Project-1

```
import pandas as pd

import numpy as np

from sklearn import tree

from sklearn import preprocessing

titanic_train=pd.read_csv("train.csv")

titanic_train["Age"].mean()
Out[7]: 32.69985376827896

new_age_var=np.where(titanic_train["Age"].isnull(),32,titanic_train["Age"])

titanic_train["Age"]=new_age_var

label_encoder=preprocessing.LabelEncoder()

encoded_sex=label_encoder.fit_transform(titanic_train["Sex"])

tree_model=tree.DecisionTreeClassifier()

tree_model.fit(X=pd.DataFrame(encoded_sex),y=titanic_train["Survived"])
Out[13]:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       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='deprecated',
            random_state=None, splitter='best')

with open("Dtree1.dot",'w') as f:
    f=tree.export_graphviz(tree_model,feature_names=["sex"],out_file=f);



predictors=pd.DataFrame([encoded_sex,titanic_train["Age"],titanic_train["Fare"]]).T

# why 6 because there are two categories in depended variable i.e survived(yes or no) i.e. (2) and (3) independent variable so (2*3=6)

tree_model=tree.DecisionTreeClassifier(max_depth=6)

tree_model.fit(X=predictors,y=titanic_train["Survived"])
Out[18]:
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
            max_depth=6, 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='deprecated',
            random_state=None, splitter='best')

with open("Dtree3.dot",'w') as f:
    f=tree.export_graphviz(tree_model,feature_names=["sex","Age","Fare"],out_file=f);
digraph Tree {
node [shape=box] ;
0 [label="sex <= 0.5\ngini = 0.472\nsamples = 889\nvalue = [549, 340]"] ;
1 [label="Fare <= 48.2\ngini = 0.384\nsamples = 312\nvalue = [81, 231]"] ;
```

```
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;

2 [label="Fare <= 27.825\ngini = 0.447\nsamples = 225\nvalue = [76, 149]"] ;

1 -> 2 ;

3 [label="Fare <= 25.698\ngini = 0.428\nsamples = 193\nvalue = [60, 133]"] ;

2 -> 3 ;

4 [label="Fare <= 24.075\ngini = 0.453\nsamples = 167\nvalue = [58, 109]"] ;

3 -> 4 ;

5 [label="Fare <= 10.481\ngini = 0.442\nsamples = 161\nvalue = [53, 108]"] ;

4 -> 5 ;

6 [label="gini = 0.489\nsamples = 66\nvalue = [28, 38]"] ;

5 -> 6 ;

7 [label="gini = 0.388\nsamples = 95\nvalue = [25, 70]"] ;

5 -> 7 ;

8 [label="Fare <= 24.808\ngini = 0.278\nsamples = 6\nvalue = [5, 1]"] ;

4 -> 8 ;

9 [label="gini = 0.444\nsamples = 3\nvalue = [2, 1]"] ;

8 -> 9 ;

10 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]"] ;

8 -> 10 ;

11 [label="Age <= 25.5\ngini = 0.142\nsamples = 26\nvalue = [2, 24]"] ;

3 -> 11 ;

12 [label="gini = 0.0\nsamples = 10\nvalue = [0, 10]"] ;

11 -> 12 ;

13 [label="Age <= 27.0\ngini = 0.219\nsamples = 16\nvalue = [2, 14]"] ;

11 -> 13 ;

14 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]"] ;

13 -> 14 ;

15 [label="gini = 0.124\nsamples = 15\nvalue = [1, 14]"] ;

13 -> 15 ;

16 [label="Fare <= 28.856\ngini = 0.5\nsamples = 32\nvalue = [16, 16]"] ;

2 -> 16 ;
```

```
17 [label="gini = 0.0\nsamples = 4\nvalue = [4, 0]"] ;

16 -> 17 ;

18 [label="Fare <= 44.24\ngini = 0.49\nsamples = 28\nvalue = [12, 16]"] ;

16 -> 18 ;

19 [label="Fare <= 36.688\ngini = 0.473\nsamples = 26\nvalue = [10, 16]"] ;

18 -> 19 ;

20 [label="gini = 0.499\nsamples = 19\nvalue = [9, 10]"] ;

19 -> 20 ;

21 [label="gini = 0.245\nsamples = 7\nvalue = [1, 6]"] ;

19 -> 21 ;

22 [label="gini = 0.0\nsamples = 2\nvalue = [2, 0]"] ;

18 -> 22 ;

23 [label="Age <= 8.0\ngini = 0.108\nsamples = 87\nvalue = [5, 82]"] ;

1 -> 23 ;

24 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]"] ;

23 -> 24 ;

25 [label="Fare <= 70.275\ngini = 0.089\nsamples = 86\nvalue = [4, 82]"] ;

23 -> 25 ;

26 [label="Fare <= 69.425\ngini = 0.211\nsamples = 25\nvalue = [3, 22]"] ;

25 -> 26 ;

27 [label="gini = 0.0\nsamples = 22\nvalue = [0, 22]"] ;

26 -> 27 ;

28 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]"] ;

26 -> 28 ;

29 [label="Age <= 25.5\ngini = 0.032\nsamples = 61\nvalue = [1, 60]"] ;

25 -> 29 ;

30 [label="Age <= 24.5\ngini = 0.117\nsamples = 16\nvalue = [1, 15]"] ;

29 -> 30 ;

31 [label="gini = 0.0\nsamples = 15\nvalue = [0, 15]"] ;

30 -> 31 ;

32 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]"] ;
```

```
30 -> 32 ;

33 [label="gini = 0.0\nsamples = 45\nvalue = [0, 45]"] ;

29 -> 33 ;

34 [label="Age <= 6.5\ngini = 0.306\nsamples = 577\nvalue = [468, 109]"] ;

0 -> 34 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;

35 [label="Fare <= 20.825\ngini = 0.444\nsamples = 24\nvalue = [8, 16]"] ;

34 -> 35 ;

36 [label="gini = 0.0\nsamples = 8\nvalue = [0, 8]"] ;

35 -> 36 ;

37 [label="Fare <= 64.379\ngini = 0.5\nsamples = 16\nvalue = [8, 8]"] ;

35 -> 37 ;

38 [label="Age <= 3.5\ngini = 0.49\nsamples = 14\nvalue = [8, 6]"] ;

37 -> 38 ;

39 [label="Fare <= 39.344\ngini = 0.496\nsamples = 11\nvalue = [5, 6]"] ;

38 -> 39 ;

40 [label="gini = 0.375\nsamples = 8\nvalue = [2, 6]"] ;

39 -> 40 ;

41 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]"] ;

39 -> 41 ;

42 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]"] ;

38 -> 42 ;

43 [label="gini = 0.0\nsamples = 2\nvalue = [0, 2]"] ;

37 -> 43 ;

44 [label="Fare <= 26.269\ngini = 0.28\nsamples = 553\nvalue = [460, 93]"] ;

34 -> 44 ;

45 [label="Age <= 13.5\ngini = 0.194\nsamples = 404\nvalue = [360, 44]"] ;

44 -> 45 ;

46 [label="Age <= 10.0\ngini = 0.375\nsamples = 4\nvalue = [1, 3]"] ;

45 -> 46 ;

47 [label="gini = 0.0\nsamples = 2\nvalue = [0, 2]"] ;

46 -> 47 ;
```

48 [label="Age <= 11.5\ngini = 0.5\nsamples = 2\nvalue = [1, 1]"] ;

46 -> 48 ;

49 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]"] ;

48 -> 49 ;

50 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]"] ;

48 -> 50 ;

51 [label="Age <= 32.5\ngini = 0.184\nsamples = 400\nvalue = [359, 41]"] ;

45 -> 51 ;

52 [label="Age <= 30.75\ngini = 0.231\nsamples = 195\nvalue = [169, 26]"] ;

51 -> 52 ;

53 [label="gini = 0.199\nsamples = 178\nvalue = [158, 20]"] ;

52 -> 53 ;

54 [label="gini = 0.457\nsamples = 17\nvalue = [11, 6]"] ;

52 -> 54 ;

55 [label="Fare <= 7.91\ngini = 0.136\nsamples = 205\nvalue = [190, 15]"] ;

51 -> 55 ;

56 [label="gini = 0.061\nsamples = 96\nvalue = [93, 3]"] ;

55 -> 56 ;

57 [label="gini = 0.196\nsamples = 109\nvalue = [97, 12]"] ;

55 -> 57 ;

58 [label="Fare <= 26.469\ngini = 0.441\nsamples = 149\nvalue = [100, 49]"] ;

44 -> 58 ;

59 [label="gini = 0.0\nsamples = 4\nvalue = [0, 4]"] ;

58 -> 59 ;

60 [label="Fare <= 387.665\ngini = 0.428\nsamples = 145\nvalue = [100, 45]"] ;

58 -> 60 ;

61 [label="Age <= 22.5\ngini = 0.421\nsamples = 143\nvalue = [100, 43]"] ;

60 -> 61 ;

62 [label="gini = 0.227\nsamples = 23\nvalue = [20, 3]"] ;

61 -> 62 ;

63 [label="gini = 0.444\nsamples = 120\nvalue = [80, 40]"] ;

```
61 -> 63 ;
64 [label="gini = 0.0\nsamples = 2\nvalue = [0, 2]"] ;
60 -> 64 ;
}
```

```
tree_model.score(X=predictors,y=titanic_train["Survived"])
Out[20]: 0.8267716535433071
```

Project-3

```
import pandas as pd

import numpy as np

from sklearn import tree

from sklearn import preprocessing

from sklearn.ensemble import RandomForestClassifier
```

```
titanic_train=pd.read_excel("Bank_Personal_Loan_Modelling.xlsx")


titanic_train.columns
Out[13]:
Index(['ID', 'Age', 'Experience', 'Income', 'ZIP Code', 'Family', 'CCAvg',
       'Education', 'Mortgage', 'Personal Loan', 'Securities Account',
       'CD Account', 'Online', 'CreditCard'],
      dtype='object')


rf_model=RandomForestClassifier(n_estimators=1000,max_features=2,oob_score=True)


features=["Age","Experience","Income","Family","CCAvg","Education","Mortgage","Securities Account","CD Account","Online","CreditCard"]


rf_model.fit(X=titanic_train[features],y=titanic_train["Personal Loan"])
Out[16]:
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features=2,
                       max_leaf_nodes=None, max_samples=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=1000,
                       n_jobs=None, oob_score=True, random_state=None,
                       verbose=0, warm_start=False)


print("OOB Accuracy:")
print(rf_model.oob_score_);
OOB Accuracy:
0.9882
```

```
for feature,imp in zip(features,rf_model.feature_importances_):
    print(feature,imp);
```

```
Age 0.04401039345617236
Experience 0.04396227270521382
Income 0.34594639739966515
Family 0.09606116118393337
CCAvg 0.18154897001324516
Education 0.16490146922331586
Mortgage 0.045134917047122865
Securities Account 0.00551063903474999
CD Account 0.054528705463591245
Online 0.008595614088972778
CreditCard 0.00979946038401734
```

```
# income,CCAvg,Education as independent variable

tree_model=tree.DecisionTreeClassifier()

predictors=pd.DataFrame([titanic_train["Income"],titanic_train["CCAvg"],titanic_train["Education"]])
.T

tree_model=tree.DecisionTreeClassifier(max_depth=6,max_leaf_node=10)


tree_model.fit(X=predictors,y=titanic_train["Personal Loan"])
```

Out[31]:

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',
                       max_depth=6, 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='deprecated',
            random_state=None, splitter='best')

with open("Dtree4.dot",'w') as f:
  f=tree.export_graphviz(tree_model,feature_names=["Income","CCAvg","Education"],out_file=f);


digraph Tree {
node [shape=box] ;
0 [label="Income <= 113.5\ngini = 0.174\nsamples = 5000\nvalue = [4520, 480]"] ;
1 [label="CCAvg <= 2.95\ngini = 0.041\nsamples = 4021\nvalue = [3937, 84]"] ;
0 -> 1 [labeldistance=2.5, labelangle=45, headlabel="True"] ;
2 [label="Income <= 106.5\ngini = 0.007\nsamples = 3723\nvalue = [3710, 13]"] ;
1 -> 2 ;
3 [label="gini = 0.0\nsamples = 3629\nvalue = [3629, 0]"] ;
2 -> 3 ;
4 [label="Education <= 1.5\ngini = 0.238\nsamples = 94\nvalue = [81, 13]"] ;
2 -> 4 ;
5 [label="CCAvg <= 0.35\ngini = 0.127\nsamples = 44\nvalue = [41, 3]"] ;
4 -> 5 ;
6 [label="CCAvg <= 0.25\ngini = 0.32\nsamples = 10\nvalue = [8, 2]"] ;
5 -> 6 ;
7 [label="gini = 0.219\nsamples = 8\nvalue = [7, 1]"] ;
6 -> 7 ;
8 [label="gini = 0.5\nsamples = 2\nvalue = [1, 1]"] ;
6 -> 8 ;
9 [label="Income <= 109.5\ngini = 0.057\nsamples = 34\nvalue = [33, 1]"] ;
5 -> 9 ;
10 [label="gini = 0.153\nsamples = 12\nvalue = [11, 1]"] ;
9 -> 10 ;
```

11 [label="gini = 0.0\nsamples = 22\nvalue = [22, 0]"] ;

9 -> 11 ;

12 [label="CCAvg <= 1.65\ngini = 0.32\nsamples = 50\nvalue = [40, 10]"] ;

4 -> 12 ;

13 [label="CCAvg <= 0.3\ngini = 0.457\nsamples = 17\nvalue = [11, 6]"] ;

12 -> 13 ;

14 [label="gini = 0.0\nsamples = 3\nvalue = [3, 0]"] ;

13 -> 14 ;

15 [label="gini = 0.49\nsamples = 14\nvalue = [8, 6]"] ;

13 -> 15 ;

16 [label="Income <= 108.5\ngini = 0.213\nsamples = 33\nvalue = [29, 4]"] ;

12 -> 16 ;

17 [label="gini = 0.5\nsamples = 4\nvalue = [2, 2]"] ;

16 -> 17 ;

18 [label="gini = 0.128\nsamples = 29\nvalue = [27, 2]"] ;

16 -> 18 ;

19 [label="Income <= 82.5\ngini = 0.363\nsamples = 298\nvalue = [227, 71]"] ;

1 -> 19 ;

20 [label="CCAvg <= 3.55\ngini = 0.142\nsamples = 117\nvalue = [108, 9]"] ;

19 -> 20 ;

21 [label="CCAvg <= 3.45\ngini = 0.269\nsamples = 50\nvalue = [42, 8]"] ;

20 -> 21 ;

22 [label="CCAvg <= 3.25\ngini = 0.245\nsamples = 49\nvalue = [42, 7]"] ;

21 -> 22 ;

23 [label="gini = 0.188\nsamples = 38\nvalue = [34, 4]"] ;

22 -> 23 ;

24 [label="gini = 0.397\nsamples = 11\nvalue = [8, 3]"] ;

22 -> 24 ;

25 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]"] ;

21 -> 25 ;

26 [label="Income <= 81.5\ngini = 0.029\nsamples = 67\nvalue = [66, 1]"] ;

```
20 -> 26 ;

27 [label="gini = 0.0\nsamples = 60\nvalue = [60, 0]"] ;

26 -> 27 ;

28 [label="CCAvg <= 3.75\ngini = 0.245\nsamples = 7\nvalue = [6, 1]"] ;

26 -> 28 ;

29 [label="gini = 0.444\nsamples = 3\nvalue = [2, 1]"] ;

28 -> 29 ;

30 [label="gini = 0.0\nsamples = 4\nvalue = [4, 0]"] ;

28 -> 30 ;

31 [label="Education <= 1.5\ngini = 0.45\nsamples = 181\nvalue = [119, 62]"] ;

19 -> 31 ;

32 [label="CCAvg <= 4.25\ngini = 0.299\nsamples = 93\nvalue = [76, 17]"] ;

31 -> 32 ;

33 [label="CCAvg <= 4.05\ngini = 0.44\nsamples = 46\nvalue = [31, 15]"] ;

32 -> 33 ;

34 [label="gini = 0.402\nsamples = 43\nvalue = [31, 12]"] ;

33 -> 34 ;

35 [label="gini = 0.0\nsamples = 3\nvalue = [0, 3]"] ;

33 -> 35 ;

36 [label="CCAvg <= 4.65\ngini = 0.081\nsamples = 47\nvalue = [45, 2]"] ;

32 -> 36 ;

37 [label="gini = 0.188\nsamples = 19\nvalue = [17, 2]"] ;

36 -> 37 ;

38 [label="gini = 0.0\nsamples = 28\nvalue = [28, 0]"] ;

36 -> 38 ;

39 [label="Income <= 92.5\ngini = 0.5\nsamples = 88\nvalue = [43, 45]"] ;

31 -> 39 ;

40 [label="Education <= 2.5\ngini = 0.388\nsamples = 38\nvalue = [28, 10]"] ;

39 -> 40 ;

41 [label="gini = 0.5\nsamples = 10\nvalue = [5, 5]"] ;

40 -> 41 ;
```

```
42 [label="gini = 0.293\nsamples = 28\nvalue = [23, 5]"] ;

40 -> 42 ;

43 [label="Income <= 109.0\ngini = 0.42\nsamples = 50\nvalue = [15, 35]"] ;

39 -> 43 ;

44 [label="gini = 0.459\nsamples = 42\nvalue = [15, 27]"] ;

43 -> 44 ;

45 [label="gini = 0.0\nsamples = 8\nvalue = [0, 8]"] ;

43 -> 45 ;

46 [label="Education <= 1.5\ngini = 0.482\nsamples = 979\nvalue = [583, 396]"] ;

0 -> 46 [labeldistance=2.5, labelangle=-45, headlabel="False"] ;

47 [label="CCAvg <= 6.633\ngini = 0.194\nsamples = 635\nvalue = [566, 69]"] ;

46 -> 47 ;

48 [label="CCAvg <= 6.55\ngini = 0.225\nsamples = 502\nvalue = [437, 65]"] ;

47 -> 48 ;

49 [label="CCAvg <= 4.05\ngini = 0.22\nsamples = 500\nvalue = [437, 63]"] ;

48 -> 49 ;

50 [label="Income <= 159.5\ngini = 0.177\nsamples = 346\nvalue = [312, 34]"] ;

49 -> 50 ;

51 [label="gini = 0.128\nsamples = 263\nvalue = [245, 18]"] ;

50 -> 51 ;

52 [label="gini = 0.311\nsamples = 83\nvalue = [67, 16]"] ;

50 -> 52 ;

53 [label="Income <= 179.5\ngini = 0.306\nsamples = 154\nvalue = [125, 29]"] ;

49 -> 53 ;

54 [label="gini = 0.349\nsamples = 129\nvalue = [100, 29]"] ;

53 -> 54 ;

55 [label="gini = 0.0\nsamples = 25\nvalue = [25, 0]"] ;

53 -> 55 ;

56 [label="gini = 0.0\nsamples = 2\nvalue = [0, 2]"] ;

48 -> 56 ;

57 [label="Income <= 176.5\ngini = 0.058\nsamples = 133\nvalue = [129, 4]"] ;
```

```
47 -> 57 ;

58 [label="CCAvg <= 6.95\ngini = 0.02\nsamples = 99\nvalue = [98, 1]"] ;

57 -> 58 ;

59 [label="CCAvg <= 6.85\ngini = 0.077\nsamples = 25\nvalue = [24, 1]"] ;

58 -> 59 ;

60 [label="gini = 0.0\nsamples = 15\nvalue = [15, 0]"] ;

59 -> 60 ;

61 [label="gini = 0.18\nsamples = 10\nvalue = [9, 1]"] ;

59 -> 61 ;

62 [label="gini = 0.0\nsamples = 74\nvalue = [74, 0]"] ;

58 -> 62 ;

63 [label="Income <= 178.5\ngini = 0.161\nsamples = 34\nvalue = [31, 3]"] ;

57 -> 63 ;

64 [label="CCAvg <= 7.6\ngini = 0.5\nsamples = 2\nvalue = [1, 1]"] ;

63 -> 64 ;

65 [label="gini = 0.0\nsamples = 1\nvalue = [1, 0]"] ;

64 -> 65 ;

66 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]"] ;

64 -> 66 ;

67 [label="CCAvg <= 8.05\ngini = 0.117\nsamples = 32\nvalue = [30, 2]"] ;

63 -> 67 ;

68 [label="gini = 0.26\nsamples = 13\nvalue = [11, 2]"] ;

67 -> 68 ;

69 [label="gini = 0.0\nsamples = 19\nvalue = [19, 0]"] ;

67 -> 69 ;

70 [label="Income <= 116.5\ngini = 0.094\nsamples = 344\nvalue = [17, 327]"] ;

46 -> 70 ;

71 [label="CCAvg <= 2.15\ngini = 0.491\nsamples = 30\nvalue = [17, 13]"] ;

70 -> 71 ;

72 [label="CCAvg <= 0.55\ngini = 0.26\nsamples = 13\nvalue = [11, 2]"] ;

71 -> 72 ;
```

```
73 [label="gini = 0.0\nsamples = 1\nvalue = [0, 1]"] ;

72 -> 73 ;

74 [label="Income <= 114.5\ngini = 0.153\nsamples = 12\nvalue = [11, 1]"] ;

72 -> 74 ;

75 [label="gini = 0.0\nsamples = 7\nvalue = [7, 0]"] ;

74 -> 75 ;

76 [label="gini = 0.32\nsamples = 5\nvalue = [4, 1]"] ;

74 -> 76 ;

77 [label="CCAvg <= 4.0\ngini = 0.457\nsamples = 17\nvalue = [6, 11]"] ;

71 -> 77 ;

78 [label="CCAvg <= 3.35\ngini = 0.497\nsamples = 13\nvalue = [6, 7]"] ;

77 -> 78 ;

79 [label="gini = 0.444\nsamples = 9\nvalue = [3, 6]"] ;

78 -> 79 ;

80 [label="gini = 0.375\nsamples = 4\nvalue = [3, 1]"] ;

78 -> 80 ;

81 [label="gini = 0.0\nsamples = 4\nvalue = [0, 4]"] ;

77 -> 81 ;

82 [label="gini = 0.0\nsamples = 314\nvalue = [0, 314]"] ;

70 -> 82 ;

}
```

```
tree_model.score(X=predictors,y=titanic_train["Personal Loan"])
Out[33]: 0.9738
```

Project-2

```
import pandas as pd

import numpy as np

from sklearn import tree

from sklearn.ensemble import RandomForestClassifier

ds=pd.read_csv("general_data.csv")

ds.columns
Out[6]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',
       'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
       'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',
       'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',
       'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],
      dtype='object')

from sklearn import preprocessing

label_encoder=preprocessing.LabelEncoder()

ds["Attrition"]=label_encoder.fit_transform(ds["Attrition"])
```

```
ds["BusinessTravel"]=label_encoder.fit_transform(ds["BusinessTravel"])

ds["Department"]=label_encoder.fit_transform(ds["Department"])

ds["EducationField"]=label_encoder.fit_transform(ds["EducationField"])

ds["Gender"]=label_encoder.fit_transform(ds["Gender"])

ds["JobRole"]=label_encoder.fit_transform(ds["JobRole"])

ds["MaritalStatus"]=label_encoder.fit_transform(ds["MaritalStatus"])

ds1=ds.drop(["EmployeeID","Over18","StandardHours","EmployeeCount"],axis=1)

ds1.columns
Out[17]:
Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',
       'Education', 'EducationField', 'Gender', 'JobLevel', 'JobRole',
       'MaritalStatus', 'MonthlyIncome', 'NumCompaniesWorked',
       'PercentSalaryHike', 'StockOptionLevel', 'TotalWorkingYears',
       'TrainingTimesLastYear', 'YearsAtCompany', 'YearsSinceLastPromotion',
       'YearsWithCurrManager'],
      dtype='object')

ds2=ds1.dropna()

ds3=ds2.drop_duplicates()

rf_model=RandomForestClassifier(n_estimators=1000,max_features=2,oob_score=True)
```

```
features=['Age','BusinessTravel', 'Department', 'DistanceFromHome','Education', 'EducationField',
'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
'NumCompaniesWorked','PercentSalaryHike', 'StockOptionLevel',
'TotalWorkingYears','TrainingTimesLastYear', 'YearsAtCompany',
'YearsSinceLastPromotion','YearsWithCurrManager']
```

```
rf_model.fit(X=ds3[features],y=ds3["Attrition"])
```

Out[22]:

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                criterion='gini', max_depth=None, max_features=2,
                max_leaf_nodes=None, max_samples=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=1000,
                n_jobs=None, oob_score=True, random_state=None,
                verbose=0, warm_start=False)
```

```
print("OOB Accuracy:")
print(rf_model.oob_score_);
```

```
OOB Accuracy:
0.8435374149659864
```

```
for feature,imp in zip(features,rf_model.feature_importance_):
   print(feature,imp);
```

```
Traceback (most recent call last):

  File "<ipython-input-29-52475b6d2c12>", line 1, in <module>
    for feature,imp in zip(features,rf_model.feature_importance_):

AttributeError: 'RandomForestClassifier' object has no attribute 'feature_importance_'
```

```
for feature,imp in zip(features,rf_model.feature_importances_):
    print(feature,imp);
```

Age 0.09722345555632105

BusinessTravel 0.027926281684840425

Department 0.026472397799854944

DistanceFromHome 0.0682332160799417

Education 0.03989682232691238

EducationField 0.042376723473310486

Gender 0.018503830359477957

JobLevel 0.037530388979159515

JobRole 0.05454731814686057

MaritalStatus 0.04004691413175912

MonthlyIncome 0.092681427219081

NumCompaniesWorked 0.056248360994626934

PercentSalaryHike 0.06530764436158845

StockOptionLevel 0.03513463939064721

TotalWorkingYears 0.08658203823618554

TrainingTimesLastYear 0.04436701420761524

YearsAtCompany 0.06904838314492537

YearsSinceLastPromotion 0.04398175870699807

YearsWithCurrManager 0.05389138519989414

```
tree_model=tree.DecisionTreeClassifier(max_depth=6,max_leaf_nodes=10)

predictors=pd.DataFrame(ds3["TotalWorkingYears"],ds3["Age"],ds3["MonthlyIncome"]).T

tree_model.fit(X=ds3[features],y=ds3["Attrition"])
```

Out[36]:

DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',

```
                    max_depth=6, max_features=None, max_leaf_nodes=10,
                    min_impurity_decrease=0.0, min_impurity_split=None,
                    min_samples_leaf=1, min_samples_split=2,
                    min_weight_fraction_leaf=0.0, presort='deprecated',
                    random_state=None, splitter='best')

tree_model.fit(X=predictors,y=ds3["Attrition"])
```

Decision TreeClassifier(class_weight=None, criterion='gini', max_depth=6, max features-None, max_leaf_nodes-10, min_impurity_decrease-0.0, min impurity split-None, min samples 1leaf-1, min samples split-2, min_weight_Fraction_leaf-0.0, presort=False, random_state=None, splitter-'best')

```
with open("Dtree6.dot",'w') as f:

f=tree.export_graphviz(tree_model,feature_names=["TotalWorkingYears","Age","MonthlyIncome"], out_file=f);
```

4444444444444444444444444444444444444444444444444444444

```
tree_model.fit(X=ds3[predictors],y=ds3["Attrition"])

import statsmodels.api as sm

Y=ds3.columns

X=ds3[['Age','BusinessTravel', 'Department', 'DistanceFromHome','Education', 'EducationField',
'Gender', 'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',
'NumCompaniesWorked','PercentSalaryHike', 'StockOptionLevel',
'TotalWorkingYears','TrainingTimesLastYear', 'YearsAtCompany',
'YearsSinceLastPromotion','YearsWithCurrManager']]

X1=sm.add_constant(X)
```

```
Logistic_Attrition=sm.Logit(y,X1)

Result= Logistic_Attrition.fit()

Result.summary()
```