**Air Quality Data**

|  |  |  |
| --- | --- | --- |
| **Group Number: 07** | | |
| Last name | First name | Matriculation Number |
| Derler | Anna | 12114827 |
| Fankhauser | Sebastian |  |
| Wallner | Selina | 12111842 |

**Introduction**

In our project we used the Air Quality Dataset we got from the UCI Machine Learning Repository. The Dataset contains the responses of an array of 5 metal oxide chemical sensors inside an Air Quality Chemical Multisensor Device which is deployed in a significantly polluted area in an Italian city. The records contain the hourly averaged responses and the gas concentrations references.

We selected this Dataset, because we are interested in the environment and its pollution.

**Code Design**

The classes, methods, and properties in the provided code are designed to be modular, reusable, clear, and easy to maintain.

The first class “DatasetPreprocessor” handles data preprocessing tasks to make sure the data is clean, formatted, and ready for analysis. This class keeps all preprocessing tasks in one place, making the code more organized, which makes it easy to update the preprocessing steps without affecting other parts of the code. The use of private methods (like \_preprocess\_data) hides the internal workings from the user, enforcing encapsulation.

The second class “SimpleBaselineClassifier” provides a simple classifier to compare against more complex models, using common strategies like "most frequent," "uniform," and "constant.".

The third class “Classifier“ manages the training and evaluation of different classifiers on the dataset. It offers an interface to train, evaluate, and compare various machine learning models. By providing specific methods for each classifier, it ensures clarity and separation of responsibilities.

If you were given another two months of time to redesign your code, what would you change?

**Experimental Setup**

We used the RidgeClassifier, the PassiveAggressive Classifier, the DecisionTree Classifier, the SGD Classifier and the Simplebaseline Classifier.

From the Dataset we used the CO(GT), C6H6(GT), NMHC(GT), NOx(GT) and the NO2(GT) features.

20% were used for testing and 80% were used for training.

To handle the missing values, we set the values < 0 to 0.0.

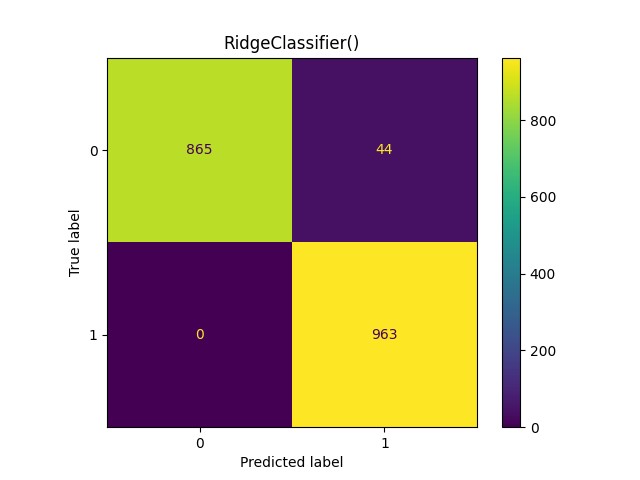
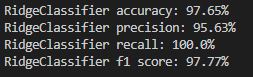
Which metrics did you rely on for evaluation and why? Accuracy, Precision, Recall, F1 Score

Our baseline was the SimplebaselineClassifier with the strategy “uniform”.

**Results and Discussion**

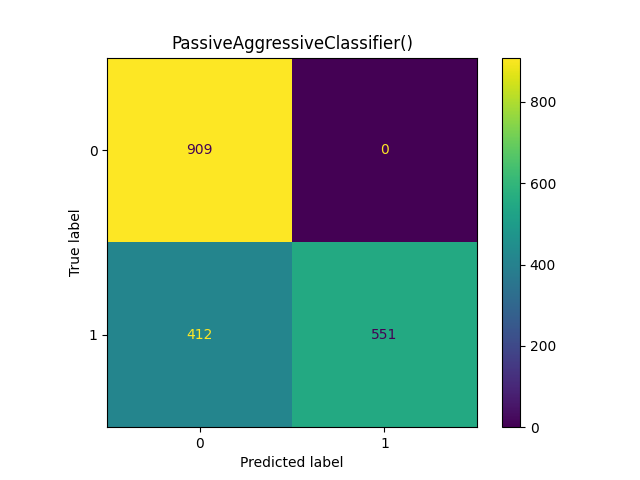
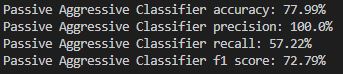
The best Classifier were the Decision Tree Classifier and the SGD Classifer. The worst Classifier was the Simple Baseline Classifier. Does this change if you use a different metric for evaluation?

In Plot 1 you can see the Confusion Matrix of the Ridge Classifier. For example, we can see the that in 963 instances the true label was 1 and the model also predicted 1 or that in 0 instances the true label was 1 and that the model predicted 0. So, this means that the Classifier is performing well.



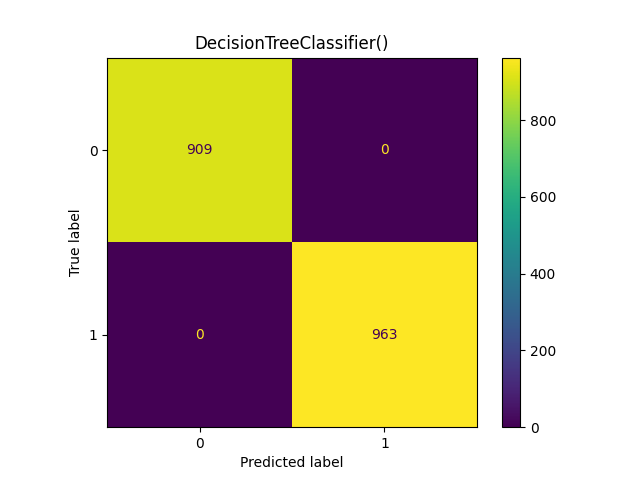
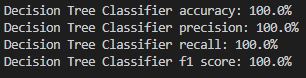
Plot 1: Ridge Classifier Terminal Output 1: Ridge Classifier

In Plot 2 you can see the Confusion Matrix of the Passive Aggressive Classifier. For example, we can see a high number of instances for True negatives, which is good. But the number of False negatives is also high. This could mean that the Classifier tries to get as few False positives as possible but for that he gets more False negatives.



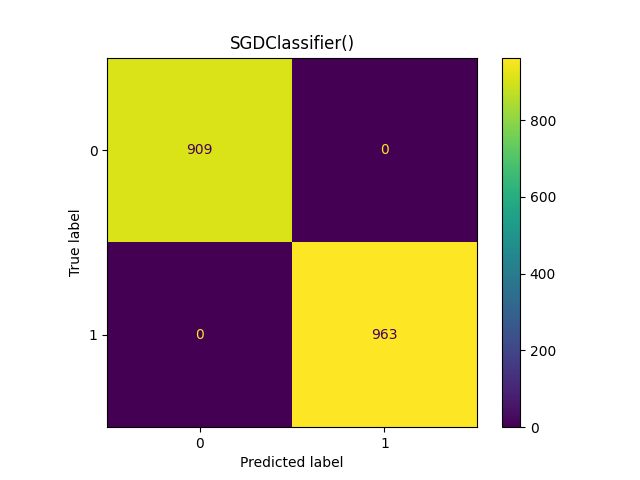
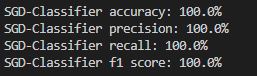
Plot 2: Passive Aggressive Classifier Terminal Output 2: PassiveAggressiveClassifier

In Plot 3 you can see the Confusion Matrix of the Decision Tree Classifier. We can see that there are only instances for True negatives and True positives. So, this means that the Classifier performed perfectly.



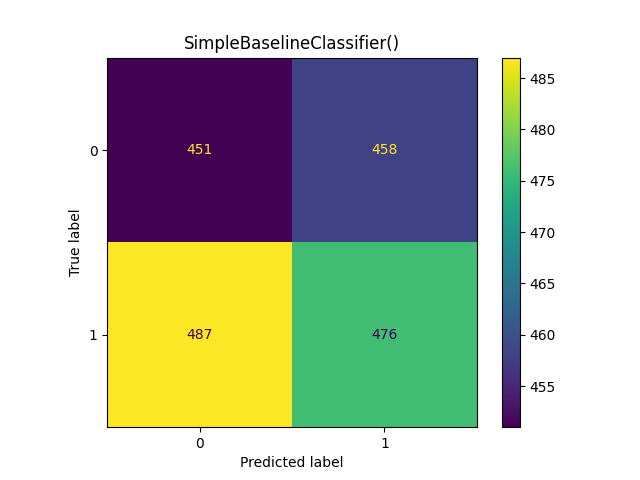
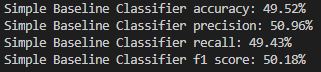
Plot 3: Decision Tree Classifier Terminal Output 3: Decision Tree Classifier

In Plot 4 you can see the Confusion Matrix of the SGD Classifier. We can see that there are only instances for True negatives and True positives. So, this means that the Classifier performed perfectly.



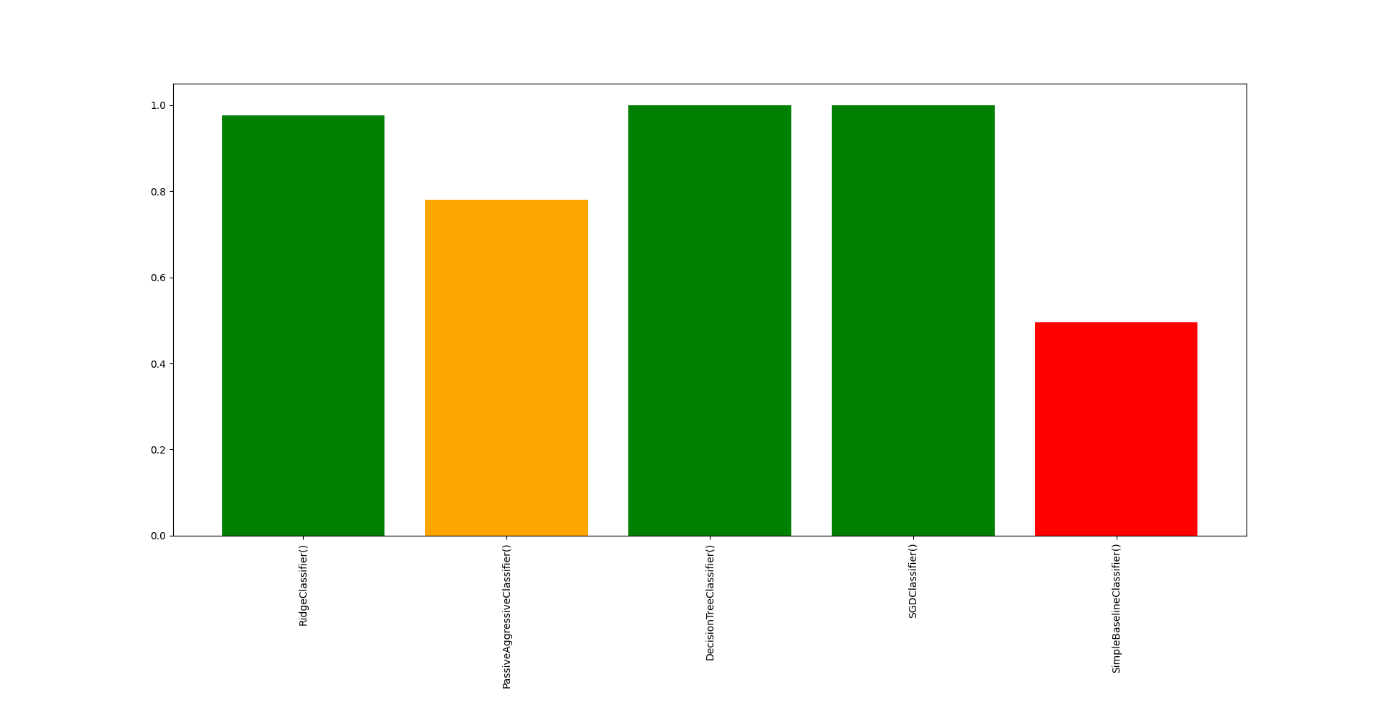
Plot 4: SGD Classifier Terminal Output 4: SGD Classifier

In Plot 5 you can see the Confusion Matrix of the Simple Baseline Classifier. We can see that instances for True negatives, False negatives, True positives and False positives are nearly equal. This means that the Classifier is not working very well, because we have a lot of False positives and negatives.



Plot 5: Simple Baseline Classifier Terminal Output 5: Simple Baseline Classifier

In Plot 6 you can see a comparison of the five different Classifiers. As you can see all other Classifiers beat the Baseline Classifier.



Plot 6: Comparison of the Classifiers

**Conclusion**