Day 54

DIY

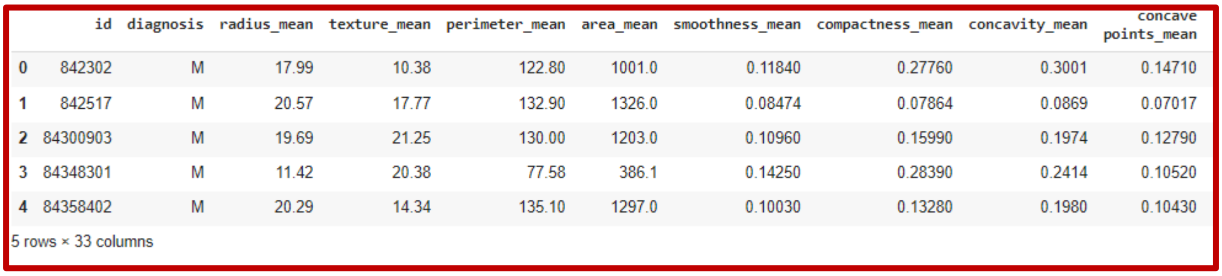
Q1. Problem Statement: Confusion Matrix

Load the 'Breast_Cancer_Dataset.csv' dataset into a DataFrame and perform the following tasks:

1. Identify the null values and remove the null rows and columns by using the `dropna()` function
2. Encode the 'diagnosis' column using the `LabelEncoder()` to convert non-numerical values to numerical labels
3. Considering the 'diagnosis' column as the target, separate the target variable and the feature vectors
4. Split the dataset into the training set and test set in a 70:30 ratio
5. Building a Logistic Regression, Naive Bayes, Decision Tree (CART), K-NN, SVM, and RandomForestClassifier models; Also, print their accuracies
6. Calculate and plot the confusion matrix

Hint: You can declare the algorithms in a list and iterate through them to build their respective models and calculate their accuracies using a for loop.

Dataset:

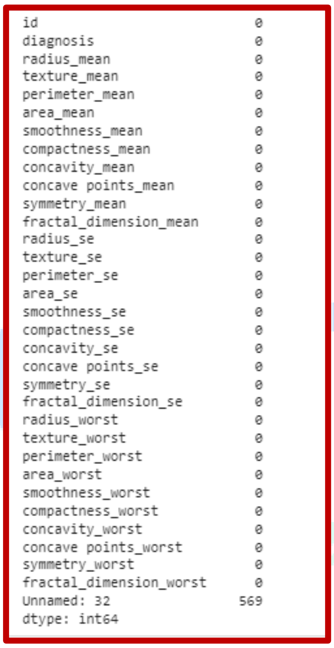


	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	M	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	842517	M	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	84300903	M	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	84348301	M	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	84358402	M	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

5 rows × 33 columns

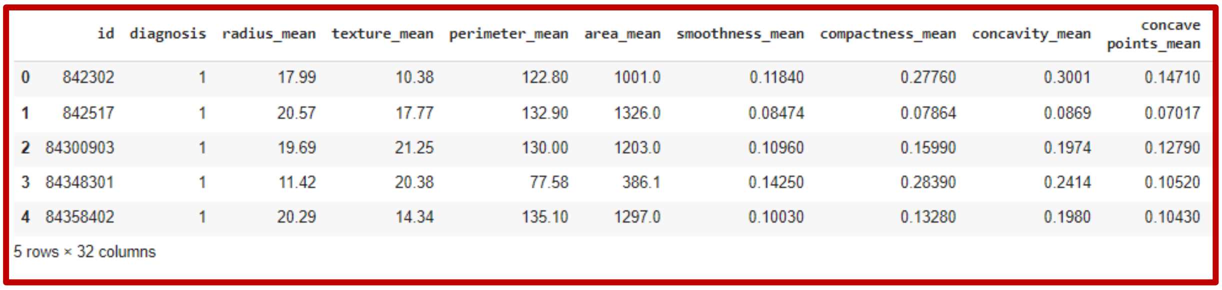
Sample Output:

1. Identify the null values and remove the null rows and columns by using the dropna() function



```
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
```

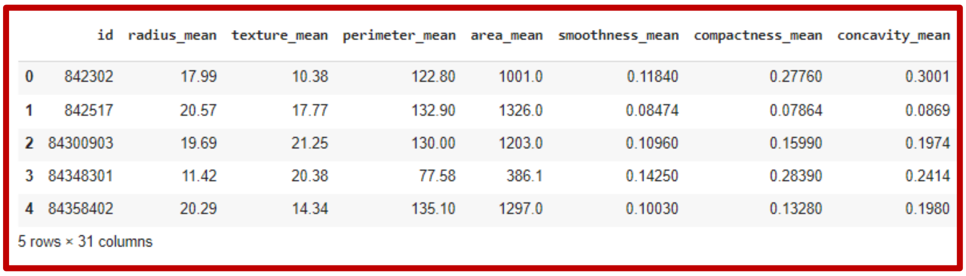
2. Encode the 'diagnosis' column using the `LabelEncoder()` to convert non-numerical values to numerical labels



	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean
0	842302	1	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001	0.14710
1	842517	1	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869	0.07017
2	84300903	1	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974	0.12790
3	84348301	1	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414	0.10520
4	84358402	1	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980	0.10430

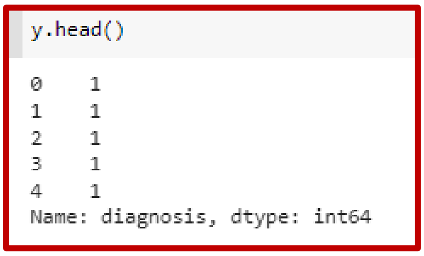
5 rows × 32 columns

3. Considering the 'diagnosis' column as the target, separate the target variable and the feature vectors



	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	842302	17.99	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	842517	20.57	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	84300903	19.69	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	84348301	11.42	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	84358402	20.29	14.34	135.10	1297.0	0.10030	0.13280	0.1980

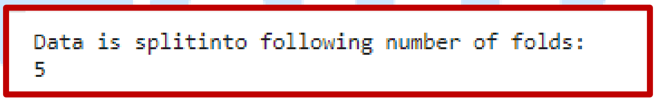
5 rows × 31 columns



```
y.head()
```

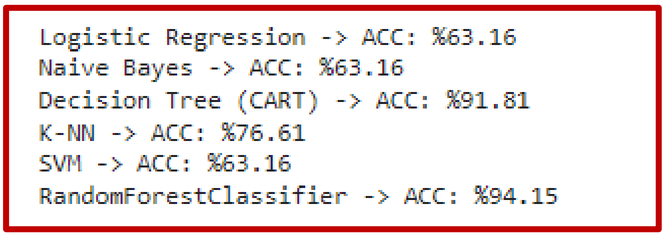
```
0    1
1    1
2    1
3    1
4    1
Name: diagnosis, dtype: int64
```

4. Split the data into five folds using `KFold()` function



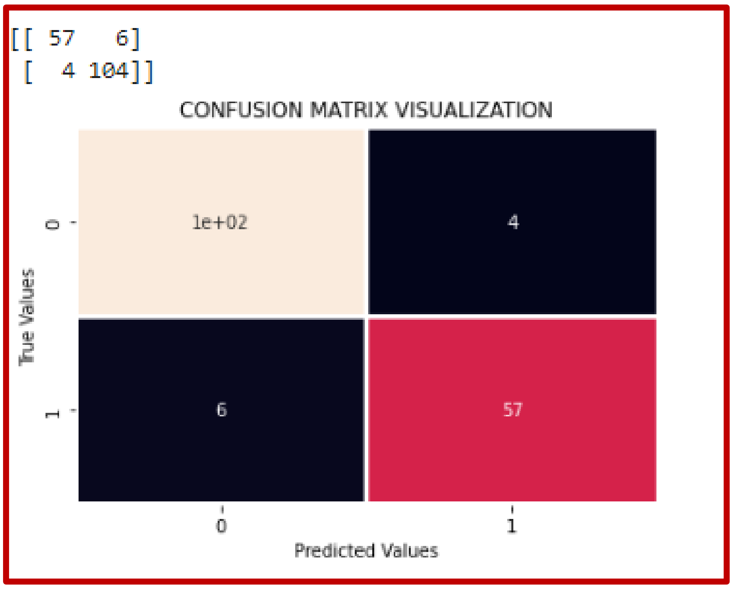
```
Data is splitinto following number of folds:
5
```

5. Building a Logistic Regression, Naive Bayes, Decision Tree (CART), K-NN, SVM, and RandomForestClassifier models. Also, print their accuracies



```
Logistic Regression -> ACC: %63.16
Naive Bayes -> ACC: %63.16
Decision Tree (CART) -> ACC: %91.81
K-NN -> ACC: %76.61
SVM -> ACC: %63.16
RandomForestClassifier -> ACC: %94.15
```

6. Calculate and plot the confusion matrix



```
[[ 57   6]
 [  4 104]]
```

CONFUSION MATRIX VISUALIZATION
1e+02
4
6
57
True Values
0
1
0
1
Predicted Values