Fake Job Posting Prediction

Abstract

In this project, we will predict fake job postings using Machine Learning techniques on the text job description data that were taken from Kaggle. A detailed understanding of data and its preprocessing is done before applying classification models.

Keywords - Fake Job, Classification models, NLP

1. INTRODUCTION

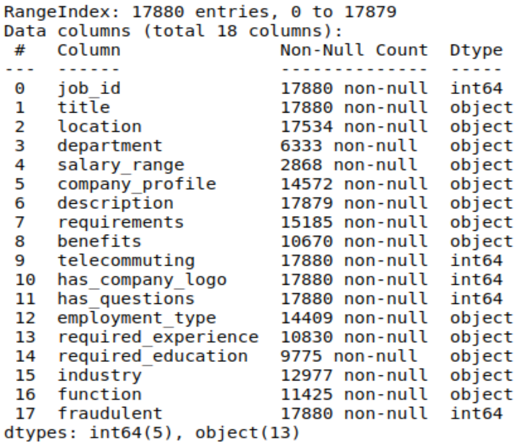
Employment is one of the most important part of a country’s economy as well in an individual’s survival and contribution. There has been a lot of frauds and scams in the job market in the recent pasts. And despite this, during this covid-19 pandemic, due to higher unemployment rate there is a good chance of people getting trapped into this fake job posts, so, here through this project. We can contribute to this jobs posting industry to classify fake jobs.

1.1 Problem Description and Data Understanding

This dataset contains 18K job descriptions out of which about 800 are fake. The data consists of both textual information and meta-information about the jobs. This dataset is used to train classification models which can learn the job descriptions which are fraudulent.

2. EXPLORATORY DATA ANALYSIS AND PREPROCESSING

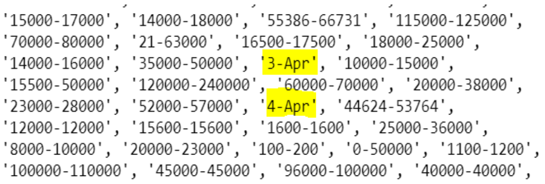
This data set has 17880 rows (job description) and 18 columns (attributes-both text and binary varibles present).



```
RangeIndex: 17880 entries, 0 to 17879
Data columns (total 18 columns):
 #   Column               Non-Null Count  Dtype 
---  ------               --------------  ----- 
 0   job_id               17880 non-null  int64 
 1   title                17880 non-null  object
 2   location             17534 non-null  object
 3   department           6333 non-null   object
 4   salary_range         2868 non-null   object
 5   company_profile      14572 non-null  object
 6   description          17879 non-null  object
 7   requirements         15185 non-null  object
 8   benefits             10670 non-null  object
 9   telecommuting        17880 non-null  int64 
 10  has_company_logo     17880 non-null  int64 
 11  has_questions        17880 non-null  int64 
 12  employment_type      14409 non-null  object
 13  required_experience  10830 non-null  object
 14  required_education   9775 non-null   object
 15  industry             12977 non-null  object
 16  function             11425 non-null  object
 17  fraudulent           17880 non-null  int64 
dtypes: int64(5), object(13)
```

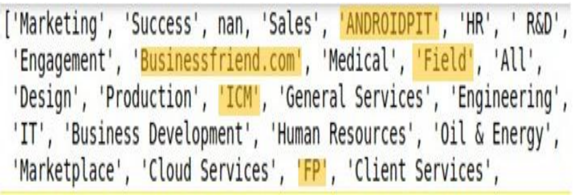
Among all attributes, Job Id is useless in identifying the fraudulent jobs so, we drop this attribute. In the given data, there are duplicates which are dropped. And also the attributes salary range and department has lots of null values.

The salary range attribute has 84% null data and the remaining 16% is noisy so we dropped this attribute.



```
'15000-17000', '14000-18000', '55386-66731', '115000-125000',
'70000-80000', '21-63000', '16500-17500', '18000-25000',
'14000-16000', '35000-50000', '3-Apr', '10000-15000',
'15500-50000', '120000-240000', '60000-70000', '20000-38000',
'23000-28000', '52000-57000', '4-Apr', '44624-53764',
'12000-12000', '15600-15600', '1600-1600', '25000-36000',
'8000-10000', '20000-23000', '100-200', '0-50000', '1100-1200',
'100000-110000', '45000-45000', '96000-100000', '40000-40000',
```

The department attribute also has 65% of the data as null and the remaining 35% is noisy, so, we dropped this attribute.



```
['Marketing', 'Success', nan, 'Sales', 'ANDROIDPIT', 'HR', ' R&D',
 'Engagement', 'Businessfriend.com', 'Medical', 'Field', 'All',
 'Design', 'Production', 'ICM', 'General Services', 'Engineering',
 'IT', 'Business Development', 'Human Resources', 'Oil & Energy',
 'Marketplace', 'Cloud Services', 'FP', 'Client Services',
```

Now the data has 17599 rows and 15 columns.

2.1 Data Visualization:

The correlation between the following binary attributes has_question, has_company_logo, telecommuting and fraudulent is represented using heatmap.

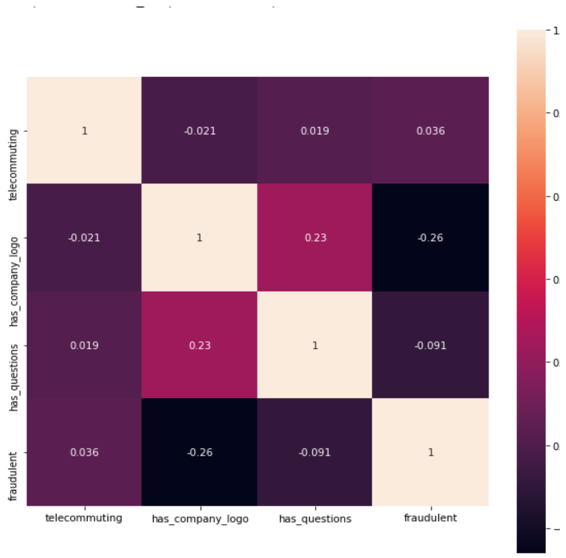
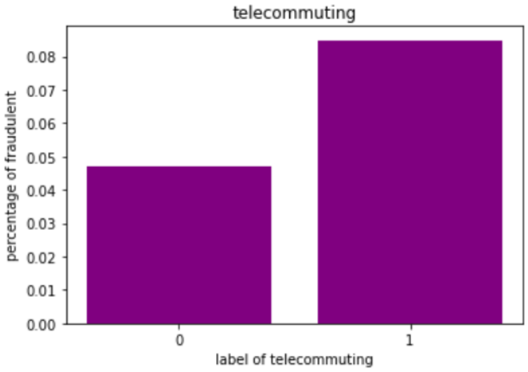


Fig: Heat-Map between Binary Attributes

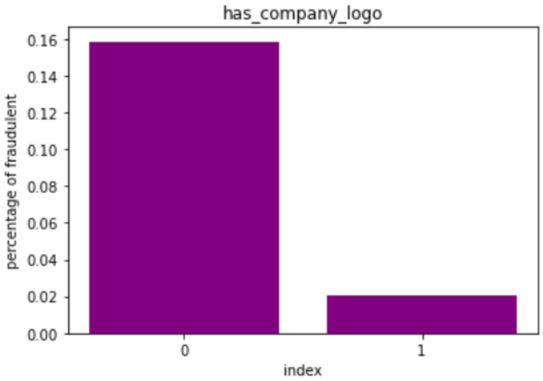
From the above fig. it can be seen that there isn't any correlation between the attributes. So, we analyze each attribute individually.

Now for data understanding we are visualizing each attribute in correspondence to real and fake jobs.

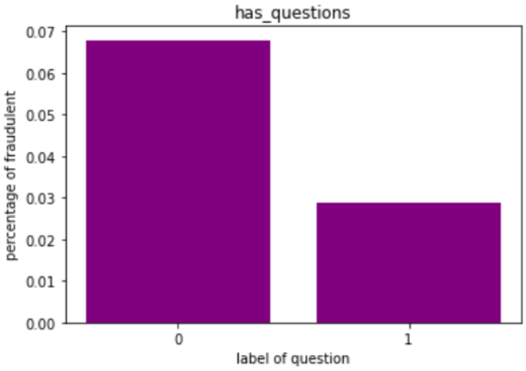
I. Jobs having telecommuting option has more chance to be fraudulent.



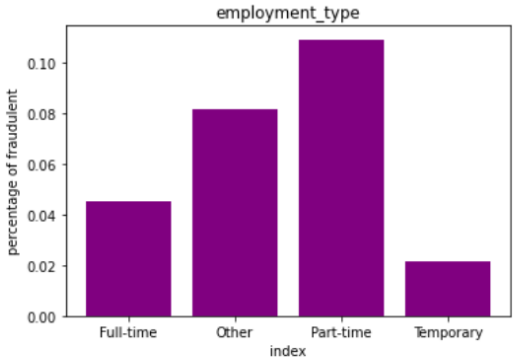
II. Jobs without company logo has more chance to be fraudulent.



III. Jobs without questions has more chance to be fraudulent.



IV. Jobs with employment type as part time has more chance to be fraudulent.



V. Jobs with Some High School Coursework as required education has more chance to be fraudulent.

VI. Jobs with required experience as Not Applicable has more chance of being fraudulent.

2.2 Text Pre-processing:

Next, we combine the description, requirement and company-profile attributes into a new attribute description. Here we split the location attribute to a new attribute Country code. After which we use Py-Country library to create a new attribute specifying Country name derived from the country code.

2.2.1 Dropping insignificant columns

After these steps, we remove some insignificant columns like Location, Country-code, description, requirement and company-profile attributes, since they do not contribute anything in the analysis.

Then Using the word-cloud we displayed the words of different size which changes with respect to the frequency of their occurrence for both fraudulent and genuine jobs. Then we also visualised these words in terms of their count in the right bar plots.

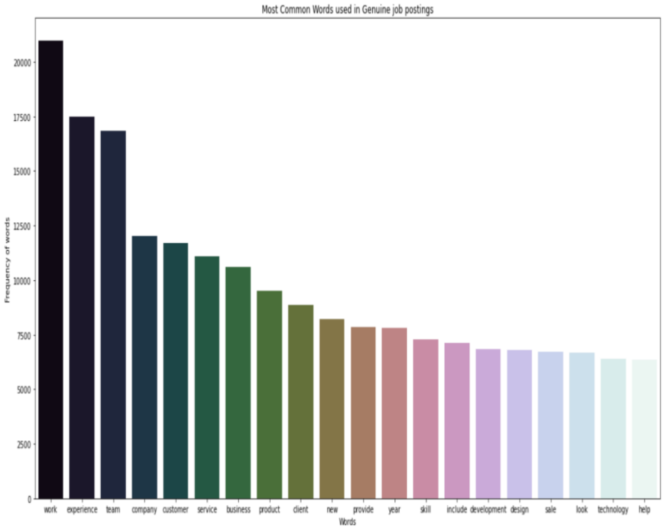


Fig: Most common words used in genuine job posting

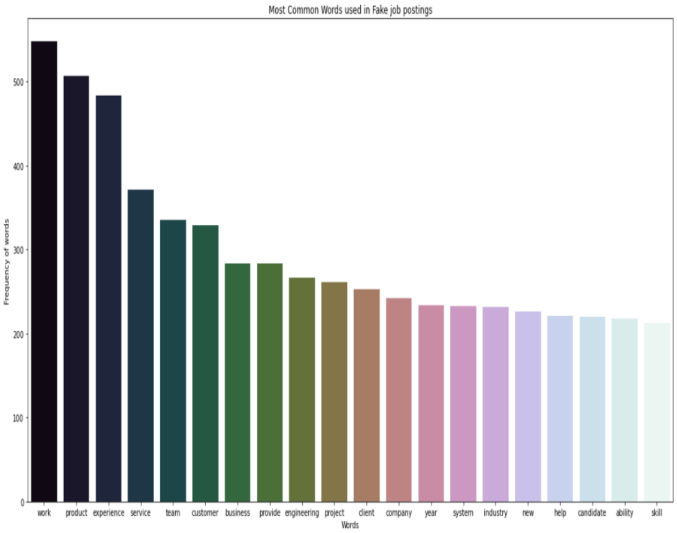


Fig: Most common words used in fake job postings

Then we used tokenizer for converting text data into tokens, eliminate the stop-words and applied lematisation.

We then use count-Vectorizer for vectorization of the text corpus to create new attributes and One Hot Encoding for categorical variables into new binary variables. Finally the cleaned data is The most common words used in genuine and the fraudulent jobs were also displayed with the help of bar graph.

3. Modelling

We use different type model such like logistic regression, KNN etc.

3.1 Model Evaluation

- To find best model among all applied models we have evaluate each model.
- Model evaluation is depending on the type of label class.
- In this problem, the label class have binary type value. We can be evaluate the model by *Classification Report.*

3.1.1 Classification Report-

There are five parameters to measure a performance of a model.

(a) Accuracy

$$\boxed{\frac{TP+TN}{TP+TN+FP+FN}}$$

(b) Precision

$$\boxed{\frac{TP}{TP+FP}} \quad \textbf{or} \quad \boxed{\frac{TN}{TN+FN}}$$

(c) Recall

$$\boxed{\frac{TP}{TP+FN}} \quad \textbf{or} \quad \boxed{\frac{TN}{FP+TN}}$$

(d) F-1 Score – F-1 score is good metric when data is highly imbalanced.

$$\boxed{\frac{2 * Precision * Recall}{Precision + Recall}}$$

(e) ROC – AUC Score-

The area under the curve is the measure the performance of a model to distinguish between classes and is used as a summary of ROC curve.

Where,
TP-True Positive
FP-False Positive
TN-True Negative
FN-False Negative

3.2 Models

(a) Logistic Regression-

	Precision	Recall	F-1 score	Support
Positive	0.26	0.16	0.2	61
Negative	0.97	0.98	0.98	1638
Accuracy			0.95	1699
Macro Avg	0.61	0.57	0.59	1699
Wt. Avg	0.94	0.95	0.95	1699

ROC -AUC Score – 0.5731

ROC-AUC score value is good for imbalanced data

(b) KNN-

	Precision	Recall	F-1 score	Support
Positive	0.00	0.00	0.00	61
Negative	0.96	1.00	0.98	1638
Accuracy			0.96	1699
Macro Avg	0.48	0.50	0.49	1699
Wt. Avg	0.93	0.96	0.95	1699

ROC-AUC Score – 0.5

Due to less ROC-AUC score, KNN performed poorly than Logistic Regression.

(c) Support Vector Classification (SVC)-

	Precision	Recall	F-1 score	Support
Positive	0.25	0.05	0.08	61
Negative	0.97	0.99	0.98	1638
Accuracy			0.96	1699
Macro Avg	0.61	0.52	0.53	1699
Wt. Avg	0.94	0.96	0.95	1699

ROC-AUC Score – 0.5218

ROC-AUC score between score of Logistic regression and KNN.

(d) Random Forest Classifier-

	Precision	Recall	F-1 score	Support
Positive	0.50	0.02	0.03	61
Negative	0.96	1.00	0.98	1638
Accuracy			0.96	1699
Macro Avg	0.73	0.51	0.51	1699
Wt. Avg	0.95	0.96	0.95	1699

ROC-AUC Score – 0.5079

Random forest is better than KNN but perform poorly than Logistic and SVC.

(e) Sklearn's MLP Classifier (solver = 'sgd')

	Precision	Recall	F-1 score	Support
Positive	0.15	0.08	0.11	61
Negative	0.97	0.98	0.97	1638
Accuracy			0.95	1699
Macro Avg	0.56	0.53	0.54	1699
Wt. Avg	0.94	0.95	0.94	1699

ROC-AUC Score – 0.5321

The MLP Classifier with ‘sgd’ solver performs slightly better than the KNN model but is still worse than what Logistic Regression reported.

(f) Sklearn's MLP Classifier (solver = 'adam')

	Precision	Recall	F-1 score	Support
Positive	0.25	0.15	0.19	61
Negative	0.97	0.98	0.98	1638
Accuracy			0.95	1699
Macro Avg	0.61	0.57	0.58	1699
Wt. Avg	0.94	0.95	0.95	1699

ROC-AUC Score – 0.5655

This is better than all expect Logistic Regression. So Logistic regression is best model among all.

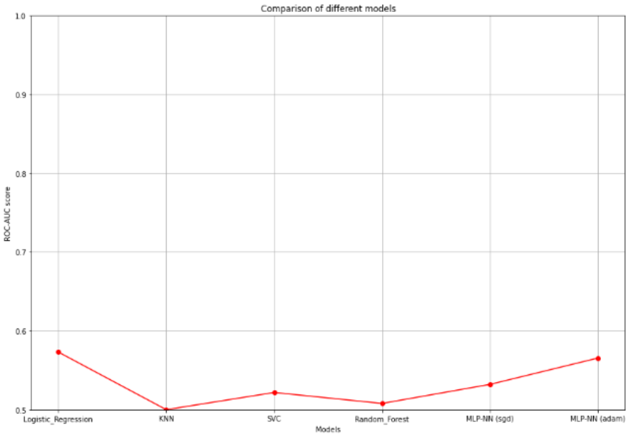


Fig: Comparison of different Models

4. Conclusion

Due to current scenario many employees are being laid off every day and the demand of jobs is much higher than number of posts available in the market which is providing the perfect opportunity for online scammers to take advantage of that. Hence predictive analysis is very much helpful to filter such type of fake jobs offered by scammers.

As we have performed various models to predict whether a given job description is fake or not we can see the best ROC-AUC score (.5731) is given by Logistic regression among other models with the 95% accuracy of predicting whether a job description is fraudulent or not, which provides perfect opportunity.