

*Middle East Technical University*

*Department of Statistics*

**STAT 493**

**NEW HORIZONS IN STATISTICS**

**TERM PROJECT**

*“Turkish Basketball Super League*

*2018-2019 / 2019-2020 / 2020-2021 Seasons”*

*Submitted to Barış Sürücü*

**ALPER TUNAHAN ÖZTÜRK 2290856**

***TABLE OF CONTENT***

***1.******Introduction………………………………………………………….3***

***2. Data description……...………………………………………..……...3***

***3.******Aim of the Study………………………….……………………….....4***

***4. Methodology ………….………………………….……………….…4***

***5. Research Questions …………………….………………………...…6***

***6***. ***Data Analysis …………………..…….……………………………....7***

***7.*** ***Conclusion………………………………………………………..….18***

***8. References………………………….……………………………….18***

***9. Appendix ………………………….………………………………..19***

1. **Introduction**

Besides just being fun, sports can help you focus better in school, relieve stress more easily and become less anxious, work better in a team and increase your overall energy. It is believed that basketball, in particular, improves cardiovascular health and builds bone strength. It is a type of sport that has millions of enjoyers of all ages around the world. To summarise, basketball is a very ambitious subject with many roots in different fields and of course in statistics. More basketball statistics information is preserved in this context. By analysing the current data set, making the required corrections and modifications, and employing a variety of computer languages, an analysis will be generated for this study. Then, given the relevant facts, these systems will provide a variety of projections. The judgments of statisticians and other professionals in the industry, as well as more readily available data, help sports teams win important matches and earn large money. This report will therefore draw attention to an in-depth analysis of *3 Season Play-by-Play Basketball* data. Machine Learning methods are used to test the provided research questions. These methods are created with the help of R Programming language on R-Studio as an IDE, Python and Microsoft Excel. The purpose of this research is to analyze the relationships of each game between a home and an away team.

## Data Description

In this research, the Turkish Basketball league data is used. The 3-season data set can be defined as in-match action data. A sensor's information on what is happening in the situation being watched is described as "action data"(1). The data collection needed to be reorganized to better fit the goals of the study and the questions that needed to be addressed. Considering that the current set includes variables that won't be included in the study. Additionally, a variety of programming techniques have been utilized to speed up the research process and produce more precise results. To fit these techniques, the data has been modified. Data manipulation was done using the statistical programming language Rstudio 4.2.2, while machine learning applications, modeling, and forecasting were done using Python 3.9.

The first two periods from our dataset using the filtering procedure are constrained because the research questions required answers from the statistics acquired in the first half. The outcomes that will result from the comparison of the data are projected.

It is possible to establish which player and which action belongs to which team by glancing at the "playersteam" column, and afterwards, which team is at home or away. Then, Home statistics and Away stats are integrated under a single line and grouped by MatchID. Using this application of combining under a single column, machine learning technologies also made it feasible to ascertain whether the opposing team had an effect on our forecasts.

Using the predictions provided, columns are made to display the game winner, the player with the most offensive rebounds, and the player with the most opponent ball thefts. The outcomes are indicated as "Home" or "Away" in these categorization columns, which are compiled from all match scores.

The variables

|  |  |  |  |
| --- | --- | --- | --- |
| MatchID | HomeTeamID | AwayTeamID |  |
| HomeTeamScore | AwayTeamScore | Home3P-Shot | Away3P-Shot |
| Home2P-Shot | Away 2P-Shot | Home Offensive  Rebound | Away Offensive  Rebound |
| Home Turnover | Away Turnover | Home Turnover  Back Court | Away Turnover  Back Court |
| Home Steal | Away Steal | Home Block | Away Block |
| Home Foul | Away Foul | Home Foulon | Away Foulon |
| Home Freethrow | Away Freethrow | Home 2Point  Percentage | Away 2Point  Percentage |
| Home 3Point  Percentage | Away 3Point  Percentage | Half Time Score |  |

Columns Created in order to make predictions:

|  |  |  |  |
| --- | --- | --- | --- |
| Winner (Home/Away) | More Offensive Rebound (Home/Away/Draw) | More Steals (Home/Away/Draw) | Half Time Winner (Home/Away/Draw) |

## Aim of the Study

The primary aim of the study is to organize the available data and make it usable in order to obtain logical and thorough answers to the issues posed in the research questions. The teams that are in the tactic selection phase can benefit from the outcomes. Using the data collected during the first half of the game, we attempted to estimate the probabilities that may occur as a result of the match in order to respond to the research questions and make use of our models. As stated in the Data description, the data has been organized.

## Methodology

This project is carried out with the help of Rstudio and Python programming languages. Decision Tree and XG Boost algorithms are used. Research questions were given and solved with examination of the data. The first quarter data is taken and 70 percent of this data is used to train the models. For the analysis each game between a home and an away team is taken into consideration. New input variables are created by using existing data to increase the accuracy of the models and target outcomes are predicted with these models.

***Graphs:***

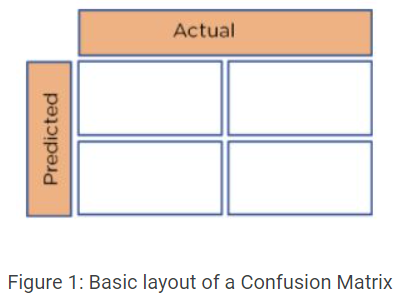
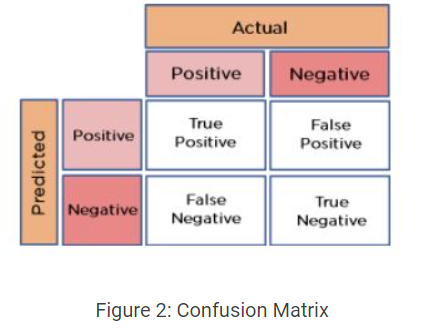
A bar chart uses bars of various lengths to graphically organize data. These bars' lengths correspond to the size of the data they represent. Bars with different vertical and horizontal lengths to show the values they stand for are used to depict categorical data. Line graph is another name for a vertical bar graph. Discrete categories are compared using a bar graph.

***Decision Tree:***

A non-parametric supervised learning approach called a decision tree is used for both classification and regression applications. It is organized hierarchically and has a root node, branches, internal nodes, and leaf nodes. A decision tree's root node is the first node and it has no incoming branches. The internal nodes, sometimes referred to as decision nodes, are fed by the root node's outgoing branches. Both node types undertake assessments based on the available attributes to create homogenous subsets, which are represented by leaf nodes or terminal nodes. All of the outcomes within the dataset are represented by the leaf nodes. This kind of flowchart layout also produces an understandable representation of decision-making, enabling a decision's rationale to be understood better.

***Confusion Matrix:***

A confusion matrix aids in visualizing the results of a classification task by providing a table arrangement of the various outcomes of the prediction and findings. It creates a table with all of a classifier's predicted and actual values.

***Model Training and Overfitting:***

Machine learning algorithms need a sample dataset to train the model when they are created. The model, however, may begin to learn the "noise"—or irrelevant information—within the dataset if it trains on sample data for an excessively long time or if the model is overly complex. The model is "overfitted" when it learns the noise and matches the training set too closely. Data scientists refer to this as overfitting, which happens when a statistical model matches its training data exactly. When this occurs, the algorithm's goal is defeated because it cannot operate accurately against unseen data.

***Cross Validation:***

A statistical technique called cross-validation is used to gauge the effectiveness (or accuracy) of machine learning models. It serves as a safeguard against overfitting in predictive models, especially when the available data may be scarce. In cross-validation, you divide the data into a predetermined number of folds (or partitions), analyze each fold individually, and then average the total error estimate.

***XGBoost:***

Extreme Gradient Boosting (XGBoost) is a distributed, scalable gradient-boosted decision tree (GBDT) machine learning framework. The top machine learning library for regression, classification, and ranking issues, it offers parallel tree boosting. A model made using GBDT is composed of several decision trees. The way the trees are constructed and combined distinguishes this machine learning model from others.

***Optuna:***

Optuna is a software framework for automatic hyperparameter tuning that was created specifically for machine learning. Optuna makes use of a feature known as define-by-run API that enables users to write highly modular code and create search spaces for hyperparameters on the fly.

## Research Questions

The solutions to the issues to be solved will be discovered using a variety of visualizations and tests. The reader of the study will be made aware of any statistical tests and modeling that were done to the data as needed.

In order, the questions that will be examined are;

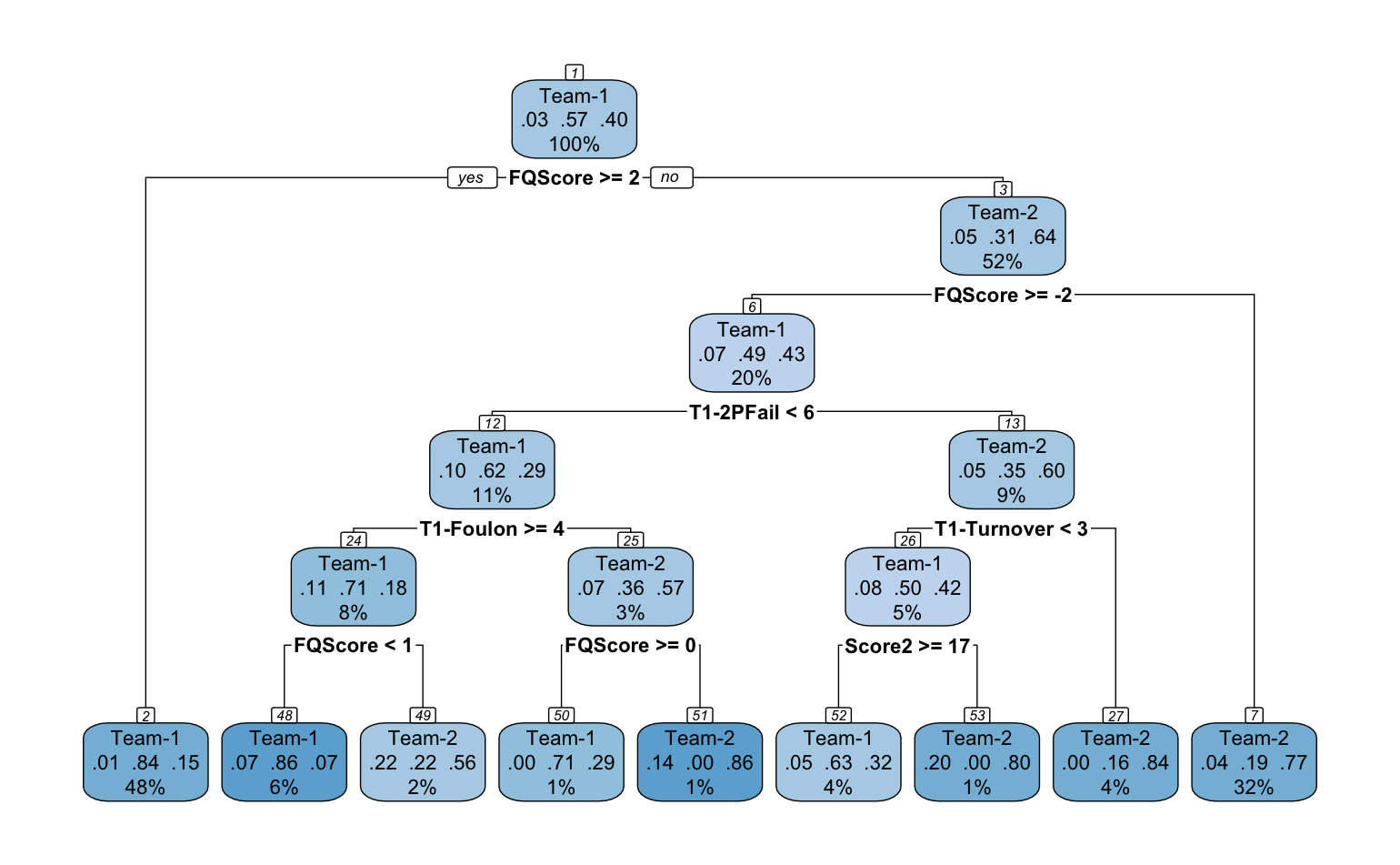
* Who leads the game at the end of the first half?
* Which team has more three-point trials at the end of the third quarter?
* Which team has more two-point success percentage?

The answers of these questions will be provided to the reader as the Home team or Away team.

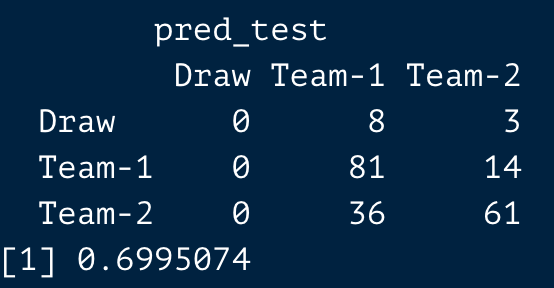
## Data Analysis

In order to obtain accurate results, it is important to conduct a thorough analysis and properly gather and organize the data that will be used. Various modeling techniques and visualizations will be employed to gain a deeper understanding and make predictions. This section will go through each question in detail, testing different models to determine which one produces the best results. The outcomes will be reported. There are certain factors that must be taken into account when building a model, which will be explained to the reader. To ensure that the data is suitable for modeling, it was divided into a 70 percent training set and 30 percent test set. This approach allows for the model to be trained effectively and make more accurate predictions, while also addressing the issue of overfitting. Additionally, a cross validation technique was also utilized to further prevent overfitting.

* **Research Question 1: Who leads the game at the end of the first half?**

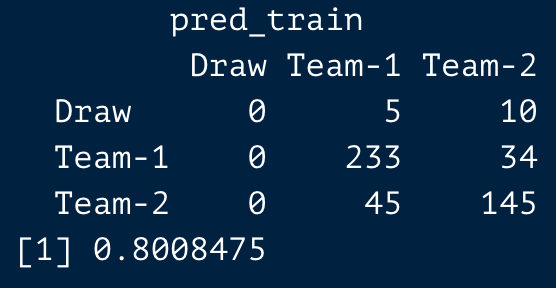
****

Above you see the Decision Tree obtained using Rstudio. To interpret it, the probability of winning the first quarter of the home team in the first place is 57%. If the first quarter score difference is greater than -2 (minus the away team dominance), the away team has a 64% chance of winning. If it is less than -2, the probability of winning home is 79%. If 2 points fail less than 6 Home has a 49% chance to win.



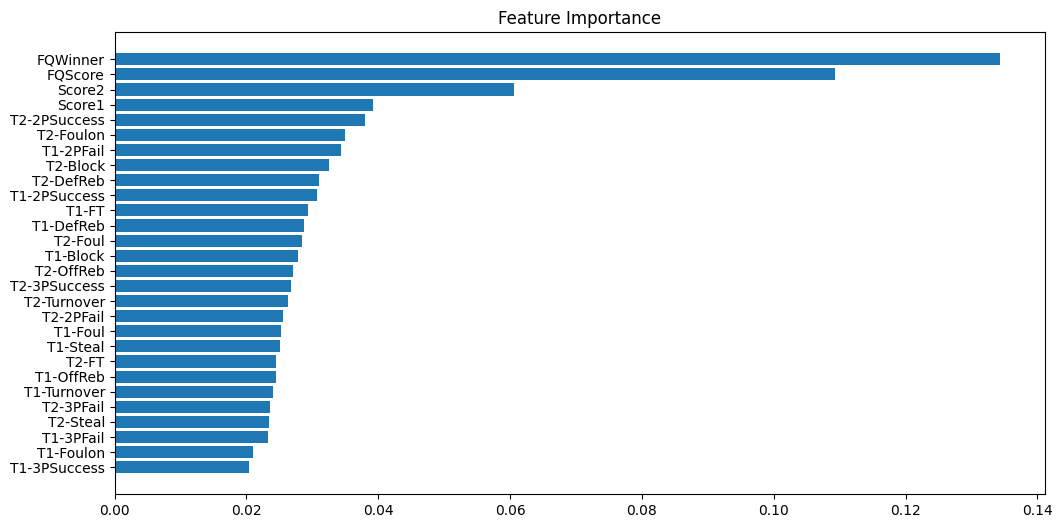
First-half winner test Accuracy 0.6995074

The accuracy of the confusion matrix that Decision Tree generated from the test data was close to 70%. In other words, out of every 100 forecasts we receive, 70 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.

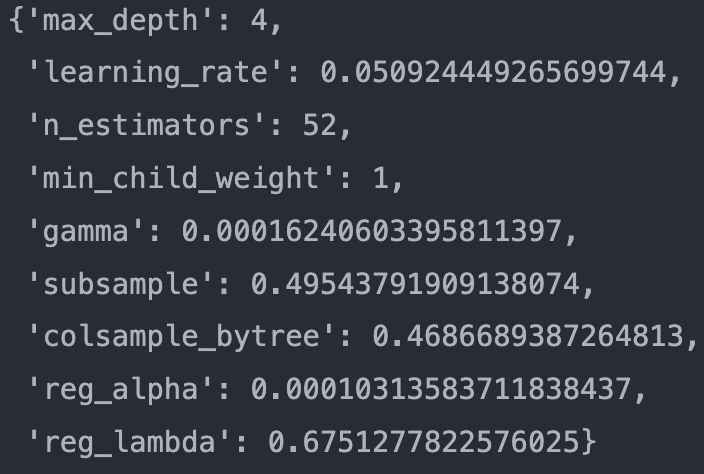


First-half winner train Accuracy 0.8008475

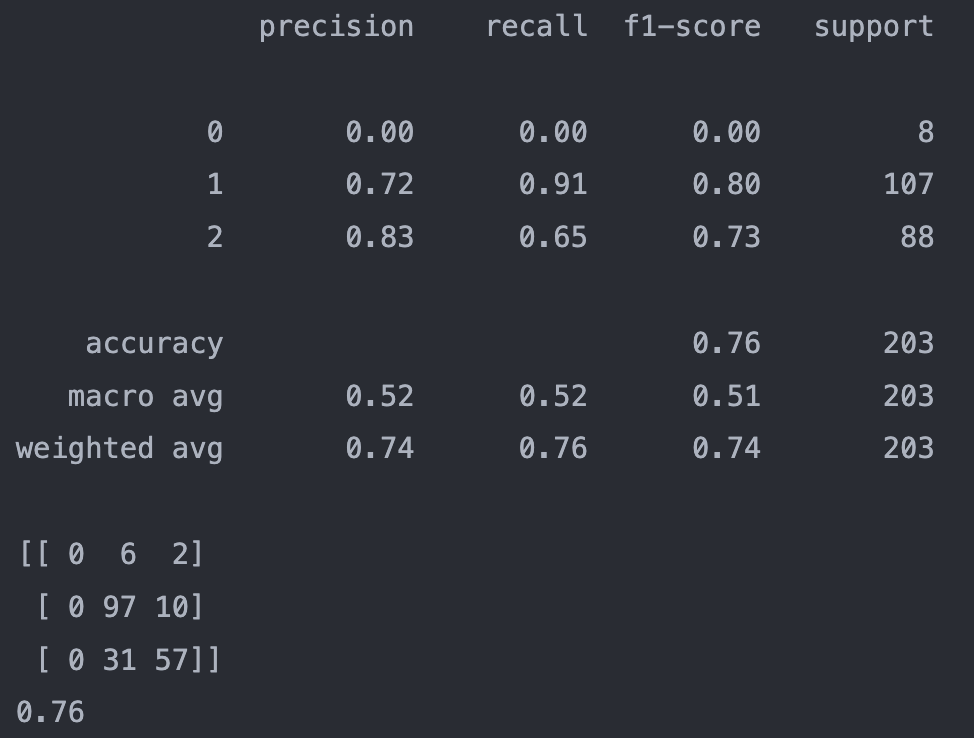
The accuracy of the confusion matrix that Decision Tree generated from the train data was 80%. In other words, out of every 100 forecasts we receive, 80 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.

**XGBoost**

Feature importance shows us which factor is effective in the XGBoost model and how much. At the same time, we see all the inputs entering the model. First Quarter Winners and First Quarter Scores affect the model the most.

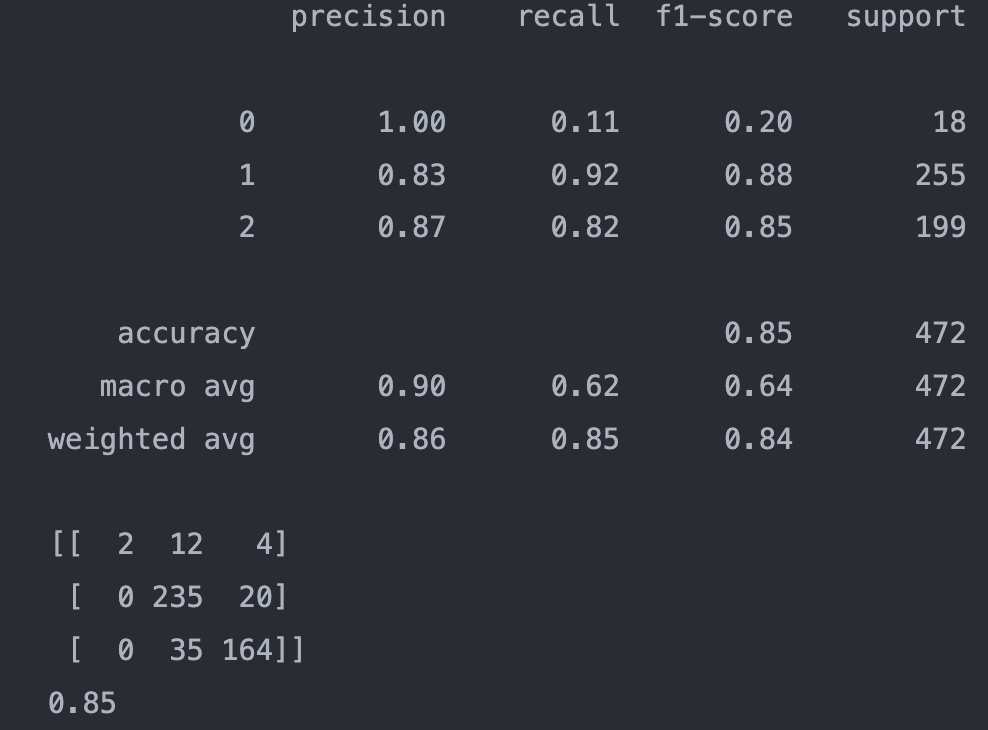


In order to get these parameters, some python packages are used. Those packages give the best parameters for the model and the output. Thanks to these hyper-tuning packages, one can extract more accuracy from the model.



On the left-hand-side, it can be seen the output from XGBoost. It obtained from XGboost package from python. At the end of the output, there is a 3x3 matrix. In this matrix, rows and columns names are Draw, Home, and Away, respectively.

First-half winner test Accuracy 0.76

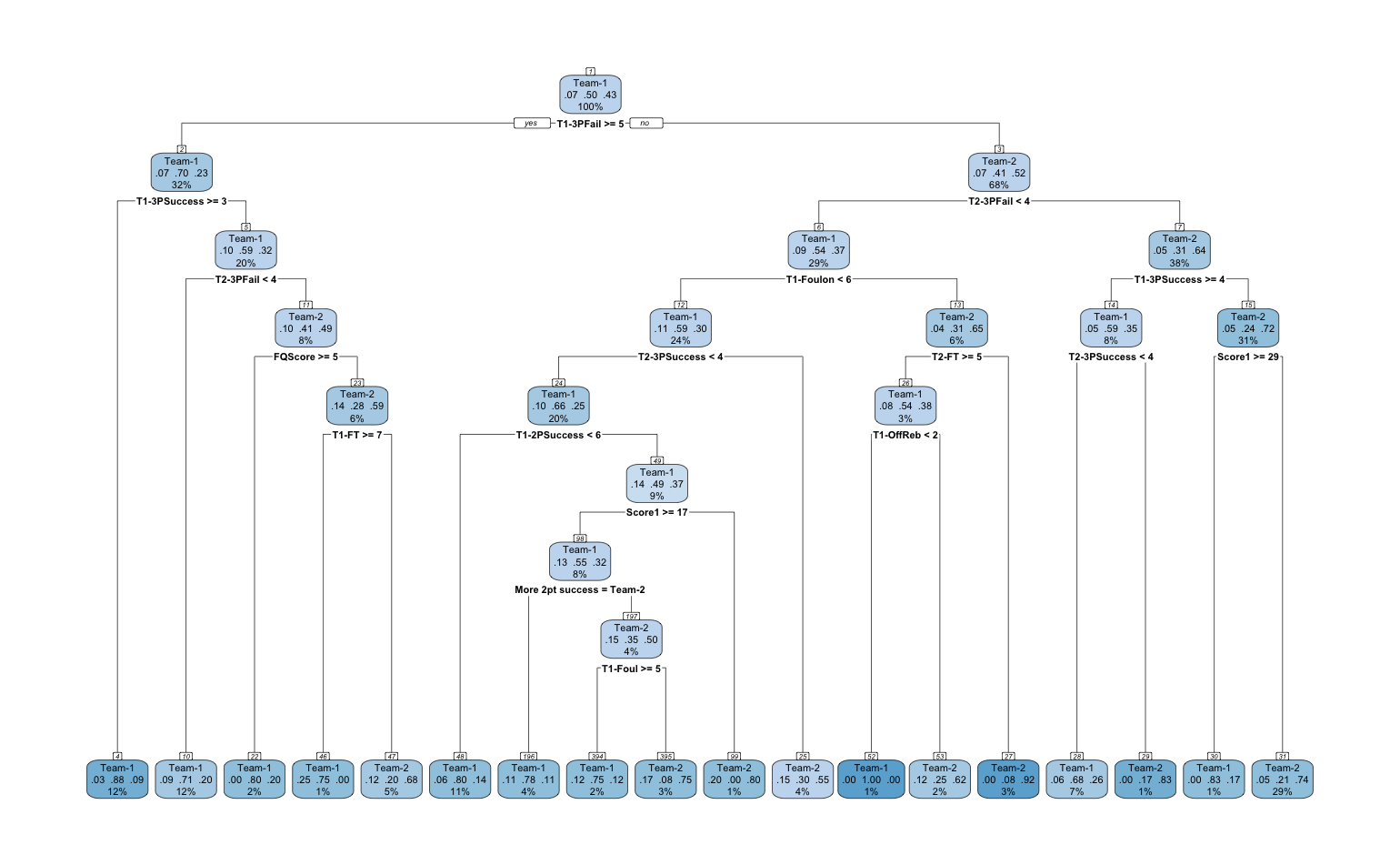


Again, on the left-hand side, the output was obtained in the exact same way. So, the numbers indicate the same descriptions.

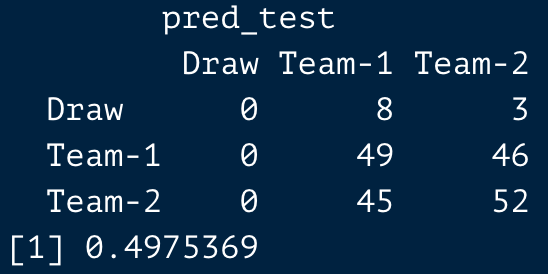
First-half winner train Accuracy 0.85

It can be seen from the results at the end of the first half, the home team leads.

* **Research Question 2: Which team has more three-point trials at the end of the third quarter?**

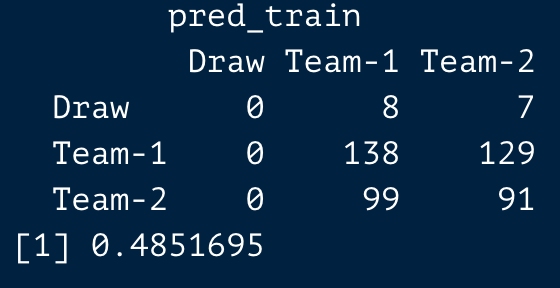


With the answer "yes" , the Decision Tree goes to the left-hand branch, and "no" to the condition goes to the right-hand side of the branch. The distribution of the data as a percentage is displayed in percentile displays. The chances are respectively Home, Draw, and Away. The Confusion Matrix generated by the Decision Tree shows more precise findings.



More three-point trials test Accuracy 0.4975369

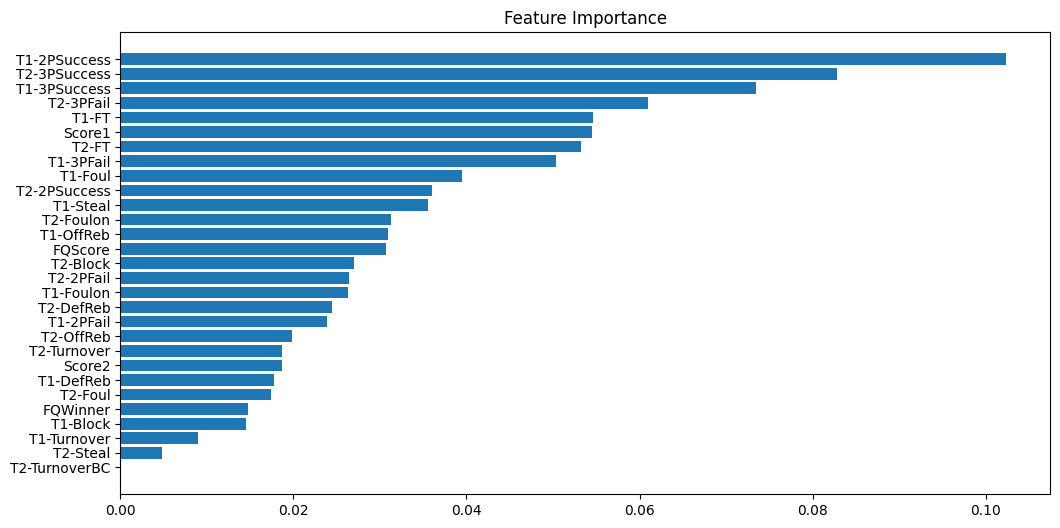
The accuracy of the confusion matrix that Decision Tree generated from the test data was close to 50%. In other words, out of every 100 forecasts we receive, 50 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.



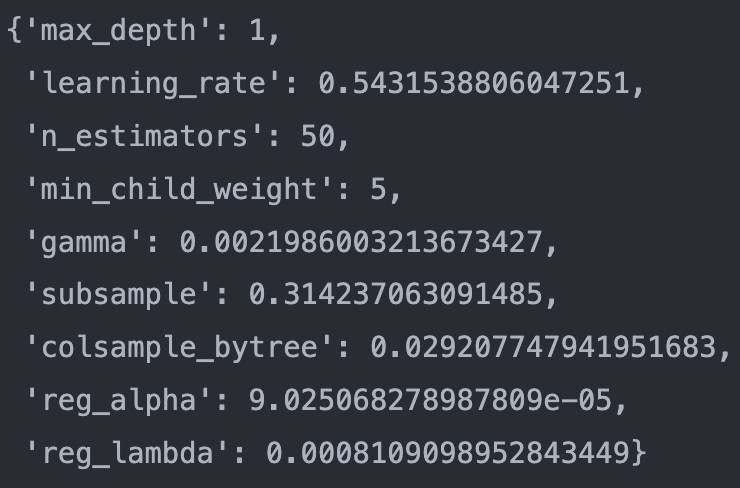
More three-point trials train Accuracy 0.4851695

The accuracy of the confusion matrix that Decision Tree generated from the train data was 48%. In other words, out of every 100 forecasts we receive, 48 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.

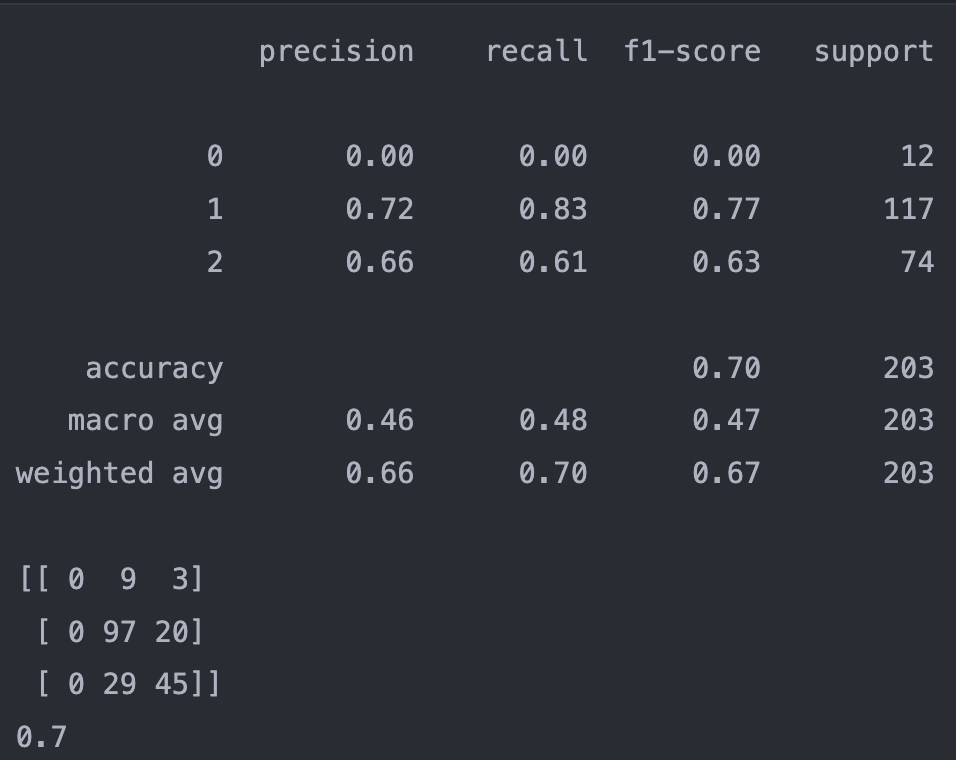
**XGBoost**



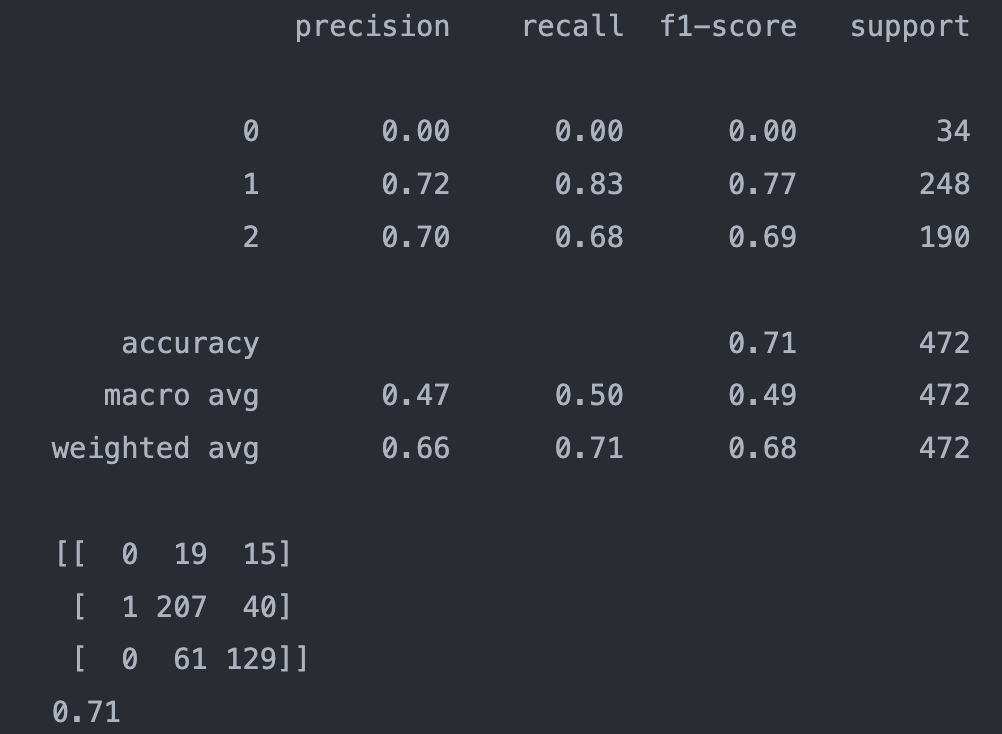
Feature importance shows us which factor is effective in the XGBoost model and how much. At the same time, we see all the inputs entering the model. Team1 2 point success and Team2 3 point success affect the model the most.



In order to get these parameters, some python packages are used. Those packages give the best parameters for the model and the output. Thanks to these hyper-tuning packages, one can extract more accuracy from the model.



On the left-hand-side, it can be seen the output from XGBoost. It was obtained from the XGboost package from python. At the end of the output, there is a 3x3 matrix. In this matrix, rows and columns names are Draw, Home, and Away, respectively.

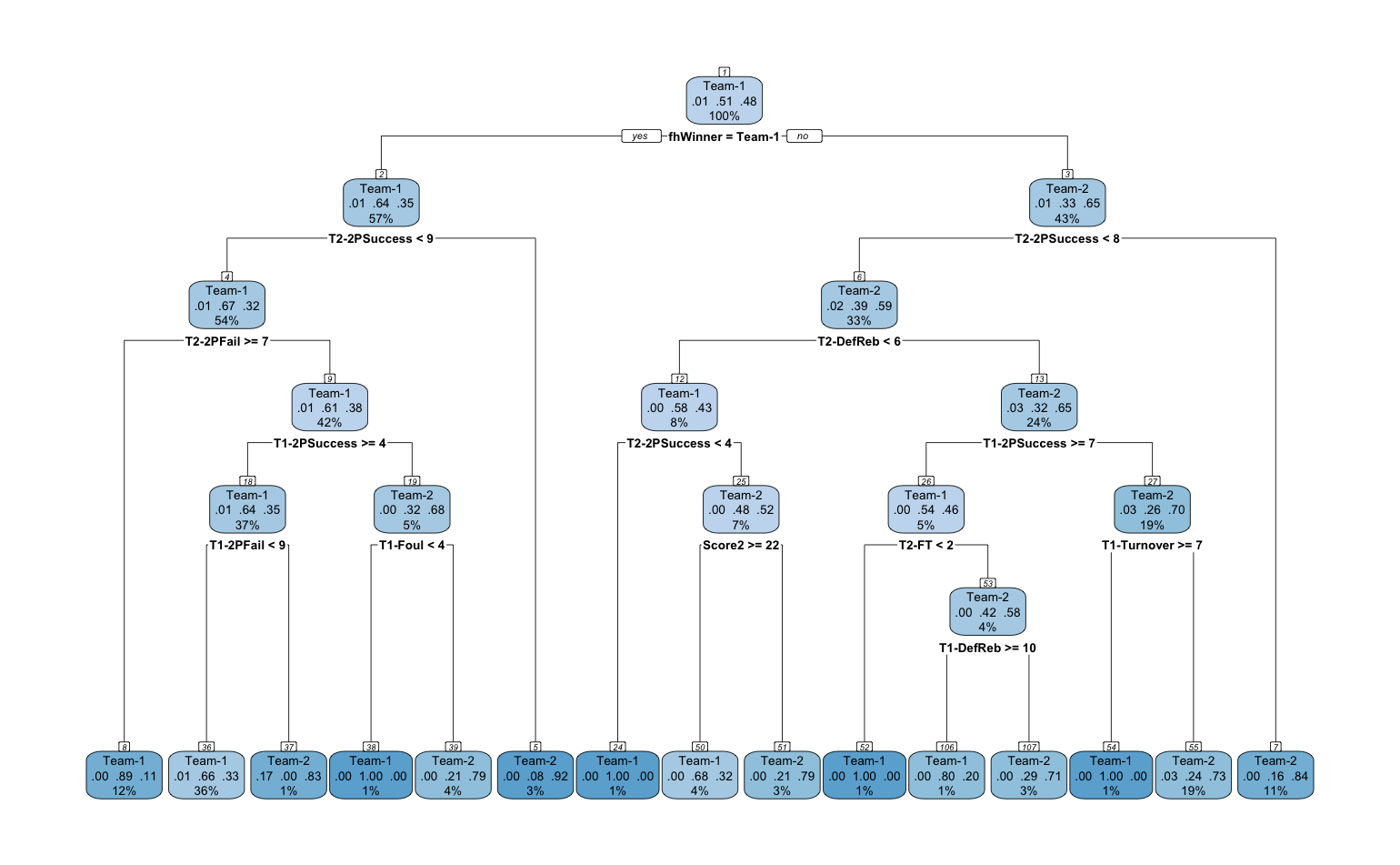
More three-point trials test Accuracy 0.7

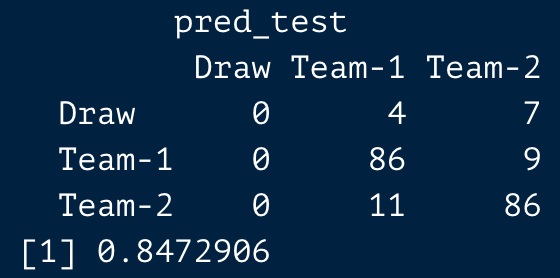
Again, on the left-hand side, the output was obtained in the exact same way. So, the numbers indicate the same descriptions.

More three-point trials train Accuracy 0.71

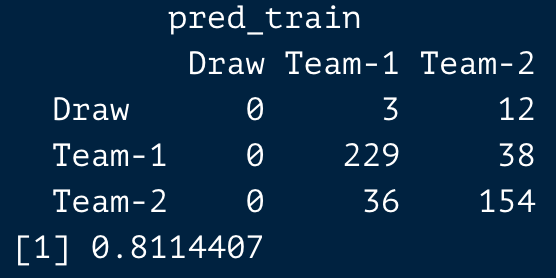
It can be seen from the results of three point trials at the end of the third quarter, the home team leads.

* **Research Question 3: Which team has more two-point success percentage?**



If the winner of the first half team1 which is the home team, go to the left. If it is team2 which is an away team, go to the right and go on.

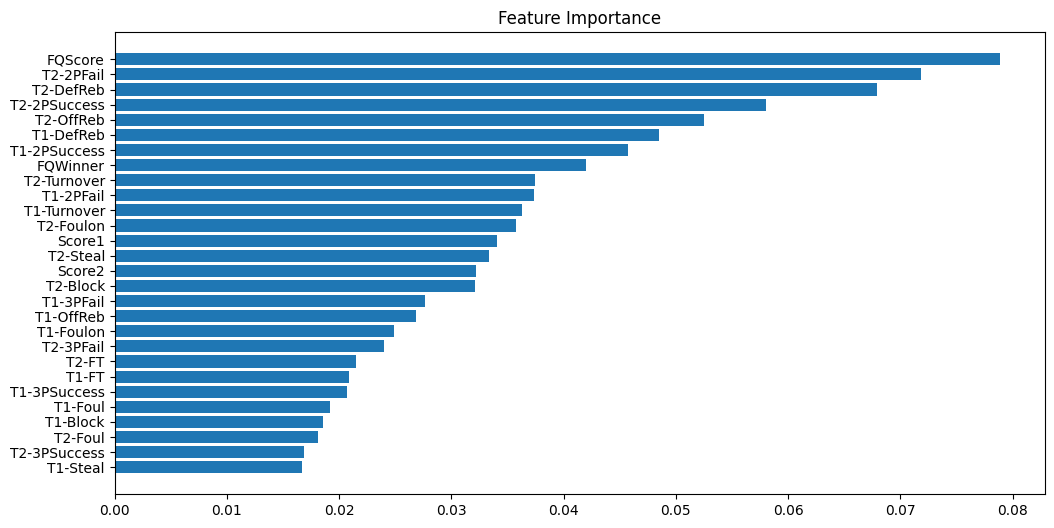
Two point success percentage test Accuracy 0.8472906

The accuracy of the confusion matrix that Decision Tree generated from the test data was 84%. In other words, out of every 100 forecasts we receive, 84 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.

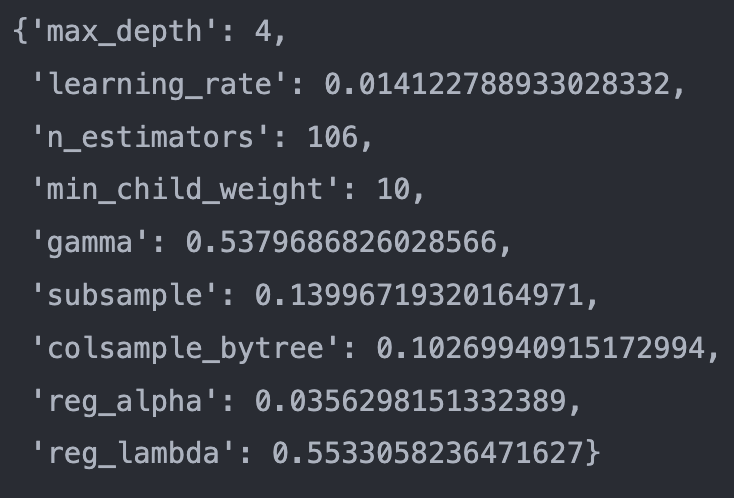
Two point success percentage train Accuracy 0.8114407

The accuracy of the confusion matrix that Decision Tree generated from the train data was 81%. In other words, out of every 100 forecasts we receive, 81 have proven to be accurate. Analyzing the matrix's indices will also reveal the quantity of Type 1 and Type 2 faults.

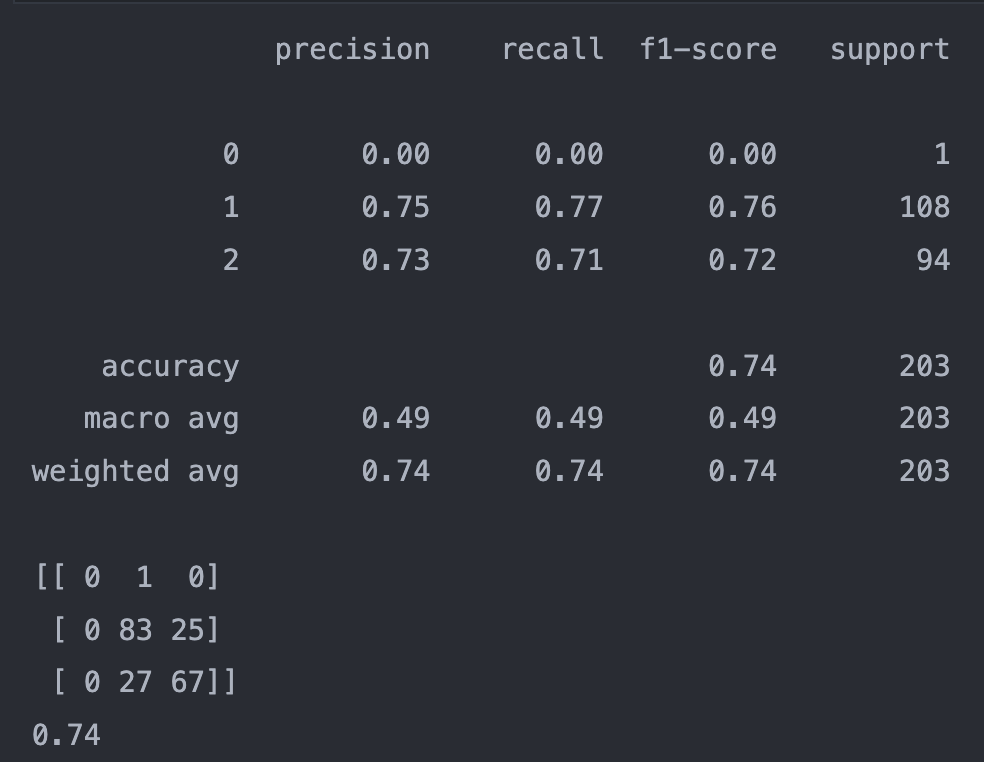
**XGBoost**



Feature importance shows us which factor is effective in the XGBoost model and how much. At the same time, we see all the inputs entering the model. The first Quarter score and Team2 2 point fail affect the model the most.

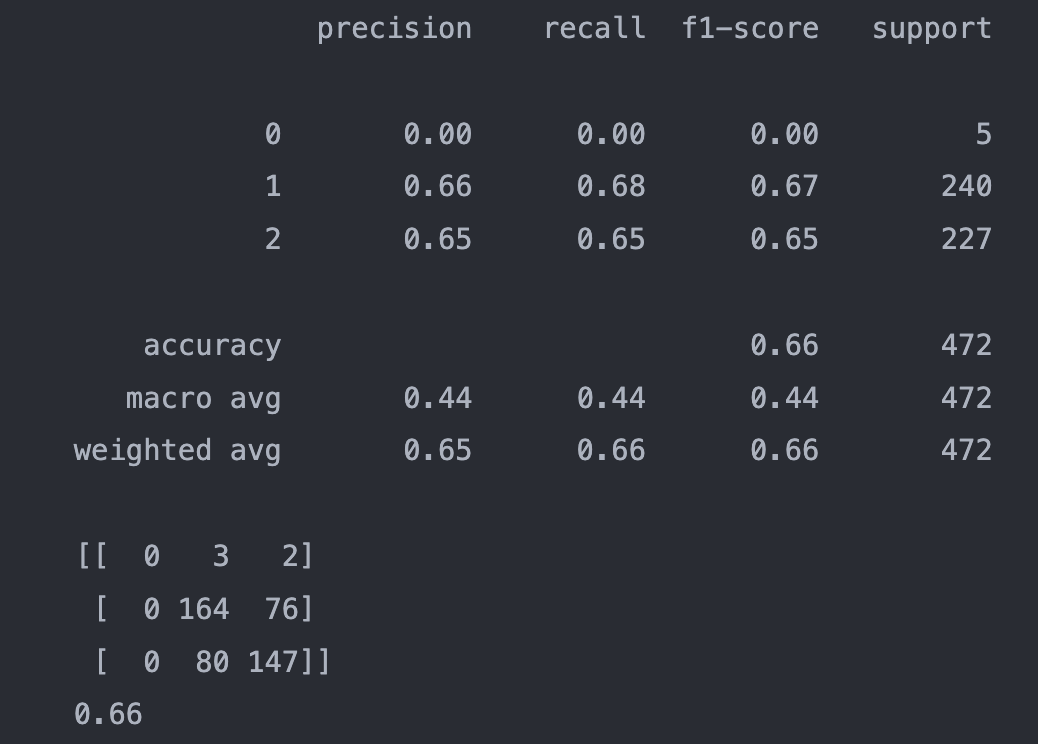


In order to get these parameters, some python packages are used. Those packages give the best parameters for the model and the output. Thanks to these hyper-tuning packages, one can extract more accuracy from the model.



On the left-hand-side, it can be seen the output from XGBoost. It was obtained from the XGboost package from python. At the end of the output, there is a 3x3 matrix. In this matrix, rows and columns names are Draw, Home, and Away, respectively.

Two point success percentage test Accuracy 0.74



Again, on the left-hand side, the output was obtained in the exact same way. So, the numbers indicate the same descriptions.

Two point success percentage train Accuracy 0.66

It can be seen from the results of two point success percentage, the home team leads.

For modeling in this study, decision trees and XGBoost machine learning algorithms were employed. These techniques provide us with beneficial findings, but there are a few things to keep in mind. We must verify these models' underlying assumptions in order to prevent any errors.

Assumptions of Decision Tree:

* The decision tree approach is non-statistical. There are no presumptions about the training data or the residuals of the predictions.
* There are no constant variance, independence, or distributional assumptions.
* Random forests are non-parametric and can handle skewed and multi-modal data as well as categorical data that are ordinal or non-ordinal because they do not require explicit distributional assumptions.

The decision tree's non-statistical assumptions are as follows:

* The entire training set is first regarded as the root.
* Categorical feature values are desired. If the values are continuous, they must first be discretized before the model can be constructed.
* On the basis of attribute values, records are dispersed recursively.
* Utilizing various statistical methods, characteristics are placed in the tree as the internal or root node.

The XGBoost’s major assumptions:

* The encoded integer values of each input variable may be assumed by XGBoost to have an ordinal connection.
* The method can handle missing values by default because it DOES NOT ASSUME that all possible values are available, which is why XGBoost thinks your data might not be accurate (it can handle missing values).
* When using tree-based techniques, missing values are discovered during the training stage. As a result, the following happens:
* XGBoost is used to handle sparsity.
* Because XGBoost only handles numeric vectors, categorical variables must be converted into numeric variables.
* It is necessary to convert a dense data frame with few zeros in the matrix into a highly sparse matrix with numerous zeros.
* This means that XGBoost can accept input in the form of a sparse matrix for variables.

## Conclusion

The goal of this study has been to create predictions using the information that has been given to the research team. The dataset has to be first organised according to the posed queries. Since there was more data than was required for the findings expected of us when we modelled in its current condition, we were unable to obtain the results we desired. Our dataset, which is described in more depth in the section on the data description, consists of the activity data from the Turkish Super League's three seasons, which begin in 2018 and finish in 2021. The Data description also includes a detailed description of the adjustments and arrangements applied to the dataset. Firstly, the research questions required us to make predictions based on the first-half outcomes. Due to this, we filtered the data from the third and fourth periods using Rstudio and Python. Then, in order to improve our forecast, we preserved the relevant columns and deleted the unnecessary ones. We employed a variety of machine learning algorithms to determine how well we predicted once the filtering techniques for each question were finished. These are XGBoost and Decision Tree. We split our complete data into two groups: 30% for the test set and 70% for the train set before creating the aforementioned models. Confusion Matrices display the outcomes from both the train set and the test set. Looking at the Type 1 and Type 2 mistakes brought on by Confusion Matrices allowed us to determine our accuracy rate. The % accuracy rating indicates how many of the 100 forecasts were correct. We could guard against overfitting by dividing them into test and train sets. By avoiding overtraining, machine learning algorithms would produce better results. Cross Validation Technique is another technique that has been utilized to reduce overfitting. When applying these techniques, there are a few things to keep in mind. Without assumptions, we are unable to apply these techniques. The presumptions have been verified, and no issues have been found.

## References

*Sicilia, A., Pelechrinis, K., & Goldsberry, K. (2019, July). Deephoops: Evaluating micro-actions in basketball using deep feature representations of spatio-temporal data. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 2096-2104).*

## Appendix

Rstudio and Python codes used for Analysis, Modelling and Visualization.