Marco Wechtitsch, BSc

Intelligent services for performance prediction of students

Master's Thesis

to achieve the university degree of

Diplom-Ingenieur

Master's degree programme: Software Engineering and Management

submitted to

Graz University of Technology

Supervisor

Univ.-Ass. Dipl.-Ing. Dr.techn. Martin Stettinger

Institute for Softwaretechnology

Graz, December 2020

This document is set in Palatino, compiled with pdfLaTeX2e and `Biber`.

The LaTeX template from Karl Voit is based on KOMA script and can be found online: https://github.com/novoid/LaTeX-KOMA-template

Affidavit

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Abstract

Machine learning algorithms have been used in many different application areas. Those range from simple object recognition to complex neural networks. The goal of this thesis is to use supervised machine learning methods to predict a student's future level of knowledge in different courses. Additionally, the exam grades can be predicted with the help of data about students. The used machine learning process can be divided into four phases: data collection, data pre-processing, machine learning phase and interpretation. The most frequently used algorithms are compared. Based on these comparisons, the algorithms with the highest prediction accuracy are applied to predict exam grades and students' level of knowledge. First, the exam grade prediction serves to reduce the exam time. Therefore, students obtain an exam grade prediction during the exam and are able to decide whether to finish the exam or take the predicted grade. Second, the prediction of knowledge level (KL)[1] is useful for students because the KL identifies which future courses will be more important than others.

[1]The knowledge level is a value that describes the knowledge of a user in a specific topic. This value is between 0 and 100 and can be considered as a percentage.

Kurzfassung

Das maschinelle Lernen wird in unterschiedlichsten Anwendungsbereichen eingesetzt. Es erstreckt sich von einfachsten Objekterkennungen bis hin zu komplexen neuronalen Netzwerken. Ziel dieser Arbeit ist es, mit Hilfe von Ansätzen aus dem maschinellen Lernen eine genaue Bestimmung des zukünftigen Wissensstandes der Studierenden in einem bestimmten Bereich vorherzusagen. Der maschinelle Lernprozess umfasst das Sammeln von Daten, das Bearbeiten von Daten, die Auswahl des richtigen Algorithmus bis hin zur Interpretation der Ergebnisse. Dafür wurden die am meisten verwendeten Algorithmen miteinander verglichen und am Ende präsentiert. Anhand dieses Vergleiches wurden die Algorithmen mit der höchsten Vorhersagegenauigkeit angewandt, um die Prüfungsnote und den zukünftigen Wissenstand der Studierenden hervorsagen zu können. Maschinelles Lernen hat erstens den Vorteil, dass die Vorhersage der Prüfungsnote zu einer Verkürzung der Prüfungszeit führt. Aufgrund dessen erhalten die Studierenden während der Prüfung eine Vorhersage der Note. Sie können entscheiden, ob sie die vorhergesagte Note als Prüfungsnote anerkennen oder sie die restlichen Fragen der Prüfung beantworten möchten. Zweitens hat es den Vorteil, dass der aktuelle Wissenstand der Studierenden aussagt, welche zukünftigen Lehrveranstaltungen aufwändig sein werden.

Acknowledgements

First of all, I want to thank my supervisor Martin Stettinger for his constant support, the long discussion rounds and good advices during those meetings.

I would like to thank Florentina Frey for proofreading my master's thesis and all my colleagues and friends for their support. Furthermore, a special thanks to all the people who have participated the evaluation process during these five weeks.

Finally, I also want to thank my parents and family for their support and motivation during some critical phases of my study.

Marco Wechtitsch
Graz, 2020

Abbreviations

KL	knowledge level
PCA	principle components analysis
LOF	local outlier factor
RD	reachability distance
LRD	local reachability density
SVM	Support Vector Machine
MSE	Mean Squared Error
BCE	Binary Crossentropy
CC	Categorical Crossentropy
MLPClassifier	Multi-layer Perceptron classifier
SGDClassifier	Stochastic Gradient Descent classifier
SVR	Support Vector Regression
RFR	Random forest Regressor
KNN	k-nearest neighbors
HAC	hierarchical agglomerative clustering
API	Application Programming Interface
REST	Representational State Transfer
CPU	Central Processing Unit
GPU	Graphics Processing Unit

RDBMS	relational database management system
SQL	Structured Query Language
PK	primary key
FK	foreign key
ID	identity
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
UML	Unified Modeling Language
MOOC	Massive Open Online Course

Contents

List of Figures

List of Tables

1 Introduction & Motivation

Nowadays, information extraction and associated values' prediction have an increasing impact on future decisions. These values are based on the huge amount of collected data in the last decades. The term artificial intelligence is often used in this context. Thereby, machine learning is a part of artificial intelligence which is able to learn by itself based on a set of data.

Many problems in everyday life could be solved with the power of machine learning. Those problems range from company issues to personal issues. Machine learning could not only serve to solve such issues but for example could also save time for companies. Chatbots are able to respond to questions of customers automatically (Nuruzzaman et al., 2018). If an answer is not satisfying, the customer's request can be forwarded to an employee. That means, that an employee only has to answer questions that have not been answered automatically in advance. That is the reason why, one of the main goals of the practical part of this thesis is to develop algorithms for the prediction of exam grades or students' knowledge level with the help of machine learning.

With the power of machine learning techniques, exam times could be reduced at universities. The developed algorithm of this thesis is able to predict students' exam grades. Thereafter, students can be asked if they want to take the whole exam or get the predicted exam grade after taking half of the exam.

Furthermore, the prediction of the KL can play an important role for students. With such knowledge, students would be able to know in advance which courses are going to be more time consuming in the future. This information could be important to plan future semesters.

Thinking in a larger area outside the university's contest, predicting a result during any kind of activity could be a game changer. For example, a survey,

where a huge amount of questions has to be answered. The developed algorithm could reduce the number of survey questions tremendously if the result can already be predicted after a few given answers of a survey. For example, answering 10 out of 50 questions could be enough for a meaningful survey's result.

Moreover, machine learning techniques are represented in a process for a more structured and easier development. This process starts with data selection and data pre-processing techniques for a more precise prediction. Next, an appropriated machine learning algorithm has to be chosen. From a wider perspective, machine learning methods can be divided into three approaches: supervised machine learning, unsupervised machine learning and reinforcement machine learning. The last step is the interpretation of the achieved results.

The goals of this thesis are to predict (1) exam grades and (2) students' KL. The realization of these goals is based on supervised machine learning algorithms. Therefore, a classification approach is used to predict the exam grades and a regression approach is used to determine the future students' KL. All steps for achieving a precise prediction are based on a machine learning process. In order to evaluate the results of the different applied prediction methods a user study have been conducted. The results of the evaluation will point out the limitations of this thesis and the associated machine learning algorithms. The evaluation will help to interpret the expressiveness of the model and additionally, to derive improvements for the future.

2 Related Work

The prediction of exam grades or students' future KL is a currently hot topic due to the large class sizes at universities and even larger class sizes in Massive Open Online Courses (MOOCs) discussed by Meier et al. (2015). With the power of machine learning algorithms, those predictions can be become true. The correct machine learning approach or the collection of data, especially the amount of data plays a crucial role. This chapter gives insights into various attempts to predict exam grades or students' future knowledge level.

Ashenafi et al. (2015) built a prediction model that tries to estimate students' performances on several tasks. The model uses data from semi-automated peer-assessment systems. The data consists of two undergraduate-level computer science courses. Furthermore, Ashenafi et al. (2015) used a supervised machine learning method to predict students' exam scores. They built different multiple linear regression models for the prediction. The Root Mean Squared Error (RMSE) is used to evaluate and compare those models. The prediction is based on fourteen features that describe various students' activities. The data set was prepared with the min-max normalization. This method converts each data point into a value between 0 and 1. Afterwards, the data set is divided into three parts for three different linear regression models. Each of the models has seven initial features. Features are properties of a data set and builds the basis for machine learning approaches. Finally, the model with the least RMSE used all fourteen features for the prediction. The evaluation value of RMSE for the prediction is under 3.0 and is therefore, better than the own created baselines.

Another approach for exam grade predictions of students is researched by Fire et al. (2012). They try to predict individual or group success in exams and courses. Therefore, they divided the features in independent

variables and dependent variables. The number of independent variables is larger than the number of dependent variables and includes age, gender and the educational level of parents. Moreover, dependent variables include course grade and the previous exam grades. Fire et al. (2012) are extracting the log data from the course's website to collect a useful data set. In a pre-processing step, this data set uses network analysis methods to detect the best set of data for an accurate prediction. The collected data contains a course and different types of homework assignments. The assignments were done either alone or in groups. For the extraction of features Fire et al. (2012) developed a Python script which includes the Networkx graph[1] package. Regressions and other machine learning techniques were used to analyse the data set and predict the grade. The goal of this paper is to support students by detecting issues with the course material early. For example, students with less knowledge can be detected early and can be provided with the necessary support.

According to Iqbal et al. (2019), it is essential to predict the grades of students in an early phase. This gives not only the students but also the professors the opportunity to adapt and improve their learning style and teaching style. Different kinds of machine learning techniques are evaluated in this paper. For example, collaborating filtering, singular value decomposition, k nearest neighbours, matrix factorization, and restricted Boltzmann machine (Hinton, 2012). The last machine learning technique is known to have the best accurate prediction of student grades. The goal of this thesis is to show, which algorithm works best and what can be done with the obtained result. On the one side, the prediction shows students the courses that are more difficult in the future. On the other side, professors are able to detect if the content of a course is too simple or too challenging.

The main points for a precise prediction, are the amount of data, the selection of features and the pre-processing of a data set. In neuronal networks, the cross-entropy loss is the most used loss function. The cross-entropy loss function measures the mutual entropy between two probability distributions. Khosla et al. (2020) developed a new loss function that outperforms the existing ones called supervised contrastive learning. This approach is embedding samples with the same classes close to each other and is dis-

[1] https://pypi.org/project/networkx

tancing samples with different classes. The prediction of deep classification models should become more exact. The supervised contrastive approach is widely used in object detection and object recognition applications. Furthermore, the supervised contrastive approach was developed by Google[2] and the Massachusetts Institute of Technology (MIT)[3]. It is said that this development could be a game changer for predictions that are based on deep classification models.

[2]https://www.google.com
[3]https://www.mit.edu

3 Machine Learning

This chapter explains the theoretical background of machine learning that serves as a basis for this thesis. Several terms and algorithms which are used in the next chapters are explained in detail. At first, a general machine learning process, including data selection and data pre-processing techniques is presented. Data selection and data pre-processing are crucial phases for achieving accurate results in machine learning. Next, the selection of the appropriate algorithms are described in detail.

To handle the complexity for the needed algorithms, feature extraction and outlier detection techniques are essential and are presented in this section.

The next sections explain the three widely used different machine learning approaches in detail (supervised machine learning, unsupervised machine learning and reinforcement machine learning).

3.1 General

According to Goodfellow et al. (2016), a machine learning algorithm is an algorithm that learns from existing data. Furthermore, machine learning can be considered as a form of applied statistics. In such a case, the statistical algorithms are very complex. In fact, the estimation is based on functions that are often calculated from computers. A function $f(x)$ can determine the result of a machine learning algorithm. In contrast to classic software development processes, the key of machine learning is the independent learning approach of data sets.

Figure 3.1 shows a general machine learning process approach. It starts with a problem that should be solved or a goal that should be reached. Next, the

selection of a data set should happen. Either the data set is prepared for the data pre-processing step or the source step is repeated until the data set is ready for further steps. Thereafter, the data pre-processing step filters out unwanted data in a data set. Moreover, the features for the algorithm are selected. The output of this step is the input for the learning phase. The machine learning phase applies an algorithm on the pre-processed data and delivers a model for further use. Furthermore, the result of a machine learning algorithm is interpreted and analyzed. If the output delivers the desired result a model can be used for predictions. If the partial result in the steps is not satisfying, steps can be repeated. The idea of a machine learning process is based on Ge et al. (2017).

The following subsections explain a machine learning process, that is mentioned in Figure 3.1, in detail. In addition to that, common terms are explained for a better understanding.

3.1.1 Data Selection and Preparation

Machine Learning is based on data sets. A data set consists of samples and features as shown in Table 3.1. A feature describes the properties of the data. The Table 3.1 consists of the features: age, gender, blood type and coronavirus. Those properties describe a data set. The selection of features is important and influences a result tremendously. A data object or sample represents the values of features. Referring to the example of Table 3.1 a row represents a sample and a column describes a feature. A well-defined and structured data set is the key to a precise result. This process can be subdivided into feature selection and feature extraction. (see Section 3.2). A common method to represent such a well-defined data set in a structured form is a data matrix as shown in Table 3.1. A result depends on features and the quality of a data set. A data set can be split into training set and test set. A training data set is a set of samples used for learning patterns and relationships within the data. A test data set is independent of the training data set and is not used for training models. Machine learning algorithms “do not know” the test data set during the learning of patterns and relationships. If a trained model has an accurate prediction on a test data set, the model has an accurate prediction on unknown data.

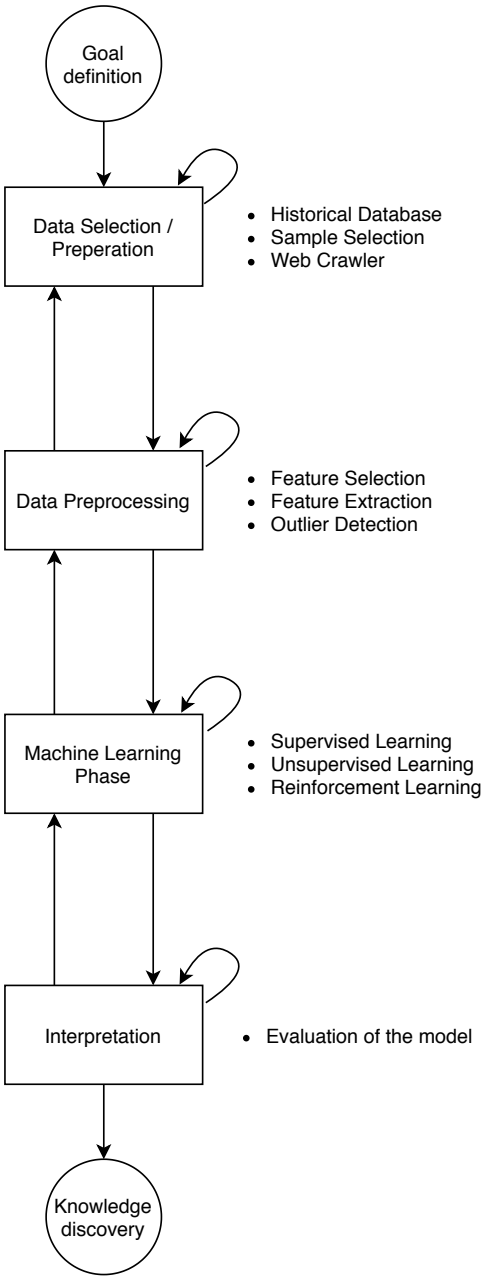


Figure 3.1: General machine learning process.

According to Ge et al. (2017), the initial step for machine learning is data selection and preparation that gives an overview of the processed data. The initial step starts with the collection of data. Therefore, a set of samples is filtered out from a raw data source and is modelled to a structured data set (Ge et al., 2017). Alternatively, the data can be collected from the web. In this case, a web crawler which is a specialized application that scans and analyses web pages, is used to store the data.

person	age	gender	blood type	coronavirus
1	45	m	A positiv	yes
2	50	m	0 positiv	yes
3	30	w	B positiv	no
4	70	w	A negativ	yes
5	81	m	AB negative	no
6	25	w	0 negative	no

Table 3.1: A sample data set describing humans attributes in correlation to the coronavirus infection.

3.1.2 Data Pre-processing

After the data collection process, the data pre-processing is executed to improve the quality of data. The goal of the pre-processing step is to filter out unwanted data. Unprepared data sets can be *incomplete*, which means they have missing values, noises, or incorrect data. This could lead to a biased result. There are many methods and algorithms, such as the expectation maximization algorithm or the principal component analyse which deal with these problems (Imtiaz et al., 2008). After cleaning a data set from the biased results mentioned above, the feature selection and extraction process, which is explained in Section 3.2, can start. Another step in pre-processing is an outlier detection, which is explained in Section 3.3.

3.1.3 Machine Learning Phase

The machine learning phase starts if the data pre-processing step is finished. First, an appropriate machine learning algorithm has to be selected. There are three main learning algorithm approaches: supervised learning, unsupervised learning and reinforcement learning, as seen in Figure 3.2. Which algorithm should be used depends on the starting problem/goal, the available data, and the preferred result. An implemented machine learning algorithm is trained with a training data set. After these steps, an algorithm or model is trained with a training data set. The output is a model that is used for the prediction of new samples.

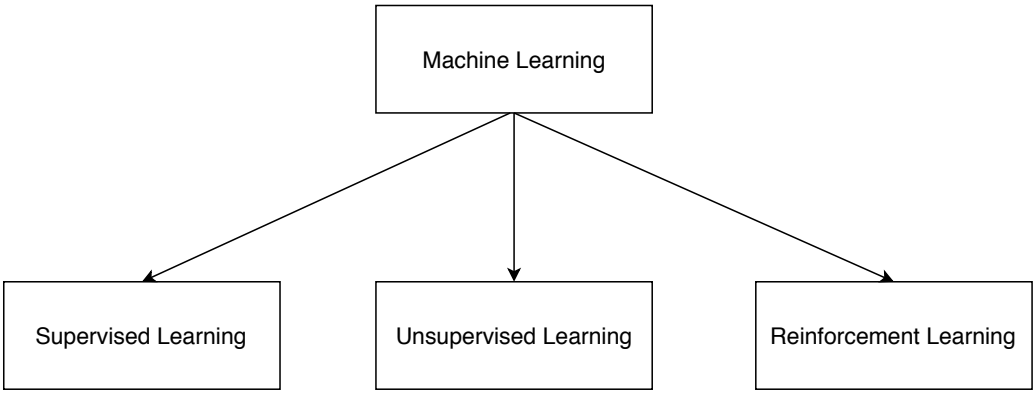


Figure 3.2: Types of machine learning.

3.1.4 Interpretation

This subsection describes how to interpret the output of the machine learning process. After a model has been trained the model has to be interpreted and evaluated. First, some frequently used evaluation tools are explained. After that, the overfitting and underfitting problem is discussed.

Evaluation

The evaluation of a model is an important step before a model can predict real-life problems. It is the foundation for further predictions. There are meaningful values that describes the performance of a model. Classification evaluation and regression evaluation have to be distinguished. The terms accuracy, precision, recall, precision-recall and F1 score are relevant for the classification evaluation. Whereas mean squared error, root mean squared error and mean absolute error for the regression. The next paragraphs explain the calculation of these values in detail. The matrix in Table 3.2 shows a simple binary classification example. True positive tp means that are also predicted positive by a model and they are actually positive. If a model predicts the means to be positive and they are actually negative, then it is called false negative fn. True negative tn is the value if the model predicts negative and they are actually negative. Finally, false negatives fn occur when a prediction is negative and the real case is actually positive.

		Real class	
		1	0
Prediction	1	true positive (tp)	false positive (fp)
	0	false negative (fn)	true negative (tn)

Table 3.2: Visualization of a simple classification example.

The following terms are based on the ideas of Saito et al. (2015) and Powers (2011).

Accuracy The accuracy A describes how accurate a model learns with a given data set. Moreover, it can be divided into train accuracy and test accuracy. The differences between these two values are the data sets, which consist on the one hand, of training data and on the other hand, of test data. The higher the percentage of the values the more precise is the output of the prediction. But on the other side, a high percentage can also be negative for the prediction. Let's assume a prediction accuracy of 99 %. This can be interpreted as bad because a model depends too much on training data. The value is not a clear indicator because the value has no validity if the classes

are imbalanced. The calculation of the accuracy is described in Formula 3.1. Finally, the significance of this value depends on an application and therefore, it should not be the only value for evaluating a model.

$$A = \frac{tp + tn}{tp + fp + tn + fn} \tag{3.1}$$

Recall The recall R describes the proportion of positive instances tp out of the total actual positive instances ($tp + fn$). It indicates the proportion of samples of the positive class that is detected by a model. For example, the value should be high for the parental controls of movies. On the other side, this value should be lower if spam emails are filtered out. Formula 3.2 shows the calculation of the recall.

$$R = \frac{tp}{tp + fn} \tag{3.2}$$

Precision The precision P shows the proportion of positive instances tp out of the total predicted positive instances ($tp + fp$). This value is a good measure to determine whether the number of false positive fp is high. Formula 3.3 displays the calculation of precision P in detail. For example, it has no consequences if a false positive fp is classified as an adult's movie instead of a children's movie. In case of an email spam filter, a false positive fp describes that an email has been detected as spam but should be detected as non-spam. Therefore, users could lose important emails.

$$P = \frac{tp}{tp + fp} \tag{3.3}$$

Precision Recall trade-off The goal is to bring both precision and recall close to one. If the recall is increased, more *ones* (tp) have to be predicted. However, by increasing the number of *ones*, the number of false positives (fp) are increased. This has the consequence that the precision is decreased. Without changing the data set or the model the precision and recall cannot be improved together.

F1 score The F1 score is the mean of recall R and precision P. The precision recall trade-off can be quantified by the F1 score (see Formula 3.4). This score is a good measure to find a balance between recall and precision.

$$F_1 = \frac{2 * P * R}{P + R} \tag{3.4}$$

Mean Squared Error The Mean Squared Error (MSE) is one of the most used methods to evaluate a regression. It is the average of the squared difference between a target value and a predicted value. The more precise the predicted values are the less an error can occur. Formula 3.5 depicts the MSE, whereas n is the number of samples, y the target value and z the predicted value.

$$\text{MSE} = \frac{1}{n} * \sum_{i=1}^{n} (y_i - z_i)^2 \tag{3.5}$$

Root Mean Squared Error The Root Mean Squared Error (RMSE) measures the average of the squares of the errors. In contrast to MSE, the target value y is subtracted from the predicted value z. Moreover, n is the number of samples. The output is a number that summarizes the error of a trained model. By squaring the difference, the RMSE ignores the data which is different to the other data in the data set of the subtraction. Formula 3.6 displays the calculation of RMSE.

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (z_i - y_i)^2}{n}} \tag{3.6}$$

Mean Absolute Error The Mean Absolute Error (MAE) is the absolute difference between the target value y and the predicted value z as shown in Formula 3.7. This value is not suitable for the cases where you pay attention

to data which does not fit into an expected data set because this method is robust against it.

$$\text{MAE} = \frac{1}{n} * \sum_{i=1}^{n} |(y_i - z_i)| \tag{3.7}$$

Overfitting and Underfitting

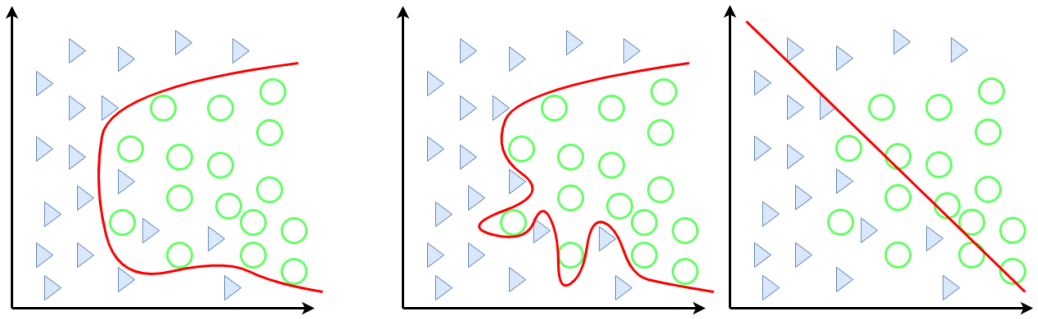
Overfitting is the term that describes a situation when a model performs well on training data but not on test data. A model learns details and noises from a training data set and this leads to an inaccurate prediction. Underfitting occurs when a model performs poor on a training data set.

The overfitting and underfitting problem can be explained as follows (see Figure 3.3). In order to distinguish whether an animal is a cat or a dog a model learns the relationships in a data set. In this example scenario, the triangles represent cats, the circles are dogs and the red line represents the model function. Overfitting occurs if a model function has too much complexity or noises. For instance, the samples in a data set are described too much in detail as the length of the fur. On the contrary, underfitting appear if a model function has not enough complexity. For example, the data set consists of few samples.

In general, overfitting leads to high variance and a low bias. Underfitting has a high bias and a low variance. Reducing complexity or learning iterations can prevent overfitting. Underfitting on the contrary should increase the complexity or iteration count. Furthermore, the features of a model should be increased.

Baseline

The evaluation of a machine learning algorithm can be done with a baseline. A baseline is the value that should be exceeded from a model. A baseline can be a static number or a value that is calculated from an algorithm.



(a) appropriate fitting (b) overfitting (c) underfitting

Figure 3.3: (a) shows an appropriate fitting which leads to a high and correct prediction. (b) displays an overfitting problem, this arises from the fact that the model function has too much complexity to fit the correct function. The underfitting problem (c) occurs if the model function has not enough complexity to fit the correct function.

Moreover, a baseline is the simplest approach to evaluate a model. Two common baselines are:

- Simple baselines
- State-of-the-art baselines

The developed algorithms for this thesis are evaluated with two simple baseline approaches. Therefore, the random baseline and constant baseline approaches are described in detail. The random baseline algorithm predicts random values in a given range. For instance, this algorithm predicts values between ten to twenty. Afterwards, the predicted values are compared with a test data set. Based on the output of the comparison, a machine learning algorithm can be evaluated. Instead, the constant baseline predicts always the same predefined value. The output of these models can be used as a first threshold for a supervised machine learning algorithm. Everything that is above the baseline values is not sufficient for a real world problem.

3.2 Feature Selection and Extraction

Features play an important role in machine learning algorithms because the quality of features in a data set has a major impact on the accuracy of a machine learning algorithm. The samples and features of a data set are key to a precise prediction.

Not every feature leads to a precise prediction because features can influence the output of a machine learning algorithm negatively. According to Blum et al. (1997), it is hard to distinguish between the relevant and irrelevant features because mostly a set of features is large and contains a lot of not meaningful features. Features should be ranked according to their relevance and afterward the first n features should be selected for a machine learning algorithm. Thereafter, the output of a machine learning algorithm with the selected features are evaluated. If the output is not adequate, the ranking step is repeated. The ranking can be done manually or with a feature importance algorithm. For example, extra tree classifier[1] can be used for the estimation of feature importance.

A more sophisticated approach is the so called feature extraction. The goal of feature extraction is to reduce the number of features in a data set. The input of a feature extraction algorithm consists of existing features and the output is a smaller set of new generated features that combine the input features. Feature extraction creates a reduced subset of new features based on features of the base data set. Therefore, the complexity of a model can be reduced.

One of the most used algorithm for feature extraction is the principle components analysis (PCA). This algorithm belongs to the category of unsupervised machine learning algorithm, therefore a data set is not labelled. The PCA is used for dimension reduction. Dimension reduction is important to reduce the complexity of a model. It transforms points from a high-dimensional space into a lower space. The goal is to reduce the dimensions and to keep most of the information. A data set is presented in a matrix D. First, the averages of the data columns are calculated and shifted by the matrix D to get a new matrix X. Next, a covariance matrix S is calculated

[1] https://scikit-learn.org

to retrieve the eigenvectors from the covariance matrix. The matrix of eigenvectors is an orthogonal matrix. This matrix M is a $n \times n$ matrix, where n represents the number of features. This matrix connects the features with each other. After the computation of the eigenvectors and eigenvalues corresponding to the M largest eigenvalues the principal components data can be determined. The output of the PCA are new variables that can be represented as a linear combination of the input variables. The PCA returns the maximum possible information. (Bishop, 2006)

Summarized, feature selection technologies choose a set of features out of the whole data set. Meanwhile, the feature extraction calculates useful features from the initial features.

3.3 Outlier Detection

A so called outlier is a data point that differs significantly from the rest of the data points (see Figure 3.4). Therefore, outliers cause noises in data sets and thus, affects the prediction negatively. A related term to outlier is the novelty. The term novelty describes whether a new observation is an outlier or not. Both are used for anomaly detection.

This section explains a common outlier detection algorithm that is used in machine learning.

3.3.1 Isolation Forest

A common way of performing outlier detection is to use isolation forest detection. The term isolation means the separation from one instance to another. Isolation forest is a tree based approach that uses decision trees with two leaves as binary trees. The tree is built randomly and the separation of instances is repeated recursively until all instances are isolated. (Liu et al., 2008)

The algorithm starts with a given data set of n instances. The isolation tree will recursively divide the data set by picking randomly an attribute and

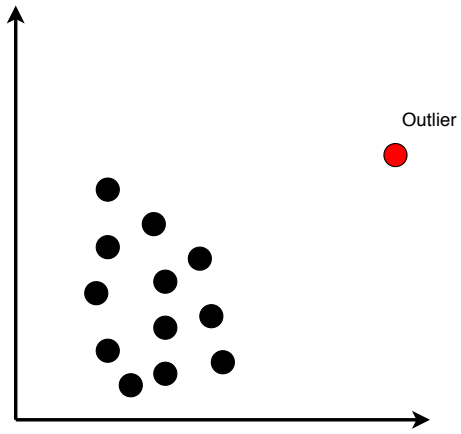


Figure 3.4: The red point indicates an outlier in a data set.

a split value from the data set. If the tree reaches a defined height limit, the data set is one, or all values of the data set are equal, the termination condition is fulfilled. Moreover, N is either an internal or an external node. Figure 3.5 shows an isolated tree. An internal node has an attribute, a split value, and two daughter nodes (T_l and T_r). The attribute and the split value partitioned the data points into T_l and T_r. The external node has no children, it is the last leaf on the tree. (Liu et al., 2008) (Medium.com, 2020)

The advantage of this method is usually a short path for anomalies. Following that, fewer instances of anomalies are found in paths and distinguishable attributes are separated earlier. Outlier detection algorithms recognize similar data first. Afterwards, the outliers are identified based on the previously calculated data. In contrast, the isolation forest tree recognizes the outliers first and therefore, this algorithm is faster in detecting outliers. (Liu et al., 2008)

3.3.2 Local Outlier Factor

One efficient way of performing outlier detection is to use the local outlier factor (LOF) algorithm. The output of the LOF is a score that tells if a given data point is an outlier or not. The score measures the local density of a data point to its neighbours. If the data point has a high density, it is an

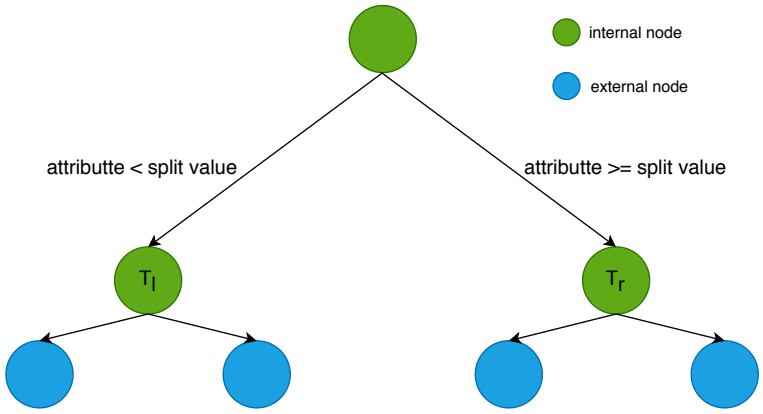


Figure 3.5: An isolation tree consists of internal and external nodes and the structure of the tree is represented as a binary tree's structure.

outlier. The algorithm is similar to the *k*-means algorithm (see Section 3.5.1). According to Breunig et al. (2000), the *K*-distance and *K*-neighbours are calculated first. The *K*-distance is the distance between the selected data point and the *K*-nearest neighbours. *K*-neighbours $K(x)$ are a set of points of the neighbours within a certain circle of radius *K*-distance (see Figure 3.6. Next, the reachability distance (RD) is determined. Figure 3.6 illustrates the idea of a reachability distance. For instance, if a point (*F*) is far away from *A*, then the reachability distance between these two points is the actual distance (*distance(A, F)*). However, if the points are close (point *A* and *D*) within the *K*-distance the actual distance is replaced by the *K*-distance (*K-distance(A)*). Formula 3.8 shows the calculation of the reachability distance.

$$RD_k(x,y) = max\{K - distance(x), distance(x,y)\} \tag{3.8}$$

Based on the example in Figure 3.6, the RD is needed to calculate the local reachability density (LRD) of data point *A*. LRD is the inverse of the average RD of *A* to its neighbours (see Formula 3.9). The result describes the distance of a point to the nearest cluster of points. If the value is high the point *A* is close to the nearest cluster.

$$LRD_K(x) = 1/\left(\frac{\sum_{y \in N_K(x)} rd(x,y)}{|N_K(x)|}\right) \tag{3.9}$$

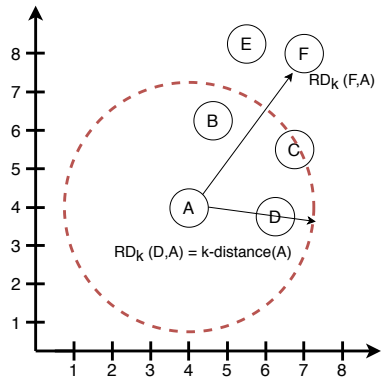


Figure 3.6: A set of K-neighbours that is in a range of K-distance of point A. The data set includes six points (A, B, C, D, E, F) whereby the points B, C and D are in the K-distance of A. The red circle indicates the K-neighbours of A with K-distance = 3 and $RD_k(D, A)$ the reachability distance between point A and D.

Finally, the LOF can be calculated. The result of LOF captures the degree of A and if it is an outlier or not. In most of the cases, if LOF is greater than one, the point is an outlier. Formula 3.10 shows, that the higher the local reachability density is and the lower the local reachability densities of the K-nearest neighbours are, the lower is the LOF.

$$LOF_K(x) = \frac{\sum_{y \in N_K(x)} \frac{LRD_K(y)}{LRD_K(x)}}{|N_K(x)|} \tag{3.10}$$

3.4 Supervised Learning

Supervised learning is a method of machine learning which makes precise predictions based on the analysis of input data with predefined target variables. Moreover, supervised machine learning algorithms are used to detect patterns in a data set which is called training phase. The output of a algorithm is a model that can predict target values of new input samples. The result of the model is influenced by both training data and features. Examples for training data properties are the size, distribution or

completeness, depending on which algorithm is chosen. Furthermore, a correct selection/extraction of relevant features is important.

A complete training data set could be the sample matrix which is displayed in Table 3.1. Features could be age, gender, blood type and the labelled target value if, for example, you are infected with the coronavirus or not. As described above, an algorithm detects patterns in a data set and derives rules for an accurate prediction of target values. Patterns for the sample matrix (see Table 3.1) could be the age in combination with the gender. Three persons are infected with the virus, most of them are men and every person is over 45 years old.

Formula 3.11 shows the simplest function of a supervised learning algorithm. The function $f(x)$ predicts a target value y for an input sample x.

$$y = f(x) \tag{3.11}$$

A mixed form of machine learning algorithms is semi-supervised learning. This method labels one part of training data. The other part is unlabelled and is assigned by the algorithm itself. An example of a semi-supervised learning method is a data set with animal images. Parts of a data sets are labelled as the images contain a cat or a dog. The rest of the images are labelled by the algorithm.

Supervised learning algorithms can be divided into classification and regression. The points below explain the two types in detail and give an example algorithm for each type.

3.4.1 Classification

A classification is a type of machine learning which predicts a class/target value/label of given data points. Thereby, an algorithm learns patterns and relationships in a data set and predict a class for new samples. First, an algorithm has to be trained with a data set that consists of data points and target values. The output of a trained algorithm is called a model. The goal of a model is to take a sample as input and assign it to a category that a model

has learned. As mentioned above, a model maps input variable with associated functions to a discrete output variable. (Goodfellow et al., 2016)

An example of a classification task is the recognition of objects in images. The input is an image and the output are the detected objects in the image (Goodfellow et al., 2016). For instance, a classification algorithm detects cars and traffic signs on a traffic control camera. Another example is the detection of an email being spam or not. In this case, the email text represents the samples and the labels indicate whether the email is spam or not.

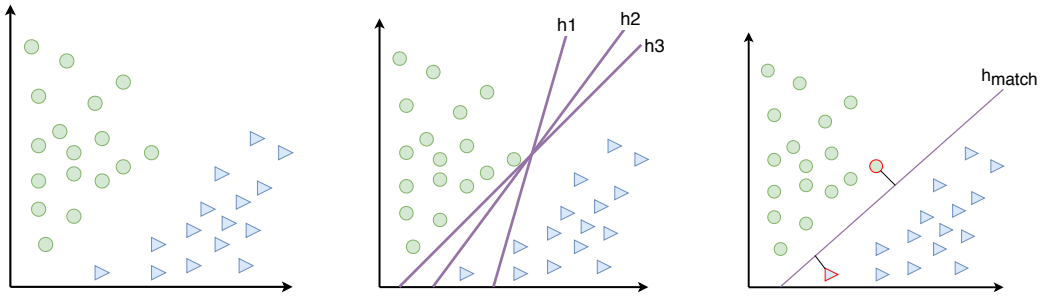
According to Goodfellow et al. (2016), a classification gets more challenging if a training data set is incomplete. This means some data points are missing. On the one hand, the samples of the missing data points can be deleted. On the other hand, if most of the data points for a feature are missing, then the whole feature is deleted. Another approach is to estimate the missing values with a probability distribution over the samples.

The next subsections describe common classification algorithms that are used in modern applications. No master classification algorithm can be used for all classification problems. An algorithm depends on a data set and the addressed problem.

Support Vector Machine

Support Vector Machines (SVMs) are used for classifying samples. Furthermore, SVM can be used for regressions and novelty detection (Bishop, 2006). The term machine does not indicate a conventional machine, but rather machine learning. As a classification algorithm, SVM is learned by examples to assign labels to objects (Boser et al., 1992). The goal of SVMs are to find a hyperplane that classifies the training data best. A hyperplane is a line which separates data into classes. The dimension for a hyperplanes depends on the number of features n. The goal of the SVM algorithm is to detect the best hyperplane that has the maximum margin between the n-dimension data. The maximum margin symbolizes the maximum distance between the single features. In Figure 3.7 (a) two features of a data set are plotted. Figure 3.7 (b), depicts possible hyperplanes that can be seen but none of them has the maximum margin. The hyperplane $h1$ does not separate the classes but

$h2$ and $h3$ do (see Figure 3.7 (b)). However, it does not maximize the distance between the two classes. The maximum margin of the data set in Figure 3.7 (a) is seen in Figure 3.7 (c). The hyperplane h_match has the maximum distance between the two features. Based on the example in Figure 3.7, the best matching hyperplane for this data set is h_{match}. The support vector would be the two data points that have a red border (see Figure 3.7 (c)). This data point influences the hyperplane and helps the algorithm to build the SVM. The maximization of the distance is calculated with a cost function (Bishop, 2006). A cost function is a function that measures the performance of a machine learning model.



(a) Two dimensional data set.

(b) Two dimensional data set with three hyperplanes (h1, h3 and h3).

(c) Two dimensional data set with the best matching hyperplane.

Figure 3.7: (a) shows the data set with two classes (green circles and blue triangles). (b) displays the data set with hyperplanes h1, h2 and h3. The best matching hyperplane for this dataset is shown in (c).

Not all data points are so well distributed and a optimal hyperplane cannot separate the classes as in Figure 3.7. Soft margin describes the term if some data points of a training set allow to be misclassified. Therefore, an optimal maximum margin between to classes is not possible.

Logistic Regression

A logistic regression algorithm is a discrete function, that assigns an input sample to a class. The logistic regression belongs to probability models.

Therefore, it calculates the probability of a sample to a class. (Bishop, 2006)

The logistic regression uses a *sigmoid* function as cost function that returns a probability value of input data points. According to Pant (2019) a logistic regression can be divided into binary functions and multi-linear classes. The output of the function is between 0 and 1 ($0 <= f(x) <= 1$). Dreiseitl et al. (2002) states, that the relation between x and y can be seen as a probability distribution $P(x, y)$ (see Formula 3.12).

$$P(y|x, z) = f(x, z) \tag{3.12}$$

The maximum likelihood estimation is used for the determination of z. As mentioned before, the function $f(x, z)$ is the logistic function and can be described as seen in Formula 3.13.

$$f(x, z) = \frac{1}{1 + e^{-z}} \tag{3.13}$$

Random Forest Classifier

The random forest algorithm is another classification method. Random forest classifier consists of trees (Liaw et al., 2002). This algorithm can be used for both, classification and regression. Furthermore, it has only two parameters and therefore, it is easy to use. The first parameter is the number of trees in a forest and the second one explains the number of features randomly selected in a random tree at each node at the creation step (Khoshgoftaar et al., 2007). Each tree predicts independent to each other a class. The prediction with the highest number of classes is the predicted class for the random forest classifier. Figure 3.8 shows four decision trees. Each one of the trees predicts a class ("0" or "1"). The random forest classifier predicts the class "1" because this class was predicted three times and the class "0" was predicted one time.

The following Formula 3.14 describes the random forest classifier algorithm, which is based on Breiman (2001). Every tree k consists of a random vector Θ_k which is separated from the previous random vectors $\Theta_{1,\ldots,}\Theta_{k-1}$. The distribution for the generation of trees are the same. The variable x is the data sample that should be assigned to a class. Because each tree is

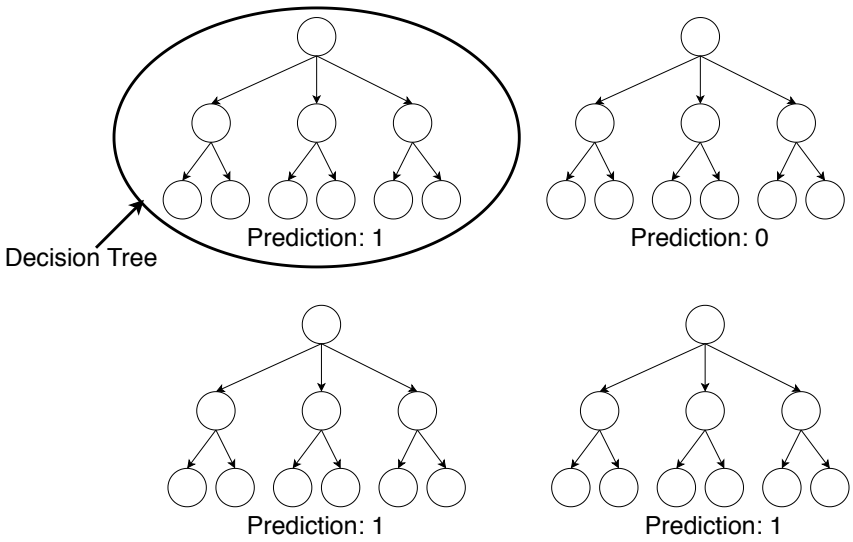


Figure 3.8: This random forest classifier uses four decision trees for a prediction. Three decision trees predict the class 1 and one decision tree predict the class "0".

independently distributed, the predicted class is the class that occurs most often in the forest. Each tree votes the most popular class for the input vector x.

$$[h(x, \Theta_k), k = 1, ...] \tag{3.14}$$

As already mentioned, the algorithm randomly chooses n samples from a initial training data set. Next, k trees are built samples under the following two criteria (Khoshgoftaar et al., 2007):

- The decision tree is unpruned
- Each node in the tree has a set of randomly selected candidate attributes from subset $(x_1, x_2, ..., x_f)$ where f represents the number of randomly selected features

During the prediction phase of an example x, each tree predicts a majority class and the class with the most votes gets the result of the model. For a better understanding Figure 3.9 gives a visual representation.

According to Breiman (2001), a random forest classifier is robust against overfitting and unbalanced training data set. Overfitting is reduced by

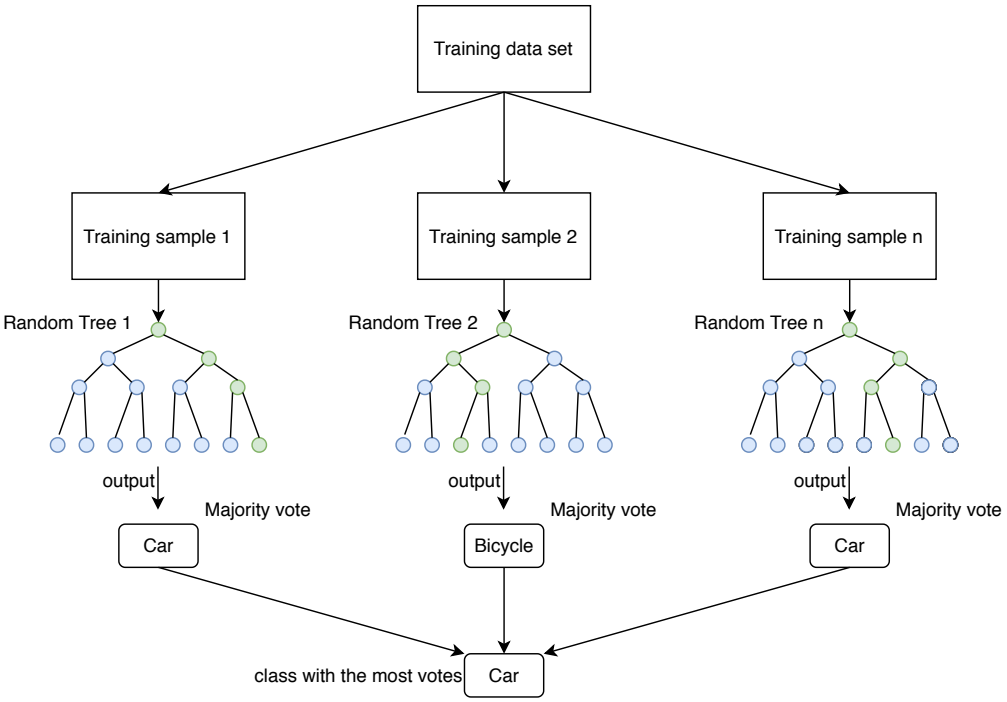


Figure 3.9: A random forest classifier can be used to choose the best vehicle for a person. In this example, the best vehicle is a car. The green points display the best matching path for the tree and the blue ones are the other nodes. The training data is split into n trees.

adding more trees to a forest. Through a tree structure, outlier and noises do not matter that much. The random forest classifier is a powerful algorithm because the trees have a low correlation. Therefore, the trees protect each other from errors.

Neuronal Networks

Neuronal networks are multi-layer networks of neurons which are used for solving machine learning tasks. A neuron is a function that takes an input and provides an output by applying a function with the given input parameters. This function is called activation function. A neuronal network

consists of a training data set with labelled features and samples. In general, neuronal networks consist of

- an input layer
- one or more hidden layers
- and an output layer.

Each of the layers is part of the neuronal network and used to calculate the probability of a sample to a class. Figure 3.10 shows a simple neuronal network. Neuronal networks used for classifications are typical deep neuronal networks that belongs to feedforward networks (Mohsen et al., 2018). This means, that each layer is connected to every other layer and the output of the previous layer is the input of the next layer. For instance, the network depicted in Figure 3.10 has an input layer with two inputs (i_1, i_2), a hidden layer with two neurons (h_1, h_2) and an output layer with one neuron (o_1). The hidden layer connects the input layer and the output layer. The initial inputs are also called input neurons and represents the number of features. In the domain of image recognition input neurons are in most case pixels of an image.

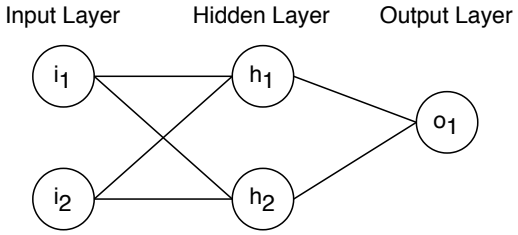


Figure 3.10: A simple neuronal network.

Usually, a neuronal network has more hidden layers and more units than Figure 3.10 shows. This can increase the complexity and the nonlinear relationships of a neuronal network (Mohsen et al., 2018). Neurons have activation functions which take some inputs $x_{1,\dots,n}$ and deliver one output y. The activation function consists of weights $w_{1,\dots,n}$ and a bias b (see Formula 3.15). Weights represent the importance of a feature and a bias is a constant vector to offset a result. Each layer has its own weights and bias.

$$y = f(\sum_{i=1}^{n}(w_i * x_i) + b) \tag{3.15}$$

The *sigmoid* function or *softmax* function are often used as activation functions.A loss function calculates the error of a neuronal network model. The better a prediction is, the smaller is the output of a loss function. A goal is to minimize the loss during a training phase of a neuronal network. Examples for a loss functions are MSE, Binary Crossentropy (BCE) or Categorical Crossentropy (CC). After a loss function was applied the backpropagation phase starts. The output of a loss function should be decreased to the point where a model has the best prediction value. The gradient descent is a part of the backpropagation that recalculates the weights and biases iterative until a loss will reach a global minimum of the function. Examples for the gradient descent are Batch Gradient Descent or Stochastic Gradient Descent. (Verma, 2019) (Mahmood, 2019)

Multi-layer Perceptron classifier (MLPClassifier) and Stochastic Gradient Descent classifier (SGDClassifier) are neuronal networks examples that use a gradient descent algorithm.

3.4.2 Regression

In contrast to classification, a regression predicts numerical values instead of categories. A regression is a supervised learning method that consists of a training data set with features and labelled samples. A function $y = f(x)$ maps the input variables x to the output variables y. For example, the price prediction of a car can be applied with a data set that contains car data. The features can be age, color, brand, condition of a car, and price. The labelled feature is the price. The output of a regression is a real value for example, between 1.5 and 10.99.

A lot of different regression algorithms exist. Support Vector Regression (SVR) is similar to the classification algorithm SVM (see Section 3.4.1) and the Random forest Regressor (RFR) to Random Forest Classifier (see Section 3.4.1). In the following regression algorithms are described.

Linear Regression

Linear regression belongs to the most applied regression. Moreover, linear regression consists of a linear model that connects independent variable x with dependent variable y. A linear regression takes a vector $x \in \mathbb{R}^n$ as input and predicts a value $y \in \mathbb{R}$ as output. The linear regression depicted in Formula 3.16 has a vector of parameters (ω) that controls the output of the prediction. The output of a prediction can be affected by each feature's ω. If ω is large, then it has a huge impact. Therefore, γ is the predicted value for a target value y. (Goodfellow et al., 2016)

$$\gamma = \omega^T * x \tag{3.16}$$

The goal is to find the best parameter vector ω for each feature. Therefore, the error between the actual output and the predicted output has to be minimized. A cost function helps to find the best vector of ω. Moreover, gradient descents are used to update the vector ω and to reduce the output of a cost function. The gradient descent is an iterative method that uses the partial deviation. The idea is to start with random values for the vector ω and to update the vector after each iteration till a minimum is reached.

Lasso[2] and Ridge[3] are other regression algorithms that are based on the linear regression method.

k-nearest neighbors (KNN)

The k-nearest neighbors (KNN) can be used both for regression and classification problems. The basic idea of this algorithm is that points which are close to each other are also similar to each other. KNN calculates the distances between the points every time. This means, that the algorithm does not create a model. Instead, the classification or regression will do the prediction in real time. The main part of KNN is to calculate the distance

[2]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
[3]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html

between each point. One of the most used algorithms for distance calculation is the Euclidean distance. The most challenging is the estimation of K which is the number of nearest neighbors.

According to Keller et al. (1985) the algorithm is as follows:

1. Let $\omega = x_1, ..., x_n$ be a data set of n labelled samples, y the input of a sample that should predict the target value
2. Choose K ($1 \leq K \leq n$)
3. For each sample x_i in ω the distance between x_i and the other samples ω is computed
4. Order the computed distances ascendingly
5. Choose the supervised machine learning approach
    a) Regression: calculate the root mean square deviation and pick the first K neighbors, that have the lowest mean
    b) Classification: determine the majority class of first K neighbors

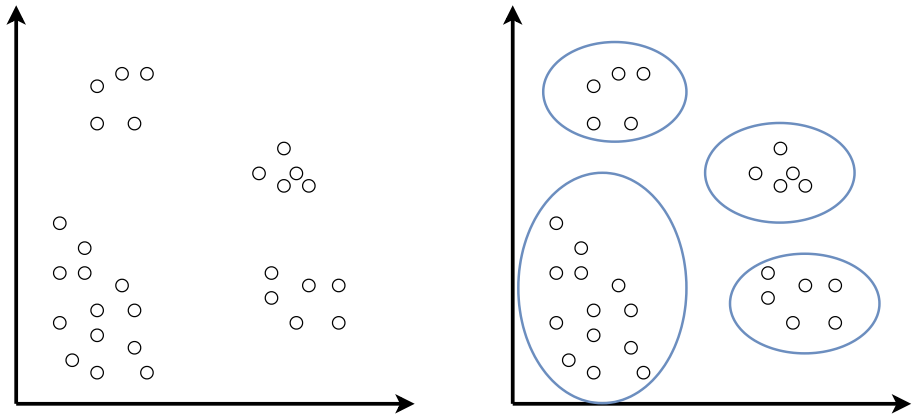
3.5 Unsupervised Learning

The second machine learning type is unsupervised learning. In contrast to supervised learning, this type of machine learning requests no labelled data set. Unsupervised learning algorithms "have to" find patterns on their own. The goal is to detect unknown structures and relationships in the data set to derive a model. Furthermore, unsupervised machine learning methods can find features that can be useful for supervised machine learning methods. Unsupervised learning methods are divided into clustering (see Section 3.5.1) and association (see Section 3.5.2).

In sections feature selection and extraction (see Section 3.2) and outlier detection (see Section 3.3) some unsupervised methods such as PCA or LOF have been presented in detail.

3.5.1 Clustering

Clustering identifies groups of instances (1) to maximize the similarity of a group and (2) minimize the similarity between groups. Figure 3.11 shows a simple clustering approach. The points in the left figure represent the data points. Each axis could be a feature as for instance the height and weight of a person. The blue ellipses in the right figure show the four clusters that have been detected by a clustering algorithm. Based on these findings, a new input vector can be clustered. The goal is to find patterns in an unlabeled data set. Data points with high similarity are in the same cluster. This similarity is often shown as a distance function, where a lower distance represents a higher similarity. The Euclidean distance is often used for distance calculation (Madhulatha, 2012) . Clustering can be divided into iterative, hierarchical, density based, or stochastic based clustering. The next sections describe the k-means (see Section 3.5.1) and the hierarchical algorithm (see Section 3.5.1) in detail. (Grira et al., 2004)



(a) Data set of samples. (b) Clustering of a data set.

Figure 3.11: (a) shows an initial data set, whereas (b) displays a data set after the k-means clustering algorithm was used.

K-Means

The K-means algorithm is a partitioning clustering algorithm which is part of iterative clustering. Moreover, K-means is one of the simplest unsupervised machine learning algorithms. The input of this algorithm is a data set which includes features and samples. The goal is to categorize samples of a data set into k groups.

First, the number of clusters k for a data set D are initialized. The idea is that each of the k clusters has a center, which are called centroids $(c_1, ...c_k)$. The centroids are randomly selected in the initialization phase. Next, each data point ($d \in D$) is assigned to the closest centroid. For that reason, the distances are squared and summed to the closest centroid. If each point is assigned to the closest centroid, the centroids are recalculated and updated. Next, the data points calculate the distances to each centroid and assign themselves to the closest centroid again. If no data point has to be reassigned the algorithm is finished. (Bishop, 2006)

Figure 3.12 shows the centroids' movement from the initial phase, where the centroids are selected randomly till the last iteration takes place. In the last iteration, the centroids have the best mean and the lowest cost.

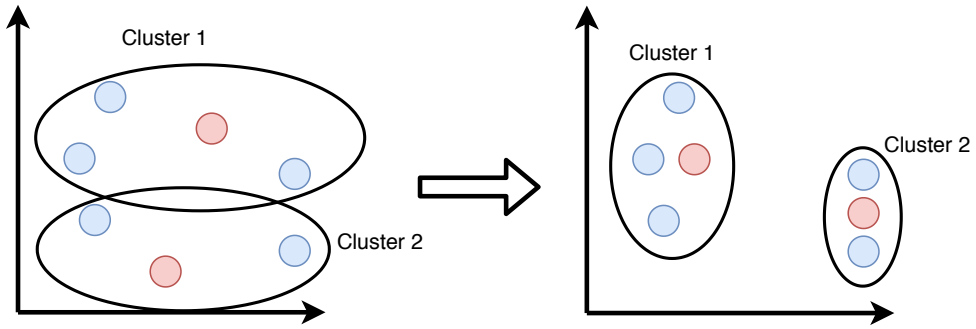


Figure 3.12: Illustration of the K-means algorithm from initial step till the last step. The blue points represent the data points and the red ones the centroids. Each sample is assigned to the cluster with the smallest error.

Hierarchical Clustering

The hierarchical clustering method has two approaches for the computation of clusters. The first approach is called hierarchical agglomerative clustering (HAC). Every data point is seen as a cluster and is merged together. The second divisive approach has the inverse idea. It starts with a single cluster of all data points and splits it into unique pieces. The first approach is also called agglomerative approach and it is based on the bottom-up principle. Meanwhile, the other approach is called divisive approach and is following the top-down principle.

The agglomerative method does not represent the clusters by centroids. Accordingly, there is not a fixed number of clusters. Zhu et al. (2008) explains this algorithm as follows:

1. Initialization of the clusters (every data point is a single cluster)
2. Compute the distance matrix between the clusters
3. Select the pair of clusters with the lowest distance between them
4. Merge the selected pair of clusters
5. Repeat the steps (from step two) until a stop criteria has been reached

A ”stop” criteria could be:

- all data points are merged to a single cluster
- a predefined size of the clusters is reached
- or a predefined number of clusters is reached

There are also various ways to calculate the distance matrix. The term linkage is used for distance calculation. The output of the algorithm is a hierarchical clustering dendrogram. (Szymkowiak et al., 2001)

The HAC algorithm is represented in Figure 3.13, Figure 3.14 and Figure 3.15. The left diagrams represent the clustering of data points whereas the hierarchical clustering dendrogram is shown on the right. The grey points in these figures represent samples or single clusters. Figure 3.13 shows the first two iterations of the HAC algorithm. On the left hand side, two new clusters are detected. The first cluster is computed from cluster A and cluster B whereas the second cluster is computed from D and E. The right hand side of Figure 3.13 shows the dendrogram. The green ellipses in Figure 3.14

represent the merged clusters. Figure 3.15 has the output cluster after the final iteration.

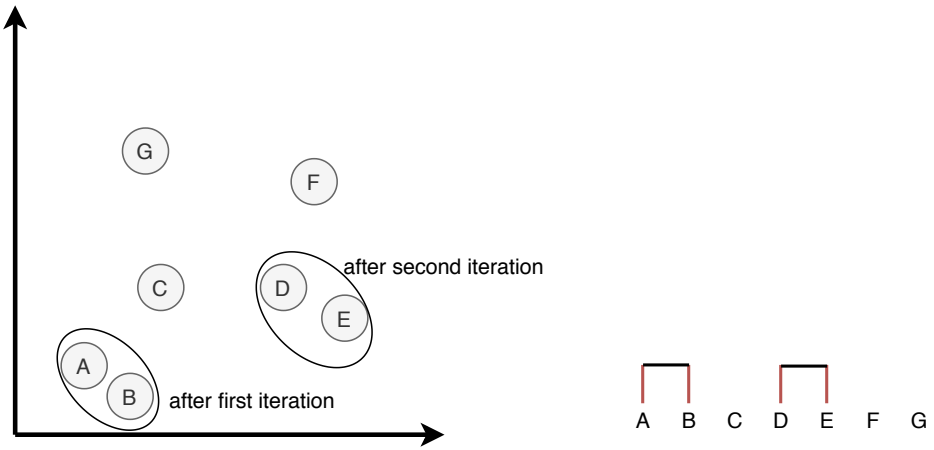


Figure 3.13: Illustration after the second iteration of the HAC algorithm. Cluster A and B, and C and D are connected first.

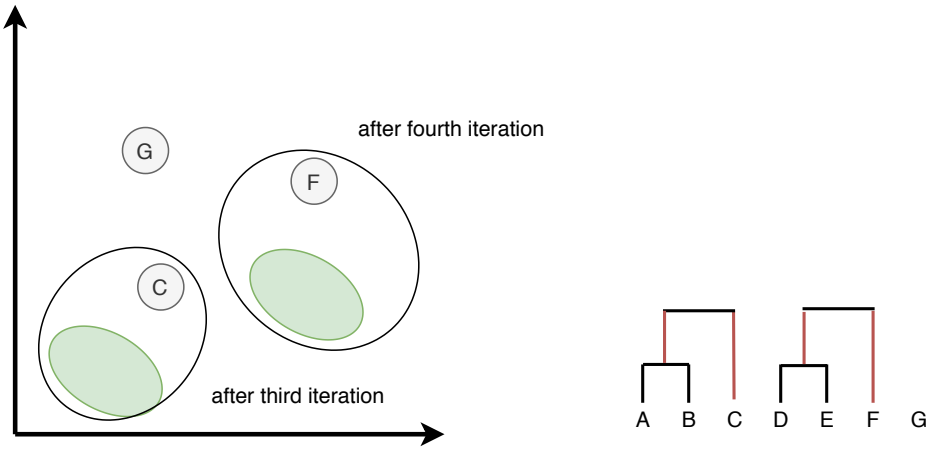


Figure 3.14: Illustration after the fourth iteration of the HAC algorithm.

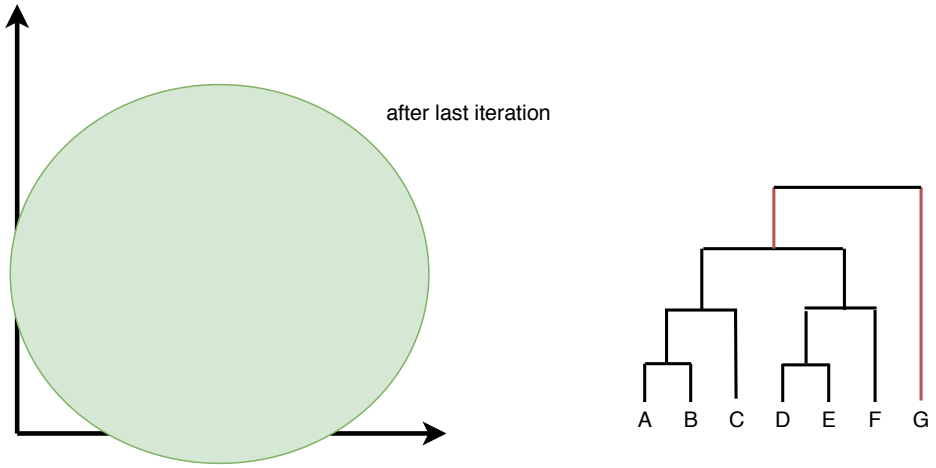


Figure 3.15: Illustration after the final iteration of the HAC algorithm.

3.5.2 Association

Association belongs to a group where an algorithm tries to learn without a data set that is assigned to a target value. Rules are used to discover relationships in a data set. For example, people that buy a computer mostly buy a new screen. For detailed insights in this area the reader is referred to Hastie et al. (2009).

3.6 Reinforcement Learning

Reinforcement learning is different from supervised learning and unsupervised learning, it follows the approach of learning from mistakes. Instead of trying to find patterns as unsupervised learning, reinforcement learning tries to map situations to actions and maximize a numerical reward signal. (Sutton et al., 2018)

Figure 3.16 is based on Sutton et al. (2018) and describes how a reinforcement process works. The problem of learning will go straightforward from interaction to achieving goals. The algorithm learns to solve a task by trial and error, so it learns from mistakes and discover which actions yield the

best reward. These mistakes influence the rewards for the next situations and the next decisions of an actor. An actor is in communication with the environment, which describes the area surrounding. The actor tries to find actions before a model is able to achieve the best output. The actor does not know which action is the best, but instead, it has to discover which actions will bring the most reward. The goal is to maximize the reward.

- Agent: An agent is a learner and decision maker
- Environment: The inputs of an environment are the current action and state. The outputs are the reward and the next state
- State: A state describes the situation an agent has to solve
- Reward: A reward is a numerical value which measures the success or failure of the actors' action in a current state
- Action: Actions are the interactions with the environment and the output from the agent
- Policy: The policy is the strategy, that maps a state to an action
- Value function: Value functions make a prediction how much rewards can be obtained for future steps in a specific scenario

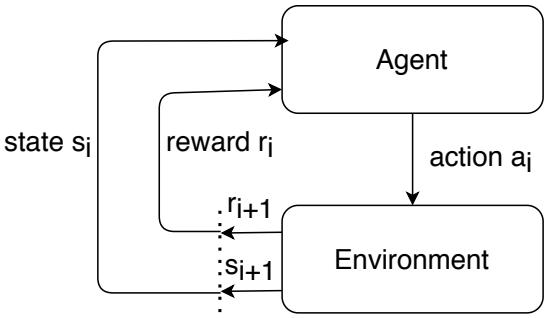


Figure 3.16: Basis algorithm of reinforcement learning. i stands for the iteration steps, s for the current state, r for the reward and a for the action.

4 Prediction of Knowledge Level and Grades Based on Artificial Intelligence

This chapter deals with the practical part of this thesis. First, the prediction scenarios of this thesis are described in detail. For the realization of the predictions based on supervised machine learning models, different approaches have been applied which are explained in detail. Thereafter, a deeper insight in the main components of the predictions are given, including a description of dataset selection and data mapping technologies. The used data set is described as well as how to map the data in the database. After that, the next steps for data pre-processing are described which are based on selection of features and outlier detection. The PCA algorithm has been applied on the data set to extract the most meaningful features. Unwanted data was recognized and removed from the data set by the isolation forest tree forest tree algorithm and the local outlier factor algorithm.

The explanation of the used machine learning types is presented in two sub-sections. Both sub-sections are based on a machine learning process (see Section 3.1). First, the classification part is explained. This section includes the comparison of the most frequently used algorithms. Furthermore, the selection criteria of the finally applied algorithms are explained. After that, regression technologies are stated in detail which have been applied to predict the KL. Four algorithms have been evaluated in detail and based on the reference values the best fitting algorithm for the prediction of the KL has been identified. Reference values are computed to make it easier to find the best fitting algorithm.

4.1 Predicting students' future knowledge level and exam grades

One goal of this thesis is to predict a user's future KL. For instance, when a person acquires knowledge in a certain area and then stops dealing with the acquisition of knowledge in a certain area, the knowledge level decreases after a certain time. A knowledge level prediction could help a person to stay motivated and to not stop dealing with a topic. Which means the knowledge level does not decrease but instead stays the same or even increases. Another goal is the successful prediction of an exam grade. The KL represents the knowledge in a given time period for a given topic. It is between 0 and 100 whereby, 0 is the lowest value and 100 the highest.

The KL prediction can be divided into two parts. On the one hand, the KL for a category of a user is calculated based on the records of other categories. On the other hand, the calculation of the KL of a category is calculated based on the previous activities in this category. The second part deals with the prediction of an exam grade. This is calculated with the help of the users' knowledge as well as other recorded values such as the users' total interactions in a course.

This thesis deals with two different ways of predicting students' future KL. For example, imagine that there is a learning application focusing on English. This application is divided into four different categories: Reading Comprehension, Listening Comprehension, Writing Comprehension and Spoken Production and Interaction. On the one hand, students' future knowledge level of a category mentioned above can be predicted based on past activities in this category on the learning application. On the other hand, students' future knowledge level of a category can be predicted using the past or current activities of other categories. In addition of dealing with two different ways of predicting students' future knowledge level, this thesis deals with the prediction of exam grades. Both, students and professors can benefit from the results of this application. For example, the approach can be activated for students to show their knowledge for the next weeks. This could be a motivation to get more familiar with the current topic and thereby, increase the level of knowledge. Furthermore, a student could get an exact feeling how much work is open to pass an exam. On the other

hand, professors can use the future students' KL to evaluate their teaching methods and teaching speed. They can react early if a student's KL does not correspond to the desired level.

4.2 Used Technologies

This section describes the technologies that have been applied in this thesis. The chapter machine learning (see Section 3) explains the machine learning process, algorithms, and frequently used terms, but not the technologies that are necessary for the implementation.

First, software libraries are explained that are used to create the prediction approach. Therefore, the programming language Python[1] is used for the implementation. Afterwards, an overview of the used database technology MySQL[2] is given, which forms the base of the persistence data storage.

Keras

Keras[3] is a deep learning library based on Python. Kera's simplest model is the sequential model that is a linear pipeline of neuronal network layers (Gulli et al., 2017). This library is a toolkit for the creation of neuronal networks because it provides blocks of functions to create a powerful model. Such a building block can be layers, activation functions, or optimizers.

Keras includes the powerful computation libraries Tensorflow[4] and Theano[5]. The computation of the model can be applied on a Central Processing Unit (CPU) or a Graphics Processing Unit (GPU). Tensorflow is developed and maintained by Google[6].

[1] https://www.python.org
[2] https://www.mysql.com/de
[3] https://keras.io
[4] https://www.tensorflow.org
[5] https://pypi.org/project/Theano
[6] https://www.google.com

Scikit-learn

Scikit-learn[7] is an open-source library developed in Python. Besides, Python can be used in other programming languages as for instance C or C++. It provides serves a lot of supervised and unsupervised learning algorithms. More precisely, it includes classification, regression, clustering, model selection, and pre-processing algorithms. Scikit-learn builds upon NumPy[8], Matplotlib[9], and Pandas[10].

4.2.1 MySQL

MySQL is a relational database management system (RDBMS). This system is an open-source solution as all the other mentioned technologies above. A database is a structured collection of data in an organized system. A relation database structures data into databases, tables, rows, columns, and entries. Furthermore, it creates relationships between tables for a structured set of data. Structured Query Language (SQL) is used to insert, update, delete, extract data, and control users' access from the relational database.

4.3 Data Selection and Preparation

The essential part of machine learning algorithms are the used data sets. Well prepared and informative data are key aspects for a precise prediction. As mentioned in Section 3.1, data selection and preparation is the initial step in a typical machine learning process. The less missing values, noises, incorrect data, or unnecessary data a raw data set contains, the better is the prediction and the faster the data pre-processing step. The data set should not be overloaded with unimportant features, but there should be enough expressive features.

[7] https://scikit-learn.org
[8] https://numpy.org
[9] https://matplotlib.org
[10] https://pypi.org/project/pandas

The following two points describe the data source and the mapping of the data set in a database.

4.3.1 Data Source

Most machine learning data sets are downloaded from the web or collected from the behaviours of users surfing online. In contrast, the data source for this thesis is based on a data set of KnowledgeCheckR[11]. KnowledgeCheckR is a web-based application for a simple and targeted distribution of knowledge. Moreover, a detailed explanation of KnowledgeCheckR is given in Section 5.2.

This application is used at Graz University of Technology. Students use the application during lectures for manifesting knowledge in a playful way. Each interaction of the user is tracked and saved for a better understanding of how users interact with the system and how this affects the knowledge of the user. A learning application is used to model the details of a course. The learning application contains categories and each category consist out of questions that a user should answer.

The data set, for the practical part of the thesis, contains data of the course "Object Oriented Analysis and Design" which deals mainly with aspects of designing and analysing an application or a system.

4.3.2 Database Mapping

A MySQL database is used as the persistent storage of the data. The data set is divided into three tables. Each table contains of primary key (PK) and foreign key (FK). A PK is a unique key to identify an entry. FK are used to create a relationship between tables. The data set is composed of the data which is the output of the feature selection (see Section 4.4.1) and feature extraction (see Section 4.4.2) phase.

[11] https://www.knowledgecheckr.com/

Table 4.1 shows the basic data for each user in a category. It contains the learning application key, the category key, the user key, and the date of the last participant in this category. Furthermore, it involves the correct interactions, total interactions, correctly answered questions, and totally answered questions of a user for a category. The exam prediction table (see Table 4.2) includes more accurate details for the prediction of exam grades. The table has an unique identity (ID) as PK and the learning application ID, user ID and the exam ID as FK. Furthermore, Table 4.2 stores the users' score, the maximum reachable points of an exam and the date of an exam (valid_from). The prediction_knowledge_level_dbo table (see Table 4.3) stores the users' knowledge level for each participated category. Moreover, the prediction_knowledge_level_dbo table has as unique ID as PK and the learning application ID, category ID and user ID as FK. All the data is updated once a week.

prediction_dbo	
PK	ID
FK	prediction_la_id
FK	prediction_ca_id
FK	prediction_registered_user_id
	correct_interactions
	total_interactions
	correct_questions
	total_interactions
	last_participation

Table 4.1: Table prediction_dbo with the associated attributes. A PK is a unique key to identify an entry. FK are used to create a relationship between tables. The attribute *ID* is the unique key of this table. Moreover the foreign keys *prediction_la_id*, *prediction_ca_id* and *prediction_registered_user_id* are used to create a relation between the table prediction_dbo and other tables as for instance the registered user table or the learning application table. The attribute *correct_interactions* describes the correct interaction of a user in a category and the attribute *total_interactions* the total number of interactions in a category. The *correct_questions* explains how many questions a user have answered correctly once in a category. In addition, *total_interactions* is the number of total questions in a category. Last, the attribute *last_participation* is the last participation date of a user in a category.

prediction_exam_dbo	
PK	ID
FK	exam_id
FK	prediction_la_id
FK	prediction_registered_user_id
	maxscore
	score
	valid_from

Table 4.2: Table prediction_exam_dbo with the corresponding attributes of this table. A PK is a unique key to identify an entry. FK are used to create a relationship between tables. The attribute *ID* is the unique key of this table. Attribute *exam_id* describes the unique ID of an exam, Furthermore the *prediction_la_id* and the *prediction_registered_user_id* are FK for link this table with another table. The *maxscore* is the maximal reachable score in an exam and the *score* is the achieved score of a user in an exam. The *valid_from* attribute describes the date of an exam.

prediction_knowledge_level_dbo	
PK	ID
FK	prediciton_la_id
FK	prediction_ca_id
FK	prediction_registered_user_id
	knowledge_level
	score_date

Table 4.3: This table represents the prediction_knowledge_level_dbo with the attributes. A PK is a unique key to identify an entry. FK are used to create a relationship between tables. The attribute *ID* is the unique key of this table and the *prediciton_la_id*, *prediction_ca_id* and *prediction_registered_user_id* are the foreign keys. The attribute *knowledge_level* describes the KL of a user in a category at a certain time (*score_date*).

The data set for the practical part consists of the learning application "Object Oriented Analysis and Design" which is separated into seven categories: *General*, *Object Relation Mapping*, *Unified Modeling Language (UML) Class Diagram*, *Unified Process*, *Use Cases*, *State Charts*, and *Sequence Diagram*. Each of them has more than ten questions. About 250 students participated in this course, this is a very small number of samples, but the next sections of this chapter explain how to deal with such situations.

4.4 Data Pre-processing

This section explains how to modify raw data sets for a precise prediction. An unprepared data set includes missing values, noises, or incorrect data (see Section 3.1.2). This data has to be detected and modified in a way that the data set is not biased. Most of the data in an existing data set is complete except the knowledge level. Nevertheless, this is a key aspect for future knowledge level prediction and in further consequence, for exam grade prediction. The next subsections will explain how to extract the most valuable features from the data set and detect conspicuous data points.

4.4.1 Feature Selection

For a precise prediction, features have to be chosen carefully. First, a list of ranked features must been created. The list starts with the most important feature and shows the most irrelevant features for the current taste at the end. Each prediction approach of the KL and the exam grade prediction (see Section 4.1) has its own feature list.

Classification: Feature Representation

The feature lists for the exam grade prediction are discussed in this section. Table 4.4 shows the first ranked list of features for the classification. The samples are on the learning application level. This means, that each sample is related to a user and the values of the cells are related to the level of a learning application. As seen in Table 4.4, the majority features of the feature list are the interactions and correct answered questions of a user. Furthermore, this feature list includes the number of days since the last participation of the user, the average complexity of correct answered questions of a user for all categories in a learning application, the complexity of the false answered questions, and the user's KL in relation to the community. The value of the KL in relation to the community KL is computed by the difference of the current KL of the user and the average KL of all other

users. The target value for the prediction is the number of points that a user achieved in the exam.

Importance	**Initial feature list**
8	total interactions of a user
7	correct interactions of a user
6	number of possible questions to be answered
5	number of correct answered questions
4	amount of days since the last participation
3	average complexity of a correct answered questions
2	average complexity of a false answered questions
1	users' knowledge level in relation to community KL (current KL - averages KL of the community)
target value:	**exam score**

Table 4.4: Initial feature list for an exam grade prediction.The list is ranked according the importance of the features, starting with the most important one on the top. The feature *total interactions of a user* describes the users' total number of interactions in a learning application. Moreover, the feature *correct interactions of a user* shows the correct number of interactions. *Number of possible questions to be answered* explains the number of possible questions a user is able to answer and feature *number of correct answered question* represents the number of questions a user answered correctly. Moreover, the *amount of days since the last participation* are the number of days that have passed since the last participation in a learning application. The *average complexity of a correct answered questions* and *average complexity of a false answered questions* describes the average complex of questions. Last, the feature *users' knowledge level in relation to community KL* is the relation between a user's KL and the average KL of other learning application participants.

The output after n iteration steps of testing and evaluating features, is a ranked feature list that is applied for the exam grade prediction. The feature list depicts in Table 4.5 shows the feature list for an exam grade prediction. The output after n iterations reflects that the KL is the key feature. The machine learning process is iterative as Figure 3.1 shows. It occurs that the process flow jumps one step back before a corresponding result can be achieved and the next phase is reached. There are not only a lot of iterations in the data pre-processing step but also between the data pre-processing step and the machine learning process. The simplest way to evaluate a good selection of features is to test it with classification algorithms.

Importance	**Regression feature list**
6	KL_1
5	KL_n
4	correct interactions of a user
3	total interactions of a user
2	correct questions of a user
1	total questions of a user
target value:	**grade**

Table 4.5: Feature list after n iterations steps of testing and evaluation for the exam grade prediction. The entries KL_1 and KL_n represent the last stored weeks of knowledge level. The variable n describes the number of weeks. The *correct interactions of a user* describes all correct answered question of a user and the *total interactions of a user* is the interactions' total number of a user. Moreover, the *correct questions of a user* represents the number of correct answered question once and the *total questions of a user* shows the total questions a user have answered. Last, the *target value: grade* describes the exam grade of a user.

Regression: Feature Representation

The knowledge level predictions of learning application participants use a regression approach for the best result. The output of a regression algorithm depends on a list of features.

The initial feature list for users' knowledge level prediction of a category based on the values of other categories is shown in Table 4.6. This list includes the same interactions' and questions' information as the classification above. In contrast to the feature list of the classification (see Table 4.5), the regression is extended with the KL of each category in a learning application. The target value is the KL of the predicting category. The feature list for the prediction of the KL is seen in Table 4.7. This list consists of categories' KL which are in relation to predict the KL of a wanted category. The feature list for the KL prediction after testing and evaluating the features is shown in Table 4.7.

The second regression approach deals with the prediction of a user's future category KL based on the data set of this category (see Section 4.1). The initial feature list of this regression includes the correct and total interactions

Importance	Initial feature list
6	correct interactions of a user
5	total interactions of a user
4	correct questions of a user
3	total questions of a user
2	KL$category_1$
1	KL$category_n$
target value:	**KL**

Table 4.6: Initial feature list for the category prediction based on the KL of other categories. The *correct interactions of a user* describes all correct answered question of a user and the *total interactions of a user* is the interactions' total number of a user. Moreover, the *correct questions of a user* represents the number of correct answered question once and the *total questions of a user* shows the total questions a user have answered. Entries $KLcategory_1$ and $KLcategory_n$ represents the KL of related categories.

Importance	Regression feature list
2	KL$category_1$
1	KL$category_n$
target value:	**KL**

Table 4.7: Initial Feature list of KL prediction based on categories. Entries $KLcategory_1$ and $KLcategory_n$ represents the KL of related categories.

of a user, the correct and total questions of a user and the KL of n weeks as seen in Table 4.8. The feature list which is used for this regression approach is displayed in Table 4.9. This feature list includes the KL of the n weeks.

Ideas of Powerful Features

A lot of different versions of features' lists for all those scenarios have been created and evaluated. For example, one feature list approach contains of a detailed view of each category for each week. This approach has about fifty features. Another approach is a detailed separation of interactions or adding a relationship between interactions and questions. The last feature list approach that is discussed in this section contains a self-conceived

Importance	**Initial feature list**
10	correct interactions of a user ($correct\_interactions_1$)
9	correct interactions of a user ($correct\_interactions_n$)
8	total interactions of a user ($total\_interactions_1$)
7	total interactions of a user ($total\_interactions_n$)
6	correct questions of a user ($correct\_questions_1$)
5	correct questions of a user ($correct\_questions_n$)
4	total questions of a user ($total\_questions_1$)
3	total questions of a user ($total\_questions_n$)
2	KL_1
1	KL_n
target value:	**KL of a category**

Table 4.8: Initial Feature list for predicting the KL of a category. The entries $correct\_interactions_1$ and $correct\_interactions_n$ describes the number of users' correct interactions of a category of the last n week. Moreover $total\_interactions_1$ and $total\_interactions_n$ represents the categories' total interaction of a user of the last n weeks. The entities $correct\_questions_1$ and $correct\_questions_n$ are the number of correct answered questions of a category of the last n weeks. Moreover the entities $total\_questions_1$ and $total\_questions_n$ have the same meaning as $total\_interactions_1$ and $total\_interactions_n$ but refer to questions instead.

Importance	**Initial feature list**
2	KL_1
1	KL_n
target value:	**KL of a category**

Table 4.9: Feature list for the KL of a category prediction. The entities KL_n and KL_n explains the KL of a category for the last n weeks.

variable. This feature derives from the KL and is divided into five possible values. The values are ranked according to the importance. Five is the most important value and one the least important value. The value is composed as follows:

- 5... current KL $>$ 90 percent and the delta KL[12] is never $<$ 0 percent

[12]The delta KL is a value that describes the difference between two successive KL.

over the last 12 weeks.
- 4... current KL > 90 percent and the average delta KL < -10 percent over the last 12 weeks
- 3... current KL ≥ 75 and < 90 percent and the average delta KL < -30 percent over the last 12 weeks
- 2... current KL≥ 50 and < 75 percent and the average delta KL < -40 percent over the last 12 weeks
- 1... everything else that do not match the other criteria

The number of features is not decisive for the accuracy of the result and depends on the quality of the features. Furthermore, the sections above (see Section 4.4.1 and Section 4.4.1) show that the number of features decreased between the initial list and the final list. The key feature for an efficient and exact prediction is the KL. The other features could have more expressiveness if the data set would contain of more samples. It is possible that the initial list leads to a more precise result if the sample size increases. The complexity which is mentioned in Table 4.4 is not helpful because each question has the same value of complexity.

4.4.2 Feature Extraction

Feature extraction gets selected features as input and reduces the number of features in a data set by creating new ones. The principle components analysis algorithm is used to create a new powerful set of features. Next, the PCA is used to give with the help of an example, how the accuracy of the prediction changes. The data set of the example consists of 230 features and 180 samples and was generated out of KnowledgeCheckr. The goal is to apply feature extraction on this example and demonstrate that the accuracy of the prediction is influenced by feature modification and the selection of the classification algorithm.

The algorithms logistic regression, support vector machine, random forest classifier, multi-layer perceptron classifier, and a neuronal network. have been applied with the PCA algorithm. Moreover, the used data set consists of about 250 samples and about 200 features. Table 4.10 shows the change

of the algorithms' accuracy with and without PCA. Summarized, PCA does not influence the accuracy significant on this data set.

	Logistic Regression	**Support Vector Machine**	**Random Forest Classifier**	**MLP Classifier**	**Neuronal Network**
without PCA	0.50	0.53	0.56	0.5	0.68
50 components	0.50	0.5 0	0.42	0.53	0.69
150 components	0.53	0.50	0.53	0.53	0.72

Table 4.10: The accuracy of a classification algorithm applied with and without PCA are shown in this table. The number of components state how many features are generated.

4.4.3 Outlier Detection

A well prepared data set is the key for a good result. Therefore, data which provides wrong predictions has to be detected. The outlier detection is a key instrument for a high accuracy of the result. The following figures use the isolation forest (see Section 3.3.1) and the local outlier factor (see Section 3.3.2) on a data set. The data set was recorded with KnowledgeCheckR and consists of 180 samples and ten features.

Table 4.11 compares the training data accuracy and test data accuracy of a neuronal network with two outlier detection algorithms (isolation forest and local outlier factor). The outlier detection does not significantly affect the results of the neuronal network based on KnowledgeCheckR data set. For instance, the data set is too small for an outlier detection to detect outliers and positively influence the accuracy. Moreover, the data set does not consist of significant outliers. Due do the observation, an outlier detection was not applied in the thesis developed algorithms.

	Without Outlier Detection	**Isolation Forest**	**Local Outlier Factor**
Accuracy on training data	0.62	0.63	0.61
Accuracy on test data	0.51	0.49	0.50

Table 4.11: The accuracy of a classification algorithm applied with and without outlier detection algorithms.

4.5 Machine Learning and Interpretation

First of all, this section describes the chosen machine learning approach, which is supervised machine learning. Based on the supervised machine learning approach, one can predict exam grades and students' future KL. This approach was chosen because the features and target values are already known. For predicting exam grades a classification algorithm is used because the exam grades can be divided in five target values (grades one to five). In addition, a regression approach was chosen to predict students' future KL because it gives one the possibility to predict target values directly.

First, the selection of the classification algorithm for the exam grade prediction is explained. Afterwards, the selection of the regression algorithm for students' future KL is described.

4.5.1 Classification Algorithms

Different classification algorithms have been evaluated within the scope of this thesis. These algorithms are logistic regression, support vector machines, random forest classifier and neuronal networks.

Each algorithm was tested and based on the defined criteria of Section *Interpretation* (3.1.4) interpreted and evaluated. The criteria was ranked (see Table 4.12) and an algorithm was chosen according to this ranking. First of all, two baseline algorithms were developed (random and constant baseline algorithm). The zero rule algorithm classification is a sub type of the constant baseline. This algorithm always predicts the most used target value. The zero rule algorithm has a significantly higher result than the random algorithm, because the data set is unbalanced.

The first classification algorithms were sorted out due to the fact that the evaluation values are lower than the baseline values. Table 4.13 represents the accuracy, precision, recall, and F1 score of each algorithm including the two baseline algorithms. Each algorithm is over the baselines. The neuronal network has the best accuracy and nearly the best recall (see Table 4.13). However, it was a head to head race between neuronal network, support vector machine, and the random forest classification because all

Criteria ranking	
1	Accuracy
2	Recall
3	Precision
4	F1 Score

Table 4.12: Classification evaluation criteria ranking.

three approaches have an accuracy over 50 percentage. A neuronal network (*Keras*) was decided to apply because it runs most precise in all test scenarios. *Keras* was tested with balanced data sets, unbalanced data sets, the most important features and a lot of unnecessary features. However, the output was almost always the same.

	Accuracy	**Recall**	**Precision**	**F1 Score**
Random algorithm	0.16	0.21	0.11	0.14
Zero Rule Algorithm Classification	0.31	0.20	0.10	0.13
Logistic Regression	0.48	0.28	0.24	0.25
Support Vector Machine	0.54	0.39	0.45	0.41
Random Forest Classifier	0.58	0.49	0.71	0.57
MLP Classifier	0.41	0.27	0.26	0.26
Neuronal Network	0.66	0.40	0.33	0.36

Table 4.13: Comparison of classification algorithms applied.

4.5.2 Regression Algorithms

To address the goal of predicting students' future KL a regression is used. This can be divided into the regression of the KL in a specific category and the prediction of the KL of a category with the help of the other categories. A random or constant baseline for the regression is not useful, because the creation of the value has a too high inaccuracy. The predicted values have a range from 0 to 100.

Table 4.14 shows the evaluation comparison of the students' future KL prediction in a category. The data set consists of about 270 samples and

has the key features that are described in Table 4.9. The decision tree, linear regression, and Lasso show all over 70 percent score as depicted in Table 4.14. The score describes the coefficient of determination R^2 of the prediction which is between 0 and 1. The other values have been described in detail in Section *Interpretation* (3.1.4). The support vector regression is the most inaccurate method for the prediction of the KL in a category. These results show that the decision tree regression has been applied for the practical part. The crucial value was the average deviation, which is always close to zero. The average deviation is the average of the difference between the target value and the predicted value. On average, the deviation is about 0.16 percent.

	Linear Regression	**Support Vector Regression**	**Decision Tree Regression**	**Lasso (Linear Model)**
Score	0.72	0.65	0.72	0.71
Average Deviation	0.91	7.52	0.16	0.55
Maximal Deviation	85.62	99.98	89.77	90.33
Mean Absolut Error	10.25	8.05	9.35	9.82
Mean Squared Error	309.48	401.80	271.49	282.70

Table 4.14: Regression algorithms evaluation based on the data set of a category. The score describes the coefficient of determination R^2 of the prediction which is between 0 and 1. The other values have been described in detail in Section *Interpretation.* (3.1.4).

Finally, the evaluation of students' future KL prediction of a category is based on other categories. This evaluation used a data set with about 180 samples and the feature set was mentioned in Table 4.7. The linear regression, support vector regression, decision tree regression and Lasso are used for the prediction of students' future KL. In contrast, the score of two algorithms are differ significantly from the other ones. On the one side, the support vector regression has a very low score and in relation to other algorithms a high mean absolute error. On the other side, the decision tree regression has a very high score of 98 percent. This is a great result if the small data set is taken into consideration. The linear regression and Lasso have similar values and form the mid-table of the ranking. The decision tree regression is the winner of these comparisons and thus was selected.

	Linear Regression	**Support Vector Regression**	**Decision Tree Regression**	**Lasso (Linear Model)**
Score	0.68	0.32	0.98	0.68
Average Deviation	0.34	4.82	0.11	0.34
Maximal Deviation	26.01	42.87	95.24	88.67
Mean Absolut Error	5.91	10.34	11.30	5.92
Mean Squared Error	104.60	238.69	289.83	104.59

Table 4.15: Evaluation of regression algorithms based on the data set of other categories. The score describes the coefficient of determination R^2 of the prediction which is between 0 and 1. The other values have been described in detail in Section *Interpretation* (3.1.4).

5 Evaluation

This chapter discusses the results of the different applied prediction technologies. The evaluation includes a user study where the user shows expertise in the area of "Object Oriented Analysis and Design".

First, the settings of the evaluation are described in detail. Afterwards, a short insight into KnowledgeCheckR, which served as the platform during the evaluation, is given. Finally, the results for the classification prediction and the two regression predictions are discussed in detail.

5.1 Preparation

KnowledgeCheckR was used for the thesis' evaluation. Thereby, the evaluation was divided into two phases which is described in this section.

The first phase consists of the evaluation of the KL prediction. Therefore, participants attended a learning application of KnowledgeCheckR. A learning application in KnowledgeCheckR is similar to a course at university. The used learning application consists of four categories: *General*, *Object Relational Mapping*, *UML Class Diagrams* and *Unified Process*. Each category has ten multiple-choice questions. The evaluation phase lasted for four weeks. In the first evaluation week, there were four questions for each category. After week one there were more questions unlocked. At the end of the evaluation phase, ten questions per category were available and should have been answered.

After answering all the questions from each category, the second phase started. This phase predicted the participants' exam grades. The exam was available after the first phase was finished. The exam included twenty

questions from all categories already done but also new ones. Moreover, two thirds of the questions were questions from the first phase and the other questions were new questions which a user never answered before.

16 people participated during these five weeks of evaluation. The participants interacted around 1.000 times with the learning application. To get the developed techniques into KnowledgeCheckR, REST API's had to be served. Additionally, some extra tables to the data base schema had to be developed. Those were needed to link the learning applications together and get more records for the data set.

5.2 KnowledgeCheckR

KnowledgeCheckR is an artificial intelligence based system that supports simple and targeted distribution of knowledge. The advantages of KnowledgeCheckR are to save time in knowledge transfer processes as well as to increase the knowledge level in a simple manner. Automated distribution of tests serves to support compliance issues. Moreover, the community knowledge level and the user knowledge level are graphically visualized. Visualizing the community level can encourage users to extend their knowledge level, because they constantly compare their own knowledge level to the community. KnowledgeCheckR can be represented by means of external content, questions and explanations. Topics such as fire protection or first aid are widely used in company settings. In these cases, KnowledgeCheckR provides a simple solution for the distribution of knowledge at employee level.

5.3 Results

The results of students' future KL and exam grades are explained in detail in this section. First, the results of the regression predictions are shown in detail. After that, the prediction of the exam grades are discussed.

5.3.1 Regression Result for the Prediction of Students' Future KL based on Previous Category Records

The prediction of students' future KL based on the previous interactions in a category, as already mentioned in Section 4.1, is discussed in this section. The predicted values represents a student's KL in a category in which the student already participated. This evaluation explains the predicted results of the categories: *General*, *Object Relational Mapping*, *Unified Modeling Language (UML) Class Diagrams* and *Unified Process* in detail. Each category has two diagrams for the graphical visualization of the evaluation. The first diagram is a histogram which represents the frequency distribution of the participants' KL. Therefore, the x-axis (KL) is divided into ten classes which represents the deviation of the knowledge level in ranges:

- 10: 0% to 10%
- 20: 11% to 20%
- 30: 21% to 30%
- 40: 31% to 40%
- 50: 41% to 50 %
- 60: 51% to 60%
- 70: 61% to 70%
- 80: 71% to 80%
- 90: 81% to 90%
- 100: 91% to 100%

The y-axis shows the number of users in each KL class. The point diagrams represent the correlation between the predicted KL and the target KL of each participant. The orange points display the target value and the blue points the predicted value. The point diagrams show one point for a user in case the predicted value and target value are the same. As already discussed, the evaluation for the regression took four weeks. The KL of the first three weeks served as a basis for the forecasts. The output of the fourth week serves as the target KL for the evaluation of the predictions.

Figure 5.1 and Figure 5.2 show the category *General*. The prediction of the future KL in this category is based on the small data set. Figure 5.1 displays, that within nine cases the prediction has a deviation smaller than 20 percent. Additionally, it can be detected that three predicted values were very close

to the target values (see Figure 5.2). This category leads to the most accurate KL prediction when applying a regression technique.

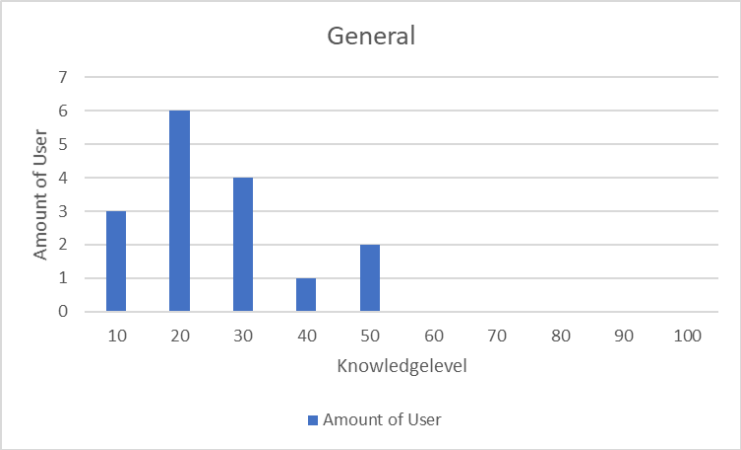


Figure 5.1: The histogram shows the KL deviations' percentage division of category *General*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

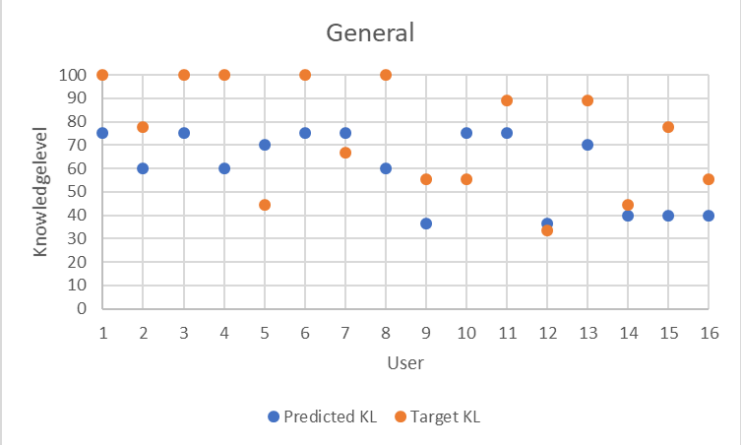


Figure 5.2: This point diagram represents the users' KL deviation of category *General*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

In contrast to category *General*, the category *Object Relation Mapping* has a balanced distribution of the classes as seen in Figure 5.3. The histogram

class *80* counts two users which is an indication of imprecise predictions. Figure 5.4 displays how far apart the values actually are. This category has the most imprecise KL prediction.

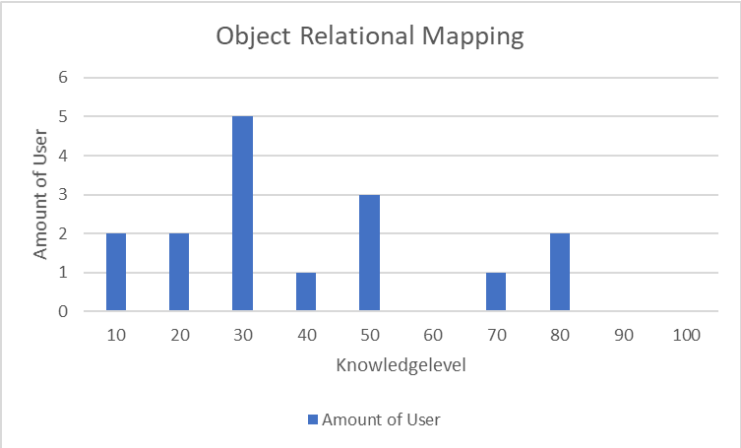


Figure 5.3: The histogram shows the KL deviations' percentage division of category *Object Relation Mapping*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

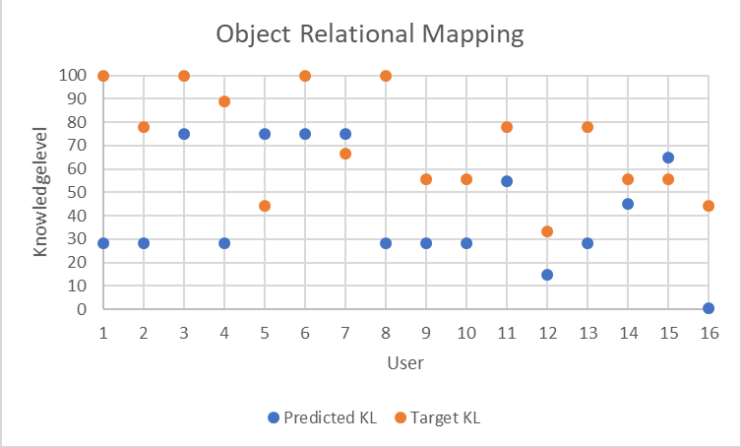


Figure 5.4: This point diagram represents the users' KL deviation of category *Object Relation Mapping*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

The categories *UML Class Diagram* and *Unified Process* have a similar deviation average. Moreover, the category *UML Class Diagram* has a maximum deviation up to 60 percent but has its peak of users at 50 percent (see Figure 5.5). On the opposite, the category *Unified Process* has a maximum deviation of 60 percent as seen in Figure 5.7. The remaining part of the prediction is well divided between the first three classes (*10*, *20* and *30*). Figure 5.6 and Figure 5.8 display a point diagram which represents the target KL and the predicted KL for the categories *UML Class Diagram* and *Unified Process*. Additionally, the category *Unified Process* shows that many values are close to the target value (see Figure 5.8). Moreover, three outliers have been detected as well.

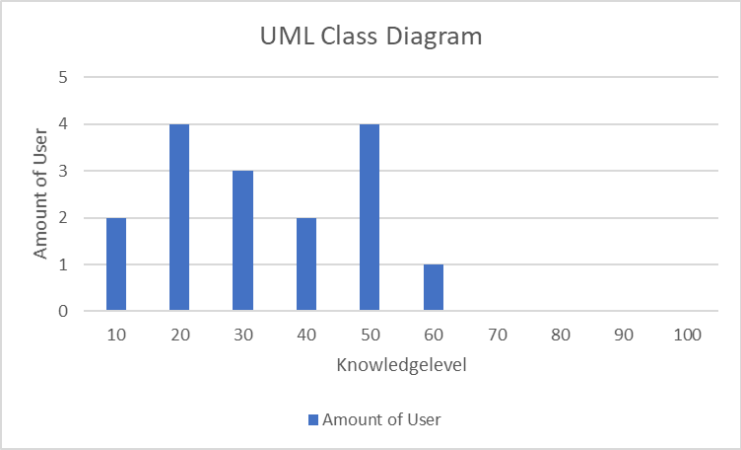


Figure 5.5: The histogram shows the KL deviations' percentage division of category *UML Class Diagram*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

Summarized, the prediction of the KL of all four categories is precise, considering the small data set used for this prediction. A histogram of all categories is shown in Figure 5.9. It is significant that most of the predictions are in the classes between 10 and 30. This implies that 44 out of 64 predictions had a deviation of 0 to 39 percent. It should also be mentioned, that in the category *Object Relational Mapping* two predictions were completely wrong (showing 80 % deviation). As mentioned above, the category *Object Relational Mapping*

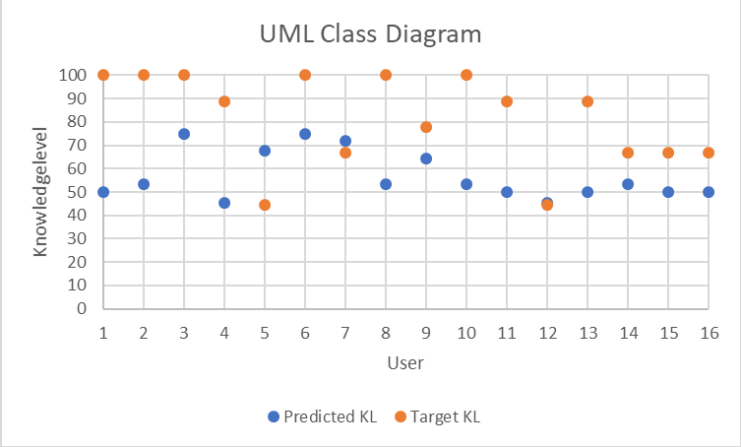


Figure 5.6: This point diagram represents the users' KL deviation of category *UML Class Diagram*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

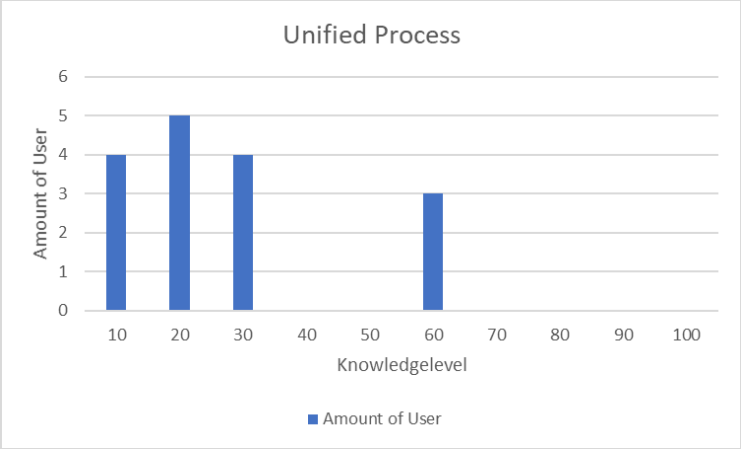


Figure 5.7: The histogram shows the KL deviations' percentage division of category *Unified Process*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

has the worst prediction rate compared to the other ones because the small data set. This can be linked to the unevenness of categories.

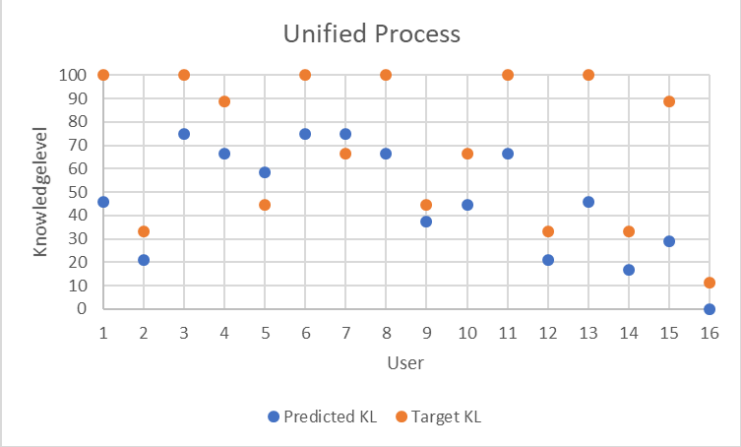


Figure 5.8: This point diagram represents the users' KL deviation of category *Unified Process*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

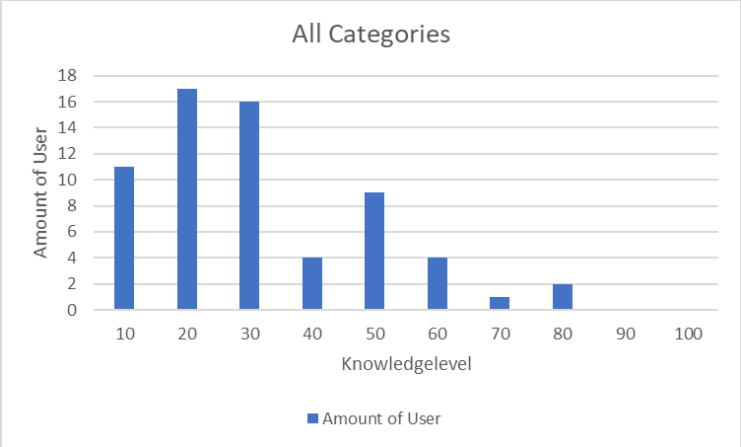


Figure 5.9: The histogram shows the KL deviations' percentage division over all categories. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

5.3.2 Regression Result for the Prediction of Students' KL based on the Past or Current Activities of Categories

This regression includes the prediction of the KL for a category based on the past or current activities of categories in a learning application (see Section 4.1). For example, a user participated in the categories *General*, *Object Relational Mapping* and *Unified Modeling Language (UML) Class Diagrams* wanted to know the KL of category *Unified Process*. The categories of this regression are the same as for the other regression. The most precise KL prediction based on the data of the other categories is the category *UML Class Diagram*. Figure 5.10 shows that half of the predicted values have a deviation lower than 10 percent to the target. Most of the other predicted values are in the histogram class *20*. Furthermore, Figure 5.11 displays a detailed overview of the predicted and target values of each user.

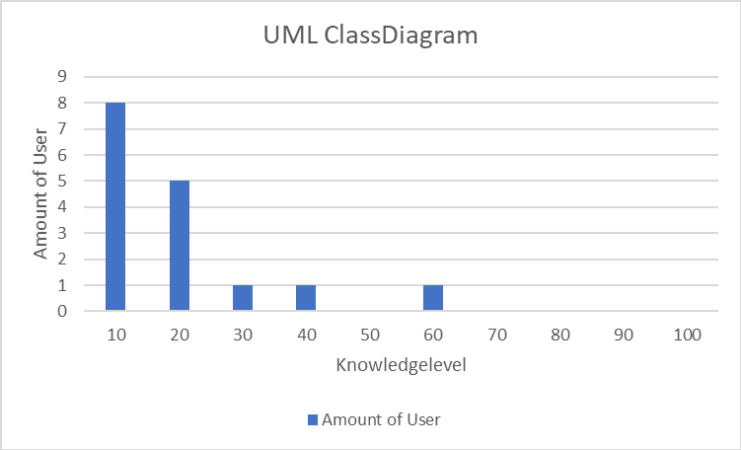


Figure 5.10: The histogram shows the KL deviations' percentage division of category *UML Class Diagram*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

On the one side, Figure 5.12 shows, that most of the category *General* predictions are in the first two knowledge level classes (*10* and *20*). On the other side, four predicted values have a derivation of over 50 percent. Most of the predicted values in category *General* are close to the target values (see Figure 5.13). The prediction of the *Unified Process* KL is similar to the

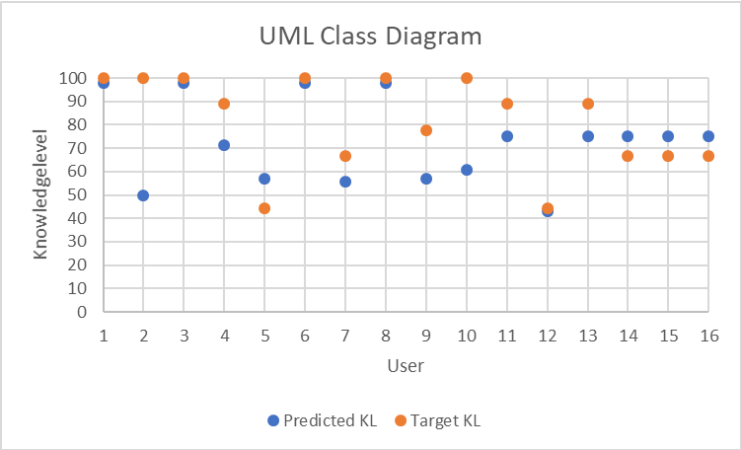


Figure 5.11: This point diagram represents the users' KL deviation of category *UML Class Diagram*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

General knowledge level prediction as presented in Figure 5.14. There are six predicted KL in category *General* which show a deviation smaller than ten percent. Figure 5.15 displays, that the predicted KL are between 40 and 80. Therefore, the deviation to the target value is larger than the deviation of the category *General*.

Figure 5.16 and Figure 5.17 represent the prediction of the category *Object Relational Mapping*. The distribution of the KL deviation is well balanced over the first eight histogram classes as presented in Figure 5.16. Looking more closely at the point diagram (see Figure 5.17), it can be recognized that most of the predicted values are close to 100. The KL prediction of *Object Relational Mapping* has the most inaccurate predicted values compared to the other predictions in this section.

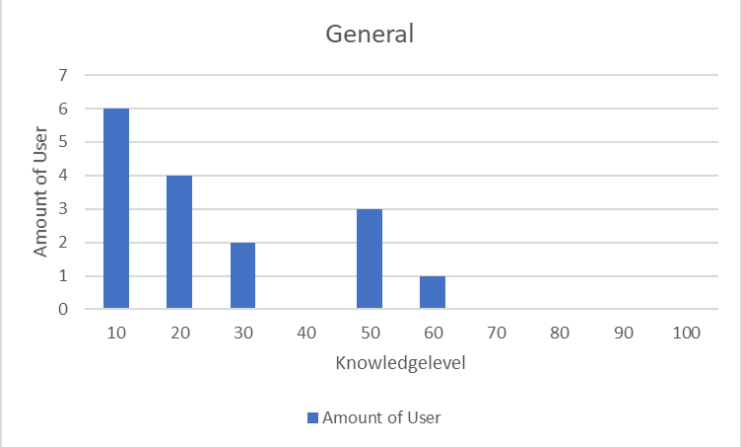


Figure 5.12: The histogram shows the KL deviations' percentage division of category *General*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

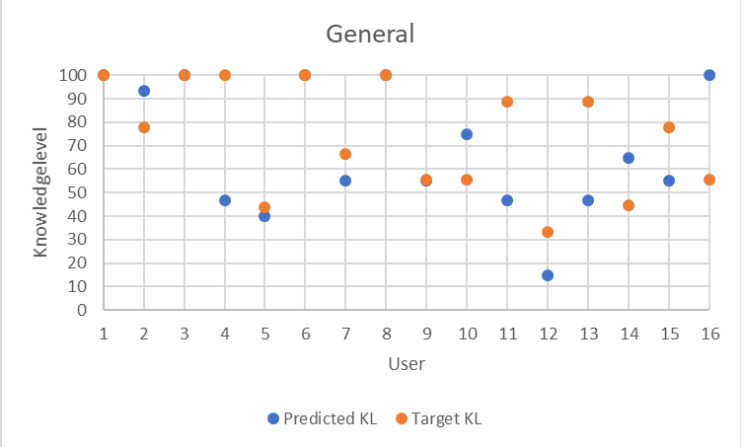


Figure 5.13: This point diagram represents the users' KL deviation of category *General*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

Summed up, the accuracy of the KL prediction depends on the category and the associated data set. The *Object Relational Mapping* data set consists of a smaller data set than the other three categories. This also reflects the accuracy of the prediction. Figure 5.18 shows the distribution of the KL deviation over all four categories. Frankly, this kind of prediction is very

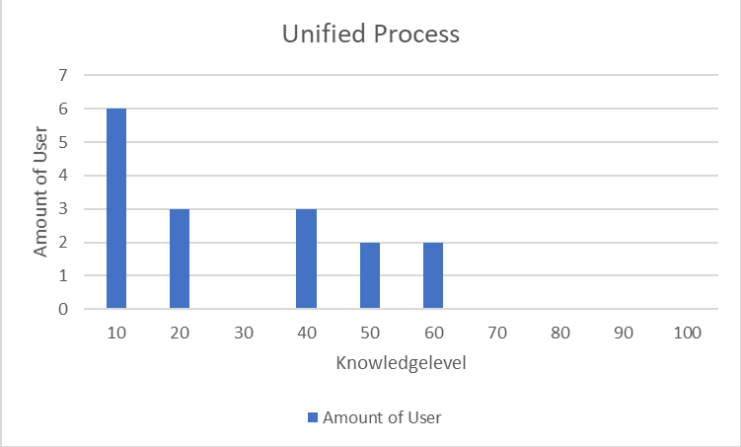


Figure 5.14: The histogram shows the KL deviations' percentage division of category *Unified Process*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

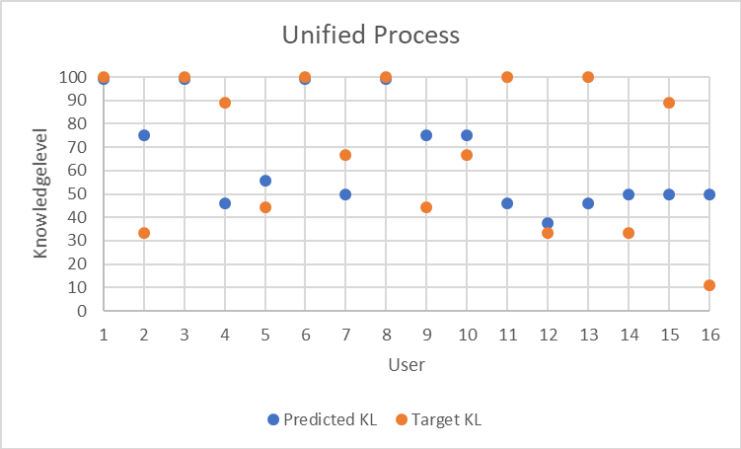


Figure 5.15: This point diagram represents the users' KL deviation of category *Unified Process*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

accurate as the histogram 5.18 shows. Most of the KL predictions have a deviation smaller than ten percent. The bars of the histogram correspond to a decreasing exponential function.

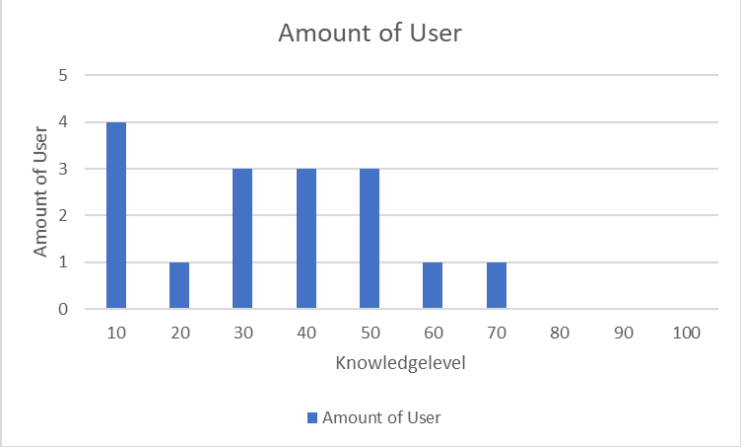


Figure 5.16: The histogram shows the KL deviations' percentage division of category *Object Relational Mapping*. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

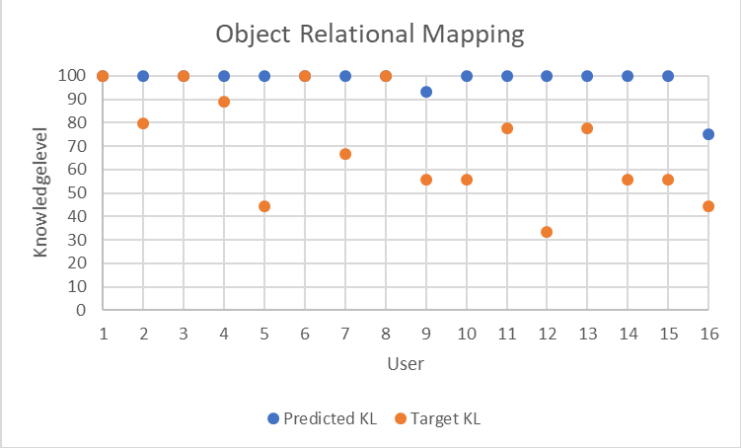


Figure 5.17: This point diagram represents the users' KL deviation of category *Object Relational Mapping*. The x-axis represents the users and the y-axis represents the predicted KL (blue) as well as the target KL (orange).

5.3.3 Classification Result

The second focus of the evaluation was the exam grade prediction. Therefore, a classification algorithm was chosen to predict the exam grade after col-

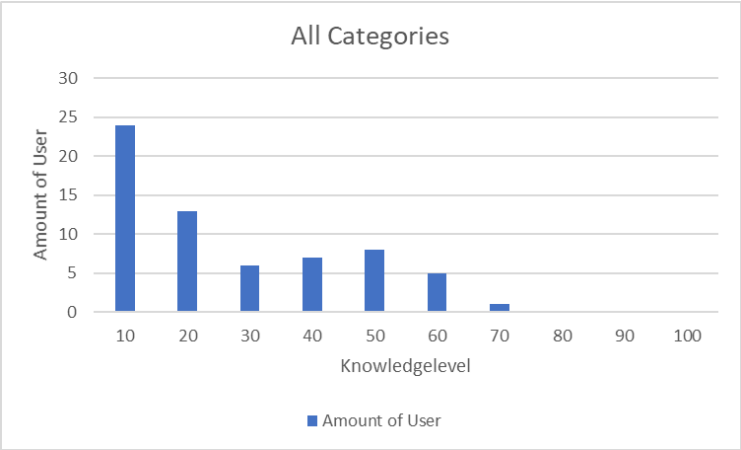


Figure 5.18: The histogram shows the KL deviations' percentage division over all categories. The x-axis represents ten classes of KL. The classes were separated in an 10% interval. The y-axis includes the number of users in a category.

lecting participants' data over four weeks. Only thirteen users participated in this exam, compared to evaluation phase one where sixteen participants participated. The maximum time for the exam was 30 minutes where the participants had to answer 20 multiple-choice questions. The diagrams for the visualization are similar to the diagrams of the regression section. The histogram includes the deviation of the grades instead of the KL. The histogram has five classes on the x-axis and each of the classes represents the difference between the predicted grade and the target grade. For example, class *0* represents the correct predicted grade and class 2 represents the difference between the predicted value and the target value.

Figure 5.19 shows that two predicted grades matched the target grades and most of the values are between the histogram classes *1* and *3*. This reflects the accuracy mentioned in section *Classification Algorithms* (4.5.1). The point diagram 5.20 displays the exact deviation between the target grade and the predicted grade.

Summarized, two exam grades are predicted precisely. Moreover, most students' exam grades are predicted close to the actual exam grad. All in all, the accuracy of exam grades is as expected (see Section 4.5.1).

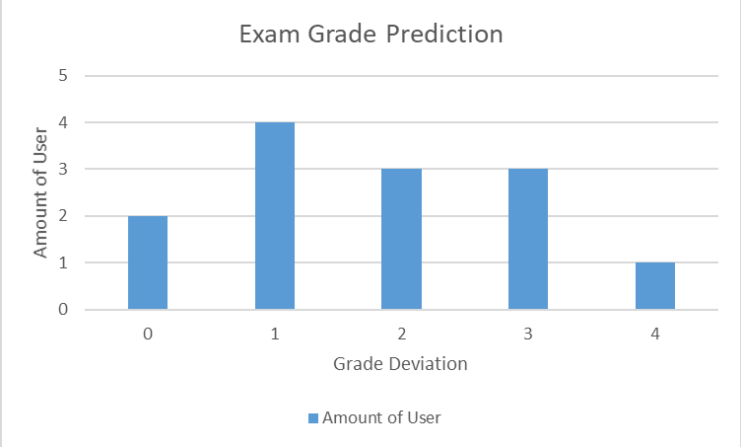


Figure 5.19: The histogram shows the percentage division of the exam grade deviation. The x-axis represents five classes of the grade. Class 0 represents the correct prediction of the grade, class 1 the deviation of one exam grade. The other classes are equally divided. The y-axis includes the number of users in a class.

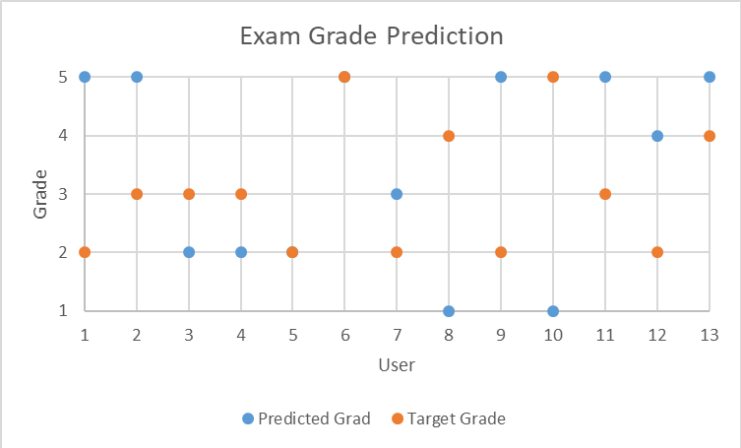


Figure 5.20: This diagram represents the exam grade deviation of each user. The x-axis represents each user and the y-axis represents the predicted grade (blue) and the target grade (orange).

6 Deployment

In this chapter the deployment of the developed approach is going to be explained. As mentioned in section *Used Technologies* (4.2) the developed machine learning algorithms are based on Python and can be called through well-defined REST API. First, the container software Docker[1], which serves as a base for the deployment, is described in detail. Afterwards, the structure of the machine learning approach as well as the communication concepts are described.

6.1 Docker

Docker is a software that makes it possible to virtualize applications in containers. All dependencies of an application can be mapped in a docker image without having a full operating system in the background. A big advantage is the easy and fast sharing of applications with docker images. It is not necessary to install all components of an application manually, because Docker takes care of it. The following list explains key terms that are often used in combination with Docker:

- image: is an immutable file that contains all components for an application
- container: is an instance of the image. Starting multiple instances with the same or different parameters from the same image is possible
- dockerfile: is a text file that contains the structure and settings of docker images. For example, the base image, install dependencies, or commands that should execute on container start. Due to this

[1] https://www.docker.com

dockerfile, Docker is able to build images based on the commands of the dockerfile

- docker-compose: is a yaml[2] file that defines and runs multiple containers. A yaml file is a settings file used in applications where data are being stored or transmitted. It is a powerful tool if an application needs more than one container and containers have to communicate within a network. A docker-compose file consists not only of individual containers that an application provides but also all relevant settings for containers

6.2 Architecture

The architecture of this thesis' practical part is built on Docker. Moreover, the architecture is divided in three main components (MySQL Database, KnowledgeCheckR Server, REST API Server[3]) as shown in Figure 6.1. Docker is easy to use and very powerful for deployment as mentioned above. The first component is the REST API server (seen in Figure 6.1). This component is the communication interface for the KL and the exam grade prediction. The second component is the web application, KnowledgeCheckR. KnowledgeCheckR is a beneficial application which helps users to get knowledge about different topics in an efficient way. In addition, the third component is called MySQL database. This component stores all relevant information for the KL and exam grade prediction.

The process for a KL or an exam grade request is depicted in Figure 6.1. Therefore, KnowledgeCheckR sends a request for a user to the REST API application. Following that, the REST API server sends a request to the MySQL database about the stored user which is essential for the prediction. The MySQL database sends the desired data back to the predictions' application server. The developed algorithm of this thesis predicts the KL or exam grade and sends the prediction to KnowledgeCheckR.

[2]https://yaml.org

[3]The REST API Server is the communication interface for KnowledgeCheckR to obtain users' KL and exam grades. Furthermore, the developed approach is deployed on this REST API Server

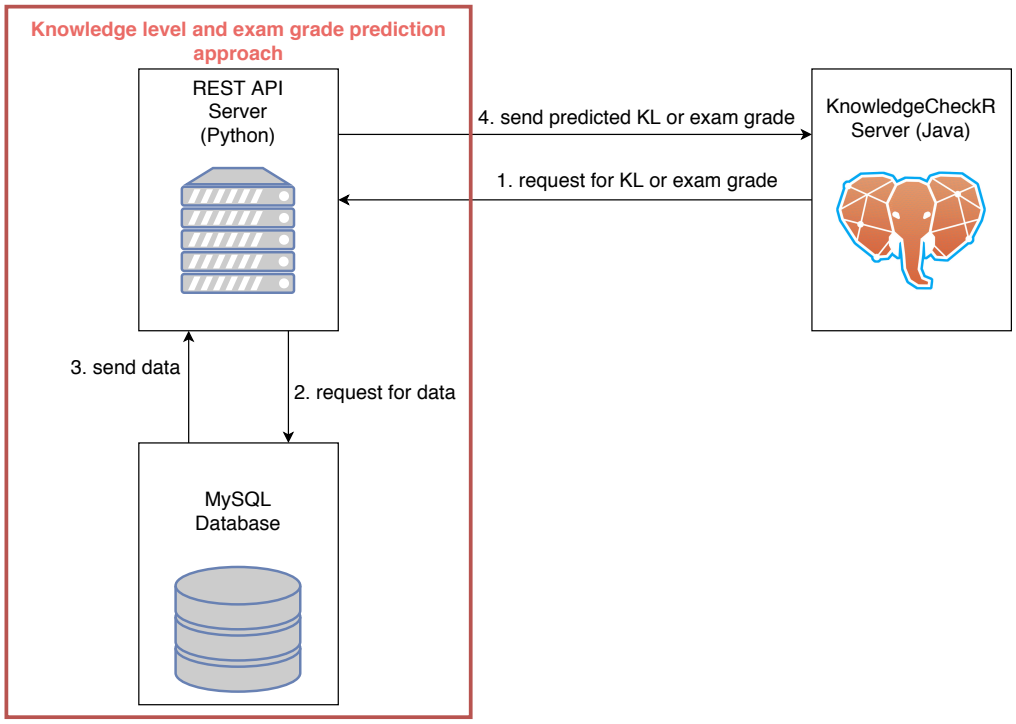


Figure 6.1: Example communication between REST API server, MySQL database, and KnowledgeCheckR server.

Figure 6.2 shows, the response of the KL prediction and the exam grade prediction. The KL prediction response includes the predicted KL (KL prediction), the mean absolute error which determines the accuracy of the regression (Mean_absolute_error), the mean value deviation of the regression (Reg AVG), and the status code which represents the correctness of the response. The exam prediction response consists of the exam grade, the accuracy of the test and training data of the classification model. Additionally, the status code is returned and has the same meaning as the status code already described at the KL prediction response above.

```
{
  "KL_Prediction":100.0,
  "Mean_Absolute_Error":"9.518",
  "Score":"0.741",
  "Status":100
}
```


(a) knowledge level (KL) prediction response.


```
{
  "Exam_Grade":"3",
  "Status":100,
  "Accuracy_Test":"0.55",
  "Accuracy_Training":"0.71"
}
```


(b) Exam prediction response.

Figure 6.2: Example response from the REST API server of a KL prediction and an exam grade prediction.

7 Limitations and Future Work

In this thesis the foundation for a successful knowledge prediction was established. As a topic for future work it is necessary to reduce the critical points, such as to increase the amount of data as well as the associated selection of features. This should serve as a basis for further developments.

Chapter *Prediction of Knowledge Level and Grades Based on Artificial Intelligence* (4) and Chapter *Evaluation* (5) mention that the size of records play a crucial role for the prediction of students' KL and exam grades. Typically, data sets are using more than 100.000 samples for an initial prediction. The data set used in this thesis had about 200 samples. The results of this thesis research conclude that the result is satisfying for such a small amount of data. An important point for future work is the collection of more data. This will drastically improve the results and enable even more accurate forecasting.

Based on the collection of data, the features will also change in the future because the data set will grow. Therefore, other features show the potential of a higher impact on the prediction. Other features may get the key for accurate prediction, as data gets more meaningful. Features can get a different meaning and feature extraction can create new correlations for a better prediction.

In connection with limitations, the terms *missing values* and the *cold start problem* should be mentioned. *Missing values* occur in an incomplete sample and represent empty cells in a data set. *Missing values* have not occurred in the data set until now, but for the future the question arises and thus it should be dealt with. The *cold start problem* is always an important topic in machine learning and recommender system approaches. Furthermore, it deals with the problem of new users because these users have no records for predicting a knowledge level or an exam grade. What should be done if there are not enough appropriate users to detect a similar user. Now, the

developed approach only works leads to an output of the prediction if a user participates for at least one week. The current data set will get bigger and bigger, because several learning applications are still in use. For example, a user B with similar characteristics as user A has already participated in the application $Application2$ and is also a participant in the current application $Application1$. Therefore, it is possible to predict the knowledge level of user A for the application $Application2$ based on previous user's performance of B in the application $Application1$.

As mentioned above, the number of data sets can have a huge impact on a prediction. Not only new patterns can be recognized but also features get new meanings. The key feature KL is unlikely to change, but new and meaningful features can be added. The exam prediction can be made even more accurate in the future by trying to determine the exam grade, based on the first five exam questions. These questions are derived from users' previous activities including previous exam questions. Supervised learning will probably reach its limits and new methods have to be applied. A promising approach to determine which question should be presented next for a user is the unsupervised machine learning method clustering. To determine the similarity of users, methods of recommender systems could also be used.

Based on the prediction of the KL, users could be addressed more individually. The goal is that a user continues their education in a playful way and thus, increases their level of knowledge. As Figure 7.1 shows, each user should learn in his or her own flow. The y-axis t represents the time since the user has started. In addition, the x-axis r show the achieved rewards in relation to the complexity of a question. Assuming a user always gets questions that are too easy or too difficult, the user would be constantly over or under-challenged. The result is that the user is not willing to deal with the application in a recalling fashion. However, if the user is in flow, the knowledge level and the difficulty of the questions can be increased.

In future settings, the approach can not only be used for predicting grades and students' future KL, but could also be of interest for companies. For example, the knowledge level of employees can be predicted even though the user is not requested to answer a single question in a specific topic such

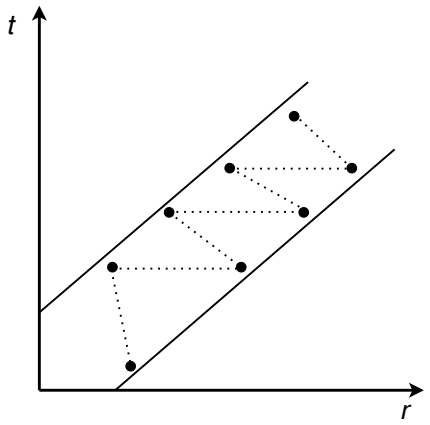


Figure 7.1: A learning flow of a user. The two lines should show a perfect flow and the points describe the rewards for a question in relation to the complexity of a question.

as first aid or fire protection. The results of the system can be used to detect problem areas in early stages.

8 Conclusion

This thesis presented the prediction of exam grades and students' future KL. Therefore, literature in this area was examined, which gives an overview of already existing approaches. These approaches aim to predict the score in a given topic or course. The results should help to derive strategies for students and professors. Students are able to see in which courses they need to invest more time. Besides, professors are able to detect in early stages, whether they are explaining the content of a course at an appropriate speed and in an understandable way.

In addition, an introduction to a machine learning was given. Thereby, a general machine learning process was examined. This iterative process includes data selection, data pre-processing, machine learning phase and interpretation of results. The data selection and pre-processing steps are very important for an accurate prediction. These steps involve feature selection, feature extraction and outlier detection. Main algorithms for feature extraction and outlier detection were explained in detail. Machine learning can be divided into three main approaches: supervised learning, unsupervised learning and reinforcement learning. Supervised machine learning consists of a data set with labelled data. Therefore, every sample consists of a target value which can be interpreted as the following: Each function $f(x)$ knows the output y. This approach includes classification and regression methods. A classification has a vector or samples as input and predicts the corresponding target value. Same as a classification, the regression has a vector or sample as input and predicts a numerical value instead of categories as output. Unsupervised learning has no labelled data and tries to detect patterns in a data set. A frequently used approach for unsupervised learning is clustering. Finally, reinforcement learning follows the approach of learning from mistakes.

The development of algorithms that predict both, the knowledge level and exam grades of users represent the main aim of this thesis. That is the reason why supervised machine learning methods were used. The prediction of knowledge level is based on regression approaches and the prediction of exam grades is based on neuronal networks. Due to the fact that the correct selection of features and the qualitative amount of data are the basis for a successful prediction, most iterations have been needed for these tasks The correlation between qualitative and quantitative data is very important. On the one hand, an enormous amount of data with "no information" is useless for a prediction. On the other hand, a small amount of qualitative data will not deliver the intended result.

The achieved results of the decision tree regression and a neuronal network were compared with the results of different approaches like logistic regression, support vector machine, linear regression or support vector regression. The main point of this comparison is to show that there are far too few samples in the data set for a more precise prediction. Moreover, an evaluation for the regression and classification has been done to evaluate the output of the developed algorithms and to create a basis for future work.

Appendix

Bibliography

Ashenafi, Michael Mogessie, Giuseppe Riccardi, and Marco Ronchetti (2015). "Predicting students' final exam scores from their course activities". In: *2015 IEEE Frontiers in Education Conference (FIE)*. IEEE, pp. 1–9 (cit. on p. 3).

Bishop, Christopher M (2006). *Pattern recognition and machine learning*. springer (cit. on pp. 17, 22–24, 32).

Blum, Avrim L and Pat Langley (1997). "Selection of relevant features and examples in machine learning". In: *Artificial intelligence* 97.1-2, pp. 245–271 (cit. on p. 16).

Boser, BE, IM Guyon, and VN Vapnik (1992). "5th Annual ACM Workshop on COLT". In: *Pittsburgh, PA*, pp. 144–152 (cit. on p. 22).

Breiman, Leo (2001). "Random forests". In: *Machine learning* 45.1, pp. 5–32 (cit. on pp. 24, 25).

Breunig, Markus M et al. (2000). "LOF: identifying density-based local outliers". In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pp. 93–104 (cit. on p. 19).

Dreiseitl, Stephan and Lucila Ohno-Machado (2002). "Logistic regression and artificial neural network classification models: a methodology review". In: *Journal of biomedical informatics* 35.5-6, pp. 352–359 (cit. on p. 24).

Fire, Michael et al. (2012). "Predicting student exam's scores by analyzing social network data". In: *International Conference on Active Media Technology*. Springer, pp. 584–595 (cit. on pp. 3, 4).

Ge, Z. et al. (2017). "Data Mining and Analytics in the Process Industry: The Role of Machine Learning". In: *IEEE Access* 5, pp. 20590–20616 (cit. on pp. 7, 9).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. MIT Press. URL: `http://www.deeplearningbook.org` (cit. on pp. 6, 22, 29).

Grira, Nizar, Michel Crucianu, and Nozha Boujemaa (2004). "Unsupervised and semi-supervised clustering: a brief survey". In: *A review of machine learning techniques for processing multimedia content* 1, pp. 9–16 (cit. on p. 31).

Gulli, Antonio and Sujit Pal (2017). *Deep learning with Keras*. Packt Publishing Ltd (cit. on p. 39).

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media (cit. on p. 35).

Hinton, Geoffrey E (2012). "A practical guide to training restricted Boltzmann machines". In: *Neural networks: Tricks of the trade*. Springer, pp. 599–619 (cit. on p. 4).

Imtiaz, SA and SL Shah (2008). "Treatment of missing values in process data analysis". In: *The Canadian Journal of Chemical Engineering* 86.5, pp. 838–858 (cit. on p. 9).

Iqbal, Zafar et al. (2019). "Early student grade prediction: an empirical study". In: *2019 2nd International Conference on Advancements in Computational Sciences (ICACS)*. IEEE, pp. 1–7 (cit. on p. 4).

Keller, James M, Michael R Gray, and James A Givens (1985). "A fuzzy k-nearest neighbor algorithm". In: *IEEE transactions on systems, man, and cybernetics* 4, pp. 580–585 (cit. on p. 30).

Khoshgoftaar, Taghi M, Moiz Golawala, and Jason Van Hulse (2007). "An empirical study of learning from imbalanced data using random forest". In: *19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007)*. Vol. 2. IEEE, pp. 310–317 (cit. on pp. 24, 25).

Khosla, Prannay et al. (2020). "Supervised contrastive learning". In: *arXiv preprint arXiv:2004.11362* (cit. on p. 4).

Liaw, Andy, Matthew Wiener, et al. (2002). "Classification and regression by randomForest". In: *R news* 2.3, pp. 18–22 (cit. on p. 24).

Liu, F. T., K. M. Ting, and Z. Zhou (Dec. 2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422 (cit. on pp. 17, 18).

Madhulatha, T Soni (2012). "An overview on clustering methods". In: *arXiv preprint arXiv:1205.1117* (cit. on p. 31).

Mahmood, Hamza (2019). *Introduction Gradient Descent*. URL: `https://towardsdatascience.com/gradient-descent-3a7db7520711#:~:text=`

Gradient%20Descent%20is%20a%20process,of%20function%20J(w). (cit. on p. 28).

Medium.com (2020). *Isolation Forest algorithm for anomaly detection*. URL: https://medium.com/@often_weird/isolation-forest-algorithm-for-anomaly-detection-f88af2d5518d (cit. on p. 18).

Meier, Yannick et al. (2015). "Predicting grades". In: *IEEE Transactions on Signal Processing* 64.4, pp. 959–972 (cit. on p. 3).

Mohsen, Heba et al. (2018). "Classification using deep learning neural networks for brain tumors". In: *Future Computing and Informatics Journal* 3.1, pp. 68–71 (cit. on p. 27).

Nuruzzaman, Mohammad and Omar Khadeer Hussain (2018). "A survey on chatbot implementation in customer service industry through deep neural networks". In: *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. IEEE, pp. 54–61 (cit. on p. 1).

Pant, Ayush (2019). *Logistic Regression*. URL: https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148 (cit. on p. 24).

Powers, David Martin (2011). *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation.(2011)* (cit. on p. 11).

Saito, Takaya and Marc Rehmsmeier (2015). "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets". In: *PloS one* (cit. on p. 11).

Sutton, Richard S and Andrew G Barto (2018). *Reinforcement learning: An introduction*. MIT press (cit. on p. 35).

Szymkowiak, Anna, Jan Larsen, and Lars Kai Hansen (2001). "Hierarchical clustering for datamining". In: *Proceedings of KES-2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, pp. 261–265 (cit. on p. 33).

Verma, Shiva (2019). *Understanding different Loss Functions for Neural Networks*. URL: https://towardsdatascience.com/understanding-different-loss-functions-for-neural-networks-dd1ed0274718 (cit. on p. 28).

Zhu, Long Leo et al. (2008). "Unsupervised structure learning: Hierarchical recursive composition, suspicious coincidence and competitive exclusion". In: *European Conference on Computer Vision*. Springer, pp. 759–773 (cit. on p. 33).