Real or Fake job prediction

CONTENTS

PROBLEM IDENTIFICATION

Data Wrangling

Exploratory Data Analysis

Feature Engineering

Modeling selection

Problem Identification

Problem Statement Worksheet

Reducing the fake job advertisements to 40-50% that were posted in various locations by end identifying their job description and salary range by the end of 15th July. So this could help the upcoming graduates and for the people hunting for jobs.

1 Context

Job postings were being posted in various locations with specifying benefits, salary_range, description etc in various platforms. Moreover, 800 fake job posting were identified in 18,000 posting. So basing on these posting fake job postings has to be reduced to 400 or more.

2 Criteria for success

Fake postings and companies has to be identified before the end of 30th June because a lot of freshers will be in search for jobs.

3 Scope of solution space

Most of the people will be blindly applying for a company that offers high packages and facilities without knowing whether it is fraudulent or not. So, we must also identify these job postings in these perspective also.

4 Constraints within solution space

Certifying a genuine job posting to fake will cause damage to company and will lose the employment opportunity for the people.

5 Stakeholders to provide key insight

Lead Data Scientist

6 Key data sources

Excel sheet- Data is collected from various sources and stored it in excel format.

Data Wrangling

```
In [6]: df.shape
Out[6]: (17880, 18)

In [7]: df.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
```

Using these we can understand how many values does dataset contains.

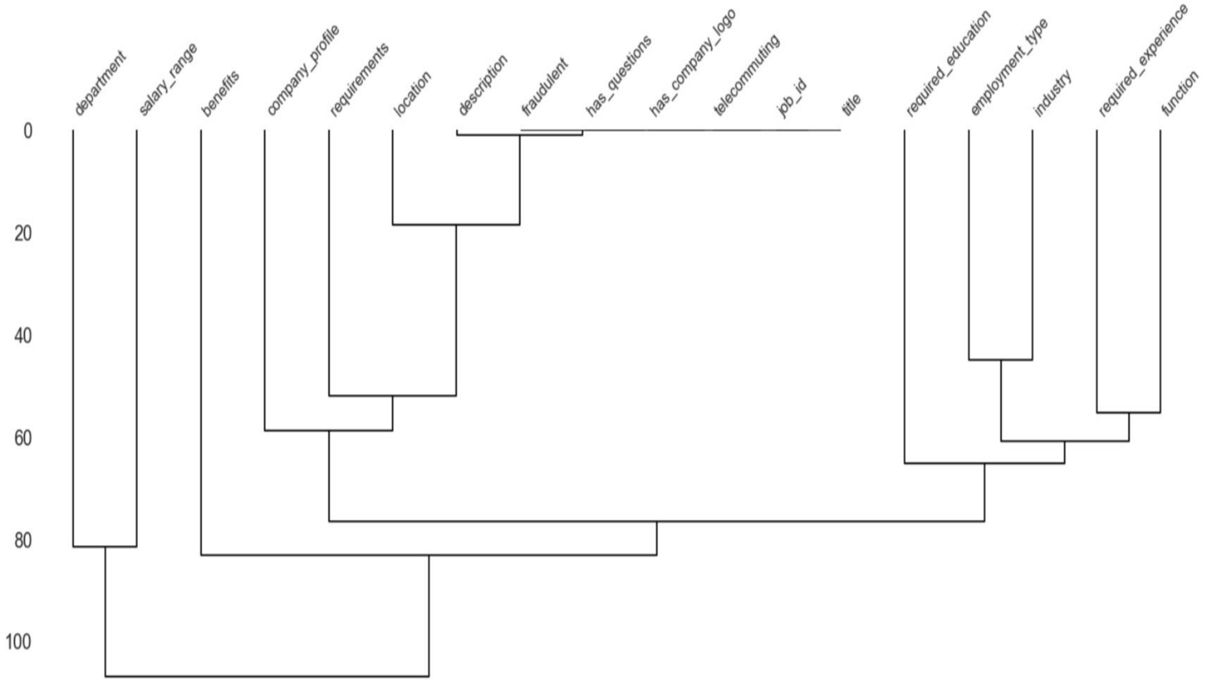
```
In [8]: df.describe()
```

Out[8]:

	job_id	telecommuting	has_company_logo	has_questions	fraudulent
count	17880.000000	17880.000000	17880.000000	17880.000000	17880.000000
mean	8940.500000	0.042897	0.795302	0.491723	0.048434
std	5161.655742	0.202631	0.403492	0.499945	0.214688
min	1.000000	0.000000	0.000000	0.000000	0.000000
25%	4470.750000	0.000000	1.000000	0.000000	0.000000
50%	8940.500000	0.000000	1.000000	0.000000	0.000000
75%	13410.250000	0.000000	1.000000	1.000000	0.000000
max	17880.000000	1.000000	1.000000	1.000000	1.000000

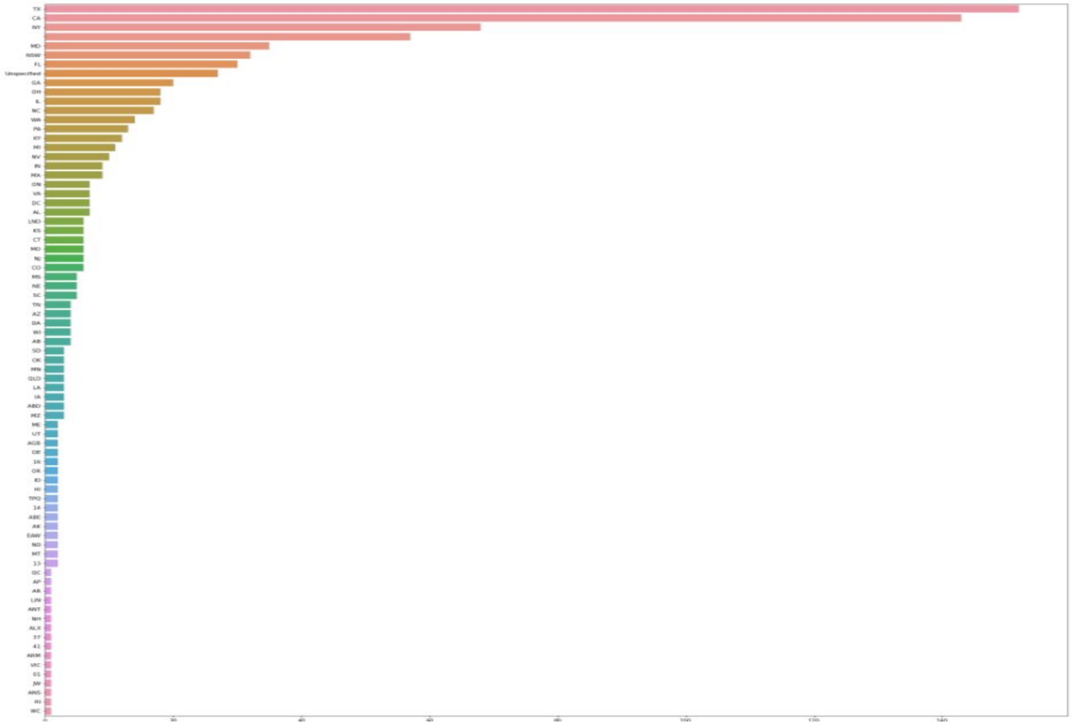
```
In [9]: df.columns

Out[9]: Index(['job_id', 'title', 'location', 'department', 'salary_range',
               'company_profile', 'description', 'requirements', 'benefits',
               'telecommuting', 'has_company_logo', 'has_questions', 'employment_type',
               'required_experience', 'required_education', 'industry', 'function',
               'fraudulent'],
              dtype='object')
```

4]: <matplotlib.axes._subplots.AxesSubplot at 0x1c98e1a2608>

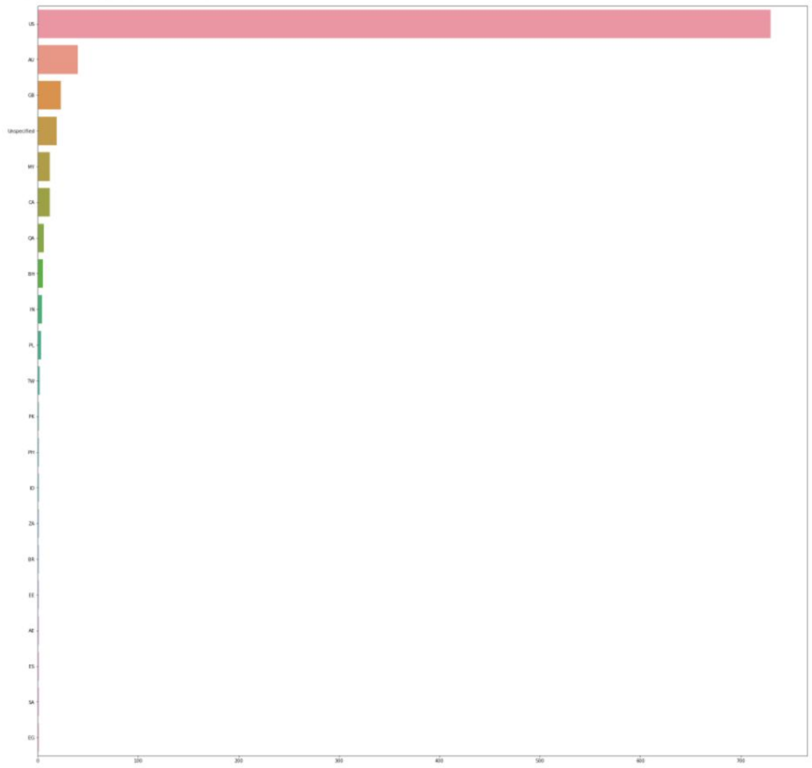


Dendogram helps us to understand the hierarchical relationships between them.

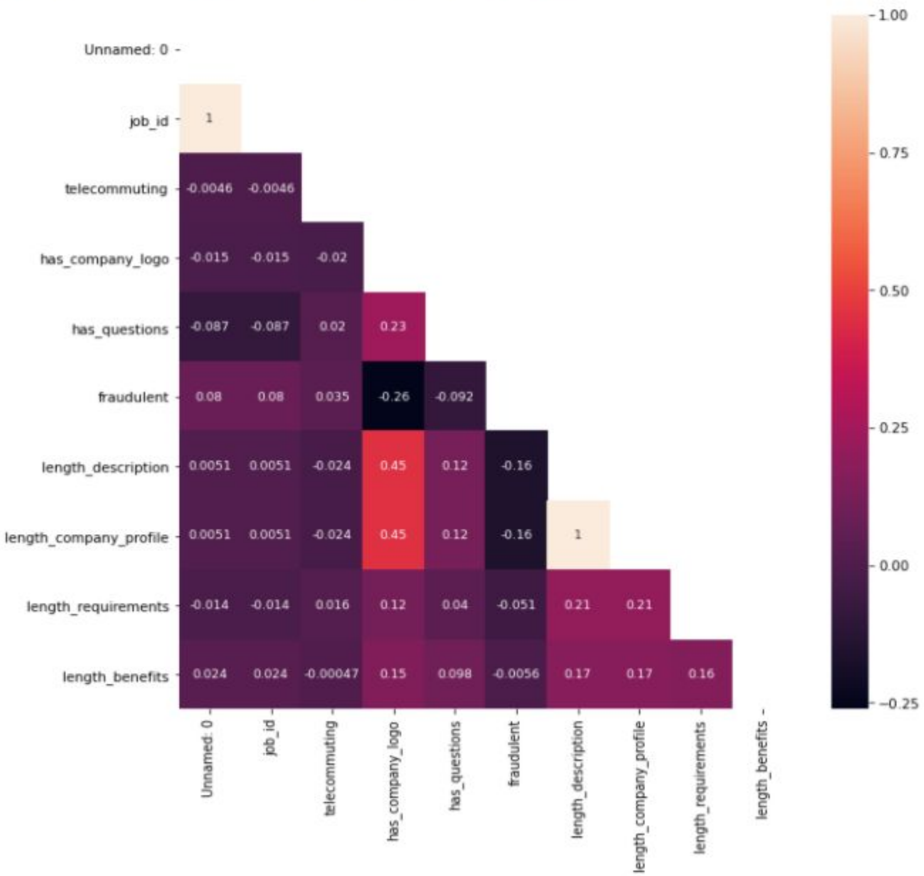
Exploratory Data Analysis



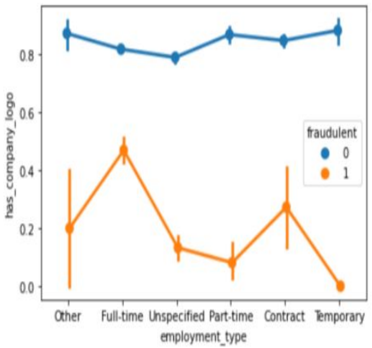
The Fake profile jobs created based on Countries.



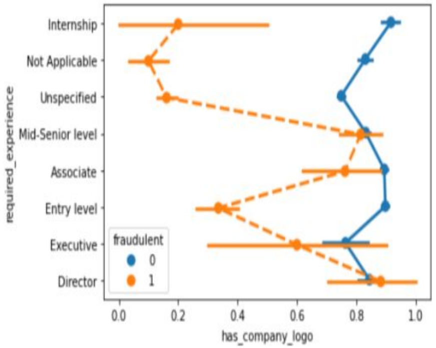
The fake job profile created based on the state.


Unnamed: 0
job_id
telecommuting
has_company_logo
has_questions
fraudulent
length_description
length_company_profile
length_requirements
length_benefits
1
-0.0046 -0.0046
-0.015 -0.015 -0.02
-0.087 -0.087 0.02 0.23
0.08 0.08 0.035 -0.26 -0.092
0.0051 0.0051 -0.024 0.45 0.12 -0.16
0.0051 0.0051 -0.024 0.45 0.12 -0.16 1
-0.014 -0.014 0.016 0.12 0.04 -0.051 0.21 0.21
0.024 0.024 -0.00047 0.15 0.098 -0.0056 0.17 0.17 0.16
1.00
0.75
0.50
0.25
0.00
−0.25

```
Out[145]: <matplotlib.axes._subplots.AxesSubplot at 0x2b9cae009c8>
```

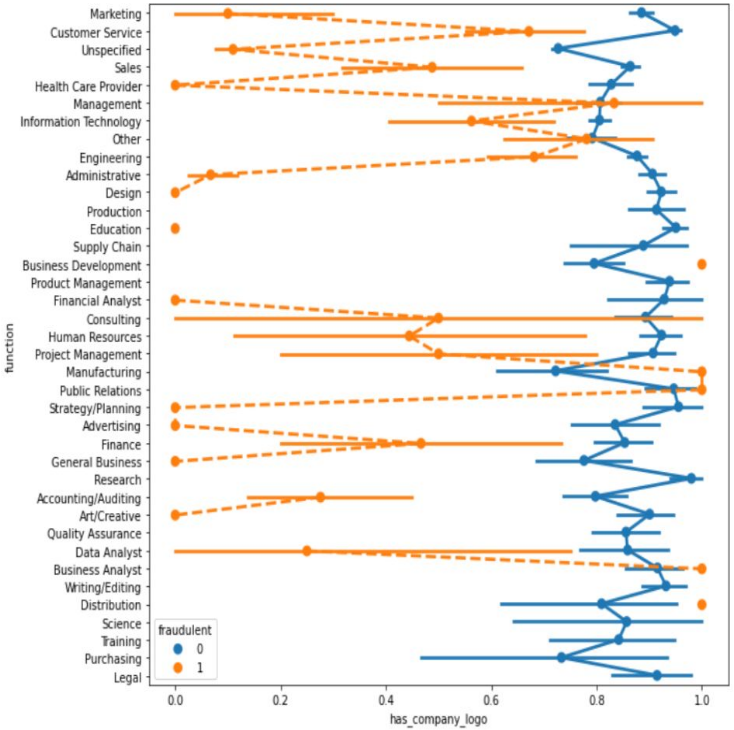



```
In [153]: sns.pointplot(x='has_company_logo',y='required_experience',hue='fraudulent',data=df,dodge=True,linestyles=['-','--'])

Out[153]: <matplotlib.axes._subplots.AxesSubplot at 0x2b9da9a91c8>
```



Percentage of fraudulent profiles based on different scenarios.

Out[180]: <matplotlib.axes._subplots.AxesSubplot at 0x2b5dc41b548>



Feature Engineering

(2)

d	benefits_d	length_description	length_company_profile	length_requirements	length_benefits	Minimum_salary_range	Maximum_salary_range
	['Unspecified']	90	90	75	1	1	3
	['get', 'usthrough', 'part', 'second', 'team',...	97	97	121	108	1	3

e	required_experience	required_education	fraudulent	Minimum_salary_range	Maximum_salary_range	text	char_length	word_length	word_density
	2	-1	0	1	3	Marketing Intern Marketing unspecified Marketi...	2825	259	10.907336
	-1	-1	0	1	3	Customer Service - Cloud Video Production Succ...	5884	532	11.060150


communication' skill'
experience' working'
written' verbal'
client'
andor'
support'
best' practice'
experience' preferred'
attention' detail'
team' member'
process'
written' communication'
verbal' written'
operation'
ideal' candidate'
minimum' year'
join' team'
fast' paced'
amp'
project'
lead'
ability' work'
data'
design'
computer' science'
application'
well'
system'
product'
technology'
program'
problem' solving'
technical'
company'
high' level'
customer' service'
least' year'
track' record'
must' able'
able' work'
social' medium'
software' development'
year' experience'
use'
skill' ability'
school' diploma'
need'
etc'
full' time'
project' management'

Model Selection

```
In [61]: predicted_t=Decision_tree_1.predict(X_test_scaled)
```

```
In [64]: print(classification_report(predicted_t,y_test))
```

```
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      3403
           1       0.88      0.77      0.82       173

    accuracy                           0.98      3576
   macro avg       0.93      0.88      0.91      3576
weighted avg       0.98      0.98      0.98      3576
```

```
In [94]: print(predicted_tree)
```

```
0.9784675615212528
```

```
5]: predicted_xgb=xgb_.predict(X_test_scaled)
```

```
5]: print(classification_report(predicted_xgb,y_test))
```

```
              precision    recall  f1-score   support

           0       1.00      0.99      1.00      3443
           1       0.86      0.98      0.92       133

    accuracy                           0.99      3576
   macro avg       0.93      0.99      0.96      3576
weighted avg       0.99      0.99      0.99      3576
```

```
5]: print(predicted_xg_)
```

```
0.9932885906040269
```

```
             verbose=False)

In [105]: predict_voting=voting_classifier.predict(X_test_scaled)

In [107]: print(classification_report(predict_voting,y_test))

              precision    recall  f1-score   support

           0       1.00      0.99      0.99      3462
           1       0.75      1.00      0.85       114

    accuracy                           0.99      3576
   macro avg       0.87      0.99      0.92      3576
weighted avg       0.99      0.99      0.99      3576

In [112]: t=(predict_voting==y_test)

In [113]: print(t.mean())

0.9890939597315436
```

Finally we had selected voting classifier that which helps to improve performance the model by using Logistic Regression,SVC, Random Forest, Xgboost, LGBM.