1 Methodology

For machine learning analysis there are some phases to be accomplished. The methods used for processing these stages are explained in this section. The first step to be done is the Data Preprocessing, wich is described in detail in section 3.1. After that the Features Selection phase gets explained in chapter 3.2. Thirdly the Machine learning models needs to be trained. This phase is explained in more depth in chapter 3.3. The final step is to evaluate the performance of the machine learning models in their application to the data set, which is declared in section 3.4.

1.1 Data Preprocessing

Data sets consisting of the real data can be partially sparse, corrupted, incomplete or noisy. The probability for this increases additionally if the data originate from various sources. Because of this data preprocessing is a common task when it comes to training a machine learning model. Data preprocessing is a task to clean up the dataset for the machine learning model to make it easier to parse the data to the model. Machine learning algorithms often fail to identify patterns in the data and do not give quality results if the dataset is inconsistent or noisy. So the quality of predictions, wich are made by machine learning models depend on the quality of the data. [**Pragati˙Preprocessing:2022**] [**kotsiantis2006data**]
The three main problems of datasets for machine learning are the following [**Pragati˙Preprocessing:2022**]:

- Missing Data
- Noisy Data
- Inconsistent Data

At first to keep the values in the dataset, for missing data the two options are ignoring them or fill them manually or with a computed value **Pragati˙Preprocessing:2022 kotsiantis2006data**. In the context of this work, the samples with missing information were mainly ignored and Manuel excluded from the dataset. However, most of the missing information has been excluded in the case of this work by removing the corresponding data columns from the set, as explained in detail in section 4.1.5 Data Preperation.
To handle the problem of noisy data and reduce the number of possible values in total, the features can be discretized. This can for example be done by calculating the maximum and the minimum for the feature and dividing it into k equal sized segments. [**kotsiantis2006data**]
According to Pragati Baheti, the only option for inconsistent data is to remove it from the data set **Pragati˙Preprocessing:2022**. This is also the method done in the context of this master thesis.
After the data is free of noisy, missing and inconsistent data, the next step of data preprocessing is the data normalization. It is a scaling down process to lower the standard deviation of the features values. [**kotsiantis2006data**]
The scaler algorithmus StandardScaler(), wich is used in this purpose, is included in the library sklearn.preprocssing. The algorithm standardizes the values by subtracting the mean and smoothing the values to unit size. [**Sklearn˙StandardScaler:2022**] This is calculated for a sample x

as follows:

$z = \frac{(x-u)}{s}$ **Sklearn˙StandardScaler:2022**

Where u is the mean value and s represents the standard deviation of the samples. The centering and scaling processes are happening independently for each features. The mean and standard deviation statistics are calculated on the samples for this process. [**Sklearn˙StandardScaler:2022**]

For most machine learning estimators it is required to use such a standartisation algorithm, because they do not behave well if the features are not standard normally distributed. [**Sklearn˙StandardScaler:2**

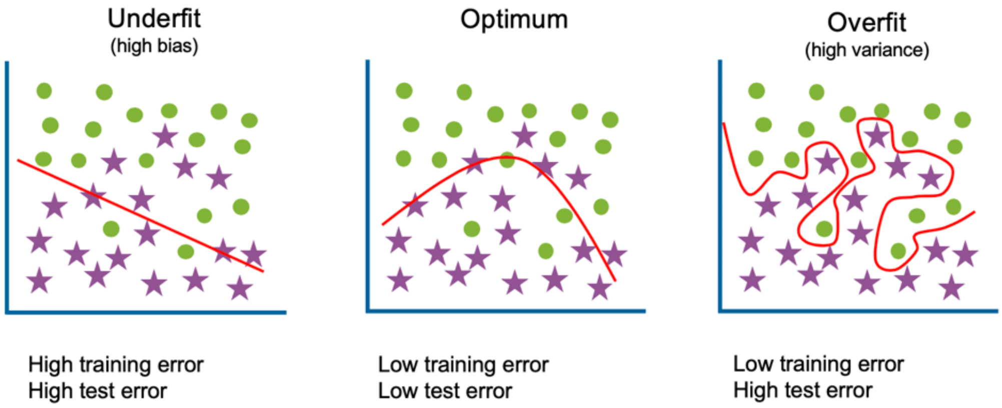
The sklearn documentation of the StandardScaler algorithm describes as an example the Support Vector Machine kernel ”RBF”, which assumes that the values of all features contained in the dataset are centered around zero. If this is not true for a feature and its values are larger, it dominates the dataset by its overweighting and the machine learning model cannot learn correctly in this case. [**Sklearn˙StandardScaler:2022**] As mentioned in the article ”Data preprocessing for supervised leaning”**kotsiantis2006data**, The next step is the features selection one, wich is described in detail in the next section. In the case of this master thesis, the normalization part of data preprocessing is done after the feature selection. This is done because the first part in the features selection process of this work contains the creation and inclusion of additional data columns.

1.2 Feature selection

It is a challenging and significant task in the field of data science to create machine learning models from high dimensional data sets. Machine learning research has assumed that too many columns of data lead to a reduction in prediction quality. This phenomenon is caused by the fact that the algorithm recognizes non-existent patterns in the data set due to the amount of features and creates its learning file based on this. These non-existent patterns are learned by the model because it tries to interpret the noise in the data or irrelevant information when the data set is too complex. So the model tries to fit too much to the training data and end up overfitting, wich means it gets good results with the train data, but has a high error on test data. [**ibm-overfitting:2022**]

The opposite can happen if there are too much important features removed from the dataset in terms of the feature selection. In this case the model underfits and it gets an high training error as well as an high test error. [**ibm-overfitting:2022**] Both overfitting and underfitting lead to a bad performance quality of the machine learning model and it is a challenging task to find the optimum between them **SUBASI202091**. The three different variants overfitting, underfitting and the optimum way a model can fit the data is visualized in figure 1. The line drawn through the data points shows how accurately the model has been fitted to the data set. So you can see that in the first visualization, only a straight line was drawn and lots of the points ended up on the wrong side of the line. In the last oft the three figures, which represents overfitting, the line is snaked so that every outliner also is on the right class. In this case the points that are closer to the other class than to their corresponding one are also correctly classified. With such a fitting it is difficult to correctly classify unseen data in border areas. The center illustration shows the optimal case that classifies many points correctly, but makes no exceptions for outliers.

A huge amount of dimensions also increases the computation costs and can reduce the performance in total. The more dimensions a data set has, the more prevalent it will contain redundant, noisy, and unimportant features, wich lead to overfitting and increase the error rate of the learning algorithm. Therefore, it can help to focus on a small subset of really important features. [**CAI201870**] **ALLAM2022329 VenkateshAnuradha:2019** Feature selection is divided into two steps. The first step is to filter the data and reducing the feature space by removing the previously mentioned irrelevant features. In the second step, an optimal subset of features of the remaining data is created using a wrapper. [**CAI201870**] This can be achieved by removing redundant and unimportant data columns to get enhance performance of prediction, scalability and generalization capability in learning efficiency and avoid overfitting. If a feature does not affect the prediction quality of the learning model, it is not important for the prediction **VenkateshAnuradha:2019**. This does not mean that this feature does not contain useful data. It only indicates that it is not statistically related to other features **VenkateshAnuradha:2019**. A good feature Selection can help to get much better predictions from the machine learning models and decrease the error rate **CAI201870**. In most feature selection methods, optimization algorithms are used to build a subset of the most relevant features. This leads to better performance and better classification results. [**ALLAM2022329**] The popular approaches to do this, are models, features quality measures, feature evaluation, search strategies and combinations of these **VenkateshAnuradha:2019**. Depending on how the training set is labeled, supervised (fully labeled), unsupervised (unlabeled) and semi-supervised (partially labeled) feature selection methods are used **CAI201870**. Feature selection methods can also be divided into the three groups Filter, Wrapper and Embedded Method, based on how they interact on the learning models.



Source: [**ibm-overfitting:2022**]

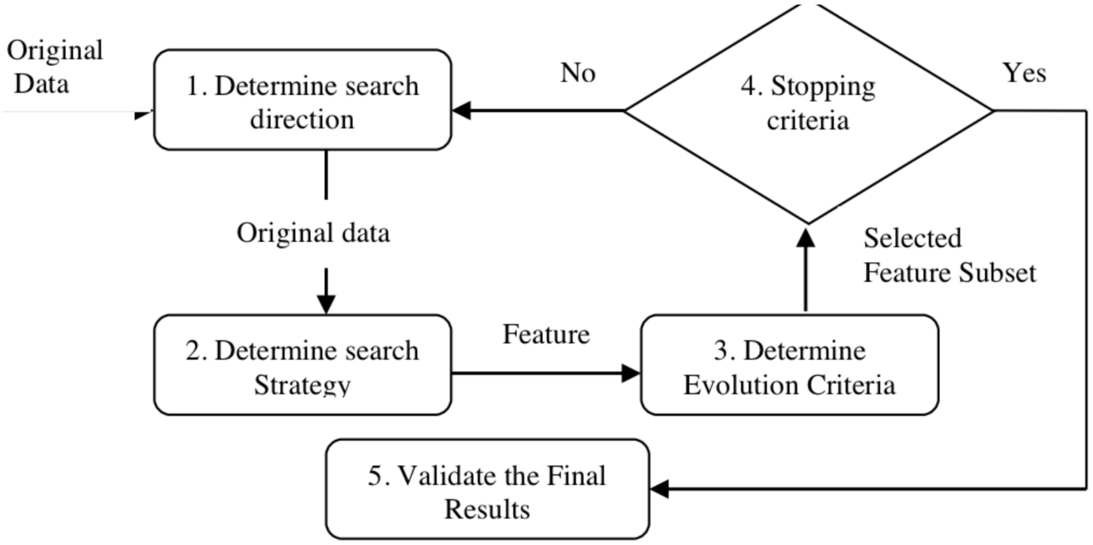
Abbildung 1: Underfitting, optimal, overfitting machine learning model.

The Filter method selects features based on statistical factors. It is used as part of preprocessing step in the feature selection, wich means this methods help to remove the not or less important features of a dataset before using the data to be fit on a classifier model. This method does not depend on the learning algorithms and therefore consumes much less time. For an Example there are correlation coefficient or the chi-square test. [**VenkateshAnuradha:2019**] **CAI201870**

PISNER2020101

The Wrapper method totally depends on the classifier used, wich means it does need more computation time than the Filter method. On the other hand the best subset of features comes directly based on the results of the classifier and they are more accurate than the filter methods. [**VenkateshAnuradha:2019**] **CAI201870** These methods train the classifier repeatedly and validate the results of the model after each iteration. In this way, the quality of the feature subset is iteratively improved. Of course, depending on the classifier and data set, this approach can lead to large computation times, which should definitely be taken into account when using it. [**PISNER2020101**] Wrapper models mostly use the accuracy rate and the classification error as default evaluation scores. The feature selection results of these models are often created at the same time as the results of the machine learning model. This is due to the fact that the learning model is embedded in the feature selection. [**CAI201870**] An examples for this method are genetic algorithms.

The third variant is the wrapper method. This performs better than the other two because it requires less computation time and makes collective decisions based on hybrid learning or ensemble learning. An example of such a method is the random forrest. [**VenkateshAnuradha:2019**] **CAI201870** Feature selection methods should have a small time and space complexity and do not generate a lot of overhead, but must also have a high learning accuracy **CAI201870**.


Original Data
1. Determine search direction
No
4. Stopping criteria
Yes
Original data
Selected Feature Subset
Feature
2. Determine search Strategy
3. Determine Evolution Criteria
5. Validate the Final Results

Source: [**VenkateshAnuradha:2019**]

Abbildung 2: Stages of the Feature Selection process.

Figure 2 shows the 5 steps of a feature selection process. The process starts by the search direction, wich can be forward, backward or random. The second step is to define wich of the three search strategies, randomized, exponential or sequential, should be used. After that the feature selection method selects features by the use of the evaluation criteria. To reduce cost, computation time and complexity, it is important to specify a stopping criteria, wich leads to stop the process earlier, for example if there were no new improvements for a while. It defines the point on wich the method should break. [**VenkateshAnuradha:2019**] For example the depth of a decision tree defines the maximum number of branches, nodes and leafs of the tree, wich also defines its maximum complexity. After the feature selection algorithm finished it search process, the results must be validated. For this step there are a lot of methods. For example cross validation or confusion matrix. [**VenkateshAnuradha:2019**] How important parameters were considered to be for the avalanches in each study seems to be strongly related to what parameters were available and wich machine learning models have been used for the study. As an example, in the study in Iran, wich is described in the article SSnow avalanche hazard prediction using machine learning methods"**Bahram:2019**, elevation was not ranked as particularly important for prediction, whereas in a study in India reported in the paper "Parameter importance assessment improves efficacy of machine learning methods for predicting snow avalanche sites in Leh-Manali Highway, India"**Tiwari:2021**, it has been ranked as the second most important feature. In the First Study, more additional meteorological and geographic parameters were available, which appear to be more important than the elevation **Bahram:2019 Tiwari:2021**. Because of its low computation time for highly dimensional datasets and good results, in context of this thesis genetic algorithm is used as search strategy. In case of this thesis decision trees, logistic regression and SVM are used as classifiers for a Genetic Algorithm. The next three chapters describe these as well as the genetic algorithm in detail and give an understanding about how they are used to find the important features of an dataset. The Decision tree and Logistic regression models are also used to give an general understanding about statistical importance of all features in the dataset.

1.2.1 Decision Tree

The Decision tree is a powerful hierarchical supervised machine learning model wich is non-parametric and can be used for both, classification and regression problems. Additionally it is a recursive build data structure based on the concept of dividing-and-conquering **SUBASI202091** Or as it is defined in the documentation for the decision tree algorithm in the Python librars Scikit-learn:

> "Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation."[**Scikit-learn-decision-tree:2022**]

These machine learning model is a representation method based on knowledge about the features of a dataset to represent classification rules **SUGUMARAN2007930**. Decision Trees use a set of if-than-else rules to decide wich value to predict. These models are good to understand, interpret and visualisable because they use white box models in wich every step is a boolean logic and easy explainable **Scikit-learn-decision-tree:2022**. For this reason, decision trees are also often preferred to other methods that actually provide accurate results **SUBASI202091**. A standard decision tree starts with a root node, does have some branches as well as child nodes and leaves **SUGUMARAN2007930**. Each of these decision nodes labels the resulting nodes or leaves with discrete scores, wich shows how much the input set has been separated. The branch wich is chosen after each node depends on the input in combination with the test function of the node. [**SUBASI202091**] The root node splits the set by a rule on the features wich provides the best classification of the instance. This goes recursive till the max depth is reached or the classification is completed. So a branch is the path from the root node to the leaf. The leaf at the end of an branch, wich represent the class labels of the feature to predict. [**SUGUMARAN2007930**] For the process of building a decision tree, the dataset is split into two or more subsets in each phase. For that in every phase it is searched for the best split for the input set. This process is continued recursively with the subsets until there is no need to split them anymore. This state can be reached by the tree it self, causing the fact that the resulting leaves are completely pure, or by the maximum depth stop criteria. This early stopping of the tree build is called prepruning, but there is also another method to simplify the tree called postpruning. This method grows the decision tree completely until all leaves are pure. Then all subtrees caused by overfitting are identified and pruned. This method can deliver better results in practice. [**SUBASI202091**] It's also possible to use them to predict multiple values at the same time, wich is a typical problem in supervised machine learning called the Multi-output problem **Scikit-learn-decision-tree:2022**. Decision Trees are likely to overfit if used on high dimensional datasets, but if used with a low tree depth, they can give a good understanding about the importance of some individual features for the prediction of multiple or specific parameters **Scikit-learn-decision-tree:2022**. The deeper the nodes are in the tree, the less important the features they represent are for the prediction. In addition, the decision tree contains only parameters that contribute to the prediction. Therefore, not only the importance of the features for classification can be determined, but also whether they can be used for classification at all. [**SUGUMARAN2007930**] This advantages of decision trees make them also useful for feature selection. In the case of the study, wich is represented in the article "Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing"**SUGUMARAN2007930** Decision trees are used for the feature selection. Similar to this work, the study was concerned with a classification problem.

1.2.2 Genetic algorithm

Genetic algorithm is an evolutionary based adaptive optimization search methodology. They contain to the feature selection category of wrapper models. As a Wrapper it is used for the second step of feature selection to find an optimal subset of features for the learning algorithm **CAI201870**. Like a lot of other technical inventions, the functionality of genetic algorithms is inspired by nature. For example Neuronal Networks are inspired by the functionality of the human brain. Genetic algorithms resemble the Darwinian natural selection and evolution of species. They use this mechanisms to optimize modeling problems and get a good subset of features. Genetic algorithms simulate the natural selection of species **LU2008887** . This means only the species who survive environmental changes can become another generation. Each generation represents a population of individuals. Each of this individuals represents a single solution for the problem and is defined by a genetic string wich is build out of chromosomes wich represent encoded features. [**LU2008887**] **DBLP:1912** Genteic algorithms are also able to handle huge dimensional datasets efficiently because of their exploitational and explorational

characteristics **LU2008887**. The algorithm starts by creating a random generated population, wich happens by generating a number of chromosomes. After that step a classification model is constructed based on the combination of variables of each chromosome. This classification model is validated, on each chromosome, with an k-fold cross validation by the use of statistical scores like the accuracy score. The fitter chromosomes have a higher chance to get passed on to the next generation. After that the genetic algorithm selects and recombinates the chromosomes by the validation of the scores from parent and offspring to get a new population. It depends on the stop criteria whether the algorithm stops or runs the same cycle again with the new population. The algorithm needs an stopping criteria on wich it will stop processing new generations. **[DBLP:1912] Yang:2018 LU2008887**
Jianjiang Lu and Tianzhong Zhao and Yafei Zhang **LU2008887** describe the three main operations for the process of a Genetic search methodology, wich are selection, crossover and mutation operation, as follows. The selection operation searches for the strongest N individuals from the current population. These are used as parents for the next generation of individual solutions. The crossover operation is spliced into three steps. At first it generates C_N^2 pairs of combinations between all parent individuals. Secondly it generates the two numbers $a(0 < a < m)$ and $b(0 < b < ma)$, in wich m represents the length of each chromosome, a indicates the start position of the crossover operation and b is the length of the crossover operation. For the last step, it is assumed for each parent pair $C_1^t = \{w_k\}$ and $C_2^t = \{w'_k\}$ with $k = a + 1, \ldots,$ where $a + b$ are two gen groups. To generate two new individuals for the pool of individuals, wich is used in the mutation operation to generate a new population, the gens in the range of $[(a+1), (a+b)]$are exchanged. The exchange is carried out on the basis of the crossover rate P_c as follows **LU2008887**:
$C_1^t{+}1 = w_1, k, C_2^t{+}1 = w_2, k,$ where $w_1, k = \gamma * w'_k + (1-\gamma) * w_k, w_2 k = \gamma * w_k + (1-\gamma) * w'_k,$ in this context γ is a predefined constant. The Mutation operation takes the, in the separation operation, created individuals into a pool with the parent individuals so that the variation in the new population is guaranteed. The K worst individuals out of this pool get a small mutation rate P_m. After that a number of genes are pickt, from every individual, by the mutation operation and a new offspring is generated as following: when a gene $w_k(w_k \in [0,1]$ is mutated and its next generation isw'_k, the mutation operation is **LU2008887**:

$$w'_k = \begin{cases} w_k + \Delta(t, 1 - w_k), & \eta = 0 \\ w_k + \Delta(t, w_k), & \eta = 1 \end{cases}$$

LU2008887

The variable η is a random number wich can be ether '1' or '0' and the return value of the function $\Delta(t, \gamma)$ is in range $[0, \gamma]$ **LU2008887**.

$$\Delta(t, \gamma) = \gamma(1 - r^{(1 - \frac{t}{M})})$$

LU2008887

r is a number wich is randomly chosen in the range of $[0, 1]$. Furthermore, t shows the value of the iterations. M represents the maximum of iterations and p indicates the predefined mutation parameter. Caused of this functions, the genetic algorithm mutates in earlier generations more than in later ones. **[LU2008887]**

In the article ”Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS”, Petro Liashchynskyi and Pavlo Liashchynskyi tested Grid Search, Random Search and Genetic Algorithm on the CIFAR-10 Dataset. They concluded that the Genetic algorithm took more time, but also produced better results. With larger numbers of features, it was even faster than the other two. [**DBLP:1912**] The authors of the article ”Predictor selection method for the construction of support vector machine (SVM)-based typhoon rainfall forecasting models using a non-dominated sorting genetic algorithm”**Yang:2018** used the genetic algorithm in combination with the SVM classifier for the prediction of typhoons. This natural disasters are dependent, as well as snow avalanches, on meteorological and topographical data.

In the context of this thesis the genetic algorithm implementation GAFeatureSelectionCV contained in the sklearn-genetic-opt Python library **Sklearn˙genetic˙feature˙docu:2022** is used.

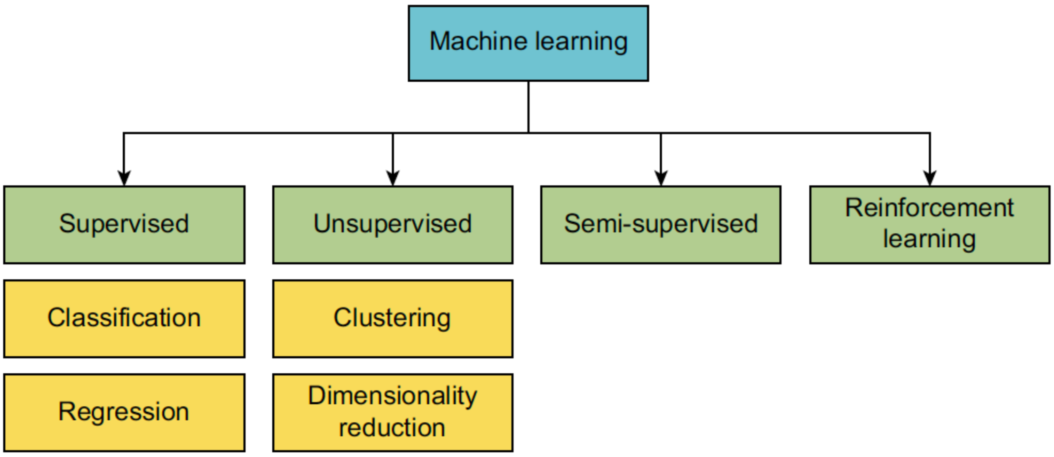
1.3 Machine Learning

The initial idea behind all artificial intelligence concepts is to make a computer able to perform tasks, that normally have to be accomplished by human brains. Machine learning is a subfield of artificial intelligence wich became popular in the 1990s and is inspired by theories from psychology and neuroscience about how humans learn. [**VIEIRA20201**] [**SUBASI202091**] Abdulhamit Subasi describes the initial goals of machine learning in his book ”Practical Machine Learning for Data Analysis Using Pythonäs follows:

> ”The goals of machine learning are defined as development and improvement of computer algorithms and models to meet decision-making needs in real-world situations.”[**SUBASI20209**

Machine learning models are supposed to recognize patterns within data based on learning data in order to be able to make predictions or decisions based on their findings in combination with new or unseen data. They are algorithms designed to automatically improve their decisions and predictions based on experience. Additional information should therefore help them make better predictions. The models learn rules wich are generalizable, since it is not very likely that the model makes predictions on exactly the same data, but in most cases receives similar data for that process. [**VIEIRA20201**] One of the biggest obstacles in the field of machine learning is obtaining good data, because ultimately the quality of the model depends directly on the quality of the data. Data acquisition can be very time-consuming and difficult, as there are no high-quality data sets for many scenarios. [**SUBASI202091**]
As shown in figure 3, in Machine learning there are four base types of models: supervised, unsupervised, semisupervised and reinforcement learning. [**VIEIRA20201**] [**ibm-supervised-learning:2022**] [**PISNER2020101**]

Supervised learning models get their names from the fact, that they are knowing both input and output variables during learning process. Therefore, they know the output values they are supposed to predict. They try to recognize the best possible relationship between input and output values while being trained on examples. Labeled datasets are used to train this models. During the training process, the weighting of the features is adjusted until the model is well fitted to the data set. [**VIEIRA20201**] [**ibm-supervised-learning:2022**] As a comparison for the



Source: [**PISNER2020101**]

Abbildung 3: Machine learning model learning types.

way a supervised learning model learns with humans, the authors Sandra Vieira and Walter Hugo Lopez Pinaya and Andrea Mechelli use in their book ”Machine Learning”the way a student learns from his teacher. The teacher knows the right answers, asks the student questions, and gives feedback on the student’s answers. [**VIEIRA20201**] As illustrated in figure 3 supervised learning is divided into two subcategories, classification and regression.

The machine learning algorithms wich are used for classification attempt to identify the relationship between the features of an observation and, on the basis of this, assign the observation to one of a number of classes, wich are known in advance **VIEIRA20201 SUBASI202091**. For example, the two classes that are in the context of this work are: än avalanche is going downänd ”no avalanche is going down”. While the classes in classification are fixed in advance, in regression any real numeric value on a continuous scale can be used for the prediction. The output variable is therefore not a categorical but a continuous one. [**VIEIRA20201**] [**SUBASI202091**] The sustainable use of supervised learning models can be challenging, since a certain level of expertise is required for their use, training the models can be time-consuming, erroneous records may have already been made when the dataset was created, and the algorithms cannot independently classify or cluster the dataset but rely on predefined classifications or regressions **ibm-supervised-learning:2022**.

In contrast to supervised learning, there is no target value to be predicted in unsupervised learning; the learning process is reduced to the structures within the data. Thus, the main applications of this type of learning are clustering, where similar data points are recognized and clustered based on the structures within the data, and the second is dimensionality reduction, wich is used to reduce the dimensionality of a dataset if the number of features is higher or near to the number of rows in the dataset.[**ibm-supervised-learning:2022**] [**VIEIRA20201**]

Semi-supervised learning is an addition to supervised learning. It is used in cases of partly labeled datasets and makes it possible to integrate the unlabeled data into a supervised learning. [**ibm-supervised-learning:2022**][**VIEIRA20201**] The reinforcement machine learning algorithms are used to learn from interactions with their environment. So in the beginning there is no dataset needed to train this type of machine learning algorithms. The learning methodology behind these algorithms is based on the concept of rewards and punishments that it receives based on its decisions, and attempts to arrive at as many rewards and as few punishments as possible in the course of learning based on trail and error. Compared to supervised learning, the algorithm is free in its behavior in the reinforcement technique. [**ionos-reinforcement-learning:2022**] [**VIEIRA20201**]

In order to achieve adequate results, a series of machine learning models are trained in the context of the thesis. Causing the fact that the prediction of avalanches for explizit defined locations is a binary classification problem, only machine learning models of the supervised learning type are applied on the task.
In the past, some models have already proven their worth in predicting natural disasters. For example, the Support Vector Machine (SVM) and the Multivariate Discriminant Analysis (MDA) models, wich is an addition to the Linear Discriminant Analysis described in chapter 3.2.3. They are useful for detecting subtle patterns in complex data sets and Flexible in handling data of different dimensions. SVM models are designed to deal with high dimensional data. Thats one aspect why they have already been used to predict natural disasters, such as earthquakes, floods, typhoons, drought, landslides and avalanches **Bahram:2019 Tiwari:2021 Pozdnoukhov:2008**. MDA forms efficient linear combinations of independent variables. MDAs have not been used that often to predict natural disasters, but shows superior performance compared to SVM in the case study in the Karaj water conservation area in predicting avalanche risk levels **Bahram:2019**. So for this master thesis a logistic regression, support vector machine and a multivariate discriminant analysis are trained, evaluated and the performance compared. The three machine learning models Logistic Regression, Support Vector Machine and Linear Discriminant Analysis are all relatively transparent about their approach to predicting observations. The three models are described in the next three chapters in detail.

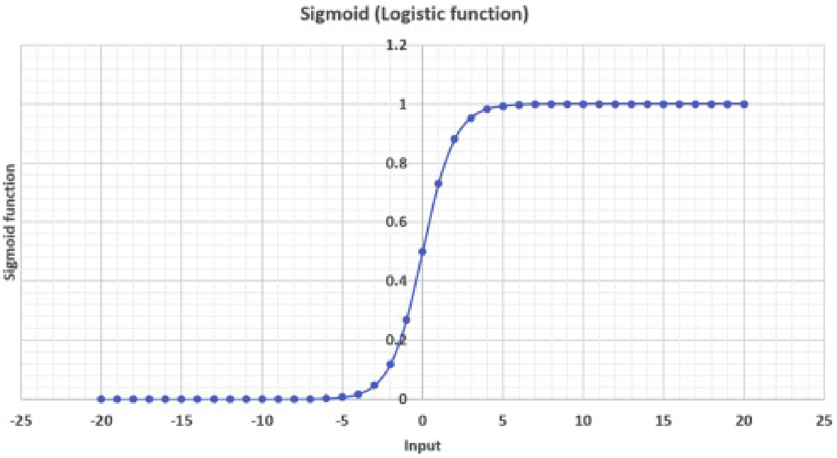
1.3.1 Logistic Regession

The Logistic Regression is a popular classification training algorithm, wich is often used in the field of predictive analytics. It is also a supervised and discriminative machine learning model. [**ibm-logistic-regression:2022**] The logistic regression and linear regression models are two of the most popular models in the field of data science, as they are very easy to execute and require little computation time. linear regression is used to find the correlation between two features. This is done by drawing the line that best fits through a number of data points. While the method, wich is used by the linear regression to calculate the loss function, is the mean squared error, in logistic regression the maximum likelihood estimation is used. Similar to the behavior of the linear regression, logistic regression is used to calculate the correlations between one or multiple features and the variable to be determined, but it is used to predict categorical variables. Binary categorical variables can only have two states, for example 1 or 0. So because lineare regression is used to predict continuous variables the major difference between them

is, that logistic regression handles binary classification problems and linear regression handles the regression problem. [**BELYADI2021169**] **ibm-logistic-regression:2022 Sourav:2020** The name ”Logistic regression”can be misleading, because it is more a classification model than a regression model **SUBASI202091**. Instead of directly searching for the best fitting regression line in the data, like the lineare regression model does, it splits this process into three steps. First, similar to the linear regression, a regression line is fit onto the data. In the case of predicting categorical variables, the line is very susceptible to outliners. Because of that fact the next step is to feed the results to the sigmoid function, wich outputs are always between 0 and 1. [**ibm-logistic-regression:2022**] **Sourav:2020 BELYADI2021169**
The sigmoid function also known as logistic function:

$S(x) = \frac{1}{1+e^{-x}}$ **Sourav:2020 BELYADI2021169**

As a last step the result values of the sigmoid function are converted to the values 0 or 1 (discrete values) based on the threshold, wichs standard value is 0.5. This means if the value is greater than 0.5 the resulting prediction value is turned to 1 and if it is smaller it is changed to 0. [**Sourav:2020**] **BELYADI2021169**



Source: [**BELYADI2021169**]

Abbildung 4: Sigmoid function curve in logistic regression.

The S-curve displayed in figure 4 is the result of the sigmoid function fed with values between -20 and 20. As the curve shows, the values resulting from the logistic function are in the range between 0 and 1.
In addition to the binary classifications variant, which is the most widely used variant of logistic regression and generally one of the most common methods for binary classification, there are two other variants. The Multinomial logistic regression and the Ordinal logistic regression. The Multinomial logistic regression is used for classification with three or more possible result values for the determined value, wich are in no particular order. Ordinal logistic regression is also used for multiclass classification tasks, but in this case the variables are in a specific order. For example a evaluation scalar with one to five stars. [**ibm-logistic-regression:2022**] **SUBASI202091**

Similar to other machine learning models, like neuronal networks, support vector machines and multiple discriminant analysis, wich are also used in context of this master thesis and described in detail in the later part of the work, logistic regression does not need linear relations between the predictor variables and the variable to be determined. They capture nonlinear relationships in the dataset. [**NUSINOVICI202056**] **BELYADI2021169**
Logistic regression models are easy to realize while they are achieving good results for binary and linear classification problems **SUBASI202091**. For Example V. Sugumaran, V. Muralidharan and K.I. Ramachandran found out, that for their case study about major chronic diseases logistic regression could keep up with the other machine learning algorithms and in two cases even delivered better results **NUSINOVICI202056**. The Python library Scikit-learn does have a highly optimized version of the logistic regression algorithm, wich also can handle dens as well as sparse input **Scikit-learn-logistic-regression:2022**. In context of this thesis, the scikit-learn implementation is used.

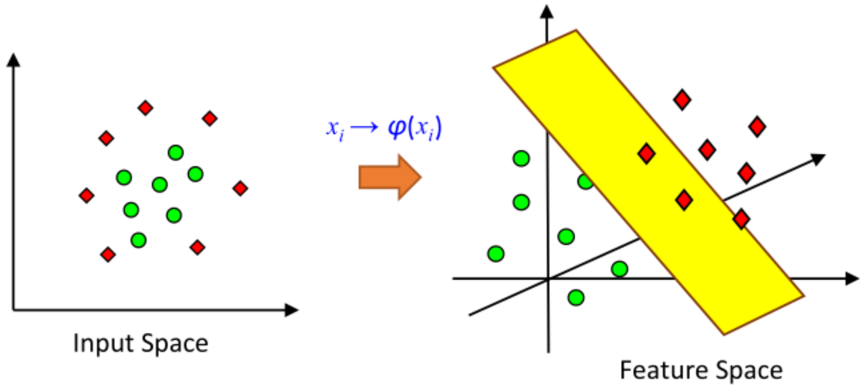
1.3.2 Support Vector Machine

Support Vector Machines (SVM) are supervised machine learning algorithms based on the statical learning theory and have been introduced the first time in the 1990s and developed by Vladimir Vapnik **ibm-supervised-learning:2022 SUGUMARAN2007930 GHOLAMI2017515**. They can be used for classification and regression problems as well as to detect outliners **ibm-supervised-learning:2022 Scikit-learn-svm:2022**. Abdulhamit Subasi gives a general overview about SVMs in his book ”Practical Machine Learning for Data Analysis Using Pythonäs follows:

> SSupport vector machines (SVMs) are one of the main machine-learning algorithms that are not only accurate but also highly robust.”[**SUBASI202091**]

The method, wich Support Vector Machines use, tries to find the best classification function that splits the training set into the classes of the variable to be determined. [**SUBASI202091**] Each feature can also be considered as a dimension in a hyperspace. The SVM creates a hyperplane to split the hyperspace into two or more parts. This depends on how many classes are to be predicted. So the SVM can be applied to cases of the multi-class problem just like decision trees. [**SUGUMARAN2007930**] Support vector machines can be linear and non-linear, but classification problems are in most cases linear, therefore mostly linear SVMs are in use **PISNER2020101**.
If the data is not linearly separable, the support vector machine can not create a good generalization. To solve this problem, it projects the data points onto a higher dimensional hyperspace. Based on the mathematical assumption that a non-linear separation in a higher dimensional space is linear. This higher dimensional space is also called Hilbert or feature space. As a result of this assumption, the input data are still non-linear, but in the feature space the application of a linear Svm and thus a better generalization is possible. Figure 5 displays the differences between the input space and the higher dimensional feature space, in wich a linear hyperplane is placed to separate the data. [**GHOLAMI2017515**]

In the case that the data set is linearly separable, the linear function is used to compare the



Source: [**GHOLAMI2017515**]

Abbildung 5: Input space in comparison to higher dimensional feature space.

separating hyperplanes. This is necessary because in this case the SVM raises the margin between the classes to the maximum based on these quantities. Contrary to the assumed definition that margin is the space between classes, the mathematical definition is the shortest distance from the hyperplane to the closest data point. Although many hyperplanes are located in hyperspace, support vector machines can only use two of them. To ensure the best possible classification of current and future data, the most extreme margin of the hyperplanes is determined. [**SUBASI202091**] The search for the maximum margin is minimizes also the generalization error of the SVM. [**SUGUMARAN2007930**]
A larger margin allows better generalization and a hard margin is the simplest way with the least computation time, but it might not be perfect in practice. In fact that, in the case of a hard marin, the hyperplane is affected even by one single outliner. This lead to hyperplane mistakes and misclassification. So another option is to use a soft margin instead. In this approach the hyperplane can get highly complex so as a compromise the penalty factor C, wich is called the ßoft margin constant"comes into the process. It is used to make a compromise between complexity and classification errors as well as reducing the chance of overfitting. For this reason, it is often argued that the soft margin variant should also be used for linearly separable datasets. **[PISNER2020101] GHOLAMI2017515**
Support vector machines can also be used for the generalization and are effective for highly distributed, high dimensional or spare datasets, even if the number of samples is smaller than the number of dimensions. It also indicates an higher accuracy in comparison to other classification machine learning methods like Neuronal Networks, because of its good generalization capacity. **[SUBASI202091] Scikit-learn-svm:2022**

As shown in Section 2 "Related Work", SVMs performed well in previous studies in the context of avalanche event prediction and avalanche hazard mapping. [**Tiwari:2021**, **THURING201560**, **Bahram:2019**, **Pozdnoukhov:2008**] In the context of this work one implementation included in the Python library Sklearn.svm is used to build the prediction model. The chosen algorithm is called C-Support Vector Classification (SVC). The implementation of this algorithm is based on another Python library with the name libsvm, wich implements a series of different SVM algorithms. This implementation scales in the computation time at least quadratically. The documentation therefore mentions that the long computation time can be impractical when the number of samples exceeds 10000 and that another implementation, such as LinearSVC, should be chosen in that case. In the case of this study, however, the number of samples is less than 10000 and the implementation can be used.

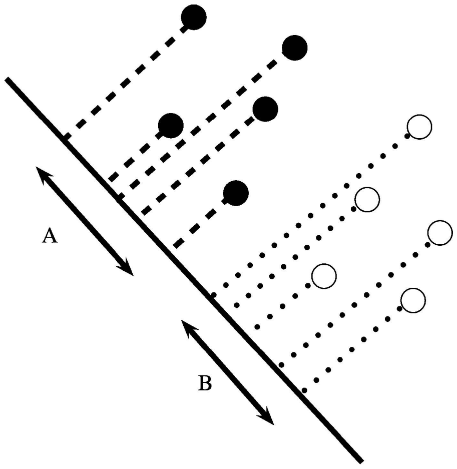
1.3.3 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a fundamental data analysis method, wich has been first time examined by Fisher in 1936 for two classes and later in 1948 C.R. Rao found it for multiple classes **DEMIR2005421 analyticsvidhyaLDA:2021 Xanthopoulos2013**. R. Fisher used the LDA to differenciate between two different types of plants **Xanthopoulos2013**. LDAs are supervised machine learning algorithms, as it depends on user input and the knowledge about class affiliation of each data point, but it is also a dimensionality reduction techinque. For this reason it can be used as classifier machine learning method to classify samples of an dataset with multiple independent variables to two or more classes and also to determine the class of unknown varibales. It can also be used for data preprocessing as a dimension reducing method or to identify the how significant the individual features are, represented by the corresponding coefficients of the hyperplane. [**MENDLEIN2013646**] [**Bahram:2019**] [**Xanthopoulos2013**] [**SUBASI202091**] [**analyticsvidhyaLDA:2021**]. Discriminate Analysis projects the data points as close as possible to the data points belonging to the same class and moves the individual classes as far away from each other as possible. This is done by defining the distance of the points from the center of their class, wich is calculated by the use of normal distribution. [**MENDLEIN2013646**] [**Bahram:2019**] [**SUBASI202091**] [**Xanthopoulos2013**] Linear Discriminant Analysis increase the variability between the classes and reducing the variability within them by projecting the data from a D dimensional feature space on a lower dimensional subspace D' and creating new discriminant axes that represent linear combinations of the individual variables **analyticsvidhyaLDA:2021 DEMIR2005421**. As an example Figure 6 shows the projection of datapoints in da two dimensional space onto a lower dimensional subspace.

This process consists of three steps. At first, the separability between the classes needs to be calculated. It represents the distance between the mean of one class to the mean of another one and is called the "between-class variance". This variance is calculated for every class and a between class matrix S_B is created. The between-class variance for the ith class S_{B_i} is the distance of the class mean μ_i and the total mean μ. [**Tharwat:2017**]

In the second step, the distances of the individual data points within a class, also known as the "within-class variance", are calculated. The within-class variance S_{W_i} is calculated based on the distance from each point within a class with the mean of the class. [**Tharwat:2017**]

In the last step, the lower dimensional subspace is constructed so that the between-class variance is as large as possible and the within-class variance is as small as possible. For this step, there are two methods for the calculation of the lower dimensional subspace. In the first one, wich is class-dependent, a lower dimensional subspace is generated for every class and project its data points onto it. The second method is called class-independent. It only calculates one subspace for all classes and projects their data points on it. [**Tharwat:2017**]



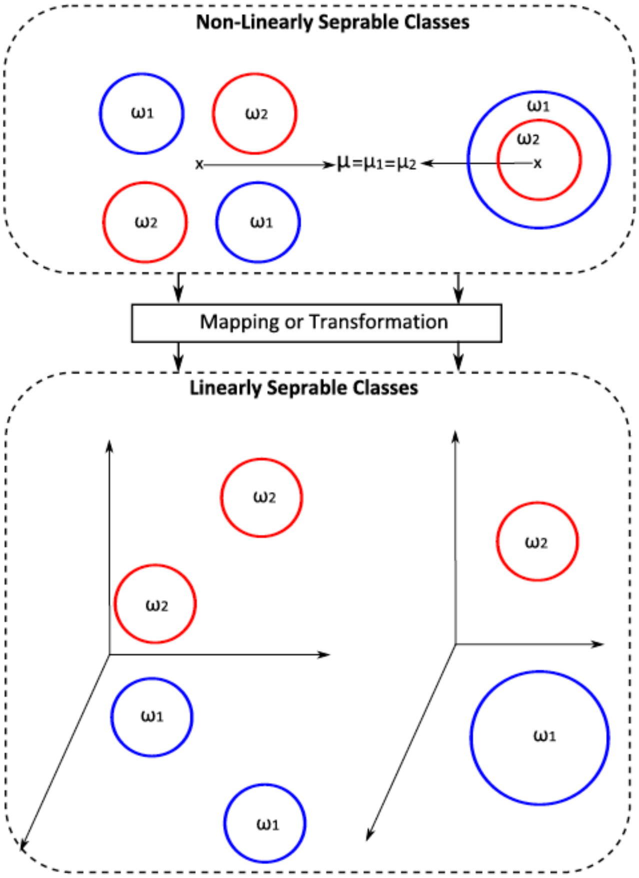
Source: [**Xanthopoulos2013**]

Abbildung 6: A Two dimensional data set is projected in a lower dimensional subspace, wich is a line. In this way the separability is increased.

LDAs do have two main problems. The Small Sample Problem and the linearity problem. The Small Sample Problem means that LDAs can not handle datasets where the number of variables is larger than the number of samples. This would cause a fail calculation of the lower dimensional subspace by the LDA. [**MENDLEIN2013646**] [**analyticsvidhyaLDA:2021**] [**Tharwat:2017**] The LDA can also run into another problem, the "linearity problem". This problem arises when the individual classes are non-linearly seperable. In this case the LDA fails to find a lower dimensional subspace. For example when the means of the classes are equal. One approach to fix this problem is to create a higher dimensional space, similar to the svm. An example for this scenario and how the problem is solved by increasing the dimensions of the space is shown in figure 7. The four datapoints in the figure are not linearly separable on the two dimensional space and the problem does not get solved by putting it onto a lower dimensional subspace. So the LDA projects them onto a three dimensional space. In this new higher dimensional space the classes are linearly separable and can be projected onto a lower dimensional subspace. [**Tharwat:2017**]
The Multivariate Discriminant Analysis (MDA), wich is an addition to the LDA, has brought good results in a study in the Karaj watershed, in wich they used SVMs and MDAs for an avalanche hazard prediction and compared the performance of both algorithms **Bahram:2019**. For the case study of this master thesis the Python implementation of Linear Discriminant Analysis, wich is included in the library sklearn.discriminant_analysis is used. [**Scikit-learn-lda:2022**]

1.4 Performance Evaluation

The Evaluation of machine learning models is core part of building an effective and robust model. It is not only a technique to get feedback, at the end of a machine learning training process, wich shows how good the quality of the models results are. The performance evaluation is also part of the optimization process while training a machine learning model. This can be an iterative process, in wich a model is trained, after that feedback of quality is obtained through metrics,


Non-Linearly Seprable Classes
ω1
ω2
ω2
ω1
x
μ=μ1=μ2
x
ω1
ω2
Mapping or Transformation
Linearly Seprable Classes
ω2
ω2
ω1
ω1
ω2
ω1

Source: [**Tharwat:2017**]

Abbildung 7: Two different examples for non-linear separable classes, in wich the problem is solved by generating a higher dimensional space and make a linear separation of the classes possible for the LDA.

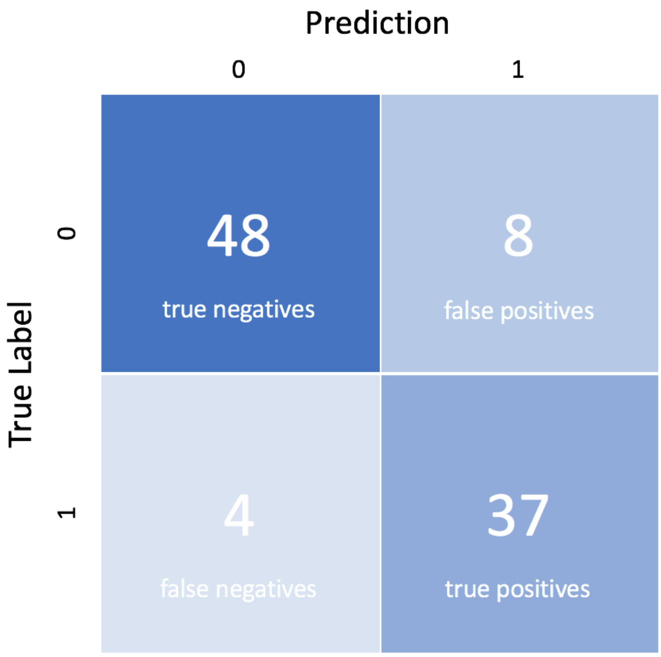
then the models hyper-parameters or features are improved (depending on the actual phase of process of creating a machine learning model) and this is repeated until a solid model has been found. [**analyticsvidhya˙evaluation:2022**]
The metrics used to evaluate supervised machine learning models are also divided into evaluation of classification and regression models **jeremyjordan˙evaluation:2022**. Causing the fact that the prediction problem of this study is a classification problem, only evaluation metrics wich are used to evaluated classifier models are mentioned in this topic.
The outcome of a binary machine learning classifier prediction has one of the four following types **jeremyjordan˙evaluation:2022**:

- True positive (TP): The model predicts that an observation belongs to a class and it actually belongs to that class.

- False positive (FP): The model predicts that an observation belongs to a class and it actually does not belong to that class.

- True negative (TN): The model predicts that an observation not belongs to a class and it does not belog to that class.
- False negative (FN): The model predicts that an observation not belongs to a class and it does belog to that class.

This four values can be plotted on a confusion matrix. An example for that matrix is shown in Figure 8. The matrix is generated by making predictions on the test data and assigning the results of the individual samples to the four types. Also different classification model evaluation metrics, like the three main scores accuracy, precision and recall score are calculated with these values. [**jeremyjordan˙evaluation:2022**]



Source: [**jeremyjordan˙evaluation:2022**]

Abbildung 8: A confusion matrix wich shows the four types of a classifier outcomes.

1.4.1 Accuracy

The Accuracy score is one of the most common evaluation metrics, when it comes to the evaluation of binary classifiers **Kartik˙evaluation:2022**. Also a lot of the related works use this score to evaluate and compare their machine learning models. Jeremy Jordans defines accuracy as follows:

> Äccuracy is defined as the percentage of correct predictions for the test data.”[**jeremyjordan˙evaluation:**

The definition can also be represented like this:
$Accuracy = \frac{Number of correct predictions}{Number of total predictions}$ **Google˙Acurracy:2022**

For binary classifiers accuracy is can be calculated as follows:
$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ **Kartik˙evaluation:2022**

As the definitions describe, the score represents the percentage of the correctness of all predictions. This implies that in the case of an unbalanced class size, the accuracy of a class can be disregarded **Kartik˙evaluation:2022**. As an example in the context of avalanche prediction: There are 90 samples that do not contain an avalanche event and only 10 wich represent an avalanche event. In the case that the algorithm predicts no avalanche event for all samples, it has an accuracy of 90%.
This does not mean that the accuracy score is not useful, the metrics gives a validation of the overall prediction performance of the model. It only signifies that it should not be the only score used for the evaluation, especially in cases of unbalanced class sizes. [**Google˙Acurracy:2022**]

1.4.2 Precision

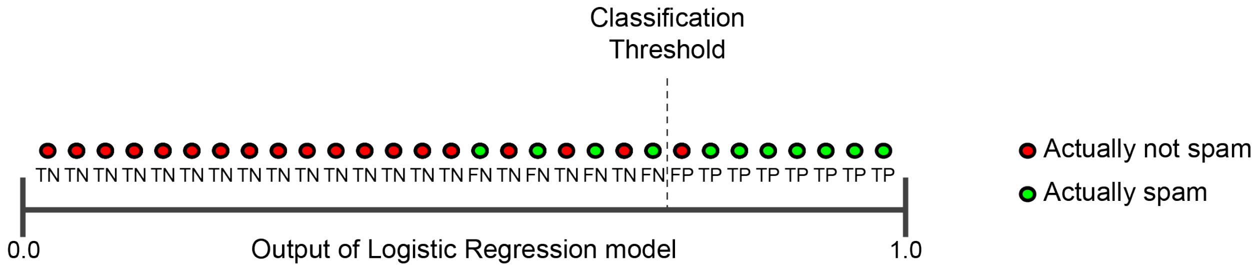
The Precision score is the percentage of how many positive predicted observations actually are positive **Kartik˙evaluation:2022**.
It is defined as follows:

$Precision = \frac{TP}{TP+FP}$ **Kartik˙evaluation:2022**

The score can be a balancing validation metric for the accuracy score, since it covers exactly the cases in which the accuracy score has the problem described above. So to extend the example mentioned in section 3.3.1, in wich the Number of Avalanches is 10 and ne number of non avalanches is 90. The model only predicts all non-avalanches correctly so the accuracy is 90% but the precision score is 0%. So in this case the accuracy shows that the model has a high prediction quality, but the precision score clarifies that none of the avalanches was predicted.
The fact the use of the Precision Score without the evaluation with another metric, has a similar problem [**Google˙Precision˙Recall:2022**]. In case of the example if only one positive samples is correctly predicted as avalanche, the precision score is 100% but nine of ten avalanche samples are rated false. So the same balancing characteristic applies the other way around from accuracy score to precision score. Figure 9 shows a set of datapoints. The precision score evaluates the right side of the classification threshold line. Since the precision score only targets the samples predicted as positive. In the case shown in figure 9, seven out of 8 datapoints are predicted right, wich results in a precision score of 0.875.

1.4.3 Recall

The Recall is another evaluation score, calculated out of the four values represented in the confusion matrix. It is defined as the percentage of actual positive datapoints predicted as positive **Google˙Precision˙Recall:2022**. In figure 9 the recall is represented as all green marked data-


Classification
Threshold
TN TN TN TN TN TN TN TN TN TN TN TN TN TN TN FN TN FN TN FN TN FN FP TP TP TP TP TP TP TP
0.0
Output of Logistic Regression model
1.0
● Actually not spam
● Actually spam

Source: [**Google˙Precision˙Recall:2022**]

Abbildung 9: A set of datapoints split by the classifier and marked as one of the four classifier output types.

points on the right sight (the TP predicted samples) of the classification threshold line divided by the all green marked samples (the TP plus FN predicted samples).
The recall evaluation metric is calculated as follows:

$Recall = \frac{TP}{TP+FN}$ **Kartik˙evaluation:2022**

1.4.4 ROC-Curve and AUC

The ROC curve (Receiver operating characteristic) is a graph representing the True Positive Rate (TPR) compared to the False Positive Rate (FPR) for different classification thresholds of a model **Google˙ROC˙AUC:2022**. In figure 10 an example ROC curve is shown. Each dot on the curve represents the TP vs. the FP rate at a specific decision threshold.
The two parameters TPR and FPR are defined as follows:

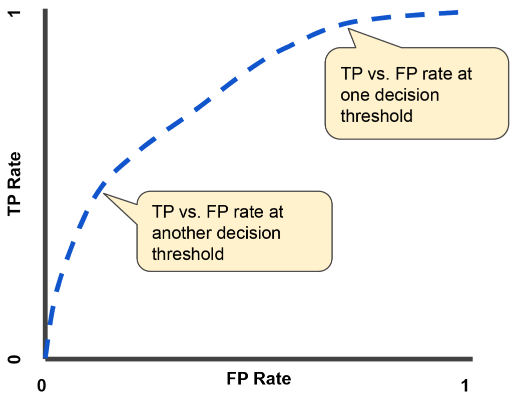
$TPR = \frac{TP}{TP+FN}$ **Google˙ROC˙AUC:2022**

$FPR = \frac{FP}{FP+TN}$ **Google˙ROC˙AUC:2022**

If the threshold is lower, the model classifies more samples as positive. The consequence of this is that both the TPR and the FPR increase. The same happens in reverse with a higher threshold. [**Google˙ROC˙AUC:2022**] [**Kartik˙evaluation:2022**]
For machine learning classifiers, wich have a class as output and do not use a threshold, the ROC curve will be represented as a single point in the plot **analyticsvidhya˙evaluation:2022**.

An interactive approach, in which a classifier model is evaluated many times with different thresholds, would be associated with high computational costs. However, there is also a more efficient approach called AUC, which can also determine this information. [**Google˙ROC˙AUC:2022**]

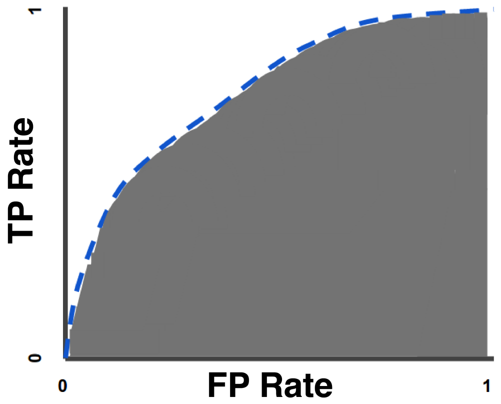
The AUC (Area under the Curve) statistic represents the integral calculus of the ROC curve



Source: [**Google˙ROC˙AUC:2022**]

Abbildung 10: The ROC curve plots the TPR vs. the FPR on all different thresholds.

from (0,0) to (1,1). Figure 11 shows an example for the AUC statistic. The grey marked area in that figure represents the AUC value of this ROC curve. It gives an aggregate measure of the models prediction quality about the whole range of possible classification thresholds. So with the AUC statistic, the ROC curve is represented by a single number. The value of AUC can be in the range between 0 and 1. If the Value is 0.0, the predictions are 1000% false. If the value is 1.0, all predictions are correct. [**Google˙ROC˙AUC:2022**, **analyticsvidhya˙evaluation:2022**] The value The higher the numerical values of the AUC statistic the better is the models performance **analyticsvidhya˙evaluation:2022** . This number is definitely meaningful, however, the entire ROC curve should always be considered as there are models that perform better in certain areas and other models in other regions **analyticsvidhya˙evaluation:2022**.



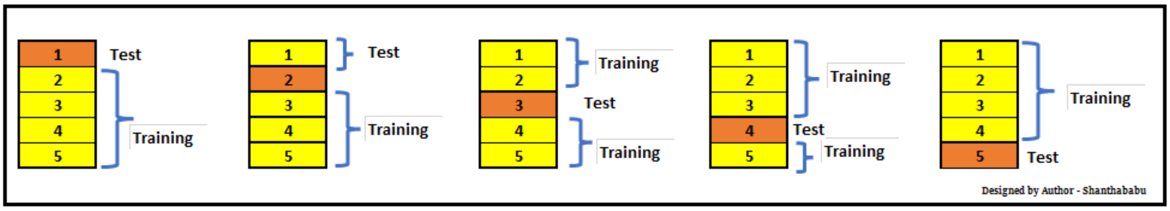
Source: [**Google˙ROC˙AUC:2022**]

Abbildung 11: The AUC statistic represents the grey marked area under the ROC curve.

In Context of this work, the implementation of the ROC-curve as well as the AUC value included in the same python library (Sklearn.metrics) as the other metrics mentioned before are used. This implementation of the ROC-curve is restricted to binary classification tasks. [**Scikit-learn-roc-curve:2022**]

1.4.5 K-Fold Cross-Validation

The k-fold cross-validation is a technique to evaluate machine learning classifier models in a balanced way and to avoid the risk of a random training test split variant with a splitting that is not meaningful. This balance is caused by the fact that with this method the model is both trained and validated with every data sample of the set. [**Refaeilzadeh2009**]
In the k-fold cross-validation, the dataset is split into k equally sized subsets. After that the model is trained iteratively with k-1 folds of the dataset and the last fold is used for validation. The one fold wich is held-out change every iteration until the model is validated with every sample of the set. The performance of each iteration is tracked by an evaluation metric like accuracy or precision. This process is shown for the example of an 5-fold cross validation in figure 12. In the figure, the test set is the orange and the training set is the yellow marked part of the set. As shown, this test part always shifts by one fifth of the total set. [**Refaeilzadeh2009**]
The 10-fold cross validation is a popular variant in terms of machine learning. the decisive advantage compared to, for example, a 70/30 train test split is that you have a large train set of 90% for each iteration. So the machine learning model does have more samples to learn from. At the same time k-fold cross-validation provides a precise test coverage. [**analyticsvidhya˙cross˙validation:2022**] In the case of this master thesis, the accuracy, precision and recall metric are all used in combination with the cross-validation implementation of the Python library sklearn.model_selection, in wich the metrics can be selected.



Source: [**analyticsvidhya˙cross˙validation:2022**]

Abbildung 12: The iteration process of a 5-fold cross-validation.