University of Essex

School of Computer Science & Electronic Engineering

CE802 Machine Learning & Data Mining

Report on Assignment: Design and Application of a Machine Learning System for a Practical Problem

Submitted By

Omkar R Kharkar

2205233

Submitted To:

Dr. Vito De Feo

Abstract

The assignment asked to perform a comparative study of different machine learning algorithms and test the algorithm with the test data provided. Learning outcomes of this assignment are: a] to learn to identify machine learning techniques appropriate for particular practical problems; and b] to undertake a comparative evaluation of several machine learning procedures when applied to specific problems. After performing the required tasks on the given datasets, herein lies my final report.

Comparative Study

A. Explorative Data Analysis:

	F1	F2	F3	F4	F5	F6	F7	F8	F9	\
0	1.6430	0	-4894.24	-13.0281	-4.793400	0	5.1270	-17.1100	-63.340	
1	0.5310	0	-5085.44	-16.2210	-3.991776	0	4.6256	-4.5800	-10.314	
2	0.2640	0	-7021.44	-11.7591	-6.161700	0	4.3628	-14.7118	-6.806	
3	0.3196	1	-4648.76	-11.8110	-4.217700	0	8.9380	-7.5360	-4.670	
4	4.0800	0	-4877.20	-11.2635	-8.061000	1	6.2800	-14.5805	-45.920	

	F10	...	F13	F14	F15	F16	F17	F18	\
0	3.61690	...	5.783440	-11315.46	22912.53	-0.4	103811.34	5.4380	
1	3.64880	...	8.180000	-12852.96	25696.44	-0.4	103884.02	5.0960	
2	3.62830	...	5.760312	-11012.16	20232.84	-1.4	103987.08	2.3652	
3	3.01503	...	6.437100	-10297.86	23592.84	-1.4	103842.08	4.4080	
4	3.60030	...	6.393200	-11527.38	24778.74	-1.4	103842.48	3.1334	

	F19	F20	F21	Class
0	1747.920	-4879.68	-41.58	False
1	1496.080	-4186.38	-45.96	True
2	1523.412	-4067.28	NaN	False
3	1506.810	1352.52	NaN	True
4	1581.790	-5095.88	-45.93	True

Fig 1. Dataset

- In this dataset, we have 22 Features and 1000 Instances.
- In Feature F21, we have 500 null values, as shown in the graph.
- To fill these null values, I counted how many True & False values are present in the F21 Column. There were 187 True values & 179 False values. Using a box plot, I found the 50-percentile value of True & False in the F21 column. After that, I filled the values in place of null values.

```
F1         0
F2         0
F3         0
F4         0
F5         0
F6         0
F7         0
F8         0
F9         0
F10        0
F11        0
F12        0
F13        0
F14        0
F15        0
F16        0
F17        0
F18        0
F19        0
F20        0
F21      500
Class      0
dtype: int64
```

Fig 2. Null Values in F21

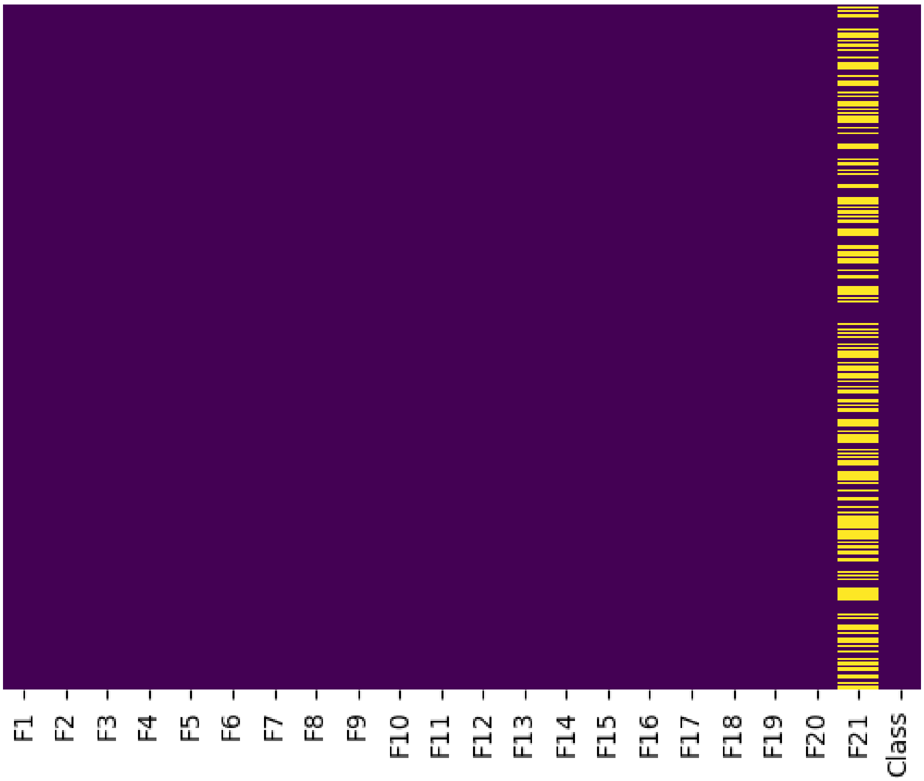


Fig 3. Showing the visualization method for null values

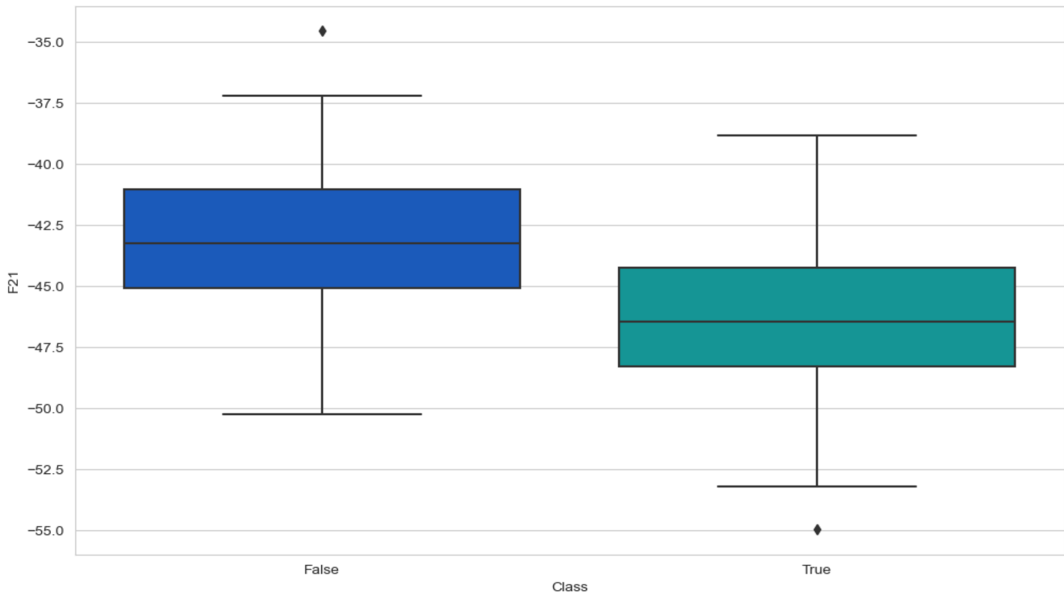


Fig 4. Showing the percentile values of Class w.r.t F21

B. Feature Engineering:

- After cleaning the data, I found the correlation between Dependent Features vs independent features as well as independent vs independent features.
- Here, Basically I have used the Person correlation in which -1,0,1 represent s negative correlation, zero correlation & positive correlation. When two features are highly correlated, we can drop off that feature.
- Furthermore, I went to investigate the skewness of the features like which features have positive skewness & negative skewness.
- Skewness is a crucial statistical method that aids in identifying the frequency distribution's asymmetrical behaviour, or more specifically, the absence of symmetry in the tails to the left and right of the frequency curve.
- F1, F6, F7, F10, F13, F15, F17, F18, F21 are Positive skew whereas, F2, F3, F4, F5, F9, F12, F19 are negative skew. So, we will apply different types of statistical methods to remove right & left skew.
- If the feature is skewed towards the positive side, then we apply Logarithm, Square-root, Cube-root, and Reciprocal to get them into symmetry whereas if the feature is skewed towards the negative, we apply Square and cube to get them into symmetry.
- After all, this we try different types of scaling like Standardization, Normalization and Robust Scaler.
- Standardization uses Z-score transformation to bring the data within a scale and Normalization use Min-Max to bring the data within the range of 0 and 1. In Robust, it removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

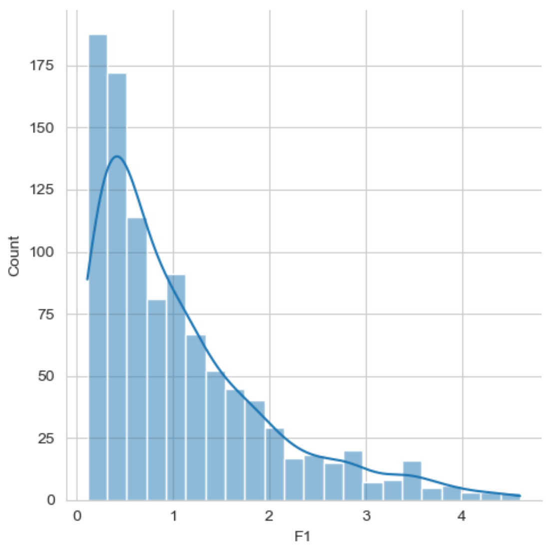


Fig 5. Right / Positive Skew

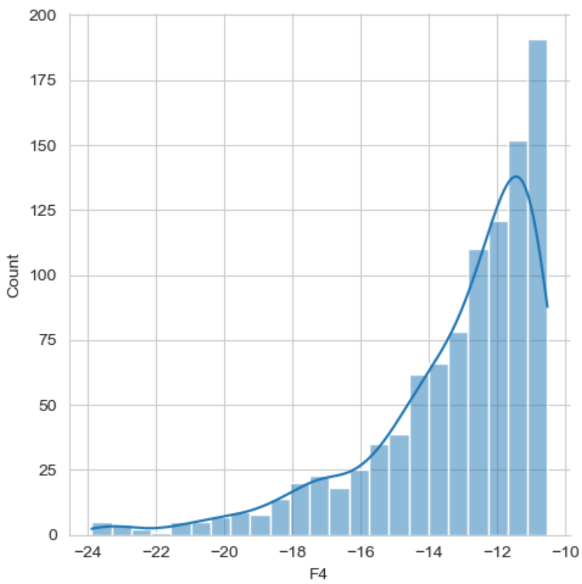


Fig 6. Left / Negative Skew

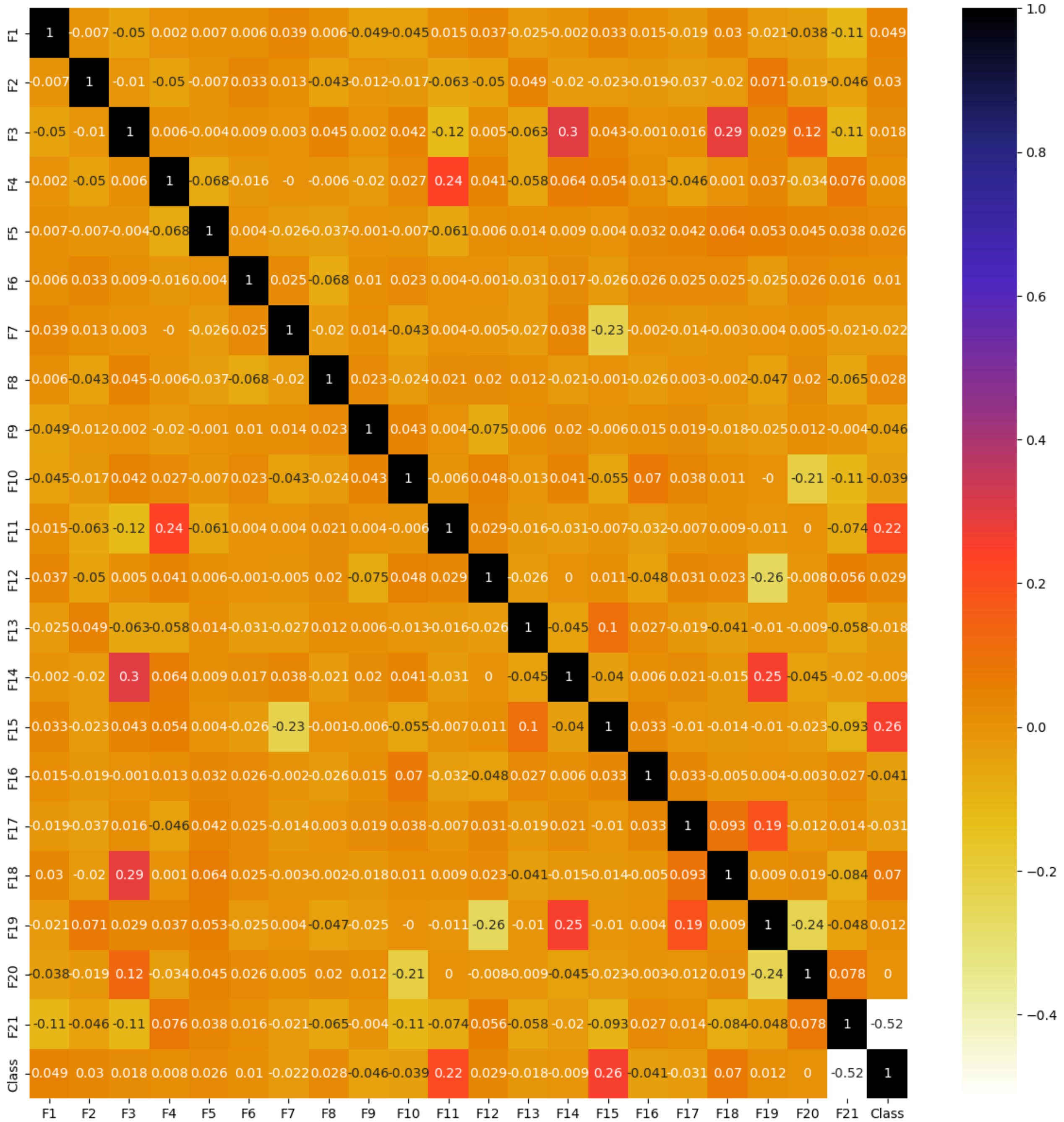


Fig 7. Correlation Heatmap

C. Splitting the Datasets:

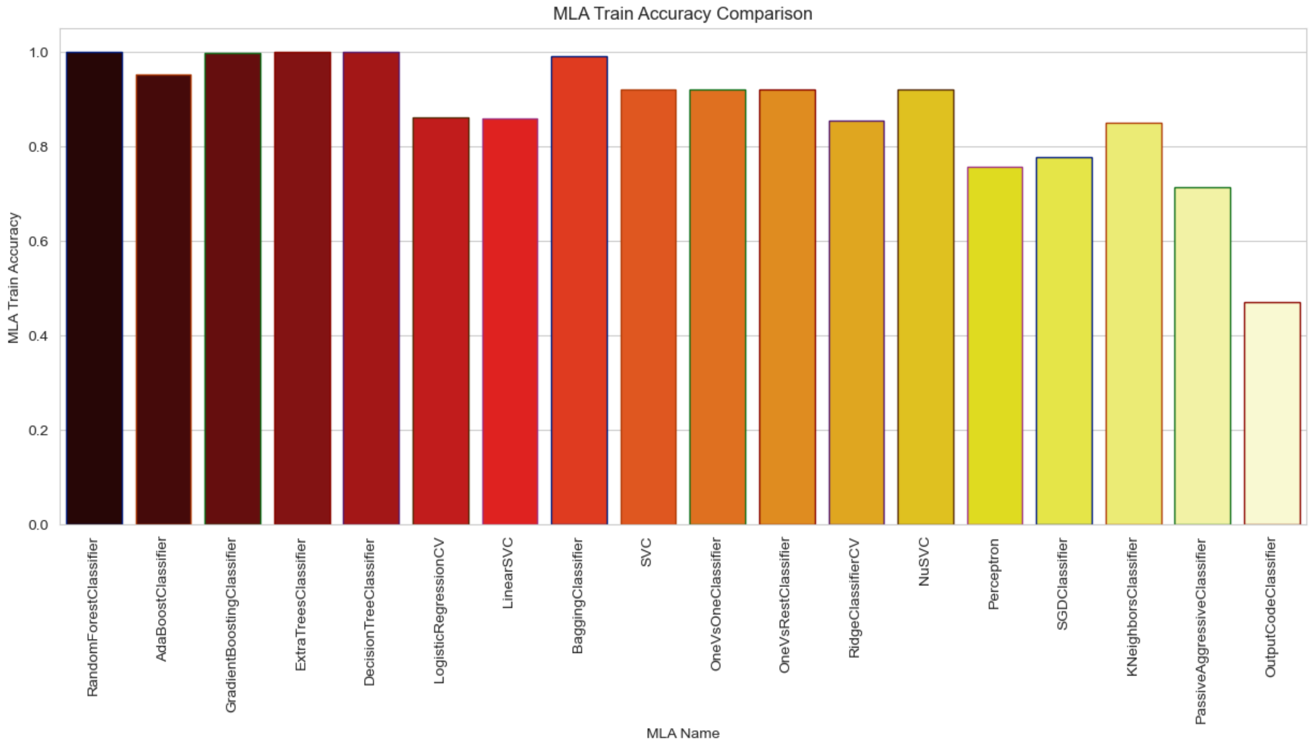
- A model validation technique called "train test split" enables you to mimic how a model would perform on brand-new or untested data It is very common to split the train test datasets in:
    I. 70:30 ratio
    II. 80:20 ratio
    III. 75:25 ratio
- So, we are going to split our data as shown in the table.

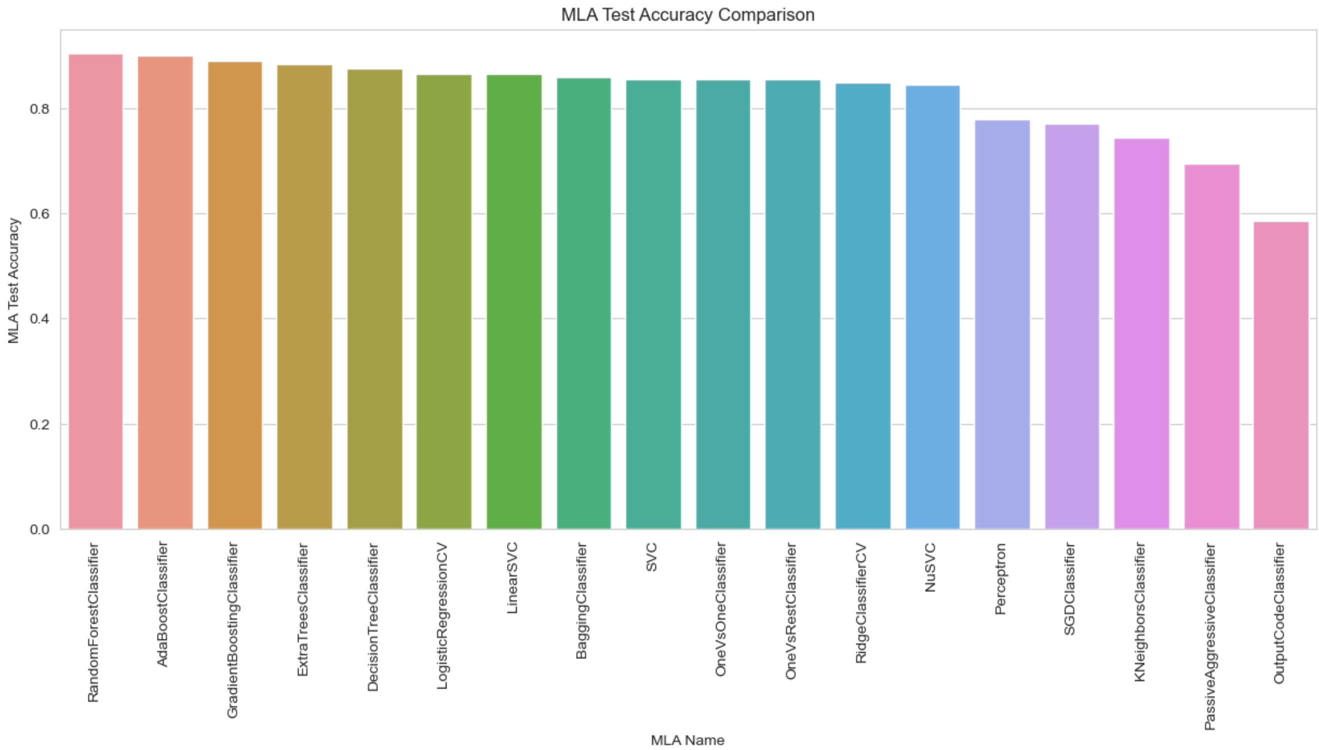
Train	**80%**
Test	**20%**

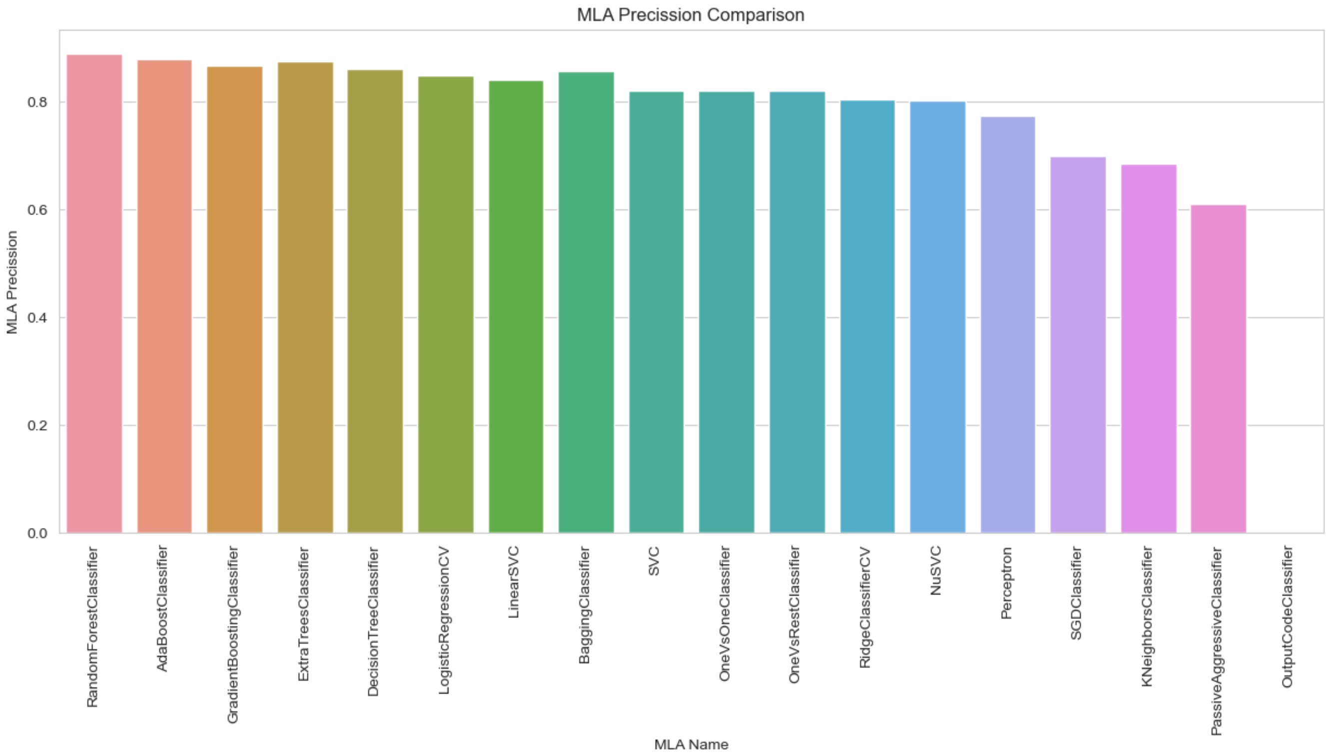
D. Machine Learning Model:

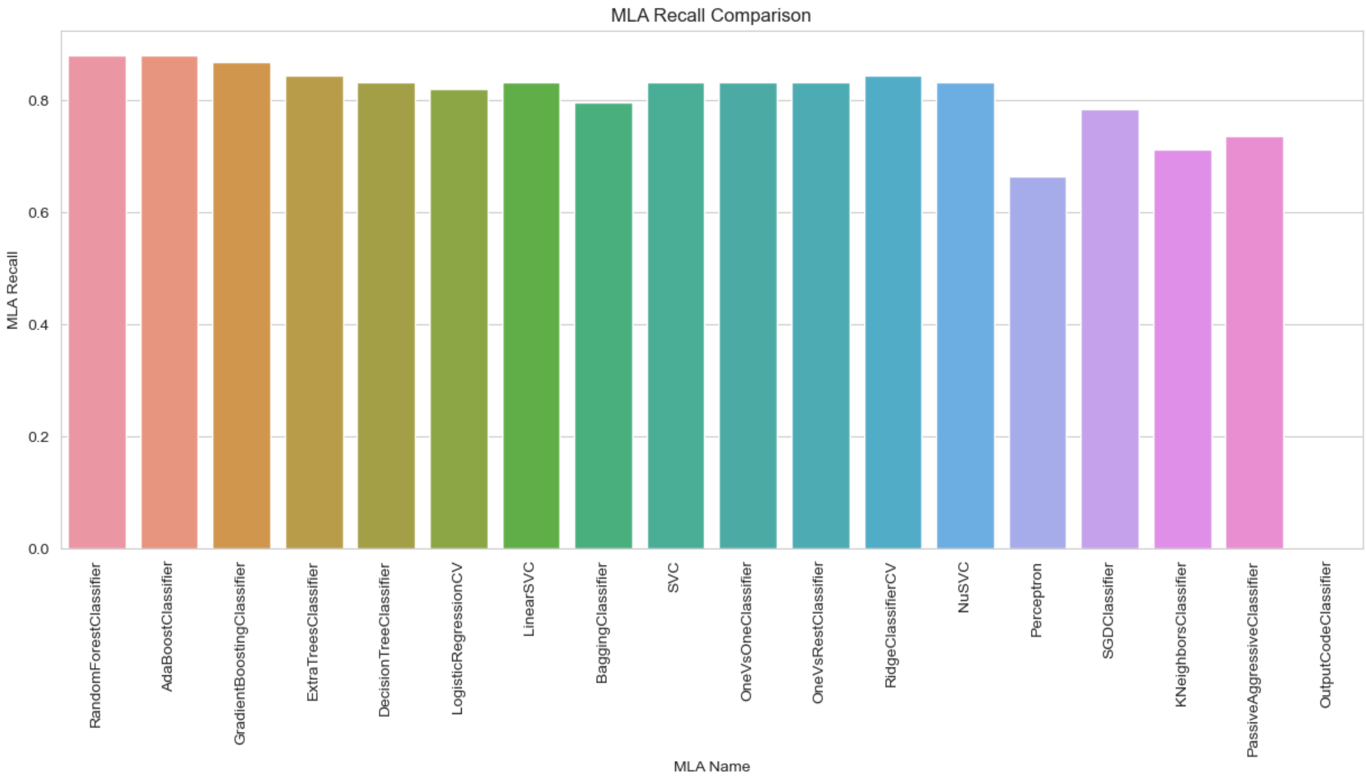
- For predicting we use the Machine Learning technique because machine learning is the best technique to provide a proper decision-making strategy. There are several machine learning algorithms like Decision Trees, Random Forest, Naïve Bayes, K-Nearest Neighbours (kNN), Support Vector Machines, Linear Regression and Logistic Regression.
- In Sklearn there are a total of 34 classification Libraries which you can use for classification tasks.
- Below table shows the accuracy of the Models that we used.

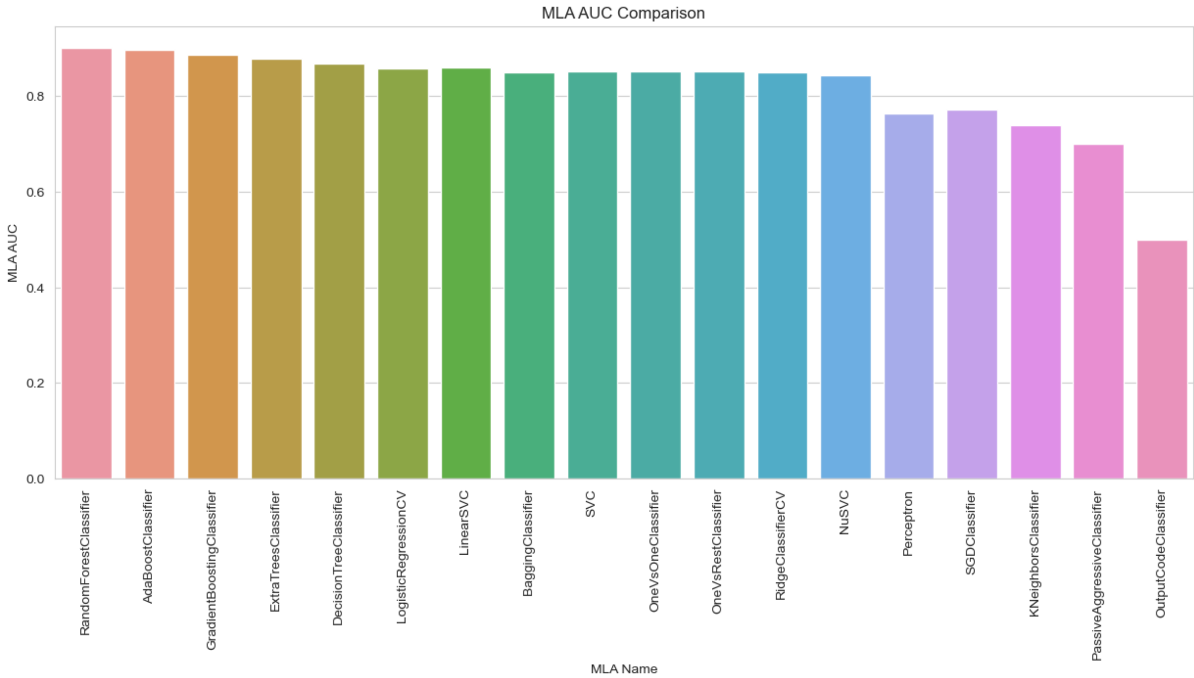
	MLA Name	MLA Train Accuracy	MLA Test Accuracy	MLA Precission	MLA Recall	MLA AUC
0	RandomForestClassifier	1.0000	0.905	0.890244	0.879518	0.901297
1	AdaBoostClassifier	0.9512	0.900	0.879518	0.879518	0.897024
2	GradientBoostingClassifier	0.9975	0.890	0.867470	0.867470	0.886726
3	ExtraTreesClassifier	1.0000	0.885	0.875000	0.843373	0.878952
4	DecisionTreeClassifier	1.0000	0.875	0.862500	0.831325	0.868654
5	LogisticRegressionCV	0.8600	0.865	0.850000	0.819277	0.858357
6	LinearSVC	0.8588	0.865	0.841463	0.831325	0.860107
7	BaggingClassifier	0.9912	0.860	0.857143	0.795181	0.850582
8	SVC	0.9200	0.855	0.821429	0.831325	0.851560
9	OneVsOneClassifier	0.9200	0.855	0.821429	0.831325	0.851560
10	OneVsRestClassifier	0.9200	0.855	0.821429	0.831325	0.851560
11	RidgeClassifierCV	0.8538	0.850	0.804598	0.843373	0.849037
12	NuSVC	0.9200	0.845	0.802326	0.831325	0.843013
13	Perceptron	0.7575	0.780	0.774648	0.662651	0.762949
14	SGDClassifier	0.7775	0.770	0.698925	0.783133	0.771908
15	KNeighborsClassifier	0.8500	0.745	0.686047	0.710843	0.740037
16	PassiveAggressiveClassifier	0.7138	0.695	0.610000	0.734940	0.700803
17	OutputCodeClassifier	0.4712	0.585	0.000000	0.000000	0.500000

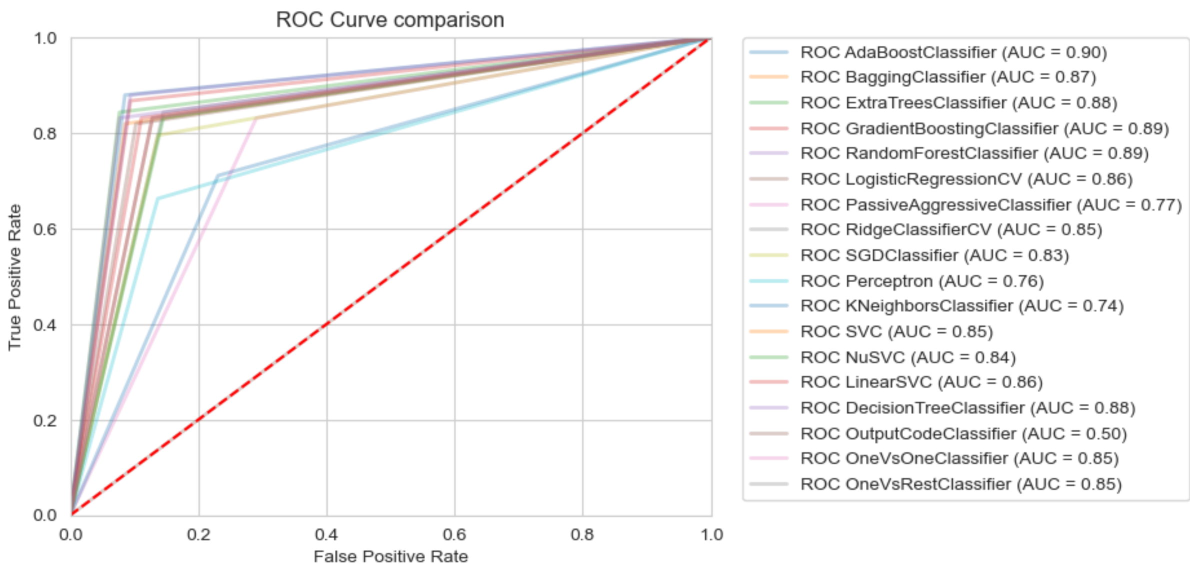

MLA Train Accuracy Comparison
MLA Train Accuracy
MLA Name
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
ExtraTreesClassifier
DecisionTreeClassifier
LogisticRegressionCV
LinearSVC
BaggingClassifier
SVC
OneVsOneClassifier
OneVsRestClassifier
RidgeClassifierCV
NuSVC
Perceptron
SGDClassifier
KNeighborsClassifier
PassiveAggressiveClassifier
OutputCodeClassifier


MLA Test Accuracy Comparison
MLA Test Accuracy
MLA Name
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
ExtraTreesClassifier
DecisionTreeClassifier
LogisticRegressionCV
LinearSVC
BaggingClassifier
SVC
OneVsOneClassifier
OneVsRestClassifier
RidgeClassifierCV
NuSVC
Perceptron
SGDClassifier
KNeighborsClassifier
PassiveAggressiveClassifier
OutputCodeClassifier


MLA Precission Comparison
MLA Precission
0.0
0.2
0.4
0.6
0.8
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
ExtraTreesClassifier
DecisionTreeClassifier
LogisticRegressionCV
LinearSVC
BaggingClassifier
SVC
OneVsOneClassifier
OneVsRestClassifier
RidgeClassifierCV
NuSVC
Perceptron
SGDClassifier
KNeighborsClassifier
PassiveAggressiveClassifier
OutputCodeClassifier
MLA Name


MLA Recall Comparison
MLA Recall
0.0
0.2
0.4
0.6
0.8
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
ExtraTreesClassifier
DecisionTreeClassifier
LogisticRegressionCV
LinearSVC
BaggingClassifier
SVC
OneVsOneClassifier
OneVsRestClassifier
RidgeClassifierCV
NuSVC
Perceptron
SGDClassifier
KNeighborsClassifier
PassiveAggressiveClassifier
OutputCodeClassifier
MLA Name


MLA AUC Comparison
MLA AUC
0.0
0.2
0.4
0.6
0.8
MLA Name
RandomForestClassifier
AdaBoostClassifier
GradientBoostingClassifier
ExtraTreesClassifier
DecisionTreeClassifier
LogisticRegressionCV
LinearSVC
BaggingClassifier
SVC
OneVsOneClassifier
OneVsRestClassifier
RidgeClassifierCV
NuSVC
Perceptron
SGDClassifier
KNeighborsClassifier
PassiveAggressiveClassifier
OutputCodeClassifier


ROC Curve comparison
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC AdaBoostClassifier (AUC = 0.90)
ROC BaggingClassifier (AUC = 0.87)
ROC ExtraTreesClassifier (AUC = 0.88)
ROC GradientBoostingClassifier (AUC = 0.89)
ROC RandomForestClassifier (AUC = 0.89)
ROC LogisticRegressionCV (AUC = 0.86)
ROC PassiveAggressiveClassifier (AUC = 0.77)
ROC RidgeClassifierCV (AUC = 0.85)
ROC SGDClassifier (AUC = 0.83)
ROC Perceptron (AUC = 0.76)
ROC KNeighborsClassifier (AUC = 0.74)
ROC SVC (AUC = 0.85)
ROC NuSVC (AUC = 0.84)
ROC LinearSVC (AUC = 0.86)
ROC DecisionTreeClassifier (AUC = 0.88)
ROC OutputCodeClassifier (AUC = 0.50)
ROC OneVsOneClassifier (AUC = 0.85)
ROC OneVsRestClassifier (AUC = 0.85)

E. Test Data:

F3	F4	F5	F6	F7	F8	F9	F10	...	F13	F14	F15	F16	F17	F18	F19	F20	F21	Class
22.64	-10.7841	-6.70380	1	5.4270	-9.679	-18.5660	5.86600	...	5.761109	-11347.230	22929.33	-0.4	103741.46	4.0760	1466.0700	-4266.40	NaN	False
93.64	-12.6885	-7.98300	0	4.5008	-11.561	-25.9200	3.18100	...	6.797000	-11289.360	25723.74	-1.4	103858.01	2.6546	1607.6600	-4802.48	-44.91	True
16.46	-11.9391	-5.11320	1	5.3808	-13.281	-20.2400	3.04817	...	6.922000	-11133.060	23138.58	-0.4	105361.06	5.6300	1543.2200	-4220.46	-45.66	True
05.44	-10.9737	-6.95640	1	6.5020	-12.101	-13.6260	3.28770	...	9.260000	-11773.530	23100.78	-0.4	103835.75	2.3680	1532.0397	-4612.88	-43.26	False
63.44	-15.9780	-10.24200	1	4.2970	-11.596	-14.6240	3.55760	...	5.997600	-11937.060	27299.64	-1.4	103877.64	2.6908	1084.3200	-4557.08	-44.82	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
38.64	-15.7710	-4.16595	0	7.3160	-19.894	-73.3600	5.45900	...	5.778300	-10418.160	23734.14	-1.4	103839.18	2.9136	1518.3900	-4146.08	NaN	False
21.32	-13.9560	-8.34000	0	4.7972	-7.204	-79.4200	3.37930	...	7.407000	-15800.760	25006.44	-0.4	103840.60	10.8800	1520.1000	-4665.08	-45.93	False
77.82	-11.5935	-4.02435	1	8.0940	-12.809	-1.9182	3.06420	...	5.960300	-4070.760	23033.76	-0.4	103993.70	9.6960	1493.1800	552.52	-43.92	False
93.36	-20.8680	-4.89450	1	5.1574	-18.655	-6.2880	3.91770	...	6.199500	-11618.257	22637.34	-0.4	103845.43	3.4610	1483.6500	-4203.62	-44.46	False
90.64	-11.3592	-4.27329	1	5.1350	-21.093	-15.7460	3.05270	...	5.768134	-11306.760	24056.04	-1.4	103801.56	3.4336	1617.3200	-6953.48	NaN	True

Additional Comparative Study

A. Explorative Data Analysis:

```
        F1       F2     F3     F4        F5    F6       F7      F8         F9  \
0 -190.11   193.53   2.32 -21.84  11289.72    UK  5730.72  638.55  Very high
1 -257.94  1934.85  24.36 -22.06   2712.12    UK  3509.94  389.53       High
2 -426.06  1071.87   0.10 -21.80   7469.01   USA  4633.20  -28.63     Medium
3 -204.48  1533.96   7.42 -17.94   4261.77  Rest  3516.06  335.36  Very high
4 -232.08  1334.88  29.48 -19.88   2941.02  Rest  3592.04  -46.68   Very low

      F10  ...     F28       F29     F30  F31     F32    F33      F34  F35  \
0  467.97  ...  174.81  23994.54  261.69    5 -177.15  17.60 -1437.20    4
1  393.42  ...  265.44  11554.06  205.14    4 -377.55  16.32 -1840.92    7
2  226.62  ...  542.94  28254.56  363.93    5 -293.67  17.64 -1207.92    5
3  912.63  ...  527.10  21449.30   23.31    4 -295.80  20.44  -139.58    6
4  601.41  ...  390.48  18060.98  248.79    3 -222.12  19.39 -2130.02    4

      F36   Target
0  178.20  1306.29
1   83.53  -118.07
2  218.54  -708.14
3  154.74  2918.75
4  178.77  1113.09

[5 rows x 37 columns]
```

Fig no.1 Dataset

- In this dataset, we have 37 Features and 1500 Instances.
- In Feature F6 & F9, we have Categorical values.

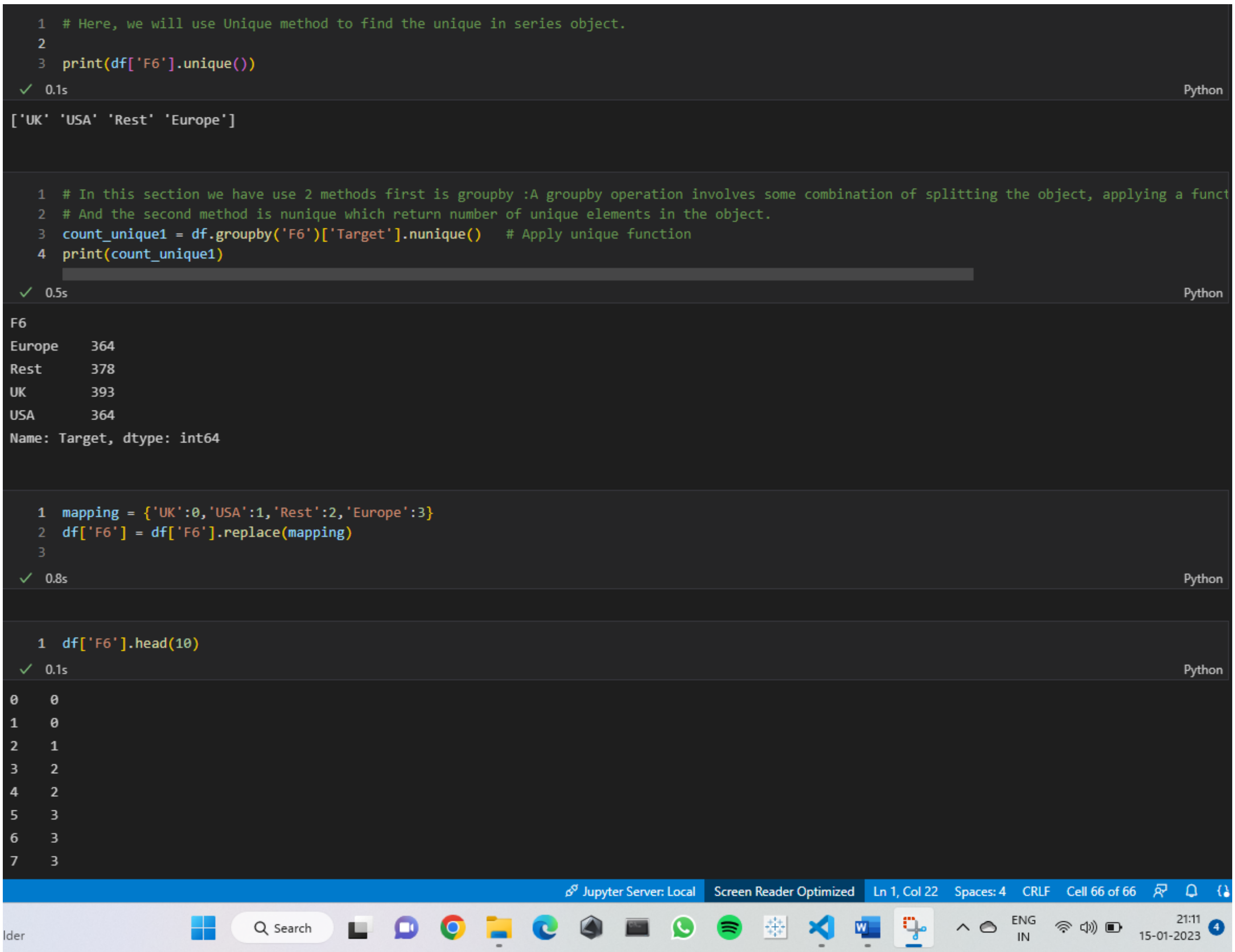


Fig no.2. Converting Categorical Values to numerical

B. Feature Engineering:

- After converting the data from categorical to numerical, I found the correlation between Dependent Features vs independent features as well as independent vs independent features.
- Here, Basically I have used the Person correlation in which -1,0,1 represent s negative correlation, zero correlation & positive correlation. When two features are highly correlated, we can drop off that feature.
- Furthermore, I went to investigate the skewness of the features like which features have positive skewness & negative skewness.
- Skewness is a crucial statistical method that aids in identifying the frequency distribution's asymmetrical behaviour, or more specifically, the absence of symmetry in the tails to the left and right of the frequency curve.
- F3 & F23 are Positive skew whereas, F4 are negative skew. So, we will apply different types of statistical methods to remove right & left skew.
- If the feature is skewed towards the positive side, then we apply Logarithm, Square-root, Cube-root, and Reciprocal to get them into symmetry whereas if the feature is skewed towards the negative, we apply Square and cube to get them into symmetry.
- After this, I have created a baseline model which will give me the importance of the feature means it will tell us that this feature has more importance than others.
- 'F1','F5','F7','F10','F11','F13','F14','F15','F16','F17','F18','F19','F20','F21','F22',' F23','F24','F26','F28','F29','F30','F32','F33','F34' these features were not much important w.r.t model.
- After all, this we try different types of scaling like Standardization, Normalization and Robust Scaler.
- Standardization uses Z-score transformation to bring the data within a scale and Normalization use Min-Max to bring the data within the range of 0 and 1. In Robust, it removes the median and scales the data according to the quantile range (defaults to IQR: Interquartile Range). The IQR is the range between the 1st quartile (25th quantile) and the 3rd quartile (75th quantile).

Fig no.3. Important Feature

	Features Name	Importance Features
34	F35	0.177070
8	F9	0.163126
3	F4	0.144781
5	F6	0.120610
2	F3	0.113393
24	F25	0.053162
1	F2	0.050727
30	F31	0.029184
15	F16	0.013321
33	F34	0.010234
10	F11	0.008336
18	F19	0.008060
20	F21	0.007540
23	F24	0.007164
13	F14	0.006632
26	F27	0.006494
27	F28	0.006436
4	F5	0.005763
12	F13	0.005528
35	F36	0.005460
7	F8	0.005057
31	F32	0.004750
16	F17	0.004595
32	F33	0.004426
6	F7	0.004230
17	F18	0.004155
14	F15	0.004022
21	F22	0.003752
28	F29	0.003746
19	F20	0.003705
22	F23	0.003566
11	F12	0.002558
9	F10	0.002511
25	F26	0.002134
29	F30	0.002044
0	F1	0.001728

Fig no.3. Important Feature

C. Splitting the Datasets:

- A model validation technique called "train test split" enables you to mimic how a model would perform on brand-new or untested data It is very common to split the train test datasets in:

    I. 70:30 ratio
    II. 80:20 ratio
    III. 75:25 ratio

- So, we are going to split our data as shown in the table.

Train	**80%**
Test	**20%**

D. Machine Learning Model:

- For predicting we use the Machine Learning technique because machine learning is the best technique to provide a proper decision-making strategy. There are several machine learning algorithms like Decision Trees, Random Forest, Naïve Bayes, K-Nearest Neighbours (kNN), Support Vector Machines, Linear Regression and Logistic Regression.
- In Sklearn there are a total of 34 classification Libraries which you can use for classification tasks.
- Below table shows the accuracy of the Models that we used.

I. Baseline Model:

- ➢ I have created a baseline model using XGBRegressor.

```
model_lr = XGBRegressor()
model_lr.fit(x_train,y_train)
y_pred_lr = model_lr.predict(x_test)
model_lr.score(x_test,y_test)


0.75648374723381
```

Fig no.4. Baseline Model

II. Comparing Model:

	MLA Name	MLA Train MSE	MLA Test MSE	MLA MAE	MLA RMSE	MLA R2square
0	GradientBoostingRegressor	2.482965e+02	402.993010	402.993010	519.931957	0.809240
1	ExtraTreesRegressor	1.278522e-12	441.911579	441.911579	571.855646	0.769236
2	RandomForestRegressor	1.673100e+02	458.124968	458.124968	595.482776	0.749774
3	BaggingRegressor	2.025056e+02	488.568670	488.568670	633.299991	0.716982
4	LinearRegression	5.152491e+02	535.228853	535.228853	680.855108	0.672882
5	Lasso	5.151773e+02	535.376582	535.376582	680.959114	0.672783
6	BayesianRidge	5.150884e+02	535.330126	535.330126	680.962327	0.672779
7	HuberRegressor	5.058973e+02	527.754236	527.754236	695.740270	0.658423
8	PassiveAggressiveRegressor	5.053822e+02	531.422493	531.422493	699.951311	0.654276
9	AdaBoostRegressor	6.196575e+02	664.612873	664.612873	761.546365	0.590752
10	ElasticNet	5.832108e+02	587.135319	587.135319	763.015289	0.589171
11	DecisionTreeRegressor	0.000000e+00	706.632967	706.632967	975.838435	0.328029

E. Test Data:

F2	F3	F4	F5	F6	F7	F8	F9	F10	...	F28	F29	F30	F31	F32	F33	F34	F35	F36	Target
995.49	0.60	-21.64	-13656.54	2	4748.60	40.72	1	686.04	...	364.17	16829.40	119.46	2	-244.62	17.19	-1171.32	4	252.58	1753.450118
2337.36	2.02	-17.90	12620.40	1	3211.32	-140.38	3	447.63	...	279.09	7122.78	274.47	3	-440.55	16.79	-2247.80	2	132.25	543.453678
2830.77	0.04	-35.14	-102.93	0	3602.00	387.18	4	379.98	...	122.79	21703.32	258.99	5	-269.04	13.77	-1335.86	5	138.75	1895.707928
1331.07	7.36	-15.38	-1616.13	0	3375.24	126.93	4	467.16	...	216.78	33672.72	253.86	4	-312.69	11.74	-2744.98	4	182.63	-293.797013
1511.70	3518.00	-29.04	4321.44	1	5541.96	34.98	3	510.45	...	348.81	15039.12	119.85	5	-415.77	16.55	-408.02	5	165.61	1231.815088
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2996.94	0.10	-18.56	4109.49	0	3201.72	-157.34	2	488.10	...	201.03	18926.82	174.06	4	-336.81	22.63	-1786.56	3	174.04	1447.151324
1908.57	0.24	-20.78	4844.52	2	3507.28	226.12	1	451.86	...	226.74	24845.00	219.75	2	-366.21	25.27	-1621.60	6	114.27	1815.640317
2215.83	3.56	-14.30	-1954.86	3	4490.90	500.26	0	421.89	...	236.61	23839.86	254.19	5	-183.87	12.09	-890.46	4	223.96	441.445043
598.17	0.96	-16.08	5670.54	2	3295.42	364.05	4	560.52	...	617.01	22592.04	271.26	2	-254.28	20.52	-2383.38	1	176.55	971.048989
818.76	224929.74	-24.02	25689.90	0	3843.04	18.85	3	162.18	...	327.00	12815.08	108.15	4	-190.02	16.74	-2001.94	2	178.09	1987.442009