import matplotlib.pyplot as plt#used for plotting graph

from sklearn.model\_selection import train\_test\_split# to split the data into two parts

from sklearn.ensemble import RandomForestClassifier #forrandomforest classifier

from sklearn.metrics import accuracy\_score# to check the error and accuracy of the model

from sklearn import svm#for support vector machine

import pandas as pd # data processing, CSV file I/O

import seaborn as sns # for statistical data visualization.

from IPython import get\_ipython

get\_ipython().run\_line\_magic('matplotlib', 'inline')

data = pd.read\_csv("C:/Users/ASUS/Desktop/PR\_Lab/a3/data.csv",header=0)# here header 0 means the 0 th row is the coloumn

# header in data

print(data.head(2))# The data have imported and having 33 columns

# head is used for to see top 5 by default . Since head(2) is given it will print top 2 rows

data.info()

# drop the column Unnamed: 32

# axis 1 means droping the column

data.columns# gives the column name which are persent in the data set

# To remove Id column for analysis

data.drop("id",axis=1,inplace=True)

# The data is divided into three parts. Divided the features according to their category

features\_mean= list(data.columns[1:11])

features\_se= list(data.columns[11:20])

features\_worst=list(data.columns[21:31])

print(features\_mean)

print("-----------------------------------")

print(features\_se)

print("------------------------------------")

print(features\_worst)

# diagnosis column is a object type , map it to integer value

data['diagnosis']=data['diagnosis'].map({'M':1,'B':0})

#data.describe() # this will describe the all statistical function of our data

# get the frequency of cancer stages

sns.countplot(data['diagnosis'],label="Count")

# draw a correlation graph so that , can remove multi colinearity

corr = data[features\_mean].corr() # .corr is used for find corelation

plt.figure(figsize=(14,14))

sns.heatmap(corr, cbar = True, square = True, annot=True, fmt= '.2f',annot\_kws={'size': 15},

xticklabels= features\_mean, yticklabels= features\_mean,

cmap= 'coolwarm')

prediction\_var = ['texture\_mean','perimeter\_mean','smoothness\_mean','compactness\_mean','symmetry\_mean']

# now these are the variables which will use for prediction

#now split the data into train and test

train, test = train\_test\_split(data, test\_size = 0.3)# in this our main data is splitted into train and test

# we can check their dimension

print(train.shape)

print(test.shape)

train\_X = train[prediction\_var]# taking the training data input

train\_y=train.diagnosis# This is output of our training data

# same to do for test

test\_X= test[prediction\_var] # taking test data inputs

test\_y =test.diagnosis #output value of test dat

model=RandomForestClassifier(n\_estimators=100)# a simple random forest model

model.fit(train\_X,train\_y)# now fit the model for traiing data

prediction=model.predict(test\_X)# predict for the test data

# prediction will contain the predicted value by our model predicted values of dignosis column for test inputs

print ("accuracy of randomforest")

print (accuracy\_score(prediction,test\_y))# to check the accuracy

# here will use accuracy measurement between the predicted value and the test output values

model = svm.SVC()

model.fit(train\_X,train\_y)

prediction=model.predict(test\_X)

print ("accuracy of SVM")

print (accuracy\_score(prediction,test\_y))

#OUTPUT::

Python 3.7.4 (default, Aug 9 2019, 18:34:13) [MSC v.1915 64 bit (AMD64)]

Type "copyright", "credits" or "license" for more information.

IPython 7.8.0 -- An enhanced Interactive Python.

runfile('C:/Users/ASUS/Desktop/PR\_Lab/a3/Cancer\_Prediction.py', wdir='C:/Users/ASUS/Desktop/PR\_Lab/a3')

id diagnosis ... symmetry\_worst fractal\_dimension\_worst

0 842302 M ... 0.4601 0.11890

1 842517 M ... 0.2750 0.08902

[2 rows x 32 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 569 entries, 0 to 568

Data columns (total 32 columns):

id 569 non-null int64

diagnosis 569 non-null object

radius\_mean 569 non-null float64

texture\_mean 569 non-null float64

perimeter\_mean 569 non-null float64

area\_mean 569 non-null float64

smoothness\_mean 569 non-null float64

compactness\_mean 569 non-null float64

concavity\_mean 569 non-null float64

concave points\_mean 569 non-null float64

symmetry\_mean 569 non-null float64

fractal\_dimension\_mean 569 non-null float64

radius\_se 569 non-null float64

texture\_se 569 non-null float64

perimeter\_se 569 non-null float64

area\_se 569 non-null float64

smoothness\_se 569 non-null float64

compactness\_se 569 non-null float64

concavity\_se 569 non-null float64

concave points\_se 569 non-null float64

symmetry\_se 569 non-null float64

fractal\_dimension\_se 569 non-null float64

radius\_worst 569 non-null float64

texture\_worst 569 non-null float64

perimeter\_worst 569 non-null float64

area\_worst 569 non-null float64

smoothness\_worst 569 non-null float64

compactness\_worst 569 non-null float64

concavity\_worst 569 non-null float64

concave points\_worst 569 non-null float64

symmetry\_worst 569 non-null float64

fractal\_dimension\_worst 569 non-null float64

dtypes: float64(30), int64(1), object(1)

memory usage: 142.4+ KB

Traceback (most recent call last):

File "<ipython-input-1-07203d9ca7f5>", line 1, in <module>

runfile('C:/Users/ASUS/Desktop/PR\_Lab/a3/Cancer\_Prediction.py', wdir='C:/Users/ASUS/Desktop/PR\_Lab/a3')

File "C:\Users\ASUS\Anaconda3\lib\site-packages\spyder\_kernels\customize\spydercustomize.py", line 827, in runfile

execfile(filename, namespace)

File "C:\Users\ASUS\Anaconda3\lib\site-packages\spyder\_kernels\customize\spydercustomize.py", line 110, in execfile

exec(compile(f.read(), filename, 'exec'), namespace)

File "C:/Users/ASUS/Desktop/PR\_Lab/a3/Cancer\_Prediction.py", line 24, in <module>

data.drop("Unnamed: 32",axis=1,inplace=True) # for dropping or the coloumn Unnamed: 32

File "C:\Users\ASUS\Anaconda3\lib\site-packages\pandas\core\frame.py", line 4102, in drop

errors=errors,

File "C:\Users\ASUS\Anaconda3\lib\site-packages\pandas\core\generic.py", line 3914, in drop

obj = obj.\_drop\_axis(labels, axis, level=level, errors=errors)

File "C:\Users\ASUS\Anaconda3\lib\site-packages\pandas\core\generic.py", line 3946, in \_drop\_axis

new\_axis = axis.drop(labels, errors=errors)

File "C:\Users\ASUS\Anaconda3\lib\site-packages\pandas\core\indexes\base.py", line 5340, in drop

raise KeyError("{} not found in axis".format(labels[mask]))

KeyError: "['Unnamed: 32'] not found in axis"

runfile('C:/Users/ASUS/Desktop/PR\_Lab/a3/Cancer\_Prediction.py', wdir='C:/Users/ASUS/Desktop/PR\_Lab/a3')

id diagnosis ... symmetry\_worst fractal\_dimension\_worst

0 842302 M ... 0.4601 0.11890

1 842517 M ... 0.2750 0.08902

[2 rows x 32 columns]

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 569 entries, 0 to 568

Data columns (total 32 columns):

id 569 non-null int64

diagnosis 569 non-null object

radius\_mean 569 non-null float64

texture\_mean 569 non-null float64

perimeter\_mean 569 non-null float64

area\_mean 569 non-null float64

smoothness\_mean 569 non-null float64

compactness\_mean 569 non-null float64

concavity\_mean 569 non-null float64

concave points\_mean 569 non-null float64

symmetry\_mean 569 non-null float64

fractal\_dimension\_mean 569 non-null float64

radius\_se 569 non-null float64

texture\_se 569 non-null float64

perimeter\_se 569 non-null float64

area\_se 569 non-null float64

smoothness\_se 569 non-null float64

compactness\_se 569 non-null float64

concavity\_se 569 non-null float64

concave points\_se 569 non-null float64

symmetry\_se 569 non-null float64

fractal\_dimension\_se 569 non-null float64

radius\_worst 569 non-null float64

texture\_worst 569 non-null float64

perimeter\_worst 569 non-null float64

area\_worst 569 non-null float64

smoothness\_worst 569 non-null float64

compactness\_worst 569 non-null float64

concavity\_worst 569 non-null float64

concave points\_worst 569 non-null float64

symmetry\_worst 569 non-null float64

fractal\_dimension\_worst 569 non-null float64

dtypes: float64(30), int64(1), object(1)

memory usage: 142.4+ KB

['radius\_mean', 'texture\_mean', 'perimeter\_mean', 'area\_mean', 'smoothness\_mean', 'compactness\_mean', 'concavity\_mean', 'concave points\_mean', 'symmetry\_mean', 'fractal\_dimension\_mean']

-----------------------------------

['radius\_se', 'texture\_se', 'perimeter\_se', 'area\_se', 'smoothness\_se', 'compactness\_se', 'concavity\_se', 'concave points\_se', 'symmetry\_se']

------------------------------------

['radius\_worst', 'texture\_worst', 'perimeter\_worst', 'area\_worst', 'smoothness\_worst', 'compactness\_worst', 'concavity\_worst', 'concave points\_worst', 'symmetry\_worst', 'fractal\_dimension\_worst']

(398, 31)

(171, 31)

accuracy of randomforest

0.9064327485380117

accuracy of SVM

0.8888888888888888

C:\Users\ASUS\Anaconda3\lib\site-packages\sklearn\svm\base.py:193: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.

"avoid this warning.", FutureWarning)



