

XGBoost

XGBOOST

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

Importing the dataset

**dataset = pd.read_csv(r'D:\Samsom - All Data\Naresh IT Institute\New
folder\Churn_Modelling.csv')**

x = dataset.iloc[:, 3:-1].values

y = dataset.iloc[:, -1].values

print(x)

print(y)

Encoding categorical data

Label Encoding the "Gender" column

from sklearn.preprocessing import LabelEncoder

le = LabelEncoder()

x[:, 2] = le.fit_transform(x[:, 2])

print(x)

One Hot Encoding the 'Geography' column

from sklearn.compose import ColumnTransformer

from sklearn.preprocessing import OneHotEncoder

**ct = ColumnTransformer(transformers=[('encoder', OneHotEncoder(), [1])],
remainder='passthrough')**

x = np.array(ct.fit_transform(x))

```
print(x)
```

```
#Splitting the dataset into the Training set and Test set
```

```
from sklearn.model_selection import train_test_split
```

```
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.2, random_state = 0)
```

```
# Training XGBoost on the Training set
```

```
from xgboost import XGBClassifier
```

```
classifier = XGBClassifier(random_state = 0)
```

```
classifier.fit(x_train, y_train)
```

```
# Predicting the Test set results
```

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

```
# Making the Confusion Matrix
```

```
from sklearn.metrics import confusion_matrix
```

```
cm = confusion_matrix(y_test, y_pred)
```

```
print(cm)
```

```
from sklearn.metrics import accuracy_score
```

```
ac = accuracy_score(y_test, y_pred)
```

```
print(ac)
```