

Prediction of Football Games Results

Roman Nestoruk¹ and Grzegorz Słowiński²

¹ Sollers Consulting Sp. z o.o., ul. Koszykowa 54, Warsaw 00-675, Poland

² University of Technology and Economics, Engineering Department, ul. Jagiellońska 82f, Warsaw 03-301, Poland

Abstract

For creation of 3 machine learning models, dataset of 50, 100 and 200 games are being used. All the models are built, using deep learning (DL) and machine technology (ML) technique with the goal to prove, that even ML algorithms can be used to predict football games result. The data set consists of different real games results, collected from the most recognizable tournaments, such as: English Premier League, Italian Seria A, German Bundesliga, Spanish La Liga and French League 1. The target values of the work are prediction of exact game score (Average accuracy obtained after the last wave of testing – 11.6%) and prediction of game result (Average accuracy obtained after the last wave of testing – 39%).

Keywords

machine learning, football games prediction, deep learning

1. Introduction

Mainly, the regular person thinking that football is unpredictable and sometimes, analogical game, but we are living in the 21st century, where technologies have become one of the biggest parts of our lives.

We are using virtual assistance, image and voice recognition, autopilots, we almost meet the era of self-driving cars. The brain of all these discoveries is Artificial Intelligence, with neural networks inside. We think these technologies are very helpful for achieving the main target of this work – proving that even football, where every match consists of thousands of different moments, can be predicted by Artificial Intelligence better than by benchmark.

2. Used Tools and technology

As football statistic is not available in the format of data files, or API communication response, scraping algorithm is needed. To not enhance existing stack with extra languages, scraping algorithm was written in Python and with use of Selenium Web Driver framework & BeautifulSoup4 library. For machine learning processes TensorFlow and keras frameworks has been used and CSV library for storing data.

3. Data for training and validation

One of the most recognized kinds of statistics in football games are possession and shots, but for this algorithm, some more data are also useful:

- Average game mark: Shows the performance of the team, during the season.
- The average amount of goals, per game: Result of dividing the number of goals, scored by the look at team, by the number of played games.
- Average possession: Average percentage of possession of the ball during the games.
- Pass accuracy: Counting by diving number of all successfully completed passes, by the number of all passes of the team.
- Shots per game: Anyone, who is connected to football knows, that goals are mainly the result of shots.
- Average players mark from most possible starting line up: Shows the performance of every single player, during the season.

Southampton Formations Summary

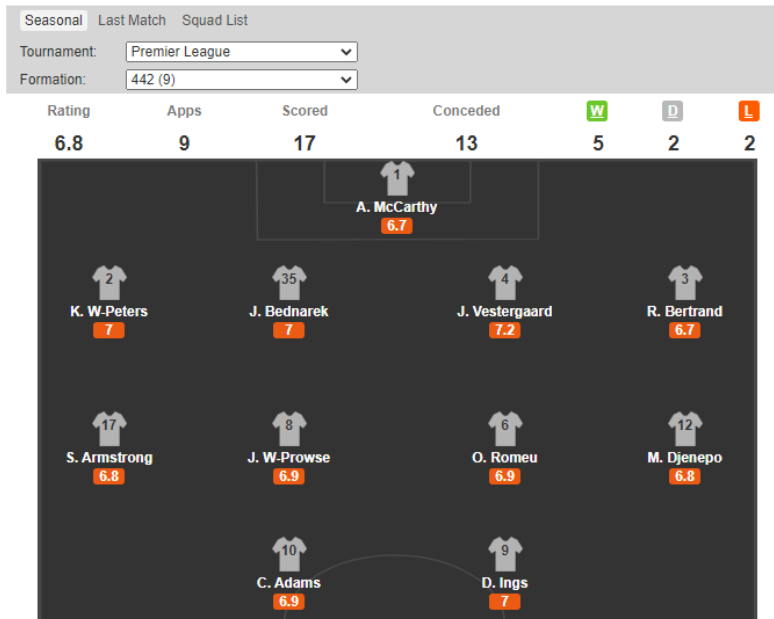


Figure 1: Table with player's mark

4. Model creation

For this experiment, model with 3 dense layers is being used. As shown on figure 2, model is consisting of:

Layer (type)	Output Shape	Param #
dense_9 (Dense)	(None, 56)	3192
dense_10 (Dense)	(None, 28)	1596
dense_11 (Dense)	(None, 2)	58

Total params: 4,846
 Trainable params: 4,846
 Non-trainable params: 0

Figure 2: Model summary

- Hidden layer 1
Consist of 56 units, with RELU activation.
- Hidden layer 2
Consist of 28 units, with RELU activation.
- Output layer
Consist just of 2 units, with Linear activation.

ReLU Function

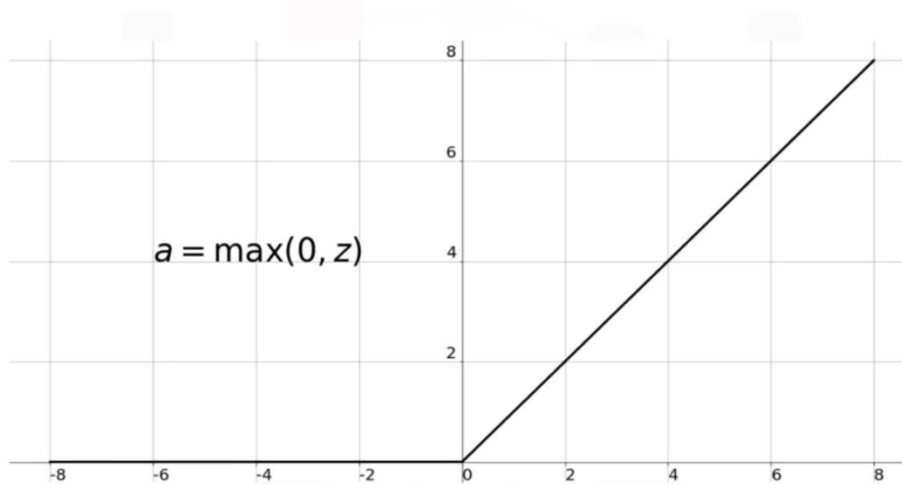


Figure 3: ReLU activation function graph [Source: 8]

For the first two layers, RELU activation helps to decrease all negative values, as team can not score -1 goals.

4.1. Data preparation

Considering that almost never in football one team is scoring more than 10 goals, expected result was transformed to the format of 0-1 value by dividing it by 10. For example: Actual score: 1:3, score after transformation: 0.1:0.3.

To be able, to better validate the result, extra 10% of the data was used for testing and validation of the model.

5. Models structure and usage

As a result of experiment, 3 models where created:

- Model 1: Trained on 50 examples of games with no unexpected result and validated on 5 extra examples.
- Model 2: Trained on 100 examples of games with no unexpected result and validated on 10 extra examples.
- Model 3: Trained on 200 examples and validated on 20 extra examples.

To make a summary, how effective are models in daily games prediction. Upcoming days games statistic were taken as input data for model.

```
'[[6.72, 1.0, 0.0, 4.0, 0.0, 0.0, 0.0, 0.0, 3.0, 0.0, 3.0, 1.0, 2.0, 1.0, 45.1, 75.0, 9.7, 6.7, 6.7, 6.8, 6.9, 6.8, 6.9, 6.8, 6.9, 6.6, 7.1, 6.8, 6.78, 2.0, 3.0, 1.0, 1.0, 1.0, 1.0, 0.0, 3.0, 1.0, 1.0, 2.0, 0.0, 1.6, 51.8, 81.2, 12.2, 6.8, 7.2, 6.8, 6.6, 6.6, 6.9, 7.3, 6.9, 7.1, 7.2, 7.5]]'
```

```
'[[7.01, 1.0, 0.0, 0.0, 2.0, 3.0, 0.0, 5.0, 0.0, 1.0, 2.0, 2.0, 0.0, 2.5, 62.0, 89.2, 19.1, 6.7, 6.9, 6.8, 7.4, 6.9, 6.9, 8.0, 7.5, 7.7, 8.2, 7.2, 7.0, 0.0, 3.0, 0.0, 2.0, 2.0, 1.0, 2.0, 3.0, 0.0, 0.0, 4.0, 0.0, 2.3, 59.5, 84.2, 15.6, 6.7, 7.3, 7.1, 7.2, 7.0, 7.4, 7.1, 6.8, 7.2, 7.5, 7.5]]'
```

Figure 4: Example of information, used for the result prediction

To simplify the process of validation, result of models prediction is stored table with following format:

- Team playing home as t1
- Team playing away as t1
- Number of goals predicted by the first model, for the home teams as m1t1
- Number of goals predicted by the first model, for the away teams as m1t2
- Number of goals predicted by the second model, for the home teams as m2t1
- Number of goals predicted by the second model, for the away teams as m2t2

[Введите текст]

- Number of goals predicted by the last model, for the home teams as m3t1
- Number of goals predicted by the last model, for the away teams as m3t2

```
[ 'Sheffield United', 'VS', 'Tottenham', 0.13859384, 0.18197364, 0.035843644, 0.12820777, 0.071412474, 0.16212471]
[ 'Manchester City', 'VS', 'Liverpool', 0.21726868, 0.18632217, 0.1908742, 0.07888002, 0.18883097, 0.103400104]
[ 'Real Sociedad', 'VS', 'Espanyol', 0.09177163, 0.1417118, 0.08841749, 0.10744436, 0.12775113, 0.12789574]
[ 'Eibar', 'VS', 'Osasuna', 0.09043123, 0.040207393, 0.09134838, 0.086297564, 0.070646524, 0.0732429]
[ 'Real Madrid', 'VS', 'Getafe', 0.101611726, 0.1555832, 0.17641327, 0.11048023, 0.21594301, 0.051858913]
[ 'Atalanta', 'VS', 'Napoli', 0.22257084, 0.2072522, 0.20862636, 0.16690905, 0.2572292, 0.07895858]
[ 'Roma', 'VS', 'Udinese', 0.13141762, 0.3416745, 0.073058605, 0.013992556, 0.2742979, 0.098182626]
[ 'Hull', 'VS', 'Middlesbrough', -0.033674814, 0.11631444, 0.011971727, 0.14647077, -0.0018174147, 0.12942076]
```

Figure 5: Data stored into the “results” variable.

6. Models evaluation

After all matches, we were interested in, have been finished. We can start comparing our predictions with the actual result. To simplify the result verification, we should transfer output of DL model to the integer, for that purpose, values were multiplied by. As a standard for this transformation, regular rules of rounding were used:

- Values are less than integer and half will be rounded to closes lower integer. For example: 1.5252->2, 2.9842->3, 0.5->1
- Values are less then integer and half, will be rounded to closes lower integer. For example: 0.4999->0, 1.1->1, 2.332->2.

Following these rules, a result like 0.5 vs 0.49 will be considered as 1 vs 0, but a result of 1.49 vs 0.5 will be considered as 1 vs 1.

The most known kind of prediction is a white guessing. Considering that probability of randomly guessing the result of any football game is 1 by the amount of possible – 3 (Winning of the home team, draw, or winning of away team), technically it is 33.3%.

The possibility of predicting the exact score of the game is more complicated because all possible combinations of the score should be considered. The chance of scoring more than 4 goals is too small, to be considered. So, to calculate the chance of predicting the exact score of the game, we should calculate the combination of 5 elements (score from 0 to 4) into 2 places (for 2 playing teams, home and away). We can calculate it by using the formula 6 – 1. The result of this calculation gives as 15 and the chance of prediction of the exact score of the game is 1 by 15 or 6.7%.

$$C(n + r - 1, r) = \frac{(n + r - 1)!}{r! (n - 1)!}$$

Formula 6: Combination with repetition

6.1. Model 1 evaluation

As mentioned in Chapter 6, model 1 was trained on the smallest amount of real data – 50 games.

- The average percentage of the predicted exact score of the games is on the level 10.3%. It's around 1.5 times more than mathematical chances to predict it. Of course, 0 predicted games for the 02-07-20 is looking not promising, but we had a very small amount of data to predict. Next days, this amount increased and was more than 2 times greater than the mathematical probability.
- The average percentage of the predicted winner or draw category is much higher. Of course, in Picture 8.2 we can see that the first day was failed again. Next days we can see results 4 and 3 times higher than on the 02-07-20, but this time average is below mathematical.

Day	Predicted exact score of the game	Percentage exact score	Predicted winner or draw	Percentage
02-07-20	0/8	0.0	1/8	12.5
11-07-20	4/25	16.0	12/25	48.0
12-07-20	3/20	15.0	6/20	30.0
Summary	7/53	10.333333333333334	19/53	30.166666666666668

Figure 7: Statistic of prediction from model 1

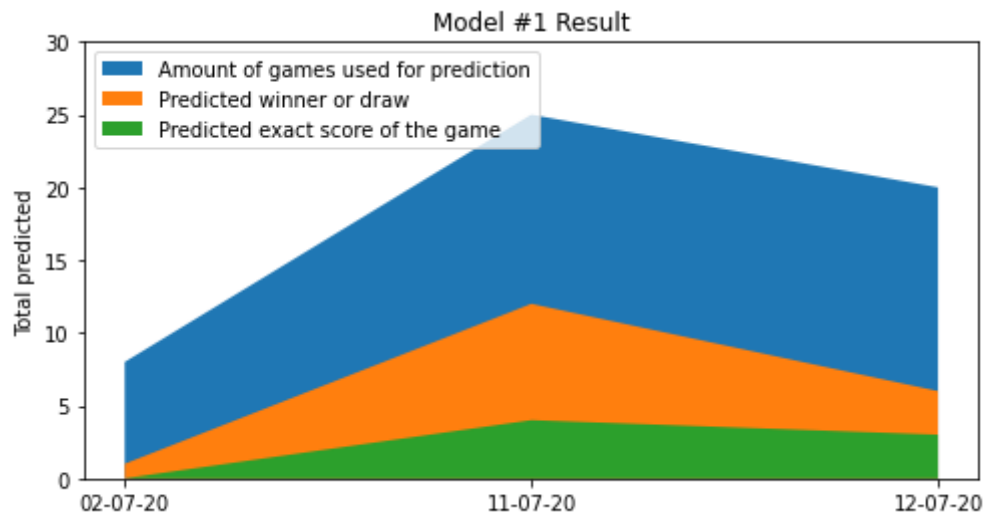


Figure 8: Diagram for the statistic of prediction from model 1

6.2. Model 2 evaluation

Model 2 was trained on the higher amount of real data – 100 games.

- This time we can see good progress on the average percent of predicted games, mainly because of the predicted games from day 02-07-20. The average percentage for the exact score category was much more stable and stands on the level 2 times more than mathematical probability.
- For the Predicted winner or draw category we can again see a big difference on the first row, but almost similar results on the day two and three. The average prognosis stands on the level 44.2%, it's now 33% more effective than mathematical probably for games prediction.

Day	Predicted exact score of the game	Percentage exact score	Predicted winner or draw	Percentage
02-07-20	1/8	12.5	5/8	62.5
11-07-20	4/25	16.0	10/25	40.0
12-07-20	3/20	15.0	6/20	30.0
Summary	8/53	14.5	21/53	44.166666666666664

Figure 9: Statistic of prediction from model 2

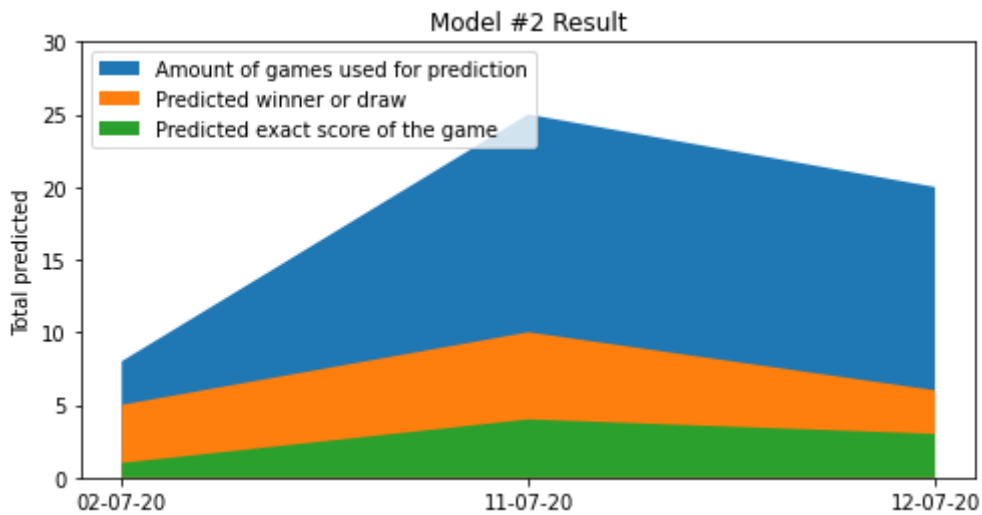


Figure 10: Diagram for the statistic of prediction from model 2

6.3. Model 3 evaluation

Model 3 was trained on the highest amount of real data – 200 games.

- The last model, we are taking into the evaluation shows a very interesting result. The average percentage of predicting the exact score is above the mathematical chances, average percentage against stands on the level of around 10%, but much more stable than the first model.
- On the Predicted winner or draw category we can see a big difference in percentages, the first day was amazingly predicted with 6 out of 8 games. This is more than two times higher than the random guess probably. But after the checking next two days we can see, that this percentage drops so much to an extremely low level.

Day	Predicted exact score of the game	Percentage exact score	Predicted winner or draw	Percentage
02-07-20	1/8	12.5	6/8	75.0
11-07-20	2/25	8.0	7/25	28.000000000000004
12-07-20	2/20	10.0	5/20	25.0
Summary	5/53	10.166666666666666	18/53	42.666666666666664

Figure 11: Statistic of prediction from model 3

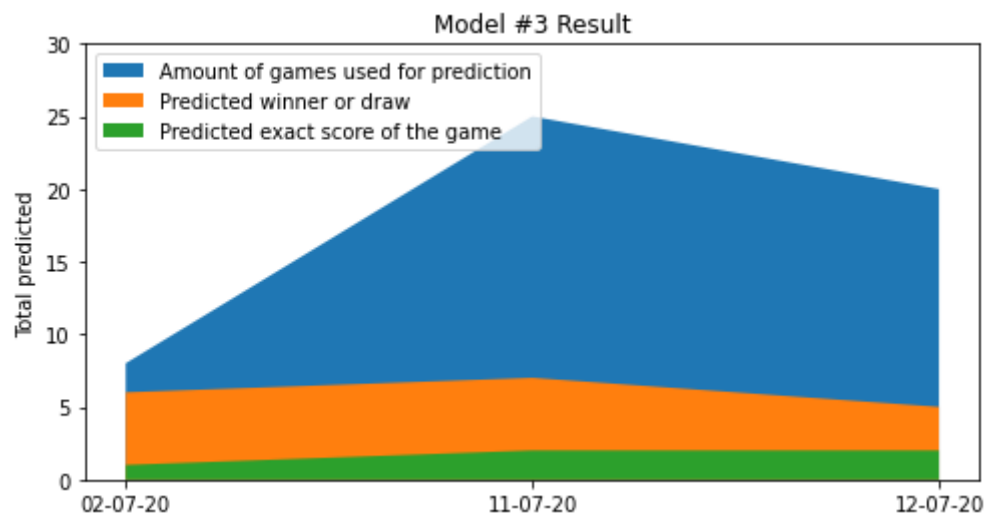


Figure 12: Diagram for statistic of prediction from model 3

[Введите текст]

6.4. Models comparison

For the model's comparison, we are using predictions, they made for real games. To discuss mainly the advantages and disadvantages of the model, we will be concentrated only on those 3 dates: 02-07-20, 11-07-20 and 12-07-20

All of them are very useful, in terms of finding the problems, which can be improved during the training of the future model.

6.4.1. First day of experiment

After the first look at the model's result, we can be somehow disappointed about their prediction for the first day. Two of them were doing very well and predicted more than 60% of the games, but one completely failed the experiment. To find the reason for this situation we should look at the games, we were trying to predict Figure 15.

Time	Home Team	Score	Away Team	Comments	Match Report
22:39	Real Madrid	1 : 0	Getafe	Spain SLL	
England - Premier League Detailed coverage					
18:00 FT	Sheffield United	3 : 1	Tottenham	33	Match Report
20:15 FT	Manchester City	4 : 0	Liverpool	74	Match Report
Spain - LaLiga Detailed coverage					
18:30 FT	Real Sociedad	2 : 1	Espanyol	1	Match Report
18:30 FT	Eibar	0 : 2	Osasuna	1	Match Report
21:00 FT	Real Madrid	1 : 0	Getafe	66	Match Report
Italy - Serie A Detailed coverage					
18:30 FT	Atalanta	2 : 0	Napoli	2	Match Report
20:45 FT	Roma	0 : 2	Udinese	2	Match Report
England - Championship Detailed coverage					
17:00 FT	Hull	2 : 1	Middlesbrough		Match Report

Figure 13: Games, took place at the date of the experiment (02-07-20)

Most of these games were very important because it was games between the table "place mates". The difference between the data was very small, but for almost all the games, except the game Roma vs Udinese, we have seen the success of a team, having a little bit better statistic.

After this analysis, we can assume that our model 1 is having a very small amount of data with different quality, which impacts a bigger range of possible results. For example:

Let's imagine the ball falls to the ground, we need to predict the height at which the ball will rise after falling. If we have seen too few examples, with different results we will think like this: The ball might have small pressure and will raise only for 40% of the initial height or this ball might have different materials and jump for 60% or even higher. Neural networks trying to consider as most data, as it can, so some of the examples from the data set can make only mistakes in prediction.

Now, we can investigate model 3, For the first day we have an amazing result, but why we have such a big drop for the next days? The problem is very similar to the one above. Because we have a big amount of data, saying for example: "Ball almost every time jumps to the 50% of the initial height", the model will

[Введите текст]

ignore extra data and always trying to make a prediction without caring about extra data. This problem is usually called overtraining.

Let us finally discuss model 2. There we have an average amount of data, so after the training model "thinks", usually the ball jumps for 50% of the initial height, but let us consider materials, this ball was created from, etc. This is the reason, why this model is not failing very much during all the games.

6.4.2. Second day of experiment

After reviewing day 2, we can see, how the overtraining problem having an impact on forecasting. According to our statistics, the best prediction for this day was made by the first model. To make the right conclusion about this day, we should investigate the games and their results (Figure 15) from football analytics way. Some of them finished with the surprising and hardly predictable result even for the specialists in analytics, like draw in the game between the English champion and the team, from the second part of the table.

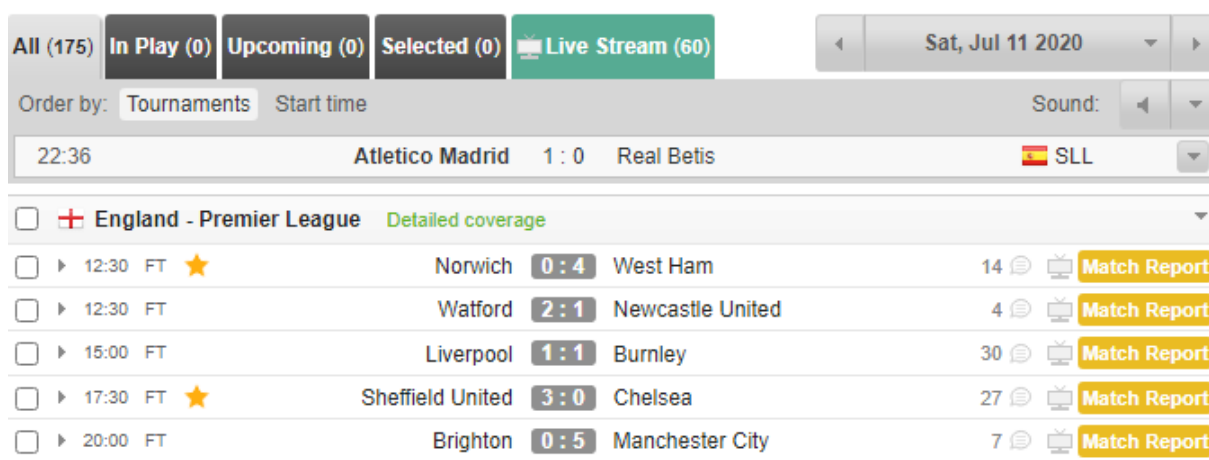


Figure 14: Games, took place at the date of the experiment (11-07-20)

The first model showed us the best result 12 out of 25 or 48% of correct predictions for the "Predicted winner or draw" category, while the second model was working also well, with 40% of predicted games. The worst result was shown by the third model. 7 out of 25 and 28%, which is also fine if it happens rarely.

6.4.3. Third day of experiment

All of the models having similar results, but they have predicted different games. Because of 20 games with different kinds of teams, we can see a very good percent of exact prediction on the 10-15%. But as well this variety of teams is having a big impact on the amount of predicted winner/draw. All of them are on the 25-30%. For this day, as well as the previous one, we can see the worst result is made by model 3.

All (171)	In Play (0)	Upcoming (0)	Selected (0)	Live Stream (68)	Sun, Jul 12 2020	
Order by: Tournaments		Start time		Sound: [Speaker Icon]		
23:20		Louisville City FC		1 : 3		Pittsburgh Riverhounds
England - Premier League Detailed coverage						
<input type="checkbox"/>	▶ 12:00 FT	★	Wolverhampton Wanderers	3 : 0	Everton	5 Match Report
<input type="checkbox"/>	▶ 14:15 FT		Aston Villa	2 : 0	Crystal Palace 1	5 Match Report
<input type="checkbox"/>	▶ 16:30 FT	★	Tottenham	2 : 1	Arsenal	46 Match Report
<input type="checkbox"/>	▶ 19:00 FT		Bournemouth	4 : 1	Leicester 1	8 Match Report
Spain - LaLiga Detailed coverage						
<input type="checkbox"/>	▶ 13:00 FT		Espanyol	0 : 2	Eibar	2 Match Report
<input type="checkbox"/>	▶ 16:00 FT		Levante	1 : 2	Athletic Bilbao	1 Match Report
<input type="checkbox"/>	▶ 18:30 FT		1 Leganes	1 : 0	Valencia	1 Match Report
<input type="checkbox"/>	▶ 21:00 FT		Sevilla	2 : 0	Mallorca	4 Match Report

Figure 15: Games, took place at the date of the experiment (12-07-20)

7. Summary

The main idea of this work – is to create a new kind of working model, for football result prediction. While other peoples are trying to predict the winner of the game, we decided to look more for trying to predict the number of goals each team should score.

After the following experiments, we can conclude that amount of data, taken for the training is not having the main impact on successful predictions. Mainly the quality of the data and a little bit of luck are making success in this field. For getting these 3 different neural networks, we have done more than 500 tries to fit them, because, for such kind of sport like football, the models should have some understanding about the usability of every component and not every time we can receive the same result even with the same statistic.

8. References

- [1] Maureen Caudill, Neural Network Primer: Part I", February 1989
- [2] Francois Chollet, Deep Learning with Python Paperback, 10 January 2018
- [3] Andreas C. Müller, Sarah Guido, Introduction to Machine Learning with Python: A Guide for Data Scientists, 2015
- [4] Sebastian Raschka, Vahid Mirjalili, Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow, 2 December 2019
- [5] Yann LeCun, Gökhan Bakır, Thomas Hofmann, Bernhard ,Alexander J. Smola, Predicting Structured Data (Neural Information Processing series), July 2007
- [6] David J. Livingstone, Artificial Neural Networks, 2009
- [7] <https://medium.com/@toprak.mhmt/activation-functions-for-deep-learning-13d8b9b20e> Access date: 11.07.2021