# Project 2 Report

## 1. Introduction:

In Machine Learning (ML), datasets often contain a high number of features. It is feasible to reduce these datasets' dimensionality to reduce computational costs while preserving predictive power. Reducing the number of attributes/features/dimensions of a dataset is known as dimensionality reduction. This project considered the impact of four dimensionality reduction techniques on the performance of two types of classification algorithms: SVM and Naive Bayes. The techniques involved were feature selection, feature importance, principal component analysis (PCA), and linear discriminant analysis (LDA).

## 2. Methodology:

### 2.1 Dataset:

The dataset consists of 5000 rows and 10 columns. Each row is a different city; the first nine columns represent a different feature containing information which combined with the information provided by the other columns would conclude the 'Air Quality' of a city, represented by the tenth column, the label; the label's possible categories are 'Good', 'Moderate', 'Poor' and 'Hazardous'. The features are 'Temperature', 'Humidity', 'PM2.5', 'PM10', 'NO2', 'SO2', 'CO', 'Proximity to Industrial Areas' and 'Population Density". The dataset was preprocessed before training any model by replacing any negative value with a 0.

### 2.2 Dimensionality reduction methods:

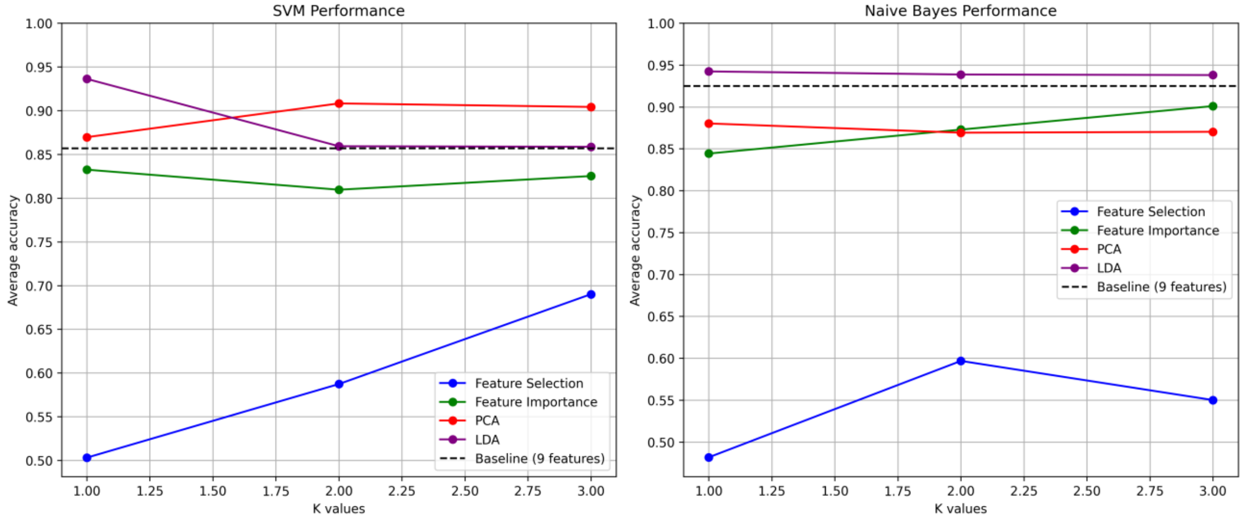
These methods can be split in two main subcategories of dimensionality reduction, feature selection, which retains the original features, and feature extraction which creates transformed features:

- Feature selection (selecting $K_f$ = 1,2,3 dimensions):
  - Univariate feature selection (FS)
  - Feature importance (FI), using a random forest to determine the feature importance scores.
- Feature extraction (with $K_s$ = 1,2,3 dimensions):
  - PCA
  - LDA

## 2.3 Classification methods:

Each reduced dataset was randomly split 50 times, using 75% of the data for training and 25% of the data for testing. For each split, SVM and Gaussian Naive Bayes classifiers were trained, calculating average test accuracy across all 50 splits for each dimensionality reduction technique. This set up provides a comparison between dimensionality techniques and how they affect each classifier's test accuracy. In order to get a better understanding of the impacts on model accuracy, each classifier was also trained and tested using all 9 features (No dimensionality reduction). In the project, this kind of training is referred to as baseline.

# 3. Results:



SVM and Naive Bayes classifiers average accuracies per dimensionality reduction technique depending on K values

The plots represent SVM and Naive Bayes classifiers performance in terms of accuracy. The average accuracy of each classifier type is plotted with respect to the dimensionality reduction technique used to preprocess the dataset used to train such classifier.

| | SVM | | | | | Naive Bayes (Gaussian) | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| $K_{f/s}$ | FS | FI | PCA | LDA | Baseline | FS | FI | PCA | LDA | Baseline |
| 1 | 0.5032 | **0.8324** | 0.8696 | **0.9362** | 0.8589 | 0.4817 | 0.8443 | **0.8802** | **0.9423** | 0.9252 |
| 2 | 0.5873 | 0.8095 | **0.9081** | 0.8592 | 0.8589 | **0.5968** | 0.8728 | 0.8692 | 0.9386 | 0.9252 |
| 3 | **0.6901** | 0.8251 | 0.9041 | 0.8586 | 0.8589 | 0.5500 | **0.9009** | 0.8703 | 0.9379 | 0.9252 |

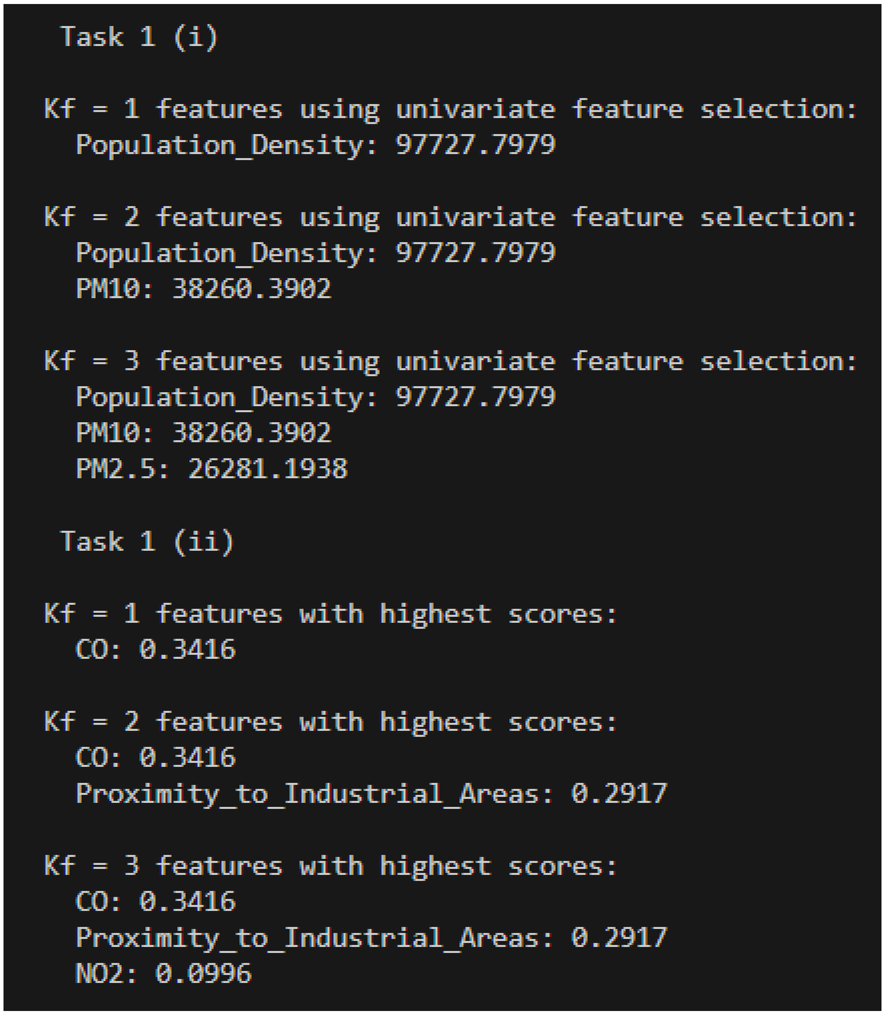
Baseline accuracy does not vary with K because it uses all 9 features.

The features selected using univariate feature selection were:

- $K_f$ = 1: ‘Population_Density’.
- $K_f$ = 2: ‘Population_Density’, ‘PM10’.
- $K_f$ = 3: ‘Population_Density’, ‘PM10’, and ‘PM2.5’.

Using feature importance:

- $K_s$ = 1: ‘CO’.
- $K_s$ = 2: ‘CO’, ‘Proximity_to_Industrial_Areas’.
- $K_s$ = 3: ‘CO’, ‘Proximity_to_Industrial_Areas’, and ‘NO2’.



```
 Task 1 (i)

Kf = 1 features using univariate feature selection:
  Population_Density: 97727.7979

Kf = 2 features using univariate feature selection:
  Population_Density: 97727.7979
  PM10: 38260.3902

Kf = 3 features using univariate feature selection:
  Population_Density: 97727.7979
  PM10: 38260.3902
  PM2.5: 26281.1938

 Task 1 (ii)

Kf = 1 features with highest scores:
  CO: 0.3416

Kf = 2 features with highest scores:
  CO: 0.3416
  Proximity_to_Industrial_Areas: 0.2917

Kf = 3 features with highest scores:
  CO: 0.3416
  Proximity_to_Industrial_Areas: 0.2917
  NO2: 0.0996
```

# 4. Discussion/Analysis Section (Task 5):

As figure 1 shows, Naive Bayes models performed better on average. Despite the better average accuracy for NB models, the only data preprocessing technique which allowed this kind of classifier to perform better than with baseline was LDA (purple line), although the table shows a slight decrease in average accuracy as $K_s$ values increase, going from 0.9423 to 0.9379. Notably, as $K_f$ values increase, feature importance (green line) improves performance for NB models, with a 6% increase in average accuracy from 0.8443 to 0.9009.

SVM models performed better than baseline with PCA and LDA. These two preprocessing techniques appear to be opposite to each other. PCA starts lower and increases while LDA starts higher and decreases. With respect to PCA (red line) the SVM side performed better than NB side too, with $K_s > 1$, there's a major improvement in average accuracy (from 0.8696), which remained steady at $K_s=2$ and $K_s=3$ (0.9081 and 0.9041). Again, on the SVM side, feature selection (blue line) only improved as $K_f$ increased, with an average accuracy increase of about 19%, from 0.5032 to 0.6901, over the span of 2 $K_f$'s (from 1 to 3), which wasn't the same for the NB side.

Overall, as K values increased, the only models which showed constant performance improvement were SVM using feature selection and NB using feature importance.