

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

```
In [2]: ads = pd.read_csv('Social_Network_Ads.csv')
ads.head()
```

Out[2]:

	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

```
In [3]: ads.shape
```

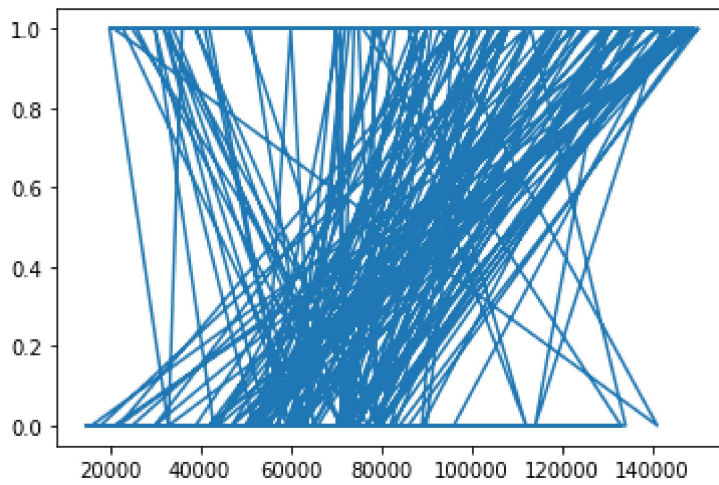
Out[3]: (400, 5)

```
In [4]: ads.describe()
```

Out[4]:

	User ID	Age	EstimatedSalary	Purchased
<b>count</b>	4.000000e+02	400.000000	400.000000	400.000000
<b>mean</b>	1.569154e+07	37.655000	69742.500000	0.357500
<b>std</b>	7.165832e+04	10.482877	34096.960282	0.479864
<b>min</b>	1.556669e+07	18.000000	15000.000000	0.000000
<b>25%</b>	1.562676e+07	29.750000	43000.000000	0.000000
<b>50%</b>	1.569434e+07	37.000000	70000.000000	0.000000
<b>75%</b>	1.575036e+07	46.000000	88000.000000	1.000000
<b>max</b>	1.581524e+07	60.000000	150000.000000	1.000000

```
In [5]: xpoints = np.array(ads['EstimatedSalary'])
ypoints = np.array(ads['Purchased'])
plt.plot(xpoints, ypoints)
plt.show()
```



```
In [6]: X = ads.iloc[:,[2,3]].values
y = ads.iloc[:,4].values
```

```
In [7]: 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)
```

```
In [8]: from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
In [12]: from sklearn.svm import SVC
classifier_rbf = SVC (random_state = 0) #by default (kernal = 'rbf',random_state=0)
classifier_rbf.fit(X_train, y_train)
y_pred_rbf = classifier_rbf.predict(X_test)
```

```
In [13]: from sklearn.metrics import confusion_matrix
cm_rbf = confusion_matrix(y_test,y_pred_rbf)
print(cm_rbf)
```

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

```
In [14]: from sklearn .metrics import classification_report
class_report_rbf = classification_report (y_test,y_pred_rbf)
print(class_report_rbf)
```

	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

```
In [ ]:
```