\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Import Dataset \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#importing the necessary libraries**

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)

import seaborn as sns

import matplotlib.pyplot as plt #matplotlib is used for plot the graphs,

%matplotlib inline

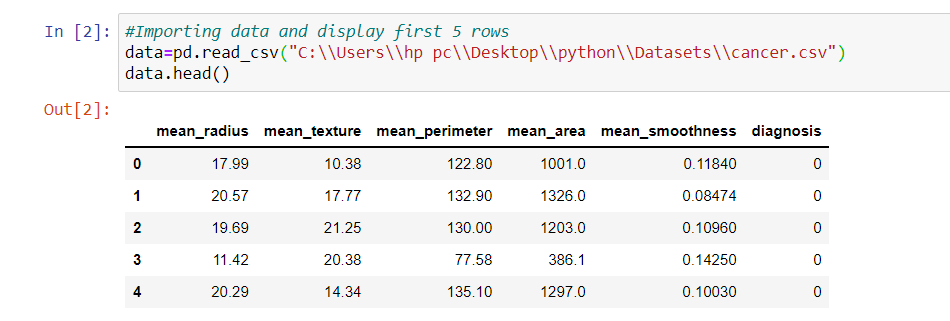
from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

**#importing the data and print first 5 rows**

data=pd.read\_csv("C:\\Users\\hp pc\\Desktop\\python\\Datasets\\cancer.csv")

data.head()



**#no of rows and columns of data**

data.shape

data.info()

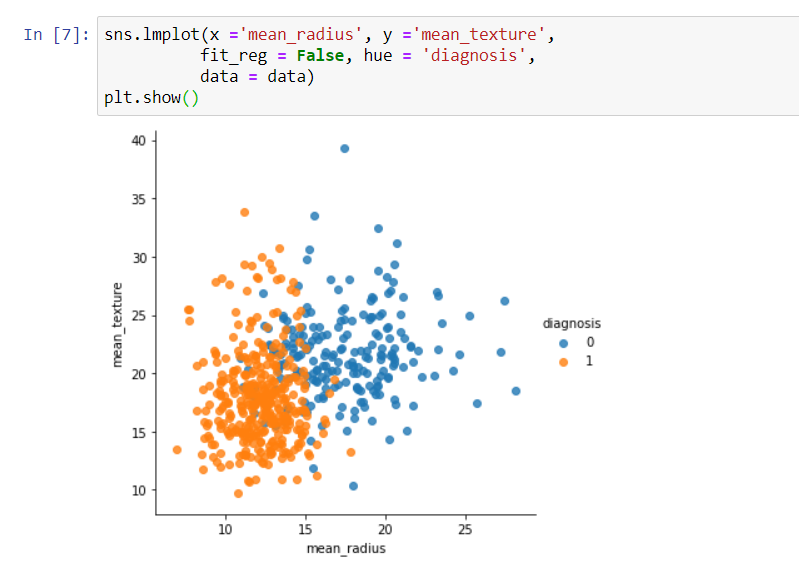
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data visualization \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

sns.lmplot(x ='mean\_radius', y ='mean\_texture',

fit\_reg = False, hue = 'diagnosis',

data = data)

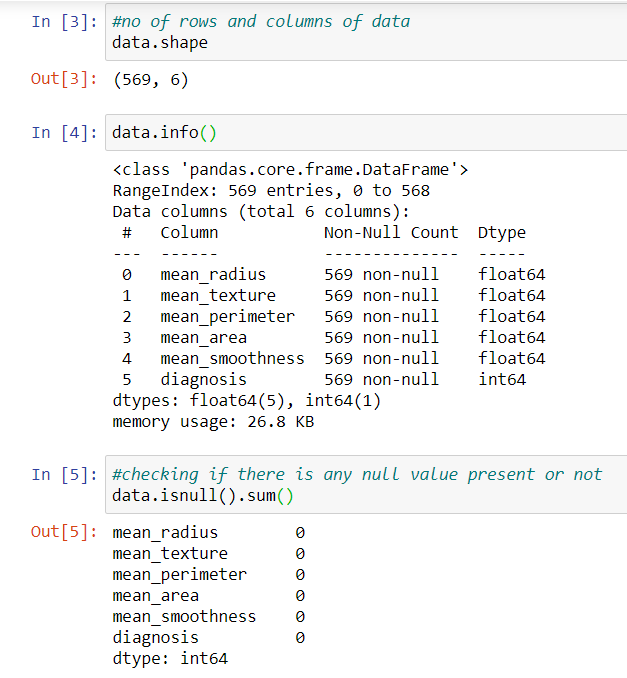
plt.show()



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data Cleaning \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#checking if there is any null value present or not**

data.isnull().sum()



**#finding the pairwise correlation of all columns in the dataframe**

corr=data.corr()

corr.nlargest(30,'diagnosis')['diagnosis']

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data Preprocessing \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#creating independent and dependent variable**

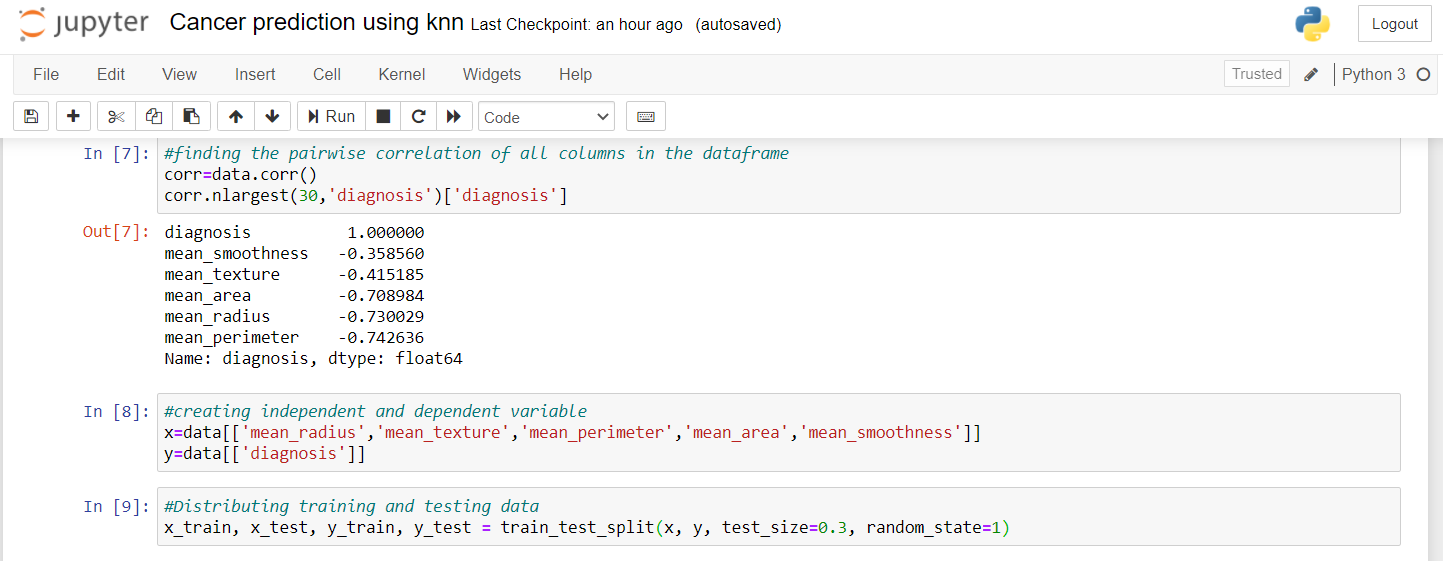
x=data[['mean\_radius','mean\_texture','mean\_perimeter','mean\_area','mean\_smoothness']]

y=data[['diagnosis']]

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Data Splitting \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#Distributing training and testing data**

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



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Create Model \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#building KNN model and fitting data into it.**

from sklearn.metrics import accuracy\_score

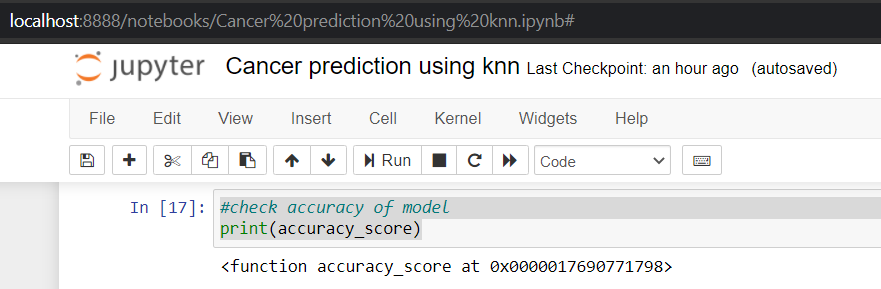
model = KNeighborsClassifier(n\_neighbors=8)

model.fit(x\_train, y\_train)

predict = model.predict(x\_test)

accuracy\_score(predict,y\_test)

print(accuracy\_score)



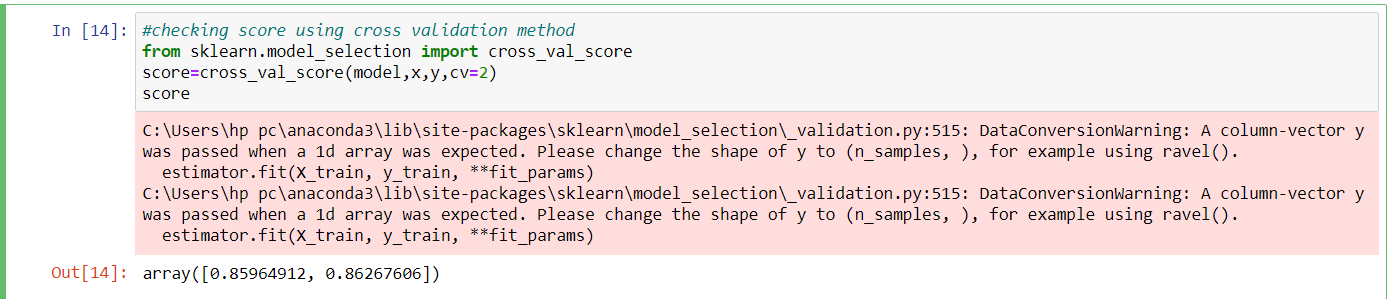
model.score(x\_train,y\_train)

**#checking score using cross validation method**

from sklearn.model\_selection import cross\_val\_score

score=cross\_val\_score(model,x,y,cv=2)

score

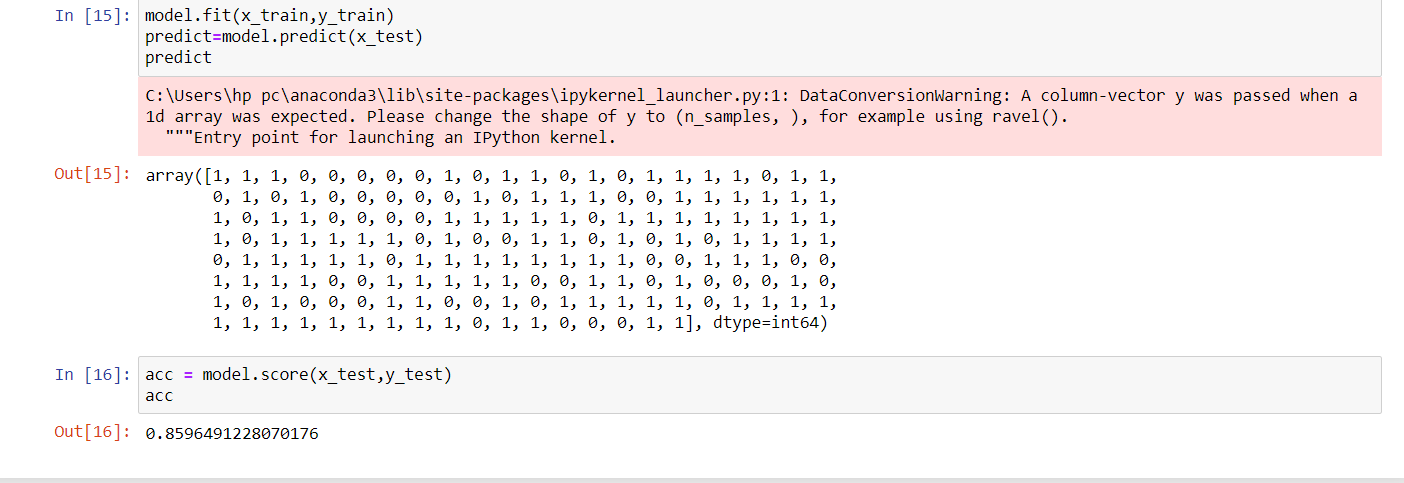


**#Display prediction**

model.fit(x\_train,y\_train)

predict=model.predict(x\_test)

predict



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Accuracy Checking \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**#accuracy of model of test data**

acc = model.score(x\_test,y\_test)

acc