FH Salzburg
MultiMediaTechnology

Predicting play types in American football using machine learning

Bachelor Thesis 2

Author: Niklas Clemens Noldin
Advisor: DI Dr. Simon Ginzinger, MSc

Salzburg, Austria, 10.08.2020

Affidavit

I herewith declare on oath that I wrote the present thesis without the help of third persons and without using any other sources and means listed herein; I further declare that I observed the guidelines for scientific work in the quotation of all unprinted sources, printed literature and phrases and concepts taken either word for word or according to meaning from the Internet and that I referenced all sources accordingly.

This thesis has not been submitted as an exam paper of identical or similar form, either in Austria or abroad and corresponds to the paper graded by the assessors.

10.08.2020
Date


Signature

Niklas Clemens Noldin
First Name *Last Name*

Kurzfassung

Diese Arbeit behandelt die Frage nach der technischen Möglichkeit und Realisierbarkeit Spielzüge, im American Football mit Machine Learning vorherzusagen. Die Vorhersagen werden für eine binäre Klassifikation zwischen Pass- und Laufspielzügen vorgenommen und weiters mit detaillierteren Ergebnissen über Passtiefe und Laufroute erstellt. Die Arbeit legt ihren Fokus auf die Anwendung von ausgeprägtem Fachwissen im American Football. Um dieses Fachwissen zu erreichen, wurde ein Interview mit dem Trainerberater und American Football Coach Max Sommer geführt, um die wesentlichen Faktoren im Play-Calling zu ermitteln. Aus dem Vergleich von einem logistischen Regressionsmodell, einer Support-Vector-Machine und einem neuronalen Netz erzielte die Support-Vector-Machine die besten Ergebnisse mit einer maximalen Vorhersagepräzision von 87,9%. Diese Ergebnisse können als nützliches Hilfsmittel für Defensiv-Koordinatoren gesehen werden. Weiters generieren die erweiterten Vorhersagen mit Mehrklassenklassifizierung hilfreiche Informationen über das Play-Calling. Mehrere Merkmalsvektoren wurden beurteilt, woraus sich ein Datensatz mit fünf Eingabeparametern als am besten geeignet herausstellte. Dieser besteht aus der verbleibenden Spielzeit, dem Abstand zur Endzone, dem Abstand zu einem neuen First Down, der Punktedifferenz und dem Prozentsatz, wie oft die Offensivmannschaft den Ball, bei dem derzeitigen Down, wirft. Die Modelle werden für jedes Down separat trainiert, um Play-Calling Strategien für jedes Down einzeln erfassen zu können.

Abstract

This thesis explores the possibility and feasibility of the prediction of play types in American football with machine learning. The projections are made for a binary classification between pass and run plays and further created with more detailed results about pass depth and run location. The thesis lays its focus on the extraction and application of strong domain knowledge. To generate this information, an interview with coaching consultant Max Sommer was held, and the essential parameters of play-calling are obtained. Out of a logistic regression model, a support vector machine and a neural network, the support vector machine generated the best results with a maximum prediction score of 87.9%. These results can be seen as a useful aid for defensive coordinators and also the extended prediction with multiclass classification generates helpful information for play-calling. Multiple sets of descriptive features were judged, and a set of five input parameters were found to be the most suited for the task. This includes the time remaining in the game, the distance to the goalline, the distance to a new first down, the score differential and the offensive team's passing percentage for each down. The models were trained separately for each down to capture the unique structure of play-calling for each down.

Contents

List of Figures

Listings

List of Tables

Abbreviations

NFL	National Football League
NCAA	National Collegiate Athletics Association
SVM	Support Vector Machine
SVC	Support Vector Classifier
MLP	Multilayer Perceptron
PDA	Predictive Data Analytics
ML	Machine Learning
CV	Cross Validation
BFGS	Broyden–Fletcher–Goldfarb–Shanno algorithm
LBFGS	Limited-memory BFGS
MLB	Major League Baseball
MIT	Massachusetts Institute of Technology
CRISP-DM	Cross-Industry Standard Process for Data Mining
CSV	Comma-Separated Values

1 Introduction

Generally, in sports, the prediction of an opponent's actions before their execution represents a tremendous advantage and opens up numerous possibilities in forming a strategy. It allows a competitor to adjust their system and to have a benefit compared to their opponents (Stefani 1987, 63). Additionally, in professional sports entire games are recorded on video and quantitative data is retained in datasets, which makes the domain of sport a uniquely authentic area for exploring research ideas and especially data analytics (Morgulev, Azar, and Lidor 2018, 213). Sports analytics is said to have its origin with the research of Charles Reed in the 1950s (Reep and Benjamin 1968, 581), who analyzed the number of passes in soccer that lead to a goal, which later constituted the "long ball" style of play and changed English soccer for decades (Morgulev, Azar, and Lidor 2018, 214).

In the 1960s, the Dallas Cowboys began to use individual play data in American football to recognize trends in the play types of their opponents and align their defensive players to stop the expected play. This lead to the Dallas Cowboys to be one of the most successful teams of that era and all other NFL teams and even most American college teams to now deploying similar techniques (Stefani 1987, 63).

Due to its iterative and less dynamic nature, which is further explained in 3, American football can be seen as more fitting to on-field sports analytics than, for example, soccer. Because every play begins with a restart, offensive and defensive players have the opportunity to align themselves against their opponents (Stefani 1987, 62). These alignments can, therefore, be improved through data analytics. Furthermore, American football relies heavily on strategy and less on spontaneous actions and hence is particularly suitable for predictive data analytics (Stefani 1987, 61-62).

Kelleher, Namee, and Arcy (2015, 41) defined predictive data analytics as "the art of building and using models that make predictions based on patterns extracted from historical data". To built the bespoken model, and therefore, to extract logical patterns from historical datasets, a variation of machine learning algorithms can be used (Kelleher, Namee, and Arcy 2015, 41-42). Due to the recent performance improvements in computers and the ongoing research in machine learning, predictive data analytics can now be deployed to various new extents, and machine learning is already used in all major league sports by most teams (Morgulev, Azar, and Lidor 2018, 213-214).

The essential research question this thesis tries to answer is if it is possible to predict play types in American football good enough to create a useful aid for defensive play-callers. Furthermore, it is elaborated to what extent of accuracy this can be achieved. In order to answer these questions, this thesis creates multiple models to predict offensive plays in American football and compares three different machine learning classifiers. Different sets of descriptive features are created and tested in order to find the best possible accuracy. Furthermore, models are trained for three different classifications. The simplest being the binary classification of either run or pass play and the most complex differentiating between a deep and a short pass, and an inside and an outside run. The goal is to develop a strong domain understanding and therefore find the most relevant descriptive features to remove unnecessary complexity and increase the feasi-

bility in a potential on-field application. This is achieved by interviewing Max Sommer, Head Coach of the National Football Team of Austria. Coach Sommer additionally is an independent American football consultant and expert for American football on Austrian national sports television.[1]

2 Methods

This thesis has the goal to predict play types as precise as possible while maintaining an appropriate accuracy. In order to explore the feasibility of multiple levels of precision, each model is trained for three different classification problems. These three classifications consist of binary classification, a ternary classification and a quaternary classification. The output classes are ”pass” and ”run” for the binary model, ”deep pass”, ”short pass” and ”run” for the ternary model and ”deep pass”, ”short pass”, ”inside run” and ”middle run” for the quaternary classification. These are the most common classifications in American football and are created after a conversation with Coach Max Sommer to fit the needs of a defensive coordinator the most.

The play-by-play datasets are taken from the public NFL.com website and are available from the 2009 season to the 2019 season. The dataset was scraped by Ronald Yurko and taken from his public Github repository. For the training of the models, each of the three classification datasets is divided into training and test datasets according to the time of their execution. This mimics a real-world workflow, in which the earlier datasets would be available to the coach or coordinator using the tool. The model is then assessed and tested on the later plays. The division into training and test datasets is also performed in three different ways. For the first dataset, each season is divided separately, for the second dataset a pair of seasons is taken and then divided chronologically, and for the third dataset the whole first eight seasons are taken as training-set and the following two seasons are taken as validation-set. This separation is explained in detail in subsubsection 8.4.2.

The candidate models for this thesis are chosen to represent a wide variety of machine learning classifiers and should try to represent the spectrum of classifiers. All datasets, and train and test splits are used on a logistic regression model, a scalable vector classifier, and a multilayer perceptron. The models are further explained in detail in subsection 5.4.

As each down in American football has a distinct characteristic, which is elaborated in section 3 and subsection 8.2, the models are trained for separately for each down. This increased the accuracy of the models significantly.

In total, 132 models are created for each logistic regression and support vector machines. This is calculated as three target datasets times four downs and 11 seasons. As the neural network is trained on the whole dataset, only 12 models are trained.

Finally, multiple subsets of the feature space are tested and assessed to find the most relevant parameters. section 8 is structured according to the cross-standard industry process for data-mining which is elaborated in section 7.

1. www.maxcoaching.at

3 Rules of American football

American football is known to have been invented in the 1880s when a rule change, proposed by Walter Camp, was accepted by all major colleges. The introduction of a quarterback and the scrimmage, the act where one person starts the play by kicking or snapping the ball backwards, ultimately distinguished the game from rugby, which was widely played in the USA. After numerous rule changes between 1880 and 1883, the game was roughly invented as it is played today (Braunwart and Carroll 1997, 3).

The goal of today's American football is to advance as far as possible down the field to either reach the endzone (touchdown) or kick the ball through the goalposts (field goal). For each touchdown, six points are awarded, and the team attempts to score an extra point which requires kicking the ball through the H-shaped goalpost. A field goal can always be attempted and if successful 3 points are awarded (Stefani 1987, 62).

A football game consists of four quarters with 15 minutes each. It starts with the kickoff, in which one team kicks the ball in the opposing team's direction for them to catch it and return the ball in the kicking team's direction (Stefani 1987, 62).

After the kickoff, the team in possession of the ball has four attempts (downs) to advance for a minimum of 10 yards in total. After advancing for 10 yards, the team is awarded another set of four attempts, with the ultimate goal to reach the endzone on a 100-yard long field. A single down ends if the ball carrier touches the ground, steps out of the field, or crosses the goal line with the ball. It also ends if an attempt to pass the ball is dropped or a penalty occurs. To move down the field one player can either run with the ball or throw to another member of the team who then proceeds to run with the ball. Each play will start from the point where the previous play ended (not taking penalties into account). The ball will change possession if the offensive team can't proceed over the ten-yard mark during their four downs, the offensive team scores, or the ball is caught or recovered by the defensive team (Stefani 1987, 62). The defending team has the task to minimize the other team's spatial profit and ultimately get into possession of the ball.[2] The type of play which is carried out by the offensive team is usually decided by the head coach or offensive coordinator before the play is carried out, but can also be decided by the quarterback on the field, depending on the situation in the game (Jordan, Melouk, and Perry 2009, 3).

In contrast to soccer, the management of the game clock in American football is different. Though the official game time sums up to 60 minutes, the average game time in 2019, according to the NFL, was approximately 3 hours and 7 minutes.[3] This results from the game time being stopped on multiple occasions during the game, e.g. after an incomplete pass, after the ball is out of bounds or after a foul. The offensive team then has up to 40 seconds to start the clock again after each play.[4] This creates the possibility to easily collect quantitative data about each play, team and player and gives time to analyze this data and decide on the subsequent strategic actions based on the gained insights. This makes American football and especially the National

2. 2019 Official Playing Rules of the National Football League
3. `https://operations.nfl.com/stats-central/chart-the-data`
4. 2019 Official Playing Rules of the National Football League

Football League, one of the most suitable sports leagues to improve their game through data analytics (Morgulev, Azar, and Lidor 2018, 213).

4 The relevance of play type prediction in American football

According to Assuncaõ and Pelechrinis (2018, 237), data and analytics have been part of the sports industry since the first box score in baseball was recorded in the 1870s. Nonetheless, due to technological advancements in recent years, advanced data mining and machine learning techniques could be used to aid the operations of sports franchises (Assuncaõ and Pelechrinis 2018, 237). Since the 1870s, as the leagues saw opportunities in the collected data, more and more data began to be collected through each game. Detailed data is now widely used throughout the entire sports industry to assist teams and leagues in multiple domains. Morgulev, Azar, and Lidor (2018, 213) defines three types of data-driven analyses:

1. the field-level analysis focused on the behaviour of athletes, coaches, and referees
2. analysis of management and policymakers' decisions
3. analysis of the literature that uses sports data to address various questions in the fields of economics and psychology

This means that next to the in-field use cases, data analytics is also heavily used in business decisions like a cost-benefit analysis of hosting the Olympics or calculating long-term impacts of major sporting events. The NFL even introduced ticket revenue sharing, equal broadcast revenue sharing, and a salary cap to balance out the league's teams, because Rottenberg (1956, 246) found that the uncertainty of outcome is needed for the consumer to be willing to pay admission to the game, suggesting, that fans enjoy viewing competitions with an unpredictable outcome more (Morgulev, Azar, and Lidor 2018, 216). Furthermore, data-driven methods are used to learn about human behaviour in certain situations in sports, e.g. learning tendencies in free throws in basketball or penalties in soccer (Morgulev, Azar, and Lidor 2018, 217-218). Predicting play types in NFL games can be classified as field-level analysis.

According to Forbes, the average value of a National Football League team in 2019 is USD 2.86 billion, ranging from USD 5.5 billion for the Dallas Cowboys to USD 1.9 billion for the Buffalo Bills. This makes the Dallas Cowboys the most valuable sports franchise in any sport, with the NFL, which consists of 32 teams, holding 29 of the 50 most valuable sports franchises.[5] Furthermore the NFL's annual revenue is the most of any professional sports league at around USD 15 billion in 2019, with the projected goal of USD 25 billion by 2027.[6] This comes even though the NFL's betting market is minimal, as sports betting is only fully allowed in 18 of 50

5. https://www.forbes.com/sites/kurtbadenhausen/2019/07/22/the-worlds-50-most-valuable-sports-teams-2019

6. https://www.chicagotribune.com/sports/ct-spt-NFL-revenue-super-bowl-20190128-story.html

states in the USA at the time of writing. The American Gaming Association projects the NFL's revenue to increase by an additional USD 2.3 billion per year due to widely available, legal sports betting.[7] This concludes the economic relevance of the National Football League with the prediction of play types gaining relevance in the future years, due to the projected increase of the sports betting market.

Nevertheless, is it strategically relevant to predict an opponent's play type? It is possible to elucidate this question with game theory. American football is a zero-sum game, which means, that the yards gained by the attacking team are yards given up by the defending team, making the one team's payoff the negative of the other team's payoff. If we break down American football strategy to play calling and reduce this to only passing and running plays this theoretically corresponds to a matching pennies game, in which it is a safe sign to win if one player knows what the opponent's strategy will be (McGarrity and Linnen 2010, 792). In this basic scenario, the accuracy of the prediction directly corresponds to the percentage of success. Translating this back to American football means that the better one's prediction of play types is, the more successful the play-calling will be.

4.1 Takeaways from the interview with Max Sommer

Max Sommer was interviewed to clarify the relevance of the prediction of play types and to further gain an elaborate domain understanding. The conversation was held as a video call, and a transcript is available in subsection B.1.

Coach Sommer explained that in the Austrian and European international leagues American football is played by the rules of the NCAA (the collegiate football league of the USA) and technological aids were only allowed around five years ago (2015). Also, the communication between coaches and players over a headset is not allowed. In contrast to the NFL, this makes it a lot more difficult to create a useful aid for defensive play-calling, considering the limited time between each play.

Nonetheless, defensive coordinators, who usually call the defensive plays, often have play sheets of the opposing team's tendencies with a table of passing percentages per down and distance.

Technology to assist the coaching decisions are very limited in the European leagues. Max Sommer clarified that the by far most used tool is Hudl, which is a sports video analytics platform. The base information for the data collection in Hudl is video, from which a coach can put in information about each play in the form of a spreadsheet. Hudl is mostly used to analyze plays of next week's opponent and to study the playing style of the team. However, if a coach has entered enough metadata to each play, Hudl tries to predict the possible outcomes of play scenarios and reveals tendencies in the playcalling of an opponent. This feature is however only available before the game and cannot be utilized during the game, on the field.

7. https://www.americangaming.org/new/nfl-could-reap-2-3-billion-annually-due-to-legalized-sports-betting

During the conversation, it became clear that a tool to assist a defensive coordinator does not primarily have to have a great prediction accuracy in terms of true positives against false positives but needs to show the probabilities of each play type for the current scenario. It has to be clear that in the end, the play is called by the defensive coordinator and not by the algorithm. Therefore it is also relevant information if the outcome is, for example, 50/50 between the classes, as this tells a coach, that there are currently no tendencies to a certain play type. Coach Sommer further clarified that while the differentiation between a pass play and a run play is very useful information, the additional classification in deep and short passes, and inside and outside runs can be a great improvement to defensive play-calling. The selected target classes for this thesis were chosen after this interview to represent the coaching needs most accurately.

Important descriptive features were discussed and evaluated with Coach Sommer, the results of this evaluation are described in detail in subsubsection 8.3.1

5 Fundamentals of machine learning

Machine learning can be broadly defined as computational methods to improve performance or predict variables using past information (Mohri, Rostamizadeh, and Talwalkar 2018, 1). More technically, machine learning is an automated process that extracts patterns from data (Kelleher, Namee, and Arcy 2015, 43). We can achieve the extraction of patterns with the combination of computer science, statistics, probability and optimization, and creating a machine learning algorithm with these methods (Mohri, Rostamizadeh, and Talwalkar 2018, 1-2). Some of the standard machine learning tasks are classification, regression, ranking, clustering and dimensionality reduction.

- Classification is referring to the assignment of a document to different categories. This can be an article that is assigned to categories like politics or business, or optical character recognition, in which single characters are registered from an image (Mohri, Rostamizadeh, and Talwalkar 2018, 3).
- Regression is the prediction of a real value depending on an item. For example, predicting stock values or variations of economic variables. In contrast to a classification task, the error of a regression model can easily be measured, while a classification can either be true or false (Mohri, Rostamizadeh, and Talwalkar 2018, 3).
- Ranking is the task of finding the best possible order of items. The canonical ranking example being the ranking of web pages depending on a search query (Mohri, Rostamizadeh, and Talwalkar 2018, 3).
- Clustering is the act of partitioning a dataset into subsets. For instance, the identification of communities in a dataset of people (Mohri, Rostamizadeh, and Talwalkar 2018, 3).
- Dimensionality reduction or manifold learning deals with the representation of multidimensional data in a lower-dimensional description. This is often used in the prepro-

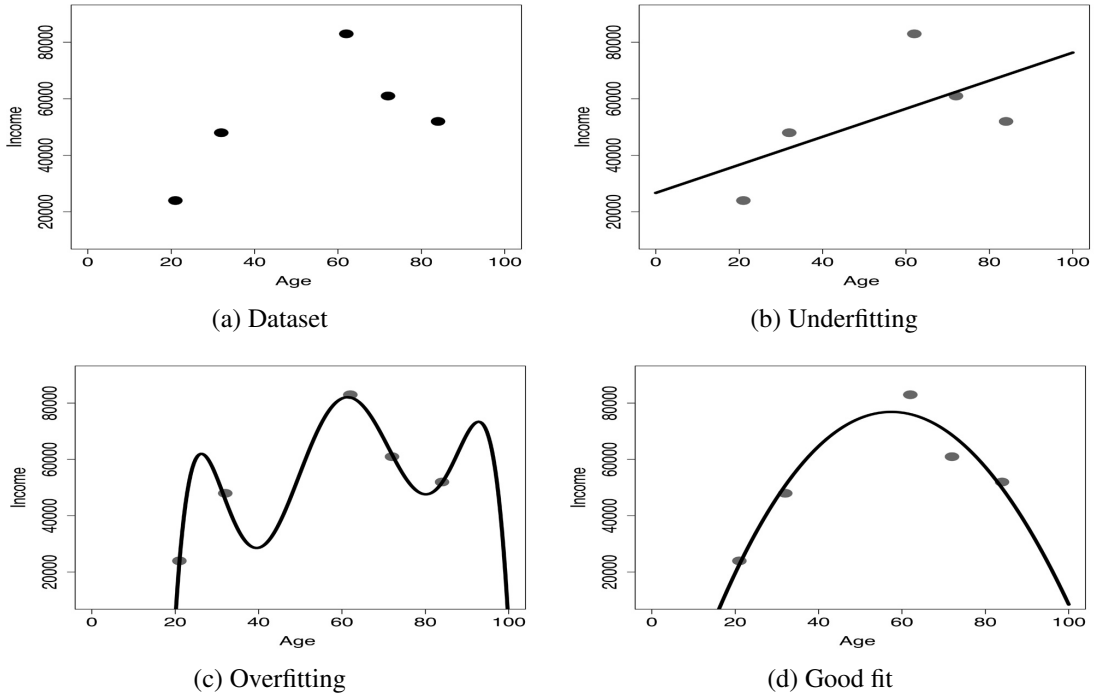
cessing of images for computer vision tasks (Mohri, Rostamizadeh, and Talwalkar 2018, 3).

Notwithstanding the task of a machine learning application, the goal of machine learning is generalization (Mohri, Rostamizadeh, and Talwalkar 2018, 7), meaning that the goal is to find a hypothesis that represents not only the dataset used for training but generalizes upon the training data and can, therefore, predict the values of data points not contained in the training set (Bishop 2007, 2). This creates one of the key problems in machine learning, as the primary goal is to find patterns in large datasets, but furthermore, these found hypotheses have to be generalized. This originates from the learning datasets often not exactly representing the underlying pattern, due to noise in the dataset, meaning that some instances are mislabeled or contain false data (Mohri, Rostamizadeh, and Talwalkar 2018, 7-8; Kelleher, Namee, and Arcy 2015, 46). Thus it is important to find a middle ground in which the model best generalizes a relationship. The sample size of the training set and the chosen algorithm are essential factors for the resulting generalization. A small sample size paired with a complex algorithm might result in overfitting and becomes sensitive to noise in the data. Contrary, a simple algorithm matched with a complex dataset could result in lacking accuracy, which is known as underfitting (Mohri, Rostamizadeh, and Talwalkar 2018, 8; Kelleher, Namee, and Arcy 2015, 47). Consequently, it is one of the core skills of a data analyst to select the appropriate algorithm for a distinct prediction task, Kelleher, Namee, and Arcy (2015, 51) describing it as one of the great arts of machine learning. Figure 1 shows the difference between underfitting and overfitting on a dataset representing the relation between age and income (Kelleher, Namee, and Arcy 2015, 47).

Depending on the types of available training data, there are multiple machine learning scenarios. The most common ones are supervised learning, unsupervised learning, semi-supervised learning, reinforcement learning, transductive interference, online learning, and, active learning (Mohri, Rostamizadeh, and Talwalkar 2018, 7). The scenarios used in this thesis are further explained in the following chapters. In practice, there may be many other intermediate and more complex scenarios (Mohri, Rostamizadeh, and Talwalkar 2018, 7).

5.1 Supervised learning

One may refer to supervised learning if the whole dataset used for training is labelled with the desired value to predict. This is the most common scenario, and predictive data analytics usually utilizes a supervised learning approach (Mohri, Rostamizadeh, and Talwalkar 2018, 6; Kelleher, Namee, and Arcy 2015, 43). In supervised learning, a prediction model is trained with the relationships between descriptive features and a target feature with the descriptive features being the whole of the data that is used to make a prediction and the target feature being the value that is predicted. Generally speaking, each value used in a predictive model is referred to as a feature. The relationships used to create a predictive model are based on historical examples or instances. Figure 2a shows that the model is trained using a dataset with several target features and a machine learning algorithm. This model can then be used to predict the target feature of a query instance, as shown in figure 2b (Kelleher, Namee, and Arcy 2015,



(a) Dataset (b) Underfitting

(c) Overfitting (d) Good fit

Figure 1: Fitting problems of predictive models. (Kelleher, Namee, and Arcy 2015, 52)

43-45). The query instance is a set of descriptive features, without their corresponding target features (Kelleher, Namee, and Arcy 2015, 173). Finding relationships between descriptive features and target features can be achieved manually on small datasets, with machine learning; however, this is easily achievable on large datasets with multiple features. These technological advancements enable a multitude of possibilities and let us create more robust models (Kelleher, Namee, and Arcy 2015, 45).

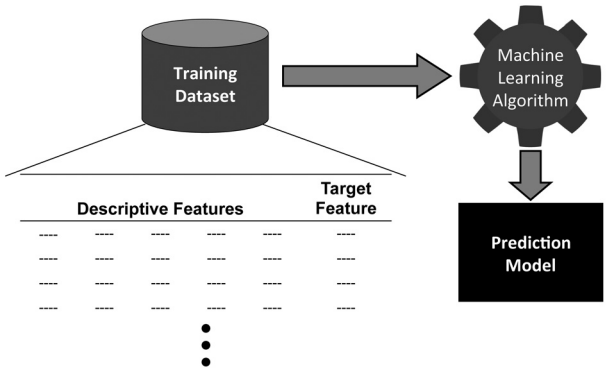
We can divide supervised learning tasks into classification and regression. If the target feature consists of discrete categorical variables, the task is a classification problem. If the desired output is one or multiple continuous variables, it is a regression problem.

5.1.1 Regression

Recapitulating, regression is any machine learning task in which the desired result is a real-values label (Mohri, Rostamizadeh, and Talwalkar 2018, 4). One of the most common models in regression is linear regression, in which the target variable is described by a linear function of the inputs. Mathematically, this can be expressed as:

$$y(x) = w^T x + \varepsilon = \sum_{j=1}^{D} w_j x_j + \varepsilon$$

where $w^T x$ is the scalar product of the model's weight vector w and the input vector x. ε is the error between the prediction and the true response and D represents the dimensionality of



(a) A predictive model is created with a historical training dataset and a machine learning algorithm.

(b) A prediction is made when the model is given a query instance.

Figure 2: The two phases of a machine learning model. (Kelleher, Namee, and Arcy 2015, 43)

the input vector. ε is assumed to be gaussian or normally distributed, which is expressed as $\varepsilon \sim N(\mu, \sigma^2)$ where μ is the mean and σ^2 is the variance. If we further connect this with the previous expression we can note the model as:

$$p(y \mid x, \theta) = N(y \mid \mu(x), \sigma^2(x))$$

This makes clear that the model is a conditional probability density. In most basic case of linear regression μ can be considered to be a linear function of x and σ^2 to be a fixed noise, therefore we get $\mu = w^T x$ and $\sigma^2(x) = \sigma^2$. In the previous expression, $\theta = (w, \sigma^2)$ are the parameters of the model. Supposed the input is one-dimensional we can expect μ to be $\mu(x) = w_0 + w_1 x = w^T x$. Here w_0 is called the bias term, and w_1 is the slope of the function. A constant 1 term is prepended to the input vector to allow the combination with other terms of the model. The vector x is therefore defined as $x = (1, x)$. In this simplified case, if w_1 is positive, the output is expected to increase as the input increases.

Linear regression can further model non-linear relationships by replacing x by a non-linear function of the inputs $\phi(x)$. Consequently, we can generally define the model as:

$$p(y \mid x, \theta) = N(y \mid w^T \phi(x), \sigma^2)$$

This is called a basis function expansion. If we define $\phi(x)$ as $[1, x, x^2, ..., x^d]$ the resulting regression is called a polynomial regression (Murphy 2012, 19-20).

In machine learning, many popular methods, such as support vector machines, neuronal networks and classification and regression trees are different ways of estimating a basis function from data (Murphy 2012, 20).

5.1.2 Classification

Classification is the mapping of an input vector x to an output value y, where $y \in 1,...,C$ and C is the number possible output classes. This means that y is a single discrete value of a collection of possible outcomes. If $C = 2$ the classification is called binary classification and if $C > 2$ it is called multiclass classification. If the output labels are not mutually exclusive, the classification is called a multi-label classification (Murphy 2012, 8).

We can generalize the linear regression introduced in subsubsection 5.1.1 to a binary classification setting, creating a logistic regression. Firstly the gaussian distribution is replaced by a Bernoulli distribution to better fit a binary output that is here considered to be $y \in 0,1$. This leads us to the model

$$p(y \mid x,w) = Ber(y \mid \mu(x))$$

where $\mu(x) = E[y \mid x] = p(y = 1,x)$. As in linear regression we create a linear combination of the inputs but pass it through a function to ensure that $0 < \mu(x) < 1$:

$$\mu(x) = sigm(w^T x)$$

The sigm represents the sigmoid, or logistic function, which is defined as

$$sigm(\eta) \triangleq \frac{1}{1+exp(-\eta)} = \frac{e^{\eta}}{e^{\eta}+1}$$

The logistic function maps all real-values numbers to $[0,1]$, which is needed for the output to be interpreted as a probability. Putting linear regression together with the Bernoulli distribution and using the sigmoid function we can define a logistic regression model as follows:

$$p(y \mid x,w) = Ber(y \mid sigm(w^T x))$$

Though this is called logistic regression, it is, in fact, a form of classification (Murphy 2012, 21-22).

5.2 Unsupervised learning

Unsupervised learning is usually used for clustering, density estimation, which means to determine distributions in the input data, or dimensionality reduction. The training data in unsupervised learning consists of descriptive features, also called input vectors, without corresponding target features (Bishop 2007, 3).

An example of clustering and unsupervised learning can be a company trying to find natural groupings in their customers. The input data can be the demographic information and buying habits of all recent customers. A clustering model can be used to divide the customers into groups with similar habits and attributes. After the different groups are found, separate strategies can be applied to maximize the company's revenue. This process is called customer segmentation (Alpaydın 2010, 11-12).

5.3 Online learning

In online learning, the training and testing phases are intermixed. Here the learner receives an unlabelled training point, makes a prediction and receives the right label afterwords. Thereby the goal of the learner is to minimize the cumulative loss over multiple iterations (Mohri, Rostamizadeh, and Talwalkar 2018, 7). This is especially useful if the whole training set is not available from the beginning, but the instances are collected or given to the learner one after another, and the model is adjusted after each iteration. Differently from other machine learning scenarios, the error function in online learning is written over a single instance and not over the whole training set. Using an online approach creates several advantages:

- In online learning, the training set doesn't have to be stored entirely but can be abolished after the training. This saves a lot of cost and storage.
- The relationship between descriptive and target features might change over time. The online learner will adjust to newly discovered patterns and can adapt to changes in the testing data.
- An online learner can adjust to physical changes, for example, sensors or components wearing out.

The training of neural networks generally uses online learning, where each parameter is tweaked a little bit for each iteration and adapting slowly to the desired results (Alpaydın 2010, 240-241).

5.4 Selected machine learning classifiers

The data mining problem discussed in this thesis is classified as a classification problem and uses supervised learning to create predictive models. The target features are binary and multivariate, but to minimize complexity all are treated as multivariate classifications. The three classifiers chosen in this thesis are selected according to three factors: The complexity of the models should vary and should capture different areas of the spectrum of machine learning algorithms, the models have to be able to predict single probabilities for each output class, and the models need to be able to predict multiclass labels. These factors arise from the educational aspect of this thesis, the technological prerequisites for multiclass classification, and the suitability to assist American football coaching staff.

Logistic regression is chosen as one of the simplest classifiers in the machine learning spectrum and the implementation in the machine learning library scikit-learn allows multivariate and probabilistic classification.

The support vector classifier is chosen as a slightly more complex classifier. It is also able to predict multiclass targets values and each class's probability by default.

To top off the spectrum a state-of-the-art neuronal network is chosen for the task. More specifically, a multilayer perceptron with backpropagation is used. This is especially common in supervised learning.

5.4.1 Logistic regression

Logistic regression is described in-depth in subsubsection 5.1.2. The parameter estimation is done using a limited-memory BFGS (LBFGS) algorithm which is a quasi-Newtonian optimization method (Mohri, Rostamizadeh, and Talwalkar 2018, 313). A detailed description of the LBFGS algorithm would go beyond the scope of this thesis. It is, however, recommended for multivariate classification by the scikit-learn documentation.[8]

5.4.2 Support vector machines

Support vector machines (SVM) are an error-based approach to predictive modelling. The goal of SVMs is to find the best decision boundary, namely the hyperplane that separates instances of each class from another class. This is achieved by maximizing the distance from the decision boundary to the nearest instance in the training set. This distance is called the margin extent and the nearest instances to the decision boundary, and therefore the most important instances in the dataset are called the support vectors. A support vector machine model is defined as

$$M_{\alpha,w_0}(q) = \sum i = 1^s (t_i \times \alpha[i] \times (d_i * q) + w_0)$$

where q is the query instance. Note that the support vectors s are defined as the combination of descriptive and target features $(d_1,t_1)...(d_s,t_s)$. w_0 represents the first weight of the decision boundary. During the training process, α is determined as a set of parameters, one for each support vector. When this equation results in a value greater than 1 the query instance is classified as the positive output class. If the value is less than -1 the instance is classified as the negative output class. The values between -1 and 1 represent the margins of the decision boundary (Kelleher, Namee, and Arcy 2015, 436-438).

To find the optimal decision boundary the support vectors, w_0 and the α parameters have to be found. This exploration is called a constrained quadratic optimization problem and is not further described in this thesis (Kelleher, Namee, and Arcy 2015, 436-438).

5.4.3 Multilayer perceptron

Artificial neural networks, which multilayer perceptrons are part of, try to mimic the human brain. The brain is much better in understanding topics that could have a significant economic impact when understood by machines. For example, vision, speech recognition, and learning. To find ways to solve these problems on a computer, it has to be studied how the human brain understands the given information. While a computer only has one processor, a brain has around 10^11 processing units, called neurons, which are believed to be significantly slower and simpler than a computer processor. The brain further differentiates from a computer in the connection between each neuron. Each neuron is connected to around 10^4 other neurons. The connections are called synapses and are believed to be the memory of the brain (Alpaydın 2010, 233-234).

8. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Neural networks can be seen as a paradigm for parallel processing. Instead of one complex processor that might communicate with another processor, as it is often used in modern hardware, the goal of neuronal networks is to create many simple processors that are connected to each other and execute the same task but with different input values. The difficulty in this model is the creation of the many local parameters to input into the processing units. Whatsoever, this is not necessary if these parameters can be learned from examples (Alpaydın 2010, 235-237).

A perceptron is a basic processing element that gets inputs from the environment or from other perceptrons and calculates a certain value from all input values. In the simplest scenario, this value could be a weighted sum. Based on this value and an activation function a binary output is then created, which corresponds to an output class. The goal is then to find the optimal weights for the calculation by assessing the binary output against a predefined target feature with an error function and changing the weights accordingly. Mathematically speaking, the function of the perceptron defines a hyperplane in the feature space which serves as the decision boundary (Alpaydın 2010, 237-240). The training of a perceptron is usually done in an online learning scenario but could be trained in an offline context and then be inserted into the network (Alpaydın 2010, 241). There is no single implementation of a perceptron but they can vary enormously in the perceptron function, the activation function and the error function. Furthermore, countless parameter estimation methods can be applied. The details on the several functions in a perceptron are not discussed in this thesis.

A simple perceptron only learns one parameter, therefore can only represent a linear decision boundary. A multilayer perceptron can use a collection of parameters to capture more complex coherences in the data (Alpaydın 2010, 245-247). A multilayer perceptron network is flexible enough to represent any relevant calculations and relationships in data (Alpaydın 2010, 248).

6 Machine learning and predictive data analytics in sports

In section 1 predictive data analytics (PDA) is defined as the art of building and using models that make predictions based on patterns extracted from historical data but it is not yet clear what such models and patterns are and how they are created and read. section 6 describes the basics of predictive data analytics and machine learning and gives an overview of applications of predictive data analytics in sports.

The most common applications of predictive data analytics include price predictions, risk assessment in managemental decisions, document classification, and medical applications like diagnoses and medicine dosage predictions. It is important to note that, in data analytics, the word prediction doesn't have a temporal aspect and can be used to describe the assignment of a value to any unknown variable. A model that's trained using machine learning predicts these variables (Kelleher, Namee, and Arcy 2015, 41).

While the prediction of future events is an integral part of multiple branches and businesses, insurance companies are particularly reliant on data analytics. Insurance companies have always relied on forecasts of policy's risks and costs, but they came from educated guesses to now having predictive data analytics as a best practice. In insurance companies, many factors, such

as income, credit history and outstanding loan balances, are combined to create a single credit score or to identify potentially fraudulent claims. Generally speaking, predictive models usually produce a score, with a higher score indicating a higher likelihood of the given behaviour or event occurring (Nyce 2007, 1).

6.1 Sports analytics

Morgulev, Azar, and Lidor (2018, 214) defines sports analytics to be the management of structured historical data, the application of predictive analytic models that use these data, and the utilization of information systems, to inform decision-makers and enable them to assist their organizations in gaining a competitive advantage on the field of play. Hence, describing the procedure of gaining business-relevant information from historical data. Moreover, the data collected in sports increased gradually since the 1870s, with the NFL fully implementing player location tracking during every game since 2015 (Assuncaõ and Pelechrinis 2018, 237).[9] Similarly, the Major League Baseball (MLB) introduced STATS SportVU, a six-camera system to track each player's real-time position. Additionally, the PITCHf/x, HITf/x, and FIELDf/x systems capture every pitching, fielding and hitting in the MLB (Morgulev, Azar, and Lidor 2018, 215; Nistala and Guttag 2019, 1). The growth of the global sports analytics market also represents the growth in the collected data. The sports analytics market is projected to grow from USD 1.9 billion in 2019 to USD 5.2 billion by 2024 which corresponds to a 22.0% compound annual growth rate.[10] Additionally, Fry and Ohlmann (2012, 105) claims that the market of spectator sports is estimated to be double the American automotive industry. With the real-time location data of each player and the ball, in addition to the already captured quantitative data, basically makes it possible to capture everything that happens on the field. This can be used to generate useful insights, not only for strategic decisions but also to decide on potential rule changes (Assuncaõ and Pelechrinis 2018, 237). Sports analytics are further known to have revolutionized the baseball player market after the Oakland Athletics used a statistical approach to create a highly efficient team in 1997. Moneyball, a best-selling book and movie, later popularized these events and introduced the broader sports community to the benefits of sports analytics (Morgulev, Azar, and Lidor 2018, 215; Fry and Ohlmann 2012, 105). Since then, many major American sports franchises, like the Dallas Mavericks and the New York Mets, employ analytics and statistics specialists in crucial roles of their teams, e.g. Daryl Morey, general manager of the Houston Rockets has degrees in statistics and computer science, but little experience with playing basketball (Fry and Ohlmann 2012, 105-106). Morey also co-founded the MIT Sloan Sports Analytics Conference, which features over 3000 annual visitors and provides a forum to discuss the increasing role of sports analytics with industry professionals.[11]

Another valuable aspect of sports analytics is health. Understanding injury occurrences and prevention strategies can significantly decrease injuries, consequently providing a significant

9. https://www.forbes.com/sites/centurylink/2014/09/12/playing-the-data-game-why-the-nfl-is-now-tracking-players-with-rfid-chips
10. https://www.researchandmarkets.com/reports/4904383/sports-analytics-market-by-sports-type
11. http://www.sloansportsconference.com/about/

advantage on the playing field and increased player safety. Using data such as the playing time and the distance run by a player, measures can be taken to minimize the risk of injury. This led to the implementation of sports injury surveillance systems in many major sports leagues, including the NFL (Wasserman et al. 2018, 387-388; Orchard, Seward, and Orchard 2013, 734-735). The ongoing increase in the collected data, especially with medical data, such as heart rate, distance run, and intensity of work, could significantly decrease injuries and especially recurring injuries (Wasserman et al. 2018, 397-398).

In summary, it can be said that sports analytics is an essential factor in the growth of the sports market and is increasingly growing in relevance and revenue. The goals of sports analytics are mainly to increase consumer interest, player health and management efficiency.

6.2 Related work in playtype prediction in American football

There have been multiple previous attempts of predicting play types in American football in the previous years. Most focused on the binary classification between pass plays and run plays and none of the reviewed papers differentiated play types any further (Fernandes et al. 2020, 35-43; Ota 2017, 1-32; Lee and Kitani 2016, 1-5).

Lee and Kitani (2016, 1-5), Ota (2017, 2), and Fernandes et al. (2020, 35-43) independently developed models to predict if a team will run or pass the ball according to different features. While all projects only created a binary classification on the play type, the descriptive features varied between all three. Table 1 shows all used descriptive features and which of the project utilized them. Unexpectedly, only three features were used by all project, but the consensus seems to be found on the current down, the distance to a first down and the score difference.

Fernandes et al. (2020, 38) used four different classification methods. A classification and regression tree reached a prediction accuracy of 73.3%, a K-nearest-neighbors classifier reached 71.3%, a random forest classifier reached 74.7%, and a neural network reached a maximum of 75.3%. These models were built over the whole dataset. In order to reach an easier implementation during a game, the thesis further created decision tree models for each team with accuracies between 64.7% and 82.5% (Fernandes et al. 2020, 38-39).

Ota (2017, 12) used a multilayer perceptron with three hidden layers and 100 nodes each. The model was tested for first, second, and third down separately and also on the total dataset. While the prediction accuracy on the total dataset lies at 68.9%, the model for the third-down achieved an accuracy score of 86.8% (Ota 2017, 13).

Lee and Kitani (2016, 3) utilized four different machine learning approaches. A logistic regression approach reached an accuracy score of 73.7%, a linear discriminant analysis reached 72.7%, a random forest classifier reached 75.1%, and a gradient boosting machine could reach the maximum of 75.7% precision. An additional model was then created by taking the average of the previously trained gradient boosting machine and the random forest classifier, which achieved an accuracy score of 75.9% on the whole dataset but was then tested on each game separately. It achieved the scores of up to 94.6% for the Tennesee Titans in a 2013 game but also became inconsistent and scored a minimum of 47.2% on a 2013 game by the Denver Broncos. It

Feature	Project		
	Fernandes et al.	Ota	Lee and Kitani
Current Down	X	X	X
Distance to First Down	X	X	X
Score Difference	X	X	X
Madden ratings	X		X
Offense is at home	X		X
Pass plays called	X		X
Current Quarter	X		X
Formation	X		X
Field position	X	X	
Time Left in Quarter			X
Time Left in Game		X	
Off. players per Position			X
Def. players per Position			X
Off. player out of position			X
Turnovers			X
Pass plays faced			X
Score difference > 7			X
Pass completion rate			X
Years	X		
Minute	X		
Second	X		
Previous play	X		
Season pass. percentage	X		
Avg. yards gained pass	X		
Avg. yards gained run	X		

Table 1: Descriptive features of the related papers described in subsection 6.2.

was further found that the model scored badly on teams with mobile quarterbacks, which mean that they often decide to run the ball themselves. This makes sense as a mobile quarterback adds more flexibility to playcalling and therefore making it harder to predict the play.

None of the found projects discussed a multinomial classification approach to classify football plays.

7 Cross-industry standard process for data mining

As the data mining industry began to gain significance in the 1990s, the urge for a standard process began to surface (Wirth and Hipp 2000, 2). Therefore the CRISP-DM Special Interest Group consisting of data mining specialists from multiple leading companies in the industry, including DaimlerChrysler, SPSS, and NCR introduced the Cross-Industry Standard Process for

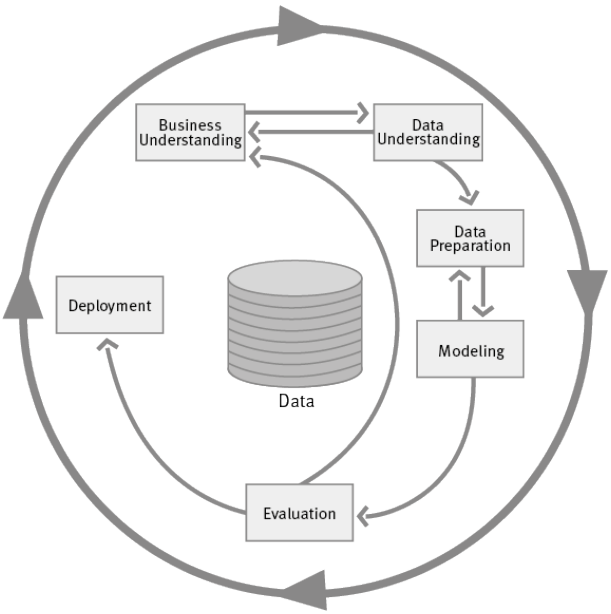


Figure 3: The six phases of the CRISP-DM process model and the relationships between the phases. (Chapman et al. 1999, 13)

Data Mining, short CRISP-DM in 1999 (Chapman et al. 1999, 3-4). CRISP-DM is a hierarchical model with four layers of abstraction. From general to specific, the layers are called phases, generic tasks, specialized tasks, and process instances (Chapman et al. 1999, 9). The CRISP-DM model is designed to work independently from industry, tool and application. Therefore the phases and generic tasks are intended to be applied as they are. Specialized tasks and process instances then have to be applied to each application to fit the needs of the project. Generally speaking, the project is divided into six phases, with each consisting of multiple second-level generic tasks. Generic tasks are general enough to cover all machine learning applications but cover the whole process for every data mining application. In the third level, the specialized tasks describe how the generic tasks are implemented. The fourth level describes the actual actions and decisions made to create the desired application. Figure 3 shows the basic structure of a CRISP-DM process. It is visible that a data mining project is not a linear process, but it is essential to step back and forth between the phases. Depending on the outcome of each phase, it has to be decided which task or phase comes next. While Figure 3 highlights the essential dependencies between business understanding and data understanding, and data preparation and modelling, it is necessary to note that this back and forth can occur after each phase. Moreover, the outer circle of the image symbolizes the repetitive nature of data mining projects, as the lessons learned from one project could lead to new, more focused applications. It is important to note that, as pictured, the data stands in the middle of each data mining project and is the most critical part of the process (Chapman et al. 1999, 9-15).

In Figure 4 an overview over each phase with their respective generic tasks and goals is given. In the following sections, each phase of CRISP-DM will be described alongside the task of predicting play types in the National Football League.

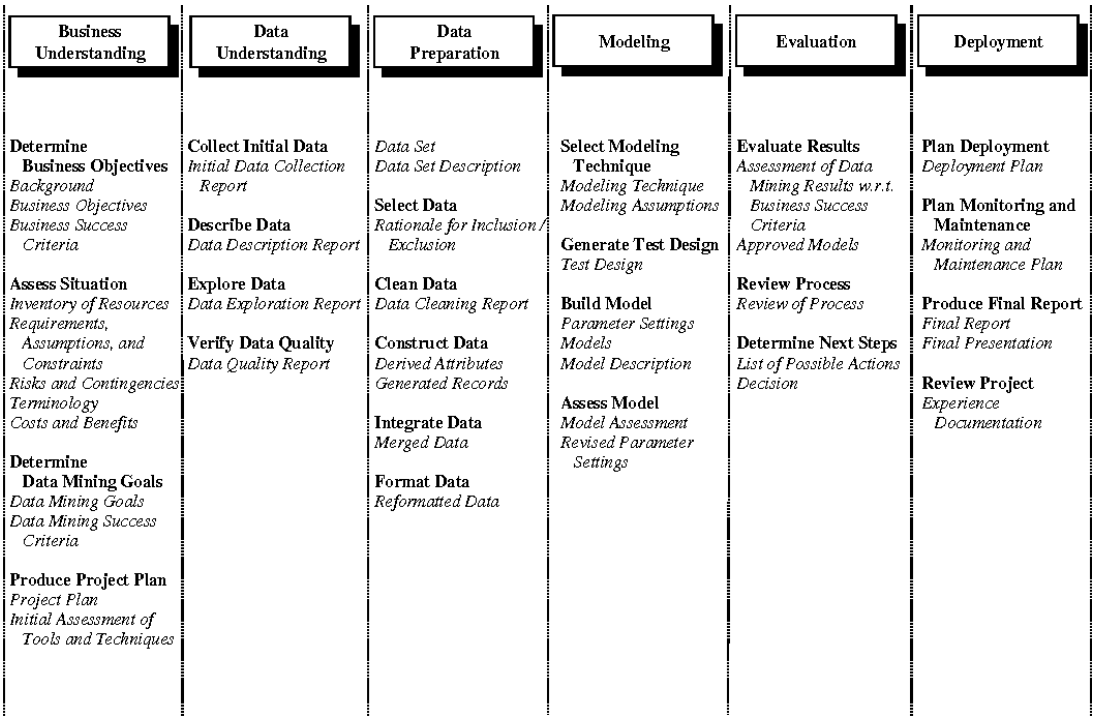


Figure 4: The phases and their respective generic tasks and goals of the CRISP-DM lifecycle. (Wirth and Hipp 2000, 6)

8 Development of the prediction model

As noted in section 1, the goal of this thesis is to predict the type of play of an American football team's offence as accurate and feasible as possible. The play-by-play data of NFL regular-season games from 2009 to 2019 is analyzed and used to create the best fitting machine learning model. The created model is further evaluated in subsection 8.5. The study is conducted roughly following the CRISP-DM process method and tries to improve the results over multiple iterations. Note that the CRISP-DM process model is designed for a commercial context and therefore can't be adhered to entirely as this study is conducted in a purely academic context. The development of the model and all following code examples are written in Jupyter Notebooks and IPython with Python 3 using sci-kit learn for machine learning, pandas for data manipulation, and matplotlib as well as seaborn for data visualization. The initially used dataset is scraped with the R-package nflscrapR from NFL.com the official website of the National Football League and is therefore considered to be a reliable source. The dataset is created by Ronald Yurko, who is also a contributor for nflscrapR, and is publicly available on github.com.[12]

12. `https://github.com/ryurko/nflscrapR-data`

8.1 Business understanding

The business understanding phase of the CRISP-DM process is used to find and understand the objectives of the client business and determine the goals of this project (Chapman et al. 1999, 13). The generic tasks in the business understanding phase are to determine the business objectives, to assess the situation and find out detailed information about the environment of the project, to determine the business and data mining goals, and to create a project plan (Chapman et al. 1999, 16-19).

8.1.1 Business background

While the project is of purely academic nature, a hypothetical client business could be imagined as an American football team in the NFL and other regional and international leagues. As noted in section 4, the NFL stands as the sports league with the highest revenue but especially American football leagues in Europe are mostly amateur leagues and have very limited financial capabilities. Furthermore, due to the limited organization and large differences in European leagues, the collection of structured data of the opposing team is less feasible as in the NFL. Therefore, the client business is assumed to be a team in the National Football League, but it is noted that the feasibility in amateur sports in Europe is a factor. Additionally, the data amount and quality available to NFL teams would be significantly higher and automatically collectable as opposed to the publicly available data used in this thesis.

8.1.2 Business objectives and success criteria

The goal of this project is to assist a defensive coordinator or defensive play-caller with a prediction on the offensive play. The gained information can further be used to assess the situation and decide on further actions. The following assumptions are taken from the conversation with Max Sommer. It is not necessary to predict the exact outcome of the play, but much more the single probabilities of certain outcomes. Sommer also said that all certain probabilities are useful for a defensive play call. For example, the knowledge that there is a 50/50 chance of a pass play versus a run play is a bad result in a machine learning context, but useful information in a football context, as the coach knows that the outcome is unclear and can adjust their defence accordingly. This further shows that it is important to create a model that can predict probabilities for each class, and not just a single class output.

8.1.3 Data mining goals and success criteria

The goals of this project in a data mining context are to further improve the results of previous projects in the field and to primarily create a better prediction model as a simple baseline assumption. The baseline algorithm is to call the most likely outcome, which is a pass on the binary classification, a run on the three-class classification and a short pass on the four-class classification. This results in an accuracy of 58.5% for binary classification, 43.1% for ternary

classification and 37.6% for quaternary classification. These values are simply the percentage of plays for each classification. The goal is to improve as much as possible above these values and therefore create a model closest to being classified as football understanding.

8.2 Data understanding

In the data understanding phase, the initial dataset is collected, and multiple techniques are applied to get familiar with the data and to find problems and qualities in the data. Furthermore, the goal is to detect first patterns and relationships (Chapman et al. 1999, 14). The generic tasks of this phase are to collect the initial data and create an initial data collection report, to create a data description report, in which the format and quantity of the data are described. Additionally, a data exploration report is created, in which distributions of key attributes and relationships between attributes are explored. This can be done using data visualizations, and interesting data characteristics and subsets are searched. Lastly, a data quality report is created. The data quality report describes found issues and how or if they are solvable (Chapman et al. 1999, 20-22).

8.2.1 Initial data collection report

As mentioned in section 8 the initial data for this project is collected from the official website of the National Football League by Ronald Yurko, who published the data on the version control platform `www.github.com`. This data was further acquired in the form of single CSV files for each season. The data is available for every game from 2009 to 2019 for each preseason, regular season and postseason. As the playcalling in preseason and postseason may differ strongly from conventional playcalling, only the data of the regular season is used in this thesis. The datasets are organized in the `datasets` folder under `play-by-play-data` and `regular-season` with each named as `reg_pbp_<full year>.csv`.

The acquired tables each consist of 256 columns including several probability measures and estimations, as well as different significant players for each play. These columns will not be discussed further, as they are not relevant to the outcome of this thesis. After filtering out irrelevant information there are 37 remaining columns.

8.2.2 Data description report

Each of the eleven datasets has between 44,596 rows for the 2009 season and 46,129 rows for the 2015 season. This results in 498,393 total plays over 2816 games in the eleven-year period. The datasets include every play, including kickoffs, field goals, extra points, timeouts and several other events that are irrelevant for this use case. Therefore, all the plays that are not classified as either a run play or a pass play are not included in the ongoing process. After this elimination 353,095 usable plays are left.

Table 2 shows all selected available parameters with their respective datatypes and possible categories for categorical variables. Furthermore, the table shows the number of missing values

Feature	Datatype	Null values
*game id**	int64	0
*play id**	int64	0
*home team**	team abbreviation	0
*away team**	team abbreviation	0
*posessing team**	team abbreviation	0
*posessing team type**	{'home', 'away'}	0
*yardline 100**	float64	0
*game date**	datetime	0
*half seconds remaining**	float64	0
*quarter seconds remaining**	float64	0
*game seconds remaining**	float64	0
*drive**	int64	0
*down**	float64	0
*goal to go**	boolean	0
*yards to go**	int64	0
*defteam timeouts remaining**	float64	0
*posteam timeouts remaining**	float64	0
*posteam score**	float64	0
*defteam score**	float64	0
*score differential**	float64	0
play type	{'run', 'pass'}	0
pass length	{'short', 'deep'}	160174
pass location	{'left', 'right', 'middle'}	160174
run location	{'left', 'right', 'middle'}	207916
run gap	{'end', 'guard', 'tackle'}	248128
yards gained	float64	210
air yards	float64	159876
yards after catch	float64	232024
incomplete pass	boolean	0
sack	boolean	0
interception	boolean	0
fumble	boolean	0
penalty	boolean	0
touchdown	boolean	0
pass touchdown	boolean	0
rush touchdown	boolean	0
desc	Textual description of play	0

Table 2: Preselected parameters from the initial dataset with each type and number of null values. Columns that are available before the outcome of the play is known are marked with *.

for each column. Seven of the 32 columns contain missing data many resulting from not applicable columns for certain play types. For example, the run gap and run location values can't be present for pass plays.

8.2.3 Data exploration report

The whole dataset consists of 206,594 passing plays and 146,501 running plays, which results in 58.51% of the total plays being passes. FigureFigure 5 shows how the percentage of play types changed over time, with a minimum of 56.8% in 2009 nearly constantly growing to a maximum of 59.9% in 2016. This shows that there are clear trends in offensive play calling and it could be an improvement to limit the selection of features to only include the past one or two years. This hypothesis is further explored in subsection 8.4.

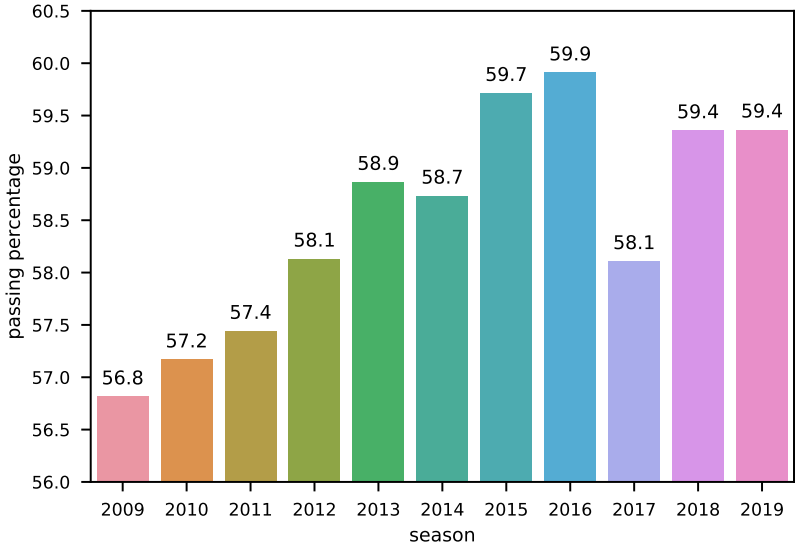


Figure 5: Percentage of passing plays per season.

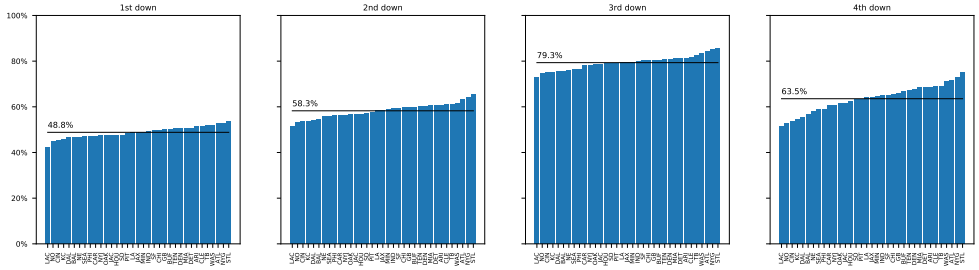
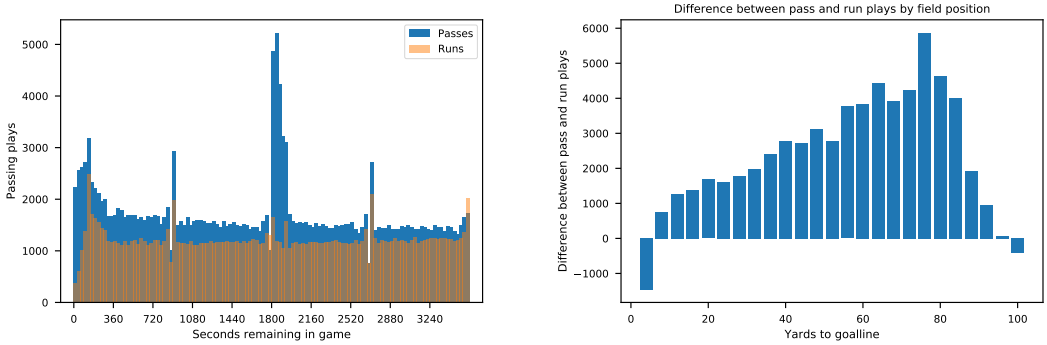


Figure 6: Passing percentage per team and down.

Besides the trends to more passing over the years, Figure 6 shows that the number of down is a significant factor in play-calling, and also fluctuates strongly across the different teams, especially on fourth down attempts. Nonetheless, this could be justified by the small amount (around 1.5%) of fourth-down attempts in the dataset. Figure 6 further shows that a passing play is much more probable to happen on a third-down than on a first down, with nearly 80% average likelihood on a third down in contrast to 48.8% on a first down.

As pictured in figure 7a, the amount of passing plays also increases drastically at the end of each quarter and especially before at the end of the second and fourth quarter. This is most certainly not the case in the first and third quarter, because the possession of the ball automatically changes after the second quarter, but not after the first and third quarter.



(a) Passing amount by time remaining in the game. (b) Histogram of the difference between passes and runs by field position.

Figure 7: Passing amounts in relation to field position and time remaining in the game.

Figure 7b shows another relevant connection between passing percentage and field position. It shows the difference between passing and running plays respective to the position on the field. While passing plays are much more common throughout the field, it is more popular to run the ball close to the goal lines.

Figure 8 shows how each play type is distributed in relation to the distance to a new first down for each down. The violin plots show how for third and fourth downs run plays are often only called when only a short distance is needed for a need first down, but passing plays are distributed much more equally. On second downs, the distribution is more similar, but it is visible that for short distances running plays are peaking while passing plays decline. The data for first downs is omitted as first downs nearly always have ten yards to go and the distribution of the play calling is nearly equal.

8.2.4 Data quality report

As noted in subsubsection 8.2.2 many features of the initial table contain missing data. This can often be justified by football rules. In this section, each feature with missing values will be discussed and reasoned.

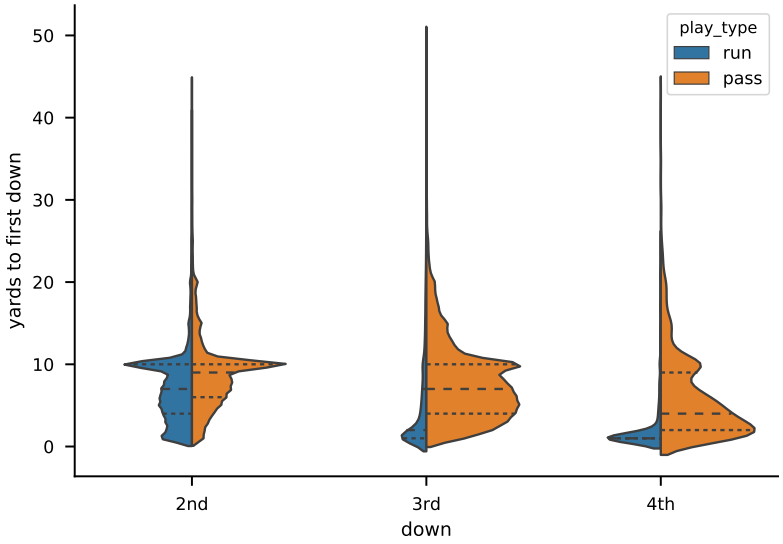


Figure 8: Distribution of pass and run plays in relation to down and distance to first down.

Yards gained The `yards_gained` column describes the total yards that are won for each play. There are 210 null values in the table, which only contain plays from two games in the 2013 season, one between the Cleveland Browns and the Jacksonville Jaguars, and one between the Cleveland Browns and the Pittsburgh Steelers. This is assumed to be an error in the collection of data and, considering the size of the dataset and the relevance of the parameter this is not viewed as a important issue.

Pass length and pass location Both `pass_length` and `pass_location` have null values for the same 160,174 plays, and therefore are estimated to have the same issues. The dataset includes 146,501 run plays which can't include the described parameters. Nonetheless, there are four run plays which contain information about pass length and pass location. Two of them are trick plays which can be classified as either pass or run, and two are misclassified passing plays according to the `desc` column. Furthermore, the dataset includes 4,769 interceptions and 13,147 sacks. While not all sacks are considered to be a pass, it makes sense that there is no available data for both columns as the quarterback isn't able to throw the ball. While the length and location of an intercepted throw could have been measured, it is not available in the dataset. After counting off both sacks and interceptions there are 461 missing values left for each column. As these values, especially the pass length, are essential features, the columns with missing values are substituted or omitted in subsection 8.3.

Air yards While the previously discussed column `pass_length` only gives a categorical overview of the passes, the `air_yards` column gives an exact amount of yards that the ball is thrown before being caught. Similar to the `pass_length` column this feature is not available

for runs, sacks, and interceptions. This value can be used to derive each `pass_length` value, but there are 70 rows which do not include either. This is not considered to be a significant flaw.

Run location The `run_location` parameter gives information if the ball carrier runs left, right or in the middle. There are 207,916 null values, which are mostly passing plays and can't be classified. After counting off the passes there are 1,322 null values left. This includes 1,098 fumbles, where the running back loses the ball and doesn't carry on running. 33 of the remaining 108 plays are quarterback scrambles that are not classifiable. 75 plays still have no run location value, this might result from missing or not classifiable plays.

Run gap The `run_gap` column describes the gap in the offensive line through which the running back tries to run. After the subtraction of pass plays, quarterback kneels and scrambles, and fumbles, 37,579 plays are missing run gap information. 37,498 of the remaining plays have a `run_location` value of `"middle"` and it is assumed that these plays are runs through the middle without a clear gap. This information has to be paid attention to in subsection 8.3.

8.3 Data preparation and feature extraction

During the data preparation phase, all activities are performed to create the final dataset that is afterwards fed into the modelling tools (Chapman et al. 1999, 14). This phase includes the selection of data according to the relevance to the data mining goal, the cleaning of the data and the creation of a data cleaning report, the construction of derived and generated features, and the integration and merging of multiple data sources. Finally, the data is formatted to meet all of the requirements of the learning model, for example, ordering the rows or columns without changing the values (Chapman et al. 1999, 23-26).

8.3.1 Feature selection

The necessary data was previously selected in subsubsection 8.2.1 on a rough estimate of relevance influenced by the information gained in subsection 6.2. The selection was further adjusted throughout the data understanding phase. To further prioritize the data and get a professional opinion on the feature relevance, Max Sommer was asked to rate several selected features according to their relevance on play calling. The results can be seen in Table 3. This information was collected through an online form designed as a multiple-choice grid. The features were given by the author. Max Sommer rates the position on the playing field, individual team strengths, down and distance and the time remaining in the current half as the five most relevant features. While the individual team strengths can't be explored in the data understanding phase because of missing player data, a relationship of all other described features with the target features can be seen. Defensive personnel grouping is also ranked very high and could be a strong indicator of offensive play-calling. Personnel grouping information is not available publicly but might be available to NFL teams.

Feature	**Importance** *1 (Irrelevant) – 5 (Crucial)*
Position on field	5
Individual team strengths	5
Current down	5
Distance to first down	5
Time remaining in half	5
Time remaining in game	4
Score Differential	4
Defensive personnel grouping	4
Time remaining in quarter	3
Offense timeouts left	3
Weather (only rain)	3
Yards gained on previous play	2
Time of season	2
Defense timeouts left	1
Previous pass was complete	1
Home or Away	1

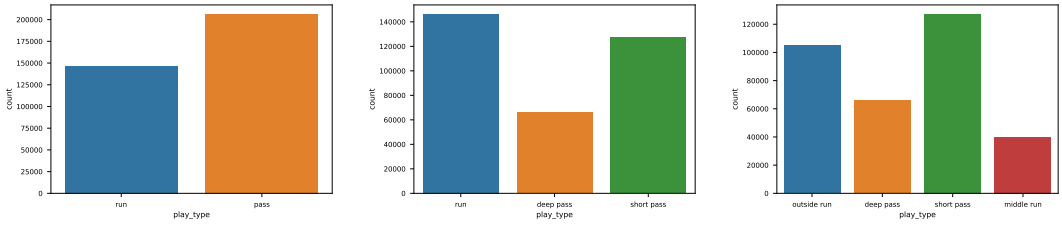
Table 3: The importance of preselected parameters for offense play calling according to Coach Max Sommer.

As seen in Table 2 only 20 of the available 37 columns can be collected prior to the play's execution and therefore being available as descriptive features in this use case. The remaining columns are eligible as target features, but as this thesis is about the prediction of play types, only `play_type`, `pass_length`, `pass_location`, `run_location`, and `run_gap` are seen as possible target features. Furthermore, the game and play IDs, as well as home and away team names and the date of the game are not seen as relevant information in the prediction process and are therefore stripped from the dataset for the modelling process. This leaves 14 features as possibly relevant as descriptive features and several selections are tried out in the modelling phase.

The target feature is constructed from the five categorical values before and will be tested on different precisions. Primarily, the prediction between pass and run plays is examined and secondly, the differentiation between deep and short passes and different run locations is investigated.

8.3.2 Data cleaning report

In this section, the issues of the dataset explored in subsubsection 8.2.4 are addressed and fixed. As there are no known issues in the descriptive features, no error handling has to be made. Whatsoever, the data is transformed to fit the needs of learning algorithms. The only categorical value in this set of features is the `posteam_type` column which can either be `home` or `away`. This feature is encoded with `home` as 1 and `"away"` as 0 using the LabelEncoder of the sklearn



(a) Two target categories (b) Three target categories (c) Four target categories

Figure 9: The distribution of each target category on three different subsets.

library for python:

```
1 from sklearn.preprocessing import LabelEncoder
2 posteam_type_encoder = LabelEncoder()
3 df['posteam_type'] = posteam_type_encoder.fit_transform(df.posteam_type)
```

Listing 1: Transforming categorical data to numeric values using a LabelEncoder.

The code assumes the data frame to be imported as `df` and the data exploration library pandas as `pd`.

For the target features, one data frame is created in which only the `play_type` is encoded with `pass` as 1 and `run` as 0. This is done in the same way as with the `posteam_type` column. Another data frame is created with a derived feature creating the categories `deep pass`, `short pass`, and `run`, to create an additional level of precision. These values are further encoded as 0, 1 and 2. A third data frame is created that further encodes run location as `middle run`, and `outside run` with both `left` and `right` of the `run_location` column being considered as `outside`. The resulting four categories are then encoded as integer values between 0 and 3.

To create the data frame with three distinctive `play_type` values, the column is calculated according to the `air_yards` values with plays with less than 10 air yards being classified as `short pass` and longer passes classified as `deep pass`. The rows where `air_yards` is unavailable are dropped. The `pass_length` column is not used as there are less available values and it is less flexible.

To create the distinction between inside and outside runs for the third data frame, the function to create the target features is extended to distinguish according to the `run_location` value. The rows with no available data are dropped, as they are mostly not classifiable.

The distribution of target features in each data frame is shown in Figure 9. All three datasets are further saved as `.csv` files for the modelling process.

8.3.3 Derived attributes

To display the teams' individual tactics one derived attribute is created. The column `team_down_passing_percentage` is added to the data frame and calculated as each

team's passing percentage per down. The exact implementation is displayed below and assumes the data frame to be available as a pandas data frame named `df`.

```
pivot = df[["down", "posteam", "play_type"]].pivot_table(index="down",
    columns=["posteam", "play_type"], aggfunc=len)
def get_down_passing_percentage(row):
    passes = pivot.loc[row.down][row.posteam, "pass"]
    runs = pivot.loc[row.down][row.posteam, "run"]
    return passes / (runs + passes)
df['team_down_passing_percentage'] = df.apply(get_down_passing_percentage,
    axis=1)
```

Listing 2: Calculating the possessing team's passing percentage for each down using a pivot table.

8.4 Modeling

Throughout the modelling phase, several modelling techniques are tried out, and the features are trimmed to the optimal values. To adjust to the different techniques, it is often necessary to step back and forth between the modelling phase and the data preparation phase (Chapman et al. 1999, 14). The generic tasks in this phase are the selection of one or multiple machine learning algorithms and techniques, the generation of a test design, which includes the division into training, test, and validation datasets. Afterwards, the models are built, which includes the discovery of optimal parameters for each model. The output of the building process is a model description for each model. The model is assessed after the training, and its parameters are tweaked to improve each model further until the best possible models are found. The assessment in the modelling phase actively integrates with the following evaluation phase (Chapman et al. 1999, 27-29).

8.4.1 Candidate models

As the problem in this classification is of multinomial nature, meaning that there could be more than two classes to output, the algorithms are chosen to be suitable for this task. As most multiclass algorithms are also suitable for binary classification, these are suitable for all three datasets. To create an overview of multiple models and to find the most fitting for this task, three candidate models are trained and tested. First of all, a logistic regression model is tested. As linear models are a much simpler and less expensive approach than most other models and might be suitable without the greater complexity of other models. To train a simpler model can be used with less computational power and could, therefore, be less expensive and more feasible in a hypothetical use case, for example, on a tablet on the playing field. Furthermore, a support vector machine is trained. Non-linear support vector classifications are prone to be time-consuming on large datasets and it is therefore used only on a limited single year subset of the whole dataset. Finally, a multilayer perceptron with backpropagation is used to take a look at a state of the art, and data-expensive approach. As neuronal networks need very large datasets, this can be used with a large period of time and with data of multiple seasons.

All three models are implemented in the scikit-learn library.

8.4.2 Test design

To mimic the real-world use case, the whole dataset is divided into training and testing subsets according to time. As the model would be trained with the past plays of a team and then be evaluated and tested on the future plays, the whole dataset is divided into the previous 80% of plays for training and the chronologically last 20% for testing. This is done on different total time frames, as certain models need to have more data than others. There are three different divisions of data:

1. The whole dataset is divided into the years from 2010 to 2017 as the training set and the 2018 and 2019 seasons are used for evaluation.
2. Each season is divided into the chronologically first 80% of plays as the training set and the rest as the test set.
3. Ten pairs of two consecutive seasons are created and then divided in the same way, with the first 80% for training and the rest for testing.

Due to the different characteristics for each down the datasets are further divided into four set for each down. Four models are then trained independently for all downs.

Each classifier is tested for all three target values (binary, 3-way, and 4-way classification). The logistic regression model is tested on all eleven seasons and is primarily evaluated with a precision score. This score is calculated as $tp/(tp+fp)$ where tp are the true positives and fp are false positives. The best possible precision score is 1 and the worst possible is 0. The models are further evaluated in the evaluation phase.

8.4.3 Fitting and assessment of the models

In the beginning, the models were trained on the whole dataset, this resulted in a precision score of around 0.65 for the binary classification. While this not a very bad score, there is still much potential. After dividing the dataset to only address one down, and training the models for each down separately, the precision score increased drastically. Each of the three classifiers is tested and trained in a loop over the three different base tables, with the different target features. The logistic regression classifier is primarily trained separately on every season. The support vector classifier is primarily trained on the paired data of each two seasons. The multilayer perceptron is primarily trained on the whole dataset. Furthermore, each descriptive feature is scaled with a standard scaler to normalize all features and fit the needs of the algorithms. In this section, each of the three models is described and assessed.

Logistic Regession The logistic regression model is trained as follows:

```
1 def logistic_regression(train, test):
2     X, y, X_test, y_test = toXy(train, test)
3     logClf = LogisticRegression(multi_class="multinomial", solver="lbfgs")
4     logClf.fit(X,y)
5     prediction = logClf.predict(X_test)
6     return precision_score(y_test, prediction, average="micro")
```

Listing 3: Training and assessing the logistic regression model.

The train and test parameters of the function are the whole datasets, including the target feature. These parameters are divided as described in subsubsection 8.4.2. Continuing, the train and test sets are then further divided into the descriptive features X and the target feature y by the `toXy` function, which also applies the standard scaler described before. Then the Logistic Regression class is instantiated with the `multi_class` flag and the `lbfgs` solver. Both values are recommended by the scikit-learn documentation for multiclass classification.[13] The classifier is further fitted to the training data with the `fit` function and a prediction for the whole testing data is created with the `predict` function. The precision score is then calculated. The `average='micro'` flag is necessary for multiclass classification.

With all 16 previously described features, the logistic regression classifier reaches a precision score of up to 83.5% for the classification between run and pass plays with a mean across all seasons and down of 70.1%. As expected the model's precision decreases when multiple classes are introduced. The model to further predict deep and short passes maxes out at 67.3% with a mean of 53.7%. The third dataset with four classification classes has a maximum accuracy of 55.3% and a mean of 46.1%.

These values are further improved by gradually removing certain features. The best results were found after removing ten of the previous descriptive features and only using the distance to the goalline, the seconds remaining in the game, the yards to go for a first down, score differential, and the team's passing percentage for the current down. These five features are easily collectable during the game and most could be collected automatically. This therefore not only improves the accuracy of the model but also the feasibility of the data collection.

The precision scores with the limited amount of features are displayed in Table 4.

	Classification of type		
	binary	ternary	quaternary
min	0.591	0.480	0.313
max	0.857	0.655	0.576
mean	0.698	0.536	0.459

Table 4: Precision scores of the logistic regression model.

13. https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Support Vector Machine The support vector machine model is trained similarly to the logistic regression model.

```
def svm(train,test):
    X, y, X_test, y_test = toXy(train, test)
    svmClf = SVC()
    svmClf.fit(X,y)
    prediction = svmClf.predict(X_test)
    return precision_score(y_test, prediction, average="micro")
```

Listing 4: Training and assessing the support vector classifier.

As visible above the code is very similar and as a support vector machine is designed to fit multiclass classification the default values are used. The support vector machine model uses the same six descriptive features as the logistic regression model but reaches a slightly better accuracy in most values.

	Classification of type		
	binary	ternary	quaternary
min	0.599	0.513	0.388
max	0.879	0.686	0.581
mean	0.717	0.565	0.471

Table 5: Precision scores of the support vector classifier.

With the larger dataset, the support vector classifier reaches slightly higher minimum values, but lower maximum values, with the mean staying rather consistent. This means that the ten additional descriptive features make the model more consistent but decreases the maximum possible accuracy. Similarly, the model was first trained on the dataset with season pairs and accomplished a more constant but overall worse prediction.

Multilayer Perceptron The multilayer perceptron is by far the most complex classifier of the three. As it is often challenging to find the most fitting hyper-parameters, meaning the predefined parameters, for such a classifier, a grid search is conducted for the multilayer perceptron classifier. This makes it possible to automatically find the best fitting hyper-parameters from a predefined parameter grid for a given estimator/model.

```
def mlp(train,test):
    X, y, X_test, y_test = toXy(train,test)
    parameter_grid = {
        'hidden_layer_sizes': [(100,),(6,6,6),(12,12)],
        'random_state': [1,2,3],
        'solver': ["lbfgs", "adam"],
        'shuffle': [True, False]
    }
    mlpClf = MLPClassifier(max_iter=300)
    gridClf = GridSearchCV(estimator=mlpClf, param_grid=parameter_grid,
        n_jobs=-1, cv=5)
    print("start fitting...")
```

		Classification of type		
		binary	ternary	quaternary
down	1st	0.637	0.580	0.451
	2nd	0.687	0.524	0.471
	3rd	0.849	0.571	0.526
	4th	0.812	0.657	0.526

Table 6: Precision scores of the multilayer perceptron.

```
12      gridClf.fit(X,y)
13      print("fitted...")
14      print(gridClf.best_params_)
15      prediction = mlpClf.predict(X_test)
16      return precision_score(y_test, prediction, average="micro")
```

Listing 5: Using a GridSearchCV to find the best parameters out of a parameter grid for the MLPClassifier.

The code is similar to the two other models, but the classifier is wrapped inside a GridSearchCV class. This class is a helper function provided by scikit-learn. The `parameter_grid` defines the different parameters to try out by the grid search. The `n_jobs` flag of the GridSearchCV constructor notes that the search should be run on the maximum available threads of the machine and the `cv` parameter defines the number of cross-validations.[14]

The grid search was conducted two times with different parameters. The search with the parameters in the code example above resulted in optimal results as two hidden layers with 12 nodes each, a random state of 2, with shuffle set to True and the `"adam"` solver. After this iteration, the search was repeated with only differentiating hidden layer sizes. The proposed layer sizes were three layers with 12 nodes each, two layers with 24 nodes each or the previously found parameter of two layers with 12 nodes each. The search resulted in the same parameters as before and could find any additional improvements in the other sizes of the hidden layers.

The classifier was then trained without the usage of the GridSeachCV class and with the described parameters. As the data for the neuronal network was only divided into one training set and one test set, the assessment slightly defers from the other models. The results are displayed in Table 6.

8.5 Evaluation

While the assessment task in the modelling phase evaluates the model on the accuracy of the model, the evaluation phase answers if the model accurately achieves all the business objectives and decides if further iterations of the CRISP-DM process are needed. At the end of this phase, it is decided if results can be deployed and integrated into the business (Chapman et al. 1999,

14. `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html`

14). The necessary tasks in this phase are the assessment of the models against the business success criteria, the review of the process, in which quality assurance issued are covered, and the determination of future steps and possible actions and deployment possibilities (Chapman et al. 1999, 30-31).

8.5.1 Model assessment and evaluation

The models were previously tested with a prediction accuracy score which is the percentage of true positives. The maximum accuracy was accomplished with the support vector machine in the binary classification scenario on fourth downs on a single season. The reached score of 87.9% is higher than any other season wide score reached by the projects presented in subsection 6.2. To reach this level of accuracy, with only six, easy collectable, descriptive features can be described as a great success.

As expected the accuracy decreased significantly with the introduction of new output classes. The ternary classification, in which each pass is additionally classified as deep or short the accuracy shrank by nearly 20% to 68.6%. While this is not nearly as accurate as of the binary classification, it is much more reliable than the baseline algorithm introduced in subsubsection 8.1.3, which achieved an accuracy of 43.1%. This is an improvement of 25.5%.

The quaternary classification reached a maximum of 58.1% which is considered to be a success as it is a significant improvement to the 37.6% achieved by the baseline algorithm.

As noted in subsection 4.1, accuracy is not the most important factor in the creation of a classifier to help a defensive play-caller. It is more important to display the probabilities of each class to assist the defensive coordinator. This can be done on each trained model with the `predict_proba` method on the classifier instance. The method takes a query instance as input and outputs the probability of each class. This could easily be implemented into an interface and could realistically help American football teams with their playcalling.

Interestingly, the support vector classifier created the best predictions in nearly all cases, with the neural network classifier often making worse predictions than the logistic regression model.

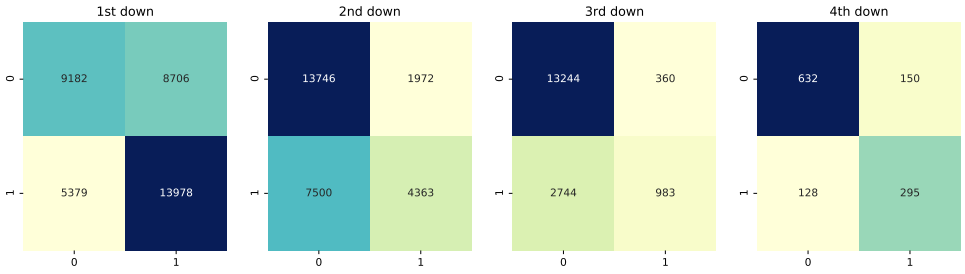


Figure 10: Sum of all confusion matrices of the support vector classifier on the binary classification scenario, divided by downs. The label 0 stands for a pass, the label 1 stands for a run.

To further evaluate the results in this thesis Figure 10 shows the confusion matrices for the support vector classifier on each down. The confusion matrices are summed up over all 11 seasons. The x-axis shows the predicted label and the y-axis the actual label, with zero being a pass and one being a run. It shows that, for example, 8,706 pass plays are predicted to be a run play on the first down scenario. Except that it is clearly visible that pass play could be classified correctly most of the time but run plays were often misinterpreted. The main diagonal of the confusion matrix shows the true positives while all other cells correspond to false positives.

As discussed before, all prediction models were able to fulfill the data mining goals and improved severely against the baseline algorithm. Furthermore, the business objectives were considered and all goals where achieved, primarily the acurate prediction of probabilities for each class.

8.6 Deployment

While the creation of a model is an essential step in data mining, it is generally not the end of the project, because the gained information is usually not readable by a company or customer. Therefore it is vital to create a usable or readable application of the data. This can, for example, be achieved by the integration of an online learning model into the decision making processes of a company or the creation of a report (Chapman et al. 1999, 14). In this phase, a deployment plan and a maintenance and monitoring plan are created, as well as a final report and if necessary a final presentation. Finally, experience documentation is written, in which essential takeaways about the project workflow and misleading approaches or hints are described (Chapman et al. 1999, 32-34).

8.6.1 Deployment of the play type prediction model

As this project is purely academic the deployment is limited to this thesis. section 8 can be seen as a final report with a detailed evaluation in subsection 8.5 and section 9. The created models are not published or distributed. The additional tasks of the deployment phase of the CRISP-DM process are left out due to the missing professional context of this project and would go beyond the scope of this thesis.

9 Discussion

The accuracy results of all trained models reached sufficiently high values to give an estimate of what type of play an offensive football team will execute. A wide variety of models, training sets and feature vectors are tried out to create the best-fitted models possible with the resources of this project. The support vector classifier accomplished better results than a multilayer perceptron and a logistic regression classifier. Given the computational complexity of a support vector machine, it would be possible to translate the models to a real-world application that runs on devices that easily can be used on the field and during competition.

Separating the datasets and creating unrelated models for each down was found to be the most effective way of improvement. Furthermore, the best results were achieved by only using a subset of five descriptive features:

1. distance to the goal line
2. seconds remaining in the game
3. yards to go for a first down
4. score differential
5. possessing team's passing percentage for the current down

Note that the separation already encodes the information about the current down into multiple models.

The logistic regression classifier and the support vector classifier improved on smaller datasets with only one season of data. This might be due to overfitting, as the tactics of American football teams might change significantly over two seasons. The reason for this might especially be the case due to the staff changes that are made between seasons, and different coaching staff might call plays differently. Generally, the models improved better with less but more meaningful data. This indicates that the models were previously overfitted.

The support vector classified trained in this thesis outperforms the theses described in subsection 6.2 in multiple factors.

The question if a play type prediction model can be created well enough to assist in the play calling of American football coaches can be answered with yes. This is especially the case, due to the possibility to give single probability scores for each class. The extension of previous related theses with multiclass prediction models also decreased the accuracy but could improve significantly against the baseline algorithm and could give essential hints for defensive coaching personnel.

Future research could improve their results with the further development of descriptive features and better representing each team's strengths. Also, this thesis ignored personnel groupings on the field which, according to Max Sommer, is a crucial factor in the play-calling process. This information was left out, as it is not publicly available by now but would be available for NFL teams.

10 Conclusion

Concluding, in this thesis, several approaches to the prediction of play types in American football were taken to find the most reasonable application for the task. Through the gathering of strong domain knowledge in the form of a conversation with American football coaching consultant Max Sommer, the most relevant features were obtained, and consistent results could be

achieved with only five input parameters. Out of a logistic regression model, a support vector machine and a neural network, the support vector machine generated the best results with a maximum prediction score of 87.9%. Compared to recent projects in the field, this is considered to be one of the best results. Furthermore, the extent of a possible prediction was explored. A binary classification between pass and run plays is found to be possible to predict with an available amount of data and can be a useful aid for defensive coordinators in American football. Moreover, a ternary and quaternary classification model was furthermore trained, which yielded results that could give a coach important information on the depth of a pass and the location of a run, but with much less confidence than the binary classification.

References

Alpaydın, Ethem. 2010. *Introduction to Machine Learning*. 2nd ed. Cambridge, Massachusetts: The MIT Press. ISBN: 9780262012430.

Assuncaõ, Renato, and Konstantinos Pelechrinis. 2018. "Sports Analytics in the Era of Big Data: Moving Toward the Next frontier." *Big Data* 6 (4): 237–238. doi:10.1089/big.2018.29028.edi.

Bishop, Christopher M. 2007. *Pattern Recognition and Machine Learning*. Edited by Michael Jordan, Jon Kleinberg, and Bernhard Schölkopf. New York, NY, USA: Springer. ISBN: 0-387-31073-8.

Braunwart, Bob, and Bob Carroll. 1997. "Camp and his Followers." In *The Journey to Camp: The Origins of American Football to 1889*. Grand Island, NY, USA: PFRA Books.

Chapman, Pete, Julian Clinton, Randy Kerber, Thomas Khabaza, Thomas Reinartz, Colin Shearer, and Rüdiger Wirth. 1999. *Step-by-step data mining guide*. Technical report. http://www.crisp-dm.org/CRISPWP-0800.pdf.

Fernandes, Craig Joash, Ronen Yakubov, Yuze Li, Amrit Kumar Prasad, and Timothy C.Y. Chan. 2020. "Predicting plays in the National Football League." *Journal of Sports Analytics* 6:35–43. doi:10.3233/jsa-190348.

Fry, Michael J, and Jeffrey W Ohlmann. 2012. "Introduction to the Special Issue on Analytics in Sports, Part I: General Sports Applications." *INFORMS Journal on Applied Analytics* (Maryland, USA) 42 (2): 105–108. doi:10.1287/inte.1120.0633.

Jordan, Jeremy D., Sharif H. Melouk, and Marcus B. Perry. 2009. "Optimizing Football Game Play Calling." *Journal of Quantitative Analysis in Sports* 5 (2). doi:10.2202/1559-0410.1176.

Kelleher, John D., Brian Mac Namee, and Aoife D' Arcy. 2015. *Fundamentals of Machine Learning for Predictive Data Analytics*. Cambridge, Massachusetts: The MIT Press. ISBN: 978-0-262-02944-5.

Lee, Namhoon, and Kris M. Kitani. 2016. "Predicting Wide Receiver Trajectories in American Football." In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 1–9. Lake Placid, NY, USA: IEEE. doi:10.1109/WACV.2016.7477732.

McGarrity, Joseph P., and Brian Linnen. 2010. "Pass or Run: An Empirical Test of the Matching Pennies Game Using Data from the National Football League." *Southern Economic Journal* 76 (3): 791–810. doi:10.4284/sej.2010.76.3.791.

Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. *Foundations of Machine Learning*. 2nd ed. Cambridge, Massachusetts: The MIT Press. ISBN: 9780262039406.

Morgulev, Elia, Ofer H. Azar, and Ronnie Lidor. 2018. "Sports analytics and the big-data era." *International Journal of Data Science and Analytics* 5 (4): 213–222. doi:10.1007/s41060-017-0093-7.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. 1st ed. Cambridge, Massachusetts: The MIT Press. ISBN: 9780262018029.

Nistala, Akhil, and John Guttag. 2019. "Using Deep Learning to Understand Patterns of Player Movement in the NBA." In *MIT Sloan Analytics Conference*, 1–14. Boston, MA, USA: Massachusetts Institute of Technology.

Nyce, Charles. 2007. *Predictive Analytics White Paper*. Malvern, PA, USA.

Orchard, John W., Hugh Seward, and Jessica J. Orchard. 2013. "Results of 2 decades of injury surveillance and public release of data in the Australian Football League." *The American Journal of Sports Medicine* 41 (4): 734–741. doi:10.1177/0363546513476270.

Ota, Karson L. 2017. "Football Play Type Prediction and Tendency Analysis." Master Thesis, Massachusetts Institute of Technology.

Reep, C., and B. Benjamin. 1968. "Skill and Chance in Association Football." *Journal of the Royal Statistical Society* 131 (4): 581–585.

Rottenberg, Simon. 1956. "The Baseball Players' Labor Market." *The Journal of Political Economy* 64 (3): 242–258. doi:10.1017/CBO9781107415324.004.

Stefani, Raymond T. 1987. "Applications of statistical methods to American football." *Journal of Applied Statistics* 14 (1): 61–73. doi:10.1080/02664768700000006.

Wasserman, Erin B., Mackenzie M. Herzog, Christy L. Collins, Sarah N. Morris, and Stephen W. Marshall. 2018. "Fundamentals of Sports Analytics." *Clinics in Sports Medicine* 37 (3): 387–400. doi:10.1016/j.csm.2018.03.007.

Wirth, Rüdiger, and Jochen Hipp. 2000. "CRISP-DM : Towards a Standard Process Model for Data Mining." In *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 29–39. New York, NY, USA: AAAI. doi:10.1.1.198.5133.

Appendices

A git-Repository

Link to the repository on the MMT-git-Server `gitlab.mediacube.at`:
`https://gitlab.mediacube.at/fhs41321/noldin_niklas_ba2`

B Templates for study material

B.1 Interview transcript

This is the complete transcript of the interview held between the author of this thesis, Niklas Noldin, and head coach of the Austria football team Max Sommer. It was held in german and is therefore displayed in the original language. Many important take aways are cited in the thesis and translated accordingly. Additional notes are displayed in bold.

The project was explained to Max Sommer before the interview began. The start is his reaction to this introduction.

Max Sommer: Ist es wichtig, dass es die NFL ist oder können es auch andere Mannschaften sein?

NN: *Der verwendete Datensatz ist von der NFL, weil es dort am meisten Daten gibt. Ich weiß aber nicht ob in Österreich Play-by-Play Daten aufgenommen werden.*

MS: Ich schätze fast alle Bundesliga Teams machen jedes Spiel vollen Dateneintrag, Play-by-Play, mit vielen Parametern. Von ”Down and Distance”, Feldposition, welche Hash und was der Ausgang ist. Pro Spielzug gibt es sicher immer 10 bis 12 eingetragene Daten. Also, das ist auf Excel basiert und in Hudl gespeichert.

[...]

NN: *Dann hätte ich die erste Frage: Was werden generell für Technologien im Football verwendet. Vor allem mit Bezug auf Daten und Playcalling.*

MS: Vor circa 10 bis 15 Jahren haben 3 Studenten von Harvard eine Plattform namens Hudl erfunden. Der große Unterschied zu den anderen Video- und Football-Plattformen ist, dass sie das damals Online- und Server-based gemacht haben. Also nicht ”on-premise” sondern Cloud-basiert. Das hat einen irren Einfluss auf das Ganze gehabt. Weil früher ist das einfach auf einem Desktop gelaufen und wer dort gesessen ist hat damit gearbeitet, und ”that’s it”. Und auf einmal hast du dein ganzes Team, alle Trainer, alle Assistenztrainer in das System hineinlassen können. Dementsprechend reguros hast du deine Daten ”reinstopfen” können. Und wie das funktioniert ist eine Play-by-Play Video-Datenbank, Clip eins bis 150, und zu jedem Clip hast du eine Zeile an wirklich x-beliebigen, also auch selbst manipulierbaren, Daten zu jedem Spielzug. Und dann

hat Hudl eingebaute Analyse-Tools, bei denen du auf Knopfdruck, die Analysen, wie ”Down-and-Distance”-Report, da gibt es ein paar standard Coaching-Fragen, und, was ein Bisschen in deine Richtung geht, haben Sie etwas das ”What’s Next” heist. Wenn du genug Daten hast und du den drüber lasst, versuchen sie dir Tendenzen zu geben. Zum Beispiel, kommen da Sachen raus wie ”Nach jedem Lauf unter zwei Yards: Erwarte einen Pass”. Wenns halt zufällig so ist. Das geht schon in deine Richtung. Video ist immer die Kerninformation und von Video tut man dann mit der Hand die Daten ”einhacken”, die man in den Videos sieht. Es gibt aber auch schon, und das ist auch interessant, Videosoftware die selbst Grunddaten eingibt. Zum Beispiel erkennt wo man am Feld ist. Ein Feld ist linear aufgebaut und die Software erkennt ob man auf der linken oder rechten Hash ist.

NN: *Das funktioniert auch?*

MS: Das funktioniert gut. Zum Beispiel auch einen Snap-Detektor. Der checkt, dass es ein Line-up ist, alles ist ruhig und markiert im Video den Snap, weil dort bewegt sich etwas. Das sind relativ simple Sachen, die aber super sind, weil wenn du einen 12-jährigen Kamerabuben hast, der einfach gern viel zu viel filmt, ist das schon recht praktisch. Dieses Tool das wir da verwendet haben, was auch weltweit das ist, ist Hudl. [...] Die haben das auch übergespielt auf Golf, auf Basketball, auf Tennis, auf Fußball mittlerweile. Haben auch die größte Fußball-Video-Analyse, was noch viel schwieriger ist, weil es noch viel fließender ist und nicht so konzipiert wie Football. [...] Eigentlich ist dieses Hudl das ”All-Inclusive-Non-Stop-Shop”. Natürlich gibt es ein paar Konkurrenten, die teurer oder billiger sind, die aber etwas Ähnliches oder das Gleiche machen.

NN: *Gibt es etwas das während dem Spiel am Feld verwendet wird?*

MS: Ja, das dürfen wir seit circa fünf Jahren: Oben filmen und unten anschauen. Früher, nach NCAA (College) Regeln, nach denen wir in Europa spielen, waren ”electronic devices” nicht erlaubt.

[...] due to technical difficulties there is no transcript available for several minutes.

MS: Das ist die Dynamik die Playcalling hat, aber 70% ist schon eine sehr gute Aussage. Und 50/50 ist auch eine gute Aussage. Das muss man auch verstehen. Das Bewusstsein jedes Verhältnisses ist super.

NN: *Alles klar. Wenn ich weiß, dass es 50/50 ist, dann...*

MS: Ist auch gut, ja. Vor allem wenn man blind ist. Wenn man gegen einen Gegner spielt, bei dem man vorher kein Video gehabt hat. Was im Internationalen Football oft der Fall ist, weil die zu ”deppert” sind ein Video zu schicken. Dann musst du das echt schon beim Warmup anfangen zu schaffen. So blöd das kling: Was übt die andere Mannschaft im Warmup? Und daraus... Das ist natürlich der schlechteste Datensatz.

NN: *Nationalteams sind (in Sachen Daten der Gegner), wahrscheinlich am wenigsten gut bedient, oder?*

MS: Das ist furchtbar, ja. Der einzige Vorteil beim Nationalteam ist... Das Repertoire der Playbooks der Nationalteams ist kleiner, weil die weniger Zeit zum vorbereiten haben. Jetzt ist aus dem sie schöpfen können kleiner. Unter anderem ist deshalb Österreich so gut: Weil wir

eine hohe Spielintelligenz in unseren Vereinen haben können wir sehr viel im Nationalteam an verschiedenen Spielzügen hineinbringen und können über Nacht, im Turnier, unser Playbook ändern. Das können andere nicht. Aber natürlich ist das Datenbasierte sehr schlecht. Sehr interessant wäre auch, welcher Spieler wird angespielt. Im Sinne der Jersey-Nummer. Das ist in Hudl auch drinnen. Sozusagen "ball from number 12 to number 80" und auch da gibt es dann Pattern um zu sagen: Bei dritter und 7+ bekommt Nummer 80 80% der Pässe. Oder wenn Nummer 25 am Feld ist, ist es ein Laufspielzug und wenn Nummer 27 am Feld ist, ist es ein Pass-Spielzug, weil Nummer 27 einfach schlecht ist. Das ist sehr oft bei Running Backs. Das schaue ich mir oft an. Welcher Running Back ist am Feld? Und wenn der Star Running Back drinnen ist, dann ist es halt meistens normal, aber was ist wenn der Backup drinnen ist. Ist der Offensive Coordinator gewillt dem Backup Running Back den Ball in die Hand zu geben oder ist, wenn der Backup-Running Back drinnen ist, automatisch ein Pass? Das ist auch interessant.

NN: *Aufstellung des Teams wird somit ein riesiger Faktor sein aus dem ich sehr viele Informationen herausziehen kann, oder?*

MS: Genau. Unser größter Filter ist Aufstellung. Wir schauen uns alles an: Wenn zwei Receiver rechts, zwei Receiver links; drei Receiver rechts, ein Receiver links; zwei Running Backs, drei Receiver. Das ist unsere größte Einteilung an Scouting-Breakdown. Und eigentlich schauen wir uns fast nur in diesen Formationen die Datensätze an. Sprich, wenn sie als "two-by-two" rauskommen, es ist Dritter und fünf. Was ist aus "two-by-two" ihr Prozentsatz? Nicht was ist gesamt der Prozentsatz, das interessiert mich auch, aber noch interessanter aus der Formation, sozusagen. Was die Middle Linebacker, das sind die Quarterbacks der Defense, eingetrichtert bekommen. Dass die da draussen stehen, sie wissen es ist Dritter und fünf. Dann kommen sie in der Formation raus und checkt er sozusagen, dabei ist 80% Pass auf die linke Seite "check blue" oder "whatever". Aber das ist jetzt schon "highly advanced", das ist jetzt nicht...

NN: *Ja klar. Ich habe rausgefunden, dass das ein relativ großer Faktor ist. Die Daten sind aber zum Beispiel in der NFL nicht öffentlich verfügbar.*

MS: Sei froh, weil die haben ja eine Million Formationen. Da wirst du ein Schwammerl. Das ist das coole hier in Österreich. Da laufen wir mit drei bis vier Formationen herum. Das ist überschaubar. Die NFL hat so viel Zeit. Die können so viele Shifts, so viele Motions, die bewegen ihre Spieler vor dem Snap. Da wärst du nicht fertig geworden.

NN: *Ja. In der NFL werden inzwischen ja alle Spieler-Positionen getracked und das sind natürlich Datenformate die zwar nicht öffentlich verfügbar sind, aber damit könnte man natürlich unendlich viel machen.*

[...] The discussion of the online form was not transcribed. The results are visible in Table 3.

MS: Es gibt so Standard-Playcalling in der Offense und du könntest immer wenn du etwas herausfindest ein bisschen schauen ob das zutrifft. Oder du könntest das auch hernehmen als; ich weiß nicht ob das analytisch etwas bring; als so zu sagen Standard. Es gibt einen Standard.

Bei "first and ten" aus Offense-Sicht versuchst du 50/50 run oder pass zu sein. Bei "second and long" versuchst du mit einem Lauf in "Dritter und Machbar" zu kommen, das ist Dritter und vier oder weniger. Bei Zweiter und Medium; Medium bedeutet 4 bis 7; also Zweiter und

vier bis sieben Yards, behandelst du wie ein First-Down, also 50/50. Zweiter und kurz, sprich drei oder weniger Yards: Eine perfekte Zeit um einen tiefen Pass, oder einen Trickspielzug zu probieren, weil du natürlich, auch wenn er nicht hinhaut, noch immer Dritter und kurz hast. Das ist noch immer eine gute Situation, ok? Third down, Dritter und lang: Pass zum First down oder Screen. Screen oder Draw, wobei Draw der angedeutete Pass ist der erlaubt ist und Screen ist der angedeutete Pass der quasi ein Lauf ist. Bei einem Draw tut der Quarterback so als würde er passen und gibt im letzten Moment dem Running Back der hinter ihm wartet. Dann haben wir Dritter und Medium, das ist "your highest percentage to get a first down". Weißt du was ich mein? Also womit du dich am wohlsten fühlst um das First Down zu erreichen. Das ist Dritter und vier bis sieben. Das ist, egal ob Lauf oder Pass: "get there". Third and short ist im Normalfall ein Lauf für das First Down.

Das ist das was man irgendwann einmal lernt und "advanced" ist dann den Spieß echt umzudrehen, und zwar individuell. Und richtig geil ist dann; das ist etwas das ich in meiner Offense, in den letzten zwei Jahren mache, was ich auch bei Trainerkongressen vorstelle, ist: Ich lasse meine Offense "no-huddle" in einer Formation aufstellen, schaue mir an wie sich die Defense der Gegner aufstellt, handle mit meinem Kopf quasi "predictive". Also mit meinem kleinen Computer da oben, habe "predicted" was die Defense also machen wird und baue darauf meinen Offense-Spielzug in Real-Time auf. Das heißt ich versuchen einen Dateninput zu bekommen, eine Aufstellung der Verteidigung, eine Personalgruppe der Verteidigung, sehe das mit meinen Augen, und aufgrund der Vorbereitung aus der Woche ergibt sich dann der "most-likely-successful-play". Und somit bin ich nicht "down-and-distance" orientiert, sondern ich bin zum Beispiel nur Verteidigungs-Aufstellung und Verteidigungs-Personalgruppen orientiert. Und mir ist mittlerweile egal ob ich bei Zweiter und zehn mein First Down mache, oder bei Dritter und zehn. Weil zehn Yards werden immer zehn Yards bleiben. Die Frage ist nur: "What's the easiest way to get it?", und nicht "What's the most proper way to get it?". Das ist natürlich auf einem bisschen anderen Level. Also das heißt: "What's the easiest way to gain yards?" und nicht was ist prozentuell "most likely to gain yards". Dann müsste man sich eigentlich anschauen: Aufstellungen, und wie erfolgreich sind andere Aufstellungen mit welchen Spielzügen? Das ist das was ich analysiere. Ich analysiere wenig "Pattern", also wenig "Pattern" in "down-and-distance", was alle anderen machen, sondern ich versuche zu analysieren wo ist in dieser Verteidigungsaufstellung die schwächste Stelle, in dem Moment. Und versuche "down-and-distance" nicht im Kopf zu haben. Je weniger du es im Kopf hast, desto "geiler" kannst du "playcallen".

NN: *Sehr cool. Die letzte Frage wäre noch: Macht es für dich einen Unterschied ob es zum Beispiel ein "Second and ten" oder ein "Second and Nnne" ist, oder ist das wirklich nur in den Kategorien "medium", "short", "long"?*

MS: Gute Frage. Was wirklich ein Unterschied ist, ist ob es "Inches" oder ein Yard ist. Das macht echt einen Unterschied. Ob das neun oder zehn ist, ist mir völlig "wurscht". Aber ob es eins oder drei ist, also je näher es zum Firstdown ist, desto mehr Unterschied macht es. Je weiter man weg ist, desto weniger Unterschied macht es. Also ich habe zum Beispiel, wenns zur Goalline, zum Touchdown, geht bin ich in Zwei-Yards-Schritten unterwegs. "Two yards and in", "four yards and in", "six yards and in", also je näher es zum wichtigsten Punkt kommt, desto interessanter wird es. So ist acht oder neun echt "wurscht". Wo es nicht "wurscht" ist,

ist für die Spielzug-Ausführung. Du willst nicht, bei Zweiter und neun, auch wenn sie auf acht konzipiert ist, auf acht Yards laufen. Das ist Spielverständnis, dass man die Route auf neun Yards läuft und dass der Pass auf neun Yards kommt. Das sieht man sehr viel in unteren Ligen, dass es Dritter und sechs ist und der Ball wird auf vier Yards gefangen und es ist aus. Aber bei Dritter und zehn fallen sehr viele Spielzüge weg die du bei Dritter und fünf spielen könntest. Das heißt das zu spielende Package wird kleiner.

NN: *Dann wäre die selbe Frage noch auf das Spielfeld abgewandelt. Macht es einen Unterschied zwischen den detaillierten Spielfeldpositionen. Macht es sinn wenn ich es, zum Beispiel, nur in Redzone und Rest unterteile?*

MS: Ja, also ganz, auf jeden Fall. "Backed-up" zum Beispiel, das ist wenn du innerhalb deiner eigenen Fünf-Yard-Linie zum Ball kommst. Was glaubt das ganze Stadion? Die werden versuchen über sichere Läufe rauszukommen und was ich mache, beziehungsweise was du fast machen musst, ist bei First Down auf "das Zweite" zu gehen. Wenn die Defense nämlich springt, gewinnst du fünf yards. Das heißt es ist nicht mehr auf der eigenen Zwei- sondern auf der eigenen Sieben-Yard-Linie. Wenn ich aber einen Fehler bei diesem "on-two" mache und als Offense einen False-Start mache, dann gehe ich "half-the-distance-to-goalline": Nur ein Yard. Das Risiko ist sehr gut. Ich riskiere ein Yard und kann aber fünf gewinnen. Also das ist einmal was auf was du achten könntest. Und das Zweite: Wie Versuchen sie sich dann von der Endzone zu befreien? Ich versuche immer, zum Überraschungsmoment, den tiefsten Spielzug aus dem Playbook zu spielen und dann versuche ich irgendwie rauszukommen. Weil du wirst sowieso nicht mehr glücklich, von da hinten. Richtig glücklich wirst du nur wenn du einen 99-Yard-Touchdown hast. Alles andere ist nur Dahin-Getümpel. Ab der 15-Yard-Linie kommt das "over-the-field". Ich weiß nicht wie das andere nennen, aber ich nenne das "over-the-field". Das is von meiner eigenen 15-Yard-Linie bis hin zur gegnerischen 25-Yard-Linie. Das ist für mich "over-the-field" und das ist für mich wirklich... Da fühle ich ich sehr frei im Playcalling. Da ist das ganze Playbook offen. Und dann, wenn ich an die 25-Yards-Linie komme fängt die Redzone an. Normal fängt sie an der 20-Yard-Linie an aber für mich an der 25-Yard-Linie. Dort wird das Feld hinten einfach wieder kürzer. Das heißt... nimmt dir aus deinem Playbook wieder ein Bisschen etwas heraus. Ab dort denke ich dann in 5-Yard-Schritten. Von der 25, von der 20, von der 15, von der zehn, und ab der zehn kommt für mich die Goldzone. Das ist noch einmal eine andere Zone, ein anderes Paket. Ich habe Playsheets, da habe ich schon sozusagen die Spielzüge für die Zonen aufgeschrieben. Es ist auf jeden Fall wichtig auf welcher Yard-Linie man ist. Das ist ein Parameter den musst du haben. Unbedingt.

NN: *Diese Fragen sind darauf bezogen ob es Sinn macht, wenn man solche Parameter wirklich als Nummer oder nur als Kategorie angibt. Also ob ich einfach sage es ist "Redzone", es ist "in-the-field" oder es ist ganz hinten.*

MS: Ich würde die Zahlen hinschreiben. Sie kategorisieren. Damit du im Overview... in der Summary ist es als Zone, aber wenn ich einen Drilldown machen will und sagen will: auf der 18-Yard-Linie, "show me". Weil es ist echt ein Unterschied ob auf der 4-Yard- oder auf der 2-Yard-Linie. Das macht echt... Wenn du es auswertest... Das Playcalling von der 5-Yard-Linie und von der 2-Yard-Linie sind völlig unterschiedlich. Würde ich auf jeden Fall eine Zahl machen.

NN: *Gut. Du hast gesagt, dass du bei der Play-Clock einfach bis vier Minuten vor jeder Halbzeit spielst, als gäbe es keine Play-Clock. Verändert sich dann in den letzten vier Minuten noch etwas? Natürlich musst du mehr planen.*

MS: Es gibt zwei Vorraussetzung: das eine ist du bist in Ballbesitz und willst den Ballbesitz nicht verlieren, das ist der 4-Minute-Drill, da wird man nur mehr laufen. Nicht mehr ”out-of-bounds” gehen. Da sieht man manchmal Spieler die mit dem Ball laufen absichtlich auf den Boden herunterfallen, nur damit sie nicht ”out-of-bounds” gehen. Das ist der 4-Minute-Drill. Und dann gibts den 2-Minute-Drill. In dem versucht man genau das Gegenteil: So schnell wie möglich in die Endzone zu können, um so schnell wie möglich Punkte zu machen. Da wird man viel passen und da gibt es sogar etwas, das nennt sich Spike, wo der Quarterback den Snap nimmt und sofort in den Boden wirft. Das ist künstliches Zeit-Töten. Und dann kommt es noch darauf an, brauchst du noch zwei Punktgewinne, also zum Beispiel einen Touchdown und ein Fieldgoal, oder brauchst du nur einen, weil im Endeffekt geht es hintenraus nur noch darum den Ball zu besitzen. Wenn ich mit plus zehn bin habe ich überhaupt kein Problem damit, wenn die in der Offense mit 30 Sekunden noch an den Ball kommen. Da muss ich überhaupt kein Risiko eingehen. Wenn ich aber nur plus sechs bin, dann muss ich mit 30 Sekunden auf der Uhr vielleicht echt noch ein Risiko nehmen damit sie nicht mehr in Ballbesitz kommen. Dahingehend ist das Ballbesitz-Thema am Schluss echt ein Ding. Was man nicht vergessen darf ist, wenn wir spielen, gegen die Raiders spielen zum Beispiel, und wir wissen die werden 40 Punkte ”draufhauen”, gegen unsere Defense. Das hab ich eh vorher schon gesagt. Dann werde ich auch vor den vier Minuten schon so viel laufen wie möglich. Ein Problem ist dann wenn sie das überzuckern und sie wissen: Der versucht nur zu laufen”, dann stellen die eine Verteidigung hinein und dann habe ich wieder nicht viel davon.

NN: *Dass heißt aber, wenn wir schon dabei sind: Dass du bei der Score-Differenz nur auf Scores schaust, oder?*

MS: Score differential ist groß aber nur in Bezug auf Possessions. Das muss man immer im Blick habe. Mit 2-Point-Conversion mitzählen. Ein Two-Possession-Game ist für mich wie ein No-Possession-Game. Das geht zu schnell. Bei einem Three-Possession-Game muss ich wirklich schon schauen, dass ich eine Possession stehle. Das heißt... Es wäre vielleicht auch etwas: Wie viele Special-Teams-Trickspielzüge gibt es mehr in Three-Possession-Games als in Two-Possession-Games? Wann passieren Surprise-Onside-Kicks? Passieren die bei One-Possession-, Two-Possession-, oder Three-Possession-Games? Ich würde mich fast trauen zu wetten bei Three-Possession-Games, weil da das Team versucht eine Possession zu stehlen. Damit sie zum Beispiel wieder in ein Two-Possession-Game kommen.

C Archived websites

`https://web.archive.org/web/20200807214152/http://www.maxcoaching.at/wp/`, last accessed 7.8.2020

`https://web.archive.org/web/20200618053627/https://operations.nfl.com/stats-central/chart-the-data/`, last accessed 15.7.2020

https://web.archive.org/web/20200723221037/https://www.forbes.com/sites/kurtbadenhausen/2019/07/22/the-worlds-50-most-valuable-sports-teams-2019/, last accessed 16.07.2020

https://web.archive.org/web/20200611032541/https://www.chicagotribune.com/sports/ct-spt-nfl-revenue-super-bowl-20190128-story.html, last accessed 16.07.2020

https://web.archive.org/web/20200807214812/https://www.americangaming.org/new/nfl-could-reap-2-3-billion-annually-due-to-legalized-sports-betting/, last accessed 16.07.2020

https://web.archive.org/web/20200626202754/https://scikit-learn.org/stable/modules/generated//sklearn.linear_model.LogisticRegression.html, last accessed 6.8.2020

https://web.archive.org/web/20160303034124/http://www.forbes.com/sites/centurylink/2014/09/12/playing-the-data-game-why-the-nfl-is-now-tracking-players-with-rfid-chips/, last accessed 17.07.2020

https://web.archive.org/web/20200525192329/http://www.sloansportsconference.com/about/, last accessed 18.07.2020

https://web.archive.org/web/20200807215541/https://www.researchandmarkets.com/reports/4904383/sports-analytics-market-by-sports-type, last accessed 17.07.2020

https://web.archive.org/web/20200714055143/https://github.com/ryurko/nflscrapR-data, last accessed 28.07.2020

https://web.archive.org/web/20200723084545/https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, last accessed 6.8.2020