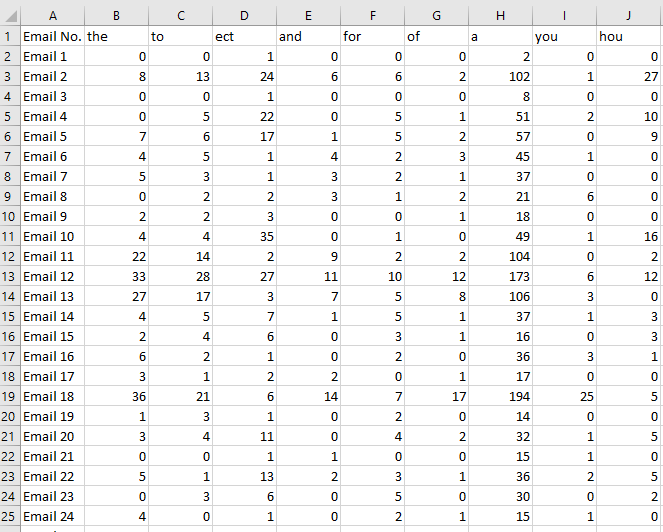
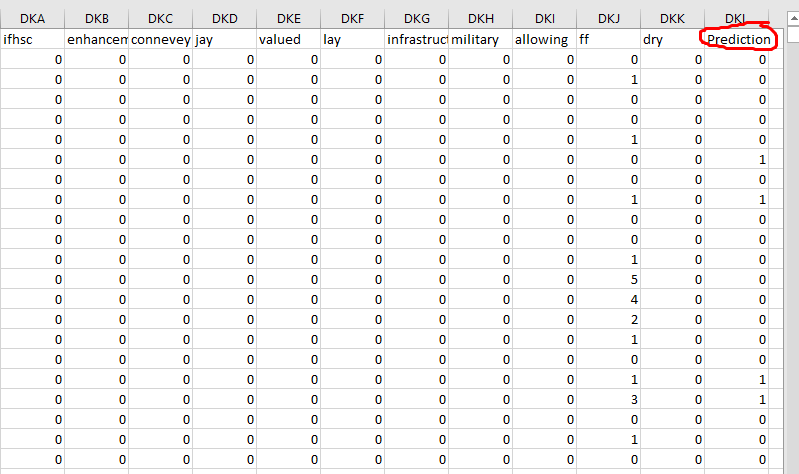
Final project: predict spam emails

By analyzing the words that arrive across most types of fake mail, the presence and frequency of certain words in the mail indicates the likelihood that this mail is fraudulent. This data therefore determines the likelihood that such mail is fraudulent.



From these words:

Politics. the offer. Security. Complimentary. Quick win. And much more.

In the attached data, you choose these messages randomly, and by the presence of words and repetitions it is clear whether the email is spam (1) or not (0).

**SVM algorithm**

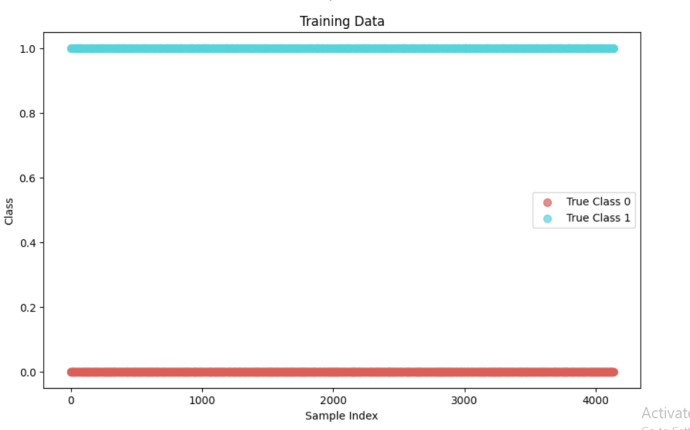
Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. Its primary goal is to find an optimal hyperplane that separates data points from different classes with the maximum margin. This margin allows SVM to be robust to noise and generalize well to unseen data

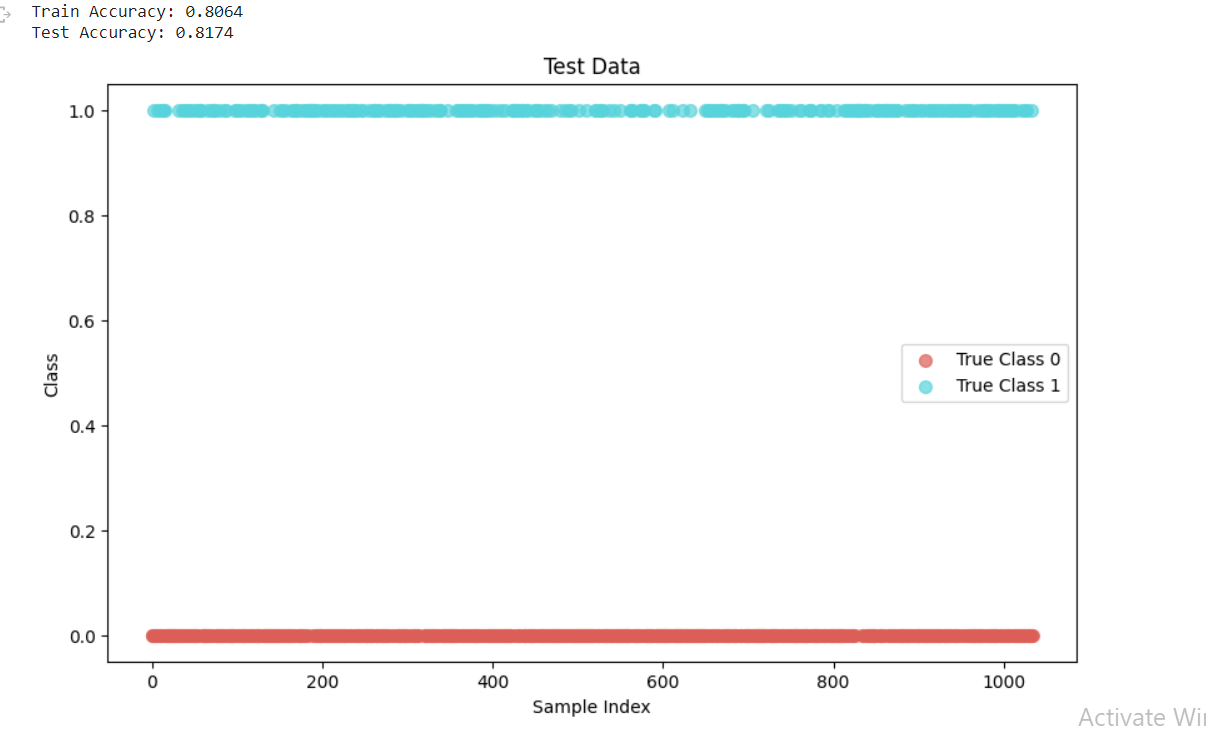
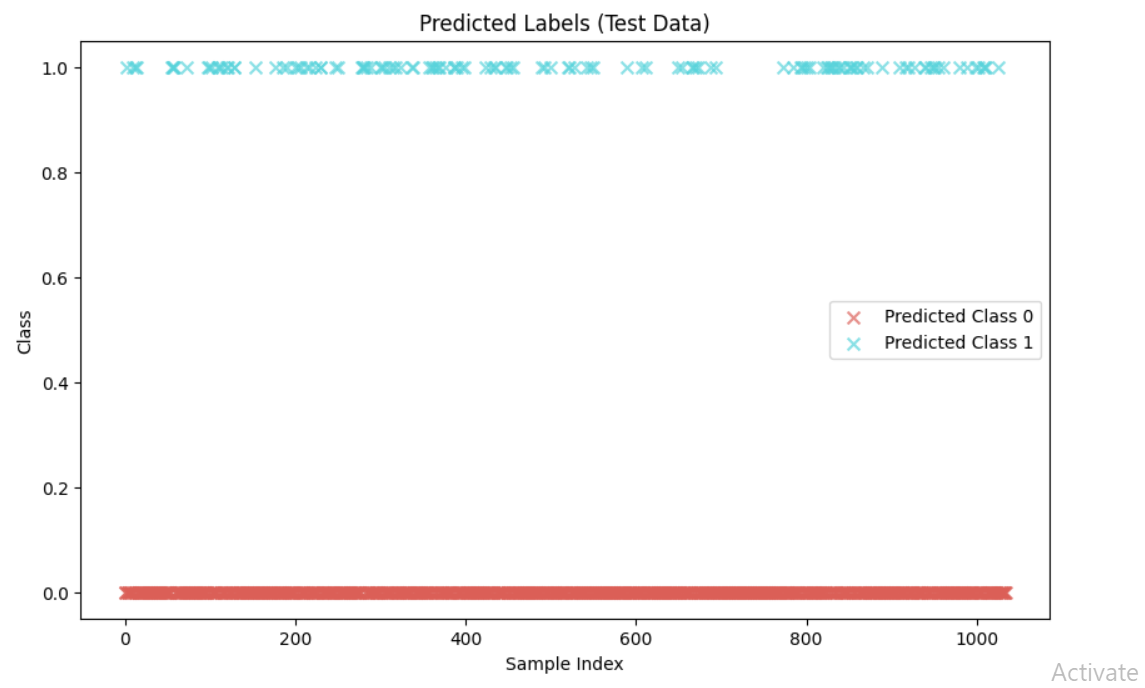
SVM was chosen due to the proximity of the data points in the dataset. In addition, it also effectively handles nonlinear data using the kernel, which transforms the data into a higher-dimensional space for better separation of categories. This capability allows us to work with data that follows nonlinear distributions and improve classification accuracy overall. Thus, using SVM seems appropriate and efficient to handle these closely-clustered data and address nonlinear challenges.

.

We notice that the selected data are close to each other, which may lead in the SVM algorithm to be difficult to draw a hyperplane that clearly separates the categories. So that the different points close to each other are more sensitive, so there must be errors, so that some points will be present in the other group, despite the fact that their final value belongs to the other party.

The accuracy is 0.8064 so this accuracy is not perfect, as there is 0.1936 that is not correct. Certainly, this ratio can be developed through the enemy of things, including: improving the data. Selection of kernel technology and others.





We note that the accuracy increased in the test, and this is due to several things, including:

This data was better than the previous one. Where also the arrangement of the girls may affect the accuracy because we distribute the data among them in a random way.

Although the accuracy increased, it is clear from the drawing that there is a problem in determining class 1, and this may be due

For data distribution: It may be an unbalanced distribution of the data, with more data in category 0 than data in category 1. In this case, you may have high precision for the most common category class 0) and low accuracy for the least common category (class 1).).

True Negatives = 2883,True Negatives (TN) The number of emails that were correctly predicted as "not spam" and which actually were., it means that there were 2883 emails that were correctly predicted and identified as "not spam" and actually belonged to this category.

True Positives (TP) = 453 The classification model successfully identified 453 emails as "spam" and worked correctly in these categories. It is an indicator of the model's ability to correctly identify spam email.

Precision (Class 0): 0.7933

this value indicates the model's ability to correctly identify non-spam email, , 79.33% of non-spam email was correctly identified.

Recall (Class 0): 0.9830

This value indicates the model's ability to correctly identify all non-spam emails, as 98.30% of non-spam emails were correctly identified.

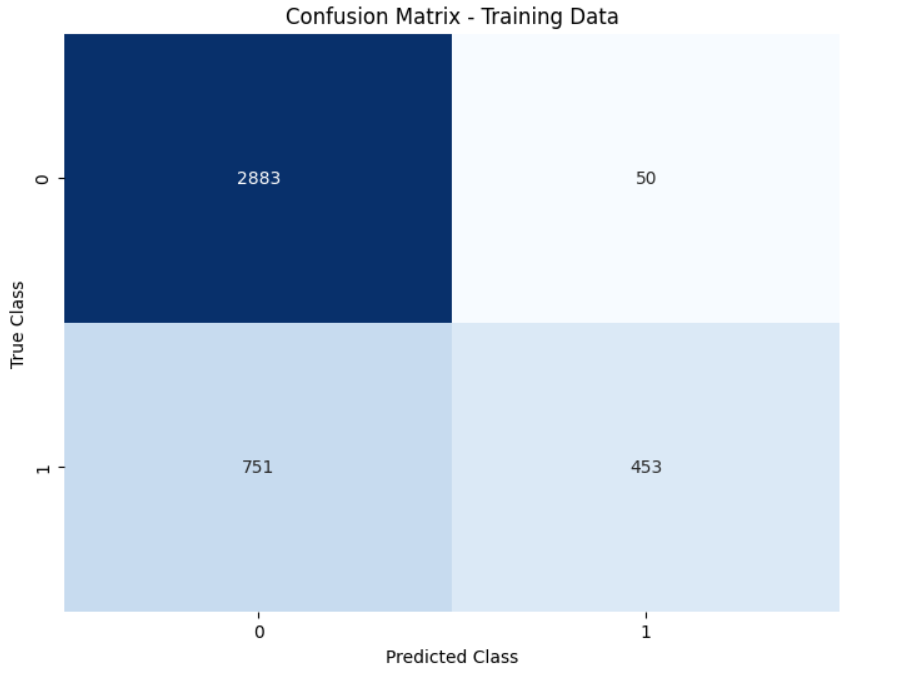
F1-Score (Class 0): 0.8780

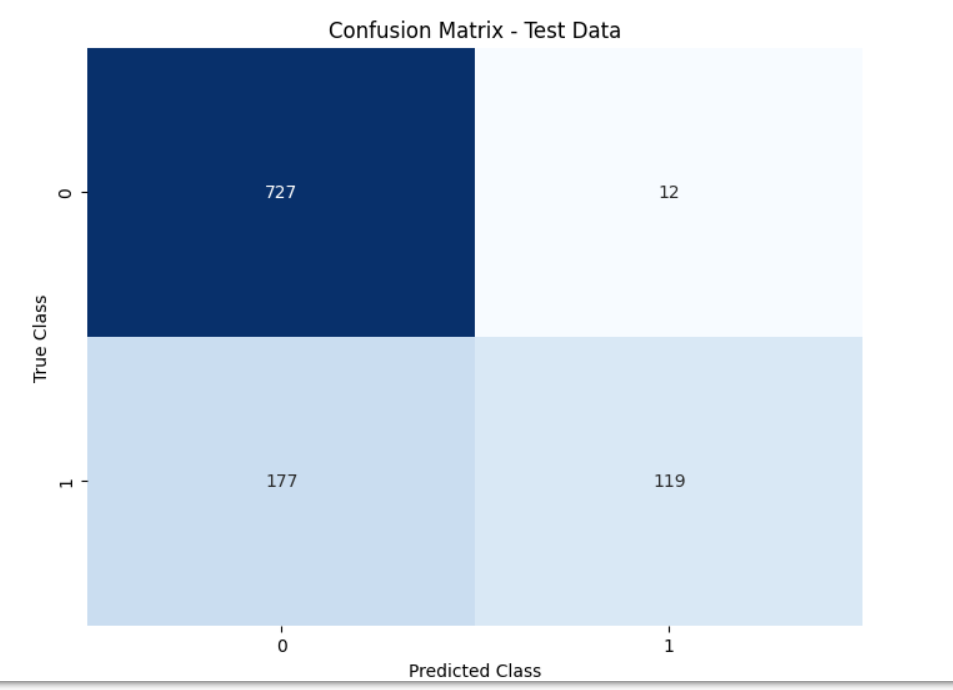
The F1-Score is the geometric mean between Precision and Recall, and it measures the balance between them. The value ranges from 0 to 1, where values close to 1 mean better performance. In our case, this value indicates that the model recognizes non-spam emails well based on accuracy and recall.

Precision (Class 1): 0.9006

Recall (Class 1): 0.3762

F1-Score (Class 1): 0.5308





Precision (Class 0): 0.8042

Recall (Class 0): 0.9838

F1-Score (Class 0): 0.8850

Precision (Class 1): 0.9084

Recall (Class 1): 0.4020

F1-Score (Class 1): 0.5574

Adaboost

adaBoost is a machine learning algorithm used for classification and iterative improvement of classification accuracy. The main goal of adaboost is to train a set of weak learners, such as short decision trees, and then combine them to create a robust model capable of high-accuracy classification.

Therefore, this algorithm mainly focuses on the examples in which there is an error in the previous time in the group given to it, and thus the weight of these examples increases each time, while the value of the correct examples decreases.

But on the other hand, if the focus is only on giving the examples that we get wrong more weight, then this leads to the opposite result, so that with the next values, the results will be wrong, and this is what is called, but this algorithm deals with this problem so that the value of all the wrong examples together is always equal to 0.5, and the sum of the correct examples is also half.

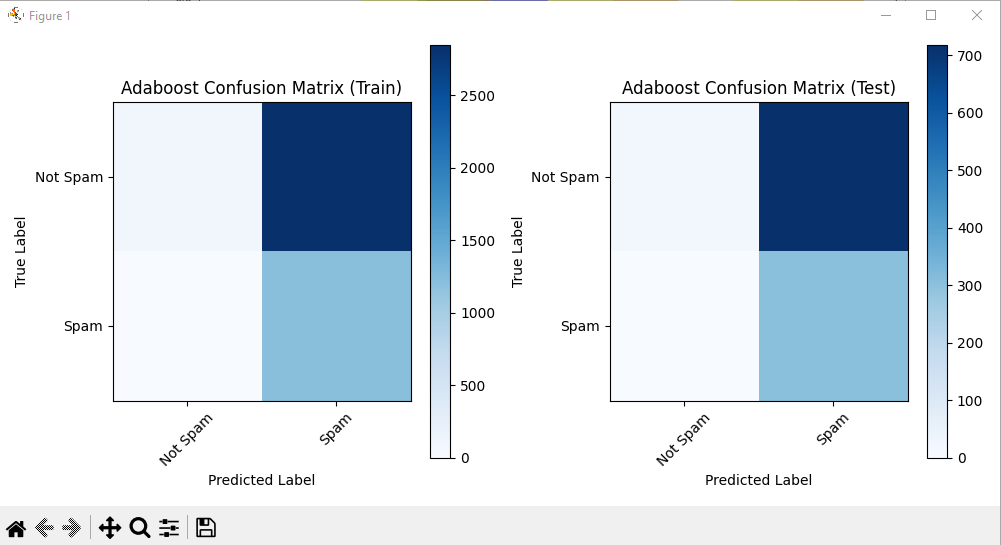
In the adaboost, simple data must be chosen, so that it does not lead to overfitting, which is the excessive increase in the training data, which leads to any distortion of the learning process and thus to a weakness in the results in addition to an excessive increase in the learning time and a high probability of failure of the learning process at times. So always with adaboost you should choose simple data and suitable amounts of data for learning.

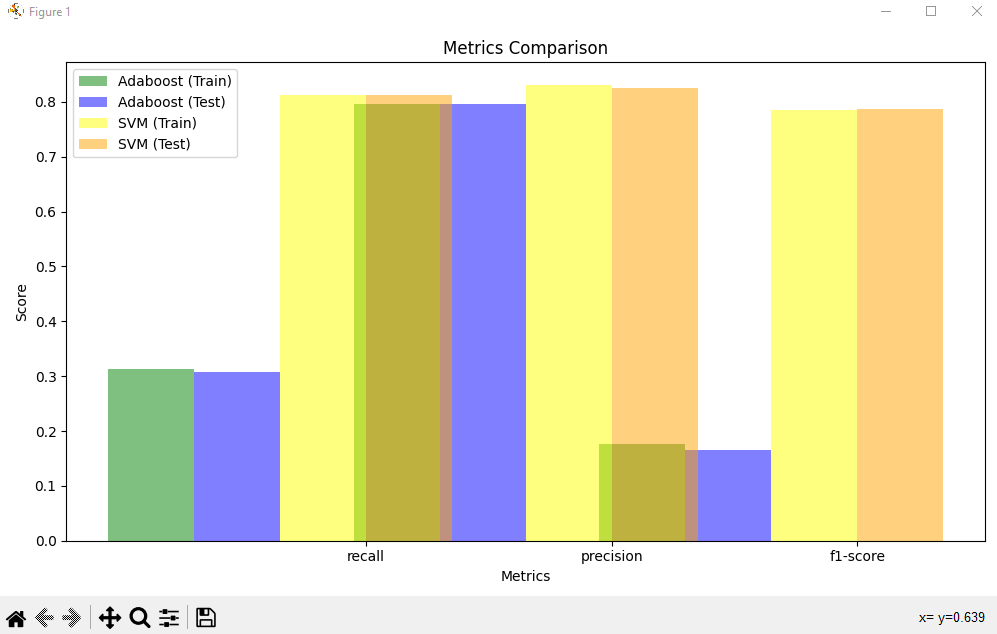
adaboost is a machine learning algorithm used for classification and iterative improvement of classification accuracy. The main goal of adaboost is to train a set of weak learners, such as short decision trees, and then combine them to create a robust model capable of high-accuracy classification.

Therefore, this algorithm mainly focuses on examples in which there is an error in the previous time in the set given to it, so the weight of these examples increases each time, while the value of the correct examples decreases.

But on the other hand, if the focus is only on giving the examples that we get more wrong, this leads to the opposite result, so that with the following values, the results will be wrong, but this algorithm deals with this problem so that the value of all the wrong examples together is always equal to 0.5, and the sum of the correct examples is also half.

In adaboost, simple data must be chosen, so that it does not lead to overfitting, which is the excessive increase in training data, which leads to any distortion in the learning process and thus to weakness in the results in addition to an excessive increase in learning time and a high probability of failure of the learning process at times. Or underfitting, In the case of lack of data, learning will certainly result in a weakness in the learning results as a result of not providing enough data for the machine learning algorithm to be able to learn from the data of the problem to be solved or learn on it, so always with adaboost, you must choose simple data and appropriate amounts of data for learning.



Adaboost is an ensemble learning technique that combines multiple weak learners (in this case, decision trees with a maximum depth of 1) to create a strong learner. The algorithm iteratively assigns higher weights to misclassified samples, allowing subsequent weak learners to focus on the previously misclassified instances. By doing so, Adaboost aims to improve the overall model performance and generalization. Through the confusion matrix we can notice the efficiently of the model to modify if the email is spam or not.

Recall= (TP+FN)/TP

Precision= TP+FP/TP

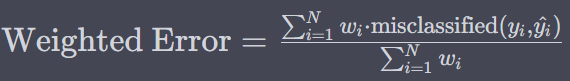
F1-score=2× Precision+Recall/ Precision×Recall

​

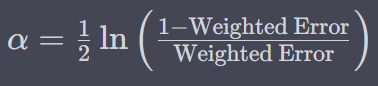
​

Weighted Error:

The weighted error of a weak learner (e.g., decision tree) on a set of samples is calculated as the sum of weights of misclassified samples divided by the total weight of all samples:



Alpha (Weight of the Weak Learner):

The alpha value represents the importance of a weak learner's prediction in the final boosted classifier. It is computed based on the weighted error of the weak learner