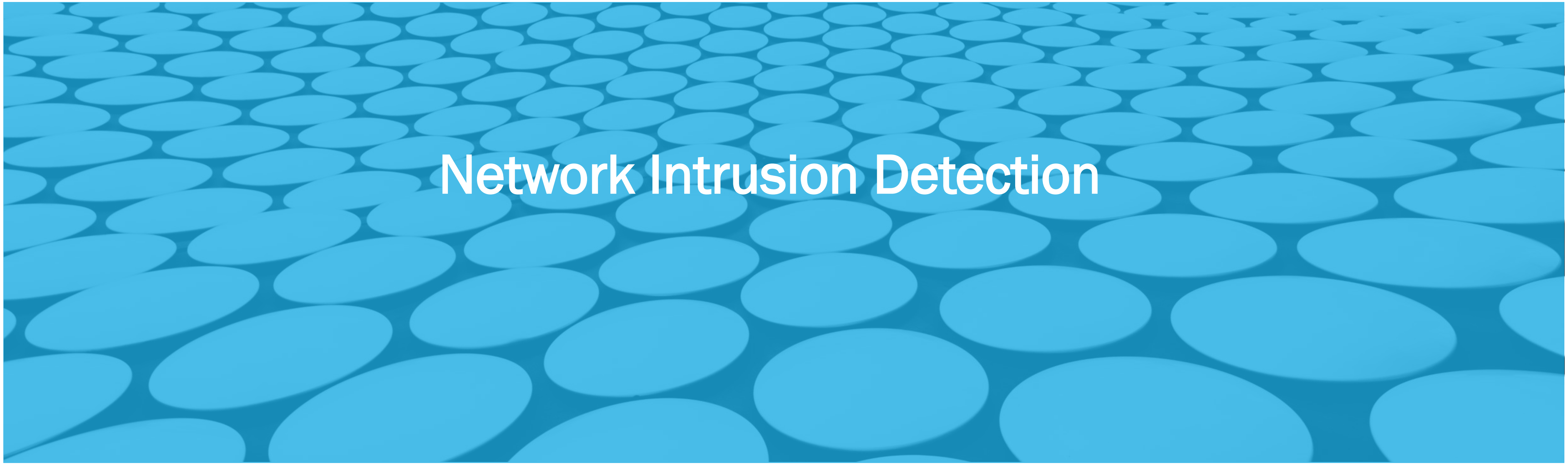
COMPUTER NETWORKS PROJECTS

- BY MANISH SRI SAI SURYA ROUTHU



Network Intrusion Detection

PROJECT 1: NETWORK INTRUSION DETECTION

INTRODUCTION:

Brief overview of modern computer networks and their growth:

Modern computer networks are interconnected systems of computers, devices, and servers that enable the exchange of data and resources. They play a pivotal role in facilitating communication, data sharing, and access to information across various industries and sectors.

IoT, Wireless Connectivity, cloud computing, Software Defined Networks, Virtualization, Big-data, Cyber threats

problem of increasing network intrusions and their potential impact:

1. **Cybersecurity Threats**
2. **Data Breaches**
3. **Financial Losses**
4. **Disruption of Critical Infrastructure**
5. **Ransomware Attacks**
6. **Intellectual Property Theft**
7. **Phishing and Social Engineering**
8. **Loss of Trust and Reputation**

...CONTINUED

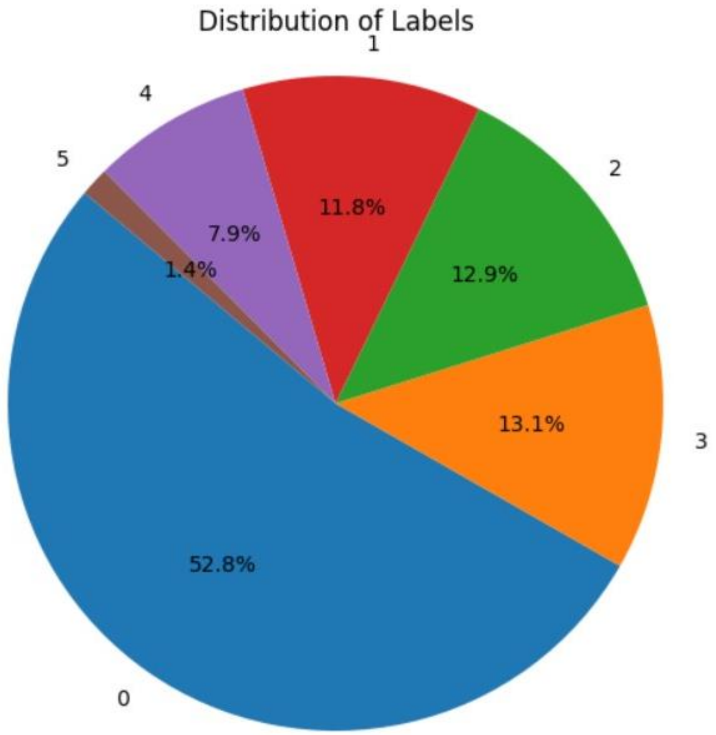
Our dataset ensures that there are enough samples for ML classifiers to achieve high F-Measure scores, uniquely. Our proposed dataset also ensures that there are no missing network metrics and that all data samples are filled.

Significance of machine learning in NIDS (Network Intrusion Detection Systems) –

1. Anomaly detection
2. pattern recognition
3. real-time monitoring
4. enhanced feature extraction
5. Automation in the system
6. threat intelligence and prediction

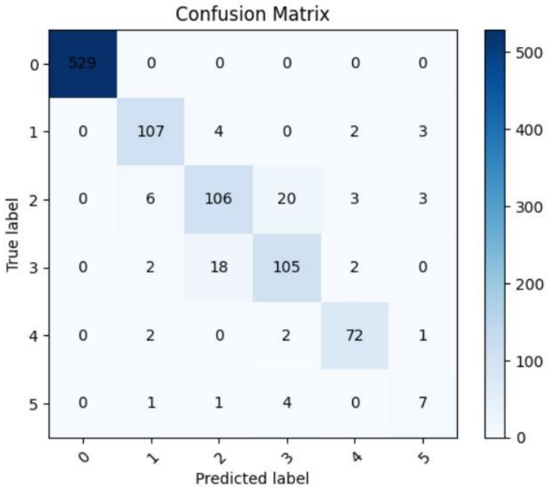
ABOUT THE DATASET

- The dataset consists of 3 csv files: train_dataset.csv, test_dataset.csv, submission.csv
- train_dataset.csv – 32 columns – 31 features + 1 label | 5000 records
- The datatype of all the columns in Int64 – no null values.
- A great difference is seen in variances and mean values of all the columns – Standardization is a must.
- There are 11 features with value count = 1 and hence those columns are removed
- ['Packets Rx Dropped', 'Packets Tx Dropped', 'Packets Rx Errors', 'Packets Tx Errors', 'Delta Packets Rx Dropped', ' Delta Packets Tx Dropped', 'Delta Packets Rx Errors', 'Delta Packets Tx Errors', 'is_valid', 'Table ID', 'Max Size’]
- 21 features left.
- Features with value counts< 10 are also checked
- Labels: 0 – Normal, 1 – Blackhole, 2 – TCP-SYN, 3 – Port Scan, 4 – Diversion, 5 - Overflow



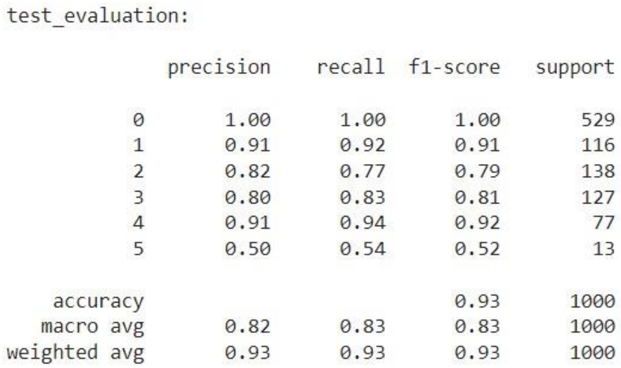
MACHINE LEARNING MODELS:

1. Decision Tree Classifier



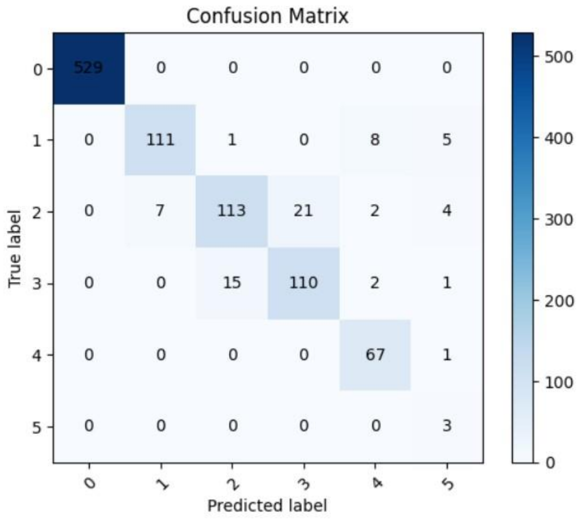
Confusion Matrix

True label \ Predicted label	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	107	4	0	2	3
2	0	6	106	20	3	3
3	0	2	18	105	2	0
4	0	2	0	2	72	1
5	0	1	1	4	0	7



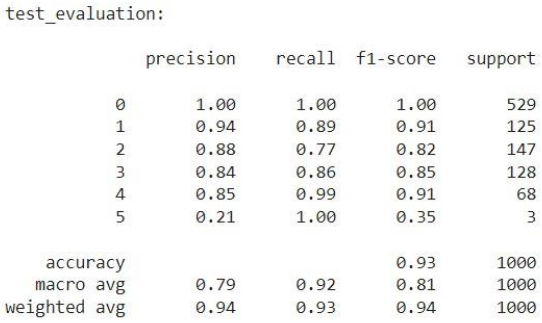
```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.91      0.92      0.91       116
           2       0.82      0.77      0.79       138
           3       0.80      0.83      0.81       127
           4       0.91      0.94      0.92        77
           5       0.50      0.54      0.52        13

    accuracy                           0.93      1000
   macro avg       0.82      0.83      0.83      1000
weighted avg       0.93      0.93      0.93      1000
```

2. Random forest Classifier



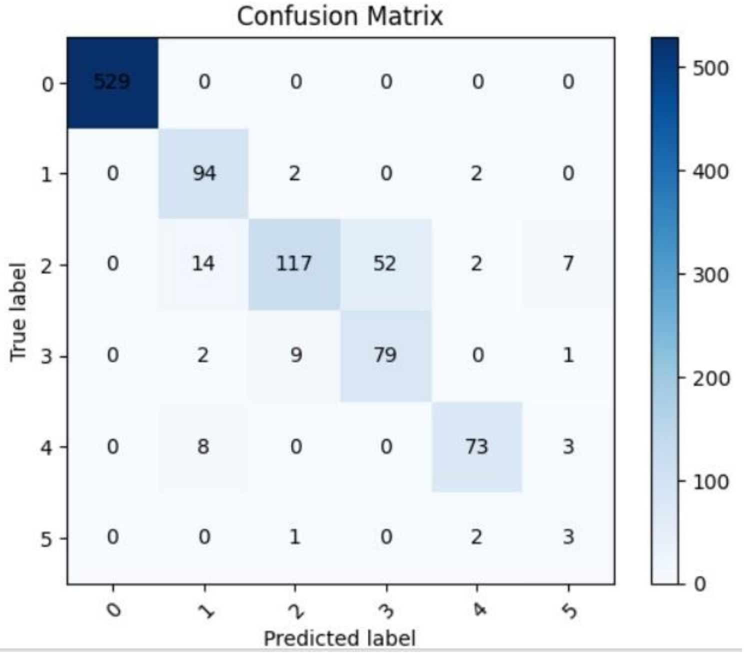
Confusion Matrix

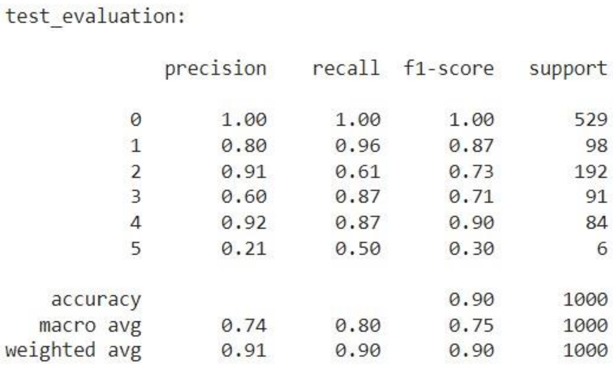
True label \ Predicted label	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	111	1	0	8	5
2	0	7	113	21	2	4
3	0	0	15	110	2	1
4	0	0	0	0	67	1
5	0	0	0	0	0	3



```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.94      0.89      0.91       125
           2       0.88      0.77      0.82       147
           3       0.84      0.86      0.85       128
           4       0.85      0.99      0.91        68
           5       0.21      1.00      0.35         3

    accuracy                           0.93      1000
   macro avg       0.79      0.92      0.81      1000
weighted avg       0.94      0.93      0.94      1000
```

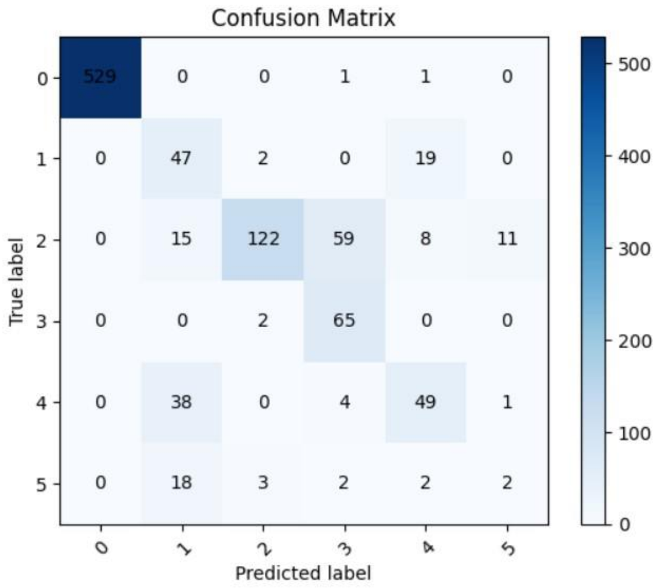
3. Logistic Regression

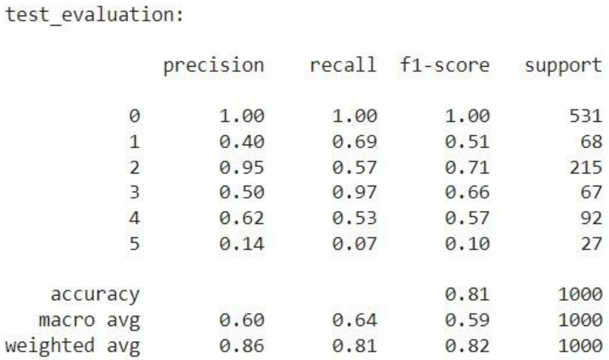




```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.80      0.96      0.87        98
           2       0.91      0.61      0.73       192
           3       0.60      0.87      0.71        91
           4       0.92      0.87      0.90        84
           5       0.21      0.50      0.30         6

    accuracy                           0.90      1000
   macro avg       0.74      0.80      0.75      1000
weighted avg       0.91      0.90      0.90      1000
```

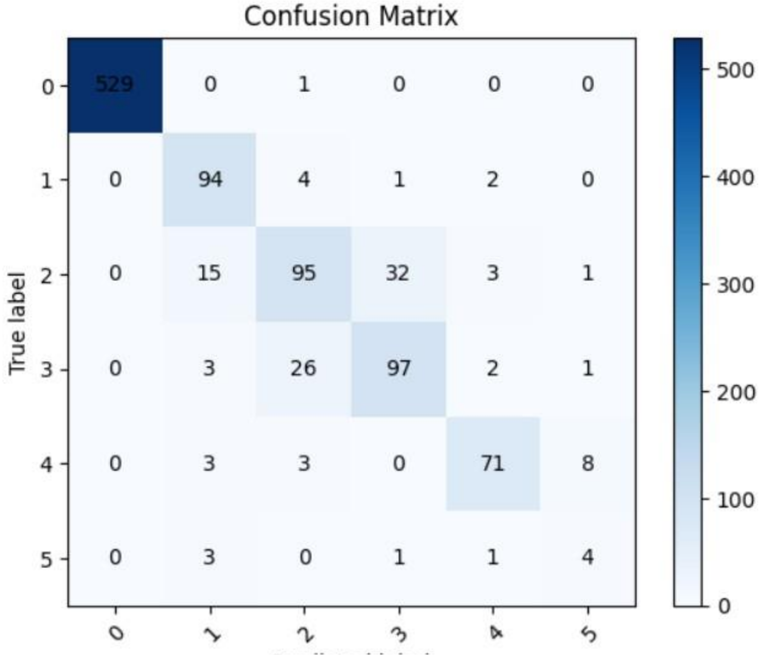
4. Naïve Bayes Classifier





```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       531
           1       0.40      0.69      0.51        68
           2       0.95      0.57      0.71       215
           3       0.50      0.97      0.66        67
           4       0.62      0.53      0.57        92
           5       0.14      0.07      0.10        27

    accuracy                           0.81      1000
   macro avg       0.60      0.64      0.59      1000
weighted avg       0.86      0.81      0.82      1000
```

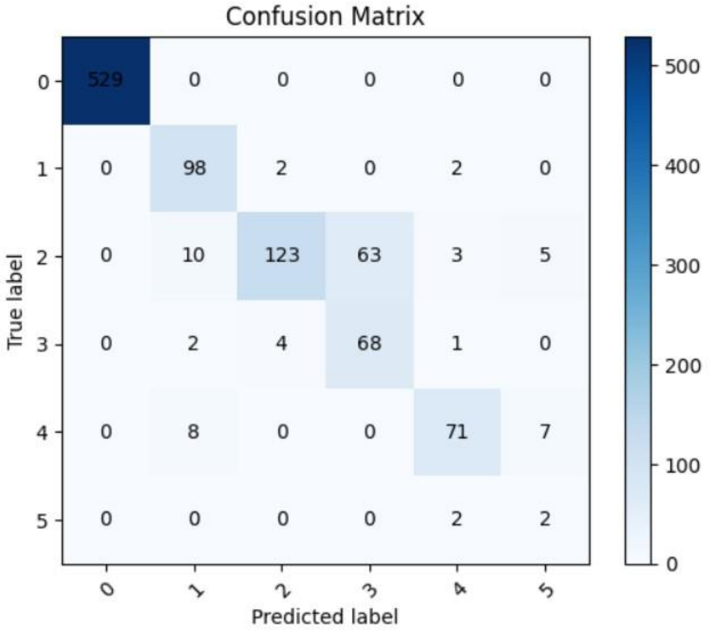
5. KNN Classifier



	0	1	2	3	4	5
0	529	0	1	0	0	0
1	0	94	4	1	2	0
2	0	15	95	32	3	1
3	0	3	26	97	2	1
4	0	3	3	0	71	8
5	0	3	0	1	1	4

```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       530
           1       0.80      0.93      0.86       101
           2       0.74      0.65      0.69       146
           3       0.74      0.75      0.75       129
           4       0.90      0.84      0.87        85
           5       0.29      0.44      0.35         9

    accuracy                           0.89      1000
   macro avg       0.74      0.77      0.75      1000
weighted avg       0.89      0.89      0.89      1000
```

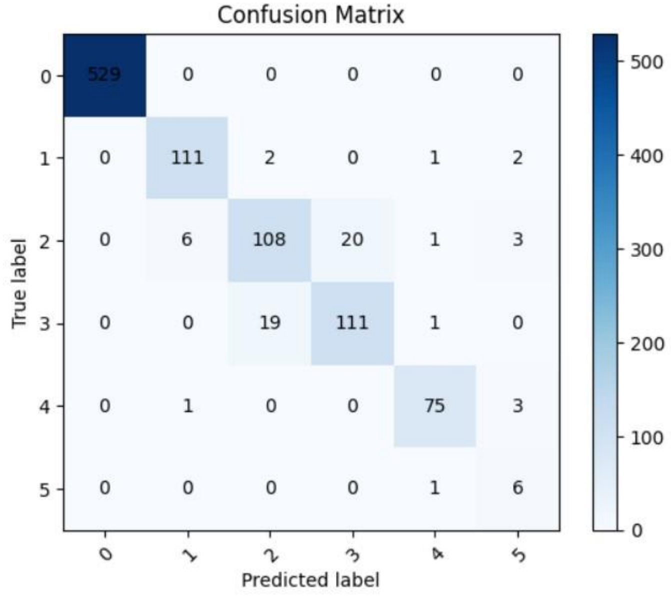
4. Support Vector Classifier



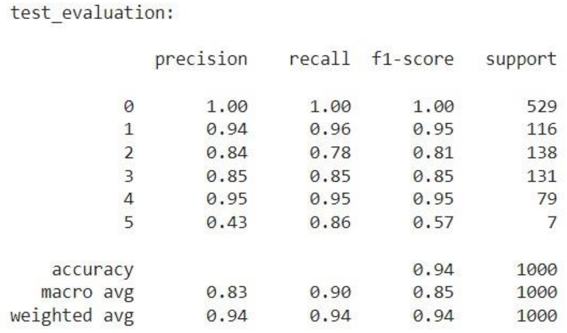
	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	98	2	0	2	0
2	0	10	123	63	3	5
3	0	2	4	68	1	0
4	0	8	0	0	71	7
5	0	0	0	0	2	2

```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.83      0.96      0.89       102
           2       0.95      0.60      0.74       204
           3       0.52      0.91      0.66        75
           4       0.90      0.83      0.86        86
           5       0.14      0.50      0.22         4

    accuracy                           0.89      1000
   macro avg       0.72      0.80      0.73      1000
weighted avg       0.93      0.89      0.89      1000
```

7. XGBoost Classifier

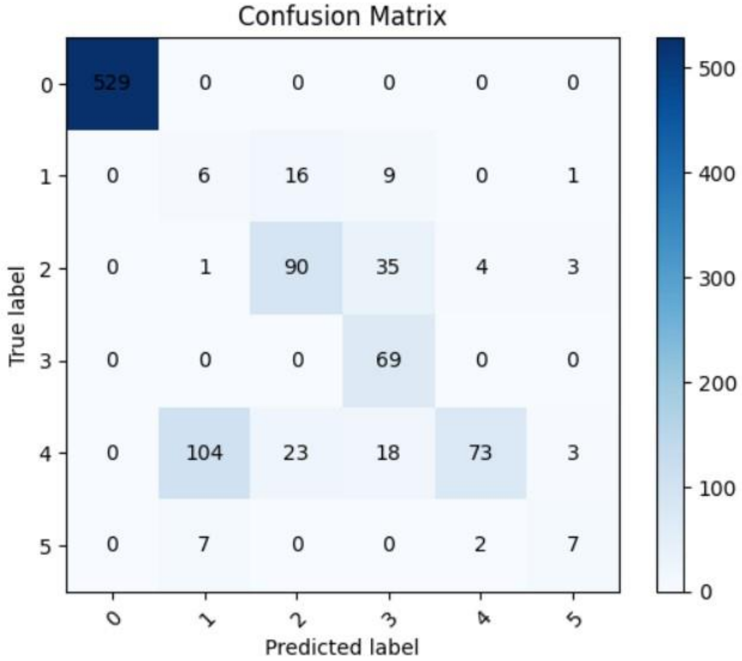


True label \ Predicted label	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	111	2	0	1	2
2	0	6	108	20	1	3
3	0	0	19	111	1	0
4	0	1	0	0	75	3
5	0	0	0	0	1	6

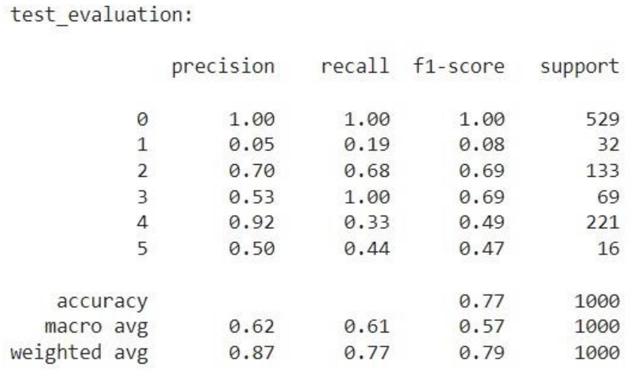


```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.94      0.96      0.95       116
           2       0.84      0.78      0.81       138
           3       0.85      0.85      0.85       131
           4       0.95      0.95      0.95        79
           5       0.43      0.86      0.57         7

    accuracy                           0.94      1000
   macro avg       0.83      0.90      0.85      1000
weighted avg       0.94      0.94      0.94      1000
```

8. Adaboost

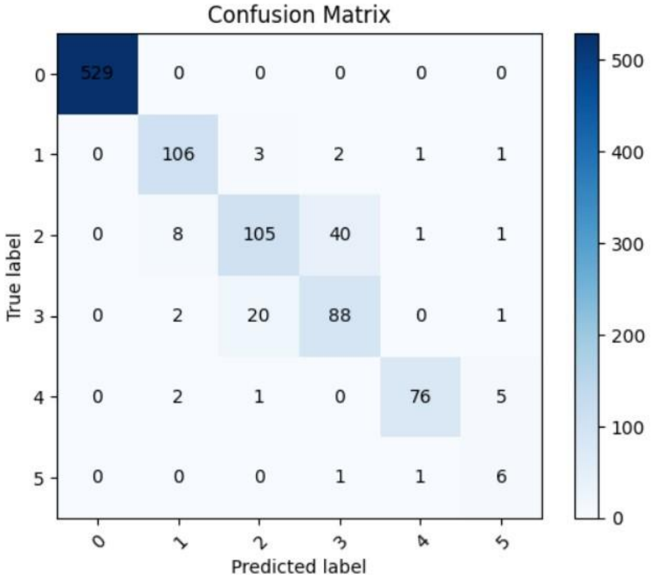


True label \ Predicted label	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	6	16	9	0	1
2	0	1	90	35	4	3
3	0	0	0	69	0	0
4	0	104	23	18	73	3
5	0	7	0	0	2	7

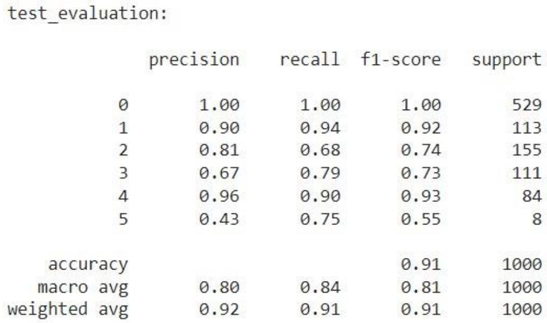


```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.05      0.19      0.08        32
           2       0.70      0.68      0.69       133
           3       0.53      1.00      0.69        69
           4       0.92      0.33      0.49       221
           5       0.50      0.44      0.47        16

    accuracy                           0.77      1000
   macro avg       0.62      0.61      0.57      1000
weighted avg       0.87      0.77      0.79      1000
```

9. MLP Classifier

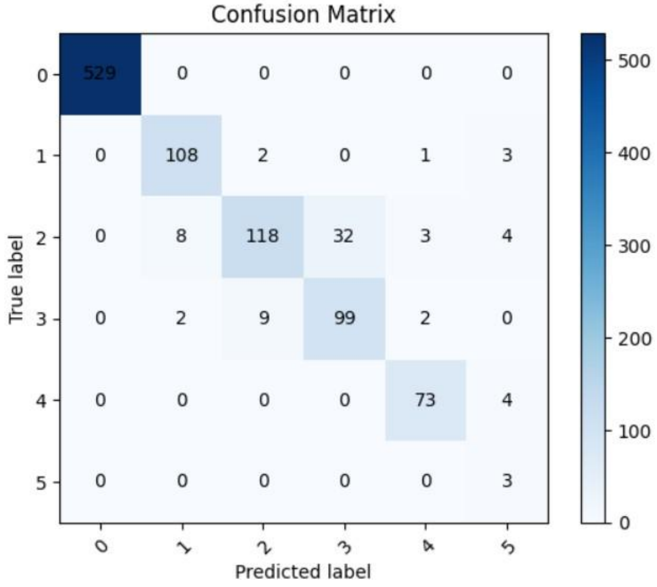


	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	106	3	2	1	1
2	0	8	105	40	1	1
3	0	2	20	88	0	1
4	0	2	1	0	76	5
5	0	0	0	1	1	6

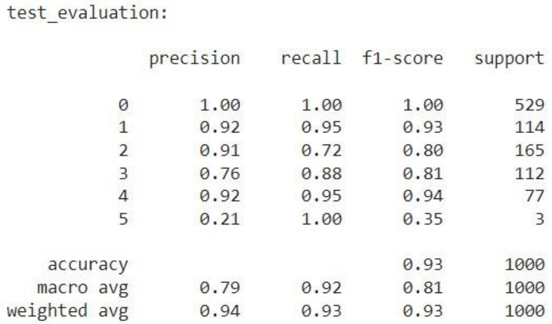


```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.90      0.94      0.92       113
           2       0.81      0.68      0.74       155
           3       0.67      0.79      0.73       111
           4       0.96      0.90      0.93        84
           5       0.43      0.75      0.55         8

    accuracy                           0.91      1000
   macro avg       0.80      0.84      0.81      1000
weighted avg       0.92      0.91      0.91      1000
```

10. Voting Classifier



	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	108	2	0	1	3
2	0	8	118	32	3	4
3	0	2	9	99	2	0
4	0	0	0	0	73	4
5	0	0	0	0	0	3



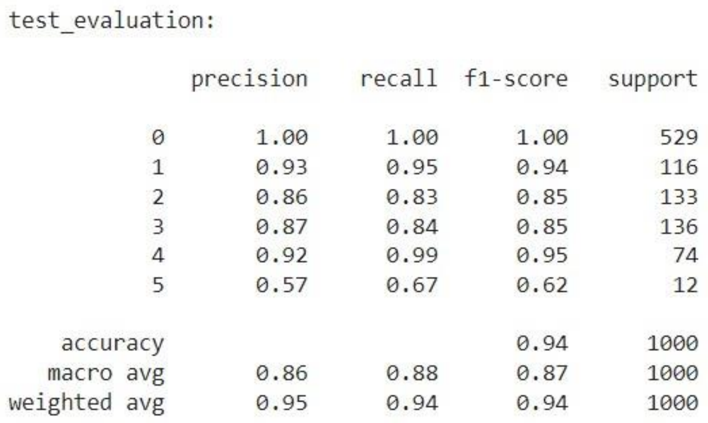
```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.92      0.95      0.93       114
           2       0.91      0.72      0.80       165
           3       0.76      0.88      0.81       112
           4       0.92      0.95      0.94        77
           5       0.21      1.00      0.35         3

    accuracy                           0.93      1000
   macro avg       0.79      0.92      0.81      1000
weighted avg       0.94      0.93      0.93      1000
```

HYPER TUNING:

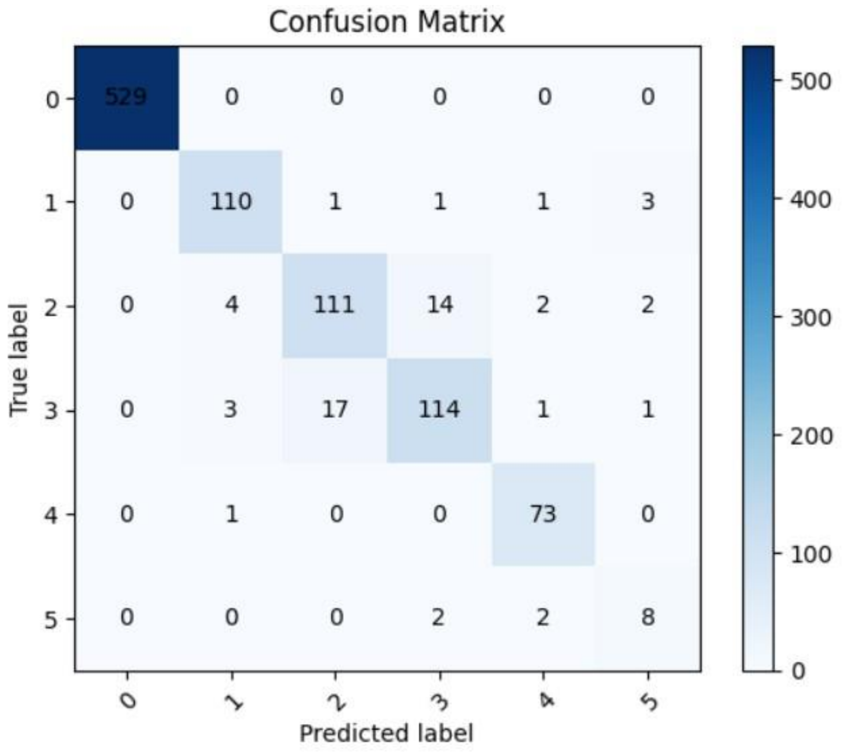
Hypertuning using 3 models:

1. Light GBM
2. RandomForestClassifier – accuracy -
3. Artificial Neural Network – accuracy – 90%

Light GBM:



```
test_evaluation:

              precision    recall  f1-score   support

           0       1.00      1.00      1.00       529
           1       0.93      0.95      0.94       116
           2       0.86      0.83      0.85       133
           3       0.87      0.84      0.85       136
           4       0.92      0.99      0.95        74
           5       0.57      0.67      0.62        12

    accuracy                           0.94      1000
   macro avg       0.86      0.88      0.87      1000
weighted avg       0.95      0.94      0.94      1000
```



Confusion Matrix

True label \ Predicted label	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	110	1	1	1	3
2	0	4	111	14	2	2
3	0	3	17	114	1	1
4	0	1	0	0	73	0
5	0	0	0	2	2	8

PROBLEMS:

1. Data imbalance: 0 – 2112, 5 – 57

To much imbalance of data – not much effect of “SMOTE” or “class_weight = ‘balanced’ “ on the model

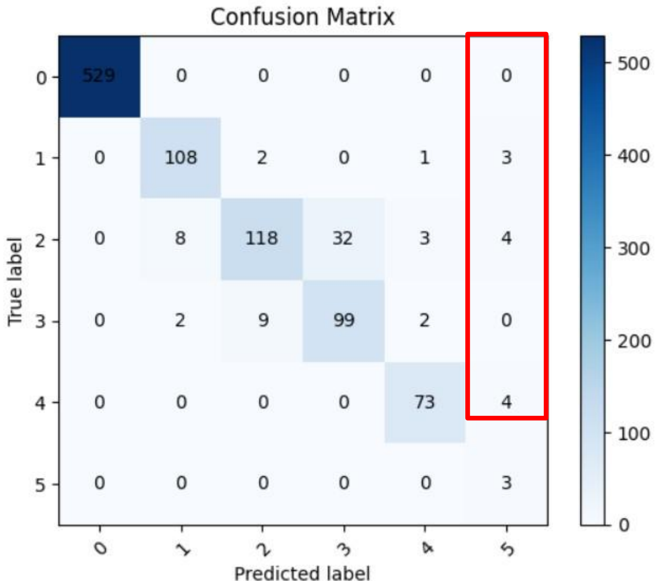
2. 20 features and 57 rows (for class 5)

Options: 1. Resampling: Oversampling/Undersampling

2. Class weight

3. Transfer Learning – TabNet (accuracy- 65%)

3. Majority of the wrong predictions despite of being less samplesare going to the final label.



	0	1	2	3	4	5
0	529	0	0	0	0	0
1	0	108	2	0	1	3
2	0	8	118	32	3	4
3	0	2	9	99	2	0
4	0	0	0	0	73	4
5	0	0	0	0	0	3

CONCLUSION:

1. Final Model – Light GBM which was giving 65% accuracy for the 5th Label.
2. Voting regressor did not perform well
3. Transfer Learning model was also not performing well because it may not be trained on that specific kind of data. [financial, customer, medical] and as we have data only 5000 samples with 20 features and 6 labels for training they are not sufficient for training very deep neural networks.

THANK YOU