In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
dataset = pd.read_csv("Social_Network_Ads.csv")
```

In [3]:

```
dataset.head()
```

Out[3]:

	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 [4]:

```
dataset.isnull().sum()
```

Out[4]:

```
User ID            0
Gender             0
Age                0
EstimatedSalary    0
Purchased          0
dtype: int64
```

In [5]:

```
mapi = {'Male':1, 'Female':0}
dataset = dataset.replace(mapi)
dataset.head()
```

Out[5]:

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

In [6]:

```
dataset.drop(['User ID'], axis=1, inplace=True)
dataset.head()
```

Out[6]:

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

In [7]:

```
x,y = dataset.drop(['Purchased'], axis=1), dataset ['Purchased']
```

In [9]:

```
from sklearn.model_selection import train_test_split
xtrain, xtest, ytrain, ytest = train_test_split(x,y, test_size = 0.25, random_state = 0)
```

In [10]:

```
from sklearn.preprocessing import StandardScaler
sc_scale = StandardScaler()
```

In [11]:

```
xtrain = sc_scale.fit_transform(xtrain)
xtest = sc_scale.transform(xtest)
```

In [13]:

```
from sklearn.naive_bayes import GaussianNB
classifier = GaussianNB()
```

classifier.fit(xtrain, ytrain)

In [14]:

```
classifier.fit(xtrain, ytrain)
```

Out[14]:

```
▾ GaussianNB

GaussianNB()
```

In [15]:

```
y_pred = classifier.predict(xtest)
```

In [16]:

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(ytest, y_pred)
```

In [17]:

```
print ("Confusion Matrix: \n", cm)
```

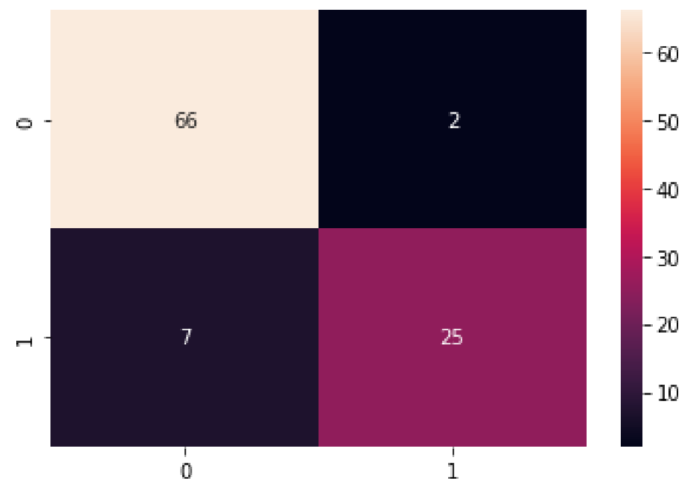
```
Confusion Matrix:
 [[66  2]
 [ 7 25]]
```

In [18]:

```
import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(cm, annot=True)
plt.show
```

Out[18]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [19]:

```
from sklearn.metrics import accuracy_score
print("Accuracy: ", accuracy_score(ytest, y_pred)*100, '%')
```

```
Accuracy:  91.0 %
```

In [20]:

```
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
```

In [21]:

```
#precision tp/(tp+fp)
precision = precision_score(ytest, y_pred)
print('Precision: %f' %precision)
#recall tp/(tp+fn)
recall = recall_score(ytest, y_pred)
print('Recall: %f' %recall)
#f1 2tp/(2tp+fp+fn)
f1 = f1_score(ytest, y_pred)
print('F1: %f' %f1)
```

```
Precision: 0.925926
Recall: 0.781250
F1: 0.847458
```

In [ ]: