

```
[1]: import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

```
[2]: data=pd.read_csv('Social_Network_Ads.csv')
print(data)
```

	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]

```
[3]: data["Gender"]=data["Gender"].replace({"Male":0,"Female":1})
```

C:\Users\DELL\AppData\Local\Temp\ipykernel_9228\483096724.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.no_silent_downcasting', True)`

```
data["Gender"]=data["Gender"].replace({"Male":0,"Female":1})
```

```
[4]: data
```

```
[4]: data
```

```
[4]:
```

	User ID	Gender	Age	EstimatedSalary	Purchased
0	15624510	0	19	19000	0
1	15810944	0	35	20000	0
2	15668575	1	26	43000	0
3	15603246	1	27	57000	0
4	15804002	0	19	76000	0
...
395	15691863	1	46	41000	1
396	15706071	0	51	23000	1
397	15654296	1	50	20000	1
398	15755018	0	36	33000	0
399	15594041	1	49	36000	1

400 rows × 5 columns

```
[7]: data=data.drop(columns=["User ID"])
print(data)
```

	Age	EstimatedSalary	Purchased	Gender_Male
0	19	19000	0	True
1	35	20000	0	True
2	26	43000	0	False

```
[7]: data=data.drop(columns=["User ID"])
print(data)
```

	Age	EstimatedSalary	Purchased	Gender_Male
0	19	19000	0	True
1	35	20000	0	True
2	26	43000	0	False
3	27	57000	0	False
4	19	76000	0	True
..
395	46	41000	1	False
396	51	23000	1	True
397	50	20000	1	False
398	36	33000	0	True
399	49	36000	1	False

[400 rows x 4 columns]

```
[8]: independent=data[["Age","EstimatedSalary","Gender_Male"]]
dependent=data[["Purchased"]]
```

```
[9]: from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(independent,dependent,test_size=0.2,random_state=42)
```

```
[10]: from sklearn.svm import SVC
Classifier=SVC(kernel='rbf',random_state=0)
Classifier.fit(X_train,Y_train)
```

C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\utils\validation.py:1339: DataConversionWarning: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples,), for example using ravel().
y = column_or_1d(y, warn=True)

```
[10]: SVC
```

```
y = column_or_1d(y, warn=True)

[10]: SVC
      SVC(random_state=0)

[11]: Y_pred=Classifier.predict(X_test)

[12]: from sklearn.metrics import confusion_matrix
      cm=confusion_matrix(Y_test,Y_pred)
      print(cm)

[[49  3]
 [18 10]]

[13]: Age=int(input("enter the prediction input value:"))
      EstimatedSalary=int(input("enter the prediction input value:"))
      Gender_Male=int(input("enter the prediction input value:"))
      future_prediction=Classifier.predict([[Age,EstimatedSalary,Gender_Male]])
      print("future_prediction={Purchased=0,Non Purchased=1}",format(future_prediction))

enter the prediction input value: 46
enter the prediction input value: 41000
enter the prediction input value: 0
future_prediction={Purchased=0,Non Purchased=1} [0]

C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but SVC was fitted with feature names
warnings.warn(
```