The prediction of snow avalanches based on selected meteorological factors for topographically defined mountain slopes using machine learning models. A case study in Kaprun/Mooserboden

2010695005

FH Salzburg

02. January 2022

Contents

1 Introduction

Avalanches are among the most dangerous natural disasters Hermann Maurer [1]. They threaten people and environment in seasonally snow-covered mountain regions such as the alps. In 2021, 265 recorded avalanches (recoreded on Lawis) in Austria killed 17 people and injured 5 others. A total of 41 people were involved in the events Lawinenwarndienst Tirol [2]. The occurrence of snow avalanches is steadily increasing due to climate change Eric Martin [3] Anuj Tiwari [4] Bahram Choubin [5]. In order to enable preventive measures to combat avalanches, a good assessment of the impending danger is necessary. The complexity of this task has already been described in several studies. It comes from the fact that there are many potentially influancial parameters, of which in most scenarios only a few are available and and they change significantly even for small geographical differences. For example, the estimation of avalanche danger levels is mostly based on weather data, which are also predicted, results of snowpack models and data on snow instability. In addition, information on the terrain is often used for these features Bahram Choubin [5]. Avalanches have been artificially triggered by explosions for a long time. For this preventive fight against snow avalanche accidents, automatic and semi-automatic snow avalanche detection systems have been developed. Of these, infrasound-based systems are the most promising, in the early detection of slow avalanches, for example ice avalanches Thüring, Schoch, van Herwijnen, et al. [6] Hermann Maurer [1]. In addition, these systems are used to detect as many avalanches as possible. Many avalanches have not been recorded, which means that the accuracy of predictions made by automatic systems is compromised. Geographic Information Systems (GIS), Hierarchy Process Methods (AHP), and Remote Sensing (RS) are used together for this purpose to assess and assess avalanche hazards. With the help of these systems avalanche hazards can be detected, but they have a high error rate. Therefore, predicting the risk of avalanches on selected slopes would be a great improvement in the precision of preventing avalanches by blasting them. Machine learning[1] is already being used for various real-world problems Subasi [7] and it also has been revealed that some machine learning models can achieve good results in predicting natural disasters and explicit avalanches Eric Martin [3] Anuj Tiwari [4] Bahram Choubin [5]. Various systems also use the Nearest Neighbor[2] method for their predictions Pozdnoukhov, Purves, and Kanevski [9]. Avalanche warning services could receive promising forecasts for the day and later for future days in advance and make them available to the public. This leads to a significant increase in safety for backcountry skiers and other winter mountain sports enthusiasts. Backcountry skiing is a sport in which the athletes are skiing unmarked or unguarded areas within or outside the boundaries of a ski resort. For this, they either climb the mountain with touring skis or are brought up by helicopter or snowmobile. Having good predictions for the avalanche hazard, makes it easier and safer to plan these tours. This has led to the author's personal interest for this topic and the resulting possibilities to get high precision predictions of avalanches hazard for topographical defined explizit mountain slopes by the use of machine learning models. Since there are people

[1] cf. chapter 3.2 of this thesis

[2] "The k-nearest neighbors algorithm, also known as KNN or k-NN, is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point. While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another." [8]

who work in avalanche-prone areas, the forecast for defined mountain slopes would also improve their work safety. For example, some employees of the editor of the data work in the area of the Glockner Group. The data provided for this work is collected purely for the purpose of improving work safety for this avalanche-prone area.

The aim of this master thesis is to predict snow avalanches for topographical explicit defined mountain slopes. In addition, another task is to find out which machine learning models are best suited for this task and predicts the snow avalanches with the best quality. As well as which factors contain the most meaningful informationS and have impact on the prediction quality of the machine learning models. This is done by using topographical data (e.g. slope, slope orientation by cardinal directions, altitude) in conjunction with meteorological data for the day to be predicted (e.g. weather, temperature, wind direction, wind speed, new snow), wich will be also used retrospectively two to four days in the past for selected data. Also Snow pack related data (e.g. snow depth, snow temperature, snow sinking depth) is used in combination with the meteorological data. Thus, the study is intended to provide an answer to the questions, ”Does the meteorological data contain enough information to apply machine learning to predict snow avalanches for topographically defined mountain slopes?” and ”With what quality can snow avalanches be predicted from meteorological data for topographically defined mountain slopes, in the specific context of this case study in Kaprun/Mooserboden?”.

2 Related Work

In the context of avalanche risk prediction, several studies have already been conducted. Machine learning methods have been applied for this purpose. Data sets of avalanche events from smaller mountain areas were used as case studies.

In the alps a dataset of avalanche events from the area around Davos, Swizerland Stephan Harvey [10] over the last 13 years is one example for these case studies. The study about the data from Davos aimed to predict the days in a winter season on which many avalanches occur for a whole winter season. To get the Meteorological[3] Data for their study, the Team form the SLF, Swizerland used the data from an automatic weather station. The authors combined these data with recorded snow avalanche events in the region of Davos and merged the simulated snowpack properties, like new snow depth, liquid water content, Stability indices, critical crack length, and the hand hardness, of the SNOWPACK model onto these data to get a dataset. For the predictions, the Team trained a Random Forest[4] machine learning model with the final dataset. Before training the Random Forrest, the authors subdivided the dataset into five groups to avoid overfitting. Each of these groups defines which criteria distinguish an avalanche day from a non-avalanche day, for a specified avalanche type. The five groups are divided into New Snow avalanches, Wet snow 1 avalanches, Wet snow 2 avalanches, Wind drift avalanches and Everything else. [10]
The authors tried five different approaches, all including Random Forrest machine learning models, and validated the results after that. For the first approach the number of features for each group is reduced by applying classification trees. Then, one random forest per group is trained with their data and the prediction of the groups per day is summed. The second method is likely the same as the first one without reducing the features of the groups. In the third the authors applied the random forest onto all 58 features of the dataset without separating into the five groups. The fourth attempt is same as the third, but all avalanche days which do not meet the criteria of the new snow, wet snow 1 or wet snow 2 groups are considered non-avalanche days. In the last method the random forest is only applied on avalanches of type new snow and measured meteorological data. [10]
One finding of the resulting study is that the predictions without the snowpack factors are just as good as those without this additional data. Furthermore, they concluded that their prediction attempts were severely limited by inaccurate in the visual observed avalanche activity data, the biased avalanche-related data and by the fact that the spatial scale is too large for their models. The Threshold, wich they did use for the study, is too different, as it allowed up to 10 large avalanches for one day but also 1 medium, for the same value. They came to the result that forecasting on a small spatial scale using only one avalanche warning system could work well and could be a good aid for avalanche hazard forecasting services. [10]

[3]"Meteorology is the study of the physical and chemical phenomena and processes in the atmosphere and their interactions with the earth's surface." [11]

[4]"Random Forest is a robust machine learning algorithm that can be used for a variety of tasks including regression and classification. It is an ensemble method, meaning that a random forest model is made up of a large number of small decision trees, called estimators, which each produce their own predictions. The random forest model combines the predictions of the estimators to produce a more accurate prediction." [12]

For another study, with the name: "Snow avalanche hazard prediction using machine learning methods" Bahram Choubin [5], avalanche hazard maps were attempted using predictions from machine learning methods. The space around the watershed of Karaj, Iran is used for the case study of this paper. Support Vector Machines (SVM), wich are described in detail in section 3.2.2 of this work as well as Multivariate Discriminant Analysis, wich are mentioned in section 3.2.3 of this thesis have been used as machnine learning techniques in this study. The goal of the study is to evaluate the performance of the two machine learning methods SVM and MDA for the prediction of snow avalanches. [5]
The dataset wich is used in that study is created with three different main categories of data, including a series of meteorological data, a map of the avalanche occurrence and terrain-related data. A snow avalanche inventory map is created for the study by mapping the location of the snow avalanches, wich were collected by field observations, onto google maps and confirming it by field surveys. Topographical[5] data like slopes, elevation, lithological and morphometric structures and 91 avalanche line locations were recorded while this process. The training of the machine learning models is done by using these locations as the dependent variable and 14 features as predictor variables. The Data is divided by an 70 to 30 ratio into a training and a testing set. To analyze the relative performance of the variables, the jackknife method is used, where the machine learning models are trained iteratively as many times as there are features in the set and one variable is removed at each iteration. This allows to investigate how important the removed variable is for the prediction quality. [5]
Resulting from the jackknife method, the performance of the MDA is more sensible than the SVM, as the performance of the MDA decreased by removing one feature from the training data. Another result of the study is that avalanches seem to slide out mainly in the vicinity of streams. In addition, Multivariate Discriminant Analysis performed better in predicting avalanche danger compared to Suppport Vector Machines. Both methods produced results with an accuracy of 0.83 for the Suppport Vector Machines and 0.85 for the Multivariate Discriminant Analysis. the study also showed for both algorithms, that the altitude has no relevant effect on the prediction of avalanches. The recorded locations had a sea level between 1911m and 3396m. [5]

Thomas Thüring, Marcel Schoch, Alec van Herwijnen and Jürg Schweizer Thüring, Schoch, van Herwijnen, et al. [6] also attempted to predict avalanches automatically using supervised machine learning methods. The goal of the study is to determine by machine learning whether the infrasonic sensor[6] array data belongs to an avalanche or not. [6]
The data area for the case study is around Lavin, switzerland in the east swiss alps for the winter of 2011-2012. Also in this study a Support Vector Machine are trained with the data. For the training set they used 26 days with 29 avalanche events and for the test set 73 days with 30 avalanche events based on the information of an infrasonic sensor. For the evaluation and optimization of the Support Vector Machine model, a 10-fold cross validation is used by the

[5]Topography is a branch of cartography and describes the characteristics of the earth's surface [13].

[6]Infrasonic sensors are avalanche detection systems, wich can monitore avalanche activity of artificially triggered or natural avalanches in the range of 3 to 5km [14].

authors of the article. [6]
The authors managed to reduce the false predictions, compared to the threshold-based classifier, wich is provided by the manufacturer of the infrasonic sensors, by training the SVM from 65% to 10%. The classification accuracy of the Support Vector Machine is 91.4% for this study. [6]

Pozdnoukhov, A. and Purves, R.S. and Kanevski, M., attempt to create forecasts for a period of several days for avalanches probabilities, by the use of machine learning methods in their article "Applying machine learning methods to avalanche forecasting" Pozdnoukhov, Purves, and Kanevski [9].
Just like in the studies mentioned before, the authors of this study use Support Vector Machines as classifier. For the study, the authors compare the performance of the SVMs to the established Nearest Neighbors models. [9]
The Dataset, wich is used to train the algorithm refers to an area around Lochaber, Scotland, UK, on wich all mountains are below 1300m elevation. The set is a combination of ten meteorological and snowpack related variables on the basis of daily measurements with data from two previous days. This combination creates a set of 30 features. For the creation of the dataset, the avalanche forecasting expert for the Lochaber region is also consulted and a number of expert features were added to the dataset. The final dataset consisted of 44 features. After data preprocessing and iterative feature selection with the SVM dataset is then reduced from 44 to 20 predictor variables and 1835 samples. In total the dataset included about 700 avalanches for 49 avalanche lines. [9]
For the training of the SVM, the dataset is divided into a train and a test set. The train set is the data for the winters from 1991-2000, wich included 1123 samples. The test set included 712 rows for the winters of 2001-2007. The performance of the Support Vector Machines is compared to Nearest Neighbors machine learning methods, wich are established methods for this field. The predicted probabilities of the SVM are close to the empirical probabilities for avalanche events. This observation can be seen especially at high probabilities. The lower the probabilities are, the less accurate they are. For another experiment, the authors added for each of the 49 avalanche lines the factors: constant meteorological and snow cover for the region, altitude, aspect ratio and gradient to each day in the data set. This was done to undertake a spatial avalanche forecast and project it onto a digital elevation model of the region. [9]
The conclusion of this study mentioned, that the Support Vector Machine at its optimum thresolds of 0.5 is broadly comparable to the Nearest Neighbors models, applied to the same dataset, but not significantly better. On the other hand, a finding of the study was that, the Nearest Neighbors models need a high number of neighbors to deliver as good results as the SVM. For this reason, the Nearest Neighbors method produces similarly good results as the SVM when 20 nearest neighbors are used, but shows a significant decrease in performance when 10 or fewer nearest neighbors are used. This can be attributed to the high dimensionality of the data needed for avalanche prediction. Since the Nearest Neighbors method is worse at handling high dimensional data compared to the SVM. For this purpose the authors see the potential of Support Vector Machines in future approaches of predicting avalanche events. [9]

Anuj Tiwari, Arun G., Bramha Dutt, Vishwakarma Anuj Tiwari [4] investigate in the context of their paper how the effectiveness of SVMs in predicting avalanches is improved by parameter importance assessment (PIA)[7]. Similar to the study of Bahram Choubin, Moslem Borji, Amir Mosavi, Farzaneh Sajedi-Hosseini, Vijay P.Singh, Shahaboddin Shamshirband Bahram Choubin [5], the main aim of this study is to generate a very accurate Avalanche susceptibility map (ASM)[8], which indicates the avalanche susceptibility, out of a prediction model based on a SVM and a PIA. [4]
The authors define the idea behind the study as follows:

> "The underlying hypothesis is that non-linear relationships between past avalanche occurrences and influencing parameters can be used for avalanche susceptibility modeling. " [4]

The study area for the case study is an avalance prone area in India which covers the greater Himalayan mountain range. The area has a mean elevation of 4430m above sea level and includes glacier covered mountains. The avalanche inventory map, wich is used for the training and validation of the machine learning models, includes 114 avalanche locations and the same number of randomly picked non-avalanche locations wich are picked from a set of location like urban settlements, water bodies, or crop land, where no avalanches can occur. The authors split the created avalanche inventory map into a randomly selected 70 to 30 ratio train and validation set. The training data for the study is a combination of eleven parameters, wich are of the meteorological and topographical data types. [4]
To get the parameter importance, the Boruta[9] algorithm was used as PIA. As mentioned before, trained machine learning algorithms in the study SVMs with linear, poynomial, sigmoid and RBF kernel functions. For the Validation of the study, the authors chose the Receiver Operating Characteristic[10] (ROC) curve and the Area Under the Curve[11] (AUC) statistic. The combination AUC-ROC technique is used to rate the accuracy of the algorithm results. The AUC statistic showed [4]
the resulting values of auc statistic showed respectively the use of selected parameters (the second value) the following values for the kernel functions: linear 88.2%, polynomial 91.6%, sigmoid 46.3% and RBF 91.5% for the use of all features and linear 88.0%, polynomial 92.1%, sigmoid 44.6% and RBF 93.4% when using selected parameters. The results of this study in terms of the importance of each parameter for the creation of an ASM show different results to the very similar study from article "Snow avalanche hazard prediction using machine learning methods" Bahram Choubin [5]. For example, in this study elevation is ranked as the third most

[7] "Parameter importance assessment (PIA) or feature selection (FS) is an essential step in susceptibility modeling applications. By eliminating irrelevant, and noisy parameters from the input datasets, PIA solves the issues of redundant information processing and enhances the accuracy of the model" [4]

[8] "Avalanche susceptibility map (ASM) is one of the essential information in spatial planning for avalanche prone areas. It gives a description about spatial probability of avalanches." [4]

[9] "Boruta is a parameter importance assessment (PIA) and influence analysis algorithm.[...]. With RF as the fundamental instrument method, Boruta integrates the association between input parameters and iteratively eliminates the unimportant ones." [4]

[10] "ROC is one of the most preferred techniques for describing the quality of susceptibility modeling techniques (Fawcett, 2006). It depicts the true positive rate on the y-axis and the false positive rate on the x-axis." [4]

[11] todo

important parameter, while in the similar study elevation is ranked as the last of the series. [4] [5]

Hong Wen and Xiyong Wu and Xin Liao and Dong Wang and Kaiyang Huang and Bernd Wünnemann Wen, Wu, Liao, et al. [15] applied a set of machine learning methods for the purpose of snow avalanche susceptibility mapping. They used an area in Parlung Tsangpo in souteastern Tibet for their case study and collected a set of 381 snow avalanches through seven field investigations. The Samples were marked with the coordinates of the location and the range of the starting points of the avalanches. The 381 locations were divided into a training set of 305 avalanche locations and a validation set with 76 locations. This makes a 80% to 20% train test ratio. The authors also created a set of sample considering the polygon attributes with 27596 points in 120m intervals within the starting zones and the same number of random non-avalanche samples, picked from outside of the avalanche zones. A number of specific conditional and topographical variables, such as Elevation, Slope, Aspect, Roughness, average annual snow fall, average temperature in January Maximum snow depth and the distance to rivers were added to the set of starting zones, to be able to present a generalized form, of the many factors that influence the complex process of the formation of a snow avalanche. [15]
For the modeling, the team used the four machine learning methods: Support Vector Machine (SVM), K-nearest neighbors[12] (KNN), Classification and Regression Tree[13] (CART) and Multilayer perceptron[14] (MLP). [15]
The importance of the conditional factor features were calculated by training a SVM with the samples. as a result, all features with an importance of less than 3%, such as the average annual snowfall days, were removed from the sample set. The authors also divided the whole study area into 845263 rasters of a 120m x 120m size and trained the machine learning algorithms on the whole set. The prediction results of the algorithms were imported into a GIS and projected onto maps of the study area. [15]
The performance of the models have been evaluated by the Kappa coefficient[15] and the ROC curve. The performance evaluation of the machine learning models with the Kappa coefficient showed that all models made good predictions. According to the order of prediction quality resulting from the evaluation with the Kappa coefficient, the SVM performs best, followed by the CART, third the MLP and last the KNN. In the evaluation with the AUC statistic, the result was that the SVM was again in first place, followed by the MLP, in third place the KNN and

[12]"KNN is a method to classify observations according to the similarity between observations and other observations, which is a mature method in theory. The idea of this method is: if most of the k most similar samples in the feature space belong to a certain category, then the sample also belongs to this category." [15]

[13]"The Classification and Regression Tree algorithm is a decision tree that can be used to predict or classify future observations." [15]

[14]"MLP is a feed-forward artificial neural network model, which maps multiple input data sets to a single output data set. Each node in the neural network is a perceptron, which models the basic function of neurons in the biological neural network." [15]

[15]"Kappa coefficient test is a method of using confusion matrix to test the consistency between model results and actual observations, also known as consistency test. Kappa coefficient test is to use confusion matrix to calculate kappa coefficient, the coefficient is between 1 and 1, usually greater than 0. The larger the value, the higher the accuracy of the evaluation model is." [15]

in last place the CART, whereby the first three models achieved an AUC value of over 0.9 and the CART a value just below 0.9, and thus high values were achieved for all. As conclusion, the SVM is, with an AUC value of 0.918 the most robust machine learning model of the study and the authors concluded that their approach, for the creation of avalanche susceptibility index maps, achieves accurate and useful results and this method is promising for future applications. [15]

In The Paper "A data efficient machine learning model for autonomous operational avalanche forecasting" Chawla and Singh [16] from the year 2021, Chawla, M. and Singh, A. describe their study about their approach for an data efficient forecast of snow avalanche events by the use of a Random Forrest machine learning model. The model is supposed to perform a binary classification into the classes "avalanche" and "non-avalanche", based on snow parameters and meteorological variables. The geographical area of this study is the Bandipore-Gurez (BG) sector at the tip of the north-west Indian Himalayan Mountains. The area observation includes over 100 avalanche paths, whose starting zones are located at the altitude between 2350m and 4800m above sea level. The authors got their Snow- and Meteorological parameters from a snow-meteorological observatory, wich is located new Kanzalwan at an elevation of 2440m. The dataset created by the team was assembled from two tables. The first table contains the meteorological and snow related features provided by the observatory. The second table consists of additional parameters derived from the features in the first table. The features from the second table represent the snow related features, the number of avalanches occurred and meteorological data from the last two to ten days. The dataset used to train the algorithm includes the three winter seasons from 2010 to 2013 of the meteorological, snow and avalanche event features. The authors want to demonstrate the data efficiency of their approach with the size of the sample set used to train the model. Different to the most studies mentioned in this chapter, the size of the test set in this study is larger than the training data sample set. It includes the four winter seasons from 2013 to 2017. [16]
The forecast, which was made for the study, distinguishes for the whole area into two classes 0 for non-avalanche days and 1 for avalanche days Chawla and Singh [16]. Therefore, no slope specific forecasts are made, but similar to the study from Davos in Switzerland Stephan Harvey [10], avalanche days are predicted for the entire area.
The authors describe in their paper a problem they encountered in their original data set. The problem is that the number of avalanche days (25%) is much lower than the number of non-avalanche days (75%). From this unbalanced distribution they conclude that the classification system model will have a bias to predict more non-avalanche days. The paper describes to solve this problem. The first is a cost-correcting classifier that gives higher weight to the samples from minority classes. The second approach they describe randomly removes samples from the majority class or artificially adds entries to the minority class. However, according to the authors, this can lead to overfitting. The method, used in the study is based the second mentioned technique but instead of removing random samples of the majority class, the authors removed samples based on domain specific knowledge. So as a result all data rows, in wich are avalanche events are unlikely because of an absence of snow cover were removed from the training set. As an example the authors removed all rows where the snow height was below 0.50m. The consequence of this filtering method is that the classification can also not be ap-

plied to days with a snow height below 0.5m. [16]
The Random Forest classification model is trained and validated by a five fold cross-validation. The authors optimized the two input parameters of the Random Forest model (maximum depth and number of trees) by performing a series of experiments in which every possible combination of the values 2, 3, 4, 5 for maximum depth and 2000, 5000, 10000, 20000 for number of trees were performed on the model. Based on the results of the test series, the authors used 3 for maximum depth and 5000 for the number of trees parameters. [16]
One result of the study showed, that the snow height, the new snow fallen in the last ten days and the wind speed were important indicators for the forecast of an avalanche-day. In summary, the authors found that the Random Forrest is suitable for the creation of an autonomous data-efficient snow avalanche forecast and provides good results. [16]

3 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.

3.1 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. [17] [18]
The three main problems of datasets for machine learning are the following [17]:

- Missing Data
- Noisy Data
- Inconsistent Data

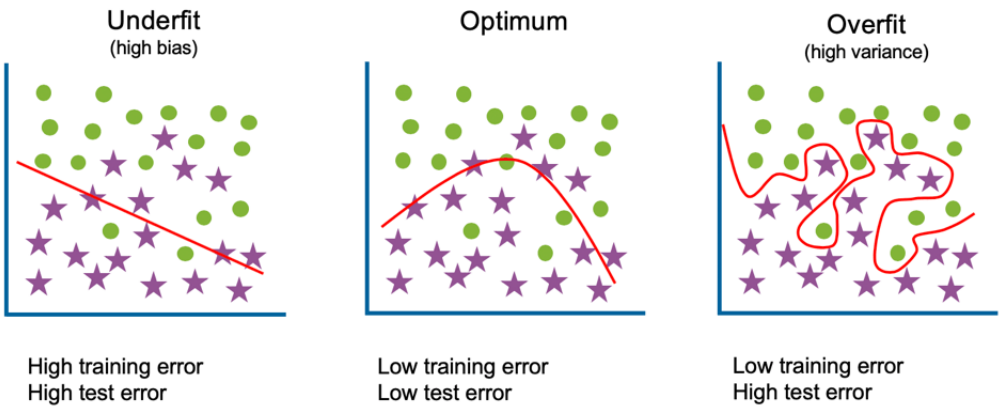
At first to keep the values in the dataset, for missing data the two options are ignoring them or fill them manually or with a computed value Baheti [17] Kotsiantis, Kanellopoulos, and Pintelas [18]. In the context of this work, the samples with missing information were mainly ignored and Manuel excluded 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. [18]
According to Pragati Baheti, the only option for inconsistent data is to remove it from the data set Baheti [17]. 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. [18]
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. [19] This is calculated for a sample x as follows:

$$z = \frac{(x-u)}{s} \quad [19]$$

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. [19] 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. [19] 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. [19] As mentioned in the article "Data preprocessing for supervised leaning" Kotsiantis, Kanellopoulos, and Pintelas [18], 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.

3.2 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. [20]
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. [20] 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 Subasi [7]. 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 learn-



Source: [20]

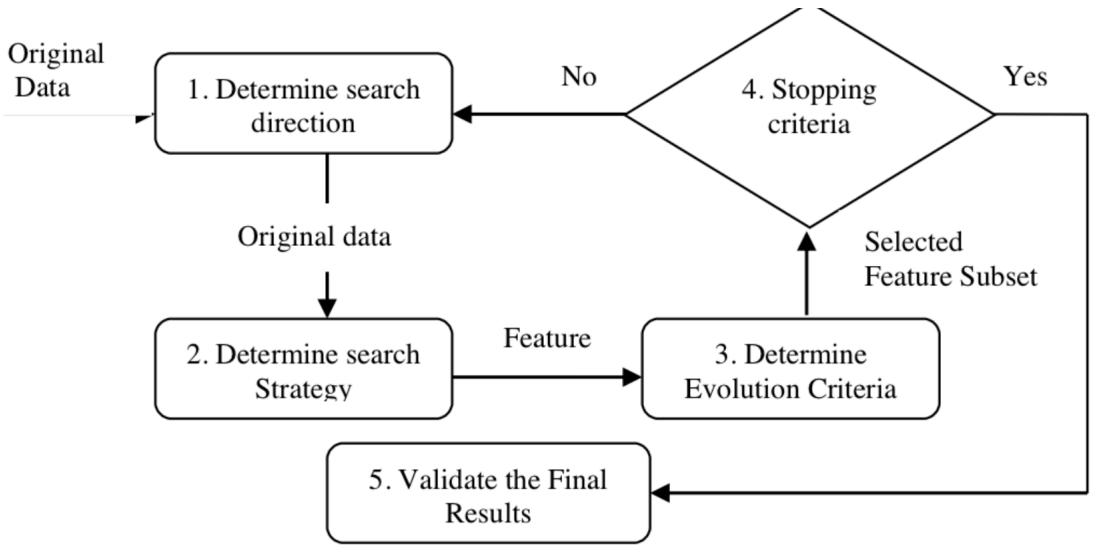
Figure 1: Underfitting, optimal, overfitting machine learning model.

ing algorithm. Therefore, it can help to focus on a small subset of really important features. [21] Allam and Nandhini [22] Venkatesh and Anuradha [23] 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. [21] 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 Venkatesh and Anuradha [23]. This does not mean that this feature does not contain useful data. It only indicates that it is not statistically related to other features Venkatesh and Anuradha [23]. A good feature Selection can help to get much better predictions from the machine learning models and decrease the error rate Cai, Luo, Wang, et al. [21]. 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. [22] The popular approaches to do this, are models, features quality measures, feature evaluation, search strategies and combinations of these Venkatesh and Anuradha [23]. Depending on how the training set is labeled, supervised (fully labeled), unsupervised (unlabeled) and semi-supervised (partially labeled) feature selection methods are used Cai, Luo, Wang, et al. [21]. 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.
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. [23] Cai, Luo, Wang, et al. [21] Pisner and Schnyer [24]

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. [23] Cai, Luo, Wang, et al. [21] 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. [24] 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. [21] 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. [23] Cai, Luo, Wang, et al. [21] 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 Cai, Luo, Wang, et al. [21].



Source: [23]

Figure 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. [23] 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. [23] 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 "Snow avalanche hazard prediction using machine learning methods" Bahram Choubin [5], 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" Anuj Tiwari [4], 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 Choubin [5] Anuj Tiwari [4]. 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.

3.2.1 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 Subasi [7] 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." [25]

These machine learning model is a representation method based on knowledge about the features of a dataset to represent classification rules Sugumaran, Muralidharan, and Ramachandran [26]. 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 [25]. For this reason, decision trees are also often preferred to other methods that actually provide accurate results Subasi [7]. A standard decision tree starts with a root node, does have some branches as well as child nodes and leaves Sugumaran, Muralidharan, and Ramachandran [26]. 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. [7] 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. [26] 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. [7] 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 [25]. 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 [25]. 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. [26] 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" Sugumaran, Muralidharan, and Ramachandran [26] Decision trees are used for the feature selection. Similar to this work, the study was concerned with a classification problem.

3.2.2 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 Cai, Luo, Wang, et al. [21]. 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 Lu, Zhao, and Zhang [27] . 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. [27] Liashchynskyi and Liashchynskyi [28] Genteic algorithms are also able to handle huge dimensional datasets efficiently because of their exploitational and explorational characteristics Lu, Zhao, and Zhang [27]. 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 fit-

ter 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. [28] Tao-Chang Yang [29] Lu, Zhao, and Zhang [27]

Jianjiang Lu and Tianzhong Zhao and Yafei Zhang Lu, Zhao, and Zhang [27] 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 Lu, Zhao, and Zhang [27]:

$C_1^t+1 = w_1, k, C_2^t+1 = w_2, k$, 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 Lu, Zhao, and Zhang [27]:

$$w'_k = \begin{cases} w_k + \Delta(t, 1 - w_k), & \eta = 0 \\ w_k + \Delta(t, w_k), & \eta = 1 \end{cases}$$ Lu, Zhao, and Zhang [27]

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 $[0, \gamma]$ Lu, Zhao, and Zhang [27].

$$\Delta(t, \gamma) = \gamma(1 - r^{(1-\frac{t}{M})})$$ Lu, Zhao, and Zhang [27]

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. [27]

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. [28] 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" Tao-Chang Yang [29] 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 [30] is used.

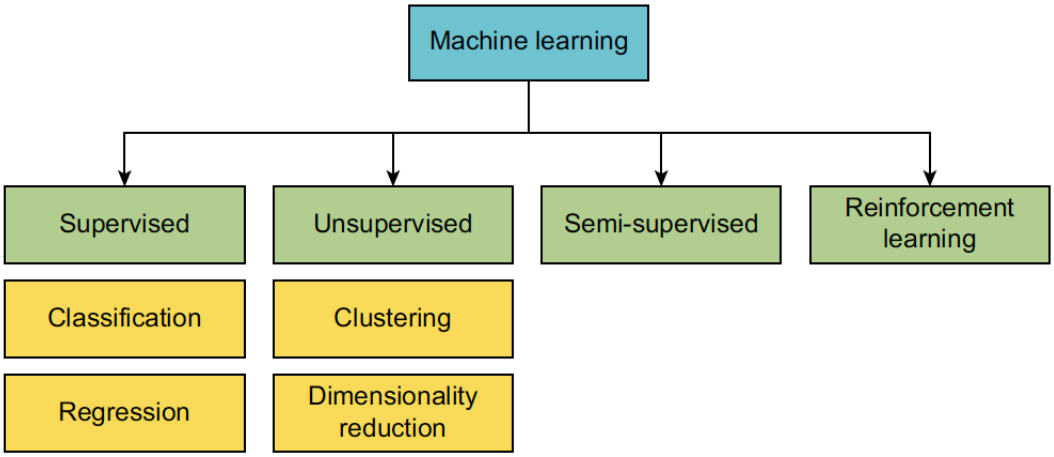
3.3 Machine Learning

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. [31] [7] Abdulhamit Subasi describes the initial goals of machine learning in his book "Practical Machine Learning for Data Analysis Using Python" as 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." [7]

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. [31] 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. [7]
As shown in figure 3, in Machine learning there are four base types of models: supervised, unsupervised, semisupervised and reinforcement learning. [31] [32] [24]

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. [31] [32] As a comparison for the 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. [31] As illustrated in figure 3 supervised learning is divided into two subcategories, classification and regression.



Source: [24]

Figure 3: Machine learning model learning types.

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 Vieira, Lopez Pinaya, and Mechelli [31] Subasi [7]. For example, the two classes that are in the context of this work are: ”an avalanche is going down” and ”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. [31] [7] 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 [32].

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.[32] [31]

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. [32][31] 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. [33] [31]

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 Choubin [5] Anuj Tiwari [4] Pozdnoukhov, Purves, and Kanevski [9]. 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 Choubin [5]. 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.

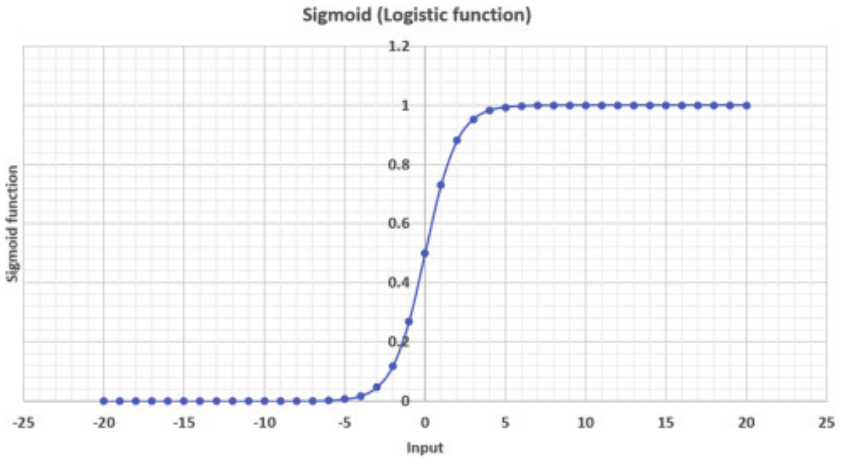
3.3.1 Logistic Regession

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. [34] 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. [35] [34] „Beginners Take: How Logistic Regression is related to Linear Regression“ [36] The name ”Logistic regression” can be misleading, because it is more a classification model than a regression model Subasi [7]. 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. [34] „Beginners Take: How Logistic Regression is related to Linear Regression“ [36] Belyadi and Haghighat [35]
The sigmoid function also known as logistic function:

$S(x) = \frac{1}{1+e^{-x}}$ „Beginners Take: How Logistic Regression is related to Linear Regression“ [36] Belyadi and Haghighat [35]

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. [36] Belyadi and Haghighat [35]



Source: [35]

Figure 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. [34] Subasi [7]
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. [37] Belyadi and Haghighat [35]
Logistic regression models are easy to realize while they are achieving good results for binary and linear classification problems Subasi [7]. 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 Nusinovici, Tham, Chak Yan, et al. [37]. The Python library Scikit-learn does have a optimized version of the logistic regression algorithm, wich also can handle dens as well as sparse input by including a set of regularization methods to convert any format to 64-Bit floats for the optimal performance of the algorithm[38]. In context of this thesis, the scikit-learn implementation is used.

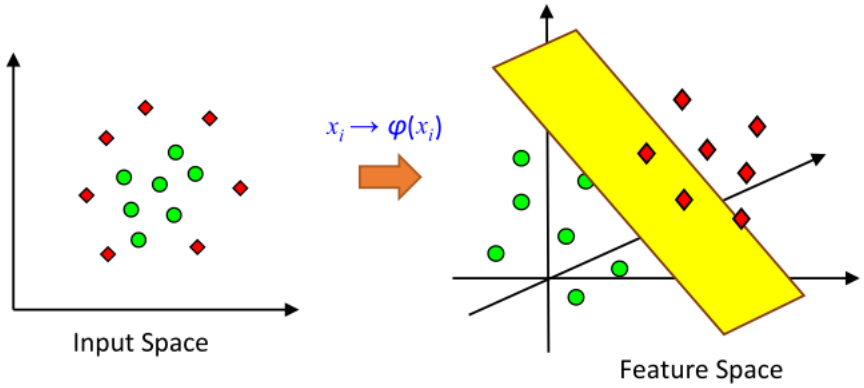
3.3.2 Support Vector Machine

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 [32] Sugumaran, Muralidharan, and Ramachandran [26] Gholami and Fakhari [39]. They can be used for classification and regression problems as well as to detect outliners [32] [40]. Abdulhamit Subasi gives a general overview about SVMs in his book "Practical Machine Learning for Data Analysis Using Python" as follows:

> "Support vector machines (SVMs) are one of the main machine-learning algorithms that are not only accurate but also highly robust." [7]

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. [7] 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. [26] Support vector machines can be linear and non-linear, but classification problems are in most cases linear, therefore mostly linear SVMs are in use Pisner and Schnyer [24].
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. [39]

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



Source: [39]

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

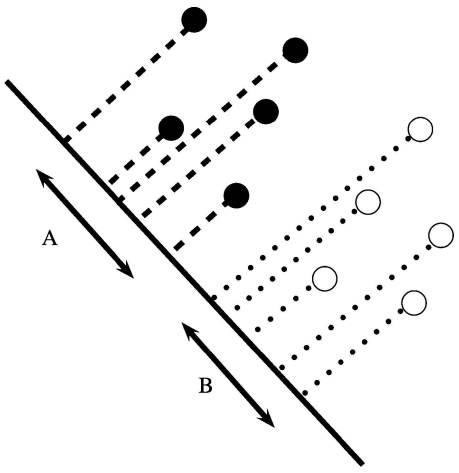
of current and future data, the most extreme margin of the hyperplanes is determined. [7] The search for the maximum margin is minimizes also the generalization error of the SVM. [26]
A larger margin allows better generalization and a hard margin is the simplest way with the least computation time, but 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 ”soft 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. [24] Gholami and Fakhari [39]
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. [7] [40]

As shown in Section 2 ”Related Work”, SVMs performed well in previous studies in the context of avalanche event prediction and avalanche hazard mapping. [4]–[6], [9] 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) [41] . 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. [41] In the case of this study, however, the number of samples is less than 10000 and the implementation can be used.

3.3.3 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 Demir and Ozmehmet [42] Dash [43] Xanthopoulos, Pardalos, and Trafalis [44]. R. Fisher used the LDA to differenciate between two different types of plants Xanthopoulos, Pardalos, and Trafalis [44]. 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. [45] [5] [44] [7] [43]. 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. [45] [5] [7] [44] 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 Dash [43] Demir and Ozmehmet [42]. 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

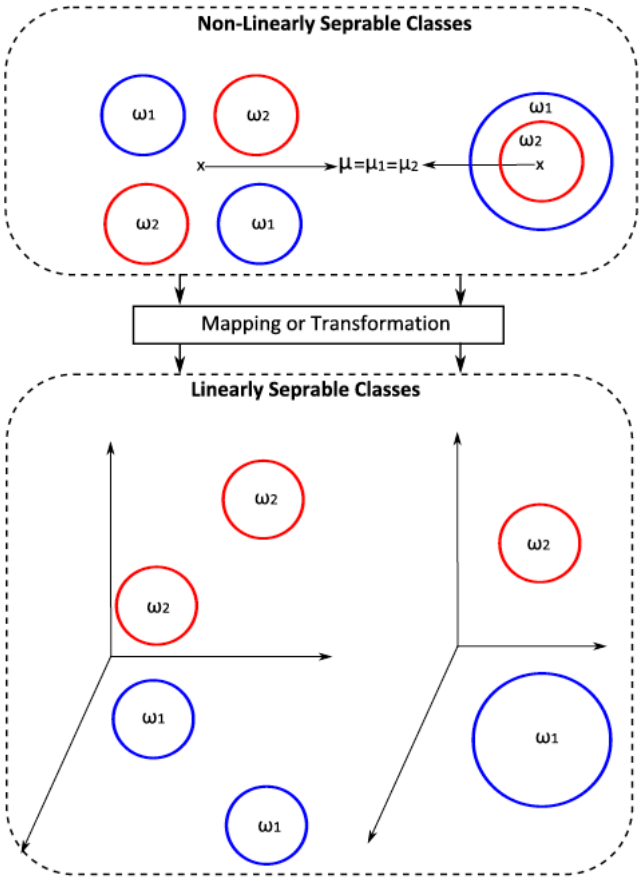


Source: [44]

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

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. [46]

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. [46]
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. [46]



Source: [46]

Figure 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.

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. [45] [43] [46] 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. [46]
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 Choubin [5].
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. [47]

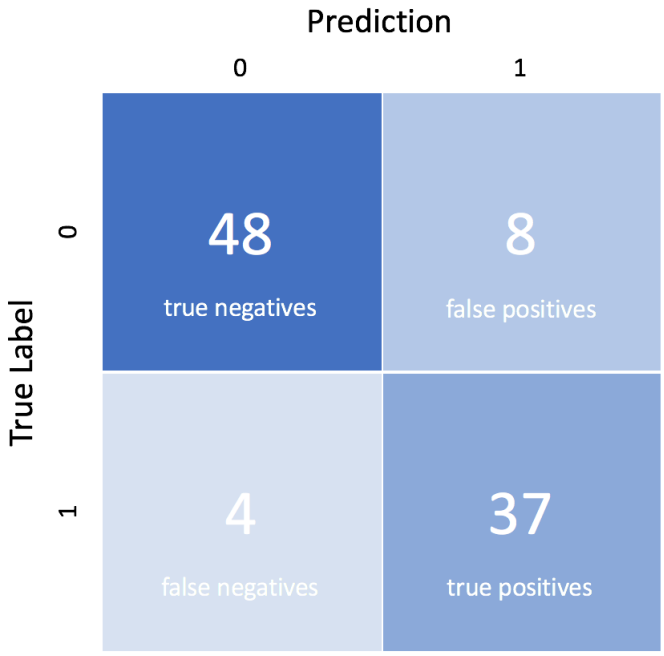
3.4 Performance Evaluation

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, 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. [48]
The metrics used to evaluate supervised machine learning models are also divided into evaluation of classification and regression models JORDAN [49]. 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 JORDAN [49]:

- 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. [49]



Source: [49]

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

3.4.1 Accuracy

The Accuracy score is one of the most common evaluation metrics, when it comes to the evaluation of binary classifiers Nighania [50]. Also a lot of the related works use this score to evaluate and compare their machine learning models. Jeremy Jordans defines accuracy as follows:

> "Accuracy is defined as the percentage of correct predictions for the test data." [49]

The definition can also be represented like this:
$Accuracy = \frac{Numberofcorrectpredictions}{Numberoftotalpredictions}$ [51]

For binary classifiers accuracy is can be calculated as follows:
$Accuracy = \frac{TP+TN}{TP+FP+TN+FN}$ Nighania [50]

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 Nighania [50]. 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. [51]

For datasets, wich classes are not balanced, the pyhton library Sklearn.metrics includes a more balanced version of the accuracy score called balanced accuracy score. For a binary classification, this score weights the class wich is less represented in the datasets more than the other one. This can indicate if the machine learning algorithm is biased by the more represented class of the dataset. [52] This balanced accuracy score is calculated for binary classification problems as follows:

$$BalancedAccuracy = \frac{1}{2}(\frac{TP}{TP+FN} + \frac{TN}{TN+FP}) \quad [52]$$

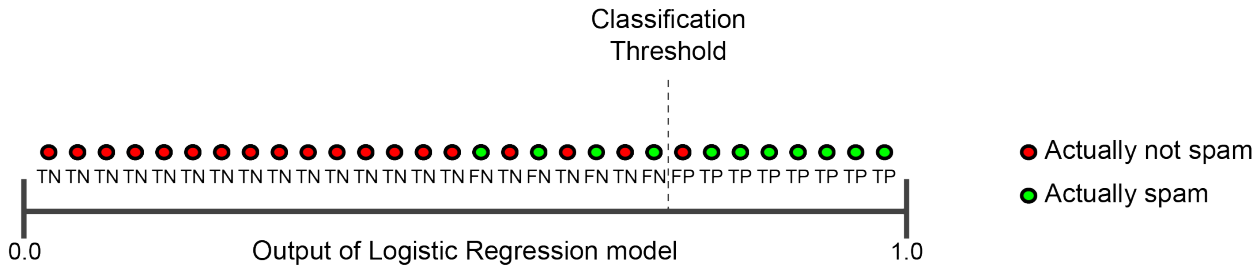
In the case of a biased machine learning model, the normal accuracy might be higher value than the balanced one. The range of the resulting score is between 0 and 1. The score can be performed on binary and multiclass classification problems. The score is defined as the average of the recall score, explained in section 3.4.3, calculated for ever class of the dataset.[52]

3.4.2 Precision

The Precision score is the percentage of how many positive predicted observations actually are positive Nighania [50].
It is defined as follows:

$$Precision = \frac{TP}{TP+FP} \quad \text{Nighania [50]}$$

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 [53]. 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.



Source: [53]

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

3.4.3 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 [53]. In figure 9 the recall is represented as all green marked datapoints 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:

$$Recall = \frac{TP}{TP+FN}$$ Nighania [50]

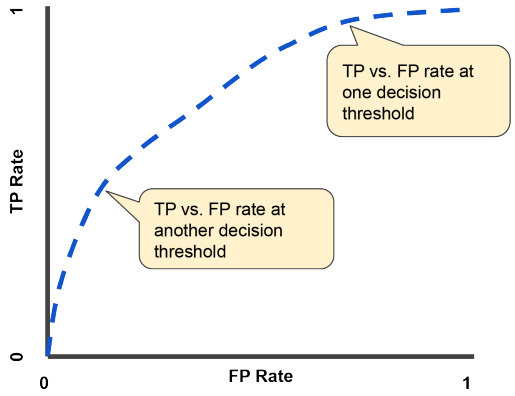
3.4.4 ROC-Curve and AUC

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 [54]. 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:

$$TPR = \frac{TP}{TP+FN}$$ [54]

$$FPR = \frac{FP}{FP+TN}$$ [54]

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. The perfect ROC-curve does go straight up to 1 and then straight to the right. So in this case the model would predict the samples perfectly. [54] [50]
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 Srivastava [48].



Source: [54]

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

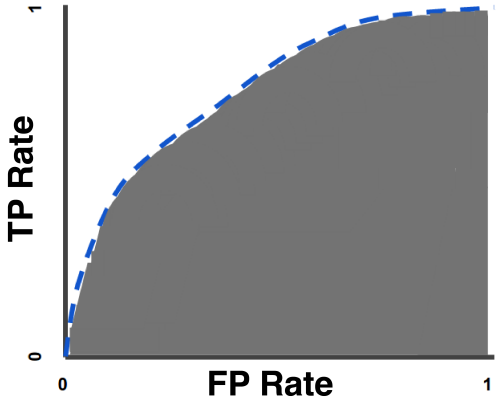
An interactive approach, in which a classifier model is evaluated many times with different thresholds, would be associated with high computational costs. However, there is also a more efficient approach called AUC, which can also determine this information. [54]

The AUC (Area under the Curve) statistic represents the integral calculus of the ROC curve 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 100% false. If the value is 1.0, all predictions are correct. If the value is below 0.5 the predictions of the model are below chance, because chance is represented by the value of 0.5 [48], [54] The value The higher the numerical values of the AUC statistic the better is the models performance Srivastava [48]. 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 Srivastava [48].

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. [55]

3.4.5 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. [56]
In the k-fold cross-validation, the dataset is split into k equally sized subsets. After that the

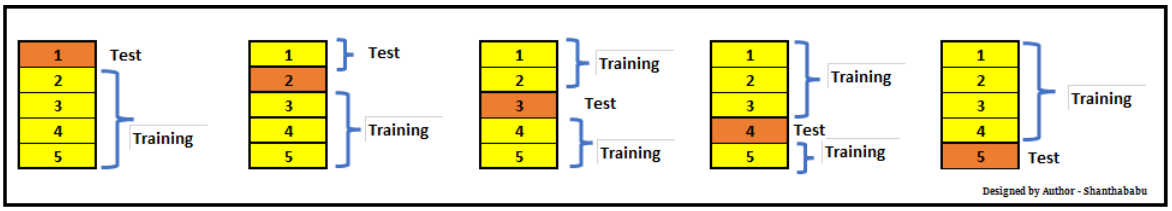


Source: [54]

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

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. [56]

The 10-fold cross validation is a popular variant in terms of machine learning. the decisive

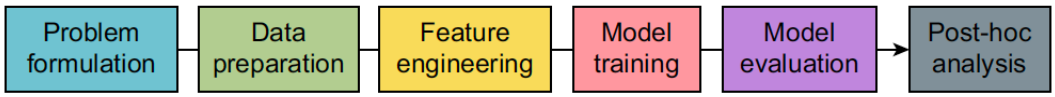


Source: [57]

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

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. [57] 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.

4 Results



Source: [58]

Figure 13: The six steps of the supervised machine learning pipeline.

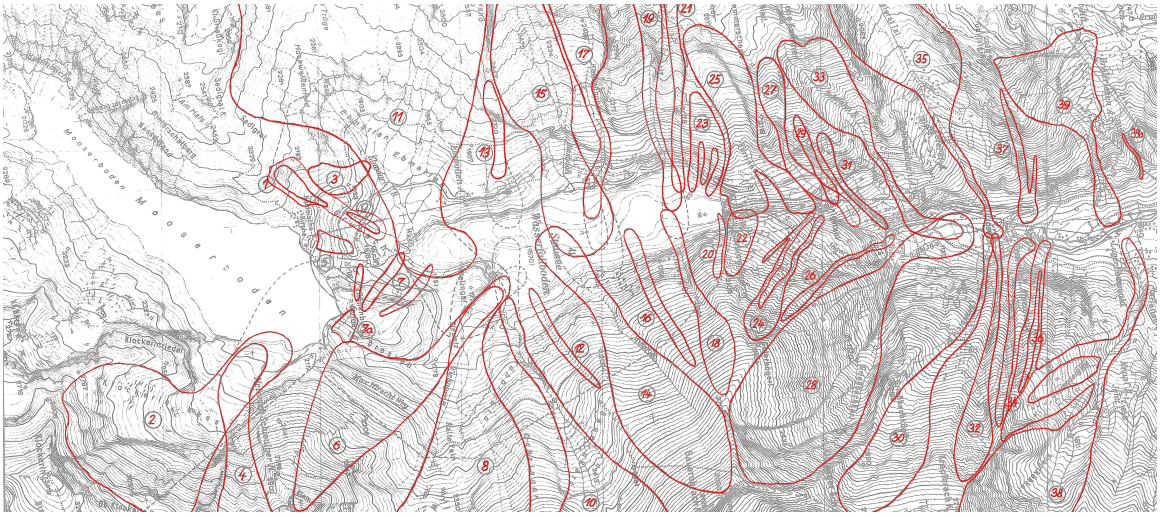
This chapter describes the actual study including the results. The study is divided into different steps. The supervised machine learning pipeline presented in Figure 13 shows the six steps of a study using supervised machine learning models. It starts with the step of defining the Problem or Goal of the study. The problem to be solved in context of this master thesis is to predict snow avalanches with meteorological dan snow pack related data for topographical defined mountain slopes and answer the two Research questions mentioned in chapter 1. The problem falls into the category of binary classifications problems, since it has the two predictive values ”Avalanche” and ”Non-avlanache”. The second phase is about data preparation. In the case of this work it starts with information about the data owner and the study area. Followed by the composition of the data set as well as its data preprocessing. Every features in the resulting data set is described. The feature selection process, which in this case consists of two steps, is also discussed. In the first step, a decision tree is used to select a number of meteorological features for which additional columns for data from the past days are added to the data set. In the second step, a genetic algorithm is executed on the three machine learning models Logistic Regression, LDA and SVM to calculate the optimal feature subset for each algorithm. The training phase of the three supervised classification machine learning models is described together with the model evaluations in section 4.3, because they represent the results of the model training phase.

4.1 Data

As mentioned in chapter 3 about machine learning, the acquisition of high quality data is an important and challenging step in the creation of machine learning models, since the quality of the data is directly related to the quality of the models Subasi [7]. Especially in the context of predicting avalanche events, the step of collecting homogeneous data and preparing a dataset is a challenge, wich is caused by the fact that avalanche events depend on many different variables. The factors that have a significant influence on avalanche release can be divided into three main categories: terrain, snowpack-related and meteorological data.[5] Data from the three categories were used for this case study, wich does not mean that the used data includes all relevant factors. Proximity to watercourses, for example, can be a terrain-related factor affecting avalanche release Bahram Choubin [5] and is not included in this study data. The process of preparing the dataset, as well as the origin and composition of the data is explained in this chapter.

4.1.1 Origin of the data

The data used for the case study of this paper were provided by the in-house avalanche warning service of the energy company VERBUND AG, headquartered in 1010 Vienna, Austria. Verbund represents Austria's largest and most environmentally friendly energy company and is one of the largest producer of electricity from hydropower in Europe. It was founded in 1947 with the name Österreichische Elektrizitätswirtschafts-AG. The hydropower plant in Kaprun was one of the first two power plants operated by Verbund AG. [59] The companies strategy slogan "With our power to a green future" [59], shows the company's intentions to promote environmentally conscious and future-oriented energy production. The company states that it obtains almost 100% of its electricity from renewable sources and 95% especially from 129 hydroelectric power plants in Austria and Germany. The electricity is mainly generated from hydro, wind and photovoltaic power plants. In addition, this is supported by gas-fired thermal power plants. The companies goals is to produce completely CO2 free electricity. [59] The companies strategy slogan "With our power to a green future" [59], shows the company's intentions to promote environmentally conscious and future-oriented energy production. The company-owned avalanche warning service of the storage power plants in the Hohe Tauern is located in Kaprun, Austria. It was initiated in 1956 due to some heavy avalanches during the construction of the power plants in order to ensure the protection of employees as well as plant and operational safety.



Source: Avalanche warning service of Verbund AG in Kaprun

Figure 14: The avalanche map of the avalanche warning service of Verbund AG in Kaprun shows the 39 avalanche lines.

The data were recorded in the vicinity of the Mooserboden storage power plant for 39 avalanche paths. The paths are represented in the avalanche map shown in figure 14. The map was created by the avalanche warning service of Verbund AG in Kaprun. It shows the avalanche prone zones of the study area, to give the avalanche warning service an orientation for their forecasts of the day. The background for the exceptionally accurate recordings of avalanches is the need for the most accurate possible prediction of avalanches in the 39 avalanche lines around the power

plant. In the moths from May to Oktober the area is also a touristic spot managed by the Verbund AG[60]. Accurate forecasting is of great importance to ensure the safety of the employees working in the areas of the avalanche lines. The power plant is part of the Kaprun power plant group, which includes both pumped and storage power plants. The power plant group is operated by Verbund Hydro Power and is located in Salzburg on the edge of the Hohe Tauern at 2040 meters above sea level and is surrounded by the over 3000 meter high mountains of the Glockner group including the Großglockner, wich is the highest mountain in Austria. [60]

4.1.2 Meteorological Data

Meteorological data are one of the most important factors to initiate snow avalanches Bahram Choubin [5]. The meteorological data used for this case study were provided by the avalanche warning service in Kaprun in the form of table Mooser_Wetter_Daten. This data table includes meteorological for each day in the months from november to may in the period from 1953 to 2022. Not all columns of the table are complete and some are not filled at all, therefore mainly columns that will be used in the further course of the work are mentioned in the following. The table contains data about the date as well as the winter season, the snow height, the precipitation, the air temperatures at the times 7:00, 14:00 and 19:00 as well as the snow temperature, the wind direction, the wind force, the snow sinking depth, the day weather as well as the weather from the day before, the snow depth, the clouds, the new snow, as well as the avalanche degree. The data from this table has been collected manually and homogoeneous by the team of the avalanche warning service. All of this factors can affect the triggering of snow avalanche events. How relevant the individual features are for the predictions with the three machine learning models used in this study, is described in more detail in the remainder of this thesis.

4.1.3 Avalanche related data

The table Allg_Lawinen_Abgänge represents general data about all recorded avalanches of the 39 avalanche lines in the area and for the same time period as the meteorological data from the table Mooser_Wetter_Data. The snow avalanches have been manually recorded by the avalanche warning service since its initiation in 1956 to get the possibility to analyze the influence of the different factors on the triggering of snow avalanches and to make a good forecast for the day possible. The table contains data such as the time and date when the avalanche was recorded, the type of avalanche, the old ID of the avalanche line where the avalanche went down, the volume of the avalanche, the general weather conditions at the time of the avalanche, the wind direction and speed, the temperature as well as snow height and new snow fallen since the day before, the danger level on the day of the avalanche and a comment of the person who recorded the avalanche event. The meteorological data shown in this table are not used in this study, because the data from the table Mooser_Wetter_Daten are homogeneous and available for each day of the winter season, also it would be redundant to use both . Another table of the database named kaplawstr contains additional information about the avalanche lines. The old and the new code of the avalanche are the only columns from this table used in context of this master thesis. and mainly to connect tables wich each other.

4.1.4 Topographical Data

Several of the studies mentioned in chapter 2 on Related work indicate that the influence of topographic factors on the occurrence of avalanche events is a non-negligible one, since these data represent the slopes.[4] [5] The topographical data is recorded in a database table called TOPP. This table contains several rows for each avalanche line, wich can be identified by the new avalanche line code. The table includes for each row the new avalanche line code, the mean slope exposition, the minimum, maximum and mean slope, the altitude of the slope as well as the orientation of the slope. The table also contains various other data columns. These are not used in the further course of the work, since they cannot be assigned to the individual avalanche lines in general, but are connected with individual avalanches, which are not allocated to them in the context of this work.

4.1.5 Data Preparation

The database tables Allg_Lawinen_Abgänge (Avalanche related data), kaplawstr (contains the old as well as the new avalanche line IDs), TOPP (Topographical Data) and Mooser_Wetter_Daten (Meteorological data) wich are already described in the previous chapters were merged into a homogeneous data set in the context of this master thesis. This section describes the data preparation process including the merging of the data tables to a homogeneous set and the removal of redundant and not adequately filled data columns.
The tables include data from 1944 to 2022. The recording of the avalanches in the past was not completely homogenous and incomplete. Table one shows a total listing of the sum of avalanches recored per winter season winter for each winter season included in the dataset. This can be shown above all by the fact that in the seasons from 1944 to the season of 1988/1989 in average, there are 22.0769 snow avalanches per season recorded. In comparison, in the seasons from 1989/1990 to season 2021/2022 the average of recorded snow avalanches is 81.636. The table gives an understanding about how less avalanches are recorded per season before the season of 1989/1990 in comparison to the seasons since 1989/1990. With the exception of a few outliers, hardly any avalanches were recorded in these years, and in the cases of the seasons 1971/ 1972, 1976/ 1977 and 1983/ 1984 there were no avalanches recorded at all. In order to increase the homogeneity of the data and decrease the bias caused through the unbalanced ratio between avalanche and non-avalanche samples, wich Chawla, M. and Singh, A. mentioned in their study about an data efficient approach of snow avalanche forecasting Chawla and Singh [16], all data outside the period from season 1989/1990 to season 2021/2022 were removed from the database tables. Subsequently to this measure the kaplawstr table has been merged to the Allg_Lawinen_Catalog table using the old avalanche line ID. This adds the associated new avalanche code and avalanche name to each avalanche, which are used as additional ID. The connection is necessary because the TOPP table, which represents the topographic data for the avalanche routes, does not contain the old avalanche line ID. In the course of this step, all lines that were labeled with the avalanche line name ”all avalanches” also have been removed. These are not included in the kaplawstr table, since this does not represent an exact departure of an avalanche in one of the avalanche lines, but only states that in many of the avalanche lines small avalanches have departed.

```
TOPP = kaplawstr['Code_neu'].apply(lambda x: TOPP.loc[TOPP['Lawinencode']
    == x].mean())
```

Listing 1: calculation of TOPP data for every avalanche line

In the second step, the average values for all columns from the associated avalanches were calculated from the TOPP table for each avalanche line to get representative values for these columns. This is important, because the recorded avalanches from the Allg_Lawinen_Abgänge table can not be directly assigned to the TOPP rows but to the avalanche lines. The python code shown in Listing 1 demonstrates this process.
By this measure, one row is created for each avalanche line. The table contained without this procedure a total of 1905 rows. In the default state, the table could not have been connected to the other data tables. Another way to get only one row per avalanche stroke would be to select a random value for the respective stroke. The reason for taking the average value is that there are not the same number of lines for all avalanche lines and the values of the individual lines per avalanche line do not differ greatly from each other. Thus, the average value represents the entirety of the lines per stroke consistently. The topographic data from the newly assembled TOPP table was then merged to the recorded avalanches in the entire dataset using the new avalanche ID.

Subsequently, these avalanches were assigned to the daily recorded meteorological data of the Mooser_Wetter_Data table by an outer join, so as result there is at least one row per day in the dataset. In cases where several large avalanches have occurred at the same day, the dataset contains one row per avalanche and each Includes the meteorological data for this day plus the topographical data for the avalanche line.

In order to train a machine learning algorithm with the aim of predicting snow avalanches for topographically defined slopes in conjunction with the meteorological data available for this study, the topographical data must also be mapped onto the days without avalanches. The algorithm needs this information, as the data set would otherwise only contain topographic data directly related to avalanches. This would mean that the machine learning algorithm learns that as soon as the topographical data of a slope is available also an avalanche is triggered.

```
for i in gesamt_df.index:
    if(pd.isnull(gesamt_df['meanExpo'][i])):
        sample = TOPP.sample(1)
        gesamt_df['meanExpo'][i] = sample['meanExpo'].values[0]
        gesamt_df['meanSlope'][i] = sample['meanSlope'].values[0]
        gesamt_df['stdDevSlope'][i] = sample['stdDevSlope'].values[0]
        gesamt_df['MinSlope'][i] = sample['MinSlope'].values[0]
        gesamt_df['MaxSlope'][i] = sample['MaxSlope'].values[0]
        gesamt_df['Altitude'][i] = sample['Altitude'].values[0]
```

Listing 2: mapping random sample lines of topographical data onto the rows of non avalanche days

Intervall	Avalanche
1956/ 1957	11
1957/ 1958	4
1958/ 1959	7
1959/ 1960	62
1964/ 1965	8
1965/ 1966	8
1966/ 1967	15
1967/ 1968	9
1968/ 1969	3
1969/ 1970	20
1970/ 1971	22
1971/ 1972	0
1972/ 1973	67
1973/ 1974	31
1974/ 1975	78
1976/ 1977	0
1979/ 1980	27
1980/ 1981	40
1981/ 1982	29
1982/ 1983	27
1983/ 1984	0
1984/ 1985	10
1985/ 1986	24
1986/ 1987	37
1987/ 1988	17
1988/ 1989	18
1989/ 1990	66
1990/ 1991	55
1991/ 1992	133

Intervall	Avalanche
1992/ 1993	80
1993/ 1994	47
1994/ 1995	93
1995/ 1996	3
1996/ 1997	19
1997/ 1998	18
1998/ 1999	90
1999/ 2000	128
2000/ 2001	89
2001/ 2002	124
2002/ 2003	84
2003/ 2004	92
2004/ 2005	97
2005/ 2006	100
2006/ 2007	40
2007/ 2008	86
2008/ 2009	79
2009/ 2010	52
2010/ 2011	52
2011/ 2012	150
2012/ 2013	121
2013/ 2014	55
2014/ 2015	75
2015/ 2016	66
2016/ 2017	67
2017/ 2018	133
2018/ 2019	177
2019/ 2020	67
2020/ 2021	130
2021/ 2022	26

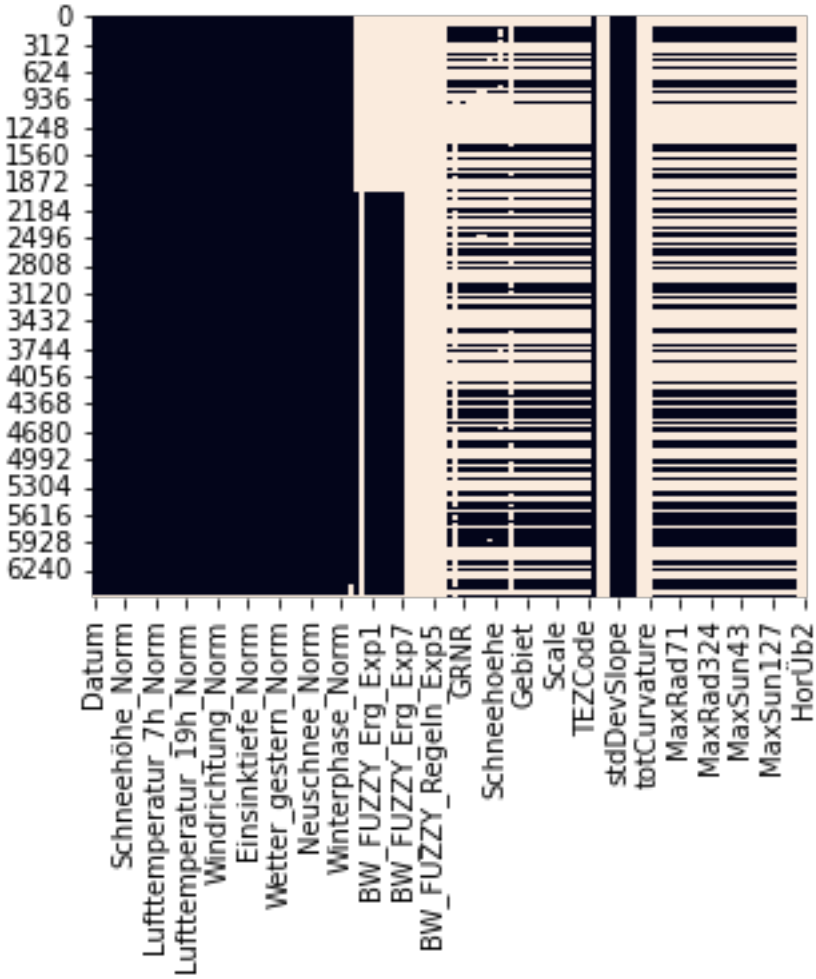
Table 1: recorded Avalanches per Season

The consequence of this is that the topographic data must also be mapped to the days without avalanches. Because these days are not connected to an avalanche line ID and an even distribution on the slopes on these days is required, random avalanche lines were picked from the calculated mean values of the TOPP table and mapped onto the non-avalanche days. Listing 2 shows how the avalanche line is randomly picked out of the modified TOPP table and the six columns of the picked row wich define the avalanche line topographical are mapped on the days without topographical information. This is shown in form of the corresponding Python code.
The resulting dataset maps 7021 rows and 139 columns. 2728 of these rows are recorded avalanches departures. This results in a data set consisting of 38.7% avalanche samples and 61.3% non-avalanche samples. The number of non-avalanche samples is therefore larger than the number of avalanche samples. Because of this, a bias can occur due to the greater number of non-avalanches. The evaluation of the three machine learning models will indicate whether such a bias is present. The dataset contains columns that are redundant, empty, sparsely filled or contain information wich can not be used to train a machine learning algorithm. Figure 15 shows a heatmap of the Nan values in the dataset, to get an overview about how many columns are sparse or incomplete. All white marked parts of the heatmap stand for Nan values in the dataset. As mentioned in chapter 3.1 about feature selection, these columns can cause a huge cost in computation time, decrease the prediction performance and increase the error rate. This requires the measure to remove all columns with these characteristics or fill their values.

In addition, a new column was added to the dataset, which contains either a 1 in case of an avalanche or a 0 in case no avalanche has occurred. This column is added to make it possible to predict whether an avalanche will occur or not. To train a machine learning algorithm on predicting whether an avalanche will go down from a topographical specified slope, wich is the specific aim of this master thesis, all features that can be used to directly and without any other features determine whether an avalanche will descend must be removed from the dataset.
The columns, wich are dropped for that reason are:
'Datum','Intervall','ZEIT','Lawinenabgänge', 'ID', 'Volumen', 'Lawinen_Art' After dropping this features the dataset includes 41 features. These columns include 25 meteorological factors, nine snowpack related features and six topographical factors and the column avalanche column wich says if this sample is an avalanche or a non-avalanche one.In the further course of the work the names of the data columns are used. For this reason and to get an overview about the dataset, table two shows all the features contained in the dataset in conjunction with their associated description.

Column Name	Description
Datum	The samples date
Intervall	The winter season in wich the sample was recorded
Schneehöhe	The snow height
Schneehöhe_Norm	The normalized snow height
Niederschlag	The rainfall
Lufttemperatur_7h	The air temperature at 7am
Lufttemperatur_7h_Norm	The normalized air temperature at 7am
Lufttemperatur_7h_Gew	The weighted air temperature at 7am
Lufttemperatur_14h	The air temperature at 2pm
Lufttemperatur_14h_Norm	The normalized air temperature at 2pm
Lufttemperatur_14h_Gew	The weighted air temperature at 2pm
Lufttemperatur_19h	The air temperature at 7pm
Lufttemperatur_19h_Norm	The normalized air temperature at 7pm
Lufttemperatur_19h_Gew	The weighted air temperature at 7pm
Schneetemperatur	The snow temperature
Schneetemperatur_Norm	The normalized snow temperature
Schneetemperatur_Gew	The weighted snow temperature
Windrichtung	The wind direction
Windrichtung_Norm	The normalized wind direction
Windrichtung_Gew	The weighted wind direction
Windstärke	The wind speed
Windstärke_Norm	The normalized wind speed
Windstärke_Gew	The weighted wind speed
Einsinktiefe	the snow sinking depth
Einsinktiefe_Norm	the normalized snow sinking depth
Einsinktiefe_Gew	The weight snow sinking depth
Wetter_akt	The encoded weather of the samples current day
Wetter_gestern	The encoded weather of the day before
Wolken	The encoded number of clouds
Wolken_Norm	The normalized encoded number of clouds
Neuschnee_x	The new snow on the samples current day
Neuschnee_Norm	The normalized new snow on the samples current day
Neuschnee_Gew	The weighted new snow on the samples current day
ID	The new avalanche ID
meanExpo	The mean slope exposition
meanSlope	The mean slope
stdDevSlope	The standard deviation of the slop
MinSlope	The minimum slope
MaxSlope	The maximum slope
Altitude	The elevation above sea level
Avalanche	The binary value if the sample is an avalanche or a non avalanche one

Table 2: The columns of the resulting dataset, in combination with their descriptions



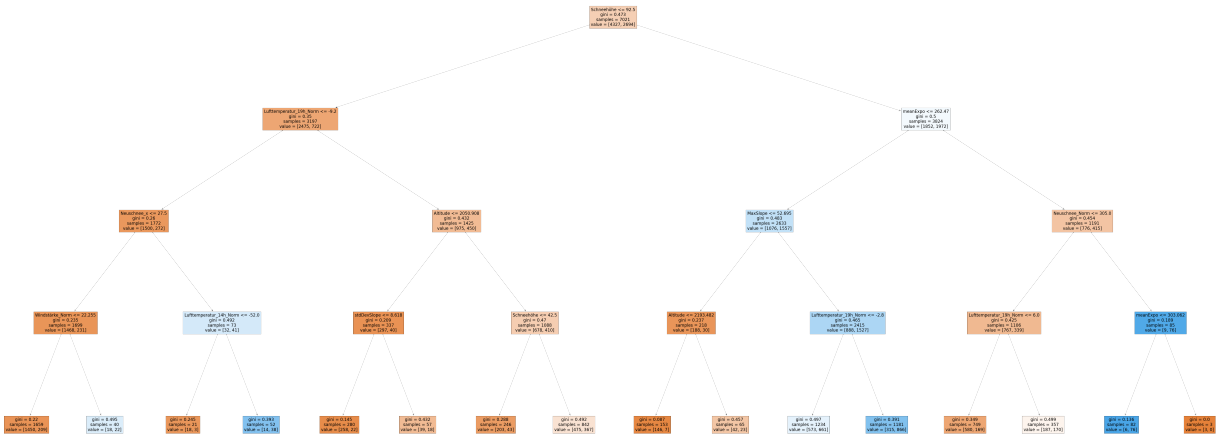
Source: created by the author

Figure 15: Heatmap to show the distribution of Nan values in the dataset

4.2 Feature Selection

This section describes the results of the feature selection process of the study. It is divided into two parts. The first part is more about figuring out which meteorological data columns have a huge impact on predicting avalanches in order to create additional features. For this purpose, a decision tree classifier is used and the features classified as significant by this classifier are added to the data set for a number of days in the past. The second part of the feature selection process is to find the approximately optimal feature set for each of the three machine learning models, trained for avalanche prediction in this study as well as to limit the number of features. For this purpose, a genetic algorithm is used to select the three features subsetssd. The two parts are described in detail in the following subsections.

4.2.1 Decision Tree



Source: created by the author

Figure 16: Decision Tree with a depth of 4 trained with the avalanche dataset described in section 4.1.5.

To gain an understanding of the importance of the individual data columns, a decision tree was trained using the data set created in section 4.1.5. Figure 18 shows a visualization of this tree. The nodes show wich features are used by the decision tree to create rules that are used for splitting the dataset. therefore, the features in higher-level nodes are the ones with the highest importance for the classification. For the purpose of this task the decision tree algorithm from the python library Scikit-learn was used [61]. The classifier model was used with a max depth of 5 and all other parameters left at the default values. Since feature selection, as described in chapter 3.1, is mainly about reducing the size of the dataset and removing irrelevant features to improve performance, the following procedure is rather unusual. The dataset contains only few meteorological data of the past days. Since this data can be crucial for the prediction of avalanches, the decision tree is used in this case to find the meteorological data with the highest priority for the decision tree. Therefore, the main reason for creating a decision tree model in

context of this master thesis is to add additionally data columns for the past two and four days to the dataset with the meteorological features used by the tree. Chawla, M. and Singh, A. Chawla and Singh [16] used a similar approach for their study. The visualization of the decision tree model trained for this reason is displayed in figure 16. As represented in that figure, the meteorological features wich are ranked by the decision tree model as features of great importance are the following: Lufttemperatur_19h_Norm, Lufttemperatur_19h , Neuschnee_x (new snow), Windstärke_Norm, Lufttemperatur_14h_Norm. The sum of normalized new snow in the period of the last two and last four days are added as two new columns to the dataset, because this feature describes the quantity of new snow it can be added as the sum and represent the full two or four days in one new data column without losing any information. The mean normalized air temperature and the normalized wind speed are added to the dataset for each day of the last four days. For these features the sum could not be taken, because they have fixed values and do not describe the quantity of something. To get a higher significance per column and less additional columns per day, the mean value of the normalized air temperature was calculated for each day out of the columns Lufttemperatur_7h_Norm, Lufttemperatur_14h_Norm as well as Lufttemperatur_19h_Norm. The values could also be added with the mean value of the last two or four days, but In this case, the significance of the values for the individual days would be limited. The four days at the beginning of a winter interval do not have at least four previous days so every value of the Lufttemperatur and Schneehöhe where no day and therefore no data is available, the columns value is set to 0.0 for this sample. The same happens with the columns for new snow in the last two and four days, if there is not at least one preceding day. This is important because otherwise either nan values are present in the samples or the samples have to be removed to enable the training of the machine learning models. Therefore this measure is essential in order not to lose the data samples. In total, ten new columns were added to the data set for the purpose of getting longer-term information on the meteorological conditions in the study area. The total avalanche dataset now has 56 features of which 40 are meteorological. The number of samples is the same as after data preparation, since none had to be discarded. Table 3 represents all new additionally included feature columns in combination with the corresponding descriptions, to give an overview about the new features. In the table it can be seen that out of the 10 additional features 2 are new snow related features, 4 are windspeed representing features and 4 are air temperature features.
Whether this additional information from the machine learning models is relevant for the prediction of avalanches will be shown in the next chapter about the results of the actual feature selection, which is performed individually for the three different classifications models using a genetic algorithm.

4.2.2 Genetic Algorithm

To get an optimal selection of features for the particular machine learning algorithms the three models Logistic Regression, Support Vector Machine and Linear Discriminant Analysis were trained repeatedly by the use of a Genetic Algorithm in the purpose of features selection. The goal of this task is to get a set of less features, which are selected as optimal as possible for the application with the corresponding machine learning model. As described in detail in section 3.1.2 the Genetic Algorithm is based on natural selection and tries to take the best features from

Column Name	Description
Neuschnee_last2	The new snow fallen in the last two days before the current
Neuschnee_last4	The new snow fallen in the last four days before the current
windstaerke_last1	The wind speed on the day before the current
windstaerke_last2	The wind speed two days before the current
windstaerke_last3	The wind speed three days before the current
windstaerke_last4	The wind speed four days before the current
Lufttemperatur_last1	The air temperature one day before the current
Lufttemperatur_last2	The air temperature two days before the current
Lufttemperatur_last3	The air temperature three days before the current
Lufttemperatur_last4	The air temperature four days before the current

Table 3: The 10 additional feature columns in combination with their descriptions

generation to generation into the new population. So that the selection approaches the optimal data set for the respective algorithm.
The library contains two main genetic algorithms one for hyper-parameter optimization and the second for the task of feature selection [30]. For this task the feature selection version, wich is called GAFeatureSelectionCV is used.

In purpose to use standardized feature set, the used estimator in all three runs for the different models is a pipeline with a data normalization scaler and the respective machine learning algorithm. As mentioned in section 3.1, the StandardScaler function from the Python library Sklearn is used as normalization scaler.

The parameters of the genetic algorithm are the same for all three machine learning models. As the first parameter the cross validation is set to a 10 fold, for the scoring of the fitness value is the accuracy score is used, the number of how many individuals are in the starting population is set to 60, the stopping criteria is the maximum generation value of 50, wich means the algorithm calculates 50 generations until it stops the process. The number of parallel running jobs is set to -1 one which stands for using all processors, therefore the learning process of the estimator and the calculation of the score are parallelized via the splitting of the cross-validation. The verbose value was set to True to reflect the progress of the algorithm in terms of the metrics of the optimization routine, as shown in Table 4 for the last generation of optimization of the Logistic Regression model. The probability that the crossover operation is performed between two individuals is left at the default value of 0.2. Also the value of the probability for child mutation is left to its default value of 0.8. The number of individuals wich perform tournament selection also remains at the default value of 3. The last parameter to mention is the maximum number of features to be selected. This value is set to None to get all features, which the algorithm thus assigns as important. All other parameters of the genetic algorithm that can be adjusted are left at their default values. [30] The output metrics of the genetic algorithms are shown in table 4, 5 and 6 for the three different results [62]:

- ”gen”: The number of the generation.

- ”nevals”: The value of the population used for the current generation.
- ”fitness”: The average fitness score, wich in the case of this study is the accuracy score, over the cross-validation folds results.
- ”fitness_std”: The representation of the standard deviation of the cross-validation accuracy.
- ”fitness_min”: The minimum fitness score of the cross-validation folds.
- ”fitness_max”: the maximum fitness score of the cross-validation.

”gen”	”nevals”	”fitness”	”fitness_std”	”fitness_max”	”fitness_min”
50	120	0.733517	0.000460338	0.733647	0.730656

Table 4: The metrics of the last generation calculated by the genetic algorithms optimization routine while optimizing the dataset for the Logistic Regression model.

First of the three models, the Logistic Regression model was trained by the genetic algorithm. As a result the genetic algorithm created a subset of 20 features as a best approximation to the optimal subset for the logistic regression model. As shown in Table 4, the process of optimizing the data set for the Logistic Regression model was completed with the 50th generation with an average accuracy of 0.733517. Where the lowest value of 10-fold cross-validation is 0.730656, the highest is 0.733647, and the standard deviation of all values in the series is 0.000460338. Table 5, which contains a count of the selected features, shows the subset of selected features for the Logistic Regression Model contains the two factors windstaerke_last3 and Lufttemperatur_last4, wich were added to the dataset by the previous process explained in section 4.2.1, as shown in Table 3. The optimized subset for the Logistic Regression model includes in total 9 meteorological features, five snowpack related columns as well as all of the six topographical factors.

The second approximately optimal subset was calculated for the LDA model by the genetic algorithm. After passing the 50 generation, the second run of the genetic algorithm to obtain an optimized set for the LDA was completed with an average accuracy of 0.735865. The lowest value of the series of accuracy scores in this case was 0.728662, the highest 0.737352, and the standard deviation of the entire series was 0.00171288. The list of values can be found in table 5. The resulting subset includes 24 features. Nine of the selected columns are meteorological features, of which the five columns Neuschnee_last4, Neuschnee_last2, windstaerke_last1, windstaerke_last2 and Lufttemperatur_last4 are from the new features added in section 4.2.1.

”gen”	”nevals”	”fitness”	”fitness_std”	”fitness_max”	”fitness_min”
50	120	0.735865	0.00171288	0.737352	0.728662

Table 5: The metrics of the last generation calculated by the genetic algorithms optimization routine while optimizing the dataset for the LDA model.

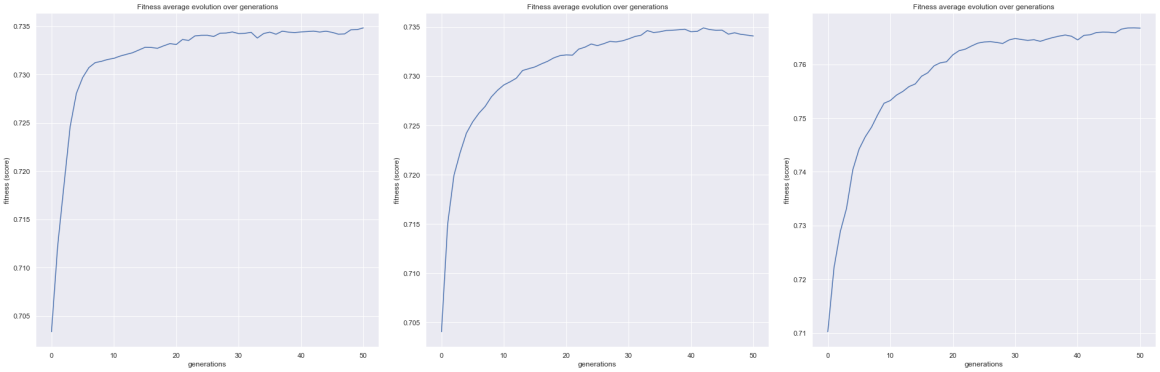
”gen”	”nevals”	”fitness”	”fitness_std”	”fitness_max”	”fitness_min”
50	120	0.768267	0.00159737	0.768972	0.762988

Table 6: The metrics of the last 10 generations calculated by the genetic algorithms optimization routine while optimizing the dataset for the SVM model.

The genetic algorithm also included all six topographic factors as well as three snowpack related features in the subset.

The last subset, wich is created by the genetic algorithm optimized for the SVM is a combination of 25 features. Therefor, this subset is the largest of the three. It includes 15 meteorological, five snowpack related and five topographical features. Also for the optimized subset for the SVM, the two new meteorological factors of the past days Neuschnee_last4 and Lufttemperatur_last4 were selected. The genetic algorithms last generation shows an accuracy of 0.768267 and completes the features selection process with the highest value of the three machine learning models. The smallest accuracy score measured in the 10-fold cross-validation series is 0.762988. The highest accuracy score is 0.768972 and the standard derivation is 0.00159737. While the the final values for the LDA and the Logistic Regression model after feature selection are close to each other, the values of the SVM model are higher than both of the other models.

The three graphs pictured in figure 17 show the increase of the fitness score over the 50 gen-



Source: created by the author

Figure 17: The fitness evolution increase over 50 generations created by the genetic algorithm for the Logistic Regression, LDA and SVM models.

erations, created by the genetic algorithm. The left graph shows the fitness evolution of the Logistic Regression model. It started an average fitness score over the 10-fold cross validation of 0.704242 increased in the first 6 generations and completed the process with the value of 0.733517. The graph in the middle of the three figures represents the fitness evolution of the LDA. As the graph shows, the fitness score in this case increases more continuously over the entire 50 generation. It is also noticeable, that it drops again somewhat towards the end. The application of the genetic algorithm to the LDA started with a fitness score of 0.703853 and ended with a value of 0.735865. The fitness evolution of the SVM model increased while the feature selection process with the genetic algorithm from an average accuracy of 0.709667 in the first generation to 0.768267 in the 50st generation.

The resulting three optimized datasets are represented in Table 7. The table contains one

Logistic Regression	LDA	SVM
Schneehöhe	Schneehöhe	Schneehöhe
Lufttemperatur_7h_Gew	Niederschlag	Lufttemperatur_7h
Lufttemperatur_14h	Lufttemperatur_7h	Lufttemperatur_14h_Norm
Lufttemperatur_14h_Gew	Lufttemperatur_7h_Gew	Lufttemperatur_14h_Gew
Lufttemperatur_19h	Lufttemperatur_14h_Norm	Lufttemperatur_19h
Schneetemperatur	Lufttemperatur_14h_Gew	Lufttemperatur_19h_Norm
Schneetemperatur_Gew	Lufttemperatur_19h	Lufttemperatur_19h_Gew
Windstärke	Schneetemperatur	Schneetemperatur
Einsinktiefe_Norm	Windrichtung_Gew	Schneetemperatur_Norm
Einsinktiefe_Gew	Windstärke_Norm	Schneetemperatur_Gew
Wetter_akt	Einsinktiefe_Norm	Windstärke
Neuschnee_Norm	Wetter_akt	Windstärke_Gew
windstaerke_last3	Neuschnee_Norm	Einsinktiefe_Gew
Lufttemperatur_last4	Neuschnee_last4	Wetter_akt
meanExpo	Neuschnee_last2	Wetter_gestern
meanSlope	windstaerke_last1	Wolken
stdDevSlope	windstaerke_last2	Neuschnee_x
MinSlope	Lufttemperatur_last4	Neuschnee_Norm
MaxSlope	meanExpo	Neuschnee_last4
Altitude	meanSlope	Lufttemperatur_last4
	stdDevSlope	meanExpo
	MinSlope	meanSlope
	MaxSlope	MinSlope
	Altitude	MaxSlope
		Altitude

Table 7: The features selected by the genetic algorithm for the three machine learning models, Logistic Regression, LDA and SVM

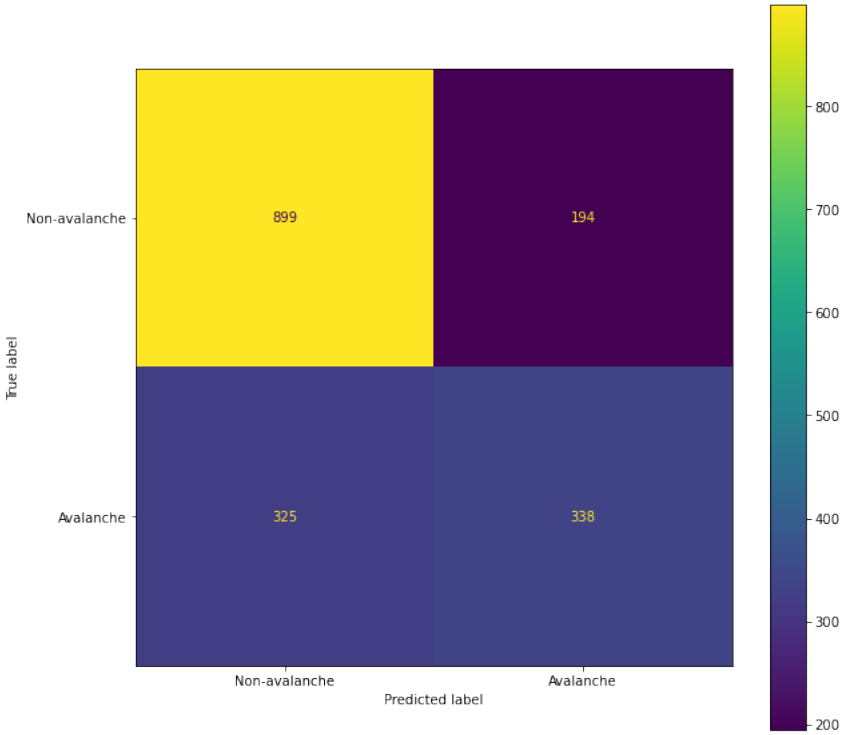
column per machine learning model. In these columns, the data sets are assigned to the corresponding machine learning models. By analyzing the datasets in the table, similarities of the selected features for the three algorithms are recognizable. The columns Schneehöhe, Lufttemperatur_14h_Gew, Lufttemperatur_19h, Schneetemperatur, Wetter_act, Neuschnee_Norm, Lufttemperatur_last4 are meteorological data that are included in all three optimized datasets, which suggests that they are of higher importance for the prediction of avalanches. Also the different features wich represent the air temperature at various points in time are represented several times in all subsets. Thus, 6 features are present in the set optimized for the logistic regression model, 5 in the set for the LDA, and 7 in the optimized subset of the SVM model features representing the air temperature. This suggests a correlation between air temperature and avalanche prediction for all three sets. Furthermore, the topographic factors are fully present in the Logistic Regression and Linear Discriminant Analysis set and in the Support Vector Machine set five

out of six features are included, whereas the standard deviation of the slope is excluded here. In comparison to the study ”Snow avalanche hazard prediction using machine learning methods”, mentioned in Chapter 2, in this study the altitude is represented in all three datasets, while the study ranked the altitude as the last important feature for avalanche prediction Bahram Choubin [5]. It can be concluded that an accurate definition of slopes based on topographic factors is related to the quality of avalanche prediction.

4.3 Model training and evaluation

In this chapter, the training phase and the evaluation of the three machine learning classification models Logistic Regression, Linear Discriminant Analysis and Support Vector Machine are explained. The structure of the models and their evaluation will be discussed in more detail. The three machine learning models are trained and evaluated in the order Logistic Regression, Linear Discriminants Analysis. The three models were trained on the respective data sets optimized for the models, wich are described before in section 4.2.2. A confusion matrix was created for every model as well as cross-validated accuracy, precision and recall metric. In addition, cross validated ROC curves were created for the Logistic Regression, LDA and SVM models. Before training and evaluating the algorithms on the optimized sets, the datasets were standardized with the StandardScaler [19] function mentioned in section 3.1 about data preprocessing.

The training and evaluation phase of the Logistic Regression model is described as the first of the three classification models. The Logistic Regression model from the Sklearn.linear_model library [38] was trained with its default parameters.
In order to gain a direct insight into the predictions of the machine learning model, the confu-



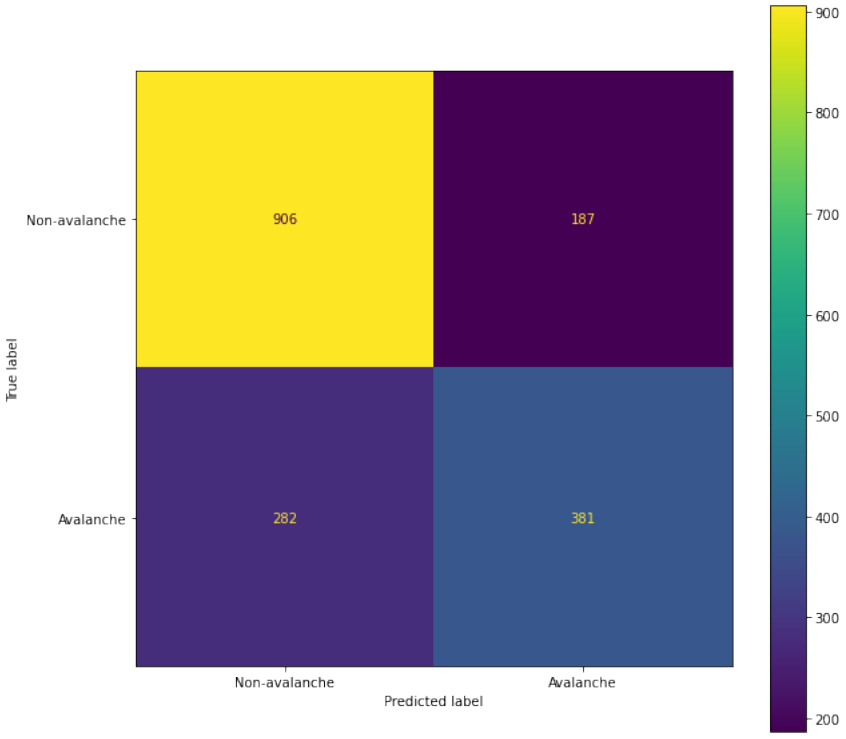
Source: created by the author

Figure 18: The confusion matrix for the Logistic Regression model.

sion matrix, wich is shown in figure 18, was created for the model in the course of the study. The confusion matrix represents the four output types of the binary classification model as

mentioned in chapter 3.4. In the case of this study the binary states are ”Avalanche” and ”Non-avalanche”. The confusion matrix in figure 18 presents the predicted states on the Y-axis and the true states on the X-axis. For the creation of the confusion matrix the train test split function from the Python library sklearn.model_selection [63] was performed on the data set with a train test ratio of 75% training set to 25% validation set. So in total 5263 training samples and 1758 test samples are used. To create the confusion matrix, the plot function ConfusionMatrixDisplay [64] from the library Sklearn.metrics is used. In the top left corner of the matrix the 899 True Negative predicted Non-avalanches are represented. 194 actual Non-avalanches were predicted as Avalanches shown in the top right quarter of the matrix. The Logistic Regression Model also predicted 325 Avalanches as Non-avalanches and 338 actual Avalanche samples as Avalanches. So the total number of samples falsely predicted by the Logistic Regression model is 519. The total number of correctly predicted samples is 1239. The number of Avalanche samples included in the test set is 663 and the number of Non-avalanche samples is 1093. As the confusion matrix shows, the percentage of correctly determined avalanches is smaller compared to the percentage of correctly determined non-avalanche samples. This could indicate a bias toward the non-avalanche samples caused by the fact that the dataset includes more Non-avalanche samples than Avalanche samples, as mentioned in chapter 4.1.5.

The second model trained and evaluated is the Linear Discriminant Analysis model. For



Source: created by the author

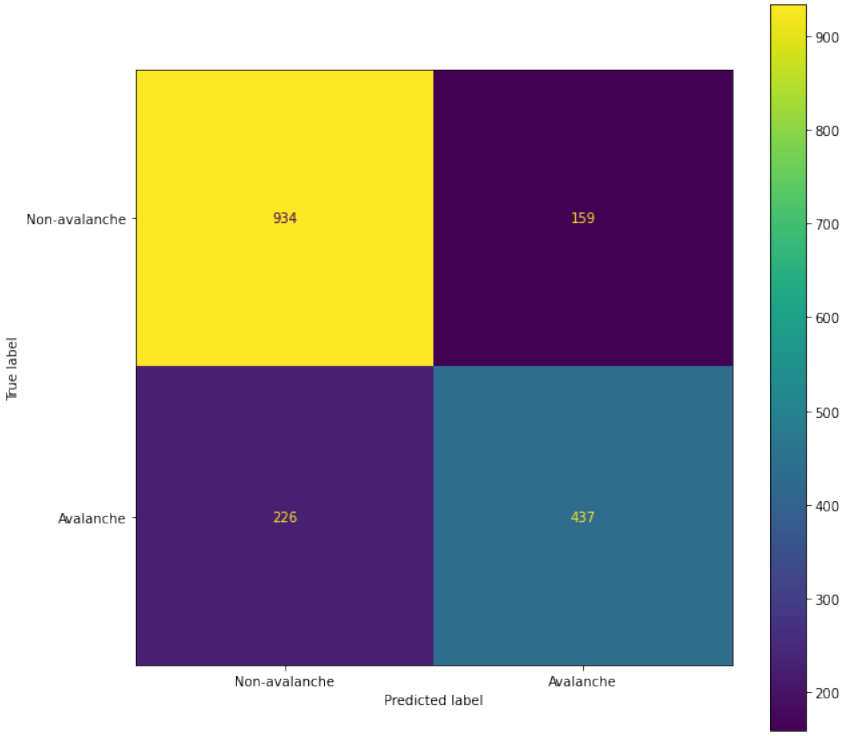
Figure 19: The confusion matrix for the Linear Discriminant Analysis model.

this task, the LDA model from the sklearn.discriminant_analysis [47] is used. The model was

also trained on with the default parameters defined by the library. For all following training and evaluation processes of the model, the data set optimized for the LDA by the genetic algorithm, which is listed in Table 7, was used.
The confusion matrix represented in figure 19 shows was created for the purpose of getting an overview over the predictions made by the Linear Discriminant Analysis model. The confusion matrix was created with the same function as that one for the Logistic Regression model, as well as the 75% to 25% split for the train and validation dataset. The number of Non-avalanche samples of the test set is 1093 and the number of Avalanches 663, wich are exactly the same numbers as for the Logistic Regression model. The number of True-Positive predicted samples shown in that confusion matrix is 381. The sum of Avalanche samples wich were predicted as Non-avalanche ones is 282. Furthermore, the number of non-avalanche samples predicted by the Linear Discriminant Analysis model True-Negative is 906. In the upper right quarter of the confusion matrix, the incorrectly predicted non-avalanches are plotted, with the number of False-Positive values at 194. As can be seen the number of True-Positive predicted samples is larger and the number True-Negative predicted samples is similar compared to the predictions of the Logistic Regression model. It is especially with the prediction of the avalanche samples further away from coincidence. This indicates that there may be sufficient information in the data to predict snow avalanches for topographically defined mountain slopes. In total 1287 samples have been predicted correctly by the LDA model. The number is slightly higher than that one of the Logistic Regression, because the LDA predicted more Avalanche samples correctly.

The Support Vector Machine is the last model trained for the study of this thesis. The SVM implementation used is the C-Support Vector Classification (SVC) [41] from the Sklearn.svm library. The parameters used for the model are also the default values from the documentation. Also in the case of the SVM model, the confusion matrix was created using a 75% to 25% train test ratio of the data. In addition, the number of avalanche and non-avalanche samples in the training and test set is the same as in the previous two models. The first thing to notice when looking at the confusion matrix in Figure 20, which was created using the SVM model, is the color difference between the correctly and incorrectly predicted samples. The value of the correctly predicted avalanche samples is the highest of the three models at 437, which makes the square stand out more from the one to the left, which represents the false negative predicted values, than in the confusion matrix for the logistic regression model. The number of False-Negative predicted values shown in the bottom left corner of the confusion matrix is 226, wich is 99 samples less than with the Logistic Regression model and 56 less then the LDA. The present difference between correctly and incorrectly predicted avalanche samples suggests that the predictions are not coincidental and that the data contain information for the prediction of avalanches. The sum of samples predicted as True-Negative is 934. In addition, 159 non-avalanche samples were predicted to be avalanches. In total, 1371 samples have been predicted correctly by the SVM model. So it predicted more samples correctly than the LDA and the Logistic Regression models. The model also predicted 385 samples incorrectly wich are 84 samples less than the LDA model.

To gain more information about the quality of the predictions, made by the three machine



Source: created by the author

Figure 20: The confusion matrix for the Support Vector Machine model.

learning models, a 10-fold cross-validation was performed on each of the models and a series of the evaluation metrics accuracy, recall and precision were created in conjunction with them. As explained in chapter 3.4.5, cross-validation offers the possibility to perform an extensive validation over the whole dataset and at the same time to use a large part of the dataset for the training of the machine learning models at each iteration. So the resulting metrics are more representative for the quality of the predictions than the 75% to 25% train test split made for the confusion matrices before. For each of the resulting metrics, the maximum, minimum, and average values that occurred during the 10-fold cross-validation are represented in table 8. The first score wich is represented in table 8 is the accuracy score. As mentioned in section 3.4.1, the accuracy score represents the total number of correctly predicted samples compared to the total number of samples. The average accuracy score is 0.735 for the Logistic Regression model, 0.732 for the LDA model and 0.754 for the SVM model. So the values are similar for all three models, but that one of the SVM is higher than the others. The maximum accuracy is 0.806 for the Logistic Regression, 0.800 for the LDA and 0.803 for the SVM. In this case the values of all three models are similar to each other. The values of the minimum accuracy in the same order as before are 0.678, 0.690, 0.713. As can be seen, the SVM has the highest minimum accuracy, the LDA follows and last is the Logistic Regression. Causing the fact that there are less Avalanche samples included in the datasets than Non-avalanches, the balanced accuracy score is also calculated for the three models. The average balanced accuracy score is 0.705 for the Logistic Regression model. This value is 0.030 smaller than the normal accuracy score for this

model. For the Linear Discriminant Analysis model the balanced accuracy score is 0.700, wich is 0.032 smaller than the average accuracy score for the same model. The value of the average balanced accuracy for the SVM model is 0.724, wich is the highest value of the three. Similar to the balanced accuracy score of the two models Logistic Regression and LDA it is about 0.030 smaller than the accuracy score for the model. As already mentioned in chapter 3.4.1 about the accuracy score, the value of the balanced accuracy score is often smaller than that one of the accuracy score in the case of an unbalanced data set. In contrast to the most previous metric values, the value of the maximum balanced accuracy score for the Logistic Regression model is the largest of the three models at 0.788. The value for the LDA is 0.765 and for the SVM 0.779. However, the value of the minimum balanced accuracy score for the Logistic Regression model is also the lowest with 0.636 compared to 0.659 and 0.663 for the LDA and SVM.

Another evaluation metric calculated in this evaluation process and represented in table 8 is

	Logistic Regression	LDA	SVM
Average accuracy	0.735356	0.732791	0.754446
Max accuracy	0.806268	0.800853	0.803419
Min accuracy	0.678063	0.690883	0.713675
Average balanced accuracy	0.705463	0.700595	0.724890
Max balanced accuracy	0.788748	0.765140	0.779399
Min balanced accuracy	0.636473	0.659482	0.663706
Average precision	0.692394	0.693510	0.720818
Max precision	0.831579	0.825000	0.809278
Min precision	0.570571	0.587859	0.613636
Average recall	0.577125	0.562296	0.597944
Max recall	0.755556	0.703704	0.762963
Min recall	0.379182	0.425926	0.449814

Table 8: A tabular arrangement showing the results of training the machine learning models and evaluating them using the metrics Accuracy, Precision and Recall in the course of 10-fold cross-validations.

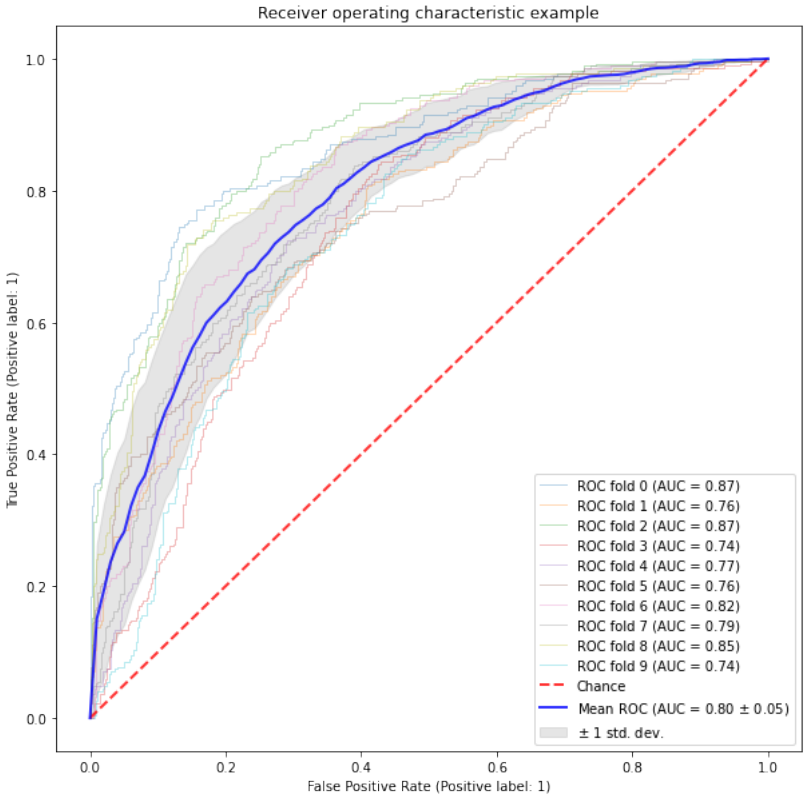
the precision score. The value of the average precision score over the 10-fold cross-validation, calculated for the Logistic Regression model, is 0.692. For the LDA the average of the score is 0.693, wich is similar to the Logistic Regression model. The SVM has the highest average precision score of 0.720 over the ten iterations of cross validation. The metric evaluates the proportion of positively predicted samples that are actually positive. The Logistic Regression model had both the highest maximum precision score of 0.831 and the lowest minimum precision score of 0.570 of the three models. The next lower maximum precision score of 0.825 and the following minimum precision score of 0.587. The SVM has the lowest maximum precision score of 0.809 and the highest minimum precision score of 0.613. As can be seen in the table, the maximum precision score of the SVM is lower than that of the other models, but the average value over the 10 iterations is higher than the scores of the other models. The last metric calculated in the course of the evaluation of the three machine learning models and shown in table 8

is the recall score. The score in the context of this work refers only to the avalanches and the proportion of avalanches that were correctly predicted. The value of the Logistic Regression models average recall is 0.577. For the LDA model the value of the score is 0.562 and for the SVM model it is 0.597. As can be seen all values are smaller than the average values of the other scores. But also all of these values are above chance. The minimum and maximum values of the recall in the cross validation of the three models drift from each other. Thus, the minimum value of the recall of the Logistic Regression model is 0.379, which is smaller than chance, which is 0.5, and the maximum value of 0.755 is similar to the average accuracy score and 0.376 larger than the minimum and thus almost twice as large. Also the LDAs maximum recall value is about 0.703 an its minimum is 0.425. Recently, the maximum and minimum recall values of the SVM model are also different with 0.762 and 0.449. While all of the maximum and average recall score values are above the chance value of 0.5, the three minimum values are all below the value.

After that step the last evaluation step made in case of this study, are ROC-curves in combination with the Area under the curve (AUC) score. For the three machine learning models the ROC curve and AUC score is calculated during 10-fold cross-validations, to obtain representative validation values. The graphs represented in the figures 21, 22 and 23 do all show a red striped straight line from $(0, 0)$ to $(1, 1)$. This line represents chance. So if the ROC-curve of a model is near to that line it is near to chance. Also if the values are below the chance, the model might have a false interpretation of the input data. A description of the ROC-curve and AUC value can be found in chapter 3.4.4.

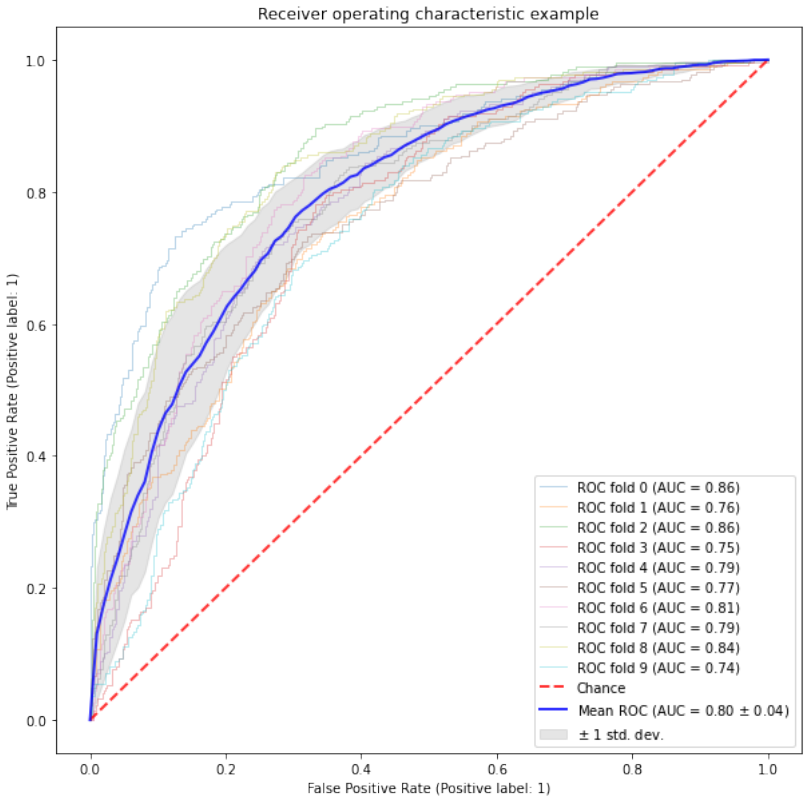
In addition, three more ROC curves and AUC values were calculated for the three machine learning models without the use of 10-fold cross-validation and visualized in a fourth plot. These curves can be seen in Figure 24.

The standard linear SVM is a discrete classifier so in terms of a roc curve it would just be a point. The ”RBF” kernel used as default for the SVC Implementation from Sklearn makes it possible to have a threshold and so also to get a ROC curve. For this study the SVC implementation is used with the default kernel ”RBF”.



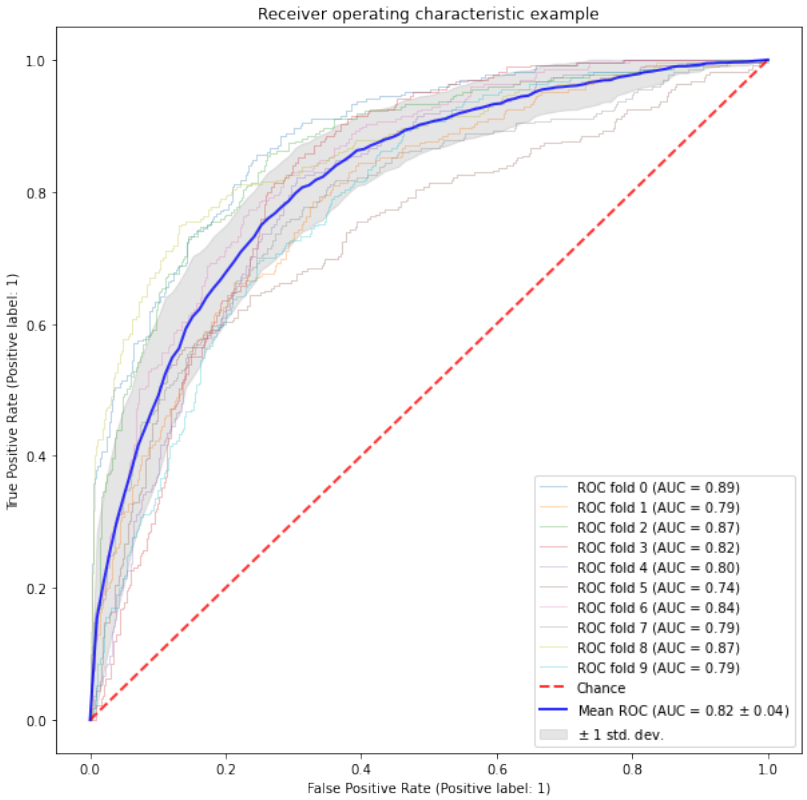
Source: created by the author

Figure 21: The ROC curve with AUC statistics with 10-fold cross-validation for the Logistic Regression model.



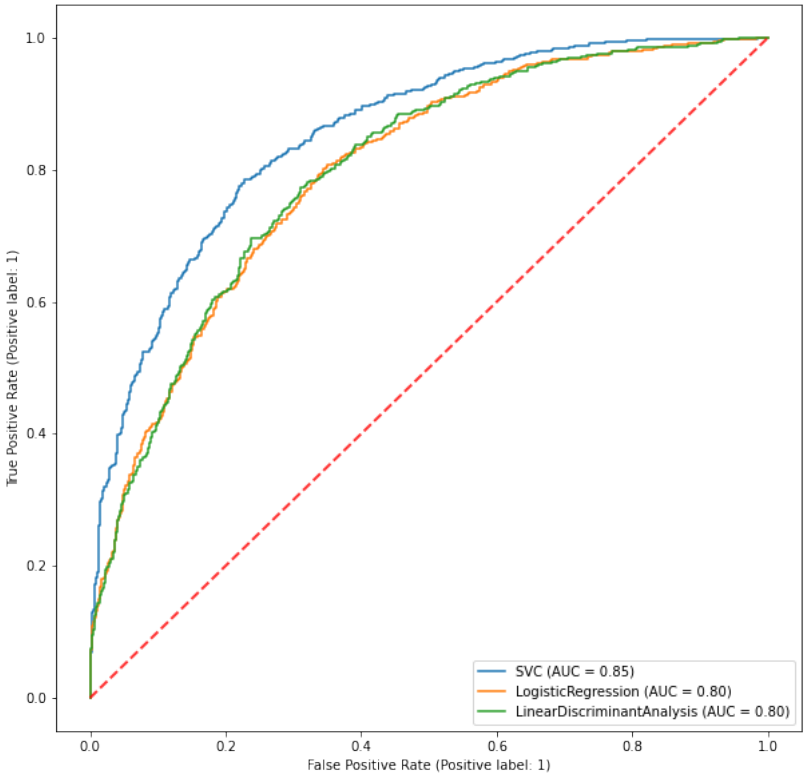
Source: created by the author

Figure 22: The ROC curve with AUC statistics with 10-fold cross-validation for the LDA model.



Source: created by the author

Figure 23: The ROC curve with AUC statistics with 10-fold cross-validation for the SVM model.



Source: created by the author

Figure 24: A comparison of the ROC curve and AUC statistic for the Logistic Regression, LDA and SVM models.

5 Discussion & Outlook

6 Conclusion and Future Work

Listings

List of Figures

References

[1] V. by Hermann Maurer, version 3, 2019. [Online]. Available: https://austria-forum.org/af/AustriaWiki/Lawine?version=3 (visited on 01/15/2022).

[2] L. S.L.O.-L.V.L.K.U. W. Lawinenwarndienst Tirol Lawinenwarndienst Steiermark. (2022). Lawinen Ereignisse, [Online]. Available: https://lawis.at/incident/ (visited on 01/10/2022).

[3] Y. L.G. B. Eric Martin Gérald Giraud, „Impact of a climate change on avalanche hazard“, Annals of Glaciology, vol. 32, pp. 163–167, 2001. [Online]. Available: https://doi.org/10.3189/172756401781819292.

[4] B. D. V. Anuj Tiwari Arun G., „Parameter importance assessment improves efficacy of machine learning methods for predicting snow avalanche sites in Leh-Manali Highway, India“, Science of The Total Environment, vol. 794, 2021. [Online]. Available: https://doi.org/10.1016/j.scitotenv.2021.148738.

[5] A. M.F.S.-H.V.P.S. S. Bahram Choubin Moslem Borji, „Snow avalanche hazard prediction using machine learning methods“, Journal of Hydrology, vol. 577, 2019. [Online]. Available: https://doi.org/10.1016/j.jhydrol.2019.123929.

[6] T. Thüring, M. Schoch, A. van Herwijnen, and J. Schweizer, „Robust snow avalanche detection using supervised machine learning with infrasonic sensor arrays“, Cold Regions Science and Technology, vol. 111, pp. 60–66, 2015, ISSN: 0165-232X. DOI: https://doi.org/10.1016/j.coldregions.2014.12.014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165232X14002419.

[7] A. Subasi, „Chapter 3 - Machine learning techniques“, in Practical Machine Learning for Data Analysis Using Python, A. Subasi, Ed., Academic Press, 2020, pp. 91–202, ISBN: 978-0-12-821379-7. DOI: https://doi.org/10.1016/B978-0-12-821379-7.00003-5. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128213797000035.

[8] (2022), [Online]. Available: https://www.ibm.com/topics/knn (visited on 06/24/2022).

[9] A. Pozdnoukhov, R. Purves, and M. Kanevski, „Applying machine learning methods to avalanche forecasting“, Annals of Glaciology, vol. 49, 107–113, 2008. DOI: 10.3189/172756408787814870.

[10] B. R. Stephan Harvey Alec van Herwijnen, „Statistical Nowcast of Avalanche Activity at the Regional Scale“, 2016. [Online]. Available: https://arc.lib.montana.edu/snow-science/item/2437.

[11] (2022), [Online]. Available: https://www.dwd.de/DE/service/lexikon/Functions/glossar.html?lv2=101640&lv3=101744 (visited on 06/24/2022).

[12] T. Wood. (2022), [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/random-forest (visited on 06/20/2022).

[13] (2022), [Online]. Available: https://www.wortbedeutung.info/Topografie/ (visited on 06/24/2022).

[14] (2022), [Online]. Available: https://www.wyssenavalanche.com/en/avalanche-detection/ida-infrasound-detection-system/ (visited on 06/24/2022).

[15] H. Wen, X. Wu, X. Liao, D. Wang, K. Huang, and B. Wünnemann, „Application of machine learning methods for snow avalanche susceptibility mapping in the Parlung Tsangpo catchment, southeastern Qinghai-Tibet Plateau“, Cold Regions Science and Technology, vol. 198, p. 103 535, 2022, ISSN: 0165-232X. DOI: https://doi.org/10.1016/j.coldregions.2022.103535. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0165232X22000544.

[16] M. Chawla and A. Singh, „A data efficient machine learning model for autonomous operational avalanche forecasting“, Natural Hazards and Earth System Sciences Discussions, vol. 2021, pp. 1–18, 2021. DOI: 10.5194/nhess-2021-106. [Online]. Available: https://nhess.copernicus.org/preprints/nhess-2021-106/.

[17] P. Baheti. (2022), [Online]. Available: https://www.v7labs.com/blog/data-preprocessing-guide (visited on 07/25/2022).

[18] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, „Data preprocessing for supervised leaning“, International journal of computer science, vol. 1, no. 2, pp. 111–117, 2006.

[19] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn-preprocessing-standardscaler (visited on 07/25/2022).

[20] (2022), [Online]. Available: https://www.ibm.com/cloud/learn/overfitting (visited on 06/07/2022).

[21] J. Cai, J. Luo, S. Wang, and S. Yang, „Feature selection in machine learning: A new perspective“, Neurocomputing, vol. 300, pp. 70–79, 2018, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.11.077. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231218302911.

[22] M. Allam and M. Nandhini, „Optimal feature selection using binary teaching learning based optimization algorithm“, Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 2, pp. 329–341, 2022, ISSN: 1319-1578. DOI: https://doi.org/10.1016/j.jksuci.2018.12.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157818306463.

[23] B. Venkatesh and J. Anuradha, „A Review of Feature Selection and Its Methods“, Cybernetics and Information Technologies, vol. 19, no. 1, pp. 3–26, 2019. DOI: doi:10.2478/cait-2019-0001. [Online]. Available: https://doi.org/10.2478/cait-2019-0001.

[24] D. A. Pisner and D. M. Schnyer, „Chapter 6 - Support vector machine“, in Machine Learning, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 101–121, ISBN: 978-0-12-815739-8. DOI: https://doi.org/10.1016/B978-0-12-815739-8.00006-7. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000067.

[25] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/tree.html (visited on 04/13/2022).

[26] V. Sugumaran, V. Muralidharan, and K. Ramachandran, „Feature selection using Decision Tree and classification through Proximal Support Vector Machine for fault diagnostics of roller bearing“, Mechanical Systems and Signal Processing, vol. 21, no. 2, pp. 930–942, 2007, ISSN: 0888-3270. DOI: https://doi.org/10.1016/j.ymssp.2006.05.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327006001142.

[27] J. Lu, T. Zhao, and Y. Zhang, „Feature selection based-on genetic algorithm for image annotation“, Knowledge-Based Systems, vol. 21, no. 8, pp. 887–891, 2008, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2008.03.051. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070510800097X.

[28] P. Liashchynskyi and P. Liashchynskyi, „Grid Search, Random Search, Genetic Algorithm: A Big Comparison for NAS“, CoRR, vol. abs/1912.06059, 2019. arXiv: 1912.06059. [Online]. Available: http://arxiv.org/abs/1912.06059.

[29] K.-H.L.C.-M.K.H.-W. T. Tao-Chang Yang Pao-Shan Yu, „Predictor selection method for the construction of support vector machine (SVM)-based typhoon rainfall forecasting models using a non-dominated sorting genetic algorithm“, Meteorological Applications, vol. 25, 2018. DOI: https://doi.org/10.1002/met.1717. [Online]. Available: https://rmets.onlinelibrary.wiley.com/doi/10.1002/met.1717.

[30] (2022), [Online]. Available: https://sklearn-genetic-opt.readthedocs.io/en/stable/api/featureselectioncv.html (visited on 07/18/2022).

[31] S. Vieira, W. H. Lopez Pinaya, and A. Mechelli, „Chapter 1 - Introduction to machine learning“, in Machine Learning, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 1–20, ISBN: 978-0-12-815739-8. DOI: https://doi.org/10.1016/B978-0-12-815739-8.00001-8. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128157398000018.

[32] (2022), [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning (visited on 06/13/2022).

[33] (2022), [Online]. Available: https://www.ionos.at/digitalguide/online-marketing/suchmaschinenmarketing/was-ist-reinforcement-learning/ (visited on 06/13/2022).

[34] (2022), [Online]. Available: https://www.ibm.com/topics/logistic-regression (visited on 05/21/2022).

[35] H. Belyadi and A. Haghighat, „Chapter 5 - Supervised learning“, in Machine Learning Guide for Oil and Gas Using Python, H. Belyadi and A. Haghighat, Eds., Gulf Professional Publishing, 2021, pp. 169–295, ISBN: 978-0-12-821929-4. DOI: https://doi.org/10.1016/B978-0-12-821929-4.00004-4. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128219294000044.

[36] „Beginners Take: How Logistic Regression is related to Linear Regression“, Data Science Blogathon, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/.

[37] S. Nusinovici, Y. C. Tham, M. Y. Chak Yan, D. S. Wei Ting, J. Li, C. Sabanayagam, T. Y. Wong, and C.-Y. Cheng, „Logistic regression was as good as machine learning for predicting major chronic diseases“, Journal of Clinical Epidemiology, vol. 122, pp. 56–69, 2020, ISSN: 0895-4356. DOI: https://doi.org/10.1016/j.jclinepi.2020.03.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0895435619310194.

[38] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html#sklearn-linear-model-logisticregression (visited on 05/21/2022).

[39] R. Gholami and N. Fakhari, „Chapter 27 - Support Vector Machine: Principles, Parameters, and Applications“, in Handbook of Neural Computation, P. Samui, S. Sekhar, and V. E. Balas, Eds., Academic Press, 2017, pp. 515–535, ISBN: 978-0-12-811318-9. DOI: https://doi.org/10.1016/B978-0-12-811318-9.00027-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128113189000272.

[40] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/svm.html (visited on 05/24/2022).

[41] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (visited on 07/25/2022).

[42] G. Demir and K. Ozmehmet, „Online local learning algorithms for linear discriminant analysis“, Pattern Recognition Letters, vol. 26, no. 4, pp. 421–431, 2005, ICAPR 2003, ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2004.08.005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865504001825.

[43] S. K. Dash, Data Science Blogathon, vol. 10, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/08/a-brief-introduction-to-linear-discriminant-analysis/ (visited on 04/05/2022).

[44] P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, „Linear Discriminant Analysis“, in Robust Data Mining. New York, NY: Springer New York, 2013, pp. 27–33, ISBN: 978-1-4419-9878-1. DOI: 10.1007/978-1-4419-9878-1_4. [Online]. Available: https://doi.org/10.1007/978-1-4419-9878-1_4.

[45] A. Mendlein, C. Szkudlarek, and J. Goodpaster, „Chemometrics“, in Encyclopedia of Forensic Sciences (Second Edition), J. A. Siegel, P. J. Saukko, and M. M. Houck, Eds., Second Edition, Waltham: Academic Press, 2013, pp. 646–651, ISBN: 978-0-12-382166-9. DOI: https://doi.org/10.1016/B978-0-12-382165-2.00259-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780123821652002592.

[46] A. Tharwat, T. Gaber, A. Ibrahim, and A. E. Hassanien, „Linear discriminant analysis: A detailed tutorial“, AI Communications, vol. 30, no. 2, pp. 169–190, 2017. DOI: 10.3233/AIC-170729.

[47] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html (visited on 07/25/2022).

[48] T. Srivastava. (2019), [Online]. Available: https://www.analyticsvidhya.com/blog/2019/08/11-important-model-evaluation-error-metrics/ (visited on 07/07/2022).

[49] J. JORDAN. (2017), [Online]. Available: https://www.jeremyjordan.me/evaluating-a-machine-learning-model/ (visited on 07/07/2022).

[50] K. Nighania. (2018), [Online]. Available: https://towardsdatascience.com/various-ways-to-evaluate-a-machine-learning-models-performance-230449055f15 (visited on 07/07/2022).

[51] (2020), [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/accuracy (visited on 07/08/2022).

[52] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/model_evaluation.html (visited on 08/05/2022).

[53] (2020), [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall (visited on 07/08/2022).

[54] (2020), [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc (visited on 07/08/2022).

[55] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_curve.html (visited on 07/26/2022).

[56] P. Refaeilzadeh, L. Tang, and H. Liu, „Cross-Validation“, in Encyclopedia of Database Systems, L. LIU and M. T. ÖZSU, Eds. Boston, MA: Springer US, 2009, pp. 532–538, ISBN: 978-0-387-39940-9. DOI: 10.1007/978-0-387-39940-9_565. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_565.

[57] S. Pandian. (2022), [Online]. Available: https://www.analyticsvidhya.com/blog/2022/02/k-fold-cross-validation-technique-and-its-essentials/ (visited on 07/08/2022).

[58] S. Vieira, W. H. Lopez Pinaya, and A. Mechelli, „Chapter 2 - Main concepts in machine learning“, in Machine Learning, A. Mechelli and S. Vieira, Eds., Academic Press, 2020, pp. 21–44, ISBN: 978-0-12-815739-8. DOI: https://doi.org/10.1016/B978-0-12-815739-8.00002-X. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B978012815739800002X.

[59] (2022), [Online]. Available: https://www.verbund.com/de-at/ueber-verbund (visited on 04/01/2022).

[60] (2022), [Online]. Available: https://www.verbund.com/de-at/ueber-verbund/besucherzentren/kaprun (visited on 04/05/2022).

[61] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (visited on 07/20/2022).

[62] (2022), [Online]. Available: https://sklearn-genetic-opt.readthedocs.io/en/latest/tutorials/basic_usage.html (visited on 07/21/2022).

[63] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html (visited on 08/01/2022).

[64] (2022), [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.ConfusionMatrixDisplay.html (visited on 08/01/2022).