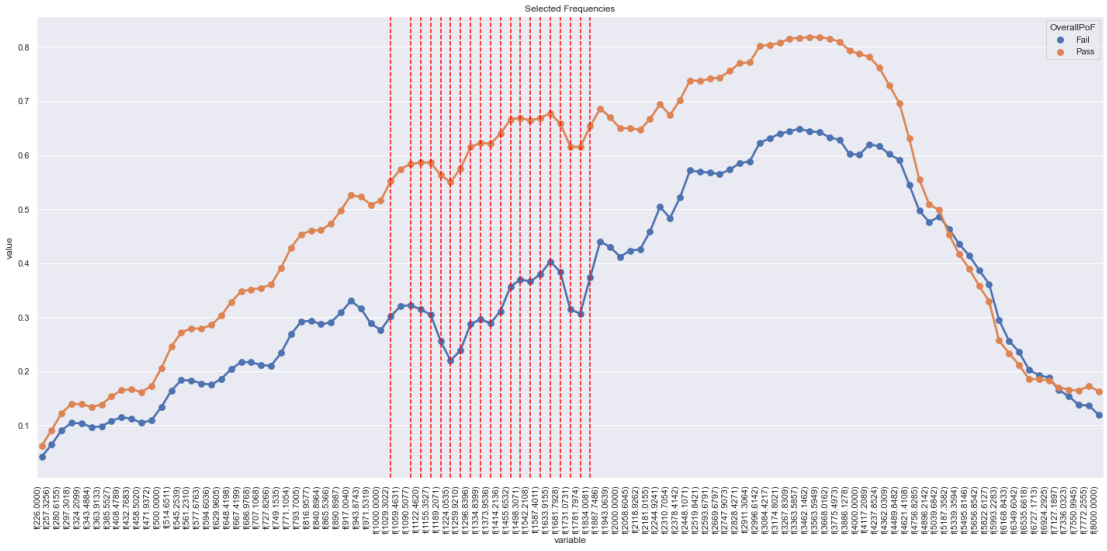
Feature Selection

As we are given features that may affect the result, to know which features or frequencies affect the prediction of the model more, in other words, whether it's possible to obtain a more accurate prediction with only some of the features out of so many, reducing data collection or input. Next, I tried four feature selection methods.

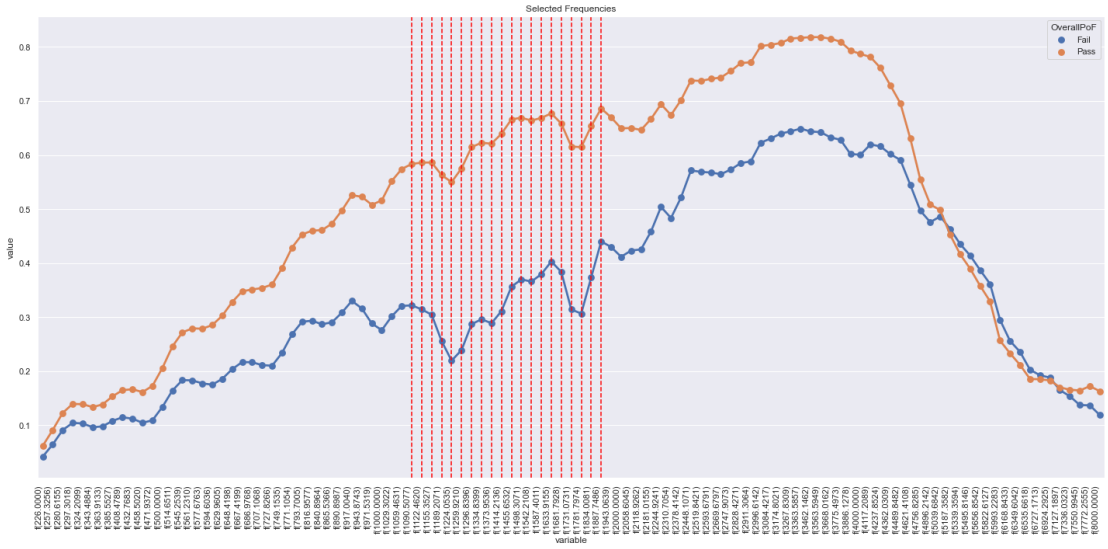
M1: Univariate Selection

This is a statistical method that uses the chi-squared statistical test to select 20 of the best features from the Frequency. Univariate Selection calculates a statistical indicator for each variable separately, based on which indicators are judged to be important and those that are not eliminated. The results show the ranking of the top 20 most relevant frequencies. It is clear from the plot presentation that these frequencies are concentrated between 1029.3022 and 1731.0731.



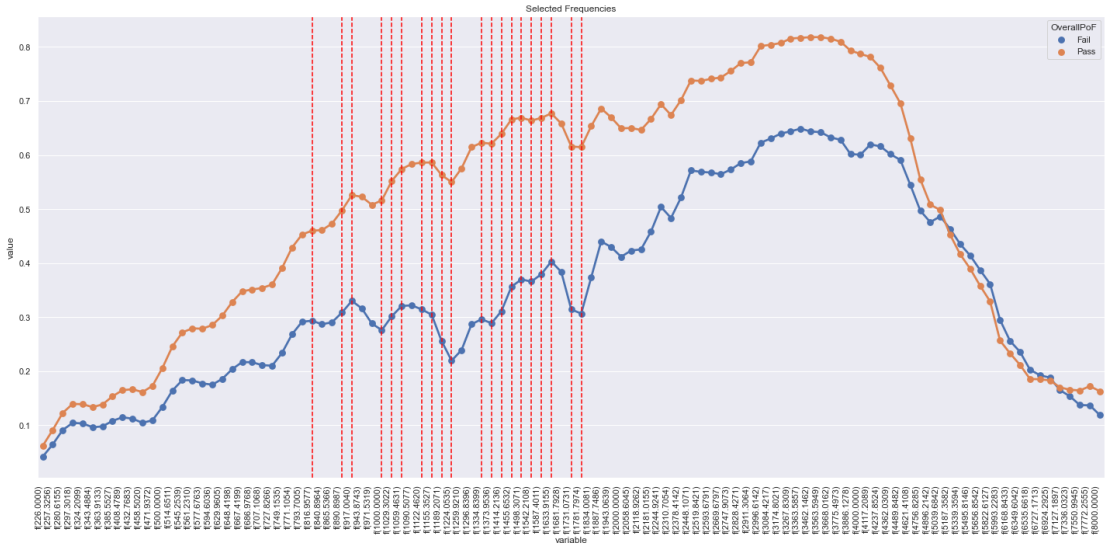
M2: Feature Selection Using the F-Test

The result is very similar to method one. The methods based on the F-test estimate the degree of linear dependency between two random variables. The following graph shows the distribution of the top 20 frequencies most relevant to the response variable using the two statistical methods, which show all the selected frequencies next to each other, which do not represent well the absorbance readings for 107 different frequencies, ranging from 226 Hz to 8000 Hz.



M3: Feature Importance based on Tree

The use of random forests for feature selection is due to the nature of decision trees, where the average impurity (gini) decay obtained from all decision trees is used to quantify the importance of features[ref]. From the figure below the selection of frequencies becomes looser, but still dense in the overall sense. Following this idea, we can change the integration method and still try again for the tree-based feature selection.

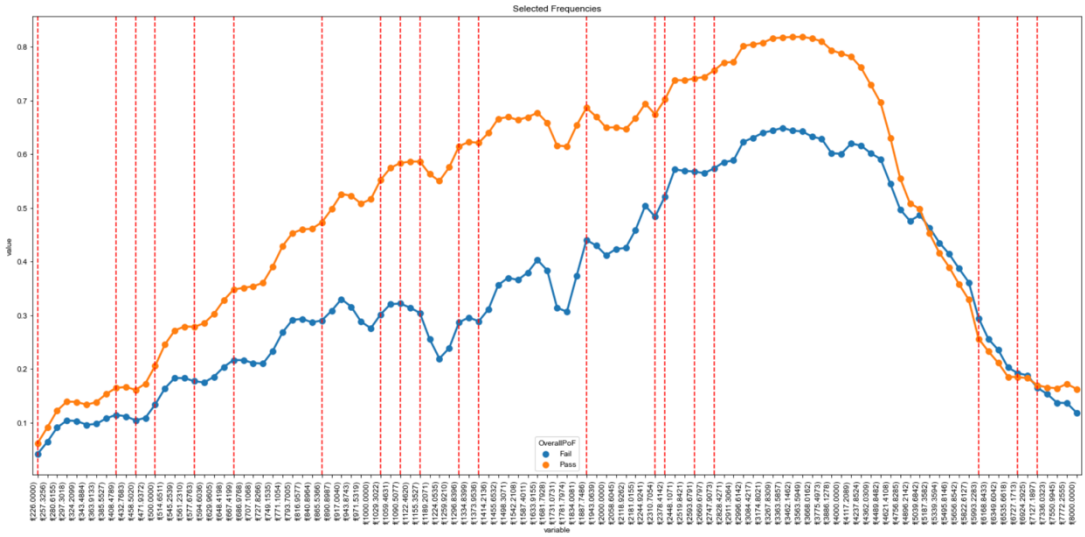


M4: Feature Importance based on XGBoost

For ensemble learning, the gradient boosting decision tree XGBoost uses regularised learning and cache-aware block structure tree learning. by the typical profit. The average gain is calculated by dividing the cumulative gain of all trees by the cumulative number of splits for each feature. The more significant and useful the relevant feature is, the higher the feature importance score of

XGBoost is. The trained XGBoost model automatically calculates the importance of features for the predictive modelling problem. From the image, the frequencies we picked were far apart from each other, and the conclusion is more representative, so I prefer this feature selection method.

Very similar to method 3, the importance of an individual decision tree is calculated by the amount of each attribute split point improvement performance measure, weighted by the number of observations the node is responsible for. The performance measure can be the purity (Gini index) used to select the split points or another more specific error function. The feature importance is then averaged over all decision trees within the model.
The trained XGBoost model automatically calculates the feature importance for the predictive modelling problem.



Show all the selected frequencies next to each other extend to xgboost the frequencies we picked were far apart from each other. As you can clearly see from the plots, the results are selected with a wider and more representative frequency distribution, so I prefer this feature selection method.

Next step:

This section shows the tree-based model and the three boost models that I trained previously with all frequencies, but the results obtained were all over 90%, which is incorrect because there is class-imbalance in our dataset, so in the next step I will train again with the dataset generated by VAE and combine it with the results obtained from feature selection.

RF and XGBoost are both classifiers that use multiple trees to train and predict samples, both based on the idea of Tree ensemble.