Experiment Number 7

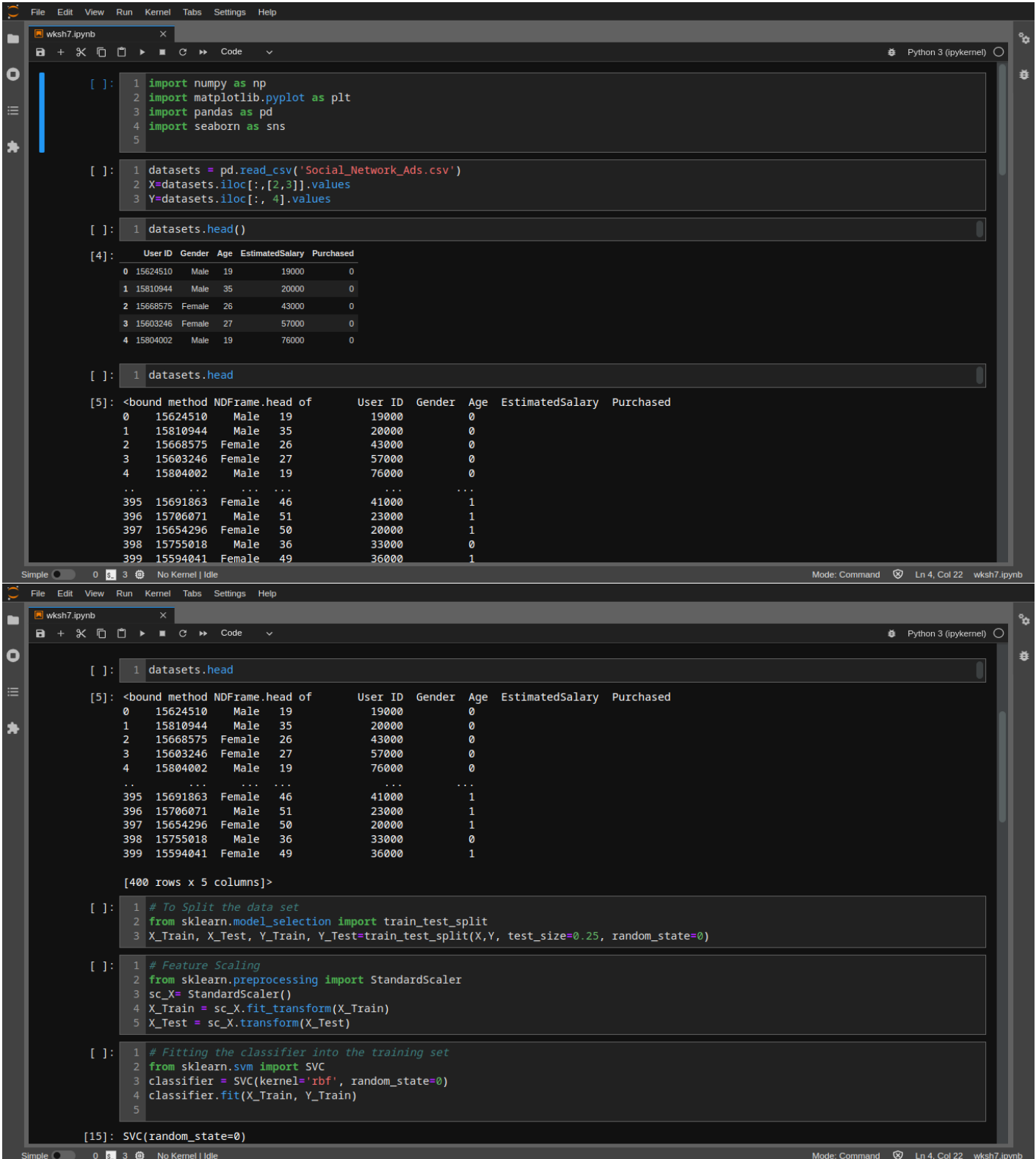|			
Name ::	Rishabh Anand	UID ::	19BCS4525
Branch ::	CSE - IoT	Sec/Grp ::	1/A
Semester ::	6th	Date ::	9th Apr, 2022
Subject ::	ML Lab	CODE ::	CSD-386

1. Aim :

To implement SVM Technique using Python programming Language.

2. Task :

1. Install the prerequisite Python.

2. Load the dataset.

3. Apply SVM Technique.

4. Analyse results/output.

4. Steps:

```python
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

```python
datasets = pd.read_csv('Social_Network_Ads.csv')
X=datasets.iloc[:,[2,3]].values
Y=datasets.iloc[:, 4].values
```

```python
datasets.head()
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	Male	19	19000	0
1	15810944	Male	35	20000	0
2	15668575	Female	26	43000	0
3	15603246	Female	27	57000	0
4	15804002	Male	19	76000	0

```python
datasets.head
```

```
[5]: <bound method NDFrame.head of        User ID  Gender  Age  EstimatedSalary  Purchased
0    15624510    Male   19            19000          0
1    15810944    Male   35            20000          0
2    15668575  Female   26            43000          0
3    15603246  Female   27            57000          0
4    15804002    Male   19            76000          0
..        ...     ...  ...              ...        ...
395  15691863  Female   46            41000          1
396  15706071    Male   51            23000          1
397  15654296  Female   50            20000          1
398  15755018    Male   36            33000          0
399  15594041  Female   49            36000          1

[400 rows x 5 columns]>
```

```python
# To Split the data set
from sklearn.model_selection import train_test_split
X_Train, X_Test, Y_Train, Y_Test=train_test_split(X,Y, test_size=0.25, random_state=0)
```

```python
# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc_X= StandardScaler()
X_Train = sc_X.fit_transform(X_Train)
X_Test = sc_X.transform(X_Test)
```

```python
# Fitting the classifier into the training set
from sklearn.svm import SVC
classifier = SVC(kernel='rbf', random_state=0)
classifier.fit(X_Train, Y_Train)
```

```
[15]: SVC(random_state=0)
```

File   Edit   View   Run   Kernel   Tabs   Settings   Help

wksh7.ipynb

Code ∨   Python 3 (ipykernel) ○

[15]: SVC(random_state=0)

```python
# Predicting the test set results
Y_Pred = classifier.predict(X_Test)
```

```python
# To check accuracy
from sklearn.metrics import classification_report
print(classification_report(Y_Test, Y_Pred))
```

```
              precision    recall  f1-score   support

           0       0.96      0.94      0.95        68
           1       0.88      0.91      0.89        32

    accuracy                           0.93       100
   macro avg       0.92      0.92      0.92       100
weighted avg       0.93      0.93      0.93       100
```

```python
from sklearn.metrics import accuracy_score
print(accuracy_score(Y_Test, Y_Pred))
```
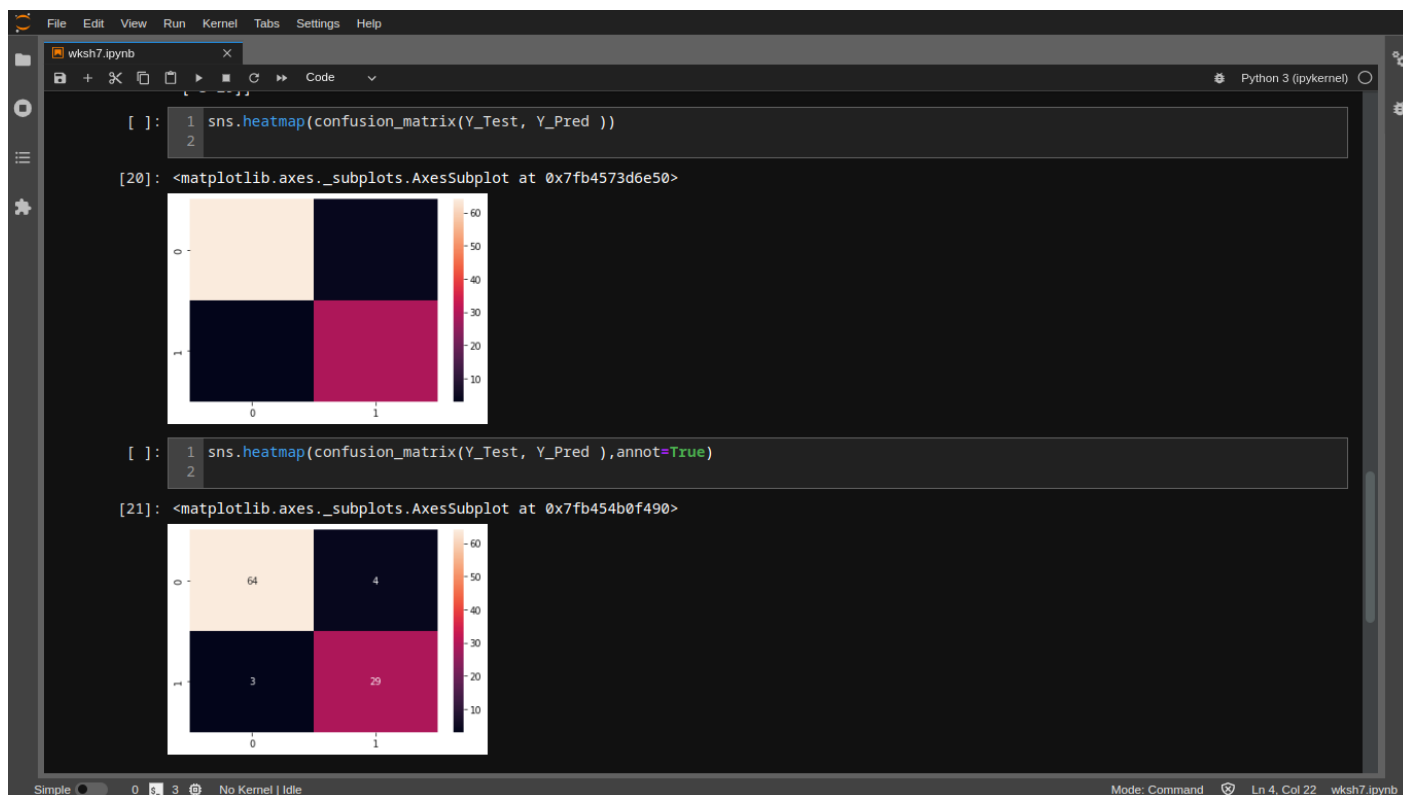
```
0.93
```

```python
# Visulize data by Confusion Matrix
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(Y_Test, Y_Pred )
print(cm)
```

```
[[64  4]
 [ 3 29]]
```

```python
sns.heatmap(confusion_matrix(Y_Test, Y_Pred ))

```

Simple ●   0 ■ 3 ⊕   No Kernel | Idle   Mode: Command ⊘   Ln 4, Col 22   wksh7.ipynb

5. Observations :


```
File   Edit   View   Run   Kernel   Tabs   Settings   Help

wksh7.ipynb

                                    Code                                              Python 3 (ipykernel)

[ ]:   1  sns.heatmap(confusion_matrix(Y_Test, Y_Pred ))
       2

[20]:  <matplotlib.axes._subplots.AxesSubplot at 0x7fb4573d6e50>
```

```
[ ]:   1  sns.heatmap(confusion_matrix(Y_Test, Y_Pred ),annot=True)
       2

[21]:  <matplotlib.axes._subplots.AxesSubplot at 0x7fb454b0f490>
```

Simple   0   3   No Kernel | Idle   Mode: Command   Ln 4, Col 22   wksh7.ipynb

Learning Outcomes :

- Implementing SVM Technique using Python Programming.

S. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			