Experiment 2.4

Decision Trees and Random Forests

Student Name: Shubham Shrivastava **UID: 20BCS5472**
Branch: CSE **Section/Group: 618-B**
Semester: 5th Semester **Date of Performance: 17/10/2022**
Subject Name: Machine Learning Lab **Subject Code: 20CSP-317**

1. **Aim:** Decision Trees and Random Forests — Explained with Python Implementation.

2. **Objective:** To prepare a model with Decision Trees and Random Forests algorithm.

3. **Data Set Chosen:** Breast Cancer Wisconsin (Diagnostic) Data Set

4. **Result and output:**

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
```

In [2]:
```
df = pd.read_csv('Breast_Cancer.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean
0	842302	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419
1	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812
2	84300903	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069
3	84348301	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597
4	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809

5 rows × 32 columns

In [4]:
```
df.set_index(['id'], inplace = True)
```

In [6]:
```
df['diagnosis'] = df['diagnosis'].map({'M':1, 'B':0})
```

In [7]:
```
df.apply(lambda x: x.isnull().sum())
```

Out[7]:
```
radius_mean               0
texture_mean              0
perimeter_mean            0
area_mean                 0
smoothness_mean           0
compactness_mean          0
concavity_mean            0
concave points_mean       0
symmetry_mean             0
fractal_dimension_mean    0
radius_se                 0
texture_se                0
perimeter_se              0
area_se                   0
smoothness_se             0
compactness_se            0
concavity_se              0
concave points_se         0
symmetry_se               0
fractal_dimension_se      0
radius_worst              0
texture_worst             0
perimeter_worst           0
area_worst                0
smoothness_worst          0
compactness_worst         0
concavity_worst           0
concave points_worst      0
symmetry_worst            0
fractal_dimension_worst   0
diagnosis                 0
dtype: int64
```

In [8]:
```
df.diagnosis.unique()
```

Out[8]:
```
array([1, 0], dtype=int64)
```

In [13]:
```
feature_space = df.iloc[:, df.columns != 'diagnosis']
feature_class = df.iloc[:, df.columns == 'diagnosis']
```

In [14]:
```
from sklearn.model_selection import train_test_split
```

In [17]:
```
from sklearn.ensemble import RandomForestClassifier
Classifier = RandomForestClassifier(random_state = 50)
Classifier.fit(training_set,class_set)
```

Out[17]:
```
RandomForestClassifier(random_state=50)
```

Out[17]:
```
RandomForestClassifier(random_state=50)
```

In [18]:
```
predict=Classifier.predict(test_set)
```

In [19]:
```
predict
```

Out[19]:
```
array([0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 0, 0, 1, 1, 1, 0,
       0, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 1, 1, 0], dtype=int64)
```

In [20]:
```
from sklearn.metrics import accuracy_score
accuracy_score(test_class_set,predict)
```

Out[20]:
```
0.956140350877193
```

Result: Accuracy of the model is approximately 95%.