

**Fachgebiet für Künstliche Intelligenz im Maschinenbau**

***Bachelor/Master Thesis***

**Titel der**

**Abschlussarbeit**

**Name:** Liwash Gurung

**Matrikelnummer:** 29367322

**Studiengang:** Fahrzeugmechatronik

**Anschrift:** Prinz Eugen Straße 4, 80804

**Betreuer:**  falls es einen gibt

**Hochschullehrer:** Prof. Dr.-Ing. Marcin Hinz

**Ausgabe:** xx.xx.xxxx

**Abgabe:** yy.yy.yyyy

**Inhaltsverzeichnis**

**1** **Einleitung 1**

1.1 Machine learning 1

**2** **Zielsetzung 2**

**3** **Aufbau der Arbeit 3**

3.1 Vorgehen bei der Erstellung einer Arbeit 3

3.2 Äußerlicher Aufbau 3

3.2.1 Titelblatt 3

3.2.2 Erklärung 4

3.2.3 Vorwort und Danksagung 4

3.3 Inhaltlicher Aufbau 4

**4** **Format 6**

4.1 Seitenformat 6

4.2 Textfluss 6

4.3 Inhaltsverzeichnis 7

4.4 Formelzeichen 7

4.5 Überschriften 7

4.6 Aufzählungen 7

4.7 Kopfzeilen und Seitenzahlen 7

4.8 Fußnoten 8

4.9 Bilder 8

4.10 Tabellen 9

4.11 Formeldarstellung 9

4.12 Literaturverzeichnis 9

4.13 Anhang 9

**5** **Inhalte 10**

5.1 Verwendung von Begriffen 10

5.2 Stand der Technik 10

5.3 Zielsetzung und Zusammenfassung 10

5.4 Dokumentation 10

5.5 Darstellung und Diskussion von Ergebnissen 11

**6** **Zusammenfassung 12**

**Literaturverzichnis I**

# Supervised Learning

This section covers the use of supervised learning algorithms to classify the raw data into three classes. Supervised learning is a type of machine learning in which an algorithm learns from labeled data. The algorithms discussed in this section are as follows:

## Random Forest Classifier(RFC)

RFC is a ensemble Method, which builds many Decision Trees and combines their Prediction through majority Voting for Classification. This section discusses the rationale behind the methods and parameter optimization to improve prediction accuracy. Since the raw data does not require feature engineering, a random variable with randomly generated values is added to the dataset. This variable serves as a benchmark: features that are less important than the random variable can be considered irrelevant and removed, thereby reducing prediction errors, as illustrated in Figure 1 below.

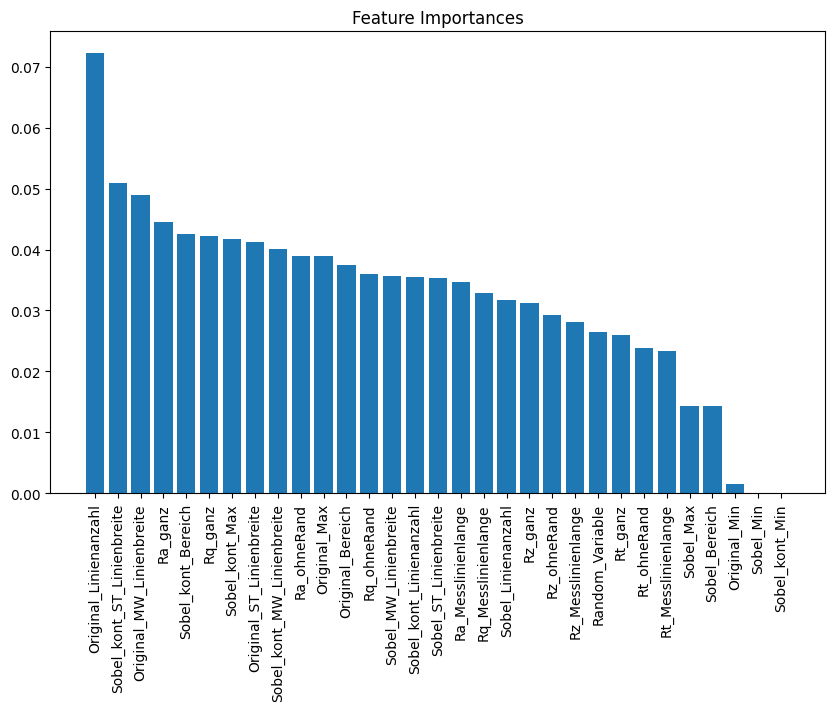


Figure 1 Feature Importance with the to be filtered Features highlighted with the red Rectangle

The filtered dataframe will be used for all algorithms considered in the supervised learning part of the project. In addition, selected parameters will be optimized using GridSearch, as shown in Figure 2 below.



Figure 2 Code Snippet of the Parameter Optimization of RFC Model

A predefined search region is specified for the parameters, and GridSearch is used to identify the optimal combination of all variables for the best possible model. The parameter “n\_estimators” determines the number of decision trees, while “max\_depth” sets the maximum depth of each tree, helping to control overfitting as well as tuning for Prepruning. The parameter “ccp\_alpha“ governs cost-complexity pruning, which can also reduce overfitting. Optimizing these hyperparameters plays a crucial role in the accuracy of tree-based models and will be performed for this project. The resulting hyperparameter optimization has led to the following model, as shown in the Figure 3 below.



Figure 3 RFC Model with the optimized Hyperparameters

From Figure 4 and Figure 5, it can be seen that the model with optimized parameters on the right performs better than the model with default parameter values on the left. The selected metrics, balanced accuracy and F1-macro, give a better understanding of model performance because the classes are very unevenly distributed. This imbalance is shown in the confusion matrices below. The optimized model shows improved metric scores compared to the model with default values.

|  |  |
| --- | --- |
| Balanced Accuracy : 0.739341063  F1 Macro : 0.758143938 | Balanced Accuracy : 0.751913399  F1 Macro : 0.769903309 |
| Figure 4 Confusion Matrix of the RFC Model with Default Parameters and its Metric scores | Figure 5 Confusion Matrix of the RFC Model with optimized Hyperparameters and its Metric scores |

The Performance of RFC Models will be compared with Models based on Decision Tree and Support Vector Classifier, which will be the Topic in the upcoming subsections.

## Decision Tree(DT)

DT is a supervised machine learning algorithm that models the relationship between input features and the target outcome using a logical, tree-like structure. It can be applied to both discrete and continuous variables. This model forms the basis for the machine learning approach in this section, where the methods for optimizing the model are explained. GridSearch will be used to optimize the model’s parameters, as described in 1.1 and shown in Figure 6.

## 

Figure 6 Code Snippet of the Parameter Optimization of DT Model

“max\_depth”, “min\_samples\_split”, “min\_samples\_leaf”, and “max\_features” are parameters that can be tuned to control overfitting and improve model accuracy. ccp\_alpha is also considered to reduce overfitting through post-pruning. As a result of parameter optimization, the model achieves the following optimized parameter values, as shown in Figure 7 below.



Figure 7 DT Model with the optimized Hyperparameters

From the confusion matrix and the metric scores (balanced accuracy and F1 score) in Figure 8 and Figure 9, the optimized model on the right performs better than the default model on the left. However, compared to the optimized RFC model discussed in 1.1, the performance of this optimized model is much worse for the given dataset.

|  |  |
| --- | --- |
| Balanced Accuracy : 0.659405962  F1 Macro : 0.665155226 | Balanced Accuracy : 0.661870142  F1 Macro : 0.672413671 |
| Ein Bild, das Text, Screenshot, Diagramm, Rechteck enthält.  KI-generierte Inhalte können fehlerhaft sein.  Figure 8 Confusion Matrix of the DT Model with Default Parameters and its Metric scores | Figure 9 Confusion Matrix of the DT Model with optimized Parameters and its Metric scores |

## Support Vector Classifier(SVC)

SVC is a powerful and popular Supervised Machine learning Algorithm, which tries to find the optimal Hyperplane which separates the data points of different classes. Since SVC is not a tree based Algorithm, which is why the data needs to be scaled before the Model can work with it. Along with optimization of the Hyperparameters of SVC, the different Types of the Scalers are also going to looked at to find the best possible combination for the model to get the best possible Accuracy for the Predictions. Gridsearch is going to be used to find the optimal Combination of the Parameters of the Model, as shown in Figure 10 below.



Figure 10 Code Snippet of the Parameter Optimization of SVC Model

“svc\_kernel” determines whether the SVC uses a linear or nonlinear hyperplane. A linear kernel restricts the model to linear separation, while nonlinear kernels (e.g., RBF, polynomial) use the kernel trick to map data into higher dimensions. Kernel-specific parameters such as gamma and degree are optimized separately. The cost parameter “C” controls the trade-off between margin width and classification errors. After optimization, the model achieves the parameter values shown in Figure 11 below.



Figure 11 SVC Model with the optimized Hyperparameters

As mentioned before, scaling the data is necessary for the SVC to work properly, which makes the choice of the right scaler very important. To determine the optimal scaler and parameters, a sensitivity analysis is performed. The data is scaled using different scalers and evaluated with the SVC model from Figure 11. This comparison of scaler–model combinations allows us to identify the optimal scaler for our use case. From Table 1, the StandardScaler performs best according to the metrics “Balanced Accuracy”,”F1 Macro”, and “Weighted Average”, which is why it will be used from here on.

Table 1 Sensitivity Analyse of the different Scalers

|  |  |  |  |
| --- | --- | --- | --- |
| **Model with Scaler Types** | **Balanced Accuracy** | **F1 Macro** | **Weighted Avg** |
| SVC Standardscaler | 0.719478201 | 0.741399426 | 0.746985174 |
| SVC Normalizer l2 norm Default | 0.443368424 | 0.442377059 | 0.535444229 |
| SVC Normalizer l1 norm | 0.505394791 | 0.525781456 | 0.596596907 |
| SVC Normalizer max nomr | 0.521330487 | 0.548178441 | 0.609113536 |
| SVC Robustscaler Quantile Range 60 | 0.673397495 | 0.691622576 | 0.70314648 |
| SVC Robustscaler Quantile Range 75 Default | 0.708927932 | 0.730586364 | 0.735840777 |
| SVC Robustscaler Quantile Range 90 | 0.660526736 | 0.691673841 | 0.705933516 |
| SVC Robustscaler Quantile Range 70 | 0.711762895 | 0.731607951 | 0.734829102 |

With all the optimizations applied, the metric scores and confusion matrix can be compared between the model with default parameter values and the optimized model. As shown in Figure 12 and Figure 13 , the optimized model on the right performs better than the model with default parameters on the left .

|  |  |
| --- | --- |
| Balanced Accuracy : 0.616025096  F1 Macro : 0.650039044 | Balanced Accuracy : 0.719478201  F1 Macro : 0.741399426 |
| Figure 12 Confusion Matrix of the SVC Model with Default Parameters and its Metric scores | Figure 13 Confusion Matrix of the SVC Model with optimized Parameters and its Metric scores |

In comparison to the RFC model discussed in 1.1, however, the performance of this model is worse. It does perform better than the DT model discussed in 1.2. Overall, we can conclude that the optimized RFC model is the best-performing among all the optimized models.