Practical Machine Learning Project

1. Introduction

In the modern society, drugs are tested on artificial cells. How do we know if the drug is successfull or not? The data represents molecular descriptors that help determine the state of the molecule.

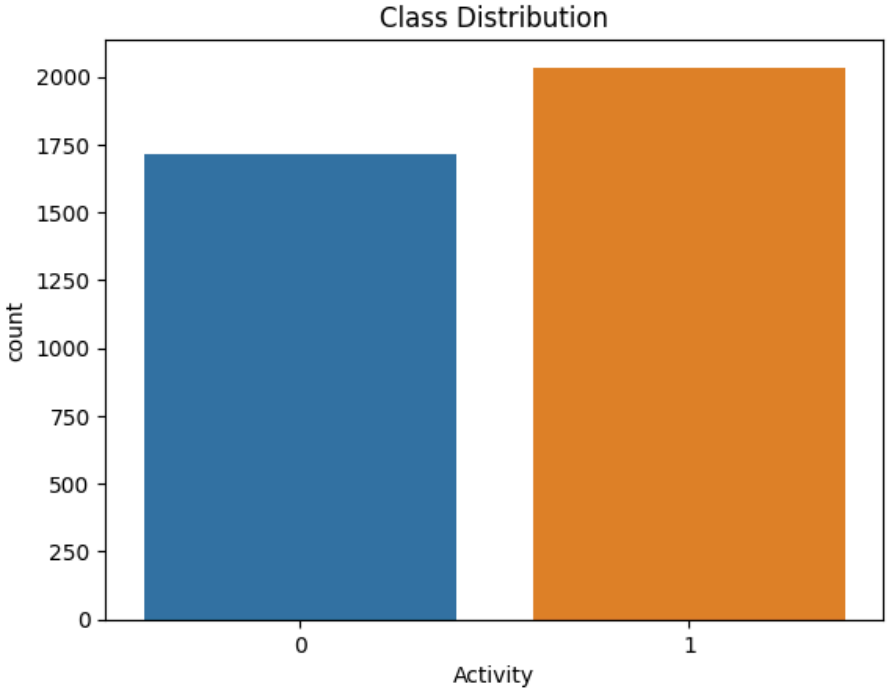
Molecular descriptors are of different categories, for example:

- Physicochemical
- Topological
- Eletrical
- Structural

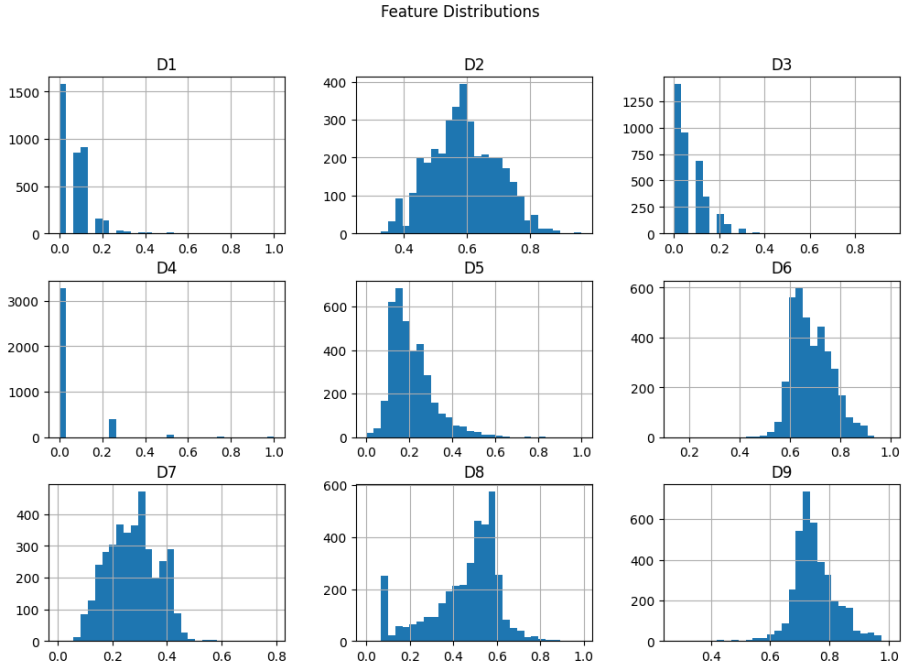
By predicting biological response we can determine if the drug has any effect or not for that specific molecule/cell. Meaning the posibility of no more human and animal trials for developing drugs.

2. Exploratory Data Analysis

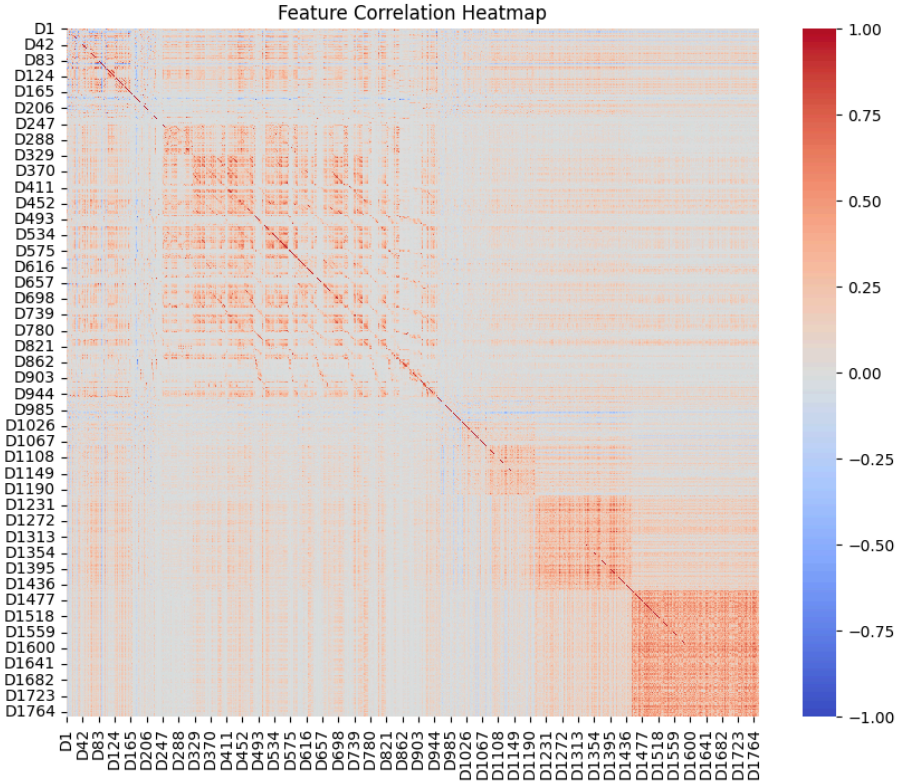
The presented dataset does not have any missing or duplicated values meaning the set is already cleaned and already has the missing values processed. From an initial look at the dataset it seems pretty consistent with all the values and the classes.



The distribution of features seems to be pretty consistent most of them being similar with the normal distribution which is a good sign of consistency. This can be observed in the below plot, also it helps identify potencial outliers.



The correlation heat-map helps us see potential influential initial features, for example we can see in the below plot that the last features are the most important regarding activity of the molecule/cell.



3. Preprocessing

For the preprocessing of the data I opted to firstly scale the features with the help of MinMaxScaler. From here i went for 2 different approaches:

1. PCA - feature extraction
2. VarianceThreshold - dropping features with low variance

Why those? Given the large amounts of data, we want to train a good model with relative low resources, that means we need to optimize the input for it. That said with PCA we only focus on the most important features as for the VT we remove the ones that has variance < 0.1. One important difference between those 2 approaches is that the PCA creates new liniar combinations of features and VT retains the more of the original features by removing only what has low variance (in the end we have a subset of the initial dataset).

4. Classifiers Used

Two widely used ensemble methods are LGBMClassifier (Light Gradient Boosting Machine) and ExtraTreesClassifier (Extremely Randomized Trees). While both models leverage decision trees as their fundamental components, they operate under different principles and are optimized for distinct use cases.

LGBMClassifier is a gradient boosting decision tree model designed for high efficiency and scalability. It follows a boosting approach, where trees are built sequentially, each attempting to correct the errors of the previous ones.

ExtraTreesClassifier is an ensemble learning method that extends the Random Forest algorithm by introducing additional randomness during tree construction. Unlike boosting methods that correct errors iteratively, ExtraTrees follows a bagging (bootstrap aggregating) approach, where multiple independent trees are trained in parallel. The final prediction is determined by aggregating the outputs of all trees.

5. Hyperparameter tunning

For tuning I used the gridsearch methodology in which i have for LGBMClassifier:

1. num_leaf = [20, 31, 40]
2. learning_rates = [0.01, 0.05, 0.1]
3. feature_fraction = [0.5, 0.7, 0.9]

4. max_depth = [-1, 5, 10]
5. metric = ['binary_error']

Best parameters: {'boosting_type': 'dart', 'feature_fraction': 0.5, 'learning_rate': 0.1, 'max_depth': -1, 'metric': 'binary_error', 'num_leaves': 40, 'objective': 'binary'}

For ExtraTreesClassifier:

1. n_estimators = [50, 100, 200]
2. max_depth = [None, 10, 20, 30]
3. min_samples_split = [2, 5, 10]
4. min_samples_leaf = [1, 2, 4]
5. bootstrap = [True, False]
6. criterion = ['gini', 'entropy']

Best parameters: {'bootstrap': False, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 200}

6. Results and metrics

For each model there were 2 training and validation rounds one with the PCA approach for the data and the other one with the VarianceThreshold they will be displayed as plots and in the end, I'll attach the accuracy and cross-validation score obtained as well as a confusion matrix of the results:

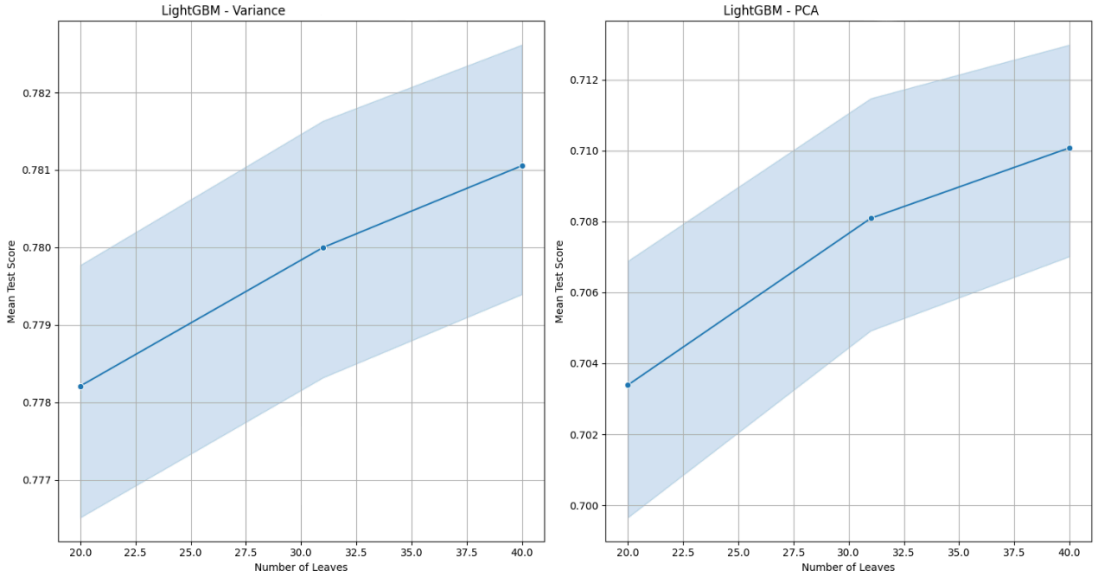
LGBMClassifier - VT:
Best cross-validation score: 0.792 ----- Validation Accuracy: 0.7523302263648469

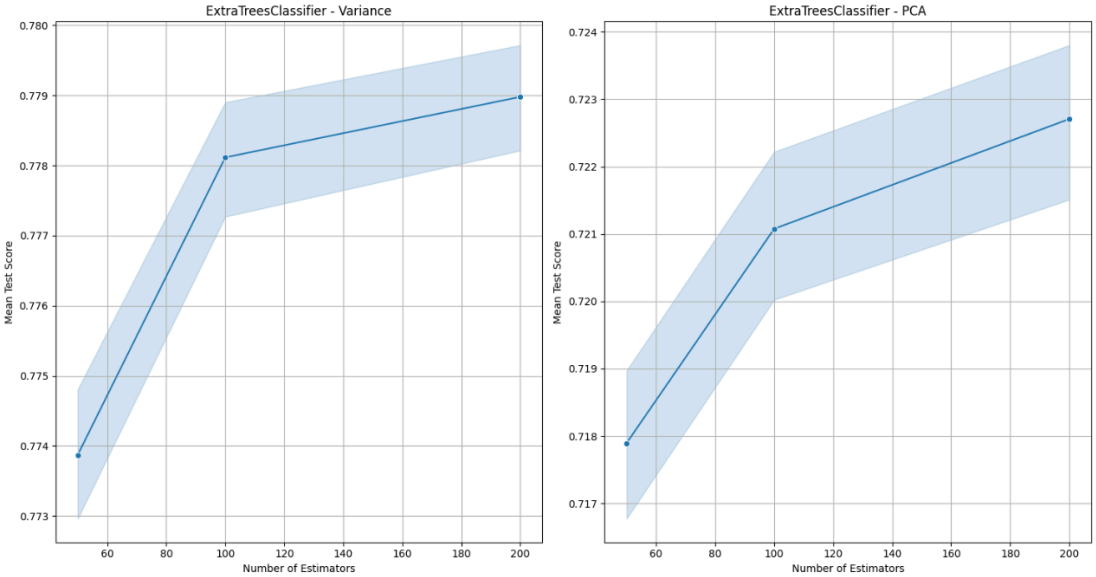
LGBMClassifier - PCA:
Best cross-validation score: 0.727 ----- Validation Accuracy: 0.7163781624500666

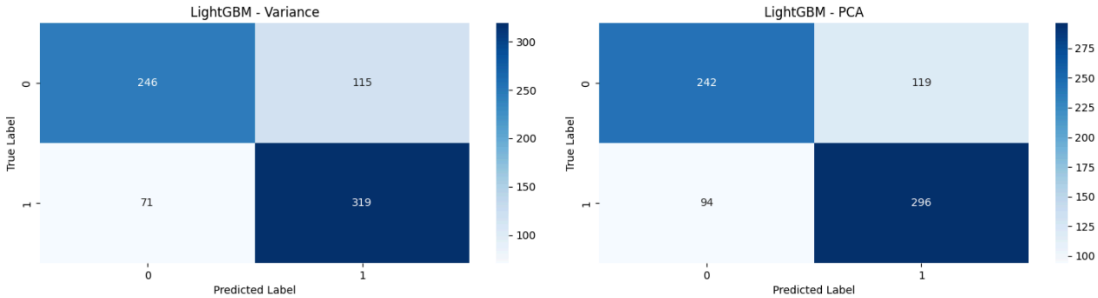
ExtraTreesClassifier - VT:
Best cross-validation score: 0.792 ----- Validation Accuracy: 0.7656458055925432

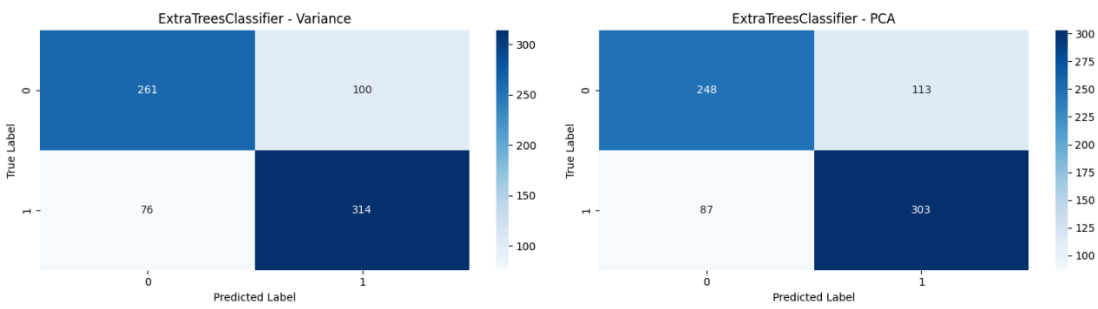
ExtraTreesClassifier - PCA:
Best cross-validation score: 0.742 ----- Validation Accuracy: 0.7336884154460719

Below are the plots:


LightGBM - Variance
Mean Test Score
Number of Leaves
LightGBM - PCA
Mean Test Score
Number of Leaves


ExtraTreesClassifier - Variance
Mean Test Score
Number of Estimators
ExtraTreesClassifier - PCA
Mean Test Score
Number of Estimators


LightGBM - Variance
True Label
Predicted Label
246
115
71
319
LightGBM - PCA
True Label
Predicted Label
242
119
94
296


ExtraTreesClassifier - Variance
True Label
Predicted Label
261
100
76
314
ExtraTreesClassifier - PCA
True Label
Predicted Label
248
113
87
303