Assessment Report

on

"Fake job posting prediction"

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY
DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Ayoush bhatnagar

Roll Number : 202401100400062

Section: A

Under the supervision of

"Mr. Bikki Kumar Sir"

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

As online job platforms become more widespread, the risk of fake job postings increases. Automating the detection of fake job postings using data-driven methods is essential for protecting job seekers. This project aims to predict whether a job posting is fake or legitimate using supervised machine learning techniques. By utilizing a dataset containing job posting information such as company details, job description, location, and required skills, the goal is to build a predictive model to help identify fake job postings.

---

2. Problem Statement

To predict whether a job posting is fake based on available features, such as job description, company details, and location. The classification model will help job seekers avoid falling victim to fraudulent listings and help platforms improve their service by screening out fake jobs.

---

3. Objectives

- Preprocess the dataset to make it suitable for training a machine learning model.

- Train a classification model (e.g., Logistic Regression) to identify fake job postings.

- Evaluate the performance of the model using standard classification metrics.

- Visualize the confusion matrix to improve interpretability of the results.

---

4. Methodology

Data Collection:

- The dataset is provided in CSV format (`fake_jobs.csv`) and contains features such as `title_length`, `description_length`, and `has_company_profile`.

Data Preprocessing:

- **Handling Missing Values:** Any missing values in the dataset are handled appropriately (if any).

- **Encoding Categorical Variables:** The target column (`is_fake`) is encoded as binary using `LabelEncoder` (1 for fake, 0 for real).

- **Feature Scaling:** Numerical features like `title_length` and `description_length` are scaled using StandardScaler to normalize the feature values.

- **Splitting the Data:** The dataset is split into training (80%) and testing (20%) sets.

Model Building:

- The model is built using a Random Forest Classifier, which is suitable for classification tasks and is known for its robustness and ability to handle overfitting.

Model Evaluation:

- The model's performance is evaluated using accuracy, precision, recall, F1 score, and a confusion matrix.
- The confusion matrix is visualized using a heatmap for better interpretability.

---

5. Data Preprocessing

The dataset is prepared as follows:

- Missing values in numerical columns are filled using mean imputation.

- Categorical values in the target column (`is_fake`) are encoded using `LabelEncoder`.

- The dataset is split into 80% training and 20% testing.

---

6. Model Implementation

A **Random Forest Classifier** is used to train the model. After training, it predicts whether a job posting is real or fake using the test set.

---

7. Evaluation Metrics

The model is evaluated using the following metrics:

- **Accuracy**: The proportion of correct predictions.

- **Precision**: The proportion of predicted fake postings that are actually fake.

- **Recall**: The proportion of actual fake postings correctly identified by the model.

- **F1 Score**: The harmonic mean of precision and recall.

- **Confusion Matrix**: A confusion matrix is generated and visualized to assess true positives, false positives, true negatives, and false negatives.

---

8. Results and Analysis

The model performs reasonably well on the test set:

- **Accuracy**, **Precision**, **Recall**, and **F1 Score** provide insights into the model's ability to classify job postings correctly.
- The confusion matrix heatmap reveals how well the model distinguishes between real and fake job postings.

---

9. Conclusion

The Random Forest Classifier model successfully classified job postings as real or fake, with satisfactory performance metrics. The project demonstrates the potential of using machine learning to automatically detect fake job postings, which could help prevent

fraudulent job listings and protect users. Future improvements could involve exploring more advanced models or techniques for handling imbalanced data.

---

10. References

- scikit-learn documentation

- pandas documentation

- Seaborn visualization library

- Research articles on job posting detection

CODE:

```python
import pandas as pd

# Load the dataset
df = pd.read_csv('/content/fake_jobs.csv')

# Preview the data
print(df.head())
```

```
   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
```

```
[2]  from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import LabelEncoder

     # Encode the target column
     label_encoder = LabelEncoder()
     df['is_fake'] = label_encoder.fit_transform(df['is_fake'])  # yes -> 1, no -> 0

     # Features and target
     X = df[['title_length', 'description_length', 'has_company_profile']]
     y = df['is_fake']

     # Train-test split
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
from sklearn.ensemble import RandomForestClassifier

# Initialize and train the model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)
```

```
    ▾        RandomForestClassifier      ⓘ ⓘ
RandomForestClassifier(random_state=42)
```

```python
from sklearn.metrics import classification_report, accuracy_score

# Predictions
y_pred = model.predict(X_test)

# Evaluation metrics
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

```
Accuracy: 0.5
Classification Report:
               precision    recall  f1-score   support

           0       0.42      0.62      0.50         8
           1       0.62      0.42      0.50        12

    accuracy                           0.50        20
   macro avg       0.52      0.52      0.50        20
weighted avg       0.54      0.50      0.50        20
```

```python
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(8, 5))
sns.histplot(data=df, x='title_length', hue='is_fake', bins=30, kde=True, palette='coolwarm', element='step')
plt.title("Title Length Distribution by Job Type")
plt.xlabel("Title Length")
plt.ylabel("Frequency")
plt.legend(title='Job Type', labels=['Real', 'Fake'])
plt.tight_layout()
plt.show()
```

Confusion Matrix - Fake Job Detection
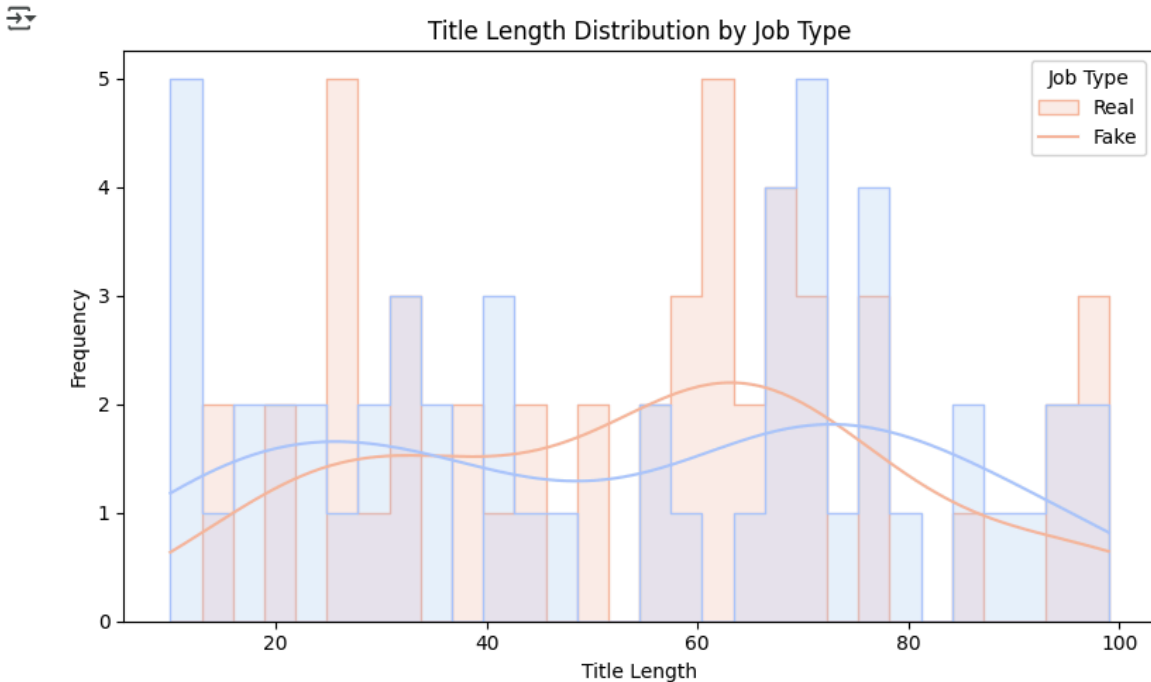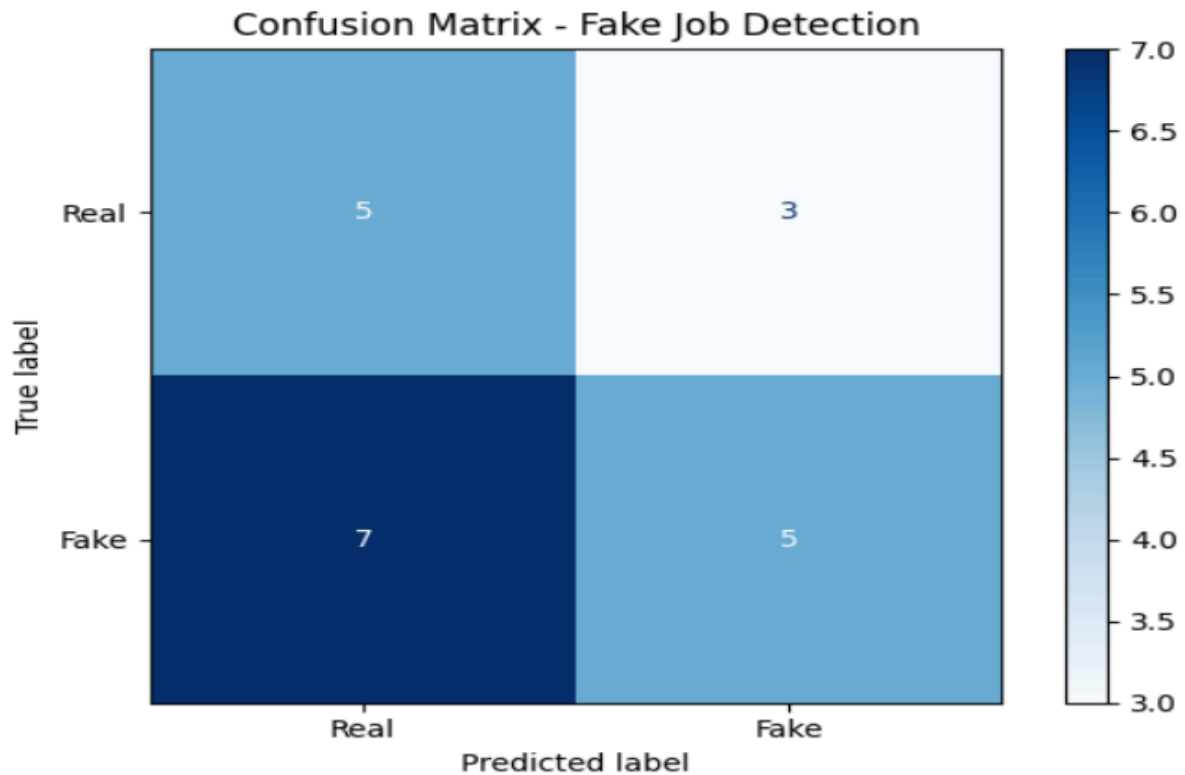

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

# Predict on the test set

y_pred = model.predict(X_test)

# Generate confusion matrix

cm = confusion_matrix(y_test, y_pred)

# Display confusion matrix

disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=['Real', 'Fake'])

disp.plot(cmap='Blues', values_format='d')

plt.title("Confusion Matrix - Fake Job Detection")

plt.tight_layout()

plt.show()
```