ADAM MICKIEWICZ UNIVERSITY IN POZNAŃ
FACULTY OF SOCIAL SCIENCES
Cognitive Science

Gracjan Popiółkowski

Effectiveness Report of Predictions From Various Machine Learning Models

Poznań 2023

Contents

Introduction

In this work, I will present the results of training an artificial neural network with different parameters, which I will use to predict the outcome of a match based on data from the first 15 minutes of the popular game "League of Legends." The first quarter of the game is a crucial moment, so I pose the following questions: How accurately can we predict the match outcome based on key statistics of both teams? Are the first 15 minutes decisive?

CHAPTER 1

Data Presentation

1.1. Dataset

I am using the "League of Legends Diamond Games (First 15 Minutes)" dataset, made available by Ben Fattori on the Kaggle platform. Access to this dataset was provided by the game creators, Riot Games. The data was collected from nearly 50,000 matches of diamond-ranked players (one of the highest ranks) on the "NA1" server and includes key statistics regarding the gameplay from the first 15 minutes of the game.

1.2. Data Description

Out of 19 data columns, 4 were removed before further analysis:

- `id` (data number) - could disrupt the performance of ML algorithms.
- `matchId` (individual match identifier) - has no impact on gameplay.
- `blueDragonKills` (how many dragons the blue team killed)
- `redDragonKills` (how many dragons the red team killed) - the value for both teams in every match is 0, so these variables are irrelevant.

Further analysis includes 15 variables, which are described below:

- `blue_win` - a categorical variable that takes the values 0 and 1. 1 if the blue team won, and 0 if the red team won.

The remaining columns are numerical variables:

- `blueGold` - the total amount of gold earned by the blue team, consisting of 5 players.
- `redGold` - the same for the red team.
- `blueMinionsKilled` - the number of minions killed by the blue team.
- `redMinionsKilled` - the same for the red team.
- `blueJungleMinionsKilled` - the number of jungle minions killed by the blue team's jungler.
- `redJungleMinionsKilled` - the same for the red team.
- `blueAvgLevel` - the average experience level of the blue team's players at the 15-minute mark.
- `redAvgLevel` - the same for the red team.
- `blueHeraldKills` - the number of "Herald" monsters killed by the blue team.

- `redHeraldKills` - the same for the red team.
- `blueTowersDestroyed` - the number of towers destroyed by the blue team.
- `redTowersDestroyed` - the same for the red team.
- `blueChampKills` - the total number of times the blue team killed an enemy champion.
- `redChampKills` - the same for the red team.

1.3. Data Preparation

As mentioned earlier, 4 data columns were removed. The categorical variable is the `blue_win` column. Additionally, I performed data standardization on all 14 numerical features, transforming them so that they have a mean value of 0 and a standard deviation of 1. This can improve the efficiency of the machine learning algorithms I used.

CHAPTER 2

Three Machine Learning Algorithms

The problem we are addressing is determining which team will win. Therefore, we will work on the binary classification variable `blue_win`. I have chosen three artificial intelligence algorithms that handle binary classification:

1. `SVM` (Support Vector Machines) - this is a machine learning algorithm used for both classification and regression. Its goal is to find a hyperplane that separates the data in the feature space. It works by maximizing the distance between the nearest points of different classes (support vectors).
2. `Artificial Neural Network` - this is a model inspired by the structure of the brain, built from units called neurons, connected to each other. Neurons are organized into layers, and each connection between them has a weight. The learning process involves adjusting the weights to make the model better at predicting outcomes.
3. `Random Forest` - an algorithm based on the idea of creating many decision trees. Each tree is trained on a different subset of data, and then either voting or averaging the results of the trees allows for a more stable prediction. It performs well when there are a large number of features. This algorithm is considered very effective for binary classification, which is why I have high expectations for it.

In the following section, I will present a comparison of the results obtained using these three algorithms.

CHAPTER 3

Analysis of Results

3.1. SVM

3.1.1. Training Set Size and Kernel Function

In the SVM algorithm, I evaluated the classification accuracy for various coefficient values: First, I tested the accuracy of predicting the match outcome for combinations of four different proportions of the training set to the test set (20% to 80%, 40% to 60%, 60% to 40%, and 80% to 20%) with different types of kernel functions: linear, polynomial (poly), and radial (rdf).

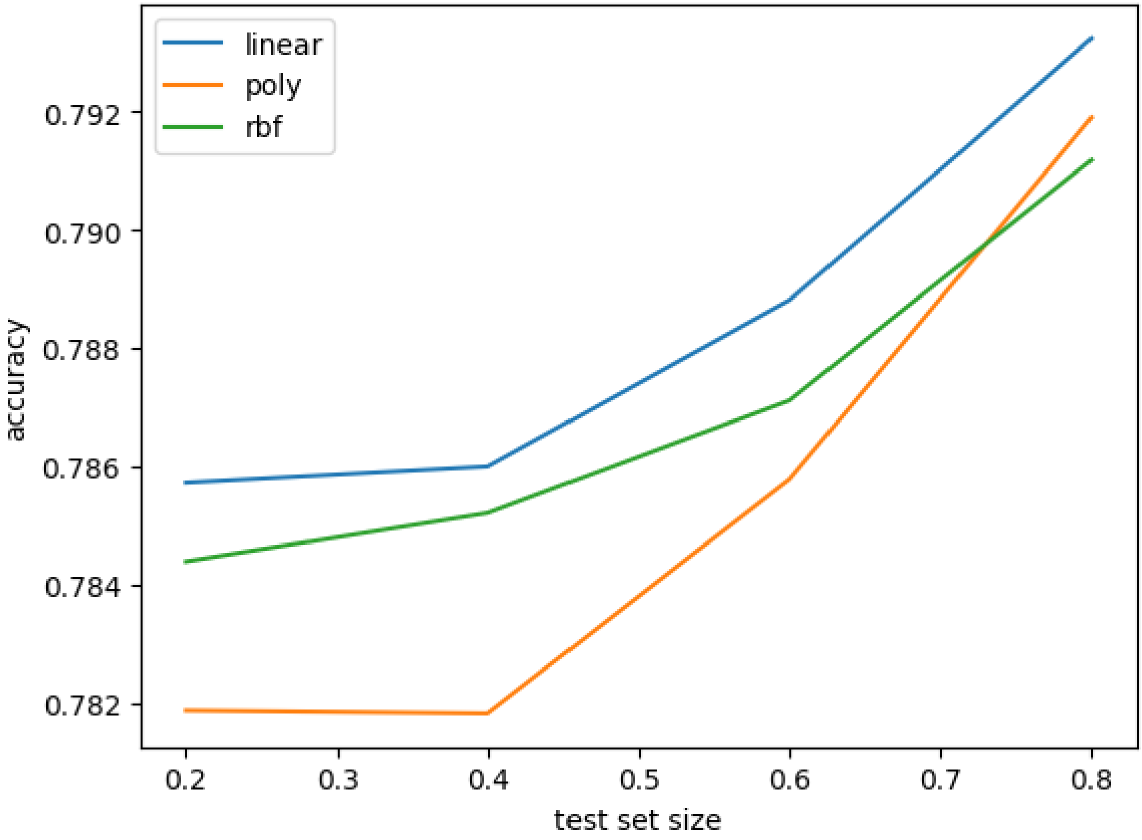

Figure 3.1: Graph showing classification accuracy for the combination of training-to-test set proportions with kernel function types.

In the above figure (3.1), we can see that the highest accuracy is achieved with a training set size of 80% and a linear kernel function. The prediction accuracy for this combination is 79.32%.

3.1.2. The `c_value` Parameter

Next, I checked the accuracy for different values of the `c_value` parameter. This parameter controls the trade-off between achieving a low training error and minimizing the norm of the weights. A low `c_value` makes the model more tolerant of training errors, while a high `c_value` penalizes training errors more strongly, leading to a more precise fit to the training points, but it may result in overfitting, where the model performs well on the training set but poorly on new data. The `c_value` values I tested were 0.001, 0.1, and 1. The best result was achieved with the middle value of 0.1, so I did not test further. With this `c_value` = 0.1, the prediction accuracy is 79.34%.

3.1.3. Degree of the Polynomial Kernel Function

The last thing I tested was the degree of the polynomial used in the kernel function. Accuracy was checked for degrees from 1 to 5, and the highest accuracy was achieved with a first-degree polynomial, at 79.4%.

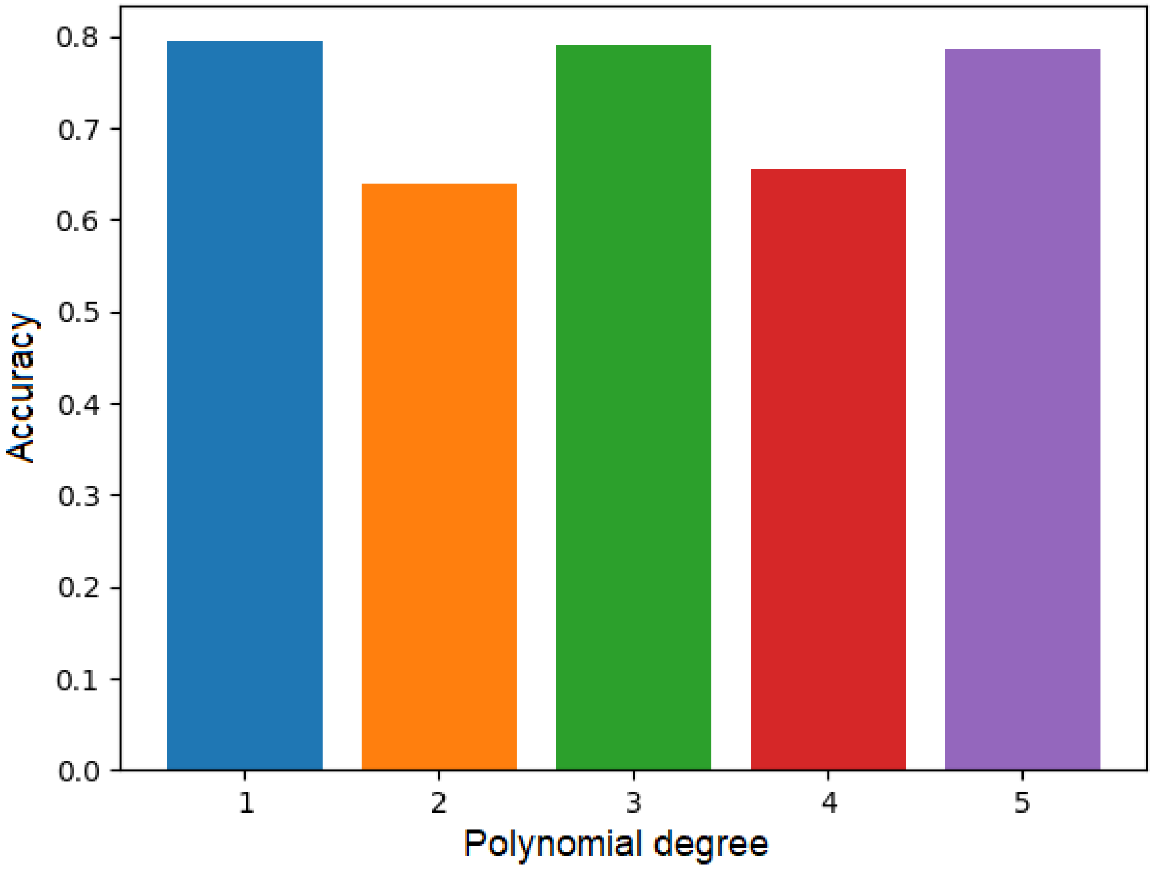

Figure 3.2: Graph showing the prediction accuracy of the match outcome for different degrees of the polynomial kernel function.

3.1.4. Summary

In summary, the best combination of SVM algorithm parameters is:

- `Kernel Function`: linear,
- `Training Set Size`: 80% of the data,
- `c_value` = 0.1,
- `Polynomial Kernel Degree` = 1.

This combination provides a classification accuracy of 79.4%.

3.2. Artificial Neural Network

In my Artificial Neural Network algorithm, I also set the training set size to 80% and the test set size to 20%. I configured the first dense layer to have 16 neurons with a `ReLU` activation function. The second dense layer had 8 neurons, also with a `ReLU` activation function, while the output layer had one neuron and a `sigmoid` activation function. After 15 training epochs, the model achieved an accuracy of 78.9%.

3.3. Random Forest

For the Random Forest model, I also used a training set size of 80% and a test set size of 20%. After testing various parameter combinations, such as `n` (number of decision trees) and `random_stat` (random seed value), the best result, with an accuracy of 78.58%, was achieved with `n` = 100 and `random_stat` = 42. Ultimately, the Random Forest model results were not better than the well-configured SVM and Artificial Neural Network models.

Summary

In conclusion, in my work, I used three machine learning algorithms to predict the outcomes of matches in the game "League of Legends," using data from the first 15 minutes of the game. The prediction results for each algorithm with the appropriate configuration were:

- `SVM` = 79.4%,
- `Artificial Neural Network` = 78.9%,
- `Random Forest` = 78.58%.

The conclusion from the analysis is that the best-performing algorithm was the properly configured SVM. However, none of the algorithms achieved an accuracy above 80%, which suggests that it is difficult to predict the outcome of a match with more than 80% accuracy based on the first 15 minutes of the game. Exceeding this threshold might require adding more features to the dataset. However, the 14 features included are the most important gameplay parameters that a player can influence in the first 15 minutes of the game. It is worth noting that a full match in "League of Legends" typically lasts between 20 and 50 minutes, so the achieved accuracy of around 80% is already quite high.