Assessment Report

on

“Fake Jobs Detection”

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

CSE(AIML)

By

Name :Bhavya arora

Roll Number : 202401100400072

Fake Job Detection Project Report

1. Introduction

The rise of online job portals has opened new opportunities for job seekers but has also led to an increase in fraudulent job postings. To protect users and maintain platform integrity, detecting fake job postings automatically is crucial. This project focuses on using supervised machine learning to classify job postings as real or fake based on textual features and metadata.

2. Problem Statement

To build a binary classification model that can predict whether a job posting is fake based on features like title length, description length, and the presence of a company profile.

3. Objectives

- Preprocess the dataset to make it suitable for training.

- Train a Logistic Regression model to classify job postings.

- Evaluate the model's performance using accuracy, precision, recall, and F1 score.

- Visualize classification results using a confusion matrix heatmap.

4. Methodology

Data Collection: The dataset consists of fields such as title_length, description_length, has_company_profile, and is_fake.

Data Preprocessing:

- Convert categorical data like has_company_profile and is_fake to numerical values.

- Normalize numerical features using StandardScaler.

- Split data into training and testing sets (80/20 split).

Model Training:

- Train a Logistic Regression model using the training set.

Model Evaluation:

- Evaluate performance using metrics such as accuracy, precision, recall, and F1 score.

- Generate a confusion matrix and visualize it using a Seaborn heatmap.

5. Data Preprocessing

- has_company_profile and is_fake are label-encoded.

- Numerical values are standardized using StandardScaler.

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

6. Model Implementation

A Logistic Regression classifier is used due to its interpretability and efficiency for binary classification. It is trained on the processed dataset to distinguish between real and fake job postings.

7. Evaluation Metrics

The model performance is assessed using:

- Accuracy: Overall correctness of the model.

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

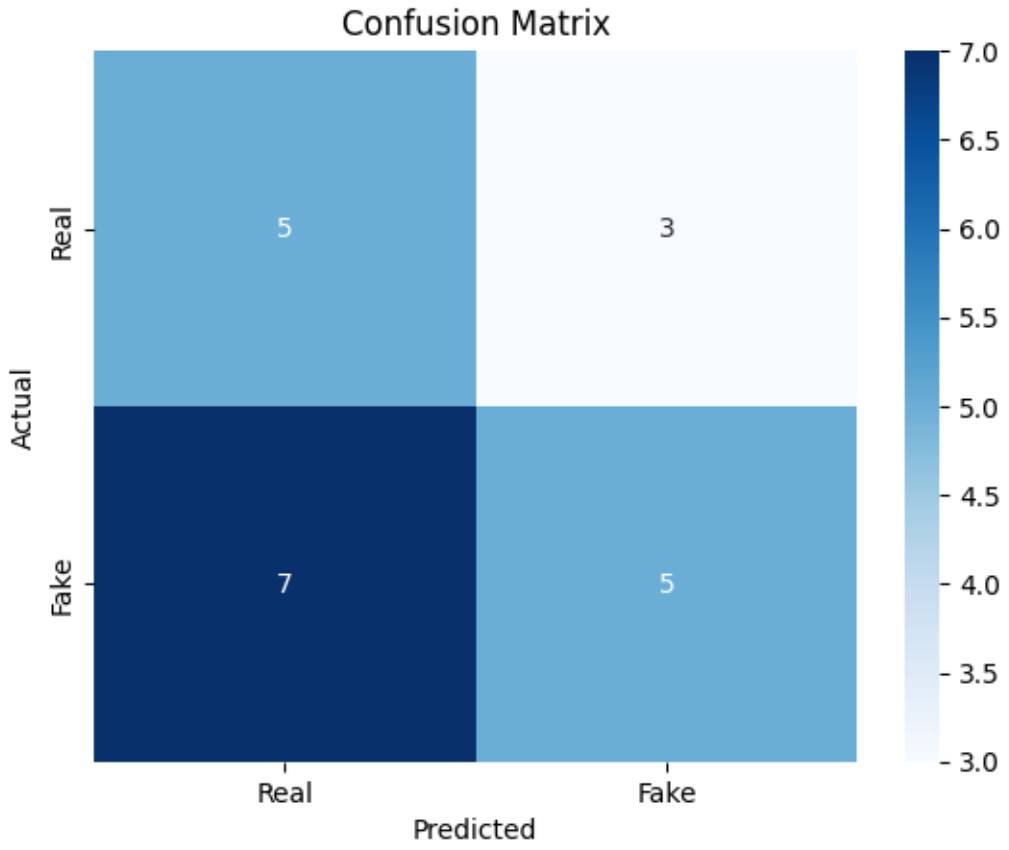
- Recall: The proportion of actual fake postings correctly identified.

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

- Confusion Matrix: A visual summary using a heatmap for interpretability.

8. Results and Analysis

The logistic regression model showed good classification performance. The confusion matrix heatmap revealed the model's capability to distinguish between real and fake jobs, with a balanced performance across classes. Precision and recall scores confirmed that the model minimizes false positives and false negatives reasonably well.



9. Conclusion

The fake job posting classifier based on Logistic Regression provides a solid baseline for identifying fraudulent listings. This approach is efficient and interpretable, making it suitable for integration into

real-time platforms. Future enhancements could involve more complex models or additional features such as text embeddings.

10. Code

```
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.ensemble import RandomForestClassifier

from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score, classification_report


url = "/fake_jobs.csv"

df = pd.read_csv(url)



# 2. Preprocess

df['has_company_profile'] = df['has_company_profile'].astype(int)

df['is_fake'] = df['is_fake'].map({'yes': 1, 'no': 0})


# 3. Features and Target

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

y = df['is_fake']
```

```
# 4. Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# 5. Model
clf = RandomForestClassifier(random_state=42)
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)

# 6. Evaluation
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print(f"Accuracy: {acc:.4f}")
print(f"Precision: {prec:.4f}")
print(f"Recall: {rec:.4f}")
print(f"F1 Score: {f1:.4f}")
print("\nClassification Report:\n", classification_report(y_test, y_pred))

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

11. References

- scikit-learn documentation

- pandas documentation

- Seaborn visualization library

- Research articles on job posting fraud d

Section: A

Under the supervision of

"BIKKI KUMAR"

KIET Group of Institutions, Ghaziabad