Sinduja Ranganathan

BrainStation - Datascience

F2 Predictions - Fake or Fact

**Fake Job Posting Predictions**

1. **Introduction:**

During this pandemic phase, unemployment rate is in great increase as more and more businesses are struggling to cope with the slowing down economy. In these desperate times, when thousands and millions of people are looking out for job, it provides the perfect opportunity for the scammers. If the job seeker looks very carefully, these fake jobs can be identified. In most of these postings, they will have titles such as ‘Immediate opening’ or will have exceptionally high salary or during an interview they might ask you for personal confidential information such as your credit card details by saying they need it for personnel verification. There are many websites which lists these traits of a fraud job postings.

In this project, I would like to discuss how through machine learning models we can predict whether a job posting is fraudulent or not.

1. **Dataset:**

The dataset for this problem was taken from Kaggle and it has 17880 job postings. There are totally 18 columns and their descriptions are given below:

1. ***Job\_id:*** Job identification number (Index column)
2. ***Title:*** The title of the job ad entry
3. ***Location:*** Geographical location of the job (country, state and city)
4. ***Department:*** Corporate department
5. ***Salary\_range:*** Indicative salary range
6. ***Company\_profile:*** A brief company description
7. ***Description:*** The details about the job
8. ***Requirements:*** Requirements for the job opening
9. ***Benefits:*** Benefits given by the employer
10. ***Telecommuting:*** True for telecommuting positions
11. ***Has\_company\_logo:*** True if the company logo is present
12. ***Has\_questions:*** True if screening questions are present
13. ***Employment\_type:*** Full-time, Part-time, Contract, etc
14. ***Required\_experience:*** Executive, Entry-level, Intern, etc
15. ***Required\_education:*** Doctorate, Master’s degree, Bachelor’s, etc
16. ***Industry:*** Automotive, IT, Health care, Real estate, etc
17. ***Function:*** Consulting, Engineering, Research, Sales, etc
18. ***Fraudulent:*** 1 if fake posting else 0

***2.1 Duplicates:***

There are totally 281 duplicated rows in this dataset. These rows were dropped from our dataset for further analysis.

***2.2 Missing Values:***

In this dataset, ‘Fraudulent’ column is the target variable and other columns are the feature variables. Almost all the features have missing values except for ‘Title’, ‘telecommuting’, ‘has\_compnay\_logo’, ‘has\_questions’ columns.

‘Description’ and ‘Salary\_range’ has more than 50% of the data missing and hence I choose to drop these columns from the dataset.

***2.3 Data clean up:***

* Job\_id looks just as the index column so I will drop that column
* ‘Location’ variable has the city, state and country information in a list. There are few data points which has more than one city in the list. There are two option to handle this, we can either duplicate the rows for each city in the job posting or just retain the country information of the job posting. I chose to retain just the country information, as I don’t want to duplicate the data as it might magnify the bias that may be present in the data.

***2.4 Handling Missing values:***

* I replaced the Nan value in the Country column with a new value ‘Unknown’.
* For all other categories, there was either ‘Unspecified’, ‘Not Applicable’ or ‘Other’ value present in the dataset. Hence the Nan values were grouped under one of the category that was present in the feature.

1. **Exploratory Data analysis**

***3.1 Distribution of target Variable:***

When we explored the distribution of the target variable we can see that the dataset is very imbalanced. This dataset has only 856 job postings (~4.8%) to be fraudulent. The dataset is very imbalanced which could be a challenge to the model as it does not have enough information about the fraudulent class.

***3.2 Insights from EDA:***

* Most fraudulent jobs did not have a company logo while real jobs did have a company logo in most cases.
* Fraudulent job description talks more on what they were looking for, while the real job post talks more about what the applicant would be performing in that role.
* We observe that fraudulent job post predominantly did not have a company profile while the real job post did have.
* In company profile, fraudulent job post had words like your career, signing bonus which tries to talk about the applicant and attract them to apply for it. But the real job post had words like clients, we which talks more about the company.
* Fraudulent job post has very less number of characters in the company profile when compared to the real job post.
* Most fraudulent job posting have title related to 'Customer Service Representative' and 'Administrative Assistant'
* There is a high percentage of fraudulent jobs for the ‘part-time’ employment type.
* It looks like the jobs with ‘Administrative’ function in various industries has the highest percentage of fraud. As these functions could have more generic description and requirements may be this makes it easier for the scammers define it.
* The fraudulent jobs posts are high in school level and college level educational requirement.
* There are more fraudulent jobs in the ‘Entry-level’ position in required\_experience field as it echo the fact there are more job post on lower education level.

1. **Feature Selection:**

For my modelling, I selected my features based on the insights derived from my EDA. I selected the following features:

* Title
* Country
* Company\_profile
* Description
* Employment\_type
* Required\_education
* Required\_experience
* Function
* Has\_company\_logo

1. **Data Preprocessing:**

As part of the data pre-processing step, I combined the ‘title’, ‘description’ and ‘company\_profile’ into a single column ‘text’. This column was then tokenised using the NLP technique TF-IDF vectoriser, after removing the punctuation, stop-words and applying stemming technique.

All the categorical features were one hot encoded using the OneHotEncoder( ) in sklearn.

This transformation was done using the column transformer function in scikit learn.

1. **Over Sampling:**

The data was over sampled using SMOTE technique. SMOTE technique creates synthetic data points of the minority class in the feature space instead of creating copies. These synthetic data points are created in the feature space after defining the boundary of the minority class using the existing values in the minority class.

1. **Modelling:**

As part of my modelling I chose three different classification models Logistic Regression, Random forest and XG boost to solve the problem statement. These models were first run on the imbalanced dataset to evaluate the base performance of the models and then it was re-run on over sampled dataset. The model performance metrics were then compared between the models for model selection.

Each of the model is hyper parameter tuned using pipeline and grid search cross validation.

Please refer to the Imbalanced\_dataset.ipynb and SMOTE\_dataset.ipynb notebooks for the pipeline and hyper parameters used for tuning the models.

1. **Model Selection:**

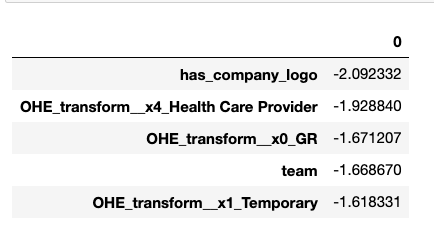
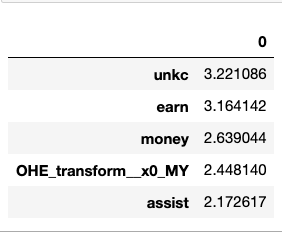
Since I had imbalance in my dataset I used recall as my model performance measure. Please find the model performance metrics for all the models below:

|  |  |  |
| --- | --- | --- |
| Models | Precision | Recall |
| Logistic Regression | 0.97 | 0.35 |
| Random Forest | 1.00 | 0.65 |
| XG Boost | 0.97 | 0.75 |
| Logistic Regression – SMOTE | 0.80 | 0.81 |
| Random Forest – SMOTE | 0.99 | 0.68 |
| XG Boost – SMOTE | 0.95 | 0.79 |

We could see from the above table that, ‘Logistic Regression – SMOTE’ has the highest recall percentage. Though the precision is not that great in this model I still chose logistic regression model for two reasons:

1. The explainability power of logistic regression
2. I feel in this problem statement the cost of recall is much higher than the cost of precision. That is the risk of sharing your personal information to a scammer is much higher losing a job opportunity by not applying for a real job.

The final model was then run on the train data with the parameters of the fitted logistic regression model with SMOTE and the coefficients were explored. Below are top 5 token with positive and negative coefficients for the model

An example of how we could interpret the above results for negative co-efficients is, when:

* ‘company\_logo’ is not present
* function is not health care provider
* country is not GR
* employment type not temporary

There is a high probability of it being a fraudulent job post

Similarly, for positive co-efficients, when:

* Company profile is missing
* The words ‘Earn’, ‘Money’, ‘assist’ are present
* When it is for the country MY

There is a high probability of the job posting to be fraudulent.

Please refer to the model\_coefficients.ipynb notebook for the code used.

1. **Future Actions:**

This model can further be improved by applying Doc2Vec vectoriser to identify if there are any similarities between the fraudulent job descriptions. Also, we could apply PCA for dimensionality reduction and check for the model performance. When we apply PCA we will loose the explainability power and hence we could chose even XG Boost for better results.

We can probably develop any application which will take the input from the job posting and could tell the job seeker if there could be a chance of the job being fraudulent. (Note: This was tried on streamlit but this app had a restriction of not being able to store the input values and run the model)

If this application is developed it could save many people from the scammers and could help us protect our confidential information.