Architectural Decisions Document (ADD)

IBM CAPSTONE PROJECT

FAKE JOB DESCRIPTION PREDICTION

1.Introduction :

In the current scenario, there is too much job posting online and we got daily mail for this and that job posting in an organisation or a company. We look at it and it seems that this is the perfect job for you. But it is also possible that there is no such job and the job description is wrong. So my project is on fake job description prediction is to predict whether the job description is given is genuine or this is for some malicious and job is fake and they are trying to just to fool you.

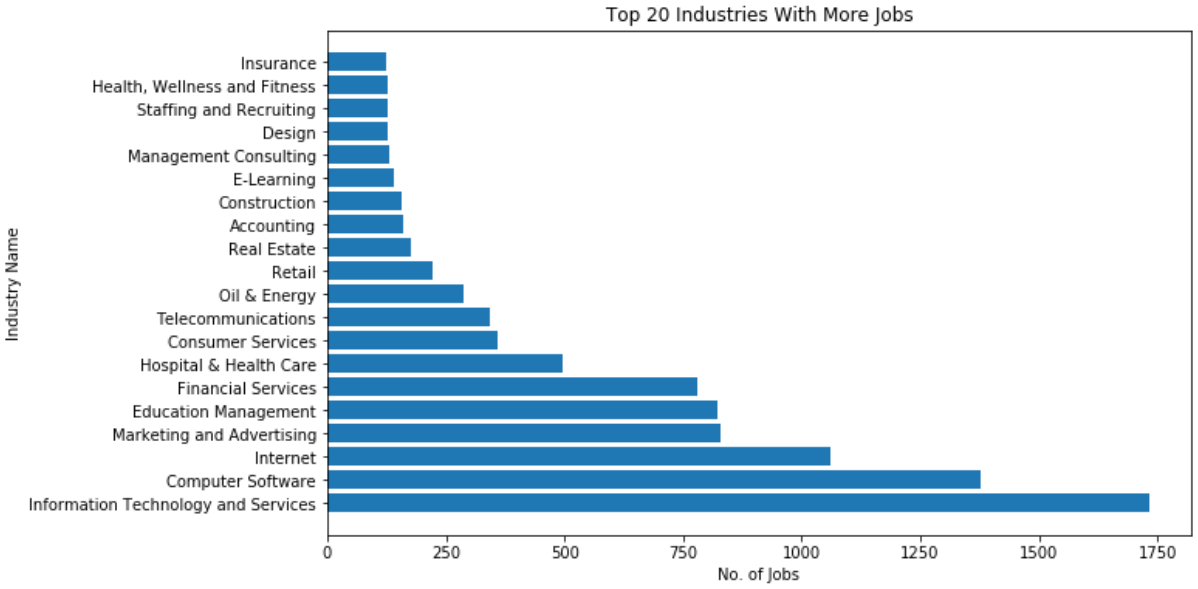
2. Data :

The data is collected from Kaggle fake job description prediction dataset. 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. The dataset can be used to create classification models which can learn the job descriptions which are fraudulent.

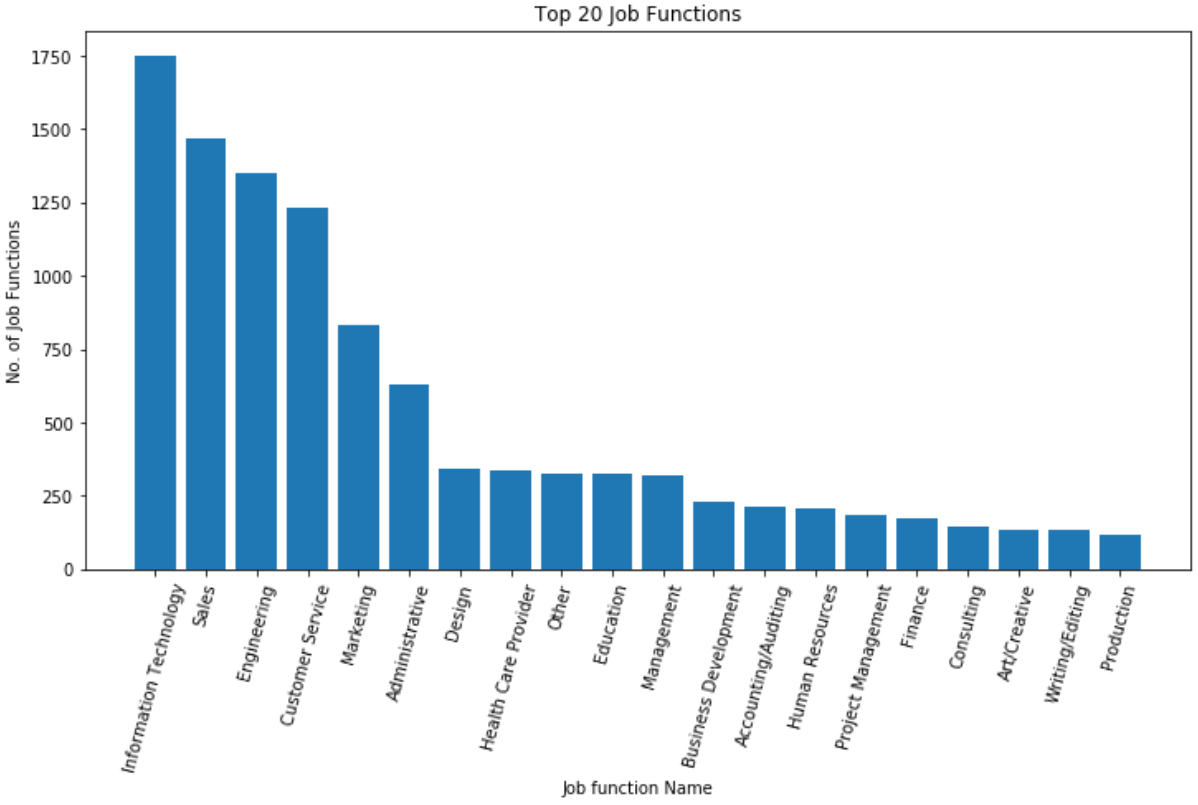
Data set contains categorical features as well as some feature containing text data. A quick overview of the features that dataset has:

```
job_id                int64
title                object
location             object
department           object
salary_range         object
company_profile      object
description          object
requirements         object
benefits             object
telecommuting         int64
has_company_logo      int64
has_questions         int64
employment_type      object
required_experience  object
required_education   object
industry             object
function             object
fraudulent            int64
dtype: object
```

From the dataset, we can see which industry is offering more jobs, the chart below shows that information



The chart below shows which job functions are more demand among the industries :



3.Data Cleaning and Preprocessing:

The dataset contains numerical and text data both and there are null also there in columns. Mostly numerical columns like telecommuting, has_company_logo, has_question have categorical data and text feature like employment_type, eductaion_requirment etc. So to deal with a categorical feature one-hot encoding is applied for the categorical columns. Preprocessing of text data are being done in several steps. The steps are as follow removing links from text data, lower all text data, remove stop words using nltk library and then check for most important words like which words have more occurrence and which words have less occurrence and according to that remove all non-important words. Finally, I deal with nan values by appending info not given with column name for each column so for categorical features they have one feature. To convert text data into numerical tfidf method is used.

4. Handling Imbalanced Dataset:

The dataset is highly imbalanced because it contains 866 fraudulent data and 17014 non-fraudulent data. Fraudulent data is only 5% of the dataset. So if we apply any machine learning algorithm without handling imbalanced data. The model gives us a high accuracy. But the model is not correct or proper. So to handle this problem I used resampling method. In this method oversampling to is used to in training data. What is basically is done that to make 50% ratio of each class we took fraudulent data and resample it of the size of non-fradulent data. Thus each class have an equal ratio in the training data and no more imbalance is there.

5. Applying Different Models:

In this project, I tried to apply several machine learning classification algorithms such as logistic regression, k nearest neighbour and random forest. Models are evaluated on precision score and recall score.

A. Result of Logistic Regression :

```
Train Data Accuracy :  0.9480567188303578
Test Data Accuracy :  0.9077181208053692

Confusion Matrix of Train Data :
 [[12552  1059]
 [  355 13256]]
Confusion Matrix of Test Data :
 [[3094  309]
 [  21  152]]
```

	precision	recall	f1-score	support
0	0.99	0.91	0.95	3403
1	0.33	0.88	0.48	173
accuracy			0.91	3576
macro avg	0.66	0.89	0.71	3576
weighted avg	0.96	0.91	0.93	3576

B. Result of K Nearest Neighbour :

```
Train Data Accuracy :  0.9530159429872896
Test Data Accuracy :  0.9544183445190156

Confusion Matrix of Train Data :
 [[13290   321]
 [  958 12653]]
Confusion Matrix of Test Data :
 [[3278  125]
 [  38  135]]
```

	precision	recall	f1-score	support
0	0.99	0.96	0.98	3403
1	0.52	0.78	0.62	173
accuracy			0.95	3576
macro avg	0.75	0.87	0.80	3576
weighted avg	0.97	0.95	0.96	3576

C. Result of Random Forest:

```
Train Data Accuracy :  0.9865549922856514
Test Data Accuracy :  0.9711968680089486

Confusion Matrix of Train Data :
 [[13431   180]
 [  186 13425]]
Confusion Matrix of Test Data :
 [[3359   44]
 [  59  114]]
```

	precision	recall	f1-score	support
0	0.98	0.99	0.98	3403
1	0.72	0.66	0.69	173
accuracy			0.97	3576
macro avg	0.85	0.82	0.84	3576
weighted avg	0.97	0.97	0.97	3576

6. Analysis:

As we can see in the results the logistic regression performs better for fraudulent data. Logistic regression has high recall value for test data very high compare to other models. But precision is very less. KNN model have also good recall and random forest have worst recall among all but the random forest model gives the best precision score for test fraudulent data. As logistic regression model gives the best recall and more correctly classify the fraudulent data, so it is the best model for the project.

7. Future Work:

Remaining is work is to apply a deep learning algorithm. To evaluate the model more correctly we can collect more data than the trained model more correctly.