Dikshita Kambri

118A2044

BE EXTC A2

EXPERIMENT 6

Implementation of Decision tree

EXPERIMENT 6

DECISION TREE

AIM: Implement of decision tree algorithm

APPARATUS: python

THEORY:

Classification is a two-step process, learning step and prediction step, in machine learning. In the learning step, the model is developed based on given training data. In the prediction step, the model is used to predict the response for given data. Decision Tree is one of the easiest and popular classification algorithms to understand and interpret.

Decision Tree Algorithm

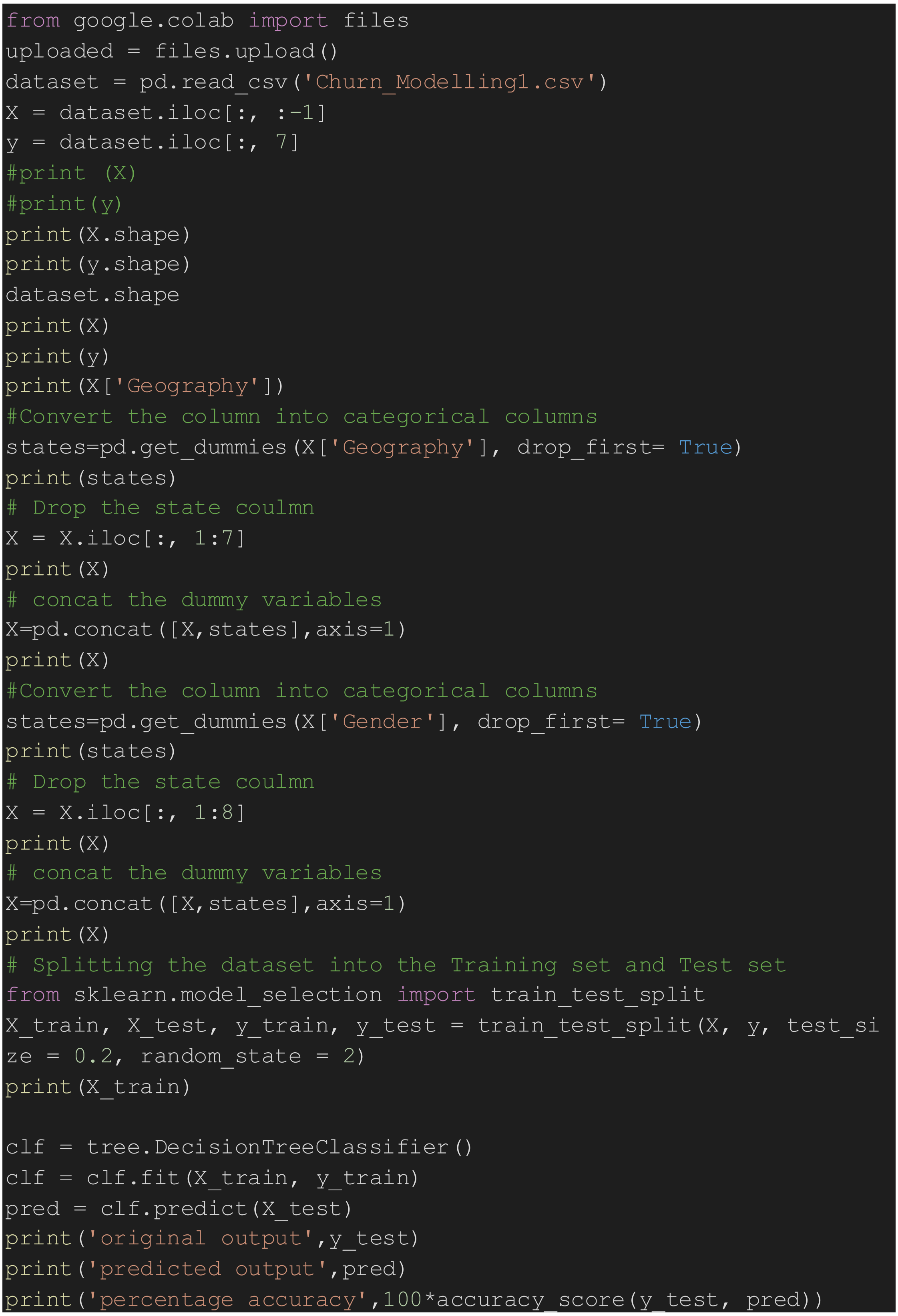
Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.
The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data).
In Decision Trees, for predicting a class label for a record we start from the root of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Trees

Types of decision trees are based on the type of target variable we have. It can be of two types:
1.Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it called a Categorical variable decision tree.
2.Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

PROGRAM:

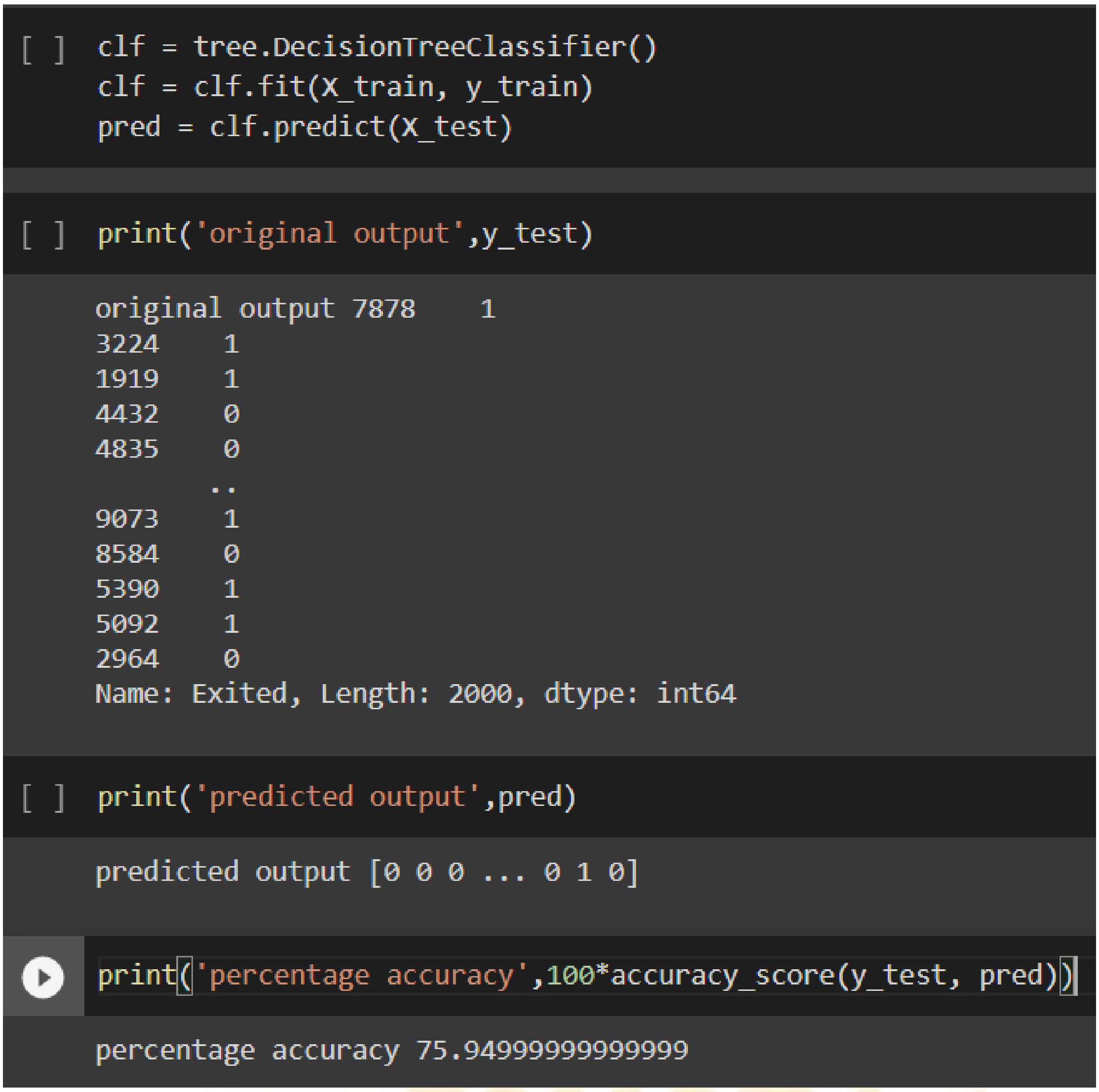
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score
from sklearn import tree
```



```
from google.colab import files
uploaded = files.upload()
dataset = pd.read_csv('Churn_Modelling1.csv')
X = dataset.iloc[:, :-1]
y = dataset.iloc[:, 7]
#print (X)
#print(y)
print(X.shape)
print(y.shape)
dataset.shape
print(X)
print(y)
print(X['Geography'])
#Convert the column into categorical columns
states=pd.get_dummies(X['Geography'], drop_first= True)
print(states)
# Drop the state coulmn
X = X.iloc[:, 1:7]
print(X)
# concat the dummy variables
X=pd.concat([X,states],axis=1)
print(X)
#Convert the column into categorical columns
states=pd.get_dummies(X['Gender'], drop_first= True)
print(states)
# Drop the state coulmn
X = X.iloc[:, 1:8]
print(X)
# concat the dummy variables
X=pd.concat([X,states],axis=1)
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_si
ze = 0.2, random_state = 2)
print(X_train)

clf = tree.DecisionTreeClassifier()
clf = clf.fit(X_train, y_train)
pred = clf.predict(X_test)
print('original output',y_test)
print('predicted output',pred)
print('percentage accuracy',100*accuracy_score(y_test, pred))
```

OUTPUT:



```
[ ]  clf = tree.DecisionTreeClassifier()
     clf = clf.fit(X_train, y_train)
     pred = clf.predict(X_test)
```

```
[ ]  print('original output',y_test)
```

```
original output 7878    1
3224    1
1919    1
4432    0
4835    0
       ..
9073    1
8584    0
5390    1
5092    1
2964    0
Name: Exited, Length: 2000, dtype: int64
```

```
[ ]  print('predicted output',pred)
```

```
predicted output [0 0 0 ... 0 1 0]
```

```
print('percentage accuracy',100*accuracy_score(y_test, pred))
```

```
percentage accuracy 75.94999999999999
```

CONCLUSION:

Created a training model using Decision tree algorithm using churn modeling dataset with accuracy of 75.94%. From this, we observed that, Decision Tree is used to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data