

A Mini-Project on “Data Analyzation and Prediction on job postings of data scientists in last 2 years”

3rd YEAR OF ENGINEERING (Computer Engineering)

By

Sudity Khushi	COTB02
Lokesh Kolte	COTB03
Rohit Mahajan	COTB08
Prathmesh Pachpute	COTB16

Under The Guidance of Prof. A. K. Magar



SINHGAD ACADEMY OF ENGINEERING

CERTIFICATE

This is to certify that the project report entitles

“Data Analyzation and Prediction on job postings of data scientists in last 2 years”

Submitted By

Sudity Khushi **COTB02**

Lokesh Kolte **COTB03**

Rohit Mahajan **COTB08**

Prathmesh Pachpute **COTB16**

Is a bonafide work carried out under the supervision of

Prof. A. K. Magar and it is approved for the partial fulfillment of the requirement of third year
computer engineering.

Prof. A. K. Magar
Guide
Dept. of Comp. Eng.

Prof. B. B. Gite
Head
Dept. of Comp. Eng.

Dr. K. P. Patil

Principal

Sinhgad Academy of Engineering Pune – 48, Pune

Date:

Working-

We have divided the working in 2 parts i.e. Analysis and Prediction. Analysis has the graphs and analyzed data, we have used **pandas**, **matplotlib**, etc. modules to analyze the dataset. Prediction has **sklearn** module included, we are training the machine by giving inputs and it can predict the future job postings in the program.

Analysis-

- Loads the dataset and plot the graph of data scientist job postings per month in last 2 years.
- Shows the Average job postings in 1 month.
- Minimum and maximum job postings in last 2 years.
- Shows pie-chart of number of data scientist job postings of every month in 2018.
- Shows pie-chart of number of data scientist job postings of every month in 2019.
- Shows pie-chart on number of total job postings of data scientists in last 2 years.
- Shows double bar graph for male and female job postings per month in 2018
- Shows double bar graph for male and female job postings per month in 2019

Prediction-

- Program prompts the user to enter the month for the prediction.
- Program prompts the user to enter the year for the prediction.
- Program uses the Linear Regression to predict the job postings in the month that user has entered.
- Shows a dialogue box through **pyautogui** module to display the prediction result.

Code of the project –

```
import pandas
import numpy as np
import matplotlib.pyplot as plt
import pyautogui as win
from sklearn import linear_model

# Declaring a function for getting average of job postings in a month
def Avg(lst):
    return sum(lst) / len(lst)

# Declaring a function for getting Minimum Job Postings in Past 2 years
def Min(lst):
    return min(lst)

# Declaring a function for getting Maximum Job Postings in Past 2 years
def Max(lst):
    return max(lst)

df = pandas.read_csv('Demo_2_.csv')
print(df)

monthyear = (df['month&year']).tolist()
male = (df['Male']).tolist()
female = (df['Female']).tolist()
monthyear = monthyear[48:]

posting = (df['Postings']).tolist()
posting = posting[48:]

plt.rcParams["figure.figsize"] = [16, 6]
ax = plt.axes()
ax.set_facecolor("yellow")
plt.title("Number of Job Postings of Data Scientists Every Month in Past 2 Years")
plt.xlabel("Months with Year")
plt.ylabel("Number of Postings")

plt.plot(monthyear, posting, color='r')

plt.ylim(5000, 10000)
plt.legend(['No. of job posted'])
```

```

plt.show()

avg = "Average Job Postings in 1 Month - {:.2f}\nMinimum Job Postings in Past 2 years
- {:.0f} \nMaximum Job Postings in Past 2 years- {:.0f} ".format(Avg(posting), Min(p
osting), Max(posting))
win.alert(avg)

# month2018 = ['Jan-18','Feb-18','Mar-18','Apr-18','May-18','Jun-18','Jul-18','Aug-
18','Sep-18','Oct-18','Nov-18','Dec-18']

# job2018= [8100,8800,8200,8300,7500,7800,8300,8000,7500,7700,8300,8500]

month2018 = []

def mol(lst):
    for i in range(0, 12):
        temp = lst[i]
        month2018.append(temp)

job2018 = []

def pos1(lst):
    for i in range(0, 12):
        temp1 = lst[i]
        job2018.append(temp1)

mol(monthyear)
pos1(posting)

explode = (0.1, 0.0, 0.1, 0.2, 0.0, 0.1, 0.1, 0.2, 0.1, 0.1, 0.2, 0.1)

# Creating color parameters
colors = ("orange", "cyan", "brown",
          "grey", "indigo", "beige", "yellow", "green", "blue", "violet", "pink", "pu
rple")

# Wedge properties
wp = {'linewidth': 1, 'edgecolor': "green"}
i = 0

def pie1():
    global i
    k = job2018[i]
    i = i + 1
    return k

```

```

plt.style.use('dark_background')
# Creating plot
fig, ax = plt.subplots(figsize=(10, 7))
wedges, texts, autotexts = ax.pie(job2018,
                                   autopct=lambda pct1: pie1(),
                                   explode=explode,
                                   labels=month2018,
                                   shadow=True,
                                   colors=colors,
                                   startangle=90,
                                   wedgeprops=wp,
                                   textprops=dict(color="red"))

# Adding legend
ax.legend(wedges, month2018,
          title="Total month",
          loc="best",
          bbox_to_anchor=(0.8, -0.4, 0.5, 1))

plt.setp(autotexts, size=8, weight="bold")
ax.set_title("Number of Job Postings of Data Scientists Every Month in 2018")

plt.show()

# month = ["Jan-19", "Feb-19", "Mar-19", "Apr-19", "May-19", "Jun-19", "Jul-19", "Aug-
19", "Sep-19", "Oct-19", "Nov-19", "Dec-19"]

# job= [8000,8200,7800,8000,8300,8800,8100,8400,8300,8500,8400,8100]
month2019 = []

def mo2(lst):
    for i in range(12, 24):
        temp = lst[i]
        month2019.append(temp)

job2019 = []

def pos2(lst):
    for i in range(12, 24):
        temp1 = lst[i]
        job2019.append(temp1)

mo2(monthyear)
pos2(posting)

```

```

explode = (0.1, 0.0, 0.1, 0.2, 0.0, 0.1, 0.1, 0.2, 0.1, 0.1, 0.2, 0.1)

# Creating color parameters
colors = ("orange", "cyan", "brown",
          "grey", "indigo", "beige", "red", "yellow", "blue", "violet", "pink", "purple")

# Wedge properties
wp = {'linewidth': 1, 'edgecolor': "green"}

i = 0

def pie2():
    global i
    k = job2019[i]
    i = i + 1
    return k

# Creating plot
fig, ax = plt.subplots(figsize=(10, 7))
wedges, texts, autotexts = ax.pie(job2019,
                                   autopct=lambda pct2: pie2(),
                                   explode=explode,
                                   labels=month2019,
                                   shadow=True,
                                   colors=colors,
                                   startangle=90,
                                   wedgeprops=wp,
                                   textprops=dict(color="green"))

# Adding legend
ax.legend(wedges, month2019,
          title="Total month",
          loc="best",
          bbox_to_anchor=(0.8, -0.4, 0.5, 1))

plt.setp(autotexts, size=8, weight="bold")
ax.set_title("Number of Job Postings of Data Scientists Every Month in 2019")

plt.show()

total = []

def tot2018(lst):
    tem = 0
    for i in lst:
        tem = tem + i

```

```

total.append(tem)

total2019 = []

def tot2019(lst):
    tem = 0
    for i in lst:
        tem = tem + i
    total.append(tem)

tot2018(job2018)
tot2019(job2019)
print(total)
year = ["2018", "2019"]

explode = (0.0, 0.1)

# Creating color parameters
colors = ("orange", "cyan",
          )

# Wedge properties
wp = {'linewidth': 1, 'edgecolor': "green"}
i = 0

def pie3():
    global i
    k = total[i]
    i = i + 1
    return k

# Creating plot
fig, ax = plt.subplots(figsize=(10, 7))
wedges, texts, autotexts = ax.pie(total,
                                   autopct=lambda jjj: pie3(),

                                   explode=explode,
                                   labels=year,
                                   shadow=True,
                                   colors=colors,
                                   startangle=90,
                                   wedgeprops=wp,
                                   textprops=dict(color="red"))

# Adding legend
ax.legend(wedges, year,

```



```
        title="year",
        loc="best",
        bbox_to_anchor=(0.8, -0.4, 0.5, 1))

plt.setp(autotexts, size=8, weight="bold")
ax.set_title("Number of Job Postings of Data Scientists in past 2 year")

plt.show()
```

```
male2018 = []
```

```
def male1(lst):
    for i in range(48,60):
        mens = lst[i]
        male2018.append(mens)
```

```
female2018 = []
```

```
def female1(lst):
    for i in range(48,60):
        female = lst[i]
        female2018.append(female)
```

```
male2019 = []
```

```
def male2(lst):
    for i in range(60,72):
        mens = lst[i]
        male2019.append(mens)
```

```
female2019 = []
```

```
def female2(lst):
    for i in range(60,72):
        female = lst[i]
        female2019.append(female)
```

```
male1(male)
```

```
female1(female)
```

```

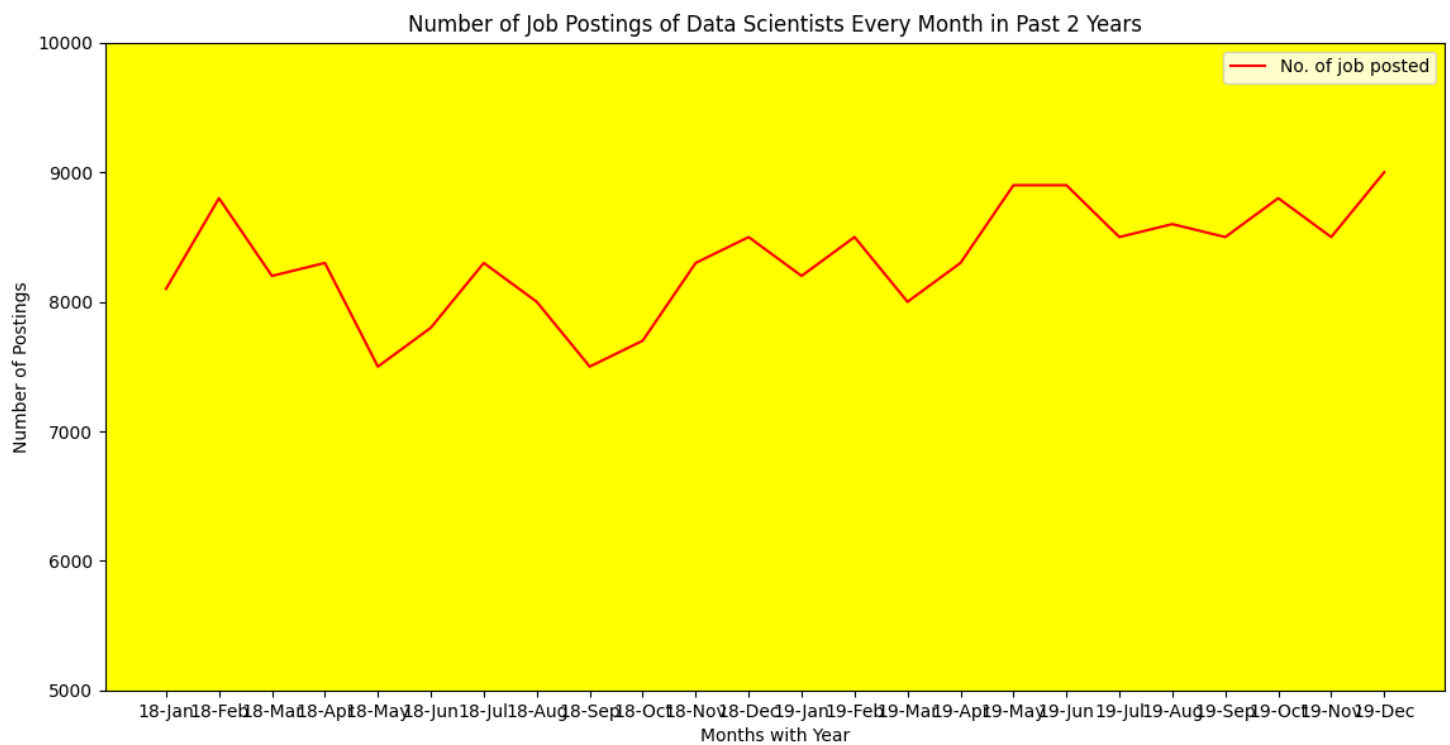
male2(male)
female2(female)
w = 0.4
bar1 = np.arange(len(month2018))
bar2 = [i + w for i in bar1]
plt.bar(bar1, male2018, w, label="male", color="violet")
plt.bar(bar2, female2018, w, label="female", color="silver")
plt.xticks(bar1 + w / 2, month2018)
plt.ylabel("job posting")
plt.xlabel("male and female")
plt.title("job posting for male and female in 2018")
plt.legend()
plt.show()

w = 0.4
bar3 = np.arange(len(month2019))
bar4 = [i + w for i in bar1]
plt.bar(bar1, male2019, w, label="male", color="cyan")
plt.bar(bar2, female2019, w, label="Female", color="pink")
plt.xticks(bar1 + w / 2, month2019)
plt.ylabel("job posting")
plt.xlabel("male and female")
plt.title("job posting for male and female in 2018")
plt.legend()

plt.show()
win.alert("Here you can check prediction in the console")
reg = linear_model.LinearRegression()
reg.fit(df[['month', 'year']], df.Postings)
x=int(input("Enter month no(ex:1/2/3/4/5/6/7/8/9/10/11/12):=>"))
y=int(input("Enter year (ex:2020/2021/2022/2023/2024/2025....):=>"))
prediction=reg.predict([(x,y)])
predict = "predicted posting for the given month is {posting:.0f} "
predic=predict.format(posting=int(prediction))
win.alert(predic)

```

Outputs-



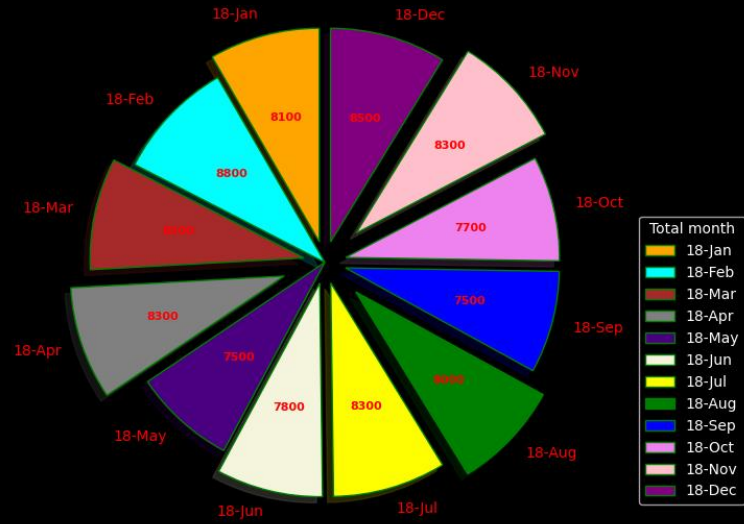
```
1 import pandas
2 import numpy as np
3 import matplotlib.pyplot as plt
4 import pyautogui as win
5 from sklearn import linear_model
6
7
8 # Declaring a function for getting average of job postings in a month
9 def Avg(lst):
10     return sum(lst) / len(lst)
11
12
13 # Declaring a function for getting Minimum Job Postings in Past 2 years
14 def Min(lst):
15     return min(lst)
16
17
18 # Declaring a function for getting Maximum Job Postings in Past 2 years
19 def Max(lst):
20     return max(lst)
21
22
23 df = pandas.read_csv('Demo_2_.csv')
24 print(df)
```

Average Job Postings in 1 Month - 8320.83
Minimum Job Postings in Past 2 years- 7500
Maximum Job Postings in Past 2 years- 9000

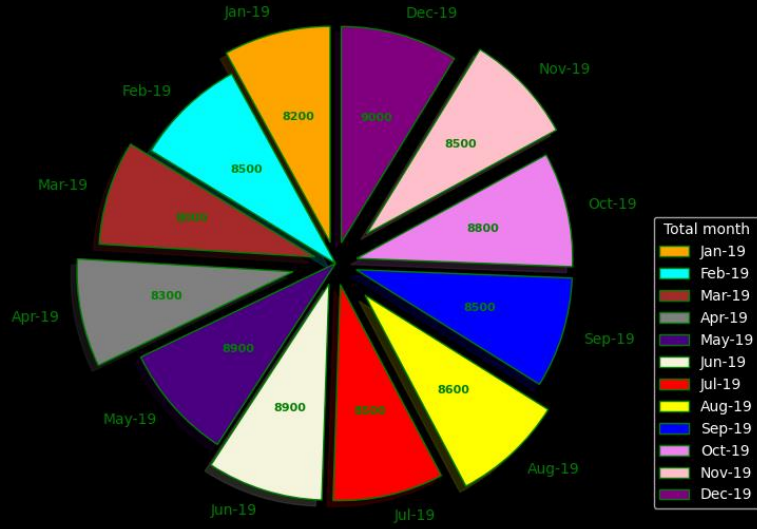
0	2014	1	14-Jan	1300	NaN	NaN
1	2014	2	14-Feb	1600	NaN	NaN
2	2014	3	14-Mar	1600	NaN	NaN
3	2014	4	14-Apr	1500	NaN	NaN
4	2014	5	14-May	1400	NaN	NaN
...
67	2019	8	19-Aug	8600	6949.0	1651.0
68	2019	9	19-Sep	8500	6860.0	1640.0
69	2019	10	19-Oct	8800	7048.0	1716.0
70	2019	11	19-Nov	8500	6821.0	1679.0
71	2019	12	19-Dec	9000	7200.0	1800.0

[72 rows x 6 columns]

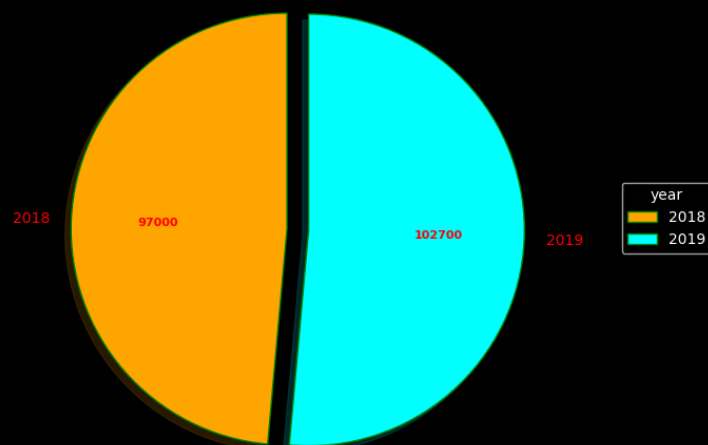
Number of Job Postings of Data Scientists Every Month in 2018

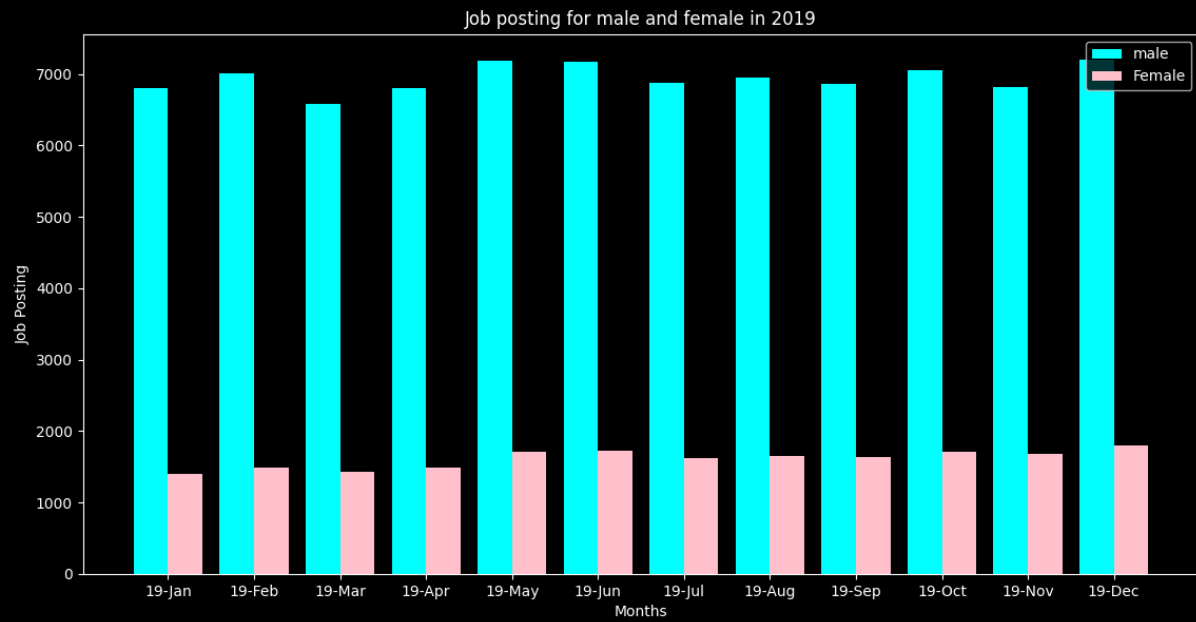
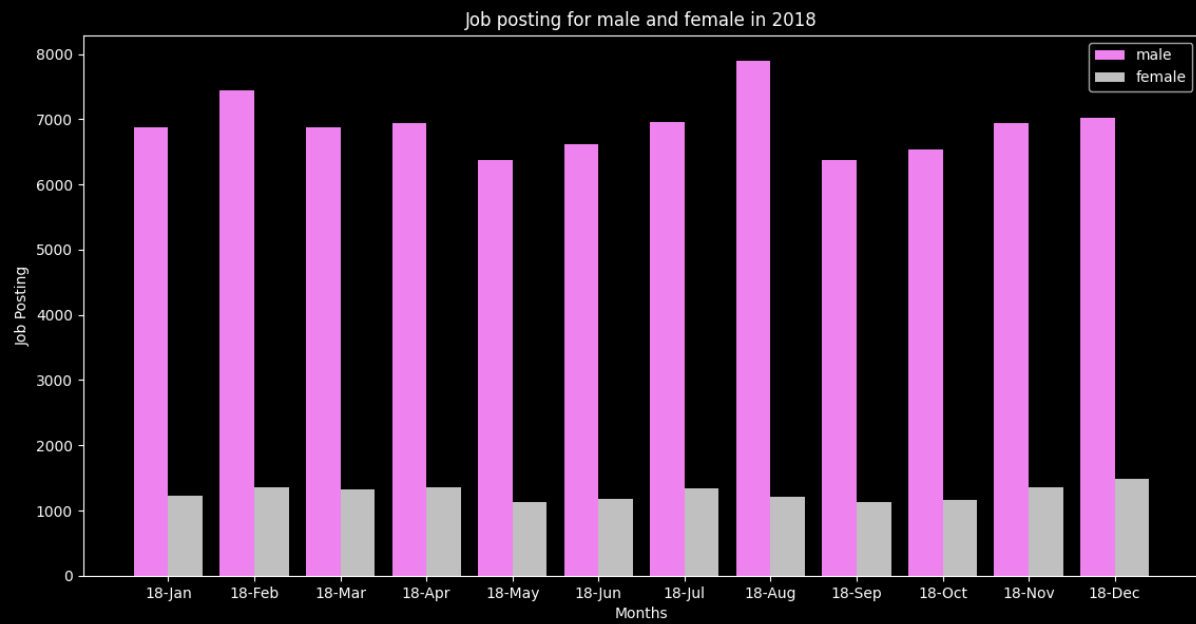


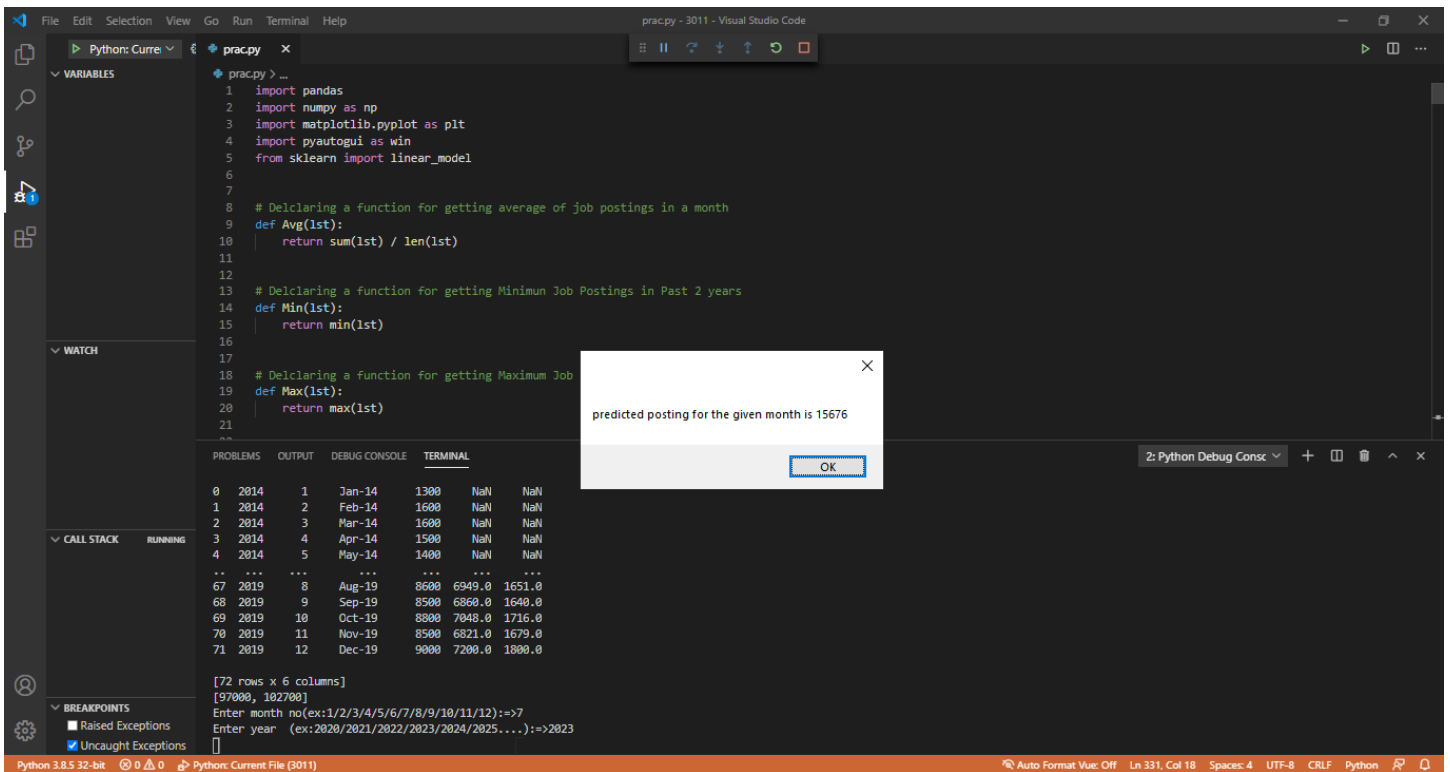
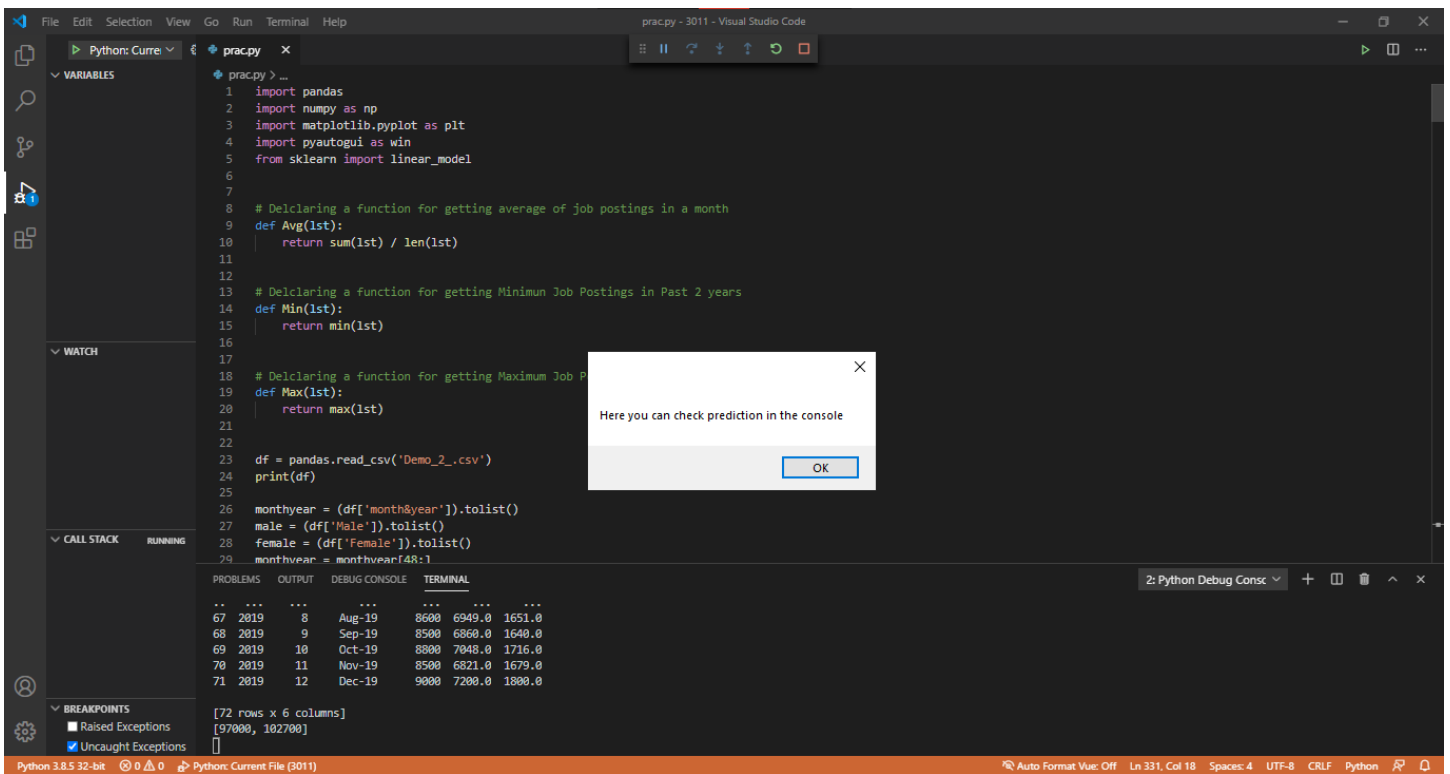
Number of Job Postings of Data Scientists Every Month in 2019



Number of Job Postings of Data Scientists in past 2 year







Thank you.