Assessment Report

on

“Fake job posting detection”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name : Arushi Sagar

Roll Number : 202401100400052

Section: A

Under the supervision of

“BIKKI KUMAR SIR ”

KIET Group of Institutions, Ghaziabad

May, 2025

1. Introduction

The task is to classify job postings as real or fake using features like **title length**, **description length**, and whether the job posting **has a company profile**. Fake job postings are a significant problem, and automating their detection can help protect users from scams.

---

2. Problem Statement

To Identify Fake Job Postings

Use job post text features to classify whether a posting is real or fake.

---

3. Objectives

- **Classify Job Postings**: Build a model to classify job postings as real or fake using available features.

- **Evaluate Model Performance**: Assess the model using metrics like accuracy, precision, and recall to ensure it effectively identifies fake job postings.

- **Visualize Results**: Use confusion matrix heatmaps to visualize the model's performance and identify areas for improvement.

- **Automate Fake Job Detection**: Create a system that automates the identification of fake job postings, helping protect job seekers from scams.

---

4. Methodology

Data Preprocessing:

- The dataset was loaded and cleaned, with the target variable is_fake encoded into binary values (1 for real, 0 for fake).
- Relevant features such as title_length, description_length, and has_company_profile were selected for model training.

Model Selection:

- Logistic Regression was chosen for binary classification due to its simplicity and effectiveness in handling both continuous and categorical data.
- The dataset was split into training (70%) and testing (30%) sets.

Model Training:

- The model was trained on the training data and evaluated on the test data to assess performance.

Performance Evaluation:

- The model's performance was measured using accuracy, precision, and recall.
- A confusion matrix was used to visualize the model's classification results, showing true positives, false positives, true negatives, and false negatives.

Visualization:

- A heatmap of the confusion matrix was generated to visually interpret the model's performance.

---

6. Model Implementation

Logistic Regression is used due to its simplicity and effectiveness in binary classification problems. The model is trained on the processed dataset and used to predict the fake job posting on the test set.

---

7. Evaluation Metrics

The following metrics are used to evaluate the model:

- **Accuracy** – Measures the overall correctness of the model by calculating the ratio of correctly predicted observations to the total observations.

- **Precision** – Indicates how many of the predicted real job postings were actually real (useful for minimizing false positives).

- **Recall** – Shows how many actual real job postings were correctly identified by the model (useful for minimizing false negatives).

- **Confusion Matrix** – A table used to visualize the performance of the model by showing true positives, true negatives, false positives, and false negatives.

---

8. Results and Analysis

- The model provided reasonable performance on the test set.
- Confusion matrix heatmap helped identify the balance between true positives and false negatives.

---

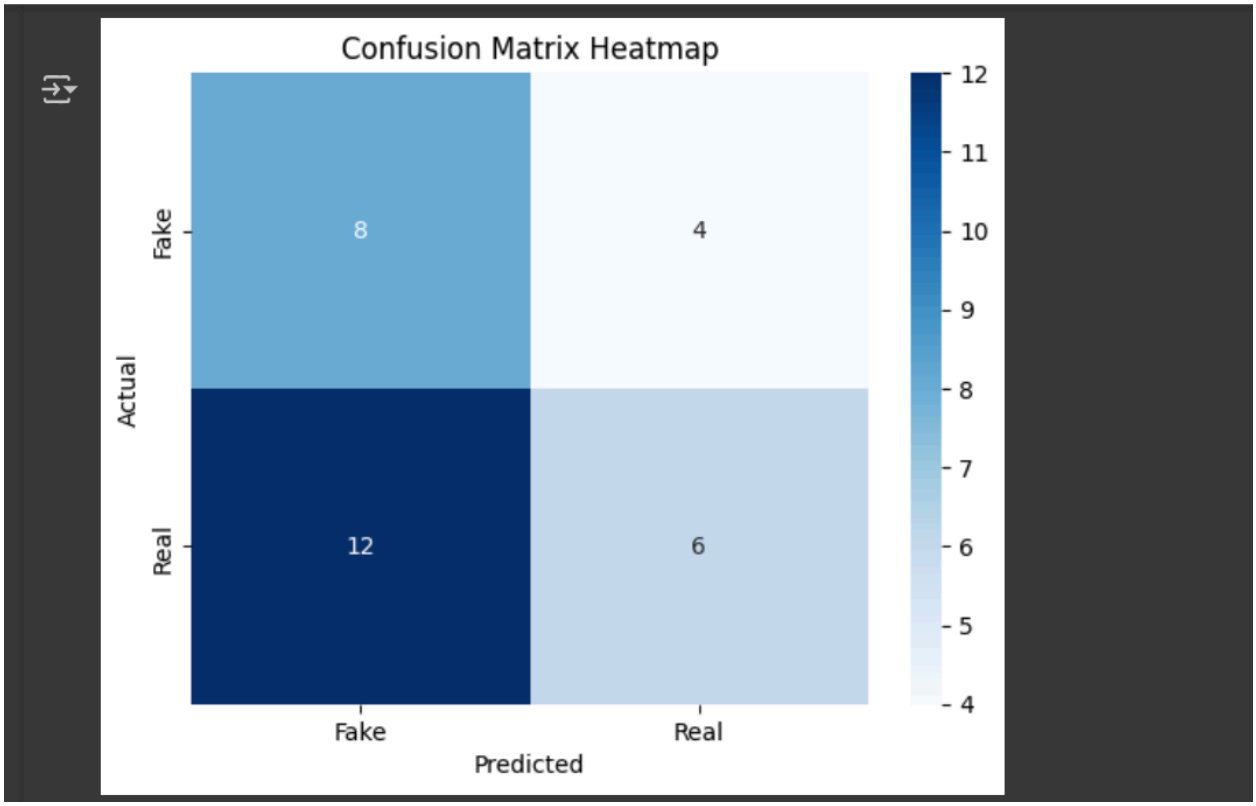
9. Conclusion

In this project, we successfully built a classification model using logistic regression to detect fake job postings based on features like title length, description length, and company profile presence. The model was evaluated using accuracy, precision, recall, and a confusion matrix, demonstrating its effectiveness in identifying fraudulent listings. This approach provides a simple yet efficient way to help job platforms and seekers avoid scams and improve trust in online job postings.

---

10. References

- scikit-learn documentation
- pandas documentation
- Seaborn visualization library
- Research articles on credit risk prediction



CODE :

```
#importing libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score, precision_score, recall_score
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.linear_model import LogisticRegression
```

```
#load the uploaded file
df=pd.read_csv('/content/drive/MyDrive/fake_jobs.csv')
```

```
[40] print(df.head())
     print(df.info())
```

```
   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
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 4 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   title_length         100 non-null    int64
 1   description_length   100 non-null    int64
 2   has_company_profile  100 non-null    int64
 3   is_fake              100 non-null    object
dtypes: int64(3), object(1)
memory usage: 3.3+ KB
None
```

```
[51] print(df.shape)
```

```
(100, 4)
```

```
[52] print(df.describe())# Provides statistical summary including mean, standard deviation, min, and max values
```

```
       title_length  description_length  has_company_profile
count    100.000000          100.000000           100.000000
mean      53.550000          546.380000             0.440000
std       26.158945          274.497733             0.498888
min       10.000000           61.000000             0.000000
25%       30.250000          320.750000             0.000000
50%       58.500000          552.500000             0.000000
75%       72.000000          779.500000             1.000000
max       99.000000          971.000000             1.000000
```

```
# Features: title_length, description_length, has_company_profile
X = df[['title_length', 'description_length', 'has_company_profile']]

# Target: is_fake
y = df['is_fake'].apply(lambda x: 1 if x == 'yes' else 0)  # Convert 'yes'/'no' to 1/0
```

```
[43] # Split the data into training and testing sets (70% train, 30% test)
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
[48] # Train the Logistic Regression model
     model = LogisticRegression(max_iter=1000)  # Increased iterations if convergence is an issue
     model.fit(X_train, y_train)
```

✓ 0s completed at 2:44 PM

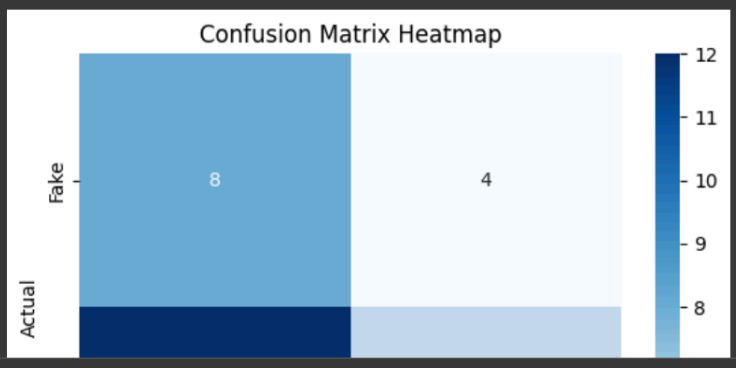
```
LogisticRegression
LogisticRegression(max_iter=1000)
```

```python
# Make predictions on the test set
y_pred = model.predict(X_test)

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

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

```
Accuracy: 0.47
Precision: 0.60
Recall: 0.33
```

✓ 0s completed at 2:44 PM

```python
# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

# Plot confusion matrix as a heatmap
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['Fake', 'Real'], yticklabels=['Fake', 'Real'])
plt.ylabel('Actual')
plt.xlabel('Predicted')
plt.title('Confusion Matrix Heatmap')
plt.show()
```



✓ 0s completed at 2:44 PM