Randomforest

Fraud data set

##random forest

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

##Importing the data set

data=pd.read\_csv("Fraud\_check.csv")

data.info()

## Three categorical variable and 3 integers variables

## so we should convert the there factor and create factor variable as fraud check

##some of teda on the variables

data.mean()

data.var()

data.std()

data.skew()

data.kurt()

data.hist()

data.type()

type(data)

##converting the independent variable which are categorical and labeling with numeric

# Considering only the string data type columns and

string\_columns=["Urban","Undergrad","Marital.Status"]

from sklearn import preprocessing

for i in string\_columns:

number = preprocessing.LabelEncoder()

data[i]=number.fit\_transform(data[i])

## Now making the dependent variable from taxable income

def fraud(x):

if x <= 30000:

return "Risky"

else:

return "good"

data["fraud\_detect"]=data["Taxable.Income"].apply(fraud)

## splitting the data into x asnd y

ind=["Urban","Undergrad","Marital.Status","City.Population","Work.Experience"]

x=data[ind]

y=data["fraud\_detect"]

##splitting the data info train and test

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=1)

##model bulding with random forest

from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n\_jobs=-1,oob\_score=True,n\_estimators=1000,criterion="entropy")

# n\_estimators -> Number of trees ( you can increase for better accuracy)

# n\_jobs -> Parallelization of the computing and signifies the number of jobs

# running parallel for both fit and predict

# oob\_score = True means model has done out of box sampling to make predictions

rf.fit(X\_train,y\_train)

rf.estimators #

rf.classes\_ # class labels (output)

rf.n\_classes\_ # Number of levels in class labels

rf.n\_features\_ # Number of input features in model 8 here.

rf.n\_outputs\_ # Number of outputs when fit performed

rf.oob\_score\_ # 0.738 is on the training test

y\_pred=rf.predict(X\_test)

## evaluation of the model

from sklearn.metrics import confusion\_matrix

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

metrics.accuracy\_score(y\_test,y\_pred)

##0.772

confusion\_matrix(y\_test,y\_pred)

pd.crosstab(y\_test,y\_pred)

Accuracy=139/180

##0.772

Company data

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

##importing the data set

data=pd.read\_csv("Company\_Data.csv")

data.info()## this function is used to known about the variable in the data set

data.head()

##To get the view of the data set

##in this data one float variable 7 int variable and 3 categorical variable are present

## we should covert the independent variable to numerical and dependent variable to categorical

data["US"].replace(['Yes','No'],[1,0],inplace=True)

data["Urban"].replace(['Yes','No'],[1,0],inplace=True)

data['ShelveLoc'].replace(['Bad','Medium','Good'],[0,1,2],inplace=True)

def sales\_to\_cat(x):

if x > 5:

return "High sales"

else :

return "low sales"

data["sales\_categorical"]=data["Sales"].apply(sales\_to\_cat)

data["sales\_categorical"].value\_count()

##We will do some analysis on data set

data.describe()

data.mean()

data.var()

data.std()

data.skew()

data.kurt()

data.hist(figsize=(20,20))

data.corr()

## variables like population,price,age,income,comprice are spred from the mean

##the main variable effecting the sales is the compprice

data.corrwith

## split the data set feauters and target variable

ind\_var=['CompPrice','Income','Advertising','Population','Price','Age','Education','US','Urban','ShelveLoc']

x=data[ind\_var]

y=data['sales\_categorical']

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.3, random\_state=1) # 70% training and 30% test

from sklearn.model\_selection import train\_test\_split # Import train\_test\_split function

##model buiding##

from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(n\_jobs=-1,oob\_score=True,n\_estimators=1000,criterion="entropy")

# n\_estimators -> Number of trees ( you can increase for better accuracy)

# n\_jobs -> Parallelization of the computing and signifies the number of jobs

# running parallel for both fit and predict

# oob\_score = True means model has done out of box sampling to make predictions

rf.fit(X\_train,y\_train)

rf.estimators #

rf.classes\_ # class labels High sales and low sales

rf.n\_classes\_ # Number of levels in class labels i.e 2

rf.n\_features\_ # Number of input features in model 10 here.

rf.n\_outputs\_ # Number of outputs when fit performed

rf.oob\_score\_ # 0.8178 is on the training test

y\_pred=rf.predict(X\_test)

## evaluation of the model

from sklearn.metrics import confusion\_matrix

from sklearn import metrics #Import scikit-learn metrics module for accuracy calculation

metrics.accuracy\_score(y\_test,y\_pred)

##0.858333

confusion\_matrix(y\_test,y\_pred)

pd.crosstab(y\_test,y\_pred)

Accuracy=103/120

##0.85833

Iris data set

##importing the dataset

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

iris = pd.read\_csv("iris.csv")

## some eda analysis

iris.mean()

#Sepal.Length 5.843333

#Sepal.Width 3.057333

#Petal.Length 3.758000

#Petal.Width 1.199333

iris.std()

#Sepal.Length 0.828066

#Sepal.Width 0.435866

#Petal.Length 1.765298

#Petal.Width 0.762238

##petal length is spred from the mean compare to other variables

iris.describe()

# Sepal.Length Sepal.Width Petal.Length Petal.Width

#count 150.000000 150.000000 150.000000 150.000000

#mean 5.843333 3.057333 3.758000 1.199333

#std 0.828066 0.435866 1.765298 0.762238

#min 4.300000 2.000000 1.000000 0.100000

#25% 5.100000 2.800000 1.600000 0.300000

#50% 5.800000 3.000000 4.350000 1.300000

#75% 6.400000 3.300000 5.100000 1.800000

#max 7.900000 4.400000 6.900000 2.500000

iX=iris[["Sepal.Length","Sepal.Width","Petal.Length","Petal.Width"]]

iy=iris[["Species"]]

##model building

from sklearn.ensemble import RandomForestClassifier

rfiris = RandomForestClassifier(n\_jobs=4,oob\_score=True,n\_estimators=100,criterion="entropy")

rfiris.fit(iX,iy)

iris["rf\_pred"] = rfiris.predict(iX)

##evaluation of the model

from sklearn.metrics import confusion\_matrix

confusion\_matrix(iris["Species"],iris["rf\_pred"])

#array([[50, 0, 0],

# [ 0, 50, 0],

# [ 0, 0, 50]], # 100 Percent

pd.crosstab(iris["Species"],iris["rf\_pred"])

#rf\_pred setosa versicolor virginica

#Species

#setosa 50 0 0

#versicolor 0 50 0

#virginica 0 0 50