Assignment-6

Problem Statement 1: 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()
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

Example:

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

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()

	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

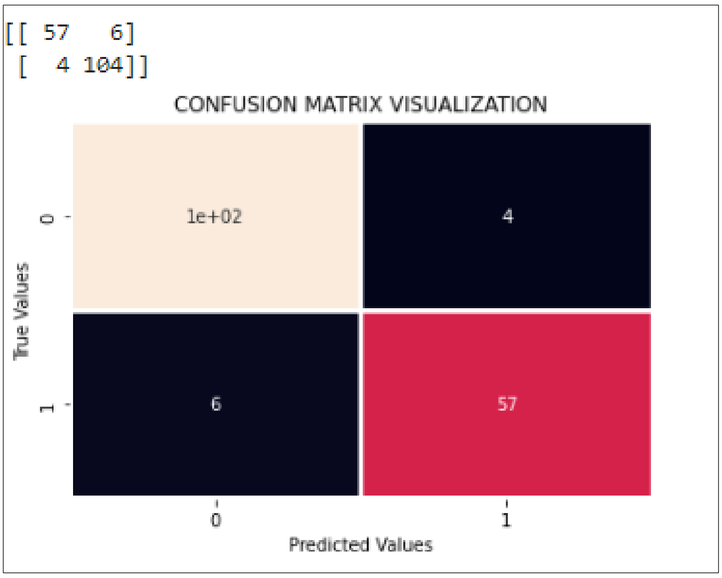
```
y.head()

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

4. Split the data into five folds using KFold() function
5. Build a decision tree classifier model and print model accuracies for all the data folds

```
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
```

7. Calculate and plot the confusion matrix.



Problem Statement 2: 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. Considering the 'diagnosis' column as the target, encode the 'diagnosis' column using the LabelEncoder()
3. 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 the ROC_AUC score based on the False Positive Rate (FPR) and True Positive Rate (TPR)
7. Plot the ROC Curve using the Matplotlib library
8. Calculate the F1 Score
9. Calculate and Print the Precision, Recall, and F1 score using the classification_report() function

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.

Example:

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

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. Considering the 'diagnosis' column as the target, encode the 'diagnosis' column using the LabelEncoder()

	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. 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
```

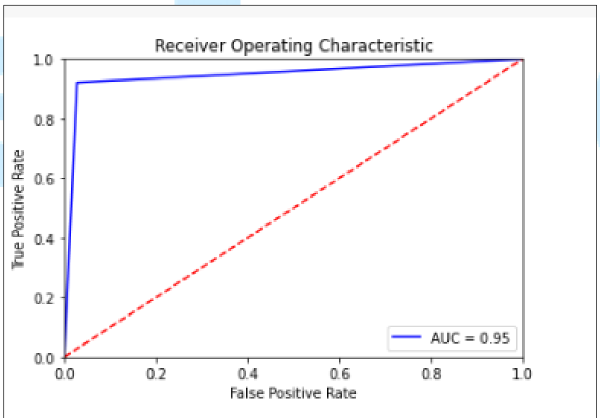
4. Split the dataset into the training set and test set in a 70:30 ratio
5. Building a Logistic Regression, Naive Bayes, Decision Tree (CART), K-NN, SVM, and RandomForestClassifier models; Also, print their accuracies

```
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 the ROC_AUC score based on the False Positive Rate (FPR) and True Positive Rate (TPR)

```
roc_auc score is:
0.943121693121693
```

7. Plot the ROC Curve using the Matplotlib library



8. Calculate the F1 Score

```
F1 score is:
0.9344262295081968
```

9. Calculate and print the Precision, Recall, and F1 score using the classification_report() function

```
              precision    recall  f1-score   support

           0       0.95      0.97      0.96       108
           1       0.95      0.92      0.94        63

    accuracy                           0.95       171
   macro avg       0.95      0.95      0.95       171
weighted avg       0.95      0.95      0.95       171
```

Problem Statement 3: Load the ‘voice.csv’ dataset into a DataFrame and perform the following tasks:

1. Considering the ‘label’ column as the target variable, rename the column as ‘Gender_Identified’
2. Using the preprocessing() function, label the target column
3. Separate the target variable and the feature vectors
4. Build a RandomForestClassifier model and find the best parameters using a Grid search
5. Print the best parameters and the best estimator

Example:

Dataset:

	meanfreq	sd	median	Q25	Q75	IQR	skew	kurt	sp.ent	sfm	...
0	0.059781	0.064241	0.032027	0.015071	0.090193	0.075122	12.863462	274.402906	0.893369	0.491918	...
1	0.066009	0.067310	0.040229	0.019414	0.092666	0.073252	22.423285	634.613855	0.892193	0.513724	...
2	0.077316	0.083829	0.036718	0.008701	0.131908	0.123207	30.757155	1024.927705	0.846389	0.478905	...
3	0.151228	0.072111	0.158011	0.096582	0.207955	0.111374	1.232831	4.177296	0.963322	0.727232	...
4	0.135120	0.079146	0.124656	0.078720	0.206045	0.127325	1.101174	4.333713	0.971955	0.783568	...

Output:

1. Considering the ‘label’ column as the target variable, rename the column as ‘Gender_Identified’

...	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	modindx	Gender_Identified
...	0.059781	0.084279	0.015702	0.275862	0.007812	0.007812	0.007812	0.000000	0.000000	male
...	0.066009	0.107937	0.015826	0.250000	0.009014	0.007812	0.054688	0.046875	0.052632	male
...	0.077316	0.098706	0.015656	0.271186	0.007990	0.007812	0.015625	0.007812	0.046512	male
...	0.151228	0.088965	0.017798	0.250000	0.201497	0.007812	0.562500	0.554688	0.247119	male
...	0.135120	0.106398	0.016931	0.266667	0.712812	0.007812	5.484375	5.476562	0.208274	male

2. Using the preprocessing() function, label the target column

...	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	modindx	Gender_Identified
...	0.186071	0.166718	0.016377	0.238806	0.549753	0.171875	6.132812	5.960938	0.072885	0
...	0.216473	0.173048	0.048387	0.275862	1.330078	0.023438	5.718750	5.695312	0.169113	0
...	0.185102	0.153339	0.048731	0.277457	1.317522	0.023438	7.078125	7.054688	0.156884	0
...	0.162429	0.086479	0.017877	0.168421	0.586458	0.093750	3.718750	3.625000	0.209821	1
...	0.165676	0.086315	0.016427	0.228571	0.676994	0.007812	3.312500	3.304688	0.272743	1
...	0.194083	0.153974	0.047904	0.279070	1.909624	0.023438	8.906250	8.882812	0.081574	0
...	0.226478	0.199270	0.049383	0.277457	0.508594	0.023438	2.789062	2.765625	0.086035	0
...	0.085832	0.097472	0.016343	0.231884	0.132308	0.007812	1.492188	1.484375	0.111579	1
...	0.122706	0.187219	0.032787	0.238806	0.610609	0.015625	3.773438	3.757812	0.190460	0
...	0.234016	0.197109	0.048000	0.279070	0.906250	0.023438	5.179688	5.156250	0.114593	0

3. Separate the target variable and the feature vectors

mode	centroid	meanfun	minfun	maxfun	meandom	mindom	maxdom	dfrange	modindx
0.000000	0.059781	0.084279	0.015702	0.275862	0.007812	0.007812	0.007812	0.000000	0.000000
0.000000	0.066009	0.107937	0.015826	0.250000	0.009014	0.007812	0.054688	0.046875	0.052632
0.000000	0.077316	0.098706	0.015656	0.271186	0.007990	0.007812	0.015625	0.007812	0.046512
0.083878	0.151228	0.088965	0.017798	0.250000	0.201497	0.007812	0.562500	0.554688	0.247119
0.104261	0.135120	0.106398	0.016931	0.266667	0.712812	0.007812	5.484375	5.476562	0.208274

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

4. Build a RandomForestClassifier model and find the best parameters using a Grid search

	mean_fit_time	std_fit_time	mean_score_time	std_score_time	param_criterion	param_n_estimators	params	split0_test_score	split1_test_score	split2_test_score	mean_test_score	std_test_score	rank_test_score
4	0.653557	0.042416	0.024817	0.000351	entropy	100	{'criterion': 'entropy', 'n_estimators': 100}	0.946970	0.982955	0.971591	0.967172	0.015019	1
5	0.974606	0.062932	0.041474	0.002761	entropy	150	{'criterion': 'entropy', 'n_estimators': 150}	0.944129	0.982008	0.970644	0.965593	0.015871	2
7	2.536522	0.395648	0.097291	0.040550	entropy	300	{'criterion': 'entropy', 'n_estimators': 300}	0.945076	0.982008	0.969697	0.965593	0.015354	2
2	0.980851	0.039669	0.046929	0.001189	gini	200	{'criterion': 'gini', 'n_estimators': 200}	0.939394	0.981061	0.974432	0.964962	0.018281	4
6	1.317470	0.075342	0.047918	0.003116	entropy	200	{'criterion': 'entropy', 'n_estimators': 200}	0.945076	0.982008	0.967803	0.964962	0.015211	4
0	0.654851	0.203673	0.024716	0.000250	gini	100	{'criterion': 'gini', 'n_estimators': 100}	0.939394	0.981061	0.973485	0.964646	0.018122	6
3	1.476880	0.060719	0.079388	0.014409	gini	300	{'criterion': 'gini', 'n_estimators': 300}	0.938447	0.981061	0.972538	0.964015	0.018411	7
1	0.737936	0.028873	0.038054	0.001538	gini	150	{'criterion': 'gini', 'n_estimators': 150}	0.940341	0.981061	0.969697	0.963699	0.017156	8

5. Print the best parameters and the best estimator

```
The best parameters are:
{'criterion': 'entropy', 'n_estimators': 100}
```

```
The best estimator is:
RandomForestClassifier(criterion='entropy')
```

Problem Statement 4: The 'seeds.csv' dataset contains the data about the wheat seeds, the 'Type' column consisit of three unique values, 1, 2, 3, which are classified based on the charecterstics of seeds entailing in other columns.

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

1. Considering the 'Type' column as target, analyze the target column by printing the unique values
2. Separate the feature vectors and the target variable
3. Split the dataset into train and test sets in a 70:30 ratio
4. Build a Decision Tree Classifier and a GaussianNB model and print their accuracy scores
5. For the Decision Tree Classifier and a GaussianNB models boost the accuracy using ADA Boost Classifier and compare the accuracy scores with original models using a bar plot

Example:

Dataset:

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove	Type
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220	1
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956	1
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825	1
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805	1
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175	1

Output:

1. Considering the 'Type' column as target, analyze the target column by printing the unique values

	Counts	Percentage
2	68	0.341709
1	66	0.331658
3	65	0.326633

2. Separate the feature vectors and the target variable

	Area	Perimeter	Compactness	Kernel.Length	Kernel.Width	Asymmetry.Coeff	Kernel.Groove
0	15.26	14.84	0.8710	5.763	3.312	2.221	5.220
1	14.88	14.57	0.8811	5.554	3.333	1.018	4.956
2	14.29	14.09	0.9050	5.291	3.337	2.699	4.825
3	13.84	13.94	0.8955	5.324	3.379	2.259	4.805
4	16.14	14.99	0.9034	5.658	3.562	1.355	5.175

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

3. Split the dataset into train and test sets in a 70:30 ratio
4. Build a Decision Tree Classifier and a GaussianNB model and print their accuracy scores

```
Accuracy score of the Decision tree model is:
0.583
```

```
Accuracy score of the GaussianNB model is:
0.933
```

5. For the Decision Tree Classifier and a GaussianNB models boost the accuracy using ADA Boost Classifier and compare the accuracy scores with original models using a bar plot