Assignment 5 - RA1911028010069

Comparing accuracy of models trained by Decision Tree Classificaton and Random Forest Classification.

Loading Dataset (heart.csv)

In [165]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn import tree
import warnings
df = pd.read_csv('heart.csv')
df.head()
warnings.simplefilter(action='ignore', category=FutureWarning)
```

Checking the datatypes of different columns

In [166]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
Age               918 non-null int64
Sex               918 non-null object
ChestPainType     918 non-null object
RestingBP         918 non-null int64
Cholesterol       918 non-null int64
FastingBS         918 non-null int64
RestingECG        918 non-null object
MaxHR             918 non-null int64
ExerciseAngina    918 non-null object
Oldpeak           918 non-null float64
ST_Slope          918 non-null object
HeartDisease      918 non-null int64
dtypes: float64(1), int64(6), object(5)
memory usage: 86.1+ KB
```

Converting objects to integers

In [167]:
```
df['Sex'] = df['Sex'].replace({'M': 0, 'F': 1})
df['ChestPainType'] = df['ChestPainType'].replace({'ASY': 0, 'NAP': 1, 'ATA':2, 'TA': 2})
df['RestingECG'] = df['RestingECG'].replace({'Normal': 0, 'LVH': 1, 'ST': 2})
df['ExerciseAngina'] = df['ExerciseAngina'].replace({'N': 0, 'Y': 1})
df['ST_Slope'] = df['ST_Slope'].replace({'Up': 0, 'Flat': 1, 'Down': 2})
df.head()
```

Out[167]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpeak	ST_Slope	HeartDisease
0	40	0	2	140	289	0	0	172	0	0.0	0	0
1	49	1	1	160	180	0	0	156	0	1.0	1	1
2	37	0	2	130	283	0	2	98	0	0.0	0	0
3	48	1	0	138	214	0	0	108	1	1.5	1	1
4	54	0	1	150	195	0	0	122	0	0.0	0	0

In [168]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 918 entries, 0 to 917
Data columns (total 12 columns):
Age               918 non-null int64
Sex               918 non-null int64
ChestPainType     918 non-null int64
RestingBP         918 non-null int64
Cholesterol       918 non-null int64
FastingBS         918 non-null int64
RestingECG        918 non-null int64
MaxHR             918 non-null int64
ExerciseAngina    918 non-null int64
Oldpeak           918 non-null float64
ST_Slope          918 non-null int64
HeartDisease      918 non-null int64
dtypes: float64(1), int64(11)
memory usage: 86.1 KB
```

In [169]:
```
df.isnull().values.any()
```

Out[169]: False

In [170]:
```
df.corr()
```

Out[170]:

	Age	Sex	ChestPainType	RestingBP	Cholesterol	FastingBS	RestingECG	MaxHR	ExerciseAngina	Oldpe
Age	1.000000	-0.055750	-0.194563	0.254399	-0.095282	0.198039	0.210498	-0.382045	0.215793	0.2586
Sex	-0.055750	1.000000	0.187720	-0.005133	0.200092	-0.120076	-0.038320	0.189186	-0.190664	-0.1057
ChestPainType	-0.194563	0.187720	1.000000	-0.037988	0.148180	-0.138012	-0.080891	0.359265	-0.433751	-0.2833
RestingBP	0.254399	-0.005133	-0.037988	1.000000	0.100893	0.070193	0.117206	-0.112135	0.155101	0.1648
Cholesterol	-0.095282	0.200092	0.148180	0.100893	1.000000	-0.260974	-0.042595	0.235792	-0.034166	0.0501
FastingBS	0.198039	-0.120076	-0.138012	0.070193	-0.260974	1.000000	0.120774	-0.131438	0.060451	0.0526
RestingECG	0.210498	-0.038320	-0.080891	0.117206	-0.042595	0.120774	1.000000	-0.093379	0.098360	0.0999
MaxHR	-0.382045	0.189186	0.359265	-0.112135	0.235792	-0.131438	-0.093379	1.000000	-0.370425	-0.1606
ExerciseAngina	0.215793	-0.190664	-0.433751	0.155101	-0.034166	0.060451	0.098360	-0.370425	1.000000	0.4087
Oldpeak	0.258612	-0.105734	-0.283397	0.164803	0.050148	0.052698	0.099935	-0.160691	0.408752	1.0000
ST_Slope	0.268264	-0.150693	-0.357587	0.075162	-0.111471	0.175774	0.085422	-0.343419	0.428706	0.5019
HeartDisease	0.282039	-0.305445	-0.514444	0.107589	-0.232741	0.267291	0.107628	-0.400421	0.494282	0.4039

Selecting X and y to train and test model

In [171]:
```
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = df.iloc[:,0:11].values
y = df.iloc[:,11].values

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=0)
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Training model using decision tree classification

In [172]:
```
from sklearn.tree import DecisionTreeClassifier
dtc = DecisionTreeClassifier(criterion='entropy', random_state=0)
dtc.fit(X_train, y_train)
```

Out[172]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=None,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=0,
            splitter='best')
```

Finding accuracy of model using K-Fold

In [173]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
k = 5
kf = KFold(n_splits=k, random_state=None)
result = cross_val_score(dtc , X_train, y_train, cv = kf)
print("Accuracy: {}".format(result.mean()))
```

```
Accuracy: 0.7762191896752354
```

Training model using random forest classification

In [174]:
```
from sklearn.ensemble import RandomForestClassifier
rfc = RandomForestClassifier(criterion='entropy', random_state=0)
rfc.fit(X_train, y_train)
```

Out[174]:
```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='entropy',
            max_depth=None, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,
            oob_score=False, random_state=0, verbose=0, warm_start=False)
```

Finding accuracy of model using K-Fold

In [175]:
```
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import KFold
k = 5
kf = KFold(n_splits=k, random_state=None)
result = cross_val_score(rfc , X_train, y_train, cv = kf)
print("Accuracy: {}".format(result.mean()))
```

```
Accuracy: 0.8561303289960866
```

On comparing the accuracies of the models trained by Decision Tree Classification(77.62%) and Random Forest Classification(85.61%) we can clearly see that Random Forest has greater accuracy.