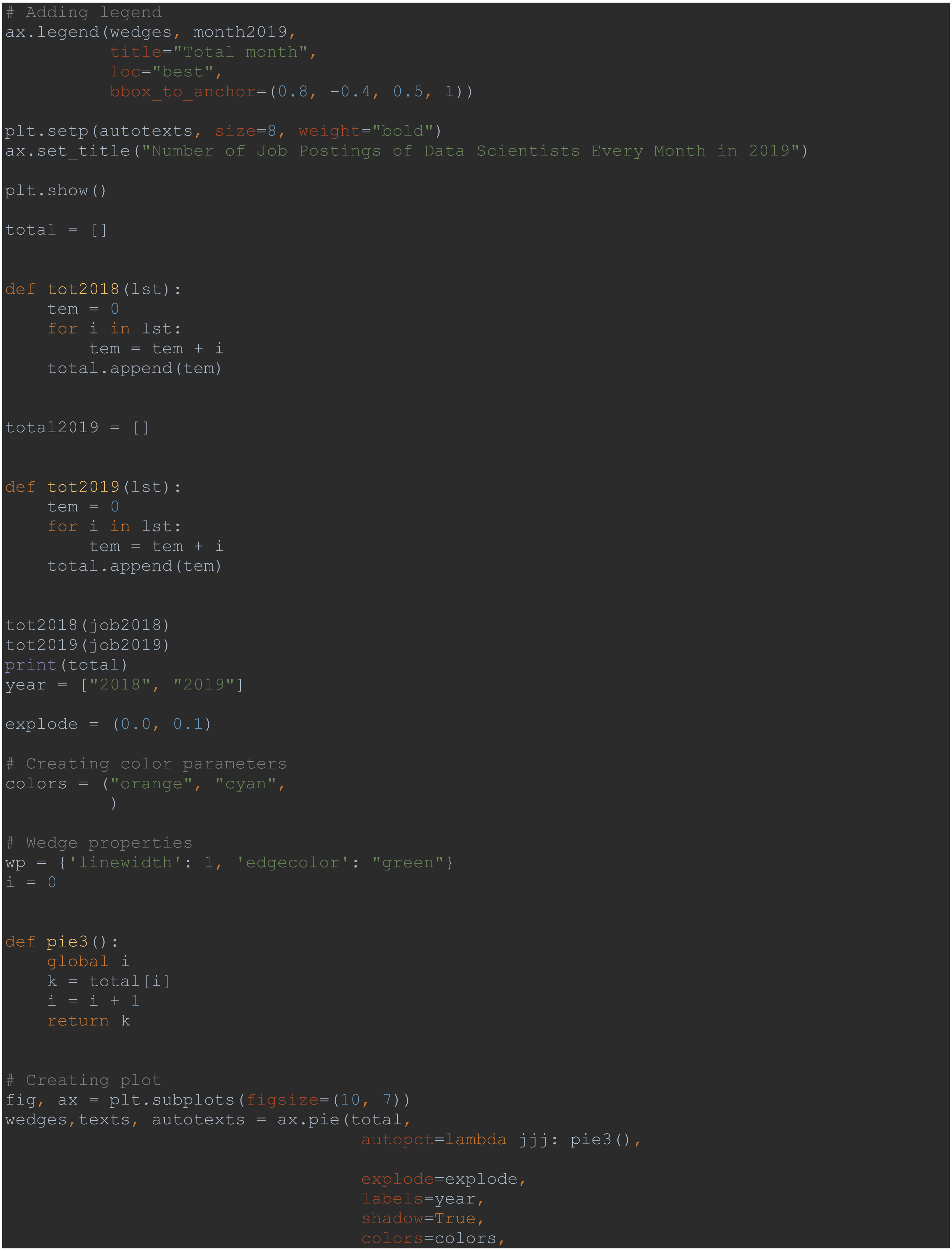
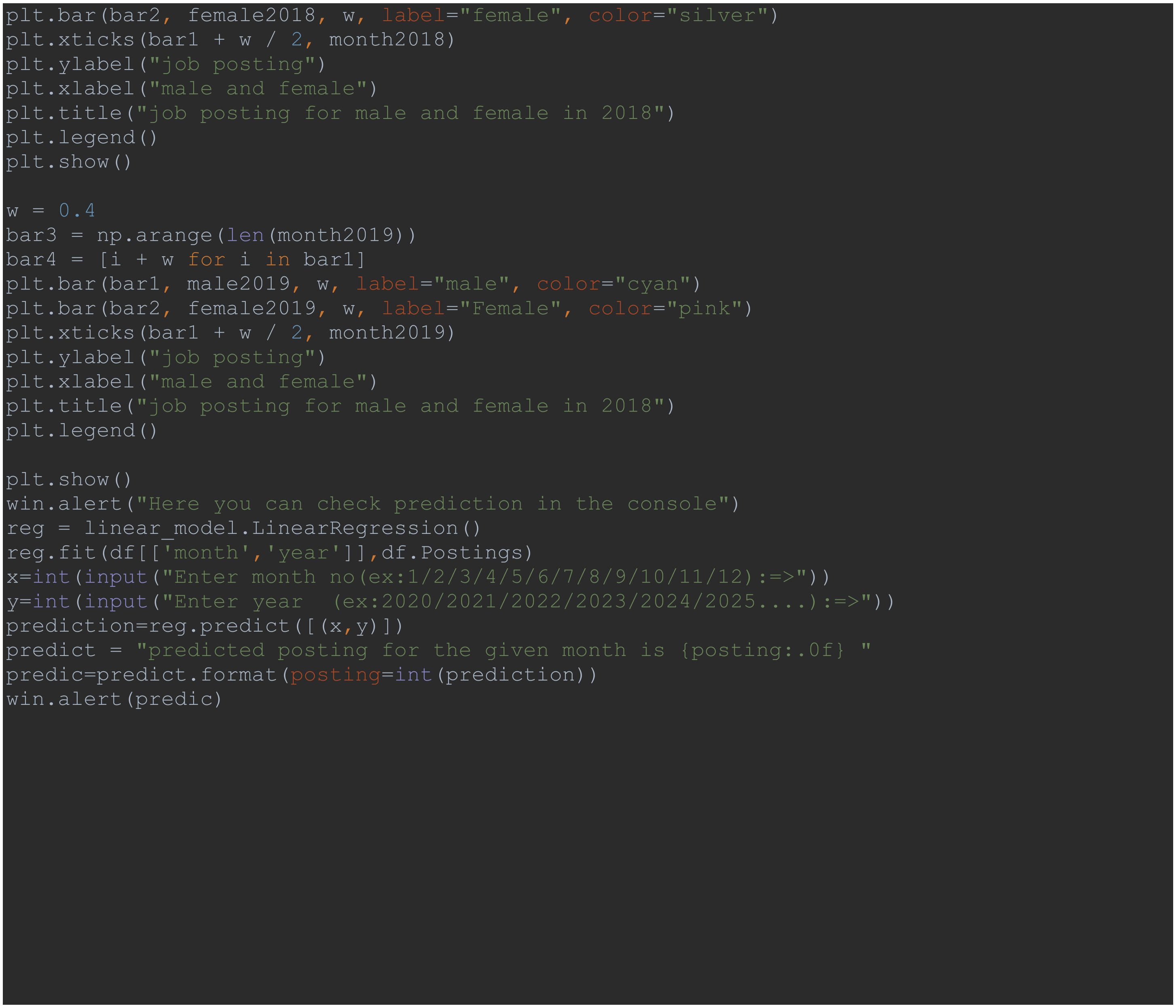
# Analyzing Job Postings in last 2 years and Predicting Job Postings for future year

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


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


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


# Delclaring 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}\nMinimun Job Postings in Past 2 years-
{:.0f} \nMaximum Job Postings in Past 2 years- {:.0f}  ".format(Avg(posting),
Min(posting), 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 mo1(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)


mo1(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",
"purple")

# 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"))
```

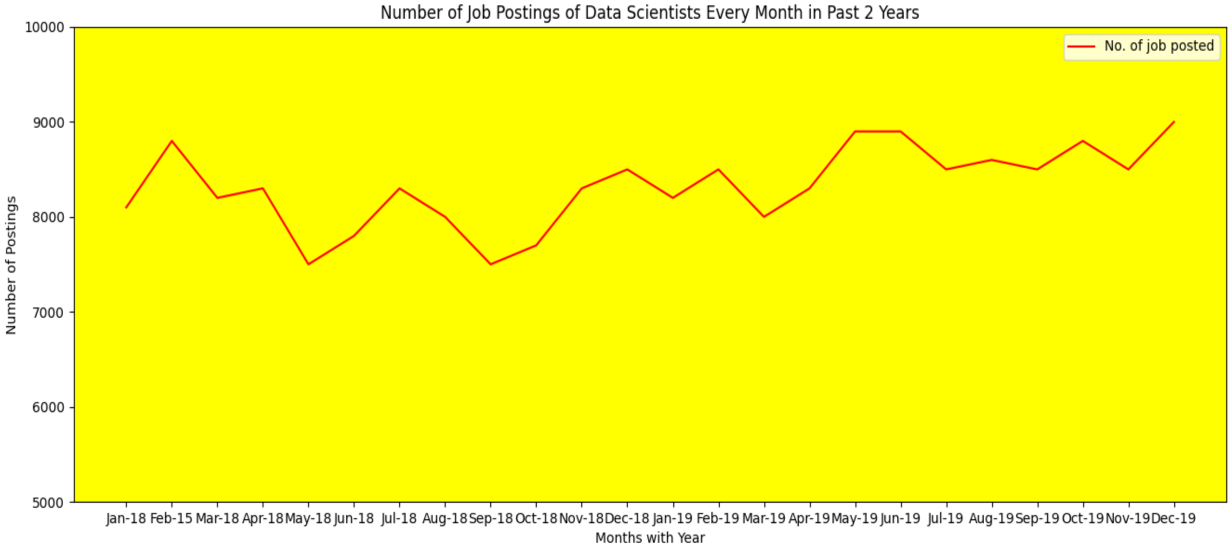


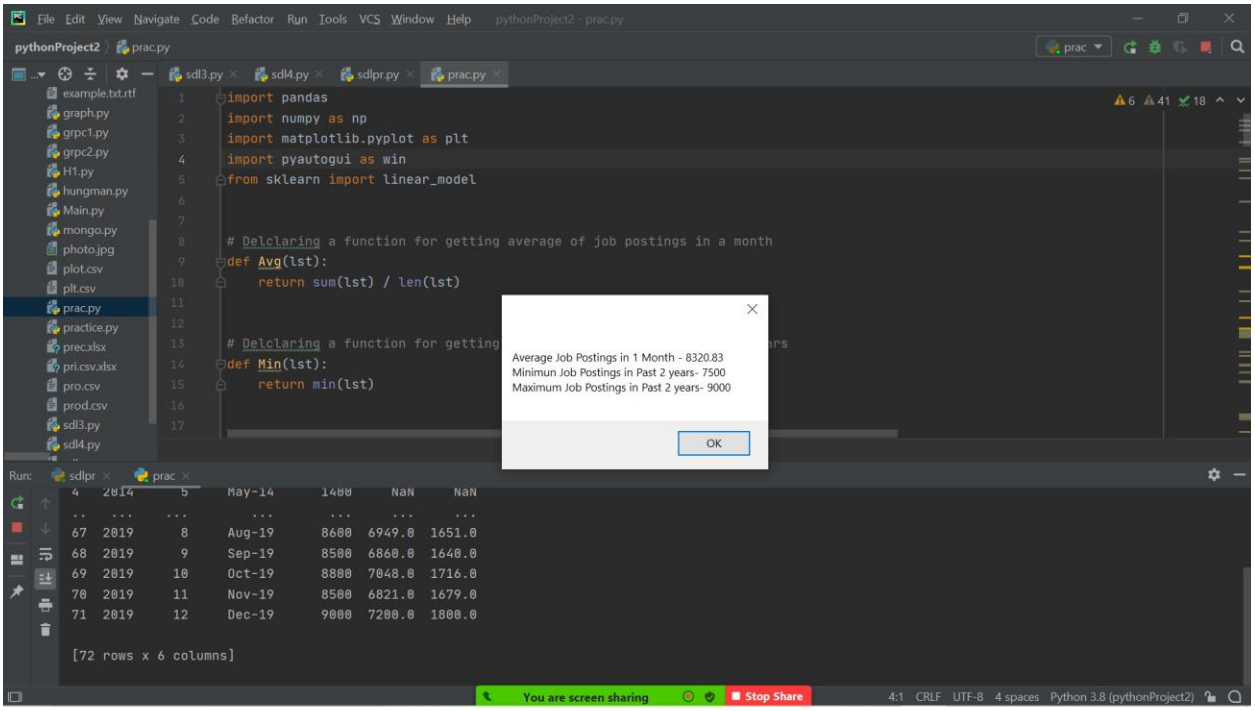
```
# Adding legend
ax.legend(wedges, month2019,
          title="Total month",
          loc="best",
          bbox_to_anchor=(0.8, -0.4, 0.5, 1))

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 of Code :-

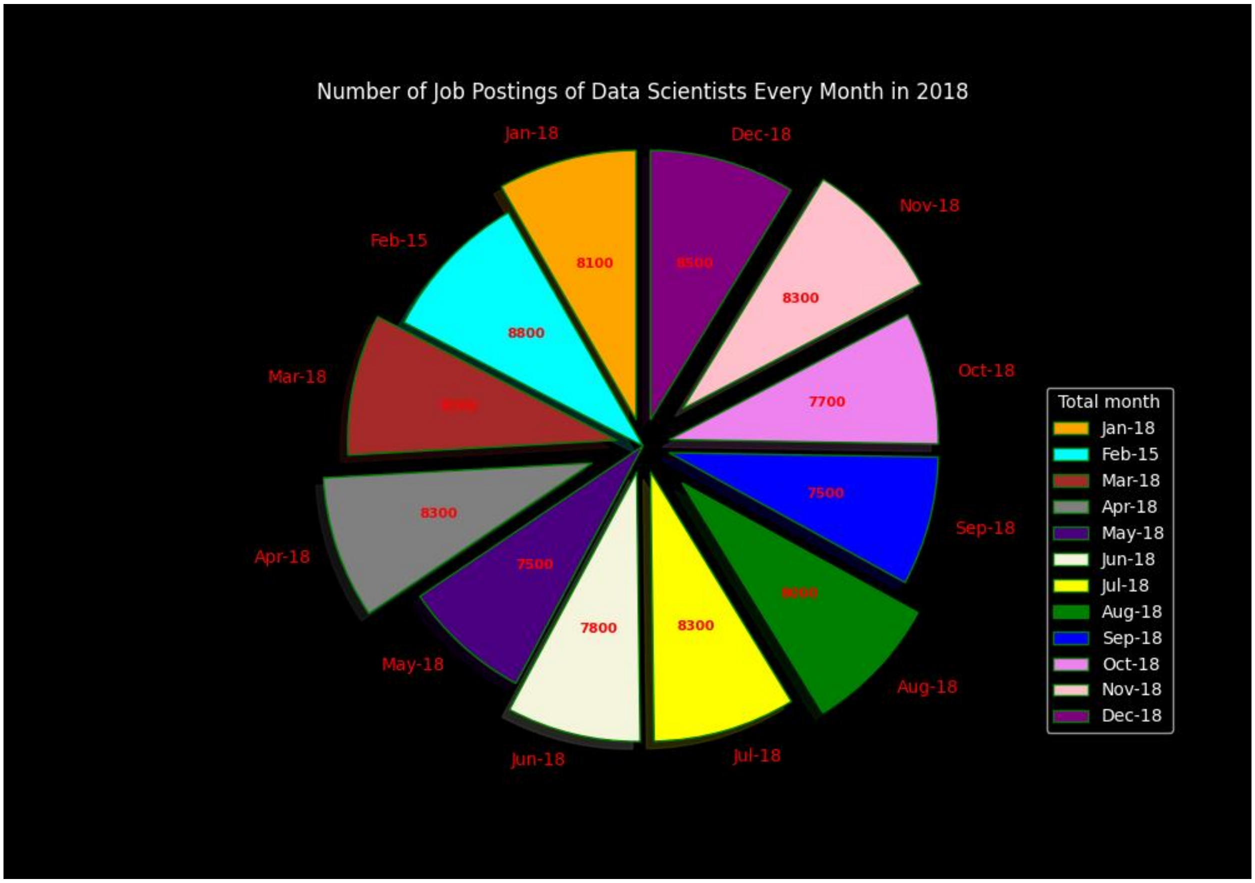


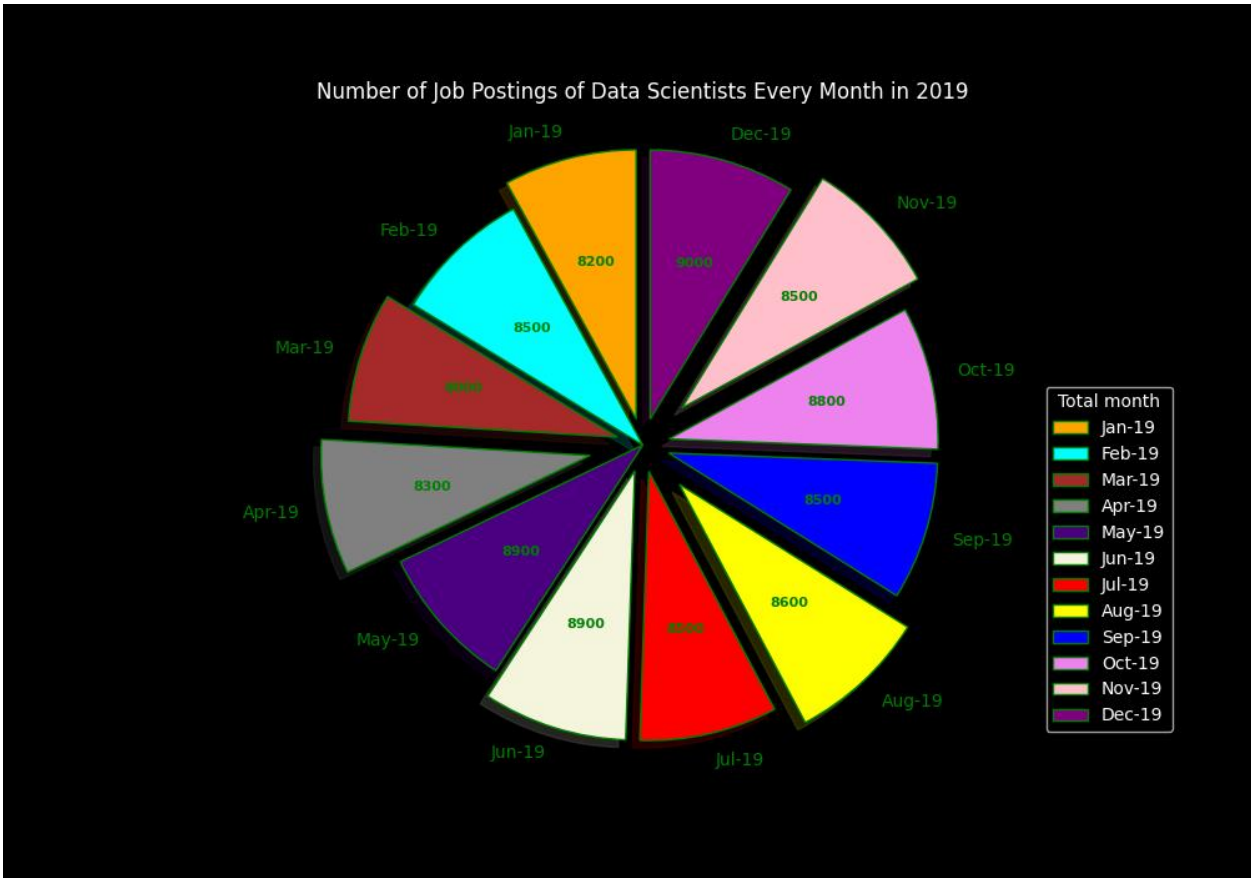


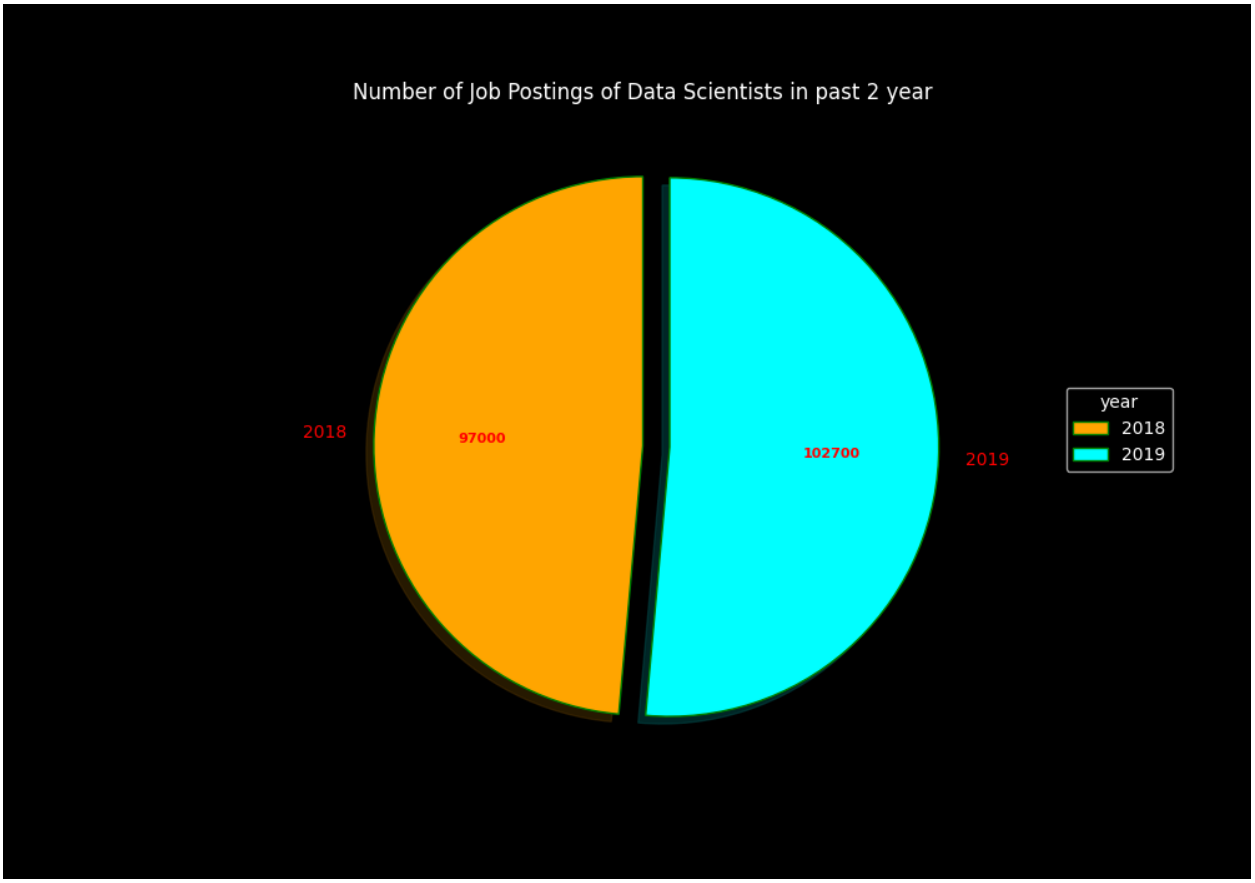
```
import pandas
import numpy as np
import matplotlib.pyplot as plt
import pyautogui as win
from sklearn import linear_model


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


# Delclaring a function for getting                              ars
def Min(lst):
    return min(lst)

```

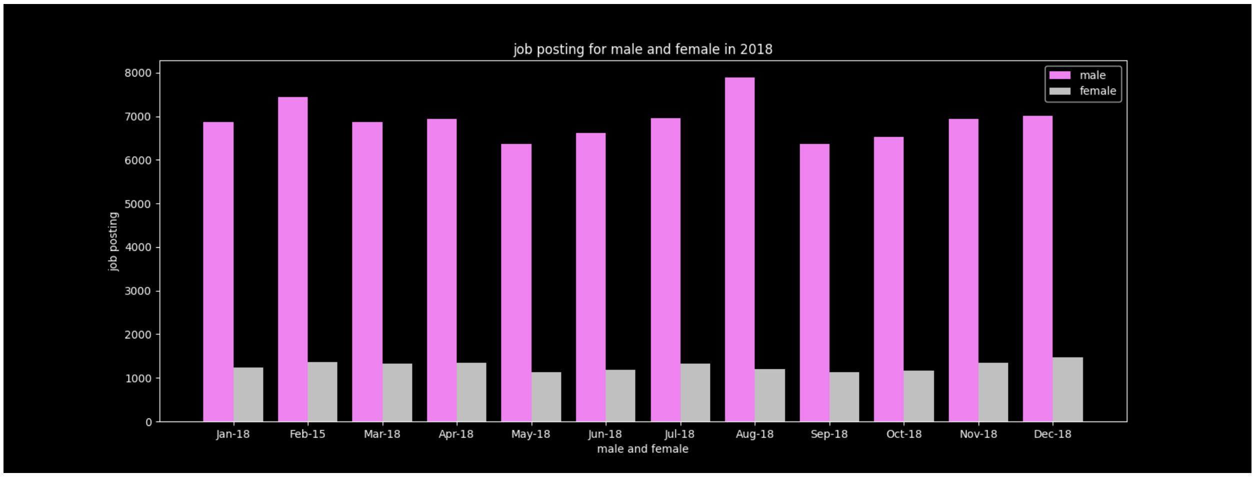
Average Job Postings in 1 Month - 8320.83
Minimun Job Postings in Past 2 years- 7500
Maximum Job Postings in Past 2 years- 9000

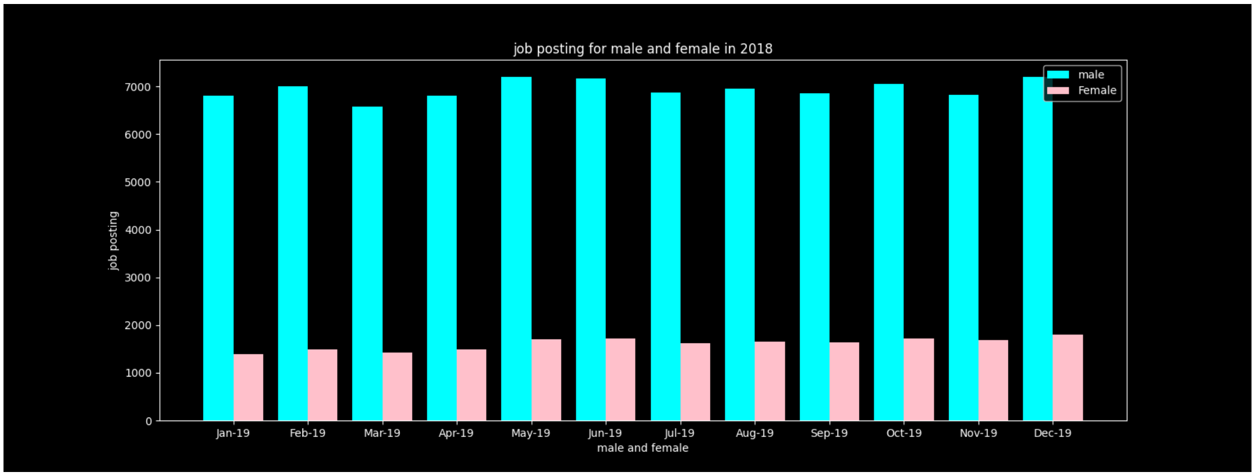
```
4   2014      5     May-14     1400     NaN     NaN
..   ...    ...        ...      ...     ...     ...
67  2019      8     Aug-19     8600  6949.0  1651.0
68  2019      9     Sep-19     8500  6860.0  1640.0
69  2019     10     Oct-19     8800  7048.0  1716.0
70  2019     11     Nov-19     8500  6821.0  1679.0
71  2019     12     Dec-19     9000  7200.0  1800.0

[72 rows x 6 columns]
```

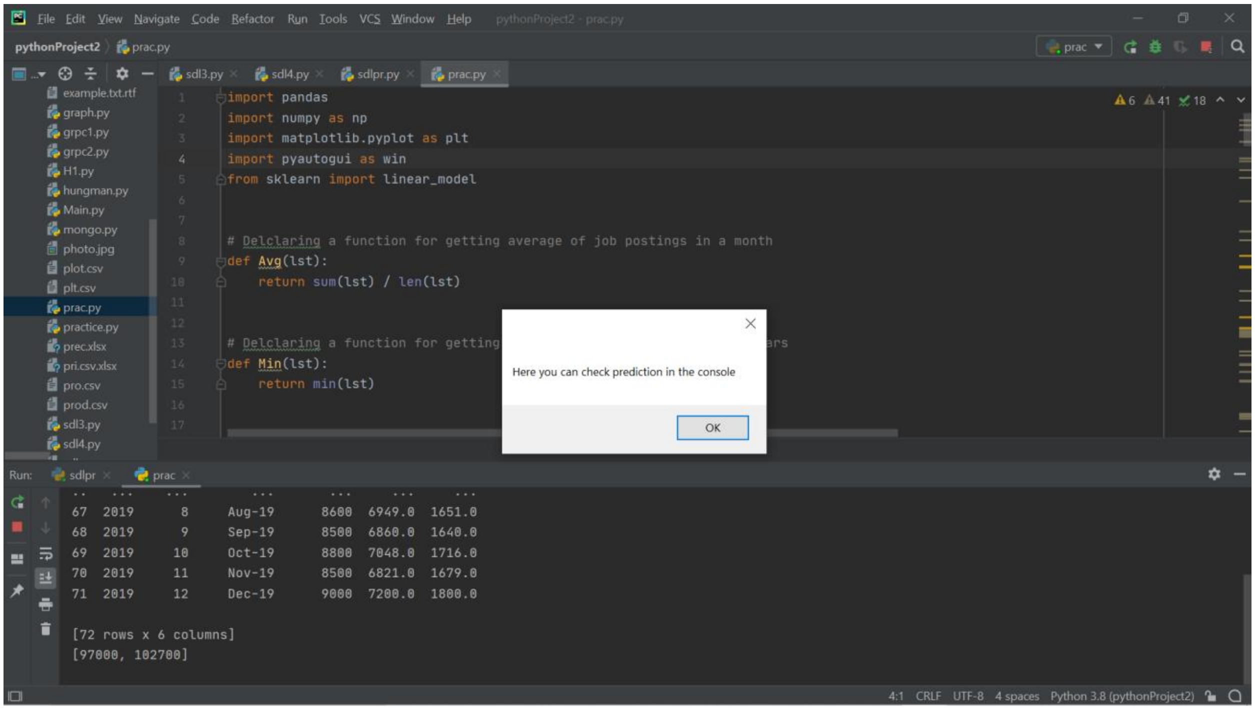

Number of Job Postings of Data Scientists Every Month in 2018
Jan-18
8100
Dec-18
8500
Nov-18
8300
Feb-15
8800
Oct-18
7700
Mar-18
Sep-18
7500
Apr-18
8300
May-18
7500
Aug-18
8000
Jun-18
7800
Jul-18
8300
Total month
Jan-18
Feb-15
Mar-18
Apr-18
May-18
Jun-18
Jul-18
Aug-18
Sep-18
Oct-18
Nov-18
Dec-18


Number of Job Postings of Data Scientists Every Month in 2019
Jan-19
8200
Dec-19
9000
Nov-19
8500
Feb-19
8500
Mar-19
9000
Oct-19
8800
Apr-19
8300
Sep-19
8500
May-19
8900
Jun-19
8900
Jul-19
8500
Aug-19
8600
Total 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


Number of Job Postings of Data Scientists in past 2 year
2018
97000
102700
2019
year
2018
2019


job posting for male and female in 2018
male
female
8000
7000
6000
5000
4000
3000
2000
1000
0
job posting
Jan-18
Feb-15
Mar-18
Apr-18
May-18
Jun-18
Jul-18
Aug-18
Sep-18
Oct-18
Nov-18
Dec-18
male and female


job posting for male and female in 2018
male
Female
7000
6000
5000
4000
3000
2000
1000
0
job posting
Jan-19
Feb-19
Mar-19
Apr-19
May-19
Jun-19
Jul-19
Aug-19
Sep-19
Oct-19
Nov-19
Dec-19
male and female

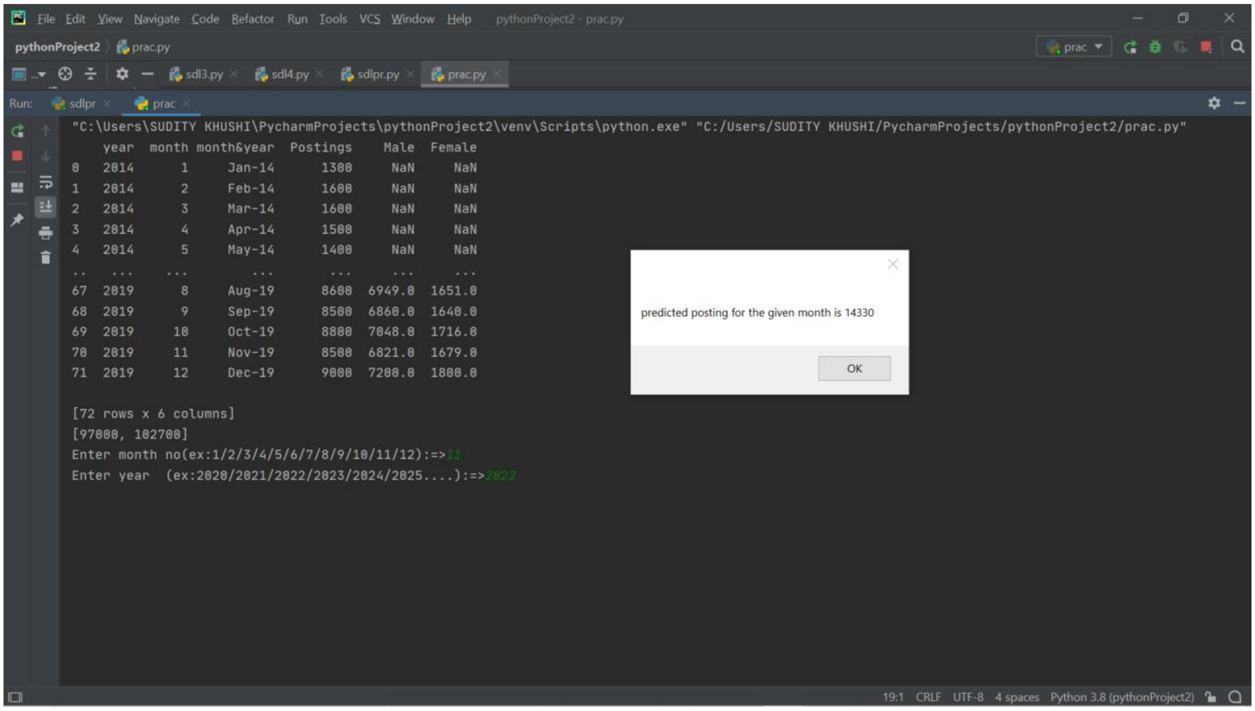


```python
import pandas
import numpy as np
import matplotlib.pyplot as plt
import pyautogui as win
from sklearn import linear_model


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


# Delclaring a function for getting                         ars
def Min(lst):
    return min(lst)
```

Here you can check prediction in the console

OK

```
..   ...    ...        ...       ...     ...     ...
67  2019      8     Aug-19      8600  6949.0  1651.0
68  2019      9     Sep-19      8500  6860.0  1640.0
69  2019     10     Oct-19      8800  7048.0  1716.0
70  2019     11     Nov-19      8500  6821.0  1679.0
71  2019     12     Dec-19      9000  7200.0  1800.0

[72 rows x 6 columns]
[97000, 102700]
```


File Edit View Navigate Code Refactor Run Tools VCS Window Help
pythonProject2 - prac.py
pythonProject2 › prac.py
prac
sdl3.py
sdl4.py
sdlpr.py
prac.py
Run:
sdlpr
prac
19:1 CRLF UTF-8 4 spaces Python 3.8 (pythonProject2)

```
"C:\Users\SUDITY KHUSHI\PycharmProjects\pythonProject2\venv\Scripts\python.exe" "C:/Users/SUDITY KHUSHI/PycharmProjects/pythonProject2/prac.py"
    year  month month&year  Postings    Male  Female
0   2014      1     Jan-14      1300     NaN     NaN
1   2014      2     Feb-14      1600     NaN     NaN
2   2014      3     Mar-14      1600     NaN     NaN
3   2014      4     Apr-14      1500     NaN     NaN
4   2014      5     May-14      1400     NaN     NaN
..   ...    ...        ...       ...     ...     ...
67  2019      8     Aug-19      8600  6949.0  1651.0
68  2019      9     Sep-19      8500  6860.0  1640.0
69  2019     10     Oct-19      8800  7048.0  1716.0
70  2019     11     Nov-19      8500  6821.0  1679.0
71  2019     12     Dec-19      9000  7200.0  1800.0

[72 rows x 6 columns]
[97000, 102700]
Enter month no(ex:1/2/3/4/5/6/7/8/9/10/11/12):=>11
Enter year  (ex:2020/2021/2022/2023/2024/2025....):=>2022
```

predicted posting for the given month is 14330

OK