Analysis and Classification of Job Postings as Fake / Legitimate

Chetana Chunduru
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
cchetan2@ncsu.edu

1. Introduction

In this time of impending recession in the IT sector, the number of cases of fraudulent job posting is increasing. According to the Federal Trade Commission, Americans were scammed out of $68 million in the first quarter of 2022 due to fraudulent business and job opportunities. Employment-related scams have been a persistent problem, but they increased in 2020 as criminals preyed on people who had lost their jobs due to Covid. Employment scams have only become more sophisticated, so it's critical to be cautious as a job seeker. To address this issue, our project aims to leverage the power of Machine Learning models to predict the genuineness of Job postings based on history data.

2. Background

Classification is a technique for categorizing data into a set number of groups. Its main goal is to determine which category/class a new data set belongs to. A classification model attempts to derive some conclusion from the training input values. It will predict the new data class labels/categories. Classification can be used to perform a wide range of tasks, including speech recognition, handwriting recognition, biometric identification, document classification, and so on.

Our project's goal is to be able to classify job postings as legitimate or not based on the data attributes used to train the model. Because we have the output label for the data points, this falls under the Supervised Machine Learning use case, and for our project, we are using classification models such as Decision Tree, K Nearest Neighbours, Bayes Classifier, etc. In addition, we hope to discover valuable patterns in the data by using various visualization techniques to gain a better understanding of the data.

3. Method

3.1 Data Selection

The dataset for this project can be found at the link: https://www.kaggle.com/datasets/shivamb/real-or-fake-fake-jobposting-prediction. The dataset consists of 17880 data points, of which 866 are classified as fake job postings under the 'fraudulent' attribute. It also consists of data elements such as job title, location of work, department of the job role, salary range and other metadata related to job profile. We are removing Job IDs as they do not contribute to any decision making process. The dataset also contains NULL elements in some of the attributes as well as duplicate entries - for which proper cleaning techniques are applied.

3.2 Data Pre-processing

The dataset used in this project has a lot of datapoints with the values of null, 'Not Applicable' and 'Unsatisfied'. We are replacing those values with the value 'No Info' because all three of these values are corresponding to the same thing. We are removing 282 duplicate rows from the dataset which reduces its size to 17598. Also, the dataset is poorly balanced as most jobs are legitimate, with only a few exceptions. Most of the data points are being classified as genuine or legitimate due to this when we analyzed the accuracy of our models.

We are therefore exploring SMOTE, which can be used to generate synthetic minority class samples. This is important as a well-balanced dataset would yield better results.

3.3 Data Transformation

There are 11 string attributes for which transformation is done to collect valuable information out of it. These categorical data points are converted to numeric data points to ease the data mining.

For the next part, the numeric / categorical part of the dataset will be converted to a different base system using Principal Component Analysis (PCA), which will help us extract only the dimensions along which the variance is the maximum. This would reduce the training overhead for the model and will result in lesser amount of time required for testing and evaluating multiple models.We are also exploring the use of Linear Discriminant Analysis (LDA) technique for the dimensionality reduction. We'll be comparing the performance based on these both and if possible, would devise our approach as per the results.

3.4 Data Mining

We are planning to evaluate the below classifiers for our problem statement-

- K- Nearest Neighbors (KNN)
- Decision Tree
- Support Vector Machine (SVM)
- Naive Bayes Classifier

Currently, we have trained different KNN models based on varying values of K and evaluated their performance. The models would be evaluated based on results from Cross Validation for the value of k=10.

3.5 Interpretation and Evaluation

There are different metrics upon which we can evaluate the performance of our models. Metrics such as Recall / Sensitivity, Precision, F1 Score etc can be utilized. We are specifically focusing on the Sensitivity metrics for evaluation.

The Sensitivity value of a machine learning model is a measure of its ability to detect positive instances. It's also referred to as the true positive rate (TPR) or recall. Sensitivity is used to assess model performance because it shows how many positive instances the model correctly identified. A model with high sensitivity will have few false negatives, implying that it will miss some positive instances. In other words, sensitivity assesses a model's ability to correctly identify positive examples. This is significant because we want our models to be able to detect all positive instances in order to make accurate predictions.

For our project, we have assumed the positive scenario to be a prediction for legitimate job posting and a negative job posting as Fraudulent. The scenario of having a False negative, i.e. predicting a job to be genuine whereas in actual it was fraudulent would carry a higher cost as it would have a critical impact on the applicant in terms of monetary status. Since, Sensitivity focuses on reducing the false negatives in the system, we are using this metric for evaluating the models.

4. Experiment Setup

We have implemented the code using the packages such as sklearn for model related functionalities, numpy and pandas for data manipulation, seaborn and matplotlib for data visualization. For collaboration purposes, we have used Git and Google Collab so as to work on the project in a distributed manner.

Currently, we have progressed with the K-Nearest Neighbor (KNN) classifier. We have created 4 different models for this classifier and below are the metrics that we have chosen for training the KNN classifiers–

- Distance metric: Euclidean distance

- k values: 1,3,5,9
- Training and testing division: 80:20

For the later part of project, we aim to implement cross validation for the above KNN models to choose the best model with the optimal k value. We aim to implement a generic flow for cross validation so as to be utilized for other classifiers as well, which would reduce code overhead.

5. Results

After experimenting, we got following test results-

- The data set has 17880 data points, of which 866 are classified as fake job postings under the ’fraudulent’ attribute as indicated in Figure 2.
- The data set has a total of 70103 null values over all columns.
- There are 282 duplicate rows which are removed from the data set.
- The unique counts for each column in the data is shown in the Figure 1.

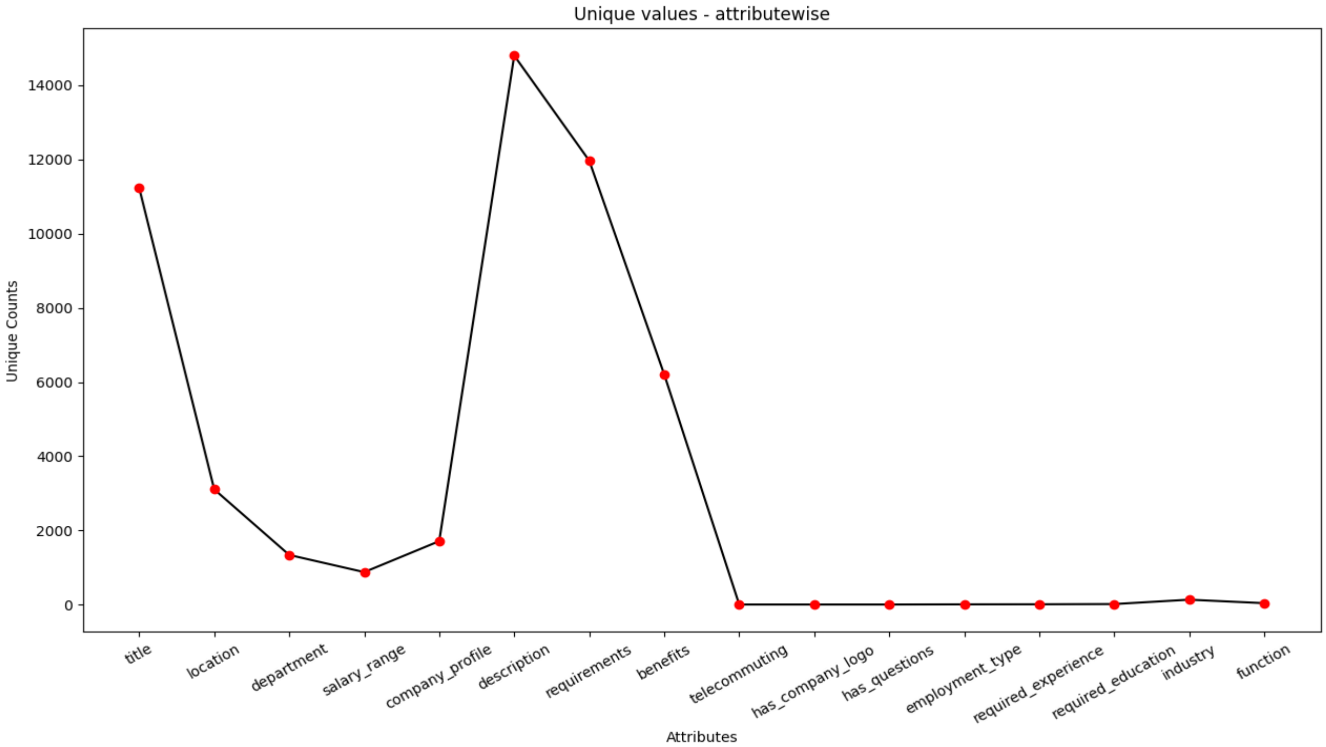
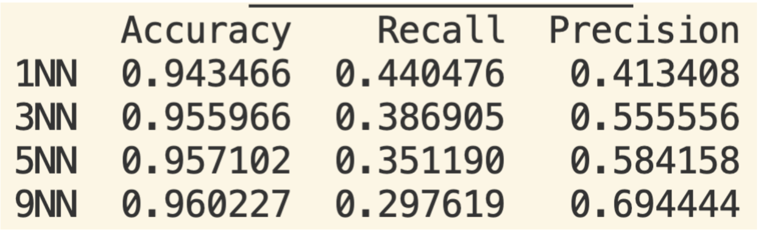


Figure 1: Unique values in the dataset

- Heatmap is used to find correlation between all attributes. This is properly visualized in Figure 3.
- We have listed the accuracy, recall and precision of KNN for different values of k (1, 3, 5, 9). Refer Figure 4 for this.



	Accuracy	Recall	Precision
1NN	0.943466	0.440476	0.413408
3NN	0.955966	0.386905	0.555556
5NN	0.957102	0.351190	0.584158
9NN	0.960227	0.297619	0.694444

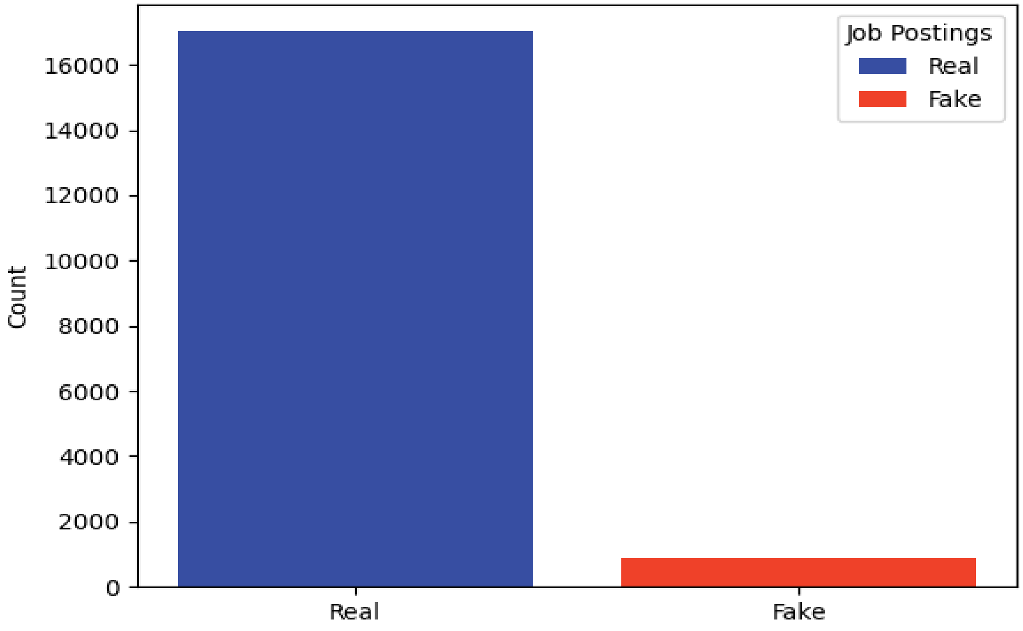


Figure 2: Job posting values in the dataset

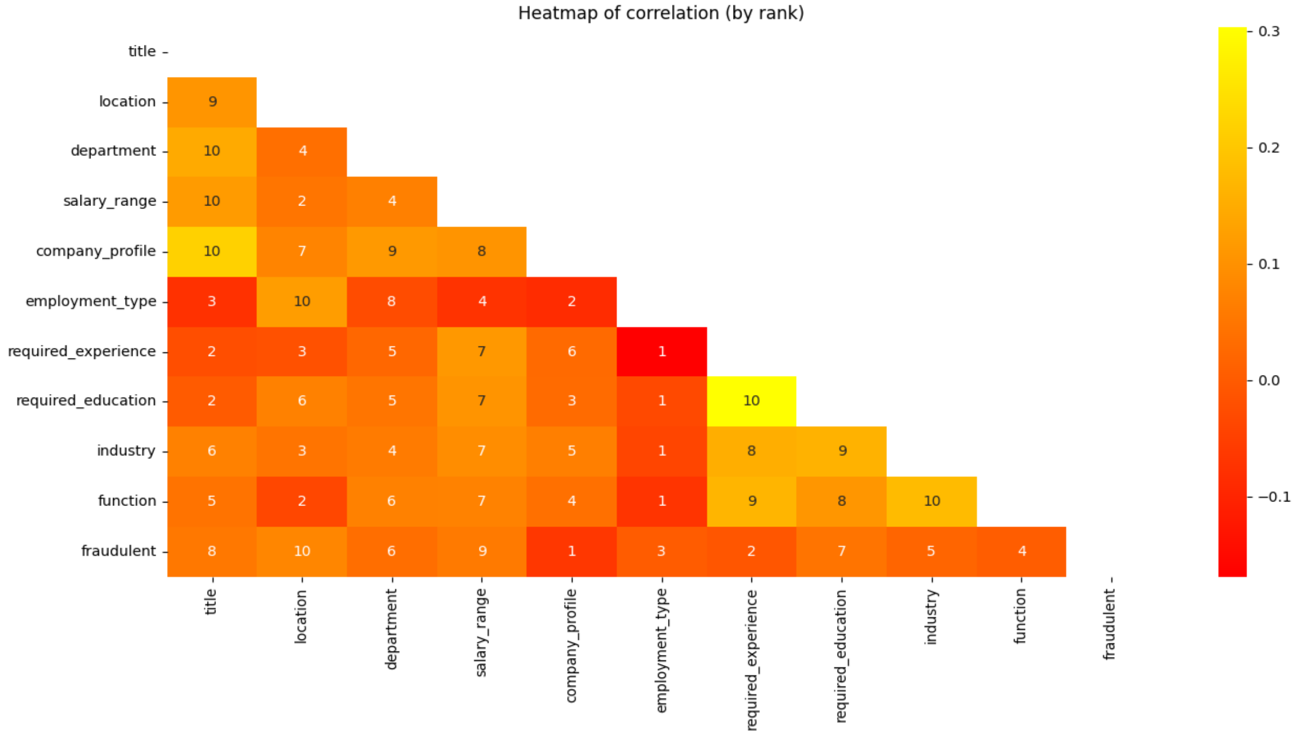


Figure 3: Heatmap of correlations

Figure 4: Evaluation Metrics for KNN

6. Conclusion

The real and fake job posting project taken was selected to solve a real world challenging issue. This project aims to provide a solution to this problem by classifying jobs as fake or real job postings. From the dataset selected for this problem, pertinent columns have been selected based on the initial data plot. The null values throughout the data have been generalized to a common value to maintain a standard value. The duplicates in the data set have been removed to have clean data. Correlation has been used to give the relation between different features and this coupled with a heatmap is used to identify the relevant features which are highly related to the class label. The dataset is then splitted into train and test sets to be used in the classification models.

Currently, as we have seen, the data is highly unbalanced according to the output class. To address this issue, we will try to implement Synthetic Minority Oversampling Technique (SMOTE) and try to resolve the imbalance problem. The future scope is to implement other classifiers and identify the best classification model for this dataset for predicting real and fake jobs.