
University of California San Diego

Course #: CSE 258

# Integrating Textual and Meta-Features for Fraudulent Job Post Detection: An XGBoost Classifier Approach

January 1, 2025

*Authors*

Bixiao Feng
Jerry Wu
Wei Zhou

## Abstract

Employment scams are a growing concern, especially in times of economic uncertainty and high unemployment, such as those brought on by the COVID-19 pandemic. Scammers exploit job seekers' desperation by creating fraudulent job postings to steal personal information, including addresses, bank account details, and social security numbers. These scams often involve enticing job offers or requests for upfront payments, posing significant risks to individuals.

This project aims to address the issue of fraudulent job postings using Machine Learning and Natural Language Processing (NLP) techniques. Leveraging a dataset from Kaggle, which includes job postings labeled as real or fake, the project focuses on developing a robust model to identify fraudulent listings. Given the imbalance in the dataset—where fake postings represent a small fraction—specific techniques are employed to handle this class imbalance effectively.

The workflow for this project follows five distinct stages, ensuring a systematic approach to understanding, preprocessing, modeling, and validating the data to provide accurate and actionable results. By detecting fake job postings, this project seeks to contribute to safeguarding job seekers and fostering a more secure employment landscape.

## 1. Introduction

Fraudulent job postings pose significant risks to job seekers and organizations, ranging from identity theft to financial loss. While existing detection methods have been implemented, their efficiency is limited by outdated features and oversimplified models. This study aims to enhance detection accuracy by exploring feature engineering techniques and reconstruct high-information features.

This study utilizes a dataset from the University of the Aegean, Laboratory of Information & Communication Systems Security. The dataset contains 17,880 records and 18 features. The numerical variables include features such as "has company logo" and "has questions", while the textual descriptions consist of fields like description, requirements, and benefits. Additionally, there are categorical variables such as "employment type" and "required experience". For more detailed information, refer to Appendix Table 1.

## 2. Literature Review

In existing studies, advanced machine learning models such as the SGD Classifier, Random Forest Classifier, and XGBoost Classifier are widely utilized, achieving an accuracy of approximately 0.97. Based on these models, this study argues that, first, relying solely on accuracy to evaluate model performance is unreliable. This is because most datasets have a much higher proportion of real jobs compared to fake jobs. For instance, if real jobs account for 95% of the data, predicting all instances as real jobs would still yield an accuracy of 95%. To address this, the study introduces the F1 score as an alternative evaluation metric. The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is especially useful for datasets with imbalanced classes. The formula is:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Second, prior studies typically use either numerical variables or text variables processed with Term Frequency-Inverse Document Frequency (TF-IDF) as input independently. The only integration of textual and numerical data in earlier attempts was converting text length into a single numerical variable, failing to fully integrate textual and numerical features. To improve upon this, the study introduces a novel method that incorporates text features processed using Bag of Words (BoW) into the numerical matrix, creating a new dataset. Additionally, it was observed that certain words appear frequently in fake job postings. Therefore, the top five most frequently occurring words were isolated into a separate matrix by BoW and included in the dataset to increase their weight in the prediction process.

## 3. Methodology

Figure 1's representation of the implementation process for this project is shown below. The dataset is divided into text, numeric, and label components. The text data is transformed into a term-frequency matrix and a bag-of-words matrix for further analysis. The numeric variables are processed to generate an additional matrix.

The baseline XGBoost model is trained to identify the most suitable combination of variables. After identifying the optimal combination of variables, we evaluated this feature set using Logistic Regression, SGD Classifier, Random Forest Classifier, and XGBoost Classifier, assessing their performance.

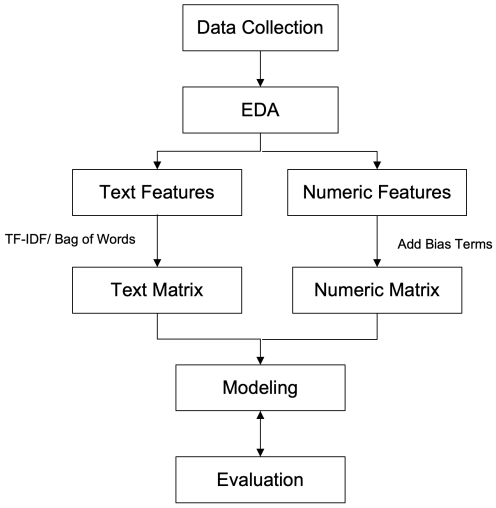


Figure 1: Stages of Development

### 3.1. Data Cleaning, Transformation, and Exploration

#### 3.1.1 Data Processing

Several preprocessing steps were carried out to prepare the dataset for analysis. First, a new column, `text`, was created by concatenating the `requirements`, `description`, and `benefits` columns. The `langdetect` library was utilized to filter rows, retaining only English text to ensure language consistency throughout the dataset.

Next, feature transformation and categorization were performed. Missing values in the `location` column were filled with the string `blank` and the column was split into `country`, `state`, and `city`, with any whitespace trimmed. For features such as `industry`, `department`, and `salary_range`, infrequent categories were grouped into an `'Other'` category, retaining only the top 20 categories. Similarly, the `country` column was simplified to include only the top 2 categories, with all others grouped into `'Other'`.

#### 3.1.2 Adding Quantitative Variables

Fraud ratios for categorical features like `required_education`, `employment_type`, and `required_experience` were calculated by comparing the occurrence of fraudulent vs. non-fraudulent entries. The ratios (`edu_ratio`, `employment_ratio`, and `exp_ratio`) function similarly to bias terms and were introduced as new features to improve the model's predictive performance.

Building on previous studies, we included text length as part of the numerical variables. Since there are multiple textual variables, such as job title and job description, we selected the variable with the biggest difference in character count distribution between fraudulent and real job postings for calculation. As shown in Figure 2, the distributions of `company_profile` and description exhibit the largest differences.

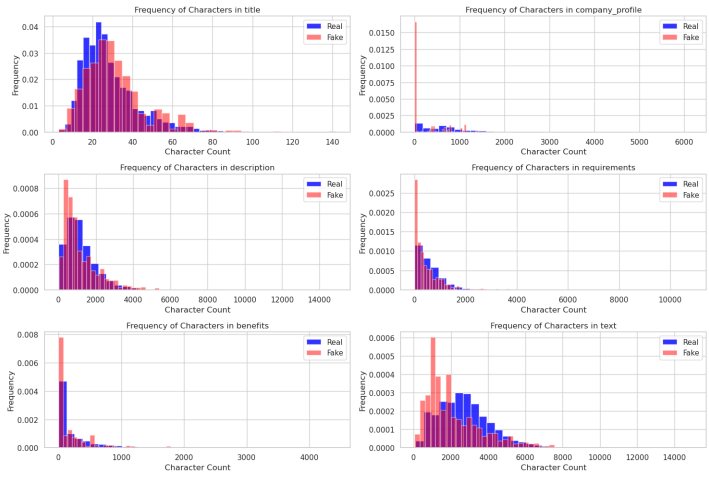


Figure 2: Frequency of Characters

Consequently, we defined `character_count` (Figure 3) as the sum of the character lengths of `company_profile` and `description`.

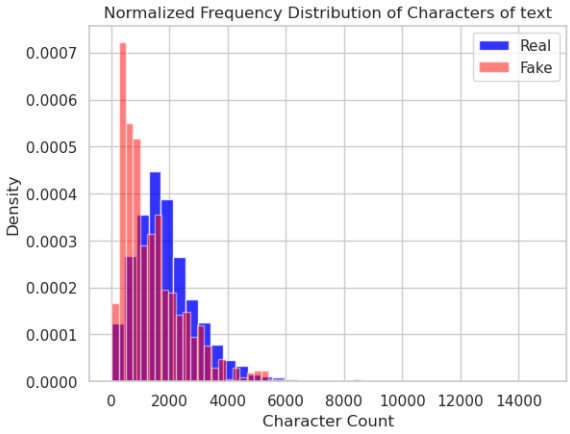


Figure 3: Characters Counts for Fraud and Real Jobs

### 3.2. Text Processing by Natural Language Processing

The text preprocessing began by downloading essential resources, including punkt, stopwords, and wordnet, from the Natural Language Toolkit (NLTK) library. The following steps were taken for text processing:

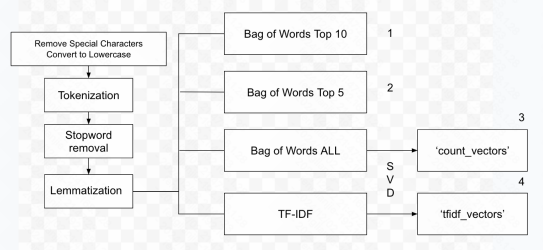


Figure 4: Text Processing Logistics

Missing text entries were replaced with empty strings and the text was tokenized, converted to lowercase, and cleaned by removing non-alphabetic characters and stopwords.

For the Bag-of-Words (BoW) model, the tokens were lemmatized using `WordNetLemmatizer` to reduce words to their base forms. The most frequent tokens were identified to create a Bag-of-Words representation. Two new columns, `bow_top10` and `bow_top5`, were added to represent feature vectors corresponding to the top 10 and top 5 most frequent words, respectively. In addition, we applied `CountVectorizer` to create a global Bag-of-Words (BoW) representation and used `TfidfVectorizer` for global TF-IDF vectorization. Due to the high dimensionality of the output matrices, we employed an SVD model to reduce the dimensionality of both matrices to 300 columns.

### 3.3. Feature Selection

We now aim to combine the NLP features with the existing numeric features and identify the best combination based on performance using the baseline model (XGBoost).

Potential Feature Combinations:

- **Combo 1:** `TF-IDF`, `bow_top10`, and `numeric features`.
- **Combo 2:** `Count`, `bow_top10`, and `numeric features`.
- **Combo 3:** `TF-IDF`, `bow_top5`, and `numeric features`.
- **Combo 4:** `Count`, `bow_top5`, and `numeric features`.

The ranking of the performance of all the potential feature combinations in descending order is **Combo 4**, **Combo 2**, **Combo 1**, and **Combo 3**. Thus, we choose **Combo 4** with `Count`, `bow_top5`, and `numeric features`.

These steps prepared the text data by removing noise, extracting key features, and generating structured numerical inputs for machine learning models. By combining the NLP features and the other numeric features, we obtain the best features for training our model.

### 3.4. Models

We chose four candidate models in this problem. Logistic Regression is the simplest one and can be easily explained. SGD has demonstrated strong performance in the literature and is often well-suited for NLP tasks. We also tried two tree-based models, Random Forest and XGBoost, as they can handle nonlinear correlation.

We implemented the features and label selected above as our data. We divided the data into

training and testing sets using an 80:20 split. We trained each of the following models with the same training data and fine-tuned the parameters to optimize the results. The parameters of each model are shown in Table 1.

| Model | Key Parameters |
|---|---|
| Logistic Regression | - `penalty`: l2<br>- `C`: 100<br>- `solver`: lbfgs<br>- `max_iter`: 100 |
| SGD | - `loss`: log loss<br>- `penalty`: l2<br>- `alpha`: 0.0001<br>- `max_iter`: 1000 |
| Random Forest | - `n_estimators`: 200<br>- `max_depth`: None<br>- `min_samples_split`: 2<br>- `min_samples_leaf`: 1<br>- `criterion`: gini |
| XGBoost | - `eval_metric`: auc<br>- `alpha`: 0.05<br>- `max_depth`: 5<br>- `min_child_weight`: 3<br>- `n_estimators`: 400<br>- `subsample`: 1<br>- `colsample_bytree`: 1<br>- `gamma`: 0 |

Table 1: Key Parameters for Different Models

### 3.5. Evaluation Metrics

The models are evaluated using the following metrics:

#### 3.5.1 Accuracy

Accuracy measures the ratio of correctly classified data points to the total number of data points. It is calculated using the formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}$$

Accuracy provides an overall measure of the model's performance in correctly categorizing both real and fake jobs. However, this metric has limitations, particularly with imbalanced datasets. In this scenario, where real job postings vastly outnumber fake ones, a high accuracy may primarily reflect the model's ability to identify the dominant class (real jobs), potentially neglecting the minority class (fake jobs).

#### 3.5.2 Recall

Recall, also known as sensitivity, quantifies the model's ability to correctly identify all instances of the positive class (fake jobs). It is calculated as:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

A high recall ensures that most fake job postings are identified, minimizing the likelihood of missing fraudulent entries.

#### 3.5.3 F1-Score

The F1-score is the harmonic mean of precision and recall, providing a balance between the two metrics. It is especially useful for datasets with imbalanced classes. The formula is:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Here:

- **Precision:** The proportion of correctly identified fake jobs among all predicted fake jobs.
- **Recall:** Measures the proportion of correctly identified fake jobs out of all actual fake jobs.

The F1-score is crucial for this project, as both false positives (real jobs incorrectly classified as fake) and false negatives (fake jobs incorrectly classified as real) carry significant consequences.

#### 3.5.4 Area Under the Curve (AUC)

The AUC score measures the model's ability to distinguish between the positive (fake jobs) and negative (real jobs) classes across various threshold settings. It is derived from the Receiver Operating Characteristic (ROC) curve, which plots the true positive rate (recall) against the

false positive rate. A high AUC score indicates a strong ability to separate the two classes, regardless of class imbalance.

By combining these metrics, the evaluation ensures that the classifier not only performs well overall (accuracy) but also excels in identifying fake job postings (recall) while maintaining precision and achieving a robust trade-off between sensitivity and specificity (F1-score and AUC).

## 4. Results

The final model selected for this analysis is the XGBoost (XGB) Classifier. This decision is based on the superior performance of the XGB Classifier across all key metrics compared to other models. As shown in the tables above, XGB achieved the highest F1 Score (0.824) and AUC Score (0.864), making it the most effective model for the given task. The detailed results and confusion matrices can be found in Appendix Figures 1 to 4.

Table 2: Comparison of LR and SGD Classifier

| **Metric** | **LR Classifier** | **SGD Classifier** |
|---|---|---|
| Accuracy | 0.972 | 0.971 |
| Recall | 0.610 | 0.599 |
| F1 Score | 0.694 | 0.679 |
| AUC Score | 0.801 | 0.795 |

Table 3: Comparison of RF and XGB Classifier

| **Metric** | **RF Classifier** | **XGB Classifier** |
|---|---|---|
| Accuracy | 0.969 | 0.984 |
| Recall | 0.396 | 0.731 |
| F1 Score | 0.567 | 0.824 |
| AUC Score | 0.698 | 0.864 |

## 5. Conclusion and Discussion

In this task of detecting fraudulent job postings, Bag-of-Words (BoW) provides a more effective text representation compared to TF-IDF scores. By integrating text features with other numeric features, we can achieve optimal predictive performance.

Among the models, XGBoost demonstrated the best performance, achieving an accuracy of 0.984 and an F1-Score of 0.824, likely due to the nonlinearity and robustness inherent in its framework.

However, due to the significant imbalance in our dataset, the optimal features and model identified for our dataset may not be the best fit for others. Nonetheless, our approach serves as a valuable guide for similar tasks in the future. Additionally, the model can be adapted to support job descriptions in other languages by incorporating appropriate text processors.

## A. Model Metrics and Visualizations

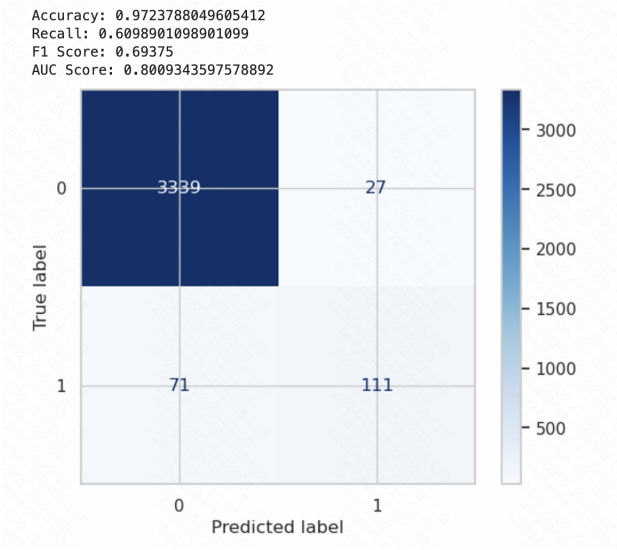


Figure 5: Detailed Result for Logistic Regression

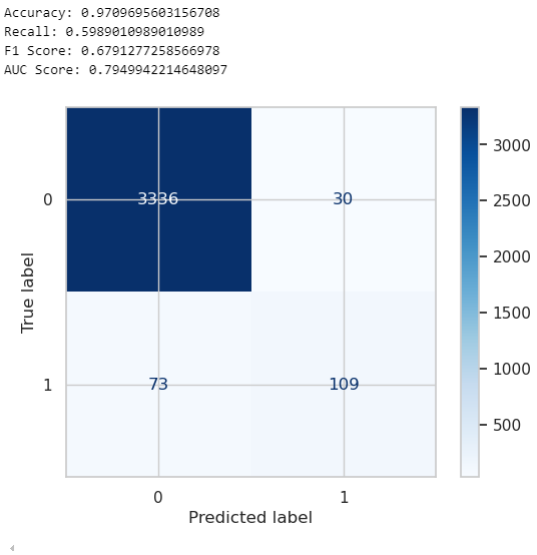


Figure 6: Detailed Result for SGD Classifier

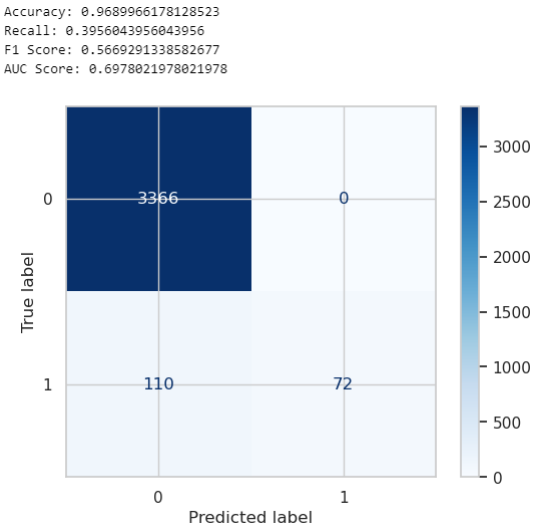


Figure 7: Detailed Result for Random Forest Classifier

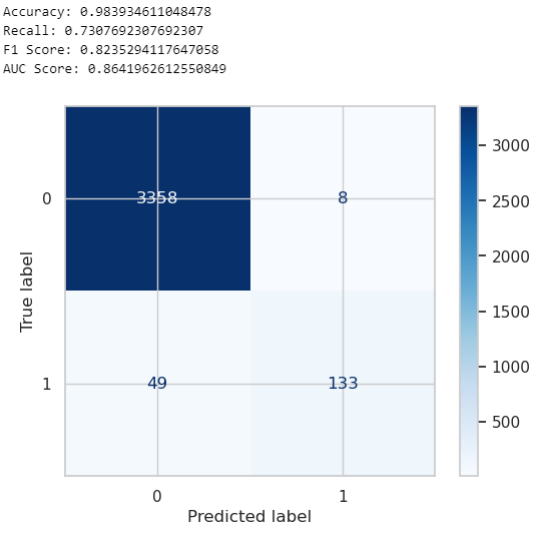


Figure 8: Detailed Result for XGB Classifier

## B. Dataset Variables

| # | Variable | Datatype | Description |
|---|---|---|---|
| 1 | job_id | int | Identification number given to each job posting |
| 2 | title | text | A name that describes the position or job |
| 3 | location | text | Information about where the job is located |
| 4 | department | text | Information about the department this job is offered by |
| 5 | salary_range | text | Expected salary range |
| 6 | company_profile | text | Information about the company |
| 7 | description | text | A brief description about the position offered |
| 8 | requirements | text | Pre-requisites to qualify for the job |
| 9 | benefits | text | Benefits provided by the job |
| 10 | telecommuting | boolean | Is work from home or remote work allowed |
| 11 | has_company_logo | boolean | Does the job posting have a company logo |
| 12 | has_questions | boolean | Does the job posting have any questions |
| 13 | employment_type | text | 5 categories – Full-time, part-time, contract, temporary, and other |
| 14 | required_experience | text | Can be – Internship, Entry Level, Associate, Mid-senior level, Director, Executive, or Not Applicable |
| 15 | required_education | text | Can be – Bachelor's degree, high school degree, unspecified, associate degree, master's degree, certification, some college coursework, professional, some high school coursework, vocational |
| 16 | industry | text | The industry the job posting is relevant to |
| 17 | function | text | The umbrella term to determine a job's functionality |
| 18 | fraudulent | boolean | The target variable (0: Real, 1: Fake) |

Table 4: Dataset Variables and Descriptions

## References

[1] Shivam Bansal. *Real or Fake: Fake Job Posting Prediction*. Kaggle. Available at: `https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction/data`. [Accessed: December 2, 2024].