Yao Yao VSCO Data Science Case Study

July 31, 2022

1 Yao Yao

VSCO Data Challenge

2 Data Science Case Study

Your company, Acme Co., sources candidates for companies hiring new employees. Recently, a number of our clients have complained that candidates have not been showing up to interviews. Your boss has provided you with the attached data set in hopes that you can find some way of identifying candidates at risk of not attending scheduled interviews

[1]:
```
import sys
try:
    sys.getwindowsversion()
except AttributeError:
    isWindows = False
else:
    isWindows = True
if isWindows:
    import win32api,win32process,win32con
    pid = win32api.GetCurrentProcessId()
    handle = win32api.OpenProcess(win32con.PROCESS_ALL_ACCESS, True, pid)
    win32process.SetPriorityClass(handle, win32process.HIGH_PRIORITY_CLASS)
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV,␣
 ↪train_test_split, KFold, StratifiedKFold
from sklearn.linear_model import LogisticRegression,LogisticRegressionCV
from sklearn.metrics import accuracy_score, classification_report,␣
 ↪confusion_matrix, roc_auc_score, roc_curve
from sklearn.tree import DecisionTreeClassifier
```

```
from sklearn.preprocessing import MinMaxScaler
from sklearn import metrics
from datetime import datetime
import folium
from folium.plugins import HeatMap
import itertools
from geopy.geocoders import Nominatim

%matplotlib inline
```

3 Automatically import NA's variations with blanks and fill them with np.nan, trim dataset to actual data

[2]:
```
df=pd.read_csv("Interview_Input.csv",nrows=1233,na_values=['na','NA','Na'])
```

4 Remove special characters, fill 'nan' as string, turn everything lowercase and trim whitespace for easier groupby functions later

[3]:
```
df=df.drop(df.columns[-5:], axis=1).replace('â€" ', '',regex=True)
df[df.select_dtypes(include='O').columns.values] = df[df.
 ↪select_dtypes(include='O').columns.values].apply(lambda x: x.astype(str).str.
 ↪lower()).apply(lambda x: x.astype(str).str.strip()).fillna('nan')
df
```

[3]:

	Date of Interview	Client name	Industry	Location	\
0	13.02.2015	hospira	pharmaceuticals	chennai	
1	13.02.2015	hospira	pharmaceuticals	chennai	
2	13.02.2015	hospira	pharmaceuticals	chennai	
3	13.02.2015	hospira	pharmaceuticals	chennai	
4	13.02.2015	hospira	pharmaceuticals	chennai	
…	…	…	…	…	
1228	06.02.2016	standard chartered bank	bfsi	chennai	
1229	30.1.16	standard chartered bank	bfsi	chennai	
1230	30.01.2016	standard chartered bank	bfsi	chennai	
1231	07.05.2016	pfizer	pharmaceuticals	chennai	
1232	06.05.2016	pfizer	pharmaceuticals	chennai	

	Position to be closed	Nature of Skillset	Interview Type	\
0	production- sterile	routine	scheduled walkin	
1	production- sterile	routine	scheduled walkin	
2	production- sterile	routine	scheduled walkin	
3	production- sterile	routine	scheduled walkin	
4	production- sterile	routine	scheduled walkin	
…	…	…	…	
1228	routine	java/j2ee/struts/hibernate	scheduled walk in	

```
1229               routine                          java  scheduled walk in
1230               routine                          java  scheduled walk in
1231                 niche                          emea          scheduled
1232                 niche          generic drugs - ra          scheduled

       Name(Cand ID)  Gender Candidate Current Location  …  \
0         candidate 1    male                     chennai  …
1         candidate 2    male                     chennai  …
2         candidate 3    male                     chennai  …
3         candidate 4    male                     chennai  …
4         candidate 5    male                     chennai  …
…               …     …                          …  …
1228  candidate 1171    male                     chennai  …
1229  candidate 1189  female                     chennai  …
1230  candidate 1207    male                     chennai  …
1231  candidate 1222    male                     chennai  …
1232  candidate 1233  female                     chennai  …

     Candidate Native location  \
0                        hosur
1                       trichy
2                      chennai
3                      chennai
4                      chennai
…                        …
1228                 hyderabad
1229                 hyderabad
1230                 hyderabad
1231                   chennai
1232                   kolkata

     Have you obtained the necessary permission to start at the required time  \
0                                                    yes
1                                                    yes
2                                                    nan
3                                                    yes
4                                                    yes
…                                                    …
1228                                                 yes
1229                                                 yes
1230                                                 yes
1231                                                 yes
1232                                                 nan

     Hope there will be no unscheduled meetings  \
0                                          yes
1                                          yes
```

```
2                                                  nan
3                                                  yes
4                                                  yes
…                                                  …
1228                                               yes
1229                                               yes
1230                                               yes
1231                                               yes
1232                                               nan

     Can I Call you three hours before the interview and follow up on your
attendance for the interview  \
0                                                    yes
1                                                    yes
2                                                    nan
3                                                     no
4                                                    yes
…                                                    …
1228                                                 yes
1229                                                 yes
1230                                                 yes
1231                                                 yes
1232                                                 nan

     Can I have an alternative number/ desk number. I assure you that I will not
trouble you too much  \
0                                                    yes
1                                                    yes
2                                                    nan
3                                                    yes
4                                                     no
…                                                    …
1228                                                 yes
1229                                                 yes
1230                                                 yes
1231                                                 yes
1232                                                 nan

     Have you taken a printout of your updated resume. Have you read the JD and
understood the same  \
0                                                    yes
1                                                    yes
2                                                    nan
3                                                     no
4                                                    yes
…                                                    …
1228                                                 yes
```

```
1229                                                  yes
1230                                                  yes
1231                                                  yes
1232                                                  nan

     Are you clear with the venue details and the landmark.  \
0                                                     yes
1                                                     yes
2                                                     nan
3                                                     yes
4                                                     yes
…                                                     …
1228                                                  yes
1229                                                  yes
1230                                                  yes
1231                                                  yes
1232                                                  nan

     Has the call letter been shared Observed Attendance Marital Status
0                                yes                  no         single
1                                yes                  no         single
2                                nan                  no         single
3                                yes                  no         single
4                                yes                  no        married
…                                …                    …            …
1228                             yes                 nan         single
1229                             yes                 nan         single
1230                             yes                 nan        married
1231                             yes                 nan         single
1232                             nan                 nan         single

[1233 rows x 22 columns]
```

5 Initial glance at data summaries and what the dataset looks like in terms of unique values and frequencies of values

[4]:
```
df.describe(include = 'all').T
```

[4]:
```
                                      count unique  \
Date of Interview                      1233     95
Client name                            1233     15
Industry                               1233      7
Location                               1233      8
Position to be closed                  1233      7
Nature of Skillset                     1233     81
Interview Type                         1233      5
```

Name(Cand ID)	1233	1233
Gender	1233	2
Candidate Current Location	1233	7
Candidate Job Location	1233	7
Interview Venue	1233	7
Candidate Native location	1233	46
Have you obtained the necessary permission to s…	1233	5
Hope there will be no unscheduled meetings	1233	5
Can I Call you three hours before the interview…	1233	4
Can I have an alternative number/ desk number. …	1233	4
Have you taken a printout of your updated resum…	1233	5
Are you clear with the venue details and the la…	1233	4
Has the call letter been shared	1233	8
Observed Attendance	1233	3
Marital Status	1233	2

	top \
Date of Interview	06.02.2016
Client name	standard chartered bank
Industry	bfsi
Location	chennai
Position to be closed	routine
Nature of Skillset	java/j2ee/struts/hibernate
Interview Type	scheduled walk in
Name(Cand ID)	candidate 1
Gender	male
Candidate Current Location	chennai
Candidate Job Location	chennai
Interview Venue	chennai
Candidate Native location	chennai
Have you obtained the necessary permission to s…	yes
Hope there will be no unscheduled meetings	yes
Can I Call you three hours before the interview…	yes
Can I have an alternative number/ desk number. …	yes
Have you taken a printout of your updated resum…	yes
Are you clear with the venue details and the la…	yes
Has the call letter been shared	yes
Observed Attendance	yes
Marital Status	single

	freq
Date of Interview	220
Client name	904
Industry	949
Location	844
Position to be closed	1023

```
Nature of Skillset                                    220
Interview Type                                        456
Name(Cand ID)                                           1
Gender                                                965
Candidate Current Location                            844
Candidate Job Location                                893
Interview Venue                                       852
Candidate Native location                             595
Have you obtained the necessary permission to s…      921
Hope there will be no unscheduled meetings            954
Can I Call you three hours before the interview…      955
Can I have an alternative number/ desk number. …      937
Have you taken a printout of your updated resum…      942
Are you clear with the venue details and the la…      948
Has the call letter been shared                       934
Observed Attendance                                   729
Marital Status                                        767
```

6 Fix date formats

[5]:
```
df['Date of Interview']
```

[5]:
```
0       13.02.2015
1       13.02.2015
2       13.02.2015
3       13.02.2015
4       13.02.2015
           …
1228    06.02.2016
1229       30.1.16
1230    30.01.2016
1231    07.05.2016
1232    06.05.2016
Name: Date of Interview, Length: 1233, dtype: object
```

7 Remove time from date with '&' delimiter and trim white space

[6]:
```
df['Date of Interview'] = df['Date of Interview'].str.split('&', expand=False).
  ↪str[0].replace(' ', '',regex=True)
```

8 Datetime variation conversions

[7]:
```
formats = ['%m.%d.%Y',
'%d.%m.%Y',
'%d.%m.%y',
```

```
'%m.%d.%y',
'%m-%d-%Y',
'%d-%m-%Y',
'%d-%m-%y',
'%m-%d-%y',
'%m/%d/%Y',
'%d/%m/%Y',
'%d/%m/%y',
'%m/%d/%y',
'%d-%b-%y',
'%d-%b-%y']
```

9 Iterate through the date formats until the coerced NaT's are filled with the correct date, write to dataframe column after loop

[8]:
```
date0 = pd.to_datetime(df['Date of Interview'], errors='coerce')

for form in formats:
    date1 = pd.to_datetime(df['Date of Interview'], errors='coerce',␣
 ↪format=form)
    date0 = date0.fillna(date1)

df['Date of Interview'] = date0
df['Date of Interview']
```

[8]:
```
0      2015-02-13
1      2015-02-13
2      2015-02-13
3      2015-02-13
4      2015-02-13
          …
1228   2016-06-02
1229   2016-01-30
1230   2016-01-30
1231   2016-07-05
1232   2016-06-05
Name: Date of Interview, Length: 1233, dtype: datetime64[ns]
```

[9]:
```
df['Date of Interview'].describe()
```

[9]:
```
count                    1233
unique                     62
top       2016-06-02 00:00:00
freq                      220
first     2014-03-18 00:00:00
```

```
last      2023-12-04 00:00:00
Name: Date of Interview, dtype: object
```

10 Test comparisons for datetime conversions before and after

[10]:
```
# for i,j in enumerate(zip(df['Date of Interview'],df['Date of Interview2'])):
#     print(i,j)
```

11 Add extra columns such as day of week, date, month, year, diff between earliest date and current date column creation

[11]:
```
df['dayofweek']=df['Date of Interview'].dt.dayofweek
df['dayofweek']
```

[11]:
```
0       4
1       4
2       4
3       4
4       4
       ..
1228    3
1229    5
1230    5
1231    1
1232    6
Name: dayofweek, Length: 1233, dtype: int64
```

[12]:
```
df['Date of Interview'].min()
```

[12]:
```
Timestamp('2014-03-18 00:00:00')
```

[13]:
```
df['month']=df['Date of Interview'].dt.month
df['month']
```

[13]:
```
0       2
1       2
2       2
3       2
4       2
       ..
1228    6
1229    1
1230    1
1231    7
1232    6
Name: month, Length: 1233, dtype: int64
```

[14]:
```
df['date']=df['Date of Interview'].dt.day
df['date']
```

[14]:
```
0       13
1       13
2       13
3       13
4       13
        ..
1228     2
1229    30
1230    30
1231     5
1232     5
Name: date, Length: 1233, dtype: int64
```

[15]:
```
df['year']=df['Date of Interview'].dt.year
df['year']
```

[15]:
```
0       2015
1       2015
2       2015
3       2015
4       2015
        …
1228    2016
1229    2016
1230    2016
1231    2016
1232    2016
Name: year, Length: 1233, dtype: int64
```

[16]:
```
df['Days since earliest date']=(df['Date of Interview'] - df['Date of␣
 ↪Interview'].min()).dt.days
df['Days since earliest date']
```

[16]:
```
0       332
1       332
2       332
3       332
4       332
       …
1228    807
1229    683
1230    683
1231    840
1232    810
```

```
Name: Days since earliest date, Length: 1233, dtype: int64
```

[17]:
```
df['Name(Cand ID)'] = df['Name(Cand ID)'].str[10:].astype(int)
df['Name(Cand ID)']
```

[17]:
```
0          1
1          2
2          3
3          4
4          5
        …
1228    1171
1229    1189
1230    1207
1231    1222
1232    1233
Name: Name(Cand ID), Length: 1233, dtype: int32
```

12 Fix spelling errors and combine extended "no" answers as 'no' for only 3 unique answers: Yes, No, Nan

[18]:
```
df['Interview Type'] = df['Interview Type'].replace('sceduled walkin',␣
 ↪'scheduled walk in').replace('scheduled walkin', 'scheduled walk in')
df['Interview Type'].describe()
```

[18]:
```
count                  1233
unique                    3
top       scheduled walk in
freq                    646
Name: Interview Type, dtype: object
```

13 Column list to fix to combine answers to their main grouping

[19]:
```
colname1=['Have you obtained the necessary permission to start at the required␣
 ↪time',
'Hope there will be no unscheduled meetings',
'Can I Call you three hours before the interview and follow up on your␣
 ↪attendance for the interview',
'Can I have an alternative number/ desk number. I assure you that I will not␣
 ↪trouble you too much',
'Have you taken a printout of your updated resume. Have you read the JD and␣
 ↪understood the same',
'Are you clear with the venue details and the landmark.',
'Has the call letter been shared']
```

14 Use df.loc to find the instance if the answer is neither 'yes' nor 'nan', set the extended 'no' answers as 'no' for the column to be properly groupedby for 3 main unique answers

[20]:
```
for col in colname1:
    df[col].loc[~((df[col] == 'yes') | (df[col] == 'nan'))]='no'
    display(df[col].describe())
```

```
count     1233
unique       3
top        yes
freq       921
Name: Have you obtained the necessary permission to start at the required time,␣
 ↪dtype: object

count     1233
unique       3
top        yes
freq       954
Name: Hope there will be no unscheduled meetings, dtype: object

count     1233
unique       3
top        yes
freq       955
Name: Can I Call you three hours before the interview and follow up on your␣
 ↪attendance for the interview, dtype: object

count     1233
unique       3
top        yes
freq       937
Name: Can I have an alternative number/ desk number. I assure you that I will␣
 ↪not trouble you too much, dtype: object

count     1233
unique       3
top        yes
freq       942
Name: Have you taken a printout of your updated resume. Have you read the JD and␣
 ↪understood the same, dtype: object

count     1233
unique       3
top        yes
freq       948
Name: Are you clear with the venue details and the landmark., dtype: object

count     1233
unique       3
```

```
top         yes
freq        934
Name: Has the call letter been shared, dtype: object
```

15 For the nature of skillset column, observe unique values and remove time and errors from the column

[21]:
```
df['Nature of Skillset'].unique()
```

[21]:
```
array(['routine', 'oracle', 'accounting operations', 'banking operations',
       'fresher', 'aml/kyc/cdd', 'cdd kyc', 'ra label', 'ra publishing',
       'lcm -manager', 'licensing - ra', 'biosimilars',
       'analytical r & d', 'analytical r&d',
       'senior software engineer-mednet', 'tech lead-mednet',
       'technical lead', 'sr automation testing', 'senior analyst',
       'production', 'regulatory', 'core java', 'oracle plsql',
       'automation testing java', 'submission management', 'publishing',
       'global labelling', 'als testing', 'java developer',
       'lending and liabilities', 'lending & liability',
       'java/j2ee/struts/hibernate', 'java/spring/hibernate/jsf',
       'java jsf', 'java,j2ee, jsf', 'java ,j2ee', 'java j2ee',
       '10.00 am', '9.00 am', 'java, j2ee', 'java,j2ee',
       'java/j2ee/core java', 'java', 'java/j2ee', 't-24 developer',
       'cots developer', 'dot net', 'testing', 'etl', 'java-sas',
       'java tech lead', 'sccm', 'sccm-(network, sharepoint,ms exchange)',
       'sccm - sharepoint', 'sas', 'java, spring, hibernate',
       'java,spring,hibernate', 'java, xml, struts, hibernate',
       'java,sql', 'biosimiliars', 'emea', 'tech lead- mednet', 'tl',
       'biosimillar', 'l & l', 'lending&liablities', '11.30 am',
       '12.30 pm', '9.30 am', 'product control', 'cots', '#name?',
       'manager', 'java, sql', 'hadoop', 'sccm- desktop support',
       'sccm- networking', 'production support - sccm',
       'basesas program/ reporting', 'generic drugs - ra', 'sccm - sql'],
      dtype=object)
```

[22]:
```
To_remove_lst = ['10.00 am', '9.00 am', '11.30 am', '12.30 pm', '9.30 am',␣
↪'#name?']
```

16 Remove special characters from the skillset column and trim whitespace and redundant white space

[23]:
```
df['Nature of Skillset'] = df['Nature of Skillset'].str.replace('|'.
↪join(To_remove_lst), '').str.replace('[^0-9a-zA-Z]', ' ').replace(r'\s+', '␣
↪', regex=True)
df['Nature of Skillset'].unique()
```

[23]:
```
array(['routine', 'oracle', 'accounting operations', 'banking operations',
       'fresher', 'aml kyc cdd', 'cdd kyc', 'ra label', 'ra publishing',
       'lcm manager', 'licensing ra', 'biosimilars', 'analytical r d',
       'senior software engineer mednet', 'tech lead mednet',
       'technical lead', 'sr automation testing', 'senior analyst',
       'production', 'regulatory', 'core java', 'oracle plsql',
       'automation testing java', 'submission management', 'publishing',
       'global labelling', 'als testing', 'java developer',
       'lending and liabilities', 'lending liability',
       'java j2ee struts hibernate', 'java spring hibernate jsf',
       'java jsf', 'java j2ee jsf', 'java j2ee', '',
       'java j2ee core java', 'java', 't 24 developer', 'cots developer',
       'dot net', 'testing', 'etl', 'java sas', 'java tech lead', 'sccm',
       'sccm network sharepoint ms exchange ', 'sccm sharepoint', 'sas',
       'java spring hibernate', 'java xml struts hibernate', 'java sql',
       'biosimiliars', 'emea', 'tl', 'biosimillar', 'l l',
       'lending liablities', 'product control', 'cots', ' ', 'manager',
       'hadoop', 'sccm desktop support', 'sccm networking',
       'production support sccm', 'basesas program reporting',
       'generic drugs ra', 'sccm sql'], dtype=object)
```

17 Count the number of unique skillsets and make that an extra column keeping track the number of skills

[24]:
```
df['Count of Skillset'] = df['Nature of Skillset'].str.count('\s+')+1
df['Count of Skillset'].unique()
```

[24]:
```
array([1, 2, 3, 4, 6], dtype=int64)
```

18 For all the words in the skillset, separate them into their own dummy variables because certain skills overlap for different positions to find the root skills per job

[25]:
```
Skillset = df['Nature of Skillset'].str.get_dummies(sep=' ')
Skillset
```

[25]:

	24	accounting	als	aml	analyst	analytical	and	automation	banking	\
0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	
…	..	…	…	…	…	…	…	…	…	
1228	0	0	0	0	0	0	0	0	0	
1229	0	0	0	0	0	0	0	0	0	

```
1230   0           0    0    0        0           0    0           0        0
1231   0           0    0    0        0           0    0           0        0
1232   0           0    0    0        0           0    0           0        0

      basesas  …  sr  struts  submission  support  t  tech  technical  \
0           0  …   0       0           0        0  0     0          0
1           0  …   0       0           0        0  0     0          0
2           0  …   0       0           0        0  0     0          0
3           0  …   0       0           0        0  0     0          0
4           0  …   0       0           0        0  0     0          0
…       …  …  ..     …          …       … ..    …        …
1228        0  …   0       1           0        0  0     0          0
1229        0  …   0       0           0        0  0     0          0
1230        0  …   0       0           0        0  0     0          0
1231        0  …   0       0           0        0  0     0          0
1232        0  …   0       0           0        0  0     0          0

      testing  tl  xml
0           0   0    0
1           0   0    0
2           0   0    0
3           0   0    0
4           0   0    0
…       …   ..  …
1228        0   0    0
1229        0   0    0
1230        0   0    0
1231        0   0    0
1232        0   0    0

[1233 rows x 81 columns]
```

19 Tally up the skillsets and limit dummy variables for words that appear more than 4 times

[26]:
```
Skillset2 = pd.DataFrame(Skillset.sum())
Skillset2
```

[26]:
```
              0
24           15
accounting   86
als          15
aml          84
analyst       5
…            ..
tech         15
```

```
technical    1
testing     46
tl           3
xml          3

[81 rows x 1 columns]
```

[27]:
```
Skillset2 =Skillset2[Skillset2[0]>4].index
Skillset2
```

[27]:
```
Index(['24', 'accounting', 'als', 'aml', 'analyst', 'analytical', 'and',
       'automation', 'banking', 'biosimiliars', 'cdd', 'control', 'core',
       'cots', 'd', 'developer', 'dot', 'emea', 'engineer', 'etl', 'fresher',
       'global', 'hadoop', 'hibernate', 'j2ee', 'java', 'jsf', 'kyc',
       'labelling', 'lead', 'lending', 'liabilities', 'mednet', 'net',
       'operations', 'oracle', 'plsql', 'product', 'production', 'publishing',
       'r', 'ra', 'regulatory', 'routine', 'sas', 'sccm', 'senior', 'software',
       'spring', 'sql', 'sr', 'struts', 'support', 't', 'tech', 'testing'],
      dtype='object')
```

20 this subset of dummy variables are the ones we will combine later to the main dataset

[28]:
```
Skillset = Skillset[Skillset2]
Skillset
```

[28]:
```
      24  accounting  als  aml  analyst  analytical  and  automation  banking  \
0      0           0    0    0        0           0    0           0        0
1      0           0    0    0        0           0    0           0        0
2      0           0    0    0        0           0    0           0        0
3      0           0    0    0        0           0    0           0        0
4      0           0    0    0        0           0    0           0        0
…    ..         …  …  …      …         …  …         …      …
1228   0           0    0    0        0           0    0           0        0
1229   0           0    0    0        0           0    0           0        0
1230   0           0    0    0        0           0    0           0        0
1231   0           0    0    0        0           0    0           0        0
1232   0           0    0    0        0           0    0           0        0

      biosimiliars  …  senior  software  spring  sql  sr  struts  support  \
0                0  …       0         0       0    0   0       0        0
1                0  …       0         0       0    0   0       0        0
2                0  …       0         0       0    0   0       0        0
3                0  …       0         0       0    0   0       0        0
4                0  …       0         0       0    0   0       0        0
…              …  …     …        …      …  …  ..     …      …
```

```
1228            0  …       0         0       0    0   0       1        0
1229            0  …       0         0       0    0   0       0        0
1230            0  …       0         0       0    0   0       0        0
1231            0  …       0         0       0    0   0       0        0
1232            0  …       0         0       0    0   0       0        0

      t  tech  testing
0     0     0        0
1     0     0        0
2     0     0        0
3     0     0        0
4     0     0        0
…   ..   …       …
1228  0     0        0
1229  0     0        0
1230  0     0        0
1231  0     0        0
1232  0     0        0

[1233 rows x 56 columns]
```

21 For the 5 location columns, correct the misspelled words

[29]:
```
df = df.replace('delhi /ncr','delhi ncr').
 ↪replace('visakapatinam','visakhapatnam').replace('gurgaonr','gurgaon').
 ↪replace('- cochin-','cochin')
df
```

[29]:
```
     Date of Interview              Client name         Industry Location  \
0           2015-02-13                  hospira  pharmaceuticals  chennai
1           2015-02-13                  hospira  pharmaceuticals  chennai
2           2015-02-13                  hospira  pharmaceuticals  chennai
3           2015-02-13                  hospira  pharmaceuticals  chennai
4           2015-02-13                  hospira  pharmaceuticals  chennai
…                 …                        …                …      …
1228        2016-06-02  standard chartered bank             bfsi  chennai
1229        2016-01-30  standard chartered bank             bfsi  chennai
1230        2016-01-30  standard chartered bank             bfsi  chennai
1231        2016-07-05                  pfizer  pharmaceuticals  chennai
1232        2016-06-05                  pfizer  pharmaceuticals  chennai

     Position to be closed          Nature of Skillset     Interview Type  \
0      production- sterile                     routine  scheduled walk in
1      production- sterile                     routine  scheduled walk in
2      production- sterile                     routine  scheduled walk in
3      production- sterile                     routine  scheduled walk in
```

```
4      production- sterile                    routine  scheduled walk in
…                    …                        …              …
1228               routine  java j2ee struts hibernate  scheduled walk in
1229               routine                        java  scheduled walk in
1230               routine                        java  scheduled walk in
1231                 niche                        emea          scheduled
1232                 niche           generic drugs ra           scheduled

      Name(Cand ID)  Gender Candidate Current Location  …  \
0                 1    male                    chennai  …
1                 2    male                    chennai  …
2                 3    male                    chennai  …
3                 4    male                    chennai  …
4                 5    male                    chennai  …
…               …     …                       …  …
1228           1171    male                    chennai  …
1229           1189  female                    chennai  …
1230           1207    male                    chennai  …
1231           1222    male                    chennai  …
1232           1233  female                    chennai  …

     Are you clear with the venue details and the landmark.  \
0                                                    yes
1                                                    yes
2                                                    nan
3                                                    yes
4                                                    yes
…                                                    …
1228                                                 yes
1229                                                 yes
1230                                                 yes
1231                                                 yes
1232                                                 nan

     Has the call letter been shared Observed Attendance Marital Status  \
0                                yes                  no         single
1                                yes                  no         single
2                                nan                  no         single
3                                yes                  no         single
4                                yes                  no        married
…                                …                    …              …
1228                             yes                 nan         single
1229                             yes                 nan         single
1230                             yes                 nan        married
1231                             yes                 nan         single
1232                             nan                 nan         single
```

```
     dayofweek month date  year Days since earliest date Count of Skillset
0            4     2   13  2015                      332                  1
1            4     2   13  2015                      332                  1
2            4     2   13  2015                      332                  1
3            4     2   13  2015                      332                  1
4            4     2   13  2015                      332                  1
…          …   …   …   …                          …              …
1228         3     6    2  2016                      807                  4
1229         5     1   30  2016                      683                  1
1230         5     1   30  2016                      683                  1
1231         1     7    5  2016                      840                  1
1232         6     6    5  2016                      810                  3

[1233 rows x 28 columns]
```

22 Find the unique places for the 5 location columns

[30]:
```
places = pd.Series(np.concatenate([df['Location'].unique(), df['Candidate␣
 ↪Current Location'].unique(), df['Candidate Job Location'].unique(),␣
 ↪df['Interview Venue'].unique(), df['Candidate Native location'].unique()])).
 ↪sort_values().unique()
places
```

[30]:
```
array(['agra', 'ahmedabad', 'allahabad', 'ambur', 'anantapur', 'baddi',
       'bangalore', 'belgaum', 'bhubaneshwar', 'chandigarh', 'chennai',
       'chitoor', 'cochin', 'coimbatore', 'cuttack', 'delhi', 'delhi ncr',
       'faizabad', 'ghaziabad', 'gurgaon', 'hissar', 'hosur', 'hyderabad',
       'kanpur', 'kolkata', 'kurnool', 'lucknow', 'mumbai', 'mysore',
       'nagercoil', 'noida', 'panjim', 'patna', 'pondicherry', 'pune',
       'salem', 'tanjore', 'tirupati', 'trichy', 'trivandrum',
       'tuticorin', 'vellore', 'vijayawada', 'visakhapatnam', 'warangal'],
      dtype=object)
```

23 use a geolocater login to retrieve the longitudes and latitudes in India to find the distance between the candidate and the other location columns

[31]:
```
geolocator = Nominatim(user_agent='myapplication')
geocode = []

for place in places:
    location = geolocator.geocode(place + ', india')
    print(place, location.raw['lat'],location.raw['lon'])
    geocode.append([place, location.raw['lat'],location.raw['lon']])
```

```
agra 27.1752554 78.0098161
```

```
ahmedabad 23.0216238 72.5797068
allahabad 25.4381302 81.8338005
ambur 12.7929067 78.6999168287325
anantapur 14.6546235 77.55625984224562
baddi 30.9763026 76.7674000810934
bangalore 12.9767936 77.590082
belgaum 15.8572666 74.5069343
bhubaneshwar 20.2602964 85.8394521
chandigarh 30.7334421 76.7797143
chennai 13.0836939 80.270186
chitoor 13.0929032 80.266619
cochin 9.931308 76.2674136
coimbatore 11.0018115 76.9628425
cuttack 20.4686 85.8792
delhi 28.6517178 77.2219388
delhi ncr 28.6882438 77.1212148
faizabad 26.63807555 82.05902434378625
ghaziabad 28.7218316 77.45268496448504
gurgaon 28.42826235 77.00270014657752
hissar 29.9918409 76.6077414
hosur 12.7328844 77.8309478
hyderabad 17.360589 78.4740613
kanpur 26.4609135 80.3217588
kolkata 22.5414185 88.35769124388872
kurnool 15.8309251 78.0425373
lucknow 26.8381 80.9346001
mumbai 19.0785451 72.878176
mysore 12.3051828 76.6553609
nagercoil 8.1880471 77.4290492
noida 28.5707841 77.3271074
panjim 15.4989946 73.8282141
patna 25.6093239 85.1235252
pondicherry 11.9340568 79.8306447
pune 18.521428 73.8544541
salem 11.6612012 78.1602498
tanjore 10.7860267 79.1381497
tirupati 13.77928955 79.83512262283737
trichy 10.804973 78.6870296
trivandrum 8.4882267 76.947551
tuticorin 8.8052602 78.1452745
vellore 12.7948109 79.0006410968549
vijayawada 16.5087586 80.6185102
visakhapatnam 17.7231276 83.3012842
warangal 17.9820644 79.5970954
```

24 Use this new dataframe to merge left onto the main dataframe for the locations provided using a prefix per location column

[32]:
```
location = pd.DataFrame(geocode,columns = ['location','lat','long'])
location[['lat','long']] = location[['lat','long']].astype(float)
location
```

[32]:

	location	lat	long
0	agra	27.175255	78.009816
1	ahmedabad	23.021624	72.579707
2	allahabad	25.438130	81.833800
3	ambur	12.792907	78.699917
4	anantapur	14.654623	77.556260
5	baddi	30.976303	76.767400
6	bangalore	12.976794	77.590082
7	belgaum	15.857267	74.506934
8	bhubaneshwar	20.260296	85.839452
9	chandigarh	30.733442	76.779714
10	chennai	13.083694	80.270186
11	chitoor	13.092903	80.266619
12	cochin	9.931308	76.267414
13	coimbatore	11.001812	76.962842
14	cuttack	20.468600	85.879200
15	delhi	28.651718	77.221939
16	delhi ncr	28.688244	77.121215
17	faizabad	26.638076	82.059024
18	ghaziabad	28.721832	77.452685
19	gurgaon	28.428262	77.002700
20	hissar	29.991841	76.607741
21	hosur	12.732884	77.830948
22	hyderabad	17.360589	78.474061
23	kanpur	26.460914	80.321759
24	kolkata	22.541418	88.357691
25	kurnool	15.830925	78.042537
26	lucknow	26.838100	80.934600
27	mumbai	19.078545	72.878176
28	mysore	12.305183	76.655361
29	nagercoil	8.188047	77.429049
30	noida	28.570784	77.327107
31	panjim	15.498995	73.828214
32	patna	25.609324	85.123525
33	pondicherry	11.934057	79.830645
34	pune	18.521428	73.854454
35	salem	11.661201	78.160250
36	tanjore	10.786027	79.138150
37	tirupati	13.779290	79.835123
38	trichy	10.804973	78.687030

```
39     trivandrum   8.488227  76.947551
40      tuticorin   8.805260  78.145274
41        vellore  12.794811  79.000641
42     vijayawada  16.508759  80.618510
43  visakhapatnam  17.723128  83.301284
44       warangal  17.982064  79.597095
```

25 as we find out, the Location and Candidate Current Location columns are redundant so Location is later removed from the dataset

[33]:
```
df = pd.merge(df,location.add_prefix('Current_'),how='left',left_on='Candidate␣
↪Current Location',right_on='Current_location').drop(columns =␣
↪'Current_location')
```

[34]:
```
df = pd.merge(df,location.add_prefix('Job_'),how='left',left_on='Candidate Job␣
↪Location',right_on='Job_location').drop(columns = 'Job_location')
```

[35]:
```
df = pd.merge(df,location.
↪add_prefix('Interview_'),how='left',left_on='Interview␣
↪Venue',right_on='Interview_location').drop(columns = 'Interview_location')
```

[36]:
```
df = pd.merge(df,location.add_prefix('Native_'),how='left',left_on='Candidate␣
↪Native location',right_on='Native_location').drop(columns =␣
↪'Native_location')
```

[37]:
```
df
```

[37]:

	Date of Interview	Client name	Industry	Location	\
0	2015-02-13	hospira	pharmaceuticals	chennai	
1	2015-02-13	hospira	pharmaceuticals	chennai	
2	2015-02-13	hospira	pharmaceuticals	chennai	
3	2015-02-13	hospira	pharmaceuticals	chennai	
4	2015-02-13	hospira	pharmaceuticals	chennai	
…	…	…	…	…	
1228	2016-06-02	standard chartered bank	bfsi	chennai	
1229	2016-01-30	standard chartered bank	bfsi	chennai	
1230	2016-01-30	standard chartered bank	bfsi	chennai	
1231	2016-07-05	pfizer	pharmaceuticals	chennai	
1232	2016-06-05	pfizer	pharmaceuticals	chennai	

	Position to be closed	Nature of Skillset	Interview Type	\
0	production- sterile	routine	scheduled walk in	
1	production- sterile	routine	scheduled walk in	
2	production- sterile	routine	scheduled walk in	
3	production- sterile	routine	scheduled walk in	

```
4      production- sterile                      routine  scheduled walk in
…                    …                          …              …
1228                routine  java j2ee struts hibernate  scheduled walk in
1229                routine                        java  scheduled walk in
1230                routine                        java  scheduled walk in
1231                  niche                        emea          scheduled
1232                  niche           generic drugs ra          scheduled

      Name(Cand ID)  Gender Candidate Current Location  …  \
0                 1    male                    chennai  …
1                 2    male                    chennai  …
2                 3    male                    chennai  …
3                 4    male                    chennai  …
4                 5    male                    chennai  …
…               …     …                       …  …
1228           1171    male                    chennai  …
1229           1189  female                    chennai  …
1230           1207    male                    chennai  …
1231           1222    male                    chennai  …
1232           1233  female                    chennai  …

     Days since earliest date Count of Skillset Current_lat Current_long  \
0                         332                 1   13.083694    80.270186
1                         332                 1   13.083694    80.270186
2                         332                 1   13.083694    80.270186
3                         332                 1   13.083694    80.270186
4                         332                 1   13.083694    80.270186
…                        …                  …          …           …
1228                      807                 4   13.083694    80.270186
1229                      683                 1   13.083694    80.270186
1230                      683                 1   13.083694    80.270186
1231                      840                 1   13.083694    80.270186
1232                      810                 3   13.083694    80.270186

        Job_lat   Job_long Interview_lat Interview_long Native_lat Native_long
0     12.732884  77.830948     12.732884      77.830948  12.732884   77.830948
1     12.976794  77.590082     12.732884      77.830948  10.804973   78.687030
2     13.083694  80.270186     12.732884      77.830948  13.083694   80.270186
3     13.083694  80.270186     12.732884      77.830948  13.083694   80.270186
4     12.976794  77.590082     12.732884      77.830948  13.083694   80.270186
…         …        …             …              …          …           …
1228  13.083694  80.270186     13.083694      80.270186  17.360589   78.474061
1229  13.083694  80.270186     13.083694      80.270186  17.360589   78.474061
1230  13.083694  80.270186     13.083694      80.270186  17.360589   78.474061
1231  13.083694  80.270186     13.083694      80.270186  13.083694   80.270186
1232  13.083694  80.270186     13.083694      80.270186  22.541418   88.357691
```

```
[1233 rows x 36 columns]
```

26 write a function to calculate the distance between 2 longitude and latitude coordinates in miles

[38]:
```
#find distance between user and restaurant
def haversine_np(lon1, lat1, lon2, lat2):

    lon1, lat1, lon2, lat2 = map(np.radians, [lon1, lat1, lon2, lat2])

    dlon = lon2 - lon1
    dlat = lat2 - lat1

    a = np.sin(dlat/2.0)**2 + np.cos(lat1) * np.cos(lat2) * np.sin(dlon/2.0)**2

    c = 2 * np.arcsin(np.sqrt(a))
    mi = 3956 * c # Radius of earth in miles
    return mi
```

27 Relative to the candidate's current location, find the distance between the candidate and the job, the candidate and the interview, and the candidate and their native location

[39]:
```
df['Location_to_Job_Dist_Miles'] = df.apply(lambda row:␣
 ↪haversine_np(row['Current_long'],
                                                          ␣
 ↪row['Current_lat'],
                                                          ␣
 ↪row['Job_long'],
                                                          ␣
 ↪row['Job_lat']), axis=1)
```

[40]:
```
df['Location_to_Interview_Dist_Miles'] = df.apply(lambda row:␣
 ↪haversine_np(row['Current_long'],
                                                          ␣
 ↪row['Current_lat'],
                                                          ␣
 ↪row['Interview_long'],
                                                          ␣
 ↪row['Interview_lat']), axis=1)
```

[41]:
```
df['Location_to_Native_Dist_Miles'] = df.apply(lambda row:␣
 ↪haversine_np(row['Current_long'],
```

```
                                                                    ␣
 ↪row['Current_lat'],
                                                                    ␣
 ↪row['Native_long'],
                                                                    ␣
 ↪row['Native_lat']), axis=1)
```

28 Most of the candidates are relatively close to the interview location and job site, while some candidates are far from their native location

[42]:
```
df[['Location_to_Job_Dist_Miles','Location_to_Interview_Dist_Miles','Location_to_Native_Dist_M
 ↪describe()
```

[42]:
```
       Location_to_Job_Dist_Miles  Location_to_Interview_Dist_Miles  \
count                 1233.000000                       1233.000000
mean                    16.476232                          4.333086
std                     67.168704                         31.973067
min                      0.000000                          0.000000
25%                      0.000000                          0.000000
50%                      0.000000                          0.000000
75%                      0.000000                          0.000000
max                    501.882489                        318.610407

       Location_to_Native_Dist_Miles
count                    1233.000000
mean                      144.669528
std                       241.568504
min                         0.000000
25%                         0.000000
50%                         0.000000
75%                       308.355896
max                      1243.882246
```

29 From the candidate's current location, plot a heatmap based on relative tally numbers. This could be done for native location and Job location as well

[43]:
```
Candidate_Current_Location = df[['Candidate Current␣
 ↪Location','Current_long','Current_lat']].groupby(['Candidate Current␣
 ↪Location','Current_long','Current_lat']).value_counts().reset_index()
Candidate_Current_Location
```

[43]:

	Candidate Current Location	Current_long	Current_lat	0
0	bangalore	77.590082	12.976794	292
1	chennai	80.270186	13.083694	844
2	cochin	76.267414	9.931308	9
3	delhi	77.221939	28.651718	1
4	gurgaon	77.002700	28.428262	34
5	hyderabad	78.474061	17.360589	38
6	noida	77.327107	28.570784	15

[44]:
```
d1 = Candidate_Current_Location
max_amount = d1[0].max()
hmap = folium.Map(location=[20.5937, 78.9629], zoom_start=5)

hm_wide = HeatMap(list(zip(d1.Current_lat, d1.Current_long, d1[0])),
                  min_opacity=0.2,
                  max_val=max_amount,
                  radius=17, blur=15,
                  max_zoom=1,
                 )

hmap.add_child(hm_wide)
```

[44]:
```
<folium.folium.Map at 0x20e9dd13850>
```

30 Visualization of continuous data by Observed Attendance, target variable

[45]:
```
plt.rcParams['figure.max_open_warning']=40
colnames=list(df.select_dtypes(exclude='O').columns.values)
for i in colnames[1:]:
    facet = sns.FacetGrid(df,hue='Observed Attendance',aspect=2)
    facet.map(sns.distplot,i)
    facet.add_legend()
    facet.fig.suptitle(''.join(map(str, list(["Figure ",colnames.index(i)+1,":␣
 ↪",i," Distribution By Label"]))))
    plt.show()
```

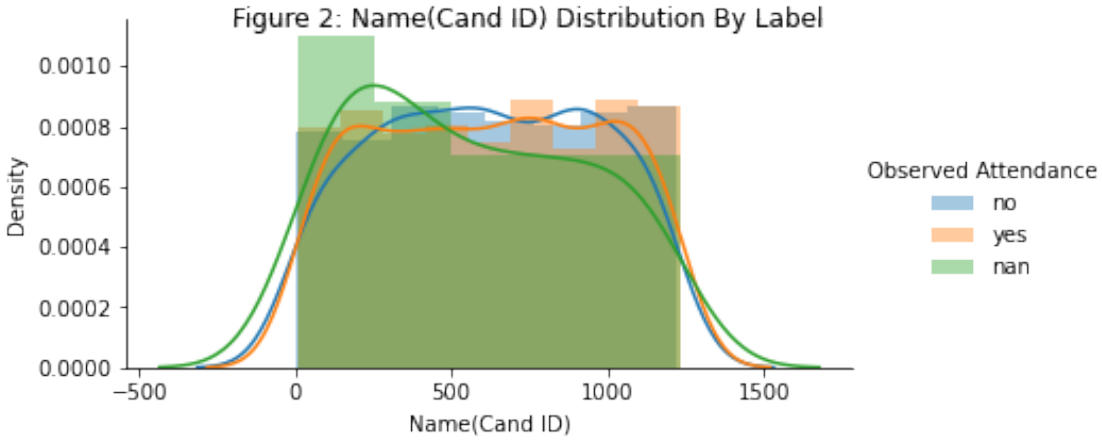


Figure 2: Name(Cand ID) Distribution By Label
Density
0.0010
0.0008
0.0006
0.0004
0.0002
0.0000
−500
0
500
1000
1500
Name(Cand ID)
Observed Attendance
no
yes
nan

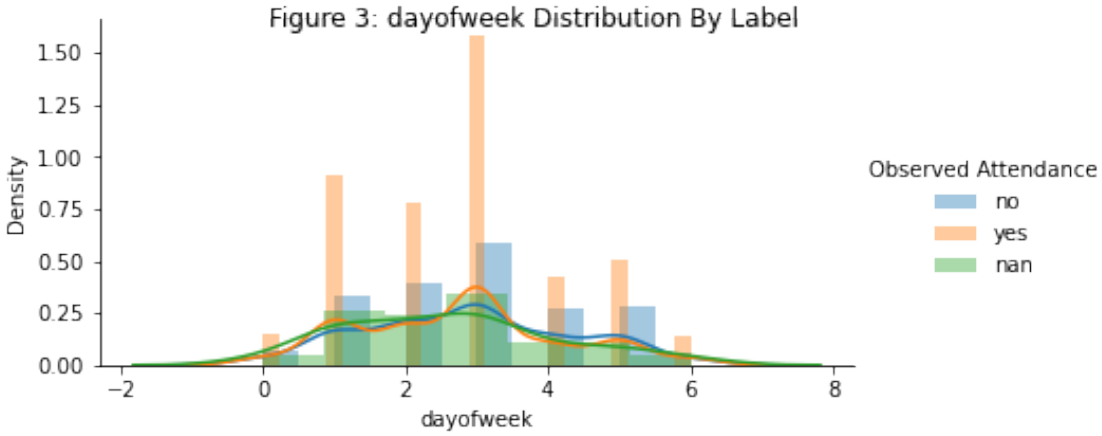


Figure 3: dayofweek Distribution By Label
Density
1.50
1.25
1.00
0.75
0.50
0.25
0.00
−2
0
2
4
6
8
dayofweek
Observed Attendance
no
yes
nan

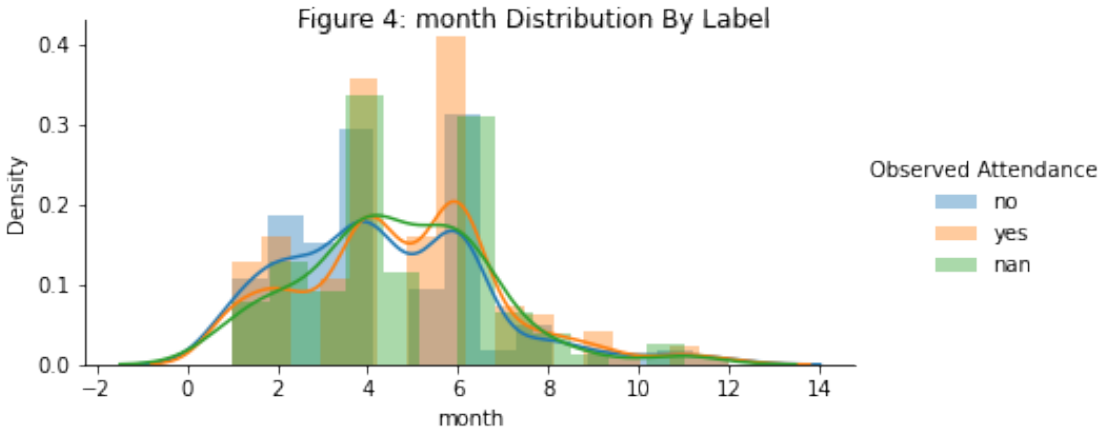


Figure 4: month Distribution By Label
Density
0.4
0.3
0.2
0.1
0.0
−2
0
2
4
6
8
10
12
14
month
Observed Attendance
no
yes
nan

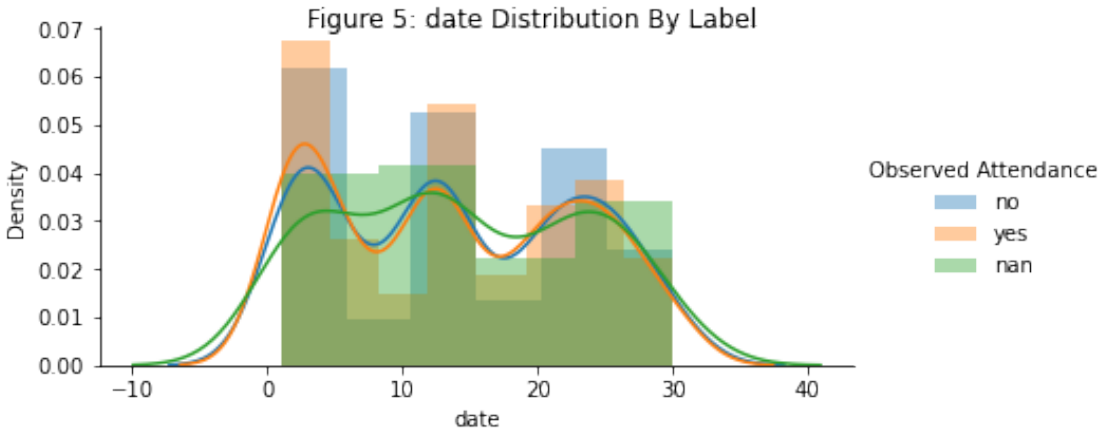


Figure 5: date Distribution By Label
Density
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0.00
−10
0
10
20
30
40
date
Observed Attendance
no
yes
nan

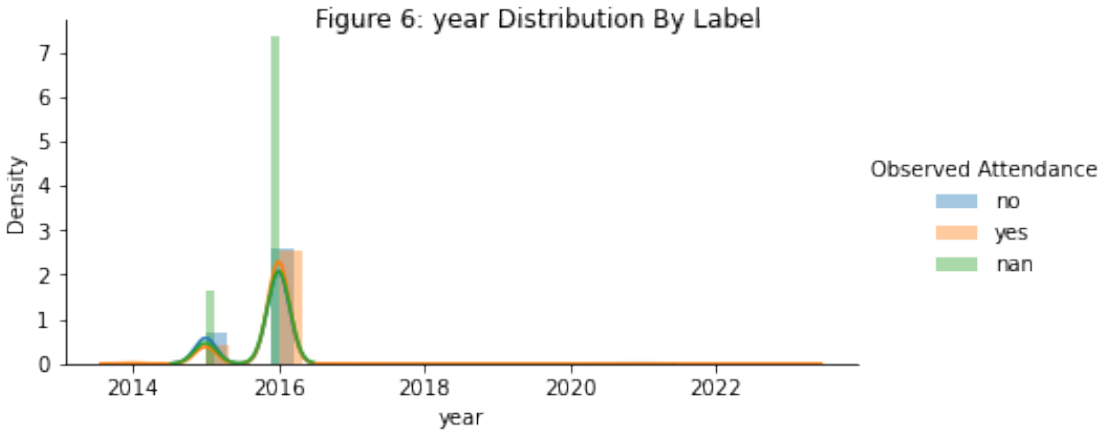


Figure 6: year Distribution By Label
Density
7
6
5
4
3
2
1
0
2014
2016
2018
2020
2022
year
Observed Attendance
no
yes
nan

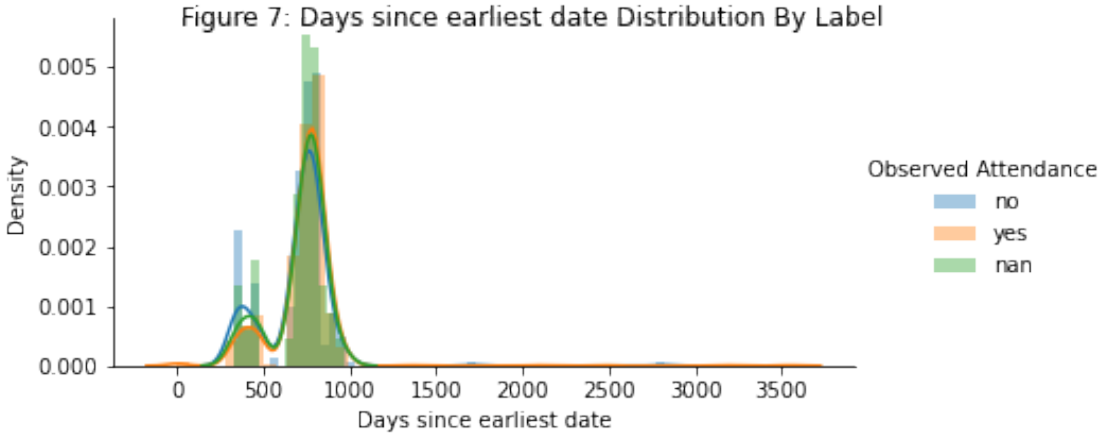


Figure 7: Days since earliest date Distribution By Label
Density
0.005
0.004
0.003
0.002
0.001
0.000
0
500
1000
1500
2000
2500
3000
3500
Days since earliest date
Observed Attendance
no
yes
nan

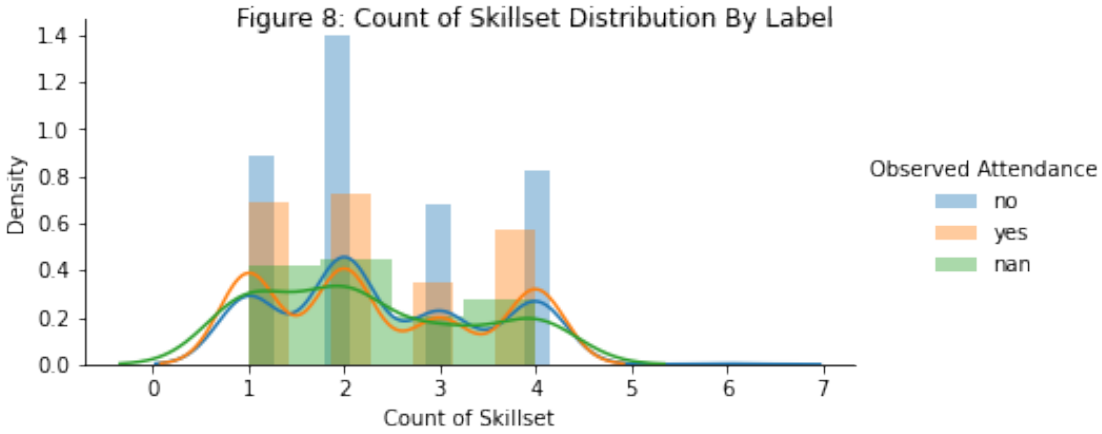


Figure 8: Count of Skillset Distribution By Label
Density
Count of Skillset
Observed Attendance
no
yes
nan

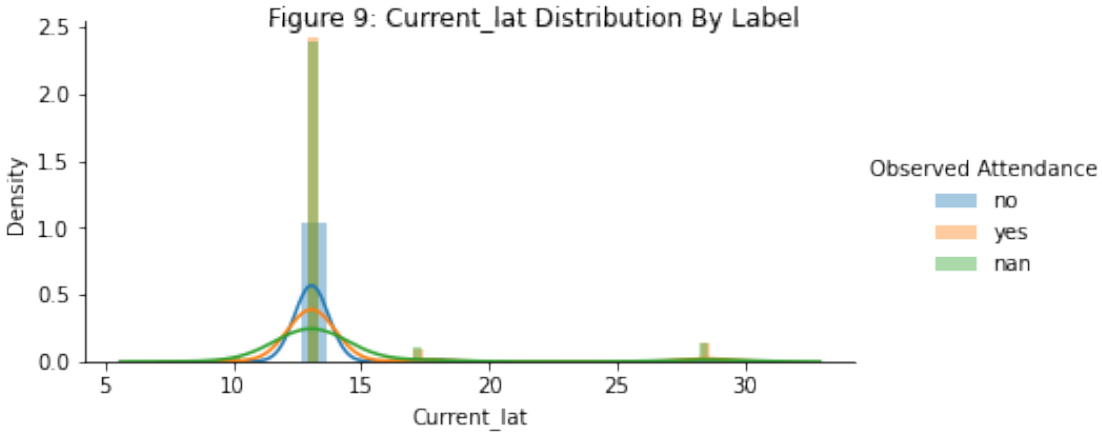


Figure 9: Current_lat Distribution By Label
Density
Current_lat
Observed Attendance
no
yes
nan

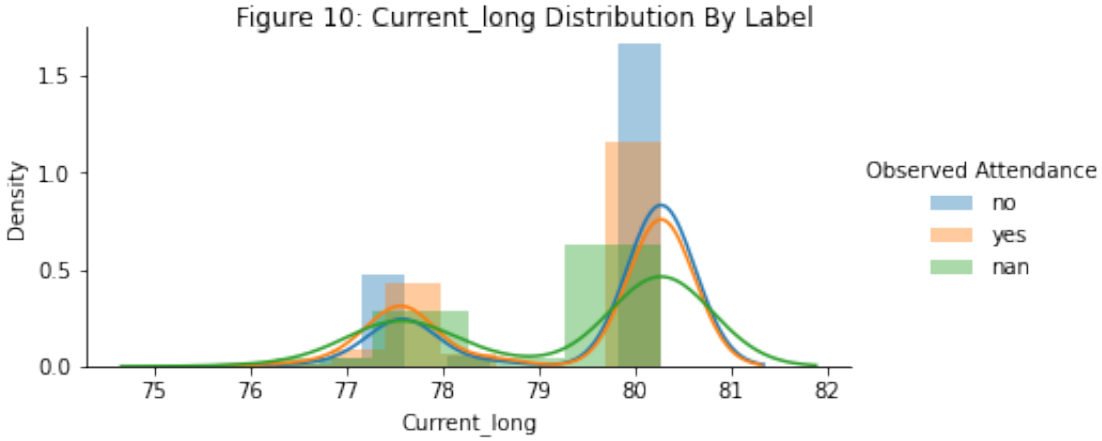


Figure 10: Current_long Distribution By Label
Density
Current_long
Observed Attendance
no
yes
nan

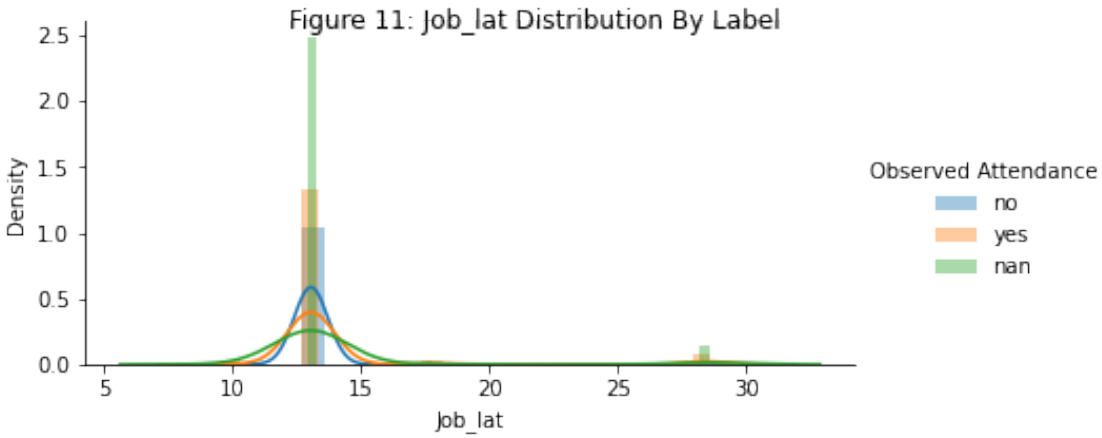


Figure 11: Job_lat Distribution By Label
Density
Job_lat
Observed Attendance
no
yes
nan

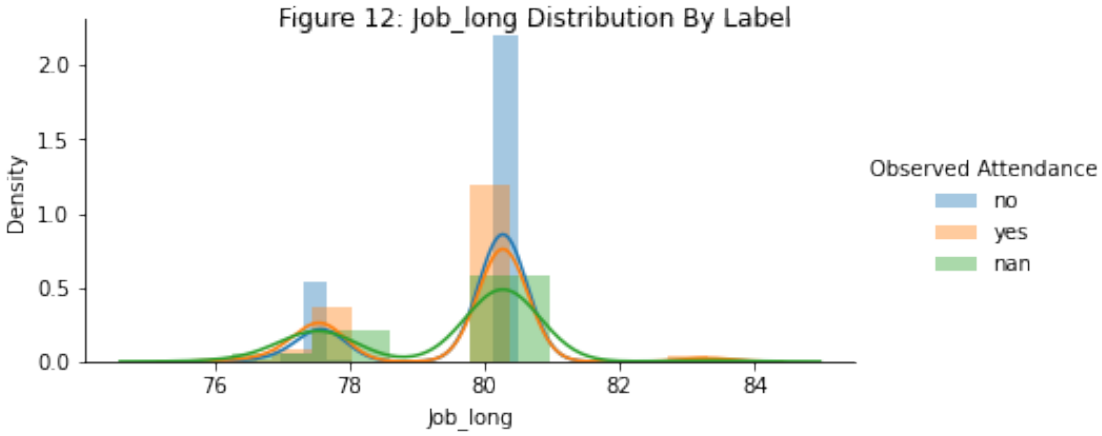


Figure 12: Job_long Distribution By Label
Density
Job_long
Observed Attendance
no
yes
nan

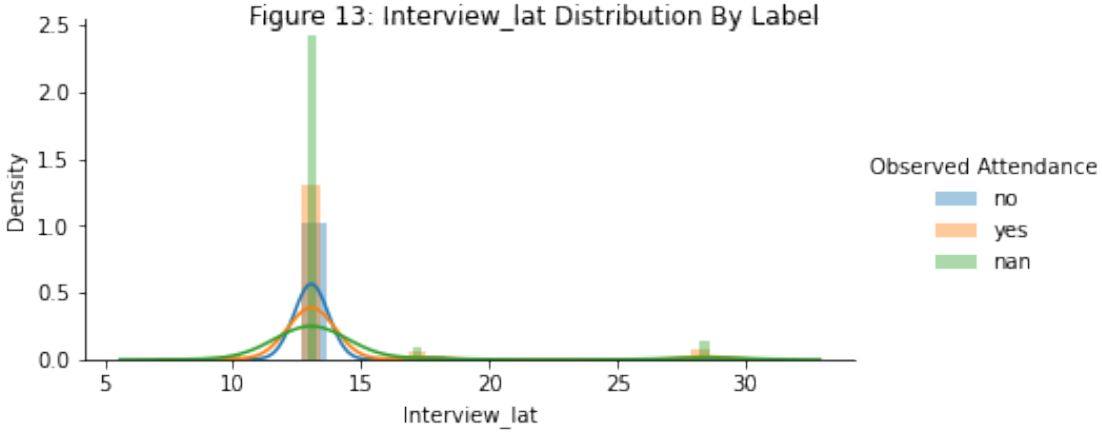


Figure 13: Interview_lat Distribution By Label
Density
Interview_lat
Observed Attendance
no
yes
nan

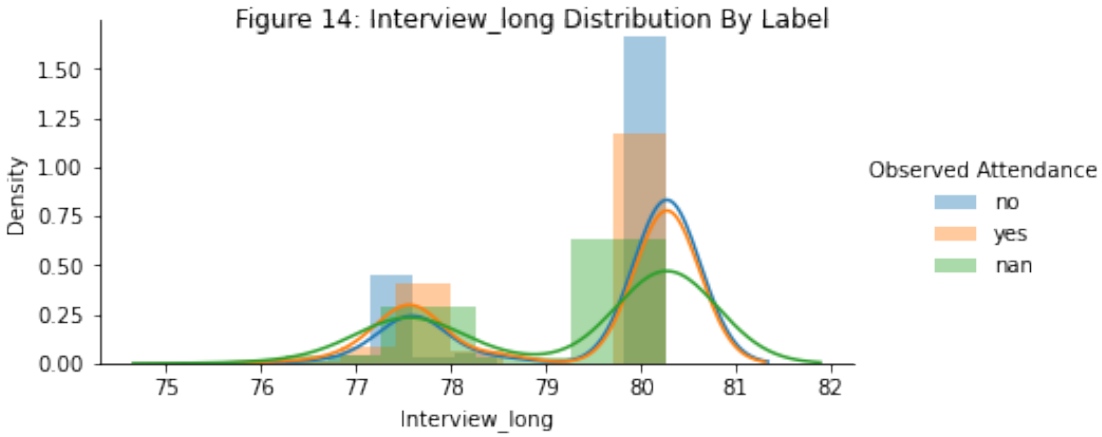


Figure 14: Interview_long Distribution By Label
Density
1.50
1.25
1.00
0.75
0.50
0.25
0.00
75
76
77
78
79
80
81
82
Interview_long
Observed Attendance
no
yes
nan

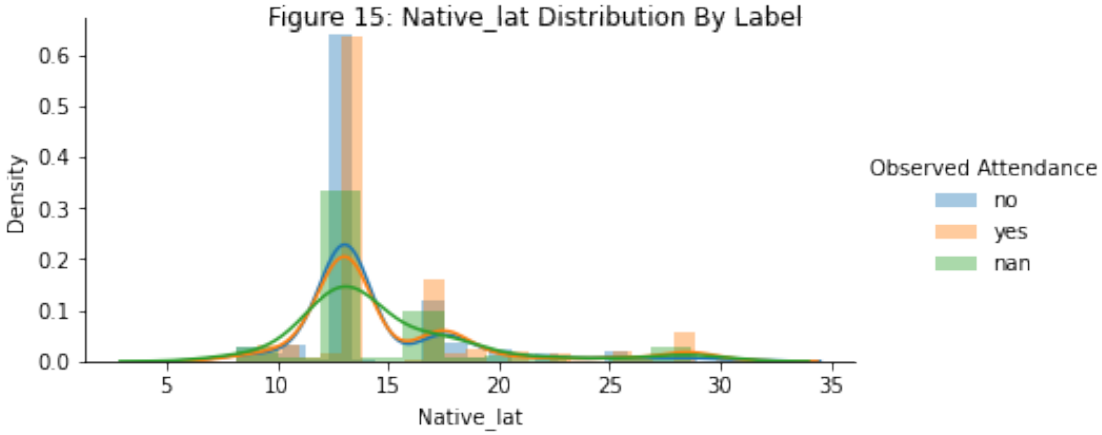


Figure 15: Native_lat Distribution By Label
Density
0.6
0.5
0.4
0.3
0.2
0.1
0.0
5
10
15
20
25
30
35
Native_lat
Observed Attendance
no
yes
nan

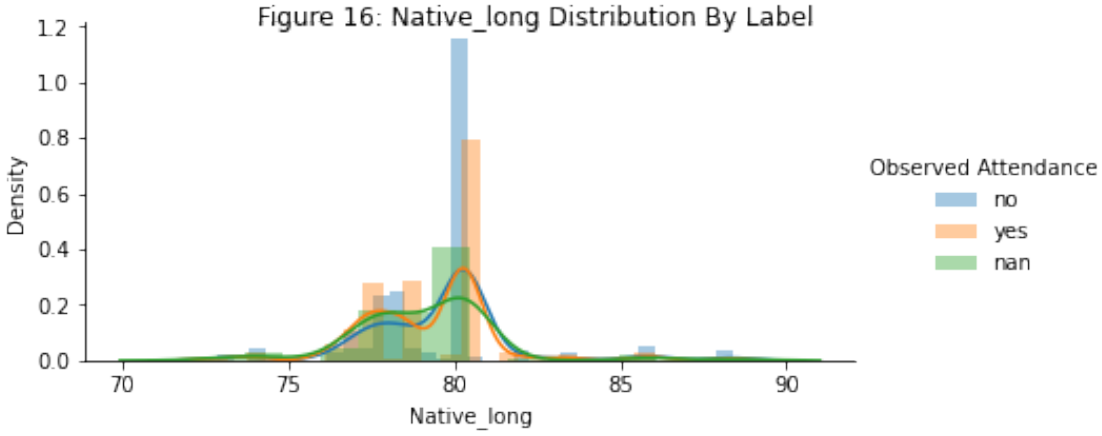


Figure 16: Native_long Distribution By Label
Density
1.2
1.0
0.8
0.6
0.4
0.2
0.0
70
75
80
85
90
Native_long
Observed Attendance
no
yes
nan

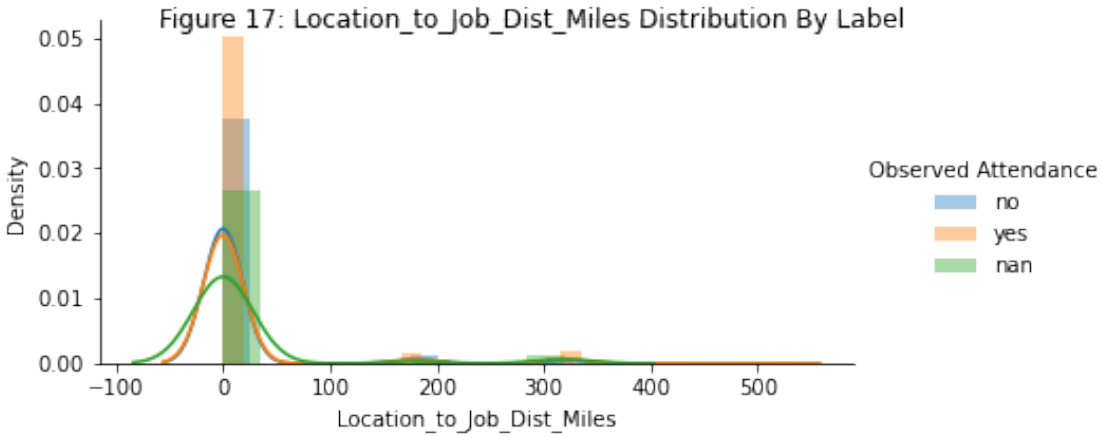


Figure 17: Location_to_Job_Dist_Miles Distribution By Label
Density
0.05
0.04
0.03
0.02
0.01
0.00
−100
0
100
200
300
400
500
Location_to_Job_Dist_Miles
Observed Attendance
no
yes
nan

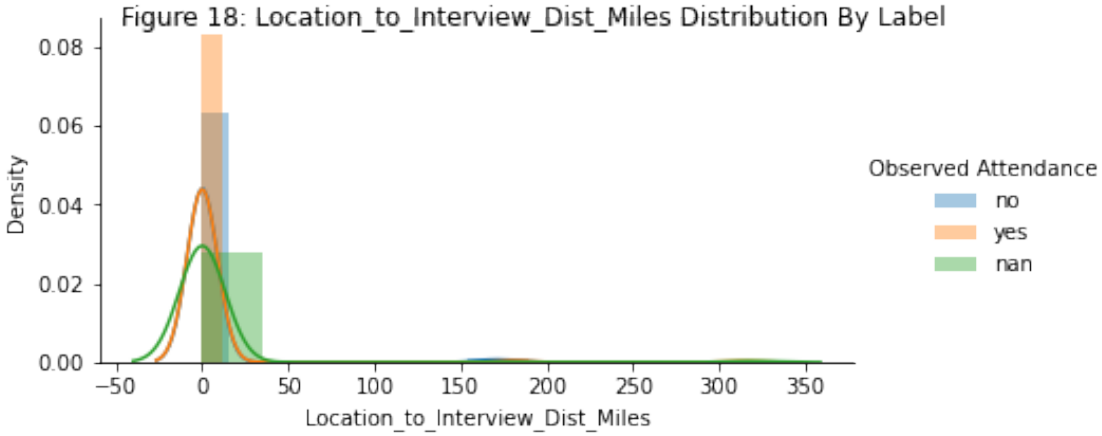


Figure 18: Location_to_Interview_Dist_Miles Distribution By Label
Density
0.08
0.06
0.04
0.02
0.00
−50
0
50
100
150
200
250
300
350
Location_to_Interview_Dist_Miles
Observed Attendance
no
yes
nan

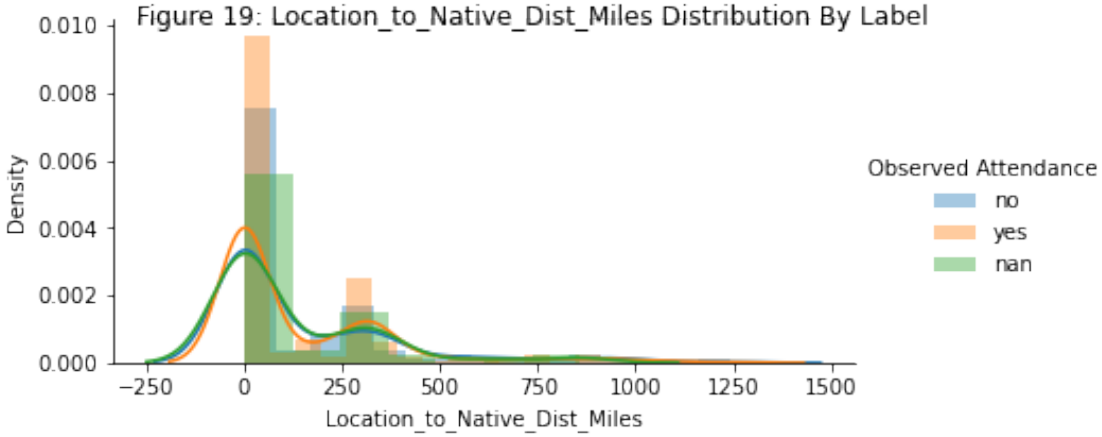


Figure 19: Location_to_Native_Dist_Miles Distribution By Label
Density
0.010
0.008
0.006
0.004
0.002
0.000
−250
0
250
500
750
1000
1250
1500
Location_to_Native_Dist_Miles
Observed Attendance
no
yes
nan

31 Visualization of categorical dataset

[46]:
```
plt.rcParams['figure.max_open_warning']=40
colnames=list(df.select_dtypes(include='O').columns.values)
for i in colnames[0:]:
    ax = plt.axes()
    sns.countplot(x=i, data=df, ax = ax)
    ax.set_title(''.join(map(str, list(["Figure ",colnames.index(i)+20,": ",i," Distribution By Label"]))))
    plt.show()
```

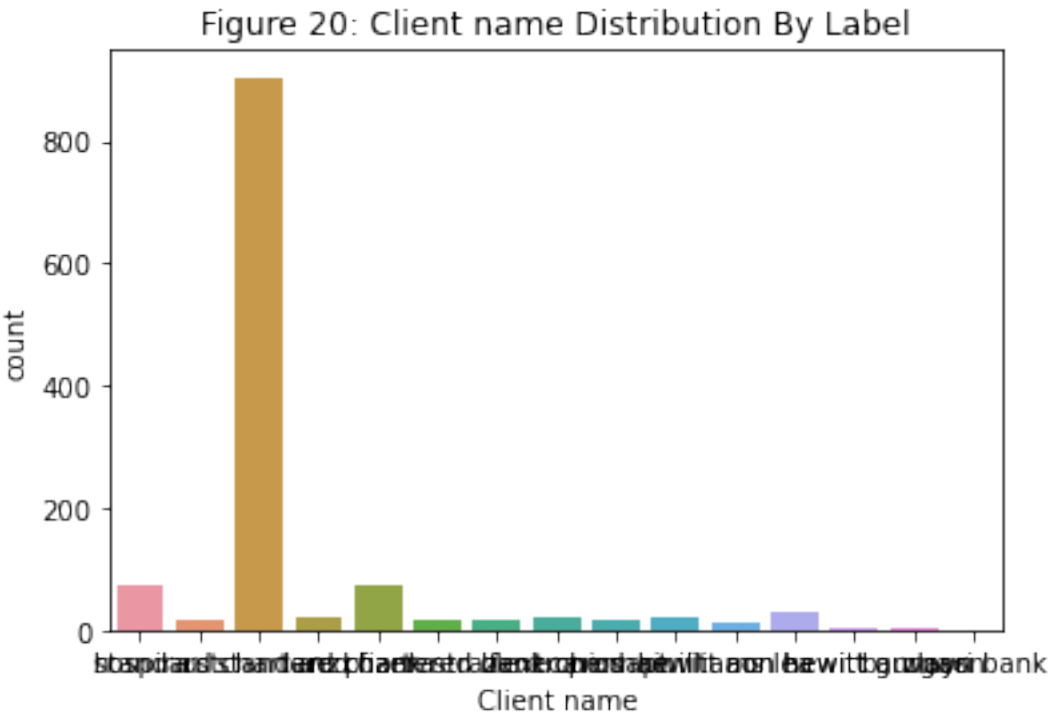


Figure 21: Industry Distribution By Label

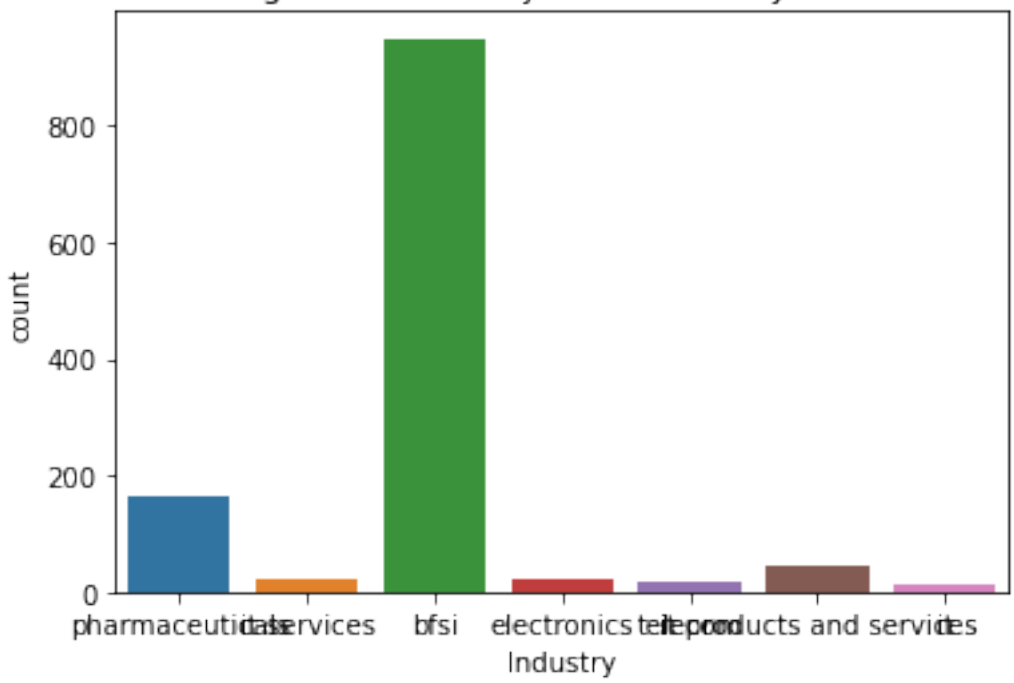


Figure 22: Location Distribution By Label

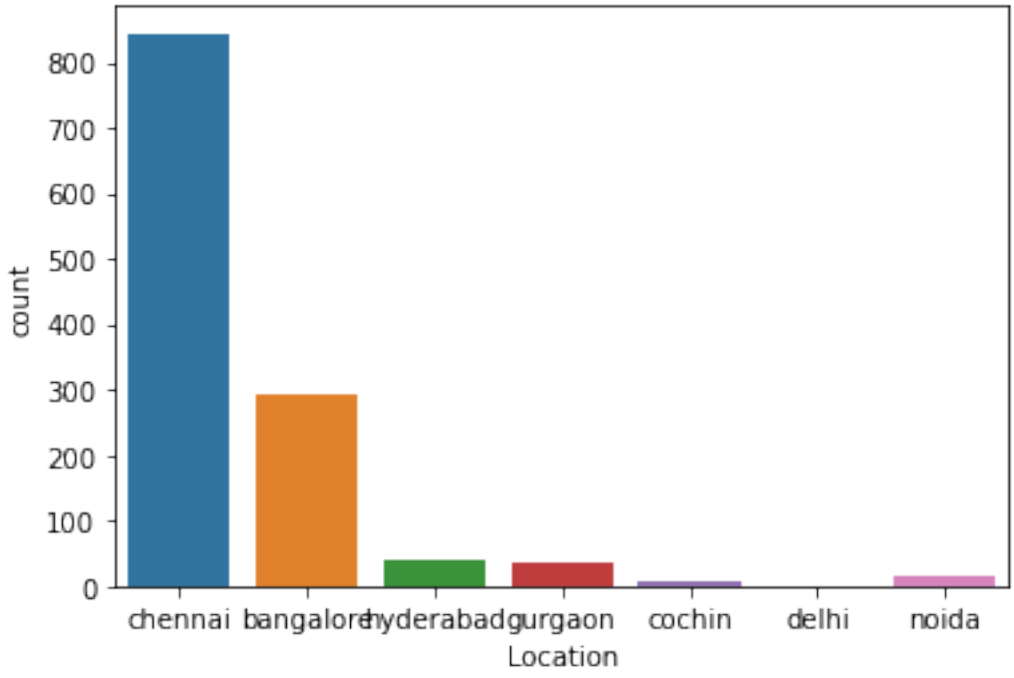


Figure 23: Position to be closed Distribution By Label

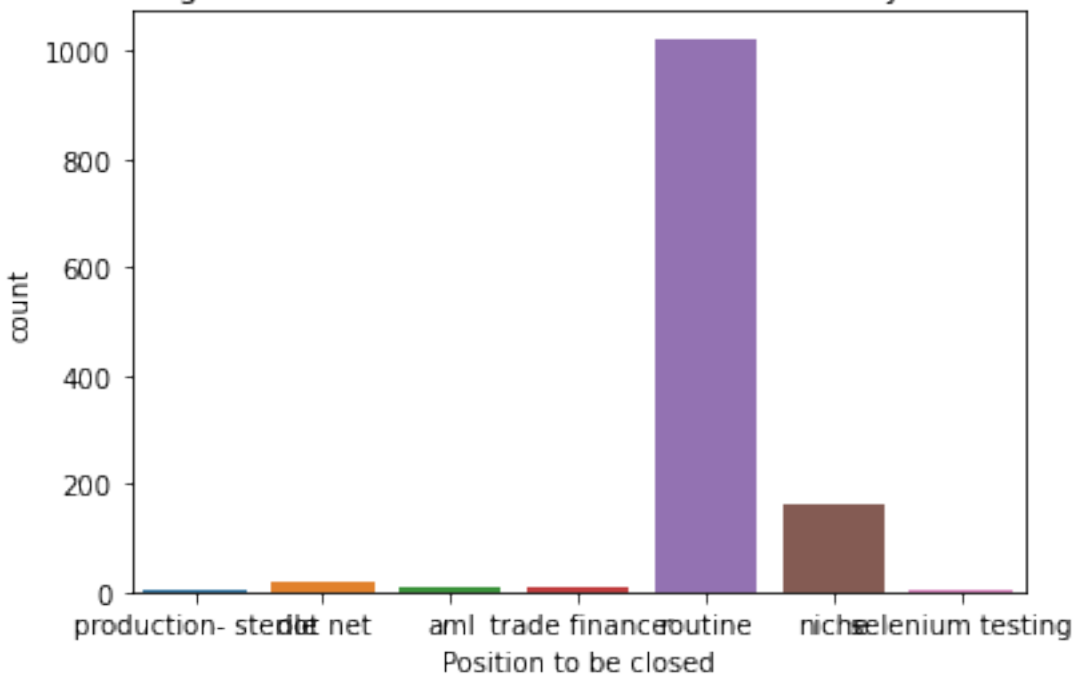


Figure 24: Nature of Skillset Distribution By Label

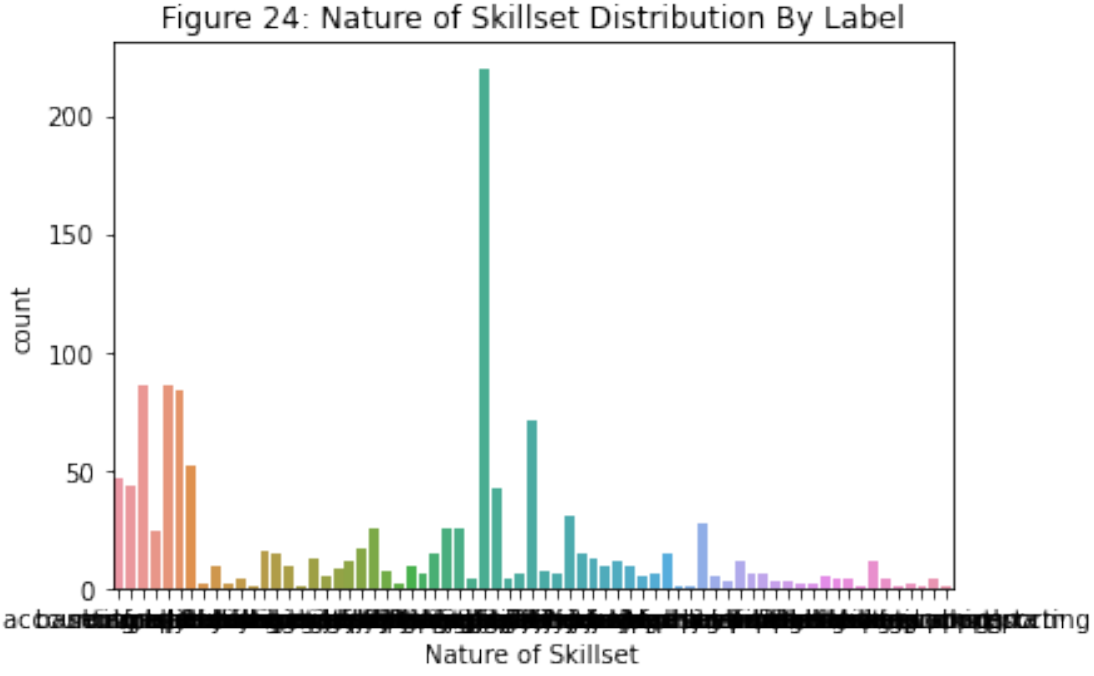


Figure 25: Interview Type Distribution By Label

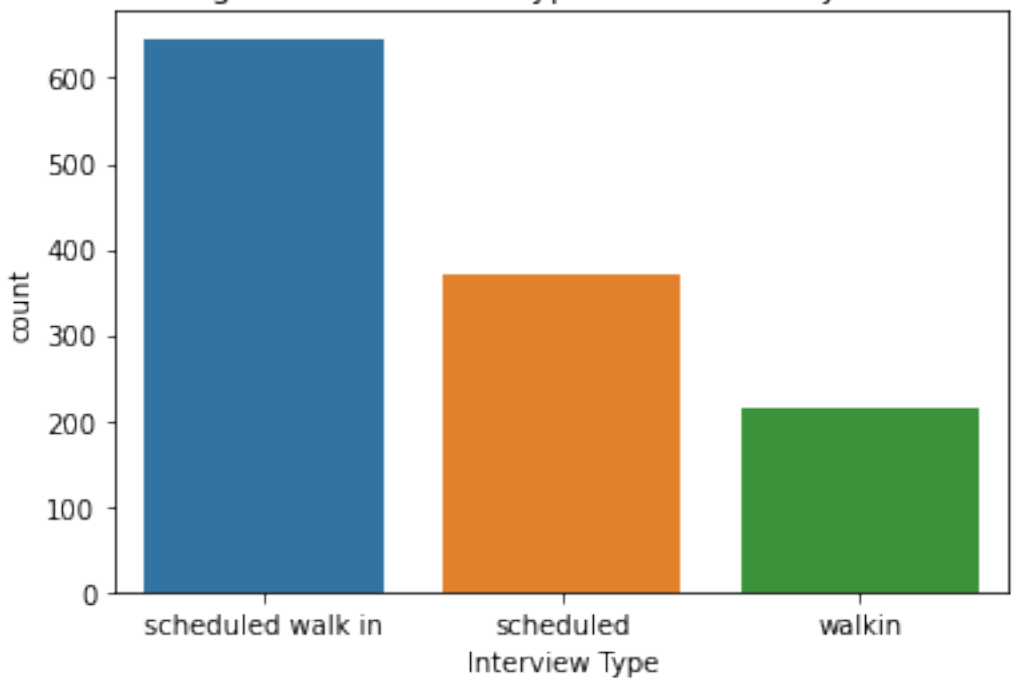


Figure 26: Gender Distribution By Label

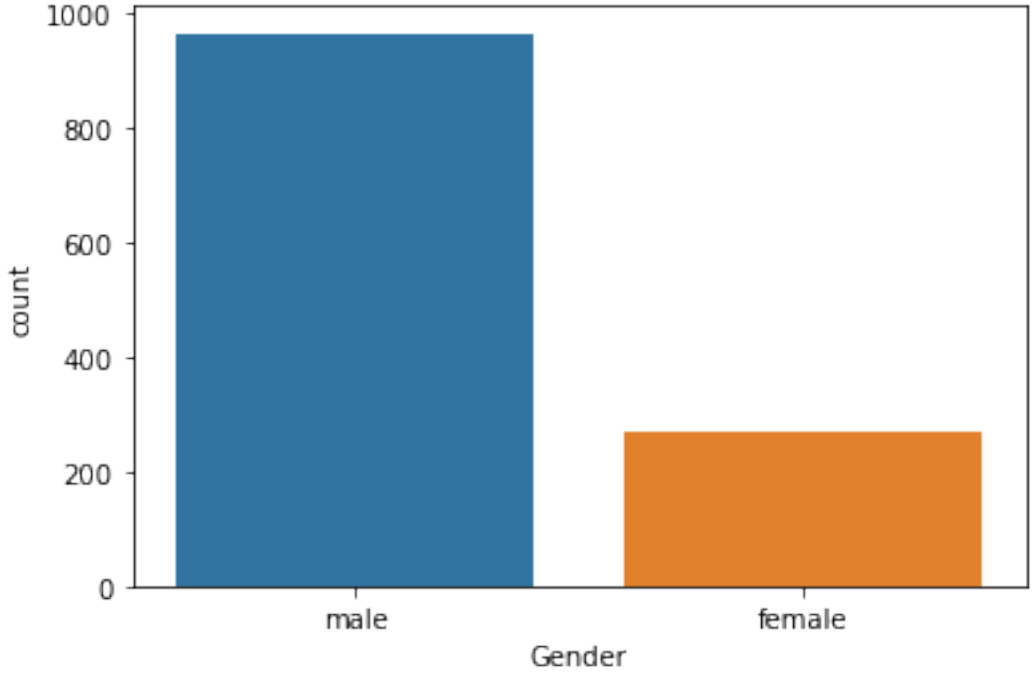


Figure 27: Candidate Current Location Distribution By Label

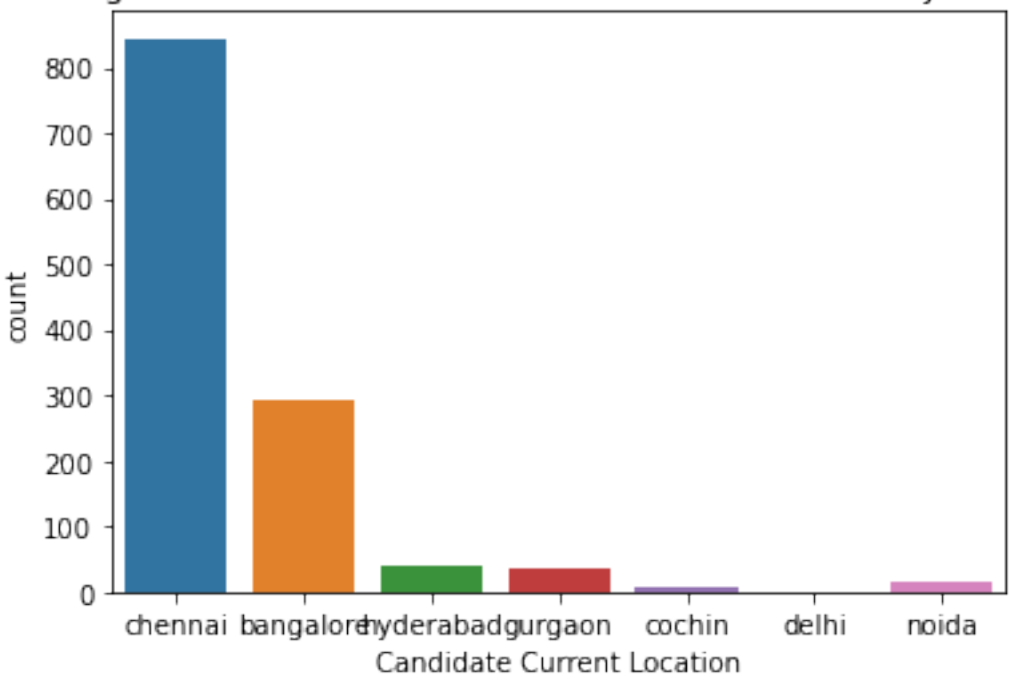


Figure 28: Candidate Job Location Distribution By Label

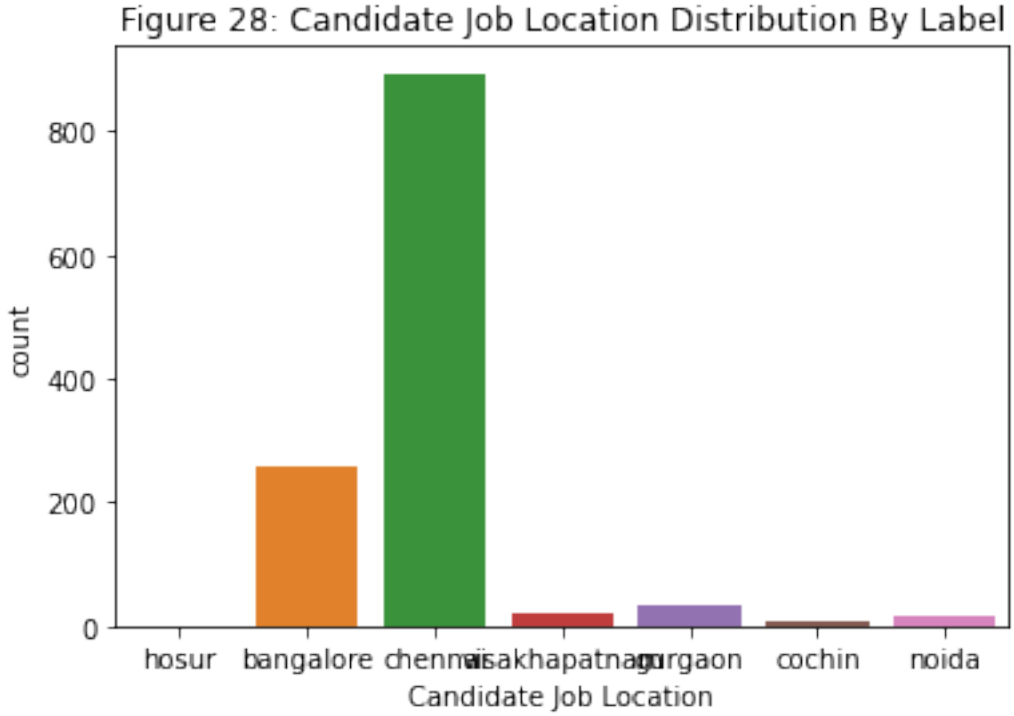


Figure 29: Interview Venue Distribution By Label

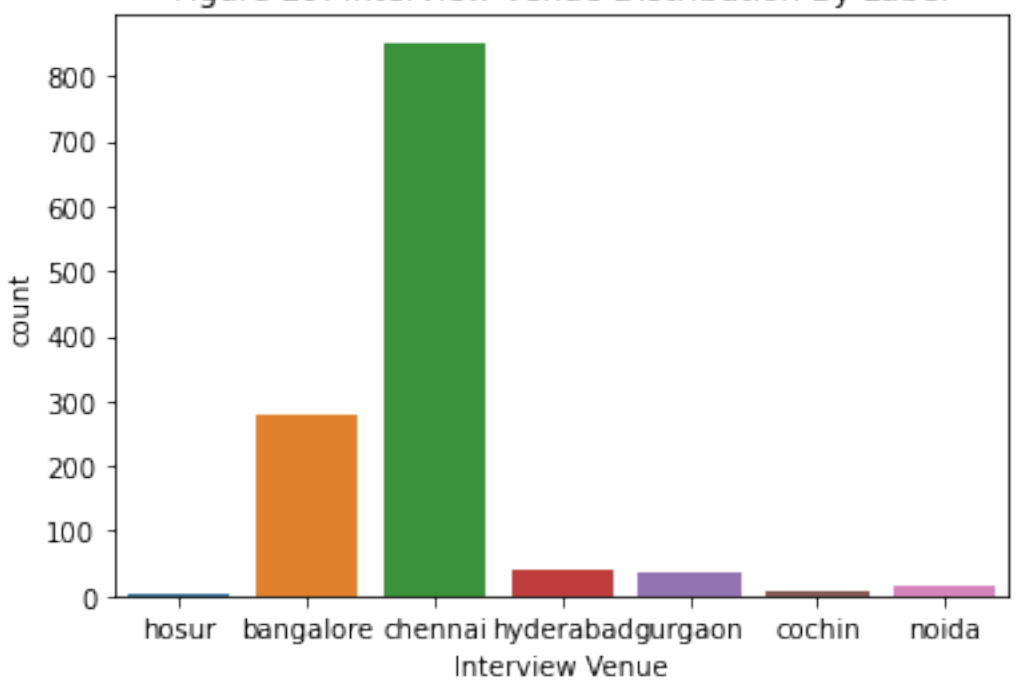


Figure 30: Candidate Native location Distribution By Label

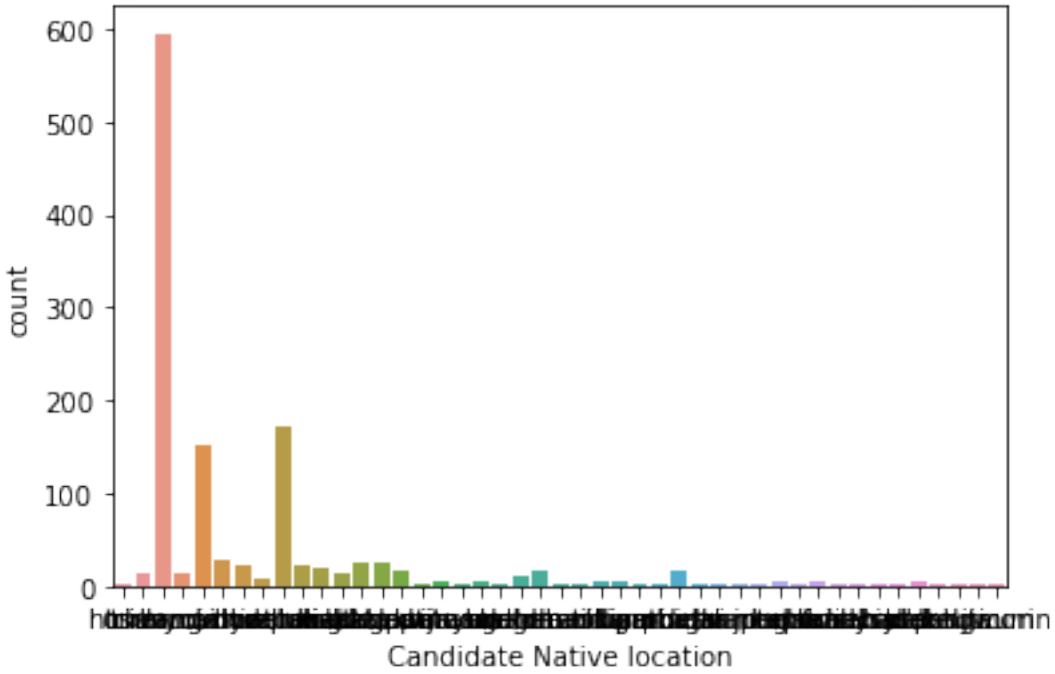


Figure 31: Have you obtained the necessary permission to start at the required time Distribution By Label

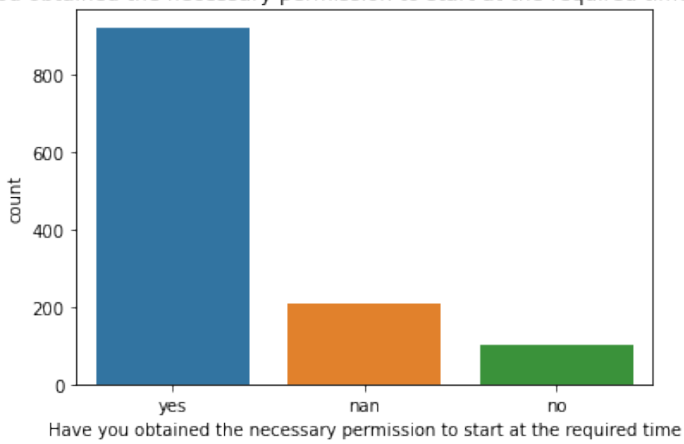


Figure 32: Hope there will be no unscheduled meetings Distribution By Label

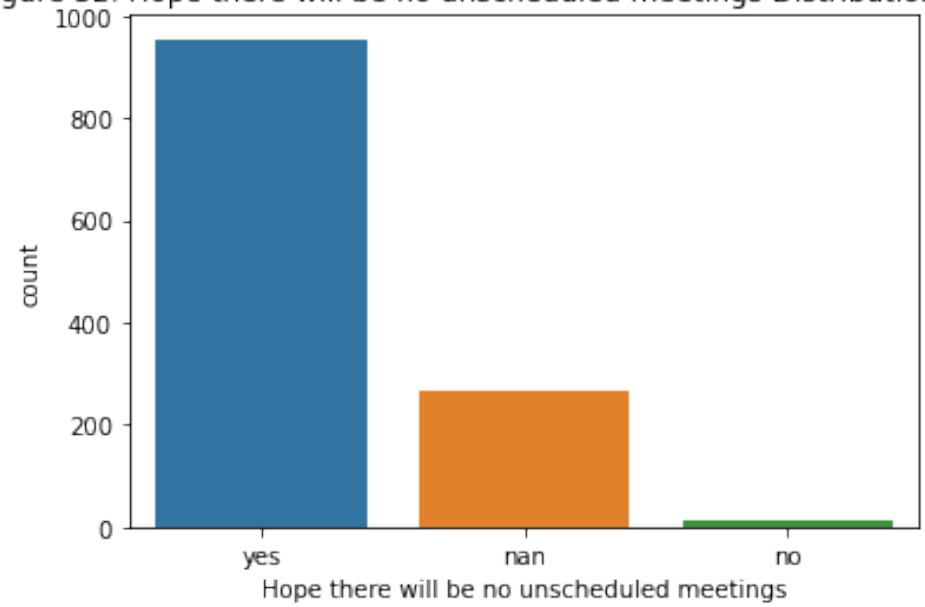


Figure 33: Can I Call you three hours before the interview and follow up on your attendance for the interview Distribution By Label

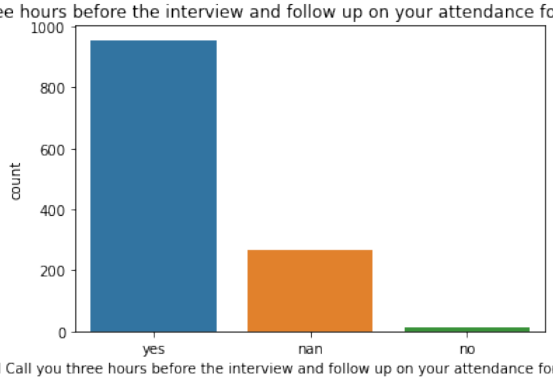


Figure 34: Can I have an alternative number/ desk number. I assure you that I will not trouble you too much Distribution By Label

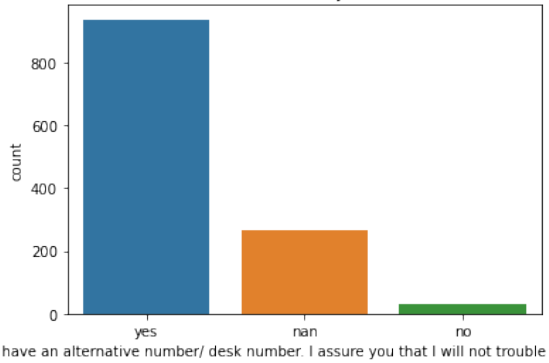


Figure 35: Have you taken a printout of your updated resume. Have you read the JD and understood the same Distribution By Label

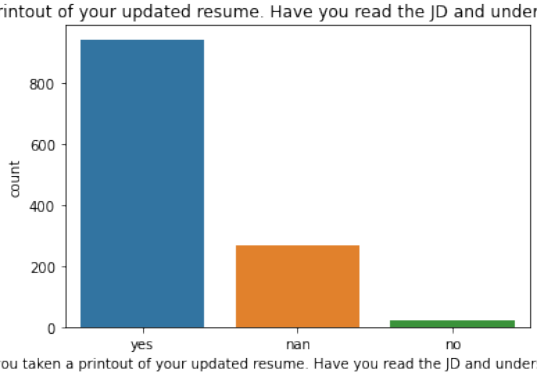


Figure 36: Are you clear with the venue details and the landmark. Distribution By Label

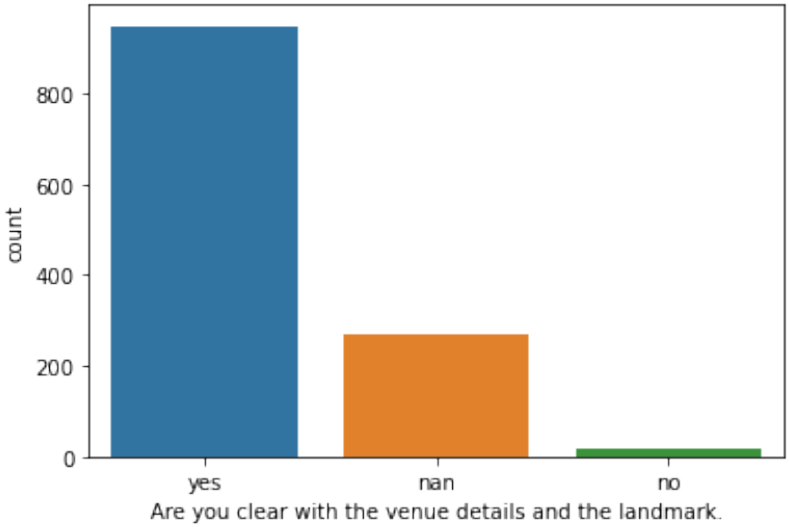


Figure 37: Has the call letter been shared Distribution By Label

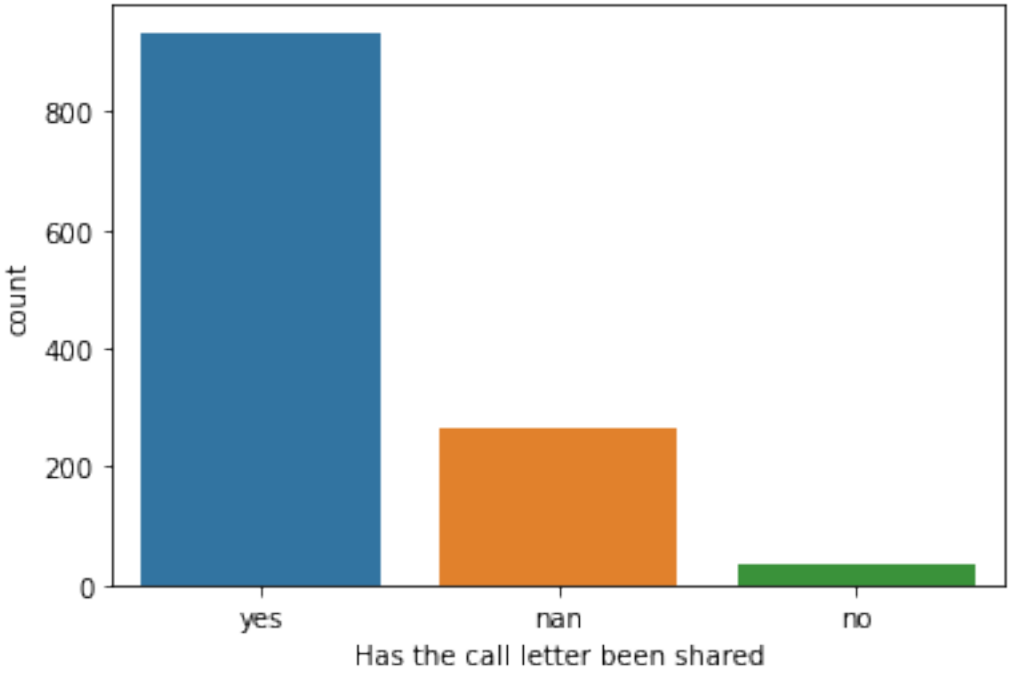


Figure 38: Observed Attendance Distribution By Label

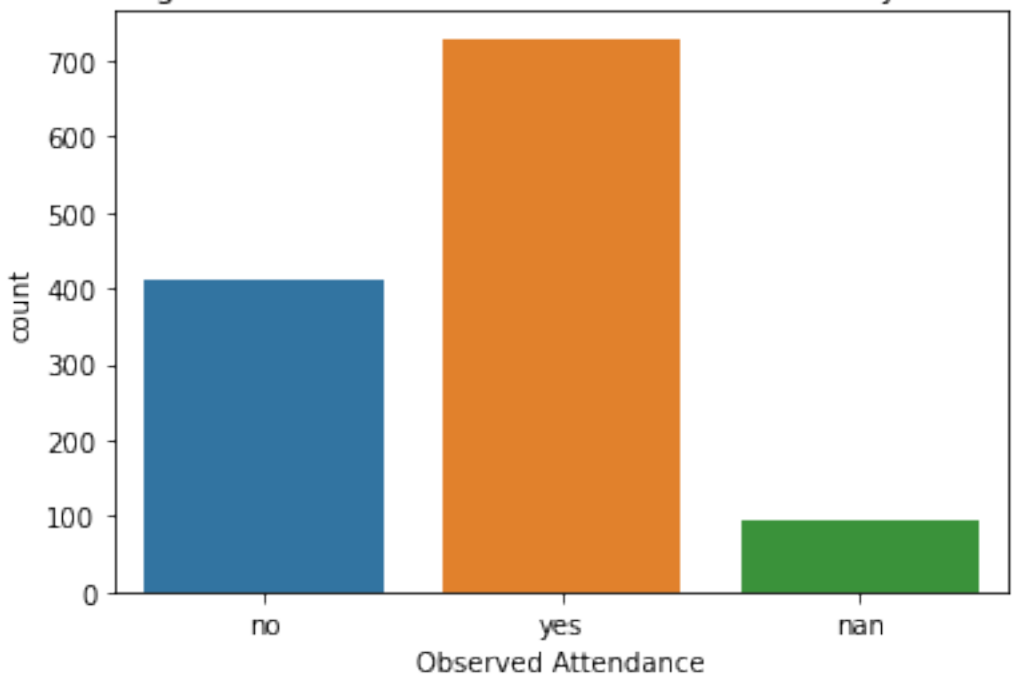
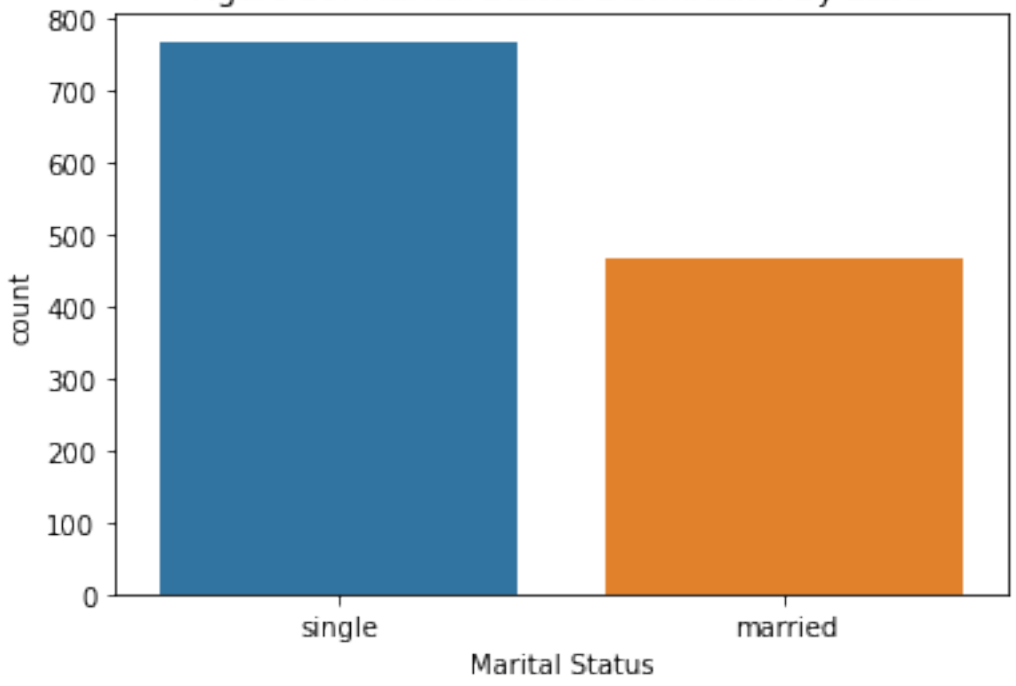


Figure 39: Marital Status Distribution By Label



32 Summary of dataset after clean up

[47]:
```
df.describe(include = 'O')[:4].T
```

[47]:

	count	unique	top
Client name	1233	15	standard chartered bank
Industry	1233	7	bfsi
Location	1233	7	chennai
Position to be closed	1233	7	routine
Nature of Skillset	1233	69	java j2ee struts hibernate
Interview Type	1233	3	scheduled walk in
Gender	1233	2	male
Candidate Current Location	1233	7	chennai
Candidate Job Location	1233	7	chennai
Interview Venue	1233	7	chennai
Candidate Native location	1233	45	chennai
Have you obtained the necessary permission to s…	1233	3	yes
Hope there will be no unscheduled meetings	1233	3	yes
Can I Call you three hours before the interview…	1233	3	yes
Can I have an alternative number/ desk number. …	1233	3	yes
Have you taken a printout of your updated resum…	1233	3	yes
Are you clear with the venue details and the la…	1233	3	yes
Has the call letter been shared	1233	3	yes
Observed Attendance	1233	3	yes
Marital Status	1233	2	

Marital Status	single

	freq
Client name	904
Industry	949
Location	844
Position to be closed	1023
Nature of Skillset	220
Interview Type	646
Gender	965
Candidate Current Location	844
Candidate Job Location	893
Interview Venue	852
Candidate Native location	595
Have you obtained the necessary permission to s…	921
Hope there will be no unscheduled meetings	954
Can I Call you three hours before the interview…	955
Can I have an alternative number/ desk number. …	937
Have you taken a printout of your updated resum…	942
Are you clear with the venue details and the la…	948
Has the call letter been shared	934
Observed Attendance	729
Marital Status	767

[48]:
```
df.describe()[:4].T
```

[48]:

	count	mean	std	min
Name(Cand ID)	1233.0	617.000000	356.080749	1.000000
dayofweek	1233.0	2.773723	1.447083	0.000000
month	1233.0	4.616383	2.228350	1.000000
date	1233.0	13.824006	9.176693	1.000000
year	1233.0	2015.847526	0.521716	2014.000000
Days since earliest date	1233.0	720.894566	212.384377	0.000000
Count of Skillset	1233.0	2.350365	1.120443	1.000000
Current_lat	1233.0	13.791338	3.123024	9.931308
Current_long	1233.0	79.422532	1.267891	76.267414
Job_lat	1233.0	13.740940	3.101452	9.931308
Job_long	1233.0	79.599086	1.321575	76.267414
Interview_lat	1233.0	13.797972	3.125425	9.931308
Interview_long	1233.0	79.442154	1.258375	76.267414
Native_lat	1233.0	14.917734	4.311093	8.188047
Native_long	1233.0	79.348968	2.190943	72.579707
Location_to_Job_Dist_Miles	1233.0	16.476232	67.168704	0.000000
Location_to_Interview_Dist_Miles	1233.0	4.333086	31.973067	0.000000
Location_to_Native_Dist_Miles	1233.0	144.669528	241.568504	0.000000

33 Due to redundancy, remove location and nature of skillset because dummy variables are created for skillset

[49]:
```
df = df.drop(columns = ['Location','Nature of Skillset'])
```

34 For the categorical data, create dummy variables and remove the original column

[50]:
```
colnames=list(df.select_dtypes(include='O').columns.values)
for i in colnames[0:]:
    # Fill missing data with the word "Missing"
    df[i].fillna("Missing", inplace=True)
    # Create array of dummies
    dummies = pd.get_dummies(df[i], prefix=i)
    # Update X to include dummies and drop the main variable
    df = pd.concat([df, dummies], axis=1)
    df.drop([i], axis=1, inplace=True)
```

35 Drop the binary variables that would be colinear to another column, drop the longitudes and latitudes once distance between cities are solved, and drop datetime variables for a relative date

[51]:
```
df = df.drop(columns = ['Date of Interview','Observed Attendance_no','Marital␣
 ↪Status_single','Gender_male','Current_lat','Current_long','Job_lat','Job_long','Interview_l
df
```

[51]:

	Name(Cand ID)	dayofweek	month	date	year	Days since earliest date	\
0	1	4	2	13	2015	332	
1	2	4	2	13	2015	332	
2	3	4	2	13	2015	332	
3	4	4	2	13	2015	332	
4	5	4	2	13	2015	332	
…	…	…	…	…	…	…	
1228	1171	3	6	2	2016	807	
1229	1189	5	1	30	2016	683	
1230	1207	5	1	30	2016	683	
1231	1222	1	7	5	2016	840	
1232	1233	6	6	5	2016	810	

	Count of Skillset	Location_to_Job_Dist_Miles	\
0	1	165.938053	
1	1	180.433771	
2	1	0.000000	

```
3                    1                    0.000000
4                    1                  180.433771
…                  …                      …
1228                 4                    0.000000
1229                 1                    0.000000
1230                 1                    0.000000
1231                 1                    0.000000
1232                 3                    0.000000

      Location_to_Interview_Dist_Miles  Location_to_Native_Dist_Miles  …  \
0                           165.938053                     165.938053  …
1                           165.938053                     190.234720  …
2                           165.938053                       0.000000  …
3                           165.938053                       0.000000  …
4                           165.938053                       0.000000  …
…                                …                              …  …
1228                          0.000000                     318.610407  …
1229                          0.000000                     318.610407  …
1230                          0.000000                     318.610407  …
1231                          0.000000                       0.000000  …
1232                          0.000000                     841.530298  …

      Have you taken a printout of your updated resume. Have you read the JD and
understood the same_yes  \
0                                                    1
1                                                    1
2                                                    0
3                                                    0
4                                                    1
…                                                  …
1228                                                 1
1229                                                 1
1230                                                 1
1231                                                 1
1232                                                 0

      Are you clear with the venue details and the landmark._nan  \
0                                                    0
1                                                    0
2                                                    1
3                                                    0
4                                                    0
…                                                  …
1228                                                 0
1229                                                 0
1230                                                 0
1231                                                 0
```

```
1232                                                    1

      Are you clear with the venue details and the landmark._no  \
0                                                       0
1                                                       0
2                                                       0
3                                                       0
4                                                       0
…                                                     …
1228                                                    0
1229                                                    0
1230                                                    0
1231                                                    0
1232                                                    0

      Are you clear with the venue details and the landmark._yes  \
0                                                       1
1                                                       1
2                                                       0
3                                                       1
4                                                       1
…                                                     …
1228                                                    1
1229                                                    1
1230                                                    1
1231                                                    1
1232                                                    0

      Has the call letter been shared_nan  Has the call letter been shared_no  \
0                                       0                                    0
1                                       0                                    0
2                                       1                                    0
3                                       0                                    0
4                                       0                                    0
…                                     …                                    …
1228                                    0                                    0
1229                                    0                                    0
1230                                    0                                    0
1231                                    0                                    0
1232                                    1                                    0

      Has the call letter been shared_yes  Observed Attendance_nan  \
0                                       1                       0
1                                       1                       0
2                                       0                       0
3                                       1                       0
4                                       1                       0
```

```
…                                    …                          …
1228                                 1                          1
1229                                 1                          1
1230                                 1                          1
1231                                 1                          1
1232                                 0                          1

      Observed Attendance_yes  Marital Status_married
0                           0                       0
1                           0                       0
2                           0                       0
3                           0                       0
4                           0                       1
…                         …                         …
1228                        0                       0
1229                        0                       0
1230                        0                       1
1231                        0                       0
1232                        0                       0

[1233 rows x 133 columns]
```

36 concat the skillset dataframe with the main dataframe

[52]:
```
df = pd.concat([df, Skillset],axis=1)
df
```

[52]:
```
      Name(Cand ID)  dayofweek  month  date  year  Days since earliest date  \
0                 1          4      2    13  2015                       332
1                 2          4      2    13  2015                       332
2                 3          4      2    13  2015                       332
3                 4          4      2    13  2015                       332
4                 5          4      2    13  2015                       332
…               …        …    …   …    …                          …
1228           1171          3      6     2  2016                       807
1229           1189          5      1    30  2016                       683
1230           1207          5      1    30  2016                       683
1231           1222          1      7     5  2016                       840
1232           1233          6      6     5  2016                       810

      Count of Skillset  Location_to_Job_Dist_Miles  \
0                     1                  165.938053
1                     1                  180.433771
2                     1                    0.000000
3                     1                    0.000000
4                     1                  180.433771
```

```
…                  …                          …
1228                   4                   0.000000
1229                   1                   0.000000
1230                   1                   0.000000
1231                   1                   0.000000
1232                   3                   0.000000

      Location_to_Interview_Dist_Miles  Location_to_Native_Dist_Miles  …  \
0                           165.938053                     165.938053  …
1                           165.938053                     190.234720  …
2                           165.938053                       0.000000  …
3                           165.938053                       0.000000  …
4                           165.938053                       0.000000  …
…                                  …                              …  …
1228                          0.000000                     318.610407  …
1229                          0.000000                     318.610407  …
1230                          0.000000                     318.610407  …
1231                          0.000000                       0.000000  …
1232                          0.000000                     841.530298  …

      senior  software  spring  sql  sr  struts  support  t  tech  testing
0          0         0       0    0   0       0        0  0     0        0
1          0         0       0    0   0       0        0  0     0        0
2          0         0       0    0   0       0        0  0     0        0
3          0         0       0    0   0       0        0  0     0        0
4          0         0       0    0   0       0        0  0     0        0
…      …       …     …  …  ..    …      … ..   …      …
1228       0         0       0    0   0       1        0  0     0        0
1229       0         0       0    0   0       0        0  0     0        0
1230       0         0       0    0   0       0        0  0     0        0
1231       0         0       0    0   0       0        0  0     0        0
1232       0         0       0    0   0       0        0  0     0        0

[1233 rows x 189 columns]
```

37 Create a checkpoint of the dataset after clean up towards the modeling of the dataset to load up later when predicting Attendance for null values

[53]:
```
checkpoint2 = df
```

38 Task 1 - Model Development

38.0.1 a. Create a model predicting if a candidate will attend an interview. This will be indicated by the "Observed Attendance" column in the data set. Create the model only using the records where this column is not null. Treat your notebook as if a technical product manager will be reading it and trying to understand your work.

[54]:
```
# df = checkpoint2
```

39 Remove the rows that has NaN's for the Observed Attendance column

[55]:
```
dataset = df[df['Observed Attendance_nan']!=1].reset_index(drop=True).
 ↪drop(columns = ['Observed Attendance_nan'])
dataset
```

[55]:

	Name(Cand ID)	dayofweek	month	date	year	Days since earliest date	\
0	1	4	2	13	2015	332	
1	2	4	2	13	2015	332	
2	3	4	2	13	2015	332	
3	4	4	2	13	2015	332	
4	5	4	2	13	2015	332	
…	…	…	…	…	…	…	
1135	1229	1	7	5	2016	840	
1136	1230	1	7	5	2016	840	
1137	1231	6	6	5	2016	810	
1138	1232	6	6	5	2016	810	
1139	107	2	5	25	2016	799	

	Count of Skillset	Location_to_Job_Dist_Miles	\
0	1	165.938053	
1	1	180.433771	
2	1	0.000000	
3	1	0.000000	
4	1	180.433771	
…	…	…	
1135	3	0.000000	
1136	1	0.000000	
1137	1	0.000000	
1138	3	0.000000	
1139	2	0.000000	

	Location_to_Interview_Dist_Miles	Location_to_Native_Dist_Miles	…	\
0	165.938053	165.938053	…	
1	165.938053	190.234720	…	
2	165.938053	0.000000	…	

```
3                                165.938053                                0.000000  …
4                                165.938053                                0.000000  …
…                                       …                                       …  …
1135                               0.000000                              180.433771  …
1136                               0.000000                              318.610407  …
1137                               0.000000                              318.610407  …
1138                               0.000000                                0.000000  …
1139                               0.000000                              318.610407  …

      senior  software  spring  sql  sr  struts  support  t  tech  testing
0          0         0       0    0   0       0        0  0     0        0
1          0         0       0    0   0       0        0  0     0        0
2          0         0       0    0   0       0        0  0     0        0
3          0         0       0    0   0       0        0  0     0        0
4          0         0       0    0   0       0        0  0     0        0
…      …       …      …   …  ..     …      … ..   …      …
1135       0         0       0    0   0       0        0  0     0        0
1136       0         0       0    0   0       0        0  0     0        0
1137       0         0       0    0   0       0        0  0     0        0
1138       0         0       0    0   0       0        0  0     0        0
1139       0         0       0    0   0       0        0  0     0        0

[1140 rows x 188 columns]
```

40 Separate the target variable from the rest of the dataset

[56]:
```
y = dataset.pop('Observed Attendance_yes')
```

41 Use minmaxscalar to scale all the dataset on a 0 to 1 scale for coefficient comparisons later for important features. Use stratified K fold with a set random state to shuffle the sampling set yet keep the results reproducible with the set random state

[57]:
```
columns = dataset.columns
sc = MinMaxScaler(feature_range=(0, 1))
dataset = sc.fit_transform(dataset)
dataset = pd.DataFrame(dataset, columns=columns)

#x_train, x_test, y_train, y_test = train_test_split(df, y_int, test_size=0.33,␣
 ↪random_state=2, stratify=y_int)
skf = StratifiedKFold(n_splits = 3, shuffle = True, random_state = 2)
```

42 Because the dataset has more attendance than nonattendance, use the balanced function in logistic binary prediction as a baseline for what the results look like based on precision, recall, and F1-score

[58]:
```
model2 = LogisticRegression(class_weight='balanced', random_state=2, n_jobs=-1)

for train_idx, test_idx in skf.split(dataset, y):
    model2.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model2.predict(dataset.
  ↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.53      0.58      0.56       137
           1       0.75      0.71      0.73       243

    accuracy                           0.67       380
   macro avg       0.64      0.65      0.64       380
weighted avg       0.67      0.67      0.67       380

              precision    recall  f1-score   support

           0       0.54      0.55      0.55       137
           1       0.74      0.74      0.74       243

    accuracy                           0.67       380
   macro avg       0.64      0.64      0.64       380
weighted avg       0.67      0.67      0.67       380

              precision    recall  f1-score   support

           0       0.56      0.43      0.49       137
           1       0.72      0.81      0.76       243

    accuracy                           0.67       380
   macro avg       0.64      0.62      0.62       380
weighted avg       0.66      0.67      0.66       380
```

43 Randomforest could also be used as an example, but the feature importance are only positive instead of negative and positive for the results, so logistic regression is used. Random forest in this case is also less accurate from the baseline metrics

[59]:
```
model3 = RandomForestClassifier(n_estimators=100, class_weight='balanced',␣
↪random_state=2, n_jobs=-1)

for train_idx, test_idx in skf.split(dataset, y):
    model3.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model3.predict(dataset.
↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.45      0.42      0.43       137
           1       0.68      0.71      0.70       243

    accuracy                           0.61       380
   macro avg       0.57      0.56      0.56       380
weighted avg       0.60      0.61      0.60       380

              precision    recall  f1-score   support

           0       0.53      0.46      0.49       137
           1       0.72      0.77      0.74       243

    accuracy                           0.66       380
   macro avg       0.62      0.61      0.61       380
weighted avg       0.65      0.66      0.65       380

              precision    recall  f1-score   support

           0       0.47      0.41      0.44       137
           1       0.69      0.74      0.71       243

    accuracy                           0.62       380
   macro avg       0.58      0.57      0.58       380
weighted avg       0.61      0.62      0.61       380
```

44 Logistic regression with cross validation is used to provide baseline metrics before hyperparameter optimization

[60]:
```
model4 = LogisticRegressionCV(class_weight='balanced', random_state=2,␣
↪n_jobs=-1)

for train_idx, test_idx in skf.split(dataset, y):
    model4.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model4.predict(dataset.
↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.59      0.39      0.47       137
           1       0.71      0.84      0.77       243

    accuracy                           0.68       380
   macro avg       0.65      0.62      0.62       380
weighted avg       0.67      0.68      0.66       380

              precision    recall  f1-score   support

           0       0.66      0.43      0.52       137
           1       0.73      0.88      0.80       243

    accuracy                           0.72       380
   macro avg       0.70      0.65      0.66       380
weighted avg       0.71      0.72      0.70       380

              precision    recall  f1-score   support

           0       0.58      0.37      0.45       137
           1       0.71      0.85      0.77       243

    accuracy                           0.68       380
   macro avg       0.64      0.61      0.61       380
weighted avg       0.66      0.68      0.66       380
```

45 Using gridsearch, find the best solver and penalty for the logistic regression with cross validation model by feeding the best estimator back into the loop

[61]:
```
hyperparameters = [{'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag',␣
 ↪'saga']},
              {'penalty':['none', 'elasticnet', 'l1', 'l2']}]

lr = LogisticRegressionCV(class_weight='balanced',n_jobs=-1, random_state=2)

for parameter in hyperparameters:
    search = GridSearchCV(lr, parameter,
                                scoring = 'accuracy',
                                cv = 5,
                                verbose=3)

    search.fit(dataset.loc[train_idx],y.loc[train_idx])
    lr = search.best_estimator_
    print(search.best_estimator_)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END …solver=newton-cg;, score=0.717 total time=   1.0s
[CV 2/5] END …solver=newton-cg;, score=0.717 total time=   0.9s
[CV 3/5] END …solver=newton-cg;, score=0.737 total time=   1.2s
[CV 4/5] END …solver=newton-cg;, score=0.697 total time=   1.1s
[CV 5/5] END …solver=newton-cg;, score=0.678 total time=   1.1s
[CV 1/5] END …solver=lbfgs;, score=0.717 total time=   0.5s
[CV 2/5] END …solver=lbfgs;, score=0.717 total time=   0.6s
[CV 3/5] END …solver=lbfgs;, score=0.737 total time=   0.6s
[CV 4/5] END …solver=lbfgs;, score=0.697 total time=   0.6s
[CV 5/5] END …solver=lbfgs;, score=0.678 total time=   0.5s
[CV 1/5] END …solver=liblinear;, score=0.704 total time=   0.1s
[CV 2/5] END …solver=liblinear;, score=0.697 total time=   0.1s
[CV 3/5] END …solver=liblinear;, score=0.737 total time=   0.1s
[CV 4/5] END …solver=liblinear;, score=0.691 total time=   0.1s
[CV 5/5] END …solver=liblinear;, score=0.684 total time=   0.1s
[CV 1/5] END …solver=sag;, score=0.717 total time=   1.2s
[CV 2/5] END …solver=sag;, score=0.717 total time=   1.2s
[CV 3/5] END …solver=sag;, score=0.737 total time=   1.2s
[CV 4/5] END …solver=sag;, score=0.697 total time=   1.3s
[CV 5/5] END …solver=sag;, score=0.678 total time=   1.3s
[CV 1/5] END …solver=saga;, score=0.717 total time=   1.5s
[CV 2/5] END …solver=saga;, score=0.717 total time=   1.5s
[CV 3/5] END …solver=saga;, score=0.737 total time=   1.5s
[CV 4/5] END …solver=saga;, score=0.697 total time=   1.5s
[CV 5/5] END …solver=saga;, score=0.678 total time=   1.5s
LogisticRegressionCV(class_weight='balanced', n_jobs=-1, random_state=2,
```

```
                   solver='newton-cg')
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END …penalty=none;, score=nan total time=   0.0s
[CV 2/5] END …penalty=none;, score=nan total time=   0.0s
[CV 3/5] END …penalty=none;, score=nan total time=   0.0s
[CV 4/5] END …penalty=none;, score=nan total time=   0.0s
[CV 5/5] END …penalty=none;, score=nan total time=   0.0s
[CV 1/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 2/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 3/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 4/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 5/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 2/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 3/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 4/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 5/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l2;, score=0.717 total time=   1.0s
[CV 2/5] END …penalty=l2;, score=0.717 total time=   1.1s
[CV 3/5] END …penalty=l2;, score=0.737 total time=   1.7s
[CV 4/5] END …penalty=l2;, score=0.697 total time=   1.2s
[CV 5/5] END …penalty=l2;, score=0.678 total time=   1.2s
LogisticRegressionCV(class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='newton-cg')
```

46 For logistic regression with CV, L2 and newton-cg are the best hyperparameters for better accuracy for the metrics provided

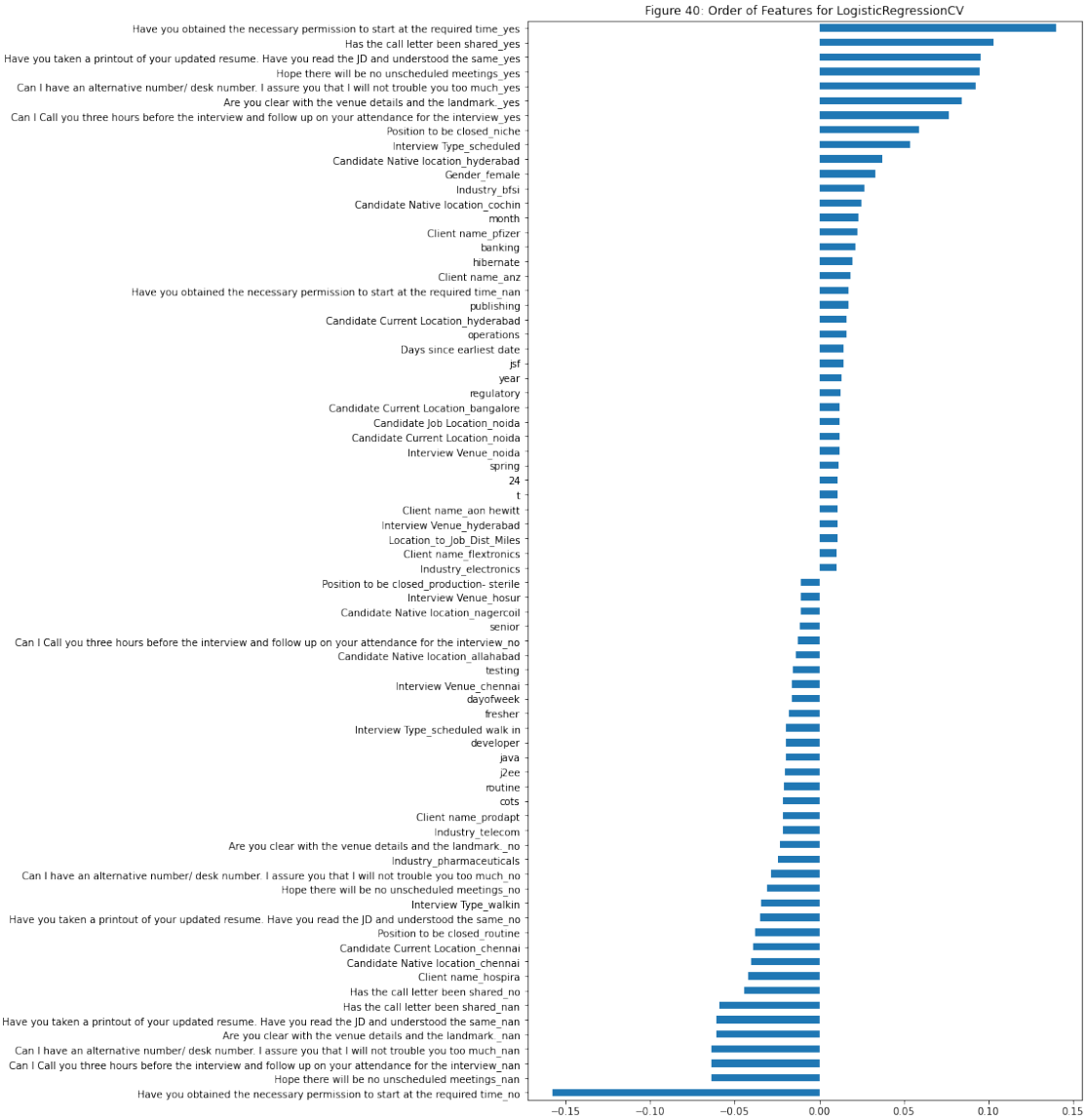
[62]:
```
model4 = search.best_estimator_

for train_idx, test_idx in skf.split(dataset, y):
    model4.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model4.predict(dataset.
 ↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.59      0.39      0.47       137
           1       0.71      0.84      0.77       243

    accuracy                           0.68       380
   macro avg       0.65      0.62      0.62       380
weighted avg       0.67      0.68      0.66       380

              precision    recall  f1-score   support

           0       0.66      0.43      0.52       137
```

```
              1       0.73      0.88      0.80       243

    accuracy                           0.72       380
   macro avg       0.70      0.65      0.66       380
weighted avg       0.71      0.72      0.70       380

              precision    recall  f1-score   support

           0       0.58      0.37      0.45       137
           1       0.71      0.85      0.77       243

    accuracy                           0.68       380
   macro avg       0.64      0.61      0.61       380
weighted avg       0.66      0.68      0.66       380
```

47 for the logistic regression CV model, here are the positive and negative features that would influence a candidate to show up to an interview, further explanations later

[63]:
```
feature_importances = pd.Series(model4.coef_[0], index=dataset.columns)
feature_importances[abs(feature_importances)>0.01].sort_values().
 ↪plot(kind="barh", figsize=(10,20),
                                        title = "Figure 40: Order of Features␣
 ↪for " + str(model4).split("(")[0]);
```



48 save the important features for a later subset of the main dataset for less overfitting of a later model

[64]:
```
features1 = pd.DataFrame(feature_importances[abs(feature_importances)>0.01])
features1
```

[64]:

	0
dayofweek	-0.016113
month	0.023317
year	0.012969
Days since earliest date	0.014221

```
Location_to_Job_Dist_Miles  0.010771
…                                …
routine                    -0.021145
senior                     -0.011523
spring                      0.011426
t                           0.011075
testing                    -0.015546

[74 rows x 1 columns]
```

49 Logistic regression with the most common parameters is used to provide baseline metrics before hyperparameter optimization

[65]:
```
model5 = LogisticRegression(class_weight='balanced',penalty='l1', C=0.5,␣
↪max_iter=100, solver='liblinear',random_state=2, n_jobs=-1)

for train_idx, test_idx in skf.split(dataset, y):
    model5.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model5.predict(dataset.
↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.54      0.55      0.55       137
           1       0.75      0.74      0.74       243

    accuracy                           0.67       380
   macro avg       0.64      0.65      0.64       380
weighted avg       0.67      0.67      0.67       380

              precision    recall  f1-score   support

           0       0.52      0.50      0.51       137
           1       0.72      0.74      0.73       243

    accuracy                           0.66       380
   macro avg       0.62      0.62      0.62       380
weighted avg       0.65      0.66      0.65       380

              precision    recall  f1-score   support

           0       0.58      0.45      0.51       137
           1       0.73      0.81      0.77       243

    accuracy                           0.68       380
```

```
   macro avg       0.65      0.63      0.64       380
weighted avg       0.67      0.68      0.67       380
```

50 Using gridsearch, find the best solver and penalty for the logistic regression model by feeding the best estimator back into the loop

[66]:
```
hyperparameters = [{'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag',␣
 ↪'saga']},
                   {'C':[0.001, 0.01, 0.1, 1, 10, 100]},
              {'penalty':['none', 'elasticnet', 'l1', 'l2']}]

lr = LogisticRegression(class_weight='balanced',n_jobs=-1, random_state=2)

for parameter in hyperparameters:
    search = GridSearchCV(lr, parameter,
                                scoring = 'accuracy',
                                cv = 5,
                                verbose=3)

    search.fit(dataset.loc[train_idx],y.loc[train_idx])
    lr = search.best_estimator_
    print(search.best_estimator_)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END …solver=newton-cg;, score=0.711 total time=   0.0s
[CV 2/5] END …solver=newton-cg;, score=0.618 total time=   0.0s
[CV 3/5] END …solver=newton-cg;, score=0.592 total time=   0.0s
[CV 4/5] END …solver=newton-cg;, score=0.572 total time=   0.0s
[CV 5/5] END …solver=newton-cg;, score=0.684 total time=   0.0s
[CV 1/5] END …solver=lbfgs;, score=0.711 total time=   0.0s
[CV 2/5] END …solver=lbfgs;, score=0.618 total time=   0.0s
[CV 3/5] END …solver=lbfgs;, score=0.592 total time=   0.0s
[CV 4/5] END …solver=lbfgs;, score=0.572 total time=   0.0s
[CV 5/5] END …solver=lbfgs;, score=0.684 total time=   0.0s
[CV 1/5] END …solver=liblinear;, score=0.711 total time=   0.0s
[CV 2/5] END …solver=liblinear;, score=0.618 total time=   0.0s
[CV 3/5] END …solver=liblinear;, score=0.592 total time=   0.0s
[CV 4/5] END …solver=liblinear;, score=0.579 total time=   0.0s
[CV 5/5] END …solver=liblinear;, score=0.684 total time=   0.0s
[CV 1/5] END …solver=sag;, score=0.711 total time=   0.1s
[CV 2/5] END …solver=sag;, score=0.618 total time=   0.1s
[CV 3/5] END …solver=sag;, score=0.592 total time=   0.2s
[CV 4/5] END …solver=sag;, score=0.572 total time=   0.2s
[CV 5/5] END …solver=sag;, score=0.684 total time=   0.2s
[CV 1/5] END …solver=saga;, score=0.711 total time=   0.2s
```

```
[CV 2/5] END …solver=saga;, score=0.618 total time=   0.2s
[CV 3/5] END …solver=saga;, score=0.592 total time=   0.2s
[CV 4/5] END …solver=saga;, score=0.579 total time=   0.2s
[CV 5/5] END …solver=saga;, score=0.684 total time=   0.2s
LogisticRegression(class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='liblinear')
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[CV 1/5] END …C=0.001;, score=0.671 total time=   0.0s
[CV 2/5] END …C=0.001;, score=0.697 total time=   0.0s
[CV 3/5] END …C=0.001;, score=0.737 total time=   0.0s
[CV 4/5] END …C=0.001;, score=0.678 total time=   0.0s
[CV 5/5] END …C=0.001;, score=0.678 total time=   0.0s
[CV 1/5] END …C=0.01;, score=0.691 total time=   0.0s
[CV 2/5] END …C=0.01;, score=0.697 total time=   0.0s
[CV 3/5] END …C=0.01;, score=0.730 total time=   0.0s
[CV 4/5] END …C=0.01;, score=0.684 total time=   0.0s
[CV 5/5] END …C=0.01;, score=0.684 total time=   0.0s
[CV 1/5] END …C=0.1;, score=0.724 total time=   0.0s
[CV 2/5] END …C=0.1;, score=0.664 total time=   0.0s
[CV 3/5] END …C=0.1;, score=0.592 total time=   0.0s
[CV 4/5] END …C=0.1;, score=0.632 total time=   0.0s
[CV 5/5] END …C=0.1;, score=0.691 total time=   0.0s
[CV 1/5] END …C=1;, score=0.711 total time=   0.0s
[CV 2/5] END …C=1;, score=0.618 total time=   0.0s
[CV 3/5] END …C=1;, score=0.592 total time=   0.0s
[CV 4/5] END …C=1;, score=0.579 total time=   0.0s
[CV 5/5] END …C=1;, score=0.684 total time=   0.0s
[CV 1/5] END …C=10;, score=0.711 total time=   0.0s
[CV 2/5] END …C=10;, score=0.645 total time=   0.0s
[CV 3/5] END …C=10;, score=0.586 total time=   0.0s
[CV 4/5] END …C=10;, score=0.500 total time=   0.0s
[CV 5/5] END …C=10;, score=0.691 total time=   0.0s
[CV 1/5] END …C=100;, score=0.704 total time=   0.0s
[CV 2/5] END …C=100;, score=0.605 total time=   0.0s
[CV 3/5] END …C=100;, score=0.599 total time=   0.0s
[CV 4/5] END …C=100;, score=0.500 total time=   0.0s
[CV 5/5] END …C=100;, score=0.539 total time=   0.0s
LogisticRegression(C=0.01, class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='liblinear')
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END …penalty=none;, score=nan total time=   0.0s
[CV 2/5] END …penalty=none;, score=nan total time=   0.0s
[CV 3/5] END …penalty=none;, score=nan total time=   0.0s
[CV 4/5] END …penalty=none;, score=nan total time=   0.0s
[CV 5/5] END …penalty=none;, score=nan total time=   0.0s
[CV 1/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 2/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 3/5] END …penalty=elasticnet;, score=nan total time=   0.0s
```

```
[CV 4/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 5/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l1;, score=0.362 total time=   0.0s
[CV 2/5] END …penalty=l1;, score=0.362 total time=   0.0s
[CV 3/5] END …penalty=l1;, score=0.362 total time=   0.0s
[CV 4/5] END …penalty=l1;, score=0.362 total time=   0.0s
[CV 5/5] END …penalty=l1;, score=0.355 total time=   0.0s
[CV 1/5] END …penalty=l2;, score=0.691 total time=   0.0s
[CV 2/5] END …penalty=l2;, score=0.697 total time=   0.0s
[CV 3/5] END …penalty=l2;, score=0.730 total time=   0.0s
[CV 4/5] END …penalty=l2;, score=0.684 total time=   0.0s
[CV 5/5] END …penalty=l2;, score=0.684 total time=   0.0s
LogisticRegression(C=0.01, class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='liblinear')
```

51 For logistic regression, C=0.01, L2, and liblinear are the best hyperparameters for better accuracy for the metrics provided

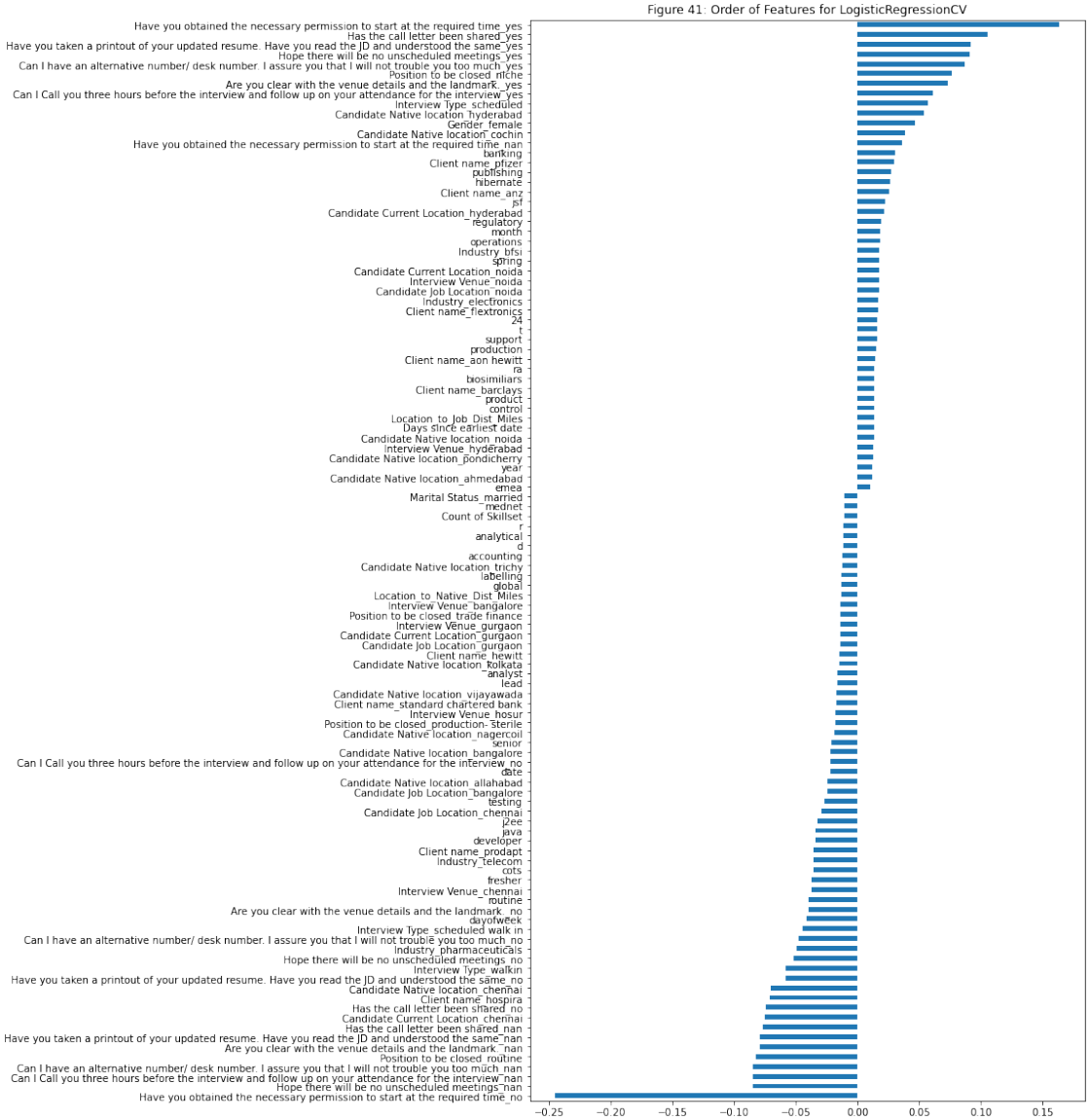
[67]:
```
model5 = search.best_estimator_

for train_idx, test_idx in skf.split(dataset, y):
    model5.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model5.predict(dataset.
 ↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.61      0.49      0.54       137
           1       0.74      0.82      0.78       243

    accuracy                           0.70       380
   macro avg       0.67      0.66      0.66       380
weighted avg       0.69      0.70      0.69       380

              precision    recall  f1-score   support

           0       0.61      0.50      0.55       137
           1       0.74      0.82      0.78       243

    accuracy                           0.71       380
   macro avg       0.68      0.66      0.66       380
weighted avg       0.70      0.71      0.70       380

              precision    recall  f1-score   support

           0       0.57      0.36      0.44       137
           1       0.70      0.84      0.77       243
```

```
    accuracy                           0.67       380
   macro avg       0.64      0.60      0.61       380
weighted avg       0.65      0.67      0.65       380
```

52 for the logistic regression model, here are the positive and negative features that would influence a candidate to show up to an interview, further explanations later

[68]:
```
feature_importances = pd.Series(model5.coef_[0], index=dataset.columns)
feature_importances[abs(feature_importances)>0.01].sort_values().
 ↪plot(kind="barh", figsize=(10,20),
                                        title = "Figure 41: Order of Features␣
 ↪for " + str(model4).split("(")[0]);
```



53 save the important features for a later subset of the main dataset for less overfitting of a later model

[69]:
```
features2 = pd.DataFrame(feature_importances[abs(feature_importances)>0.01])
features2
```

[69]:

	0
dayofweek	-0.041428
month	0.018515
date	-0.022098
year	0.011767

```
Days since earliest date  0.013491
…                            …
senior                   -0.021201
spring                    0.017529
support                   0.015665
t                         0.016356
testing                  -0.026776

[110 rows x 1 columns]
```

54 Given that logistic regression and logistic regression with CV are similar with the same model, the important features are combined to form a subset of the main dataset for further iterative fitting

[70]:
```
keep = pd.DataFrame(pd.concat([features1,features2],axis=1).fillna(0).T.mean()).
↪sort_values(0)
keep
```

[70]:
```
                                                          0
Have you obtained the necessary permission to s… -0.201503
Can I have an alternative number/ desk number. … -0.074429
Hope there will be no unscheduled meetings_nan     -0.074429
Can I Call you three hours before the interview… -0.074429
Are you clear with the venue details and the la… -0.070167
…                                                       …
Can I have an alternative number/ desk number. …  0.089728
Hope there will be no unscheduled meetings_yes      0.092643
Have you taken a printout of your updated resum…  0.093737
Has the call letter been shared_yes                 0.104489
Have you obtained the necessary permission to s…  0.151906

[111 rows x 1 columns]
```

55 from 188 columns, the dataset is reduced to 111 columns after removing nonessential features

[71]:
```
dataset = dataset[keep.index]
dataset
```

[71]:
```
      Have you obtained the necessary permission to start at the required
time_no  \
0                                                     0.0
1                                                     0.0
```

```
2                                                    0.0
3                                                    0.0
4                                                    0.0
…                                                    …
1135                                                 0.0
1136                                                 0.0
1137                                                 0.0
1138                                                 0.0
1139                                                 0.0

      Can I have an alternative number/ desk number. I assure you that I will
not trouble you too much_nan  \
0                                                    0.0
1                                                    0.0
2                                                    1.0
3                                                    0.0
4                                                    0.0
…                                                    …
1135                                                 0.0
1136                                                 0.0
1137                                                 0.0
1138                                                 0.0
1139                                                 0.0

      Hope there will be no unscheduled meetings_nan  \
0                                                  0.0
1                                                  0.0
2                                                  1.0
3                                                  0.0
4                                                  0.0
…                                                  …
1135                                               0.0
1136                                               0.0
1137                                               0.0
1138                                               0.0
1139                                               0.0

      Can I Call you three hours before the interview and follow up on your
attendance for the interview_nan  \
0                                                    0.0
1                                                    0.0
2                                                    1.0
3                                                    0.0
4                                                    0.0
…                                                    …
1135                                                 0.0
1136                                                 0.0
```

```
1137                                                 0.0
1138                                                 0.0
1139                                                 0.0

      Are you clear with the venue details and the landmark._nan  \
0                                                    0.0
1                                                    0.0
2                                                    1.0
3                                                    0.0
4                                                    0.0
…                                                    …
1135                                                 0.0
1136                                                 0.0
1137                                                 0.0
1138                                                 0.0
1139                                                 0.0

      Have you taken a printout of your updated resume. Have you read the JD and
understood the same_nan  \
0                                                    0.0
1                                                    0.0
2                                                    1.0
3                                                    0.0
4                                                    0.0
…                                                    …
1135                                                 0.0
1136                                                 0.0
1137                                                 0.0
1138                                                 0.0
1139                                                 0.0

      Has the call letter been shared_nan  Position to be closed_routine  \
0                                     0.0                            0.0
1                                     0.0                            0.0
2                                     1.0                            0.0
3                                     0.0                            0.0
4                                     0.0                            0.0
…                                     …                              …
1135                                  0.0                            0.0
1136                                  0.0                            0.0
1137                                  0.0                            0.0
1138                                  0.0                            0.0
1139                                  0.0                            1.0

      Has the call letter been shared_no  Candidate Current Location_chennai  \
0                                    0.0                                 1.0
1                                    0.0                                 1.0
```

```
2                                    0.0                                    1.0
3                                    0.0                                    1.0
4                                    0.0                                    1.0
…                                    …                                      …
1135                                 0.0                                    1.0
1136                                 0.0                                    1.0
1137                                 0.0                                    1.0
1138                                 0.0                                    1.0
1139                                 0.0                                    1.0

      …  Candidate Native location_hyderabad  Interview Type_scheduled  \
0     …                                  0.0                       0.0
1     …                                  0.0                       0.0
2     …                                  0.0                       0.0
3     …                                  0.0                       0.0
4     …                                  0.0                       0.0
…   …                                    …                         …
1135  …                                  0.0                       1.0
1136  …                                  1.0                       1.0
1137  …                                  1.0                       1.0
1138  …                                  0.0                       1.0
1139  …                                  1.0                       0.0

      Position to be closed_niche  \
0                             0.0
1                             0.0
2                             0.0
3                             0.0
4                             0.0
…                             …
1135                          1.0
1136                          1.0
1137                          1.0
1138                          1.0
1139                          0.0

      Can I Call you three hours before the interview and follow up on your
attendance for the interview_yes  \
0                                                    1.0
1                                                    1.0
2                                                    0.0
3                                                    0.0
4                                                    1.0
…                                                    …
1135                                                 1.0
1136                                                 1.0
1137                                                 1.0
```

```
1138                                                  1.0
1139                                                  1.0

      Are you clear with the venue details and the landmark._yes  \
0                                                     1.0
1                                                     1.0
2                                                     0.0
3                                                     1.0
4                                                     1.0
…                                                     …
1135                                                  1.0
1136                                                  1.0
1137                                                  1.0
1138                                                  1.0
1139                                                  1.0

      Can I have an alternative number/ desk number. I assure you that I will
not trouble you too much_yes  \
0                                                     1.0
1                                                     1.0
2                                                     0.0
3                                                     1.0
4                                                     0.0
…                                                     …
1135                                                  1.0
1136                                                  1.0
1137                                                  1.0
1138                                                  1.0
1139                                                  1.0

      Hope there will be no unscheduled meetings_yes  \
0                                                   1.0
1                                                   1.0
2                                                   0.0
3                                                   1.0
4                                                   1.0
…                                                   …
1135                                                1.0
1136                                                1.0
1137                                                1.0
1138                                                1.0
1139                                                1.0

      Have you taken a printout of your updated resume. Have you read the JD and
understood the same_yes  \
0                                                     1.0
1                                                     1.0
```

```
2                                                    0.0
3                                                    0.0
4                                                    1.0
…                                                    …
1135                                                 1.0
1136                                                 1.0
1137                                                 1.0
1138                                                 1.0
1139                                                 1.0

      Has the call letter been shared_yes  \
0                                     1.0
1                                     1.0
2                                     0.0
3                                     1.0
4                                     1.0
…                                     …
1135                                  1.0
1136                                  1.0
1137                                  1.0
1138                                  1.0
1139                                  1.0

      Have you obtained the necessary permission to start at the required
time_yes
0                                                    1.0
1                                                    1.0
2                                                    0.0
3                                                    1.0
4                                                    1.0
…                                                    …
1135                                                 1.0
1136                                                 1.0
1137                                                 1.0
1138                                                 1.0
1139                                                 1.0

[1140 rows x 111 columns]
```

56 Fit the truncated dataset with hyperparameter optimization using gridsearch by feeding the best estimator back into the logistic regression with CV model

[72]:
```
hyperparameters = [{'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag',␣
 ↪'saga']},
              {'penalty':['none', 'elasticnet', 'l1', 'l2']}]

lr = LogisticRegressionCV(class_weight='balanced',n_jobs=-1, random_state=2)

for parameter in hyperparameters:
    search = GridSearchCV(lr, parameter,
                                scoring = 'accuracy',
                                cv = 5,
                                verbose=3)

    search.fit(dataset.loc[train_idx],y.loc[train_idx])
    lr = search.best_estimator_
    print(search.best_estimator_)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END …solver=newton-cg;, score=0.724 total time=   0.6s
[CV 2/5] END …solver=newton-cg;, score=0.711 total time=   0.6s
[CV 3/5] END …solver=newton-cg;, score=0.737 total time=   0.8s
[CV 4/5] END …solver=newton-cg;, score=0.697 total time=   0.8s
[CV 5/5] END …solver=newton-cg;, score=0.691 total time=   0.6s
[CV 1/5] END …solver=lbfgs;, score=0.724 total time=   0.5s
[CV 2/5] END …solver=lbfgs;, score=0.711 total time=   0.5s
[CV 3/5] END …solver=lbfgs;, score=0.737 total time=   0.6s
[CV 4/5] END …solver=lbfgs;, score=0.697 total time=   0.6s
[CV 5/5] END …solver=lbfgs;, score=0.691 total time=   0.5s
[CV 1/5] END …solver=liblinear;, score=0.724 total time=   0.1s
[CV 2/5] END …solver=liblinear;, score=0.697 total time=   0.1s
[CV 3/5] END …solver=liblinear;, score=0.737 total time=   0.1s
[CV 4/5] END …solver=liblinear;, score=0.671 total time=   0.1s
[CV 5/5] END …solver=liblinear;, score=0.691 total time=   0.1s
[CV 1/5] END …solver=sag;, score=0.724 total time=   1.1s
[CV 2/5] END …solver=sag;, score=0.711 total time=   1.1s
[CV 3/5] END …solver=sag;, score=0.737 total time=   1.2s
[CV 4/5] END …solver=sag;, score=0.697 total time=   1.2s
[CV 5/5] END …solver=sag;, score=0.691 total time=   1.0s
[CV 1/5] END …solver=saga;, score=0.724 total time=   1.3s
[CV 2/5] END …solver=saga;, score=0.711 total time=   1.3s
[CV 3/5] END …solver=saga;, score=0.737 total time=   1.3s
[CV 4/5] END …solver=saga;, score=0.697 total time=   1.1s
[CV 5/5] END …solver=saga;, score=0.691 total time=   1.2s
LogisticRegressionCV(class_weight='balanced', n_jobs=-1, random_state=2,
```

```
                   solver='newton-cg')
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END …penalty=none;, score=nan total time=   0.0s
[CV 2/5] END …penalty=none;, score=nan total time=   0.0s
[CV 3/5] END …penalty=none;, score=nan total time=   0.0s
[CV 4/5] END …penalty=none;, score=nan total time=   0.0s
[CV 5/5] END …penalty=none;, score=nan total time=   0.0s
[CV 1/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 2/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 3/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 4/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 5/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 2/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 3/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 4/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 5/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l2;, score=0.724 total time=   0.7s
[CV 2/5] END …penalty=l2;, score=0.711 total time=   0.6s
[CV 3/5] END …penalty=l2;, score=0.737 total time=   0.8s
[CV 4/5] END …penalty=l2;, score=0.697 total time=   0.7s
[CV 5/5] END …penalty=l2;, score=0.691 total time=   0.6s
LogisticRegressionCV(class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='newton-cg')
```

57 in this case, logistic regression with CV has the same hyper-parameters as last time for higher metrics in precision, recall, and f1-score

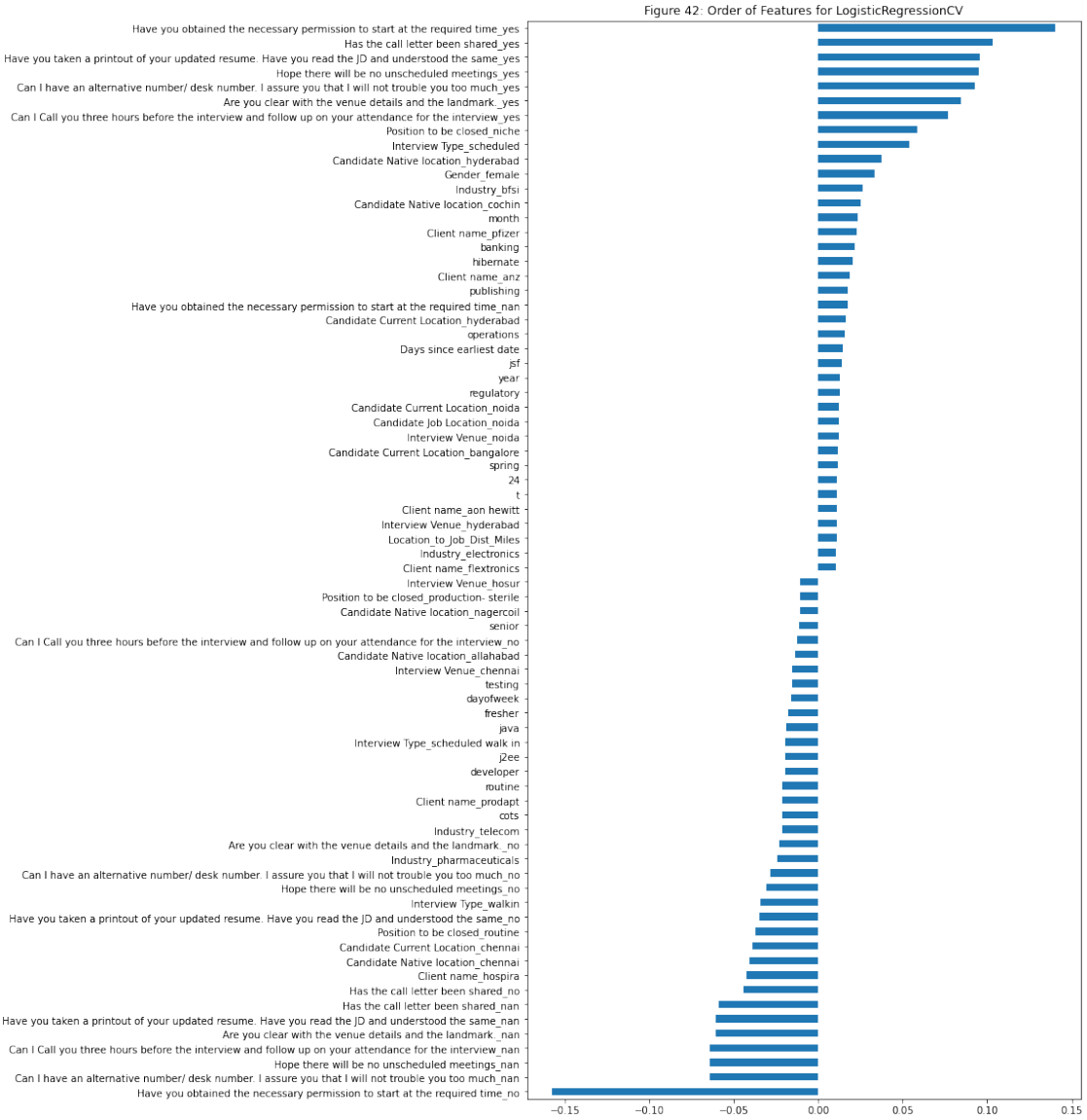
[73]:
```
model6 = search.best_estimator_

for train_idx, test_idx in skf.split(dataset, y):
    model6.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model6.predict(dataset.
  ↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.59      0.39      0.47       137
           1       0.71      0.84      0.77       243

    accuracy                           0.68       380
   macro avg       0.65      0.62      0.62       380
weighted avg       0.67      0.68      0.66       380

              precision    recall  f1-score   support
```

```
              0       0.66      0.42      0.52       137
              1       0.73      0.88      0.80       243

       accuracy                           0.71       380
      macro avg       0.69      0.65      0.66       380
   weighted avg       0.70      0.71      0.70       380

                  precision    recall  f1-score   support

              0       0.57      0.36      0.45       137
              1       0.70      0.85      0.77       243

       accuracy                           0.67       380
      macro avg       0.64      0.61      0.61       380
   weighted avg       0.66      0.67      0.65       380
```

58 feature importance of the new logistic regression with CV model is ordered for the target variable for attending, more explanations later

[74]:
```
feature_importances = pd.Series(model6.coef_[0], index=dataset.columns)
feature_importances[abs(feature_importances)>0.01].sort_values().
 ↪plot(kind="barh", figsize=(10,20),
                                        title = "Figure 42: Order of Features␣
 ↪for " + str(model6).split("(")[0]);
```



59 For the features importance, only keep the features that is absolute value 0.01 or more

[75]:
```
features3 = pd.DataFrame(feature_importances[abs(feature_importances)>0.01])
features3
```

[75]:

	0
Have you obtained the necessary permission to s…	-0.157748
Can I have an alternative number/ desk number. …	-0.064122
Hope there will be no unscheduled meetings_nan	-0.064122
Can I Call you three hours before the interview…	-0.064122

```
Are you clear with the venue details and the la… -0.061014
…                                                       …
Can I have an alternative number/ desk number. …  0.092641
Hope there will be no unscheduled meetings_yes      0.094844
Have you taken a printout of your updated resum…  0.095640
Has the call letter been shared_yes                 0.103315
Have you obtained the necessary permission to s…  0.140389

[74 rows x 1 columns]
```

60 Fit the truncated dataset with hyperparameter optimization using gridsearch by feeding the best estimator back into the logistic regression model

[76]:
```
hyperparameters = [{'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag',␣
 ↪'saga']},
                   {'C':[0.001, 0.01, 0.1, 1, 10, 100]},
              {'penalty':['none', 'elasticnet', 'l1', 'l2']}]

lr = LogisticRegression(class_weight='balanced',n_jobs=-1, random_state=2)

for parameter in hyperparameters:
    search = GridSearchCV(lr, parameter,
                                scoring = 'accuracy',
                                cv = 5,
                                verbose=3)

    search.fit(dataset.loc[train_idx],y.loc[train_idx])
    lr = search.best_estimator_
    print(search.best_estimator_)
```

```
Fitting 5 folds for each of 5 candidates, totalling 25 fits
[CV 1/5] END …solver=newton-cg;, score=0.724 total time=   0.0s
[CV 2/5] END …solver=newton-cg;, score=0.697 total time=   0.0s
[CV 3/5] END …solver=newton-cg;, score=0.612 total time=   0.0s
[CV 4/5] END …solver=newton-cg;, score=0.579 total time=   0.0s
[CV 5/5] END …solver=newton-cg;, score=0.697 total time=   0.0s
[CV 1/5] END …solver=lbfgs;, score=0.724 total time=   0.0s
[CV 2/5] END …solver=lbfgs;, score=0.697 total time=   0.0s
[CV 3/5] END …solver=lbfgs;, score=0.612 total time=   0.0s
[CV 4/5] END …solver=lbfgs;, score=0.579 total time=   0.0s
[CV 5/5] END …solver=lbfgs;, score=0.697 total time=   0.0s
[CV 1/5] END …solver=liblinear;, score=0.724 total time=   0.0s
[CV 2/5] END …solver=liblinear;, score=0.678 total time=   0.0s
[CV 3/5] END …solver=liblinear;, score=0.612 total time=   0.0s
[CV 4/5] END …solver=liblinear;, score=0.586 total time=   0.0s
```

```
[CV 5/5] END …solver=liblinear;, score=0.697 total time=   0.0s
[CV 1/5] END …solver=sag;, score=0.724 total time=   0.0s
[CV 2/5] END …solver=sag;, score=0.697 total time=   0.0s
[CV 3/5] END …solver=sag;, score=0.612 total time=   0.0s
[CV 4/5] END …solver=sag;, score=0.579 total time=   0.1s
[CV 5/5] END …solver=sag;, score=0.697 total time=   0.1s
[CV 1/5] END …solver=saga;, score=0.724 total time=   0.1s
[CV 2/5] END …solver=saga;, score=0.691 total time=   0.1s
[CV 3/5] END …solver=saga;, score=0.612 total time=   0.1s
[CV 4/5] END …solver=saga;, score=0.579 total time=   0.1s
[CV 5/5] END …solver=saga;, score=0.697 total time=   0.2s
LogisticRegression(class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='newton-cg')
Fitting 5 folds for each of 6 candidates, totalling 30 fits
[CV 1/5] END …C=0.001;, score=0.671 total time=   0.0s
[CV 2/5] END …C=0.001;, score=0.704 total time=   0.0s
[CV 3/5] END …C=0.001;, score=0.737 total time=   0.0s
[CV 4/5] END …C=0.001;, score=0.704 total time=   0.0s
[CV 5/5] END …C=0.001;, score=0.678 total time=   0.0s
[CV 1/5] END …C=0.01;, score=0.684 total time=   0.0s
[CV 2/5] END …C=0.01;, score=0.717 total time=   0.0s
[CV 3/5] END …C=0.01;, score=0.724 total time=   0.0s
[CV 4/5] END …C=0.01;, score=0.697 total time=   0.0s
[CV 5/5] END …C=0.01;, score=0.678 total time=   0.0s
[CV 1/5] END …C=0.1;, score=0.724 total time=   0.0s
[CV 2/5] END …C=0.1;, score=0.697 total time=   0.0s
[CV 3/5] END …C=0.1;, score=0.592 total time=   0.0s
[CV 4/5] END …C=0.1;, score=0.632 total time=   0.0s
[CV 5/5] END …C=0.1;, score=0.691 total time=   0.0s
[CV 1/5] END …C=1;, score=0.724 total time=   0.0s
[CV 2/5] END …C=1;, score=0.697 total time=   0.0s
[CV 3/5] END …C=1;, score=0.612 total time=   0.0s
[CV 4/5] END …C=1;, score=0.579 total time=   0.0s
[CV 5/5] END …C=1;, score=0.697 total time=   0.0s
[CV 1/5] END …C=10;, score=0.678 total time=   0.0s
[CV 2/5] END …C=10;, score=0.678 total time=   0.0s
[CV 3/5] END …C=10;, score=0.605 total time=   0.0s
[CV 4/5] END …C=10;, score=0.513 total time=   0.0s
[CV 5/5] END …C=10;, score=0.684 total time=   0.0s
[CV 1/5] END …C=100;, score=0.671 total time=   0.0s
[CV 2/5] END …C=100;, score=0.678 total time=   0.0s
[CV 3/5] END …C=100;, score=0.586 total time=   0.0s
[CV 4/5] END …C=100;, score=0.493 total time=   0.0s
[CV 5/5] END …C=100;, score=0.678 total time=   0.0s
LogisticRegression(C=0.01, class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='newton-cg')
Fitting 5 folds for each of 4 candidates, totalling 20 fits
[CV 1/5] END …penalty=none;, score=0.592 total time=   0.3s
```

```
[CV 2/5] END …penalty=none;, score=0.671 total time=   0.2s
[CV 3/5] END …penalty=none;, score=0.566 total time=   0.8s
[CV 4/5] END …penalty=none;, score=0.401 total time=   0.4s
[CV 5/5] END …penalty=none;, score=0.618 total time=   0.5s
[CV 1/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 2/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 3/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 4/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 5/5] END …penalty=elasticnet;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 2/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 3/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 4/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 5/5] END …penalty=l1;, score=nan total time=   0.0s
[CV 1/5] END …penalty=l2;, score=0.684 total time=   0.0s
[CV 2/5] END …penalty=l2;, score=0.717 total time=   0.0s
[CV 3/5] END …penalty=l2;, score=0.724 total time=   0.0s
[CV 4/5] END …penalty=l2;, score=0.697 total time=   0.0s
[CV 5/5] END …penalty=l2;, score=0.678 total time=   0.0s
LogisticRegression(C=0.01, class_weight='balanced', n_jobs=-1, random_state=2,
                   solver='newton-cg')
```

61 in this case, logistic regression has the same hyperparameters as last time for higher metrics in precision, recall, and f1-score

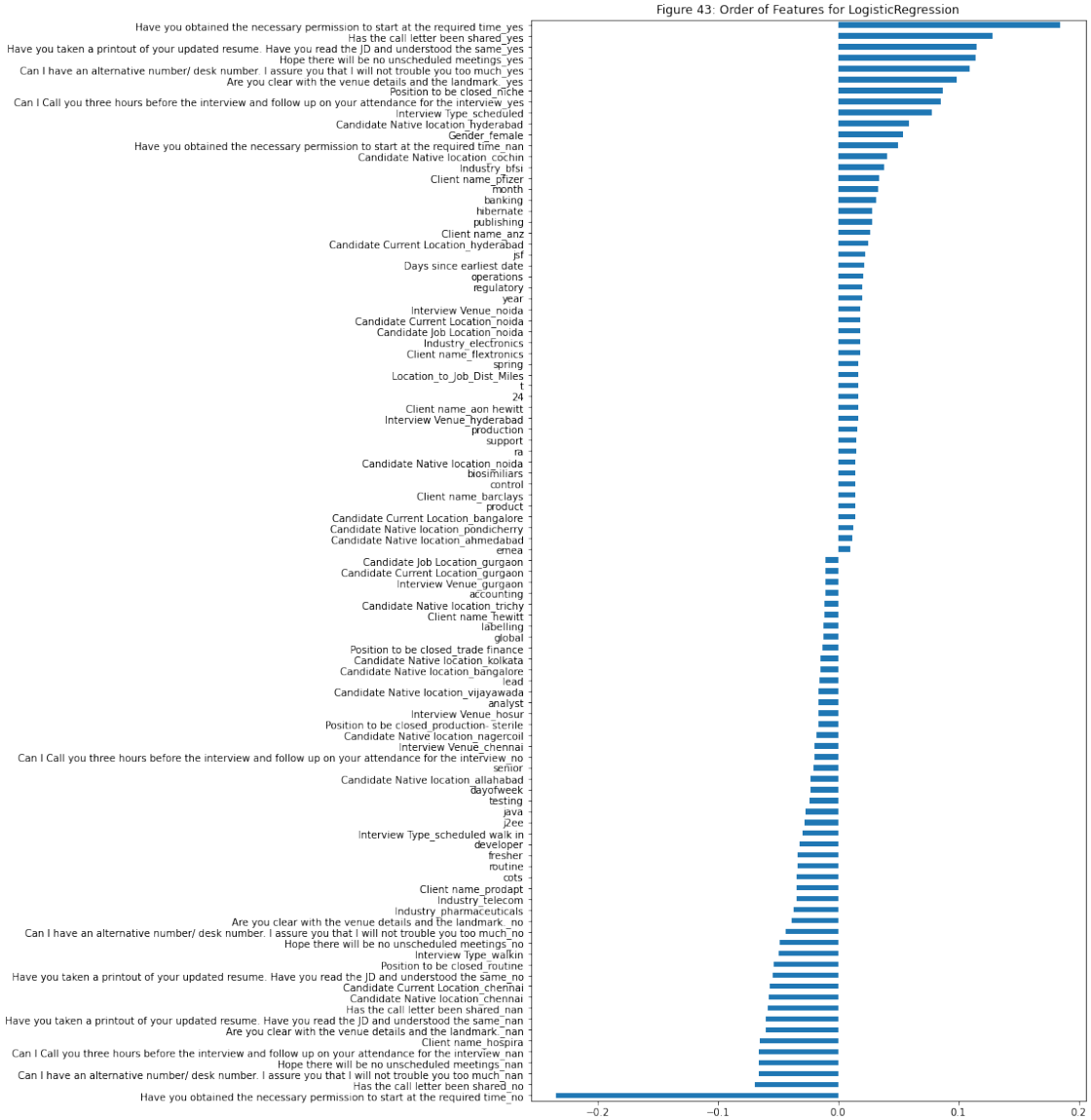
[77]:
```
model7 = search.best_estimator_

for train_idx, test_idx in skf.split(dataset, y):
    model7.fit(dataset.loc[train_idx],y.loc[train_idx])
    print(classification_report(y.loc[test_idx],model7.predict(dataset.
  ↪loc[test_idx])))
```

```
              precision    recall  f1-score   support

           0       0.62      0.51      0.56       137
           1       0.75      0.82      0.78       243

    accuracy                           0.71       380
   macro avg       0.68      0.67      0.67       380
weighted avg       0.70      0.71      0.70       380

              precision    recall  f1-score   support

           0       0.62      0.50      0.55       137
           1       0.74      0.83      0.78       243

    accuracy                           0.71       380
```

```
   macro avg       0.68      0.66      0.67       380
weighted avg       0.70      0.71      0.70       380

              precision    recall  f1-score   support

           0       0.57      0.38      0.46       137
           1       0.71      0.84      0.77       243

    accuracy                           0.67       380
   macro avg       0.64      0.61      0.61       380
weighted avg       0.66      0.67      0.65       380
```

62 feature importance of the new logistic regression with CV model is ordered for the target variable for attending, more explanations later

[78]:
```
feature_importances = pd.Series(model7.coef_[0], index=dataset.columns)
feature_importances[abs(feature_importances)>0.01].sort_values().
 ↪plot(kind="barh", figsize=(10,20),
                                        title = "Figure 43: Order of Features␣
 ↪for " + str(model7).split("(")[0]);
```



63 For the features importance, only keep the features that is absolute value 0.01 or more

[79]:
```
features4 = pd.DataFrame(feature_importances[abs(feature_importances)>0.01])
features4
```

[79]:

	0
Have you obtained the necessary permission to s…	-0.234597
Can I have an alternative number/ desk number. …	-0.066045
Hope there will be no unscheduled meetings_nan	-0.066045
Can I Call you three hours before the interview…	-0.066045

```
Are you clear with the venue details and the la… -0.060510
…                                                      …
Can I have an alternative number/ desk number. …  0.109380
Hope there will be no unscheduled meetings_yes      0.114410
Have you taken a printout of your updated resum…  0.115055
Has the call letter been shared_yes                 0.128045
Have you obtained the necessary permission to s…  0.184766

[99 rows x 1 columns]
```

64 given that the two models both use logistic regression, take the average of the important features and filter out the non essential features

[80]:
```
pd.concat([features3,features4],axis=1)
```

[80]:
```
                                                    0         0
Have you obtained the necessary permission to s… -0.157748 -0.234597
Can I have an alternative number/ desk number. … -0.064122 -0.066045
Hope there will be no unscheduled meetings_nan     -0.064122 -0.066045
Can I Call you three hours before the interview… -0.064122 -0.066045
Are you clear with the venue details and the la… -0.061014 -0.060510
…                                                       …         …
control                                               NaN  0.013902
biosimiliars                                          NaN  0.014289
ra                                                    NaN  0.014787
production                                            NaN  0.015601
support                                               NaN  0.015327

[99 rows x 2 columns]
```

65 The total number of important features is reduced to 77 from 77. Given the iterative nature of reducing the important features, another cycle could be done to reduce more features, but for the simplicity of the exercise, only two cycles of this process is done

[81]:
```
tot_Features = pd.DataFrame(pd.concat([features3,features4],axis=1).fillna(0).T.
  ↪mean()).sort_values(0)
tot_Features = tot_Features[abs(tot_Features[0])>.01]
print(tot_Features.to_string())
```

```
                                                    0
Have you obtained the necessary permission to start at the required time_no
```

-0.196172
Can I have an alternative number/ desk number. I assure you that I will not trouble you too much_nan -0.065084
Hope there will be no unscheduled meetings_nan
-0.065084
Can I Call you three hours before the interview and follow up on your attendance for the interview_nan -0.065084
Are you clear with the venue details and the landmark._nan
-0.060762
Have you taken a printout of your updated resume. Have you read the JD and understood the same_nan -0.060762
Has the call letter been shared_nan
-0.058800
Has the call letter been shared_no
-0.056880
Client name_hospira
-0.053595
Candidate Native location_chennai
-0.049237
Candidate Current Location_chennai
-0.047662
Position to be closed_routine
-0.045373
Have you taken a printout of your updated resume. Have you read the JD and understood the same_no -0.044586
Interview Type_walkin
-0.041684
Hope there will be no unscheduled meetings_no
-0.039544
Can I have an alternative number/ desk number. I assure you that I will not trouble you too much_no -0.035928
Industry_pharmaceuticals
-0.030827
Are you clear with the venue details and the landmark._no
-0.030802
Client name_prodapt
-0.027983
cots
-0.027983
Industry_telecom
-0.027983
routine
-0.027615
fresher
-0.025759
developer
-0.025638
Interview Type_scheduled walk in

-0.024299
j2ee
-0.023499
java
-0.023029
testing
-0.019584
dayofweek
-0.019385
Candidate Native location_allahabad
-0.018344
Interview Venue_chennai
-0.017466
Can I Call you three hours before the interview and follow up on your attendance for the interview_no -0.016223
senior
-0.016040
Candidate Native location_nagercoil
-0.014548
Position to be closed_production- sterile
-0.013650
Interview Venue_hosur
-0.013650
Candidate Current Location_bangalore
0.012756
Interview Venue_hyderabad
0.013661
Location_to_Job_Dist_Miles
0.013863
Client name_aon hewitt
0.013918
24
0.013964
t
0.013964
spring
0.014180
Client name_flextronics
0.014281
Industry_electronics
0.014281
Candidate Job Location_noida
0.015307
Interview Venue_noida
0.015307
Candidate Current Location_noida
0.015307
regulatory

0.016536
year
0.016580
Days since earliest date
0.018041
jsf
0.018215
operations
0.018519
Candidate Current Location_hyderabad
0.020713
Client name_anz
0.022549
publishing
0.022713
hibernate
0.024485
banking
0.026763
month
0.028334
Client name_pfizer
0.028485
Industry_bfsi
0.032440
Candidate Native location_cochin
0.033158
Have you obtained the necessary permission to start at the required time_nan
0.033595
Gender_female
0.043667
Candidate Native location_hyderabad
0.048306
Interview Type_scheduled
0.065983
Position to be closed_niche
0.072988
Can I Call you three hours before the interview and follow up on your attendance
for the interview_yes 0.081306
Are you clear with the venue details and the landmark._yes
0.091564
Can I have an alternative number/ desk number. I assure you that I will not
trouble you too much_yes   0.101011
Hope there will be no unscheduled meetings_yes
0.104627
Have you taken a printout of your updated resume. Have you read the JD and
understood the same_yes   0.105347
Has the call letter been shared_yes

```
0.115680
Have you obtained the necessary permission to start at the required time_yes
0.162577
```

66 This is the final visualization of the important features that influnce job interview attendance positively or negatively

66.0.1 For the features to be more interpretable, logistic regression is used to have both positive and negative features: conclusions:

66.0.2 1) For the candidate questionaires regardng permission to start at required time, call letter been shared, print out of resume, no unscheduled meetings, alternative number, venue details, call 3 hours before interview, if the candidate answered yes to all of them, they are more likely to show up for the interview. Alternatively, if the candidate said no to those questions or even worse left those questions blank, they are less likely to show up for the interview

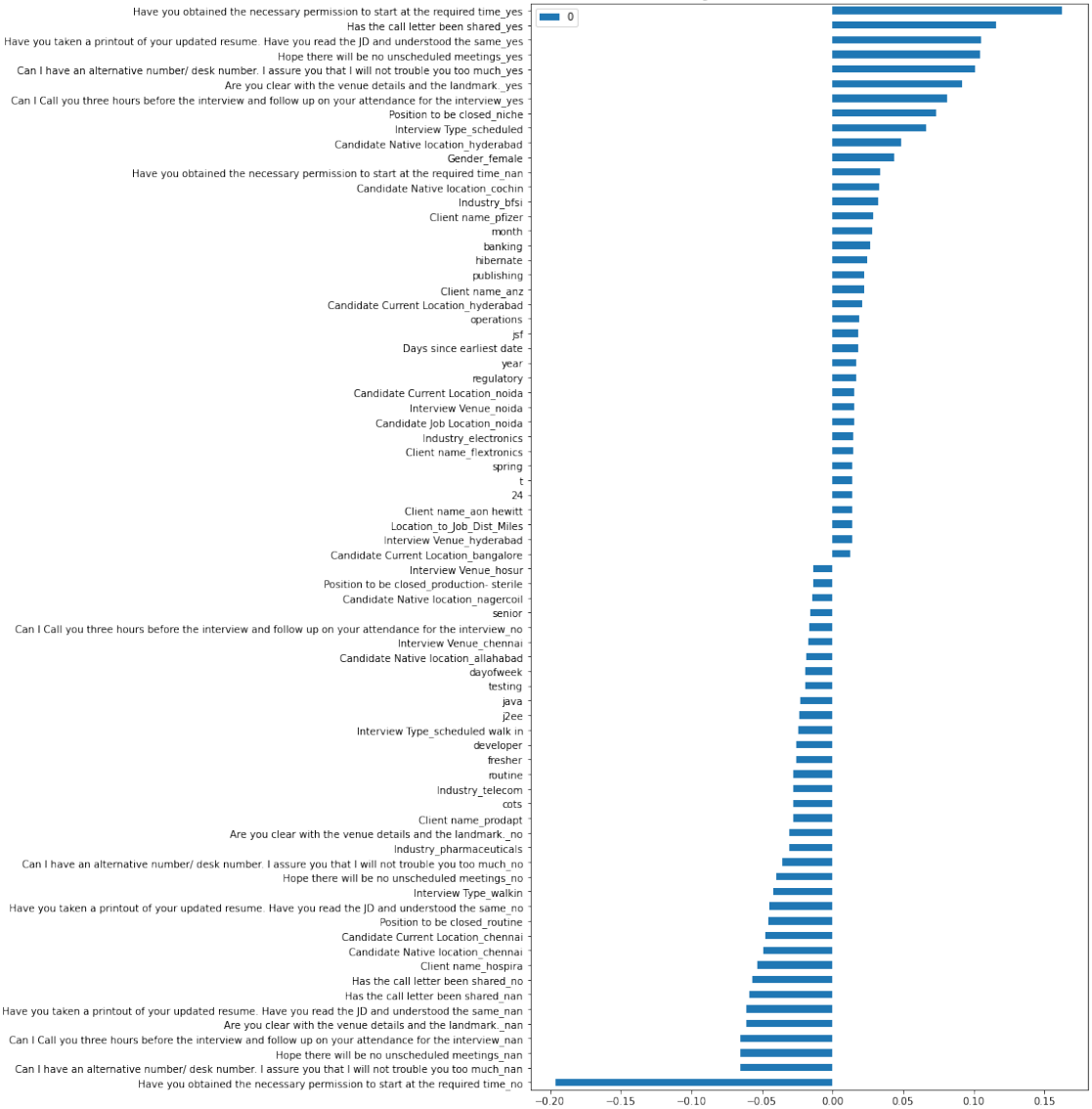
66.0.3 2) If the position is to be closed niche, interview is scheduled, industry is bfsi or electronics, client name is pfizer, anz, flextronics, or aon hewitt, if it is later during the year, the location of the job is far from the candidate, the candidate is more likely to show up for interview. If the position is to be closed sterile or routine, more closer to the beginning of the week, is a walk in, industry is telecom or pharmaceuticals, client name is hospira or prodapt, the candidate is less likely to show up for the interview

66.0.4 3) There are some data that would be improper to judge the candidate if they would show up for an interview such as their current location, native location, and gender because it may induce classism or gender bias even though it is in the data and prediction model. However, if the interview venue is in noida or hyderabad, candidates are more likely to show up. If the interview is in hosur or chennai, candidates are less likely to show up.

66.0.5 4) Based on skillset keywords, if the job entails cots, routine, fresher, developer, j2ee, java, testing, and senior, the candidate is less likely to show up. If the job entails t-24, spring, regulatory, jsf, operations, publishing, hibernate, and banking, the candidate is more likely to show up.

[82]:
```
tot_Features.plot(kind="barh", figsize=(10,20),
                                        title = "Figure 44: Order of Features␣
↪Combined");
```

Figure 44: Order of Features Combined



67 c. Pick one or more accuracy metrics for training and testing sets.

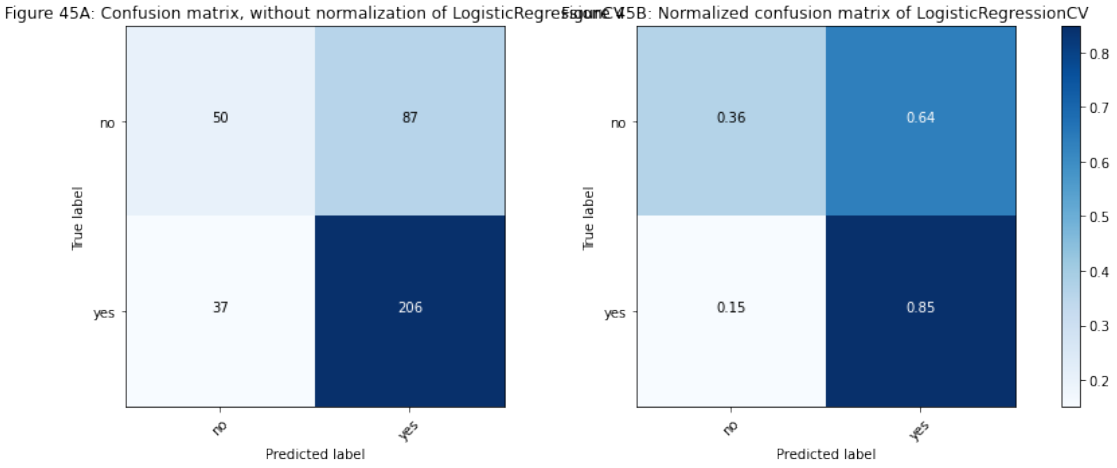
[83]:
```
def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):
    """
    This function prints and plots the confusion matrix.
    Normalization can be applied by setting `normalize=True`.
    """
```

```
    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()
    plt.ylabel('True label')
    plt.xlabel('Predicted label')
```

68 Using a confusion matrix, here is the normalized and non-normalized labeling of the logistic regression with and without CV prediction lavel compared to the actual label. Type 1 error is a false positive depicted in the upper right of the square while type 2 error is a false negative depicted in the bottom left of the square. Given that management is more concerned about no-shows to interviews, they should be more concerned about false positives because it is better to have more candidates in the interview pipeline rather than to overestimate people showing up for interviews and yet not have enough candidates interviewing.

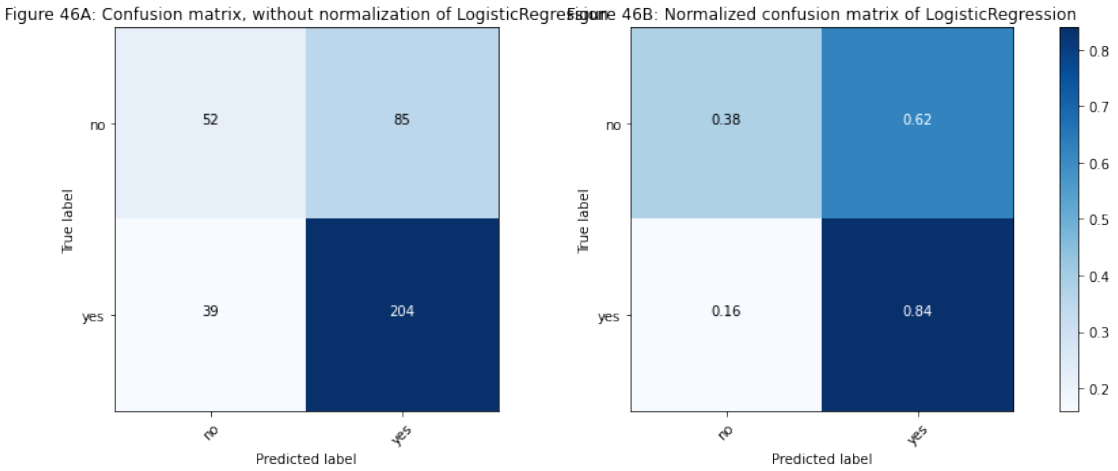
[84]:
```
# Compute confusion matrix
cnf_matrix = confusion_matrix(y.loc[test_idx],model6.predict(dataset.
 ↪loc[test_idx]))
np.set_printoptions(precision=2)

class_names = ['no', 'yes']

# Plot non-normalized confusion matrix
plt.figure(figsize = (15,5))
plt.subplot(1, 2, 1)
```

```
plot_confusion_matrix(cnf_matrix, classes=class_names, title='Figure 45A:␣
 ↪Confusion matrix, without normalization of ' + str(model6).split("(")[0])

# Plot normalized confusion matrix
plt.subplot(1, 2, 2)
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,␣
 ↪title='Figure 45B: Normalized confusion matrix of ' + str(model6).
 ↪split("(")[0])

plt.tight_layout()
plt.show()
```



[85]:
```
# Compute confusion matrix
cnf_matrix = confusion_matrix(y.loc[test_idx],model7.predict(dataset.
 ↪loc[test_idx]))
np.set_printoptions(precision=2)

class_names = ['no', 'yes']

# Plot non-normalized confusion matrix
plt.figure(figsize = (15,5))
plt.subplot(1, 2, 1)
plot_confusion_matrix(cnf_matrix, classes=class_names, title='Figure 46A:␣
 ↪Confusion matrix, without normalization of ' + str(model7).split("(")[0])

# Plot normalized confusion matrix
plt.subplot(1, 2, 2)
plot_confusion_matrix(cnf_matrix, classes=class_names, normalize=True,␣
 ↪title='Figure 46B: Normalized confusion matrix of ' + str(model7).
 ↪split("(")[0])
```

```
plt.tight_layout()
plt.show()
```



69 Using train test split, calculate the metrics of Roc area under the curve, accuracy, precision, recall, and F1 score for the 80-20 split of the dataset for both logistic regression with and without cross validation models

[86]:
```
x_train, x_test, y_train, y_test = train_test_split(dataset, y, test_size=0.2,␣
↪random_state=2)
```

[87]:
```
def get_metrics(y_true, y_pred, treshhold=.5):
    y_pred = (y_pred > treshhold).astype(np.int32)
    return dict(
        RocAuc="{:.4f}".format(metrics.roc_auc_score(y_test, y_pred)),
        Accuracy="{:.4f}".format(metrics.accuracy_score(y_test, y_pred)),
        Precision="{:.4f}".format(metrics.precision_score(y_test, y_pred)),
        Recall="{:.4f}".format(metrics.recall_score(y_test, y_pred)),
        F1="{:.4f}".format(metrics.f1_score(y_test, y_pred))
    )
```

70 The metrics for both models are about the same with the same shape underneath the curve at 71% accuracy

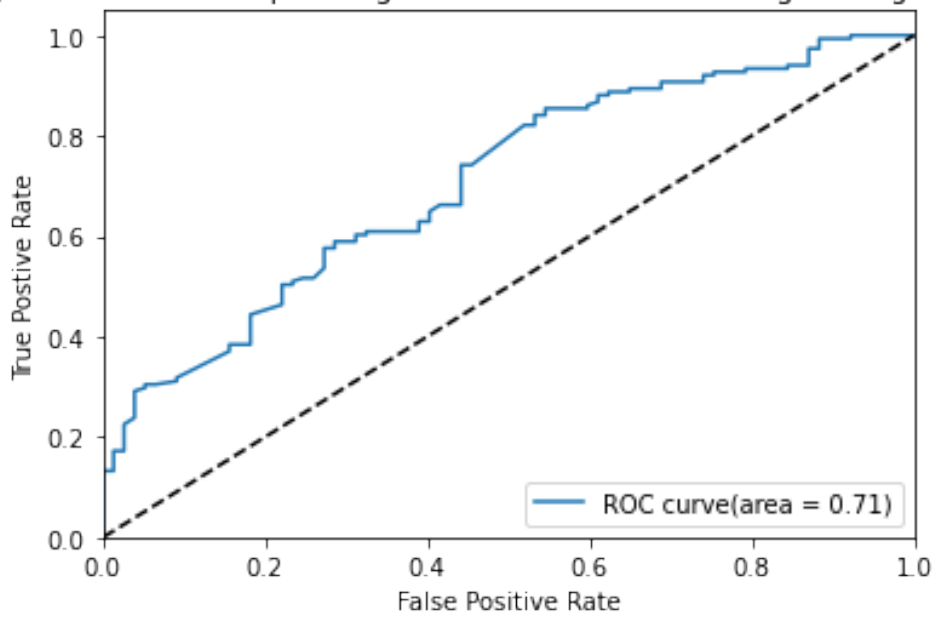
[88]:
```
print("Test Dataset :", get_metrics(y_test, model6.predict(x_test)))
```

```
Test Dataset : {'RocAuc': '0.6287', 'Accuracy': '0.7105', 'Precision': '0.7348',
```

```
'Recall': '0.8808', 'F1': '0.8012'}
```

[89]:
```
fpr, tpr, thresholds = roc_curve(y_test, model6.predict_proba(x_test)[:,1])

plt.figure()
plt.plot(fpr, tpr, label='ROC curve(area = %0.2f)' %roc_auc_score(y_test,␣
 ↪model6.predict_proba(x_test)[:,1]))
plt.plot([0,1], [0,1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Postive Rate')
plt.title('Figure 47: Receiver operating characteristic (ROC) of ' +␣
 ↪str(model6).split("(")[0])
plt.legend(loc='lower right')
plt.show()
```

Figure 47: Receiver operating characteristic (ROC) of LogisticRegressionCV

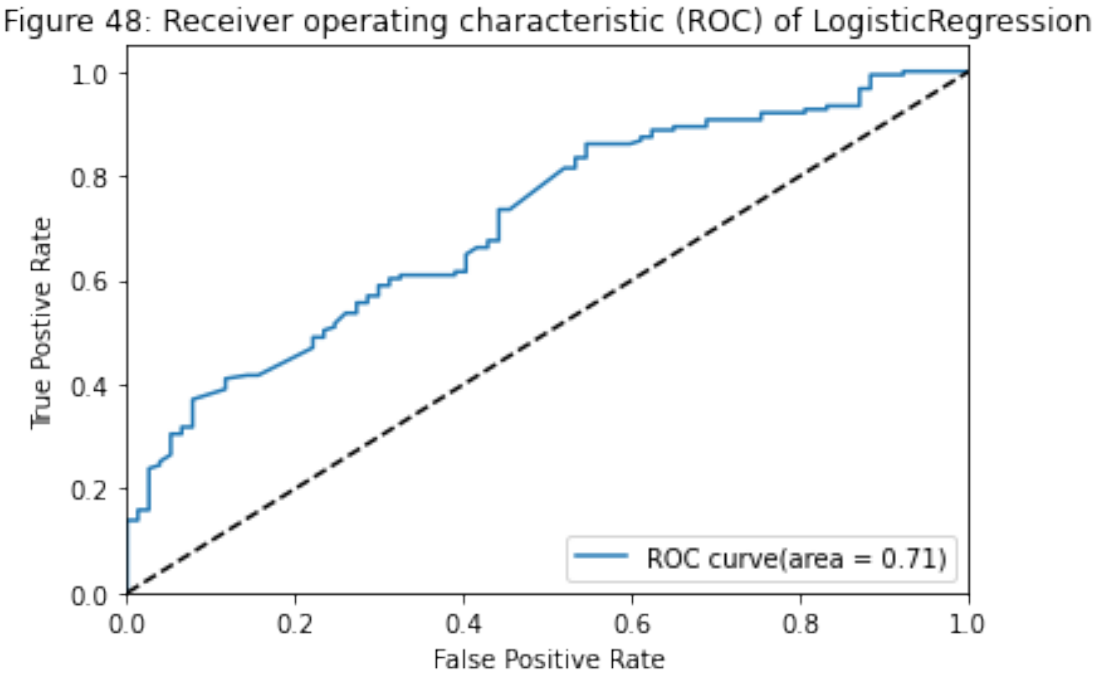


[90]:
```
print("Test Dataset :", get_metrics(y_test, model7.predict(x_test)))
```

```
Test Dataset : {'RocAuc': '0.6319', 'Accuracy': '0.7105', 'Precision': '0.7374',
'Recall': '0.8742', 'F1': '0.8000'}
```

[91]:
```
fpr, tpr, thresholds = roc_curve(y_test, model7.predict_proba(x_test)[:,1])

plt.figure()
```

```
plt.plot(fpr, tpr, label='ROC curve(area = %0.2f)' %roc_auc_score(y_test,␣
 ↪model7.predict_proba(x_test)[:,1]))
plt.plot([0,1], [0,1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Postive Rate')
plt.title('Figure 48: Receiver operating characteristic (ROC) of ' +␣
 ↪str(model7).split("(")[0])
plt.legend(loc='lower right')
plt.show()
```



71 b. Provide a probability and a prediction output for the candidates where the "Observed Attendance" column is null.

[92]:
```
df = checkpoint2
prediction = pd.DataFrame()
```

72 Using the previous checkpoint, retrieve only the dataset where the observed attendance is NaN and predict the attendance for those candidates. Given that the more accurate model requires the dataset with priority on important features, only that subset of the dataset is used

[93]:
```
dataset = df[df['Observed Attendance_nan']==1].reset_index(drop=True).
 ↪drop(columns = ['Observed Attendance_nan'])
prediction['Name(Cand ID)'] = dataset['Name(Cand ID)']
dataset = dataset[keep.index]
dataset
```

[93]:
```
    Have you obtained the necessary permission to start at the required time_no
\
0                                                     0
1                                                     0
2                                                     0
3                                                     0
4                                                     0
..                                                  …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    0

    Can I have an alternative number/ desk number. I assure you that I will not
trouble you too much_nan  \
0                                                     0
1                                                     0
2                                                     0
3                                                     0
4                                                     0
..                                                  …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    1

    Hope there will be no unscheduled meetings_nan  \
0                                                   0
1                                                   0
2                                                   0
3                                                   0
4                                                   0
```

```
..                                                    …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    1

    Can I Call you three hours before the interview and follow up on your
attendance for the interview_nan  \
0                                                     0
1                                                     0
2                                                     0
3                                                     0
4                                                     0
..                                                    …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    1

    Are you clear with the venue details and the landmark._nan  \
0                                                     0
1                                                     0
2                                                     0
3                                                     0
4                                                     0
..                                                    …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    1

    Have you taken a printout of your updated resume. Have you read the JD and
understood the same_nan  \
0                                                     0
1                                                     0
2                                                     0
3                                                     0
4                                                     0
..                                                    …
88                                                    0
89                                                    0
90                                                    0
91                                                    0
92                                                    1
```

```
    Has the call letter been shared_nan  Position to be closed_routine  \
0                                     0                              0
1                                     0                              0
2                                     0                              0
3                                     0                              0
4                                     0                              1
..                                  …                              …
88                                    0                              1
89                                    0                              1
90                                    0                              1
91                                    0                              0
92                                    1                              0

    Has the call letter been shared_no  Candidate Current Location_chennai  \
0                                    0                                   0
1                                    0                                   0
2                                    0                                   1
3                                    0                                   1
4                                    0                                   1
..                                 …                                   …
88                                   0                                   1
89                                   0                                   1
90                                   0                                   1
91                                   0                                   1
92                                   0                                   1

    …  Candidate Native location_hyderabad  Interview Type_scheduled  \
0   …                                    0                         0
1   …                                    0                         0
2   …                                    0                         0
3   …                                    0                         0
4   …                                    0                         0
..  …                                  …                         …
88  …                                    1                         0
89  …                                    1                         0
90  …                                    1                         0
91  …                                    0                         1
92  …                                    0                         1

    Position to be closed_niche  \
0                             0
1                             0
2                             0
3                             0
4                             0
..                          …
```

```
88                              0
89                              0
90                              0
91                              1
92                              1

    Can I Call you three hours before the interview and follow up on your
attendance for the interview_yes  \
0                                                       1
1                                                       1
2                                                       1
3                                                       1
4                                                       1
..                                                    …
88                                                      1
89                                                      1
90                                                      1
91                                                      1
92                                                      0

    Are you clear with the venue details and the landmark._yes  \
0                                                       1
1                                                       1
2                                                       1
3                                                       1
4                                                       1
..                                                    …
88                                                      1
89                                                      1
90                                                      1
91                                                      1
92                                                      0

    Can I have an alternative number/ desk number. I assure you that I will not
trouble you too much_yes  \
0                                                       1
1                                                       1
2                                                       1
3                                                       1
4                                                       1
..                                                    …
88                                                      1
89                                                      1
90                                                      1
91                                                      1
92                                                      0
```

```
    Hope there will be no unscheduled meetings_yes  \
0                                                    1
1                                                    1
2                                                    1
3                                                    1
4                                                    1
..                                                   …
88                                                   1
89                                                   1
90                                                   1
91                                                   1
92                                                   0

    Have you taken a printout of your updated resume. Have you read the JD and
understood the same_yes  \
0                                                      1
1                                                      1
2                                                      1
3                                                      1
4                                                      1
..                                                     …
88                                                     1
89                                                     1
90                                                     1
91                                                     1
92                                                     0

    Has the call letter been shared_yes  \
0                                        1
1                                        1
2                                        1
3                                        1
4                                        1
..                                       …
88                                       1
89                                       1
90                                       1
91                                       1
92                                       0

    Have you obtained the necessary permission to start at the required time_yes
0                                                      1
1                                                      1
2                                                      1
3                                                      1
4                                                      1
..                                                     …
```

```
88                                                  1
89                                                  1
90                                                  1
91                                                  1
92                                                  0

[93 rows x 111 columns]
```

73 For both models, the prediction for those candidates is that they would show up to the interview. The probability that they would not show up for the interview is on the scale of 1e-14 to 1e-16, so these results are highly probable

[94]:
```
model6.predict(dataset)
```

[94]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1], dtype=uint8)
```

[95]:
```
model6.predict_proba(dataset)[:,0]
```

[95]:
```
array([3.26e-14, 1.07e-14, 4.44e-15, 4.22e-15, 4.88e-15, 4.66e-15,
       4.66e-15, 4.88e-15, 8.22e-15, 0.00e+00, 0.00e+00, 2.22e-16,
       0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
       0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
       2.22e-16, 2.22e-16, 2.22e-16, 4.44e-16, 6.66e-16, 2.22e-16,
       4.44e-16, 2.22e-16, 2.22e-16, 4.44e-16, 0.00e+00, 0.00e+00,
       0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00,
       2.22e-16, 2.22e-16, 2.22e-16, 0.00e+00, 0.00e+00, 0.00e+00,
       2.22e-16, 6.66e-16, 2.22e-16, 6.37e-14, 2.22e-16, 0.00e+00,
       4.44e-16, 4.44e-16, 2.22e-16, 0.00e+00, 0.00e+00, 0.00e+00,
       0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 1.39e-13, 2.22e-16,
       1.75e-14, 2.22e-16, 6.66e-16, 0.00e+00, 0.00e+00, 2.71e-14,
       9.48e-14, 7.86e-14, 1.80e-14, 2.78e-14, 2.71e-14, 0.00e+00,
       0.00e+00, 0.00e+00, 2.22e-16, 2.44e-15, 8.88e-16, 0.00e+00,
       0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 0.00e+00, 6.66e-16,
       6.66e-16, 0.00e+00, 1.78e-15])
```

[96]:
```
model7.predict(dataset)
```

[96]:
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
       1, 1, 1, 1, 1], dtype=uint8)
```

[97]:
```
model7.predict_proba(dataset)[:,0]
```

[97]:
```
array([0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.,
       0., 0., 0., 0., 0., 0., 0., 0.])
```

74 In this case, the logistic regression with cross validation is used because the probability of the candidate not showing up still has a chance to be wrong, while the overall probability of the model predicting correctly is 71%. The reason that the probability of the prediction is so high might be due to the optimization of the hyper parameters and modeling the data on the important features only.

[98]:
```
no_show = pd.DataFrame(model6.predict_proba(dataset)[:,0],columns =␣
↪['Probability of candidate not showing up'])
no_show
```

[98]:

	Probability of candidate not showing up
0	3.264056e-14
1	1.065814e-14
2	4.440892e-15
3	4.218847e-15
4	4.884981e-15
..	…
88	0.000000e+00
89	6.661338e-16
90	6.661338e-16
91	0.000000e+00
92	1.776357e-15

[93 rows x 1 columns]

75 The candidate ID, the predicted attendance, the probability of them not showing up to interview, and the overall probability of the prediction are concatenated to a new dataframe and exported as a .csv.

[99]:
```
prediction = pd.concat([prediction,pd.DataFrame(model7.
 ↪predict(dataset),columns=['Predicted Attendance']).
 ↪replace(1,'yes'),no_show],axis=1)
prediction.loc[:,'Overall Probability of Prediction'] = 0.7105
prediction
```

[99]:

	Name(Cand ID)	Predicted Attendance	Probability of candidate not showing up	Overall Probability of Prediction
0	10	yes	3.264056e-14	0.7105
1	20	yes	1.065814e-14	0.7105
2	30	yes	4.440892e-15	0.7105
3	40	yes	4.218847e-15	0.7105
4	50	yes	4.884981e-15	0.7105
..	…	…	…	…
88	1171	yes	0.000000e+00	0.7105
89	1189	yes	6.661338e-16	0.7105
90	1207	yes	6.661338e-16	0.7105
91	1222	yes	0.000000e+00	0.7105
92	1233	yes	1.776357e-15	0.7105

[93 rows x 4 columns]

[100]:
```
prediction.to_csv('prediction.csv',index=False)
```

76 Part 2 - Model Interpretation

76.0.1 A client has scored a candidate with your model and it gave the candidate a 30% chance of attending the interview. However, the candidate did come to the interview. The client would like to know why there is this apparent discrepancy in your model.

Even with feature engineering, hyperparameter optimization, and choosing only the important features to model on, there is 29% that the model could be inaccurate. If the candidate is predicted not to show up and shows up, this is a type 2 error, as described above. It is better to have too many candidates interviewing for a better pool of talent to choose from than to have less candidates showing up for interviews and a lesser talent pool, where in this instance type 2 errors are better than type 1 errors to have.

76.0.2 - How would you explain this occurrence? What would you have ideally done to prevent this confusion with the client? Do your accuracy metrics help explain this?

The occurance happens because the model is 71% accurate with the presented data, where the recall score would depict true positives divided by the summation of true positives and false negatives hovering around 82-88% depending on the split of the random state. I would explain that type 1 errors are not as much as a concern compared to type 2 errors when if less talent is available to interview rather than too many candidates interviewing.

76.0.3 - Could you provide a better way for the client to evaluate your model's performance?

To evaluate the model's performance, use the model on more data to see how well it performs outside of the dataset in which it is trained to see how it performs. Also provide more data that the model could learn from in terms of more rows and columns for more relevant feature engineering and higher accuracy.

76.0.4 - What accuracy metrics would help explain this gap in understanding?

Recall is type 1 errors, which is when a candidate is predicted not to interview and shows up. Precision is type 2 errors, which is when a candidate is predicted to interview and not shows up, which would waste more time and resources if they were to reschedule. F1 score is the balance between recall and precision, ROC area under the curve is how the model is performant when evaluating positive to negative prediction classifications while accuracy is the overall reliability of the prediction.

[ ]: