International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 04 | April - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Supervised ML Approach to Fake Job Post Detection

K. Aswani[1], M. Gayathri[2], K. Arun Babu[3], K. Sri Vijaya[4]

Department of Information Technology[1,2,3,4]

Prasad V Potluri Siddhartha Institute of Technology[1,2,3,4]

Abstract - The significance of online job platforms has expanded considerably over time. However, along with technological advancements, there has been a rise in fraudulent job postings that aim to exploit job seekers. These fake job listings often mislead users by offering unrealistic opportunities, requesting personal information, or demanding money, which may lead to financial loss and data misuse. Such fraudulent postings pose a serious threat to users by spreading misleading and harmful content. Therefore, there is a strong need for an efficient and reliable detection system. This project addresses the problem of identifying fake job postings by analyzing job-related textual data using machine learning techniques. Various supervised learning algorithms, including Random Forest, Decision Tree, Logistic Regression, and Naive Bayes, along with TF-IDF-based feature extraction, are utilized to classify job postings as real or fake.

Key Words: Fake Job Detection, Machine Learning, Random Forest, Naive Bayes, Logistic Regression, TF- IDF, NLP

1. INTRODUCTION

In today's digital world, online job portals are widely used by job seekers. However, the increase in online job postings has also led to a rise in fake and fraudulent job advertisements. These fake jobs can mislead users, causing financial loss and misuse of personal information, making it important to identify them effectively.

The **Fake Job Post Detection System** is a machine learning-based application that helps classify job postings as real or fake. It analyzes job descriptions or job links using text processing techniques and trained models to predict authenticity. The system provides a simple interface using Streamlit, allowing users to easily check job postings and avoid potential scams.

1.2 EXISTING SYSTEM

In the existing system, job seekers rely on online job portals and social media to find opportunities, but there is no proper way to verify if a job is real or fake. Users depend on manual judgment, which can be unreliable, leading to risks like financial loss and misuse of personal data. Additionally, the absence of automated tools to detect suspicious patterns makes it difficult to identify fraudulent job postings an increases the chances of trusting unverified job offers.

1.2.1 Disadvantages of Existing System

- No proper system to verify whether a job posting is real or fake.
- Depends on manual judgment, which is not always accurate.
- High risk of financial loss due to fraudulent job offers.
- Possibility of personal data theft and misuse.
- Time-consuming process to check job authenticity.
- Lack of automated tools to detect fake job postings.
- Users may trust unverified sources easily.

2. PROPOSED SYSTEM

The proposed system is a machine learning-based application designed to automatically detect whether a job posting is real or fake. It analyzes job descriptions and job links using text processing techniques and trained models to classify the authenticity of the job.

The system uses methods like TF-IDF vectorization and classification algorithms to make accurate predictions. It also provides an interactive interface using Streamlit, allowing users to easily input job details and view results. This helps users quickly identify fraudulent job postings and make safer decisions.

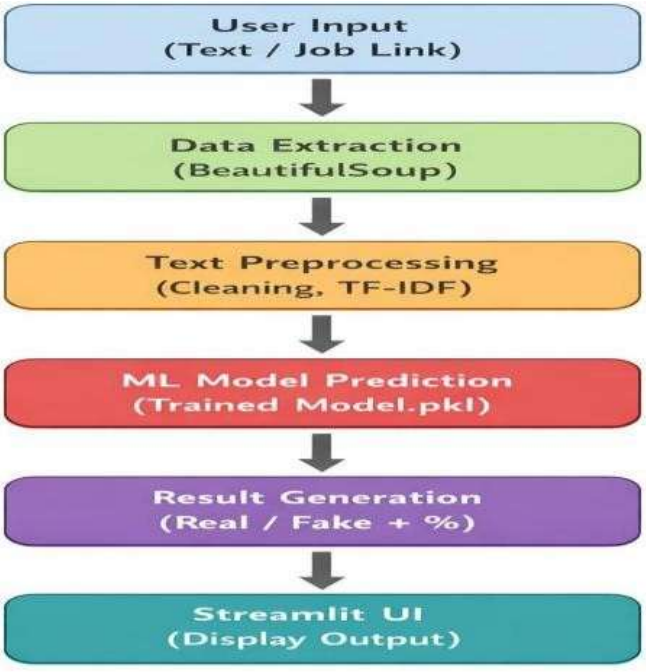


Fig -1: Proposed system model

2.1 Dataset Description

The dataset used in this project is the **Fake Job Postings Dataset**, which contains a total of **17,880 job postings** collected from online job platforms. Out of these, approximately **17,014 are real job postings (label 0)** and **866 are fake or fraudulent job postings (label 1)**.

The dataset consists of several important columns such as **job_id, title, location, department, salary_range, company_profile, description, requirements, benefits, telecommuting, has_company_logo, has_questions, employment_type, required_experience, required_education, industry, function, and fraudulent (target column)**. In this project, the main focus is on **title and description columns**, which are combined and converted into numerical features using TF-IDF for training the machine learning model.

This dataset helps in training the model to effectively distinguish between real and fake job postings based on textual patterns and features.

- **Random Forest :**

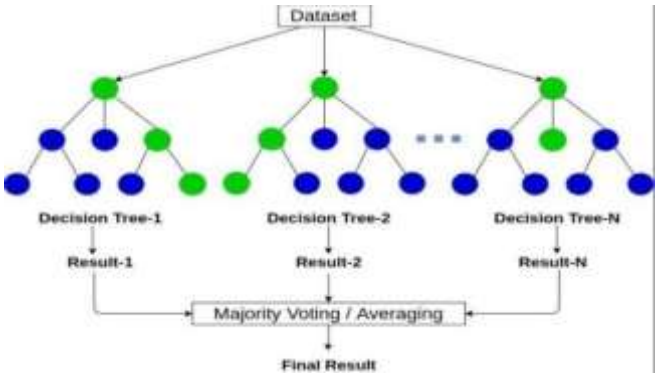


Fig -2: Implementation of Random forest

In this, the dataset is passed through multiple decision trees, where each tree gives its own prediction (real or fake job). Finally, all the results are combined using **majority voting**, and the final prediction is generated. This helps improve accuracy and gives a more reliable result for detecting fake job postings.

- **Decision Tree**

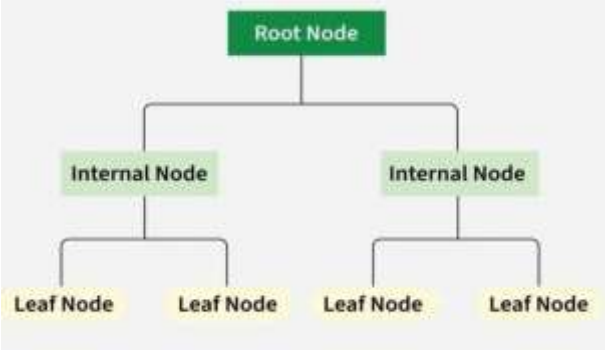


Fig -3: Implementation of Decision Tree

This diagram represents the Decision Tree algorithm used in your project. It works by splitting the data based on different conditions and features of the job description, forming a tree-like structure. Each branch represents a decision, and the final leaf node gives the prediction as real or fake. This helps in understanding how the model makes decisions step by step for classifying job postings.

- **Logistic Regression**

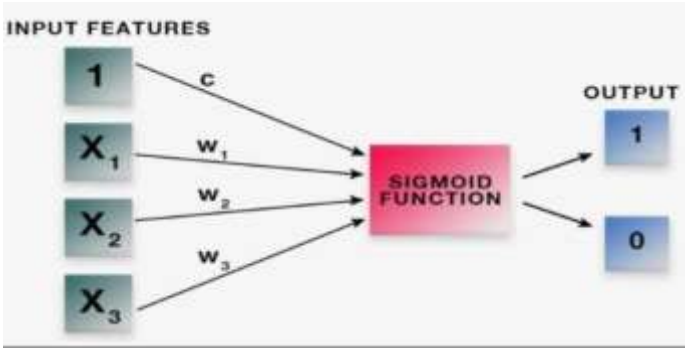


Fig -4: Implementation of Logistic Regression

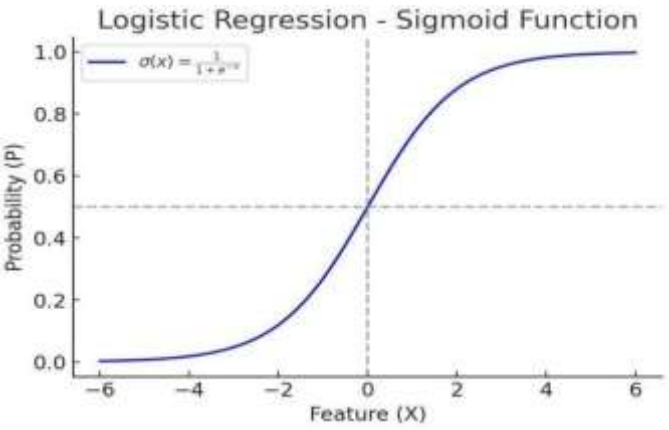


Fig -5: Implementation of Logistic Regression

These diagrams represent the **Logistic Regression algorithm** used in your project. The input features (from job description text after TF-IDF) are combined and passed through a **sigmoid function**, which converts them into a probability value between 0 and 1. Based on this probability, the model classifies the job posting as **real (0)** or **fake (1)**. This helps in making accurate binary predictions for detecting fraudulent job postings.

- **Naive Bayes**

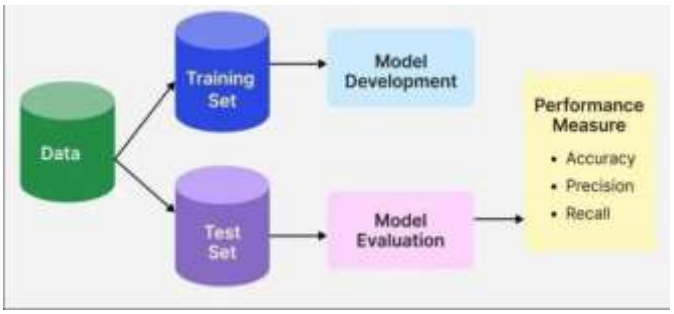


Fig -6: Implementation of Naive Bayes

This diagram shows the process of training and evaluating the machine learning model in the project. The dataset is divided into **training set** and **test set**. The training set is used to build the model, while the test set is used to evaluate its performance. Metrics like **accuracy, precision, and recall** are used to measure how well the model detects real and fake job postings.

- **TF-IDF (Feature Extraction - NLP)**

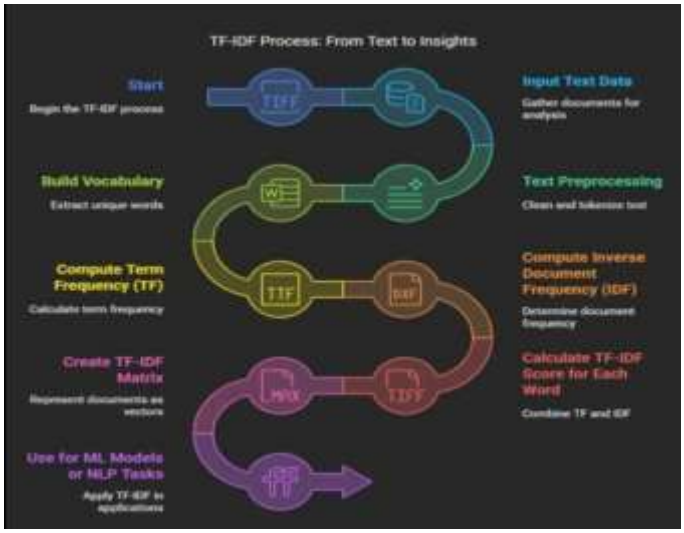


Fig -7: Implementation of TF-IDF (Feature Extraction - NLP)

TF-IDF (Term Frequency–Inverse Document Frequency) is a widely used Natural Language Processing technique for converting textual data into numerical form. It measures the importance of a word in a document relative to a collection of documents. Term Frequency (TF) calculates how often a word appears in a document, while Inverse Document Frequency (IDF) reduces the importance of commonly occurring words across all documents. By combining these two measures, TF-IDF assigns higher weights to important and unique words.

3. TECHNOLOGIES USED

3.1 Google Colab

Google Colab is used as the main platform for developing and training the machine learning model. It provides an online environment to write and execute Python code without installation. It also supports easy dataset uploading, visualization, and model training, making it suitable for building the fake job detection system.

3.2 PYTHON LIBRARIES

3.2.1 NumPy

NumPy is used for performing numerical operations and handling arrays efficiently during data processing.

3.2.2 Pandas

Pandas is used for data handling, cleaning, and preprocessing. It helps in loading the dataset and combining job title and description.

3.2.3 Scikit-learn (Sklearn)

Scikit-learn is used to implement machine learning algorithms like Logistic Regression, Decision Tree, and Random Forest. It also provides evaluation metrics.

3.2.4 BeautifulSoup & Requests

These libraries are used for extracting job descriptions from job links through web scraping.

3.2.5 Matplotlib & Seaborn

Used for visualizing data through graphs like confusion matrix and performance charts.

3.2.6 Streamlit

Streamlit is used to build the interactive web interface for the project, allowing users to check job postings easily.

3.2.7 Pickle

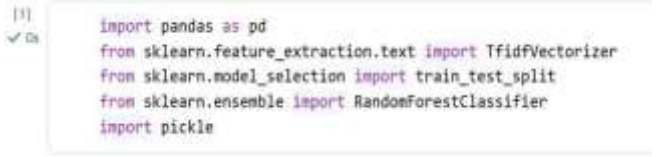
Pickle is used to save and load the trained machine learning model (model.pkl) and vectorizer (vectorizer.pkl) for prediction.

4. RESULTS

4.1 Installing Required Packages

This step installs the required Python libraries such as **Pandas** and **Scikit-learn**. These libraries are essential for data handling, preprocessing, and building machine learning models.

4.2 Importing Required Libraries



```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
import pickle
```

In this step, necessary libraries are imported:

- **Pandas** → for handling dataset
- **TF-IDF Vectorizer** → for text feature extraction
- **Train-Test Split** → for splitting data
- **Random Forest** → for classification
- **Pickle** → for saving trained model

4.3 Uploading Dataset

This step allows the user to upload the dataset file (CSV format) into Google Colab for processing.

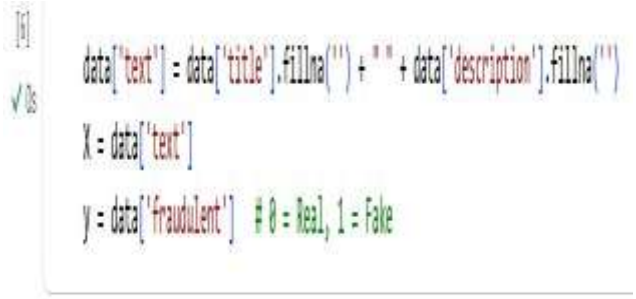
4.4 Loading Dataset

```
data = pd.read_csv("Fake_Job_postings.csv")
data.head()
```

	job_id	title	location	department	salary_range	company_profile	description	
0	1	Marketing Intern	US, NY, New York	Marketing	NaN	We're Food52, and we've created a groundbreaki...	Food52, a fast-growing, James Beard Award-winn...	Experien manage
1	2	Customer Service - Cloud Video Production	NZ, , Auckland	Success	NaN	90 Seconds, the worlds Cloud Video Production ...	Organised - Focused - Vibrant - Awesome!Do you...	What

The dataset is loaded using Pandas. The `head()` function displays the first few rows to understand the structure of the data.

4.5 Data Preprocessing & Feature Selection



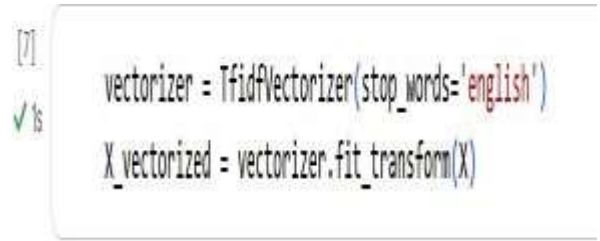
```
data['text'] = data['title'].fillna('') + ' ' + data['description'].fillna('')
X = data['text']
y = data['fraudulent']  # 0 = Real, 1 = Fake
```

- Missing values are handled using `fillna()`
- Job title and description are combined into a single text column

This improves model understanding of job content.

- **X (Input)** → Job text data
- **y (Output)** → Labels (Real or Fake job)

4.6 Feature Extraction using TF-IDF



```
vectorizer = TfidfVectorizer(stop_words='english')
X_vectorized = vectorizer.fit_transform(X)
```

TF-IDF converts text into numerical vectors so that machine learning models can process the data effectively.

4.7 Training Multiple Machine Learning Models

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import MultinomialNB

rf = RandomForestClassifier()
lr = LogisticRegression()
dt = DecisionTreeClassifier()
nb = MultinomialNB()

rf.fit(X_vectorized, y)
lr.fit(X_vectorized, y)
dt.fit(X_vectorized, y)
nb.fit(X_vectorized, y)
```

```
MultinomialNB()
```

Different machine learning algorithms such as Random Forest, Logistic Regression, Decision Tree, and Naive Bayes are trained on the dataset. This helps in comparing their performance and selecting the best model.

4.8 Model Evaluation and Comparison

```
from sklearn.metrics import accuracy_score

print("Random Forest:", accuracy_score(y, rf.predict(X_vectorized)))
print("Logistic Regression:", accuracy_score(y, lr.predict(X_vectorized)))
print("Decision Tree:", accuracy_score(y, dt.predict(X_vectorized)))
print("Naive Bayes:", accuracy_score(y, nb.predict(X_vectorized)))
```

```
Random Forest: 1.0
Logistic Regression: 0.9707494407158836
Decision Tree: 1.0
Naive Bayes: 0.9582237136465325
```

```
model = rf  # choose best one
```

The trained models are evaluated using accuracy score to determine which model performs best on the dataset. This step is important for selecting the most suitable algorithm.

4.9 Model Saving (Pickle Files)

```
pickle.dump(model, open("model.pkl", "wb"))
```

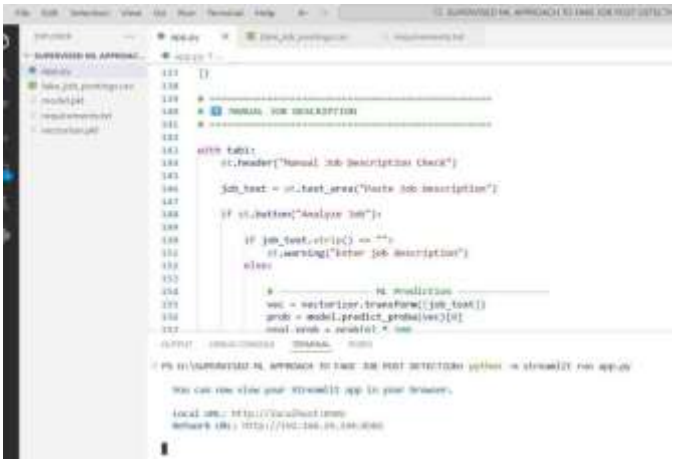
```
pickle.dump(vectorizer, open("vectorizer.pkl", "wb"))
```

The trained model and vectorizer are saved as `.pkl` files. These files are later used for prediction without retraining.

4.10 Downloading Model Files

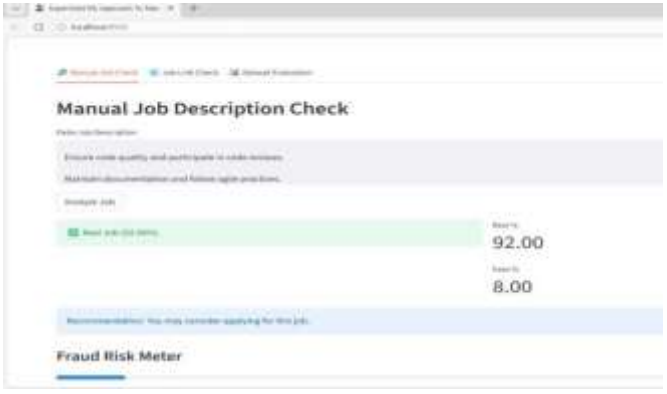
This step downloads the trained model and vectorizer to your system for further in application.

4.11 Execution of Streamlit Application



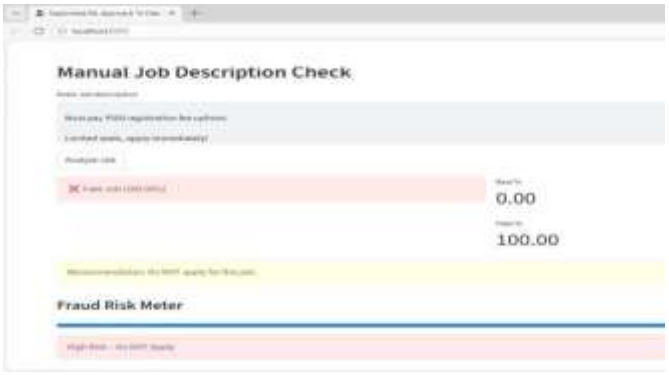
The figure shows the successful execution of the Streamlit-based Fake Job Detection System in VS Code. The application is run using the command `streamlit run app.py`, which generates a local URL to access the web interface. After running, the system loads the trained model and allows users to analyze job descriptions, job links, and datasets, confirming that the application is working correctly.

4.12 Real Job Description Result



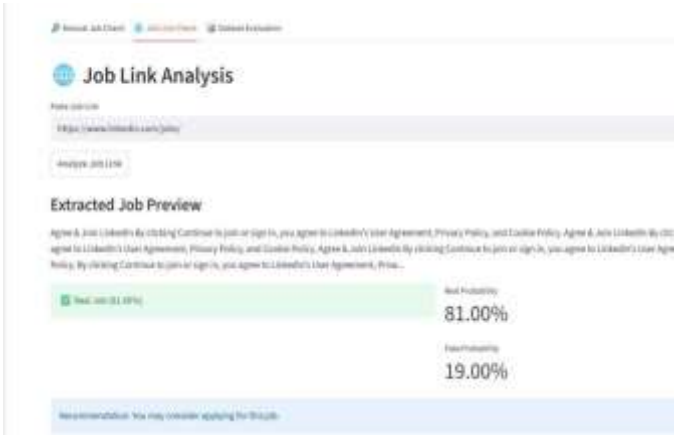
The system analyzes a normal job description and correctly classifies it as **Real Job (92%)**. It shows high confidence with low fake probability and recommends that the user can safely apply for the job.

4.13 Fake Job Description Result



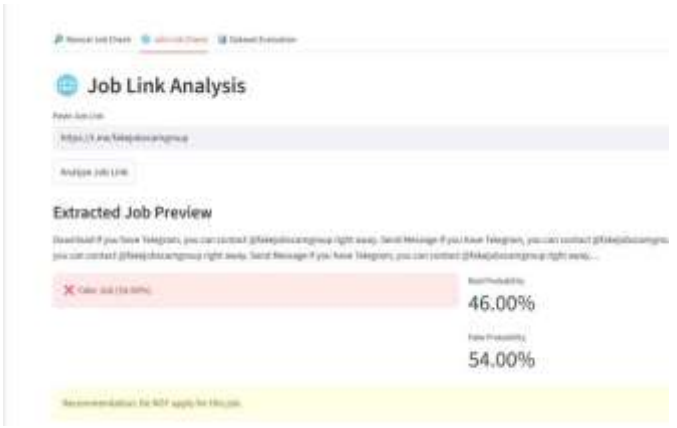
The system detects suspicious content like registration fees and urgency, classifying it as **Fake Job (100%)**. It gives a strong warning and recommends **not to apply**.

4.14 Real Job Link Results



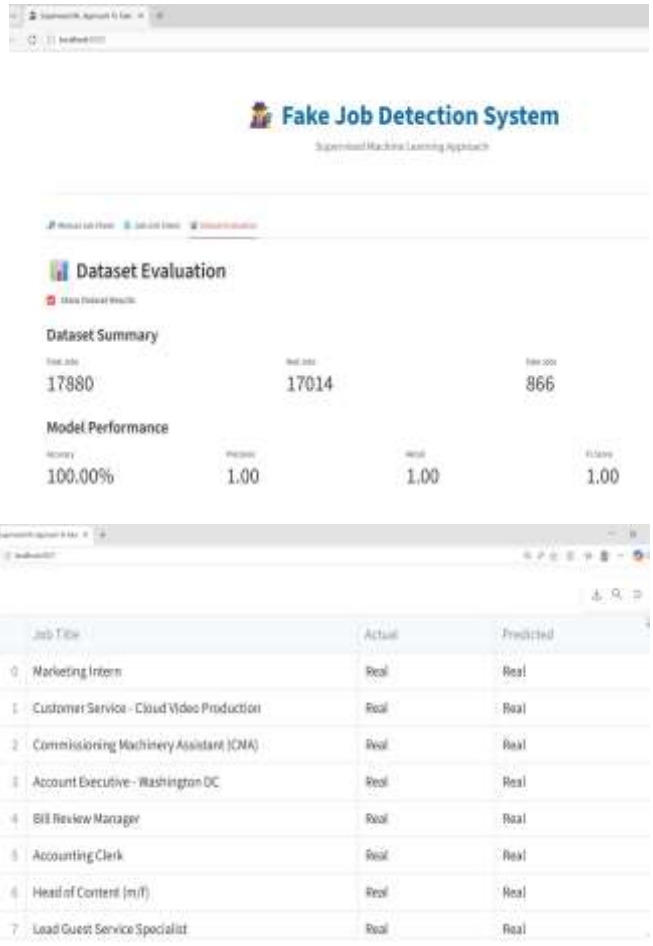
The system analyzes a job link from LinkedIn and classifies it as a **Real Job (81%)**. It shows a higher real probability compared to fake, indicating that the job is likely genuine. Since LinkedIn is a trusted platform, the system considers it safe but with moderate confidence. Hence, it recommends that users can apply, but should still verify job details before proceeding.

4.15 Fake Job Link Results

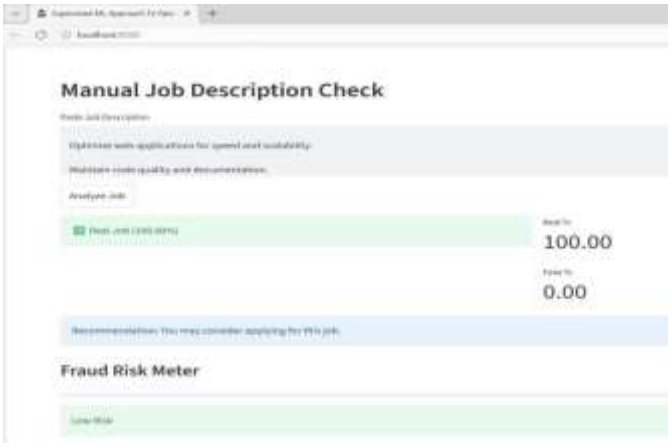


The system analyzes a Telegram job link and identifies it as **Fake (54%)**. It detects risky patterns like external messaging and repeated contact prompts, advising users to avoid such links.

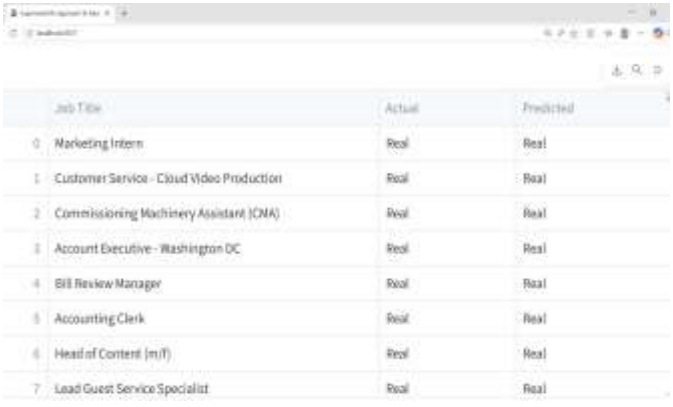
4.16 Dataset Evaluation & Prediction Table



The dataset contains **17,880 jobs**, with most being real and a smaller portion fake. The model achieves **100% accuracy, precision, recall, and F1-score**, showing excellent performance in detecting fake jobs.



5. OUTPUTS



6. FUTURE SCOPE

The future scope of this project focuses on improving the accuracy of the Fake Job Detection System by using larger and more diverse datasets. Additional features like company verification, salary patterns, and user feedback can be included to enhance the system's ability to detect fake job postings. Advanced NLP techniques can also be applied to better understand job descriptions.

The system can be further developed by using deep learning models and real-time job data from online portals. It can also be extended as a browser extension or mobile application for easy access. Adding a feedback system and supporting multiple languages will make the application more effective and useful for a wider range of users.

7. CONCLUSION

In this project, a supervised machine learning approach is used to detect fake job postings. Algorithms like Random Forest, Decision Tree, Logistic Regression, and Naive Bayes are applied, and Random Forest gives the best performance. The system uses TF-IDF and basic text processing to analyze job descriptions and classify them as real or fake.

The Streamlit application provides a simple interface for users to check job postings easily. Overall, the system helps in reducing job fraud and protecting users from scams, showing how machine learning can be used to solve real-world problems effectively.

ACKNOWLEDGEMENT

We express our sincere gratitude to our guide **Ms. K. Sri Vijaya, M.Tech., (Ph.D)**, Assistant Professor, Department of Information Technology, for her continuous guidance, valuable suggestions, and constant support throughout the development of this project. Her encouragement and insights played a significant role in the successful completion of this work.

We also extend our sincere thanks to the **Head of the Department, Dr. B. V. Subba Rao, M.Tech., Ph.D**, for his support and motivation. We are grateful to all the faculty members of the Department of Information Technology for their assistance and encouragement during this project.

Finally, we would like to thank our institution for providing the necessary facilities and a supportive environment to carry out this work successfully.

We also acknowledge the use of various research papers, online resources, and datasets that supported the successful completion of this work.

REFERENCES

1. Fake Job Post Detection Using Machine Learning, IEEE Conference/Journal Publication, Year. This paper proposes a machine learning-based approach to classify fraudulent job advertisements using supervised learning techniques and text preprocessing methods. Available on: https://ieeexplore.ieee.org/
2. R. R. Bhamare et al., "Employment Scam Detection Using Machine Learning Techniques," IEEE Xplore, 2020–2022. This research addresses employment fraud detection using classification models and structured feature analysis. Available on: https://ieeexplore.ieee.org/
3. V. Itnal, I. Pande, S. Nimkar, A. Padwal, A. Pampattiwar, and A. Patil, "Fake/Real Job Posting Detection Using Machine Learning," International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2025. DOI: https://doi.org/10.22214/ijraset.2025.74635
4. P. Khandagale, A. Utekar, A. Dhonde, and S. S. Karve, "Fake Job Detection Using Machine Learning," IJRASET, 2022. DOI: https://doi.org/10.22214/ijraset.2022.41641
5. K. Sridevi, G. Likitha, P. Chandana, and S. Shamarthi, "Real or Fake Job Posting Detection," International Research Journal on Advanced Engineering and Management (IRJAEM), 2024. DOI: https://doi.org/10.47392/IRJAEM.2024.0361
6. K. G., P. M., R. G. Naik, R. K. R., and S. A. Sharma, "Detection of Fake Job Listings Using Text Classification and SMOTE-Enhanced Training," International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), 2025. DOI: https://doi.org/10.17148/IJARCCE.2025.141063
7. J. Fating, J. Tumdam, A. Raut, A. Ladke, and A. Shewale, "Fake Job Listing Detection Using Machine Learning Approach," International Journal of Engineering Trends and Technology (IJETT), 2023. Available at: https://ijettjournal.org/
8. Kaggle Dataset – "Fake Job Postings Dataset." This dataset contains real and fraudulent job advertisements used for training machine learning models in classification tasks. Available at: https://www.kaggle.com/shivamb/real- or-fake-fake-jobposting-prediction