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 : Ayush Pandey

Roll Number : 202401100300085

Section: B

Under the supervision of

“Shivansh Prasad”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

With the increasing number of job seekers turning to online platforms, fake job postings have become a growing concern. These deceptive listings can lead to financial fraud and identity theft. This project focuses on detecting such fake job postings using machine learning techniques to assist job platforms in filtering malicious content and ensuring a safer experience for users.

---

2. Problem Statement

To classify job postings as real or fake based on text-based features such as title length, description length, and whether a company profile exists.

---

3. Objectives

- Preprocess and analyse the dataset.
- Train a machine learning classification model.
- Evaluate model performance using accuracy, precision, and recall.
- Visualize model performance using a confusion matrix heatmap.

---

4. Methodology

Data Collection:
The dataset used contains 100 job postings with labeled outcomes ("yes" for fake, "no" for real) and features like title length, description length, and presence of a company profile.

Data Preprocessing:

- **The target label is_fake was encoded into binary format (1 = fake, 0 = real).**
- **Features were selected directly without need for missing value imputation or scaling, as they are already numeric and clean.**

Model Building:

- **The dataset was split into training and testing sets (80%-20%).**
- **A Random Forest Classifier was trained on the training data.**

Model Evaluation:

- **Metrics such as Accuracy, Precision, and Recall were calculated.**
- **A confusion matrix was generated and visualized using a heatmap for better interpretability.**

---

5. Data Preprocessing

- **Label encoding was used to convert the categorical target variable (is_fake) to numeric.**
- **The dataset was already clean and had no missing values.**
- **No scaling was needed as tree-based models like Random Forest are not sensitive to feature magnitude.**
- **The data was split into 80% training and 20% testing.**

6. Model Implementation

A **Random Forest Classifier** was chosen for its robustness and ability to handle small datasets efficiently. The model was trained on features like:

- title_length
- description_length
- has_company_profile

These features helped the model learn patterns associated with fake versus real job postings.

---

7. Evaluation Metrics

The model was evaluated using the following metrics:

- **Accuracy**: 50% — Indicates how often the model was correct.
- **Precision**: 62.5% — Of all jobs predicted as fake, 62.5% were actually fake.
- **Recall**: 41.7% — Of all actual fake postings, only 41.7% were correctly identified.

A **confusion matrix** was also created to visualize how well the model performed in predicting fake vs real postings.

---

8. Results and Analysis

The model showed moderate performance with:

- A higher precision, meaning fewer false alarms.
- A lower recall, meaning many fake postings were missed.
- The confusion matrix helped analyze the trade-offs between false positives and false negatives.

Although basic, the model shows promise and can be enhanced using more advanced NLP-based features like TF-IDF, word embeddings, or deep learning models.

---

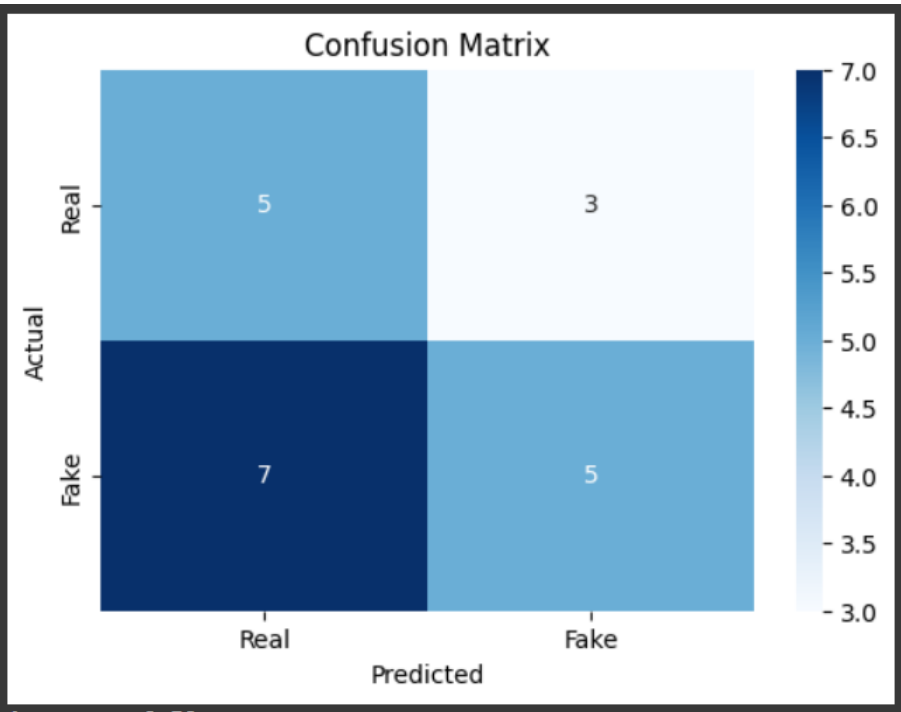
9. Conclusion

The project successfully demonstrated a basic machine learning approach to classifying fake job postings. While the initial results are modest, the groundwork has been laid for more sophisticated detection systems that can use full-text analysis and additional features. Enhancing data richness and exploring ensemble or deep learning techniques could further improve accuracy.

---

10. References

- **scikit-learn documentation**
- **pandas documentation**
- **seaborn library for visualization**
- **Research papers on spam detection and online fraud**




```
Accuracy: 0.50
Precision: 0.62
Recall: 0.42
```

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score
from sklearn.preprocessing import LabelEncoder

df = pd.read_csv("fake_jobs.csv")


label_encoder = LabelEncoder()
df['is_fake_encoded'] = label_encoder.fit_transform(df['is_fake'])


X = df[['title_length', 'description_length', 'has_company_profile']]
y = df['is_fake_encoded']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)


y_pred = clf.predict(X_test)


cm = confusion_matrix(y_test, y_pred)

plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Real', 'Fake'], yticklabels=['Real', 'Fake'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()


accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)

print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
```