**Decision Tree**

import pandas as pd

import numpy as np

from sklearn import preprocessing

from sklearn import tree

dataset=pd.read\_csv("C:/Users/PC/Downloads/train.csv")

dataset['Age'].mean()

Output: 32.69985376827896

new\_age=np.where(dataset['Age'].isna(),32,dataset['Age'])

encoder=preprocessing.LabelEncoder()

new\_Gen=encoder.fit\_transform(dataset['Sex'])

tree\_model=tree.DecisionTreeClassifier()

predictors=pd.DataFrame([new\_Gen,dataset['Age'],dataset['Fare']]).T

tree\_model.fit(X=predictors,y=dataset['Survived'])

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

f=tree.export\_graphviz(tree\_model,feature\_names=['Sex','Age','Fare'],out\_file =f);

from sklearn.metrics import accuracy\_score

tree\_model.score(X=predictors,y=dataset['Survived'])

Output

0.9763779527559056

Accuracy is more with 97%

**Bank\_Modelling**

import pandas as pd

import numpy as np

from sklearn import preprocessing

from sklearn import tree

from sklearn.ensemble import RandomForestClassifier

dataset1=pd.read\_excel("C:/Users/PC/Downloads/Bank\_Personal\_Loan\_Modelling.xlsx",sheet\_name=1)

dataset1.columns

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=dataset1[features],y=dataset1['Personal Loan'])

for feature,imp in zip(features,rf\_model.feature\_importances\_):

print(feature,imp);

Age 0.04445988762522036

Experience 0.04385797849898187

Income 0.3468708539270932

Family 0.09974914511058638

CCAvg 0.1853870560410674

Education 0.15929971324104122

Mortgage 0.04324401873730802

Securities Account 0.0054211194216508855

CD Account 0.05348325652285715

Online 0.008445963107871246

CreditCard 0.009781007766322321

We got three important variables from random forest classifier.

Now we are doing decision tree with Income, Education,CCAvg.

tree\_model1=tree.DecisionTreeClassifier()

predictor=pd.DataFrame([dataset1['Income'],dataset1['Education'],dataset1['CCAvg']]).T

tree\_model1.fit(X=predictor,y=dataset1['Personal Loan'])

with open("Dtree1.dot",'w') as f:

f=tree.export\_graphviz(tree\_model1,feature\_names=['Income','Education','CCAvg'],out\_file =f);

tree\_model1.score(X=predictor,y=dataset1['Personal Loan'])

Output:

0.9972

Conclusion: We got model accuracy of 99% with Income, CCAvg and Education.

So, to get loan from bank these three attributes are required.

**ATTRITION**

import pandas as pd

import numpy as np

from sklearn import preprocessing

from sklearn import tree

dataset2=pd.read\_csv("C:/Users/PC/Downloads/general\_data.csv")

coder=preprocessing.LabelEncoder()

new\_Gen=coder.fit\_transform(dataset2['Gender'])

dataset2['Gender']=new\_Gen

new\_Att=coder.fit\_transform(dataset2['Attrition'])

dataset2['Attrition']=new\_Att

new\_job=coder.fit\_transform(dataset2['JobRole'])

dataset2['JobRole']=new\_job

dataset2.columns

rf\_model=RandomForestClassifier(n\_estimators=1000,max\_features=2,oob\_score=True)

features=['Age','Education','EmployeeCount','EmployeeID','Gender','JobLevel','JobRole','MonthlyIncome','PercentSalaryHike','StandardHours']

rf\_model.fit(X=dataset2[features],y=dataset2['Attrition'])

for feature,imp in zip(features,rf\_model.feature\_importances\_):

print(feature,imp);

Output:

Age 0.22399967804175203

Education 0.07315128653039071

EmployeeCount 0.0

EmployeeID 0.13657779194556952

Gender 0.02378565829476121

JobLevel 0.06167399387698692

JobRole 0.10052500840954305

MonthlyIncome 0.24526119756981532

PercentSalaryHike 0.1350253853311813

StandardHours 0.0

From Random Forest test we got 5 important attributes

Age,EmployeedID,JobRole,MonthlyIncome,PercentSalaryHike.

tree\_model2=tree.DecisionTreeClassifier()

predictor=pd.DataFrame([dataset2['Age'],dataset2['EmployeeID'],dataset2['JobRole'],dataset2['MonthlyIncome'],dataset2['PercentSalaryHike']]).T

tree\_model2.fit(X=predictor,y=dataset2['Attrition'])

with open("Dtree2.dot",'w') as f:

f=tree.export\_graphviz(tree\_model2,feature\_names=['Age','EmployeeID','JobRole','MonthlyIncome','PercentSalaryHike'],out\_file =f);

tree\_model2.score(X=predictor,y=dataset2['Attrition'])

1.0

Conclusion: Model Accuracy is 100% which means attritions are happening because of Age, percentsalaryhike and monthly income.