A Project Report on

Securing Employment Opportunities: Machine Learning Solutions for Fake Job Detection

Submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

in

Computer Science and Engineering

By

TIPPANI KAVYA SRI	**20A91A0560**
VELECHETY NAGA SAI SRINITHA	**20A91A0562**
BHUMIKA SREE SARELLA	**20A91A0508**
NAMEPALLI V R VISHAL VARMA	**21A91A0501**

Under the Esteemed Supervision of

Mrs. T. Sudha Rani M. Tech.,(Ph.D.)

Associate Professor

Department of Computer Science and Engineering

ADITYA ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to JNTUK Kakinada, Accredited by NBA & NAAC with 'A++' Grade)

Aditya Nagar, ADB Road, Surampalem

2020 – 2024

A Project Report on

Securing Employment Opportunities: Machine Learning Solutions for Fake Job Detection

Submitted in partial fulfillment of the requirements for the award of the Degree of

Bachelor of Technology

in

Computer Science and Engineering

By

TIPPANI KAVYA SRI	**20A91A0560**
VELECHETY NAGA SAI SRINITHA	**20A91A0562**
BHUMIKA SREE SARELLA	**20A91A0508**
NAMEPALLI V R VISHAL VARMA	**21A91A0501**

Under the Esteemed Supervision of

Mrs.T. Sudha Rani M. Tech.,(Ph.D.)

Associate Professor

Department of Computer Science and Engineering

ADITYA ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to JNTUK Kakinada, Accredited by NBA & NAAC with 'A++' Grade)

Aditya Nagar, ADB Road, Surampalem

2020 – 2024

ADITYA ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Affiliated to JNTUK Kakinada, Accredited by NBA & NAAC with 'A++' Grade)

Aditya Nagar, ADB Road, Surampalem

2020 – 2024

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project work entitled "**Securing EmploymentOpportunities: Machine Learning Solutions for Fake Job Detection"** is being submitted by

TIPPANI KAVYA SRI	**20A91A0560**
VELECHETY NAGA SAI SRINITHA	**20A91A0562**
BHUMIKA SREE SARELLA	**20A91A0508**
NAMEPALLI V R VISHAL VARMA	**21A91A0501**

in partial fulfillment of the requirements for the award of degree of **B.Tech**in Computer Science and Engineering from **Jawaharlal Nehru Technological University Kakinada** is a record of bonafide work carried out by them at **Aditya Engineering College**

The results embodied in this Project report have not been submitted to any other University or Institute for the award of any degree or diploma.

PROJECT GUIDE	**HEAD OF THE DEPARTMENT**
T. Sudha Rani	**Dr.A.Vanathi**
M. Tech.,(Ph.D.)	**M.E.,Ph.D**

EXTERNAL EXAMINER

DECLARATION

We hereby declare that the project entitled **"SECURING EMPLOYMENT OPPORTUNITIES: MACHINE LEARNING SOLUTIONS FOR FAKE JOB DETECTION"** is a genuine project. This work has been submitted to the **ADITYA ENGINEERING COLLEGE,** Surampalem, permanently affiliated to **JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, KAKINADA** in partial fulfillment of the **B.Tech**degree**.** We further declare that this project work has not been submitted in full or part of the award for any degree of this or any other educational institutions.

By

TIPPANI KAVYA SRI	**20A91A0560**
VELECHETY NAGA SAI SRINITHA	**20A91A0562**
BHUMIKA SREE SARELLA	**20A91A0508**
NAMEPALLI V R VISHAL VARMA	**21A91A0501**

ACKNOWLEDGEMENT

It is with immense pleasure that we would like to express our indebted gratitude to our project guide **SUDHA RANI, M. Tech.,(Ph.D.),Associate Professor , Department of CSE** who has guided us a lot and encouraged us in every step of the project work, her valuable moral support and guidance throughout the project helped us to a greater extent.

Our sincere thanks to project coordinator **Dr.V.Ravi Kishor M.Tech., Ph.D, Associate Professor, Department of CSE** for providing a great support and suggestions during our project work.

Our deepest thanks to our HOD **Dr. A.Vanathi, M.E.,Ph.D, Associate Professor** for inspiring us all the way and for arranging all the facilities and resources needed for our project.

We wish to thank **Dr. S. Rama Sree, Professor** in CSE and Dean (Academics) for her support and suggestions during our project work.

We owe our sincere gratitude to **Dr. M.Sreenivasa Reddy, Principal** for providing a great support and for giving us the opportunity of doing the project.

We are thankful to our **College Management** for providing all the facilities in time to us for completion of our project.

Not to forget, **Faculty, Lab Technicians, Non-teaching staff and our friends** who have directly or indirectly helped and supported us in completing our project in time.

VISION & MISSION OF THE INSTITUTE

Vision:

To emerge as a premier institute for quality technical education and innovation.

Mission:

M1: Provide learner centric technical education towards academic excellence

M2: Train on technology through collaborations

M3: Promote innovative research & development

M4: Involve industry institute interaction for societal needs

PRINCIPAL
PRINCIPAL
ADITYA ENGINEERING COLLEGE
SURAMPALEM - 533 437

VISION & MISSION OF THE DEPARTMENT

VISION:

To emerge as a competent Centre of excellence in the field of Computer Science and Engineering for industry and societal needs.

MISSION:

- Impart quality and value based education.
- Inculcate the inter personal skills and professional ethics.
- Enable research through state-of-the-art infrastructure.
- Collaborate with industries, government and professional societies.

Head of the Department

Head of the Department
Department of CSE
ADITYA ENGINEERING COLLEGE (A)

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

Graduates of the Program will

- **PEO 1:** Adopt new technologies and provide innovative solutions.
- **PEO 2:** Be employable, become an entrepreneur or researcher for a successful career.
- **PEO 3:** Demonstrate interpersonal, multi-disciplinary skills and professional ethics to serve society.

Head of the Department

Head of the Department
Department of CSE
ADITYA ENGINEERING COLLEGE (A)

PROGRAM OUTCOMES (POs)

After successful completion of the program, the graduates will be able to

PO 1 **Engineering Knowledge:** Apply knowledge of mathematics, science, engineering fundamentals and an engineering specialization to the solution of complex engineering problems.

PO 2 **Problem Analysis:** Identify, formulate, research literature and analyze complex engineering problems, reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.

PO 3 **Design/Development of Solutions:** Design solutions for complex engineering problems and design systems, components or processes that meet specified needs with appropriate consideration for public health and safety, cultural, societal, and environmental considerations.

PO 4 **Conduct Investigations of Complex Problems:** Conduct investigations of complex problems using research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of information to provide validconclusions.

PO 5 **Modern Tool Usage:** Create, select and apply appropriate techniques, resources, and modern engineering and IT tools, including prediction and modelling, to complex engineering activities, with an understanding of the limitations.

PO 6 **The Engineer and Society:** Apply reasoning informed by contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to professional engineering practice.

PO 7 **Environment and Sustainability:** Understand the impact of professional engineering solutions in societal and environmental contexts and demonstrate knowledge of, and need for sustainable development.

PO 8 **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of engineering practice.

PO 9 **Individual and Teamwork:** Function effectively as an individual, and as a member or leader in diverse teams and in multidisciplinary settings.

PO 10 **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO 11 **Project Management and Finance:** Demonstrate knowledge and understanding of engineering management principles and apply these to one's own work, as a member and leader in a team and to manage projects in multidisciplinary environments.

PO 12 **Life-Long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Head of the Department
Head of the Department
Department of CSE
ADITYA ENGINEERING COLLEGE (A

PROGRAM SPECIFIC OUTCOMES (PSOs)

After successful completion of the program, the graduates will be able to

PSO 1: Develop efficient solutions to real world problems using the domains of Algorithms, Networks, database management and latest programming tools and techniques.

PSO 2: Provide data centric business solutions through emerging areas like IoT, AI , data analytics and Block Chain technologies.

Head of the Department
Head of the Department
Department of CSE
ADITYA ENGINEERING COLLEGE (A)

ABSTRACT

In today's digital world, technology and social media have made job postings more common. Unfortunately, this increase in job ads has also brought about more fake listings, which pose a big problem for both job seekers and employers. Being able to spot these fake job postings is crucial to protect people from scams and ensure that the hiring process is fair. However, telling apart real job ads from the fraudulent ones is tough, just like solving other prediction problems.To address this challenge, our project suggests using different machine learning techniques to classify job postings. We want to create a strong system that can accurately flag fraudulent job ads among the many online listings. Our experiments use a dataset called EMSCAD, which contains a variety of real job postings. Through careful testing and analysis, we aim to see how well these machine learning methods can separate genuine job ads from the fake ones.Our early findings are encouraging, especially with deep neural network classification. By using a deep neural network with three dense layers, we achieved impressive results, accurately identifying about 98% of fake job postings. These results show the potential of advanced machine learning tools in tackling the problem of fake job listings. It highlights the need for ongoing research and innovation to ensure online job searches are safe and reliable.

Moreover, our study emphasizes the need to continuously update and improve our methods to stay ahead of scammers who create fraudulent job postings. As these individuals adapt and develop new techniques, our classification models must remain adaptable and resilient. Additionally, while achieving high accuracy in detecting fake job ads is important, we also need to consider the possibility of incorrectly flagging legitimate postings as fraudulent and vice versa. Finding a balance between precision and recall ensures that our system effectively identifies scams while minimizing the risk of overlooking genuine opportunities.Furthermore, our project highlights the broader impact of using machine learning to address online fraud. By showing how these techniques can successfully identify fake job postings, we open up possibilities for their application in other areas where deception is a concern. This demonstrates the versatility and potential benefits of using advanced technology to protect individuals and businesses online. Ultimately, our efforts contribute to building trust and confidence in online job searches, making the recruitment process safer and more reliable for job seekers worldwide.

TABLE OF CONTENTS

LIST OF FIGURES

1. INTRODUCTION

1.1 INTRODUCTION TO PROJECT

In today's dynamic employment landscape, the fusion of industry and technology has ushered in an era of unprecedented opportunities for job seekers. The advent of online job advertisements has significantly expanded the scope of available options, offering tailored opportunities that align with individual qualifications, schedules, and experiences. This transition towards digital platforms for recruitment has been driven by the pervasive use of the internet and social media, providing job seekers with unparalleled convenience in navigating and applying for positions.

However, alongside these advancements comes the challenge of fraudulent job postings, which have proliferated in tandem with the rise of online recruitment platforms. The prevalence of fake job listings has raised concerns among job seekers, who are increasingly wary of engaging with new opportunities to protect their personal and professional information. Legitimate job postings shared through social and electronic media channels often face credibility issues, overshadowed by the prevalence of deceptive listings. Restoring trust in job advertisements and ensuring the integrity of the recruitment process are paramount objectives in addressing this issue.

To mitigate the impact of fraudulent job postings, there is a pressing need for technological innovations aimed at effectively identifying and filtering out deceptive listings. Automated systems equipped with advanced algorithms can play a pivotal role in detecting false job postings, thereby providing job seekers with a safer and more efficient experience. By implementing such solutions, organizations can minimize the time and effort wasted by job seekers on fraudulent listings, while also simplifying the recruitment process for human resource management professionals.

In addition to technological solutions, collaborative efforts between stakeholders are essential in combating the proliferation of fake job postings. Job boards, recruitment agencies, and regulatory bodies must work together to implement robust verification processes and stringent measures for posting job advertisements. Moreover, educating both job seekers and employers about the signs of fraudulent job postings and best practices for job searching is

crucial in empowering individuals to make informed decisions and avoid falling victim to scams.

1.2 EXISTING SYSTEM AND DISADVANTAGES

Tingminet al. provide a survey of new methods and techniques to identify Twitter spam detection. The above survey presents a comparative study of the current approaches.On the other hand, S. J. Somanet. al. conducted a survey on different behaviors exhibited by spammers on Twitter social network. The study also provides a literature review that recognizes the existence of spammers on Twitter social network. Despite all the existing studies, there is still a gap in the existing literature. Therefore, to bridge the gap, we review state-of-the-art in the spammer detection and fake user identification on Twitter.

In the current model aimed at detecting fraudulent activity in an online recruitment system, researchers utilized the EMSCAD dataset and employed data mining algorithms. Their approach involved three main steps: data pre-processing, feature selection, and fraud detection using a classifier.During the pre-processing phase, efforts were made to eliminate noise and HTML tags from the dataset while preserving the general text pattern. This step was crucial in ensuring that the data remained clean and suitable for subsequent analysis. Following pre-processing, a feature selection technique was applied to streamline the dataset by effectively reducing the number of attributes.

This step aimed to enhance the efficiency and effectiveness of the analysis process by focusing on the most relevant features for fraud detection.Finally, the researchers utilized a classifier to detect instances of fraudulent activity within the refined dataset. By leveraging the selected features and employing appropriate classification techniques, the model was able to accurately identify potential instances of fraud within the online recruitment system.

DISADVANTAGES OF EXISTING SYSTEM:

Limited Data Availability: The existing system often relies on datasets from platforms like Facebook, which may have limited access to detailed information due to privacy concerns. This limitation restricts the amount of data available for analysis and can hinder the effectiveness of the detection algorithms.

Low Accuracy: Traditional methods based on data mining algorithms tend to suffer from lower accuracy rates. This can lead to false positives or false negatives in identifying fake job postings, potentially causing confusion and distrust among job seekers.

High Complexity: The complexity of data mining algorithms can lead a significant challenges, both in terms of the implementation and computational resources required. This complexity may hinder scalability and efficiency, making it difficult to process large volumes of job postings in real-time.

1.3 PROPOSED SYSTEM AND ADVANTAGES

In the proposed system, we have identified individuals engaging in fraudulent activities by posing as legitimate job advertisers. These scammers exploit data pertaining to reputable companies to fabricate job advertisements with malicious intent. Our experimentation involved the utilization of the EMSCAD dataset, where we applied various classification algorithms, including the Naive Bayes classifier and the Random Forest classifier. Notably, the Random Forest Classifier demonstrated superior performance, achieving a classification accuracy of 94.5%. Conversely, the Naive Bayes classifier yielded unsatisfactory results when applied to the dataset. Additionally, we observed that the logistic regression classifier performed effectively when the dataset was appropriately balanced.

ADVATAGES OF PROPOSED SYSTEM :

Enhanced Security: By leveraging machine learning algorithms for detecting fake job postings, the proposed system offers improved security for job seekers browsing online platforms. This reduces the risk of falling victim to scams or fraudulent activities, thereby safeguarding users' personal and financial information.

Mitigation of Social Networking Issues: The proposed system addresses various issues associated with social networking platforms, such as privacy concerns, online bullying, misuse, and trolling, which are often exacerbated by fake job postings. By effectively identifying and filtering out fake job posts, the system helps mitigate these issues, fostering online environment for safety.

In the proposed system, we identified job scammers as fake online job advertiser. They found statistics about many real and renowned companies and enterprises who produced fake job advertisements or vacancy posts with ill-motive.we experimented on EMSCAD dataset using several classification algorithms like naive bayes classifier, random forest

classifier,etc. Random Forest Classifier showed the best performance on the dataset with 94.5% classification accuracy. we found navy bayes performing very poor on the dataset. Logistic regression classifier performed well when they balanced the dataset and experimented on that.

1.4 OBJECTIVES OF THE PROJECT

The objectives of a "Fake Job Post Prediction using Machine Learning Techniques" project typically include:

- **Identification of Fraudulent Job Postings:** The primary objective is to develop a model that can accurately identify fake or fraudulent job postings.
- **Protecting Job Seekers:** By detecting fake job postings early, the project aims to protect job seekers from potential scams, fraud, or exploitation
- **Enhancing Trust in Online Job Platforms:** By implementing effective fake job post prediction models, online job platforms can enhance trust among their users
- **Reducing Economic Losses:** Fake job postings can lead to economic losses for both job seekers and businesses.
- **Improving Platform Reputation:** Online job platforms rely on their reputation to attract both job seekers and employers.
- **Optimizing Resource Allocation:** By automating the process of identifying fake job postings, the project can help online job platforms allocate their resources more effectively.
- **Continuous Improvement:** The project may also focus on developing models that can adapt to new trends and techniques used by fraudsters to create fake job postings.

1.5 ORGANISATION OF THE PROJECT

Based on the provided information, the organization of the "Fake Job Post Prediction using Machine Learning Techniques" project could be structured as follows:

- **Introduction:**

Introduction to the project: Provides an overview of the project's context, emphasizing the challenges posed by fraudulent job postings in online recruitment platforms and the need for technological solutions.

- **Existing System and Disadvantages:** Reviews previous research efforts and identifies limitations in the current approach to detecting fake job postings.

- **Proposed System and Advantages:** Introduces the proposed solution, highlighting its advantages over existing systems.

Literature Review:

Survey of Related Work: Summarizes existing research and methodologies related to fake job post detection, including approaches used in social media spam detection and previous studies on identifying fraudulent activities in online recruitment systems.

2. REQUIREMENT ANALYSIS

2.1 INTRODUCTION TO REQUIREMENT ANALYSIS

In any software development endeavor, understanding the software and hardware requirements is paramount for successful project execution. These requirements serve as the foundation upon which the entire development process is built, ensuring that the final product meets the needs and expectations of stakeholders while operating within the constraints of the available infrastructure.

The hardware requirements outline the necessary physical components and specifications needed to support the software system effectively. For instance, specifying an i3 processor, 500GB hard disk, 4GB RAM, and essential input devices like a keyboard and mouse ensures that the system has the computational power and storage capacity to run the software smoothly. These hardware specifications are carefully chosen to accommodate the expected workload and provide users with a seamless experience.On the other hand, the software requirements detail the necessary software components and technologies required for developing and running the software system. In this case, the software requirements include the operating system (Windows 10), coding language (Python), front-end development (Python), back-end framework (Django-ORM), designing tools (HTML, CSS, and JavaScript), and database (MySQL running on a WAMP server). These software components are selected based on their compatibility, functionality, and suitability for the intended purpose of the software system.

2.2HARDWARE REQUIREMENTS:

- **System**: i3 processor
- **Hard Disk**: 500GB
- **Input Devices**: keyboard, mouse
- **RAM**: 4GB

2.2.1SOFTWARE REQUIREMENTS:

- **Operating system**: Windows 10
- **Coding Language**: Python
- **Front-end:** Python

- **Back-end:** Django-ORM
- **Designing:** HTML, CSS and JavaScript
- **Data Base**: MYSQL (WAMP Server)

2.3DATASETS

EMSCAD dataset is used which contains 17000+

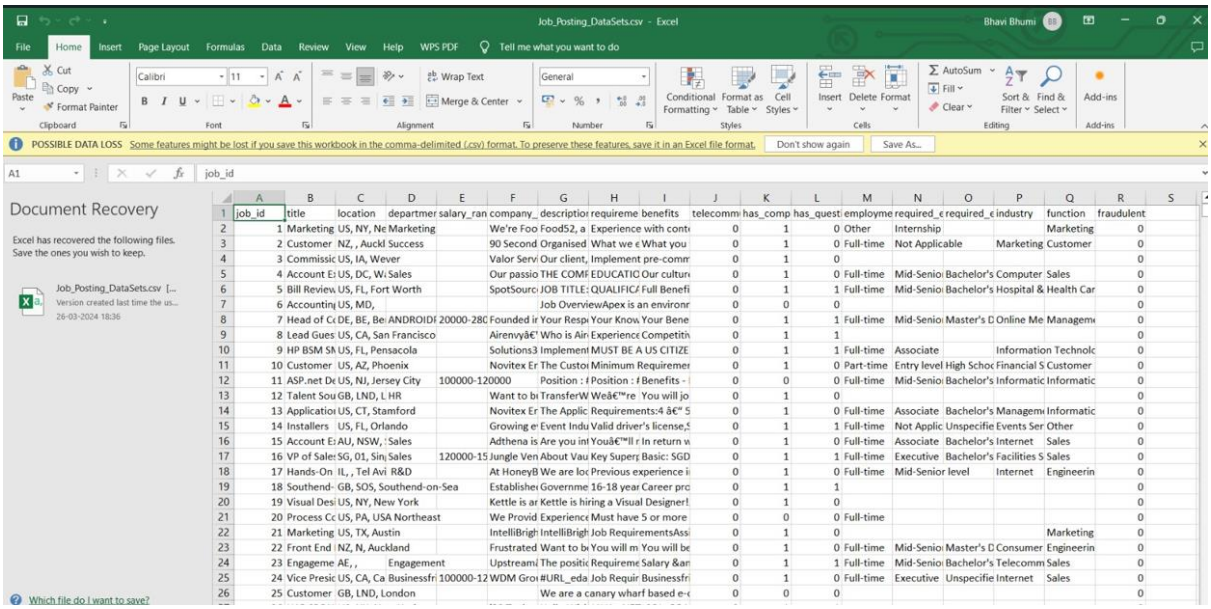


Fig 2.3.1 Datasets

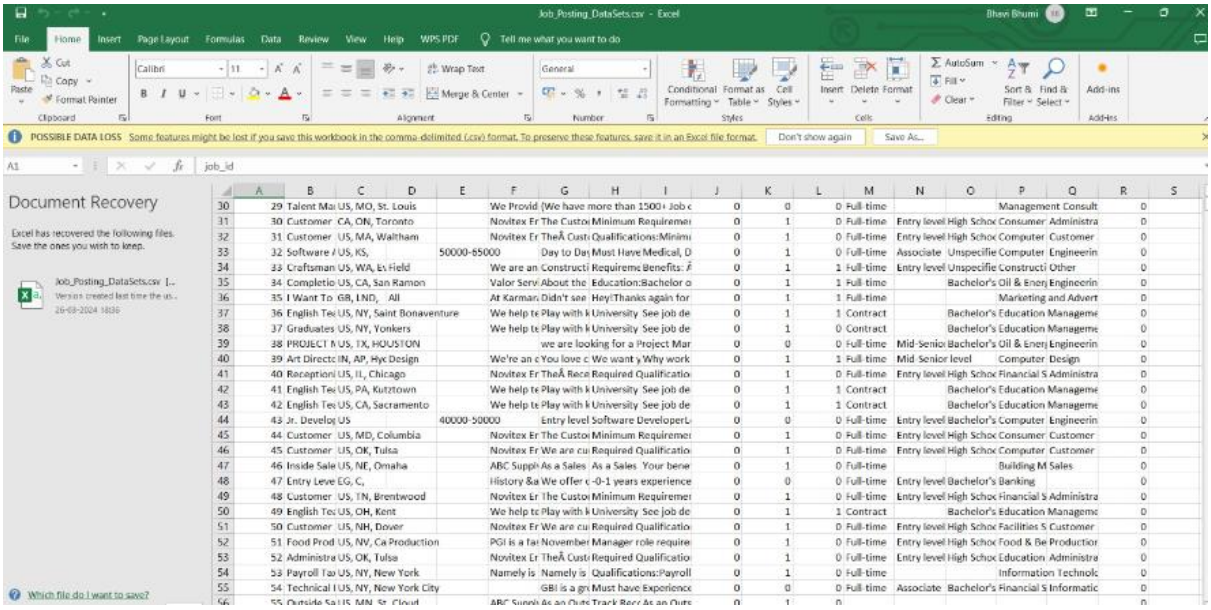


Fig 2.3.2 Datasets

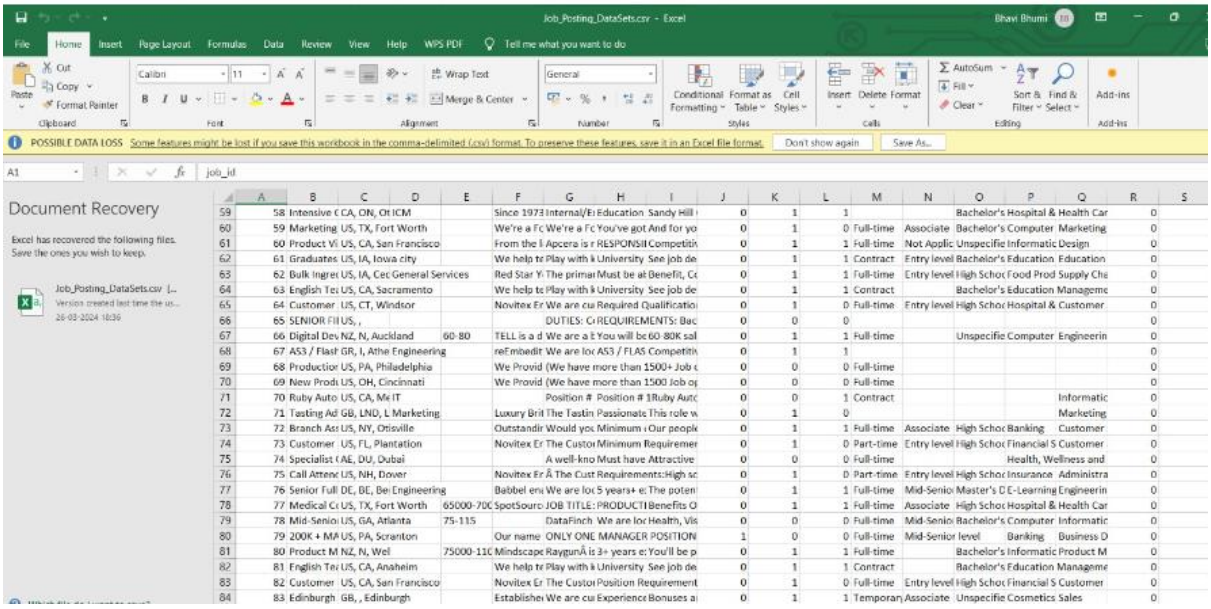


Fig 2.3.3 Datasets

2.4SOFTWARE REQUIREMENTS SPECIFICATIONS

Operating System:

The software shall be compatible with Windows 10 operating system.

Coding Language:

The primary coding language for the software development shall be Python.

Front-end:

The front-end of the software application shall be developed using Python.The graphical user interface (GUI) elements shall be designed and implemented using appropriate Python libraries and frameworks.

Back-end:

The back-end of the software application shall be implemented using Django-ORM framework.Django-ORM shall be utilized for handling database interactions, authentication, and business logic implementation.

Designing:

The user interface shall be designed using HTML, CSS, and JavaScript.HTML shall be used for structuring the content of web pages.CSS shall be used for styling the visual presentation of web pages.JavaScript shall be used for implementing interactive features and enhancing user experience.

Database:

The software shall utilize MySQL as the relational database management system (RDBMS).MySQL shall be installed and configured to run on a WAMP server environment.

Overall, the software requirements specify the necessary technologies, frameworks, and tools that will be utilized in the development of the software application. These specifications provide a clear guideline for the development team, ensuring that the software is developed in accordance with the desired functionality, compatibility, and performance criteria outlined by the stakeholders.

2.5SOCIAL NEEDS

Employment Security: Machine learning techniques are utilized to predict fake job postings, safeguarding job seekers from fraudulent schemes and ensuring confidence in genuine job opportunities.

Consumer Protection: By detecting fake job posts, the project protects job seekers from financial loss, identity theft, and exploitation, promoting safer online job searches.

Digital Literacy and Awareness: Public access to predictive tools raises awareness about fake job risks, fostering digital literacy and empowering informed decision-making in the digital job market.

Trust in Online Platforms: Implementing machine learning for fake job prediction enhances the credibility and reliability of online job platforms, improving user trust and engagement.

Social Inclusion and Equity: Predicting fake job posts promotes fairness and equal access to employment opportunities, contributing to a more inclusive and equitable job market for all individuals.

3. LITERATURE SURVEY

3.1 RELATED WORKS

[3.1.1] Spam detection of Twitter traffic: A framework based on random forests and non-uniform feature sampling

AUTHORS: C. Meda, E. Ragusa, C. Gianoglio, R. Zunino, A. Ottaviano, E. Scillia, and R. Surlinelli

Law Enforcement Agencies play a vital role in analyzing open data, particularly on social networks like Twitter, to monitor events and profile accounts. However, amidst the vast number of users, there are individuals who misuse microblogs for harassment or spreading harmful content. Identifying and classifying users, particularly spammers, is essential to filter out uninformative content. This study presents a framework that utilizes a non-uniform feature sampling approach within a gray-box Machine Learning System. It employs a variant of the Random Forests Algorithm to detect spammers within Twitter traffic. The effectiveness of this approach is demonstrated through experiments conducted on a popular Twitter dataset and a new dataset consisting of users labeled as either spammers or legitimate users, described by 54 features. The experimental results highlight the efficacy of the enriched feature sampling method.

[3.1.2] Automatically identifying fake news in popular Twitter threads

AUTHORS: C. Buntain and J. Golbeck 1

The importance of addressing information quality in social media is increasingly recognized. However, the sheer volume of data online makes it challenging for experts to effectively identify and rectify inaccurate content, commonly referred to as "fake news." This study presents a novel approach for automatically detecting fake news on Twitter by leveraging two credibility-focused Twitter datasets: CREDBANK, a dataset where accuracy assessments are crowdsourced for Twitter events, and PHEME, a dataset containing potential rumors on Twitter along with assessments of their accuracy by journalists. The proposed method is applied to Twitter content from BuzzFeed's fake news dataset, demonstrating that models trained using assessments from crowdsourced workers outperform those relying solely on assessments by journalists or models trained on a combined dataset of both crowdsourced

workers and journalists. Additionally, all three datasets have been standardized into a uniform format and made publicly accessible. Through feature analysis, the study identifies the most predictive features for both crowdsourced and journalistic accuracy assessments, with findings consistent with prior research. The paper concludes with a discussion on the distinction between accuracy and credibility, highlighting why models based on assessments from non-experts are more effective than those relying solely on assessments from journalists for detecting fake news on Twitter.

[3.1.3] Statistical features-based real-time detection of drifted Twitter spam

AUTHORS: C. Chen, Y. Wang, J. Zhang, Y. Xiang, W. Zhou, and G. Min

Twitter spam has emerged as a significant issue in recent times, prompting researchers to explore machine learning techniques for its detection. These methods typically rely on statistical features extracted from tweets. However, our examination of labeled tweet datasets reveals a phenomenon known as "Twitter Spam Drift," wherein the statistical properties of spam tweets vary over time, leading to decreased classifier performance. To address this challenge, we conduct an in-depth analysis of statistical features extracted from one million spam tweets and one million non-spam tweets. Subsequently, we propose a novel scheme called Lfun, which identifies "changed" spam tweets from unlabeled data and integrates them into the classifier's training process. Through a series of experiments, we demonstrate the effectiveness of the Lfun scheme in improving spam detection accuracy under real-world conditions.

[3.1.4] A model-based approach for identifying spammers in social networks

AUTHORS: F. Fathaliani and M. Bouguessa

In this paper, we view the task of identifying spammers in social networks from a mixture modeling perspective, based on which we devise a principled unsupervised approach to detect spammers. In our approach, we first represent each user of the social network with a feature vector that reflects its behaviour and interactions with 1 other participants. Next, based on the estimated users feature vectors, we propose a statistical framework that uses the Dirichlet distribution in order to identify spammers. The proposed approach is able to automatically discriminate between spammers and legitimate users, while existing unsupervised approaches require human intervention in order to set informal threshold parameters to detect spammers. Furthermore, our approach is general in the sense that it can

be applied to different online social sites. To demonstrate the suitability of the proposed method, we conducted experiments on real data extracted from Instagram and Twitter.

3.2 SURVEY TABLE

S.No	Reference Paper Title	Purpose	Drawbacks
1	S. Vidros, C. Kolias, G. Kambourakis and L. Akoglu, "Automatic Detection of Online Recruitment Frauds: Characteristics Methods and a Public Dataset"	For fake job prediction	The paper acknowledges that the dataset used for experimentation is small and curated.
2	B. Alghamdi and F. Alharby, "An Intelligent Model for Online Recruitment Fraud Detection", *Journal of Information Security*, vol. 10,	For fake job prediction	The paper acknowledges that the dataset used for experimentation is small and curated.
3	T. Van Huynh, V. D. Nguyen, K. Van Nguyen, N. L.-T. Nguyen, and A.G.-T. Nguyen, "Hate Speech Detection on Vietnamese Social Media Text using the Bi-GRU-LSTM-CNN Model	For fake job prediction	The paper acknowledges that the dataset used for experimentation is small and curated.

Fig:3.2 Survey Table

4. MODULES

4.1 INTRODUCTION TO MODULE

- Data Collection
- Pre-Processing
- Train and Test
- Machine Learning Technique
- Detection of Fake Job post

MODULE DESCRIPTION

4.1.1 Data Collection

The dataset was collected from online. The dataset contains 17000+ different samples of job posts. It contains different parameters of JOB ID, DESCRIPTION

4.1.2 Pre-Processing

We convert data in to scalar format and then create new features which are passed to algorithm and features are saved in x and labels in y.

4. Train and Test

we will split the dataset into training dataset and test dataset. We will use 70% of our data to train and the rest 30% to test. To do this, we will create a split parameter which will divide the data frame in a 70-30 ratio.

4.1.4 Machine Learning Technique

After splitting the dataset into training and test dataset, we will instantiate Random Forest Classifier and KNN classifier fit the train data by using ‘fit’ function. Then we will store as model.

4.1.5 Fake Job Post Prediction

In this step details are fed as input in the form of csv of various profiles and prediction is performed. New input csv file with different job profile details is

given as input and prediction is performed and details are stored in new csv with

prediction results.

In this project three modules are used mostly. The following are the modules used

Service Provider

- In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as Train and Test Data Sets, View Trained and Tested Accuracy in Bar Chart, View Trained and Tested Accuracy Results, Predict Job Post Type Details, Find Job Post Type Prediction Ratio, Download Trained Data Sets, View Job Post Type Prediction Ratio Results, View All Remote Users**.**

View and Authorize Users

- In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

Remote User

- In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like REGISTER AND LOGIN, POST JOB POST DATA SETS, PREDICT JOB POST PREDICTION, VIEW YOUR PROFILE.

4.2 MODULE IMPLEMENTATION

The Architecture diagram tells us the basic idealogy of the project.

so we have two main domains namely service provider and remote user which have there own duties. The user is the one that register or login to the website and post job posts and data sets, and he/she can click and can predict the job the fake or real and can also the view their profile that they have created during registration.

The next domain is service provider. He can login using the admin ceredentials, he can train and test datasets, he also can view the trained and tested accuracy in bar chart,

accuracy results, predict job post type details, find job post type prediction ratio, download trained datasets, view job post type prediction ratio results, and view all remote users actions in the website.

We will have a webserver that accepts all the information and stores the data in the database it acts like a medium between the web database and service provider and remote user. And we also have a wed database that process all the queries of the user and gives the resultant output. And it all stores all the inputs given by the user.

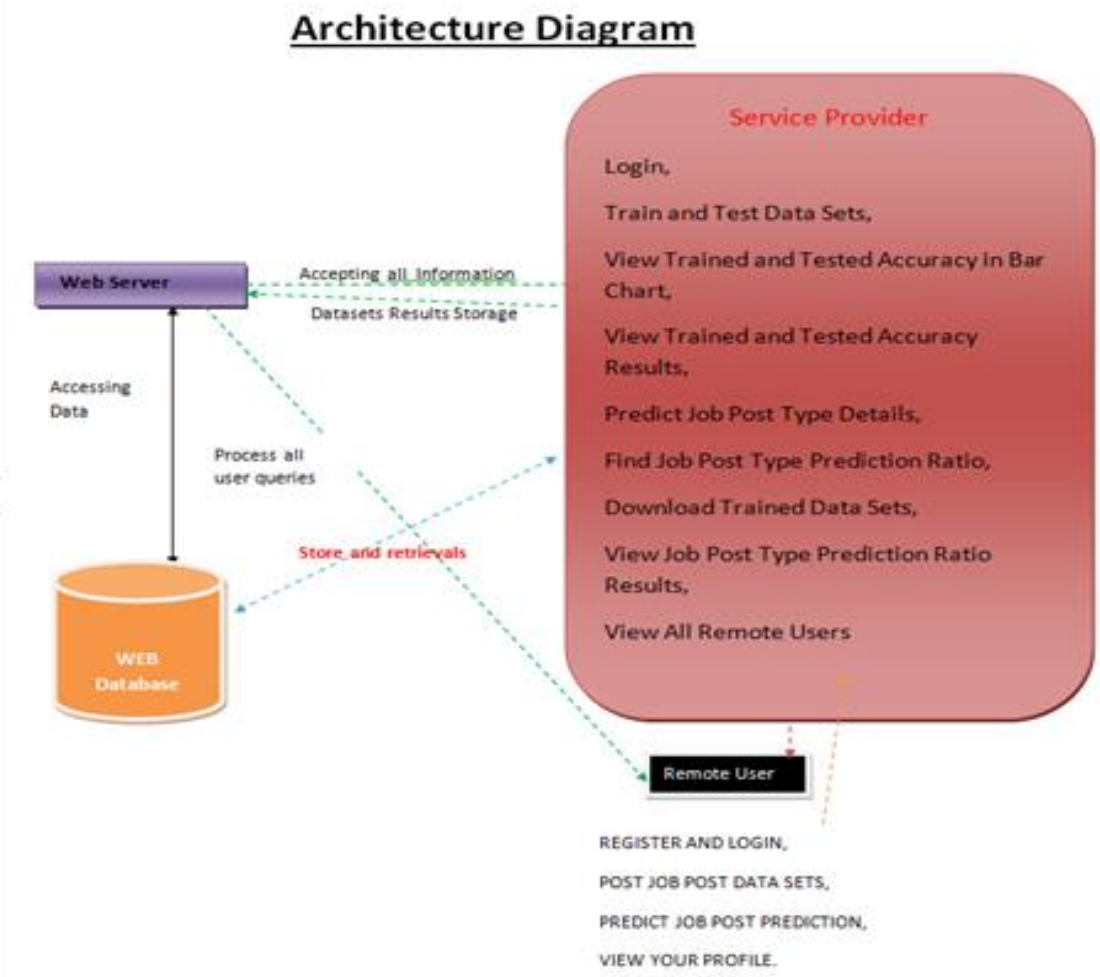


Fig: 4.2 Architecture Diagram

ALGORITHMS

- Random Forest Algorithm
- KNN
- Naïve Bayes
- SVM

4.3.1 RANDOM FOREST ALGORITHM

Definition: The random forest algorithm is a powerful supervised learning technique used for both classification and regression tasks. It operates by constructing a multitude of decision trees during the training phase and then aggregating their predictions to make a final decision.

Working Principle: Random forest works by creating an ensemble of decision trees, where each tree is trained on a random subset of the training data and a random subset of features. During prediction, the algorithm aggregates the predictions of all the decision trees to arrive at a final output. For classification tasks, the mode (most frequent class) of the individual tree predictions is typically chosen, while for regression tasks, the mean or median of the predictions is used.

Key Features:

1.Ensemble Learning: Random forest leverages the power of ensemble learning by combining multiple decision trees to improve predictive accuracy and robustness.

2. Random Subsampling: The algorithm uses random subsampling of both the training data and features to create diverse and uncorrelated decision trees, which helps prevent overfitting.

3. Feature Importance: Random forest provides a measure of feature importance, allowing users to understand which features contribute the most to the predictive performance of the model.

4. Flexibility: Random forest is versatile and can handle a wide range of data types and structures, making it suitable for various real-world applications.

4.3.2 KNN

Definition: K-Nearest Neighbors (KNN) is a simple yet effective supervised learning algorithm used for both classification and regression tasks. It operates on the principle that similar data points tend to belong to the same class or have similar numerical values.

Working Principle: In KNN, when presented with a new data point for prediction, the algorithm looks for the K nearest data points in the training set based on a chosen distance metric, such as Euclidean distance. These nearest neighbors are determined by measuring the distance between the new data point and all other data points in the training set.

For classification tasks, KNN assigns the majority class among the K nearest neighbors to the new data point. In other words, the class label of the new data point is determined by the most common class among its K nearest neighbors. For regression tasks, KNN computes the average value among the target values of the K nearest neighbors and assigns this average value as the prediction for the new data point.

Key Features:

1. Non-parametric: KNN is a non-parametric algorithm, meaning it does not make any assumptions about the underlying data distributions. This makes it particularly useful for data with complex or unknown distributions.

2. Distance Metric: KNN typically uses distance metrics such as Euclidean distance or Manhattan distance to measure the similarity between data points. The choice of distance metric can impact the performance of the algorithm.

3. Flexibility: KNN is versatile and can be used for both classification and regression tasks, making it suitable for a wide range of applications.

4. Intuitive Concept: The underlying principle of KNN, which is based on the idea of similarity among data points, is intuitive and easy to understand, making it accessible to users without a deep understanding of complex mathematical concepts.

4.3.3 NAÏVE BAYES

Definition: Naive Bayes is a probabilistic supervised learning algorithm used for classification tasks. It is based on Bayes' theorem and assumes that features are conditionally independent given the class label.

Working Principle: In Naive Bayes classification, the algorithm calculates the probability of each class given a set of features using Bayes' theorem. It assumes that each feature contributes independently to the probability of the class label. Despite its simplifying

assumption of feature independence, Naive Bayes often performs well in practice and is particularly effective for text classification tasks.

Key Features:

1. Probabilistic Approach: Naive Bayes calculates probabilities to determine the likelihood of each class given the observed features. It computes the posterior probability of each class based on the prior probability of the class and the likelihood of the features given the class.

2. Independence Assumption: The "naive" assumption of feature independence simplifies the computation of probabilities and makes Naive Bayes computationally efficient, especially for high-dimensional data.

3. Versatility: Naive Bayes can handle both binary and categorical features, as well as continuous features when combined with appropriate probability density estimation techniques.

4. Fast Training and Prediction: Naive Bayes models are relatively simple and have fast training and prediction times, making them suitable for large datasets and real-time applications.

4.3.4 SUPPORT VECTOR MACHINE

Definition: Support Vector Machine (SVM) is a supervised learning algorithm used for both classification and regression tasks. It aims to find the hyperplane that best separates the data points into different classes while maximizing the margin between the classes.

Working Principle: In SVM classification, the algorithm seeks to find the hyperplane that maximizes the margin, which is the distance between the hyperplane and the nearest data points from each class, known as support vectors. SVM can handle both linearly separable and non-linearly separable datasets through the use of kernel functions, which transform the original feature space into a higher-dimensional space where the data points become linearly separable.

Key Features:

1. Maximum Margin: SVM aims to find the hyperplane that maximizes the margin between the classes, resulting in a robust decision boundary that generalizes well to unseen data.

2. Kernel Trick: SVM utilizes kernel functions to map the original feature space into a higher-dimensional space, allowing it to handle non-linearly separable data and discover complex decision boundaries.

3. Margin-based Loss Function: SVM uses a hinge loss function, which penalizes misclassifications based on their distance from the decision boundary. This margin-based loss function encourages the model to prioritize correctly classifying data points near the decision boundary.

4. Effective in High-Dimensional Spaces: SVM is effective in high-dimensional feature spaces, making it suitable for tasks with a large number of features, such as text classification and image recognition.

5. SYSTEM DESIGN

5.1 INTRODUCTION TO SYSTEM DESIGN

System design encompasses various aspects of designing an information system to meet the needs of users efficiently and effectively. It involves designing the input and output components of the system to ensure smooth operation and optimal utilization of resources. Here, we discuss the input design and output design aspects of the system.

Input Design:

Input design serves as the bridge between the user and the information system, facilitating the conversion of user requirements into a computer-based system. The primary objective of input design is to minimize errors in the data input process while guiding users to input the correct information for accurate results. Key considerations in input design include determining the data to be input, organizing or coding the data, providing user-friendly interfaces for data entry, implementing input validation mechanisms, and offering guidance to users in case of errors. The ultimate goal is to create an input layout that is easy to navigate, minimizes user confusion, and ensures data accuracy.

Output Design:

Output design focuses on presenting the processed information to users in a clear and meaningful manner. The quality of output is crucial as it directly impacts users' ability to make informed decisions. Efficient output design enhances the system's usability and aids in user decision-making by conveying information about past activities, current status, or future projections. Key considerations in output design include organizing output elements systematically, selecting appropriate methods for presenting information, creating various output formats such as documents or reports, and ensuring that outputs convey relevant information effectively. The output design should also aim to signal important events, trigger actions, or confirm user actions to facilitate efficient interaction with the system.

Overall, effective system design requires careful consideration of both input and output components to create a user-friendly, error-free, and efficient information system that meets the needs of its users.

5.2 FLOW OF THE PROJECT

Flow of the Project for Fake Job Post Prediction:

5.2.1 Data Collection:

Gather a dataset containing information about job postings from various sources. This dataset should include features such as job title, description, company details, salary, location, etc. Ensure that the dataset includes both legitimate and fake job postings.

5.2.2 Data Preprocessing:

Perform data cleaning to handle missing values, outliers, and inconsistencies in the dataset.

Normalize or standardize numerical features to ensure uniformity in scale. Convert categorical variables into numerical representations using techniques like one-hot encoding or label encoding. Split the dataset into training and testing sets.

5.2.3 Feature Engineering:

Extract relevant features from the dataset that can help distinguish between legitimate and fake job postings. Feature engineering may involve text preprocessing techniques such as tokenization, removing stopwords, and stemming or lemmatization for textual data.

5.2.4 Model Selection:

Choose appropriate machine learning algorithms for fake job post prediction. Commonly used algorithms include Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machines (SVM). Train multip3le models using the training dataset and evaluate their performance using appropriate evaluation metrics such as accuracy, precision, recall, and F1-score.

5.2.5 Model Training:

Train the selected machine learning models on the training dataset using the engineered features. Tune hyperparameters of the models using techniques like grid search or random search to optimize performance.

5.2.6 Model Evaluation:

Evaluate the trained models using the testing dataset to assess their performance on unseen data. Compare the performance of different models based on evaluation metrics and select the best-performing model for deployment.

5.2.7 Deployment:

Deploy the selected model into a production environment where it can be used for real-time prediction of fake job postings. Integrate the model into a web application or API that accepts job posting data as input and returns predictions on whether the job posting is fake or legitimate.

5.2.8 Monitoring and Maintenance:

Monitor the performance of the deployed model over time and retrain it periodically with new data to ensure its accuracy and reliability. Update the model as needed to adapt to changes in the data distribution or new patterns of fake job postings.

By following this flow, the project for fake job post prediction can effectively leverage machine learning techniques to identify and prevent fraudulent activities in the job market.

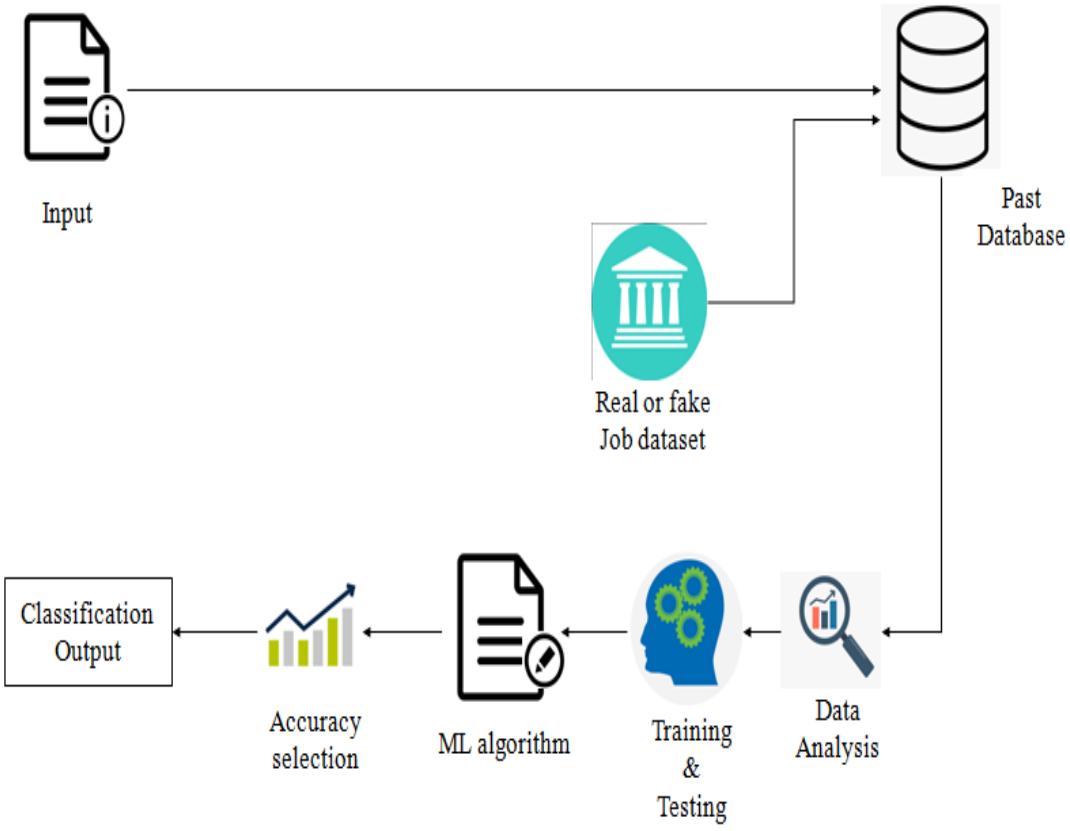


Fig : 5.2.8 Flow Diagram

5.3 FLOWCHART

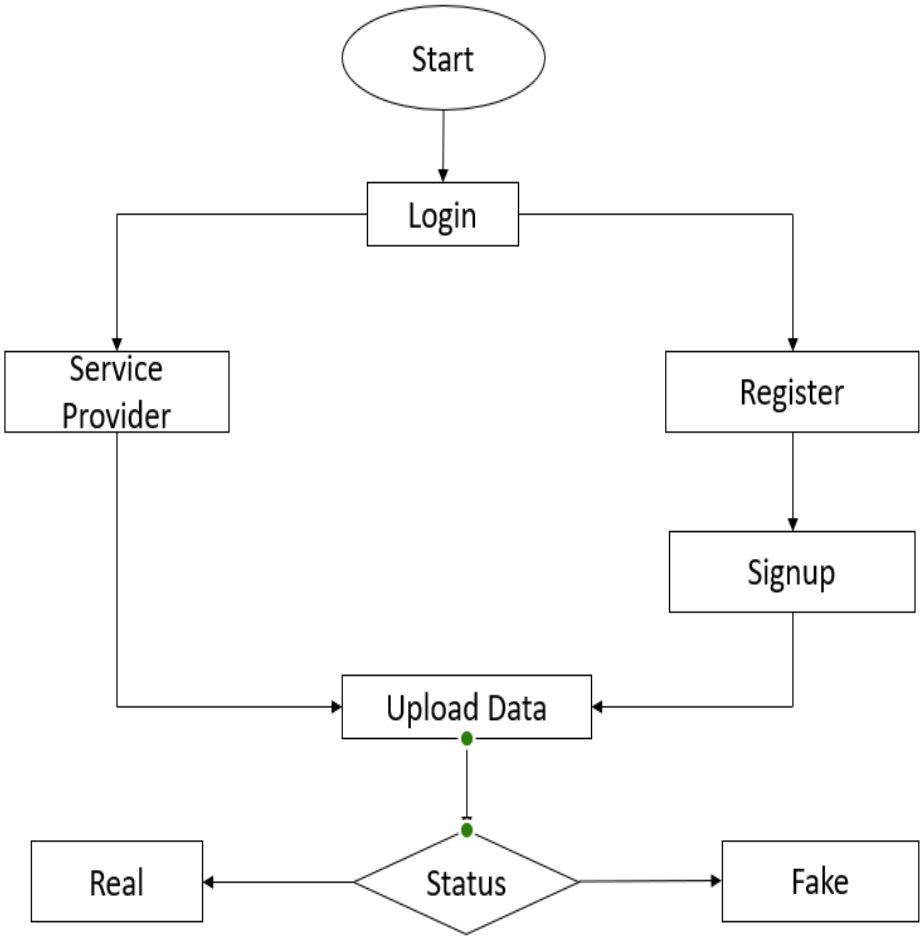


Fig : 5.2 Flow Chart

5.4 UML DIAGRAMS

The Unified Modeling Language (UML) is a standardized, general-purpose modeling language utilized in object-oriented software engineering. Managed by the Object Management Group (OMG), UML aims to establish a common language for creating models of object-oriented computer software. It consists of two main components: a Meta-model and a notation, with the possibility of incorporating additional methods or processes in the future.

UML serves as a standard means for specifying, visualizing, constructing, and documenting software system artifacts, as well as for modeling business and non-software systems. It embodies a collection of best engineering practices proven effective in modeling large and complex systems. Employing mostly graphical notations, UML plays a significant role in developing object-oriented software and the software development process.

The goals underlying the design of UML are:

1. Offer users an expressive visual modeling language that facilitates the development and exchange of meaningful models.
2. Provide mechanisms for extendibility and specialization to expand upon core concepts.
3. Maintain independence from specific programming languages and development processes.
4. Establish a formal basis for comprehending the modeling language.
5. Stimulate the growth of the object-oriented tools market.
6. Support advanced development concepts such as collaborations, frameworks, patterns, and components.
7. Integrate best practices into the modeling process.

5.4.1 USE CASE DIAGRAM:

The following shown figure is the Use-case diagram of our proposed project. It describes what the system does and how the system operates internally.

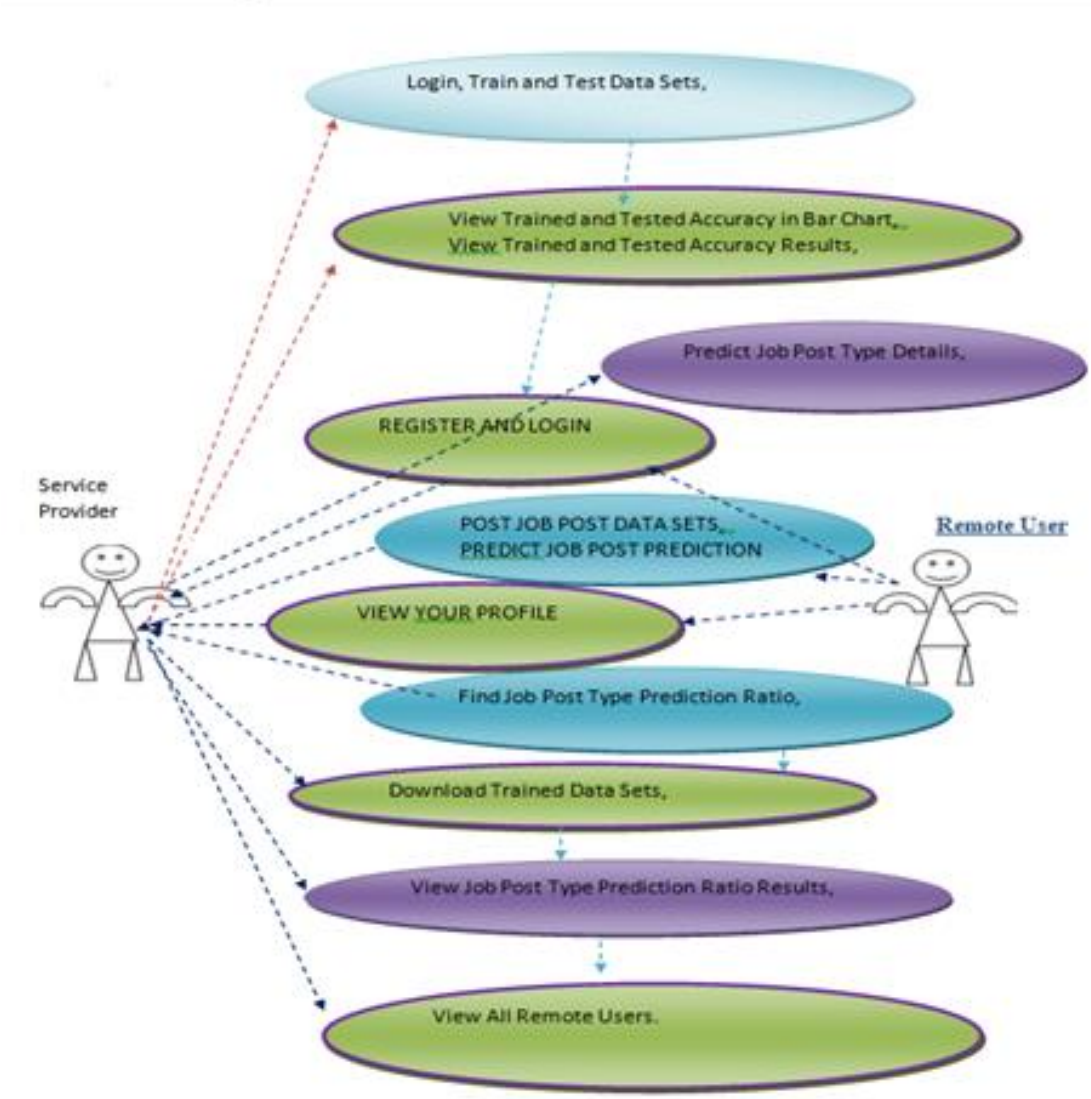


Fig : 5.4.1 Use Case Diagram

5.4.2 CLASS DIAGRAM:

The following shown figure is the Class diagram of our proposed project . It is used to model the objects that make up the system, to display the relationships between the objects and to describe what those objects do and the services that they provide.

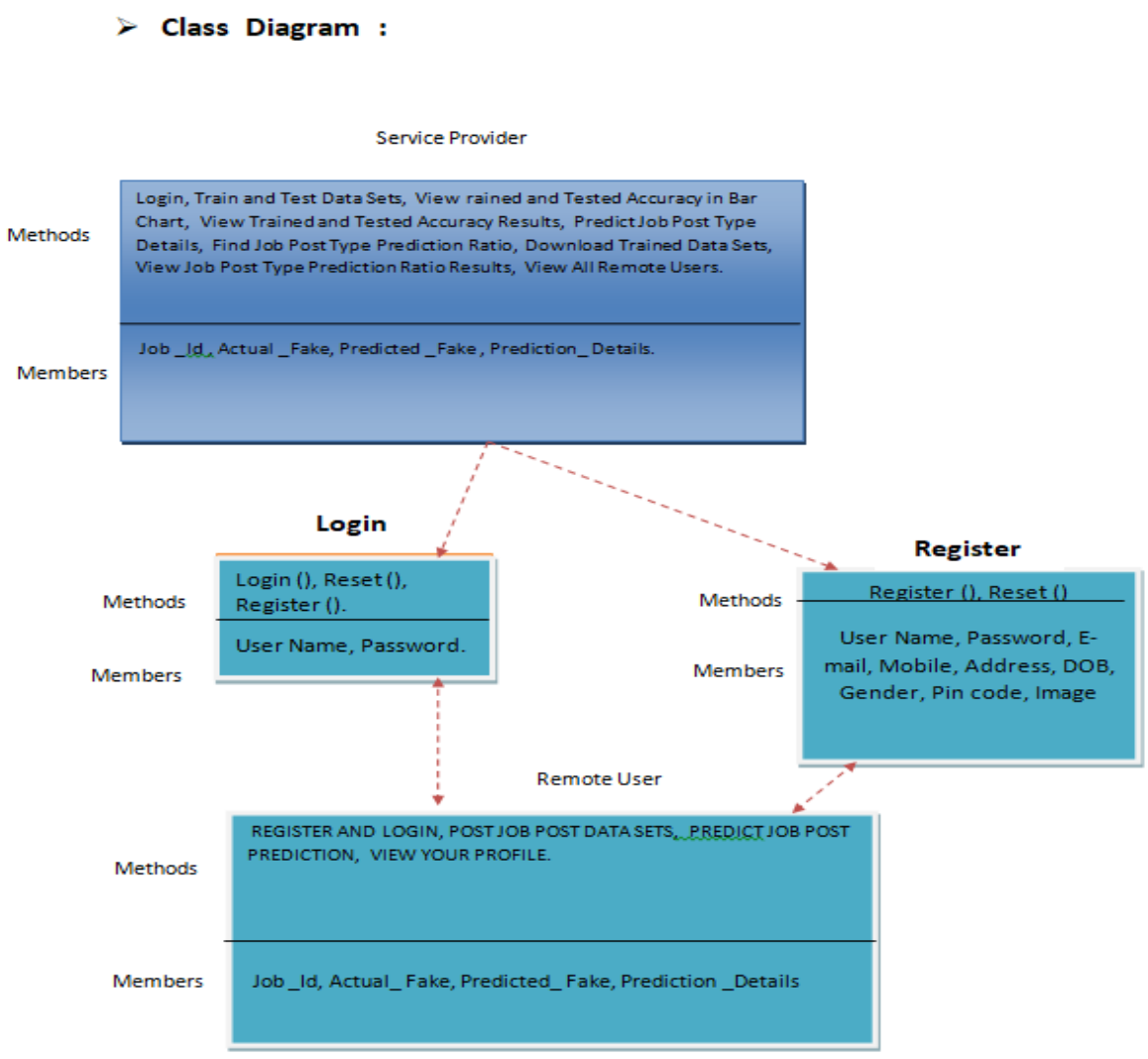


Fig: 5.4.2 Class Diagram

5.4.3 SEQUENCE DIAGRAM:

- The following diagram is the **Sequence diagram.** This sequence diagram consists of four objects the User, Main, and Audio to Sign. It shows the sequence of actions between the user and the feature Audio to Sign Language Conversion.

➢ **Sequence Diagram**

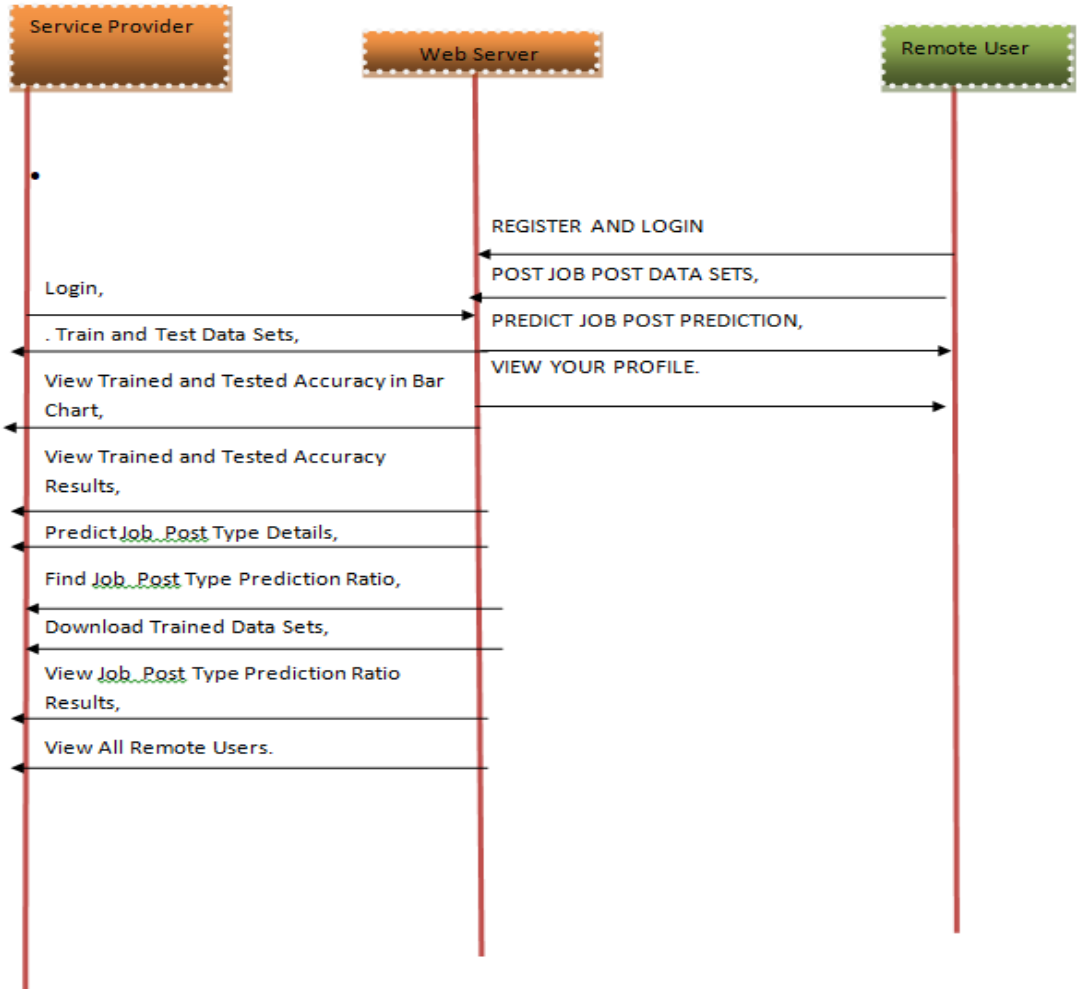


Fig : 5.4.3 Sequence Diagram

5.4.4 DEPLOYMENT DIAGRAM:

- A Deployment Diagram shows the hardware of your system and the software in that hardware of the proposed project. The below shown figure is the Deployment diagram of the proposed project.

Fig 5.4.4 Deployment Diagram

6. SYSTEM IMPLEMENTATION

6.1 INTRODUCTION TO SYSTEM IMPLEMENTATION

Implementing a fake job post detection system using machine learning (ML) involves several steps. Here's an overview of the process:

1. Data Collection: Gather a dataset of job postings. This dataset should include both genuine job postings and fake ones. You can collect data from job boards, online forums, or use existing datasets.

2. Data Preprocessing: Clean and preprocess the collected data. This may involve tasks such as removing duplicates, handling missing values, and normalizing text data.

3. Feature Extraction: Extract relevant features from the job postings. Features could include text-based features like the job description, title, location, salary range, and company information.

4. Labeling: Label the data as genuine or fake. This can be done manually or using a semi-supervised approach where some fake postings are labeled initially, and then the model learns to identify similar patterns.

5. Model Selection: Choose an appropriate machine learning model for classification. Common models for text classification tasks include Logistic Regression, Support Vector Machines (SVM), Random Forest, and Neural Networks.

6. Training: Split the dataset into training and testing sets. Train the selected model on the training data.

7. Evaluation: Evaluate the performance of the trained model using the testing dataset. Metrics such as accuracy, precision, recall, and F1-score can be used to assess the model's performance.

8. Deployment: Once satisfied with the model's performance, deploy it into a production environment. This could involve integrating the model into a web application or API where users can input job postings, and the model outputs predictions.

9. Monitoring and Maintenance: Continuously monitor the performance of the deployed model and retrain it periodically with new data to ensure that it remains effective over time.

Throughout the implementation process, it's essential to consider ethical considerations, such as bias detection and mitigation, especially when dealing with sensitive topics like job postings.

6.2 SELECTED SOFTWARE & PROGRAMMING LANGUAGE

- **DJANGO**

We Choosing Django for implementing a fake job post detection system brings several advantages, which can be justified based on the specific requirements and characteristics of the project:

1. Rapid Development: Django is known for its "batteries-included" approach, providing a wide range of built-in features and functionalities that streamline web development. This includes an ORM (Object-Relational Mapping) for database interaction, built-in authentication and authorization system, admin interface, and URL routing

2. Scalability: Django is designed to scale efficiently, making it suitable for handling large volumes of data and user interactions. With features like database connection pooling, caching mechanisms, and support for asynchronous tasks (via libraries like Celery), Django can accommodate the potential scalability needs of a fake job.

3. Security: Security is paramount, especially when dealing with sensitive data and user information. Django provides built-in security features to help mitigate common web vulnerabilities such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). Additionally, Django's authentication system makes it easier to implement user authentication and access control, ensuring that only authorized users can access certain features or data within the system.

4. Community and Ecosystem: Django has a large and active community of developers, which means extensive documentation, tutorials, and a wide range of third-party packages and libraries available for integration. This can be particularly beneficial for implementing machine learning functionalities within the fake job post detection system.

5. Flexibility and Customization: While Django provides a lot of built-in functionality out-of-the-box, it's also highly customizable and extensible. This allows you to tailor the fake job post detection system to meet specific requirements and integrate additional features or third-party services as needed.

6. Community Support and Maintenance: Django's popularity ensures ongoing community support and maintenance, which is essential for the long-term sustainability of the project. Updates, security patches, and new features are regularly released, ensuring that the fake job post detection system remains up-to-date and secure against emerging threats or vulnerabilities.

Overall, choosing Django for implementing a fake job post detection system provides a robust and reliable foundation for developing, deploying, and maintaining a scalable and secure web application with machine learning capabilities.

- **PYTHON**

Choosing Python as the programming language for implementing fake job post detection using machine learning (ML) offers several advantages:

1.Rich Ecosystem for Machine Learning: Python has a vast ecosystem of libraries and frameworks specifically designed for machine learning, such as TensorFlow, Scikit-learn, PyTorch, and Keras. These libraries provide powerful tools for building and deploying ML models, making Python a natural choice for ML-based projects.

2.Ease of Prototyping and Experimentation: Python's simplicity and readability make it well-suited for rapid prototyping and experimentation. This is crucial for developing and refining machine learning algorithms, as it allows data scientists and developers to quickly iterate on ideas and explore different approaches.

3.Community Support and Resources: Python has a large and active community of developers, data scientists, and researchers who contribute to the development of machine learning tools and resources. This vibrant community ensures extensive documentation, tutorials, and online support forums, making it easier to learn and implement machine learning solutions in Python.

4.Interoperability with Other Technologies: Python seamlessly integrates with other technologies commonly used in web development, such as Django, Flask, and FastAPI. This facilitates the integration of machine learning models into web applications, allowing you to deploy ML-powered features alongside other application functionalities.

5.Scalability and Performance: While Python is not typically known for its raw performance, libraries like TensorFlow and PyTorch offer support for distributed computing

and GPU acceleration, enabling scalable and high-performance machine learning computations. Additionally, Python's ease of integration with languages like C and C++ allows for performance-critical components to be optimized if needed.

6.Availability of Pre-trained Models and Transfer Learning: Python's extensive library ecosystem includes pre-trained models and pretrained embeddings that can be readily used for various natural language processing (NLP) and text classification tasks, which are relevant for fake job post detection. Additionally, techniques like transfer learning allow developers to leverage pre-trained models and fine-tune them for specific tasks, reducing the amount of data and computational resources required for training new models.

7.Support for Data Analysis and Visualization: Python's rich ecosystem includes libraries like Pandas, NumPy, and Matplotlib, which are widely used for data analysis, manipulation, and visualization. These tools are invaluable for preprocessing data, extracting relevant features, and visualizing the results of machine learning models, enhancing the overall effectiveness of fake job post detection systems.

Overall, Python's versatility, extensive library ecosystem, and community support make it an excellent choice for implementing fake job post detection using machine learning, enabling developers to build robust, scalable, and efficient solutions.

We selected Django as the software framework and Python as the programming language for implementing our fake job post detection system due to their combined strengths in rapid development, scalability, security, and extensive ecosystem support. Python's simplicity, readability, and vast ecosystem of libraries make it ideal for implementing machine learning algorithms, while Django's "batteries-included" approach provides built-in features and functionalities that streamline web development. Together, Python and Django offer a powerful and flexible platform for building web applications with machine learning capabilities, enabling us to develop a robust and scalable solution for detecting fake job postings efficiently.

6.3 STEPS INVOLVED IN IMPLEMENTATION OF ALGORITHM

The implementation of an algorithm for predicting fake job posts using various machine learning techniques involves several steps. Here are the general steps involved:

1.Data Collection:

Gather a dataset containing features related to job posts, such as job description, required qualifications, salary offered, etc. Additionally, label each job post as either real or fake based on ground truth information.

2. Data Preprocessing:

Clean the data by removing any irrelevant or duplicate entries. Handle missing values by imputing them or removing rows/columns with missing data. Perform text preprocessing techniques such as tokenization, removing stopwords, and stemming/lemmatization.Encode categorical variables into numerical representations using techniques like one-hot encoding or label encoding.

3. Feature Engineering:

Extract relevant features from the dataset that may be useful for distinguishing between real and fake job posts. Generate additional features if necessary, such as word embeddings or TF-IDF vectors to represent text data.

4. Splitting Data:

Divide the dataset into training and testing sets to evaluate the performance of the machine learning models.

5. Model Selection:

Choose the machine learning algorithms to use for fake job post prediction, including Random Forest, k-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machines (SVM).

6. Model Training:

Train each selected machine learning model on the training dataset.

7. Model Evaluation:

Evaluate the performance of each model on the testing dataset using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

8. Hyperparameter Tuning:

Optimize the hyperparameters of each model using techniques like grid search or random search to improve their performance.

9. Cross-Validation:

Perform k-fold cross-validation to ensure the robustness and generalization of the models.

10. Model Comparison:

Compare the performance of different machine learning algorithms based on evaluation metrics to determine the most effective approach for predicting fake job posts.

By following these steps, you can implement and deploy an algorithm for predicting fake job posts using various machine learning techniques like Random Forest, k-Nearest Neighbors (KNN), Naive Bayes, and Support Vector Machines (SVM).

6.4 SAMPLE CODE

SNIPPET 1

```
from django.db import models

# Create your models here.

from django.db.models import CASCADE

class ClientRegister_Model(models.Model):

    username = models.CharField(max_length=30)

    email = models.EmailField(max_length=30)

    password = models.CharField(max_length=10)

phoneno = models.CharField(max_length=10)

    country = models.CharField(max_length=30)

    state = models.CharField(max_length=30)
```

```
    city = models.CharField(max_length=30)
class Prediction_Results(models.Model):
Job_Id=models.CharField(max_length=300)
Actual_Fake=models.CharField(max_length=300)
Predicted_Fake=models.CharField(max_length=300)
   class Job_Post_Prediction(models.Model):
Job_Id=models.CharField(max_length=300)
Actual_Fake=models.CharField(max_length=300)
Predicted_Fake=models.CharField(max_length=300)
Prediction_Details=models.CharField(max_length=300)
    class detection_accuracy(models.Model):


    names = models.CharField(max_length=300)
    ratio = models.CharField(max_length=300)
class detection_ratio(models.Model):


    names = models.CharField(max_length=300)
    ratio = models.CharField(max_length=300)
```

SNIPPET 2

```
from django import forms
from Remote_User.models import ClientRegister_Model
class ClientRegister_Form(forms.ModelForm):
```

```
password = forms.CharField(widget=forms.PasswordInput())

email = forms.EmailField(required=True)

class Meta:

   model = ClientRegister_Model

   fields = ("username","email","password","phoneno","country","state","city")
```

SNIPPET 3

```
import os

# Build paths inside the project like this: os.path.join(BASE_DIR, ...)
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(_file_)))

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.0/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'm+1edl5m-5@u9u!b8-=4-4mq&o1%agco2xpl8c!7sn7!eowjk#'

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = []

# Application definition

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
```

```
    'Remote_User',
    'Service_Provider',
]

MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
    'django.contrib.auth.middleware.AuthenticationMiddleware',
    'django.contrib.messages.middleware.MessageMiddleware',
    'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'aComparative_study_onFake_jobPost.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [(os.path.join(BASE_DIR,'Template/htmls'))],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    },
]

WSGI_APPLICATION = 'aComparative_study_onFake_jobPost.wsgi.application'
```

```
# Database
# https://docs.djangoproject.com/en/3.0/ref/settings/#databases

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'aComparative_study_onFake_jobPost',
        'USER':'root',
        'PASSWORD': '',
        'HOST' :'127.0.0.1',
        'PORT' :'3306',
    }
}

# Password validation
# https://docs.djangoproject.com/en/3.0/ref/settings/#auth-password-validators

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]
```

```
# Internationalization
# https://docs.djangoproject.com/en/3.0/topics/i18n/

LANGUAGE_CODE = 'en-us'
TIME_ZONE = 'UTC'
USE_I18N = True
USE_L10N = True
USE_TZ = True

# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/3.0/howto/static-files/

STATIC_URL = '/static/'
STATICFILES_DIRS = [os.path.join(BASE_DIR,'Template/images')]
MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'Template/media')

STATIC_ROOT = '/static/'

STATIC_URL = '/static/'
```

7. TESTING

7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

TYPES OF TESTS

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were

individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional testing

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

7.1 Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page

7.2 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

7.3 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

SYSTEM TESTING

TESTING METHODOLOGIES

The following are the Testing Methodologies:

- Unit Testing
- Integration Testing
- User Acceptance Testing
- Output Testing
- **Validation Testing**

Unit Testing:

Unit testing focuses verification effort on the smallest unit of Software design that is the module. Unit testing exercises specific paths in a module's control structure to ensure complete coverage and maximum error detection. This test focuses on each module individually, ensuring that it functions properly as a unit. Hence, the naming is Unit Testing.

During this testing, each module is tested individually and the module interfaces are verified for the consistency with design specification. All important processing path are tested for the expected results. All error handling paths are also tested.

Integration Testing:

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order tests are conducted. The main objective in this testing process is to take unit tested modules and builds a program structure that has been dictated by design.

The following are the types of Integration Testing:

1. Top Down Integration

This method is an incremental approach to the construction of program structure. Modules are integrated by moving downward through the control hierarchy, beginning with the main program module. The module subordinates to the main program module are incorporated into the structure in either a depth first or breadth first manner.

In this method, the software is tested from main module and individual stubs are replaced when the test proceeds downwards.

2. Bottom-up Integration

This method begins the construction and testing with the modules at the lowest level in the program structure. Since the modules are integrated from the bottom up, processing required for modules subordinate to a given level is always available and the need for stubs is eliminated. The bottom up integration strategy may be implemented with the following steps:

- The low-level modules are combined into clusters into clusters that perform a specific Software sub-function.
- A driver (i.e.) the control program for testing is written to coordinate test case input and output.
- The cluster is tested.
- Drivers are removed and clusters are combined moving upward in the program structure

The bottom up approaches tests each module individually and then each module is module is integrated with a main module and tested for functionality.

OTHER TESTING METHODOLOGIES

User Acceptance Testing

User Acceptance of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with the prospective system users at the time of developing and making changes wherever required. The system developed provides a friendly user interface that can easily be understood even by a person who is new to the system.

Output Testing

After performing the validation testing, the next step is output testing of the proposed system, since no system could be useful if it does not produce the required output in the specified format. Asking the users about the format required by them tests the outputs generated or displayed by the system under consideration. Hence the output format is considered in 2 ways – one is on screen and another in printed format.

Validation Checking

Validation checks are performed on the following fields.

Text Field:

The text field can contain only the number of characters lesser than or equal to its size. The text fields are alphanumeric in some tables and alphabetic in other tables. Incorrect entry always flashes and error message.

Numeric Field:

The numeric field can contain only numbers from 0 to 9. An entry of any character flashes an error message. The individual modules are checked for accuracy and what it has to perform. Each module is subjected to test run along with sample data. The individually tested modulesare integrated into a single system. Testing involves executing the real data information is used in the program the existence of any program defect is inferred from the output. The testing should be planned so that all the requirements are individually tested. A successful test is one that gives out the defects for the inappropriate data and produces and output revealing the errors in the system.

Preparation of Test Data:

Taking various kinds of test data does the above testing. Preparation of test data plays a vital role in the system testing. After preparing the test data the system under study is tested using that test data. While testing the system by using test data errors are again uncovered and corrected by using above testing steps and corrections are also noted for future use.

Using Live Test Data:

Live test data are those that are actually extracted from organization files. After a system is partially constructed, programmers or analysts often ask users to key in a set of data from

their normal activities. Then, the systems person uses this data as a way to partially test the system. In other instances, programmers or analysts extract a set of live data from the files and have them entered themselves.

It is difficult to obtain live data in sufficient amounts to conduct extensive testing. And, although it is realistic data that will show how the system will perform for the typical processing requirement, assuming that the live data entered are in fact typical, such data generally will not test all combinations or formats that can enter the system. This bias toward typical values then does not provide a true systems test and in fact ignores the cases most likely to cause system failure.

Using Artificial Test Data:

Artificial test data are created solely for test purposes, since they can be generated to test all combinations of formats and values. In other words, the artificial data, which can quickly be prepared by a data generating utility program in the information systems department, make possible the testing of all login and control paths through the program.

The most effective test programs use artificial test data generated by persons other than those who wrote the programs. Often, an independent team of testers formulates a testing plan, using the systems specifications. The package "Virtual Private Network" has satisfied all the requirements specified as per software requirement specification and was accepted.

USER TRAINING

Whenever a new system is developed, user training is required to educate them about the working of the system so that it can be put to efficient use by those for whom the system has been primarily designed. For this purpose, the normal working of the project was demonstrated to the prospective users. Its working is easily understandable and since the expected users are people who have good knowledge of computers, the use of this system is very easy.

MAINTAINENCE

This covers a wide range of activities including correcting code and design errors. To reduce the need for maintenance in the long run, we have more accurately defined the user's

requirements during the process of system development. Depending on the requirements, this system has been developed to satisfy the needs to the largest possible extent. With development in technology, it may be possible to add many more features based on the requirements in future. The coding and designing is simple and easy to understand which will make maintenance easier.

TESTING STRATEGY:

A strategy for system testing integrates system test cases and design techniques into a well-planned series of steps that results in the successful construction of software. The testing strategy must co-operate test planning, test case design, test execution, and the resultant data collection and evaluation. A strategy for software testing must accommodate low-level tests that are necessary to verify that a small source code segment has been correctly implemented as well as high level tests that validate major system functions against user requirements.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification design and coding. Testing represents an interesting anomaly for the software. Thus, a series of testing are performed for the proposed system before the system is ready for user acceptance testing.

SYSTEM TESTING:

Software once validated must be combined with other system elements (e.g. Hardware, people, database). System testing verifies that all the elements are proper and that overall system function performance is achieved. It also tests to find discrepancies between the system and its original objective, current specifications and system documentation.

UNIT TESTING:

In unit testing different are modules are tested against the specifications produced during the design for the modules. Unit testing is essential for verification of the code produced during the coding phase, and hence the goals to test the internal logic of the modules. Using the detailed design description as a guide, important Conrail paths are tested to uncover errors within the boundary of the modules. This testing is carried out during the programming stage itself. In this type of testing step, each module was found to be working satisfactorily as regards to the expected output from the module.

In Due Course, latest technology advancements will be taken into consideration. As part of technical build-up many components of the networking system will be generic in nature so that future projects can either use or interact with this.The future holds a lot to offer to the development and refinement of this project.

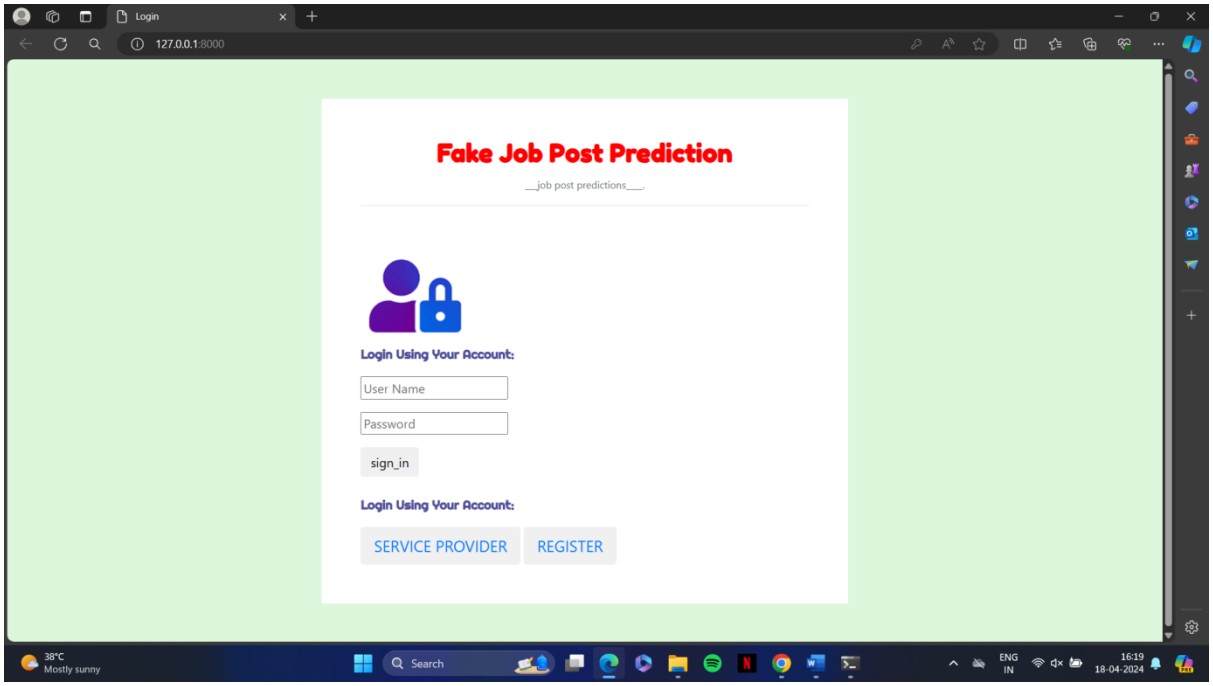
7.2 TEST CASES

Test case id	Test case description	Actual value	Entered value	Status
1	Register user details in registration page	Fill all the fields while registering user	All the fields are filled	Pass
2	Give user name in text box	User name must be given in alphabets	User name given in alphabets and numeric values	Fail
3	Password to be entered in password box	Password must be given correctly	Password is entered wrongly	Fail
4	Phone number must be entered in phone number box during registration	Phone number must be given in 10 digits	Phone number given in 10 digits	pass

Fig : 7.2 Test Cases

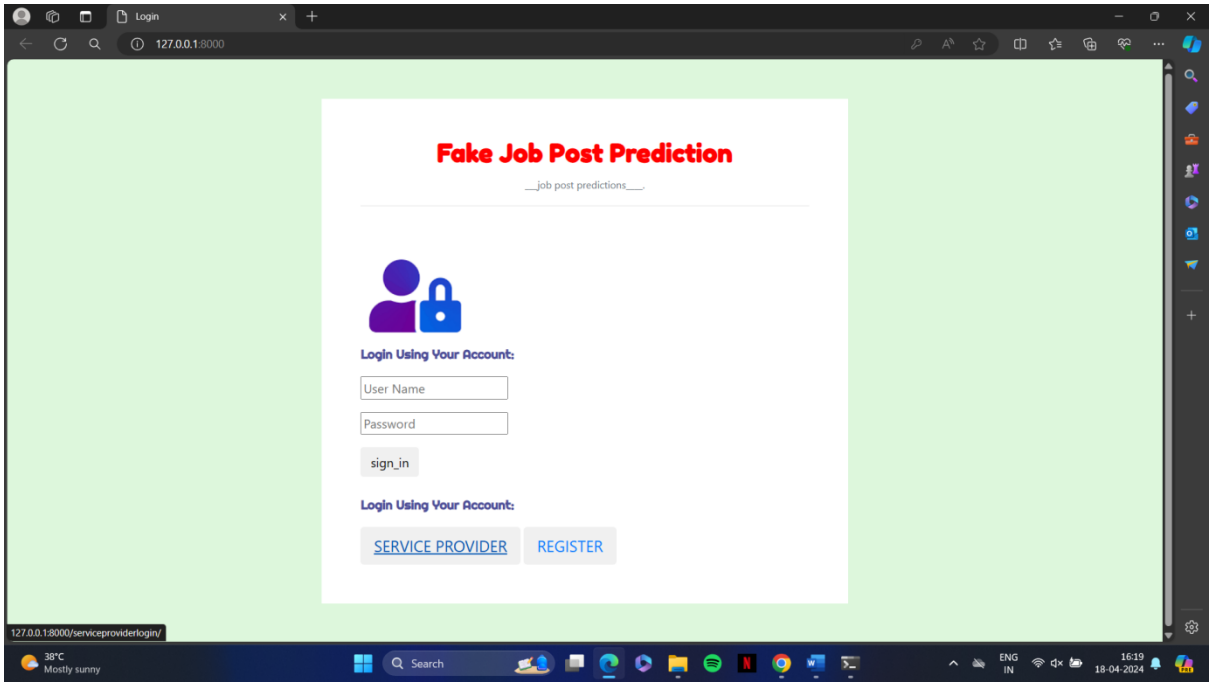
8.SCREEN AND OUTPUTS

Login page of website



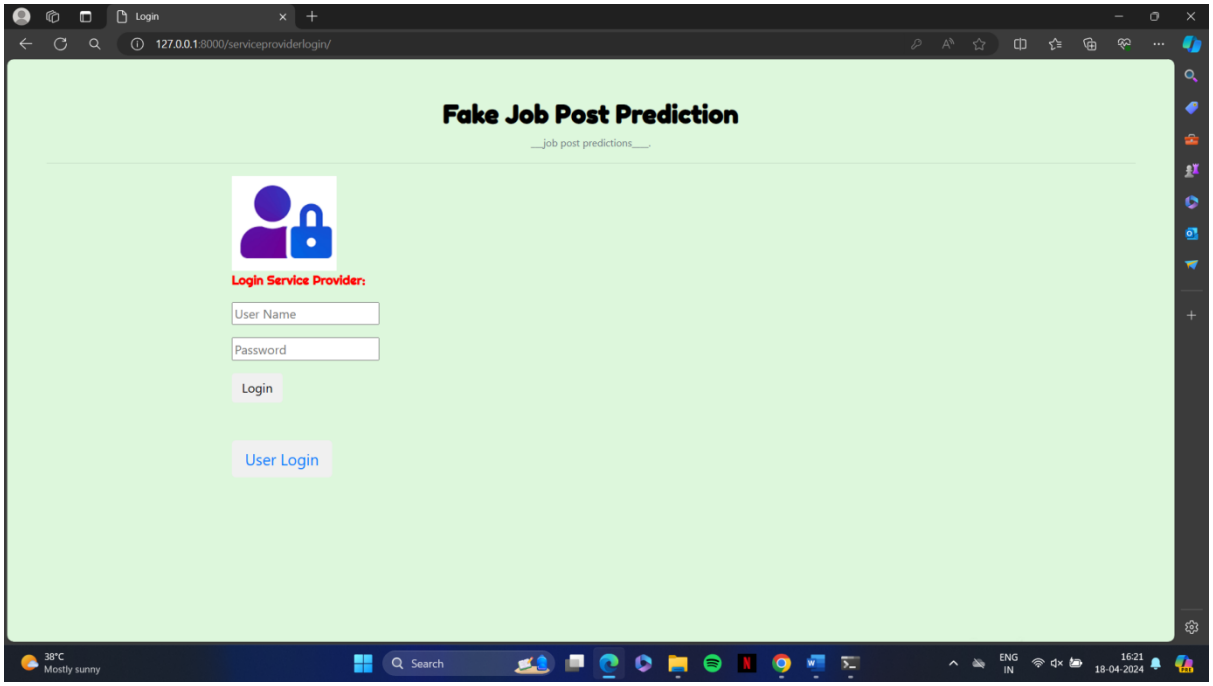
Screen 8.1 : Login page

To access Service Provider of the website.



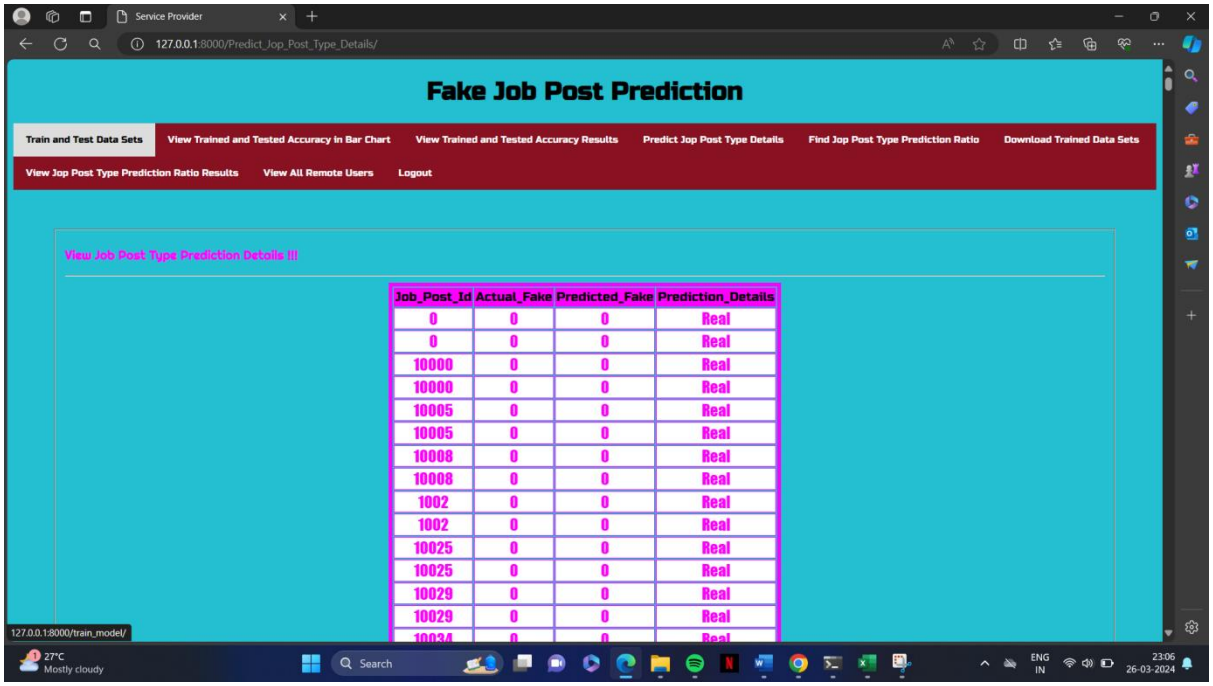
Screen 8.2 : Service Provider

To access the Service Provider, only the authorize can access it.

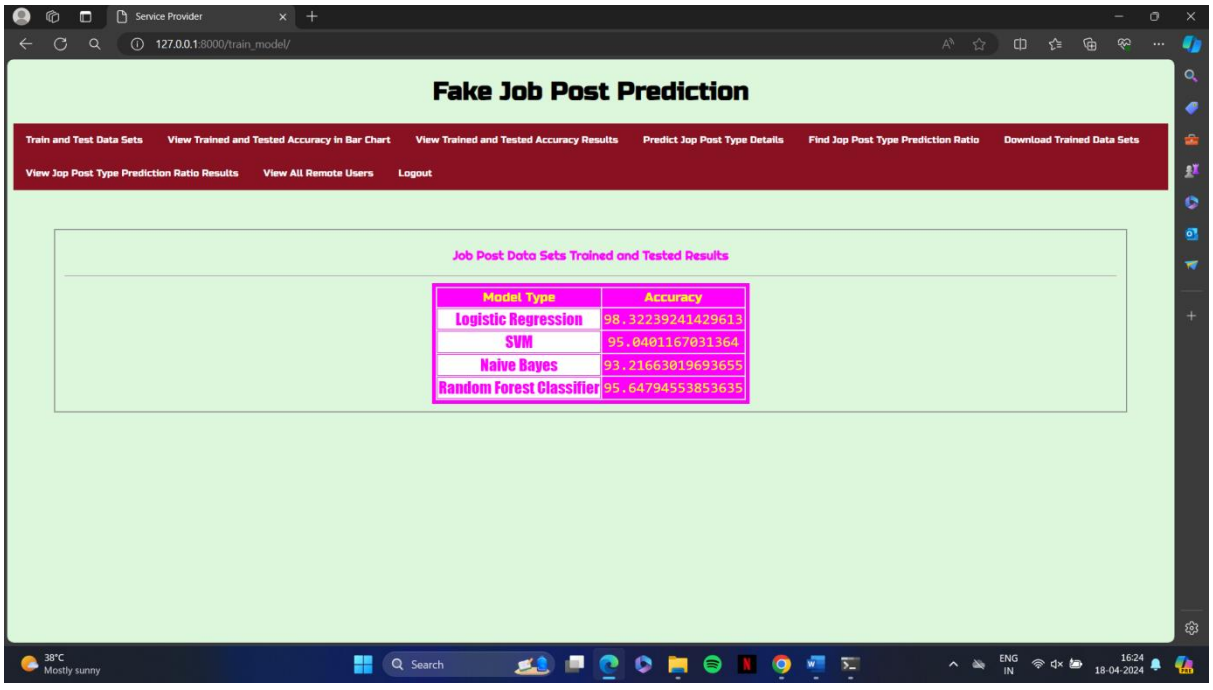


Screen 8.3 : Service Provider Authorized

Displaying Train and Test data sets.

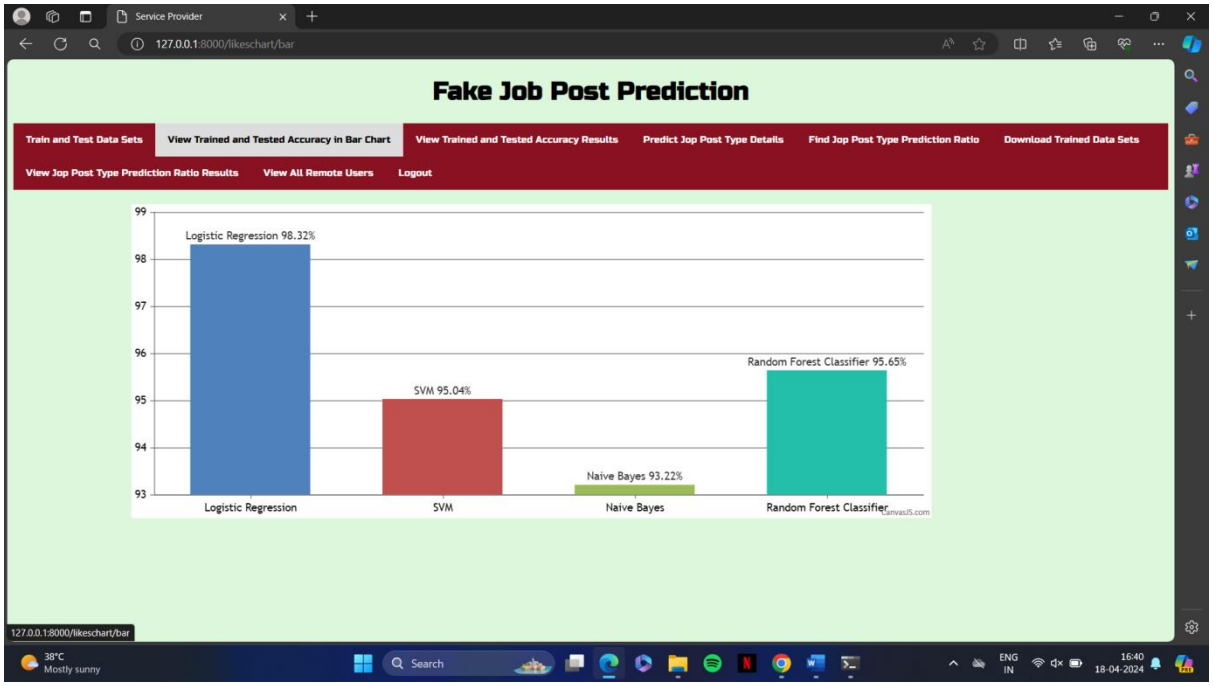


Screen 8.4 : Train and Test data sets



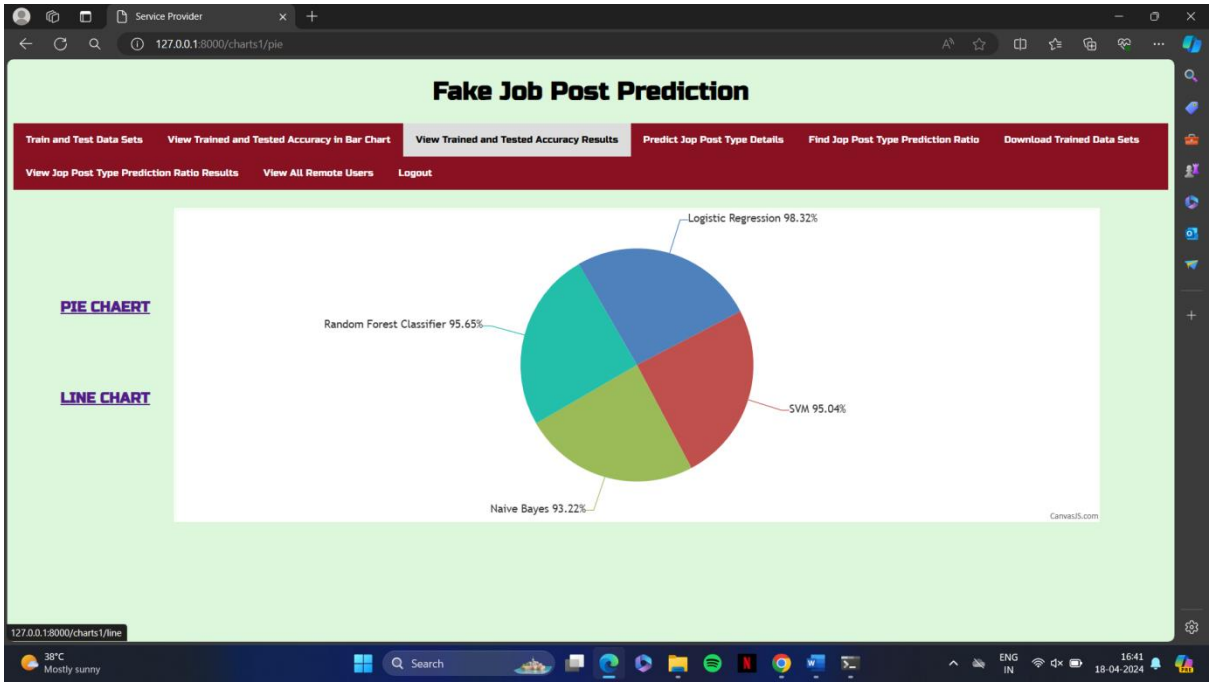
Screen 8.4.2 :Train and Test data sets (1)

Viewing Trained and Tested accuracy in Bar Chart for the algorithms.



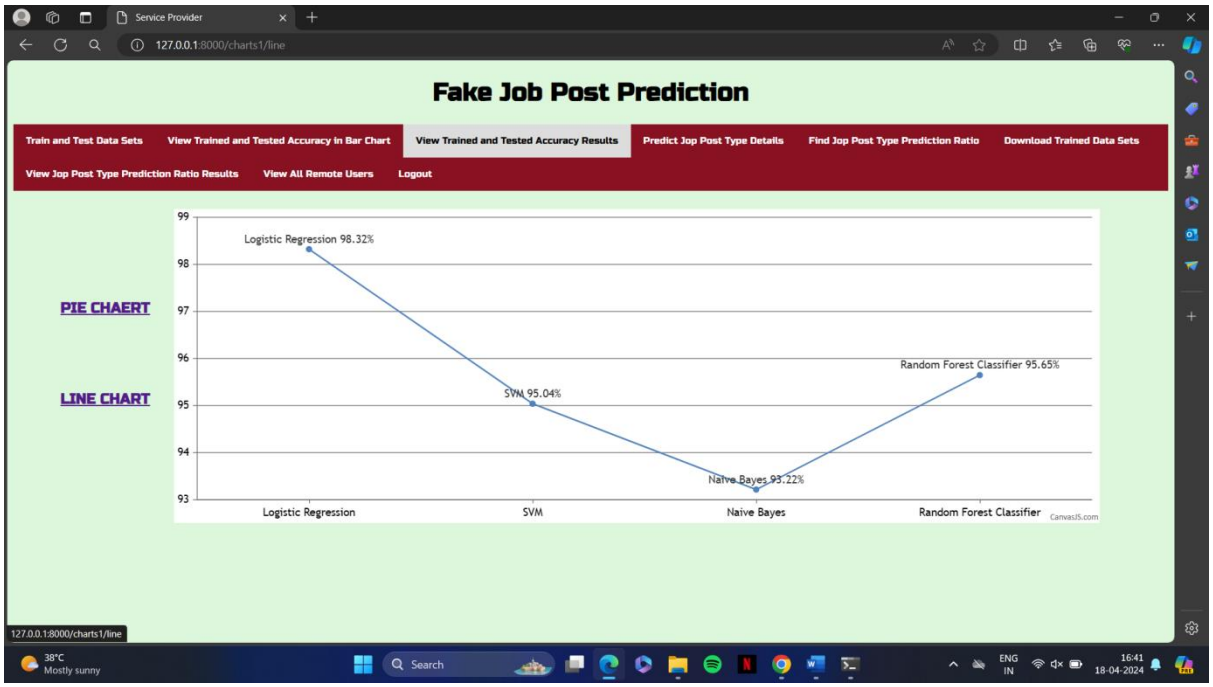
Screen 8.5 : Bar Chart

Viewing Trained and Tested Accuracy Results using Pie Chart.



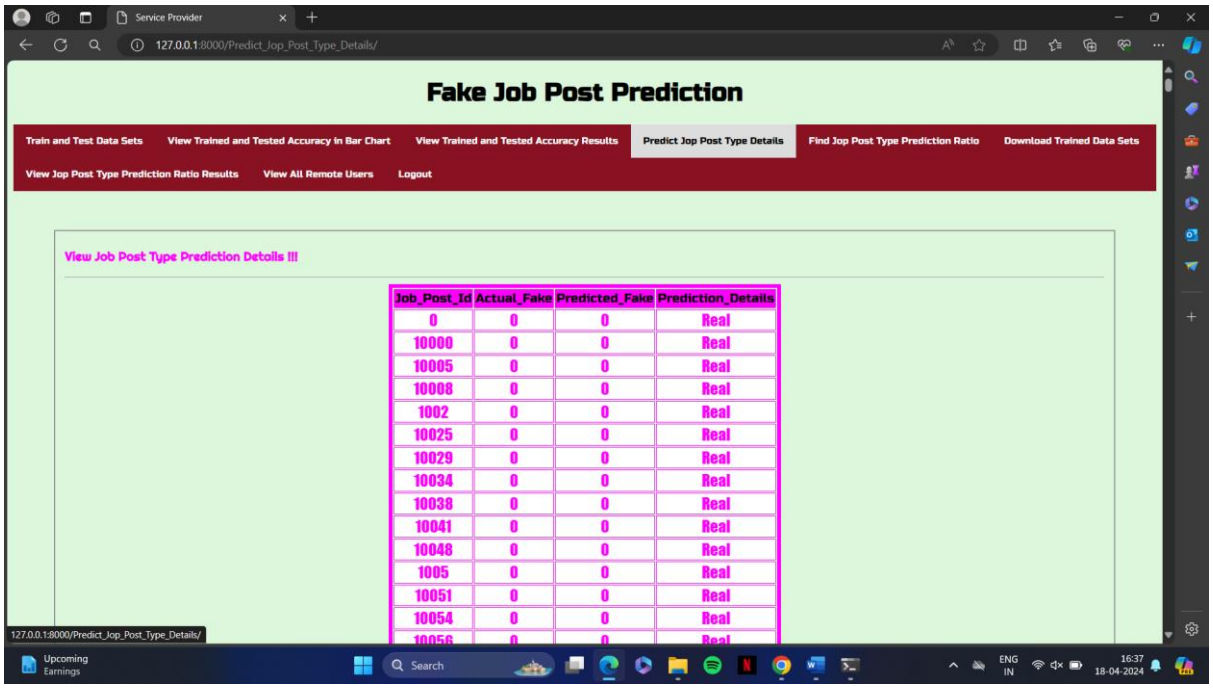
Screen 8.6 : Pie Chart

Viewing Trained and Tested Accuracy Results using Line Chart.



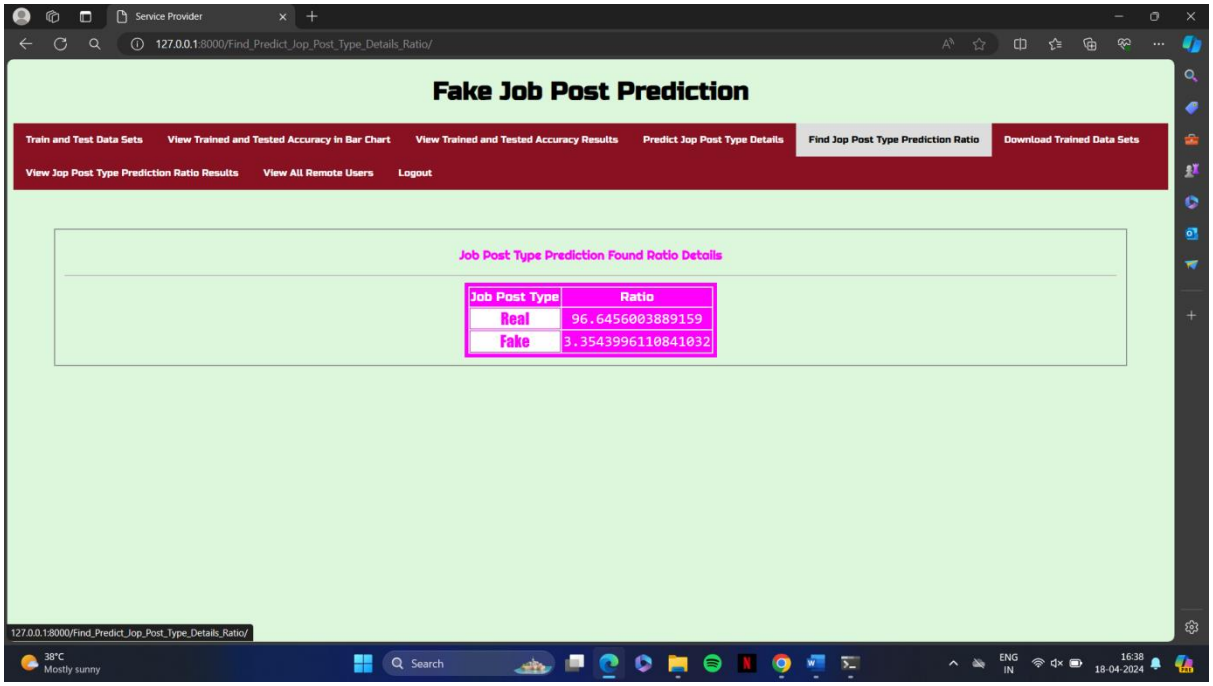
Screen 8.7 : Line Chart

Displaying job post type details predictions.



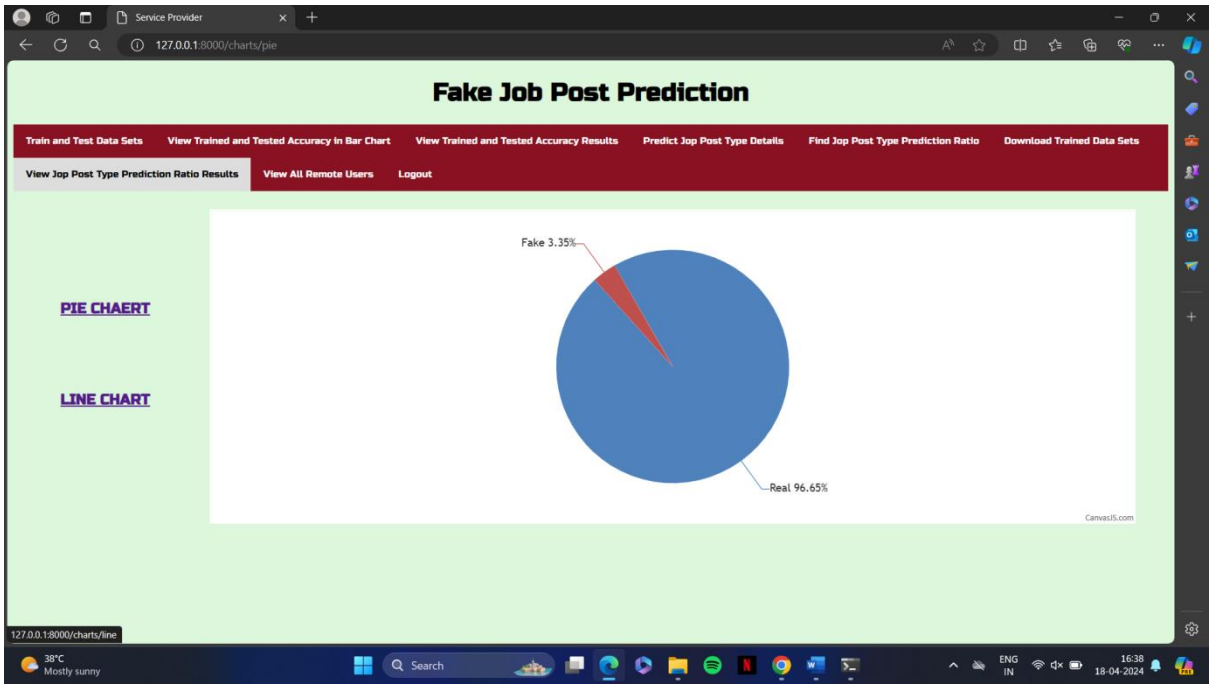
Screen 8.8 : job post type details prediction

Displaying job post type prediction ratio.



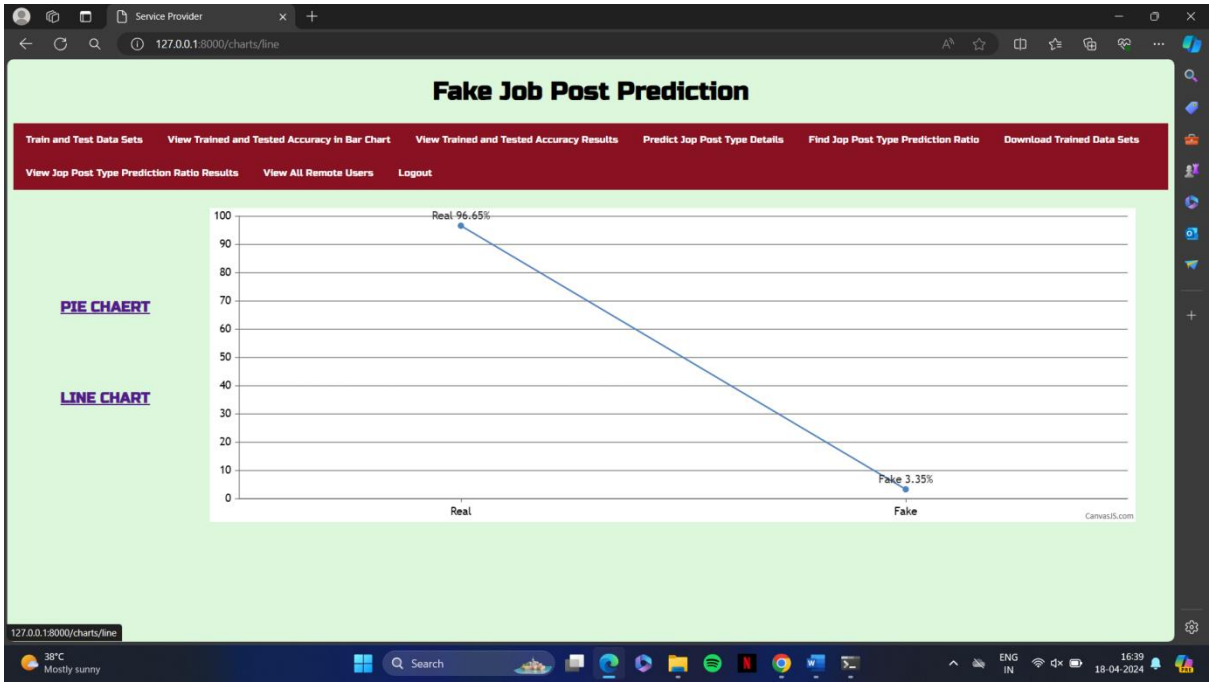
Screen 8.9 : Job post type prediction ratio

Viewing job post type prediction ratio results in Pie Chart.



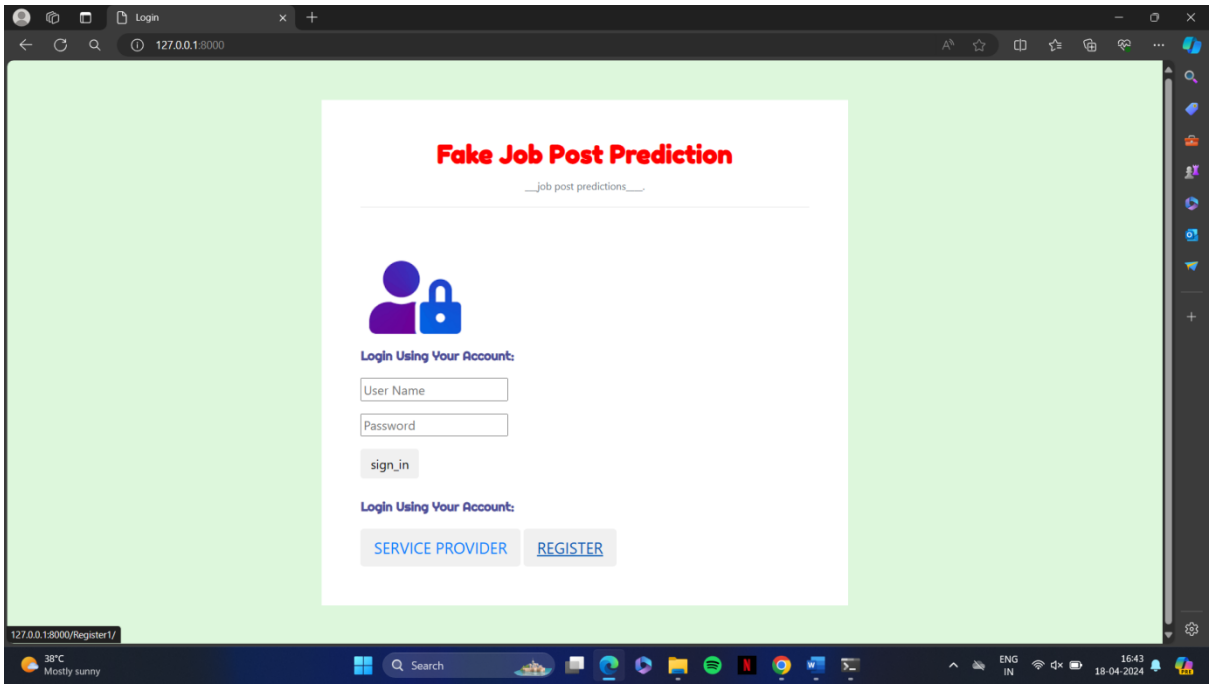
Screen 8.10 : Job post type prediction in pie chart

Viewing job post type prediction ratio results in Line Chart.



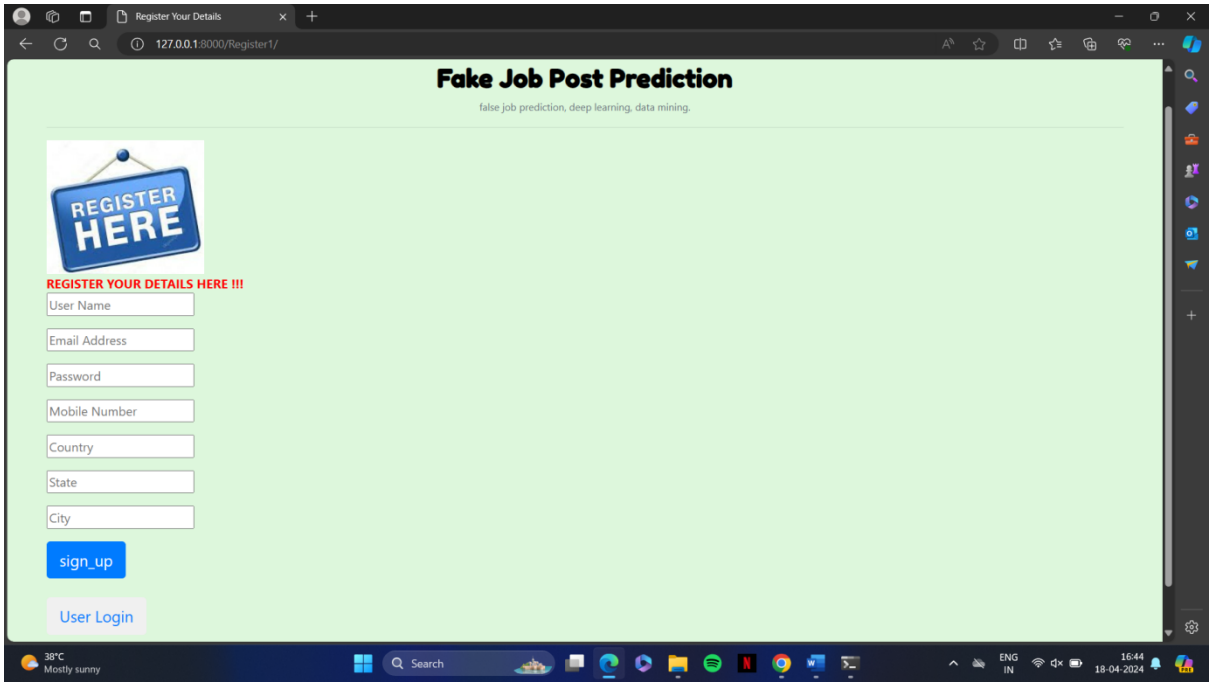
Screen 8.11 : Job post type prediction in Line Chart

For registering a new user.

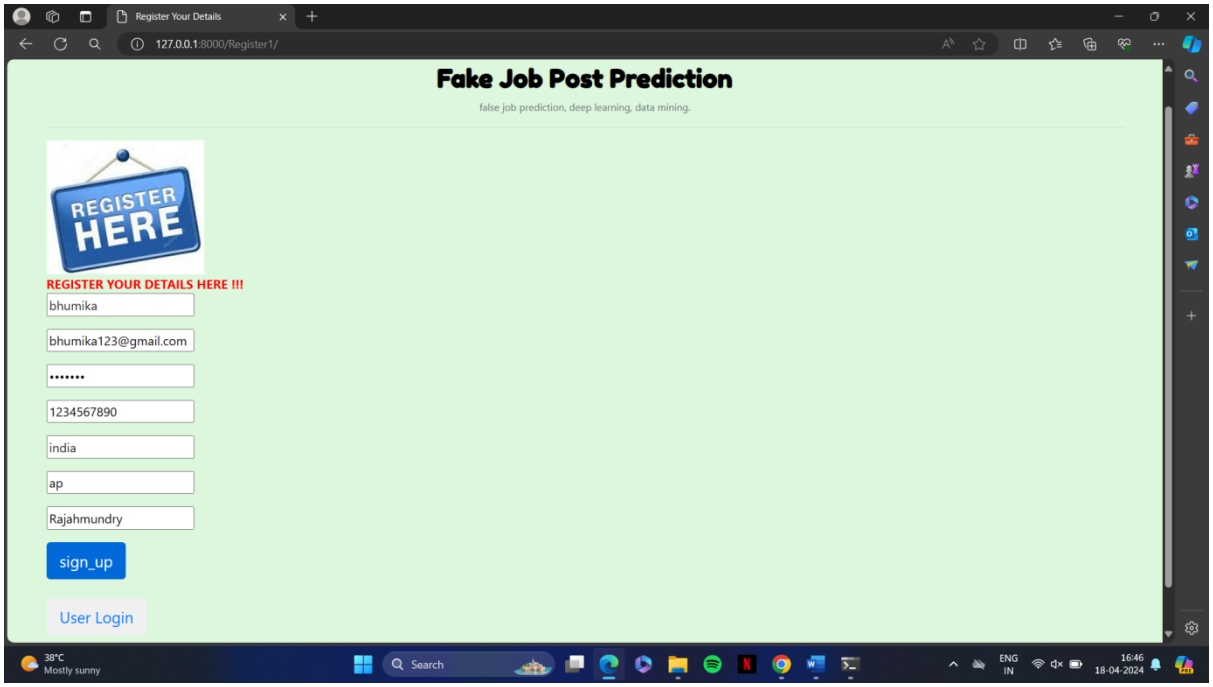


Screen 8.12 : New user

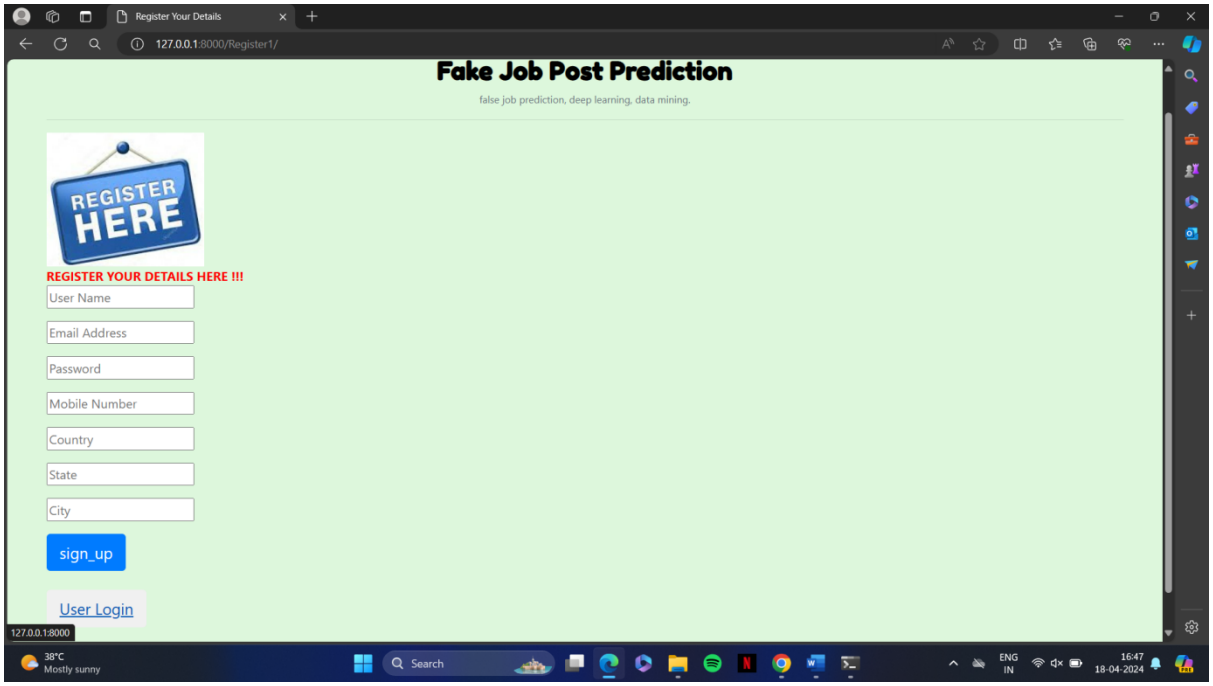
Enter the details and sign up.



Screen 8.13.1 : Sign Up (1)

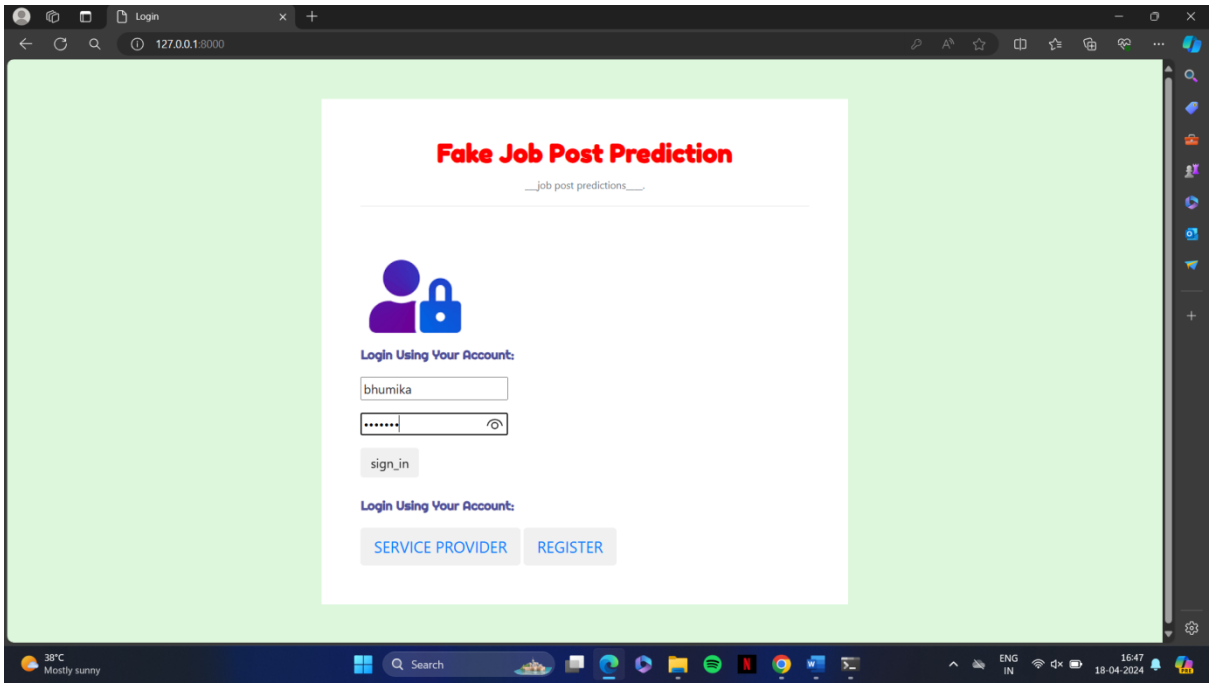


Screen 8.13.2: Sign Up (2)



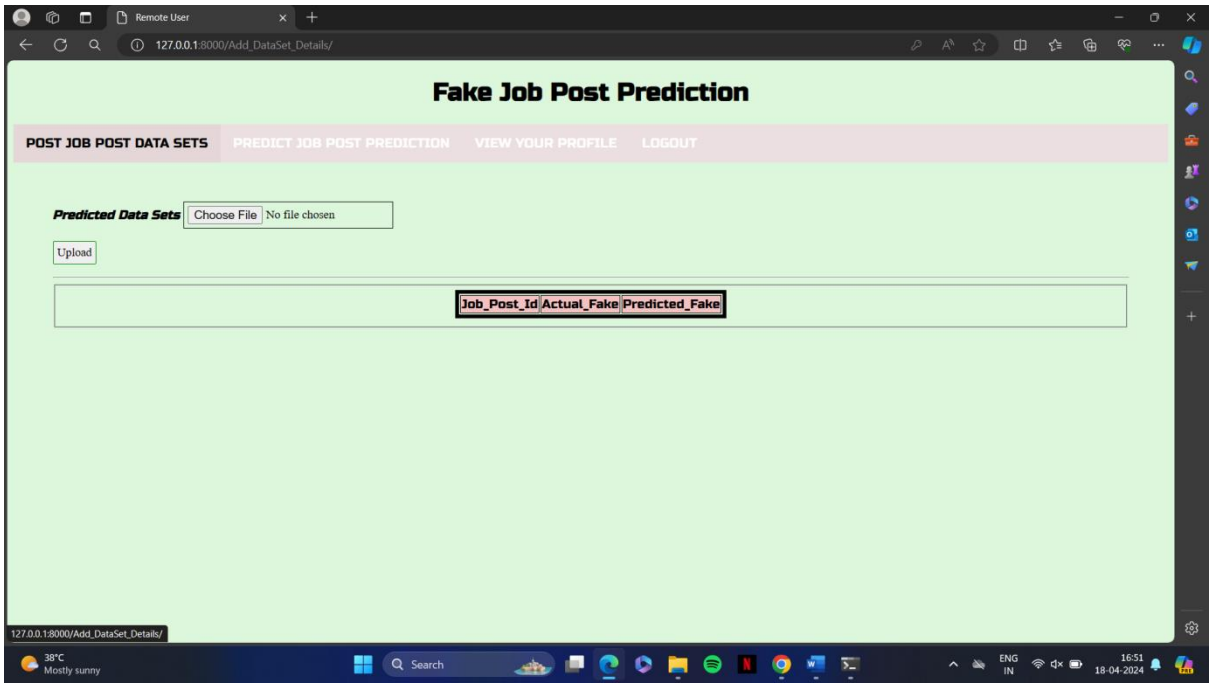
Screen 8.13.3 : Sign Up (3)

Login with creditials.



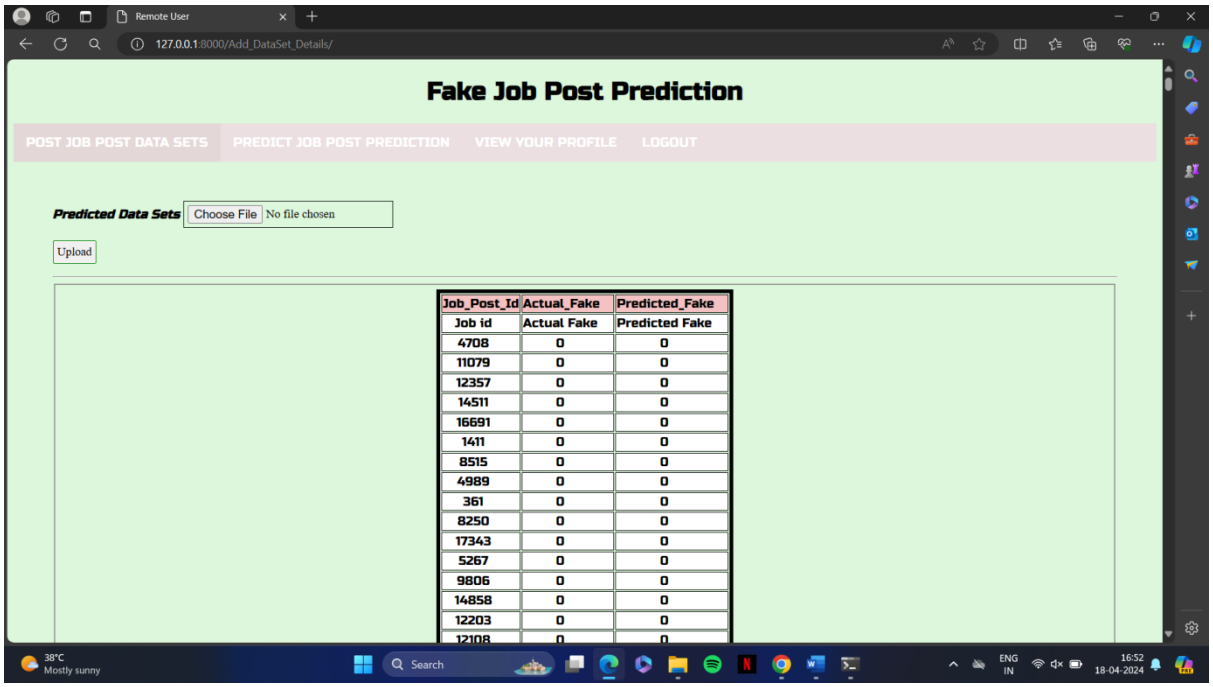
Screen 8.14 : Credentials

Select post job post data sets.



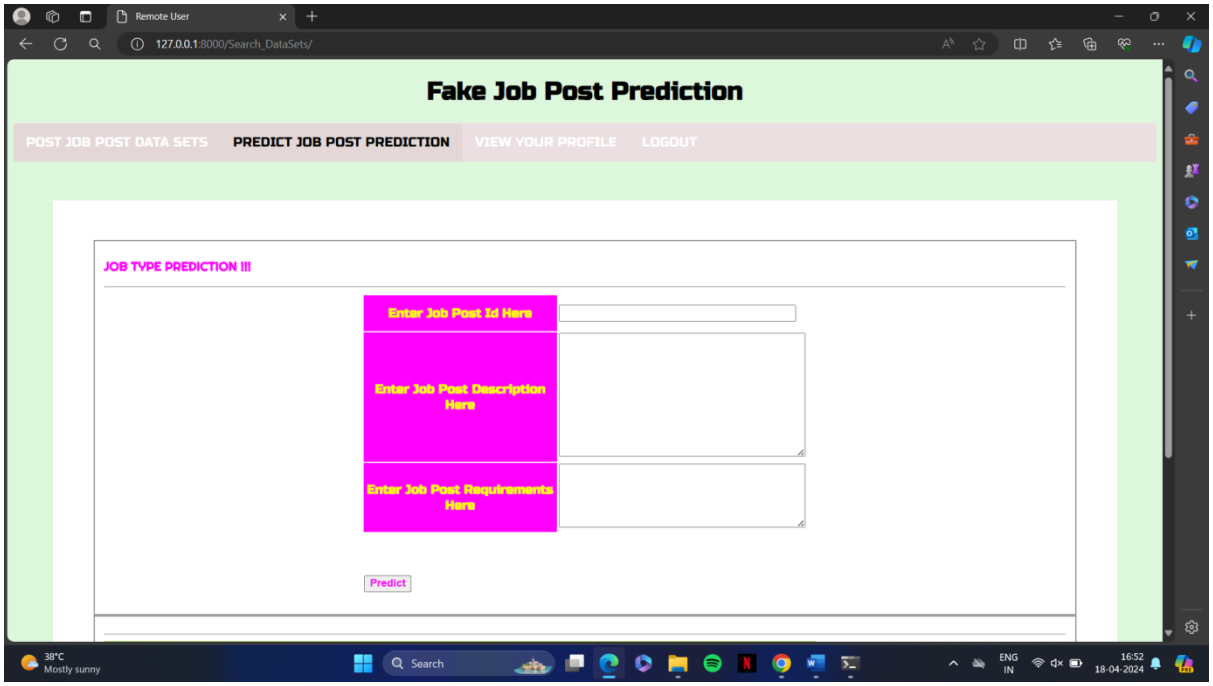
Screen 8.15 : Select post job post data sets

Upload the required files and the predictions are displayed.

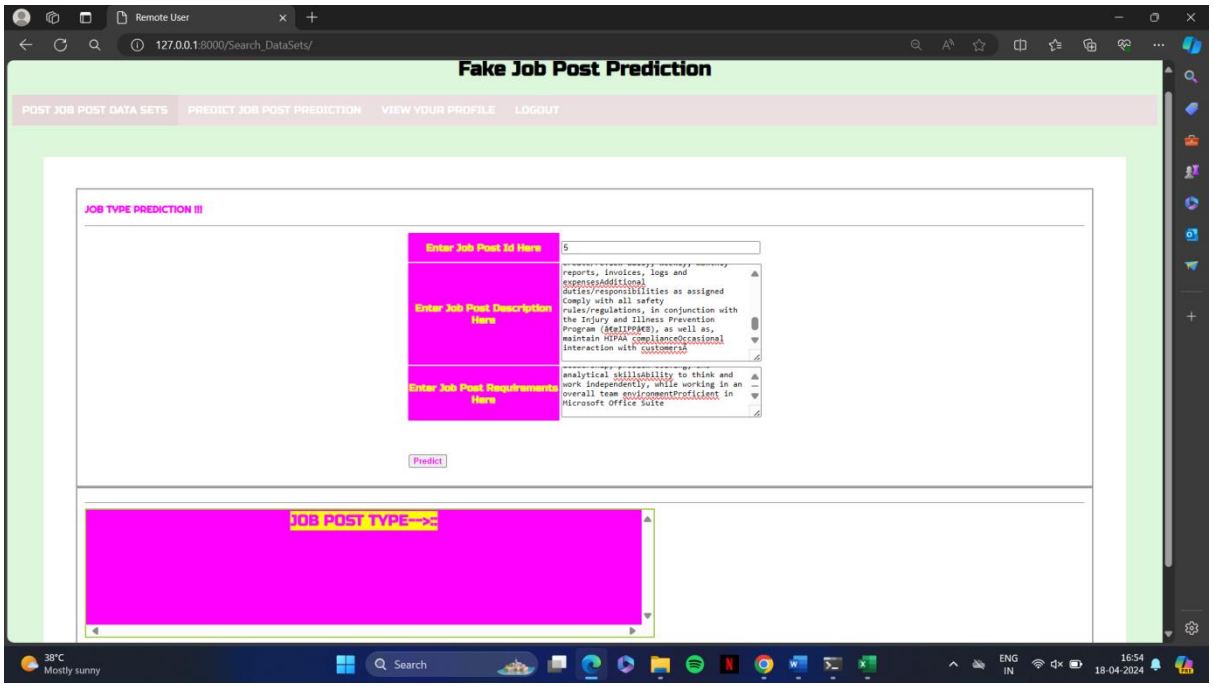


Screen 8.16 : upload file

Select predict job post prediction and enter the job post id, description and requirements. Click predict.

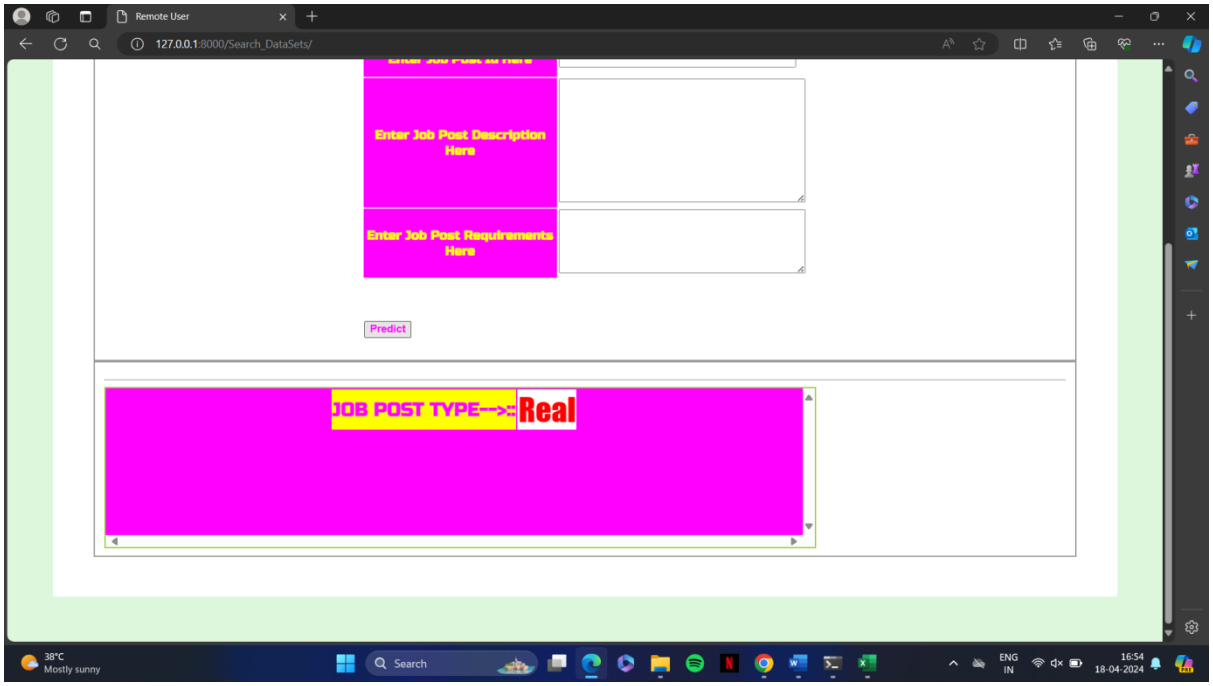


Screen 8.17 : Select predict job post prediction



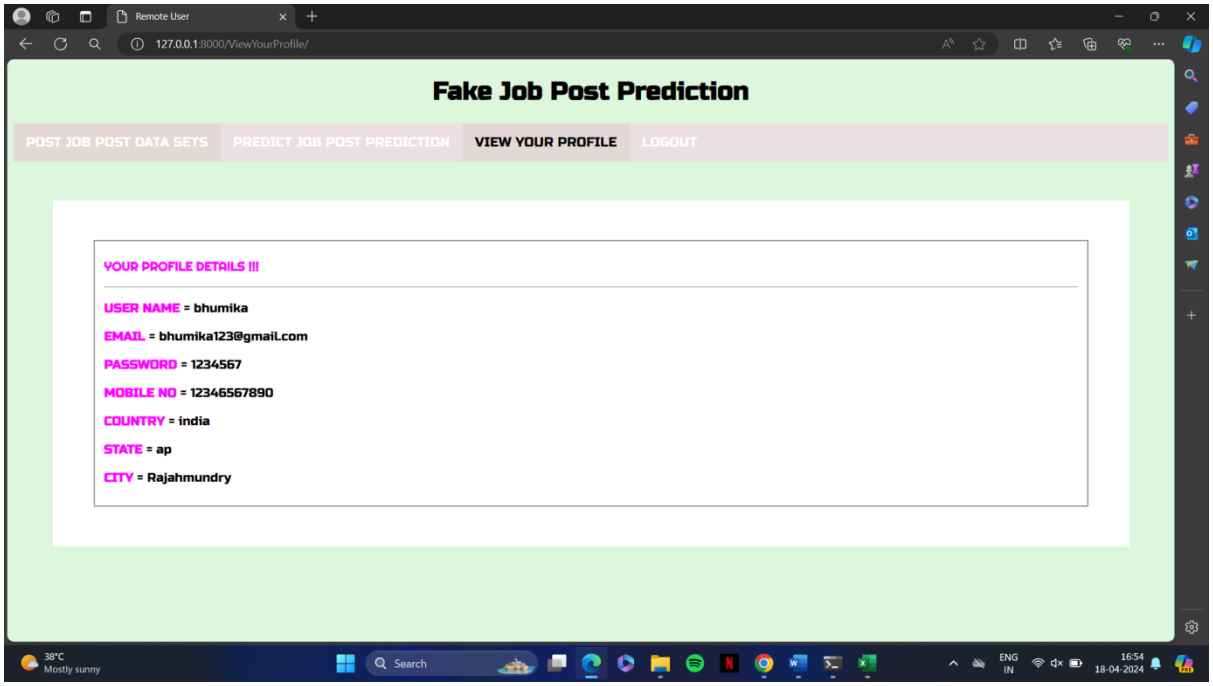
Screen 8.18 : execution

Then the result is displayed.



Screen 8.19 : Result

Displaying the profile of user who registered.



Screen 8.20 : Displaying the profiles

9. CONCLUSION and FUTURE SCOPE

9.1 Conclusion

In conclusion, our study focused on the critical task of predicting fake job postings using various machine learning techniques, namely KNN, Naive Bayes, Random Forest, and SVM. Through rigorous experimentation and analysis, we aimed to assess the efficacy of these methods in combating the pervasive issue of job scams.

Our findings reveal that each technique exhibited varying degrees of effectiveness in distinguishing between genuine and fraudulent job postings. Specifically, Random Forest emerged as the top performer among the traditional machine learning algorithms, showcasing superior classification accuracy. Its ability to handle complex datasets and mitigate overfitting contributed to its success in identifying fake job postings with remarkable precision.

Additionally, while KNN, Naive Bayes, and SVM demonstrated respectable performance, they fell short of matching the predictive power of Random Forest. Nonetheless, their

contributions remain valuable, particularly in scenarios where computational efficiency and interpretability are paramount.

Moving forward, the integration of these machine learning techniques into automated job screening systems holds immense promise for safeguarding job seekers against deceptive practices. By leveraging the strengths of each method and continuously refining their models through ongoing research and data collection, stakeholders can enhance the detection and prevention of fake job postings, thereby fostering a more secure and trustworthy online job marketplace.

9.2 FUTURE SCOPE

The future scope of fake job post prediction using machine learning techniques is promising and multifaceted. Here are several avenues for further exploration and development:

1. Enhanced Feature Engineering: Future research can delve deeper into identifying and extracting more informative features from job postings. This may involve incorporating textual analysis techniques, such as sentiment analysis or topic modeling, to capture nuances in language that indicate fraudulent intent.

2. Ensemble Methods: Exploring ensemble methods, such as stacking or boosting, could offer improvements in predictive accuracy by combining the strengths of multiple machine learning models. Ensemble techniques have shown effectiveness in various domains and could be applied to enhance fake job post detection as well.

3. Deep Learning Advancements: Continued advancements in deep learning architectures and techniques offer opportunities for improving the performance of fake job post prediction models. Investigating more complex neural network architectures, transfer learning approaches, or attention mechanisms could lead to better understanding and detection of fraudulent patterns.

4. Real-time Detection Systems: Developing real-time detection systems capable of identifying and flagging suspicious job postings as soon as they are posted would be invaluable for preventing job seekers from falling victim to scams. Integrating machine learning models into job platforms and recruitment websites could facilitate proactive monitoring and immediate action.

5. Data Collaboration and Standardization: Collaboration among researchers, industry stakeholders, and government agencies to share data and insights can lead to the creation of standardized datasets and benchmarks for evaluating fake job post prediction models. This would enable more comprehensive comparisons and advancements in the field.

6. User Feedback Integration: Incorporating user feedback mechanisms into detection systems can help improve model performance over time. Leveraging crowdsourced annotations or user-reported incidents of fake job postings can provide valuable training data for refining machine learning models.

7. Ethical and Fairness Considerations: As with any predictive model, it's crucial to address ethical concerns and ensure fairness in decision-making processes. Future research should focus on developing methods to mitigate bias and discrimination in fake job post prediction models, thereby promoting equitable outcomes for all job seekers.

By pursuing these avenues of research and innovation, the field of fake job post prediction using machine learning techniques can continue to evolve, providing increasingly effective solutions to combat online job scams and protect job seekers worldwide.

10.BIBLIOGRAPHY

10.1 Reference:

[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 NeighbourClassifiers,‖ 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.

[11]. N.Hussain, H.T.Mirza, G.Rasool, I.Hussain, and M.Kaleem, Spam review detection techniques: A systematic

literature review,‖ Appl. Sci., vol. 9, no. 5, pp. 1–26, 2019, doi: 10.3390/app9050987.

[12]. Mandru, D.B., Aruna Safali, M., Raghavendra Sai, N., Sai Chaitanya Kumar, G. (2022). Assessing Deep Neural

Network and Shallow for Network Intrusion Detection Systems in Cyber Security. In: Smys, S., Bestak, R.,

Palanisamy, R., Kotuliak, I. (eds) Computer Networks and Inventive Communication Technologies . Lecture

Notes on Data Engineering and Communications Technologies, vol 75. Springer, Singapore.

https://doi.org/10.1007/978-981-16-3728-5_52

[13]. Mandru, Deena Babu. "An Improved K-Means Algorithm for Web Page Clustering."

[14]. Krishna, Paruchuri Jeevan, M. Deena Babu, and G. Manoj Someswar. "Design & Development of an

improvised PACK System using TRE Technique for Cloud Computing Users." International Journal of

Research 3.2 (2016): 384-393.

[15]. Rakesh, Ganya, M. D. Babu, and G. Manoj Someswar. "A Novel Integrated Attestation Graph Analysis Scheme

for Enhancing Result Quality and Higher Attacker Pinpointing Accuracy." International Journal of

Research 3.2 (2016): 214-225.

[16]. Mandru, Deena Babu, and Y. K. Krishna. "Enhanced Cluster Ensemble Approach Using Multiple Attributes

in Unreliable Categorical Data." International Journal of Psychosocial Rehabilitation 23.1 (2019).

[17]. Mandru, Deena Babu, and Y. S. Krishna. "Multi-view Cluster Approach to Explore Multi-Objective Attributes

based on Similarity Measure for High Dimensional Data." International Journal of Applied Engineering

Research 13.15 (2018): 12289-12295.