A PROJECT REPORT ON

"DETECTION OF FAKE JOB RECRUITMENT USING MACHINE LEARNING TECHNIQUES"

Submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS

Submitted By

PRIYA SAI DEEKSHITH

Reg No. 2251916026

Under the esteemed guidance of

Dr .G. RAMAKRISHNA, M. Tech, Ph.D

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COLLEGE OF ENGINEERING

Dr. B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM

2023-2024

Department of Computer Science & Engineering.

CERTIFICATE

This is to certify that the project entitled "DETECTION OF FAKE JOB RECRUITMENT USING MACHINE LEARNING TECHNIQUES" that is being submitted by **PRIYA SAI DEEKSHITH (2251926026)** in partial fulfilmentof requirements for the award of the degree in **MASTER OF COMPUTERAPPLICATIONS** during 2023 - 2024, in **Dr. B. R. AMBEDKAR UNIVERSITY ,SRIKAKULAM , COLLEGE OF ENGINEERING** is a record of bonafide work carried out by him under our guidance and super vision. The results embodied in this workhave not being submitted to any other university or institute for the award of any degree or diploma.

PROJECT SUPERVISOR
Dr. G.RAMAKRISHNA,M.Tech.Ph.D
Assistant Professor

HEAD OF THE DEPARTMENT
Mr. R. SRIDHAR, M. Tech
Assistant professor

External Examiner

DECLARATION

I hereby declare that the project work entitled "DETECTION OF FAKE JOB RECRUITMENT USING MACHINE LEARNING TECHNIQUES" submitted by me for the award of the degree of **MASTER OF COMPUTER APPLICATIONS (MCA) under the guidance of Dr.G.RAMAKRISHNA M.Tech,Ph.D, Assistant Professor.** Dr. B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM. Is original and it has not been submitted earlier.

Place: Srikakulam PRIYA SAI DEEKSHITH

Date: Regd.No:2251926026

ACKNOWLEDGEMENT

A Project is a golden opportunity for learning and self-development. I consider myself lucky and privileged to express my gratefulness and deep sense of gratitude and to our guide **Dr.G.RAMAKRISHNA,M.Tech,Ph.D. Assistant professor, DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,** for stimulating suggestions and encouragement that helped me at every stage of our project work, which made this project successful.

I also express my gratefulness gratitude to our **Dr.R.SRIDHAR ,M.Tech** Head of the Department of Computer Science and Engineering, **Dr.B.R. AMBEDKAR UNIVERSITY, SRIKAKULAM for** his support and encouragement throughout the project.

I also express my gratefulness gratitude to our **Prof. Ch.RAJSEKHARRAO, Ph,D., Principal, College of Engineering, Dr.B.R.AMBEDKAR UNIVERSITY, SRIKAKULAM for** his support and encouragementthroughout the project.

Further-more, I would also like to acknowledge my thankfulness with much appreciation the crucial role of our **TEACHING STAFF, NONTEACHING STAFF, PARENTS AND FRIENDS** for their love,
support, encouragement and cooperation.

Place: Srikakulam
Date:

PRIYA SAI DEEKSHITH
Regd.No: 2251926026

ABSTRACT

In our society, especially the freshers are coming out of their education level to job experience level. In such a process of finding their suitable jobs they are giving preference to some fake jobs spending time for that recruitment process. So, in order to find the fake recruitment, our project has come into existence, we are using machine learning approaches using classification techniques able to detect such fake recruitment detection processes. Different classifiers are used for checking fraudulent posts on the web and the results of those classifiers are compared for identifying the best employment scam detection model. It helps in finding the fake posts from an enormous number of posts .We can use both the single classifier and ensemble classifier .However in experimental analysis, ensemble classifiers may be showing good results over single classifiers in fake job detection

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ML: Machine Learning

CV: Classification Techniques or Cross Validation (depending on the context)

SVM: Support Vector Machine

RF: Random Forest

NB: Naive Bayes

DT: Decision Tree

NN: Neural Network

LR: Logistic Regression

KNN: K-Nearest Neighbors

AdaBoost: Adaptive Boosting

XGBoost: Extreme Gradient Boosting

GBM: Gradient Boosting Machine

MLP: Multi-Layer Perceptron

ANN: Artificial Neural Network

CHAPTER-1

1.NOTATIONS

Employment scam is one of the serious issues in recent times addressed in the domain of Online Recruitment Frauds (ORF). In recent days, many companies prefer to post their vacancies online so that these can be accessed easily and timely by the jobseekers. However, this intention may be one type of scam by the fraud people because they offer employment to job-seekers in terms of taking money from them. Fraudulent job advertisements can be posted against a reputed company for violating their credibility. These fraudulent job post detection draws good attention for obtaining an automated tool for identifying fake jobs and reporting them to people for avoiding application for such jobs. For this purpose, a machine learning approach is applied which employs several classification algorithms for recognizing fake posts. In this case, a classification tool isolates fake job posts from a larger set of job advertisements and alerts the user. To address the problem of identifying scams on job posting, supervised learning algorithms as classification techniques are considered initially. A classifier maps input variables to target classes by considering training data. Classifiers addressed in the paper for identifying fake job posts from the others are described briefly. These classifier based predictions may be broadly categorized into -Single Classifier based Prediction and Ensemble Classifiers based Prediction.

1.1 Objectives

The main objective is to detect the fake job post, which is a classic text classification problem with a straightforward proposition. It is needed to build a model that can differentiate between a “Real” job post and “Fake” job post.

1.2 Methodology

To predict job posts a large collection of the company's job posts are required. Employment Scam Aegean Dataset (EMSCAD) dataset is used to predict fake job posts. In this section the methodology followed is discussed in detail**.**

1.2.1 Dataset

Dataset collection :

Data is a set of records. This step is concerned with selecting the subset of all available data. EMSCAD dataset is used to train ML algorithms. Employment Scam Aegean Dataset (EMSCAD) dataset which is provided publicly by the University of the Aegean Laboratory of Information & Communication Systems Security. This dataset contains 17,880 real-life job postings in which 17,014 are real and 866 are fake. Dataset contains labelled data.

Attribute	Description
Job_id	Unique Job ID
Title	The title of the job ad entry
Location	Geographical location of the job ad
Department	Corporate department
Salary_range	Indicative salary range
Company_profile	A brief company description.
Description	The details description of the job ad.
Requirements	Enlisted requirements for the job opening
benefits	Enlisted offered benefits by the employer.

Fig 1.2.1.1: Dataset - Attributes with description

Attribute	Description
telecommuting	True for telecommuting positions.
has_company_logo	True if company logo is present.
has_questions	True if screening questions are present.
employment_type	Full-type, Part-time, Contract, etc.
required_experience	Executive, Entry level, Intern, etc.
required_education	Doctorate, Master's Degree, Bachelor, etc
industry	Automotive, IT, Health care, Real estate, etc.
function	Consulting, Engineering, Research, Sales etc.
fraudulent	target - Classification attribute.

Fig 1.2.1.2: Dataset - Attributes with description

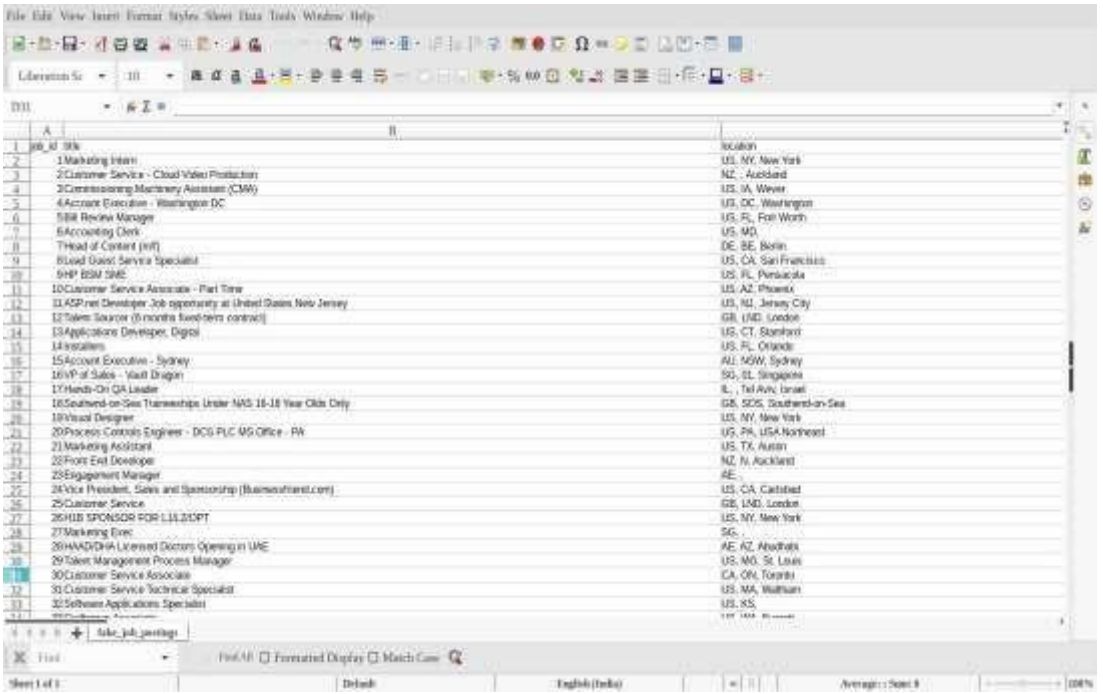


Fig 1.2.1.3: Sample dataset of job_id, title, location attributes

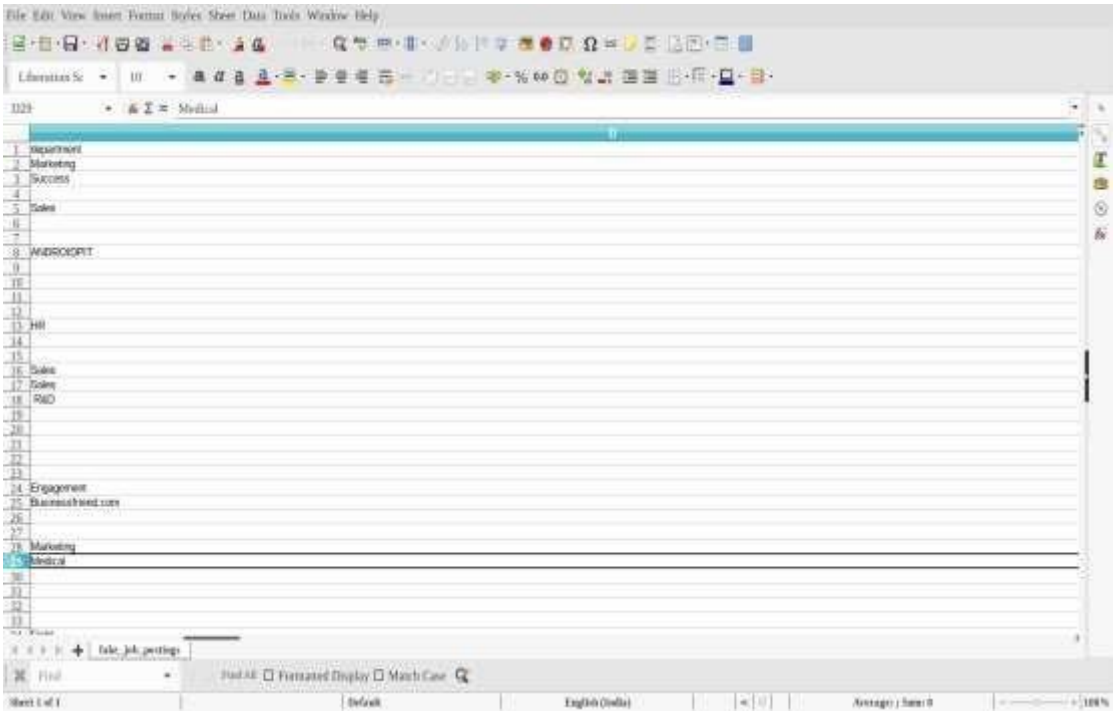


Fig 1.2.1.4: Sample dataset of department attribute

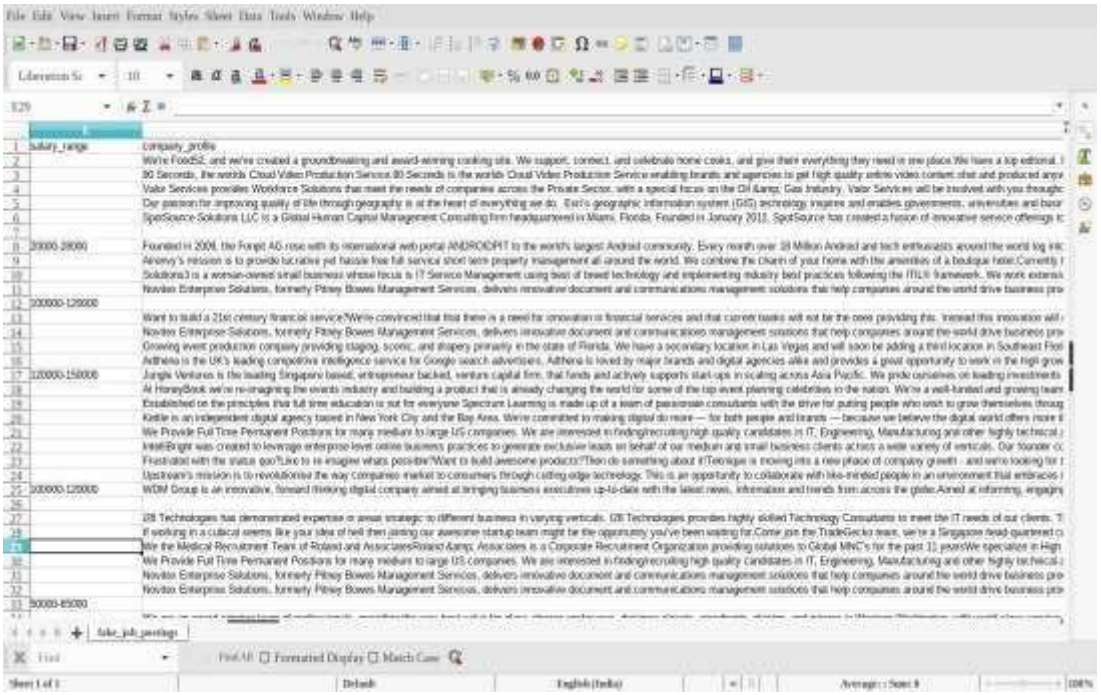


Fig 1.2.1.5: Sample dataset of salary_range and company_profile attributes

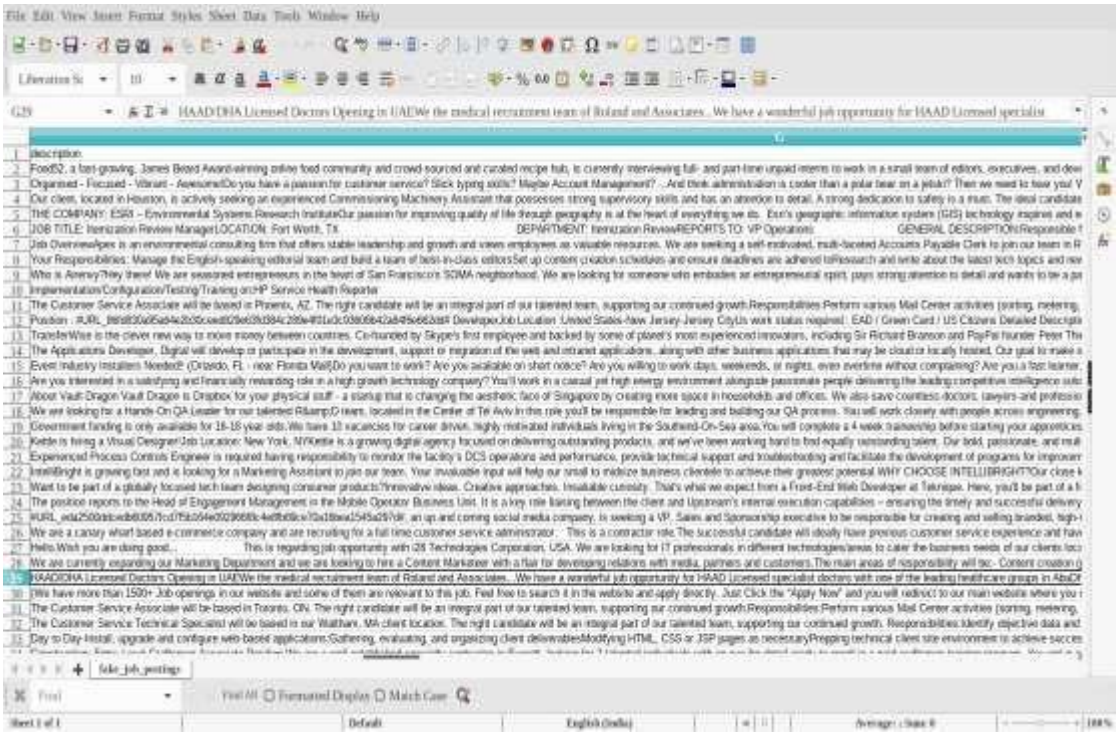


Fig 1.2.1.6: Sample dataset of description attribute

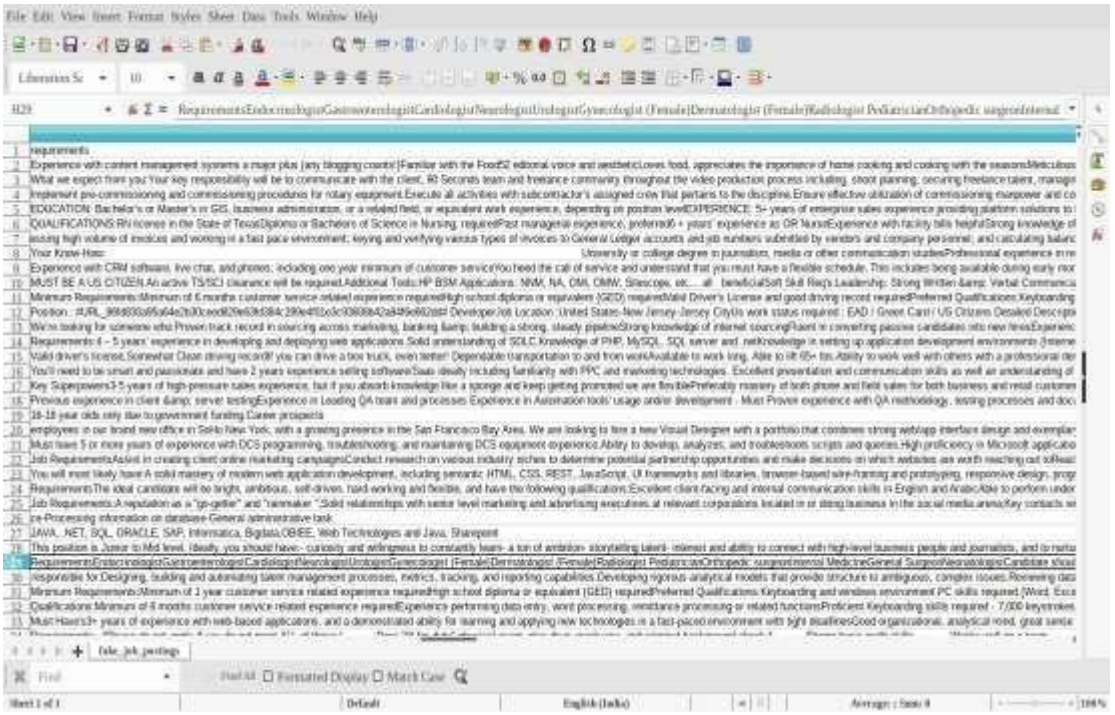


Fig 1.2.1.7: Sample dataset of requirements attribute

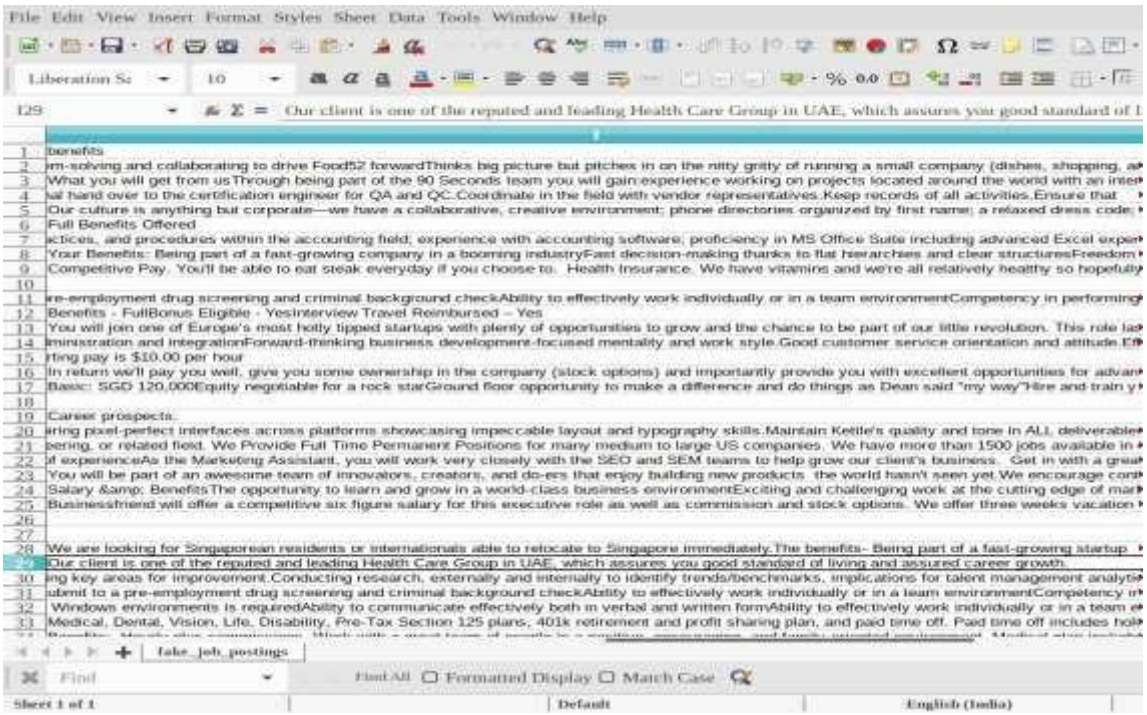
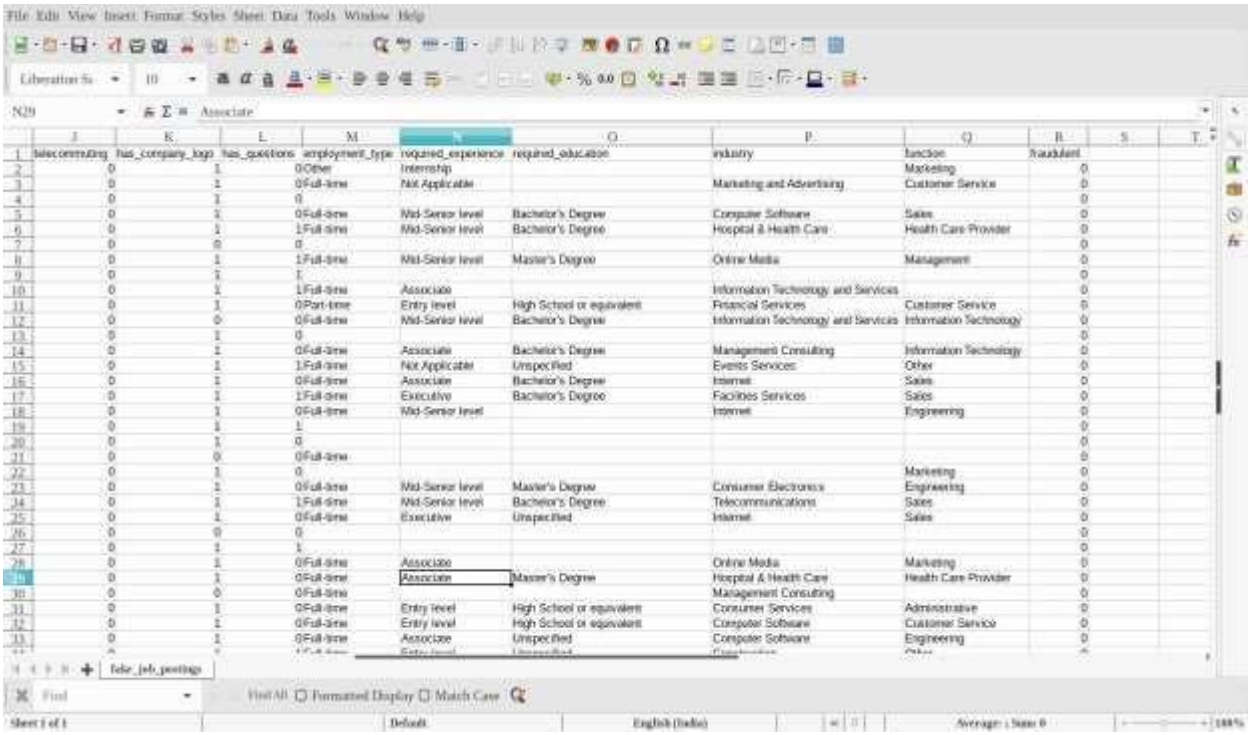


Fig 1.2.1.8: Sample dataset of benefits attribute



	J	K	L	M	N	O	P	Q	R	S	T
1	telecommuting	has_company_logo	has_questions	employment_type	required_experience	required_education	industry	function	fraudulent		
2	0	1	0	Other	Internship			Marketing	0		
3	0	1	0	Full-time	Not Applicable		Marketing and Advertising	Customer Service	0		
4	0	1	0						0		
5	0	1	0	Full-time	Mid-Senior level	Bachelor's Degree	Computer Software	Sales	0		
6	0	1	1	Full-time	Mid-Senior level	Bachelor's Degree	Hospital & Health Care	Health Care Provider	0		
7	0	0	0						0		
8	0	1	1	Full-time	Mid-Senior level	Master's Degree	Online Media	Management	0		
9	0	1	1						0		
10	0	1	1	Full-time	Associate		Information Technology and Services		0		
11	0	1	0	Part-time	Entry level	High School or equivalent	Financial Services	Customer Service	0		
12	0	0	0	Full-time	Mid-Senior level	Bachelor's Degree	Information Technology and Services	Information Technology	0		
13	0	1	0						0		
14	0	1	0	Full-time	Associate	Bachelor's Degree	Management Consulting	Information Technology	0		
15	0	1	1	Full-time	Not Applicable	Unspecified	Events Services	Other	0		
16	0	1	0	Full-time	Associate	Bachelor's Degree	Internet	Sales	0		
17	0	1	1	Full-time	Executive	Bachelor's Degree	Facilities Services	Sales	0		
18	0	1	0	Full-time	Mid-Senior level		Internet	Engineering	0		
19	0	1	1						0		
20	0	1	0						0		
21	0	0	0	Full-time					0		
22	0	1	0					Marketing	0		
23	0	1	0	Full-time	Mid-Senior level	Master's Degree	Consumer Electronics	Engineering	0		
24	0	1	1	Full-time	Mid-Senior level	Bachelor's Degree	Telecommunications	Sales	0		
25	0	1	0	Full-time	Executive	Unspecified	Internet	Sales	0		
26	0	0	0						0		
27	0	1	1						0		
28	0	1	0	Full-time	Associate		Online Media	Marketing	0		
29	0	1	0	Full-time	Associate	Master's Degree	Hospital & Health Care	Health Care Provider	0		
30	0	0	0	Full-time			Management Consulting		0		
31	0	1	0	Full-time	Entry level	High School or equivalent	Consumer Services	Administrative	0		
32	0	1	0	Full-time	Entry level	High School or equivalent	Computer Software	Customer Service	0		
33	0	1	0	Full-time	Associate	Unspecified	Computer Software	Engineering	0		

Fig 1.2.1.9: Sample dataset of remaining attributes

Data preprocessing :

Three common data pre-processing steps are:

- Formatting :
    - The data selected may not be in a format that is suitable to work with. The data may be in a relational database and to be converted into a flat file, or the data may be in a proprietary file format and to be converted to a relational database or a text file.
- Cleaning :
    - Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data needed to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonymized or removed from the data entirely.

- Sampling :
    - There may be far more selected data available than is needed to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. Smaller representative sample of the selected data can be taken that may be much faster for exploring and prototyping solutions before considering the whole dataset.

.

CHAPTER 2
LITERATURE SURVEY

According to several studies, Review spam detection, Email Spam detection, Fake news detection have drawn special attention in the domain of Online Fraud Detection.

A. Review Spam DetectionPeople often post their reviews online forum regarding the products they purchase. It may guide other purchaser while choosing their products. In this context, spammers can manipulate reviews for gaining profit and hence it is required to develop techniques that detect one of these spam reviews. This can be implemented by extracting features from the reviews by extracting features using Natural Language Processing (NLP). Next, machine learning techniques are applied on these features. Lexicon based approaches may be one alternative to machine learning techniques that use the dictionary or corpus to eliminate spam reviews.

Email Spam Detection

Unwanted bulk mails, belonging to the category of spam emails, often arrive in the user's mailbox. This may lead to unavoidable storage crises as well as bandwidth consumption. To eradicate this problem, Gmail, Yahoo mail and Outlook service providers incorporate spam filters using Neural Networks. While addressing the problem of email spam detection, content based

filtering, case based filtering, heuristic based filtering, memory or instance based filtering, adaptive spam filtering approaches are taken into consideration.

Fake News Detection

Fake news in social media characterizes malicious user accounts, echo chamber effects. The fundamental study of fake news detection relies on three perspectives- how fake news is written, how fake news spreads, and how a user is related to fake news. Features related to news content and social context are extracted and machine learning models are imposed to recognize fake news.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in systems engineering, information systems and software engineering, is the process of creating or altering systems, and the models and methodologies that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of software development methodologies.

Existing System

Efficient Detection: By utilizing machine learning and classification techniques, the system can quickly analyze a large number of job posts to identify potential scams. This saves time and effort for job seekers.

Accuracy: Machine learning algorithms can learn patterns and characteristics of fake job postings, allowing the system to make accurate predictions and identify fraudulent posts more effectively than manual screening.

Scalability: The system can handle a vast amount of data, making it suitable for analyzing a large number of job postings from various sources on the web..

Disadvantages

The existing system for detecting fake job postings relies on machine learning approaches and classification techniques. It aims to identify fraudulent recruitment processes by comparing the results of different classifiers. The advantages of this approach include:

Efficient Detection: By utilizing machine learning and classification techniques, the system can quickly analyze a large number of job posts to identify potential scams. This saves time and effort for job seekers.

Accuracy: Machine learning algorithms can learn patterns and characteristics of fake job postings, allowing the system to make accurate predictions and identify fraudulent posts more effectively than manual screening.

Scalability: The system can handle a vast amount of data, making it suitable for analyzing a large number of job postings from various sources on the web.

However, the existing system also has some disadvantages:

False Positives/Negatives: Like any machine learning system, there is a possibility of false positives (legitimate job posts being identified as fake) or false negatives (fake job posts being missed). This can happen if the classifiers are not perfectly trained or if scammers find new ways to deceive the system.

Dependency on Training Data: The performance of machine learning classifiers heavily relies on the quality and representativeness of the training data. If the training data does not adequately cover all types of fake job postings, the system's effectiveness may be limited.

Adaptability to Evolving Scams: Scammers constantly evolve their tactics to bypass detection systems. The existing system may struggle to keep up with new and sophisticated techniques used in fake job postings.

Interpretability: Some machine learning classifiers, such as deep learning models, can be challenging to interpret. This lack of transparency may make it difficult to understand the reasons behind the system's predictions or identify potential biases.

Proposed system

The target of this study is to detect whether a job post is fraudulent or not. Identifying and eliminating these fake job advertisements will help the jobseekers to concentrate on legitimate job posts only a couple of classifiers are employed such as Naive Bayes Classifier, Decision Tree Classifier, K-nearest Neighbor Classifier, and Random Tree Classifier for classifying job post as fake. It is to be noted that the attribute ‘fraudulent’ of the dataset is kept as target class for classification purpose. At first, the classifiers are trained using 80% of the entire dataset and later 20% of the entire dataset is used for the prediction purpose. The performance measure metrics such as Accuracy, are used for evaluating the prediction for each of these classifiers. Finally, the classifier that has the best performance with respect to the metrics is chosen as the best candidate model.

Organization of Project

The technique which is developed is taking input as a job_id and compares the input from the label encoded dataset. If the input matches, then it predicts using Random Forest Classifier and displays the result.

We have four modules in our project.

- Data Collection
- Data Pre-Processing
- Apply Algorithm
- Evaluation

Project Planning:

Define the project scope: Clearly define the objectives, deliverables, and boundaries of the project. Identify the key stakeholders and their roles.

Conduct a feasibility study: Evaluate the project's feasibility in terms of resources, time, and budget. Identify potential risks and constraints that may impact project success.

Develop a project plan: Create a comprehensive project plan that includes a detailed breakdown of tasks, timelines, resources, and dependencies.

Define project milestones: Identify significant milestones or checkpoints in the project timeline to track progress and ensure timely completion.

Allocate resources: Identify the resources required for each task and allocate them accordingly. Consider human resources, equipment, and materials needed.

Create a project schedule: Develop a timeline that outlines the start and end dates of each task, considering dependencies and resource availability.

Establish a communication plan: Define communication channels and protocols to ensure effective collaboration and information flow among team members and stakeholders.

Identify and manage risks: Identify potential risks and develop a risk management plan to mitigate or address them. Continuously monitor and update the plan as needed.

Set project milestones: Define specific milestones or deliverables to measure progress and provide checkpoints for evaluation.

Monitor and control progress: Regularly track project progress, compare it against the project schedule, and implement necessary adjustments to keep the project on track.

Document and report project status: Maintain accurate documentation of project activities, milestones, and changes. Regularly report project status to stakeholders to ensure transparency and alignment. Bellow table show my different activity and planning of my complete project. Whole project was developed in 3months time period from March 2023 to May 2023

Week	Activity	Status
1	Project planning	Completed
2	Analysis of current scenario	Completed
3	Gathered requirements	Completed
4	Design databases	Completed
5	Design web forms	Completed
6	Develop coding	Completed
7	Testing	Completed
8	Implemented	Completed
9	Maintain	Completed

Table 1 Project plan and Schedule

Bellow is Gantt chart for my project plan

Project Timeline

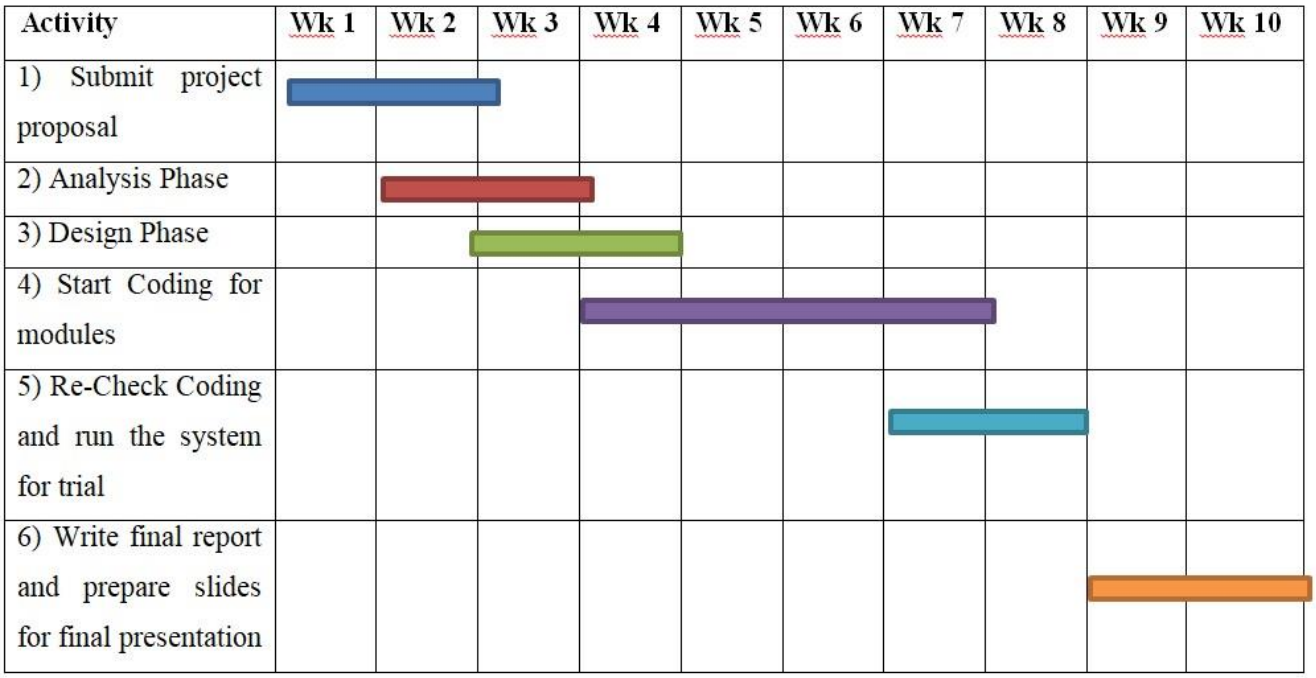


Figure 3.1 Gantt chart

PERT Chart:

A PERT (Program Evaluation and Review Technique) chart is a visual representation of a project's tasks, their dependencies, and their estimated durations. It helps in identifying critical path activities and provides a visual overview of the project timeline.

Gantt Chart:

A Gantt chart is another visual representation of a project schedule. It displays project tasks as horizontal bars along a timeline, showing their start and end dates, durations, and dependencies. Gantt charts provide a clear view of task interdependencies and resource allocation.

Both PERT and Gantt charts can be used for project scheduling and planning purposes. While the PERT chart focuses on task dependencies and critical path analysis, the Gantt chart emphasizes the timeline, milestones, and resource allocation.

CHAPTER 4

IMPLEMENTATION

Machine Learning

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.

The process of learning begins with observations or data in order to look for patterns in data and make better decisions in the future based on the examples that are provided. The primary aim is to allow the computers to learn automatically without human intervention or assistance and adjust actions accordingly.

Machine learning algorithms are often categorized as supervised or unsupervised.

- Supervised machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events.
- Unsupervised machine learning algorithms are used when the information used to train is neither classified nor labeled.

Algorithms used in our project are :

- Logistic Regression
- Random Forest Classifier
- Support Vector Machine
- Decision Tree
- K-Nearest Neighbours
- Naive-Bayes

Single Classifier based Prediction

Classifiers are trained for predicting the unknown test cases. The following classifiers are used while detecting fake job posts.

Naive Bayes Classifier :

The Naive Bayes classifier is a supervised classification tool that exploits the concept of Bayes Theorem of Conditional Probability. The decision made by this classifier is quite effective in practice even if its probability estimates are inaccurate. This classifier obtains a very promising result in the following scenario- when the features are independent or features are completely functionally dependent. The accuracy of this classifier is not related to feature dependencies; rather it is the amount of information loss of the class due to the independence assumption is needed to predict the accuracy. In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values Decision Tree Classifier :

A Decision Tree (DT) is a classifier that exemplifies the use of tree-like structure. It gains knowledge on classification. Each target class is denoted as a leaf node of DT and non-leaf nodes of DT are used as a decision node that indicates a certain test. The outcomes of those tests are identified by either of the branches of that decision node. Starting from the beginning at the root this tree goes through it until a leaf node is reached. It is the way of obtaining classification results from a decision tree. Decision tree learning is an approach that has been applied to spam filtering. This can be useful for forecasting the goal based on some criterion by implementing and training this model.

Important Terminology related to Decision Trees :

1. **Root Node:** It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:** It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:** When a sub-node splits into further sub-nodes, then it is called the decision node.
4. **Leaf / Terminal Node:** Nodes that do not split are called Leaf or Terminal nodes.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.

6. **Branch / Sub-Tree:** A subsection of the entire tree is called branch or subtree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.

Support Vector Machine :

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

The advantages of support vector machines are:

- Effective in high dimensional spaces.
- Still effective in cases where the number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

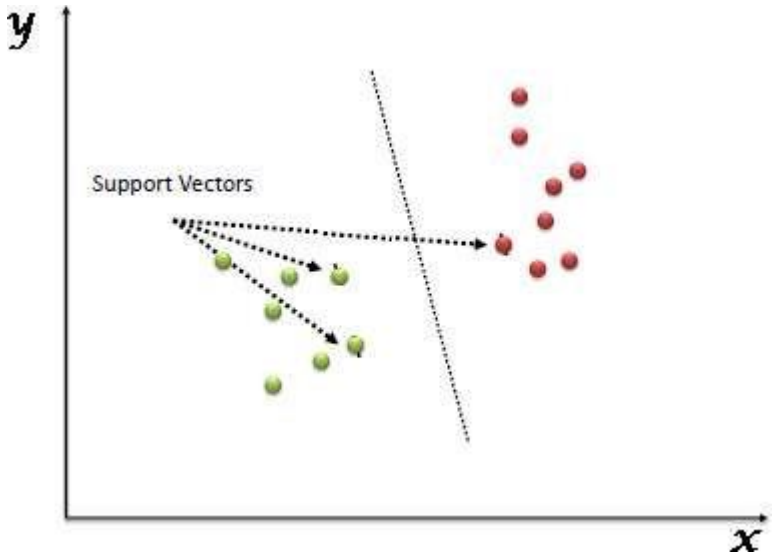


Fig 2.2.1: SVM

Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

Logistic Regression : Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc. Types of Logistic Regression

- **Binary or Binomial:** In such a kind of classification, a dependent variable will have only two possible types either 1 and 0
- **Multinomial:** In such a kind of classification, dependent variable can have 3 or more possible unordered types or the types having no quantitative significance.
- **Ordinal:** In such a kind of classification, dependent variable can have 3 or more possible ordered types or the types having a quantitative significance

K-Nearest Neighbor(KNN) : K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

- The K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.

- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.
- The K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- The KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Ensemble Approach based Classifiers

Ensemble approach facilitates several machine learning algorithms to perform together to obtain higher accuracy of the entire system. Random forest (RF) exploits the concept of ensemble learning approach and regression technique applicable for classification based problems. This classifier assimilates several tree-like classifiers which are applied on various sub-samples of the dataset and each tree casts its vote to the most appropriate class for the input. Boosting is an efficient technique where several unstable learners are assimilated into a single learner in order to improve accuracy of classification. Boosting technique applies the classification algorithm to the reweighted versions of the training data and chooses the weighted majority vote of the sequence of classifiers. AdaBoost is a good example of boosting technique that produces improved output even when the performance of the weak learners is inadequate. Boosting algorithms are quite efficient in solving spam filtration problems. Gradient boosting algorithm is another boosting technique based classifier that exploits the concept of decision tree. It also minimizes the prediction loss.

Random Forest Algorithm :

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in

ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

CHAPTER 5

SOFTWARE REQUIREMENT SPECIFICATION

5. SOFTWARE REQUIREMENT SPECIFICATION

5.1 Requirements Specification:

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

5.1.1 HARDWARE REQUIREMENTS:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

- System : Pentium IV 2.4 GHz.
- Hard Disk : 100 GB.
- Monitor : 15 VGA Color.
- Mouse : Logitech.
- RAM : 1 GB.

5.1.2 SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

- Domain : Machine Learning
  Programming Language : Python 3.6
- Dataset : fake_job_postings.csv
- Packages : Numpy, Pandas, Matplotlib, Scikit-learn, Seaborn, Tkinter
  Tool : Jupyter Notebook, Google Colab

5.2 FUNCTIONAL REQUIREMENTS:

The functional requirement refers to the system needs in an exceedingly computer code engineering method.

The key goal of determinant “functional requirements” in an exceedingly product style and implementation is to capture the desired behavior of a software package in terms of practicality and also the technology implementation of the business processes.

System Modules:

- Data Collection:
- This module focuses on collecting data from social media platforms, specifically Twitter, Facebook, and Instagram. It involves accessing public posts and tweets related to women's safety in Indian cities. Data collection techniques may include API integration, web scraping, and data crawling.
- Data Preprocessing:
- The data preprocessing module involves cleaning and transforming the collected data to prepare it for analysis. This includes removing irrelevant or duplicate posts, handling missing data, normalizing text, and performing language processing tasks like tokenization, stemming, and lemmatization. It also involves handling multimedia content such as images and videos associated with the tweets.

- Sentiment Analysis:
- The sentiment analysis module aims to analyze the sentiment expressed in tweets and social media posts related to women's safety. It uses machine learning or natural language processing techniques to classify the sentiment as positive, negative, or neutral. This analysis helps in understanding public perception and emotional response towards women's safety in Indian cities.
- Topic Modeling:
- The topic modeling module focuses on identifying and extracting topics or themes from the collected tweets and social media posts. It uses techniques such as Latent Dirichlet Allocation (LDA) or Non-Negative Matrix Factorization (NMF) to cluster similar tweets and identify common discussion topics. This analysis provides insights into the key concerns and issues related to women's safety.
- Text Classification:

- The text classification module aims to classify tweets and social media posts into different categories or labels, such as harassment, assault, safety tips, awareness campaigns, and support initiatives. Machine learning algorithms, such as Naive Bayes, Support Vector Machines (SVM), or deep learning models like Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN), can be used for this task.

- Network Analysis:

- The network analysis module focuses on analyzing the network structure and relationships among users discussing women's safety on social media platforms. It identifies influential users, communities, and patterns of interactions. Network analysis techniques like social network analysis (SNA) and graph algorithms can be applied to identify key players and understand the information flow within the network.

- Visualization and Reporting:

- This module involves visualizing the analyzed data and generating informative reports. Visualizations can include word clouds, sentiment distribution charts, topic distribution plots, network graphs, and interactive dashboards. These visual representations help in presenting the findings and insights to stakeholders, policymakers, and the general public.

- Recommendations and Interventions:

- Based on the analysis and insights gained from the previous modules, this module focuses on generating actionable recommendations and interventions to improve women's safety in Indian cities. These recommendations can be directed towards policymakers, law enforcement agencies, NGOs, and other stakeholders involved in promoting women's safety.

The system modules described above work together to analyze the data collected from social media platforms and provide valuable insights into women's safety in Indian cities. Each module performs specific tasks and contributes to the overall goal of leveraging social media to promote women's safety and create awareness among the general public.

5.3 NON FUNCTIONAL REQUIREMENTS

All the other requirements which do not form a part of the above specification are categorized as Non-Functional needs. A system perhaps needed to gift the user with a show of the quantity of records during info. If the quantity must be updated in real time, the system architects should make sure that the system

is capable of change the displayed record count at intervals associate tolerably short interval of the quantity of records dynamic. Comfortable network information measure may additionally be a non-functional requirement of a system.

The following are the features:

- Accessibility
- Availability
- Backup
- Certification
- Compliance
- Configuration Management
- Documentation
- Disaster Recovery
- Efficiency(resource consumption for given load)
- Interoperability

5.4 PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

- The system should be able to interface with the existing system
- The system should be accurate
- The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

5.5 Feasibility Study:

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

- Technical Feasibility
- Operation Feasibility

Economical Feasibility

5.5.1 Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

- Does the necessary technology exist to do what is suggested?
- Do the proposed equipments have the technical capacity to hold the data required to use the new system?
- Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
- Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

5.5.2 Operational Feasibility

User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-friendly to the Client.

Reliability

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

Security

The web server and database server should be protected from hacking, virus etc **Portability**

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

Availability

This software will be available always.

Maintainability

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front-end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server.

Users access these forms by using the user-ids and the passwords.

5.5.3 Economic Feasibility

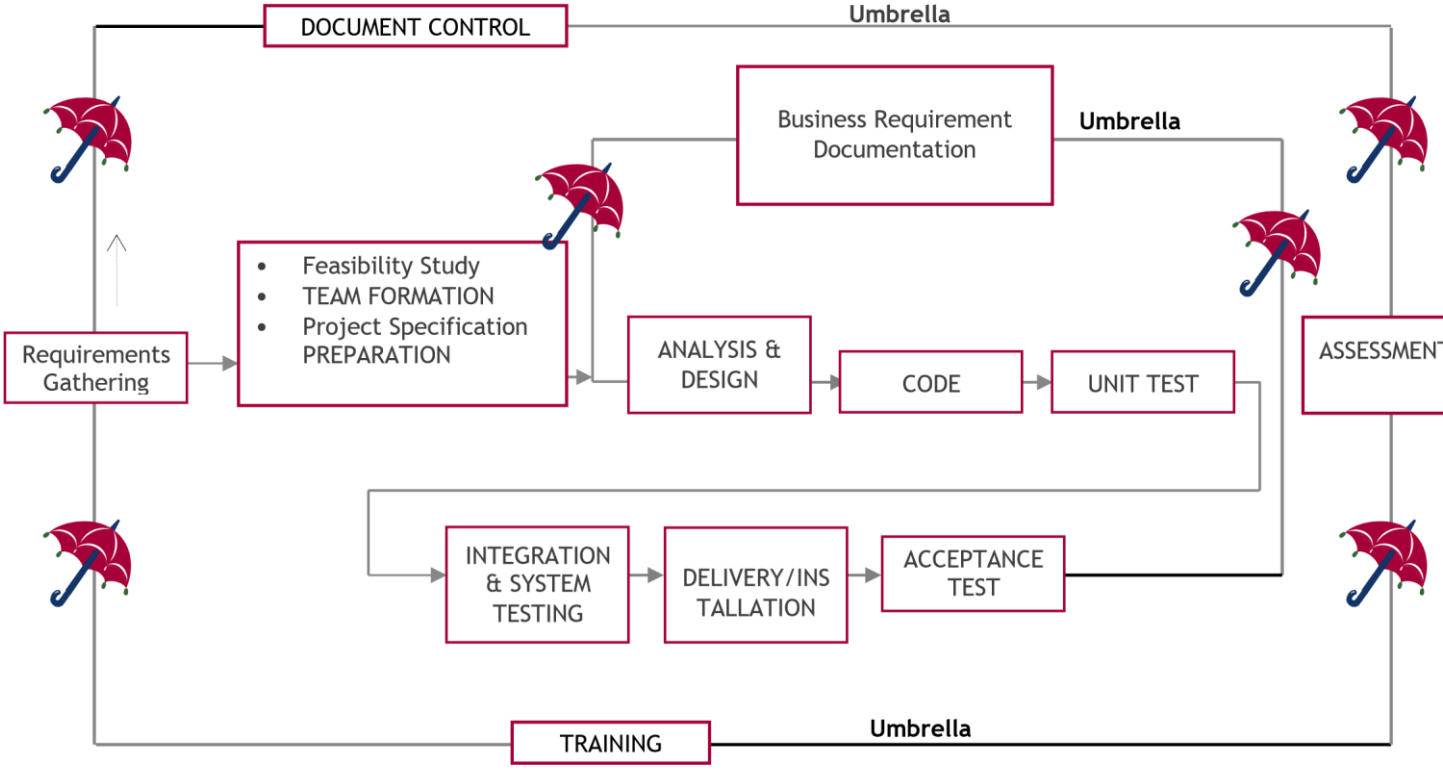
The computerized system takes care of the present existing system's data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database.

Further some of the linked transactions take place in different locations.

CHAPTER 6

METHODOLOGY



6. Methodology

SDLC (Software Development Life Cycle) – Umbrella Model

Fig no. 6.1 Umbrella model

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

Requirements Gathering Stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be

managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.

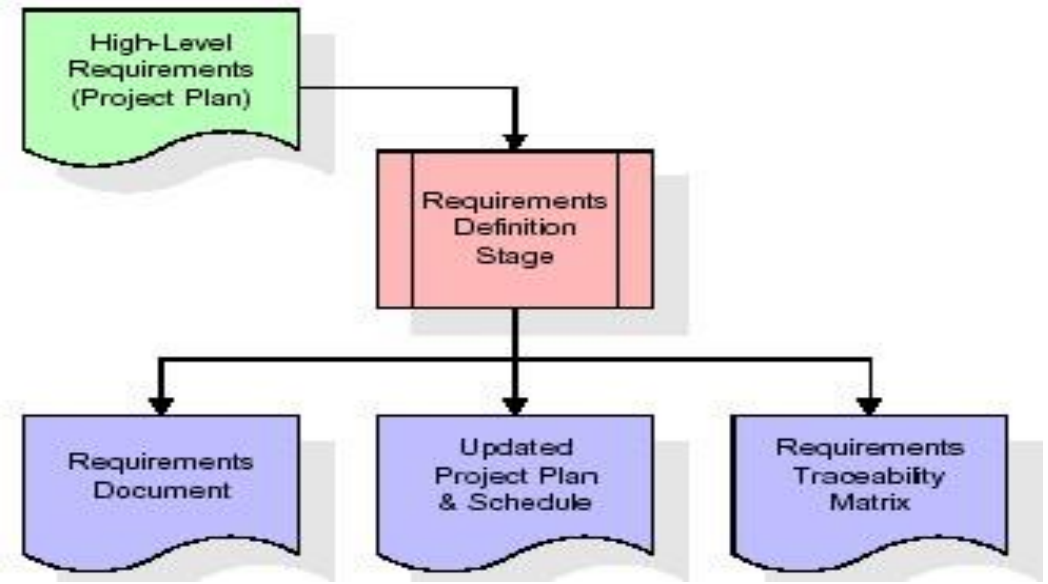


Fig no. 6.2 Requirements Gathering stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Feasibility study is all about identification of problems in a project, number of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

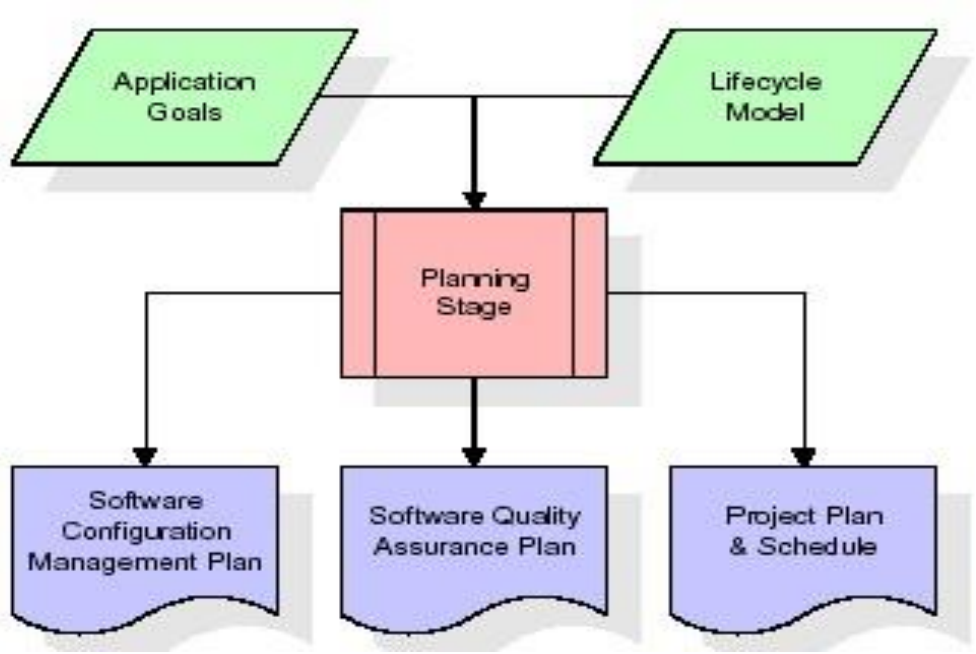


Fig no. 6.3 Analysis stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality

assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

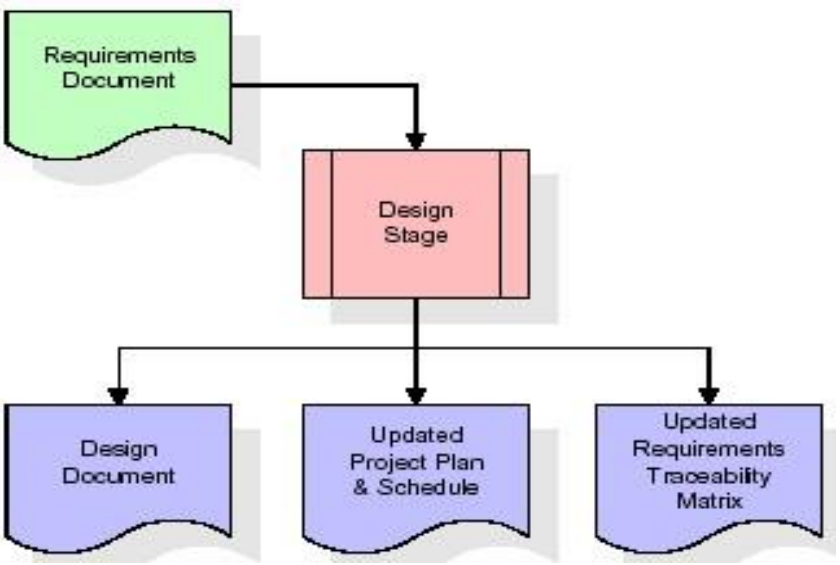


Fig no. 6.4 Designing stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats,

and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.

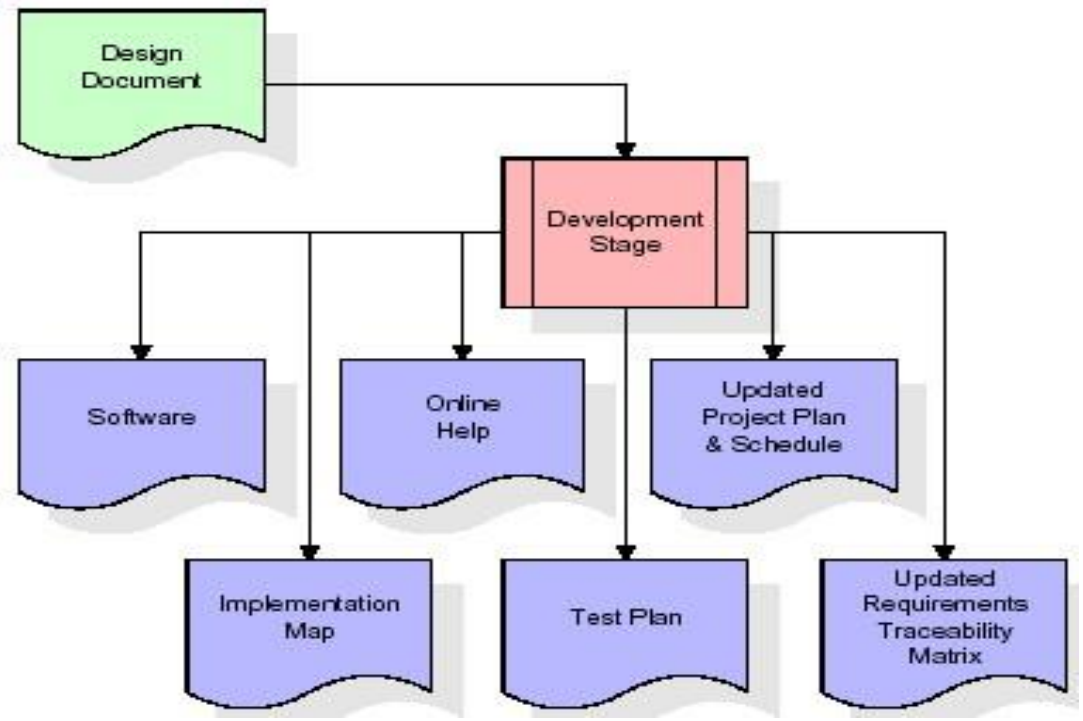


Fig no. 6.5 Coding stage

Integration & Test Stage

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

Installation & Acceptance Test

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.

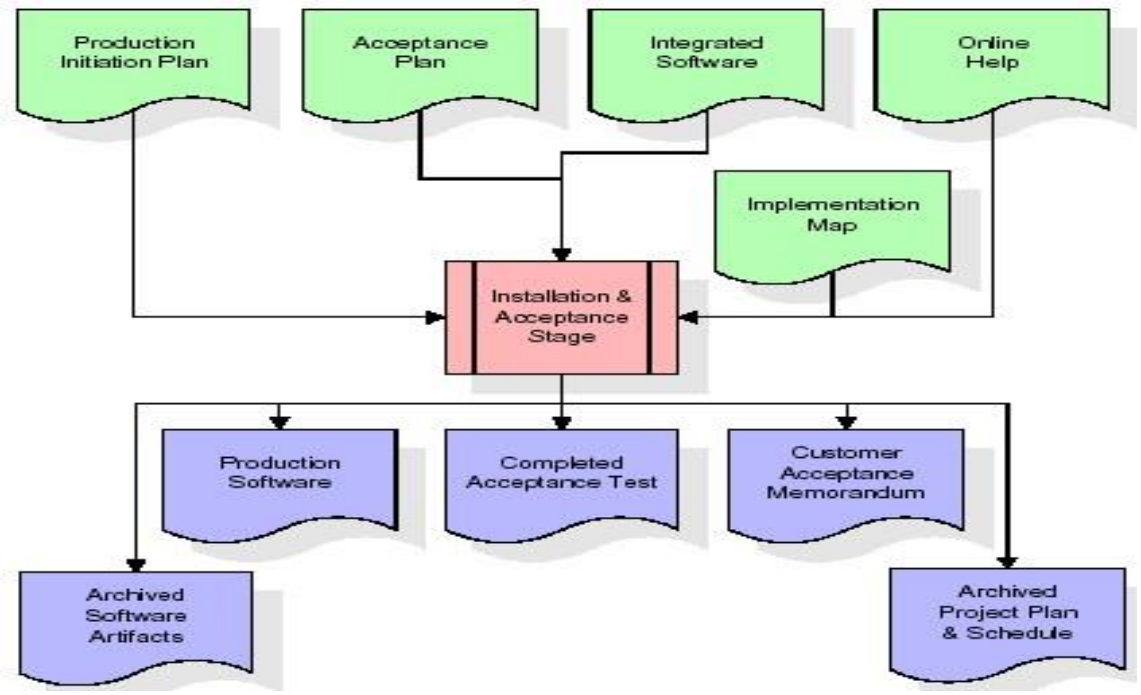


Fig no. 6.7 Installation

Maintenance

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

CHAPTER 7

SYSTEM DESIGN & UML DESIGN

7. System Design

Introduction

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer's goal is to produce a model or representation of an entity that will later be built. Once system requirements have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word "Quality". Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess quality. Design is the only way that we can accurately translate a customer's view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed, reviewed and documented. System design can be viewed from either a technical or project management perspective. From the technical point of view, design consists of four activities – architectural design, data structure design, interface design and procedural design.

Architecture Diagram

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through a network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client. For the enterprise, software

on potentially thousands of client computers is a key reason for their popularity. Web applications are used for web mail, online retail sales, discussion boards, weblogs, online banking, and more. One web application can be accessed and used by millions of people.

Like desktop applications, web applications are made up of many parts and often contain mini programs and some of which have user interfaces. In addition, web applications frequently require an additional markup or scripting language, such as HTML, CSS, or JavaScript programming language. Also, many applications use only the Python programming language, which is ideal because of its versatility.

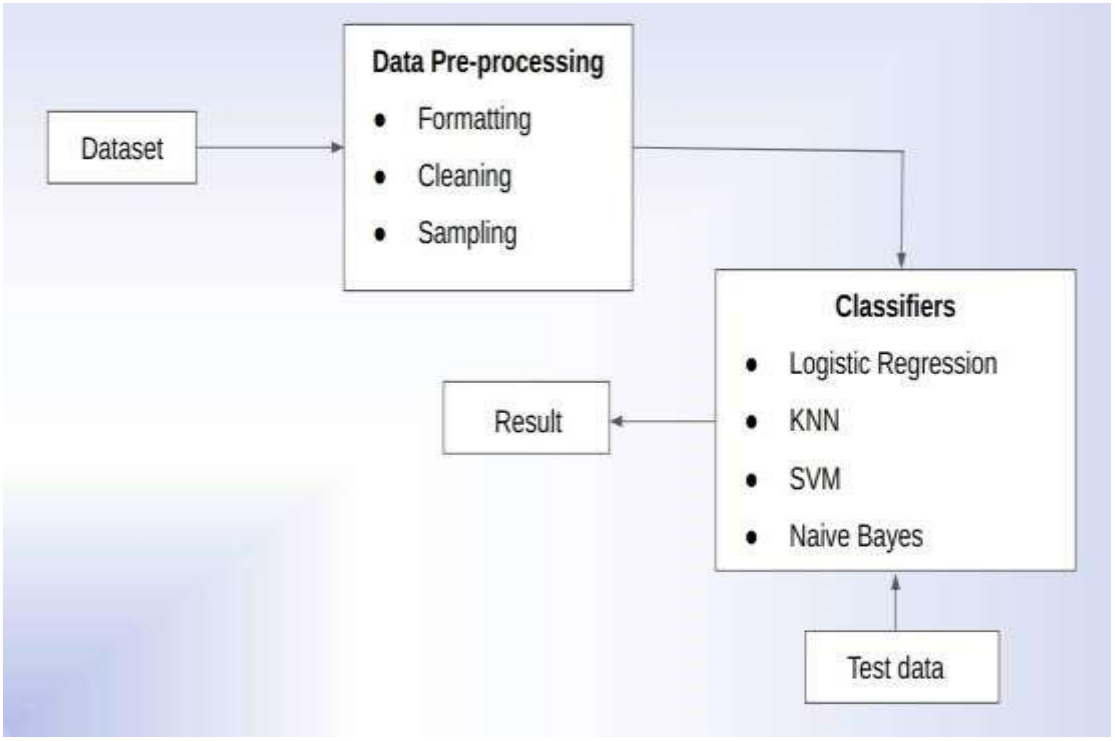


Fig 7.2: Architecture Diagram

UML Diagrams

UML stands for Unified Modeling Language. It's a rich language to model software solutions, application structures, system behavior and business processes. UML is a standard language for specifying, visualizing, constructing, and documenting the

artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997. It was initially started to capture the behavior of complex software and non-software systems and now it has become an OMG standard.

Use Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior. Dynamic Behavior means the behavior of the system when it is running/operating. Only static behavior is not sufficient to model a system; rather dynamic behavior is more important than static behavior. In UML, there are five diagrams available to model the dynamic nature and use case diagrams are one of them. Now as we have to discuss that the use case diagram is dynamic in nature, there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. Use case diagrams consist of actors, use cases and their relationships. The diagram is used to model the system subsystem of an application. A single use case diagram captures a particular functionality of a system.

The purpose of a use case diagram is to capture the dynamic aspect of a system.
However, this definition is too generic to describe the purpose, as other four diagrams (activity, sequence, collaboration, and Statechart) also have the same purpose. We will look into some specific purpose, which will distinguish it from the other four diagrams. Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

Hence to model the entire system, a number of use case diagrams are used.

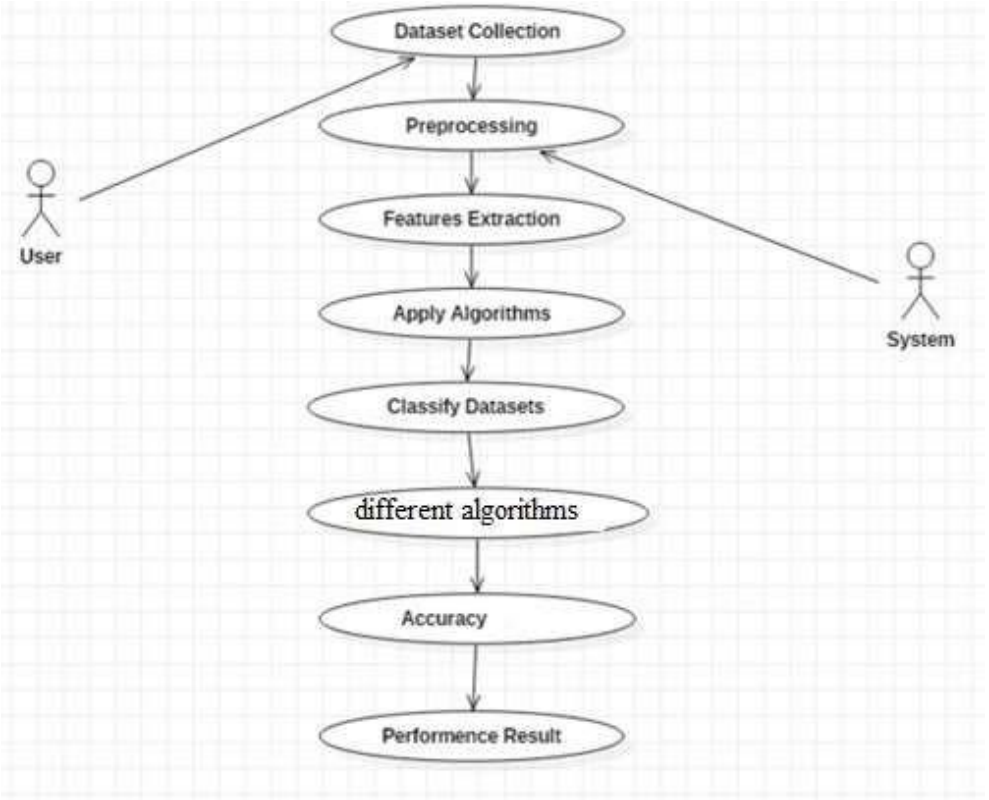


Fig 7.3.1: Use Case Diagram

Sequence Diagram

Sequence Diagrams Represent the objects participating in the interaction horizontally and time vertically. A Use Case is a kind of behavioral classifier that represents a declaration of an offered behavior. Each use case specifies some behavior, possibly including variants that the subject can perform in collaboration with one or more actors. Use cases define the offered behavior of the subject without reference to its internal structure. These behaviors, involving interactions between the actor and the subject, may result in changes to the state of the subject and communications with its environment. A use case can include possible variations of its basic behavior, including exceptional behavior and error handling.

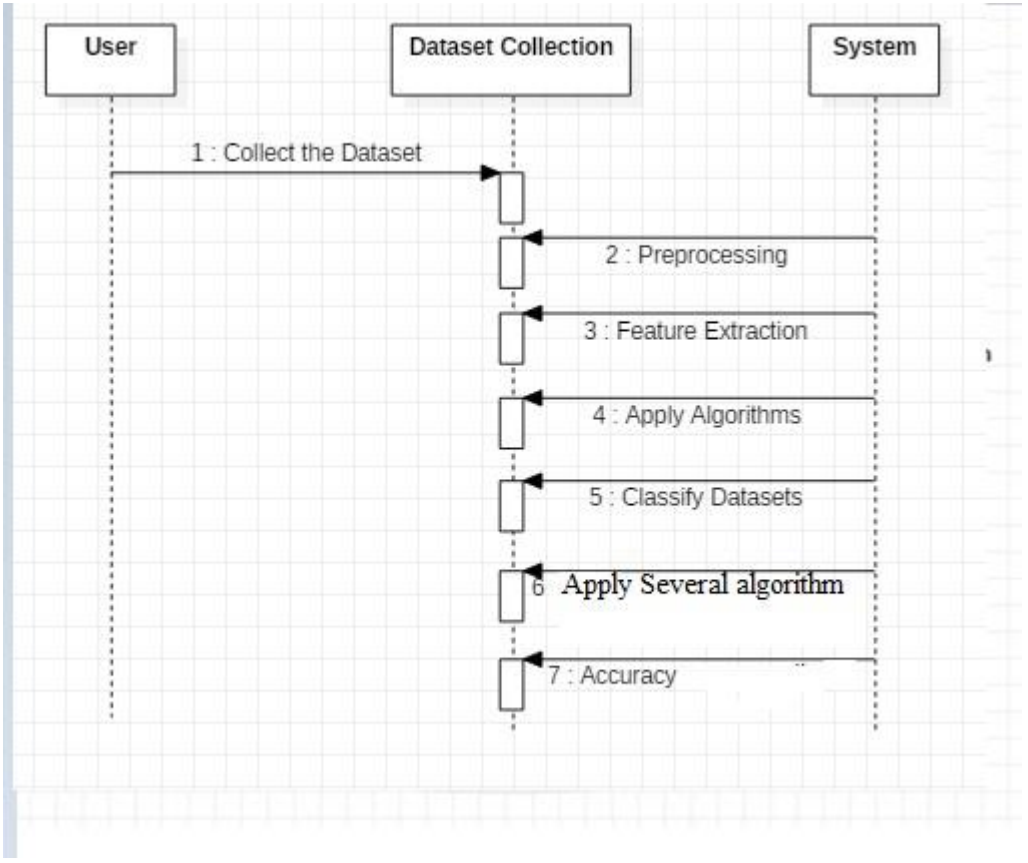


Fig 7.3.2: Sequence Diagram Activity Diagram

Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

The basic purpose of activity diagrams is similar to the other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but the activity diagram is used to show message flow from one activity to another.

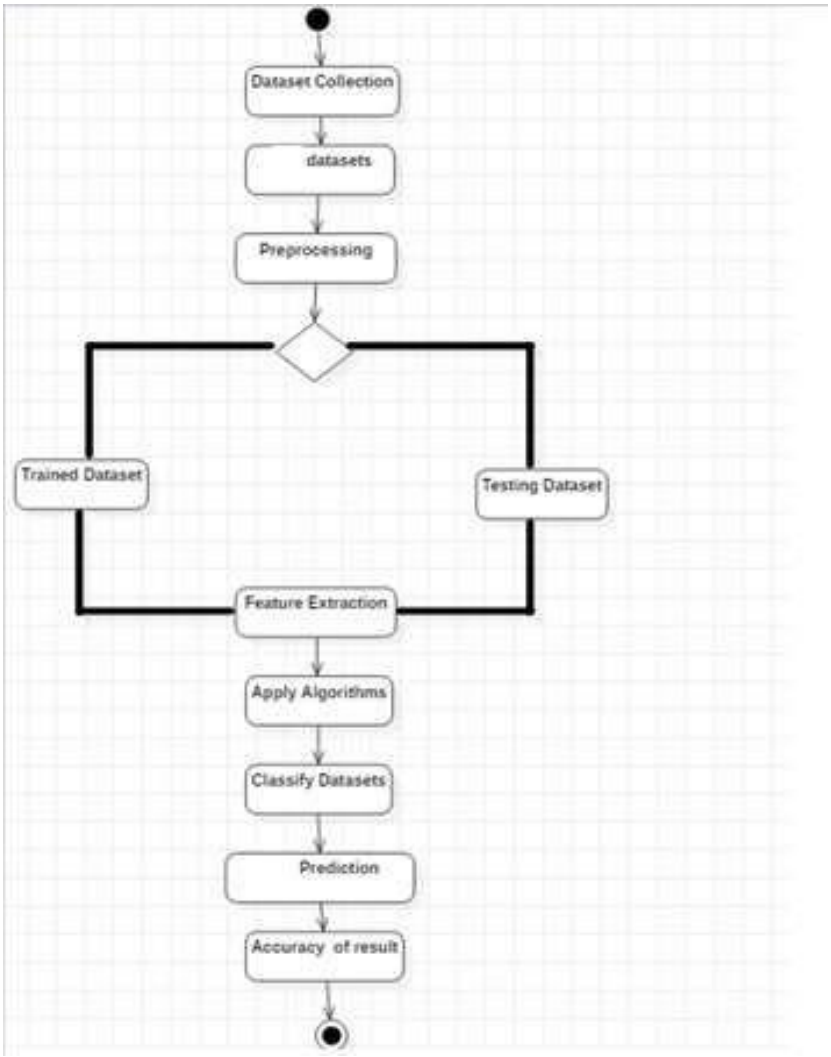


Fig 7.3.3: Activity Diagram

Class Diagram

The class diagram is the main building block of object-oriented modeling. It is used for general conceptual modeling of the system of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

The purpose of the class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction.

UML diagrams like activity diagram, sequence diagrams can only give the sequence flow of the application, however the class diagram is a bit different. It is the most popular UML diagram in the coder community.

The purpose of the class diagram can be summarized as − ●

Analysis and design of the static view of an application.

- Describe responsibilities of a system.
- Base for component and deployment diagrams.
- Forward and reverse engineering.

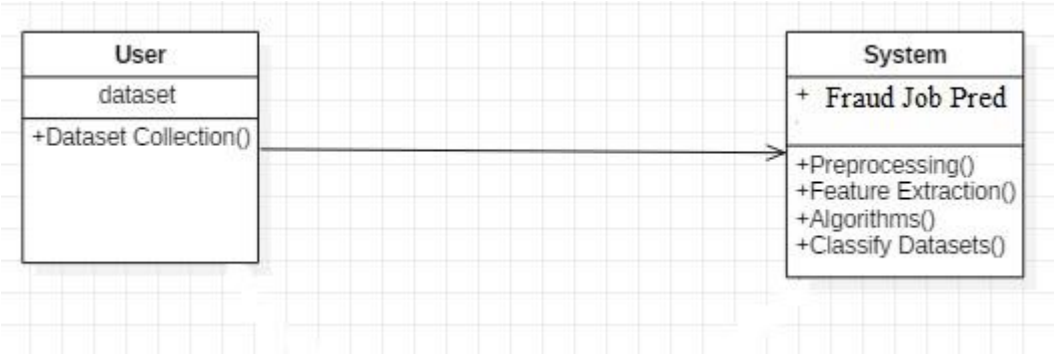


Fig 7.3.4: Class Diagram

CHAPTER 8

SYSTEM TESTING

TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

Software testing is crucial for ensuring the functionality, reliability, and accuracy of the analysis of women's safety in Indian cities using machine learning on tweets. It involves testing the software implementation of the system, including the data processing, machine learning algorithms, and result reporting. Here are some test cases that can be used for software testing:

Input Data Validation Test Cases:

Test the system's ability to handle different types of input data, including various formats of tweet data, such as JSON or CSV files.

Verify that the system performs proper data validation and handles invalid or corrupted data gracefully.

Test the system's response to missing or incomplete data fields in the input tweets.

Machine Learning Algorithm Test Cases:

Test the accuracy and performance of the machine learning algorithms used for sentiment analysis, topic modeling, and text classification.

Verify that the algorithms handle different types of tweet data, including different languages, writing styles, and sentiments.

Validate the accuracy of the algorithms by comparing their predictions with manually labeled ground truth data.

System Performance Test Cases:

Test the system's performance by processing a large volume of tweets, simulating real-world data scenarios.

Measure the system's response time and resource utilization under different loads to ensure it can handle a significant number of tweets efficiently.

Validate the system's scalability by gradually increasing the number of concurrent requests or tweet data size.

Boundary and Error Handling Test Cases:

Test the system's behavior when handling extreme or boundary cases, such as tweets with extremely long text or excessive multimedia content.

Verify that the system handles errors gracefully, displaying appropriate error messages and recovering from failures without data loss or corruption.

Validate the system's behavior when encountering unexpected situations, such as network connectivity issues or service interruptions.

Reporting and Visualization Test Cases:

Test the accuracy and completeness of the reports and visualizations generated by the system.

Validate the system's ability to generate informative and visually appealing representations of the analyzed tweet data.

Verify that the reports provide meaningful insights and accurately reflect the sentiment, topics, and classifications obtained from the analysis.

Integration and Compatibility Test Cases:

Test the integration of the analysis system with other components, such as data storage systems, APIs, or visualization tools.

Validate the compatibility of the system with different operating systems, web browsers, or platforms on which it will be deployed.

Verify the system's ability to handle data from multiple social media platforms, ensuring compatibility with Twitter, Facebook, and Instagram APIs.

By conducting thorough software testing and executing these test cases, you can ensure that the analysis of women's safety in Indian cities using machine learning on tweets is robust, reliable, and delivers accurate results.

8.1 SYSTEM TESTING

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

8.2 MODULE TESTING

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

8.3 INTEGRATION TESTING

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

8.4 ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

8.5 TEST CASES:

Unit testing strategy is used in this application for testing.

Test Case Id	Test Scenario	Expected Result	Actual Result	Pass/Fail
TO1	Check whether jupyter notebook is installed	Jupyter notebook should be opened after executing command	As expected	Pass
TO2	Check if all the packages are installed	Error should not be displayed	As expected	Pass
TO3	Check if all the modules are correctly imported	Error should not be displayed	As expected	Pass
TO4	Check for empty input	Warning message should be given	As expected	Pass
TO5	Check for string input	Warning message should be given	As expected	Pass
TO6	Check for out of range input	Warning message should be given	As expected	Pass
TO7	Check whether button is working	Should give result	As expected	Pass
TO8	Check whether getting correct output	It should correctly predict output	As expected	Pass

CHAPTER 9

CODE

File name : Fake_Job_Recruitment_Detection.ipynb

```
# import libraries import numpy
as np import pandas as pd import
seaborn as sns from sklearn
import preprocessing import
matplotlib.pyplot as plt
    ax = plt.subplot(grid[n]) # feeding the figure of grid
    sns.countplot(x=col, data=df, hue='fraudulent', palette='Set2')
    ax.set_ylabel('Count', fontsize=12) # y axis label
    ax.set_title(f'{col} Distribution by Target', fontsize=15) # title
    label ax.set_xlabel(f'{col} values', fontsize=12) # x axis label
    xlabels = ax.get_xticklabels() ylabels = ax.get_yticklabels()
    ax.set_xticklabels(xlabels, fontsize=10) ax.set_yticklabels(ylabels,
    fontsize=10) plt.legend(fontsize=8) plt.xticks(rotation=90) total =
    len(df)
    sizes=[] # Get highest values in y for p
    in ax.patches: # loop to all objects
       height = p.get_height()
       sizes.append(height)
       ax.text(p.get_x()+p.get_width()/2.,
       height + 3,
            '{:1.2f}%'.format(height/total*100), ha="center",
            fontsize=10)
    ax.set_ylim(0, max(sizes) * 1.15) #set y limit based on highest heights
plt.show() fig,(ax1,ax2)= plt.subplots(ncols=2, figsize=(17,
5), dpi=100)
length=df[df["fraudulent"]==1]['requirements'].str.len()
ax1.hist(length,bins = 20,color='orangered')
ax1.set_title('Fake Post')
```

```
length=df[df["fraudulent"]==0]['requirements'].str.len()
ax2.hist(length, bins = 20) ax2.set_title('Real Post')
fig.suptitle('Characters in description') plt.show()
fig,(ax1,ax2)= plt.subplots(ncols=2, figsize=(17, 5), dpi=100)

num=df[df["fraudulent"]==1]['company_profile'].str.split().map(lambda x: len(x)) ax1.hist(num,bins
= 20,color='orangered')
ax1.set_title('Fake Post')

num=df[df["fraudulent"]==0]['company_profile'].str.split().map(lambda x: len(x))
ax2.hist(num, bins = 20) ax2.set_title('Real Post') fig.suptitle('Words in company
profile') plt.show() fraud = df[df['fraudulent']== 1] fraud.shape
fraud

not_fraud = df[df['fraudulent']== 0] not_fraud.shape
not_fraud
```

CHAPTER 10

10. OUTPUT SCREENS

Visualization Screenshots

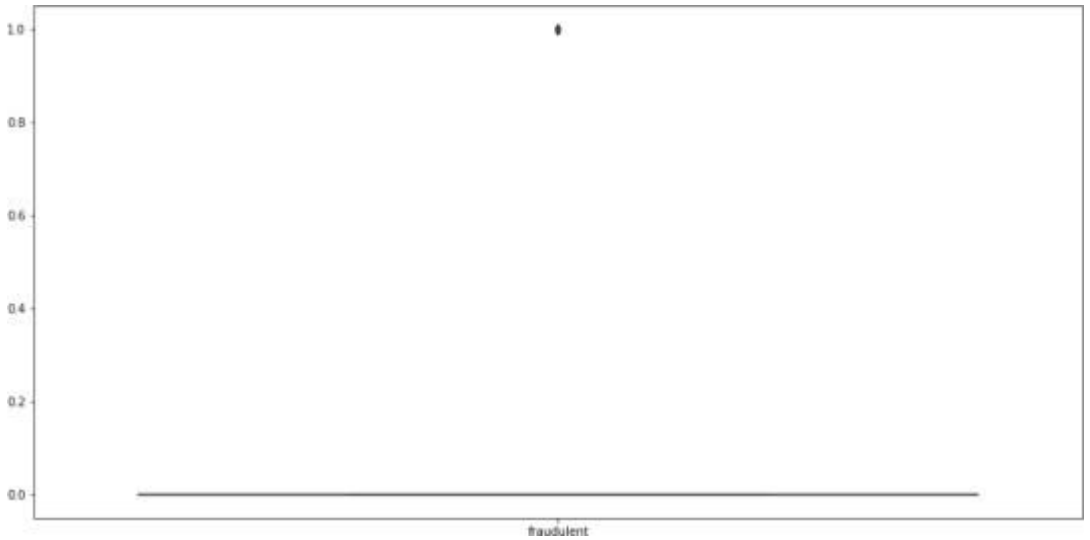


Fig 10.4.1: Outliers in fraudulent attribute

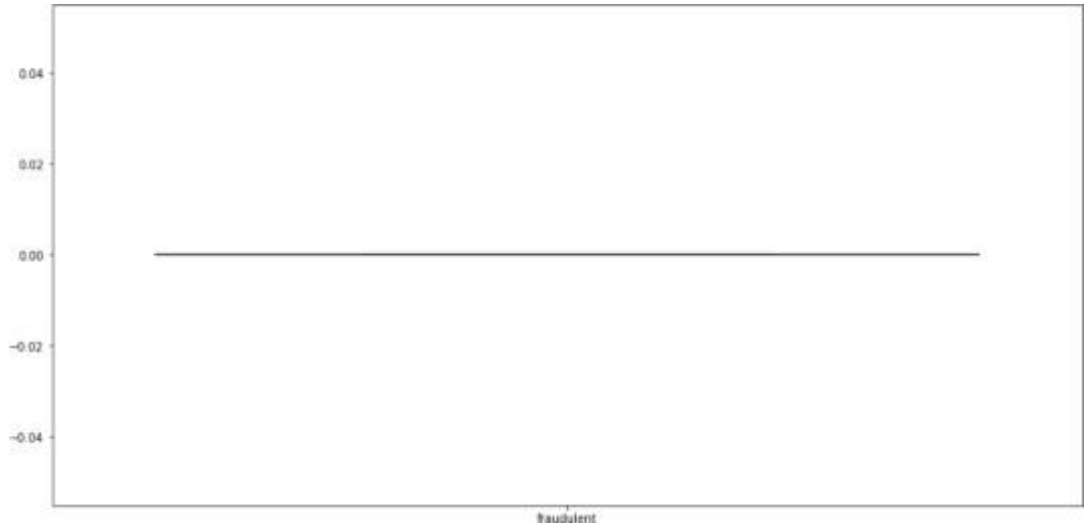


Fig 10.4.2: After removing outliers in fraudulent attribute

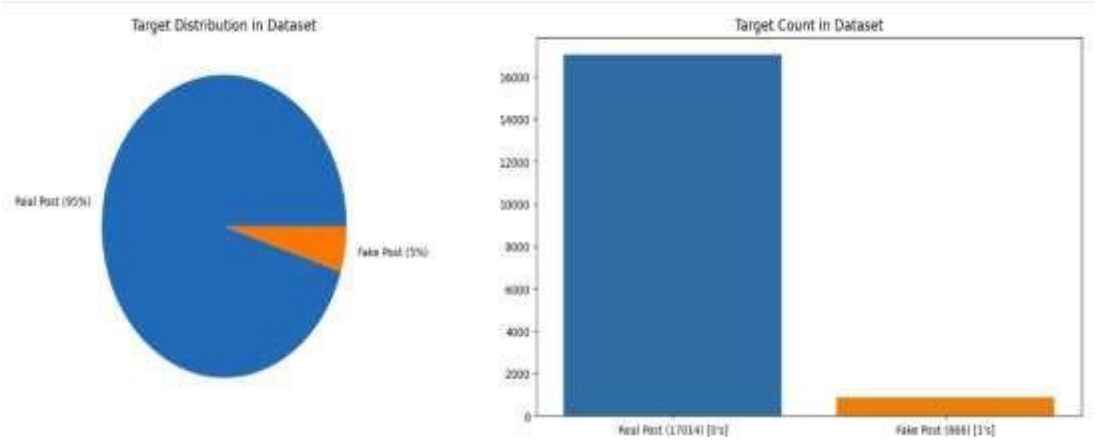


Fig 10.4.3: Fraudulent percentage in dataset

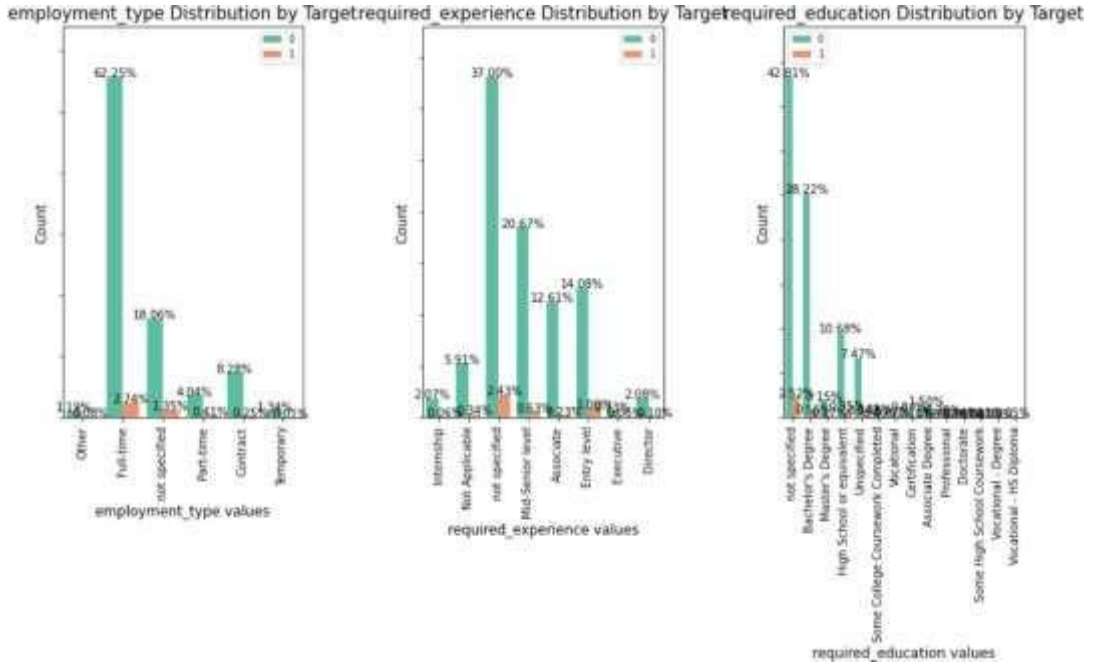


Fig 10.4.4: Visualizing categorical variable by target

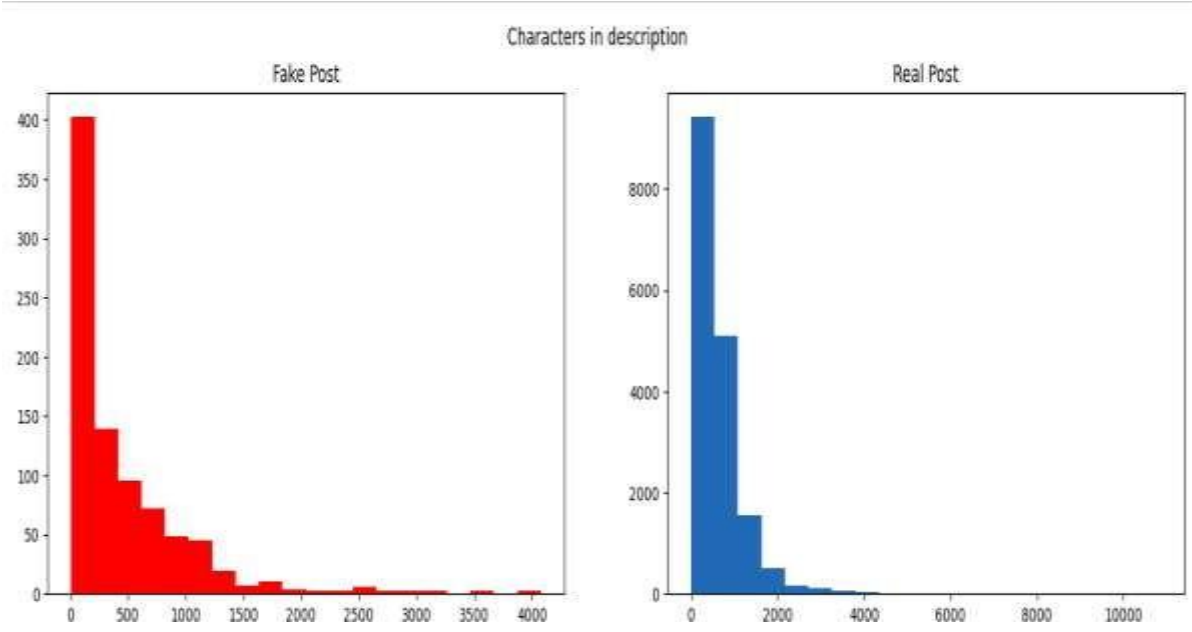


Fig 10.4.5: Comparison on characters in description

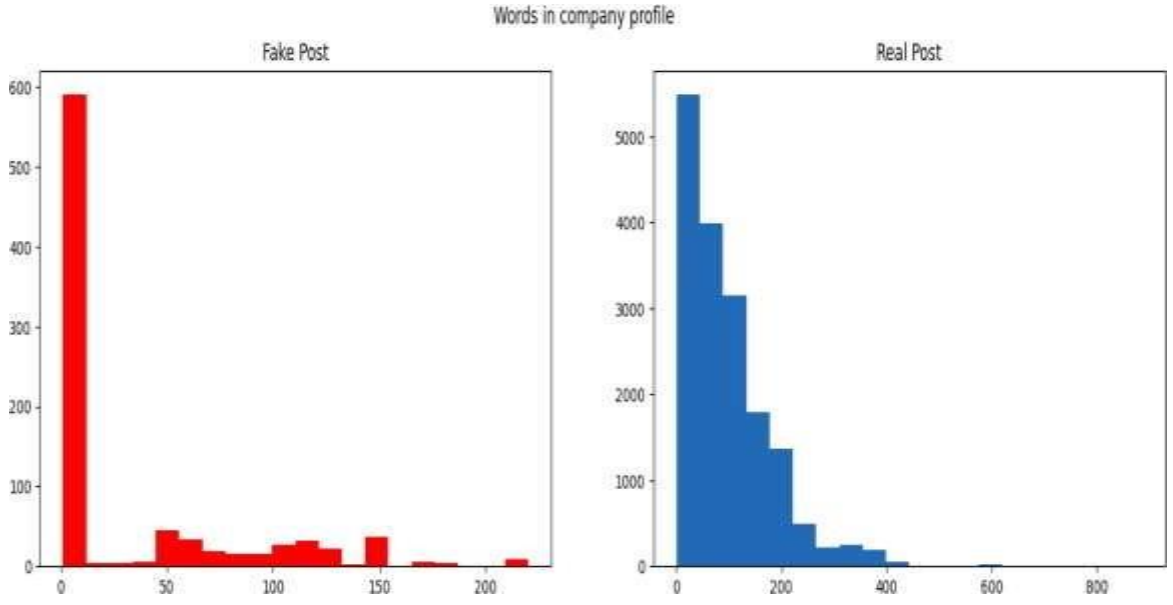


Fig 10.4.6: Comparison on number of words in Company profile

5.1 Input Screenshots

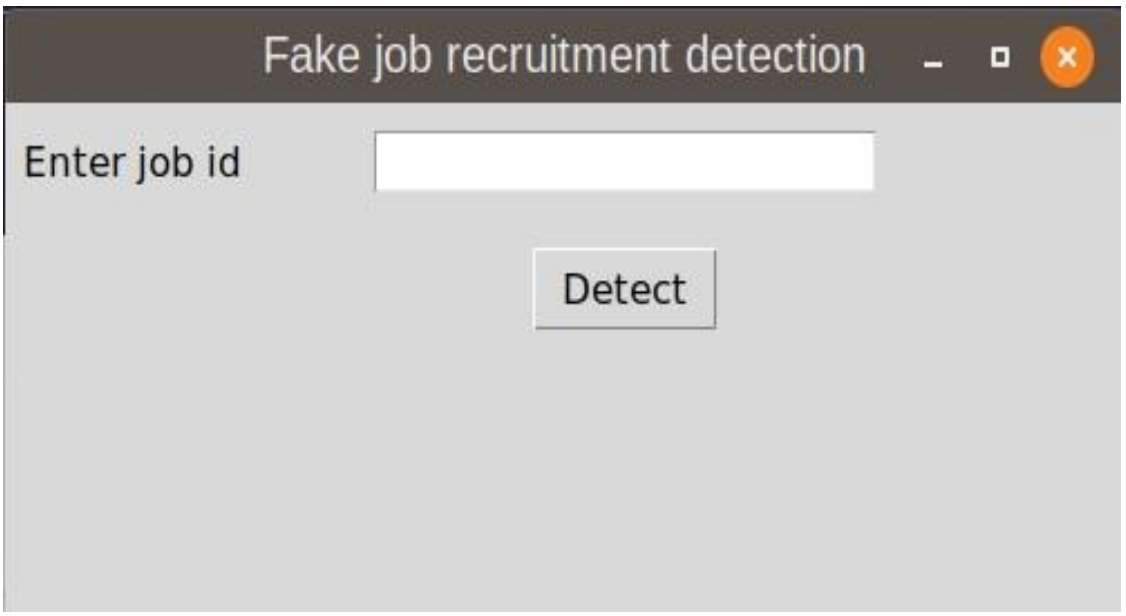


Fig 10.5: Input screen

```
accuracy with Logistic Regression: 0.9250814332247557 %
accuracy with Random Forest: 0.9876221498371336 %
accuracy with Support Vector Machine: 0.9761129207383279 %
accuracy with Decision Tree: 0.9765472312703583 %
accuracy with K-Nearest Neighbors : 0.9454940282301846 %
accuracy with Naive Bayes: 0.9274701411509229 %
```

Fig 10.1: Accuracy of algorithms

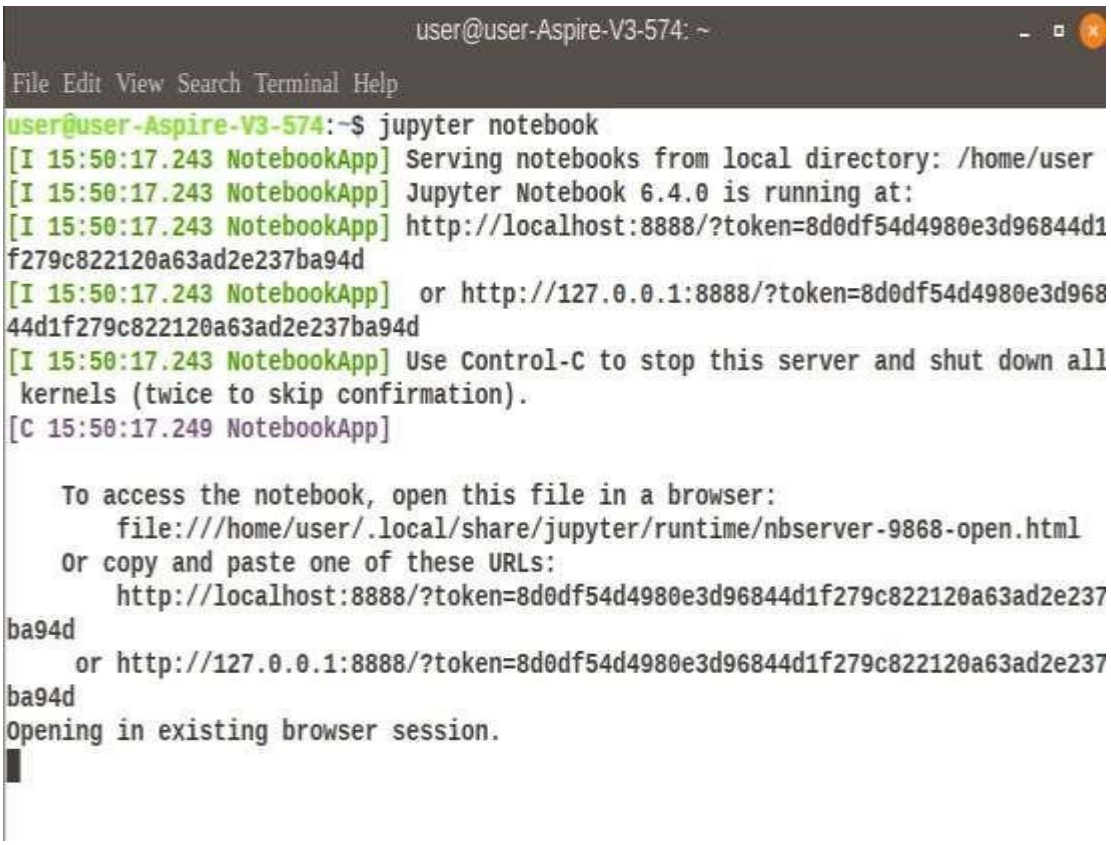


Fig 10.2: Command to open jupyter book

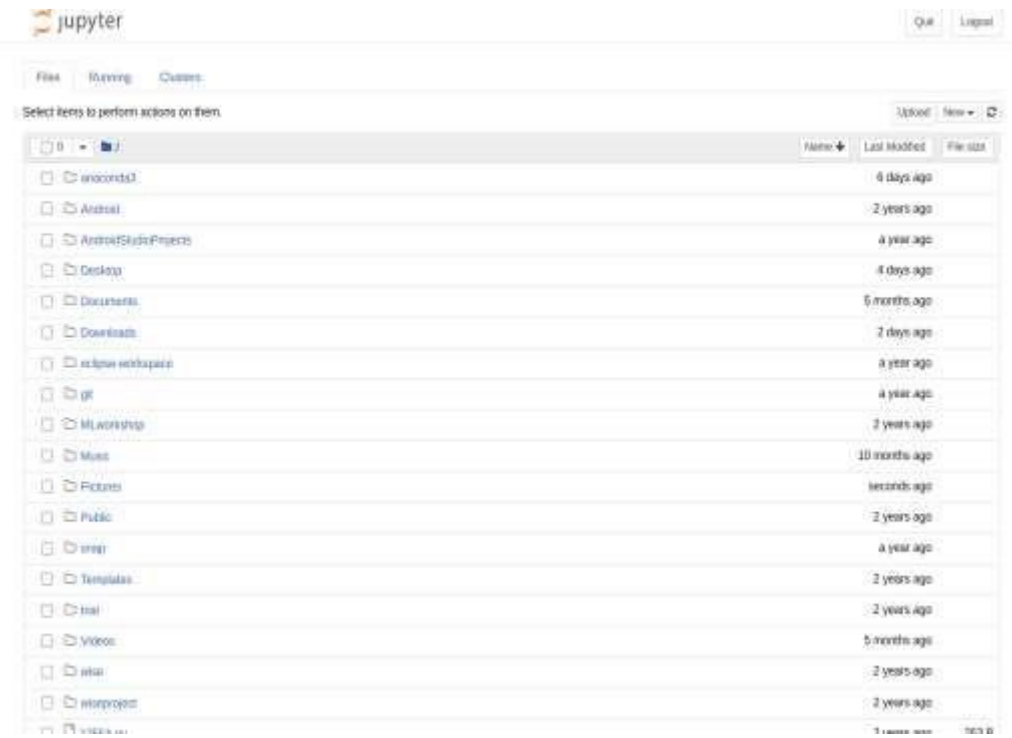


Fig 10.3: Test case showing jupyter notebook has opened

```
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn import preprocessing
import matplotlib.pyplot as plt
```

Fig 10.4: Test case showing there is no error while importing modules and packages

Fig 10.5: If no input is given, "Please enter Id" is displayed

Fig 10.6: If string is given as input, "Please enter a number" is displayed

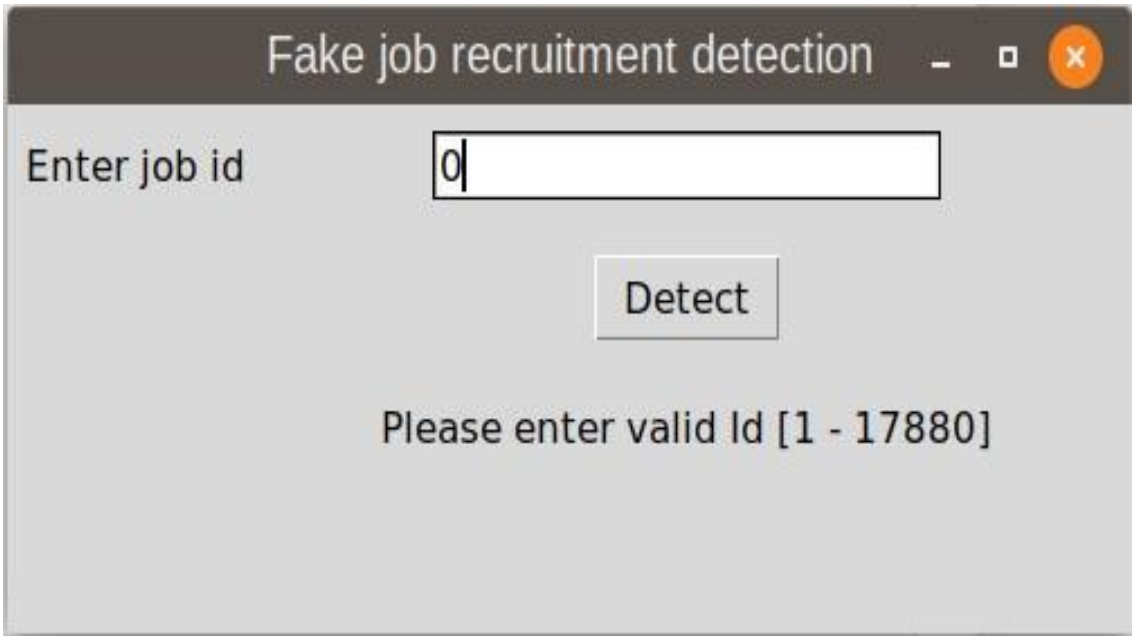


Fig 10.7: If a number out of range is given as input, "Please enter valid Id [1-17880]" is displayed

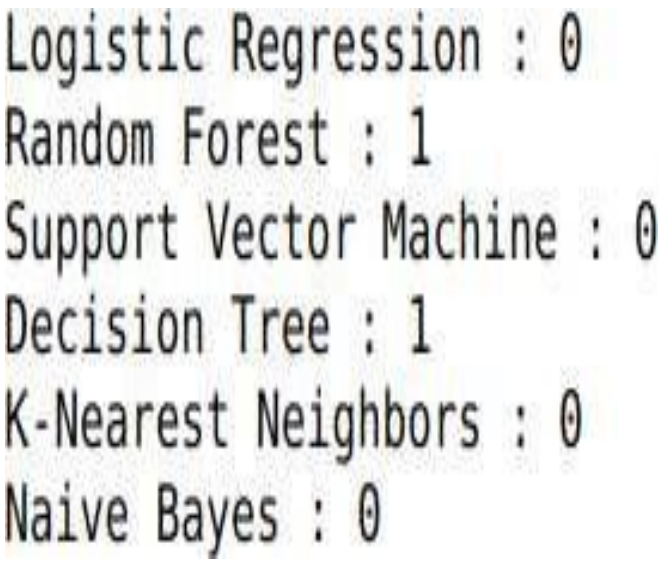


Fig 10.9: Predicted output by all algorithms (0 - real; 1 - fake)

Fig 10.9: Predicted output by all algorithms (0 - real; 1 - fake)

Fig 10.1: Given input '98' output is predicted

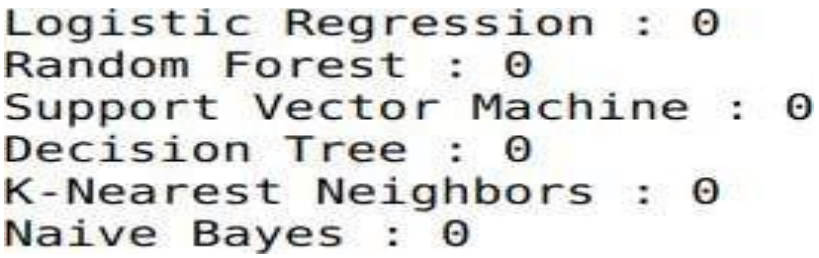


Fig 10.11: Predicted output by all algorithms (0 - real; 1 - fake)

CHAPTER 11

CONCLUSION

Fake Job detection will guide job-seekers to get only legitimate offers from companies. For tackling employment scam detection, several machine learning algorithms are proposed as countermeasures in this paper. Supervised mechanism is used to exemplify the use of several classifiers for Fraudulent Job detection. Experimental results indicate that Random Forest classifier outperforms over its peer classification tool. From the proposed approaches highest achieved accuracy is 98.76% which is much higher than the existing methods.

Future Enhancements:

It is not possible to develop a system that makes all the requirements of the user. User requirements keep changing as the system is being used. Some of the future enhancements that can be done to this system are:

- As the technology emerges, it is possible to upgrade the system and can be adaptable to desired environment.
- Based on the future security issues, security can be improved using emerging technologies like single sign-on.

REFERENCES

[1] B. Alghamdi and F. Alharby, ―An Intelligent Model for Online Recruitment Fraud Detection,‖ J. Inf. Secur., vol. 10, no. 03, pp. 155–176, 2019, doi: 10.4236/jis.2019.103009.

[2] I. Rish, ―An Empirical Study of the Naïve Bayes Classifier An empirical study of the naive Bayes classifier,‖ no. January 2001, pp. 41–46, 2014.

[3] D. E. Walters, ―Bayes's Theorem and the Analysis of Binomial Random Variables,‖ Biometrical J., vol. 30, no. 7, pp. 817–825, 1988, doi: 10.1002/bimj.4710300710.

[4] F. Murtagh, ―Multilayer perceptrons for classification and regression,‖ Neurocomputing, vol. 2, no. 5–6, pp. 183–197, 1991, doi: 10.1016/0925- 2312(91)90023-5.

[5] P. Cunningham and S. J. Delany, ―K -Nearest Neighbour Classifiers,‖ Mult. Classif. Syst., no. May, pp. 1–17, 2007, doi: 10.1016/S0031-3203(00)00099-6.

[6] H. Sharma and S. Kumar, ―A Survey on Decision Tree Algorithms of Classification in Data Mining,‖ Int. J. Sci. Res., vol. 5, no. 4, pp. 2094–2097, 2016, doi: 10.21275/v5i4.nov162954.

[7] E. G. Dada, J. S. Bassi, H. Chiroma, S. M. Abdulhamid, A. O. Adetunmbi, and O. E. Ajibuwa, "Machine learning for email spam filtering: review, approaches and open research problems,‖ Heliyon, vol. 5, no. 6, 2019, doi: 10.1016/j.heliyon.2019.e01802.

[8] L. Breiman, ―ST4_Method_Random_Forest,‖ Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1017/CBO9781107415324.004.

[9] B. Biggio, I. Corona, G. Fumera, G. Giacinto, and F. Roli, ―Bagging classifiers for fighting poisoning attacks in adversarial classification tasks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 6713 LNCS, pp. 350–359, 2011, doi: 10.1007/978-3-642- 21557-5_37.

[10] A. Natekin and A. Knoll, ―Gradient boosting machines, a tutorial,‖ Front. Neurorobot., vol. 7, no. DEC, 2013, doi: 10.3389/fnbot.2013.00021.