Identifying fake job posts

Himadri Halder(1932300012), Akamal Khan(193230013), Girish Maske(193230014), Shubham Bhise(193230015),

June 10, 2020

Problem Definition

It is a supervised learning problem in which the classification models are created which can learn the job descriptions which are fraudulent.

The dataset used in this project contains 18K job descriptions out of which about 800 are fake. The data consists of both textual information and meta-information about the jobs.

Method or Outline

Following are the objectives of this project.

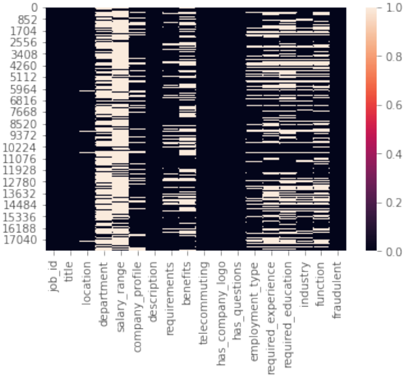
1. Perform data cleaning and Exploratory Data Analysis on the dataset to identify interesting insights from this dataset.
2. Identify key traits/features (words, entities, phrases) of job descriptions which are fraudulent in nature.
3. Create a classification model that uses text data features and meta-features and predict which job description are fraudulent or real.
4. Compare the different classification models by their accuracy and precision.

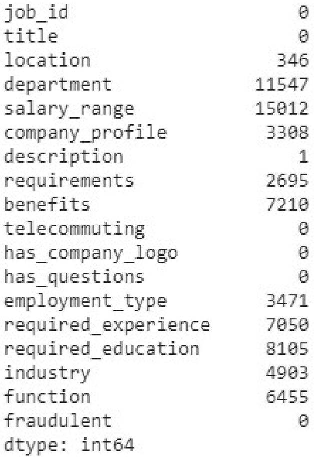
Contribution of the team members

After selecting a supervised learning classification problem, we found that the data has too many null spaces and some features have information in paragraphs or textual format. For this we need to search and learn the methods for cleaning or processing the data (such as bfill, ffill, drop). Also we got to learn alot about the NLP toolkit while doing this project. It was important to learn that how to extract the main words or information from the sentences. After this, in prediction part we tried to use gridsearchcv. It is a very useful function for performing cross validation and hyper parameter tuning. So in the end we all discussed, searched and analysed the things. Still the main focus of 2 members 193230012 and 193230013 were on data cleaning and EDA and others 193230014 and 193230015 have more contributed to NLP, prediction and hyperparameter tuning part.

Data cleaning

The following heatmap shows features and null values in this dataset.





```
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
dtype: int64
```

Figure: Null values in original dataset

Data cleaning(Contd...)

Data is cleaned by performing the following actions on certain features.

- Dropped: job id, department, salary range, benefits.
- Back filled: employment type, required experience, required education, industry, function.
- Row elimination: remaining data
- Duplicate rows were removed
- Concatenate: description, requirements, company profile
- Features like country and city are separated from location

Reasons of these actions are mentioned below the respective codes in the code file.

With this the data dimension is reduced from (17880, 18) to (11272, 13)

Exploratory Data Analysis(EDA)

Following count plots and bar graphs are drawn from the cleaned data gives some insights.

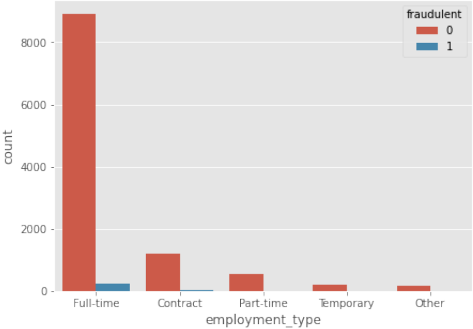


Figure: Feature employment type

It shows that the more fake job posts are contained with full time employment offer.

Exploratory Data Analysis(EDA)Contd...

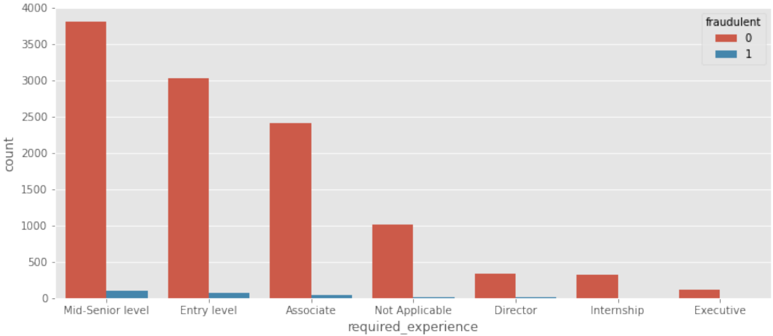


Figure: Feature experience

Here we can see that more fake posts having demanded experience of mid-senior level than entry level.

Exploratory Data Analysis(EDA)Contd...

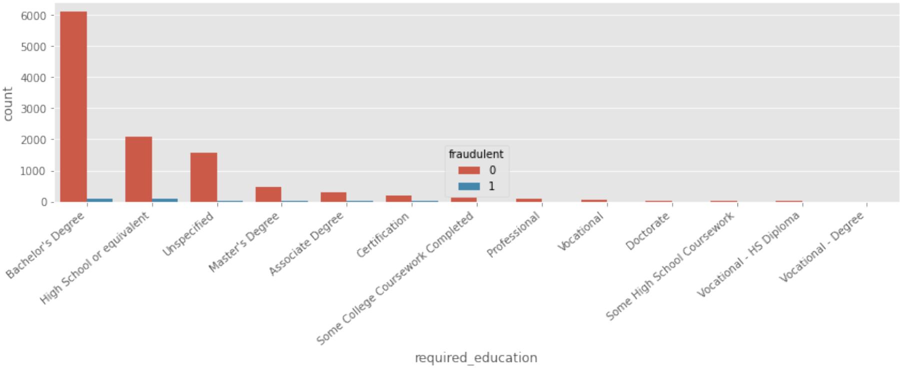


Figure: Feature education

Bachelors degree and High school passed out are the most demanded categories in the fraudulent posts.

Exploratory Data Analysis(EDA)Contd...

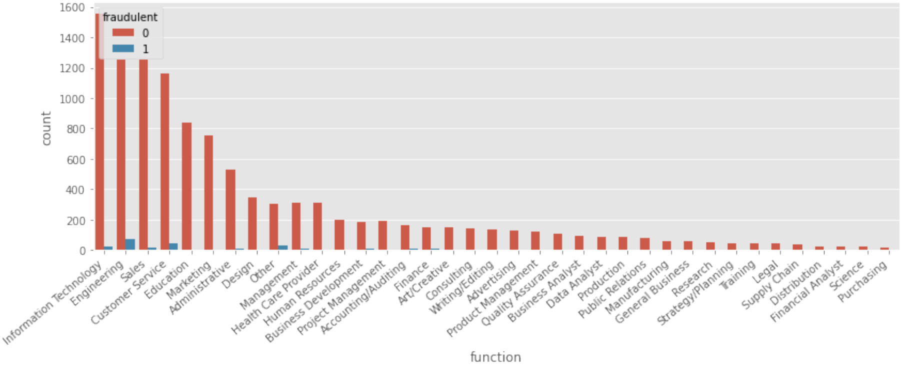


Figure: Feature Function

Functions or sectors like engineering, Information technology and customer service are having more fake job posts.

Exploratory Data Analysis(EDA)Contd...

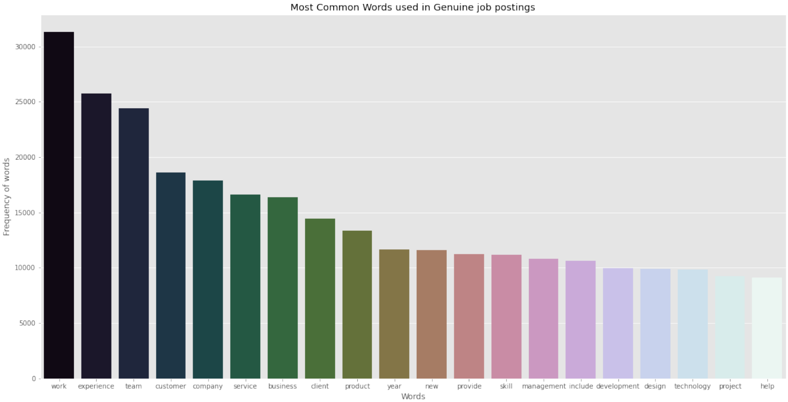


Figure: Most common words from the feature description used in genuine job postings

Exploratory Data Analysis(EDA)Contd...

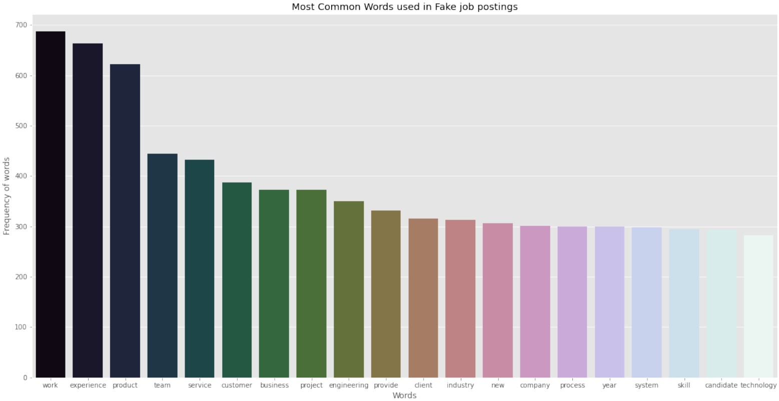
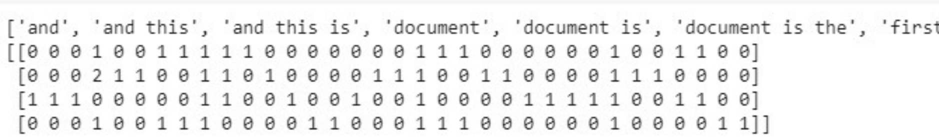


Figure: Most common words from the feature description used in fake job postings

From these last 2 graphs we can try to predict the fake post by finding out the main words from the concatenated feature description.

Natural Language Processing(NLP)

- In this section the features which have sentences and paragraphs are filtered and main or relevant words are converted to vectors.
- The two libraries used for this work are corpus from sklearn and spacy.
- Using this a stopwords and symbols are removed from sentences.
- And vectorize the the words like following. (ngram range = (1,3))



```
['and', 'and this', 'and this is', 'document', 'document is', 'document is the', 'first
[[0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 1 0 0]
 [0 0 0 2 1 1 0 0 1 1 0 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 1 1 1 0 0 0 0]
 [1 1 1 0 0 0 0 0 1 1 0 0 1 0 0 1 0 0 1 0 0 0 0 1 1 1 1 1 0 0 1 1 0 0]
 [0 0 0 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 1 1]]
```

Figure: Example of vectorization different from actual data

Feature Selection

After vectorization of feature description, other feature are one hot encoded.

- Significant features: decripction(description + requirements + company profile), employment type, required experience, required education, industry, function.
- Insignifcant features: title, city, country name.

Even though some features are insignificant for the prediction purpose but country name and city are separated from the feature location for cleaning the data. And these can be used for better visualisations like by plotting fraudulent cases on world map.

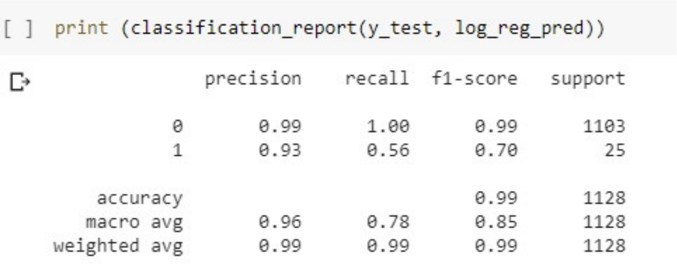
Prediction

- In this part we split the data into train test at 9:1 proportion.
- And used 3-fold cross validation. This method of CV is efficient because it test the train model on whole data.
- For hyper parameters (eg. c and gamma), a GridsearchCv function is used. It tests each model by making all combinations of mentioned hyper parameters and gives best fit model.
- We tested the following 4 models and compared their results at the end.

Prediction(Contd...)

Logistic Regression

After tuning it gives hyper parameters c = 1 and regularization selected as L2(lasso regression).



```
[ ]  print (classification_report(y_test, log_reg_pred))
```

```
              precision    recall  f1-score   support

           0       0.99      1.00      0.99      1103
           1       0.93      0.56      0.70        25

    accuracy                           0.99      1128
   macro avg       0.96      0.78      0.85      1128
weighted avg       0.99      0.99      0.99      1128
```

Figure: Classification report

Comparison

And like this we tested other models such as KNN, SVC and MLP classifier. Their results are compared by using ROC scores.

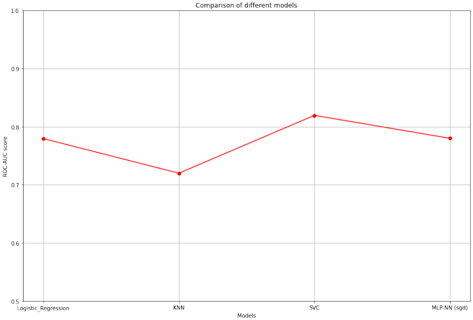


Figure: Comparison of models

It shows that SVC classifier gives best results for this dataset.