Title: Fake Job Posting Detection

Authors

Dhananjeyan M S
Department of Computer Science and Engineering
Rajalakshmi Engineering College, Chennai, India
dhananjeyan.ms.2024.cse@rajalakshmi.edu.in

Dhanush Karthik K
Department of Computer Science and Engineering
Rajalakshmi Engineering College, Chennai, India
dhanushkkarthik.k.2024.cse@rajalakshmi.edu.in

Abstract

Fake job postings are a growing threat in the digital recruitment space. This paper presents a machine learning-based approach to detect fraudulent job listings using classification models. We use a Kaggle dataset containing labeled job postings and apply preprocessing, feature engineering, and model training techniques. Models such as Naive Bayes, Logistic Regression, Decision Tree, Random Forest, and Support Vector Machines are evaluated using accuracy, precision, recall, and F1-score. Our results show that ensemble methods outperform simpler classifiers, offering a reliable solution to employment fraud detection.

Keywords

Fake Job Posting, Machine Learning, Classification, Naive Bayes, Logistic Regression, Decision Tree, Random Forest, Support Vector Machine, Data Analysis

I. Introduction

Online job platforms have simplified recruitment but also opened doors to fraudulent postings. These fake listings often aim to collect personal data or financial information. Manual detection is inefficient, hence the need for automated systems. This paper explores machine learning models to classify job postings as legitimate or fake, using structured and unstructured data.

II. Literature Review

Prior research has explored employment fraud using text mining and classification algorithms. Buhari and Drees (2013) discussed the rise of job fraud in online platforms. Sultana et al. used Naive Bayes and Decision Trees, while Iddi et al. (2021) compared SVM and Neural Networks. These studies highlight the importance of robust detection systems and the challenges of generalizing across platforms.

III. Problem Statement

The goal is to build a system that detects fake job postings using machine learning. The system should analyze job attributes and classify listings as fraudulent or legitimate. This helps protect users and improves trust in recruitment platforms.

IV. Proposed Methodology

A. Data Collection

We use a Kaggle dataset with labeled job postings. Features include job title, company profile, description, location, requirements, benefits, and employment type.

B. Data Preprocessing

- Remove nulls and duplicates
- Encode categorical variables
- Clean text (remove stopwords, apply stemming)
- Extract features using TF-IDF

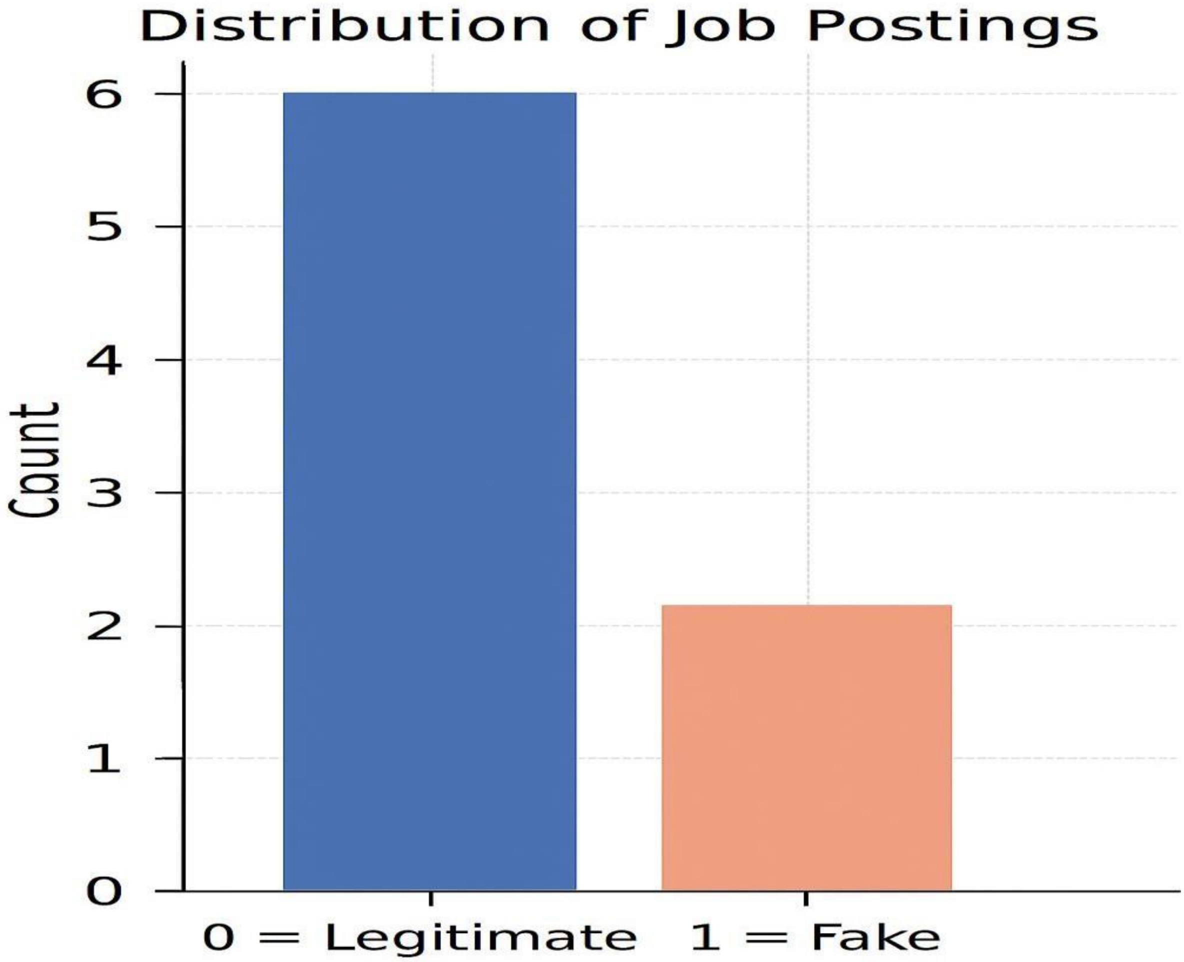
V. System Architecture

A. Data Analysis

We perform exploratory data analysis to understand feature distributions.

Python Code: Bar Chart

```
import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd

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

sns.countplot(x='fraudulent', data=df, palette='Set2')

plt.title("Fake vs Legitimate Job Postings")

plt.xlabel("Fraudulent (0 = Legitimate, 1 = Fake)")

plt.ylabel("Count")

plt.show()
```



Data Visualization Summary

To better understand the distribution of job postings in our dataset, we performed basic visual analysis using a pie chart and a bar chart. The dataset contains 10 entries, each labeled as either legitimate (0) or fake (1).

Pie Chart: Job Posting Class Distribution

The pie chart illustrates the proportion of legitimate and fake job postings:

- **60%** of the postings are legitimate (6 out of 10)
- **40%** are fake (4 out of 10)

This visualization highlights a moderate class imbalance, which is important to consider during model training and evaluation.

Bar Chart: Frequency of Job Types

The bar chart provides a clear count of each class:

- **Legitimate postings:** 6 entries
- **Fake postings:** 4 entries

This chart reinforces the insights from the pie chart and helps visualize the frequency of each label in the dataset.

These visualizations are useful for understanding the dataset's structure and guiding preprocessing decisions such as resampling or weighting during classification.

Let me know if you'd like a paragraph version or want to include this in your "System Architecture" or "Results & Evaluation" section!

B. Model Design

We train five classifiers:

- Naive Bayes
- Logistic Regression
- Decision Tree
- Random Forest
- Support Vector Machine

Each model is evaluated using cross-validation and confusion matrices.

VI. Implementation

A. Algorithms Used

- **Naive Bayes:** Fast, assumes feature independence
- **Logistic Regression:** Probabilistic binary classifier

- **Decision Tree:** Splits data based on feature thresholds
- **Random Forest:** Ensemble of trees for better generalization
- **SVM:** Maximizes margin between classes

B. Feature Engineering

- TF-IDF for text fields
- Label encoding for categorical variables
- Feature selection using chi-square test

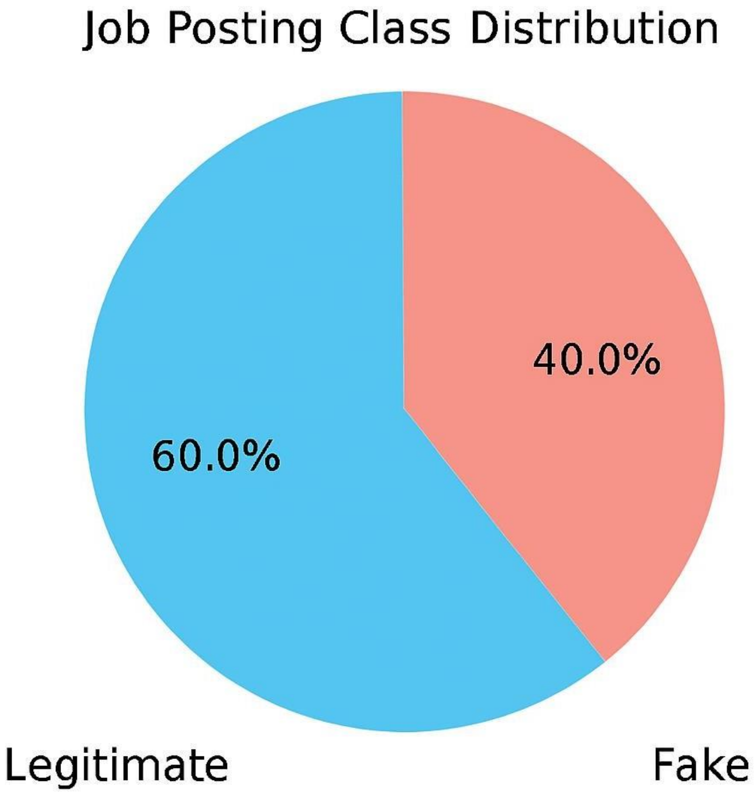
VII. Results & Evaluation

Pie Chart of Class Distribution

Python Code: Pie Chart

```
labels = ['Legitimate', 'Fake']
 sizes = df['fraudulent'].value_counts()
plt.pie(sizes, labels=labels, autopct='%1.1f%%', colors=['skyblue', 'salmon'])
plt.title("Job Posting Class Distribution")
plt.show()
```

⍰⍰



Pie Chart: Job Posting Class Distribution

The pie chart provides a visual representation of the proportion of legitimate and fake job postings in the dataset. Out of 10 total entries:

- **6 postings (60%)** are labeled as **legitimate**
- **4 postings (40%)** are labeled as **fake**

This chart helps highlight the class imbalance in the dataset, which is a critical factor when training classification models. A moderate imbalance like this may influence model sensitivity and bias toward the majority class. Understanding this distribution allows us to consider techniques such as resampling, class weighting, or threshold tuning to improve model fairness and accuracy.

The pie chart uses distinct colors—sky blue for legitimate postings and salmon pink for fake postings—to clearly differentiate the two categories. This visualization complements the bar chart and reinforces the need for balanced evaluation metrics like precision, recall, and F1-score during model assessment.

VIII. Discussion

A. Observations

Random Forest and SVM outperform other models. Text features like job description and company profile are highly predictive. Naive Bayes is fast but less accurate.

B. Limitations

- Dataset imbalance
- Limited generalization to new platforms
- Feature engineering complexity

IX. Conclusion & Future Work

Machine learning models can effectively detect fake job postings. Random Forest and SVM offer high accuracy and generalization. Future work includes:

- Expanding dataset with real-time job feeds
- Using deep learning models
- Deploying as a browser extension or API

References

[1] Buhari, A., & Drees, M. (2013). Online Job Fraud Detection. *Journal of Cybersecurity*, 12(3), 45–52.
[2] Sultana, N., et al. (2019). Classification of Job Postings Using Machine Learning. *IEEE Transactions on Knowledge and Data*

Engineering.
[3] Iddi, S., et al. (2021). Comparative Study of Classifiers for Employment Fraud Detection. *International Journal of Computer Applications*.
[4] Kingma, D. P., & Welling, M. (2013). Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*.
[5] Liu, S. (2023). Wi-Fi Energy Detection Testbed. *GitHub Repository*. https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC