For cited: *Seyrek, M., Kiriktir, E., Doğan, Y. & Yapıcıoğlu, B. (2021). Football Match Result Prediction Decision Support System. Journal of Current Research on Engineering, Science and Technology, 7 (2), 173-202.*

Journal of Current Research on Engineering, Science and Technology (JoCREST)
ISSN: 2651-2521

Year: 2021
Volume: 7
Issue: 2

Crossref doi: 10.26579/jocrest.85

Research Article/Araştırma Makalesi

Football Match Result Prediction Decision Support System

Mustafa SEYREK[1], Enes KİRİKTİR[2], Yunus DOĞAN[3] & Bülent YAPICIOĞLU[4]

Keywords
Data Mining, Decision Support Systems, Sport Science.

Abstract
Today, predicting the results of football matches is as important as the results. Before the match, people try to predict the outcome of the next match by considering the performances of the teams in the previous matches on various social media platforms. Along with the results of the matches, the outcome of the second half has become a subject that football fans and technical teams meticulously think about. There are many statistical metrics for making score predictions of matches. It is a very laborious task to analyze all these metrics one by one and to come up with a result. In this study, score estimation of the matches was made by considering the statistical metrics. Here, based on the previously played match data, it is determined what kind of result the team can get in the second half, what they need to do to win the match and what the match result will be. The estimation process takes place in the light of data collected from a particular website. Machine learning algorithms were used while performing these operations. In addition, this study has been made useful with the user interface.

Article History
Received
16 Oct, 2021
Accepted
30 Dec, 2021

1. Introduction

In today's world, many data are emerging in different fields. This data needs to be processed in order to make sense. The processed data is of great importance for the progress of the business both in personal and corporate structures. One of the systems used to analyze data is decision support systems. Decision support systems provide support to the user in decision-making in various situations by using different models. In this way, the person in the position of decision maker can make faster and more accurate decisions. These systems obtain information about the flow of data through various algorithms using previously obtained data. Then, by applying the learned flow on new incoming data, it draws an estimation result close to the truth. Thanks to decision support systems, corporate control in

[1] ORCID: 0000-0003-3204-6620. BSc., Dokuz Eylul University, Faculty of Engineering, Department of Computer Engineering, Izmir/Turkey, mustafa.seyrek@ceng.deu.edu.tr
[2] ORCID: 0000-0001-9783-4041. BSc., Dokuz Eylul University, Faculty of Engineering, Department of Computer Engineering, Izmir/Turkey, enes.kiriktir@ceng.deu.edu.tr
[3] Corresponding Author. ORCID: 0000-0002-0353-5014. Asst. Prof., Dokuz Eylul University, Faculty of Engineering, Department of Computer Engineering, Izmir/Turkey, y.dogan@deu.edu.tr
[4] ORCID: 0000-0002-0395-4585. Asst. Prof. Ege University, Faculty of Sport Science, Departemnet of Coaching Education, Izmir/Turkey, bulent.yapicioglu@ege.edu.tr

companies increases and at the same time, it saves time by speeding up problem solving. In addition to these, managerial processes are becoming more automated. These systems are widely used in many areas such as finance, health, and sports. In this work to be done, a decision support system has been created by using statistics from football matches. Thanks to this system, a prediction about the match results can be made. Today, football has become a sport branch that is much more popular than other sports branches. In football, which is of great importance both in material and spiritual terms, there is not only national but also international competition. Football has a significant financial value based on companies, states and football employees. Especially during the transfer periods, player and technical team transfers significantly increase the monetary transition between countries and create great competition among the supporters of the teams. Considering that hundreds of matches are played every week around the world, it can be clearly seen how common it is as a sport. Football matches are played under much better conditions than before. The development of factors such as the field, equipment and technical team makes this sport much more common day by day. With the improvement of such conditions, technology has begun to enter football. With the introduction of technology, much deeper match and player analyzes are made. Based on these analyzes, technical teams can determine their own game styles. It should be noted that the priority for this kind of analysis is the data to be collected during the match. Keeping these collected data only as raw data does not matter for technical teams. Correct analysis of these collected data is very important in determining the play styles of the teams. The first problem that arises here is which method and algorithm have been used to process the obtained data. It is also very important high the accuracy percentage of the results to be obtained in the process of data processing. Another issue that should not be ignored here is what kind of result has been obtained according to the available data. It is important not only to process the data, but also to carry out a prediction process with new data in the light of the processed data. In short, in the light of the match statistics obtained in the past, being able to make a prediction for the matches to be played later is another problem that arises. In order to overcome these problems, football match statistics has been collected in this study and analyzes have been made with these statistics. A decision support system has emerged in the light of the analyzes to be made.

In this study, a decision support system was obtained by using only the match and player statistics of the Spanish La Liga league (Zaveri et al., 2018). The data used here was obtained from the database of the game named “FIFA 18”. Here, it is aimed to analyze situations such as the rate of winning a match and how the staff has been shaped. Various machine learning algorithms have been used to analyze the obtained data. These algorithms are implemented in the Python programming language. These algorithms were run using the available data and their accuracy values were calculated. The most successful algorithm has been obtained in this way. The starting point of this study is to process the match data and make it more meaningful, so that the users have had an idea of what kind of outcome may occur in the next matches. In addition, thanks to this study, technical teams have been able to see what kind of match results have emerged during the half-time of the matches according to the statistics in the first half. Consequently, the technical

team has been able to take to the field in the second half of the match with a different style of play. It was aimed to collect the necessary data during the study and to process these data with various algorithms. With the decision support system that has emerged in this way, analyzes can be performed using match data.

Two different implementations have been made in this study. The first of these concerns the acquisition of data. It is the process of retrieving statistics by accessing the source codes of a specified website. The data received has been stored in data files for use. The second part is about the analysis process of previously obtained data. At this point, the data has been made more meaningful by going through various processes. Later, these data have been analyzed and predicted using algorithms. Both implementations have been realized as separate desktop applications.

In this study, a desktop application has been implemented by using C# programming language in the process of obtaining data. Visual Studio environment has been used during the implementation phase. HtmlAgilityPack package has been used for data extraction from the website. In this way, statistics can be obtained from the source codes of the site.

In the process of data analysis, the desktop application has been implemented in the Visual Studio environment with the C# programming language. Weka has been used for analysis. In this way, algorithms can be accessed, and necessary analysis processes can be performed.

2. Related Works

While searching the literature regarding this study, the subjects included in this study were taken into consideration. How the methodology, tools and subjects used in this study are used in other studies has been investigated. While doing this research, the advantages and disadvantages of each are examined in detail. Information on these examinations is presented in the rest of the section.

In this study, a decision support system was developed for nurses to use (Purkuloğlu et al., 2019). Thanks to this system, it is aimed that nurses do their work such as recording and document preparation in a shorter time and more effectively. At the same time, the process of diagnosing patients is managed with this system. The system has been developed as a desktop application. Various units in the hospital were used to test the accuracy of the study. Thanks to the feedback received from these units, improvements were made in the decision support system.

In this study, a decision support system is presented for the detection of breast cancer, which is common in women (Alpaslan, 2019). In this system, cancerous structures were identified and classified. During the creation of this system, Convolutional Neural Networks were used. The results of the experiments were found to give better results when compared with the results of the articles published in this field. As a result, it is aimed to provide doctors with a great advantage in the diagnosis of breast cancer in this decision support system.

In this study, a decision support system is proposed to assist buyers in purchasing commercial taxis (Canbazoğlu et al., 2018). This decision support system is

presented to the user with a mobile-based application. Thanks to this system, the person who will buy a commercial taxi will be able to choose a vehicle according to the criteria he has determined instead of getting advice from the people around him. There are a certain number of tools available here. AHS and MOORA techniques were used in this study. These techniques have been used to identify and rank criteria and tools.

In this study, a decision support system created for universities is proposed (Aydemir, 2019). In this system, the conditions for the appointment and promotion of academic staff were examined, and as a result, some criteria were determined. These criteria determined in this decision support system were used. In this way, different universities will be able to create success rankings in different fields according to their criteria. Thanks to this system, individuals using it will be able to see their success rankings and evaluate themselves.

In this study, a decision support system related to the stage of changing computers in institutions is presented (Yorulmaz et al., 2019). In this system, the hardware features of the computers are determined and listed in certain ways. The TOPSIS method was used in this system. This method is used to enumerate the selected states within themselves. This system was implemented and used in a faculty. The computers here are listed with this system and it has facilitated the administrators to examine and rate the computer systems.

In this study, the heart sound recordings were determined by using various data sets (Yıldız and Zan, 2019). K-Nearest Neighbors and Support Vector Machine algorithms were used during the classification of heart sounds data. Here first, the normalization of the data has been done. Afterwards, the features that will not be used in the data set were removed and the data obtained were separated as test and train in order to be used in algorithms. As a result of the study, a system was developed to assist doctors in making a diagnosis by analyzing data with these algorithms.

This article covers a study of detecting fake websites (Korkmaz and Büyükgöze, 2019). In this study, the status of the website was determined by using certain features of the websites. The data set in the study was taken from UCI. The study is programmed in the R programming language. Support Vector Machine, Random Forest, J48, Naïve Bayes, K-Nearest Neighbors algorithms were used to detect fake websites. When the values of these algorithms on the data set were compared, it was seen that the Random Forest algorithm was more successful in detecting fake websites.

In this study, a system to assist doctors in the treatment of warts is proposed (Tanyıldızı et al., 2018). Here, it is aimed to determine the patient-specific treatment method by means of various data mining algorithms. At the stage of determining the treatment method, the data in the data set was classified and the patterns in these data sets were taught to the algorithms. Afterwards, the accuracy values of these algorithms were compared, and it was determined which algorithms would be successful in determining the treatment method of warts. According to the results obtained, K-Star and Random Forest algorithms have yielded successful results. The reason for the emergence of two different

algorithms is that there are two different treatment methods. In this way, it can be concluded that which treatment method will be more successful to apply to the patient.

In this study, intrusion detection systems, which are generally used to secure internet services, are emphasized (İnce et al., 2021). The success of these systems is determined according to the results they give, that is, success is measured according to the correct classification of attack types. In this study, measurements and comparisons have been made using some methods. These methods; K-Nearest Neighbors, Extreme Learning Machines, Support Vector Machines and Random Forest methods. Results were obtained according to the data set given to these methods. As a result of this study, the most successful method was found to be the excessive learning machines method.

In this study, a study is presented for users who want to reach the websites to be accessed in a fast, accurate and safer manner (Chouseinoglou and Şahin, 2019). Web sites can be classified for these requests. In this study, the classification processes have been tested with the Binary Classification and Multiple Classification methods. Different algorithms are used in these methods, and they are coded in the Python programming language. As a result, the test results of both methods used and the algorithms in this method are shown separately.

In this study, research was presented to predict the academic success of students using particular software (Güldal and Çakıcı, 2017). These estimates have been made with various classification algorithms. These; The Decision Tree (C4.5) is the K-Nearest Neighbors and Naïve Bayes algorithms. For the analysis of the data, the data set was run with these algorithms. Weka software was used for this. Algorithms and accuracy values were compared, and it was seen that the most successful algorithm in this study was the K-Nearest Neighbors algorithm.

In this study, it was aimed to predict the breast cancer potential with the help of various data collected (Sarıtaş and Yaşar, 2019). While making this estimation, routine blood analysis parameters and anthropometric data were used. These data used were collected from patients admitted to the clinic with suspicion of breast cancer. Artificial Neural Networks and Naïve Bayes classification algorithms are used for the estimation process. Then the performances of these algorithms were compared. Matlab R2017b's Neural Network Toolbox and Naïve Bayes were used to implement these algorithms. As a result, the accuracy values were obtained as 86.95% with ANN and 83.54% with Naïve Bayes. Thus, it was concluded that these two algorithms can be used for early breast cancer diagnosis.

In this study, the performance evaluation of the strikers in football has been made (Evwiekpaefe et al., 2020). The aim is to create a prediction mechanism by selecting the most effective striker in the matches. Data of 100 strikers playing in different leagues in Europe were used in the study. Artificial Neural Networks and J48 classification algorithm were used in the analysis of the data. The study was completed in the Weka environment, and it was observed that the Artificial Neural Networks classification algorithm gave more successful results. Considering the data set used in the study, it was observed that the cross-validation score of the Artificial Neural Networks algorithm was 68%.

In this study, a prediction model has been developed to predict the match results in the English Premier League (Baboota and Kaur, 2019). The data set used includes data sets of matches played over 11 seasons. After the received data passed through certain data pre-processing stages, estimations were made with various algorithms. Here, Naïve Bayes, Random Forest, Support Vector Machine and Gradient Boosting estimation algorithms are used. Scikit-Learn and Pandas packages are used in the study using the Python programming language. As a result of the study, it was seen that the Gradient Boosting algorithm provided the best result among the prediction algorithms used.

In this study, a model for predicting football match results is proposed (Rudrapal et al., 2020). The data obtained from various sites were made ready to be analyzed by passing through pre-processing stages. Multi-Layer Perceptron, Support Vector Machine, Gaussian Naïve Bayes and Random Forest algorithms are used as algorithms. As a result of running these algorithms with the data set, the algorithm with the highest accuracy score and F1-score was determined as Multi-Layer Perceptron. The accuracy score of this algorithm was 73.57% and the F1-score was 71.45%.

In this study, the tactical behavior of the players in beach volleyball has been classified (Wenninger et al., 2020). The data set used in the study includes data from 569 male and 787 female beach volleyball players. The data were made ready for analysis by passing through pre-processing stages. In the process of data analysis, Multi-Layer Perceptron, Recurrent Network, Convolutional Network and Gradient Boosted Tree algorithms were used.

In this study, a multiple regression was developed to analyze team performances in basketball game and to predict the game (Lumbao et al., 2017). Here, dependent and independent variables are determined to be used in analysis and estimation processes. The prepared data has been subjected to some preliminary processes and arrangements. These data were analyzed with the help of Multiple Regression. With this study, a prediction has been made that will benefit basketball players and basketball coaches for the competitions in the upcoming seasons.

In this article, a decision support system is presented to help football coaches choose between different players (Salman et al., 2018). Different characteristics and criteria of football players were used in the selection phase. These criteria have been kept in a database. From here, it is sent to the model that will decide and is presented to the user. All these operations have been implemented as an interface. Visual Basic .net is used for these. As a result of this study, an easy-to-use product has emerged.

In this study, a decision support system has been put forward by predicting which position a football player should play in the match (Jauhari et al., 2018). Thanks to this system, it will be easier for the football coach to make decisions about the players and maximum efficiency will be tried to be achieved. Player data will be sent to this system created and the system's suggestion will be received. Simple Additive Weighting method is used here. It has been observed that the system built gave results with 50% accuracy.

This article presents a system that can help football managers decide which player to choose when transferring football players (Young and Weckman, 2020). Here, the Heuristic Evaluation of Artificially Replaced Teammates (HEART) methodology is proposed. This method is a mathematical model that uses machine learning, statistical-based methodology and helps managers with recruitment. Thanks to this method, the estimated contribution that the chosen player can make to win the team will be determined. As a result of this study, it has been seen that this method provides significant benefit in the decision-making phase.

3. Functional Requirements

3.1. Functional Requirements for Data Collection

With the "private void button1_Click()" function, the previously created txt file containing the match links is selected. With this button, the user is directed to the file manager on his computer and the user can select the txt file he wants. Here, a single file selection and only file selection with txt extension are allowed. If the file selection fails, an error message is given to the user. If the file selection is successful, this selected file is read in the function.

With the "private void button2_Click()" function selects the csv file where the data will be saved. Clicking the button here, the user is directed to the file manager on his computer and the user is expected to select the file with csv extension. In case of any error in file selection, an error message is given to the user. Only file selection with csv extension is allowed. If the write to file process is successful, the user is informed that the transaction was successful.

With the "private void button3_Click()" function selects the csv file where the data will be saved for analysis. Clicking the button here is directed to the file manager on the user's computer and the user is expected to select the file with the csv extension. If there is any error in the file selection, an error message will be given to the user. Only file selection with csv extension is allowed. If the writing to the file is successful, the user will be notified that the operation was successful.

Thanks to the "private void islemler()" function, data is received from the site according to the connections. The received data is shown with datagridview on the form screen. The function does not take any parameters and there is no return value.

Thanks to the "private void csvYazma(DataGridView gridIn, string outputFile)" function, the data in the datagridview is written to the file with csv extension. Here, datagridview with data is given as a parameter to the function. In addition, the path of the file with the csv extension is given as a string value to the function.

Thanks to the "private void analiz(DataGridView gridIn, string outputFile)" function, the data in the datagridview is written to the file with csv extension for analysing. Here, datagridview with data is given as a parameter to the function. In addition, the path of the file with the csv extension is given as a string value to the function.

3.2. Functional Requirements for Analysis Processes

"public static double NaiveBayesTest(weka.core.Instances insts)", "public static double LogRegressionTest(weka.core.Instances insts)", "public static double Knn (weka.core.Instances insts)", "public static double J48classifyTest(weka.core.Instances insts)", "public static double RandomForestTest(weka.core.Instances insts)", "public static double RandomTreeTest(weka.core.Instances insts)", "public static double ArtificialNN(weka.core.Instances insts)", "public static double SVM (weka.core.Instances insts)" functions show the functions of the algorithms. Here, algorithms are used by importing the Weka library. These functions belong to 8 different algorithms and the data set obtained is run in each algorithm. Functions return the results of accuracy as a double.

Thanks to the "private void clears()" function, it clears the previous texts on the form screen. The function does not take a parameter and there is no return value. The csv file selected with the "private void btnBrowse_Click()" function is sent to the previously mentioned 8 algorithms and the results are calculated. The calculated results are shown on the form screen and the algorithm with the highest score is determined as the most successful algorithm. In addition, a dynamic textbox list is opened according to the attributes in the data set used. the user can enter the desired values to be predicted into the textboxes that find here. With the "private void btnDiscover_Click()" function, the values entered by the user for estimation are sent to the algorithm that was previously selected as the most successful. The value obtained as a result of the estimation process is presented to the user on the form screen.

4. Non-Functional Requirements

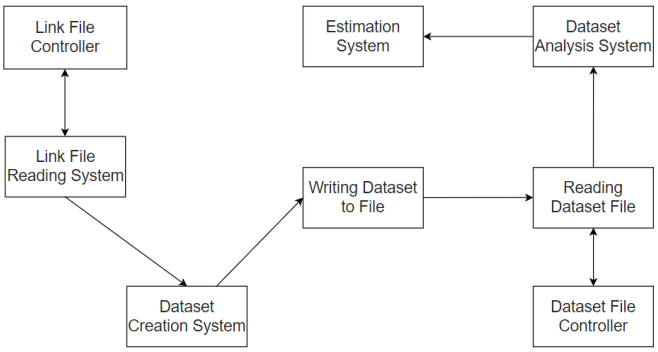
This study will be developed by using C# programming language to be used in Windows operating system. This application developed will be presented to the user as a desktop application. The user will be able to use the application without any registration and login process. The application can work without any problems during use. It will provide quick feedback to the user during the withdrawal of data from the website and during the analysis process. This will show how low the response time of the application is.

In the data collection part of the application, the accuracy of the data received from the site will be tested by comparing it with the data on the site. Here it will be seen that the data has been received correctly. During the analysis process, the results will be compared manually. The interface design of the application will be simple in terms of ease of use. The user will be able to do all his operations easily. In this way, the user will not have any problems while using the application. This will increase the usability of the application. The user will also be directed with the necessary error and information messages. Thus, the application will be a user-friendly application.

5. Design

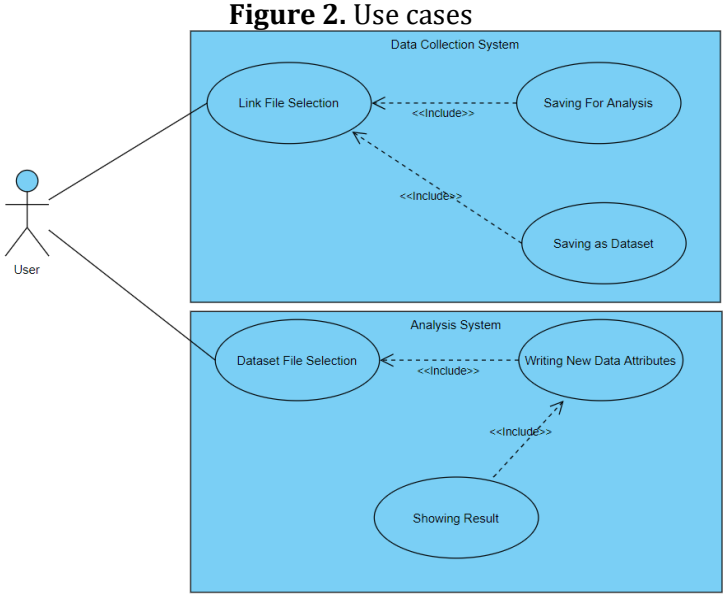
Figure 1 shows the architectural view design for this project. In this design, two controllers and six systems are shown. These control the operation of the user and enable the program to fulfil their mechanism.

Figure 1. Architectural view



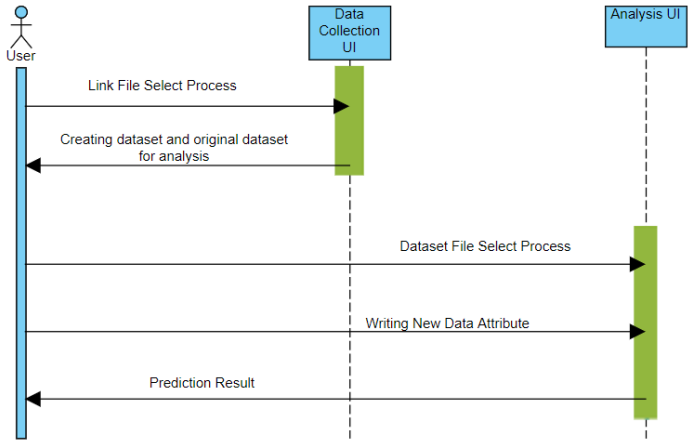
With Link File Controller and Link File Reading System, the user selects the correct link file and performs reading of this file. With Dataset Creation System, it collects statistical data from previously read links. Data collected also with Writing Dataset to File are stored in files of appropriate formats. Thanks to Reading Dataset File, previously collected data is read from the file and controlled with Dataset File Controller. After these, data sets are analyzed by Dataset Analysis System. Finally, estimation processes are carried out with the Estimation System. The use cases diagram of the project is shown in Figure 2. The use case diagram is used here to explain the functionality of the system. Thanks to this diagram, the functionality of the entire system can be shown. As can be seen in this diagram, there are two different systems. The user can perform different operations on both systems.

Figure 2. Use cases



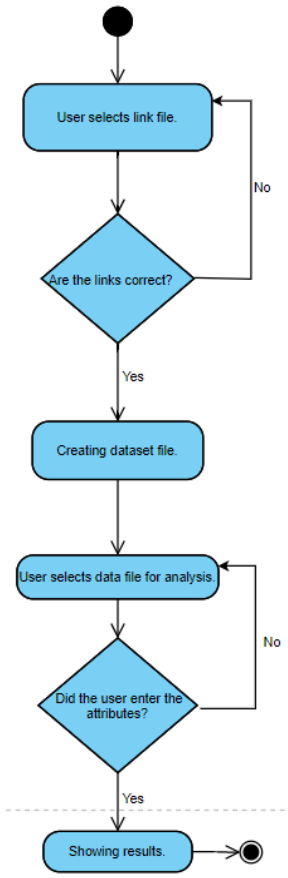
The sequence diagram of the project is shown in Figure 3. With this diagram, it is aimed to show the communication of objects with each other in order.

Figure 3. Sequence diagram



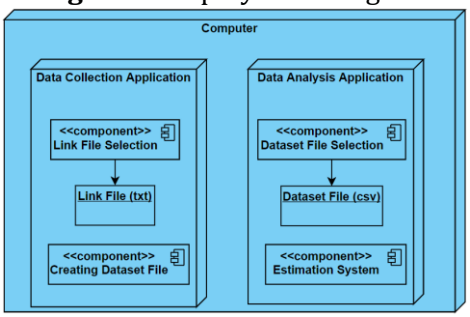
The activity diagram of the project is shown in Figure 4. Thanks to the activity diagram, it can be seen how the activities work in coordination with each other. First, the user is expected to select the file containing the links of the matches. Then it is checked whether the connections are correct or not. In case the connections are correct, dataset files are created by taking the data. In the next step, the user is expected to select the file containing the data set created for analysis. As a result of the analysis made according to the selected data set, the user is expected to enter new data for estimation. In the last step, the result of the estimation process for the new data entered by the user is shown.

Figure 4. Activity diagram



The distribution scheme of the project is shown in Figure 5. Thanks to this diagram, it can be seen how the hardware and software of the system are placed. There are two different applications in the system. Here, the output of an application is the input for the other application.

Figure 5. Deployment diagram



6. Implementation

6.1. Data Collection Operations

In this section, first, how the links of the match statistics are collected and what has been done while collecting is explained. Afterwards, it was explained how the data was obtained from the match statistics links collected. It was previously stated in the report that the data of the matches were received via links. The links of the match data were collected in a single file and transferred to the next data acquisition stage. In this section will explain how these links are obtained. The basic algorithm established during the collection of links is to check whether the links created with the code are valid. The current link is stored for writing to the txt file. Finally, valid links are written to the file with "txt" extension by means of a function. While receiving links from the site, it was observed that the statistics page of some matches was empty. For this reason, after checking whether the generated link exists or not, the status of the statistics page on the existing link is checked. The general algorithm logic for collecting links is as described above. A more detailed explanation of this algorithm with code fragments is given in the continuation of this part of the report. The description of the structures used and deemed important in the link collection operation is given in Figure 6.

Figure 6. Data Structure definitions

```
HtmlWeb web = new HtmlWeb();
DateTime matchDate; // to receive matches before a certain date
List<String> urls = new List<String>(); // for correct urls
List<String> teamNames = new List<String>(); // for spain league teams
Dictionary<String, int> findingTeam = new Dictionary<String, int>();//for matched teams
```

Here, a "web" variable of "HtmlWeb" type that will be used to get html codes has been created. A "matchDate" of "DateTime" type was created in order to scan the days of the matches. In order to store valid links for the purpose of writing to the file, a "List" of "String" type has been created. Also, in Figure 6 there is "teamNames" where the names of the Spanish league teams are stored. Finally, the "Dictionary" structure was used to determine the teams that were found to have matches within a week. Details of the dictionary structure will be explained later in the report. An example of team names is given in Figure 7. Here, the team names are written in the format specified in the links of the site. The figure shows two

teams in the Spanish league as an example. Team names are kept in the “List” structure as can be seen in Figure 7.

Figure 7. Team name list example

```
teamNames.Add("atletico-madrid");
teamNames.Add("fc-barcelona");
```

It was mentioned in the report that the Dictionary structure was used. The aim here is to determine the team whose matches are determined within a week. As seen in Figure 8, an example of two teams in the Dictionary structure is given. Here, the name of the team is used as the “key” value, and 0 and 1 are used as the “value” value. A “0” means that the team's match in that week has not been found yet. “1” indicates that the link of the team's match with another team has been detected. In this way, while checking whether the teams have matches or not, the teams that are already matching are not taken into account. As a result, thanks to this structure, which was established especially during the search for links, it was possible to save search time.

Figure 8. Usage example of the dictionary structure

```
findingTeam.Add("atletico-madrid",0);
findingTeam.Add("fc-barcelona", 0);
```

In the code snippet seen in Figure 9, the “newMatchDate” variable is used to check the matches of the teams in the list with other teams day by day. Here, “dateCounter” variable is used to control the previous days. Here, the day name of that day is kept with the variable “matchDay”. Here, considering only match days, no search was made on other days. In this way, no calls were made on days without a match.

Figure 9. Determination of match days

```
var newMatchDate = matchDate.AddDays(-dateCounter);//scanning backwards matches
String matchDay = newMatchDate.DayOfWeek.ToString();
if (matchDay == "Friday" || matchDay == "Saturday" || matchDay == "Sunday" || matchDay == "Monday")...
```

In the code fragment seen in Figure 10, it is checked whether each team has a match with other teams. The basic logic of the algorithm here is that a selected team is considered the host and individual links are created for their matches with other teams.

Figure 10. Match searching

```
for (int i = 0; i < teamNames.Count; i++)
{
    String homeTeam = teamNames[i];
    // If this week's match of the wanted team has been found before
    if (findingTeam[homeTeam] == 1)
    {
        continue;
    }

    for (int j = 0; j < teamNames.Count; j++)...
}
```

Later, it is checked whether these links really exist or not on the site. If the link is correct, it will be saved for later printing. The team is selected with the first for loop in Figure 10 and a comparison is made with the other for loop inside. Here, before making a comparison, it is checked to see if the chosen team's match of that week has been found before. This control section is expressed in the code with an “if statement”. If the match of the team has been played before, the other team is

passed without comparing it with the “continue” structure. In the code snippet in Figure 11, link creation process is included by taking the date together with the host and away teams. Here, the fixed parts of the links are combined with the team names and date parts, which are variable data, and placed in the variable named “url”. Later, thanks to the “Load()” function, the html codes of the page with the link are assigned to the “doc” variable in HtmlDocument type.

Figure 11. Link building and receiving Html codes

```
String url = "https://skorexpress.com/football/matches/"; // the fixed part of the url
url = url + homeTeam + "-" + awayTeam + "-" + strMatchDate + "/statistics"; //creating url
HtmlDocument doc = web.Load(url);//getting the html codes of the page
```

After checking whether the html codes are “null”, if it is not “null”, that is, if the link is valid, the codes in Figure 12 are executed. Here, the team names in the Html code are taken with the “SelectSingleNode()” function. After the necessary split operations are done, it is checked whether the word “Bayanlar” is present in the team’s name. Since women's football teams are not widespread, this study is carried out only on men's football teams. After the examination made on the site, it was determined that the matches of the women's football teams of the same two teams were found on the same link. In order to avoid confusion here, the “if” structure shown in Figure 12 has been used. If the link contains the statistics of the matches of men's football teams, this link is stored in the List structure. Finally, “value” values are assigned as “1” so that teams with matches do not participate in the search over again.

Figure 12. Operations made if the link is valid

```
//if url is valid
var teams = doc.DocumentNode.SelectSingleNode("/html/body/div[3]/div[1]/main/div[3]/div[3]/h4");
string[] temp = teams.InnerText.Split('-');
homeTeamName = temp[0].Substring(14);
if (homeTeamName.Contains("Bayanlar"))
{
    continue;
}
urls.Add(url);
findingTeam[homeTeam] = 1;
findingTeam[awayTeam] = 1;
break;
```

In the code snippet seen in Figure 13, it is seen that the “dateCounter” variable has been increased in order to scan the days of the past. In the “if” structure here, if the current day is “Friday”, because a new week will start, the “value” values in the Dictionary of all teams are assigned “0”. Finally, the List structure containing the links is sent to the function that enables writing to the file, and all the links obtained are written to the file with the “txt” extension.

Figure 13. The function of determining the new week and writing to the file

```
    dateCounter++;// to look back to the day before
    //if a new match week is starting
    if (matchDay == "Friday")...
}
writeToFile(urls);
Console.WriteLine("Operation is successful !");
Console.ReadKey();
```

In the screenshot shown in Figure 14, there is a console screen indicating that the collection of the links and the writing to the file were successfully completed.

Figure 14. Result screen

```
C:\Users\enesa\source\repos\VeriCekmeOtomatik\VeriCekmeOtomatik\bi...
Operation is successful !
```

6.2. Data Sets Creation Operations

In this section, the process of taking the data belonging to the matches in the links collected before, from the website and writing to files in csv format is explained. These operations are implemented as a form application in the C# programming language.

HtmlAgilityPack package, shown in Figure 15, used to extract data in C#, has been added to the C# form application in Visual Studio from the Nuget Package Manager section. This package provides the proper parsing of the source code in Html format on the website. Thanks to this package, statistics data could be drawn easily.

Figure 15. HtmlAgilityPack package installations

```
HtmlAgilityPack by ZZZ Projects,Simon Mourrier,Jeff Klawiter,Stephan Grell                v1.11.31
This is an agile HTML parser that builds a read/write DOM and supports plain XPATH or XSLT (you actually don't HAVE to understand XPATH nor XSLT to
use it, don't worry...). It is a .NET code library that allows you to parse "out of the web" HTML files. The parser is very tolerant with "real world" malfor...
```

HtmlAgilityPack package, which was previously added to the project, was used in C# as seen in the code snippet in Figure 16. In this 2-line piece of code, the necessary variables are defined and the links read line by line from the txt file are processed here. Source codes in Html format in the link are kept in the variable named “doc”. In this way, the connection between the website and the application is established.

Figure 16. Using HtmlAgilityPack.

```
HtmlAgilityPack.HtmlWeb web = new HtmlAgilityPack.HtmlWeb();
HtmlAgilityPack.HtmlDocument doc = web.Load(line);
```

It is necessary to retrieve the data in the desired parts of the source code previously kept in the variable named “doc”. This requirement is provided in Figure 17. The XPaths of the site have been added to the code in order to access the desired points in the source code. If there is more than one data, “DocumentNode.SelectNodes” expression is used, if there is a single data, “DocumentNode.SelectSingleNode” expressions are used. As can be seen from the figure, these data are assigned to the necessary variables. To give an example of one of these, the “e-Gol” variable here refers to the number of goals scored by the home team. And because it is a single data, “DocumentNode.SelectSingleNode” expression is used.

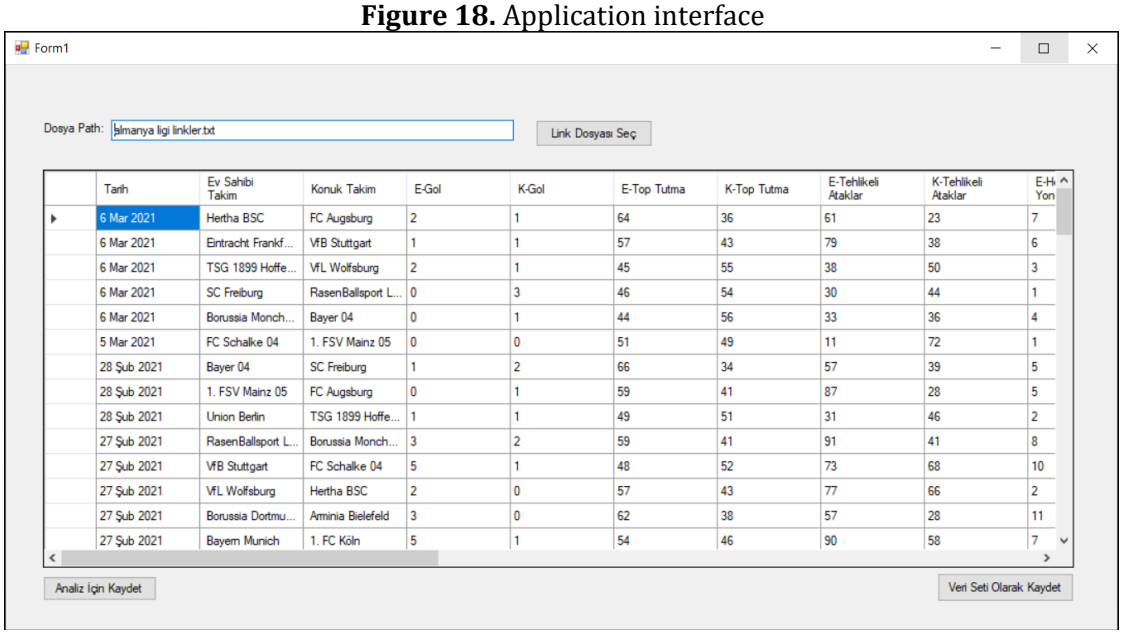
Figure 17. Assigning data to variables using XPaths.

```
//siteden veri alma
var evSahibiVeri = doc.DocumentNode.SelectNodes("/html/body/div[3]/div[1]/main/div[3]/div[1]/div[*]/div[1]/span[1]");
var ist = doc.DocumentNode.SelectNodes("/html/body/div[3]/div[1]/main/div[3]/div[1]/div[*]/div[1]/span[2]");
var konukTakimVeri = doc.DocumentNode.SelectNodes("/html/body/div[3]/div[1]/main/div[3]/div[1]/div[*]/div[1]/span[3]");
var tarih = doc.DocumentNode.SelectSingleNode("/html/body/div[3]/div[1]/main/div[2]/div[2]/div[2]/div[3]/span[1]");
var eGol = doc.DocumentNode.SelectSingleNode("/html/body/div[3]/div[1]/main/div[2]/div[2]/div[2]/div[1]/span[1]");
var kGol = doc.DocumentNode.SelectSingleNode("/html/body/div[3]/div[1]/main/div[2]/div[2]/div[2]/div[1]/span[2]");
var takimlar = doc.DocumentNode.SelectSingleNode("/html/body/div[3]/div[1]/main/div[3]/div[3]/h4");
```

In Figure 18, the C# form interface belonging to the application can be seen. There are three buttons, one label, textbox and datagridview in this interface. Here, the

path of the link file is written in the textbox. In the datagridview, the data of the matches in the selected links are listed. On the sample form screen, the data of 100 games played in the German league are listed.

Figure 18. Application interface



Tarih	Ev Sahibi Takim	Konuk Takim	E-Gol	K-Gol	E-Top Tutma	K-Top Tutma	E-Tehlikeli Ataklar	K-Tehlikeli Ataklar	E-H Yon
6 Mar 2021	Hertha BSC	FC Augsburg	2	1	64	36	61	23	7
6 Mar 2021	Eintracht Frankf...	VfB Stuttgart	1	1	57	43	79	38	6
6 Mar 2021	TSG 1899 Hoffe...	VfL Wolfsburg	2	1	45	55	38	50	3
6 Mar 2021	SC Freiburg	RasenBallsport L...	0	3	46	54	30	44	1
6 Mar 2021	Borussia Monch...	Bayer 04	0	1	44	56	33	36	4
5 Mar 2021	FC Schalke 04	1. FSV Mainz 05	0	0	51	49	11	72	1
28 Şub 2021	Bayer 04	SC Freiburg	1	2	66	34	57	39	5
28 Şub 2021	1. FSV Mainz 05	FC Augsburg	0	1	59	41	87	28	5
28 Şub 2021	Union Berlin	TSG 1899 Hoffe...	1	1	49	51	31	46	2
27 Şub 2021	RasenBallsport L...	Borussia Monch...	3	2	59	41	91	41	8
27 Şub 2021	VfB Stuttgart	FC Schalke 04	5	1	48	52	73	68	10
27 Şub 2021	VfL Wolfsburg	Hertha BSC	2	0	57	43	77	66	2
27 Şub 2021	Borussia Dortmu...	Arminia Bielefeld	3	0	62	38	57	28	11
27 Şub 2021	Bayern Munich	1. FC Köln	5	1	54	46	90	58	7

The codes of the "Link Dosyası Seç" button, one of the interface elements, are shown in Figure 19. In these codes, it is possible to select files with txt extension on the computer. Depending on whether the selected file is in a correct link format, an error message is given or the process continues. Links in the selected file are read line by line. Each line is assigned to a variable named "line". This variable is also sent into the "web.load()" function in Figure 16. In this way, the situations described in Figure 17 are realized step by step correctly.

Figure 19. "Link Dosyası Seç" button code

```
//link secme butonu
1 reference
private void button1_Click(object sender, EventArgs e)
{
    //dosya secme ve dosya okuma islemleri
    OpenFileDialog file = new OpenFileDialog();
    file.Filter = "Text|*.txt|All|*.*";
    file.Multiselect = false;
    file.Title = "Lütfen link dosyasını seçiniz!";
    if (file.ShowDialog() == DialogResult.OK)
    {
        textBox1.Text = file.FileName;

        try
        {
            if (textBox1.Text.Length < 1)
            {
                MessageBox.Show("Lütfen dosya seçiniz!", "Error Message!");
                textBox1.Text = "";
            }
            else
            {
                StreamReader sr = new StreamReader(textBox1.Text);
                line = sr.ReadLine();
                while (line != null)
                {
                    islemler();
                    line = sr.ReadLine();
                }
                sr.Close();
            }
        }
        catch (Exception e2)
        {
            MessageBox.Show(e2.Message, "Error Message!");
        }
    }
}
```

The function of the "Veri Seti Olarak Kaydet" button, which is one of the interface elements, is shown in Figure 20. In this function, a csv file in the computer is selected. Thanks to the "file.Filter" expression in this function, file type filtering can be done. Again, the "file.Multiselect" statement here determines whether more than one file can be selected at the same time. If no error is encountered, the data in datagridview is saved in the csv file. An information message appears on the screen after the data has been successfully saved. The data recorded with this function holds more detailed information about the matches. These are not used for analysis.

Figure 20. "Veri Seti Olarak Kaydet" button code

```
1 reference
private void button2_Click(object sender, EventArgs e)
{
    OpenFileDialog file = new OpenFileDialog();
    file.Filter = "CSV (*.csv)|*.csv";
    file.Multiselect = false;
    file.Title = "Lütfen veri seti dosyasını seçiniz!";
    String path = "";
    if (file.ShowDialog() == DialogResult.OK)
    {
        path = file.FileName;
         try
         {
             if (path.Length < 1)
             {
                 MessageBox.Show("Lütfen dosya seçiniz!", "Error Message!");
                 textBox1.Text = "";
             }
             else
             {
                 csvYazma(dataGridView1, path);
                 MessageBox.Show("Veri seti başarılı bir şekilde kaydedildi!", "Bilgi Mesajı");
             }
         }
         catch (Exception e2)
         {
             MessageBox.Show(e2.Message, "Error Message!");
         }
    }
}
```

The codes of the "csvYazma()" function in the "Veri Seti Olarak Kaydet" button shown in Figure 20 are shown in Figure 21. With this function, the data in the datagridview is added line by line, including the headings, to a selected csv file. This function takes two parameters. The first is datagridview, the second is the path to the csv file. After getting these parameters, it browses the datagridview with for loops and retrieves the headers and data one by one. Here, the output file is given in the "StreamWriter" and thus the information here is written to the csv file. While doing this, the expression "Encoding.UTF8" is used to avoid any corruption in the texts.

The function of the "Analiz İçin Kaydet" button, which is one of the interface elements, is shown in Figure 22. In this function, a target csv file has been selected and the analysis data has been saved there. These data are taken from datagridview. However, the data in the datagridview was not taken as it is. Some deductions have been made in the data. An information message is displayed on the screen after the operation has been completed successfully. If it is not successful, an error message is shown

Figure 21. Writing to csv from datagridview

```
//csv ye yazma fonksiyonu
1 reference
private void csvYazma(DataGridView gridIn, string outputFile)
{
    if (gridIn.RowCount > 0)
    {
        string value = "";
        DataGridViewRow dr = new DataGridViewRow();
        StreamWriter swOut = new StreamWriter(outputFile);

        for (int i = 0; i <= gridIn.Columns.Count - 1; i++)
        {
            if (i > 0)
            {
                swOut.Write(",");
            }
            swOut.Write(gridIn.Columns[i].HeaderText);
        }
        swOut.WriteLine();
        for (int j = 0; j <= gridIn.Rows.Count - 2; j++)
        {
            if (j > 0)
            {
                swOut.WriteLine();
            }

            dr = gridIn.Rows[j];

            for (int i = 0; i <= gridIn.Columns.Count - 1; i++)
            {
                if (i > 0)
                {
                    swOut.Write(",");
                }
                value = dr.Cells[i].Value.ToString();
                value = value.Replace(',', ' ');
                value = value.Replace(Environment.NewLine, " ");
                swOut.Write(value,Encoding.UTF8);
            }
        }
        swOut.Close();
    }
}
```

The “analiz()” function in the “Analiz İçin Kaydet” button function in Figure 22 is shown in Figure 23. This function takes two parameters. The first is datagridview, the second is the csv file path.

Figure 22. “Analiz İçin Kaydet” button code

```
//analiz butonu
1 reference
private void button3_Click(object sender, EventArgs e)
{
    OpenFileDialog file = new OpenFileDialog();
    file.Filter = "CSV (*.csv)|*.csv";
    file.Multiselect = false;
    file.Title = "Lütfen analiz dosyasını seçiniz!";
    String path = "";
    if (file.ShowDialog() == DialogResult.OK)
    {
        path = file.FileName;
        try
        {
            if (path.Length < 1)
            {
                MessageBox.Show("Lütfen dosya seçiniz!", "Error Message!");
                textBox1.Text = "";
            }
            else
            {
              analiz(dataGridView1, path);
                MessageBox.Show("Analiz için veri seti başarılı bir şekilde kaydedildi!", "Bilgi Mesajı");
            }
        }
        catch (Exception e2)
        {
            MessageBox.Show(e2.Message, "Error Message!");
        }
    }
}
```

Some for loops in this function start from the value of “i=3”, unlike the loops in Figure 21. This is because the data in the first three titles are not received. Thanks to these for loops, the desired information in the datagridview can be successfully saved to the specified csv file. Here, too, it was saved to the file by means of “StreamWriter”. And “Encoding.UTF8” expression has been added to avoid character problems.

Figure 23. Saving to csv for analysis

```
//analiz için csv yazma fonksiyonu
1 reference
private void analiz(DataGridView gridIn, string outputFile)
{
    if (gridIn.RowCount > 0)
    {
        string value = "";
        DataGridViewRow dr = new DataGridViewRow();
        StreamWriter swOut = new StreamWriter(outputFile);

        for (int i = 3; i <= gridIn.Columns.Count - 1; i++)
        {
            if (i > 3)
            {
                swOut.Write(",");
            }
            swOut.Write(gridIn.Columns[i].HeaderText);
        }
        swOut.WriteLine();
        for (int j = 0; j <= gridIn.Rows.Count - 2; j++)
        {
            if (j > 0)
            {
                swOut.WriteLine();
            }

            dr = gridIn.Rows[j];

            for (int i = 3; i <= gridIn.Columns.Count - 1; i++)
            {
                if (i > 3)
                {
                    swOut.Write(",");
                }
                value = dr.Cells[i].Value.ToString();
                value = value.Replace(',', ' ');
                value = value.Replace(Environment.NewLine, " ");
                swOut.Write(value, Encoding.UTF8);
            }
        }
        swOut.Close();
    }
}
```

While performing data collection operations in the C# form application, multiple different data types were used. In Figure 24, a string array, a datatable and a dictionary structure in which the headings of the data sets are kept can be shown as an example. Dictionary structure consists of key and value values. In this example, both are of type strings. The elements in the “basliklar” array are added as the key value, and the value of the place belonging to the same title in the statistics data is added as the value. In this way, the data listed in the datagridview are taken from “dict”.

Figure 24. Different types of data used

```
DataTable table = new DataTable("Veriler");
Dictionary<string, string> dict = new Dictionary<string, string>();
string[] basliklar = {"Tarih","Ev Sahibi Takim", "Konuk Takim",
            "E-Gol","K-Gol","E-Top Tutma","K-Top Tutma",
        "E-Tehlikeli Ataklar","K-Tehlikeli Ataklar","E-Hedefe Yonelik Sutlar",
        "K-Hedefe Yonelik Sutlar","E-Kose Vuruslari","K-Kose Vuruslari",
        "E-Faul","K-Faul", "E-Kirmizi Kartlar","K-Kirmizi Kartlar",
        "E-Sari Kartlar","K-Sari Kartlar","E-Tac Atisi",
        "K-Tac Atisi", "E-Ofsaytlar", "K-Ofsaytlar","Sonuc" };
```

Figure 25 shows the saved format of the data in the csv file after clicking the “Veri Seti Olarak Kaydet” button.

Figure 25. Sample data set in csv file

```
6 Mar 2021, Hertha BSC , FC Augsburg,2,1,64,36,61,23,7,4,6,1,16,13,0,0,2,3,30,20,2,2,E-Galibiyet
6 Mar 2021, Eintracht Frankfurt , VfB Stuttgart,1,1,57,43,79,38,6,1,13,4,15,16,0,0,2,3,26,23,1,1,Beraberlik
6 Mar 2021, TSG 1899 Hoffenheim , VfL Wolfsburg,2,1,45,55,38,50,3,3,2,6,8,11,0,1,2,4,13,16,0,5,E-Galibiyet
6 Mar 2021, SC Freiburg , RasenBallsport Leipzig,0,3,46,54,30,44,1,9,1,2,10,14,0,0,1,1,22,23,0,1,K-Galibiyet
6 Mar 2021, Borussia Monchengladbach , Bayer 04,0,1,44,56,33,36,4,5,7,8,6,10,0,0,0,2,20,20,3,2,K-Galibiyet
```

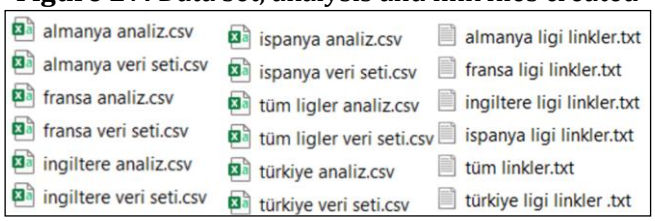
Figure 26 shows the analysis data saved in the csv file after clicking the "Analiz İçin Kaydet" button.

Figure 26. Sample analysis data set in csv file

```
2,1,64,36,61,23,7,4,6,1,16,13,0,0,2,3,30,20,2,2,E-Galibiyet
1,1,57,43,79,38,6,1,13,4,15,16,0,0,2,3,26,23,1,1,Beraberlik
2,1,45,55,38,50,3,3,2,6,8,11,0,1,2,4,13,16,0,5,E-Galibiyet
0,3,46,54,30,44,1,9,1,2,10,14,0,0,1,1,22,23,0,1,K-Galibiyet
0,1,44,56,33,36,4,5,7,8,6,10,0,0,0,2,20,20,3,2,K-Galibiyet
```

Figure 25 and Figure 26 show the values of 5 sample matches of the German league. However, there are currently no statistics titles in these images. The first line in the csv file belongs to the headers. The data set section here also keeps the date and team names of the matches. However, these 3 values were not included in the analysis data set. The reason for this is that a decision support system will be established later with the analysis data. Figure 27 shows the txt link files created to be used in creating data sets. These link files are processed one by one in the application. As a result of this process, the data of all these leagues have been saved in csv formats, which are also seen in the same image. As a result of these processes, various data sets were made ready to be used later.

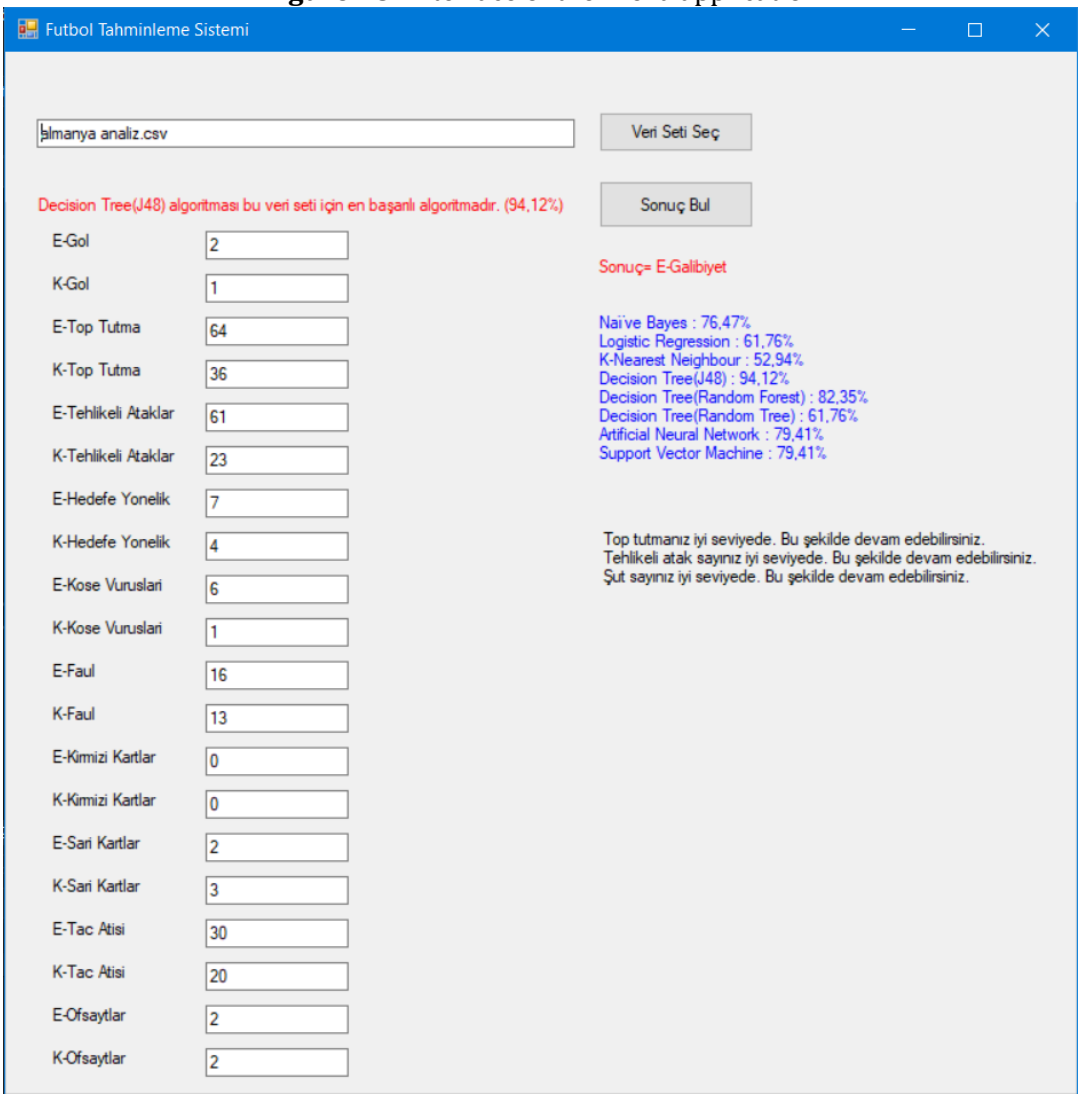
Figure 27. Data set, analysis and link files created



6.3. Analysis Operations

In this section, the stages of creating a decision support system with Weka using C# are explained. In these analysis processes, previously collected analysis data sets were used. A C# form application has been developed for analysis with Weka. First of all, the dll files of Weka have been added to the "\bin\Debug" folder in the project. In this way, easy access to ready-made Weka libraries within the application is provided. The sample form screen of the Weka analysis performed using C# is shown in Figure 28. There are two fixed buttons, four labels and one textbox on this screen. Their number is fixed. Buttons are used to select the data set file and to find the estimated result value. It shows the path to the textbox data set file. Labels show the most successful algorithm sentence, the estimated result, the estimation values found by all the algorithms used, and the suggestion sentences, respectively. The remaining labels and textboxes are dynamically formed. These are formed in a specific field in the form according to the number of entries in the data set and title names. Depending on the input value being numeric and categorical, textboxes can be replaced by comboboxes that can be selected.

Figure 28. Interface of the Weka application



When using Weka, processes are usually performed with arff extension files. It is necessary to use "CSVLoader" to process files in csv format. This requirement is shown in Figure 29. After pressing the "Veri Seti Seç" button on the form screen, the files with csv extension from the files on the computer are filtered. And the user is given the chance to choose a single file. These file selection steps have been explained in the previous sections of the study. If the file selection process is successful, this file with csv extension is sent to "csvLoader.setSource()" shown in Figure 29. And the data in this file is assigned to the "insts" variable.

Figure 29. Reading csv file in Weka

```
weka.core.converters.CSVLoader csvLoader = new weka.core.converters.CSVLoader();
csvLoader.setSource(new java.io.File(txtPath.Text));
insts = csvLoader.getDataSet();
```

Classification algorithms used in analysis using Weka are shown in Figure 30. These algorithms are; Naïve Bayes, Logistic Regression, K-Nearest Neighbors, Decision Tree (J48), Decision Tree (Random Forest), Decision Tree (Random Tree), Artificial Neural Network and Support Vector Machine algorithms. Each of these algorithms has its own specific functions and they give their own estimation values

as a result of a number of operations. Normalization, conversion from nominal to numeric and conversion from numerical to nominal can be given as examples of these operations. In the list shown in blue text in Figure 28, the results of these algorithms for sample data in the German league are listed in percentage.

Figure 30. Classification algorithms used in Weka

```
static weka.classifiers.Classifier NaiveBayescl = null;
static weka.classifiers.Classifier LogRegressioncl = null;
static weka.classifiers.Classifier Knncl = null;
static weka.classifiers.Classifier J48cl = null;
static weka.classifiers.Classifier RandomForestcl = null;
static weka.classifiers.Classifier RandomTreecl = null;
static weka.classifiers.Classifier Anncl = null;
static weka.classifiers.Classifier Svmcl = null;
```

Figure 31 is the function of the Naïve Bayes algorithm used in analysis operations. This function returns a value back in the double data type. It also takes one parameter into it. This parameter is the "insts" variable mentioned earlier, that is, the variable where the data read from the file is kept. "NaiveBayescl" defined in Figure 30 is equal to the function belonging to Weka with the expression "weka.classifiers.bayes.NaiveBayes()" in this function. Later, numerical data were converted into verbal data. While doing this, Weka's "weka.filters.unsupervised.attribute.Discretize()" method was used.

Figure 31. Naïve Bayes algorithm in Weka

```
//Naive Bayes
2 references
public static double NaiveBayesTest(weka.core.Instances insts)
{
    try
    {
        insts.setClassIndex(insts.numAttributes() - 1);

        NaiveBayescl = new weka.classifiers.bayes.NaiveBayes();

        //discretize
        weka.filters.Filter myDiscretize = new weka.filters.unsupervised.attribute.Discretize();
        myDiscretize.setInputFormat(insts);
        insts = weka.filters.Filter.useFilter(insts, myDiscretize);

        weka.filters.Filter myRandom = new weka.filters.unsupervised.instance.Randomize();
        myRandom.setInputFormat(insts);
        insts = weka.filters.Filter.useFilter(insts, myRandom);

        int trainSize = insts.numInstances() * percentSplit / 100;
        int testSize = insts.numInstances() - trainSize;
        weka.core.Instances train = new weka.core.Instances(insts, 0, trainSize);

        NaiveBayescl.buildClassifier(train);

        int numCorrect = 0;
        for (int i = trainSize; i < insts.numInstances(); i++)
        {
            weka.core.Instance currentInst = insts.instance(i);
            double predictedClass = NaiveBayescl.classifyInstance(currentInst);
            if (predictedClass == insts.instance(i).classValue())
                numCorrect++;
        }
        return (double)numCorrect / (double)testSize * 100.0;
    }
    catch (java.lang.Exception ex)
    {
        ex.printStackTrace();
        return 0;
    }
}
```

In the next steps, test and learning data were selected. Then, the result value of this algorithm was found with the calculations made. Here, as an example, only the code for the Naïve Bayes algorithm is shown. The codes of other algorithms are also more or less written in this format.

As a result of the calculations, the most successful algorithm should be selected. This process is shown as an example in Figure 32. In this example, the result values of Naïve Bayes and Logistic Regression algorithms are compared. This "if" structure is made for all algorithms, but the figure for only two algorithms is shown as an example. Here, the success result of the Naïve Bayes algorithm has been assigned to the "max_value" and "naivevalue" values as a starting point. The name values of the algorithms are also kept in the variable named "name". Then, each algorithm is compared with "if" blocks and the most successful value and algorithm name is determined. In order to keep the model of the algorithm, the algorithm variable is assigned to the variable named "model". Thus, the algorithm to run the manually entered estimation values was determined. According to the values found here, the most successful algorithm sentence in the interface and the list of these algorithms are shown in the required labels.

Figure 32. Finding the biggest value and algorithm name.

```
//naive bayes
double max_value = NaiveBayesTest(insts);
double naivevalue= NaiveBayesTest(insts);
model = NaiveBayescl;
name = "Naïve Bayes";

//logistic regression
double LogRegressionvalue = LogRegressionTest(insts);
if (LogRegressionvalue > max_value)
{
    max_value = LogRegressionvalue;
    model = LogRegressioncl;
    name = "Logistic Regression";
}
```

In the previous figure, the name of the variable in which the algorithm variables named "model" are kept is kept in the variable named "type" whose type is string type. This variable is the model for the most successful algorithm. This variable is seen in Figure 33. In this "if" block, if the "type" variable belongs to Naïve Bayes, that is, if this is the most successful algorithm, the entered values are processed according to this algorithm. These calculations are calculated by pressing the "Sonuç Bul" button after the values entered in dynamic textboxes and dynamic comboboxes, which are among the interface elements. After these operations, the value in the label with the result value is displayed in the interface.

Figure 33. Checking the most successful algorithm

```
if (type == "weka.classifiers.bayes.NaiveBayes")
{
    weka.filters.Filter myDiscretize = new weka.filters.unsupervised.attribute.Discretize();
    myDiscretize.setInputFormat(insts2);
    insts2 = weka.filters.Filter.useFilter(insts2, myDiscretize);
}
```

The creation of dynamic interface elements mentioned in the previous sections is shown in Figure 34. The array named "labels" in the codes shown in the figure holds the headings in the data set. In this way, these names can be listed on the dynamic labels in the interface. In order to keep numerical data, the array named

“numeric” was created. If the incoming data is digital, texboxes are formed in front of the appropriate title dynamically in the interface. If there are verbal data, the “nominal” sequence performs these operations and dynamically comboboxes are formed in the interface.

Figure 34. Dynamic interface elements

```
//seçme işlemleri
numAtt = insts.numAttributes() - 1;

int x = 30, y = 125, t = 30, l = 110;
int txt = 0, cmb = 0, r1 = 0, r2 = 0;
labels = new Label[insts.numAttributes()];
for (int i = 0; i < numAtt; i++)...

nominal = new ComboBox[cmb];
numeric = new TextBox[txt];
typeAtt = new bool[numAtt];
this.Height += (numAtt + 1) * t;

for (int i = 0; i < numAtt; i++)
{

    if (insts.attribute(i).isNominal())...
    else if (insts.attribute(i).isNumeric())...

    btnDiscover.Enabled = true;
}
```

The label with the suggestion sentence, which is one of the interface elements, is explained in Figure 38. These suggestion statements are created with a Python Script run in C# application.

The study on the minimum level of statistics for the football team to win is included. This study was carried out using the Python programming language in the PyCharm environment. This work has been integrated into the form screen. Information on this integration will be explained in detail later in this section. The report continues with the details of the work done with the Python programming language.

First of all, it would be logical to mention the algorithm logic of this study. All match data were used in this study. From these data, the attributes required for offense, that is to win a win, were taken. Then, the offensive features of the data whose result was a winner were averaged. The averages obtained here represent the minimum data limit required for a team to win. In the study with C#, the necessary data was taken from the data entered by the user and written to a file. This file is read in the Python environment and compared with the average values obtained before. As a result of the comparison made, a feedback is made to the user whether his data is at a sufficient level to win. Details on the implementation of this algorithm can be found later in the report.

Library usage, file reading and split operations are given in Figure 35. Pandas was used as a library. File reading operations have been performed thanks to the “read_csv()” function in this library. First, the file containing all match data was read. While reading this file and assigning it to the “matchData” variable, offensive-weighted attributes such as “E-Top Tutma, K-Top Tutma” are taken into account. Later, the “input.csv” file containing the data entered by the user was read, split and transferred into the “allCSVdata” value.

Figure 35. Reading file and splitting

```
import pandas as pd

matchData = pd.read_csv("C:\\Masaüstü\\tüm ligler analiz.csv", usecols=['E-Top Tutma','K-Top Tutma','E-Tehlikeli Ataklar',
                                                                        'K-Tehlikeli Ataklar','E-Hedefe Yonelik Sutlar',
                                                                        'K-Hedefe Yonelik Sutlar','Sonuc'])

fileCSV = open("C:\\Masaüstü\\input.csv", 'r', encoding="UTF-8")
CSVdata = fileCSV.read()
allCSVdata = CSVdata.split(",")
```

In the code fragment shown in Figure 36, the averaging process has been performed. First of all, the data set containing the data of all matches was converted into a dataframe with the "DataFrame()" function. Afterwards, data with target attributes "E-Galibiyet" and "K-Galibiyet" in the dataframe were transferred to "homeWin" and "awayWin" dataframes. Finally, the average of the attributes in "homeWin" and "awayWin" dataframes has been transferred into "homeStatisticsDict" and "awayStatisticsDict" dictionaries.

Figure 36. Creating data frame and averaging process

```
matchDataDF = pd.DataFrame(data=matchData) #dataframe was created

homeWin = matchDataDF[(matchDataDF.Sonuc == "E-Galibiyet")]  # data of the match won by the host has been collected
awayWin = matchDataDF[(matchDataDF.Sonuc == "K-Galibiyet")]  # data of the match won by the visiting team were collected in one place

homeStatisticsDict = {} # Dictionary will include the average of homeWin
awayStatisticsDict = {} # Dictionary will include the average of awayWin

homeStatisticsDict = homeWin.mean() # averages were obtained
awayStatisticsDict = awayWin.mean() # averages were obtained
```

In Figure 37, the general average of the averages of the features previously taken as home and away is taken. In this way, the minimum attack values required for a team to win were obtained. The general average of the data for possession of the ball is assigned to the variable "meanBallPossesion". In order to write the decimal part of float type data in a certain format, the "round()" function has been used. In this way, the two numbers after the point were taken into account. The averages of the dangerous attack and shot data of the teams are kept in "meanAttack" and "meanTargetShot" variables, respectively. Later, in order to inform the user, the data received from the user and the average values were compared. Finally, the analysis results obtained were printed on the console screen with "print()" function.

Figure 37. Obtaining the general average and printing the result

```
meanBallPossesion = (homeStatisticsDict["E-Top Tutma"] + awayStatisticsDict["K-Top Tutma"]) / 2
meanBallPossesion = round(meanBallPossesion,2)
meanAttack =int((homeStatisticsDict["E-Tehlikeli Ataklar"] + awayStatisticsDict["K-Tehlikeli Ataklar"]) / 2)
meanTargetShot = int((homeStatisticsDict["E-Hedefe Yonelik Sutlar"] + awayStatisticsDict["K-Hedefe Yonelik Sutlar"]) / 2)

if float(allCSVdata[0]) < meanBallPossesion:
    resultBallPossession = "Top tutmanızı en az {}'a yükseltmelisiniz.".format(meanBallPossesion)
else:
    resultBallPossession = "Top tutmanız iyi seviyede. Bu şekilde devam edebilirsiniz."
if int(allCSVdata[1]) < meanAttack:
    resultMeanAttack = "Tehlikeli atak sayınızı en az {}'a yükseltmelisiniz.".format(meanAttack)
else:
    resultMeanAttack = "Tehlikeli atak sayınız iyi seviyede. Bu şekilde devam edebilirsiniz."
if int(allCSVdata[2]) < meanTargetShot:
    resultMeanTargetShot = "Şutlarınızı en az {}'a yükseltmelisiniz.".format(meanTargetShot)
else:
    resultMeanTargetShot = "Şut sayınız iyi seviyede. Bu şekilde devam edebilirsiniz."

print("\n",resultBallPossession , "\n" ,resultMeanAttack ,"\n" ,resultMeanTargetShot)
```

At the beginning of this chapter, it was mentioned that the work done with Python is integrated into the form screen. The code snippet written for this integration is given in Figure 38. The basic logic here is to run the Python script written in C# programming language. Here, the attribute values required from the information received from the user have been written to the "input.csv" file with the "SaveArrayAsCSV()" function. Here, a new process has been created to run the Python script. Then the paths of the "python.exe" file and the "main.py" file to be run are defined to the process. The output value obtained after the process is run with the "Start()" function has been assigned to the "output" variable. The output value obtained here is the analysis evaluation result written on the console screen with the "print()" function in the Python script. Finally, the value in the "output" variable is printed on the form screen so that the user can see the result.

Figure 38. Running Python script on C# and showing results to the user.

```
public void runPythonScript(string[] inputArray)
{
    SaveArrayAsCSV(inputArray, @"C:\Masaüstü\input.csv");
    Process p = new Process(); // Yeni nesne yarat…
    p.StartInfo.UseShellExecute = false; // Shell kullanma…
    p.StartInfo.RedirectStandardOutput = true; // Çıkışı yönlendir….
    // Python klasörümüz ve derleyicimizin adı…
    p.StartInfo.FileName = @"C:\Python\Python38\python.exe";
    // verilecek yani çalıştırılacak python scriptimizin yolu…
    p.StartInfo.Arguments = @"C:\PycharmProjects\decisionAnalysis\main.py";
    p.Start(); // işlemeye başla…
    string output = p.StandardOutput.ReadToEnd(); // çıkışı sakla
    p.WaitForExit(); // çıkış için zaman tanıyalım
    label1.Text = output;
}
```

Finally, if the study is to be summarized, the necessary data and analysis result for the win were obtained with the Python programming language. After that, these results are presented to the user by running the Python script in the C# programming language.

7. Conclusion

In this study, football match result prediction decision support system is proposed. In this study, the collection and processing of data is of great importance. The data obtained from a certain website is made ready for the analysis phase after passing through some preprocessing steps. The data prepared for analysis were run on various algorithms as mentioned in the previous sections. With the interpretation of the results obtained here, a decision support system has emerged that can predict the result of the match. With this study, an application has been developed that includes a user interface that users can use using different software languages. The aim here was to transform the study not only in theory but also into practice.

As Future work, it is aimed to develop a player-based decision support system in the later stages of the project. Here, it will be determined which player should start in the starting 11 according to the performance of the players. In this way, it is aimed to propose a decision support system based on players.

References

Alpaslan, N. (2019). Meme kanseri tanisi için derin öznitelik tabanli karar destek sistemi. Selçuk Üniversitesi Mühendislik, Bilim Ve Teknoloji Dergisi, 7(1), 213-227.

Aydemir, E. (2019). Akademik Personel Performans Değerlendirmesinde Bir Karar Destek Sistemi Önerisi. Bilişim Teknolojileri Dergisi, 12 (2), 131-140. DOI: 10.17671/gazibtd.484509

Baboota, R., & Kaur, H. (2019). Predictive analysis and modelling football results using machine learning approach for English Premier League. International Journal of Forecasting, 35(2), 741-755.

Canbazoğlu, E , Ercan, U , İpekçi Çetin, E . (2018). Ticari Taksi Araç Yenilemelerinde Ahs Ve Moora Yöntemlerine Dayali Karar Destek Mobil Uygulamasi. Uluslararası İktisadi ve İdari İncelemeler Dergisi, 18. EYİ Özel Sayısı, 117-134. DOI: 10.18092/ulikidince.354009

İnce, C., İnce, K., & Hanbay, D. (2021). Saldırı Tespit Sistemlerinde Sınıflandırma Yöntemlerinin Kıyaslanması. Bilgisayar Bilimleri, 6(1), 1-10.

Chouseinoglou, O., & Şahin, İ. (2019). Metin Madenciliği, Makine ve Derin Öğrenme Algoritmaları ile Web Sayfalarının Sınıflandırılması. Yönetim Bilişim Sistemleri Dergisi, 5(2), 29-43.

Evwiekpaefe, A. E., Bitrus, E., & Ajakaiye, F. (2020). Selecting Forward Players in a Football Team using Artificial Neural Networks. International Journal of Computer Applications, 975, 8887.

Güldal, H., & Çakici, Y. (2017). Ders yönetim sistemi yazılımı kullanıcı etkileşimlerinin sınıflandırma algoritmaları ile analizi. Atatürk Üniversitesi Sosyal Bilimler Enstitüsü Dergisi, 21(4), 1355-1367.

Jauhari, A., Suzanti, I. O., Putro, S. S., Anamisa, D. R., Abdillah, M., Khozaimi, A., & Rochman, E. M. S. (2018). Determining Football Players Position Using SAW Method. In The 1st International Conference on Computer Science and Engineering Technology Universitas Muria Kudus (pp. 54-60).

Korkmaz, A , Büyükgöze, S . (2019). Sahte Web Sitelerinin Sınıflandırma Algoritmaları İle Tespit Edilmesi. Avrupa Bilim ve Teknoloji Dergisi, (16), 826-833. DOI: 10.31590/ejosat.598036

Lumbao, J. I. P. I., Sindac, K. K. V., Lagunzad, H. C., Paed, F. C. A. B., Gonzaga, J. M. M., Sillano, L. A. S., & Sunga, A. L. (2017). Implementation of Multiple Regression in Predicting Basketball Game.

Purkuloğlu, E., Arzu, Ü. N., & Yürürdurmaz, F. (2019). Hemşire Karar Destek Sistemleri Uygulamalari. Hacettepe Sağlık İdaresi Dergisi, 22(3), 491-514.

Rudrapal, D., Boro, S., Srivastava, J., & Singh, S. (2020). A Deep Learning Approach to Predict Football Match Result. In Computational Intelligence in Data Mining (pp. 93-99). Springer, Singapore.

Salman, B., Alhakkak, N. M., & Jaber, M. M. (2018). Football Player Decision Support System Baghdad-City as a Case Study. International Journal of Engineering & Technology, 7(3.20), 406-411.

Saritas, M. M., & Yasar, A. (2019). Performance analysis of ANN and Naive Bayes classification algorithm for data classification. International Journal of Intelligent Systems and Applications in Engineering, 7(2), 88-91.

Tanyilidizi, E., Karabatak, M., Yildirim, G., & Özpolat, Z. (2018). Siğil Tedavisinde Sınıflandırma Algoritmalarının Performans Analizi. Fırat Üniversitesi Mühendislik Bilimleri Dergisi, 30(2), 249-256.

Wenninger, S., Link, D., & Lames, M. (2020). Performance of machine learning models in application to beach volleyball data. International Journal of Computer Science in Sport, 19(1), 24-36.

Yıldız, A , Zan, H . (2019). Segmantasyon yapmadan patolojik kalp sesi kayıtlarının tespiti için bir örüntü sınıflandırma algoritması . Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi , 10 (1) , 77-91 . DOI: 10.24012/dumf.476437

Yorulmaz, M , İç, Y , Seyrek, A . (2019). Bilgisayar Değiştirme Kararları için Bir Karar Destek Sisteminin Geliştirilmesi . Bilişim Teknolojileri Dergisi , 12 (3) , 195-202 . DOI: 10.17671/gazibtd.510657

Young, W. A., & Weckman, G. R. (2020). A Team-Compatibility Decision Support System for the National Football League.

Zaveri, N., Shah, U., Tiwari, S., Shinde, P., & Teli, L. K. (2018). Prediction of football match score and decision making process. International Journal on Recent and Innovation Trends in Computing and Communication, 6(2), 162-165.