Advance Software Engineering (SE-406)

LAB A1-G3

Laboratory Manual

Department of Software Engineering

DELHI TECHNOLOGICAL UNIVERSITY(DTU)

Shahbad Daulatpur, Bawana Road, Delhi-110042

Submitted to: -

Ms. Parul Sharma

Submitted by:-

Name: ASHISH KUMAR

Roll number: 2K18/SE/041

EXPERIMENT 9

- ASHISH KUMAR
- 2K18/SE/041

Aim:- To train and test fault prediction model using Decision tree.

Introduction:- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. The decisions or the test are performed on the basis of features of the given dataset. It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.

Pros

- Decision trees are easy to interpret and visualize.
- It can easily capture Non-linear patterns.
- It requires fewer data preprocessing from the user, for example, there is no need to normalize columns.
- It can be used for feature engineering such as predicting missing values, suitable for variable selection.
- The decision tree has no assumptions about distribution because of the non-parametric nature of the algorithm.

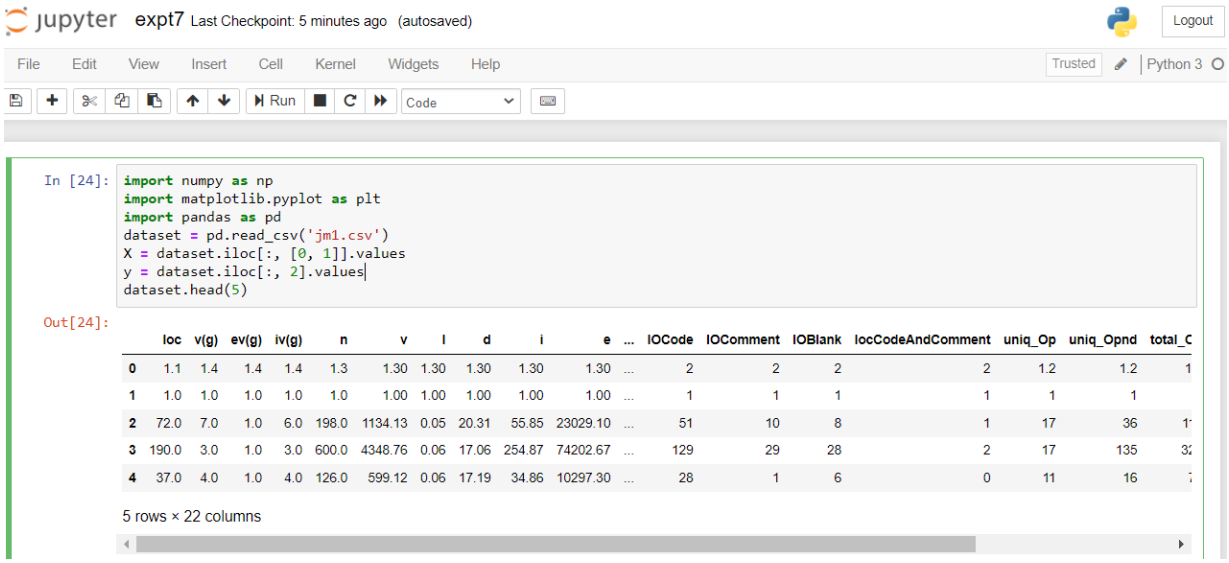
Cons

- Sensitive to noisy data. It can overfit noisy data.
- The small variation (or variance) in data can result in the different decision tree. This can be reduced by bagging and boosting algorithms.
- Decision trees are biased with imbalance dataset, so it is recommended that balance out the dataset before creating the decision tree.

Code & Output:-

Link to dataset: http://promise.site.uottawa.ca/SERepository/datasets/cm1.arff
This is a PROMISE data set made publicly available in order to encourage repeatable, verifiable, refutable, and/or improvable predictive models of software engineering.

```
# Importing the dataset
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
dataset = pd.read_csv('jm1.csv')
X = dataset.iloc[:, [0, 1]].values
y = dataset.iloc[:, 2].values
dataset.head(5)
```



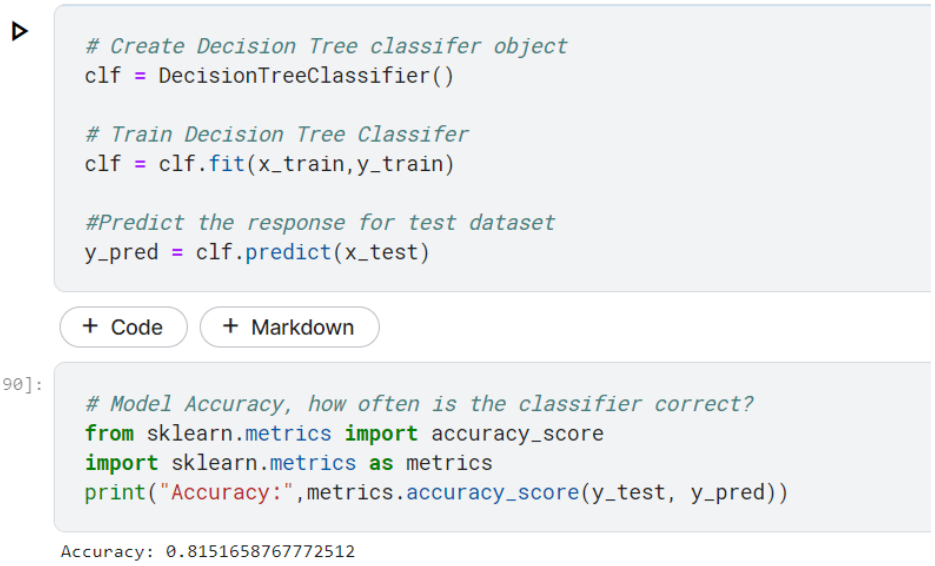

```
# Splitting the dataset into the Training set and Test set
from sklearn.model_selection import train_test_split
x_train, x_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=1)

# Feature Scaling
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)
```

#Applying Decision tree Classifier on the Training Set
# Create Decision Tree classifer object
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier()

# Train Decision Tree Classifer
clf = clf.fit(x_train,y_train)

#Predict the response for test dataset
y_pred = clf.predict(x_test)

# Model Accuracy, how often is the classifier correct?
from sklearn.metrics import accuracy_score
import sklearn.metrics as metrics
print("Accuracy:",metrics.accuracy_score(y_test, y_pred))



Result:- The accuracy of the Decision tree classifier comes out to be 81.51%.

Learning from experiment:- We have successfully been able to train and test fault prediction model using Decision tree and learnt its advantages as well as its disadvantages.