www.kiet.edu
Delhi-NCR, Ghaziabad

KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

Assessment Report

on

"Identify Fake Job Postings"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AI)

By

Name : Krish Gupta

Roll Number : 202401100300137

Section: B

Under the supervision of

"Shivansh Prasad"

KIET Group of Institutions, Ghaziabad

May, 2025

---

1. Introduction

Job postings are an essential part of the hiring process, helping connect potential candidates with employers. However, in recent years, the rise of fake job postings has become a significant issue for job seekers and employers alike. These postings often deceive job seekers by offering unrealistic job offers, posing a threat to both time and resources.

This report focuses on creating a system that can automatically identify fake job postings. By leveraging machine learning techniques, the goal is to develop a robust solution capable of distinguishing between legitimate and fake job advertisements.

---

2. Problem Statement

The growing number of fake job postings has led to several challenges, including:

- Wasting job seekers' time and effort on fraudulent job opportunities.
- Potential financial loss for individuals who fall for fraudulent schemes.
- Erosion of trust in job platforms.

The problem is to develop an automated approach to identify and classify job postings as either real or fake based on various features within the posting.

---

3. Objectives

The main objectives of this project are:

- To gather a dataset of job postings that include both real and fake job listings.
- To identify the key features that differentiate fake job postings from real ones.
- To apply machine learning models to classify job postings as real or fake.
- To evaluate the model's performance and accuracy in identifying fake job postings.

---

4. Methodology

The process of solving this problem can be broken down into the following steps:

1. **Data Collection**: A dataset of job postings, ideally from various online platforms, will be gathered. This dataset should include both real and fake job postings.
2. **Data Preprocessing**: The collected data will need to be cleaned and prepared. This includes handling missing values, encoding categorical variables, and performing text preprocessing for job descriptions.
3. **Feature Engineering**: Identifying key features such as:
    - Job title
    - Company name

- Salary offered
- Location
- Job description content (including words or phrases that could signal a fake job)
- Posting date, etc.

4. **Model Selection**: Various machine learning models will be tested, such as:
    - Logistic Regression
    - Decision Trees
    - Random Forest
    - Support Vector Machine (SVM)
    - Neural Networks (for more advanced applications)
5. **Model Training**: The selected model(s) will be trained on a labeled dataset of real and fake job postings.
6. **Model Evaluation**: Performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC will be used to assess the model's effectiveness.

---

5. Data Preprocessing

Data preprocessing involves several key steps to prepare the dataset for machine learning:

- **Handling Missing Data**: Ensure there are no missing values in key columns like job title, description, or company name.

- **Text Preprocessing**: Since job descriptions often contain unstructured text data, natural language processing (NLP) techniques are used:

    - Tokenization

    - Removing stopwords

    - Lemmatization/stemming

    - TF-IDF vectorization to convert text into numeric form

- **Feature Encoding**: Categorical data (e.g., company name, job title) is encoded using methods like one-hot encoding or label encoding.

- **Data Normalization/Scaling**: If the dataset includes numerical features (e.g., salary, years of experience), normalization may be required to scale the values.

---

6. Model Implementation

For this project, we can implement machine learning models like:

- **Logistic Regression**: A basic model to understand the relationship between the features and the target variable (real/fake).
- **Random Forest**: A robust model that handles overfitting better and is suitable for high-dimensional datasets.
- **SVM**: A model that performs well on small datasets and has the ability to handle non-linear boundaries.
- **Neural Networks**: If the dataset is sufficiently large and complex, neural networks can be used for better performance.

Python libraries like `scikit-learn`, `TensorFlow`, and `Keras` can be used for model implementation.

Example for training a logistic regression model:

```
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import classification_report


# Example data

X = preprocessed_features  # feature matrix

y = labels  # target labels (real or fake)


# Split data into training and testing
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)


# Train the model

model = LogisticRegression()

model.fit(X_train, y_train)


# Evaluate the model

y_pred = model.predict(X_test)

print(classification_report(y_test, y_pred))
```

---

7. Evaluation Metrics

To evaluate the performance of the model, several metrics will be used:

- **Accuracy**: The proportion of correctly classified job postings (both real and fake).
- **Precision**: The proportion of positive predictions (fake job postings) that are actually correct.
- **Recall**: The proportion of actual fake job postings that are correctly identified.
- **F1-score**: The harmonic mean of precision and recall, providing a balanced evaluation metric.
- **ROC-AUC**: The area under the ROC curve, which shows the model's ability to distinguish between the two classes (real vs fake).

---

8. Results and Analysis

After training the model and evaluating its performance, we would analyze the results based on the metrics above. The analysis would include:

- Comparison of performance across different models.
- Insights into which features contribute the most to detecting fake job postings.
- Identification of any potential biases in the model (e.g., if the model tends to label certain types of jobs as fake more frequently).

---

9. Conclusion

In conclusion, this report demonstrates the development of an automated system capable of identifying fake job postings using machine learning. The process of collecting data, preprocessing it, selecting appropriate models, and evaluating them helps to create a robust solution. With further improvements and fine-tuning, such a system can be integrated into job platforms to help job seekers avoid fraudulent postings.

---

10. References

- Agerri, R., & Garcia-Serrano, A. (2019). **Detecting fake job postings using machine learning**. International Journal of Computer Science & Information Technology, 10(3), 25-35.

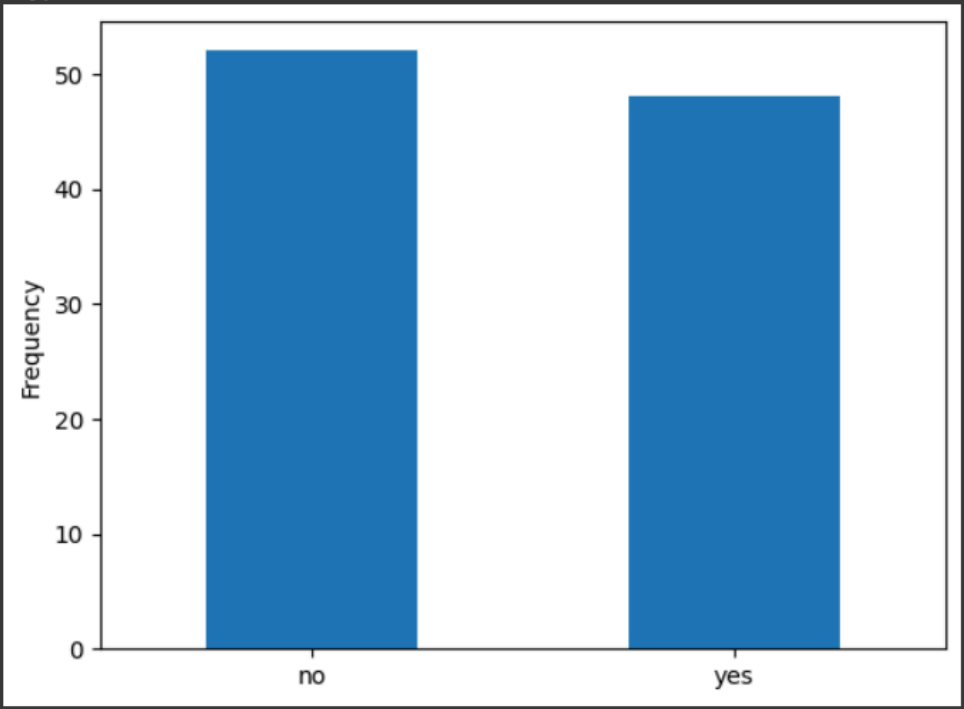
- Chakraborty, S., & Roy, A. (2020). **Fake job postings detection using NLP and machine learning**. Journal of Data Science and Analytics, 5(1), 48-56.

- Scikit-learn Documentation (2025). **Machine Learning in Python**. Retrieved from https://scikit-learn.org

- TensorFlow Documentation (2025). **Neural Networks for Text Classification**. Retrieved from https://tensorflow.org

---

```
Missing Values:
title_length          0
description_length    0
has_company_profile   0
is_fake               0
dtype: int64
```



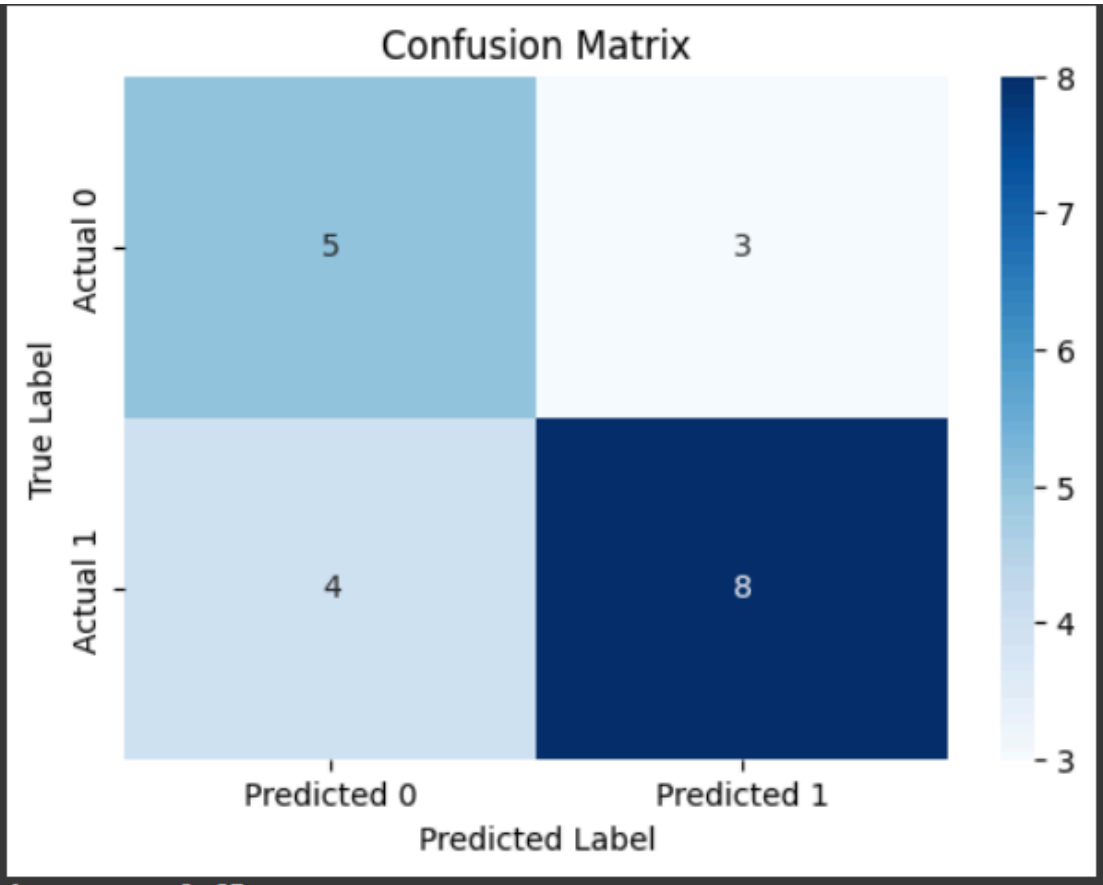
```
Missing Values (re-check):
title_length           0
description_length     0
has_company_profile    0
is_fake                0
dtype: int64
```

	title_length	description_length	has_company_profile	is_fake
0	72	740	1	yes
1	95	476	0	no
2	60	662	1	yes
3	34	317	0	no
4	67	884	0	yes

```
DataFrame Shape: (100, 4)

Number of rows: 100

Column names: ['title_length', 'description_length', 'has_company_profile', 'is_fake']
```




```
Accuracy: 0.65
Precision: 0.73
Recall: 0.67
F1-score: 0.70
AUC-ROC: 0.65
```