**Northeastern University**
**CS 6120 Natural Language Processing**

**April 2022**

# Detection of Fake Job Postings Using Natural Language Processing and Machine Learning

**Chamanthi Aki**
**Sri Harika Cherukuri**

## Abstract

A popular scam nowadays is fake job advertisements. Ignorantly, people share their data with the scammers by applying to these fake jobs. For this purpose, we proposed a methodology that uses text and classify the jobs as fake or real from online recruitment portals by using Natural Language Processing and Supervised Machine Learning techniques. Initially, data is pre-processed and then various models namely RF (Random Forest), KNN (K-Nearest Neighbour), SVM (Support Vector Machine), LR (Logistic Regression) are implemented where KNN gave the best F-score when compared with all the other mentioned models.

## 1. Introduction

Scams involving employment are on the rise. According to CNBC, the number of job frauds have been doubled in 2018 over 2017. Unemployment is at an all-time high due to the current market condition. For many, the coronavirus impact and economic hardships have greatly reduced availability of work and resulted in job loss. Scammers benefit from a situation like this. Many individuals are unknowingly falling prey to these fraudsters who are preying on people's desperation. Scammers use this to obtain personal including Addresses, bank account numbers, and social security numbers.

University students receive multiple scam emails of this nature. Scammers provide customers with a fantastic job offer and then demand money in exchange. This is a dangerous problem that can be solved using Machine Learning and Natural Language Processing approaches.

## 2. Challenges

The primary motivation of this project is to develop a good-fit model on the balanced dataset. Many researchers have done work on the fake jobs postings and did not consider the data balancing, causing models over-fitting on majority class data. The ratio of real and fake job posts samples is unequal, which caused the model over-fitting on majority class data. To overcome this limitation, up sampling technique is used which helps to balance the ratio between target classes by generating the number of samples for minority class artificially.

## 3. Dataset

Dataset is taken from Kaggle[1]. This data contains features that define a job posting. These job postings are categorized as either real or fake. Fake job postings are a tiny fraction of this dataset. That is as excepted as we do not expect a lot of phony job postings. The below fig-1 clearly explains the features in the dataset. The dataset consists of 17,880 observations and 18 features.

| # | 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 determining a job's functionality |
| 18 | Fraudulent | boolean | The target variable ▤ 0: Real, 1: Fake |

Fig-1

## 4. Process flow

Fig-2 explains the process flow of the Model development which consists of 7 different phases namely Data Collection and Understanding, Exploratory Data Analysis, Data Pre-processing, Resampling, Model Selection, Data Classification, Model Evaluation.

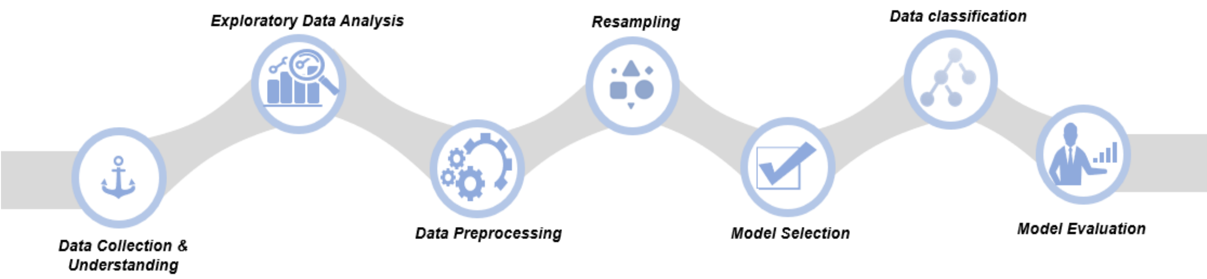


Fig-2

## 5. Data Preprocessing

Most of the features are either text or Boolean. Job_id is the only interger feature which is not relevant for this analysis. The dataset is further explored to identify null values. Fig-3 shows the null values in each column.

```
job_id                   0
title                    0
location               346
department           11547
salary_range         15012
company_profile       3308
description              1
requirements          2695
benefits              7210
telecommuting            0
has_company_logo         0
has_questions            0
employment_type       3471
required_experience   7050
required_education    8105
industry              4903
function              6455
fraudulent               0
```

Fig-3

All the unnecessary columns are dropped as we are mainly interested in text related features. The features that are considered are title, location, department, description, requirements, benefits, industry and function. These are combined to form a single text. One of our basic findings could be seen from Fig-4 that all these job postings have been extracted from several countries and United States has a greater number of job postings in general.

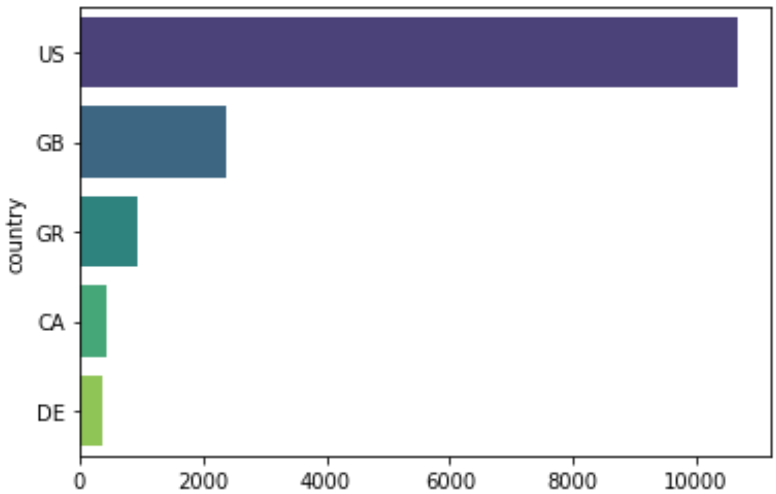


Fig-4

In addition to these, the dataset is highly imbalanced with 93% of the jobs being actual and the rest 7% being fraud. A count plot of the same can show the discrepancy clearly such as in Fig-5. Due to this data imbalance, the machine learning model might be biased towards the dominant class, and this is solved using one of the sampling techniques.

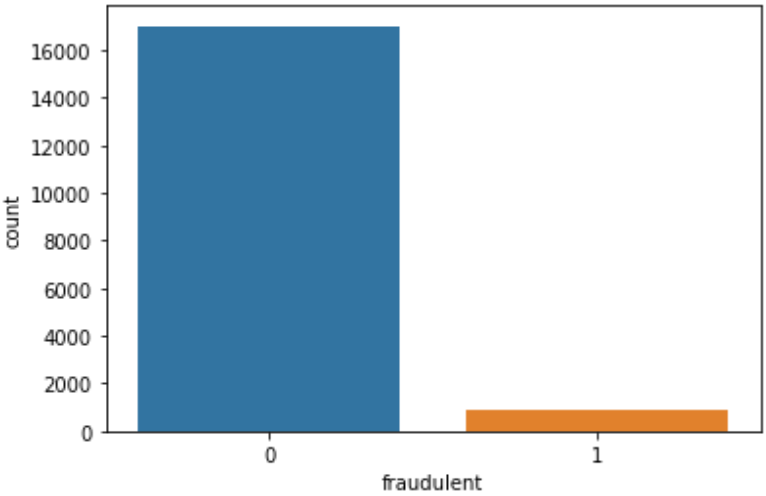


Fig-5

Now the relevant text data is combined in one column and the rest are excluded except the target column, for the dataset to be pre-processed for training. The column with text data has been cleaned by removing the stop-words, punctuations, case-normalisation and stemming using porter-stemming library.

To visualize the fraud and real job postings, the Word Cloud has been generated to see the top occurring keywords in the data. To do so, the text data for fraud (Fig-6) and real (Fig-7) job postings are separated and then the Word Cloud has been plotted accordingly.


Fig-6

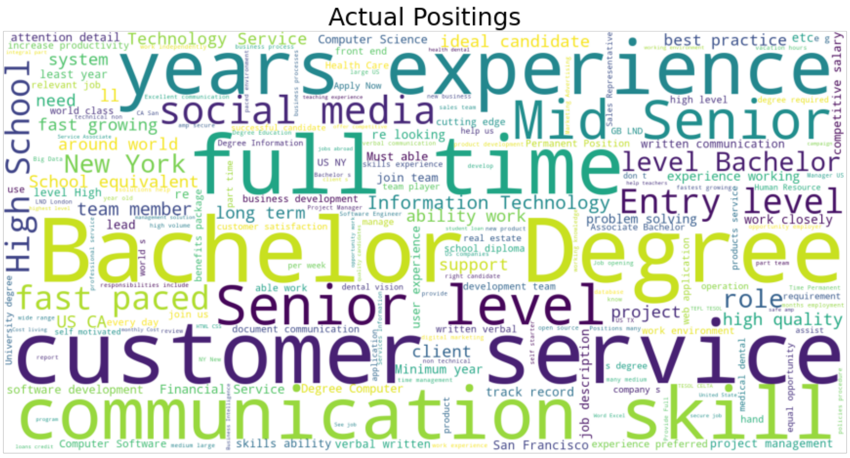


Fig-7

The job postings being fraud or real cannot be judged just by observing the obtained Word Clouds. Customer centric postings seem fake whereas the postings that require experience seem original, as can be inferred from the Word Clouds.

## 6. N-gram Analysis

N-gram analysis is implemented for words upto 4 grams and verified if some additional information can be obtained. But similar information is obtained like from the Word Cloud. The customer service related roles are large in number under each class.

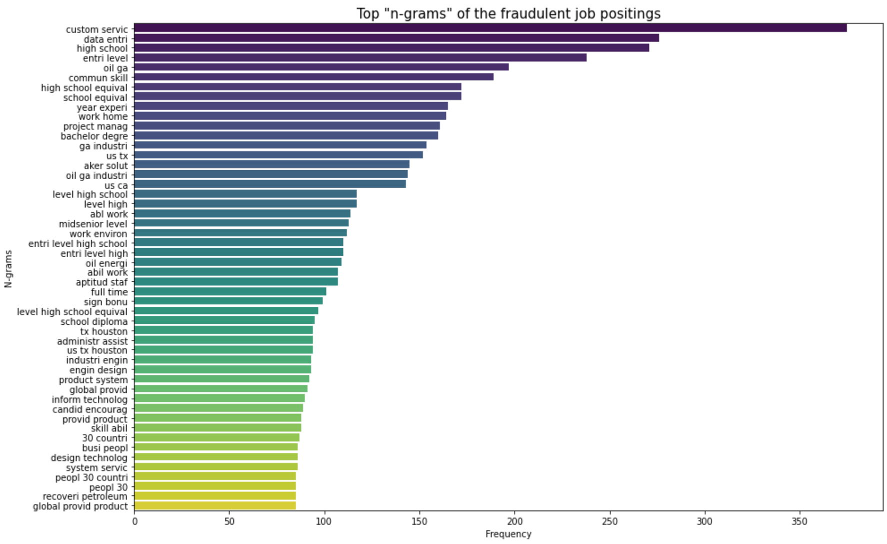


Fig-8

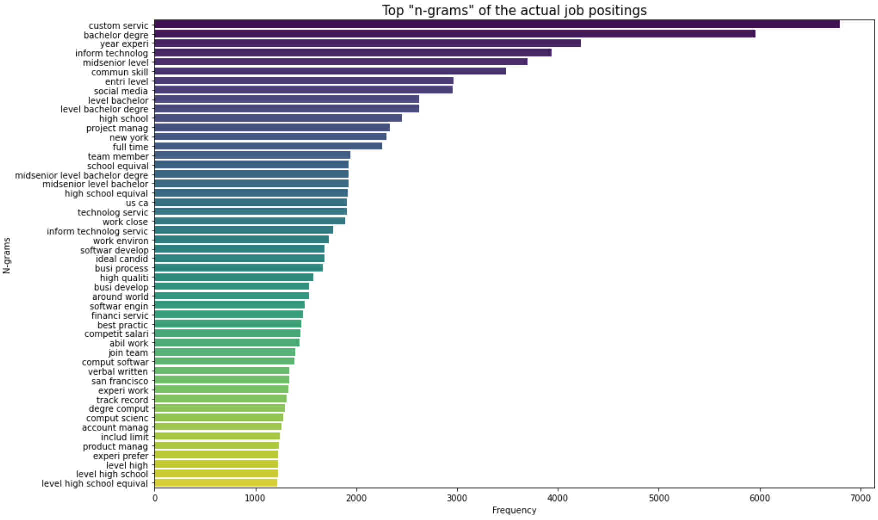


Fig-9

## 7. Modeling

The machine learning model is trained to classify the fraudulent and real job postings. We have used a pipleline to combined the cleaned text, vectorization and classification. The dataset is split into test and train in the ratio of 80:20. Stratify parameter is used for using train test split of sklearn package that makes the split so that proportion of values in the sample produced is the same as that of the original data set. This can be observed from the below figures Fig-10. This one is to make sure that test evaluation metrics are not biased. This is not to remove the class imbalance but to reduce the bias.

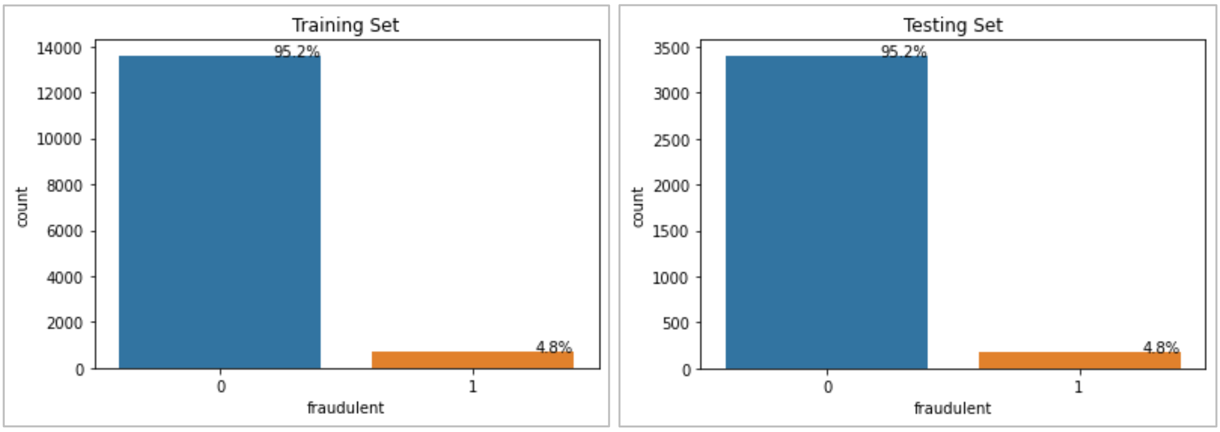


Fig-10

Then later, after splitting, the train and test data are transformed into Term Frequency – Inverse Document Frequency matrices using sklearn's TF-IDF vectorizer function and then the model has been fit using the training TF-IDF matrices and evaluated using the test. Here, are the tranformations as shown in Fig-12.

| | X_train | X_test |
|---|---|---|
| **Original** | **(14304,)** | **(3576,)** |
| After TF-IDF | (14304, 129948) | (3576, 129948) |

Fig-12 *TF-IDF Transformation*

## 8. Results and Discussion

Various Evaluation Metrics namely:

1) Recall,
2) Precision,
3) Accuracy,
4) F- measure are used to evaluate all the mentioned models.

| model | data_type | accuracy | error | precision | recall | f1_score |
|---|---|---|---|---|---|---|
| KNeighborsClassifier | train | 0.985389 | 0.014611 | 0.921603 | 0.763348 | 0.835043 |
| | test | 0.981544 | 0.018456 | 0.908397 | 0.687861 | 0.782895 |
| LogisticRegression | train | 0.975252 | 0.024748 | 0.997067 | 0.490620 | 0.657640 |
| | test | 0.969519 | 0.030481 | 0.984848 | 0.375723 | 0.543933 |
| RandomForestClassifier | train | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 |
| | test | 0.982103 | 0.017897 | 1.000000 | 0.630058 | 0.773050 |
| SVC | train | 0.995666 | 0.004334 | 1.000000 | 0.910534 | 0.953172 |
| | test | 0.980145 | 0.019855 | 1.000000 | 0.589595 | 0.741818 |

Fig-13

Clearly, Random Forest was overfitting and other models were not that well performing with respect to recall and F1-score. Since our data is imbalanced, there are high chances for the precision to be high and recall to be low. So, it is better to look at F1 score while dealing with such problems. As highlighted, KNN performs relatively well when F1-score is considered. Still, the problem of data imbalance exists, hence the minority class has been oversampled and models have been re-run.

| model | data_type | accuracy | error | precision | recall | f1_score |
|---|---|---|---|---|---|---|
| KNeighborsClassifier | train | 0.982732 | 0.017268 | 0.737234 | 1.000000 | 0.848745 |
| | test | 0.975391 | 0.024609 | 0.685590 | 0.907514 | 0.781095 |
| LogisticRegression | train | 0.992240 | 0.007760 | 0.861940 | 1.000000 | 0.925852 |
| | test | 0.980984 | 0.019016 | 0.772021 | 0.861272 | 0.814208 |
| RandomForestClassifier | train | 1.000000 | 0.000000 | 1.000000 | 1.000000 | 1.000000 |
| | test | 0.980425 | 0.019575 | 0.990476 | 0.601156 | 0.748201 |
| SVC | train | 0.999720 | 0.000280 | 0.994261 | 1.000000 | 0.997122 |
| | test | 0.984899 | 0.015101 | 0.983740 | 0.699422 | 0.817568 |

Fig-14

This sampling has been done to reduce the bias that the model has towards the dominant class. There are disadvantages to these kind of strategies; up sampling the minority class might add bias our model to emphasizing certain words, while down sampling the majority class might also add bias[2]. We have implemented oversampling since it has usually been used in dealing with imbalanced classes. However, recall here, seems alarmingly high. Not to create any additional bias than that already, as per the original data, without oversampling, KNN is suggested to be the benchmark model after careful evaluation.

## 9. Reflection and Improvement

The insights obtained from the model are- most  of the entry level jobs seem to be fraudulent and the Scammers seem to target job seeking people with Bachelor's degree or a high school diploma. The model can be further improved using suitable sampling technique to treat data imbalance, more like SMOTE analysis which involves synthetic generation of samples. Adding to this, models can be fine-tuned using cross validation to get the best hyper parameters.

## References


1. https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction
2. http://michael-harmon.com/blog/NLP1.html#fifth-bullet