

```
In [55]: import pandas as pd
df = pd.read_csv('Social_Network_Ads.csv')
df.head()
```

```
Out[55]:
```

	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 [56]: df.describe
```

```
Out[56]: <bound method NDFrame.describe of
User ID  Gender  Age  EstimatedS
alary  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]>
```

```
In [57]: df.shape
```

```
Out[57]: (400, 5)
```

```
In [58]: x=df.iloc[:,2:4]
```

```
In [59]: y=df.iloc[:,4]
```

```
In [60]: x.head()
```

```
Out[60]:
```

	Age	EstimatedSalary
0	19	19000
1	35	20000
2	26	43000
3	27	57000
4	19	76000

```
In [61]: x.describe
```

```
Out[61]: <bound method NDFrame.describe of      Age  EstimatedSalary
0      19          19000
1      35          20000
2      26          43000
3      27          57000
4      19          76000
...    ...          ...
395    46          41000
396    51          23000
397    50          20000
398    36          33000
399    49          36000

[400 rows x 2 columns]>
```

```
In [62]: x.shape
```

```
Out[62]: (400, 2)
```

```
In [63]: y.head()
```

```
Out[63]: 0      0
1      0
2      0
3      0
4      0
Name: Purchased, dtype: int64
```

```
In [64]: y.shape
```

```
Out[64]: (400,)
```

```
In [ ]:
```

```
In [105... 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 [106... x_train.shape
```

```
Out[106... (300, 2)
```

```
In [107... x_test.shape
```

```
Out[107... (100, 2)
```

```
In [108... from sklearn.preprocessing import StandardScaler
cs = StandardScaler()

x_train = cs.fit_transform(x_train)
x_test = cs.fit_transform(x_test)
```

```
In [109... from sklearn.svm import SVC
```

```
In [110... svc = SVC(kernel='linear',random_state=0)
svc.fit(x_train, y_train)
```

Out[110...

▼ SVC

SVC(kernel='linear', random\_state=0)

In [111... `y_pred = svc.predict(x_test)`In [112... `from sklearn.metrics import classification_report`In [113... `print(classification_report(y_test,y_pred))`

	precision	recall	f1-score	support
0	0.90	0.93	0.91	68
1	0.83	0.78	0.81	32
accuracy			0.88	100
macro avg	0.87	0.85	0.86	100
weighted avg	0.88	0.88	0.88	100

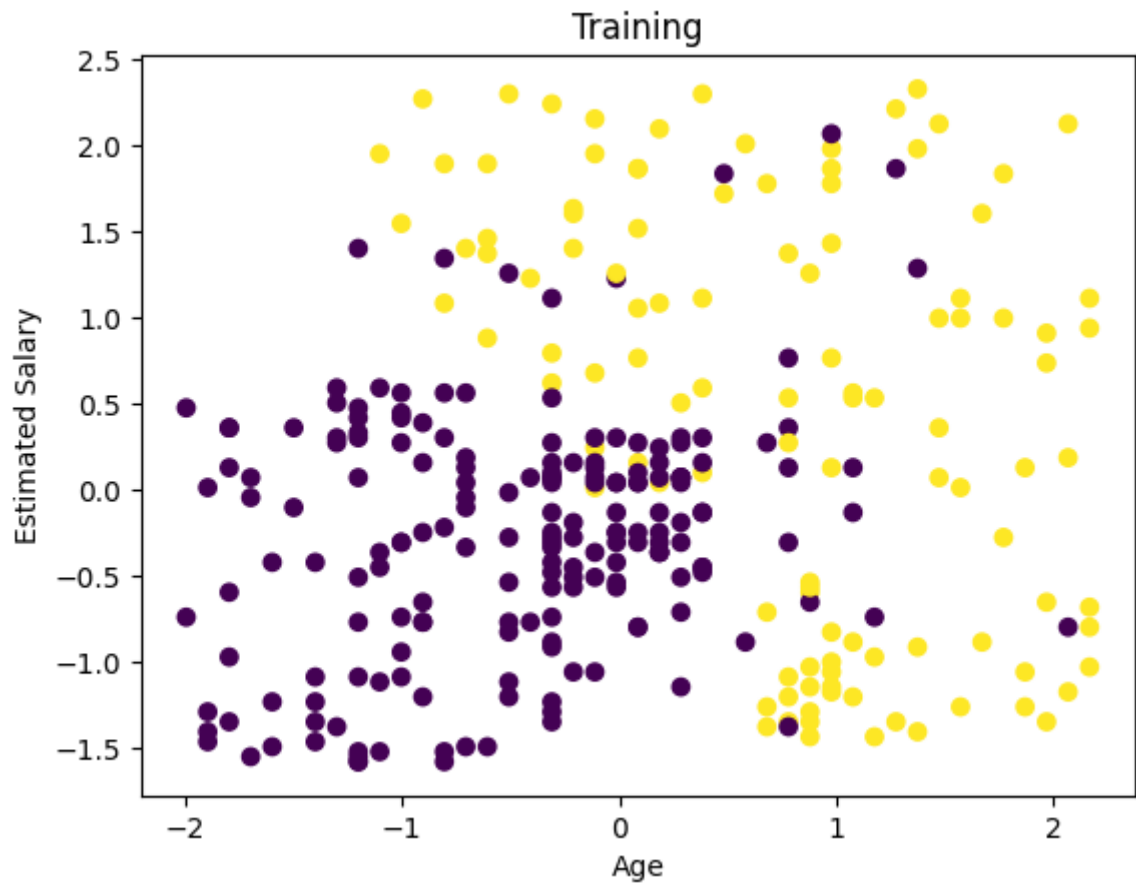
In [114... `svc = SVC(kernel='rbf',random_state=0)`  
`svc.fit(x_train, y_train)`  
`y_pred = svc.predict(x_test)`  
`print(classification_report(y_test,y_pred))`

	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

In [115... `svc = SVC(kernel='poly',random_state=0)`  
`svc.fit(x_train, y_train)`  
`y_pred = svc.predict(x_test)`  
`print(classification_report(y_test,y_pred))`

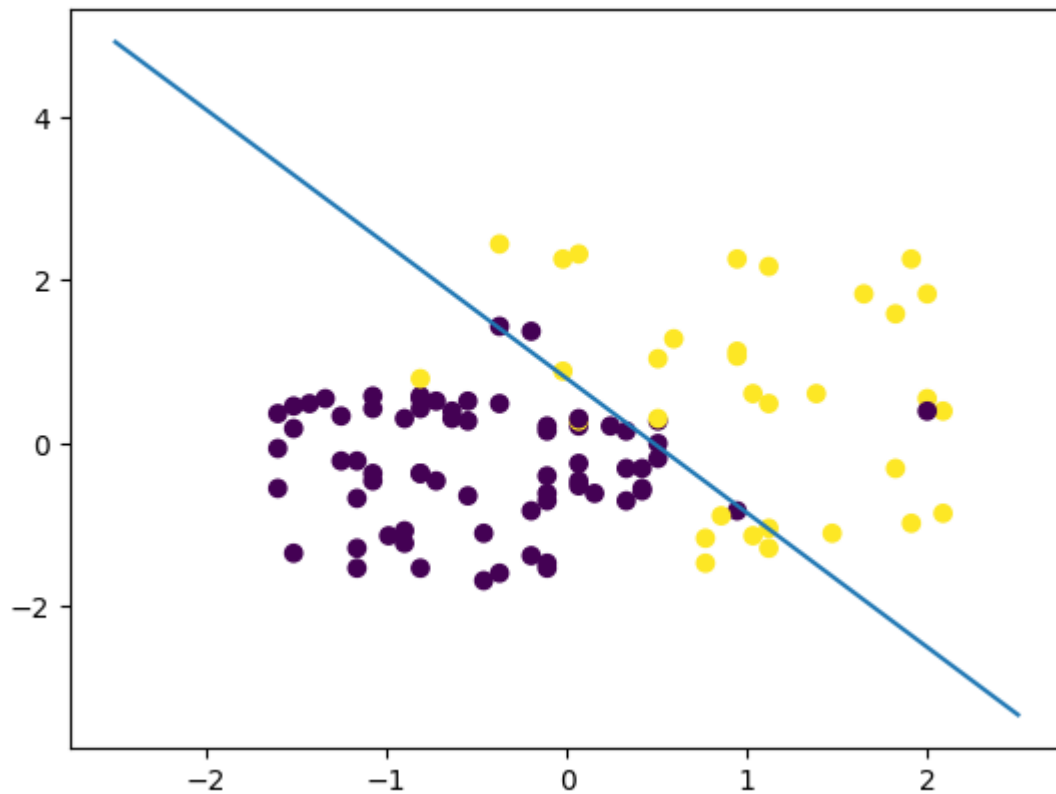
	precision	recall	f1-score	support
0	0.84	0.94	0.89	68
1	0.83	0.62	0.71	32
accuracy			0.84	100
macro avg	0.84	0.78	0.80	100
weighted avg	0.84	0.84	0.83	100

In [116... `import matplotlib.pyplot as plt`  
`plt.scatter(x_train[:,0],x_train[:,1],c=y_train)`  
`plt.xlabel("Age")`  
`plt.ylabel("Estimated Salary")`  
`plt.title('Training')`  
`plt.show()`



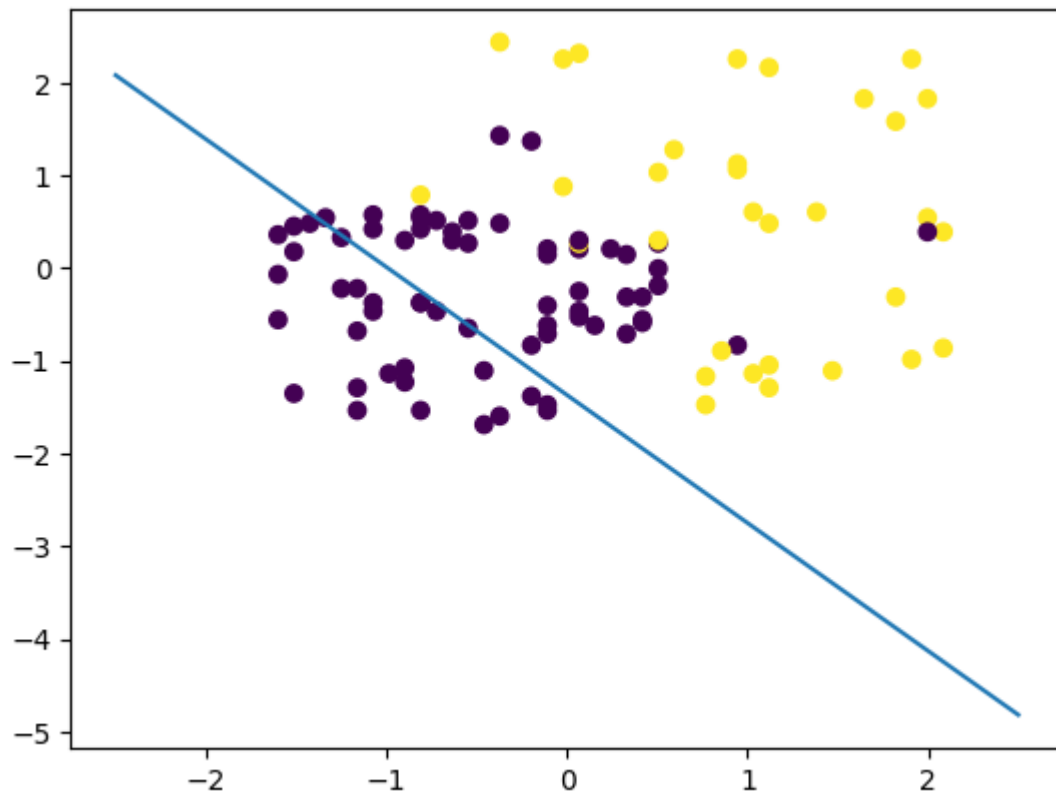
```
In [117...] svc = SVC(kernel='linear', random_state=0)
            svc.fit(x_train, y_train)
            y_pred = svc.predict(x_test)
```

```
In [118...] import numpy as np
            plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
            w=svc.coef_[0]
            a=-w[0]/w[1]
            xx=np.linspace(-2.5, 2.5)
            yy=a*xx-(svc.intercept_[0]/w[1])
            plt.plot(xx, yy)
            plt.show()
```



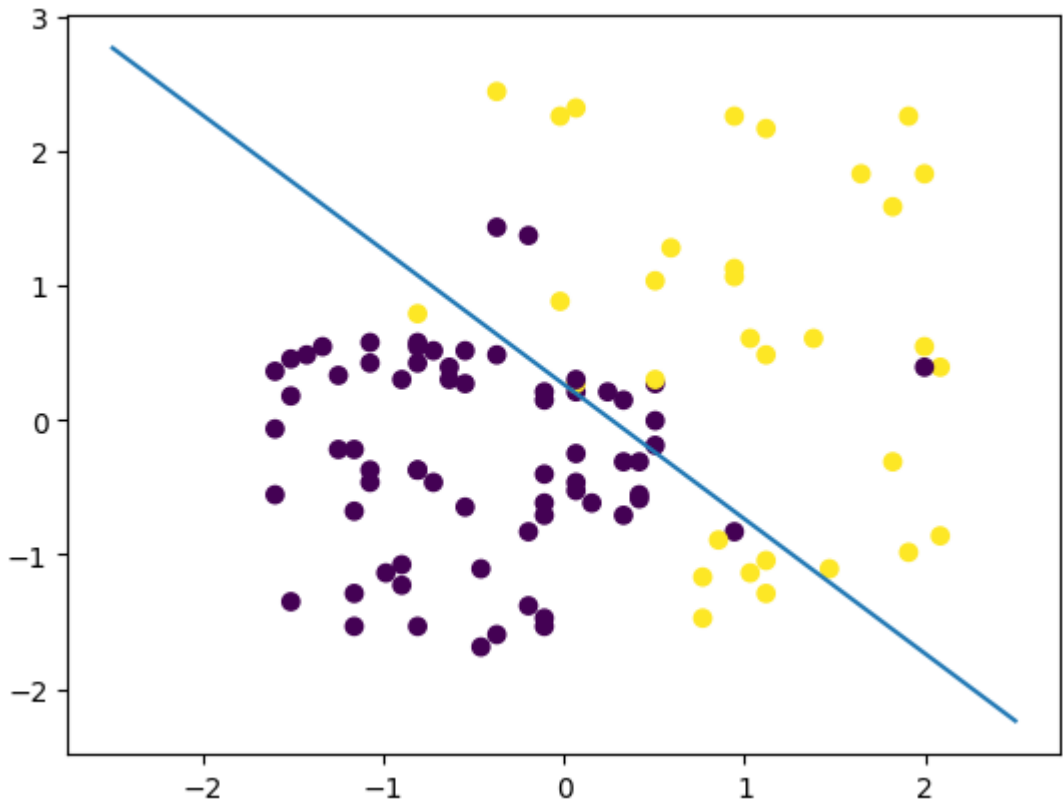
```
In [121... svc = SVC(kernel='poly', random_state=0)
svc.fit(x_train, y_train)
y_pred = svc.predict(x_test)

plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
w=svc.dual_coef_[0]
a=-w[0]/w[1]
xx=np.linspace(-2.5, 2.5)
yy=a*xx-(svc.intercept_[0]/w[1])
plt.plot(xx, yy)
plt.show()
```



```
In [122... svc = SVC(kernel='rbf', random_state=0)
svc.fit(x_train, y_train)
y_pred = svc.predict(x_test)

plt.scatter(x_test[:,0], x_test[:,1], c=y_test)
w=svc.dual_coef_[0]
a=-w[0]/w[1]
xx=np.linspace(-2.5, 2.5)
yy=a*xx-(svc.intercept_[0]/w[1])
plt.plot(xx, yy)
plt.show()
```



```
In [ ]:
```