CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Comparison of Machine Learning Approaches Applied to Predicting Football Players Performance

Master's thesis in Computer science and engineering

ADRIAN LINDBERG
DAVID SÖDERBERG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Comparison of Machine Learning Approaches Applied to Predicting Football Players Performance

ADRIAN LINDBERG
DAVID SÖDERBERG

UNIVERSITY OF GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Comparison of Machine Learning Approaches Applied to Predicting Football Players Performance

ADRIAN LINDBERG
DAVID SÖDERBERG

Supervisor: Carl Seger, Research Professor, Functional Programming division, Computer Science and Engineering.

Supervisor: Yinan Yu, Postdoc, Functional Programming division, Department of Computer Science and Engineering.

Examiner: Andreas Abel, Senior Lecturer, Logic and Types division, Department of of Computer Science and Engineering.

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Comparison of Machine Learning Approaches Applied to Predicting Football Players Performance

ADRIAN LINDBERG
DAVID SÖDERBERG

Department of Computer Science and Engineering
Chalmers University of Technology

Abstract

This thesis investigates three machine learning approaches: Support Vector Machine (SVM), Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) on predicting the performance of an upcoming match for a football player in the English Premier League. Each approach is applied to two problems: regression and classification. The last four seasons of English Premier League is collected and analyzed. Each approach and problem is tested several times with different hyperparameters in order to find the best performance. We evaluate on five game weeks by picking a lineup for each model that is then measured by its collective score. The results indicate that regression outperforms classification, with LSTM being the best performing model. The score ends up outperforming the average of all managers during the evaluated period in the online football game, Fantasy Premier League. The findings could be used to assist in providing insight from historical data that might be too complex to find for humans.

Keywords: SVM, SVR, MLP, LSTM, Predicting Athletic Performance, Computer Science.

Acknowledgements

We would especially like to thank our supervisors Carl Seger and Yinan Yu, for their guidance and valuable feedback during this thesis.

Contents

List of Figures

List of Tables

1 Introduction

Many attempts at predicting the outcome of football matches around the world have been made, mainly to beat a bookmaker to make a profit from the predictions. However, predicting the performance of a specific player in a specific football-match is more of an undiscovered field.

One application for predicting the performance of a player in a football match is when picking a lineup for that match. Picking out a good lineup is a challenging task for all football managers as each match has different components that require different lineups. Some examples of components that affect lineups are the current combination of team members, as well as the opponents. Each player that is available for picking has their characteristics, such as strengths/weaknesses as well as player condition (unable to play for some reason). As many different factors affect the player performances, it is difficult to filter out the noise and consider all inputs. The idea is that a machine learning model can assist when managers are choosing their lineups by predicting the performance of players before the match has been played.

1.1 Aim

The thesis aims to analyze these two main problems:

- Is it possible to predict a player's performance for an upcoming football match using historical information about the match-participants?
- How does a time-series based approach like long short-term memory (LSTM) compare to a non-time-series based approach like support vector machine (SVM)/support vector regression (SVR) and multi layer perceptron (MLP) in terms of picking the best lineup?

In order to benchmark the chosen approaches, the football player will be evaluated with a performance score. The performance is measured based on a scoring system publicly available online. The system was chosen as it is both available online and extensive in the sense that it measures many different aspects of the match. More information about why this thesis uses an external scoring model will be discussed in Section 1.3 Evaluation. The scoring model is from the online game Fantasy Premier League (FPL), and its grading system is displayed in the Appendix A.7.

Example:

The football player John Lundstram has taken part in three matches so far and has received a score of 3, 14, 0 points for his respective performances. This week John is going to play in his fourth match, and this thesis challenge is to predict John's Fantasy Premier League-points before he plays the fourth (named y in the table below). This prediction will be based on previous matches played by others as well as John and information about his teammates as well as the upcoming opponent.

John Lundstram, Defender, Sheffield United (2019/20)					
Gameweek	Minutes Played	Goals Scored	Assists	...	Points
1	77	0	1	...	3
2	90	2	1	...	14
3	90	0	0	...	0
4	?	?	?	...	y
	Input				Output

1.2 Limitations

Even though the data set is in-depth and captures the football match in detail, it is impossible to record all aspects that could have an impact on the football players' performance. For example, it is difficult to measure if the player is having a good or a bad day, and therefore it is not in the data set.

The data set will contain information about all players' matches. On an individual level, the data set will contain information about the player and a summary of all player events in a specific match. It will also contain aggregated information about the player, his team members as well as the opponents of that match. However, there's no individual information about each team members/opponent in the data set as it would increase the data set immensely. An argument for this is that with more data it could predict more accurately; however, the uncertainty regarding which players the opponent will play, increases the risk of predicting on incorrect data.

The results are limited to the English Premier League (EPL).

1.3 Evaluation

A player's performance in a match is an abstract measurement that is hard to define. To help define this, we used the Fantasy Football model of scoring, which can help put a number on how well a player performed [19]. The system is well documented and publicly available online.

A short background on Fantasy Premier League:
In the Fantasy Premier League game, players of the game try to pick the favourite team for the next game week [21]. Each real-life football player gets a score for their performance for each game week. Those players of the game get points for having said football player in their chosen team. In FPL, the managers are able to select players from each team in the Premier League and use these in their Fantasy team. Since there are four different positions in Fantasy Premier League which has separate scoring functions, models were implemented for each position. The different positions are goalkeepers (gk), defenders (df), midfielders (md) and forwards (fw).

To compare the approaches further, five Premier League game weeks were selected that were not included in the training session. The models then selected the best possible lineup for these game weeks to inspect which players the model selects. The Fantasy Premier League score of this predicted lineups were then compared against how other Fantasy Premier League managers performed during the same game weeks.

These thesis models will have an advantage compared to other managers in FPL during this process since they did not have to care about the constraints of the game, such as being limited to a budget or not being able to make unlimited transfers between the game weeks. The models predicted an optimized lineup without any restrictions. Since some of the game weeks contain multiple matches for some teams (called double game weeks), and some game weeks miss matches for some teams (called blank game weeks), five game weeks were selected where all teams play exactly one match.

The selected game weeks are:

- Season 19/20, game week 21 (matches played 1 January 2020 and 2 January 2020)
- Season 19/20, game week 22 (matches played between 10 January 2020 and 12 January 2020)
- Season 19/20, game week 23 (matches played 18 January 2020 and 19 January 2020)
- Season 19/20, game week 25 (matches played 1 February 2020 and 2 February 2020)
- Season 19/20, game week 27 (matches played between 22 February 2020 and 24 February 2020)

1.4 Structure

The thesis is structured as follows: it starts in Section 2, Theory, with a background on the area of Machine Learning as well as Neural Networks for the reader unfamiliar with the topic.
Section 3, Data, explains in detail how the data was collected and later pre-processed. It also digs deeper into understanding the data by visualizing the different features

used. Methods (Section 4), go over how the models were implemented and later trained. The code used in this section is uploaded to the following GitHub repository: `https://github.com/dasovm/fpl-ml-comparision`. The results are presented in Section 5. Section 6 continues the background, but on a more specific field around the thesis. Several related works are summarized and what the key takeaways were. The Section discusses related work on predicting outcomes of football matches, predicting performance on the field by several different performance measurements and predictions on Fantasy Football in general. Section 7, Conclusion, end the thesis with a summary and a conclusion on what has been learned and what can be improved upon in future work.

2

Theory

This chapter provides a background on the machine learning concepts used in this thesis. It is intended for the reader who is not familiar with these concepts and therefore, experts in the area can skip this chapter.

2.1 Machine Learning

Machine learning is the study of mathematical models that improve autonomously by experience. The idea of machine learning is to provide a model with data and letting the model tweak itself until finding an optimal solution based on the provided data. The learning is done without having to manually tweak and set conditions of the model as a programmer or mathematician would do. Machine learning approaches are often divided into three categories: supervised-, unsupervised- and reinforcement learning. However, this thesis will focus solely on supervised learning.

2.2 Supervised Learning

Supervised learning uses past labelled data to predict future events. With the labelled data as ground truth, the error of the predicted data will iteratively be reduced until a sufficient model is found. The model takes inputs and attempts to learn by example, through a teacher who corrects the student when the output is wrong [9]. With enough corrections, the student learns to correctly, with some error margin, output an acceptable result. Supervised learning only works when the ground truth is known so that an error can be calculated.

There are different types of supervised learning algorithms, two of them being classification and regression. Classification algorithms are used when the outputs correspond to a limited set of classes. The algorithm outputs a line (or a hyperplane in higher dimensions) that separates the classes. Regression algorithms map the output to a point on a line (or a hyperplane) in order to reduce it to a value. The most commonly used learning algorithms are: support vector machines, linear- and logistic regression, naive Bayes, decision trees and k-nearest neighbours.

2.2.1 Support Vector Machines

Support vector machines (SVM) are a model in supervised learning. A support vector machine creates a hyperplane that acts as the boundary between sets of classes. When a new input is given, and there are several classes available, then the goal is to decide to which class the input belongs. In a mathematical sense, the algorithm maximizes the margin between the classes. Its called a linear classifier if the classification is based on a linear combination of the properties of the inputs. One of the main advantages of using SVM is that the learning process is often fast, compared to other machine learning algorithms [6].

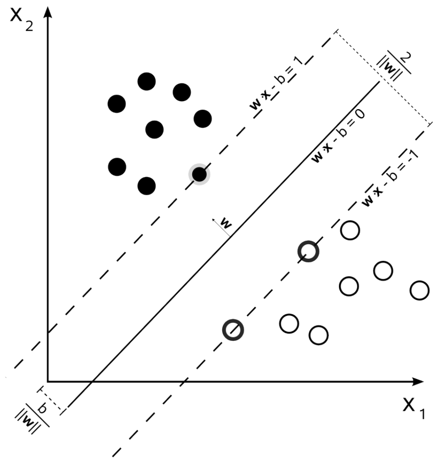


Figure 1: SVM-model that creates a line (hyperplane in larger dimensions) to separate the two classes (black & white) by maximizing the margin. The points that are positioned on the margin lines (the dashed ones) are called support vectors.[1]

2.2.2 Support Vector Regression

Vapnik et al., introduced in 1995 a new version of SVM for regression called xsupport vector regression [4]. Support vector regression (SVR) contains the same properties as a SVM. However, the main difference is that SVR outputs a hyperplane that most closely fits the inputs instead of a hyperplane that classifies the inputs into different categories.
There are different types of kernels one can use. This thesis will use a linear kernel for both the SVM and SVR, but a non-linear kernel is also often used. The kernel type specifies which type of function to use. Other hyperparameters used in a SVR model are C and ϵ. The equation for the line is $y = wx+b$. N is the amount of inputs.

Minimize:

$$\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

Constraints:

$$\begin{gathered} y_i - wx_i - b \leq \epsilon + \xi_i \\ wx_i + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{gathered}$$

[1] https://commons.wikimedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png

C: is a regularization parameter. By increasing its value, you are also increasing how much you want to avoid faulty classify the labels. However, increasing the value of C too much can lead to the model overfitting (Section 2.6). C is able to tune in both SVR-models and SVM-models for classification.

ϵ: The ϵ parameter specifies how much error to tolerate before penalizing in the SVR. The ϵ is unique for SVR-models since it's only applicable for regression optimizations.

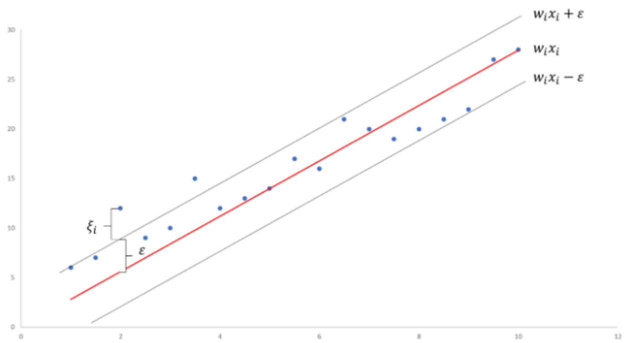


Figure 2: An illustrative example on how SVR works. The red line represents the best fit and the grey lines represent the tolerance of error ϵ.[2]

2.3 Artificial Neural Network

Artificial neural networks (ANN) are digital representations that try to mimic how the human brain learns. They are based on mathematical functions called artificial neurons, which models the neurons in a biological brain. ANN is a subfield of machine learning, meaning these networks also learn by provided data. The difference between these systems and SVM/SVR is mainly that it is possible to model more complex functions by connecting multiple artificial neurons.

Pitts and McCulloch introduced the first concept of an artificial neurons as a mathematical function [1]. A wiring of these is what later researchers would describe as a neural network.

2.3.1 Perceptron

A perceptron is a simplified model of a biological neuron, introduced in 1958 by Rosenblatt [2]. The perceptron is an algorithm for supervised learning of binary classification. A binary classifier is a function which can decide whether or not inputs belong to a specified class or not. A perceptron is a type of linear classifier, i.e. an algorithm that makes predictions based on a linear predictor function.

[2] `https://miro.medium.com/max/1400/1*nrXHNqC_hqpyux7GUbtqAQ.png`

The perceptron uses a threshold function as defined below: A function that maps its input $\vec{x}$ to an output value $f(\vec{x})$ (a single binary value, usually 0 or 1 if its a binary classification problem):

$$f(\vec{x}) = \begin{cases} 1, & \text{if } \vec{w} \cdot \vec{x} + b > 0 \\ 0, & \text{otherwise} \end{cases}$$

where $\vec{w}$ is a vector of weights and $\vec{w} \cdot \vec{x}$ is the dot product of $\vec{w}$ and $\vec{x}$.
The bias b acts as a threshold. The value of $f(\vec{x})$ is used to classify $\vec{x}$ as either a positive or a negative instance, in a binary classification problem.

The perceptron algorithm is also called a single-layer perceptron network. This is to differentiate it from its predecessor, the multi-layer perceptron network. As a linear classifier, the single-layer perceptron network is the simplest feedforward neural network.

2.3.2 Feedforward network

A feedforward neural network is an artificial neural network where connections between the nodes do not form a cycle, thus it differs from another network type called a recurrent neural network. In this network, the information moves in only one direction, forward, from the input nodes, through the hidden nodes (if any) and to the output nodes.

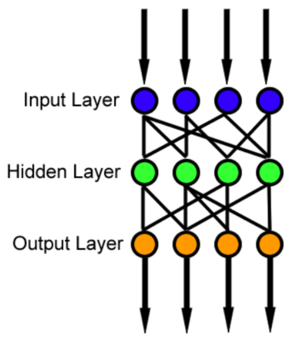


Figure 3: A typical feedforward network. Inputs traverse through each hidden layer until it reaches the output layer. [3]

The single-layer perceptron network consists of a single layer of input nodes and a single layer of output nodes. The inputs are fed directly to the outputs via a series of weights. A multi-layer perceptron network (MLP) is a standard feedforward network [3]. Which is comprised of several layers between the input and the output layers. The layers between input and output are often referred to as hidden layers, see figure 3.

[3] `https://upload.wikimedia.org/wikipedia/en/5/54/Feed_forward_neural_net.gif`

In machine learning, backpropagation is a widely used algorithm in training feedforward neural networks for supervised learning. In fitting a neural network, backpropagation computes the gradient of the loss function, which are introduced in Section 2.5, with respect to the weights of the network for a single feedforward of the input through the network. This is mainly done after one epoch, i.e. one completed training run of the data set.

2.3.3 Recurrent Neural Networks

A recurrent neural network (RNN) is a type of neural network where connections between nodes form a directed graph with cycles. With cycles in the network, the RNN can use the internal state (memory) to process variable-length sequences of inputs.

2.3.4 Long Short-term Memory

Long short-term memory (LSTM) is a type of recurrent neural network. A standard LSTM network has a cell that in turn has an input gate, an output gate and a forget gate. The cell remembers values over specific time intervals, and the three gates regulate the flow of information into and out of the cell. LSTM networks speciality is when basing predictions on time series data. The exact definition can be found in the authors, Hochreiter & Schmidhuber's paper, published in 1997 [5].

2.3.5 Masking

Even though RNN networks can handle variable length of inputs, i.a. different amount of time steps in the samples, the inputs do still need to be in fixed length when passed to the network. To process the variable inputs that are shorter then the fixed length, a popular method is to pad the inputs with zeros to match the shape needed for the network. By initially using a masking-layer in the network, the network ignores the timesteps where all values are zero, or any other given masking value.

2.4 Data normalization

Normalization of data sets is a common practice for many machine learning estimators. The aim of normalization is to get each feature in the data to be on the same scale. By not using normalization, some features risk dominating the objective function and make the estimator unable to learn from the other features.

2.5 Loss functions

Machines uses loss functions to learn. The loss functions evaluates how good the given algorithm is at modelling the given data.

2.5.1 Mean Squared Error

A popular loss function in the scope of regression problems is *mean squared error* (MSE), which is defined as follow:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n} (y_i - \hat{y}_i)^2$$

where y are the ground truth values and $\hat{y}$ are the predictions from the model. Models can be optimized by trying to minimize the evaluated mean squared error. In this thesis, MSE is used as loss function for the MLP- and LSTM-models solving the regression problem and used to evaluate all of the regression approaches.

2.5.2 Mean Absolute Error

Very similar to the mean squared error, *mean absolute error* (MAE) is another loss function to evaluate the performance of the predictions compared to the ground truth. It is defined as follow:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n} |y_i - \hat{y}_i|$$

where y are the ground truth values and $\hat{y}$ are the predictions from the model. Models can be optimized by trying to minimize the evaluated mean absolute error. In this thesis, MAE is used to determinate the similarities between two different data sets.

2.5.3 Binary Cross-Entropy

Binary cross-entropy is another popular loss function, mainly used for binary classification problems. It is defined as follow:

$$BC(y, \hat{y}) = -\frac{1}{n} \sum_{i=0}^{n} (y * \log \hat{y}_i + (1 - y_i) * \log (1 - \hat{y}_i))$$

where y are the ground truth values and $\hat{y}$ are the predictions from the model. In this thesis, binary cross-entropy is used to optimize the neural network approaches of the classification problem.

2.6 Overfitting

Even though a model seems to minimize the loss during training, it still does not have to perform well in general. Models that perform too well on the training data might have learned that noise and outliers have too great importance, which makes the model tailored to the particular training data. Therefore, even though the model continues to decrease the loss during training, the general loss might even increase and the gap between the training loss and testing loss may get bigger. Models that behave like this are said to be *overfitted* [16].

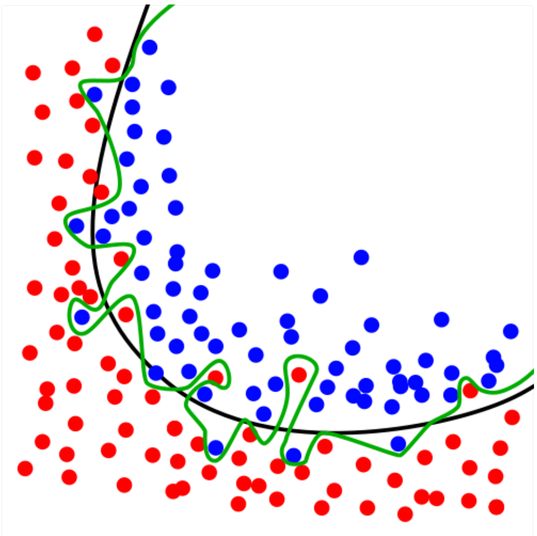


Figure 4: Illustration where the green line is overfit while the blackline fits the data well.[4]

2.6.1 Dropout

Dropout is a technique used in neural networks to reduce overfitting. This is done by ignoring (dropping out) a fraction p of neurons in a given layer to make the neurons less dependent on each other (and more general). Therefore, for each epoch neurons in the given layer are either dropped out with the probability p or kept with the probability $1 - p$.

2.6.2 Regularization

Regularization is a collection of techniques where information is added to the loss function to reduce overfitting. Two popular techniques are called L1 and L2 regularization, which are named after the L1 and L2 norm of a vector $\vec{w}$. The idea is to add a penalty based on the parameters in the model on the loss, in order to keep the parameters small in the model. These are defined as follow:

$$\text{L1-reguralized loss} = Loss(y, \hat{y}) + \lambda \Sigma_{i=0}^{n} |w_i|$$

$$\text{L2-reguralized loss} = Loss(y, \hat{y}) + \lambda \Sigma_{i=0}^{n} w_i^2$$

where λ is the rate of regularization. By adding a penalty to the models, the parameters are forced to be smaller, which makes it less likely that the model will learn from noise or outliers, and might reduce the chance of overfitting.

[4] `https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Overfitting.svg/1200px-Overfitting.svg.png`

3

Data

3.1 Collecting data from various sources

The data was collected from three different sources:

- A public repository from GitHub called *Fantasy-Premier-League* by a user called *vaastav* [20]. This repository provided us with data regarding all the players who were involved in any of the last four seasons of Fantasy Premier League (2016/17, 2017/18, 2018/19, 2019/20), alongside with their given points for every match of the season.
- The website *SofaScore* [22], where statistics were collected about the performance of each of the matches where the players were involved. For instance, shots, tackles, etc.
- The website *understat* [23], where even more detailed statistics were collected about each game, such as how many chances that were created from freekicks etc.

The data, by and large, consisted of:

- all matches for all seasons,
- players data for each match,
- home and away team data for each match,
- aggregated player data for each season, and
- aggregated team data for each season.

With "all seasons" the authors mean the last four seasons mentioned above (2019/20 season was not played in full and at the time of writing only 29/38 game weeks were completed). A season typically starts in the late summer and lasts to the late spring the next year (August-May). As the goalkeeper is vastly different from the other positions (defender, midfielder and forwards), the authors chose to separate the data into two data sets where the goalkeeper has features that the other positions do not have, and vice versa.

The total amount of data consists of 2820 matches, 984 players and 29 teams.

Game week data columns (outfielder):

ID	Description	Data Type
accurate-passes	Nr of accurate passes	int
accurate-passes-rate	Accurate passes out of all passes	float

aerial-duels-attempts	Nr of aerial duels attempts	int
aerial-duels-won	Nr of aerial duels won	int
assists	Nr of assists	int
big-chance-missed	Nr of times the player should have scored but missed	int
big-chances-created	Nr of times player created chances where the recieveing player should score	int
clearances	Nr of clearances	int
crosses-attempts	Nr of crosses	int
crosses-successful	Nr of successful crosses	int
dribble-attempts	Nr of dribble attempts	int
dribble-successful	Nr of successful dribbles	int
dribbled-past	Nr of times player got dribbled past	int
duels	Nr of duels	int
error-led-to-shot	Nr of errors that led to attempt at own goal	int
fouls	Nr of fouls	int
fpl-position	Position on the field	[1, 2, 3, 4]
goals	Nr of goals	int
goals-conceded	Nr of goals conceded	int
ground-duels-attempts	Nr of ground duels attempts	int
ground-duels-won	Nr of ground duels won	int
h-a	Player team is home or away	[0, 1]
hit-woodwork	Nr of times player hit woodwork	int
interceptions	Nr of interceptions	int
key-passes	Nr of important passes	int
long-balls-attempts	Nr of long balls attempts	int
long-balls-successful	Nr of long balls successful	int
minutes-played	Minutes played	int
outfielder-blocks	Nr of outfielder blocks	int
own-goals	Nr of own goals	int
penalties-missed	Nr of penalties missed	int
penalty-comitted	Nr of penalty comitted	int
penalty-won	Nr of penalties won	int
playing-probability	playing Probability	[0, 0.5, 1]
red-card	Nr of red cards	int
shots	Nr of shots	int
shots-off-target	Nr of shots off target	int
shots-on-target	Nr of shots on target	int
tackles	Nr of Tackles	int
total-points	Total Points that game week	int
touches	Nr of touches	int
was-fouled	Nr of times player was fouled	int
yellow-card	Nr of yellow cards	int

Table 3.1: All features for a player (outfielder) for a specific match.

Game week data columns (goalkeeper):

ID	Description	Data Type
accurate-passes	Nr of accurate passes	int
accurate-passes-rate	Accurate passes out of all passes	float
aerial-duels-attempts	Nr of aerial duels attempts	int
aerial-duels-won	Nr of aerial duels won	int
assists	Nr of assists	int
clean-sheets	Goalkeeper held the goal without letting any goals in	[1, 0]
clearances	Nr of clearances	int
dribbled-past	Nr of times player got dribbled past	int
duels	Nr of duels	int
fouls	Nr of fouls	int
goals-conceded	Nr of goals conceded	int
ground-duels-attempts	Nr of ground duels attempts	int
ground-duels-won	Nr of ground duels won	int
interceptions	Nr of interceptions	int
key-passes	Nr of important passes	int
long-balls-attempts	Nr of long balls attempts	int
long-balls-successful	Nr of long balls successful	int
minutes-played	Minutes played	int
own-goals	Nr of own goals	int
penalties-saved	Nr of penalties saved	int
red-card	Nr of red cards	int
tackles	Nr of Tackles	int
total-points	Total Points that game week	int
touches	Nr of touches	int
was-fouled	Nr of times player was fouled	int
yellow-card	Nr of yellow cards	int
goalkeeper-high-claims	Nr of goalkeeper high claims	int
goalkeeper-punches	Nr of goalkeeper punches	int
goalkeeper-runs-out-attempts	Nr of goalkeeper runs out attempts	int
goalkeeper-runs-out-successful	Nr of goalkeeper runs out successful	int
goalkeeper-saves	Nr of goalkeeper saves	int
h-a	Player team is home or away	[0, 1]
playing-probability	Playing Probability	[0, 0.5, 1]

Table 3.2: All features for a player (goalkeeper) for a specific match.

Since the SVR/SVM/MLP models are not time-based, a different data set had to be built. This is unlike LSTM that utilizes time in the network to learn the data set. The first data set had to be created with the time series aggregated on each feature. Based on the related work, it was concluded that using several time steps was the way to go. The chosen time steps were: 1, 3, 5 and all previous matches. This is notated like below:

Aggregated feature data on x previous matches: Where: -1: no previous matches and -all: all previous matches.

```
    Index(['assists-1', 'assists-3', 'assists-5',
       'assists-all', 'goals-1', ...],
      dtype='object')
```

The aggregated team data, both team members and opponents, is shown below:

Team data over a season:

ID	Description	Data Type
opponent-team-GA-direct-freekick	Opponent Team Goals Against using free kicks	int
opponent-team-GA-from-corner	Opponent Team Goals Against from corners	int
opponent-team-GA-open-play	Opponent Team Goals Against from open field	int
opponent-team-GA-penalty	Opponent Team Goals Against from penalties	int
opponent-team-GA-set-piece	Opponent Team Goals Against from set pieces	int
opponent-team-G-direct-freekick	Opponent Team Goals using free kicks	int
opponent-team-G-from-corner	Opponent Team Goals from corners	int
opponent-team-G-open-play	Opponent Team Goals from open field	int
opponent-team-G-penalty	Opponent Team Goals from penalties	int
opponent-team-G-set-piece	Opponent Team Goals from set pieces	int
team-GA-direct-freekick	Team Goals Against using free kicks	int
team-GA-from-corner	Team Goals Against from corners	int
team-GA-open-play	Team Goals Against from open field	int
team-GA-penalty	Team Goals Against from penalties	int
team-GA-set-piece	Team Goals Against from set pieces	int
team-G-direct-freekick	Team Goals using free kicks	int
team-G-from-corner	Team Goals from corners	int
team-G-open-play	Team Goals from open field	int
team-G-penalty	Team Goals from penalties	int
team-G-set-piece	Team Goals from set pieces	int

Table 3.3: All aggregated features for a team

The aggregated player data, both team members and opponents, is shown below:

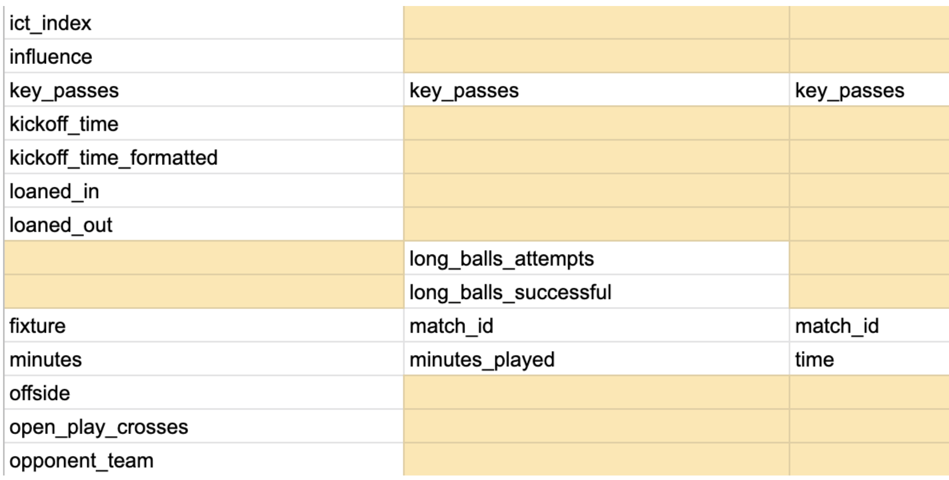
Player data over a season:

ID	Explanation	Data Type
goals	Nr of goals	int
G-direct-freekick	Nr of goals from a free kick	int
G-from-corner	Nr of goals from corner	int
G-open-play	Nr of goals from open field	int
G-penalty	Nr of goals from a penalty	int
G-set-piece	Nr of goals from a set piece	int
assists	Nr of assists	int

Table 3.4: All aggregated features for a player

3.2 Data pre-processing

After collecting the data, there exists three source files with different features. Each source sometimes uses their name for the same respective feature. In order to create a single extensive data set, a mapping structure was built in Excel. The mapping was made manually, with each source as its column.



ict_index		
influence		
key_passes	key_passes	key_passes
kickoff_time		
kickoff_time_formatted		
loaned_in		
loaned_out		
	long_balls_attempts	
	long_balls_successful	
fixture	match_id	match_id
minutes	minutes_played	time
offside		
open_play_crosses		
opponent_team		

Figure 1: The features location in the three sources here displayed as each source has its own column.

Problems arose where data in some cases was not indexed. This meant that data could not be joined together with an index, like an id, but had to be joined by comparing player names. More problems arose when different sources chose to spell the players' name either in a simplified way or correctly with the local encoding.

Example: Three examples on how the data could differ between the different sources.

```
Gabriel Armando de Abreu, Gabriel Paulista, Gabriel
Ezekiel Fryers, Zeki Fryers
Diego Da Silva Costa, Diego Costa
```

For each row, the three names refer to the same player, but the question is how to tell they are the same. A solution to find a correct matching player, a technique called Fuzzy/Approximate String Matching was used. It uses Levenshtein Distance to calculate the differences between sequences [24].

```
Input: Diego Da Silva Costa
Output: [('Diego Costa', 80), ('Diogo Jota', 75),
('Diogo Dalot', 64), ('Theo Walcott', 57)]
```

In the example above the string matching would return a list of probable players with the levenshtein distance. This worked well for most of the players as the main differences were encoding, but some players had to be manually mapped as the string matching returned wrong the result.

Example: Most of the differences were encoding related like the one below, but in some cases, the sources differed in showing the famous name/nickname as compared to the full name. Here are two examples of where the first name differs in encoding and the second in if the source used the full name or the nickname:

```
Encoding: Petr Cech,Petr Čech
Full/Nickname: Ezekiel Fryers,Zeki Fryers
```

The accuracy limit was set to 75% to make sure that each player mapped to their respective name correctly with a small error rate. The first name in the list is the most probable and was chosen if it had an accuracy over the limit. The output was then manually checked to make sure there were no errors in the mapping. The ones that did not meet the accuracy limit were placed in a different file and manually added later.

3.3 Data Analysis

When all data were collected and combined, a quality check was performed. This was done with the help of plotting distributions of the data. To better understand what the data consisted of, several distributions were plotted.

3.3.1 Position Analysis

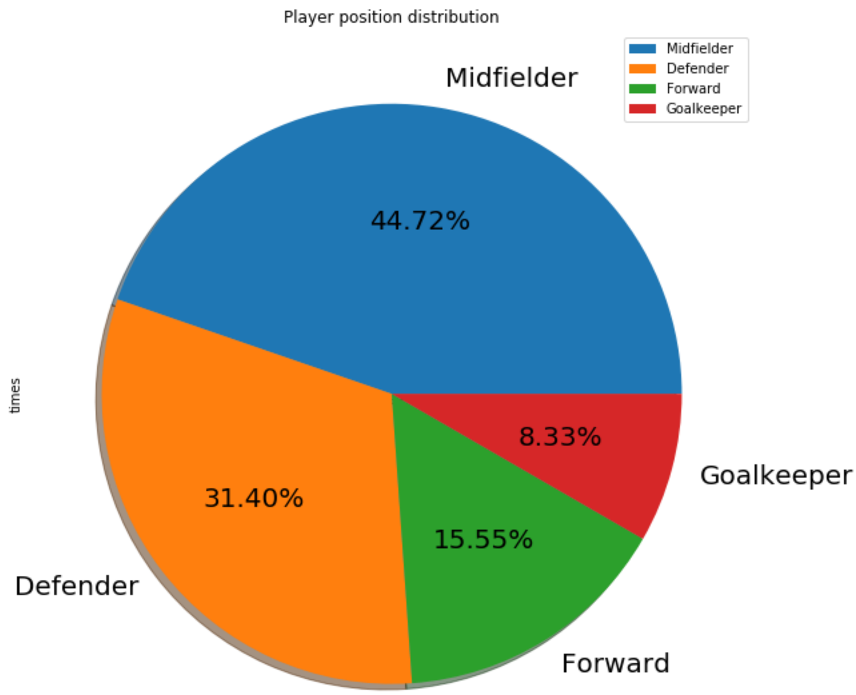


Figure 1a: Player position distribution in the data

Figure 1 shows how the majority of the data set consists of midfielders and defenders; thus, one might predict that performance for these two positions would be predicted more accurately. On a football field, each team has eleven players: one goalkeeper, ten outfielders. The outfielders are different combinations of defenders, midfielders and forwards. The distribution seems to follow a normal distribution of positions in a football match where usually the match will have more defenders and midfielders than other positions. Since the distribution looks correct, the quality check is approved.

3.3.2 Team Analysis

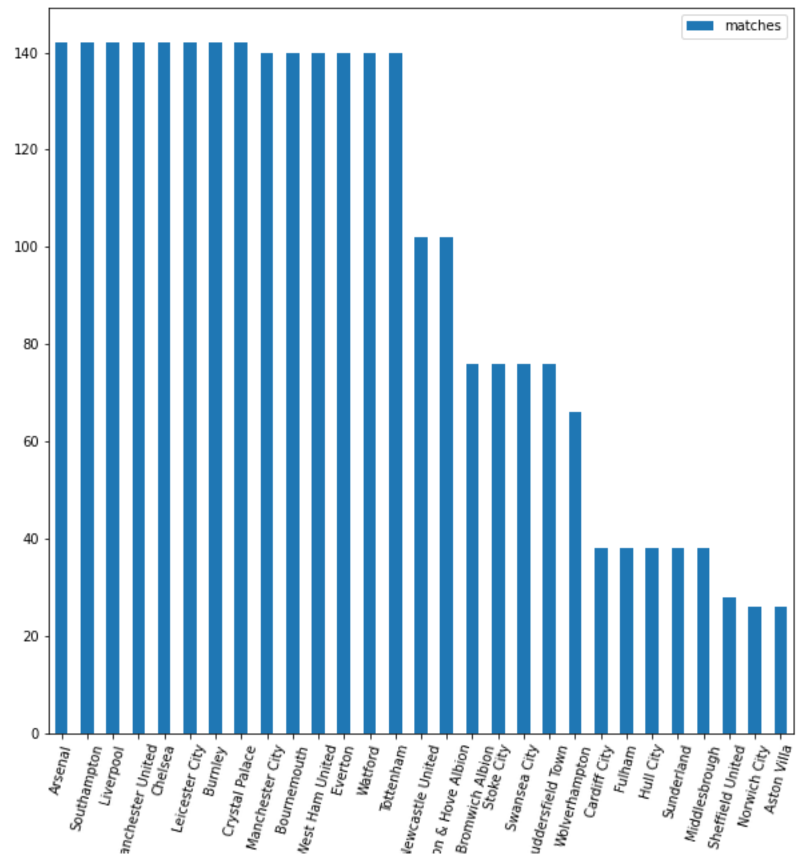


Figure 2: Team distribution in the data

Figure 2 shows how the team distribution looks to follow a standard order where some teams stay in the top and get to participate next year while others fall behind, and another team takes their places instead. In Premier League, each season, 20 teams compete against each other. The last three teams are moved to a lower division; thus, replaced by three teams that move up from a lower division. The first 17 teams get to stay to compete for next year. In the last four season, three teams had to step down into a lower division and have yet to return. This explains why the data set contains 29 teams, 20 teams + 3 new teams each season. From the Figure, it can be read that the teams with the lowest amount of matches are the teams that had to leave which coincides with previous seasons standings. An explanation on why some teams differ than others is that the 2019/20 season was cut short to only 29 games out of 38. As the distribution meets the expectations, the second quality check is approved.

3.3.3 Feature Analysis

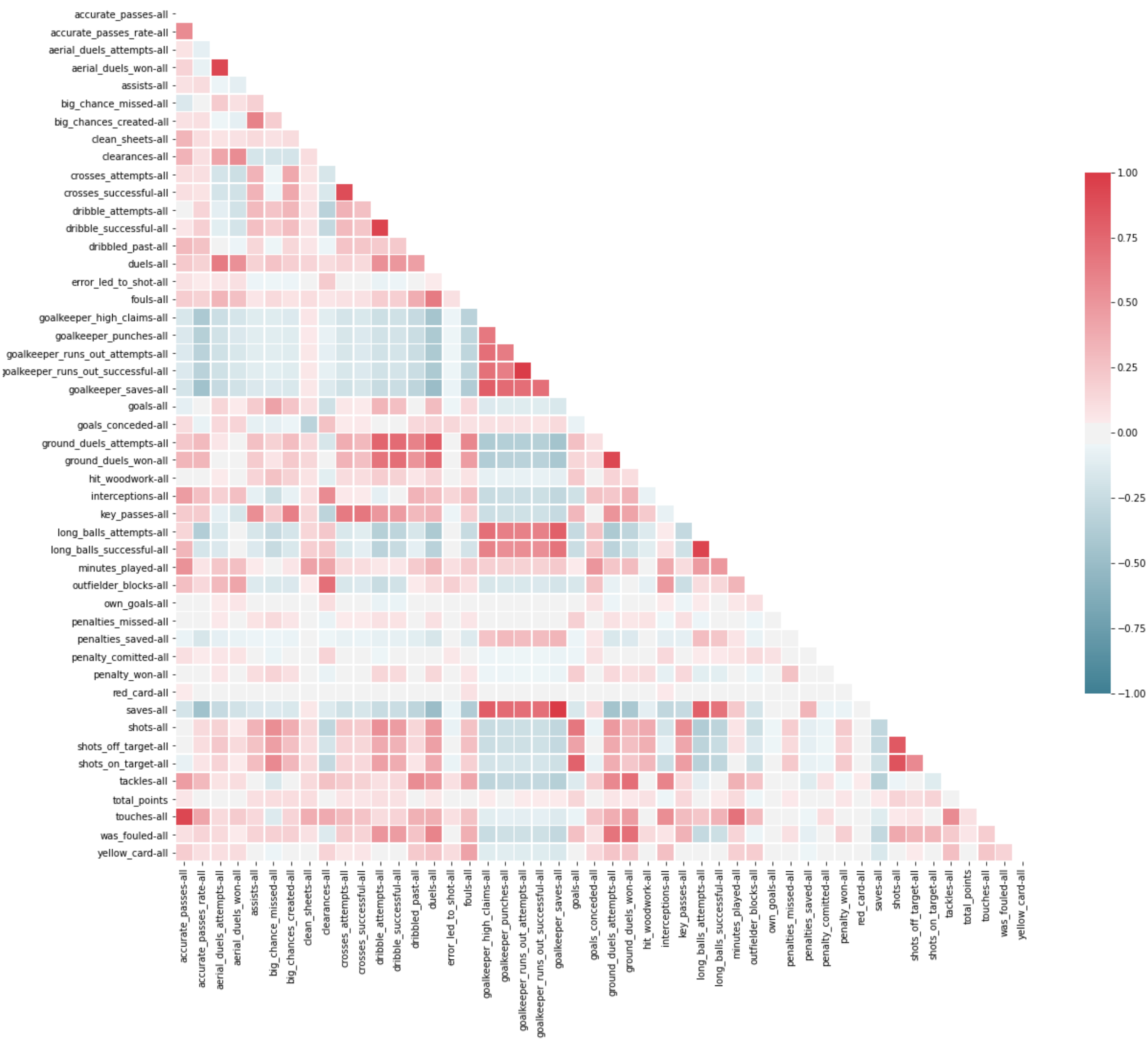


Figure 3: Correlation matrix of the data

The correlation matrix is another tool to prove the quality of the data. If two features that are related to each other do not correlate, it is an indication that something is wrong. Figure 3 shows how the different features correlate with each other. The correlation matrix is based on the cumulative sum of the features and shows how some of the features correlate with other while others do not.

Some features that should correlate with each other are for example touches vs accurate passes, as accurate passes can not increase without touches increasing. The same can be said for saves vs the goalkeeper features as reasonably the number of saves increase as goalkeeper high claims/punches increases. A negative correlation can be seen on long balls & saves vs duels & accurate passes. It is trivial that these are close to opposites in a game.

No apparent outliers that showed instances of incorrect data that would mislead the models in a negative manner were found. The third quality check passes, and thus, it can be established that the data looks to be in well enough shape to be used.

4

Method

4.1 Splitting into regression and classification

Originally the problem was only stated as a regression problem. However, an idea was formed that the problem might not be suitable to always maximize the potential output but instead to secure a good result even though it might not be the maximum. That is why the problem became split in two: regression and classification, where both methods were to be used on all approaches.

The intuition is that in the regression problem, the goal is to select the players that the model predicts will obtain the highest points. In the classification problem, however, the goal is to select the players that the model predicts having the highest probability of performing well. The regression model acts like a risk taker and is more likely to select those players that might receive a high score or a low score as compared to the classification model. The classification model is more conservative and selects players that are more likely to perform well, but not necessarily excellently.

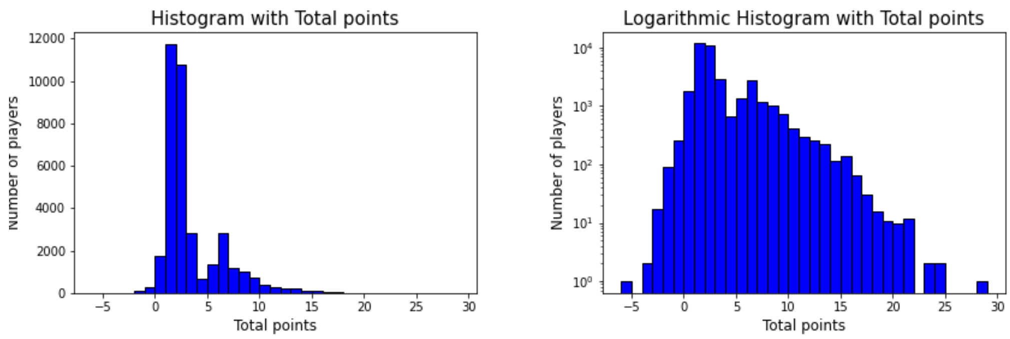


Figure 1: Histograms of total points in data set. The average is approximately 3 and the median is 2.

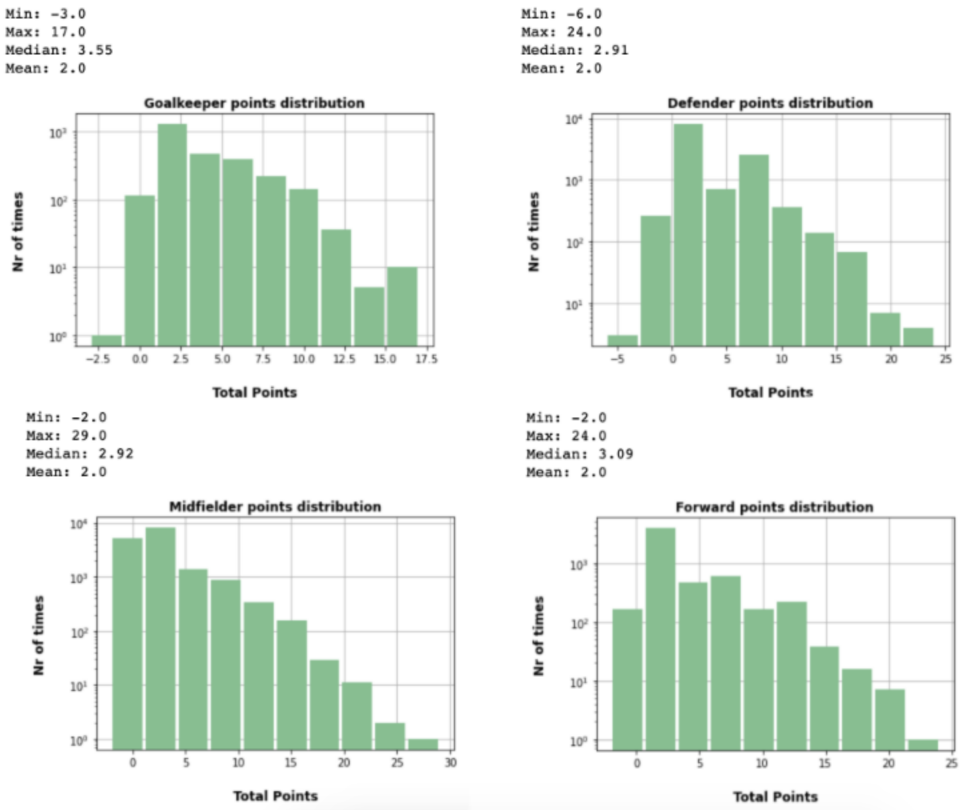


Figure 2: The different points distributions for all positions.

The score in the data set had a high variance from the range of -6 to 28 with the vast majority of the points as 2 or 3. To convert the existing regression problem into a classification problem, we used a "threshold-function" $f(y)$:

$$f(y) = \begin{cases} 0, & y \leq 4 \\ 1, & \text{otherwise} \end{cases}$$

Where y is the total points of a player in a given match and $f(y) = 1$ is defined as a good performance ($f(y) = 0$ the opposite).

4.2 Splitting the data set

In each of the approaches used in the scope of this thesis, the data set was split up into a training-part, a validation-part and a testing-part using 60% for training, 20% for validation and 20% for testing.

4.3 Implementing the models

All of the models were implemented using the Python programming language. The tools used to implement the networks were Scikit Learn [10] and Tensorflow [15]. Both of these are open-source machine learning libraries for Python development. The differences between them are that Scikit Learn is more for the traditional field

of Machine Learning, which makes it suitable for implementing the SVM-model. Tensorflow, on the other hand, is more focused on deep learning, which was needed in the two latter networks.
The different types of models that were implemented were a SVM, a MLP, and a LSTM. The models were implemented both solving the regression- and classification problem. These models are all part of the Machine Learning-space but contain different characteristics as mentioned above. The SVM model was selected to be the baseline.
In order to train the models, a "correct" output was to be evaluated. The correct output was obtained from the scoring model mentioned previously in Section 1.1.
The code used to implement the different approaches is uploaded to the following GitHub repository: `https://github.com/dasovm/fpl-ml-comparision`.

4.3.1 SVR

The SVR model was implemented using Scikit Learn. Each model was trained once using every combination of the following hyperparameters:

$$C = \{1, 2, 3, 4\},\ \epsilon = \{0.2, 0.4, 0.6, \ldots, 3.6, 3.8, 4.0\}$$

The hyperparameters were chosen after studying the related works. In total, each model was trained 80 times on the training data set. After every time, the model was evaluated against the validation data set, where only the model with the lowest validation loss (mean squared error) was saved and later used for evaluation on the test data.

4.3.2 SVM

The implementation of the SVM-model became quite similar to the SVR-model above since they contain the same properties. The Python package Scikit Learn was used, but since the SVM-model does not contain any ϵ-hyperparameter when solving classification problems, each model was trained once for every C:

$$C = \{1, 2, 3, 4\}$$

In total, each model was therefore trained four times on the training data set. After every time, the model was evaluated against the validation data set, where only the model with the lowest validation loss (binary cross-entropy) was saved and later used for evaluation on the test data.

4.3.3 MLP

The implementation of MLP was done in a similar matter for both the regression and the classification problem. The models were implemented using a Python package called Tensorflow [15]. Each model was trained once for 1000 epochs using every combination of the following hyperparameters:

$$\text{Amount of hidden layers} = \{1, 2, 3, 4, 5, 6\}$$
$$\text{Amount of neurons in each hidden layer} = \{64, 128, 256, 512\}$$

In total, each model was trained 24 times on the training data set. To counteract the models from overfitting, an l2-regularizer were used for every input- and hidden layer with a value of 0.001, together with a dropout-layer after every input- and hidden layer with a rate of 0.5.
After each epoch of training, the validation loss (mean squared error for the regression problem and binary cross-entropy for the classification problem) were obtained and evaluated against the previously trained epochs. If the validation loss was lower than the previous best model, the model was saved as the best one. If the performance of the model did not improve for 50 epochs, i.e. the validation loss of the model did not decrease, the training session was considered finished. The idea of stopping a non-improving model from training is very convenient called *early stopping* [8].

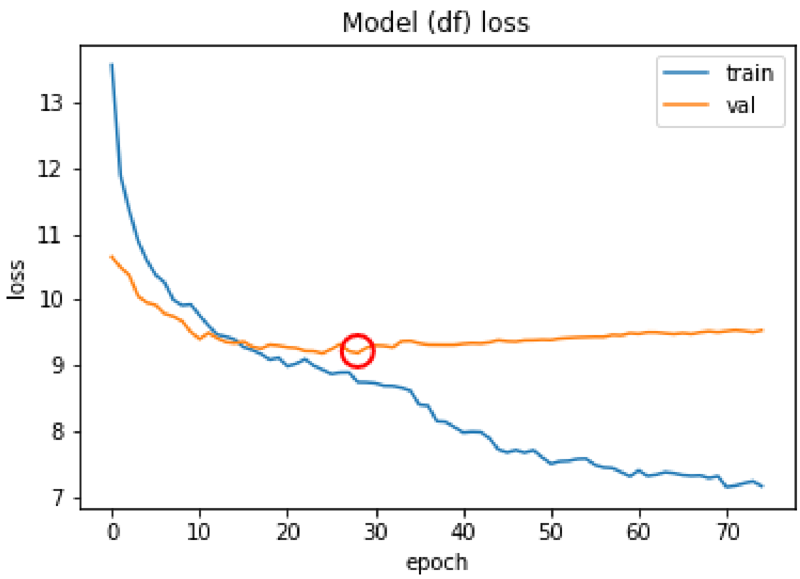


Figure 3: A training history of a MLP-model. The red circle (where the validation loss are the lowest) indicates where the model were saved and later on used for evaluation.

4.3.4 LSTM

As well as with the MLP-models, the LSTM-models were implemented in a similar matter for both the regression- and the classification problem. The models were implemented using Tensorflow[15]. Each model was trained once for 1000 epochs using every combination of the following hyperparameters:

$$\text{Amount of hidden LSTM-layers} = \{1, 2, 3\}$$
$$\text{Amount of neurons in each hidden LSTM-layer} = \{64, 128, 256\}$$

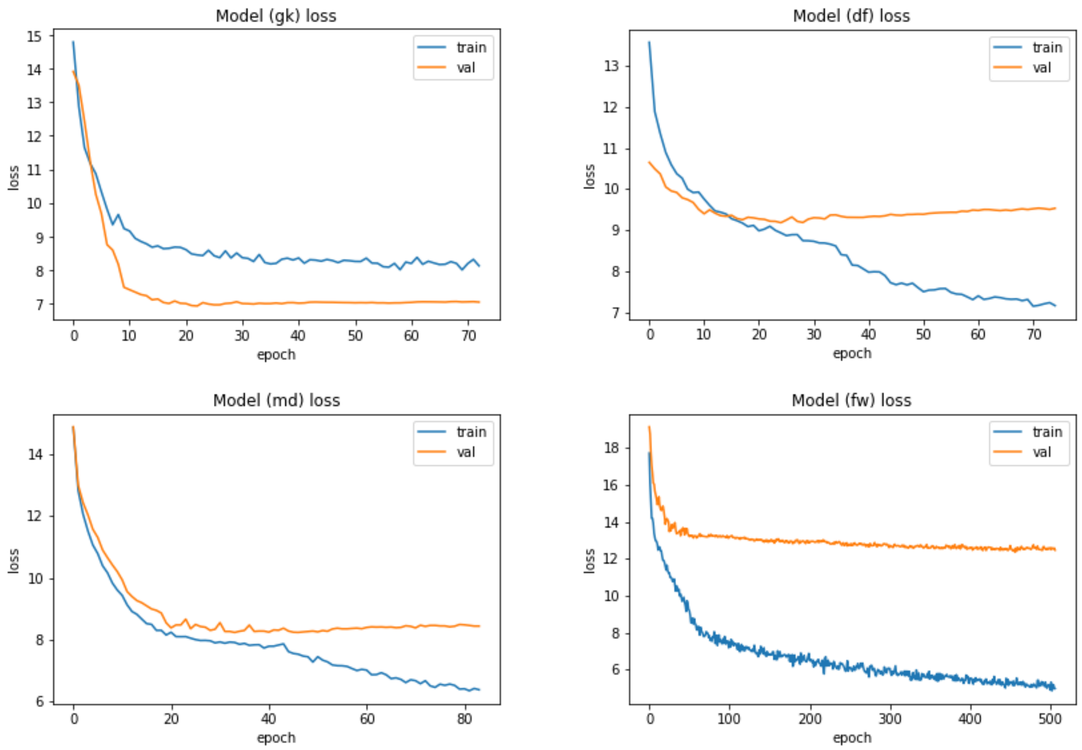


Figure 4: Training histories for the different regression MLP-models.

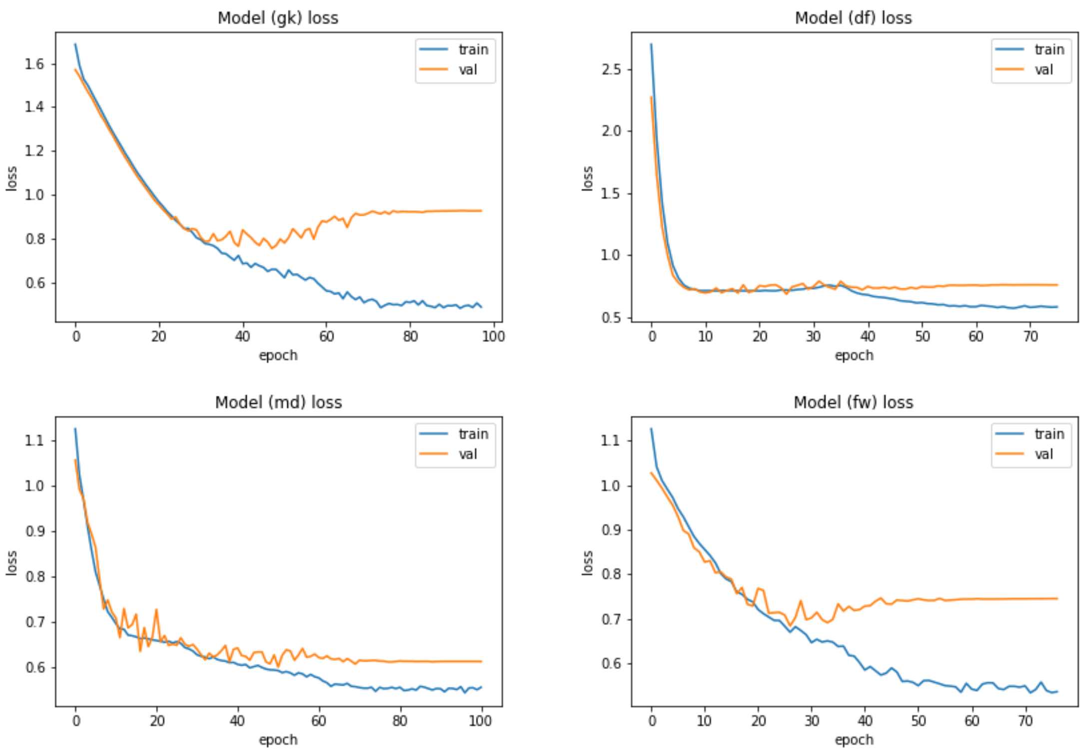


Figure 5: Training histories for the different classification MLP-models.

$$\text{Amount of neurons in the last hidden feedforward layer} = \{64, 128, 256\}$$

The first layer of every model was a masking layer to skip the zero-padded timesteps in the data set. Then came the hidden LSTM-layers, but these layers were tweaked a bit compared to the traditional ones, visually described in Figure 6. The data set mainly consisted of historical features, such as previous goals and passes. It also included some features that applied to the upcoming match, for instance, the probability of the player to participate in the upcoming match and if the upcoming match were played at home or away. To combine these features with the historical ones, an encoder was implemented for every LSTM-layer. This encoder consisted of a simple feedforward layer that took these features regarding the upcoming match as input and encoded them to the same size as a timestep in the historical data. The encoded row then became the initial state of the current LSTM-layer.

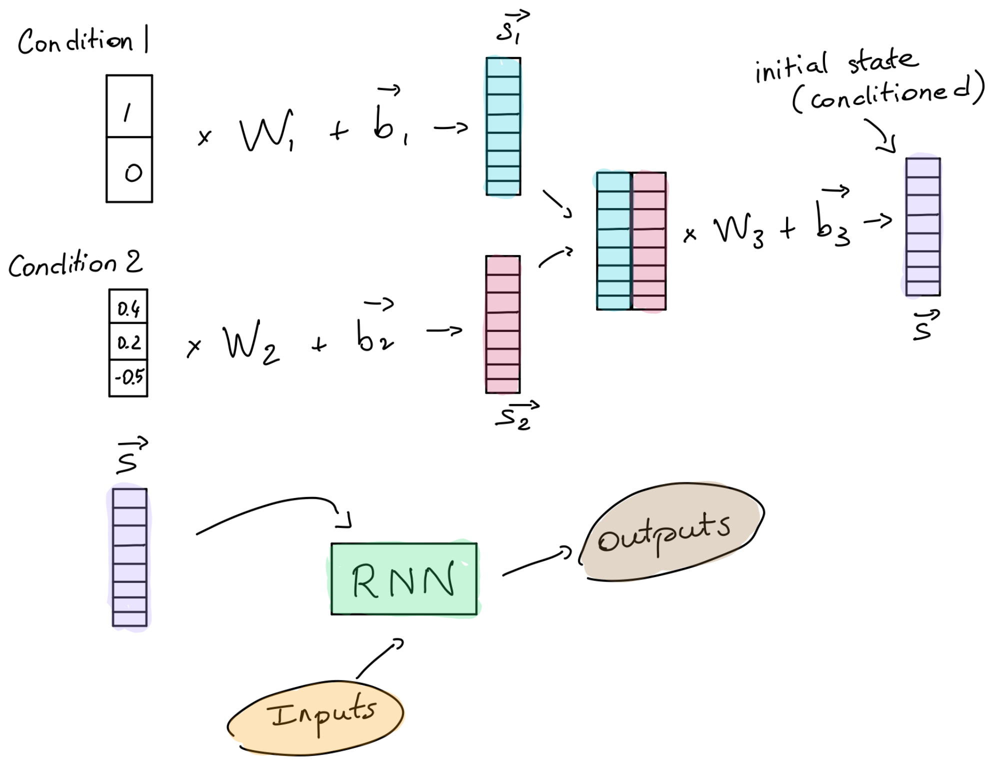


Figure 6: Visual description of the modification that was made in the LSTM-layers used in this thesis.[1]

The output from the hidden LSTM-layers was then passed to the last hidden layer, which consisted of a feedforward layer, before the output layer.
Dropout-layers with a rate of 0.5 were used in between every hidden layer, as well as an l2-regularizer with a value of 0.001, to counteract overfitting.
As with the implementation of the MLP-models, the validation loss (mean squared error for the regression problem and binary cross-entropy for the classification problem) were monitored after each epoch to save the best model during the training session and to finish the session if no improvements were made withing 50 epochs.

[1] `https://github.com/philipperemy/cond_rnn/raw/master/misc/arch.png`

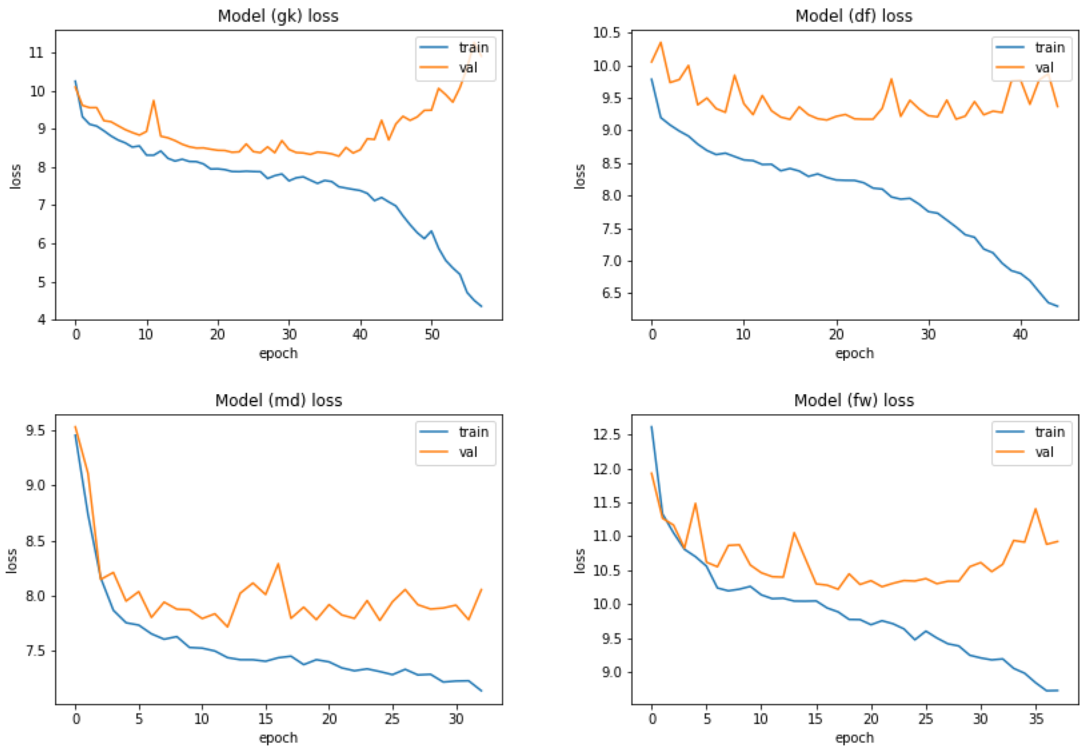


Figure 7: Training histories for the different regression LSTM-models.

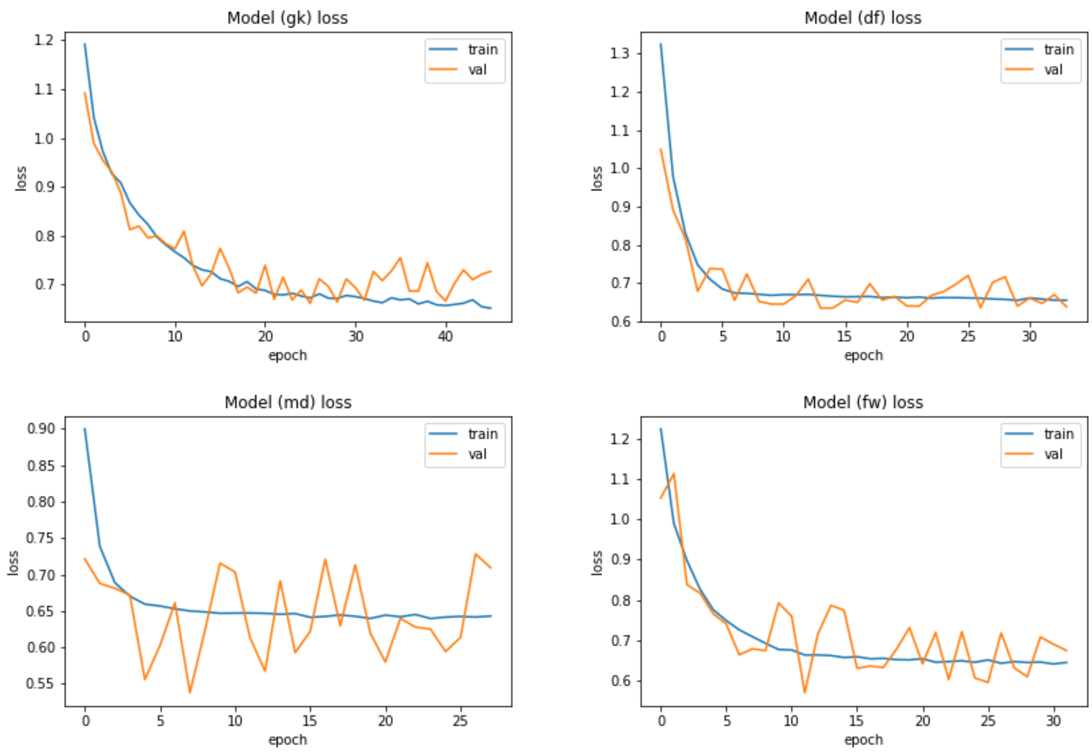


Figure 8: Training histories for the different classification LSTM-models.

5
Results

5.1 SVR training

After training and validating all SVR-models, the following hyperparameters were used as the best models:

Position	C	Epsilon
Goalkeeper	1	1.8
Defender	1	2.0
Midfielder	2	1.6
Forward	4	2.2

Table 5.1: Best hyperparameters for the SVR-models

To understand the models better, analyzing the feature importance was done by reading the coefficients of the regression. By displaying the five most- and least important features for the forward-model in Table 5.2, it becomes more understandable of how the model treats similar features.

Feature	Description	Importance
touches-all	Total amount of touches aggregated	1.11
ground_duels_won-all	Total amount of ground duels won aggregated	0.62
minutes_played-5	Amount of minutes played aggregated for the last five matches	0.51
dribble_successful-5	Amount of successful dribbles aggregated for the last five matches	0.50
assists-3	Amount of assists aggregated for the last five matches	0.41
...	...	...
accurate_passes-all	Total amount of accurate passes aggregated	-0.40
dribble_successful-all	Total amount of successful dribbles aggregated	-0.43
ground_duels_won-5	Amount of ground duels won aggregated for the last five matches	-0.43
minutes_played-all	Total amount of minutes played aggregated	-0.47
touches-5	Amount of touches aggregated for the last five matches	-0.86

Table 5.2: Feature importance for the SVR forward model

As displayed in Table 5.2, the model is balancing similar features by treating one variant of the feature with positive importance, while then treating another variant of the feature with negative importance. For instance, the feature with the highest positive importance in the forward model is *touches-all* with an importance of 1.11, while the feature with the highest negative importance is *touches-5* with an importance of -0.86. By analyzing these two features more deeply, it is possible to see how similar they are. The mean value for *touches-all* for all forwards are 25.70, while it is 26.17 for *touches-5*. The mean absolute error between them is only 3.39, which emphasizes how similar they are.

By evaluating the different models on the test data, using the best hyperparameters from the validation data, the following results were obtained:

Position	Loss (MSE)
Goalkeeper	9.58
Defenders	9.59
Midfielders	7.88
Forwards	9.87

Table 5.3: Test losses for the SVR-models

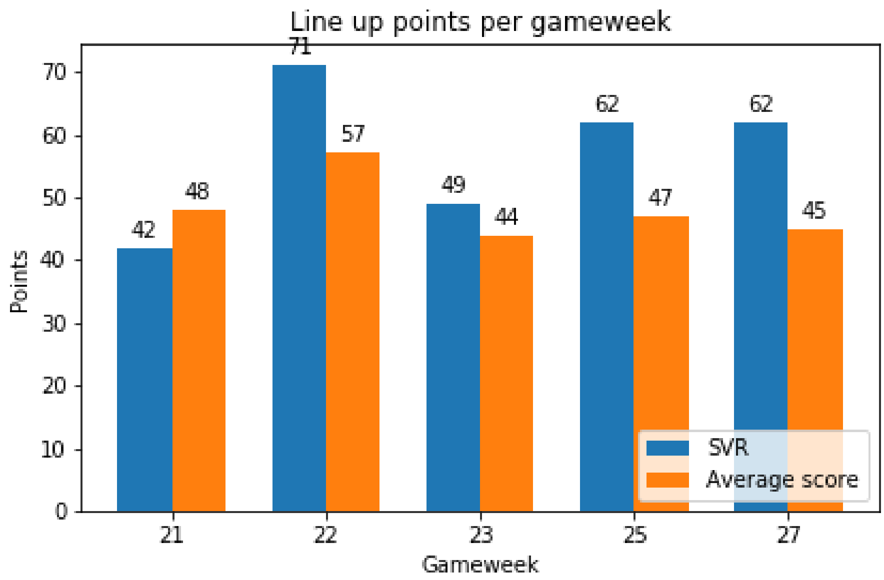


Figure 1: Points obtained from the line-ups generated by the SVR-models for the five evaluation game weeks, compared to the average points at the same time.

As seen in Figure 1, the SVR-models managed to beat the average Fantasy Premier League managers in 4/5 game weeks. For all selections for the five evaluation game weeks, see Appendix A.1.

Although the SVR-models did not seem to be performing well based on the loss function, they still were able to predict players that performed better than the average managers of the evaluation game-weeks. The models seem to pick popular choices of players, mainly from the best teams in the league. For instance, Mohamed Salah is a midfielder from one of the best team in the Premier League, Liverpool, that the models selected as a midfielder for all five game-weeks. For all of the evaluated game weeks, he has been a popular pick in the game with more than 20% of the managers selecting him in their team.

However there are some exceptions, and one of the more impressive ones being the forward Diogo Jota in game week 27. Jota's team Wolverhampton played at home against Norwich, an opponent who had struggled defensively recently. Instead of picking the more popular Wolverhampton-forward Raul Jimenez who was selected by more than 30% of all managers, the model predicted a higher score of the more rare forward Diogo Jota, chosen by less than 5%. Jimenez obtained a score 5 points during the game-week, while Jota obtained a score of 16 points.

5.2 MLP training (regression)

After training and validating all MLP-models for regression, the following hyperparameters were used as the best models:

Position	Hidden layers	Neurons in each hidden layer
Goalkeeper	6	64
Defenders	1	128
Midfielders	1	64
Forwards	1	64

Table 5.4: Best hyperparameters for the MLP-models

By evaluating these models by obtaining the mean squared error from the test-data predictions, the following result were obtained:

Position	Loss (MSE)
Goalkeeper	6.44
Defenders	9.55
Midfielders	7.99
Forwards	10.64

Table 5.5: Test losses for the MLP-models

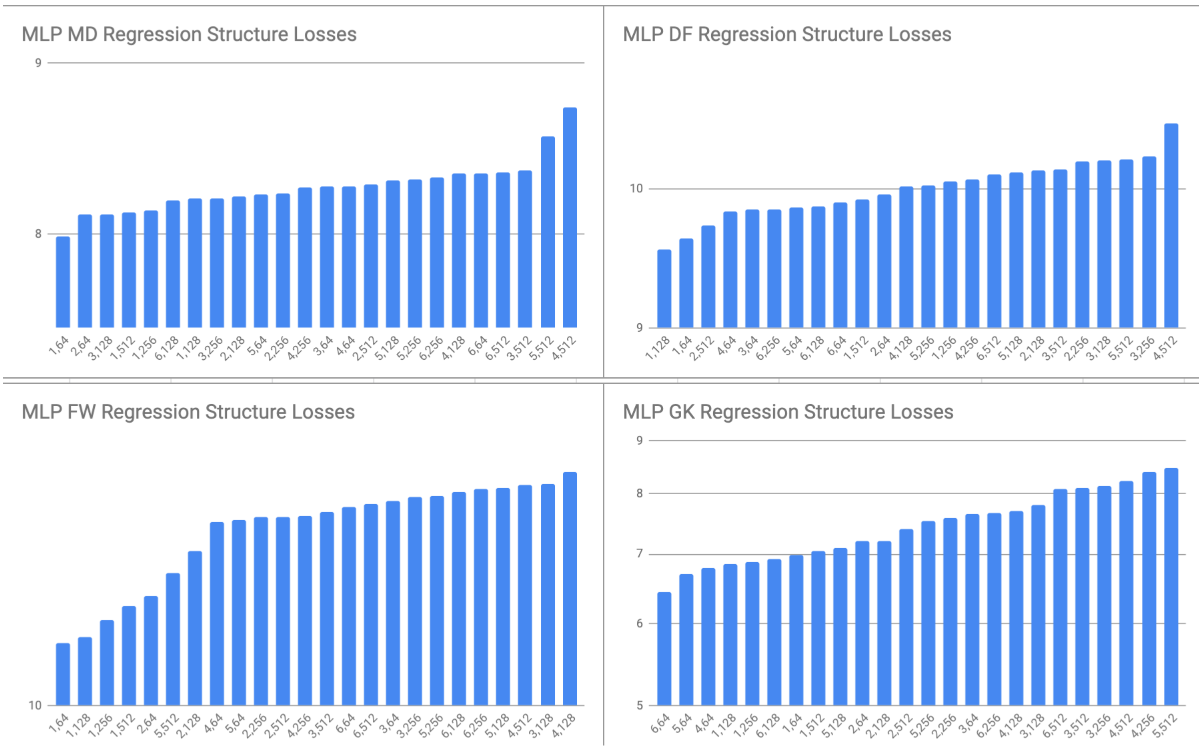


Figure 2: The different losses for all tested structures. The structure is (layers, neurons).

In Figure 2 it is clear that as the number of neurons increase, with the number of layers, the network overfits more; thus the validation loss becomes higher. All positions (except goalkeeper) have a similar structure between with one hidden layer of 64-128 neurons. Goalkeeper instead had the lowest loss with 6 hidden layers of 64 neurons.

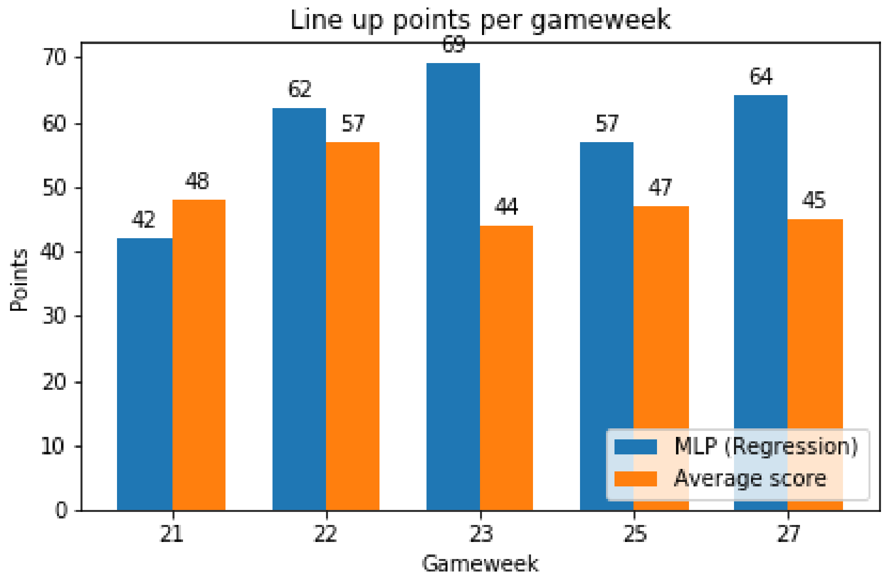


Figure 3: Points obtained from the line-ups generated by the MLP-models for the five evaluation game weeks, compared to the average points at the same time

As seen in Figure 3, the MLP-models seem to perform quite similar to the SVR-models, with some improvements on the defender model while the midfielder model obtained a slightly worse result. The selected lineups did beat the average score in 4/5 of the evaluation game-weeks. For all selections for the five evaluation game weeks, see Appendix A.2.

As well as with the SVR-models, most of the picks were popular choices, especially the selected midfielders. However, some selections are more interesting. One of them is the selection of the Manchester City-midfielder Riyad Mahrez in game week 22, which the model predicted to be the second-best performing midfielder. Mahrez was selected by $\sim 5\%$ of all managers at the start of the game week, which is low compared to some of the other Manchester City-midfielders such as Kevin De Bruyne ($\sim 50\%$).

Although being a less popular choice, Mahrez obtained 17 points, while De Bruyne got 9 points (the third midfielder selection by the model). By analyzing the importance of the different features in the midfielder model, it is possible to see that five of the most important features are: *h_a* (Home or Away), *was_fouled-all*, *penalty_comitted-1*, *goals-3* and *shots_on_target-all*. Since Mahrez and De Bruyne play for the same team, they have the same value of *h_a*. None of them did also commit any penalty during the last game; therefore, they have the same value of *penalty_comitted-1*. However, De Bruyne scored one goal in the last three matches while Mahrez did not score any goals, which might be one of the reasons for De Bruyne being a far more popular choice than Mahrez. On the other hand, Mahrez has higher value in both *was_fouled-all* and *shots_on_target-all* compared to De Bruyne (0.82 and 1.18 vs 0.75 and 0.9), which might be one of the reasons why the model found Mahrez more interesting than De Bruyne.

Another interesting pick was the Tottenham defender Japhet Tanganga in game week 25, which the model predicted as the best performing defender despite playing versus one of the best offensive team in the league, Manchester City. This is an interesting pick since Tanganga was a relatively new defender for Tottenham with only two matches played before, and was only selected by $\sim 1.5\%$ of the managers.

By analyzing the feature importance, *h_a* (the team to be playing at home) was found to be the most important feature, which might be one reason why Tanganga became a good pick according to the model. Tottenham managed to keep a clean sheet against Manchester City, and Tanganga obtained a score of 6 points. The same prerequisites goes with the Leicester goalkeeper Kasper Schmeichel, who was selected as a goalkeeper by the model in game week 27 despite facing Manchester City. Although Leicester conceding one goal, Schmeichel saved a penalty and obtained an impressive score of 12 points.

5.3 LSTM training (regression)

After training and validating all LSTM-models for regression, the following hyper-parameters were used as the best models:

Position	**Hidden LSTM-layers**	**Neurons in each hidden LSTM-layer**	**Neurons in last hidden layer**
Goalkeeper	2	256	64
Defenders	1	64	64
Midfielders	3	128	128
Forwards	2	64	128

Table 5.6: Best hyperparameters for the LSTM-models

By evaluating these models by obtaining the mean squared error from the test-data predictions, the following result were obtained:

Position	**Loss (MSE)**
Goalkeeper	8.59
Defenders	8.31
Midfielders	6.88
Forwards	9.65

Table 5.7: Test losses for the LSTM-models

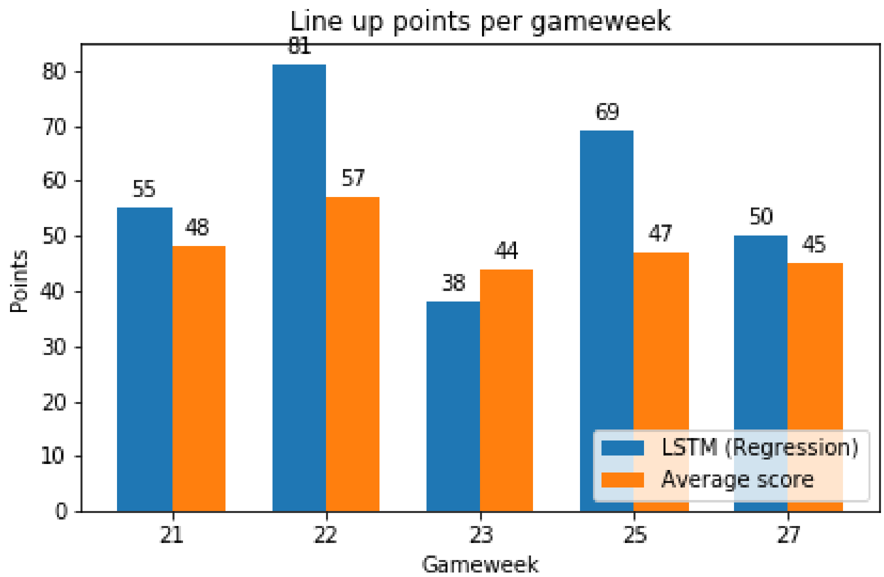


Figure 4: Points obtained from the line-ups generated by the LSTM-models for the five evaluation game weeks, compared to the average points at the same time

As seen in Figure 4, the LSTM-models for regression managed to beat the average

Fantasy Premier League managers in 4/5 game weeks. For all selections for the five evaluation game weeks, see Appendix A.3.

The trained LSTM-models outperformed both the SVR-models and MLP-models in every position for the test data. By analyzing the evaluation game weeks, it is possible to see that a majority of the players selected are playing in either Liverpool or Manchester City. However, an interesting selection made by the model was the Chelsea defender Antonio Rüdiger in game week 22, which the model predicted to be the best performing defender for the entire game week. Rüdiger was only selected by $\sim 0.3\%$ of the managers in Fantasy Premier League and had not been perfoming anything special pointwise (average 2.2 points per match for the last five games), without any scoring any goals or assists for the entire season. For game week 22, when Chelsea played at home against Burnley, Chelsea managed to keep a clean sheet and Rüdiger obtained 6 points. However, the underlying data that made the model select Rüdiger might have been indicating that he was about to perform better offensive as well, only three matches later Rüdiger scored two goals and obtained a score of 16 points when Chelsea faced Leicester City away from home.

5.4 Regression comparison

	Goalkeepers	Defenders	Midfielders	Forwards	Average
SVR	9.58	9.59	7.88	9.87	9.23
MLP	9.65	9.20	8.05	9.82	9.03
LSTM	**8.59**	**8.31**	**6.88**	**9.65**	**8.36**

Table 5.8: Comparison between the different regression approaches using MSE on the test data (the lower, the better)

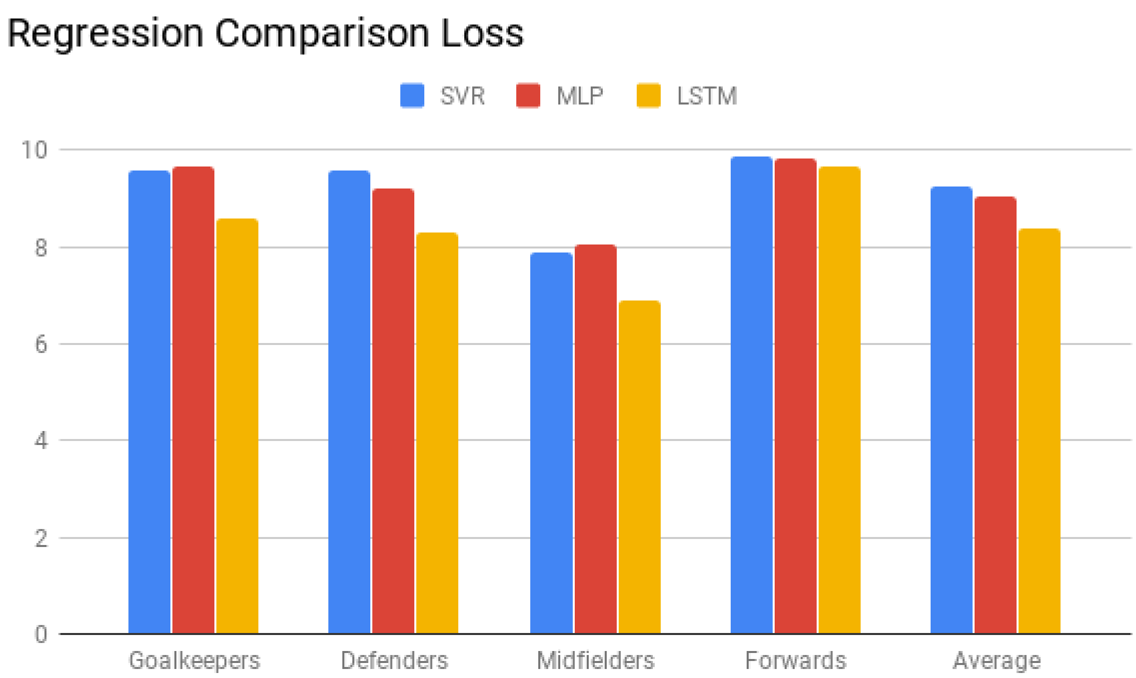


Figure 5: A graph displaying Table 5.8

As mentioned before, LSTM outperforms the other approaches for all positions.

5.5 SVM training (classification)

After training and validating all SVM-models, the following hyperparameters were used as the best models:

Position	C
Goalkeeper	1
Defenders	4
Midfielders	2
Forwards	4

Table 5.9: Best hyperparameters for the SVM-models

By evaluating these models by obtaining the accuracy from the test-data predictions, the following result were obtained:

Position	Accuracy
Goalkeeper	68.20%
Defenders	60.02%
Midfielders	68.37%
Forwards	60.35%

Table 5.10: Test accuracies for the SVM-models

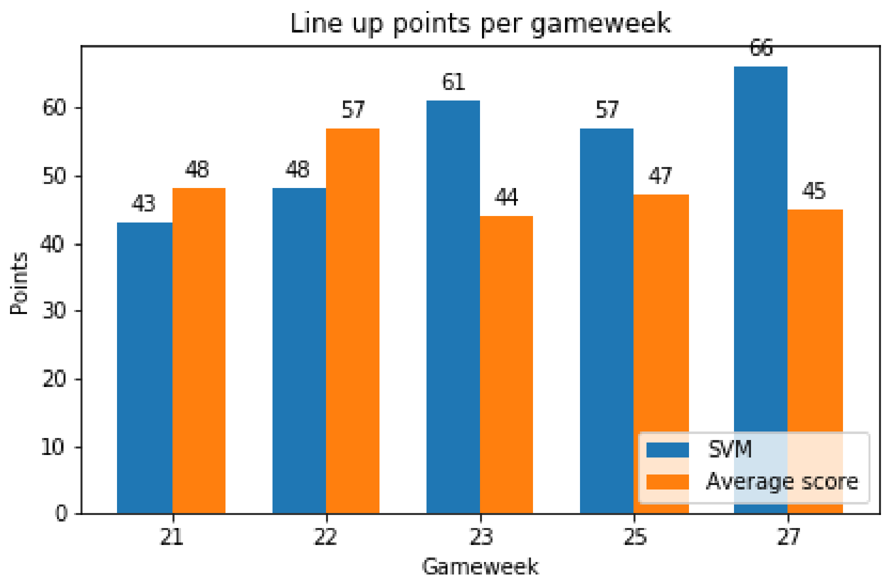


Figure 6: Points obtained from the line-ups generated by the SVM-models for the five evaluation game weeks, compared to the average points at the same time

As seen in Figure 6, the SVM-models managed to beat the average Fantasy Premier League managers in 3/5 game weeks. For all selections for the five evaluation game weeks, see Appendix A.4.

The SVM-model obtained lower scores than all of the other approaches on the evaluation game weeks and the although the accuracies on the test data set did not tell us anything yet, it was a bit lower than expected. Some interesting selections were the pick of three Brighton defenders in a relatively tough game home versus Chelsea. Even though the model picked the defenders as third, fourth and fifth choice, none of them were especially popular by the Fantasy Premier League managers. Also, none of them performed well in game with obtained scores of 2, 1 and 2 points respectively. The model did also select Riyad Mahrez as first midfielder in game week 22, as the MLP-model for regression did, and Diogo Jota as first forward in game week 27, as the SVR-model did.

5.6 MLP training (classification)

After training and validating all MLP-models for classification, the following hyperparameters were used as the best models:

Position	Hidden layers	Neurons in each hidden layer
Goalkeeper	5	128
Defenders	5	256
Midfielders	6	512
Forwards	6	256

Table 5.11: Best hyperparameters for the MLP-models (classification)

By evaluating these models by obtaining the accuracy from the test-data predictions, the following result were obtained:

Position	Accuracy
Goalkeeper	66.10%
Defenders	69.89%
Midfielders	82.04%
Forwards	75.25%

Table 5.12: Test accuracies for the MLP-models

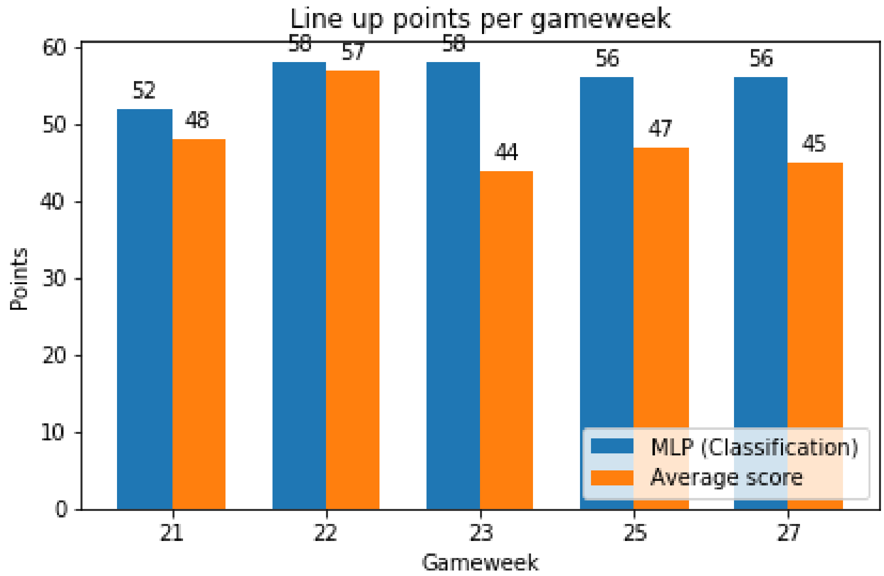


Figure 7: Points obtained from the line-ups generated by the MLP-models for the five evaluation game weeks, compared to the average points at the same time

As seen in Figure 7, the MLP-models for classification managed to beat the average Fantasy Premier League managers in all five game weeks. For all selections for the five evaluation game weeks, see Appendix A.5.

The MLP-model increased the accuracy of the training data quite significantly, with only the goalkeeper model to perform slightly worse then the SVM-model. The model obtained 52, 58, 58, 56 and 56 points which is noticeable if it is compared to for instance, the MLP regression, which obtained 42, 62, 69, 57 and 64 points. Even though the regression approach overall got a higher score, the classification is a lot more stable with a lower variance (4.8 for classification compared to 85.36 for regression).

One of the more interesting selections made by this model is the selection of the Burnley-defender Phil Bardsley in game week 27. Even though Bardsley were selected as the fifth defender by the model, he was only selected by $\sim 2.6\%$ of the Fantasy Premier League managers. Burnley played a relatively easy match at home against Bournemouth and managed to keep a clean sheet while Barsley also assisted to one of Burnley's three goals, meaning that Barsley obtained a score of 10 points.

5.7 LSTM training (classification)

After training and validating all LSTM-models for classification, the following hyperparameters were used as the best models:

Position	**Hidden LSTM-layers**	**Neurons in each hidden LSTM-layer**	**Neurons in last hidden layer**
Goalkeeper	1	128	64
Defenders	1	256	64
Midfielders	1	64	256
Forwards	1	128	256

Table 5.13: Best hyperparameters for the LSTM-models (classification)

By evaluating these models by obtaining the accuracy from the test-data predictions, the following result was obtained:

Position	**Accuracy**
Goalkeeper	70.00%
Defenders	69.70%
Midfielders	82.78%
Forwards	75.00%

Table 5.14: Test accuracies for the LSTM-models

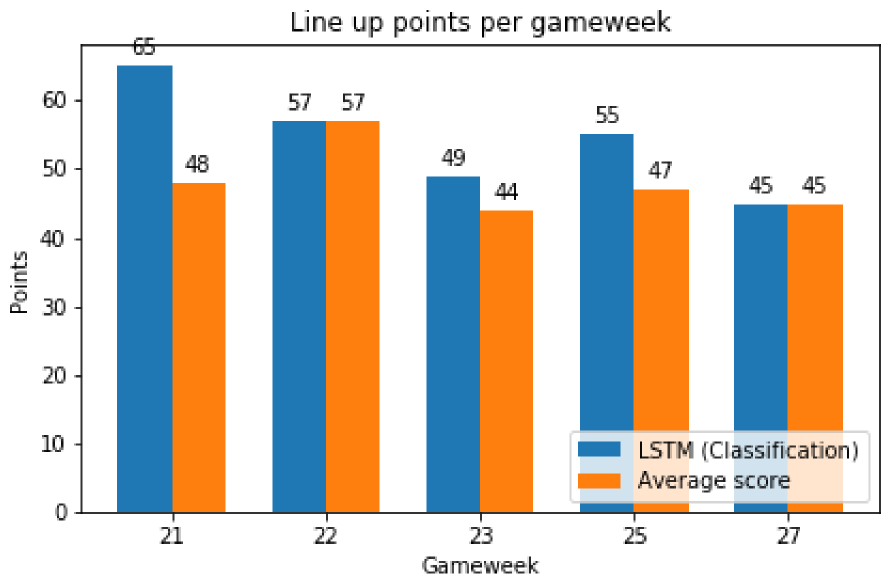


Figure 8: Points obtained from the line-ups generated by the LSTM-models for the five evaluation game weeks, compared to the average points at the same time

As seen in Figure 8, the LSTM-models for classification managed to beat the average Fantasy Premier League managers in 3/5 game weeks, and being on par with the average in the other two. For all selections for the five evaluation game weeks, see Appendix A.6.

The LSTM-model for classification performed similar to the MLP-model seen above.

However, when analyzing the evaluating game weeks, some interesting selections were found. Probably the most interesting one is the Arsenal-defender Sokratis Papastathopoulos in game week 21, who before the game only had played 76 minutes for the last three matches. The model ignored that data and selected Sokratis as a the first defender pick, in spite of the fact that Arsenal played a tough match at home against Manchester United. Sokratis were only selected by $\sim 1.2\%$ of the Fantasy Premier League managers and are not known for a defender with offensive capacity, but somehow during game week 21 he managed to both score a goal and keep a clean sheet, which resulted in 15 points.

Another impressive pick is the selection of the Chelsea-defender Reece James in game week 22, when Chelsea played at home versus Burnely. At the time, James was only selected by $\sim 0.2\%$ of all Fantasy Premier League managers, but the model selected the defender anyway as the third defender-pick. Chelsea won the game with 3-0 and James managed to get an assist, ending up with 11 points.

5.8 Classification comparision

	Goalkeepers	Defenders	Midfielders	Forwards	Average
SVM	68.20%	60.02%	68.37%	60.35%	64.23%
MLP	66.10%	**69.89%**	82.04%	**75.25%**	73.32%
LSTM	**70.00%**	69.70%	**82.78%**	75.00%	**74.37%**

Table 5.15: Comparision between the different classification approaches using accuracy on the test data (the higher, the better)

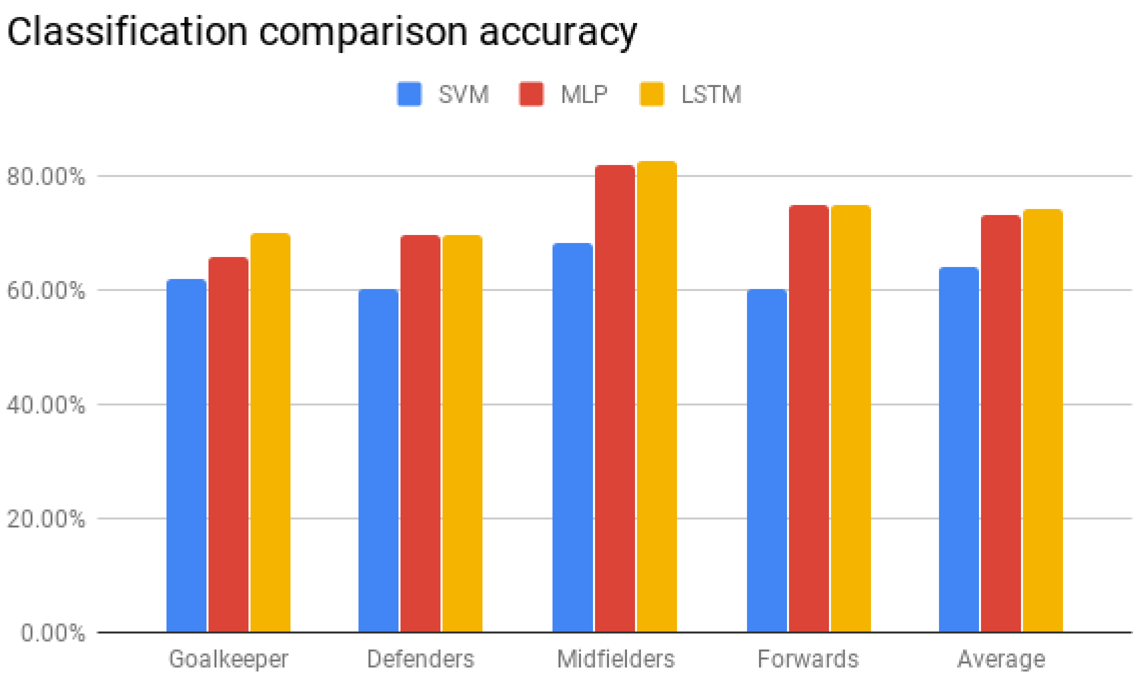


Figure 9: A graph displaying Table 5.15

The LSTM- and MLP-models are performing quite similar on the test data, with a slight but not significant advantage for LSTM.

5.9 Evaluation on game weeks

	GW 21	**GW 22**	**GW 23**	**GW 25**	**GW 27**	Average	Total
Game average	48	57	44	47	45	48.2	241
SVR (regression)	42	71	49	62	62	57.2	286
MLP (regression)	42	62	**69**	57	64	**58.8**	**294**
LSTM (regression)	55	**81**	38	**69**	50	58.6	293
SVM (classification)	43	48	61	57	**66**	55	275
MLP (classification)	52	58	58	56	56	54	280
LSTM (classification)	**65**	57	49	55	45	54.2	271

Table 5.16: All evaluations of the different approaches

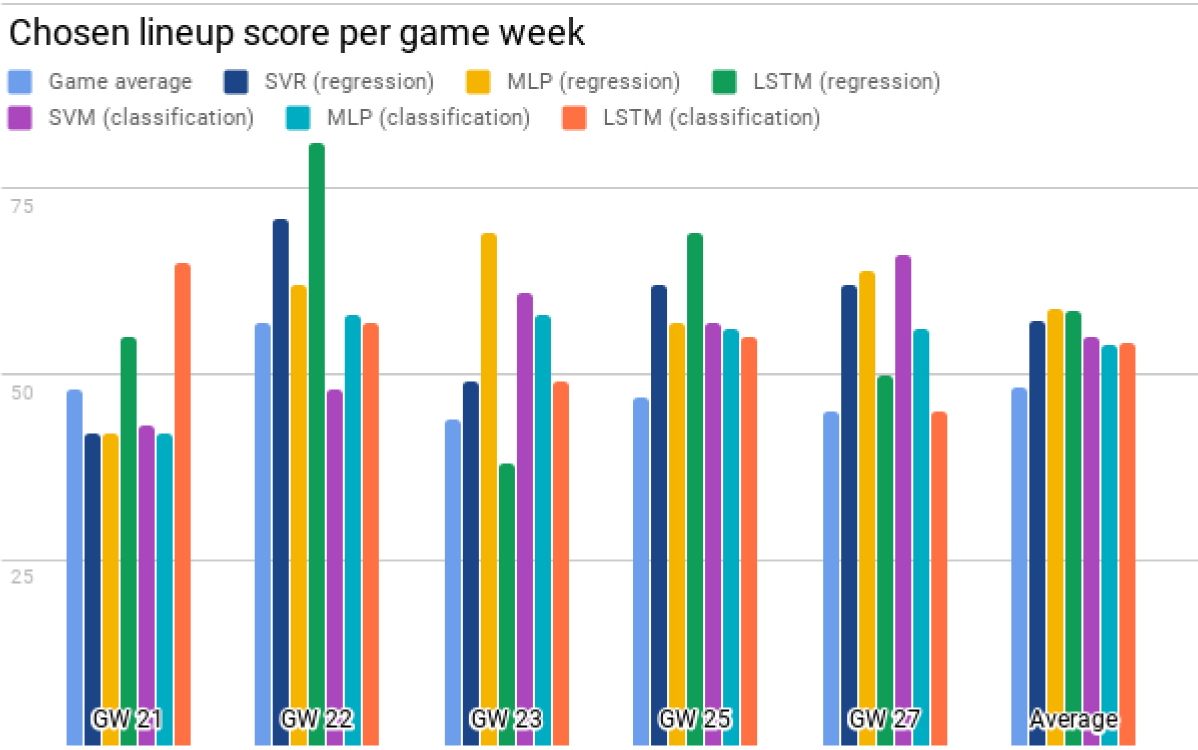


Figure 10: A graph displaying Table 5.16

As seen in Table 5.16, the regression approaches outperformed the classification approaches in terms of selecting players for a game week, with the MLP- and the LSTM-models being the best ones. As mentioned in Section 5.6, the classification approaches tend to obtain scores with a lower variance then the regression approaches, which comes naturally because of the way the problem is formulated.

When comparing the results to other managers the models are in the top 12k managers. In season 2019/20, 29 game weeks have been played, and the best manager has 1894p so far. 1894p in 29 game weeks is on average roughly 65p per game weeks. The best model, MLP with regression, reached an average of 58.8p during 5 game weeks. If extrapolated to 29 game weeks, it sums up to 1705p. This would be in the range of the top 12k managers or top 0,17%.

6

Related Work

King, 2017, predicted NFL quarterback performance using amongst others, support vector regression and principal component regression [18]. King found that using principal component analysis, 95% of the cumulative proportion of the variance could be explained with 6 principal components. King started with 50 features but later reduced it to 10, believing that a large number of features could be detrimental to the models as there was too much noise for the models to learn.

Hoerning, Moallemi, & Wilson, 2014, used logistic regression and support vector machines on predicting NFL quarterback performance [13]. They used rolling averages on their features, as their data, like this thesis, varies significantly from one game to the next. They found that using $m = 1$ and $m = 3$, where m is the number of previous games to use in their average, reported the best results on predicting the number of touchdowns. Hoerning et al., as opposed to King, did not seem to suffer from too much noise in their data set when using over 100 features.

Hamadani, 2005, used rolling averages to predict the winner of an NFL game with logistic regression and support vector regression [7]. Hamadani found that by using a SVM with a linear kernel gave the best results. Other kernels obtained similar results however some were significantly worse. Instead of using a small m, like Hoeming et al., Hamadani used all matches that have been played previously. If a game for week 16 was to be predicted, all weeks' averages up to week 15 were to be calculated. This thesis will also include rolling averages when modelling time-series data to a non-time-series model.

Ulmer & Fernandez, 2013, used, amongst other models, linear support vector machine to predict English Premier League match results [12]. Their data is similar to this thesis; thus, they suffer from some problems that will be encountered in this thesis. One of their problems is the high entropy, the value for randomness in their data, causing their models to be less accurate. They also used rolling averages when taking into account teams that were on a hot streak and found that using $m = 7$ as the best number of games to look back.

Timmaraju, Palnitkar, & Khanna, 2013, predicted English Premier League outcomes with a high entropy in their data and by measuring past performances [11]. They defined a teams performance by only three features: goals, shots on goal and corners. They trained on one season of data and tested on one season of data leading to a small data set. This thesis model will include the features Timmaraju et al. used as well as many others in order to create a more general model for EPL and football.

Lutz, 2015, performed predictions with promising results on the performance of NFL quarterbacks with SVR and MLP, despite using limited data [14].

Sierra, Fosco & Fierro, 2011, used support vector machines to predict a winning team with a high accuracy based on a small data set in the NFL [25]. They tried several different algorithms (linear, polynomial, gaussian and logistic regression) but found that the linear SVM had by far the best results.

Teich, Lutz & Kassarnig, 2016, studied how to predict the accuracy of a football play/choice made in the game [17]. By coming up with metrics such as "progress", they could predict the success of a play with high accuracy by using Decision Trees. Several machine learning algorithms ended up with similar results (Linear SVM, RBF SVM (radial basis function/gaussian) and SVD-LDA (singular value decomposition-linear discriminant analysis).

Many papers have in common that using a Linear SVM gives the most promising results, which is why it was chosen as a baseline for this thesis. Most of the research has been focused on the American football league NFL, where for example, King, Lutz and Sierra, Fosco & Fierro appoint similar techniques that this thesis will use but on different data sets.

7

Conclusion & Future Work

7.1 Conclusion

In this thesis, different machine learning models have been studied and benchmarked on their capability of predicting upcoming performances for football players. In total, three different approaches have been implemented and analyzed. All three have been applied to two problems: regression and classification. The models learned from past football matches played in the last four seasons of the English top division, Premier League. The point system used in the game Fantasy Premier League were used to evaluate how well a player performed. The models were fed with features regarding historical data regarding the player, the team and the opponent team as input.

Both regression and classification proved to be useful techniques for picking a good lineup for next match. All approaches achieved a better score in total on the five game weeks than the average manager in Fantasy Premier League managed to. By comparing the average and the top managers output to the model's output, the conclusion can be drawn the models can prove themselves useful in predicting performances based on match history. However, as mentioned before, the models will have an advantage in not being limited by a budget as the managers in Fantasy Premier League are. With that in mind, it is not enough to say that the models will always beat the average but can instead be used as an indicator on performance rather than a prediction on the exact truth.

By analyzing the evaluation game weeks further, it was found that regression approaches obtained higher points than the classification approaches in all cases. As mentioned before, the regression model takes higher risks by choosing the players with the highest potential performance rather than the player who is most likely to succeed. This should adequately explain the difference between the two methods. Thus with the two methods, more possibilities open up to use the models not only for best performances prediction but also for safest lineup prediction.

After a comparison between the test losses, it is clear that the LSTM model outperformed the rest of the models in both the regression and classification approach. The performance of the LSTM does not come as a shock, hinted by the way the problem was stated in Section 1.1. It was suspected that the time-based approach would benefit of the time-based data as compared to non-time based approaches like

SVM/SVR and MLP.

The test-loss obtained by the LSTM-model is still relatively high. With the limitation in mind mentioned in Section 1.2, that it seems impossible to predict with zero error how good a player will perform, the conclusion is made that it can instead be used to give a strong indication for an upcoming match. In summary, these findings can bring useful information for football managers responsible for picking the best lineup in their respective team by providing insights from historical data that might be complex to find for humans.

7.2 Future Work

With more data, the models' performance could be improved and generalized. In this thesis, only the last four seasons of the Premier League were used. By including more seasons and more leagues, the models would have more data to train and validate on. Sometimes multiple tournaments are played in parallel, such as some teams might play a Premier League game during the weekend, then a Champions League game in the middle of the week and then a FA Cup game the weekend ahead before their next Premier League game. In scenarios like these, the models currently only know about the data from the last Premier League game and are then ignoring the two latter games, even though they most likely will affect a player's performance in the upcoming game.

In this thesis, the models are compared only by their loss as well as comparing the score of the best lineups compared to the average Fantasy Premier League score. However, one could perhaps evaluate the models using other comparisons as well, since the average Fantasy Premier League score might be a good guideline of what to expect as a score, but does not give us anything more than an indication. The models could, for instance, be compared against the top 10% of the Fantasy Premier League managers, or some experts picking their best lineups as well.

It would also be interesting to compare more models against the one covered in this thesis. Some of the work discussed in Section 6 showed promising results when using principal component regression on American Football data, and therefore it would be interesting to see if that stands for this problem as well.

References

[1] Warren McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[2] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[3] Frank Rosenblatt. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. Tech. rep. Cornell Aeronautical Lab Inc Buffalo NY, 1961.

[4] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine learning* 20.3 (1995), pp. 273–297.

[5] Sepp Hochreiter and Jürgen Schmidhuber. “Long short-term memory”. In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[6] Vladimir Vapnik. “An overview of statistical learning theory”. In: *IEEE transactions on neural networks* 10.5 (1999), pp. 988–999.

[7] Babak Hamadani. “Predicting the outcome of NFL games using machine learning”. In: (2005). Accessed: 2020-06-15. URL: `http://cs229.stanford.edu/proj2006/BabakHamadani-PredictingNFLGames.pdf`.

[8] Christopher Bishop. *Pattern recognition and machine learning*. springer, 2006.

[9] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[10] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[11] Aditya Srinivas Timmaraju, Aditya Palnitkar, and Vikesh Khanna. *Game ON! Predicting English Premier League Match Outcomes*. Stanford university. 2013.

[12] Ben Ulmer, Matthew Fernandez, and Michael Peterson. *Predicting soccer match results in the English Premier League*. Stanford. 2013.

[13] Steven Hoerning, Bobak Moallemi, and Matthew Wilson. *Any Given Sunday: Forecasting NFL Quarterback Performance*. Stanford university. 2014.

[14] Roman Lutz. “Fantasy football prediction”. In: *arXiv preprint arXiv:1505.06918* (2015). URL: `https://arxiv.org/pdf/1505.06918`.

[15] Martın Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[17] Brendan Teich, Roman Lutz, and Valentin Kassarnig. "NFL Play Prediction". In: *arXiv preprint arXiv:1601.00574* (2016). URL: http://homepages.cae.wisc.edu/~ece539/project/f03/kahn.pdf.

[18] Nicholas Aaron King et al. "Predicting a Quarterback's Fantasy Football Point Output For Daily Fantasy Sports Using Statistical Models". UT Arlington. PhD thesis. 2017.

[19] Premier League. *Fantasy Premier League Rules*. 2019. URL: https://fantasy.premierleague.com/help/rules.

[20] vaastav. *Fantasy-Premier-League*. 2019. URL: https://github.com/vaastav/Fantasy-Premier-League.

[21] Premier League. *Fantasy Premier League*. 2020. URL: https://fantasy.premierleague.com.

[22] SofaScore. *SofaScore*. 2020. URL: https://www.sofascore.com/.

[23] Understat. *Understat*. 2020. URL: https://www.understat.com/.

[24] Wikipedia. *Levenshtein Distance*. Accessed: 2020-06-15. 2020. URL: https://en.wikipedia.org/wiki/Levenshtein_distance.

[25] Adrian Sierra et al. *Football futures*. Accessed: 2020-06-15. URL: http://cs229.stanford.edu/proj2011/SierraFoscoFierro-FootballFutures.pdf.

A

Appendix 1

A.1 SVR evaluation on gameweeks

Gameweek 21
Selected lineup points: 42
Goalkeeper

Player	Team	Predicted Points	Actual Points
Rui Patrício	Wolverhampton	4.23	1

Defenders

Player	Team	Predicted Points	Actual Points
Adam Webster	Brighton	3.89	2
Trent Alexander-Arnold	Liverpool	4.11	6
Virgil van Dijk	Liverpool	4.54	7

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.29	10
Sadio Mané	Liverpool	4.34	8
Raheem Sterling	Manchester City	4.40	1
Kevin De Bruyne	Manchester City	4.80	2

Forwards

Player	Team	Predicted Points	Actual Points
Neal Maupay	Brighton	4.43	1
Marcus Rashford	Manchester United	4.63	2
Raúl Jiménez	Wolverhampton	4.76	2

Gameweek 22
Selected lineup points: 71

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	6.25	2

Defenders

Player	Team	Predicted Points	Actual Points
Lucas Digne	Everton	3.94	12
Harry Maguire	Manchester United	4.21	6
Ricardo Pereira	Leicester City	4.41	1

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.25	6
Sadio Mané	Liverpool	4.37	3
Kevin De Bruyne	Manchester City	4.60	9
Riyad Mahrez	Manchester City	4.81	17

Forwards

Player	Team	Predicted Points	Actual Points
Marcus Rashford	Manchester United	4.67	12
Andy Carroll	Newcastle	5.07	1
Raúl Jiménez	Wolverhampton	5.34	2

Gameweek 23
Selected lineup points: 49

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	6.70	1

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	4.45	6
Trent Alexander-Arnold	Liverpool	4.52	10
Virgil van Dijk	Liverpool	4.53	15

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.12	7
Sadio Mané	Liverpool	4.27	3
Kevin De Bruyne	Manchester City	4.53	2
Riyad Mahrez	Manchester City	4.67	1

Forwards

Player	Team	Predicted Points	Actual Points
Callum Wilson	Bournemouth	4.18	1
Jordan Ayew	Crystal Palace	4.28	1
Danny Ings	Southampton	4.76	2

Gameweek 25
Selected lineup points: 62

Goalkeeper

Player	Team	Predicted Points	Actual Points
Alisson Becker	Liverpool	5.32	8

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	3.84	6
Virgil van Dijk	Liverpool	3.86	6
Matt Doherty	Wolverhampton	4.03	6

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.10	16
Riyad Mahrez	Manchester City	4.28	2
Anthony Martial	Manchester United	4.44	3
Dele Alli	Tottenham	4.54	3

Forwards

Player	Team	Predicted Points	Actual Points
Chris Wood	Burnley	4.24	2
Troy Deeney	Watford	4.27	5
Gerard Deulofeu	Watford	5.14	5

Gameweek 27

Selected lineup points: 62

Goalkeeper

Player	Team	Predicted Points	Actual Points
Lukasz Fabianski	West Ham	4.50	2

Defenders

Player	Team	Predicted Points	Actual Points
Marcos Alonso	Chelsea	4.13	8
Trent Alexander-Arnold	Liverpool	4.21	10
Virgil van Dijk	Liverpool	4.42	1
Matt Doherty	Wolverhampton	4.44	10

Midfielders

Player	Team	Predicted Points	Actual Points
Harvey Barnes	Leicester City	4.13	1
Mohamed Salah	Liverpool	4.20	7
Kevin De Bruyne	Manchester City	4.37	3

Forwards

Player	Team	Predicted Points	Actual Points
Roberto Firmino	Liverpool	4.58	2
Danny Ings	Southampton	4.87	2
Diogo Jota	Wolverhampton	5.73	16

A.2 MLP (regression) evaluation on gameweeks

Gameweek 21
Selected lineup points: 42

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ben Foster	Watford	3.39	2

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	4.75	12
Trent Alexander-Arnold	Liverpool	5.48	6
João Cancelo	Machester City	5.70	2

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.90	10
Raheem Sterling	Manchester City	4.97	1
Kevin De Bruyne	Manchester City	5.20	2
Anthony Martial	Manchester United	5.28	2

Forwards

Player	Team	Predicted Points	Actual Points
Pierre-Emerick Aubameyang	Arsenal	5.50	2
Neal Maupay	Brighton	5.80	1
Raúl Jiménez	Wolverhampton	6.00	2

Gameweek 22
Selected lineup points: 62

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	3.42	2

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	5.03	6
Trent Alexander-Arnold	Liverpool	5.14	6
John Lundstram	Sheffield United	5.15	6

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.69	6
Kevin De Bruyne	Manchester City	4.77	9
Riyad Mahrez	Manchester City	4.91	17
Heung-Min Son	Tottenham	5.26	2

Forwards

Player	Team	Predicted Points	Actual Points
Dominic Calvert-Lewin	Everton	4.78	1
Jamie Vardy	Leicester	6.16	5
Raúl Jiménez	Wolverhampton	7.42	2

Gameweek 23
Selected lineup points: 69

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	3.48	1

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	5.84	6
Trent Alexander-Arnold	Liverpool	5.89	10
Virgil van Dijk	Liverpool	6.46	15

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.55	7
Sadio Mané	Liverpool	4.73	3
Raheem Sterling	Manchester City	4.78	2
Kevin De Bruyne	Manchester City	5.07	2
Riyad Mahrez	Manchester City	5.33	1

Forwards

Player	Team	Predicted Points	Actual Points
Sergio Agüero	Manchester City	4.81	13
Teemu Pukki	Norwich City	5.76	9

Gameweek 25

Selected lineup points: 57

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	3.48	1

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	3.95	6
Trent Alexander-Arnold	Liverpool	4.21	6
Japhet Tanganga	Tottenham	5.01	6

Midfielders

Player	Team	Predicted Points	Actual Points
Mohamed Salah	Liverpool	4.89	16
Kevin De Bruyne	Manchester City	4.95	2
Riyad Mahrez	Manchester City	4.99	2
Anthony Martial	Manchester United	5.00	3
Heung-Min Son	Tottenham	5.03	8

Forwards

Player	Team	Predicted Points	Actual Points
Sergio Agüero	Manchester City	4.39	2
Gerard Deulofeu	Watford	5.14	5

Gameweek 27
Selected lineup points: 64

Goalkeeper

Player	Team	Predicted Points	Actual Points
Kasper Schmeichel	Leicester City	3.38	12

Defenders

Player	Team	Predicted Points	Actual Points
James Tarkowski	Burnley	4.35	6
Marcos Alonso	Chelsea	4.54	8
Matt Doherty	Wolverhampton	4.83	10

Midfielders

Player	Team	Predicted Points	Actual Points
Jack Grealish	Aston Villa	4.67	1
Mohamed Salah	Liverpool	4.87	7
Kevin De Bruyne	Leicester City	5.09	3
Anthony Martial	Manchester United	5.15	8

Forwards

Player	Team	Predicted Points	Actual Points
Jamie Vardy	Leicester City	4.53	2
Danny Ings	Southampton	5.00	2
Raúl Jiménez	Wolverhampton	7.05	5

A.3 LSTM (regression) evaluation on gameweeks

Gameweek 21
Selected lineup points: 55

Goalkeeper

Player	Team	Predicted Points	Actual Points
Claudio Bravo	Manchester City	4.67	2

Defenders

Player	Team	Predicted Points	Actual Points
João Cancelo	Machester City	4.23	2
Virgil van Dijk	Liverpool	4.11	7
Trent Alexander-Arnold	Liverpool	5.38	6

Midfielders

Player	Team	Predicted Points	Actual Points
Sadio Mané	Liverpool	5.54	8
Jack Grealish	Aston Villa	5.60	13
Raheem Sterling	Manchester City	5.62	1
Kevin De Bruyne	Manchester City	5.97	2
Mohamed Salah	Liverpool	6.02	10

Forwards

Player	Team	Predicted Points	Actual Points
Roberto Firmino	Liverpool	5.62	2
Marcus Rashford	Manchester United	5.67	2

Gameweek 22

Selected lineup points: 81

Goalkeeper

Player	Team	Predicted Points	Actual Points
Kepa Arrizabalaga	Chelsea	5.27	6

Defenders

Player	Team	Predicted Points	Actual Points
John Lundstram	Sheffield United	4.75	6
Enda Stevens	Sheffield United	5.05	8
Antonio Rüdiger	Chelsea	5.06	6

Midfielders

Player	Team	Predicted Points	Actual Points
Willian	Chelsea	5.51	6
Anthony Martial	Manchester United	5.58	8
Mohamed Salah	Liverpool	5.75	6
Kevin De Bruyne	Manchester City	5.86	9

Forwards

Player	Team	Predicted Points	Actual Points
Roberto Firmino	Liverpool	5.59	9
Marcus Rashford	Manchester United	5.84	12
Jamie Vardy	Leicester City	5.85	5

Gameweek 23
Selected lineup points: 38

Goalkeeper

Player	Team	Predicted Points	Actual Points
Ederson	Manchester City	4.38	1

Defenders

Player	Team	Predicted Points	Actual Points
Lewis Dunk	Brighton	4.84	1
João Cancelo	Manchester City	4.92	1
Trent Alexander-Arnold	Liverpool	5.70	10

Midfielders

Player	Team	Predicted Points	Actual Points
Raheem Sterling	Manchester City	5.53	2
Kevin De Bruyne	Manchester City	5.90	2
Sadio Mané	Liverpool	5.98	3
Mohamed Salah	Liverpool	6.02	7

Forwards

Player	Team	Predicted Points	Actual Points
Teemu Pukki	Norwich City	5.63	9
Jamie Vardy	Leicester City	5.83	0
Roberto Firmino	Liverpool	5.89	2

Gameweek 25

Selected lineup points: 69

Goalkeeper

Player	Team	Predicted Points	Actual Points
Alisson	Liverpool	4.99	8

Defenders

Player	Team	Predicted Points	Actual Points
Andrew Robertson	Liverpool	4.55	6
Virgil van Dijk	Liverpool	4.58	6
Trent Alexander-Arnold	Liverpool	5.49	6

Midfielders

Player	Team	Predicted Points	Actual Points
Raheem Sterling	Manchester City	5.53	1
Heung-Min Son	Tottenham	5.55	8
Anthony Martial	Manchester United	5.64	3
Kevin De Bruyne	Manchester City	5.65	2
Mohamed Salah	Liverpool	6.04	16

Forwards

Player	Team	Predicted Points	Actual Points
Teemu Pukki	Norwich City	5.41	2
Roberto Firmino	Liverpool	6.00	11

Gameweek 27
Selected lineup points: 50

Goalkeeper

Player	Team	Predicted Points	Actual Points
Alisson	Liverpool	4.96	1

Defenders

Player	Team	Predicted Points	Actual Points
Virgil van Dijk	Liverpool	5.55	1
Andrew Robertson	Liverpool	5.77	4
Trent Alexander-Arnold	Liverpool	6.27	10

Midfielders

Player	Team	Predicted Points	Actual Points
Kevin De Bruyne	Manchester City	5.62	3
Anthony Martial	Manchester United	5.69	8
Adama Traoré	Wolverhampton	5.85	1
Sadio Mané	Liverpool	5.94	8
Mohamed Salah	Liverpool	6.12	7

Forwards

Player	Team	Predicted Points	Actual Points
Raúl Jiménez	Wolverhamton	5.88	5
Roberto Firmino	Liverpool	6.17	2

A.4 SVM (classification) evaluation on gameweeks

Gameweek 21
Selected lineup points: 43

Goalkeeper

Player	Team	Actual Points
Alisson Becker	Liverpool	6

Defenders

Player	Team	Actual Points
Martín Montoya	Brighton	2
Dan Burn	Brighton	1
Adam Webster	Brighton	2
Andrew Robertson	Liverpool	12
João Cancelo	Machester City	2

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	10
Kevin De Bruyne	Manchester City	2
Anthony Martial	Manchester United	2

Forwards

Player	Team	Actual Points
Marcus Rashford	Manchester United	2
Harry Kane	Tottenham	2

Gameweek 22
Selected lineup points: 48

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
Çaglar Söyüncü	Leicester City	1
John Stones	Manchester City	1
John Lundstram	Sheffield United	6
Ricardo Pereira	Leicester City	1

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	6
Kevin De Bruyne	Manchester City	9
Riyad Mahrez	Manchester City	17

Forwards

Player	Team	Actual Points
Kelechi Iheanacho	Leicester City	1
Andy Carroll	Newcastle	1
Raúl Jiménez	Wolverhampton	2

Gameweek 23

Selected lineup points: 61

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
Andrew Robertson	Liverpool	6
Trent Alexander-Arnold	Liverpool	10
Virgil van Dijk	Liverpool	15
Benjamin Mendy	Manchester City	5
João Cancelo	Manchester City	1

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	7
Sadio Mané	Liverpool	3
Kevin De Bruyne	Manchester City	2

Forwards

Player	Team	Actual Points
Tammy Abraham	Chelsea	2
Teemu Pukki	Norwich City	9

Gameweek 25
Selected lineup points: 57

Goalkeeper

Player	Team	Actual Points
Alisson Becker	Liverpool	8

Defenders

Player	Team	Actual Points
Andrew Robertson	Liverpool	6
Virgil van Dijk	Liverpool	6
Japhet Tanganga	Tottenham	6
Matt Doherty	Wolverhampton	6
João Cancelo	Manchester City	1

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	16
Kevin De Bruyne	Manchester City	2
Riyad Mahrez	Manchester City	2

Forwards

Player	Team	Actual Points
Chris Wood	Burnley	2
Teemu Pukki	Norwich City	2

Gameweek 27
Selected lineup points: 66

Goalkeeper

Player	Team	Actual Points
Alisson Becker	Liverpool	2

Defenders

Player	Team	Actual Points
Phil Bardsley	Burnley	10
Marcos Alonso	Chelsea	8
Virgil van Dijk	Liverpool	1
Matt Doherty	Wolverhampton	10
Jonny	Wolverhampton	6

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	7
Kevin De Bruyne	Manchester City	3

Forwards

Player	Team	Actual Points
Callum Wilson	Bournemouth	2
Danny Ings	Southampton	2
Diogo Jota	Wolverhampton	16

A.5 MLP (classification) evaluation on gameweeks

Gameweek 21
Selected lineup points: 52

Goalkeeper

Player	Team	Actual Points
Alisson Becker	Liverpool	6

Defenders

Player	Team	Actual Points
Dan Burn	Brighton	1
Andrew Robertson	Liverpool	12
Trent Alexander-Arnold	Liverpool	6
João Cancelo	Machester City	2

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	10
Sadio Mané	Liverpool	8
Raheem Sterling	Manchester City	1
Anthony Martial	Manchester United	2

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	2
Marcus Rashford	Manchester United	2

Gameweek 22
Selected lineup points: 58

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
Steve Cook	Bournemouth	1
Michael Keane	Everton	6
Fabian Delph	Everton	1
Andrew Robertson	Liverpool	6
João Cancelo	Machester City	2

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	6
Sadio Mané	Liverpool	3

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	9
Marcus Rashford	Manchester United	12
Gabriel Jesus	Machester City	10

Gameweek 23
Selected lineup points: 58

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
Andrew Robertson	Liverpool	6
Trent Alexander-Arnold	Liverpool	10
Virgil van Dijk	Liverpool	15
John Stones	Manchester City	1
João Cancelo	Manchester City	1

Midfielders

Player	Team	Actual Points
Sadio Mané	Liverpool	3
Raheem Sterling	Manchester City	2

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	2
Sergio Agüero	Manchester City	13
Gabriel Jesus	Manchester City	4

Gameweek 25
Selected lineup points: 56

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	1

Defenders

Player	Team	Actual Points
Andrew Robertson	Liverpool	6
Virgil van Dijk	Liverpool	6
Nicolás Otamendi	Manchester City	1

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	16
Raheem Sterling	Manchester City	1
Anthony Martial	Manchester United	3
Heung-Min Son	Tottenham	8

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	11
Sergio Agüero	Manchester City	2
Gabriel Jesus	Manchester City	1

Gameweek 27
Selected lineup points: 56

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	7

Defenders

Player	Team	Actual Points
Phil Bardsley	Burnley	10
Trent Alexander-Arnold	Liverpool	10
Joe Gomez	Liverpool	1
Joël Matip	Liverpool	1
David Luiz	Arsenal	4

Midfielders

Player	Team	Actual Points
Mohamed Salah	Liverpool	7
Anthony Martial	Manchester United	8

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	2
Sergio Agüero	Manchester City	0
Gabriel Jesus	Manchester City	6

A.6 LSTM (classification) evaluation on gameweeks

Gameweek 21
Selected lineup points: 65

Goalkeeper

Player	Team	Actual Points
Claudio Bravo	Manchester City	2

Defenders

Player	Team	Actual Points
Benjamin Chilwell	Leicester City	6
Harry Maguire	Manchester United	1
Çaglar Söyüncü	Leicester City	8
Lewis Dunk	Brighton	4
Sokratis Papastathopoulos	Arsenal	15

Midfielders

Player	Team	Actual Points
Anthony Martial	Manchester United	2
Jack Grealish	Aston Villa	13
Mohamed Salah	Liverpool	10
Kevin De Bruyne	Manchester City	2

Forwards

Player	Team	Actual Points
Raúl Jiménez	Wolverhamton	2

Gameweek 22
Selected lineup points: 57

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
John Stones	Machester City	2
Virgil van Dijk	Liverpool	6
Reece James	Chelsea	11
Andreas Christensen	Chelsea	6
Antonio Rüdiger	Chelsea	6

Midfielders

Player	Team	Actual Points
Adama Traoré	Wolverhampton	2
Ayoze Pérez	Leicester City	2
Mohamed Salah	Liverpool	6
Kevin De Bruyne	Manchester City	9

Forwards

Player	Team	Actual Points
Jamie Vardy	Leicester City	5

Gameweek 23
Selected lineup points: 49

Goalkeeper

Player	Team	Actual Points
Ederson	Manchester City	2

Defenders

Player	Team	Actual Points
César Azpilicueta	Chelsea	2
Mason Holgate	Everton	8
Benjamin Mendy	Manchester City	5
Reece James	Chelsea	8
Trent Alexander-Arnold	Liverpool	10

Midfielders

Player	Team	Actual Points
Riyad Mahrez	Manchester City	1
Kevin De Bruyne	Manchester City	2
Mohamed Salah	Liverpool	7
Sadio Mané	Liverpool	3

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	2

Gameweek 25
Selected lineup points: 55

Goalkeeper

Player	Team	Actual Points
Alisson	Liverpool	8

Defenders

Player	Team	Actual Points
Patrick van Aanholt	Crystal Palace	2
Lewis Dunk	Brighton	1
Gary Cahill	Crystal Palace	2
Adam Smith	Bournemouth	2
Adam Webster	Brighton	1

Midfielders

Player	Team	Actual Points
Kevin De Bruyne	Manchester City	2
Riyad Mahrez	Manchester City	2
Heung-Min Son	Tottenham	8
Mohamed Salah	Liverpool	16

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	11

Gameweek 27
Selected lineup points: 45

Goalkeeper

Player	Team	Actual Points
Alisson	Liverpool	1

Defenders

Player	Team	Actual Points
Joe Gomez	Liverpool	1
Conor Coady	Wolverhampton	5
Benjamin Mendy	Manchester City	6
Shkodran Mustafi	Arsenal	1
Trent Alexander-Arnold	Liverpool	10

Midfielders

Player	Team	Actual Points
Kevin De Bruyne	Manchester City	3
Adama Traoré	Wolverhampton	1
Sadio Mané	Liverpool	8
Mohamed Salah	Liverpool	7

Forwards

Player	Team	Actual Points
Roberto Firmino	Liverpool	2

A.7 Fantasy Premier League Point System

Figure 1: Fantasy Premier League Point System

Action	Points
For playing up to 60 minutes	1
For playing 60 minutes or more (excluding stoppage time)	2
For each goal scored by a goalkeeper or defender	6
For each goal scored by a midfielder	5
For each goal scored by a forward	4
For each goal assist	3
For a clean sheet by a goalkeeper or defender	4
For a clean sheet by a midfielder	1
For every 3 shot saves by a goalkeeper	1
For each penalty save	5
For each penalty miss	-2
Bonus points for the best players in a match	1-3
For every 2 goals conceded by a goalkeeper or defender	-1
For each yellow card	-1
For each red card	-3
For each own goal	-2

Figure 2: Fantasy Premier League Bonus Point System

The Bonus Points System (BPS) utilises a range of statistics to create a BPS score for every player. The three best performing players in each match will be awarded bonus points. 3 points will be awarded to the highest scoring player, 2 to the second best and 1 to the third.

Examples of how bonus point ties will be resolved are as follows:

- If there is a tie for first place, Players 1 & 2 will receive 3 points each and Player 3 will receive 1 point.
- If there is a tie for second place, Player 1 will receive 3 points and Players 2 and 3 will receive 2 points each.
- If there is a tie for third place, Player 1 will receive 3 points, Player 2 will receive 2 points and Players 3 & 4 will receive 1 point each.

How is the BPS score calculated?

Players score BPS points based on the following statistics (one point for each unless otherwise stated):

Action	BPS
Playing 1 to 60 minutes	3
Playing over 60 minutes	6
Goalkeepers and defenders scoring a goal	12
Midfielders scoring a goal	18
Forwards scoring a goal	24
Assists	9
Goalkeepers and defenders keeping a clean sheet	12
Saving a penalty	15
Save	2
Successful open play cross	1
Creating a big chance (a chance where the receiving player should score)	3
For every 2 clearances, blocks and interceptions (total)	1
For every 3 recoveries	1
Key pass	1
Successful tackle (net*)	2
Successful dribble	1
Scoring the goal that wins a match	3
70 to 79% pass completion (at least 30 passes attempted)	2
80 to 89% pass completion (at least 30 passes attempted)	4
90%+ pass completion (at least 30 passes attempted)	6
Conceding a penalty	-3
Missing a penalty	-6
Yellow card	-3
Red card	-9
Own goal	-6
Missing a big chance	-3
Making an error which leads to a goal	-3
Making an error which leads to an attempt at goal	-1
Being tackled	-1
Conceding a foul	-1
Being caught offside	-1
Shot off target	-1