```
import pandas as pd #useful for loading the dataset
import numpy as np #to perform array
```

```
from google.colab import files
uploaded = files.upload()
```

Choose Files DigitalAd_dataset.csv

- **DigitalAd_dataset.csv**(text/csv) - 17285 bytes, last modified: 3/20/2023 - 100% done

```
Saving DigitalAd_dataset.csv to DigitalAd_dataset.csv
```

```
dataset = pd.read_csv('DigitalAd_dataset.csv')
```

```
print(dataset.shape)
print(dataset.head(5))
```

```
(1425, 3)
   Age   Salary  Status
0   18  82000.0       0
1   29  80000.0       0
2   47  25000.0       1
3   45  26000.0       1
4   46  28000.0       1
```

```
X = dataset.iloc[:, :-1].values
X
Y = dataset.iloc[:, -1].values
Y
```

```
array([0, 0, 1, ..., 1, 0, 1])
```

```
Y = dataset.iloc[:, -1].values
Y
```

```
array([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1,
       0, 1, 0, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1,
       0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 0, 1,
       1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1,
       1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 1,
       0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,
       1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
       0, 0, 0, 0])
```

```
X = dataset.iloc[:, :-1].values
X
```

```
array([[1.80000000e+01, 8.20000000e+04],
       [2.90000000e+01, 8.00000000e+04],
       [4.70000000e+01, 2.50000000e+04],
       ...,
       [5.60000000e+01, 4.29418333e+04],
       [2.30000000e+01, 4.26616667e+04],
       [6.70000000e+01, 4.23815000e+04]])
```

```
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)
```

```
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

```
from sklearn.linear_model import LogisticRegression
model = LogisticRegression(random_state = 0)
model.fit(X_train, y_train)
```

```
LogisticRegression
LogisticRegression(random_state=0)
```

```
age = int(input("Enter New Customer Age: "))
sal = int(input("Enter New Customer Salary: "))
newCust = [[age,sal]]
result = model.predict(sc.transform(newCust))
print(result)
if result == 1:
  print("Customer will Buy")
else:
  print("Customer won't Buy")
```

```
Enter New Customer Age: 34
Enter New Customer Salary: 54444
[0]
Customer won't Buy
```

```
y_pred = model.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
```

```
 [1 1]
 [0 0]
 [1 1]
 [1 0]]
```

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

print("Confusion Matrix: ")
print(cm)

print("Accuracy of the Model: {0}%".format(accuracy_score(y_test, y_pred)*100))
```

```
Confusion Matrix:
[[152  43]
 [103  59]]
Accuracy of the Model: 59.103641456582636%
```

✓ 0s completed at 3:24 AM