

## Machine Learning Lab : Feature Engineering

M.H.A.Ahmed


Department of Computer Science & Engineering

University of Moratuwa, Sri Lanka

[akeelahmed.19@cse.mrt.ac.lk](mailto:akeelahmed.19@cse.mrt.ac.lk)

# Introduction

In the realm of machine learning, raw data is the foundation upon which models are built. However, not all data is created equal. The process of feature engineering plays a pivotal role in transforming this raw data into meaningful and relevant features, thereby empowering machine learning models to better capture patterns and make accurate predictions.

This report elaborates on some techniques used and some experimental details in relation to the Google Colab Notebook :  final.ipynb

# Preprocessing

## Scaling

- Used Robust Scaler from scikit-learn  
The Robust Scaler is a feature scaling technique used in machine learning to mitigate the impact of outliers on the scaling process. Outliers are data points that significantly differ from the rest of the data. These extreme values can distort the scaling of features, particularly in methods that are sensitive to the range of values, such as Min-Max scaling and Standardization (Z-score normalisation).

## Handling Missing Values

- Removed the records with NaN  
Imputing missing values can introduce bias into the data, potentially affecting the integrity of our analysis or model. By removing missing values, we avoid introducing any potentially biased assumptions about the missing data.

## Handling Unequal Distribution

- Using the *class\_weight* parameter with balanced SVC (Support Vector Classifier) is a common and effective approach to address the issue of unequal class distribution in classification tasks. When dealing with imbalanced datasets, where one class has significantly fewer examples than the others, traditional machine learning algorithms might struggle to correctly classify the minority class.  
We could observe such unequal distribution in label\_4.

```
▼ Bias observed in label_4  
[271] TRAIN_DF[LABEL_4].value_counts()  
6    19938  
2     1449  
0      955  
12     954  
7      938  
13     482  
1      481  
11     480  
10     480  
3      479  
5      478  
9      472  
4      469  
8      465  
Name: label_4, dtype: int64
```

The `class_weight` parameter in SVC allows you to assign different weights to different classes. When set to 'balanced', the algorithm automatically adjusts the weights inversely proportional to the class frequencies. This means that the algorithm assigns higher weights to the minority class and lower weights to the majority class. This helps in giving more importance to the minority class during the training process, allowing the algorithm to learn from it more effectively.

## Classification Model

### Support Vector Classification

Support Vector Classifier (SVC) is chosen for its effectiveness in handling complex and non-linear relationships within data. It is a powerful supervised learning algorithm commonly used for classification tasks. SVC is particularly well-suited when data points are not linearly separable in their original feature space.

SVC employs the concept of "support vectors" to create a decision boundary that maximises the margin between different classes. These support vectors are the data points closest to the decision boundary and have the most influence on its position. By selecting the right kernel function (e.g., linear, polynomial, or radial basis function), SVC can transform the original feature space into a higher-dimensional space where the data might become linearly separable. This allows SVC to capture complex relationships that might not be evident in the original feature space.

In this lab, we have used the linear kernel function to train the model.

# Feature Engineering Techniques

## SelectKBest

SelectKBest is a feature selection technique used to select the most important features from a dataset based on a scoring function. The commonly used scoring function ANOVA F-test (Analysis of Variance F-test).is used with SelectKBest to rank and select top features. Selecting the top 'k' features based on ANOVA F-test reduces the dimensionality of the dataset, which leads to improved model performance..

## Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimensionality reduction technique widely used in machine learning and statistics. Its primary purpose is to transform high-dimensional data into a lower-dimensional representation while preserving as much of the original data's variance as possible. PCA achieves this by finding new orthogonal axes, known as principal components, along which the data varies the most.

A **combination** of the above two techniques were used in order to reduce the number of features while maintaining the accuracy without a considerable drop.

# Experiment Results

Label	Accuracy (Before)	K value	N_components	Accuracy (After)	No_of Features Remain
Label_1	0.9907	230	0.99	0.989	104
Label_2	0.895	240	0.99	0.82	105
Label_3	0.9986	100	0.99	0.9986	73
Label_4	0.9587	250	0.99	0.9306	106

## Conclusion

In summary, even though there has been a slight dip in accuracy, it's worth noting that this decline can be attributed to the intentional reduction of several features. This trade-off between accuracy and feature reduction is a common consideration during optimization and refinement phases. By focusing on essential features and eliminating unnecessary elements, the system's efficiency and user-friendliness are enhanced. While there may have been a

minor accuracy reduction, the advantages of streamlined operation, improved performance, and simplified user interaction resulting from the reduction of non-essential features far outweigh the marginal accuracy trade-off. This strategic choice to prioritise usability and efficiency through feature reduction underscores the idea that in specific contexts, a modest concession in one aspect can lead to noteworthy overall enhancements.