PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY, THANE

(ALL PROGRAMS ACCREDITED BY NBA)

APSIT Skills

We Set Trends

APSIT SKILLS INTERNSHIP – PROJECT REPORT

PO COMPUTER

Project Batch: B1

Team Members Names:

Team Leader: - Rahul J. Bhiwande

Member 1: - Saurabh V. Yadav

Member 2: - Urvi V. Aryamane

Technology Selected:

Python

Project Topic Name:

Breast Cancer Classification using Machine Learning

Problem statement:

During their life, among 8% of women are diagnosed with Breast cancer (BC), after lung cancer, BC is the second popular cause of death in both developed and undeveloped worlds. BC is characterized by the mutation of genes, constant pain, changes in the size, color(redness), skin texture of breasts. Today, Machine Learning (ML) techniques are being broadly used in the breast cancer classification problem. They provide high classification accuracy and effective diagnostic capabilities. The early diagnosis of BC can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. Further accurate classification of benign tumors can prevent patients undergoing unnecessary treatments. To observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection. The goal is to classify whether the breast cancer is benign or malignant.

Detailed Workflow:

- Functions/modules used for the program:
    1. NumPy
    2. Pandas
    3. Matplotlib
    4. Scikit learn

- Flow of the code:

Step 1: Data Preparation

We will use the UCI Machine Learning Repository for breast cancer dataset.

Step 2: Data Exploration

We will import the necessary libraries and use the library functions to examine the available data.

Step 3: Handling the Categorical Data

The data in the dataset is not always in the numeric form. Further processing requires the data to be in numeric form. Hence, in this step, we will convert the non-numeric data into numeric data.

Step 4: Splitting the dataset

The data we use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. We have the test dataset (or subset) in order to test our model's prediction on this subset.

Step 5: Feature Scaling

Most of the times, the data will be varying over a large range. Hence, this data is needed to be brought in a lower range using normalization.

Step 6: Model Selection, classification and visualization

We will be using various classification models. In our dataset we have the outcome variable or Dependent variable i.e. Y having only two set of values, either M (Malign) or B(Benign). So, we will use Classification algorithm of supervised learning. Also, the data will be visualized in the form of graphs, using python libraries.

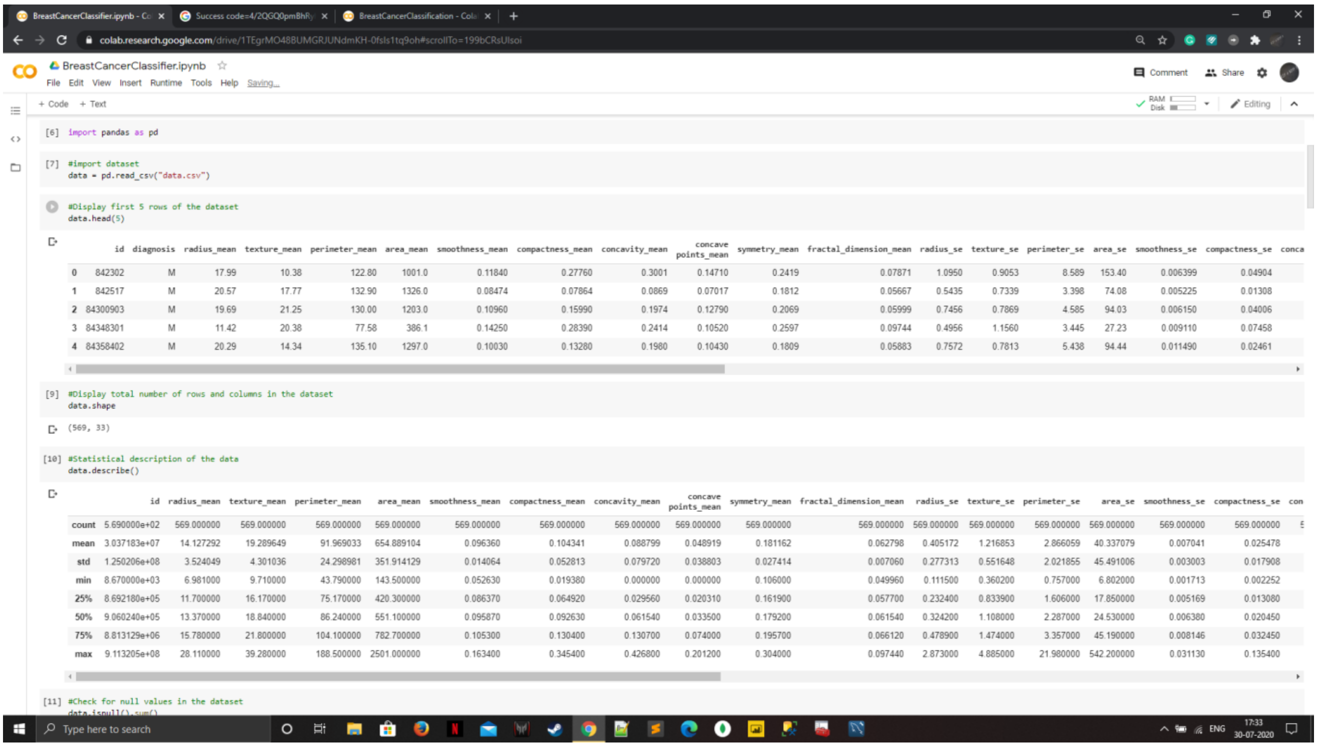
GitHub / Drive link of project code:

Data model:

https://colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsIs1tq9oh

Output Screenshots:

Data Model:


BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help Saving...
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
ENG 17:33 30-07-2020

```
[6] import pandas as pd
```

```
[7] #import dataset
    data = pd.read_csv("data.csv")
```

```
#Display first 5 rows of the dataset
data.head(5)
```

	id	diagnosis	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	conca
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710	0.2419	0.07871	1.0950	0.9053	8.589	153.40	0.006399	0.04904	
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398	74.08	0.005225	0.01308	
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790	0.2069	0.05999	0.7456	0.7869	4.585	94.03	0.006150	0.04006	
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520	0.2597	0.09744	0.4956	1.1560	3.445	27.23	0.009110	0.07458	
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430	0.1809	0.05883	0.7572	0.7813	5.438	94.44	0.011490	0.02461	

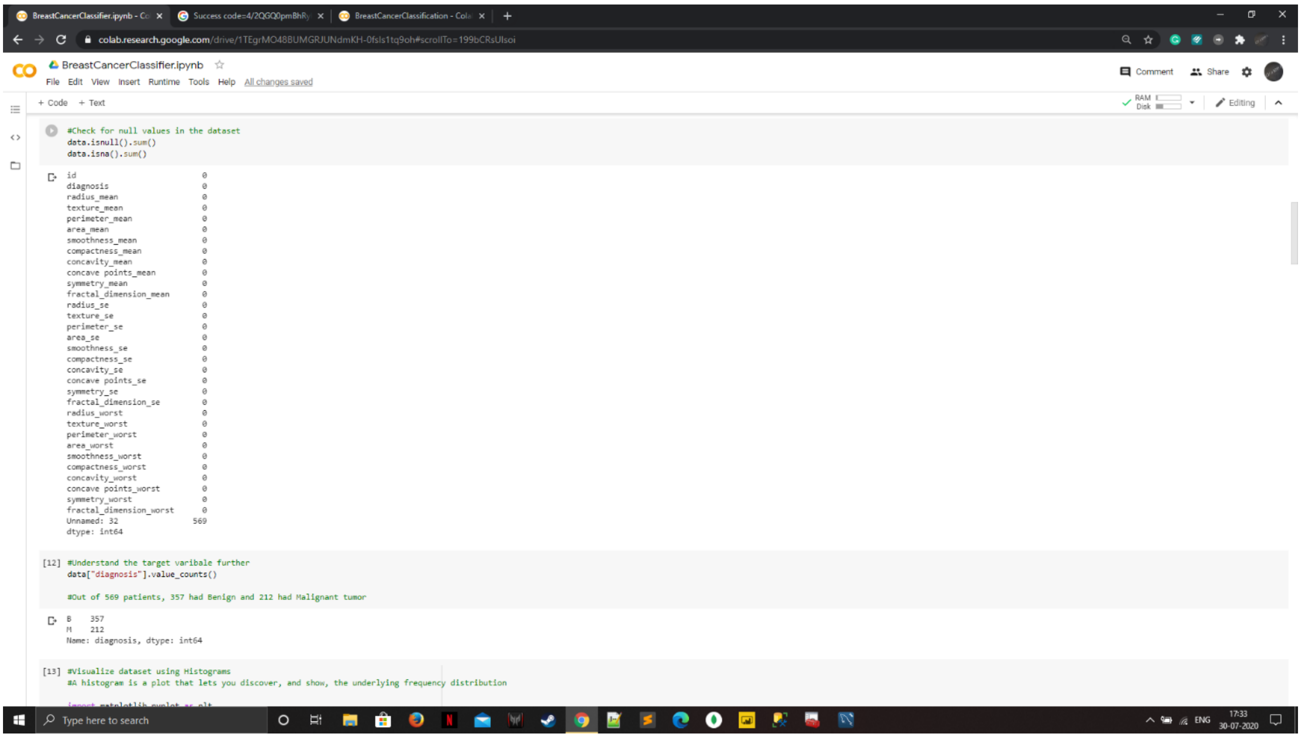
```
[9] #Display total number of rows and columns in the dataset
    data.shape
```

```
(569, 33)
```

```
[10] #Statistical description of the data
     data.describe()
```

	id	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	con
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	5
mean	3.037183e+07	14.127292	19.289649	91.969033	654.889104	0.096360	0.104341	0.088799	0.048919	0.181162	0.062798	0.405172	1.216853	2.866059	40.337079	0.007041	0.025478	
std	1.250206e+08	3.524049	4.301036	24.298981	351.914129	0.014064	0.052813	0.079720	0.038803	0.027414	0.007060	0.277313	0.551648	2.021855	45.491006	0.003003	0.017908	
min	8.670000e+03	6.981000	9.710000	43.790000	143.500000	0.052630	0.019380	0.000000	0.000000	0.106000	0.049960	0.111500	0.360200	0.757000	6.802000	0.001713	0.002252	
25%	8.692180e+05	11.700000	16.170000	75.170000	420.300000	0.086370	0.064920	0.029560	0.020310	0.161900	0.057700	0.232400	0.833900	1.606000	17.850000	0.005169	0.013080	
50%	9.060240e+05	13.370000	18.840000	86.240000	551.100000	0.095870	0.092630	0.061540	0.033500	0.179200	0.061540	0.324200	1.108000	2.287000	24.530000	0.006380	0.020450	
75%	8.813129e+06	15.780000	21.800000	104.100000	782.700000	0.105300	0.130400	0.130700	0.074000	0.195700	0.066120	0.478900	1.474000	3.357000	45.190000	0.008146	0.032450	
max	9.113205e+08	28.110000	39.280000	188.500000	2501.000000	0.163400	0.345400	0.426800	0.201200	0.304000	0.097440	2.873000	4.885000	21.980000	542.200000	0.031130	0.135400	

```
[11] #Check for null values in the dataset
     data.isnull().sum()
```


BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
ENG 17:33 30-07-2020

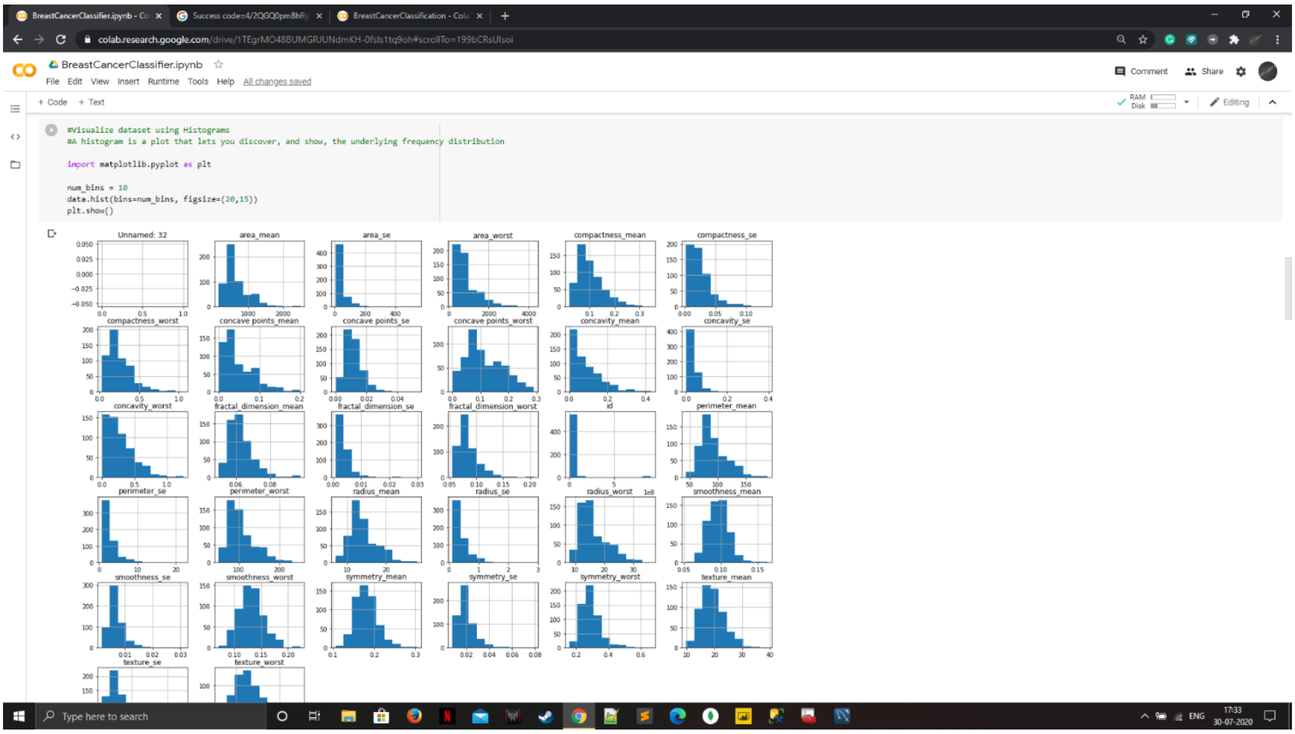
```
#Check for null values in the dataset
data.isnull().sum()
data.isna().sum()
```

```
id                         0
diagnosis                  0
radius_mean                0
texture_mean               0
perimeter_mean             0
area_mean                  0
smoothness_mean            0
compactness_mean           0
concavity_mean             0
concave points_mean        0
symmetry_mean              0
fractal_dimension_mean     0
radius_se                  0
texture_se                 0
perimeter_se               0
area_se                    0
smoothness_se              0
compactness_se             0
concavity_se               0
concave points_se          0
symmetry_se                0
fractal_dimension_se       0
radius_worst               0
texture_worst              0
perimeter_worst            0
area_worst                 0
smoothness_worst           0
compactness_worst          0
concavity_worst            0
concave points_worst       0
symmetry_worst             0
fractal_dimension_worst    0
Unnamed: 32              569
dtype: int64
```

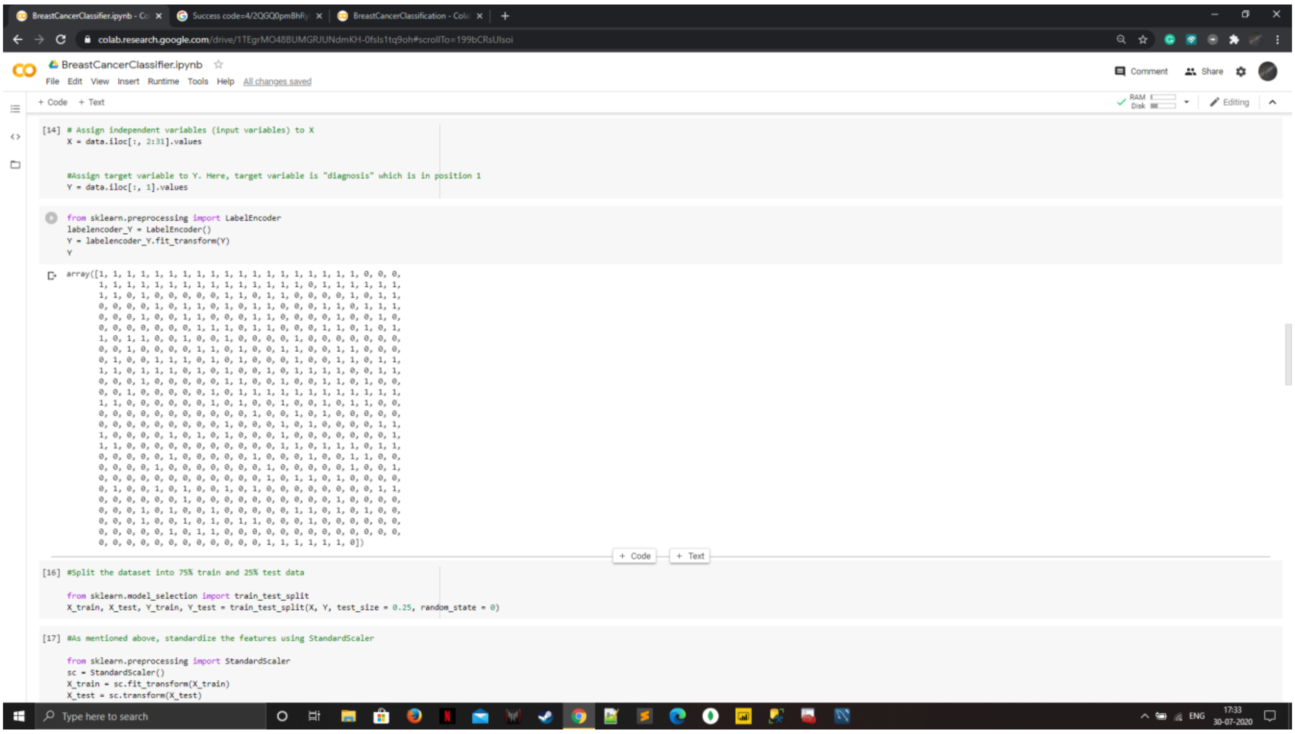
```
[12] #Understand the target varibale further
     data["diagnosis"].value_counts()

     #Out of 569 patients, 357 had Benign and 212 had Malignant tumor
```

```
B    357
M    212
Name: diagnosis, dtype: int64
```

```
[13] #Visualize dataset using Histograms
     #A histogram is a plot that lets you discover, and show, the underlying frequency distribution

     import matplotlib.pyplot as plt
```



```
#Visualize dataset using Histograms
#A histogram is a plot that lets you discover, and show, the underlying frequency distribution

import matplotlib.pyplot as plt

num_bins = 10
data.hist(bins=num_bins, figsize=(20,15))
plt.show()
```


BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
Type here to search
ENG 17:33 30-07-2020

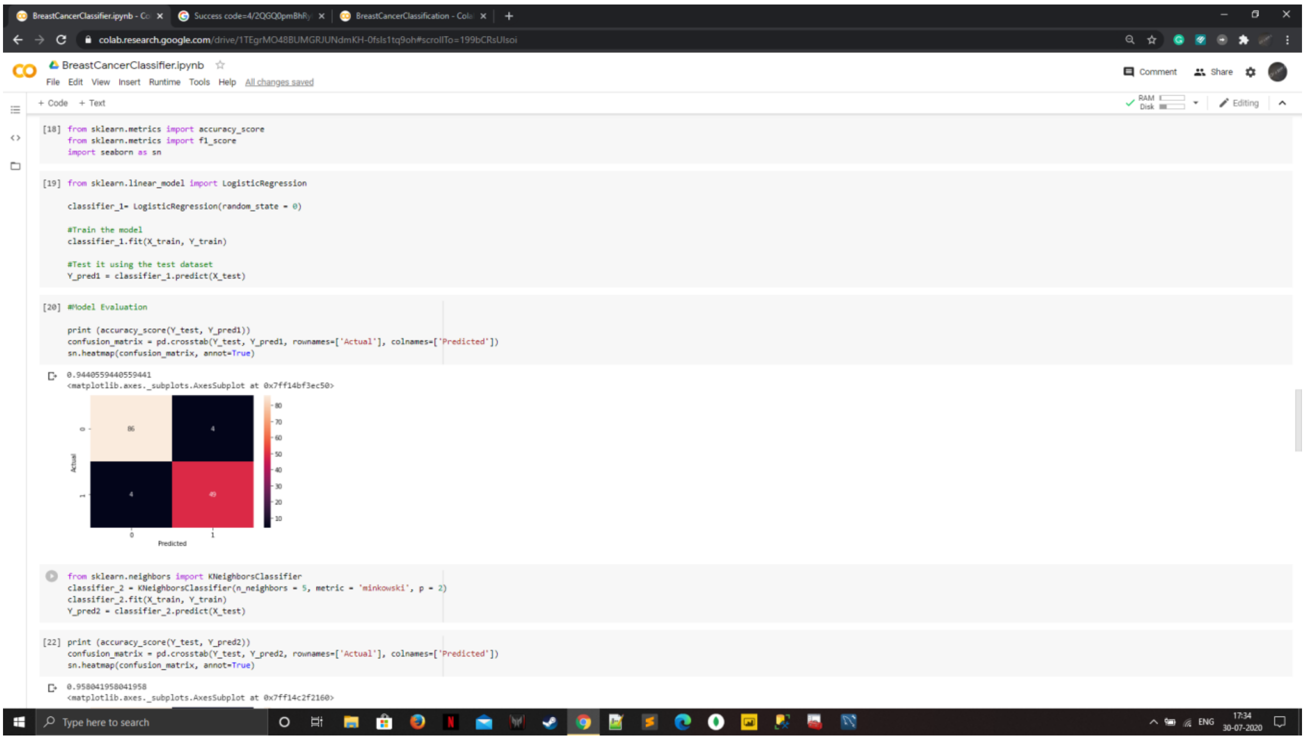
```
[14] # Assign independent variables (input variables) to X
     X = data.iloc[:, 2:31].values


     #Assign target variable to Y. Here, target variable is "diagnosis" which is in position 1
     Y = data.iloc[:, 1].values
```

```
from sklearn.preprocessing import LabelEncoder
labelencoder_Y = LabelEncoder()
Y = labelencoder_Y.fit_transform(Y)
Y
```

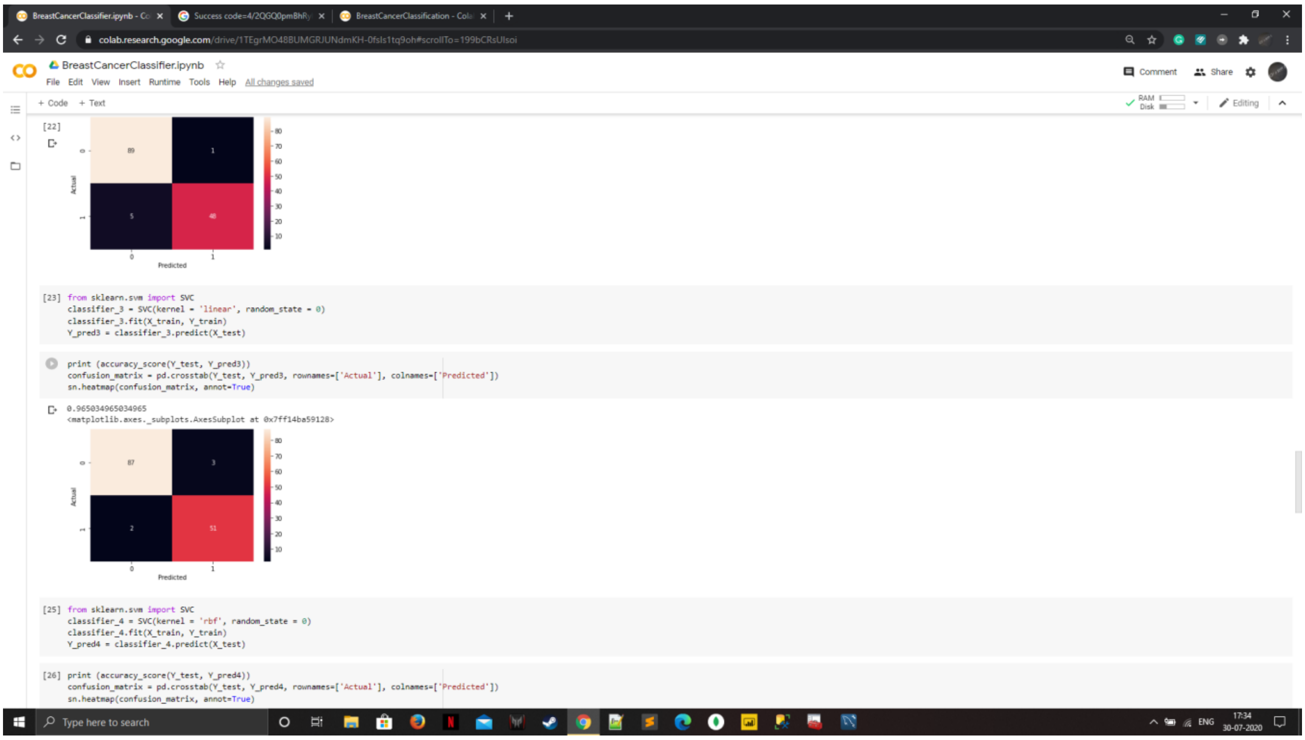
```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 1, 1,
       0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1,
       0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1,
       1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0,
       0, 1, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1,
       1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 1, 1,
       1, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0])
```

```
[16] #Split the dataset into 75% train and 25% test data

     from sklearn.model_selection import train_test_split
     X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

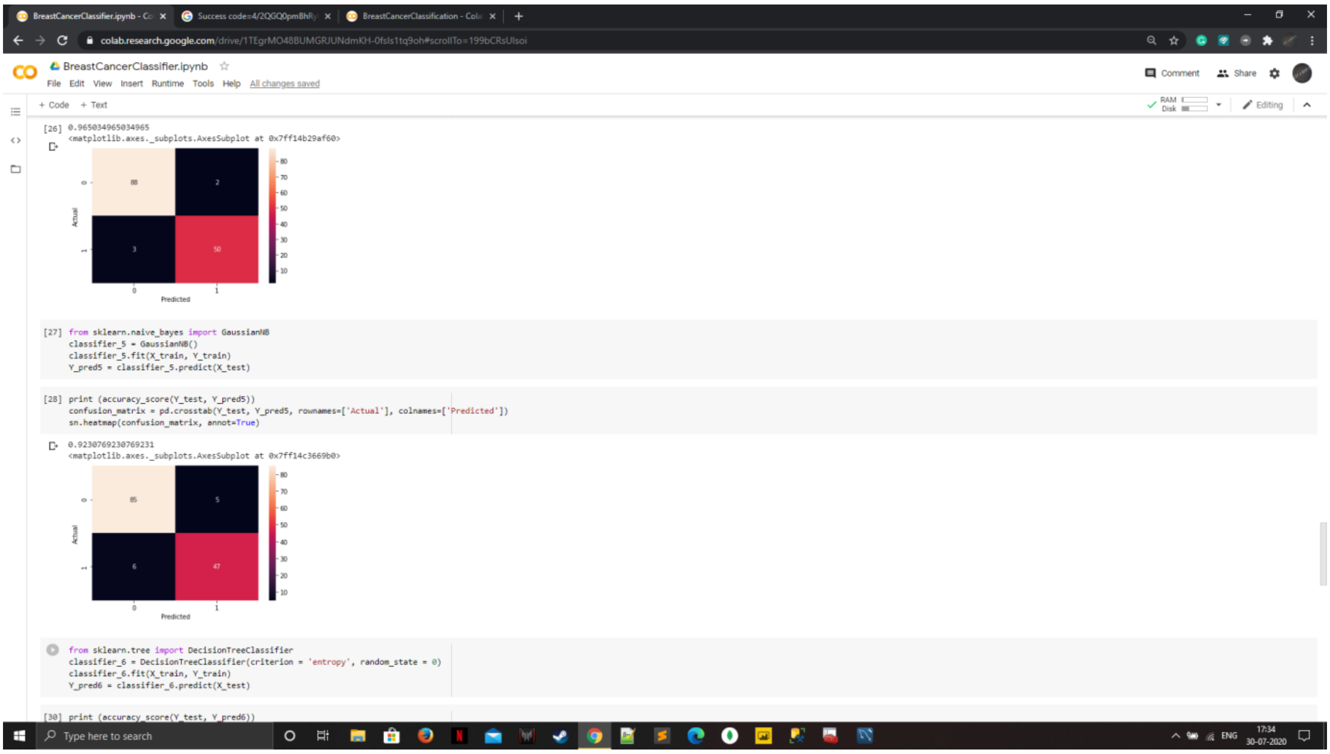
```
[17] #As mentioned above, standardize the features using StandardScaler

     from sklearn.preprocessing import StandardScaler
     sc = StandardScaler()
     X_train = sc.fit_transform(X_train)
     X_test = sc.transform(X_test)
```


BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
86
4
4
49
0
1
Actual
Predicted
80
70
60
50
40
30
20
10
Type here to search
ENG
17:34
30-07-2020

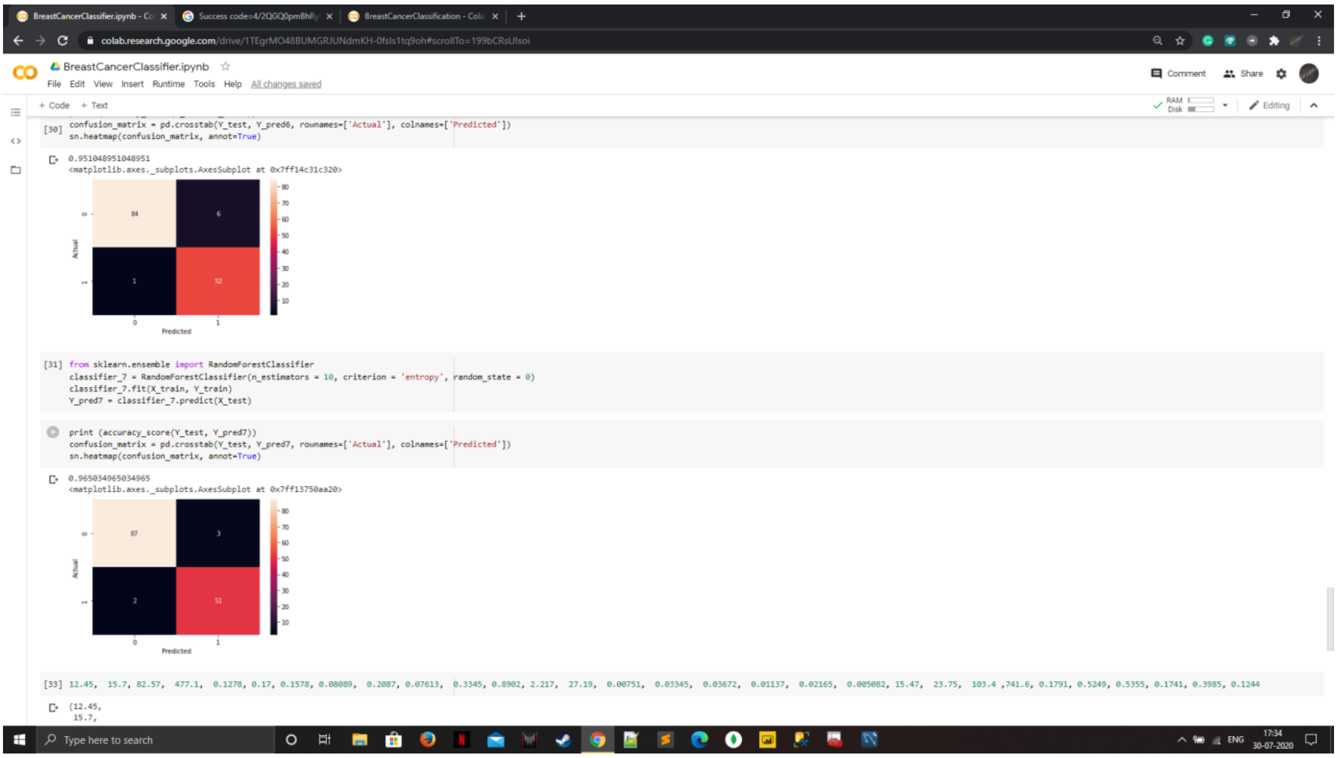
```
[18] from sklearn.metrics import accuracy_score
     from sklearn.metrics import f1_score
     import seaborn as sn

[19] from sklearn.linear_model import LogisticRegression

     classifier_1= LogisticRegression(random_state = 0)

     #Train the model
     classifier_1.fit(X_train, Y_train)

     #Test it using the test dataset
     Y_pred1 = classifier_1.predict(X_test)

[20] #Model Evaluation

     print (accuracy_score(Y_test, Y_pred1))
     confusion_matrix = pd.crosstab(Y_test, Y_pred1, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.9440559440559441
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14bf3ec50>

     from sklearn.neighbors import KNeighborsClassifier
     classifier_2 = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
     classifier_2.fit(X_train, Y_train)
     Y_pred2 = classifier_2.predict(X_test)

[22] print (accuracy_score(Y_test, Y_pred2))
     confusion_matrix = pd.crosstab(Y_test, Y_pred2, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.958041958041958
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14c2f2160>
```


BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment Share
+ Code + Text
RAM Disk
Editing
89
1
5
48
87
3
2
51
0
1
Actual
Predicted
80
70
60
50
40
30
20
10
Type here to search
ENG
17:34
30-07-2020

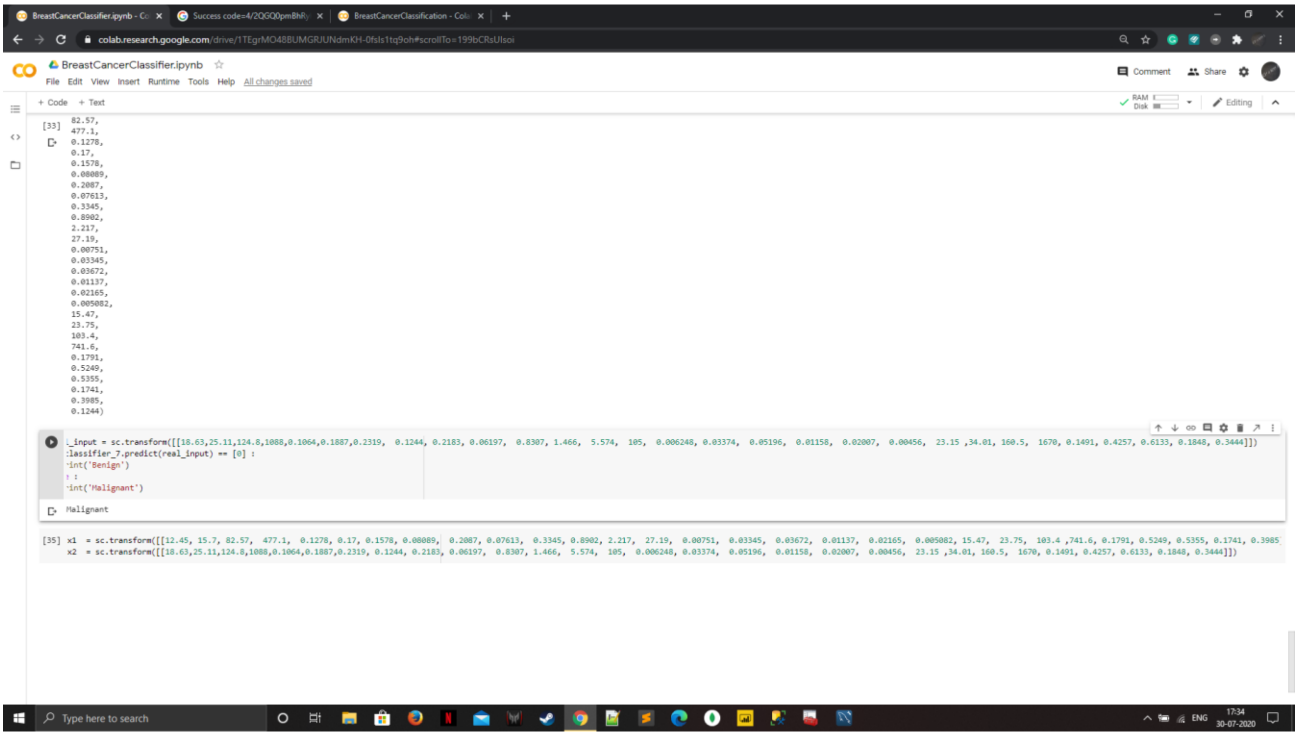
```
[22]

[23] from sklearn.svm import SVC
     classifier_3 = SVC(kernel = 'linear', random_state = 0)
     classifier_3.fit(X_train, Y_train)
     Y_pred3 = classifier_3.predict(X_test)

     print (accuracy_score(Y_test, Y_pred3))
     confusion_matrix = pd.crosstab(Y_test, Y_pred3, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.965034965034965
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14ba59128>

[25] from sklearn.svm import SVC
     classifier_4 = SVC(kernel = 'rbf', random_state = 0)
     classifier_4.fit(X_train, Y_train)
     Y_pred4 = classifier_4.predict(X_test)

[26] print (accuracy_score(Y_test, Y_pred4))
     confusion_matrix = pd.crosstab(Y_test, Y_pred4, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)
```


BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Actual
Predicted
0
1
88
2
3
50
85
5
6
47
80
70
60
50
40
30
20
10

```
[26] 0.965034965034965
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14b29af60>

[27] from sklearn.naive_bayes import GaussianNB
     classifier_5 = GaussianNB()
     classifier_5.fit(X_train, Y_train)
     Y_pred5 = classifier_5.predict(X_test)

[28] print (accuracy_score(Y_test, Y_pred5))
     confusion_matrix = pd.crosstab(Y_test, Y_pred5, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.9230769230769231
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14c3669b0>

     from sklearn.tree import DecisionTreeClassifier
     classifier_6 = DecisionTreeClassifier(criterion = 'entropy', random_state = 0)
     classifier_6.fit(X_train, Y_train)
     Y_pred6 = classifier_6.predict(X_test)

[30] print (accuracy_score(Y_test, Y_pred6))
```


BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
Actual
Predicted
0
1
84
6
1
52
87
3
2
51
80
70
60
50
40
30
20
10

```
[30] confusion_matrix = pd.crosstab(Y_test, Y_pred6, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.951048951048951
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff14c31c320>

[31] from sklearn.ensemble import RandomForestClassifier
     classifier_7 = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 0)
     classifier_7.fit(X_train, Y_train)
     Y_pred7 = classifier_7.predict(X_test)

     print (accuracy_score(Y_test, Y_pred7))
     confusion_matrix = pd.crosstab(Y_test, Y_pred7, rownames=['Actual'], colnames=['Predicted'])
     sn.heatmap(confusion_matrix, annot=True)

     0.965034965034965
     <matplotlib.axes._subplots.AxesSubplot at 0x7ff13750aa20>

[33] 12.45,  15.7, 82.57,  477.1,  0.1278, 0.17, 0.1578, 0.08089,  0.2087, 0.07613,  0.3345, 0.8902, 2.217,  27.19,  0.00751,  0.03345,  0.03672,  0.01137,  0.02165,  0.005082, 15.47,  23.75,  103.4 ,741.6, 0.1791, 0.5249, 0.5355, 0.1741, 0.3985, 0.1244

     (12.45,
      15.7,
```

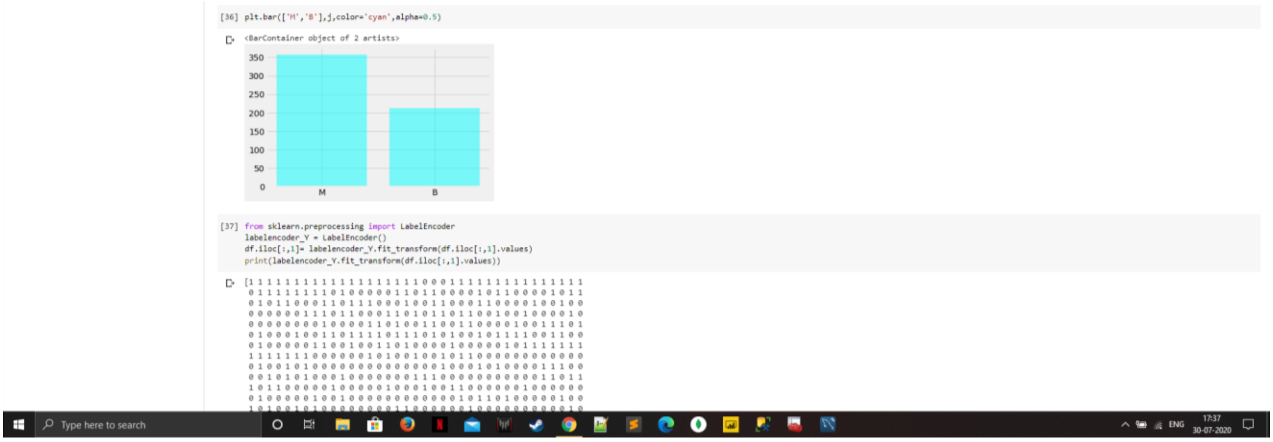

BreastCancerClassifier.ipynb - Co
Success code=4/2QGQ0pmBhRy
BreastCancerClassification - Cola
colab.research.google.com/drive/1TEgrMO48BUMGRJUNdmKH-0fsls1tq9oh#scrollTo=199bCRsUlsoi
BreastCancerClassifier.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Type here to search
ENG
17:34
30-07-2020

```
[33]  82.57,
      477.1,
      0.1278,
      0.17,
      0.1578,
      0.08089,
      0.2087,
      0.07613,
      0.3345,
      0.8902,
      2.217,
      27.19,
      0.00751,
      0.03345,
      0.03672,
      0.01137,
      0.02165,
      0.005082,
      15.47,
      23.75,
      103.4,
      741.6,
      0.1791,
      0.5249,
      0.5355,
      0.1741,
      0.3985,
      0.1244)
```

```
l_input = sc.transform([[18.63,25.11,124.8,1088,0.1064,0.1887,0.2319,  0.1244, 0.2183, 0.06197,  0.8307, 1.466,  5.574,  105,  0.006248, 0.03374,  0.05196,  0.01158,  0.02007,  0.00456,  23.15 ,34.01, 160.5,  1670, 0.1491, 0.4257, 0.6133, 0.1848, 0.3444]])
:lassifier_7.predict(real_input) == [0] :
·int('Benign')
: :
·int('Malignant')
```

```
Malignant
```

```
[35] x1  = sc.transform([[12.45, 15.7, 82.57,  477.1,  0.1278, 0.17, 0.1578, 0.08089,  0.2087, 0.07613,  0.3345, 0.8902, 2.217,  27.19,  0.00751,  0.03345,  0.03672,  0.01137,  0.02165,  0.005082, 15.47,  23.75,  103.4 ,741.6, 0.1791, 0.5249, 0.5355, 0.1741, 0.3985
     x2  = sc.transform([[18.63,25.11,124.8,1088,0.1064,0.1887,0.2319, 0.1244, 0.2183, 0.06197,  0.8307, 1.466,  5.574,  105,  0.006248, 0.03374,  0.05196,  0.01158,  0.02007,  0.00456,  23.15 ,34.01, 160.5,  1670, 0.1491, 0.4257, 0.6133, 0.1848, 0.3444]])
```

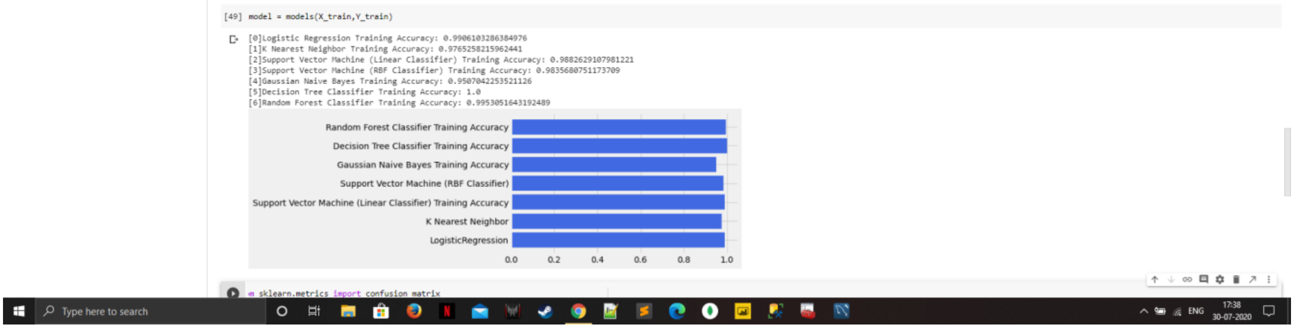


```
[36] plt.bar(['M','B'],j,color='cyan',alpha=0.5)
```

```
<BarContainer object of 2 artists>
```

```
[37] from sklearn.preprocessing import LabelEncoder
     labelencoder_Y = LabelEncoder()
     df.iloc[:,1]= labelencoder_Y.fit_transform(df.iloc[:,1].values)
     print(labelencoder_Y.fit_transform(df.iloc[:,1].values))
```

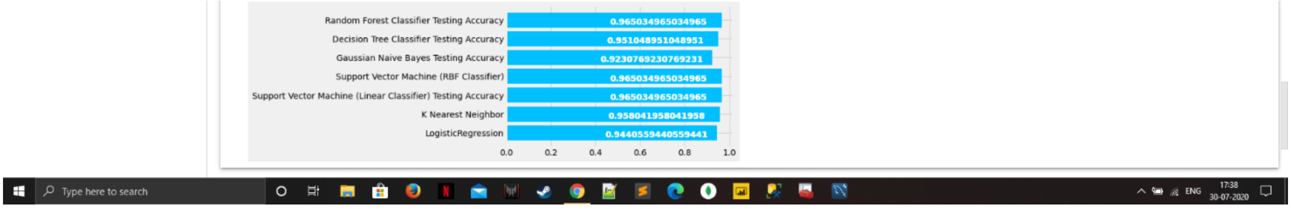
```
[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 0 1 1 1 1 1 1 1 1 0 1 0 0 0 0 0 1 1 0 1 1 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1
 0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0 0 0 0 1 0 0 1 0 0
 0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0
 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 1 1 1 0 1
 0 1 0 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 0 0 1 1 0 0
 0 1 0 0 0 0 0 1 1 0 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0 0 1 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0
 0 0 1 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0
 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0
 1 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
```


Random Forest Classifier Training Accuracy
Decision Tree Classifier Training Accuracy
Gaussian Naive Bayes Training Accuracy
Support Vector Machine (RBF Classifier)
Support Vector Machine (Linear Classifier) Training Accuracy
K Nearest Neighbor
LogisticRegression
0.0
0.2
0.4
0.6
0.8
1.0
Type here to search
ENG
17:38
30-07-2020

```
[49] model = models(X_train,Y_train)
```

```
[0]Logistic Regression Training Accuracy: 0.9906103286384976
[1]K Nearest Neighbor Training Accuracy: 0.9765258215962441
[2]Support Vector Machine (Linear Classifier) Training Accuracy: 0.9882629107981221
[3]Support Vector Machine (RBF Classifier) Training Accuracy: 0.9835680751173709
[4]Gaussian Naive Bayes Training Accuracy: 0.9507042253521126
[5]Decision Tree Classifier Training Accuracy: 1.0
[6]Random Forest Classifier Training Accuracy: 0.9953051643192489
```

```
m sklearn.metrics import confusion matrix
```


Random Forest Classifier Testing Accuracy
0.965034965034965
Decision Tree Classifier Testing Accuracy
0.951048951048951
Gaussian Naive Bayes Testing Accuracy
0.923076923076923l
Support Vector Machine (RBF Classifier)
0.965034965034965
Support Vector Machine (Linear Classifier) Testing Accuracy
0.965034965034965
K Nearest Neighbor
0.958041958041958
LogisticRegression
0.944055944055944l
0.0
0.2
0.4
0.6
0.8
1.0
Type here to search
ENG
17:38
30-07-2020

Acknowledgment:

We have great pleasure in presenting our topic: Breast Cancer Classification using Machine Learning. We take this opportunity to express our sincere thanks towards our guide Prof. Amol R. Kalugade Department of Computer Engineering, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project. We also thank the coordinators of APSIT for their invaluable help rendered during the course of this work.

Student Name1: Urvi Aryamane

Student ID1: 15102001

Student Name2: Saurabh Yadav

Student ID2: 15102021

Student Name3: Rahul Bhiwande

Student ID3: 15102068

Reference:

1. http://archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+%28diagnostic%29
2. https://www.udemy.com/course/machinelearning/learn/lecture/19041246?start=1#overview