SIES (NERUL) COLLEGE OF ARTS, SCIENCE AND COMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE
(ISO 9001:2008 CERTIFIED INSTITUTION)
NERUL, NAVI MUMBAI – 400706

PROJECT REPORT ON

ONLINE RECRUITMENT FRAUD DETECTION WITH COMPARATIVE PATTERN ANALYSIS ON MULTINANTIONAL AND INDIAN BASED FRADULENT JOBS

SUBMITTED BY

- KEDAR KOTWAL

UNDER THE GUIDANCE OF

ASST. PROF. MANASVI SHARMA

SUBMITTED IN THE PARTIAL FULLFILMENT FOR THE DEGREE OF

MSc. COMPUTER SCIENCE
SEMESTER – IV, 2021-2022

SIES (NERUL) COLLEGE OF ARTS, SCIENCE ANDCOMMERCE

NAAC ACCREDITED 'A' GRADE COLLEGE(ISO 9001:2015 CERTIFIED INSTITUTION) NERUL, NAVI MUMBAI – 400706

Certificate

THIS IS TO CERTIFY THAT THE PROJECT TITLED

ONLINE RECRUITMENT FRAUD DETECTION WITH COMPARATIVE PATTERN ANALYSIS ON MULTINATIONAL AND INDIA BASED FRADULENT JOBS

IS UNDERTAKEN BY

KEDAR PRASHANT KOTWAL

Seat No: **07**

In partial fulfillment of the MSc - IT / CS Degree (Semester IV) Examinationin the academic year **2021-2022** and has not been submitted for any other examination and does not form part of any other course undergone by the candidate. It is further certified that he/she has completed all the required phases of the Project.

Project Guide — External Examiner

Head of Department — Principal

ACKNOWLEDGEMENT

I extend my heartfelt gratitude and thanks to Asst. Professor Manasvi Sharma for providing me excellent guidance to work on this project and for their understanding and assistance by providing all the necessary information needed for my project.

I would also like to acknowledge all the staffs for providing a helping hand to us in times of queries & problems. The project is a result of the efforts of all the peoples who are associated with the project directly or indirectly, who helped us to work to complete the project within the specified time frame. They motivated me in the project and gave a feedback on it to improve my adroitness. Thanks to all my teachers, who were a part of the project in numerous ways and for the help and inspiration they extended to me and for providing the needed motivation.

With all Respects & Gratitude, I would like to thanks to all the people, who have helped for the development of the Project.

Kedar Kotwal

M.Sc. Computer Science Part(II)

SIES(Nerul) College of Arts, Science and Commerce

ABSTRACT

Online job portals have an enormous number of job adverts uploaded on them every day. This project aims to detect the job adverts which have more possibility of being fraudulent based on some factors like, whether a company logo is present on the advert, is there a job description, whether the description very short, etc.

For this project, two datasets are collected, one has internationally available jobs while the other has only India-based jobs. As mentioned, this project aims to classify fraudulent and genuine job adverts from these two datasets. Another aim is to find out which factors contribute in determining the credibility of job i.e. whether it is genuine or fraudulent. A comparative pattern analysis will be done on both of these datasets to find out whether there is any similarity on the factors of two datasets that determine fraudulent-ness of the job adverts.

INTRODUCTION

Since March 2020, there was a series of lay-offs and job cuts after the nation-wide lockdown and many people have not been called back to their respective workplaces; most of the working class has experienced salary cut, students who were offered placements haven't received their offer letters from companies, the labour class has migrated back to their home-town with no signs of returning back. A global employment crisis is already looming over and people are aware about the efforts they need to put in to secure a job back. Students who have graduated and wish to apply for foreign universities are now searching for remote work from home to not waste the year and rather to add practical training experience in their CVs.

This has been a boon to fraudsters to take advantage of people's anxiety and worry regarding the grim future of job prospects to lure them into false high pay promising jobs. Often in a state of anxiety and believing it to be a fate or destiny calling, people end up subscribing at multiple platforms to get their CVs or resumes recognized.

Students, freshers, people who have lost jobs or are temporarily not given work end up signing and subscribing at various job listings. Many times these job listings just pop up as an advertisement and often people willingly share their email id and previous organization details.

One of the most common ways job seekers find out about job opening is by looking up online job portals such as Naukri.com, Internshala.com, Monster.com, Indeed.com, etc. Job seekers can search about jobs in which they are interested. The amount of number of job adverts that are added on these sites every day is quite enormous. There are often postings of job adverts with suspicious information that goes unnoticed by the site.

Fake job adverts remain live on recruitment sites until they are detected, which often takes weeks or months. In the meantime, individuals unwittingly waste time and effort filling out job applications - complete with personal details - for non-existent roles. What's more, if the contact details provided on the fake advert are those of a real business, it will have to deal with receiving applications for a vacancy that doesn't exist.

There are also financial consequences. Jobseekers have been known to travel considerable distances at significant cost to interview for these fake roles. They have been tricked into calling premium rate phone lines for interviews,

participating in money laundering via work-from-home scams, and paying extortionate fees for non-existent background checks, online training, visas or insurance.

This project proposes developing a system that is able to classify job adverts as fraudulent or genuine. Another aim of this project is to find out the factors that point towards a job advert that is fraudulent, which will be achieved by comparative pattern analysis.

This project aims to classify job advertisements as genuine or fake as well as to specify what factors seem to be prevalent for fake job adverts. Exploratory Data Analysis (EDA) will be performed on the dataset to find out whether factors such as telecommuting, has company logo, etc. has any effect on job being fraudulent.

IMPLEMENTATION DETAILS

Data Collection:

There will be two datasets, primary(Kaggle) and secondary(web scraped). The secondary dataset is scraped from job sites such as Naukri.com, Monster.com, Internshala.com, Shine.com using several web scraper tools. There are about 220 entries in this dataset. The primary dataset is from Kaggle. There are about 17,880 entries in this dataset. This is an unbalanced dataset i.e. about 866 fraudulent jobs and 17014 genuine jobs.

The data cleaning, model implementation, model evaluation, EDA will be done separately on both the datasets.

Finally, a comparative pattern analysis is done to verify whether the factors that point towards the credibility of a job advert are similar or different in both the datasets. This is a contribution to the research about whether scammers incorporate similar techniques in internationally available job openings and India based job openings.

System Requirement Specification:-

• Google Colab:-

To work on this project I am using Anaconda, as it is a distribution of the Python programming languages for scientific computing i.e. data science, machine learning applications, large-scale data processing, predictive analytics, etc., that aims to simplify package management and deployment.

• Python:-

Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python has been built with extraordinary Python libraries that are used in Big Data for solving problems that are as follows.

- NumPy
- Seaborn
- Pandas
- Matplotlib etc.

Algorithms:-

Classification is a technique where we categorize data into a given number of classes. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Here I am using the classification algorithms as my data set consists of multiple features having categorical data.

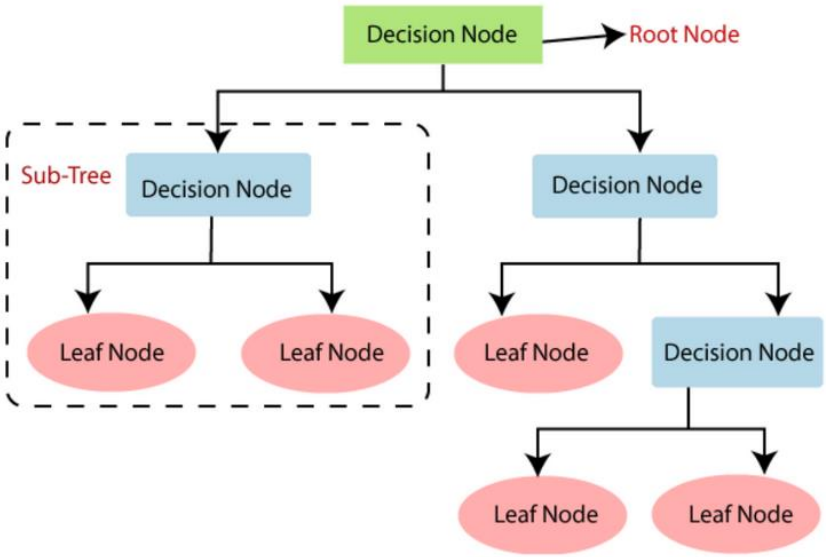
The classification algorithms that I have implemented are as follows.

- Logistic Regression:

Logistic regression is a process of modelling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on.

- Decision Tree:

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. A Decision Tree is traversed by asking Yes/No questions and moving down the tree either to the left subtree or right subtree.

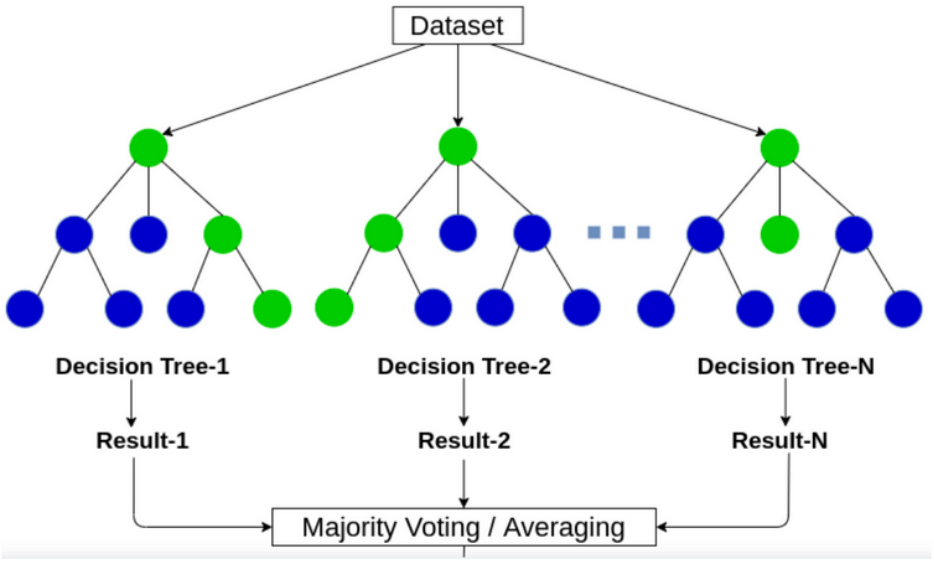


The main objective of the creation of a decision tree is to build a training model or training set. This training model is used to predict the value or class of the recipient variables or categorical variables.

- Random Forest:

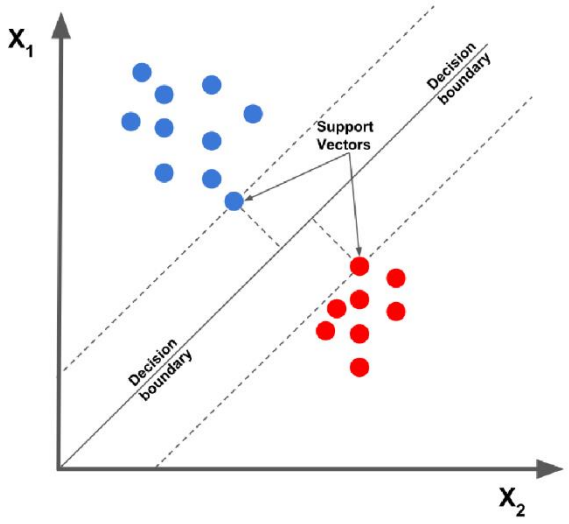
The random forest algorithm is also known as the random forest classifier. The RF algorithm comprises a random collection or a random selection of a forest tree. It creates a random sample of multiple decision trees and merges them together to obtain a more stable and accurate prediction through cross validation.

The algorithm will choose random subsets of the job advert dataset and create multiple decision trees with randomly selected features and samples. These multiple decision trees with their accuracy and their predictions i.e. job being fraud or genuine, will vote a decision tree output with highest fraud job detection accuracy and will be considered as the fraudulent or genuine.



- SVM:

Support Vector Machine(SVM) is a supervised machine learning algorithm used for both classification and regression. Though we say regression problems as well its best suited for classification. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points.



The main reason behind using Logistic Regression, Decision Tree, Random Forest and SVM is that these algorithms work better with categorical data. Although Regressor models and SVM can be tuned to work with numerical data as well.

I have compared the performances of these models. I have observed the Accuracy, Precision and Recall of all of these algorithms and found out which one works best with the data at hand.

Experimental Setup and Results

Dataset:

The dataset sourced from Kaggle.

```
[ ]  print(mydf.info())

<class 'pandas.core.frame.DataFrame'>
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)
memory usage: 2.5+ MB
```

The dataset consists of 17,880 rows and 18 columns.

Web scraped Dataset:

```
print(scjobdf.info())

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 220 entries, 0 to 219
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Unnamed: 0           220 non-null    int64
 1   title                220 non-null    object
 2   location             220 non-null    object
 3   salary_range         88 non-null     object
 4   company_profile      166 non-null    object
 5   description          214 non-null    object
 6   requirements         183 non-null    object
 7   benefits             84 non-null     object
 8   telecommuting        220 non-null    int64
 9   has_company_logo     220 non-null    int64
 10  has_questions        220 non-null    int64
 11  employment_type      201 non-null    object
 12  required_education   142 non-null    object
 13  required_experience  165 non-null    object
 14  industry             177 non-null    object
 15  function             182 non-null    object
 16  fraudulent           220 non-null    int64
dtypes: int64(5), object(12)
memory usage: 29.3+ KB
```

This dataset contains 220 rows and 17 columns.

Feature Enginering and Data Cleaning:

The columns I was concerned with are, Department, Company profile, Description, Requirements, Benefits, Telecommuting, has_company_logo, has_questions, employment_type, required_education, required_experience, industry, function and fraudulent.

The remaining columns were dropped.

The null values from the columns were replaced with 'Empty' indicating that the job advert posters hadn't filled the respective field data.

There are some hypothesis I have put forward:

- Fraudulent job adverts have more instances of company logo being absent as well as not having any questions mentioned in the advert when compared to Genuine job adverts.
- Fraudulent job adverts have short word count length for textual features when compared to Genuine job adverts.
- Fraudulent job adverts have more instances of empty textual features than Genuine job adverts.

Finding out which job adverts have empty textual features data:

The job adverts which do not have any content for aforementioned textual features. I have filled these NA values with 'Empty'. As the job adverts which do not have data for these textual features, have impact on credibility of the job advert, I have created new columns, respectively, 'empty_desc, empty_req, empty_benefits, empty_compprof' having Boolean values 0 if there is some content present and 1 if there isn't any content present.

Finding the word count for textual features:

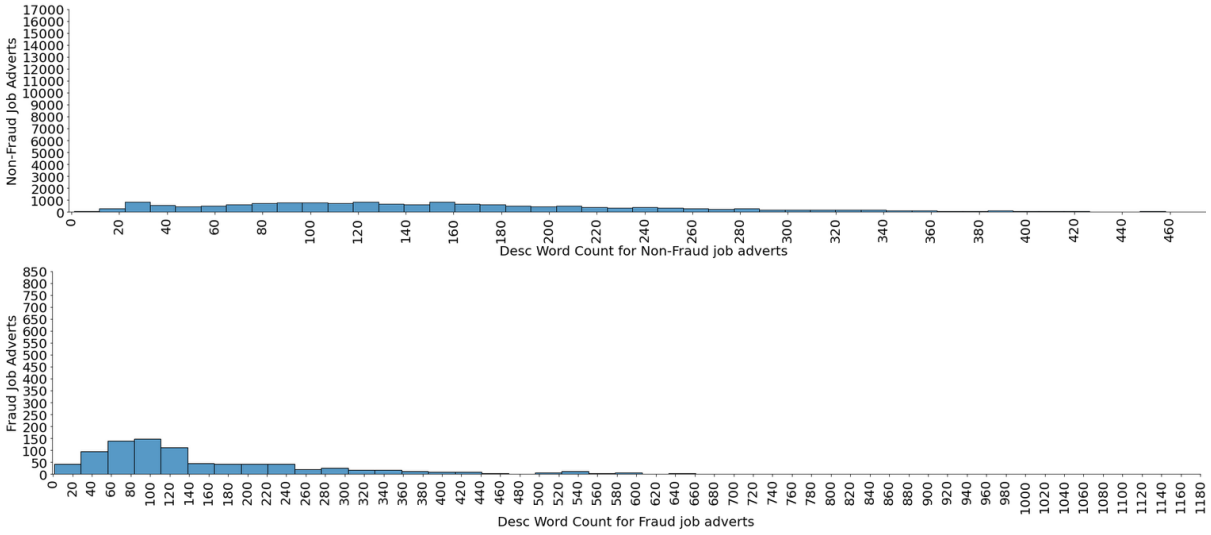
Finding word length count of company_profile, job description, requirements, benefits.

```
#print(mydf['required_education'].unique())
function = lambda x:len(x.split(' '))

len_compprof = jobdf['company_profile'].astype(str).apply(function)
len_desc = jobdf['description'].astype(str).apply(function)
len_req = jobdf['requirements'].astype(str).apply(function)
len_benefits = jobdf['benefits'].astype(str).apply(function)

#Creating Columns
jobdf['compprof_len'] = len_compprof
jobdf['desc_len'] = len_desc
jobdf['req_len'] = len_req
jobdf['benefits_len'] = len_benefits
```

Finding out the word count for textual features, description, company_profile, requirements and benefits. This feature creation will be used for deciding threshold for short word count.

Setting a threshold for short word count of textual features:

For Description column:



Here we can see 393 out of 866 fraudulent job adverts have less than or equal to 100 word count for description column i.e. 49% fraudulent job adverts. While

on the other hand, 5284 out of 17014 genuine job adverts have less than 100 word count for description i.e. 29% genuine job adverts.

I applied this method for remaining textual columns and found out that company profile word count threshold as 10. 587 out of 866 fraudulent job adverts i.e. 68% have word count less than 10. While for genuine job adverts 3544 out of 17014 i.e. 20% have word count less than 10 word count length.

Threshold wasn't found for textual feature requirements and benefits as the percent ratio of fraud and genuine job adverts is almost the same.

Replaced NaN values of columns having Boolean values to 'Empty'

Created a new feature 'has_special_attr', which has Boolean values 1 if the columns, 'required_education, required_experience, function, department' has content present and 0 if all of the mentioned columns are empty. This indicates that the job advert poster left these fields empty when creating the advert.

Creating new feature has_special_attributes

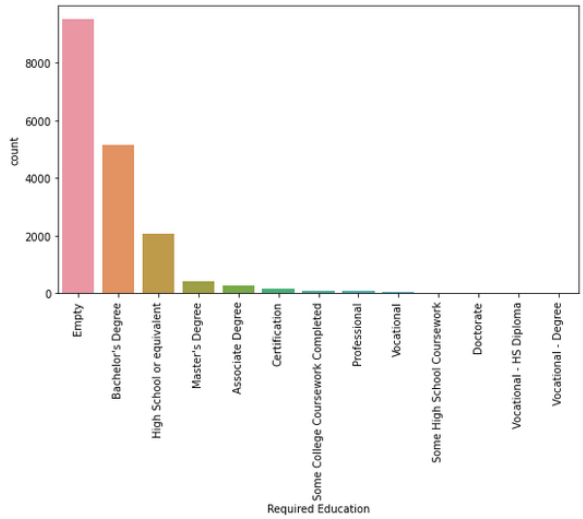
```
[ ]  special_attr_list = []

     for i in range(len(jobdf['required_education'])):
       a = jobdf['required_education'][i]
       b = jobdf['required_experience'][i]
       c = jobdf['function'][i]
       d = jobdf['department'][i]

       if a == 'Empty' and b == 'Empty' and c == 'Empty' and d == 'Empty':
         special_attr_list.append(0)
       else:
         special_attr_list.append(1)
```

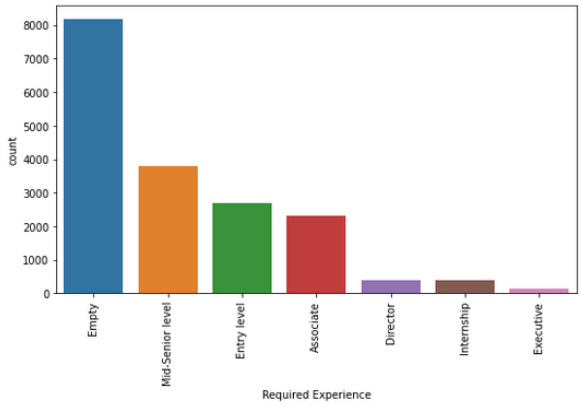
```
[ ]  jobdf['has_spec_attr'] = special_attr_list
```

Data Visualization

For Primary dataset (sourced from Kaggle):

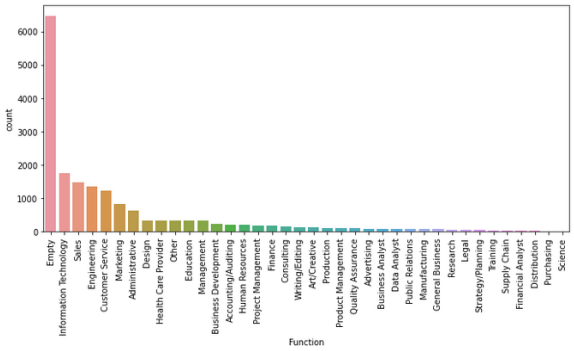
Most desired education and experience for the whole dataset:

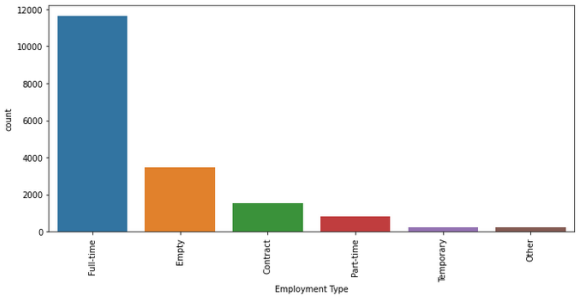




Bachelor's degree seems to be most desired required education followed by High school or equivalent. While the most desired required experience seems to be Mid Senior level followed by Entry level and associate. It also seems that employers often leave these fields empty while creating job adverts.

Most occurring Function and Employment type in overall dataset.

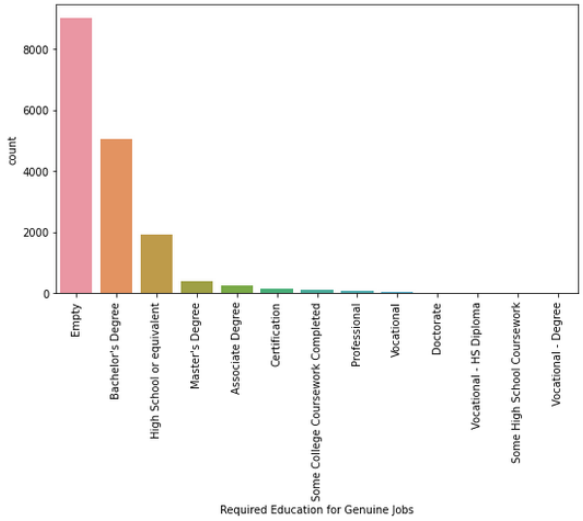


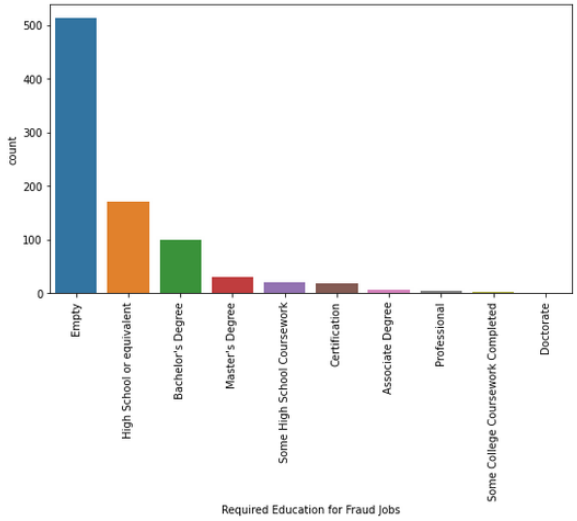


Information Technology is the most popular function that is observed in the overall job adverts followed by Sales, Engineering, etc.

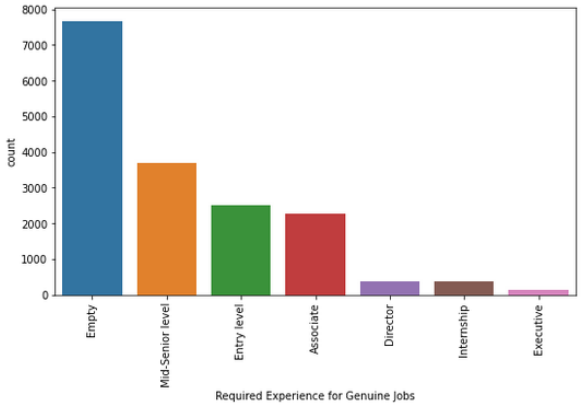
Most employers prefer that their potential employees should do full time jobs.

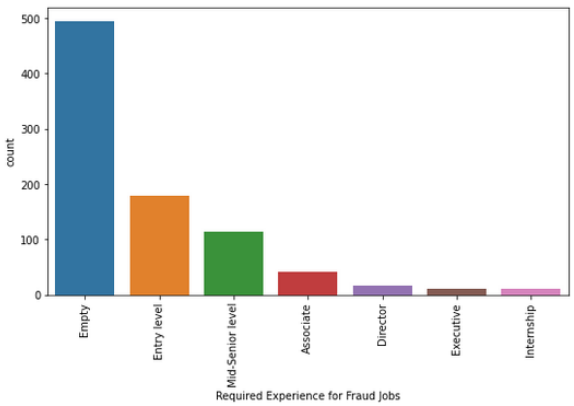
Above visualizations comparing Fraud and Genuine job adverts:



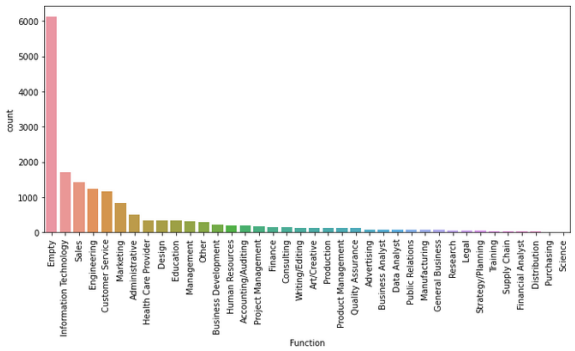


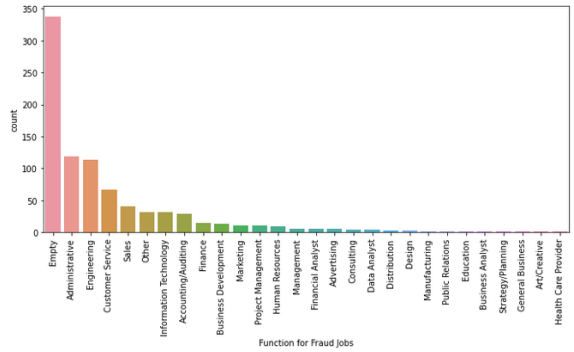
Bachelor's degree is highly desired education for genuine job adverts while High school or equivalent is the highly desired education for fraudulent job adverts.



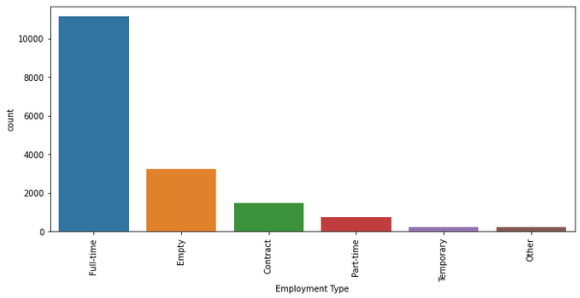


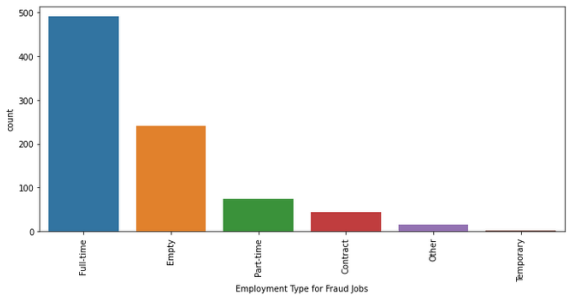
Mid-Senior level is the highly desired experience for genuine jobs while on the other hand, Entry level is the highly desired experience for fraudulent job adverts.





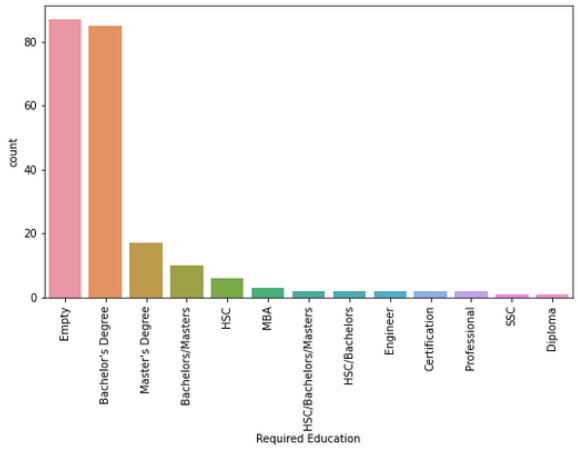
Genuine job adverts have IT dominating, followed by Sales and Engineering while on the other hand, Fraudulent job adverts prefer Administrative, Engineering and Customer service more.

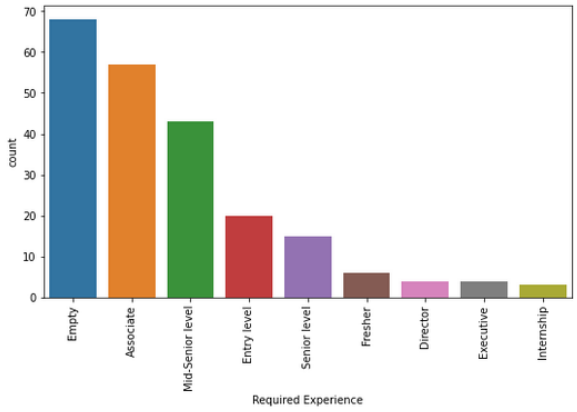




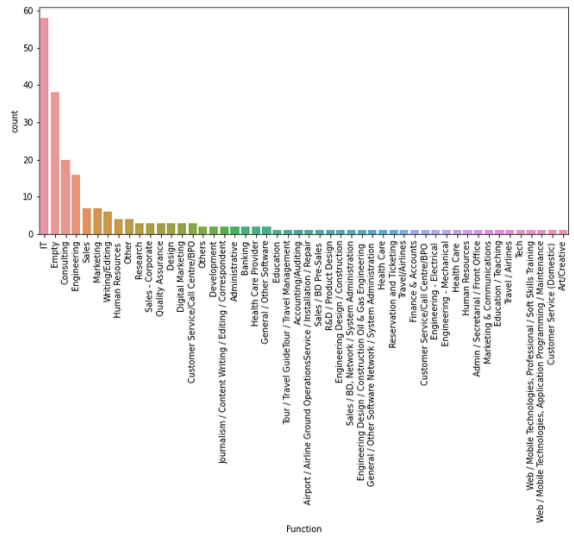
Genuine Job adverts have Full time followed by contract and part time, while fraudulent job adverts prefer Full time followed by part time and contract. There isn't much difference between preferred employment type in both Genuine as well as fraudulent job adverts.

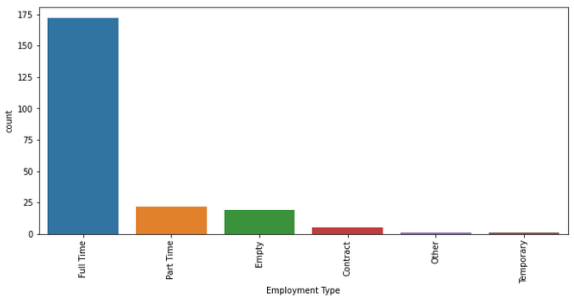
For secondary dataset (web scraped)





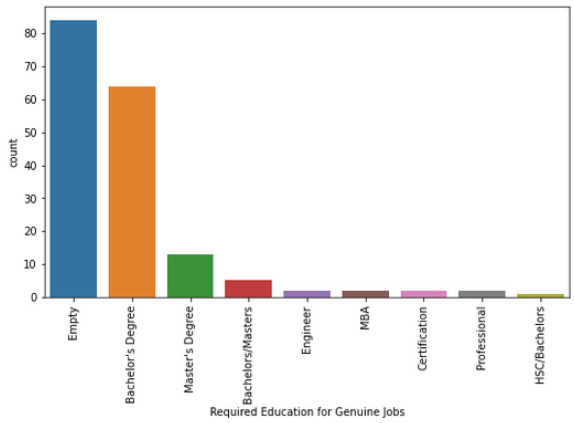
For the overall dataset, Bachelor's degree seems to be in most demand followed by Master's degree and Associate is the most desired experience followed by mid-senior level. Most job adverts have left these fields empty.

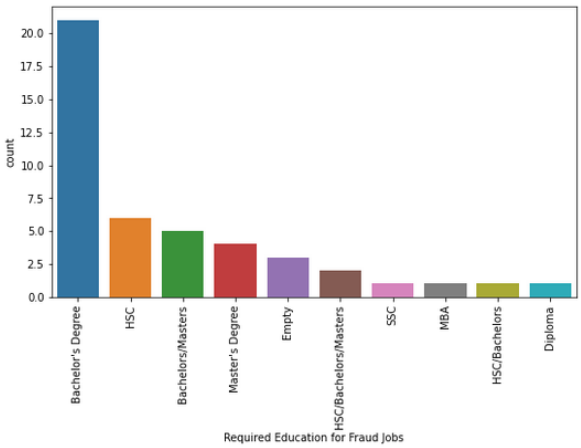




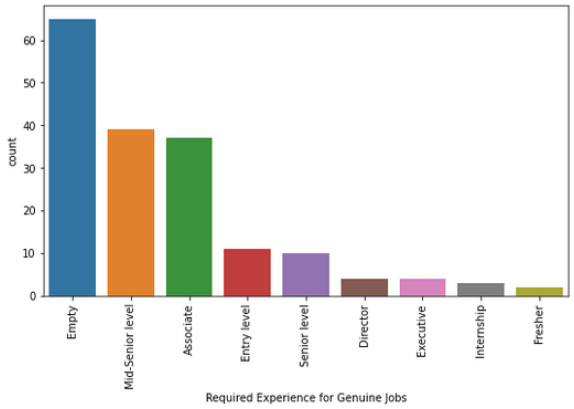
Information technology is the most desired function observed in overall job adverts, followed by consulting and engineering. Full time is still more preferable to employers than part time and contract.

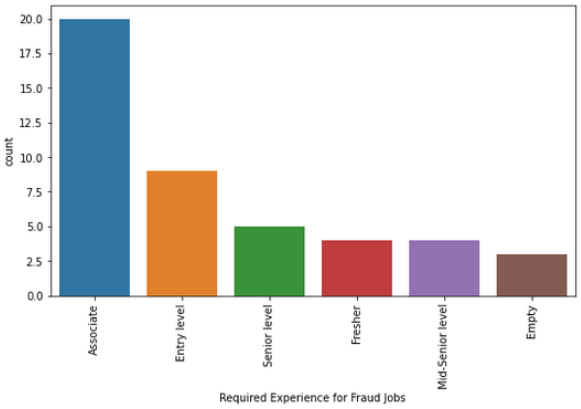
Above visualizations comparing Fraud and Genuine job adverts:



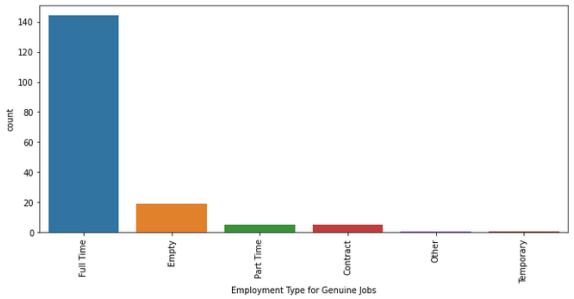


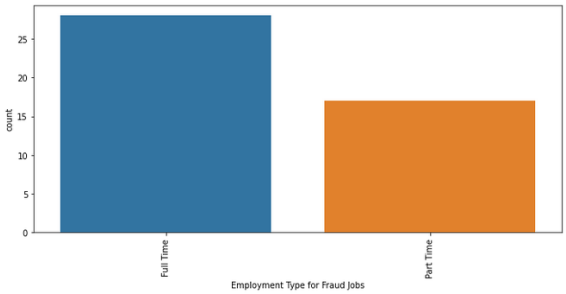
Bachelor's degree is the most desirable required education in both genuine and fraudulent job adverts. It is seen that HSC, SSC and HSC/ Bachelors/ Masters is only seen in fraudulent job adverts.





Mid senior level followed by associate is the most popular required experience observed in genuine job adverts while on the other hand, Associate is most popular followed by entry level for fraudulent job adverts.





For fraudulent jobs, there are only two type of employments that can be observed, and part time is still contributing significantly compared to the job adverts who have part time employment in genuine job adverts.

One Hot Encoding:

The columns ‘employment_type, required_education, required_experience, function’ consist of categorical entries, I have used one hot encoding for turning these entries into values of either 1 or 0.

```
[ ] detectdf = pd.get_dummies(detectdf, columns = ['employment_type','required_education','required_experience','industry','function'])
    detectdf.head()
```

	telecommuting	has_company_logo	has_questions	empty_compprof	empty_desc	empty_req	empty_benefits	has_short_desc	has_short_compprof	has_short_req	has_spec_attr	fraudulent	employment_type_Contract
0	0	1	1	0	0	0	1	1	0	1	1	0	0
1	0	0	1	0	0	0	1	0	1	1	1	0	0
2	0	1	1	0	0	0	1	1	0	0	1	0	0
3	1	0	0	0	0	0	1	1	1	0	1	1	0
4	1	0	0	1	0	1	1	1	1	1	1	1	0

The Python library Pandas provides a function called get_dummies to enable one-hot encoding.

pd.get_dummies when applied to a column of categories where we have one category per observation will produce a new column (variable) for each unique categorical value. It will place a one in the column corresponding to the categorical value present for that observation.

Algorithms:

For the primary dataset, as the dataset is highly unbalanced, 17014 genuine jobs and 866 fraudulent jobs. I have balanced the dataset using RandomUnderSampler from imblearn.under_sampling.

Random Undersampling

```
from imblearn.under_sampling import RandomUnderSampler
```

```
[ ]  rus  = RandomUnderSampler(random_state=42)
     jdf,y = rus.fit_resample(jdf, y)

     jdf = pd.DataFrame(jdf)
     y = pd.DataFrame(y)
```

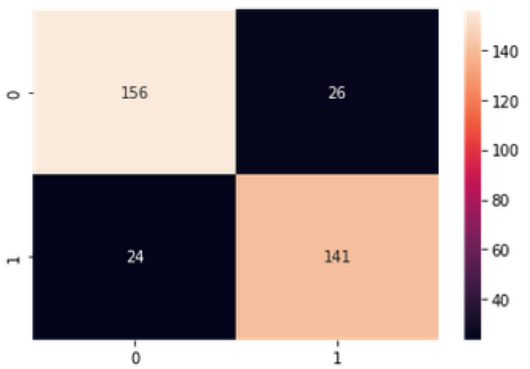
```
[ ]  print(jdf.shape)
     print(y.shape)
```

```
     (1732, 1546)
     (1732, 1)
```

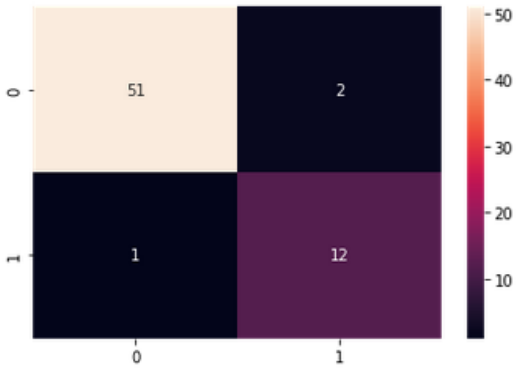
Now the balanced dataset has 1732 rows i.e. 866 genuine jobs and 866 fraudulent jobs.

For the secondary dataset, where there are 175 genuine jobs and 45 fraudulent jobs present. There is no need to make the dataset balanced because there isn't much difference in two classes.

Logistic Regression:

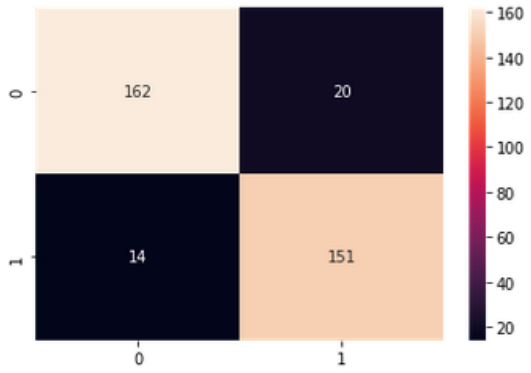


Primary Dataset

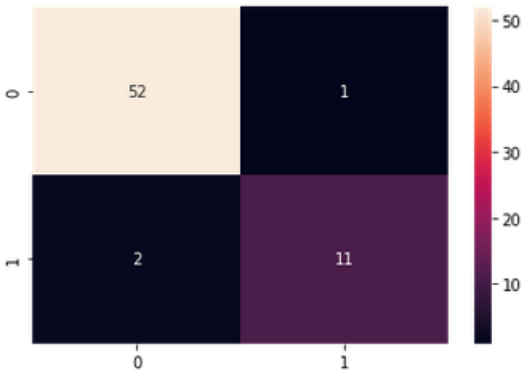


Secondary Dataset

Decision Tree:

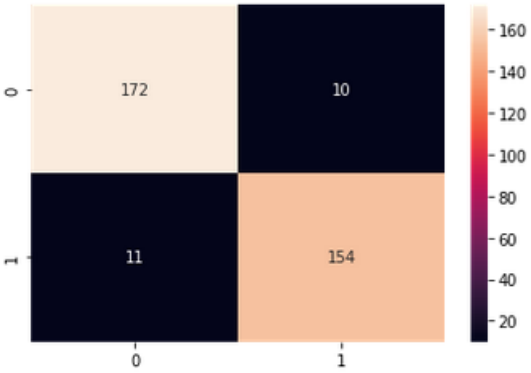


Primary Dataset

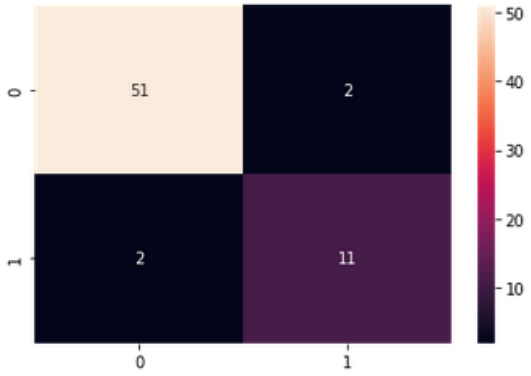


Secondary Dataset

Random Forest:



Primary Dataset

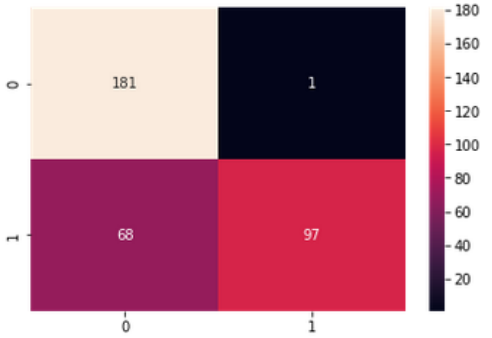


Secondary Dataset

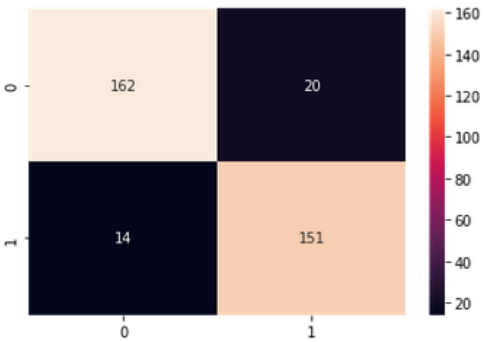
SVM (Simple Vector Machines):

For primary dataset, as the confusion matrix results were not accurate for fraudulent class. I had to implement Hyper Parameter Tuning, after which the results improved significantly.

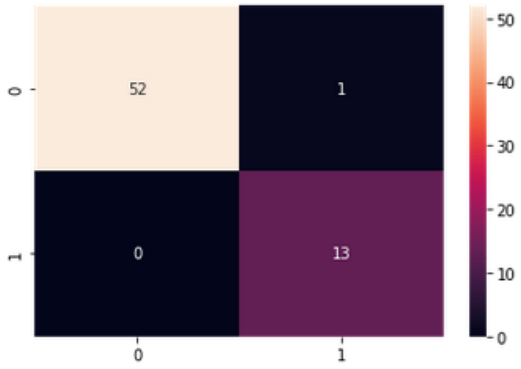
Before Hyper parameter tuning



After Hyper parameter tuning



For Primary Dataset



For Secondary Dataset

Analysis of the results

As we are more concerned with 1 class i.e. yes for fraudulent class, positive class will be 1 and negative class will be 0.

1. **Logistic Regression**:

   (For Primary Dataset)

   We have 87% precision with 86% recall which means out of all predicted values of 0 as True, 87% were predicted accurately and out of all actual True values of 0, 86% were predicted accurately.

   We have 84% precision with 85% recall which means out of all predicted values of 1 as True, 84% were predicted accurately and out of all actual True values of 1, 85% were predicted accurately.

   Confusion Matrix results:

   - True Positive: 141 of fraudulent jobs were classified correctly as fraudulent jobs.
   - True Negative: 156 of genuine jobs were classified correctly as genuine jobs.
   - False Positive: 26 job adverts were classified as fraudulent when in reality they were genuine job adverts.
   - False Negative: 24 job adverts were classified as genuine when in reality they were fraudulent job adverts.

   (For Secondary Dataset)

   We have 98% precision with 96% recall which means out of all predicted values of 0 as True, 98% were predicted accurately and out of all True values of 0, 96% were predicted accurately.

We have 86% precision with 92% recall which means out of all predicted values of 1 as True, 86% were predicted accurately and out of all actual True values of 1, 92% were predicted accurately.

Confusion Matrix results:

- True Positive: 12 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 51 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 2 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 1 job adverts were classified as genuine when in reality they were fraudulent job adverts.

2. **Decision Tree:**

(For Primary Dataset)

We have 92% precision with 89% recall which means out of all predicted values of 0 as True, 92% were predicted accurately and out of all True values of 0, 89% were predicted accurately.

We have 88% precision with 91% recall which means out of all predicted values of 1 as True, 88% were predicted accurately and out of all True values of 1, 91% were predicted accurately.

Confusion Matrix results:

- True Positive: 150 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 162 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 20 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 15 job adverts were classified as genuine when in reality they were fraudulent job adverts.

(For Secondary Dataset)

We have 96% precision with 98% recall which means out of all predicted values of 0 as True, 96% were predicted accurately and out of all True values of 0, 98% were predicted accurately.

We have 92% precision with 85% recall which means out of all predicted values of 1 as True, 92% were predicted accurately and out of all True values of 1, 85% were predicted accurately.

Confusion Matrix results:

- True Positive: 11 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 52 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 1 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 2 job adverts were classified as genuine when in reality they were fraudulent job adverts.

3. **Random Forest:**

(For Primary Dataset)

We have 94% precision with 94% recall which means out of all predicted values of 0 as True, 94% were predicted accurately and out of all True values of 0, 94% were predicted accurately.

We have 93% precision with 93% recall which means out of all predicted values of 1 as True, 93% were predicted accurately and out of all True values of 1, 93% were predicted accurately.

Confusion Matrix results:

- True Positive: 154 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 171 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 11 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 11 job adverts were classified as genuine when in reality they were fraudulent job adverts.

(For Secondary Dataset)

We have 96% precision with 96% recall which means out of all predicted values of 0 as True, 96% were predicted accurately and out of all True values of 0, 96% were predicted accurately.

We have 85% precision with 85% recall which means out of all predicted values of 1 as True, 85% were predicted accurately and out of all True values of 1, 85% were predicted accurately.

Confusion Matrix results:

- True Positive: 11 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 51 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 2 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 2 job adverts were classified as genuine when in reality they were fraudulent job adverts.

4. **SVM:**

(For Primary Dataset)

We have 92% precision with 89% recall which means out of all predicted values of 0 as True, 92% were predicted accurately and out of all True values of 0, 89% were predicted accurately.

We have 88% precision with 91% recall which means out of all predicted values of 1 as True, 88% were predicted accurately and out of all True values of 1, 91% were predicted accurately.

Confusion Matrix results:

- True Positive: 150 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 162 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 20 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 15 job adverts were classified as genuine when in reality they were fraudulent job adverts.

(For Secondary Dataset)

We have 100% precision with 98% recall which means out of all predicted values of 0 as True, 100% were predicted accurately and out of all True values of 0, 98% were predicted accurately.

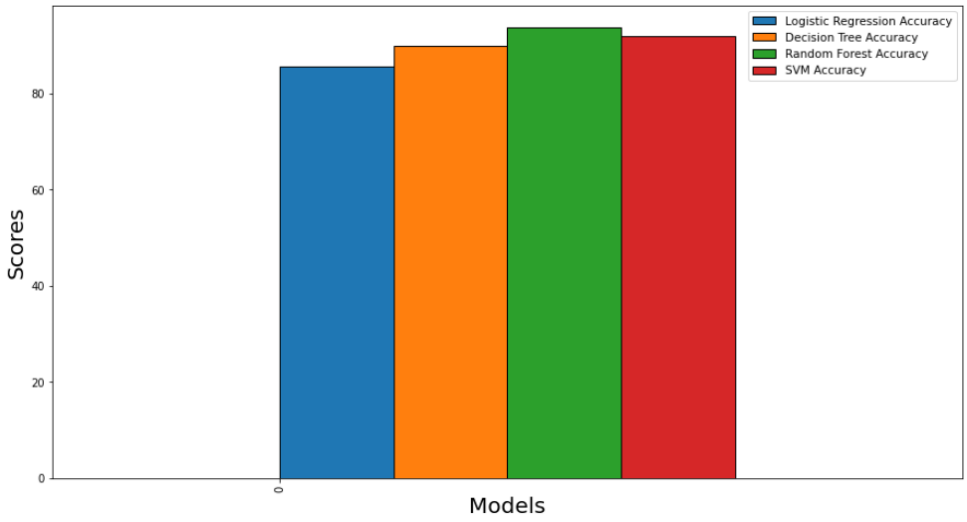
We have 93% precision with 100% recall which means out of all predicted values of 1 as True, 93% were predicted accurately and out of all True values of 1, 100% were predicted accurately.

Confusion Matrix results:

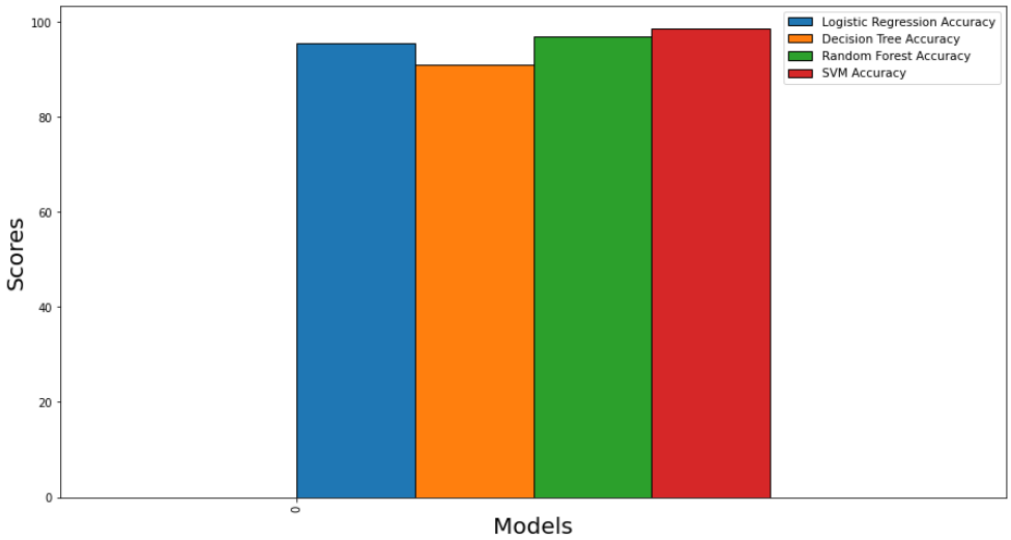
- True Positive: 13 of fraudulent jobs were classified correctly as fraudulent jobs.
- True Negative: 52 of genuine jobs were classified correctly as genuine jobs.
- False Positive: 1 job adverts were classified as fraudulent when in reality they were genuine job adverts.
- False Negative: 0 job adverts were classified as genuine when in reality they were fraudulent job adverts.

Performance of models:

Accuracy:

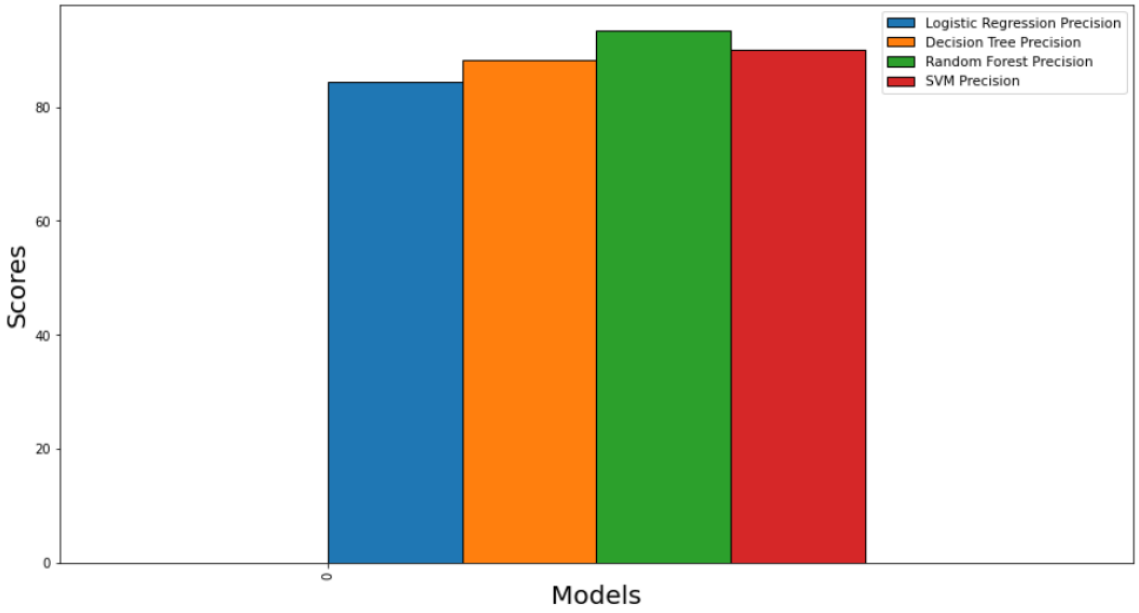


For Primary Dataset

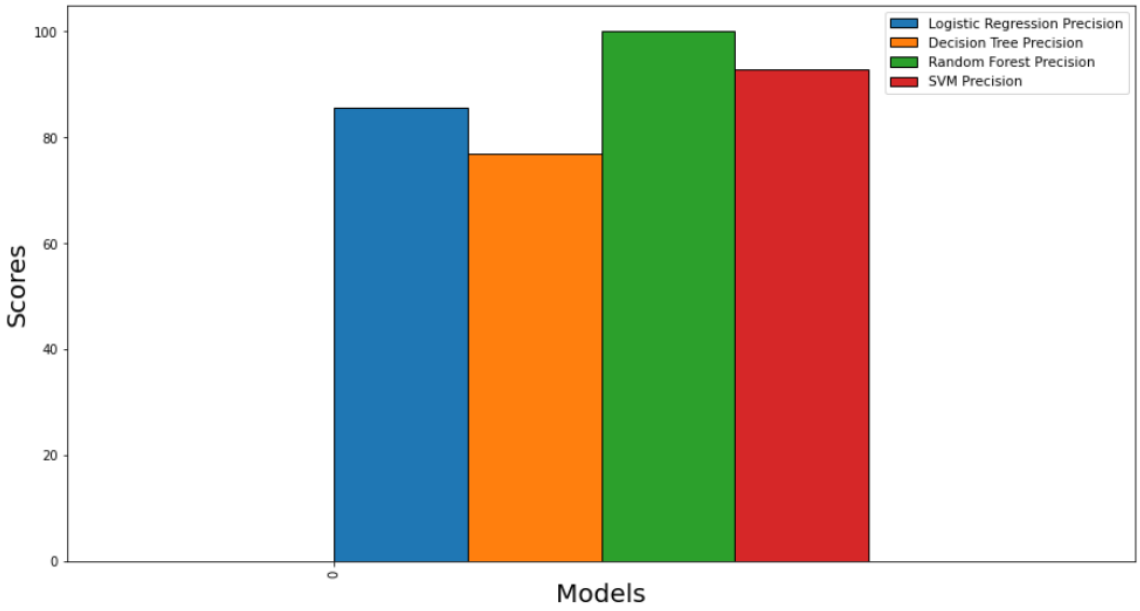


For Secondary Dataset

Precision:

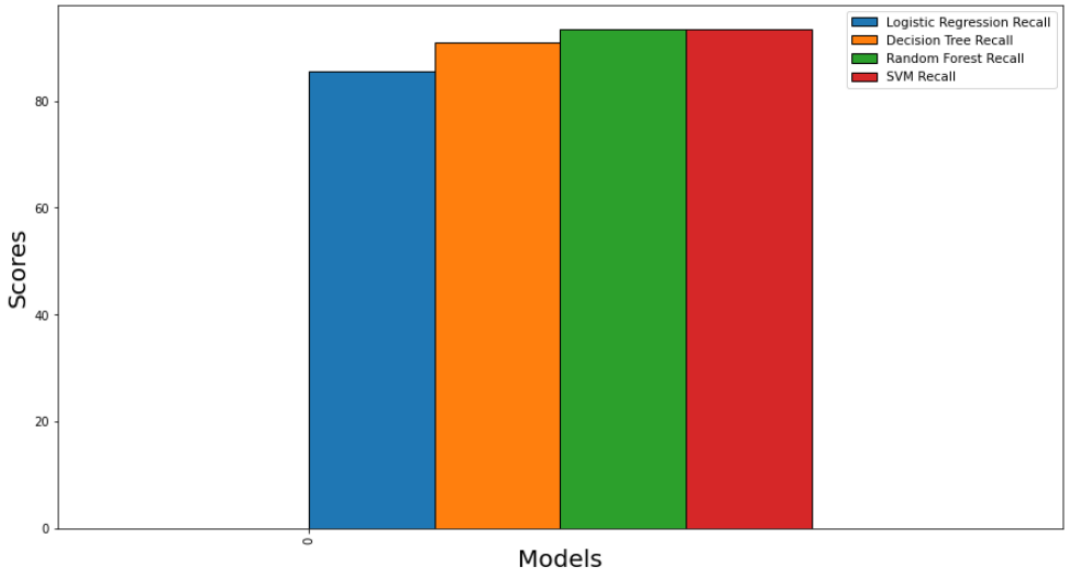


For Primary Dataset

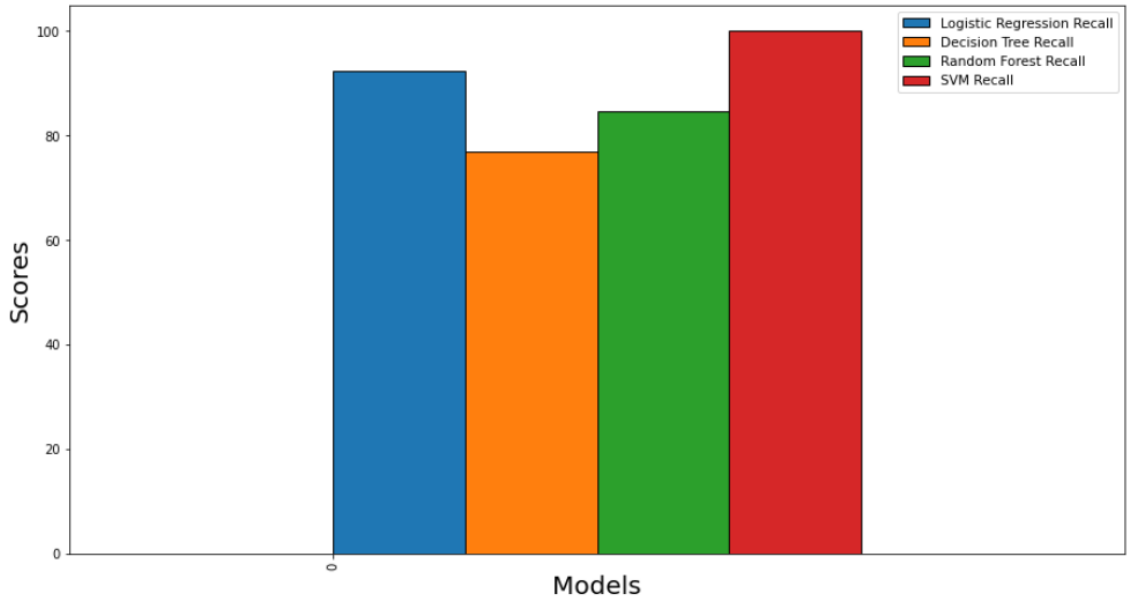


For Secondary Dataset

Recall:



For Primary Dataset



For Secondary Dataset

Random Forest had the highest accuracy, precision and recall among the algorithms used for the primary dataset.

Simple Vector Machine (SVM) was the best model for secondary dataset as it had highest accuracy and recall. Random forest followed with highest precision and decent accuracy but had the second lowest recall just above Decision Tree.

CONCLUSION

Results obtained through visualizations done on the dataset:

As we are more concerned with detection of fraudulent job adverts, the following observations are a comparison of results obtained through visualization of primary dataset against secondary dataset.

1. For the primary dataset, the most desired required education was high school or equivalent.
   For the secondary dataset, the most desired required education was Bachelor's degree followed by HSC. Although the number of job adverts with Bachelor's degree was high, contribution of job adverts with HSC as a required education was significant nonetheless.
   This indicates that these fraudulent job advert posters want less educated people to apply making them an easy target for further proceedings.

2. For the primary dataset, highly desired required experience was Entry level while for the secondary dataset, it was Associate followed by Entry level.
   Job seekers who are new to the job industry would likely to apply to these job adverts, although scammers based in India, preferred people with some industry experience.

3. Most desired function was Customer service followed by Engineering and Administrative for primary dataset, while IT, HR, Digital marketing and Customer Service/ Call Centre were the most desired functions for secondary dataset.

4. For primary dataset, fraudulent job advert posters preferred full time employment type while for the secondary dataset, Full time as well as part time was preferred.
   Another detail that was observed, there is a variety of employment types other than Full time and Part time preferred by job adverts in primary dataset, for example, contract, other, temporary. This diversity is missing from secondary dataset as most of the job adverts contain either Full time or Part time.

5. Fraudulent Job adverts often seem to not have company logo present which is a factor that contributes to deciding credibility of a job advert. Presence of questions in the job adverts is also another factor that contributes to deciding credibility of a job advert.

Use cases:

- This model can be used to classify fraudulent job adverts from genuine ones.
- This model can be implemented by newly started job portals to filter out fraudulent job adverts and show genuine job adverts to job seekers more.
- The comparative pattern analysis done on both the datasets which reflected how fraudulent Indian job adverts are structured which then can be used to tune the model for classification of Indian job adverts specifically.

FUTURE ENHANCEMENT

- Textual features can be further analysed using NLP and more granular feature engineering can be done to gain more insights.

PROGRAM CODE

```
import sys
import pandas as pd
import nltk
import numpy as np
import re
import seaborn as sns
import matplotlib.pyplot as plt

pd.set_option('display.max_columns',None)
pd.set_option('display.max_rows',None)
pd.set_option('display.width',None)

from google.colab import drive
drive.mount('/content/drive')

cd drive/MyDrive/MSC_Project



mydf = pd.read_csv('fake_job.csv')

print(mydf.head())

mydf.shape

print(mydf.info())
print(mydf.describe())

mydf['fraudulent'].value_counts()

"""Creating a copy of mydf and storing it into jobdf"""

jobdf = mydf.copy()

print(jobdf.head(2))

"""Total Null values in the dataset"""
```

```
nullval = jobdf.isnull().sum()
plt.figure(figsize=(10,5))
plt.bar(jobdf.columns,nullval,color = 'orange', edgecolor = 'white')
plt.xlabel('Null values')
plt.xticks(rotation='vertical')
plt.show()

"""#Feature Engineering

Finding word length count of company_profile, job description, requirements, benefits.
"""

#print(mydf['required_education'].unique())
function = lambda x:len(x.split(' '))

len_compprof = jobdf['company_profile'].astype(str).apply(function)
len_desc = jobdf['description'].astype(str).apply(function)
len_req = jobdf['requirements'].astype(str).apply(function)
len_benefits = jobdf['benefits'].astype(str).apply(function)


#Creating Columns
jobdf['compprof_len'] = len_compprof
jobdf['desc_len'] = len_desc
jobdf['req_len'] = len_req
jobdf['benefits_len'] = len_benefits

"""Display description of the job advert with maximum description word count"""

print("job advert with maximum word count", jobdf['desc_len'].max())
#jobdf['desc_len'].loc[:,201]

"""Replacing benefits NaN values by 'empty'"""

jobdf['company_profile'].fillna('Empty',inplace=True)
jobdf['requirements'].fillna('Empty',inplace=True)
jobdf['description'].fillna('Empty',inplace=True)
jobdf['benefits'].fillna('Empty',inplace=True)
```