Coursework Summary

Pranav Gaikwad
230276671
MSc Artificial Intelligence

1. INTRODUCTION

The summary explores a thorough analysis carried out as a requirement for the Artificial Intelligence Master's program's Data Mining coursework. In order to find actionable insights and predictive models, the project involved applying a number of data mining techniques to a predetermined dataset. This report's main objective is to provide an in-depth explanation of the procedures that must be followed in order to provide predictions on a holdout set, including exploratory data analysis, feature reduction, data pre-processing, and model creation.

2. EDA AND DATA PRE-PROCESSING

We started the EDA by basic summary of the dataset using pd.describe() feature of pandas, here is the snippet of the info we got,

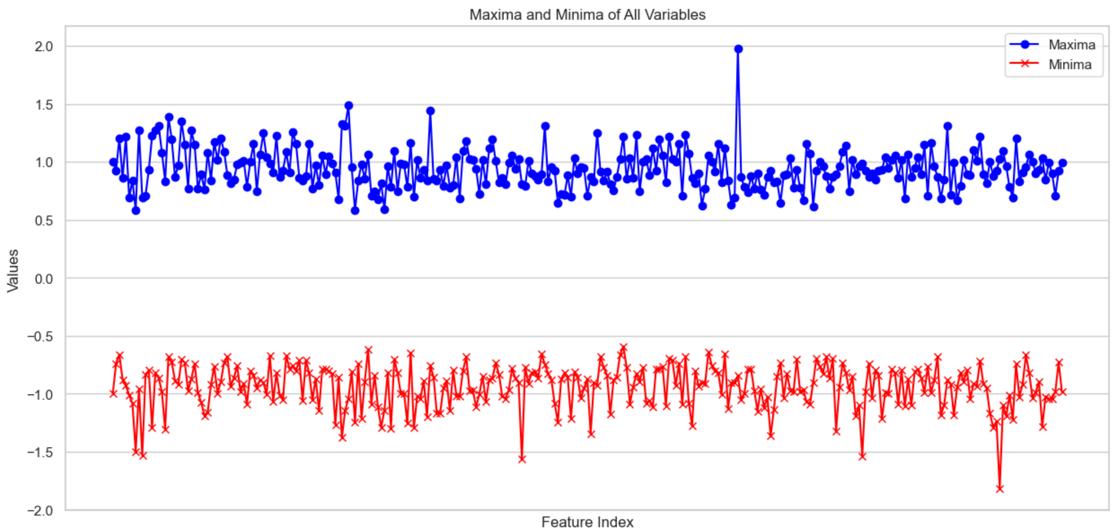
	Info_cluster	Class	feat_esm1b_0	feat_esm1b_1	feat_esm1b_2	feat_esm1b_3	feat_esm1b_4	feat_esm1b_5	feat_esm1b_6
count	12402.000000	12402.000000	12388.000000	12388.000000	12390.000000	12389.000000	12390.000000	12388.000000	12388.000000
mean	139.667634	-0.970005	0.040924	0.150334	0.068379	0.077347	0.023808	-0.136817	-0.094994
std	77.944928	0.243095	0.194674	0.180886	0.206083	0.183967	0.193493	0.193170	0.219265
min	7.000000	-1.000000	-0.739531	-0.664717	-0.918128	-0.931084	-1.010501	-1.086608	-1.499195
25%	70.000000	-1.000000	-0.084004	0.040529	-0.065351	-0.039926	-0.097352	-0.260591	-0.207169
50%	145.000000	-1.000000	0.034410	0.145975	0.069221	0.081350	0.035148	-0.135299	-0.071583
75%	205.000000	-1.000000	0.160128	0.255262	0.203641	0.201316	0.155194	-0.012340	0.051047
max	283.000000	1.000000	0.925082	1.203393	0.974194	1.217404	0.709081	0.838205	0.582508

8 rows × 292 columns

Next we checked for class imbalance, the dataset had a huge class imbalance with '-1' label having 12216 count and '1' as 186 count, later with sampling we can overcome this.

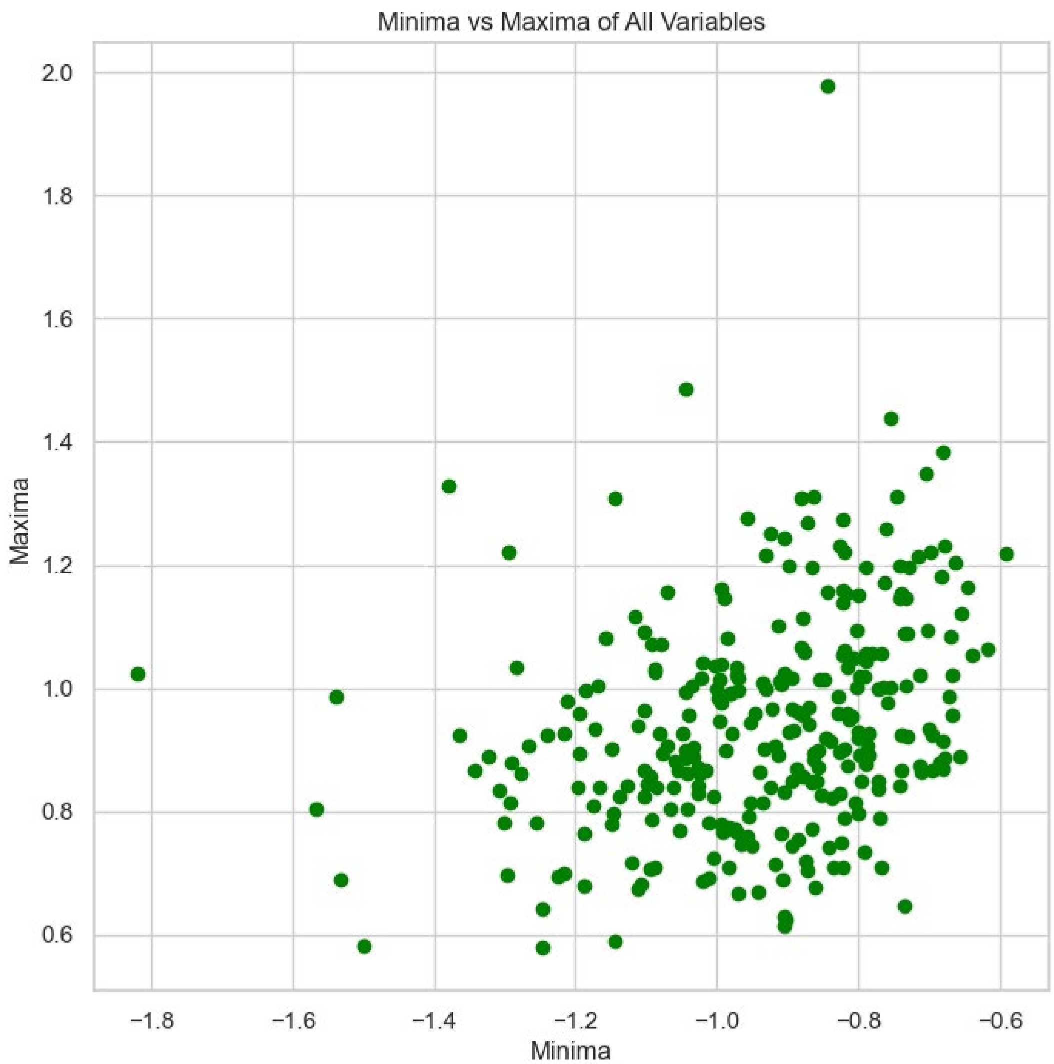
- **Scale Checking and Treatment**

In this step of EDA we check for the scale of all variables so that are they in need of scaling the features. Here are the graphs of Maxima and Minima of variables , Maxima vs Minima of the variables.

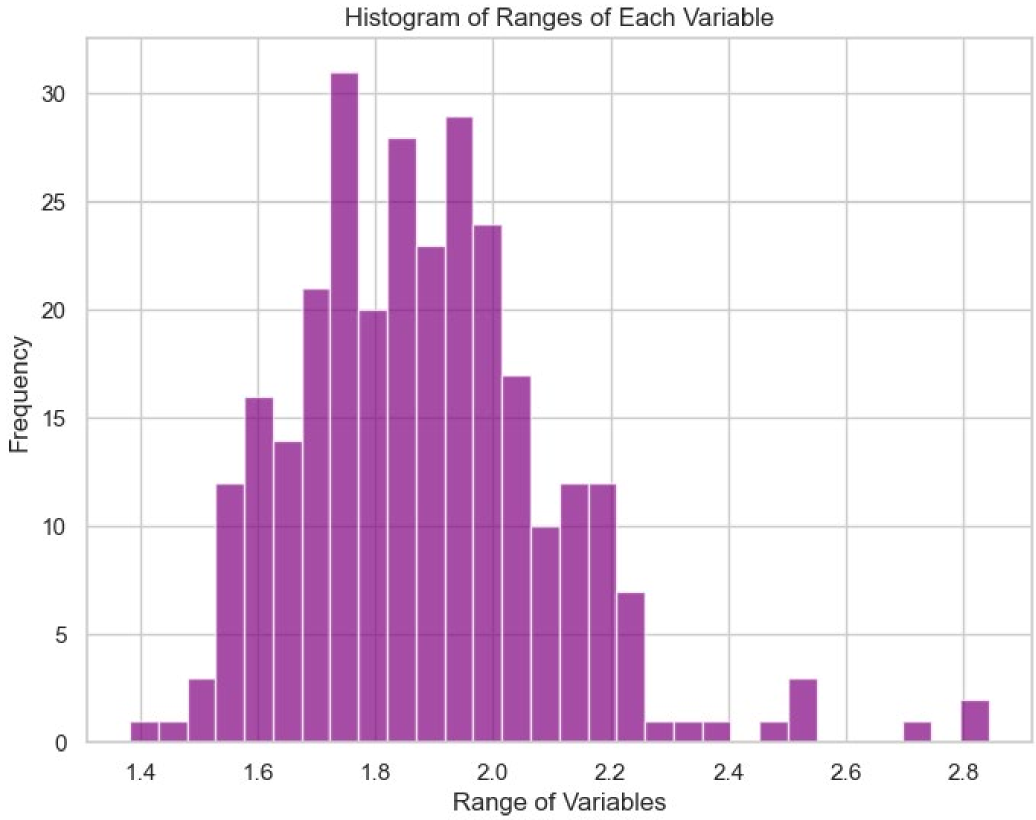


From the graph we notice few notable outliers, particularly one maximum value that is much higher than the rest and one minimum value that is much lower than the rest. These extreme values might indicate that the corresponding features have a wider range or are more volatile.

For further analysis we plot maxima vs minima of all features :



From this graph we inferred that there were two distinct outliers. One in the bottom left, which indicates a variable with both a very low minimum and a very low maximum. Another outlier is located near the top of the y-axis, indicating a variable with a small or moderate minimum but a very high maximum. Moreover, histogram of ranges of each variable pointed out that scaling of features is necessary in preprocessing step.



To overcome this we used MinMaxScaler() [1] to normalise the data between 0 and 1 except 'Info_Cluster' and 'Class'.

- **Outliers and Missing Values**

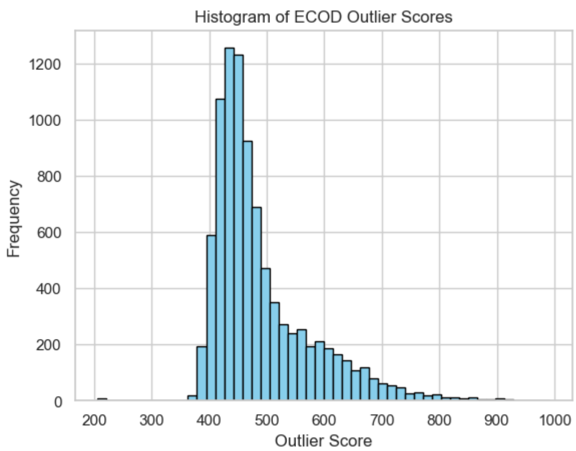
  We checked for missing values in the dataset and calculated the percentage of missing values :

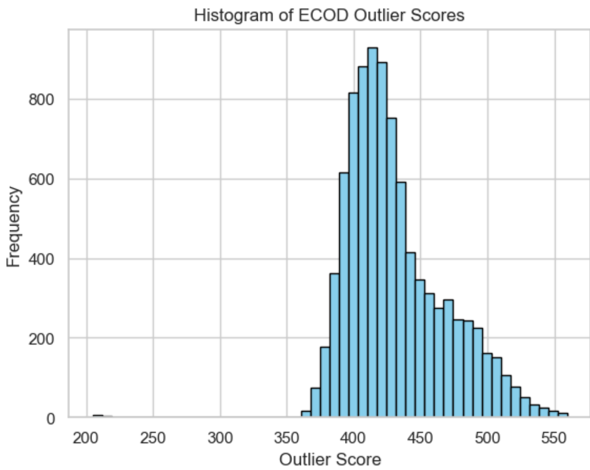
```
... (Info_cluster       0
     Class              0
     feat_esm1b_0      11
     feat_esm1b_1      11
     feat_esm1b_2      10
                       ..
     feat_esm1b_285    11
     feat_esm1b_286    11
     feat_esm1b_287     9
     feat_esm1b_288    11
     feat_esm1b_289    11
     Length: 292, dtype: int64,
     feat_esm1b_148    90.104338
     feat_esm1b_289     0.120813
     feat_esm1b_120     0.120813
     feat_esm1b_112     0.120813
     feat_esm1b_220     0.120813
                          ...
     feat_esm1b_267     0.087864
     feat_esm1b_270     0.076881
     feat_esm1b_119     0.065898
     Class              0.000000
     Info_cluster       0.000000
     Length: 292, dtype: float64,
        Info_cluster  Class  feat_esm1b_0  feat_esm1b_1  feat_esm1b_2  \
    ...
```

As we can see feature_1b_148 had 90% null or missing values so we dropped it and for the rest of the null values we used a **Simple Imputer** that replaces null values with **median**. After carrying out these steps there where no null values and missing values.

- **Outliers**

For outlier identification we used Empirical-Cumulative-distribution-based Outlier Detection (ECOD) [2] and for treatment of the outlier we used **Winisorize** [3] method and winisorize the data by keeping the upper limit and lower limit 10 %. Below are the ECOD scores histogram after Winisorization and before Winisorization :





- **Splitting According to ‘Info_Cluster’**

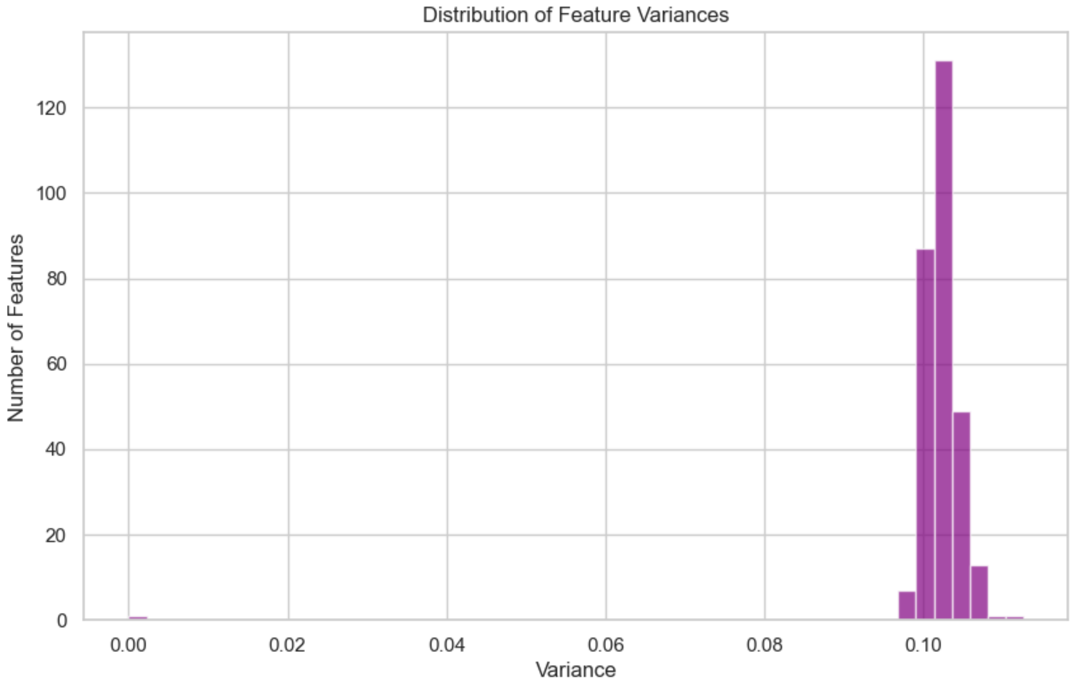
We split the data into train and test around Info_Cluster column for that we used **GroupShuffleSplit** [4] method from sklearn also while maintaining the class imbalance same across train data as well as test data.

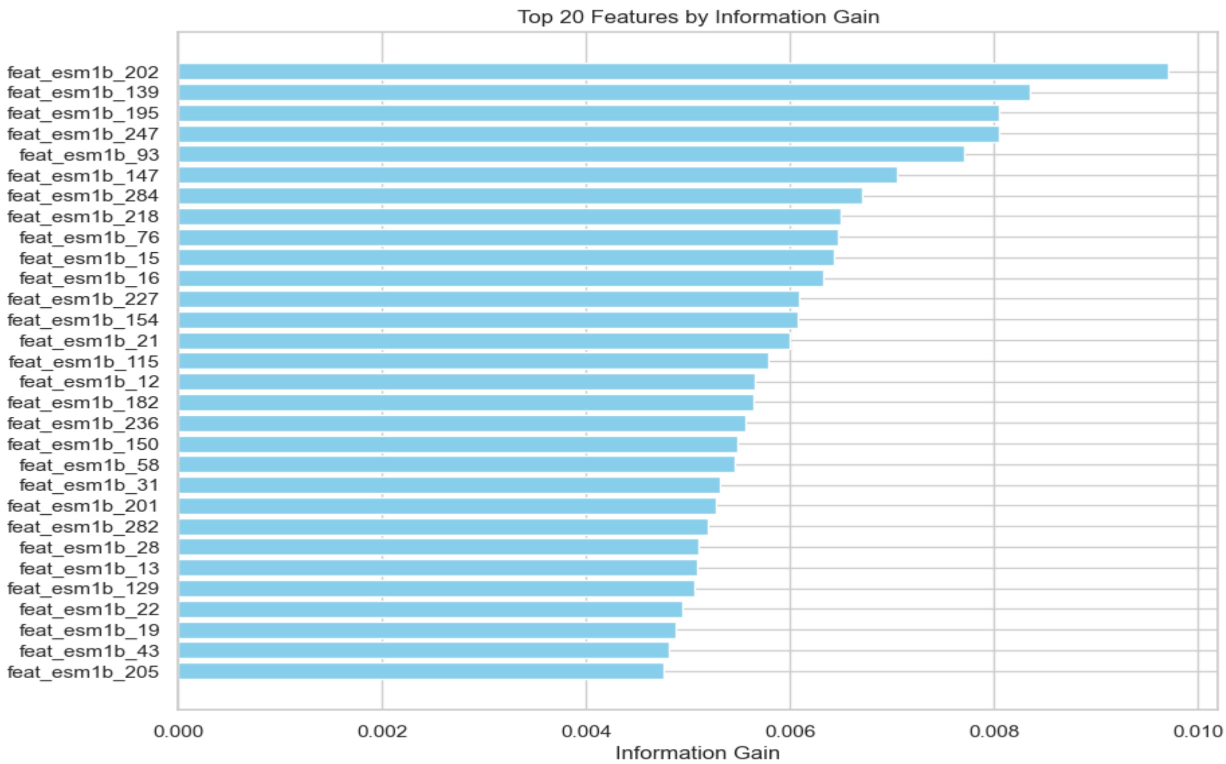
```
Training set class balance:
 -1    0.984514
 1    0.015486
Name: Class, dtype: float64

Test set class balance:
 -1    0.986351
 1    0.013649
Name: Class, dtype: float64
```

3. FEATURE REDUCTION

For Feature reduction we tried different methods like Variance Threshold , Information Gain and   (MRMR) [5].

We used Variance Threshold and Information Gain to select Top 200 features and then we used MRMR to select Top 30 features, below are the graphs showing outputs of each method and the final image contains the selected Top_30 features.





These are the Top_30 features selected after MRMR:

```
Selected features by MRMR: ['feat_esm1b_247', 'feat_esm1b_251',
'feat_esm1b_94', 'feat_esm1b_179', 'feat_esm1b_66', 'feat_esm1b_198',
'feat_esm1b_104','feat_esm1b_116','feat_esm1b_166','feat_esm1b_215',
'feat_esm1b_231','feat_esm1b_19', 'feat_esm1b_127', 'feat_esm1b_158',
'feat_esm1b_58','feat_esm1b_270', 'feat_esm1b_205', 'feat_esm1b_218',
'feat_esm1b_115','feat_esm1b_22', 'feat_esm1b_276', 'feat_esm1b_278',
'feat_esm1b_138','feat_esm1b_164', 'feat_esm1b_120', 'feat_esm1b_31',
'feat_esm1b_264', 'feat_esm1b_244', 'feat_esm1b_271', 'feat_esm1b_72'
]
```

- **Class Imbalance and Train_Val Split**

  As discussed in the EDA section we had a huge class imbalance to overcome it we tried different Oversmapling , Undersampling methods. These are the outputs of different sampling methods [6] :

```
Tomel Links :
 -1    8940
 1     141
Name: Class, dtype: int64
CNN :
 -1    278
 1    141
Name: Class, dtype: int64
ASASYN :
 -1    8964
 1    8955
Name: Class, dtype: int64
BorderlineSMOTE :
 -1    8964
 1    8964
Name: Class, dtype: int64
```

  After this we tried each sampling technique using Decesion Tree Classifier in which ASADYN had the highest balanced accuracy and F1 score, so we decided to use it for final modelling.

  We then next splitted the data into train and val for parameters tuning using **GroupKFold** method of sklearn.

4. MODELLING AND ASSESSMENT

For modelling and hyper-parameter tuning we use different classification models with different parameters and find the best parameters using grid search CV.

We used classification models like Logistic Regression, RandomForest Classifier and Gradient Boosting Classfier with different parameters and tested them on the validation set.

Below are the classification report and balanced accuracy of each model:

Logistic Regression :

```
Best parameters for LogisticRegression: {'C': 10, 'class_weight': {-1: 1, 1: 10}, 'penalty': 'l2'}
Classification report for LogisticRegression:
              precision    recall  f1-score   support

          -1       0.75      0.61      0.67      1603
           1       0.73      0.83      0.78      1980

    accuracy                           0.73      3583
   macro avg       0.74      0.72      0.72      3583
weighted avg       0.74      0.73      0.73      3583

Balanced Accuracy :  0.7223879468420953
```

Random Forest Classifier :

```
Best parameters for RandomForestClassifier: {'max_depth': 20, 'n_estimators': 100}
Classification report for RandomForestClassifier:
              precision    recall  f1-score   support

          -1       0.60      1.00      0.75      1603
           1       0.99      0.47      0.64      1980

    accuracy                           0.71      3583
   macro avg       0.80      0.73      0.70      3583
weighted avg       0.82      0.71      0.69      3583

Balanced Accuracy :  0.7338683150910225
```

Decision Tree Classifier :

```
Classification report for DecisionTreeClassifier:
              precision    recall  f1-score   support

          -1       0.58      0.96      0.73      1603
           1       0.94      0.44      0.60      1980

    accuracy                           0.67      3583
   macro avg       0.76      0.70      0.66      3583
weighted avg       0.78      0.67      0.65      3583

Balanced Accuracy :  0.7009820601523659
```

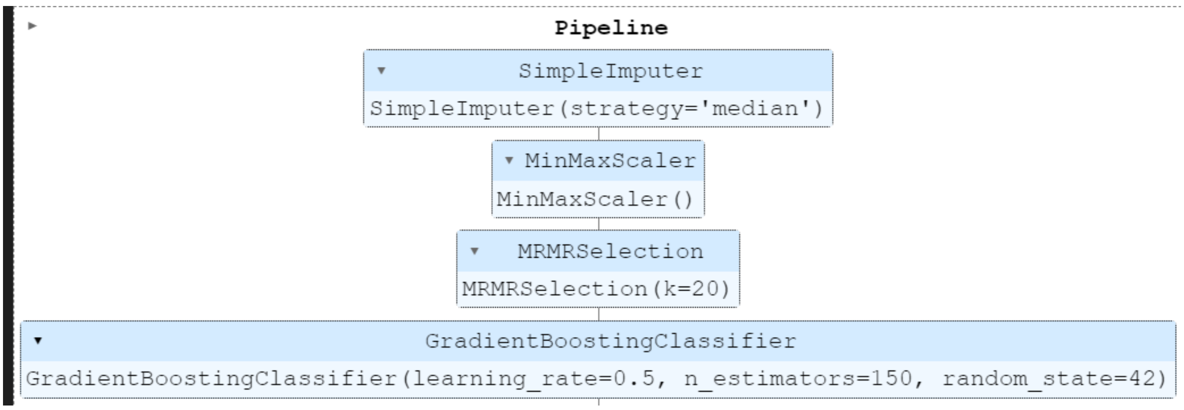
Gradient Boosting Classifier :

```
Best parameters for GradientBoostingClassifier: {'learning_rate': 0.5, 'n_estimators': 150}
Classification report for GradientBoostingClassifier:
              precision    recall  f1-score   support

          -1       0.63      0.99      0.77      1603
           1       0.98      0.52      0.68      1980

    accuracy                           0.73      3583
   macro avg       0.80      0.75      0.72      3583
weighted avg       0.82      0.73      0.72      3583

Balanced Accuracy :  0.7546053170507319
```

As we can see after using various model Gradient Boosting Classifier had the best **F1 Score**, **Accuracy** and **Balanced Accuracy** of **0.77, 0.73,0.75.**

This is the model we use for are final pipeline and other Data-Preprocessing steps. Below is the final pipeline.

Note - While creating pipeline we needed a custom MRMR function as MRMR doesn't integrate with sklearn.



Finally the created pipeline was used on holdout dataset to carry out predictions and create a CSV file of predictions.

5. CONCLUSIONS AND DISCUSSION

The analysis effectively demonstrated the application of data mining techniques in extracting meaningful patterns and predictions from the dataset. Key findings indicated significant predictors and their impacts on the target variable, as revealed by the feature importance scores from the final model.

Limitations of this analysis were primarily related to data quality and size, which may have influenced the model's performance and generalizability. Specific features exhibited a high degree of missing data, which posed challenges in imputation and might have skewed the results.

Future Directions could include incorporating additional data sources to enhance the robustness of the findings and exploring more advanced machine learning techniques such as deep learning for potentially improved predictive performance.