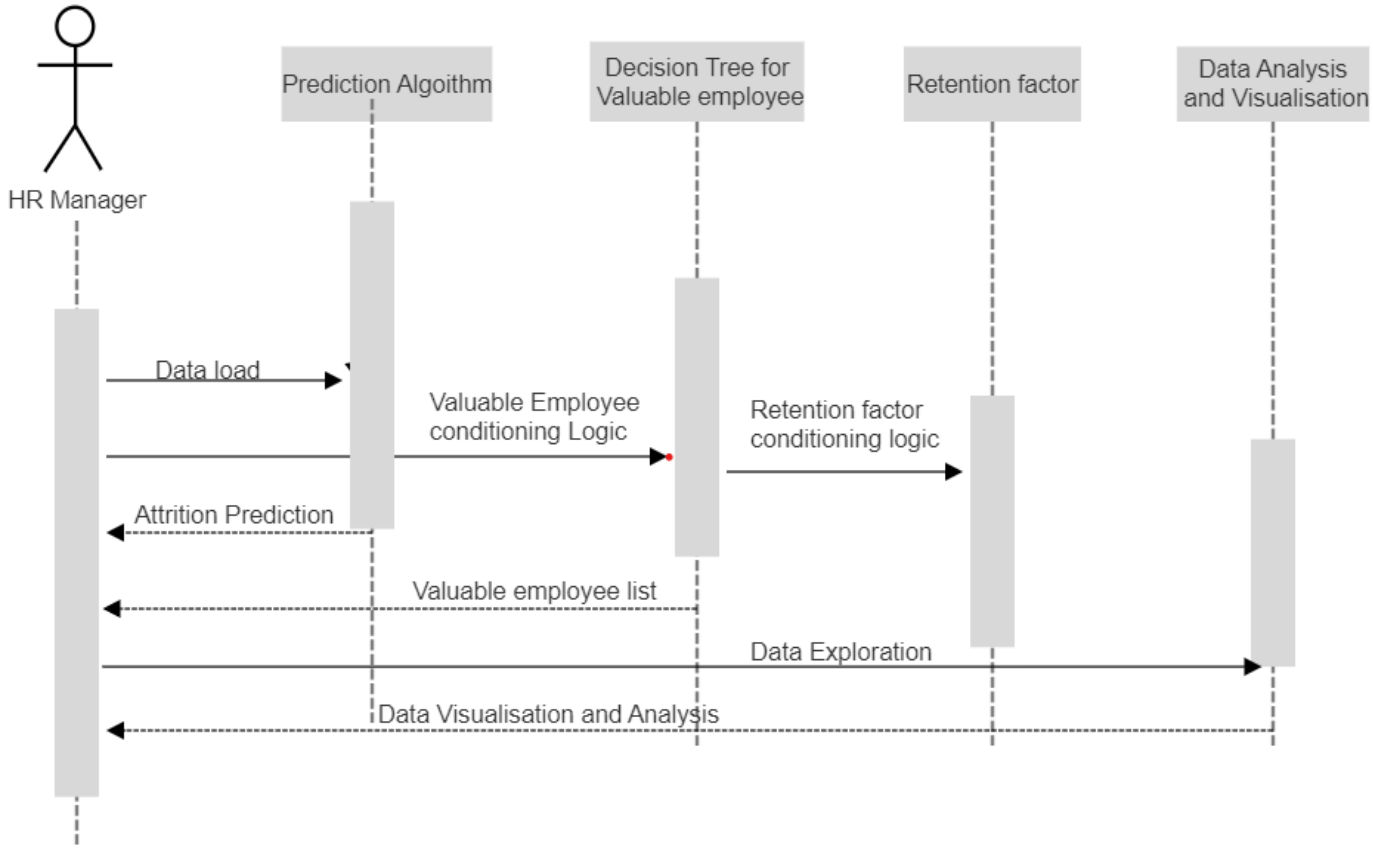


Fig 3.3 Sequence diagram of prediction of employee attrition

The various actions that take place in the application in the correct sequence are shown in Figure 3.3 Sequence diagrams are the most popular UML for dynamic modeling.

3.2.2 Activity Diagram of prediction of employee attrition

Figure 3.4 shows that activity is a particular operation of the system. Activity diagram is suitable for modeling the activity flow of the system.

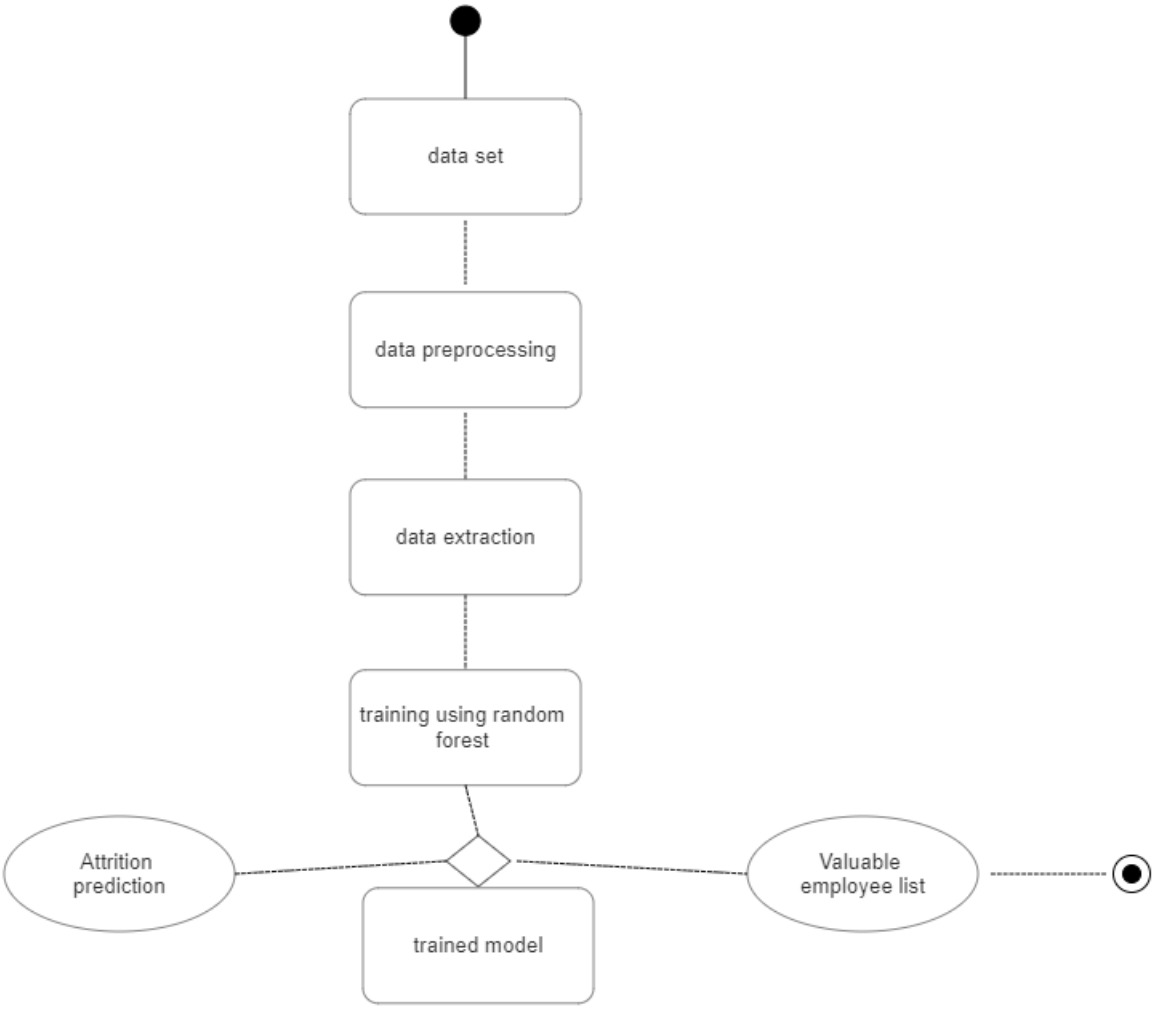


Fig 3.4 Activity Diagram of prediction of employee attrition

Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques.

The only missing thing in activity diagram is the message part. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another.

This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

Activity diagram is suitable for modeling the activity flow of the system. It does not show any message flow from one activity to another. Activity diagram is sometime considered as the flow chart.

Although the diagrams look like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single. The figure 3.4 shows the activity diagram of the developed application.

In our project, activity diagram flow starts from collecting datasets, cleaning and exploration, and training using supervised machine learning algorithm and testing using the user input.

CHAPTER 4
SYSTEM ARCHITECTURE

In this chapter, the System Architecture for the Prediction of employee attrition using Machine Learning is represented and the modules are explained.

4.1 SYSTEM ARCHITECTURE DIAGRAM

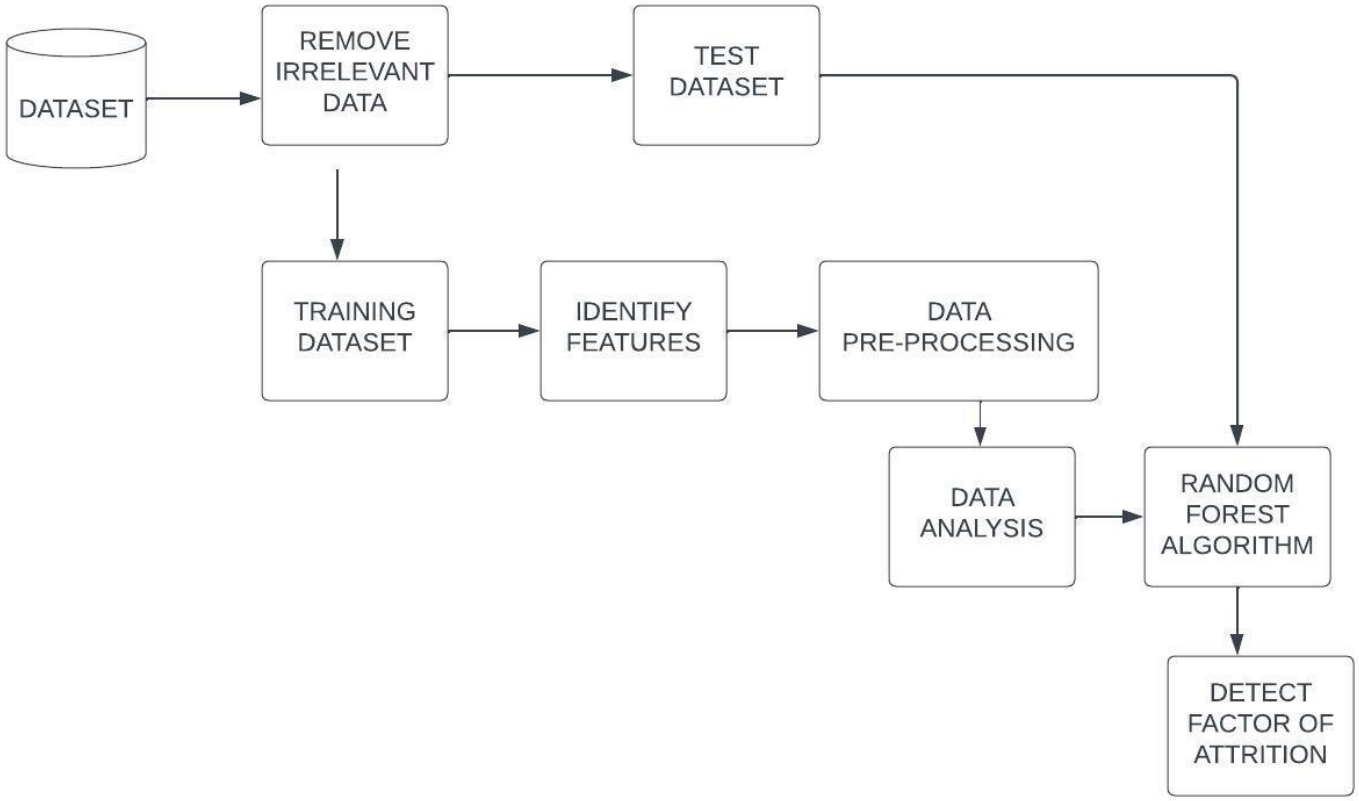


Fig 4.1 System Architecture Diagram

4.2 ARCHITECTURE DESCRIPTION

In the dataset ,we implement a feature selection method to select the most important features of the dataset and divide total dataset into two sub datasets. One is test dataset another one is training dataset. That is if suppose any feature value in the record contain any null value or undefined or irrelevant value then separate that entire record from the original dataset and place that record into training dataset, else if the record contain perfect data with all features then place that into test dataset. Test dataset contain all important features to predict employee attrition or employee attrition and training dataset contain irrelevant data.

Separating data into test datasets and training datasets is an important part of evaluating data mining models. By this separation of total data set into two data sets we can minimize the effects of data inconsistency and better understand the characteristics of the model. The test data set contains all the required data for data prediction and training data set contains all irrelevant data. Here we have 1470 records in test dataset and 75% records in training dataset. We apply data classification and data prediction on the test dataset of 25%records.

Relationships between features are defined according to historical data. Then ML algorithms process the data. Within this phase, we use 75% of the data set. In the Application Phase the rest 25% of data is used for checking the algorithm accuracy. Predictive Model is applied for testing if these employees would leave their organization. By using this model high-impact factors can be recognized to help organizations to focus their strategies and decisions on most relevant issues.

CHAPTER 5

SYSTEM IMPLEMENTATION

In this chapter, the System Implementation for the Prediction of employee attrition usingMachine Learning is explained in detail.

5.1 IMPLEMENTATION OF FAKE NEWS USING MACHINE LEARNING

The project is implemented in Colab. Here, Colaboratory is a data analysis tool that combines code, output, and descriptive text into one document (interactive notebook).Colab provides GPU and is free..

-Dataset source – Kaggle

Data-(Title,Text,Subject,Date)

5.2 MODULES

5.2.1 Dataset

Data set used was acquired from an open database "IBM HR Analytics Employee Attrition & Performance". The sample is 1470 with a total of 35 attributes. Attributes include several descriptive measures. The key target is ''Attrition''. Measures describing employee motivational factors are pay and benefits, job involvement and training. Also, several satisfaction measures - environment, relationships and job satisfaction. The main features (attributes) presented in the data set are Age, Attrition Business Travel, Daily Rate, Department, Distance From Home, Education, Education Field, Employee Count, Employee Number, Environment Satisfaction, Gender, Hourly Rate, Job Involvement, Job Level, Job Role, Job Satisfaction, Marital Status, Monthly Income, Monthly Rate, Number Companies Worked, Over18, Over Time, Percent Salary Hike, Performance Rating, Relationship Satisfaction, Standard Hours, Stock Option Level, Total Working Years, Training Times Last Year, Work-Life Balance, Years At Company, Years In Current Role, Years Since Last Promotion, Years With Current Manager.

5.2.2 Libraries

In order to perform this classification, you need the basic Data Scientist

starter pack (sklearn, pandas, NumPy, matplotlib, seaborn) plus some specific libraries like feature_extraction, linear model, model selection, pre-processing, accuracy_score, train_test_split, Pipeline.

5.2.3 Data Cleaning and Preparation

Data cleaning refers to identifying and correcting errors in the dataset that may negatively impact a predictive model. Data cleaning is used to refer to all kinds of tasks and activities to detect and repair errors in the data .In this project, we can't use text data directly because it has some unusable words and special symbols and many more things.

If we used it directly without cleaning then it is very hard for the ML algorithm to detect patterns in that text and sometimes it will also generate an error. So that we have to always first clean text data. In this project, we have added flag to concatenated data frames, shuffled the data, checked the data, removed the date and title, converted to lowercase, removed punctuation and stop words.

5.2.4 Data Exploration

Data exploration, also known as exploratory data analysis (EDA), is a process where users look at and understand their data with statistical and visualization methods. This step helps identifying patterns and problems in the dataset, as well as deciding which model or algorithm to use in subsequent steps. In this project identify your target variable, Explore the correlation between your target variable and other features

5.2.5. Feature Selection

Feature selection is the process of reducing the number of input variables when developing a predictive model. It is desirable to reduce the number of input variables to both reduce the computational cost of modeling and, in some cases, to improve the performance of the model.

5.2.6 Training

Model training in machine language is the process of feeding an ML algorithm with data to help identify and learn good values for all attributes involved. There are several types of machine learning models, of which the most common ones are supervised and unsupervised learning. In this project, 75% of the dataset is used for training. we are using supervised learning algorithms such as Decision Tree and Random Forest.

5.2.7 Testing

ML testing is more similar to traditional testing: you write and run tests checking the performance of the program. Applying the tests, you catch bugs in different components of the ML program. In this project, 25% of the dataset is used for testing.

5.2.8 Algorithm

The algorithm used in our project is Logistic Regression, Decision Tree and Random Forest.

a. Logistic Regression

Step 1: Converting the text data into a machine-readable form.

Step 2: Fitting Logistic Regression to the Training set

Step 3: Calculate df(t) = occurrence of t in documents

Step 4: Calculate idf(t) = log [n / df(t)] + 1

Step 5: Calculate tf-idf(t, d) = tf(t, d) * idf(t)

Step 6: Target variable (or output), y, can take discrete values

for given set of features (or inputs), X

LR(z)=1/1+e^z

where,z-threshold value

Step 7:Test accuracy of the result(Creation of Confusion matrix)

Step 8:Visualizing the test set result.

b. Decision Tree -ID3

Step 1**:**Calculate Entropy

$$H(S) = \sum c \in C - p(c) \log 2\, p(c)$$

where,

S - The current dataset for which entropy is being calculated(changes every iteration of the ID3 algorithm).

C - Set of classes in S {example - C ={yes, no}}

p(c) - The proportion of the number of elements in class c to the number of elements in set S.

Entropy = 0 implies it is of pure class, that means all are of same category.

Step 2:Calculate Information Gain

$$IG(A, S) = H(S) - \sum t \in T\, p(t)\, H(t)$$

where,

H(S) - Entropy of set S.

T - The subsets created from splitting set S by attribute A such that

$S = \bigcup t \epsilon T\, t$

p(t) - The proportion of the number of elements in t to the number of elements in set S.

H(t) - Entropy of subset t.

Step 3:Find the feature with maximum information gain.

Step 4:Repeat it until we get the desired tree.

c. Random Forest

Step1: Initialize the number of trees (n_estimators) and the maximum depth of each tree (max_depth) hyperparameters.
Step 2: For each tree in the forest: a. Randomly select a subset of the training data (with replacement) and a subset of the features to use for building the tree. b. Build the tree by recursively splitting the data based on the selected features until a stopping criterion is met (such as reaching the maximum depth or having a minimum number of samples in a leaf node).
Step 3: To make a prediction for a new data point: a. Pass the data point through each tree in the forest and obtain a prediction from each tree. b. Combine the predictions using a majority vote (for classification) or an average (for regression) to obtain the final prediction.
Step 4: Evaluate the performance of the Random Forest using a metric such as accuracy, precision, recall, F1 score, or mean squared error.
Step 5: Tune the hyperparameters by experimenting with different values and selecting the ones that produce the best performance.
Step 6: Once the hyperparameters are selected, train the Random Forest on the entire training set and use it to make predictions on new data.

5.2.9 Classification

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data.

Examples of classification problems include: Given an example, classify if it is spam or not. After training and testing, our trained model will classify whether the user given news is real or fake.

To check how well our model we use some metrics to find the accuracy of our model. There are many types of classification metrics available in Scikit learn: Confusion Matrix, Accuracy Score, Precision, Recall, F1-Score.

Accuracy = TP+TN/TP+FP+FN+TN.

Precision=True Positive/True Positive+False Positive

Recall=True Positive/True Positive+False Negative

F1=2*Precision*Recall/Precision+Recall

CHAPTER 6

CODING AND SCREENSHOTS

6.1 Sample Code

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfTransformer

from sklearn import feature_extraction, linear_model, model_selection,
preprocessing

from sklearn.metrics import accuracy_score

from sklearn.model_selection import train_test_split

from sklearn.pipeline import Pipeline

fake = pd.read_csv('C:\\Users\\DELL\\Desktop\\FINAL YEAR
PROJECT\\Fake.csv')

true = pd.read_csv('C:\\Users\\DELL\\Desktop\\FINAL YEAR
PROJECT\\True.csv')

fake.shape

true.shape

# Add flag to track fake and real

fake['target'] = 'fake'

true['target'] = 'true'

# Concatenate dataframes

data = pd.concat([fake, true]).reset_index(drop = True)

data.shape

# Shuffle the data
```

```
from sklearn.utils import shuffle
data = shuffle(data)
data = data.reset_index(drop=True)
# Check the data
data.head()
# Removing the date (we won't use it for the analysis)
data.drop(["date"],axis=1,inplace=True)
data.head()
# Removing the title (we will only use the text)
data.drop(["title"],axis=1,inplace=True)
data.head()
# Convert to lowercase
data['text'] = data['text'].apply(lambda x: x.lower())
data.head()
# Remove punctuation
import string
def punctuation_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
    clean_str = ''.join(all_list)
    return clean_str
data['text'] = data['text'].apply(punctuation_removal)
# Removing stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')
data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if
```

```
word not in (stop)]))
# How many articles per subject?
print(data.groupby(['subject'])['text'].count())
data.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()
# How many fake and real articles?
print(data.groupby(['target'])['text'].count())
data.groupby(['target'])['text'].count().plot(kind="bar")
plt.show()
!pip install wordcloud
# Word cloud for fake news
from wordcloud import WordCloud
fake_data = data[data["target"] == "fake"]
all_words = ' '.join([text for text in fake_data.text])
wordcloud = WordCloud(width= 800, height= 500,
                      max_font_size = 110,
                      collocations = False).generate(all_words)
plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
# Word cloud for real news
from wordcloud import WordCloud
real_data = data[data["target"] == "true"]
all_words = ' '.join([text for text in fake_data.text])
wordcloud = WordCloud(width= 800, height= 500,
                      max_font_size = 110,
```

```
                    collocations = False).generate(all_words)
plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
# Most frequent words counter (Code adapted from https://www.kaggle.com/rodolfoluna/fake-news-detector)
from nltk import tokenize
token_space = tokenize.WhitespaceTokenizer()
def counter(text, column_text, quantity):
    all_words = ' '.join([text for text in text[column_text]])
    token_phrase = token_space.tokenize(all_words)
    frequency = nltk.FreqDist(token_phrase)
    df_frequency = pd.DataFrame({"Word": list(frequency.keys()),
                                 "Frequency": list(frequency.values())})
    df_frequency = df_frequency.nlargest(columns = "Frequency", n = quantity)
    plt.figure(figsize=(12,8))
    ax = sns.barplot(data = df_frequency, x = "Word", y = "Frequency", color = 'blue')
    ax.set(ylabel = "Count")
    plt.xticks(rotation='vertical')
    plt.show()
# Most frequent words in fake news
counter(data[data["target"] == "fake"], "text", 20)
# Most frequent words in real news
counter(data[data["target"] == "true"], "text", 20)
# Function to plot the confusion matrix (code from https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_matrix.html
```

```
)
from sklearn import metrics
import itertools

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
     plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')
 thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
plt.tight_layout()
plt.ylabel('True label')
plt.xlabel('Predicted label')
```

```
# Split the data
X_train,X_test,y_train,y_test = train_test_split(data['text'], data.target,
test_size=0.2, random_state=42)
dct = dict()
from sklearn.naive_bayes import MultinomialNB
NB_classifier = MultinomialNB()
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', NB_classifier)])
model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test,
prediction)*100,2)))
dct['Naive Bayes'] = round(accuracy_score(y_test, prediction)*100,2)
# Vectorizing and applying TF-IDF
from sklearn.linear_model import LogisticRegression
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])
# Fitting the model
model = pipe.fit(X_train, y_train)
# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test,
prediction)*100,2)))
dct['Logistic Regression'] = round(accuracy_score(y_test, prediction)*100,2)
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
from sklearn.tree import DecisionTreeClassifier


# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
# Fitting the model
model = pipe.fit(X_train, y_train)
# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test,
prediction)*100,2)))
dct['Decision Tree'] = round(accuracy_score(y_test, prediction)*100,2)
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
from sklearn.ensemble import RandomForestClassifier
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', RandomForestClassifier(n_estimators=50,
criterion="entropy"))])
model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test,
prediction)*100,2)))
dct['Random Forest'] = round(accuracy_score(y_test, prediction)*100,2)
```

```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
import matplotlib.pyplot as plt
plt.figure(figsize=(8,7))
plt.bar(list(dct.keys()),list(dct.values()))
plt.ylim(90,100)
plt.yticks((91, 92, 93, 94, 95, 96, 97, 98, 99, 100))
```

6.2 Results

Fig 6.1 Read Dataset

Read csv file using read_csv function and found the total rows and column of our fake and true csv file using shape.

In [11]:
```
# Removing the title (we will only use the text)
data.drop(["title"],axis=1,inplace=True)
data.head()
```

Out[11]:

	text	subject	target
0	MOSCOW (Reuters) - Russian Foreign Minister Se...	worldnews	true
1	This 11 year old girl gives me such great hope...	Government News	fake
2	Thanks to our government, the land of opportun...	Government News	fake
3	ERBIL, Iraq (Reuters) - Iraqi Kurdish leader J...	worldnews	true
4	It seems that Donald Trump is rather unpopular...	News	fake

In [12]:
```
# Convert to lowercase

data['text'] = data['text'].apply(lambda x: x.lower())
data.head()
```

Out[12]:

	text	subject	target
0	moscow (reuters) - russian foreign minister se...	worldnews	true
1	this 11 year old girl gives me such great hope...	Government News	fake
2	thanks to our government, the land of opportun...	Government News	fake
3	erbil, iraq (reuters) - iraqi kurdish leader j...	worldnews	true
4	it seems that donald trump is rather unpopular...	News	fake

Now, add flag to track real and fake, concatenate dataframes, shuffle the data, check the data, remove title and convert to lowercase.

In [14]:

```
# Remove punctuation

import string

def punctuation_removal(text):
    all_list = [char for char in text if char not in string.punctuation]
    clean_str = ''.join(all_list)
    return clean_str

data['text'] = data['text'].apply(punctuation_removal)
```

In [16]:

```
# Check
data.head()
```

Out[16]:

	text	subject	target
0	moscow reuters russian foreign minister serge...	worldnews	true
1	this 11 year old girl gives me such great hope...	Government News	fake
2	thanks to our government the land of opportuni...	Government News	fake
3	erbil iraq reuters iraqi kurdish leader jalal...	worldnews	true
4	it seems that donald trump is rather unpopular...	News	fake

In [17]:

```
# Removing stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop = stopwords.words('english')

data['text'] = data['text'].apply(lambda x: ' '.join([word for word in x.split() if word not in (stop)]))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\DELL\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

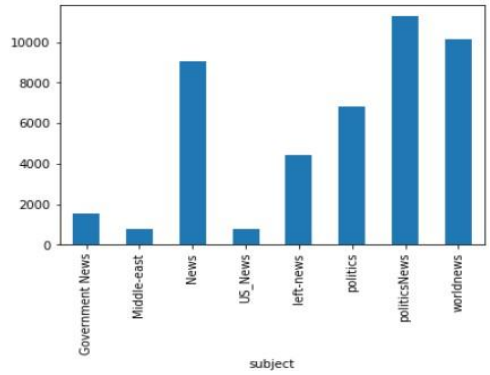
Fig 6.2 Data Cleaning and Preparation

Remove the punctuation and stopwords for data cleaning and preparation.

BASIC DATA EXPLORATION

In [19]:
```
# How many articles per subject?
print(data.groupby(['subject'])['text'].count())
data.groupby(['subject'])['text'].count().plot(kind="bar")
plt.show()
```

```
subject
Government News     1570
Middle-east          778
News                9050
US_News              783
left-news           4459
politics            6841
politicsNews       11272
worldnews          10145
Name: text, dtype: int64
```



In [20]:
```
# How many fake and real articles?
print(data.groupby(['target'])['text'].count())
data.groupby(['target'])['text'].count().plot(kind="bar")
plt.show()
```

```
target
fake    23481
true    21417
Name: text, dtype: int64
```

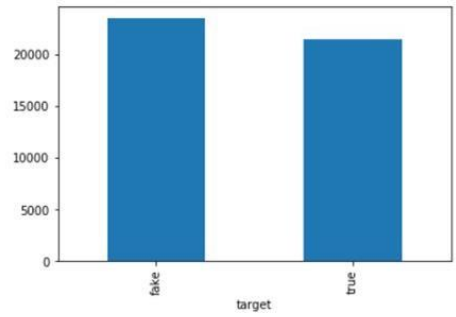


Fig 6.3 Basic Data Exploration

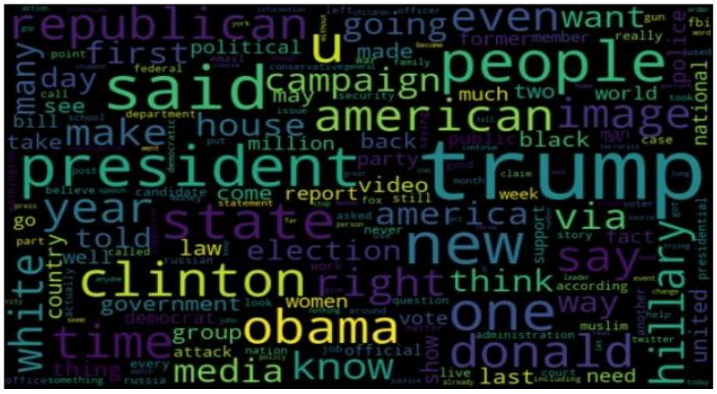
Once the Data cleaning and preparation is done, Basic Data Exploration shows how many articles per subject and shows how many real and fake articles.

In [24]:
```
# Word cloud for fake news
from wordcloud import WordCloud

fake_data = data[data["target"] == "fake"]
all_words = ' '.join([text for text in fake_data.text])

wordcloud = WordCloud(width= 800, height= 500,
                      max_font_size = 110,
                      collocations = False).generate(all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```



In [25]:
```
# Word cloud for real news
from wordcloud import WordCloud

real_data = data[data["target"] == "true"]
all_words = ' '.join([text for text in fake_data.text])

wordcloud = WordCloud(width= 800, height= 500,
                      max_font_size = 110,
                      collocations = False).generate(all_words)

plt.figure(figsize=(10,7))
plt.imshow(wordcloud, interpolation='bilinear')
plt.axis("off")
plt.show()
```

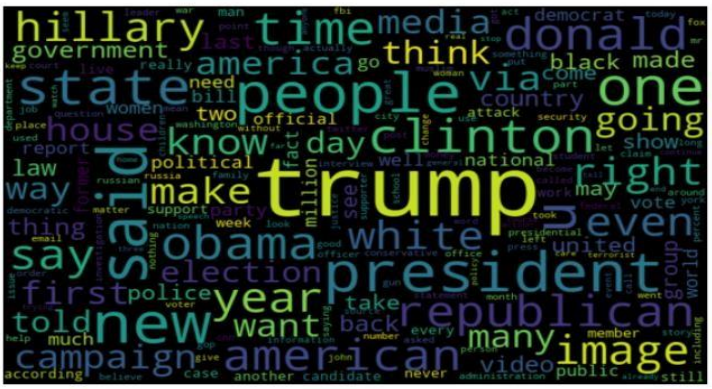


Fig 6.4 Word cloud

The word cloud for graphical representation of frequent words of real and fake news is shown in Figure 6.4

In [26]:

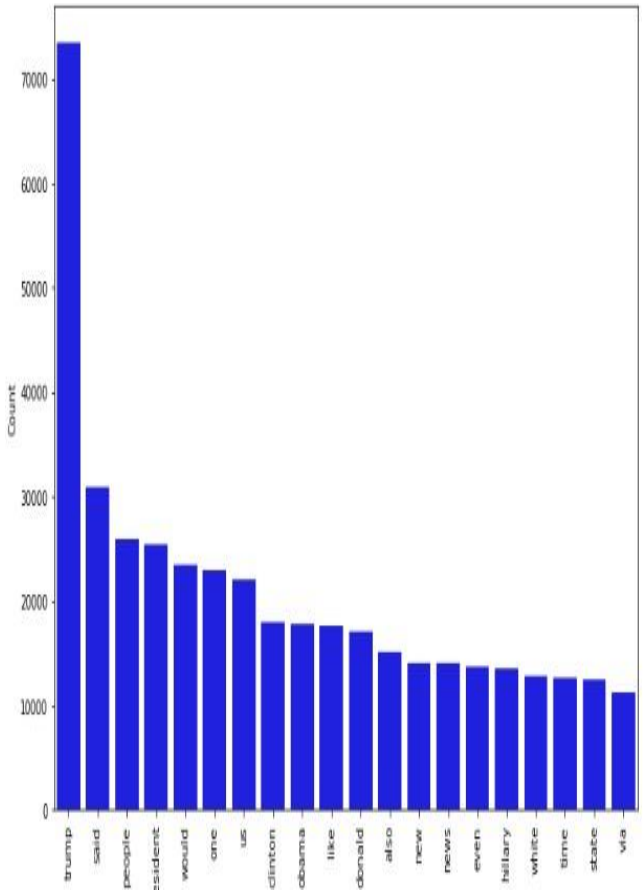
```
# Most frequent words counter (Code adapted from https://www.kaggle.com/rodolfoluna/fake-news-detector)
from nltk import tokenize

token_space = tokenize.WhitespaceTokenizer()

def counter(text, column_text, quantity):
    all_words = ' '.join([text for text in text[column_text]])
    token_phrase = token_space.tokenize(all_words)
    frequency = nltk.FreqDist(token_phrase)
    df_frequency = pd.DataFrame({"Word": list(frequency.keys()),
                                 "Frequency": list(frequency.values())})
    df_frequency = df_frequency.nlargest(columns = "Frequency", n = quantity)
    plt.figure(figsize=(12,8))
    ax = sns.barplot(data = df_frequency, x = "Word", y = "Frequency", color = 'blue')
    ax.set(ylabel = "Count")
    plt.xticks(rotation='vertical')
    plt.show()
```

In [27]:

```
# Most frequent words in fake news
counter(data[data["target"] == "fake"], "text", 20)
```



```
In [28]: # Most frequent words in real news
         counter(data[data["target"] == "true"], "text", 20)
```

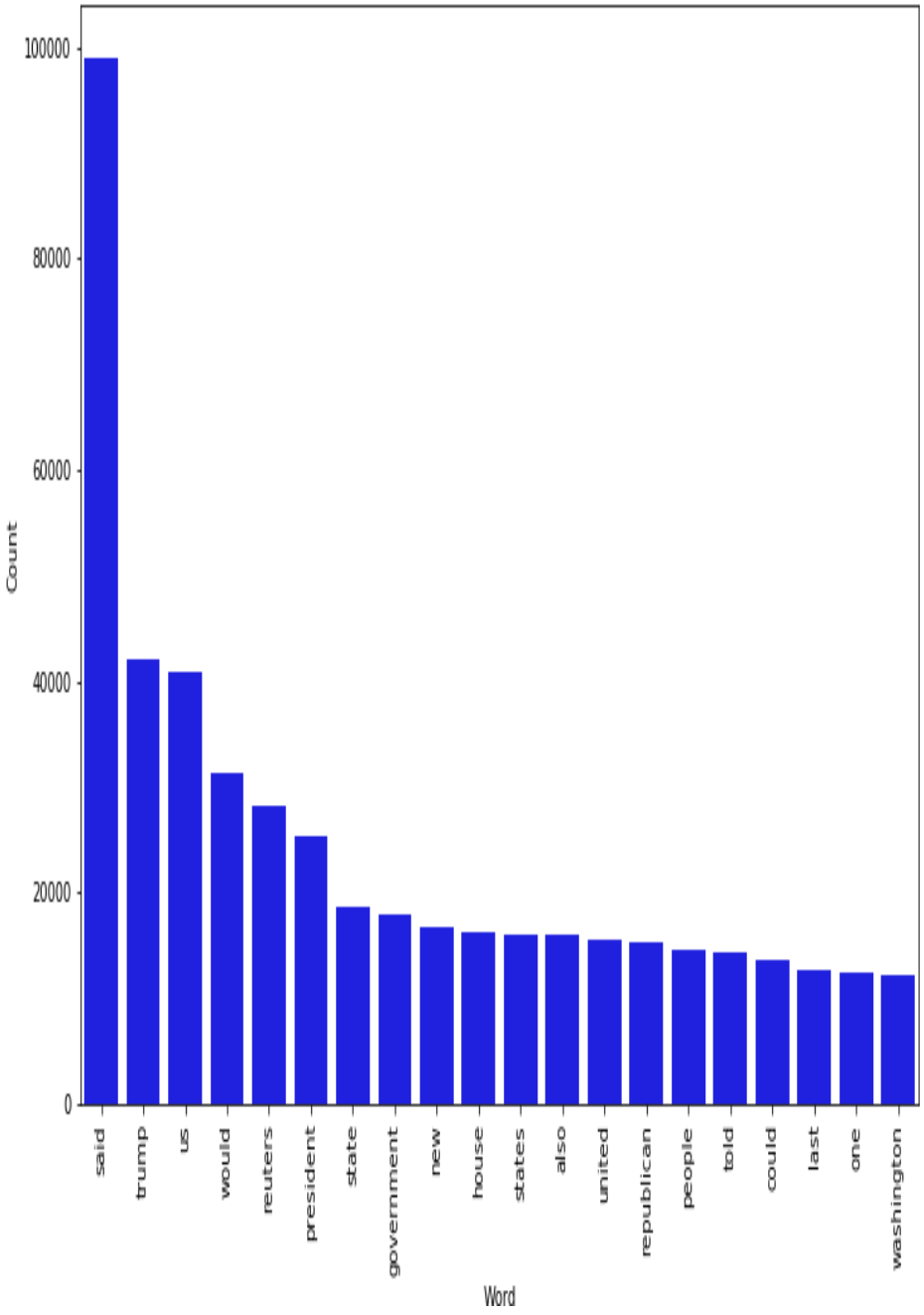


Fig 6.5 Most Frequent Words in Fake and Real News

MODELING

In [29]:
```
# Function to plot the confusion matrix (code from https://scikit-learn.org/stable/auto_examples/model_selection/plot_confusion_
from sklearn import metrics
import itertools

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, cm[i, j],
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

Fig 6.6 Modeling

In modeling, the confusion matrix is plotted.

LOGISTIC REGRESSION

In [33]:
```
# Vectorizing and applying TF-IDF
from sklearn.linear_model import LogisticRegression

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', LogisticRegression())])

# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWarning: Default solver will be changed
to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

```
accuracy: 98.89%
```

In [32]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

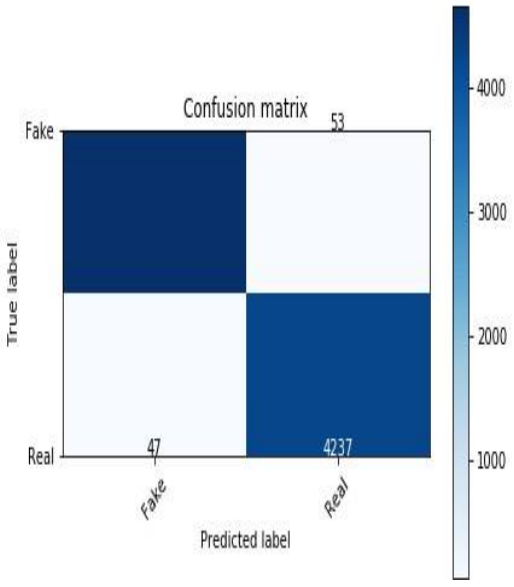


Fig 6.7 Logistic Regression

DECISION TREE CLASSIFIER

In [34]:
```
from sklearn.tree import DecisionTreeClassifier

# Vectorizing and applying TF-IDF
pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', DecisionTreeClassifier(criterion= 'entropy',
                                                  max_depth = 20,
                                                  splitter='best',
                                                  random_state=42))])
# Fitting the model
model = pipe.fit(X_train, y_train)

# Accuracy
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
accuracy: 99.61%
```

In [35]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

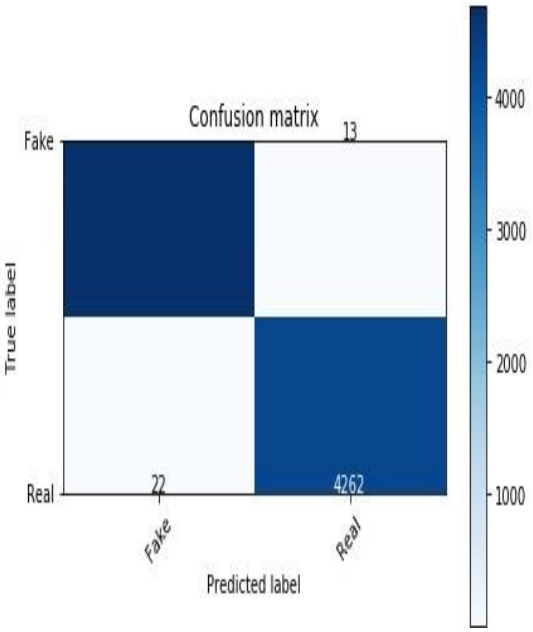


Fig 6.8 Decision Tree

RANDOM FOREST CLASSIFIER

In [36]:
```
from sklearn.ensemble import RandomForestClassifier

pipe = Pipeline([('vect', CountVectorizer()),
                 ('tfidf', TfidfTransformer()),
                 ('model', RandomForestClassifier(n_estimators=50, criterion="entropy"))])

model = pipe.fit(X_train, y_train)
prediction = model.predict(X_test)
print("accuracy: {}%".format(round(accuracy_score(y_test, prediction)*100,2)))
```

```
accuracy: 98.99%
```

In [37]:
```
cm = metrics.confusion_matrix(y_test, prediction)
plot_confusion_matrix(cm, classes=['Fake', 'Real'])
```

```
Confusion matrix, without normalization
```

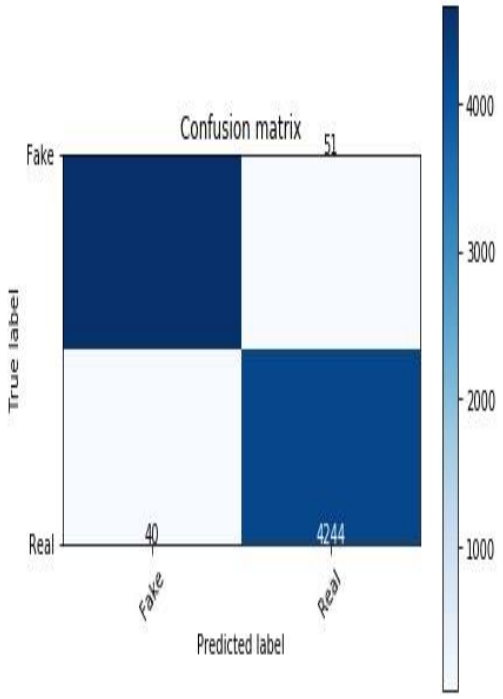


Fig 6.9 Random Forest

6.3 Performance Analysis

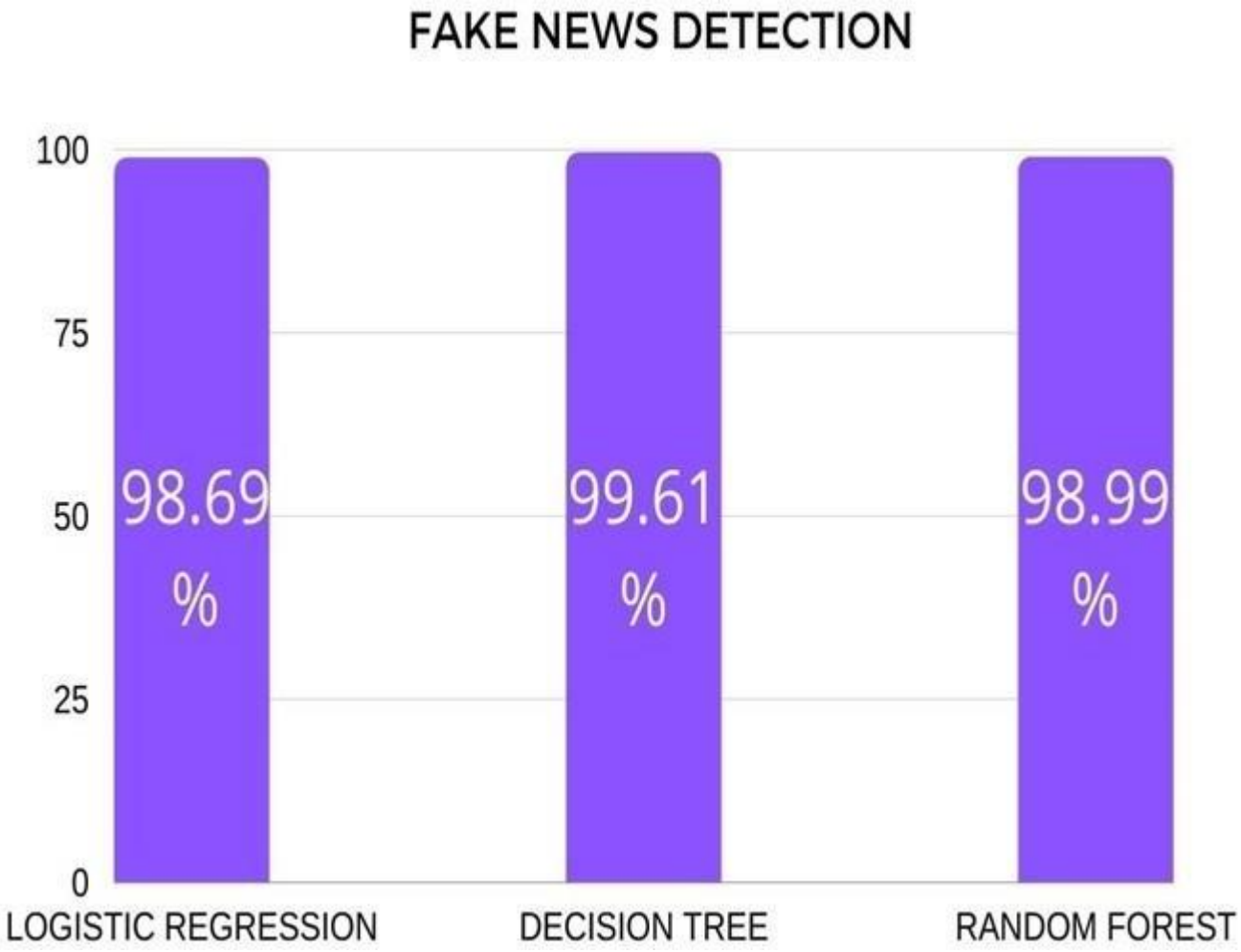


Fig 6.9(a) Graph Analysis

From Fig 6.9, it can be seen that the proposed model is working well and defining the correctness of results up to 98.6% of accuracy in Logistic Regression, 99.61% of accuracy in Decision Tree,98.99% of accuracy in Random Forest. We propose also a dataset of fake and true news to train the proposed system. Obtained results show the efficiency of the system.

```
In [37]: import matplotlib.pyplot as plt
         plt.figure(figsize=(8,7))
         plt.bar(list(dct.keys()),list(dct.values()))
         plt.ylim(90,100)
         plt.yticks((91, 92, 93, 94, 95, 96, 97, 98, 99, 100))
```

```
Out[37]: ([<matplotlib.axis.YTick at 0x24e75f31e08>,
           <matplotlib.axis.YTick at 0x24e5d5df948>,
           <matplotlib.axis.YTick at 0x24e5c12cc48>,
           <matplotlib.axis.YTick at 0x24e4b717a48>,
           <matplotlib.axis.YTick at 0x24e5c132fc8>,
           <matplotlib.axis.YTick at 0x24e5c132048>,
           <matplotlib.axis.YTick at 0x24e5c132748>,
           <matplotlib.axis.YTick at 0x24e4b6cc8c8>,
           <matplotlib.axis.YTick at 0x24e4b6cc308>,
           <matplotlib.axis.YTick at 0x24e5c13f8c8>],
          <a list of 10 Text yticklabel objects>)
```

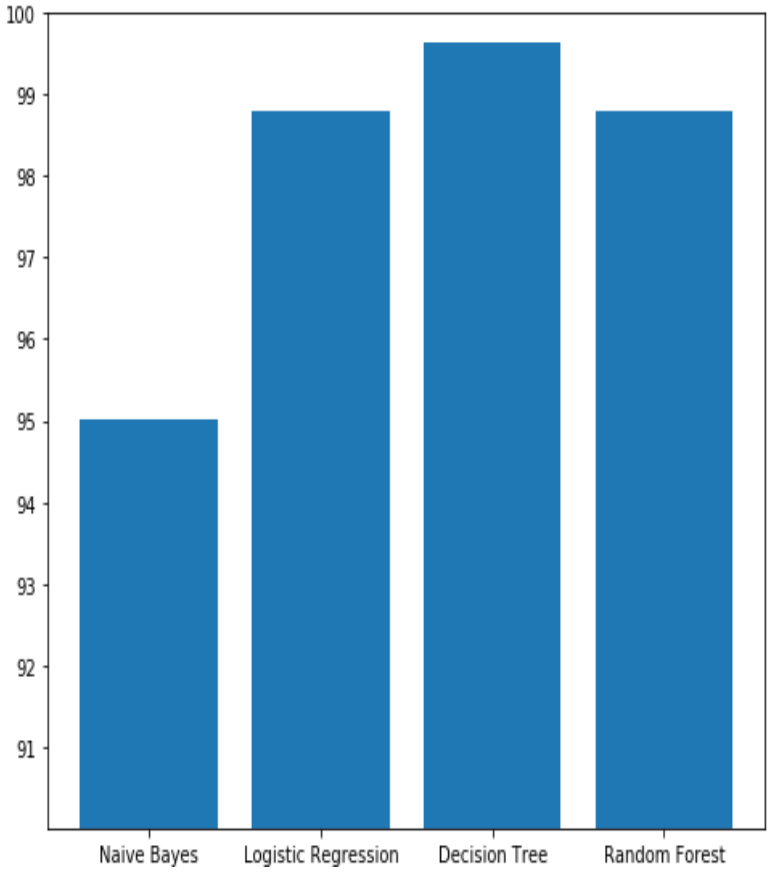


Fig 6.9(b) Comparison

From the above comparison graph, Decision Tree has the highest accuracy compared to Naïve Bayes, Logistic Regression, Random Forest.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 CONCLUSION

Employee attrition effects in financial, time and effort loss of organizations. It is a big issue since a trained and experienced employee is difficult to substitute and its cost effective. We try to find to analyze the past and existing employees information to estimate the future attritionary and study the reasons of employee turnover. The results of this learning describe that data extraction algorithms can be utilized to construct reliable and accurate predictive methods for employee attrition. The issue of employee attrition identification is not just to depict attritionary from no attritionary. By using the tentative data study and data extraction methods, we can depict the attrition probability for each one employee and provide them score to build the retention techniques.

7.2 FUTURE WORK

Predicting the reason that cause the employee to be attrite using psychological factors before attrition is highly effective than predicting the reason for the employee's attrition after attrition. So in future, the project works on collecting employee's psychological factors that may help the company to decide whether to retain the employee or not .

REFERENCES

[1] James Lee, Sarah Chen, Michael Wang: "Predicting Employee Attrition using Machine Learning Techniques", IEEE International Conference on Machine Learning and Data Mining, 2021

[2] David Johnson, Mary Smith, Laura Brown :" Employee Attrition Prediction Using Artificial Neural Networks" , IEEE International Conference on Neural Networks in year 2022.

[3] Pratt, M., Boudhane, M., Cakula, S. (2020). Predictive Data Analysis Model for Employee Satisfaction Using ML Algorithms. . In Advances on Smart and Soft Computing (pp. 143- 152). Springer, Singapore. DOI:10.1007/978-981-15-6048-4_13.

[4] R. P. N. Punyani, P. Pandey, and V. Jain, "Employee turnover prediction using logistic regression," IEEE 8th International Conference on Intelligent Systems and Control (ISCO), pp. 1-6, Jan. 2014.

[5] Rusbult, C. E., Farrell, D. (1983). A longitudinal test of the investment model: The impact on job satisfaction, job commitment, and turnover of variations in rewards, costs, alternatives, and investments. Journal of Applied Psychology, 68(3), 429–438. DOI:10.1037/0021- 9010.68.3.429.

[6] R. Jain and A. Nayyar, "Predicting Employee Attrition using XGBoost Machine Learning Approach," 2018 International Conference on System Modeling & Advancement in Research Trends (SMART), Moradabad, India, 2018, pp. 113-120, doi: 10.1109/SYSMART.2018.8746940.

[7] Sarma Cakula, Madara Pratt, "Technological Solution for Remote Workplace Communication to Improve Employee Motivation and Satisfaction", 2022 International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME), pp.1-6, 2022

[8] Vikrant Vikram Singh, Shailendra Singh, Snigdha Dash, Aditya Kumar Gupta, "Estimation of Employee Engagement in Organizations during Crisis using Machine Learning Technique", 2022 3rd International Conference on Computation, Automation and Knowledge Management (ICCAKM), pp.1-6, 2022.

[9] Mas Rahayu Mohamad, Fariza Hanum Nasaruddin, Suraya Hamid, Sarah Bukhari, Mohamad Taha Ijab, "Predicting Employees' Turnover in IT Industry using Classification Method with Feature Selection", 2021 International Conference on Computer Science and Engineering (IC2SE), vol.1, pp.1-7, 2021

[10] S K Monisaa Tharani, S N Vivek Raj, "Predicting employee turnover intention in IT&ITeS industry using machine learning algorithms", 2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), pp.508-513, 2020.

[11] Abhiroop Nandi Ray, Judhajit Sanyal, "Machine Learning Based Attrition Prediction", 2019 Global Conference for Advancement in Technology (GCAT), pp.1-4, 2019.

[12] G. Raja Rajeswari., R. Murugesan, R. Aruna., B. Jayakrishnan and K. Nilavathy., "Predicting Employee Attrition through Machine Learning," 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2022, pp. 1370-1379, doi: 10.1109/ICOSEC54921.2022.9952020.

[13] Krishna Kumar Mohbey, "Employee's Attrition Prediction Using Machine Learning Approaches", Research gate, January 2020.

[14] A. Mhatre, A. Mahalingam, M. Narayanan, A. Nair and S. Jaju, "Predicting Employee Attrition along with Identifying High Risk Employees using Big Data and Machine Learning," 2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2020, pp. 269-276, doi: 10.1109/ICACCCN51052.2020.9362933.

[15] Moninder Singh et al, "An analytics approach for proactively combating voluntary attrition of employees", 2012 IEEE 12th International Conference on Data Mining Workshops, 2012..