SPORTS BETTING RISK MANAGEMENT USING MACHINE LEARNING

HEL 2602 Honors Thesis II Project Report

Supervising Committee

Supervisor: Dr. Ioannis Christou
Second Reader: Prof. E. Vagianou
Additional Members: Dr. Georgia Giannakopoulou, Dr. Sophia Zevgoli

Dimitrios Mavrofrydis
24/2/2021

Abstract

The purpose of this report is to analyze the development, implementation and outcomes of a machine learning software package solution, which can identify potential mistakes and value errors in the odd generation systems of two major sports betting providers Bet 365 and Bwin. Nevertheless, the architectural structure of the software solution promotes the scalability and adaptability of the machine learning algorithm to operate on a wide variety of sporting events and betting providers. The initial goal of this solution was the evaluation and generation of an accurate win/lose risk factor for upcoming soccer matches. This would allow the players to better understand the actual risk associated with the bet they are about to place. However, after careful examination of the historical soccer odds and data structure, the main functionality of the software solution was set to operate as a risk management alert tool opting to identify miscalculated or exposed odd betting values for any upcoming soccer matches. This report will extensively focus on the collection of the historical data using a custom data mining API while also analyzing the functionality and results of the supervised machine learning algorithm.

Contents

Abstract ..... *iii*

Introduction *1*

Online Sports Betting Odds Generation (Problem Definition) *2*

Solution *3*

Data Mining & Processing *4*

Data Analysis & Supervised Machine Learning *8*

Results *13*

Evaluation & Limitations *15*

Future Work & Improvements *15*

Conclusion *17*

Appendix *18*

BSON Documents Structure **18**

Historical Soccer Matches (Training Data) 18

Upcoming Soccer Matches & 2019 Selected Data (Testing Data) 24

Machine Learning Results & Findings **30**

Tools & Technologies Used **35**

Editors Used: 35

Programming Languages: 35

Libraries: 35

Third Party Applications & Drivers: 35

Source Code **36**

Bibliography *49*

Introduction

Sports and especially competitive sporting events have been gaining more and more popularity due to the recent growth and development of the online sports betting industry. With the exponential growth of the internet and the online streaming services that are accessible through any smart device with an active internet connection, players from all around the world can access a plethora of online sport betting services with a single press of a button. This rapid evolution of the online web services and web apps has allowed the online sports betting industry to expand and take over the professional sports industry (James et al. 2017). As a result, most sports fans are starting to unconsciously mistake sports as a form of betting and betting as a form of sporting activity (Lopez-Gonzalez H Griffiths M.D 2018). Nevertheless, many countries and especially specific states in the United States still do not allow to this date any provider to purchase a license for providing their online sports betting services in their area (Cantinotti et al. 2004).

These online sport betting platforms allow the bettors to place bets on any sport of their choice, by assigning a value which represents the odds of a certain event happening in that specific match. These odd values that are publicly accessible through any betting provider's online platform, are either generated directly by the sports betting providers themselves or imported by a third-party odd-generation service. It is important to highlight that most of the online sports betting companies in the market are partnering with such service providers to import the odds into their online platforms. One of the most major online sports betting odd-generation companies is Sports Radar. This service provider located in Switzerland is the main odds provider for companies such as Bet 365, Betfair, William Hill as well as the Greek betting provider OPAP (Sports Radar, 2021). By simply using a betting provider's platform, the bettors are prompted to place an x amount of their money in the probability of an event happening in a certain sporting match. Consequently, the bettor could receive an amount of money equal to the initial amount multiplied by the odd value for the specific event. In the case where

the event is not validated, the players end up losing the wager to the bookmaker (Ondřej Hubáček et al. 2019, Bar-Eli et al. 2011).

Online Sports Betting Odds Generation (Problem Definition)

When it comes to the accuracy as well as the validity of the generated betting values, companies like Sports Radar base their calculations on mathematical formulas and algorithms to perform statistical analysis of the collected sports data (Sport Radar, 2021). As a result, a wide variety of mathematical equations and statistical analysis formulas are implemented during the development process of these algorithms that gather and analyze meaningful statistics regarding past soccer matches. This results in a fast and automated process of calculating and exporting specific odd betting values for every available soccer betting category in the world. However, algorithms that are solely based on mathematical formulas and statistics to calculate and generate betting odds, allow for a slight change of odd miscalculations. This means that when a sports betting platform receives the betting odds from a service provider, there might be a significant percentage of odds which have been miscalculated or wrongly exposed due to unfortunate or mathematically unpredictable events.

Thus, it is important to acknowledge that bettors who understand the accuracy of these fixed prices as well as the true probability of an event happening based on the odds provided in these online platforms, can estimate whether the specific generated bet value has been wrongly placed at a higher or lower price than it should have initially been placed at. Such techniques of studying the odds of specific sporting events before placing a bet, does not only allow the players to get a better understanding of the risk they are about to take, but it also allows them to predict the success rate of their placed bet (Gaspero 2015). Therefore, it is important for the online sports betting providers to identify such odd generation mistakes before they are made publicly accessible on their online platforms. This would allow them to minimize any room for error when it comes to paying back their customers. Experienced bettors will mostly focus on studying previous game

statistics before placing their bets, instead of relying on rapid decision making or on personal beliefs (Cantinotti et al. 2004). By following risk management analysis and data processing, the bettors can achieve higher return rates, while also lowering the risk of losing their placed bet (Gainsbury Russel 2015). Due to the recent development and growth of the online live sports betting services, the betting companies are constantly attempting to come up with new ways of limiting their player's revenue by altering the values of specific odds to act in their favor (Winston & Wayne 2012, Bar-Eli et al. 2011).

There are many ways in which the online sports betting companies ensure that they have absolute control during a live sporting event. According to Glanz, the most used technique that bookmakers use in order to limit their loss during a live event, is to employ agents at major events to record in-game incidents that may affect the price of the odds for the respective game (2018). Additionally, bookmakers mostly rely on their ability to outperform the average bettor in forecasting the results of the games rather than on focusing on increasing their profit with simple calculations and margin values (Levitt 2004). Nevertheless, while it is easier to manipulate odd values and data during a live match, it is essential to ensure that the pre-game odds are carefully examined and validated as to minimize any potential risk. Most players will take advantage of the pre-game odd values as they are able to place bets before the game starts on a fixed price, which ensures that even when the game starts their payout will not be altered. As stated previously in this paper, this is a crucial point for the online sports betting companies to ensure the validity of their pre-game odds. In detail, most of the pre-game odds are instantly imported from the odd service providers and are only altered for the addition of the house edge which is the fee charged by the betting provider.

Solution

As a result, this thesis project attempts to resolve and prevent errors or issues by developing an all-in-one software solution. By implementing data mining

technologies such as web scrapping for the collection of the data from various web sources, as well as machine learning algorithms for the processing and analysis of the data, this software successfully identifies potential mistakes or risky bet values based on previously identified patterns and calculated statistics. This software solution is not limited to a specific online sports betting company neither to a specific sport or league. Nevertheless, due to the time constraints as well as the limited resources that were available for this project, it was essential to narrow down the algorithm to solely run for any soccer related matches from a variety of leagues, countries and time periods.

The development of this software solution can be broken down into two major parts. The first and the most important part includes both the collection and processing of the collected data. This part is followed by the extensive and precise searching and supervised machine learning algorithm which is responsible for the analysis of the collected data. According to the Salian, a supervised machine learning algorithm can be best described as the following “in a supervised learning model, the algorithm learns on a labeled dataset, providing an answer key that the algorithm can use to evaluate its accuracy on training data.” (2021). The finalized version of this software solution automates the process of collecting and updating the training data of the machine learning algorithm, using a variety of data mining Java algorithms which execute when new training data is available online. Finally, for the completion of this project it was essential to implement a supervised machine learning algorithm which can provide great and accurate results especially when dealing with historical data and sports analytics (Apostolou Konstantinos 2019). All the parts and algorithms that were used for the collection, analysis and generation of the outcomes, are going to be analyzed in detail throughout the rest of this paper.

Data Mining & Processing

First and foremost, the most important step that must be taken in order to ensure that a supervised machine learning algorithm will generate significant

results, is to provide it with sufficient and meaningful training data which are stored using a specific and easy to process data-structure. The online sports betting industry is built on top of an enormous amount of information and data which derives from every match that has been completed for every available sport in the world throughout the past few decades. This means that in order to generate, process and provide meaningful odd betting values for a specific bet, a great amount of work and statistical/historical analysis is required. As mentioned earlier in this paper, this is an action which is completed by either a dedicated trading/bookmaking team which is a part of the online sports betting provider's environment, or by an external company which specializes in performing such odd generation services that are later provided as a service or sold.

The ML algorithm that is used in this project, requires a significant amount of soccer related statistical data to correctly operate and identify odd generation errors or error prone calculated odds. As a result, the most important step for the development of this project was the identification and collection of every available historical soccer match data and statistics. In order to allow the ML algorithm to study and potentially identify any patterns in the available odds for each soccer match, all the collected data had to be accompanied by every available bet and odds provided by the most successful betting providers for each specific match. However, such data and statistics are not easily accessible to the public since most of this information either belongs to specific betting companies or can only be accessed with the use of very expensive third-party APIs.

Due to the research nature and the budget limitations of this project, it was essential for a custom data mining/web crawling API to be developed, in order to acquire all the required data. This API is very significant for the completion of the project because it automates the process of accessing and searching the web for any soccer data statistics, while ensuring that all this information is collected in the appropriate form and stored in a very specific and easy to manipulate structure. Despite the limited timeframe that was available for the completion of this Thesis

Project and the complexity of the algorithms evolved, an advanced data mining API was successfully developed which was based on an adaptable custom web crawling and scrapping algorithm capable of identifying and automating the process of retrieving, collecting and finally inserting the retrieved data into an internal database.

The data mining algorithm in discussion was written in Java and uses two very important libraries which enabled the automation and extraction of the soccer data from various websites. These two libraries are the Selenium and Jsoup libraries. The Selenium library allows access to specific web browser drivers that must be manually installed into the working environment. As a result, these web-drivers are accessible by the written Java code, allowing for the complete control over executing and automatically navigating throughout any web page on the internet. The Jsoup library is used as a tool which enriches the Java syntax by embodying Html and CSS commands which can be used to breakdown and retrieve data based on the publicly accessible web elements of each sports data provider's website (JSoup Org 2021). However, having access to such tools in addition to the written Java code, would be meaningless without having the knowledge and information of which websites can provide the data required to support the ML algorithm.

For a great amount of time, there had been many attempts of identifying potential websites which legally provide free historical data on the betting values and odds for each soccer match to the public. Nevertheless, after studying all the major online sports betting platforms and betting provider websites, the Odds Portal website was selected which covered most of the needs for acquiring the required soccer historical data and statistics. This website provides all the necessary betting values, odds and categories of the available soccer bets. Additionally, the Odds Portal website includes soccer leagues and historical data from every part of the world that took place during the last two decades. Finally, it is a very well structured, well written and easy to reverse-engineer website in terms

of its HTML, CSS and JavaScript code. All these traits have been proven to be the strongest factors that led to using the Odds Portal website as the main source of data and information for this project. The functionalities and characteristics that the Odds Portal website offers, allowed for an easy understanding of how the website works, promoting the web automation process while reducing any room for error.

Furthermore, the collected data required a great amount of processing and preparation in order to allow the machine learning algorithm to understand their weight of importance and generate the appropriate results. In order to achieve this, the collected data had to be cleaned and stripped of any illegal or unnecessary characters that were extracted during the data mining process. Additionally, the respective odd values and historical score results were converted into the appropriate programming data types in order to ease the saving process. The final step includes the data formation and processing which allows the processed data to be inserted into the computer's internal database. After taking into consideration the volume of soccer data collected, ranging between 450.000 and 500.000 historical soccer matches from all around the world, a fast and reliable database structure had to be used. Thus, the Mongo DB database was selected in order to ensure the speed, efficiency and safety of the data. The selected database is used by some of the most major companies in the world including Google, EA sports, Facebook and many more (MongoDB Inc. 2021). This database brings many advantages for such big datasets as it ensures that each historical soccer match stored as a BSON document, will maximize efficiency both in storage space and speed.

These BSON Documents follow a structural hierarchy like those of the JSON files. For the purpose of reducing this project's algorithm complexity as well as to allow the machine learning and analysis part of this project to be developed and implemented within the given time period, the betting types and historical odds that were collected were limited to the 1X2 category for every betting provider. The 1X2 bet type has been selected for the data collection part of this project since it

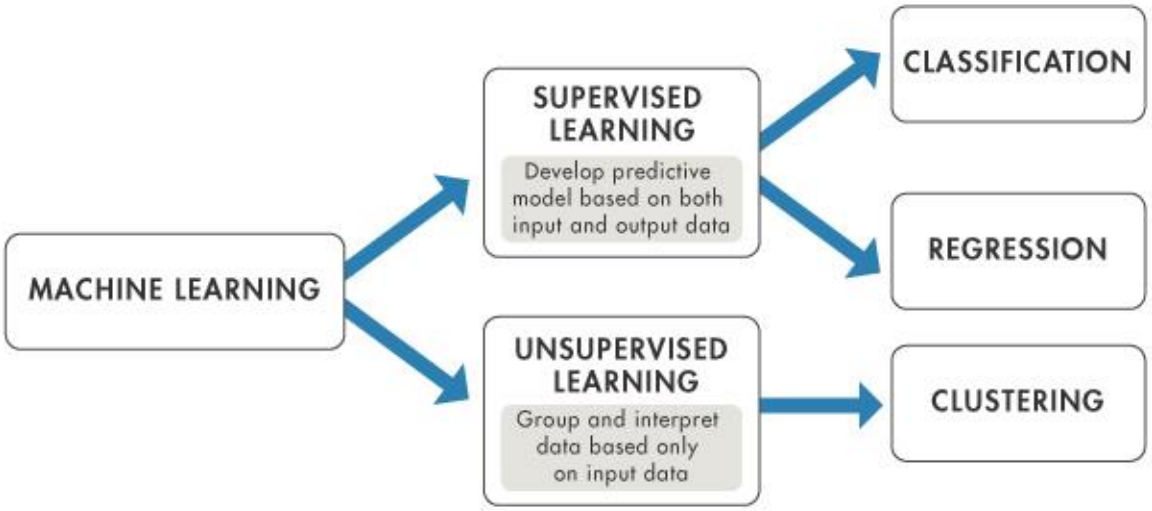
is the easiest and most straightforward betting type category which can be applied to almost every sport, especially in soccer matches. The 1 in the 1X2 betting category of every sporting event, represents the chance of the home team winning the game. Similarly, the X value represents the chance of the game ending in a draw. Finally, the 2 of the 1X2 category represents the chance of the away team winning the game.

This data structure allows this paper to provide the audience with a clearer and easier to understand view of how such specific statistical and historical betting data enable the algorithm to calculate and identify patterns which indicate the miss-calculated or exposed pre-game odds. Every generated BSON Document inherits the following structure: a unique ID for each match (the match's web link), the type of the collected bet (e.g., 1X2), the time period collected for this bet type (e.g., Full Time), the collected odd values for each time period accompanied by the Bookmaker's name, collected sport, league, day, year, month, time and the scores for both the final result and for the end of every period. Examples of validated, collected and successfully processed BSON Documents can be found in the appendix section of this paper.

Data Analysis & Supervised Machine Learning

The second and most important step for the completion of this software solution was the development of the machine learning algorithm based on a supervised learning adaptation. As stated earlier in this paper, the machine learning algorithm is implemented into the software solution in order to enable the synthesis, validation and identification of potential mistakes and exposed betting errors in the odd generation systems of Bet 365 and Bwin providers. The main reason behind the development of a machine learning algorithm was the need to enable the online betting providers to correct any potentially exposed values and errors in their pre-game soccer odds. The machine learning algorithm which works as an all-in-one software solution with the custom-made data mining API, requires a great amount of data in order to generate the appropriate results. There are two

main types of machine learning algorithms that could be used in this case. The supervised and unsupervised machine learning algorithms.



A supervised learning classification approach is most used when a prediction is ought to be generated for a specific set of data test-dataset, by building a classification model on a given set of data training dataset (Witten et al. 2011). For the development of this supervised ML algorithm, a neural network was created by following a classification approach, due to the nature of the soccer match's results which include a win, lose or a draw (Prasitio et al. 2016).

After careful examination and preprocessing of the given training dataset, the machine learning algorithm received an enormous amount of input data that was broken down into different groups consisted of every team's country and tournament. In order to predict and identify any potentially exposed or faulty betting odd values, each soccer match's historical data from the training-dataset had to be analyzed. This analysis generates a different classification model for each one of the testing matches based on the related country and tournament. As a result, the algorithm can train the data for each country and tournament sequentially, in order to reduce any room for error and provide more accurate results and predictions for each tournament separately.

The chosen testing-dataset includes historical data for each soccer match for every available country and tournament of the year 2019. The chosen training

dataset for the Bet 365 provider includes 26.188 soccer matches and 32.190 soccer matches for Bwin. Additionally, the training dataset for Bet 365 consists of 330.910 soccer matches and 318.499 for Bwin, corresponding to all the testing dataset's countries and tournaments ranging from collected matches between the years of 1999 – 2018. A similar machine learning algorithm was also developed in order to perform similar calculations, predictions and identifications for all the upcoming matches for both providers. The upcoming matches are received each day at a certain timeframe by using the custom-made data mining API in order to be used by the supervised machine learning algorithm as a testing-dataset. This allows the algorithm to perform the appropriate calculations and classifications using the already collected historical data to predict and identify whether any of the retrieved match's odd values are either overvalued or undervalued. However, due to certain timeframe and sport related restrictions that the worldwide pandemic has caused, the 2019 historical dataset was eventually chosen as the testing dataset which also allowed for the evaluation and validation of the algorithm's predictions. This is made possible since the 2019 testing dataset soccer matches' results are already available in the database.

The classification process of this algorithm is based on the generation of four different classes (observations) for each soccer match. These classes are generated for each match in the testing-dataset based on the provider's historical success in assigning the appropriate 1X2 values in similar matches. The main hypothesis in which the machine algorithm is based on is the following: classify the testing data (upcoming matches), into four different classes based on whether the given home or away odd values for each match are undervalued or overvalued. As far as the mathematics are concerned, the classification formula can be broken down into four different steps.

The first step of the classification process involves the retrieval of the home and away odds of all the required betting providers for the given input soccer match. Following the retrieval of the input, a new range of betting values is

generated by incrementing both the home and away odd values by a constant decimal equal to 0,01. This ensures that the next step of the classification process will be given as an input, a range of betting odds between the initial odd value plus 0,01.

The second step includes the retrieval of all the available historical soccer matches from the internal database which satisfy certain requirements. Therefore, the algorithm initiates a searching sequence by filtering all the historical matches that belong in the same country and tournament as the input, whose home or away odd values fall into the range of the input's odds plus 0.01. In that event, a new collection of historical soccer matches is created all of which are associated with the input match as training data. In order to provide meaningful weights to the new collection of matches, an algorithm is used which categorizes and counts the number of matches which verify the value of the given home or away odds based on their result. In other words, every historical match belonging in this collection is cross validated in order to provide an insight on the number of matches resulting in the home team or the opponent winning. This validation outputs the number of matches whose home odds belong in the given range and resulted in a home win. An identical validation takes place for the away team odd values.

The third step is initiated after acquiring the number of historical training matches which validate their result based on their corresponding odd value. The mean of the odds input range is generated in order to create a point of reference, which is then converted to a percentage indicating the true probability of an event happening. Due to the mathematical nature of this algorithm, a statistical percentage is added and subtracted from this percentage in order to allow room for error and miscalculations. This room for error has been set to 2.5% of the probability's true value. The final step of the classification process is the calculation of the percentage of success which, represents the number of matches validating the team winning hypothesis divided by the total number of matches found in the generated collection. Hence, the classification process for the input match is

initiated which runs a check on whether the percentage of success is less or greater than the generated probability, including the room for error of the input match's odd values. In the case where the percentage of success is less than the probability percentage, the input match is classified as overvalued for its corresponding home or away odds. In the case where the value is greater, the input match is classified as undervalued. Ultimately, if the percentage of success is included in the input range, the match is classified as having correct odds for both home and away teams.

This classification for every testing match is based on the results of the training data for the same country as well as tournament. As a result, all the testing or upcoming matches are now separately classified for each one of the given provider's odds as overvalued or undervalued for the corresponding home or away odds. Eventually, the provider that was used as an input for the testing dataset, is now assigned a class which characterizes its provided odd values as accurate or inaccurate based on the historical analysis performed on the training dataset for the specific match. It is very important to highlight that when a given soccer match's home value is classified as overvalued, it is up to the betting provider to revisit the specific match that the machine learning algorithm found to have an exposed home value and reconsider altering the odds.

Finally, the speed and efficiency of the machine learning agent had to be taken into consideration due to the importance of quickly analyzing the enormous amounts of collected data. The classification process for each one of the testing matches requires a great amount of processing and data analysis from the machine learning agent. An outstanding number of thousands and thousands of historical soccer matches must be analyzed each second for a testing match to be successfully classified. However, taking into consideration the amount of soccer matches that the algorithm is dealing with, the process would eventually take hours or even days to complete for every country and tournament for every iteration of the algorithm. According to Schuessler and Loyola "training a neural network for

complicated and multi-dimensional problems normally means using very large amounts of training examples with hundred thousand or even millions of patterns" (2011). Thus, in order to increase the efficiency of the generated neural network, the training process was parallelized.

By parallelizing the training data to run simultaneously using multithreading java techniques due to the multi-core and multi-thread working computers, the machine learning agent is up to fifteen times faster in training and finally testing the data. The training dataset is broken into several layers during the training process, forming several sub-training datasets equal to the number of threads the running machine can process plus one. As a result, each thread and process are trained simultaneously which allows the neural network to perform more efficiently depending on the CPU power of each device (Schuessler et al. 2011). Finally, all the appropriate measures have been taken in order to ensure that the machine learning agent can be easily upscaled, improved and expanded in the near future with new historical data and more advanced algorithms.

Results

After the machine learning algorithm has finished training and testing for a given dataset, the findings are ready to be presented. This step includes the analysis of the observations given for any classified soccer match provided by the ML algorithm. The generated observations are based on two main assumptions in order to examine the statistically significant observations.

The first assumption focuses on all the soccer matches that have been classified as having overvalued home or away odds that exceed a 60% threshold of winning rates, respectively. In the case where both home and away odds are classified as overvalued, the one with the highest winning rate is marked as significant. From a betting company's point of view, this would be a sign of risk exposure. Such a scenario indicates that the company would have to double check and adjust the odds accordingly. For example, after training with the appropriate data for Sweden's Allsvenskan league, the machine learning algorithm

successfully validated an outstanding 75% of the classified test matches for having exposed overvalued home team odds. This specific case involves betting odds for the Bet 365 provider. In other words, the specific provider assigned overexposed and extremely high betting values on the probability of the home teams winning in the 2019 Allsvenskan league. By implementing this machine learning algorithm into their systems, the betting providers could have avoided exposing their odds in such sporting events which would result the minimization of their loss.

The second assumption involves the soccer matches that have been classified as having undervalued odds for either home or away, with 25% or less winning rate. Given the scenario that both home and away odds are marked as undervalued, the one with the lowest winning rate is marked as significant. Thus, a betting company could take advantage of this information as to manipulate the odds in their favor. This would mean that for these specific matches, the company has the elasticity to slightly increase the odds, resulting in attracting more customers. Another observation that derived from the thousands of the machine learning classification results, is a case of the 2019 Mexico Liga Mx league regarding the Bet 365 provider. After studying and classifying the appropriate data, the ML agent detected that only an impressive 9.38% of the undervalued testing matches for the away team were validated. This means that the betting providers could effortlessly apply the ML agent into identifying such matches and eventually tricking their customers by raising the value of the probability of the away team winning. Thus, by minimizing their risk based on the statistics, the companies could attract more bettors by tricking them into placing a bet for the away team.

However, it is important to acknowledge that two more scenarios may exist based on these assumptions. The first case would be a classified match with overvalued home and undervalued away odds. Vice versa, the opposite case would be a classified match with undervalued home and overvalued away odds. In both cases, the betting company is advised to check the probability of the match ending in a draw and proceed with corrective adjustments. Additionally, this case

provides the company with the ability to easily increase the undervalued odds and decrease the overvalued odds. For instance, after running the ML algorithm for the 2019 Primiera Division in Venezuela, a significant 57% of the undervalued home and overvalued away matches were validated for resulting in a draw. In this case the respective betting provider Bwin would be able to manipulate the odd betting values for all three results as only 17% and 26% of the soccer matches in this testing dataset end up in the home or away team winning respectively.

Evaluation & Limitations

Developing such an advanced and sophisticated software solution has proved to be a very challenging and time-consuming task. Due to the recent events caused by the covid-19 pandemic, the sports industry has been greatly affected in terms of its upcoming events and match consistency. A great number of soccer matches are being canceled everyday due to covid-19 related issues. Consequently, the initial consideration to deploy the supervised machine learning algorithm into performing soccer forecasting and odd prediction accuracy validation for any upcoming soccer matches was not feasible. Thus, as explained earlier in this paper a historical pool of testing-data was used which allowed for an easier and more efficient validation of the provided machine learning results.

Future Work & Improvements

By following some of the most major software development and machine learning principles, this software solution has been developed with extreme attention to detail and detailed class and code relations. Additionally, the introduction of the custom multilayering neural networks used by the machine learning algorithm allow room for future improvements and were designed with one single goal in mind, the scalability of the solution. Furthermore, the custom-made data-mining API allows the user to gather data from any website in the world in order to provide the machine learning agent with the appropriate training and testing data to perform its calculations. As a result, the scalability, flexibility and

adaptability of the implemented algorithms enable the easy integration or expansion of the software solution in many different fields of study.

Consequently, the existing custom-made classification machine learning algorithm could be improved by implementing and testing different classification approaches into the existing model. First, a Naïve Bayes classifier approach will be examined in order to determine whether the performance as well as the overall results of the existing custom-made algorithm will be improved. The Naïve Bayes classification algorithm is very efficient especially when dealing with large sets of training data. The Naïve Bayes classification algorithm could be used especially for predicting and identifying faulty odds in live sports, due to its ability to perform calculations in real time. Similar to the already developed machine learning classification algorithm, the Bayes approach could enable the formulation of individual matrices, for classifying each soccer match as overvalued or undervalued for any kind of bet type.

Additionally, a non-probabilistic binary linear classifier approach will be followed by implementing support vector machines “SVM”. The implementation of an SVM classifier would enable the machine learning agent to improve its accuracy in predicting and identifying potential odd value errors. Algorithms in the SVM category have the ability to handle and classify high dimensional data very effectively, especially in the case of sports betting which offers thousands of bet types for each sporting event. Furthermore, having clearly defined and separated the training datasets for each sport, betting type and league, can dramatically improve the predictions of the SVM algorithm. However, due to the already large sets of data and the growing popularity of the online sports betting industry, the SVM could actually underperform in the case where the size of the training dataset increases dramatically.

The final classification approach will involve an implementation of a decision tree machine learning classification algorithm. A decision tree classification approach would enable the algorithm to be more flexible in terms of predicting the

expected outcomes with certain missing values (e.g., missing home or away scores for the first half of the game). What is most important is that the input data does not need to be extensively normalized. This can maximize the efficiency and time required in the data processing stage, while also allowing for an easy visualization of the classified results. Nevertheless, such an approach may require more time than the average classifying algorithm. However, it is the most appropriate solution for predicting errors and faults before they actually occur. Thus, a decision tree algorithm would also be suitable for predicting and identifying miscalculated odds during live sporting events for a variety of betting providers and betting types.

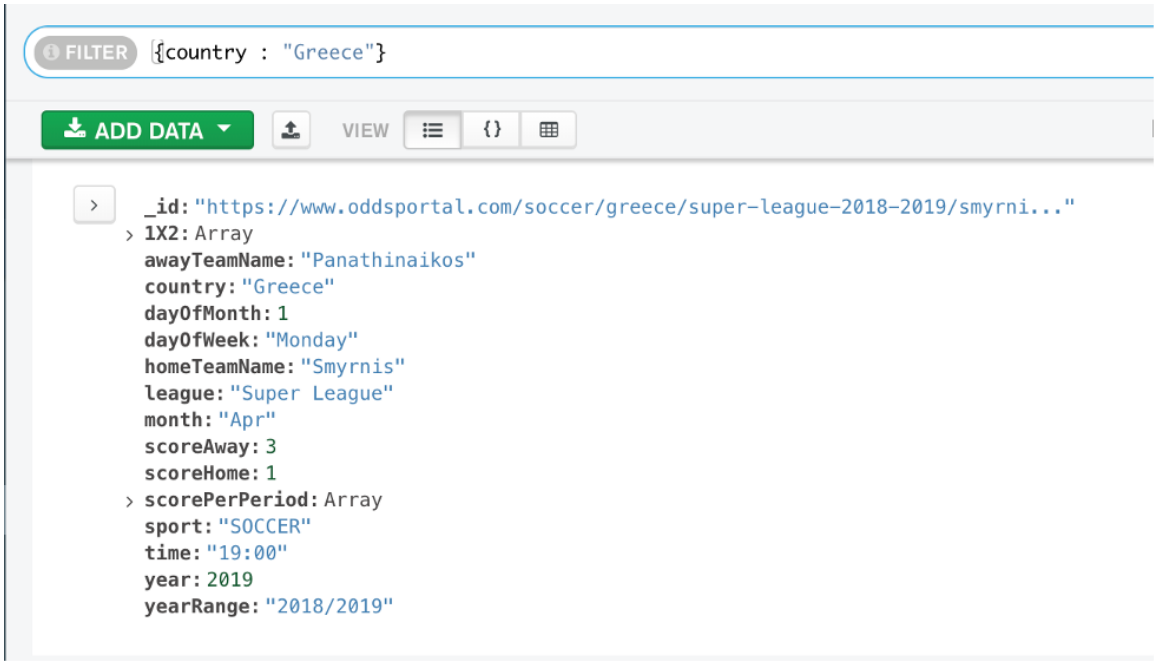
Conclusion

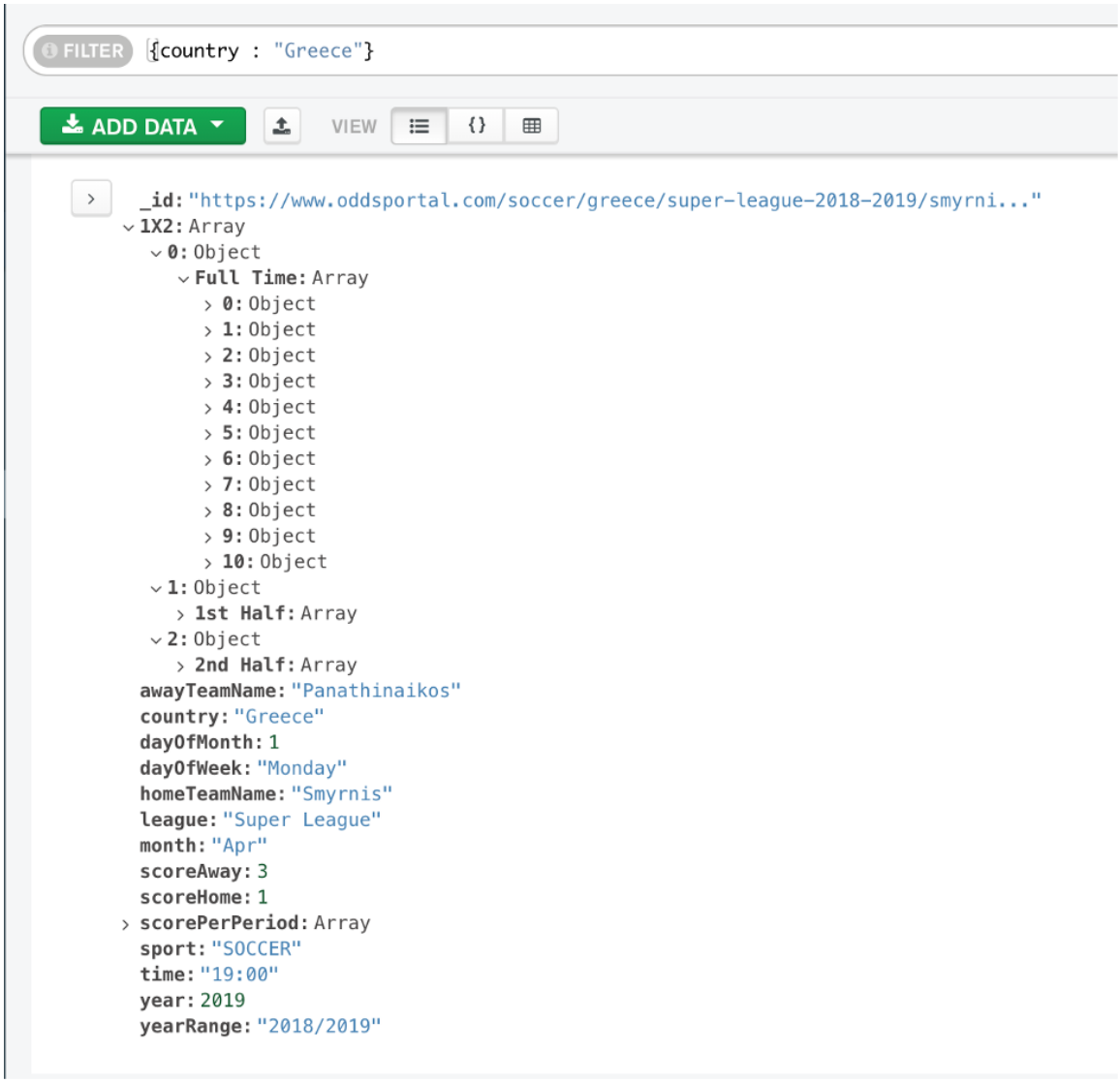
Overall, based on theoretical and empirical evidence presented in this paper, this software solution can prove to be a very reliable risk management tool for any sports betting company. While there is still a lot of room for improvement, both the company as well as the bettors can benefit from this software package. This solution can reduce the risk exposure of overvalued odds, thus increasing the profits of the company. Additionally, it would allow the company to manipulate undervalued odds, by making them more appealing to the customer's eye. This would increase the company's competitiveness and popularity. The bettors would also benefit from this solution, provided that they will have access to more consistent and appealing odd values that are closer to the true probability of a sporting event happening. To summarize, by implementing technologies such as automated data mining and supervised machine learning algorithms, betting companies can increase their risk management efficiency and build a stronger brand name in the industry.

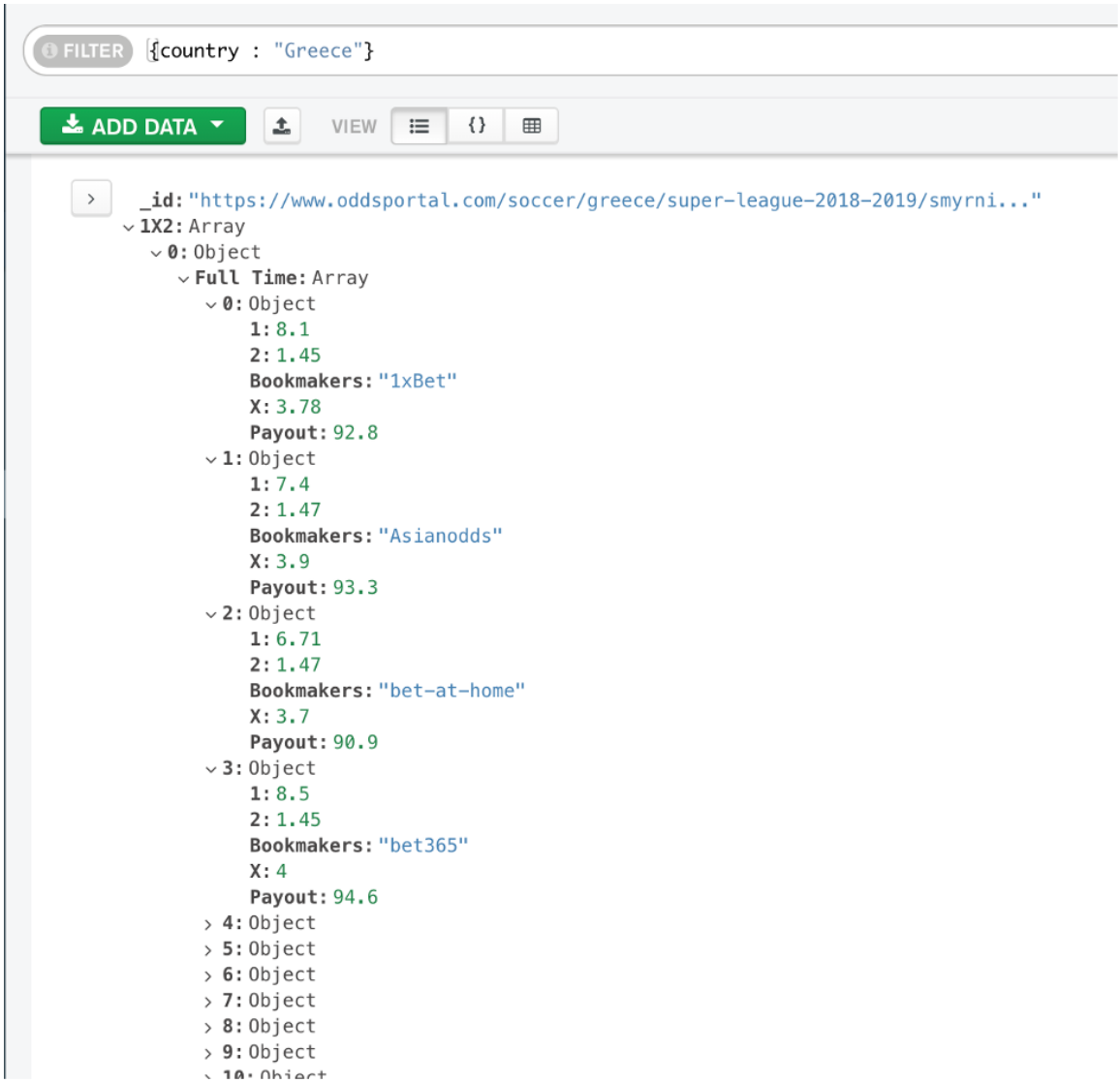
Appendix

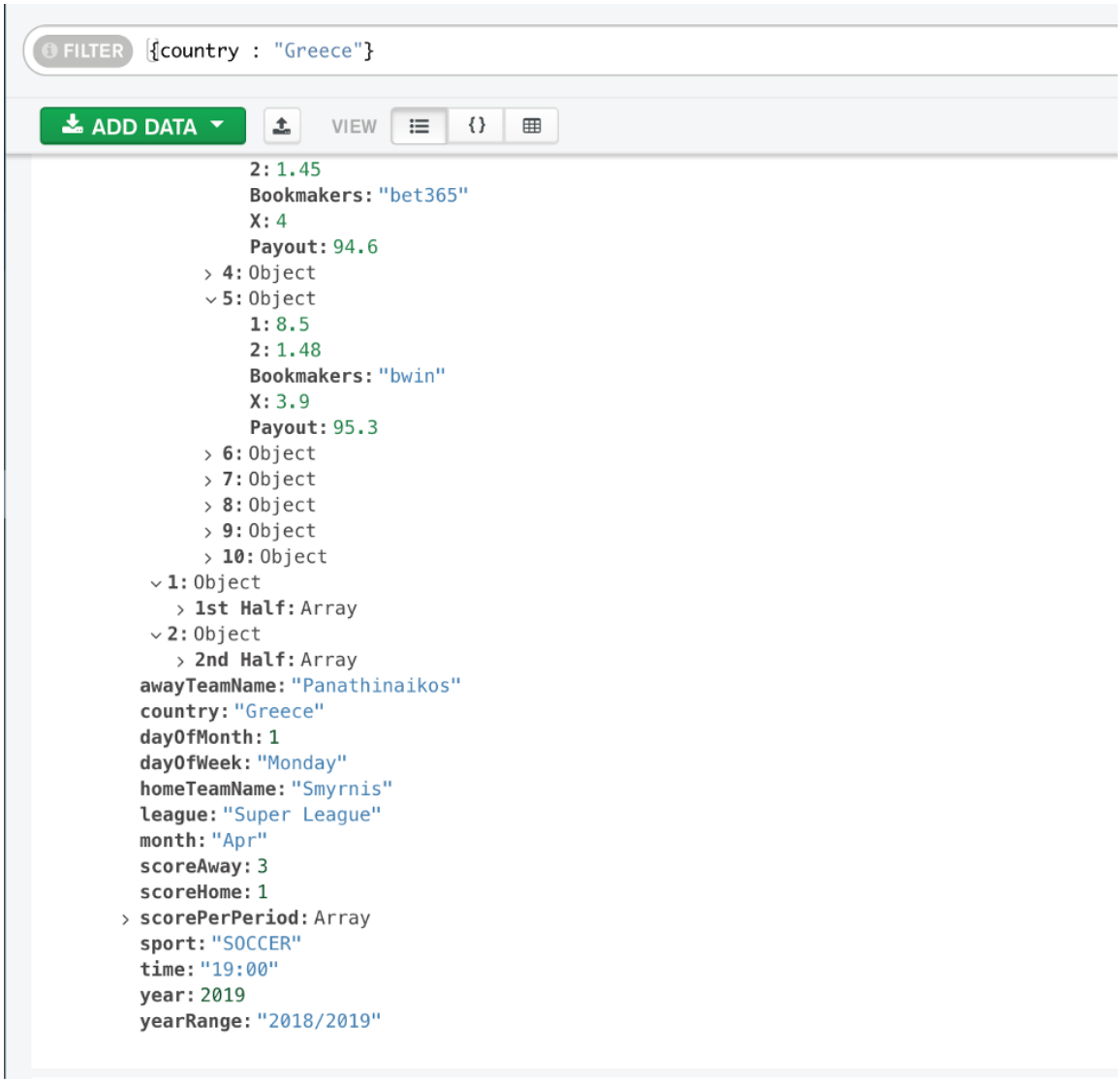
BSON Documents Structure

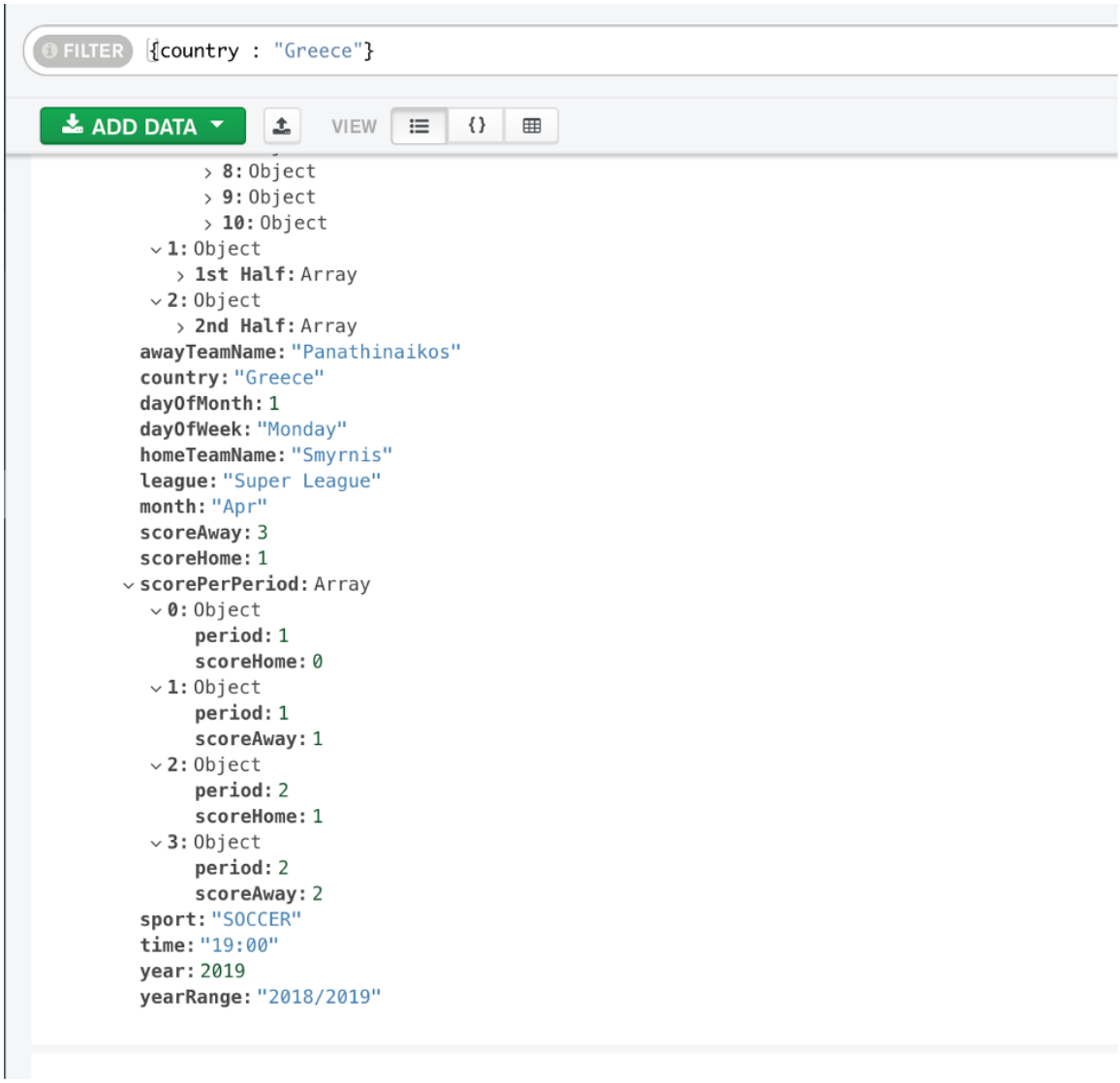
Historical Soccer Matches (Training Data)


FILTER {country : "Greece"}
ADD DATA
VIEW
_id: "https://www.oddsportal.com/soccer/greece/super-league-2018-2019/smyrni..."
1X2: Array
awayTeamName: "Panathinaikos"
country: "Greece"
dayOfMonth: 1
dayOfWeek: "Monday"
homeTeamName: "Smyrnis"
league: "Super League"
month: "Apr"
scoreAway: 3
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "19:00"
year: 2019
yearRange: "2018/2019"

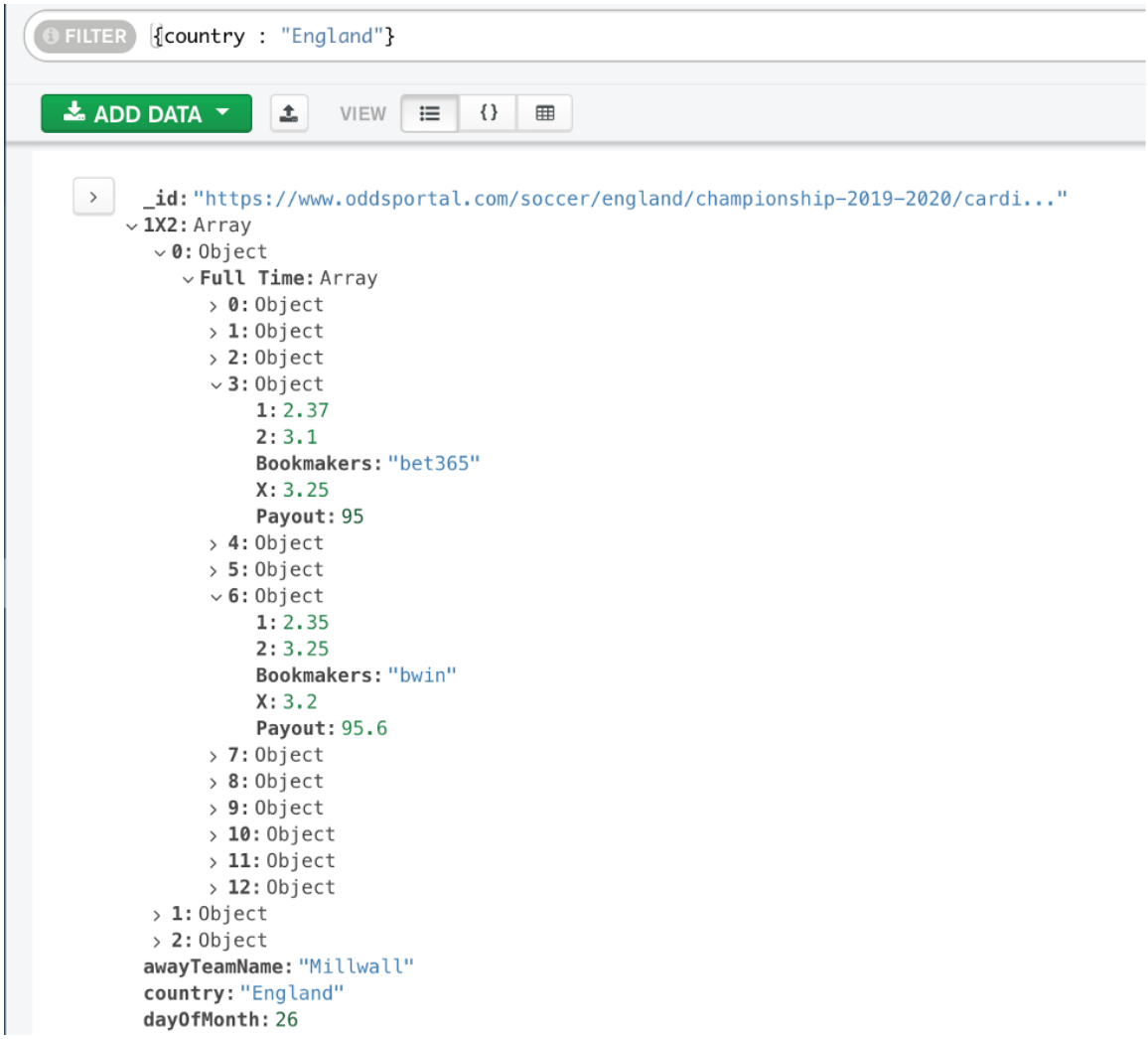

FILTER {country : "Greece"}
ADD DATA
VIEW
_id: "https://www.oddsportal.com/soccer/greece/super-league-2018-2019/smyrni..."
1X2: Array
0: Object
Full Time: Array
0: Object
1: Object
2: Object
3: Object
4: Object
5: Object
6: Object
7: Object
8: Object
9: Object
10: Object
1: Object
1st Half: Array
2: Object
2nd Half: Array
awayTeamName: "Panathinaikos"
country: "Greece"
dayOfMonth: 1
dayOfWeek: "Monday"
homeTeamName: "Smyrnis"
league: "Super League"
month: "Apr"
scoreAway: 3
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "19:00"
year: 2019
yearRange: "2018/2019"

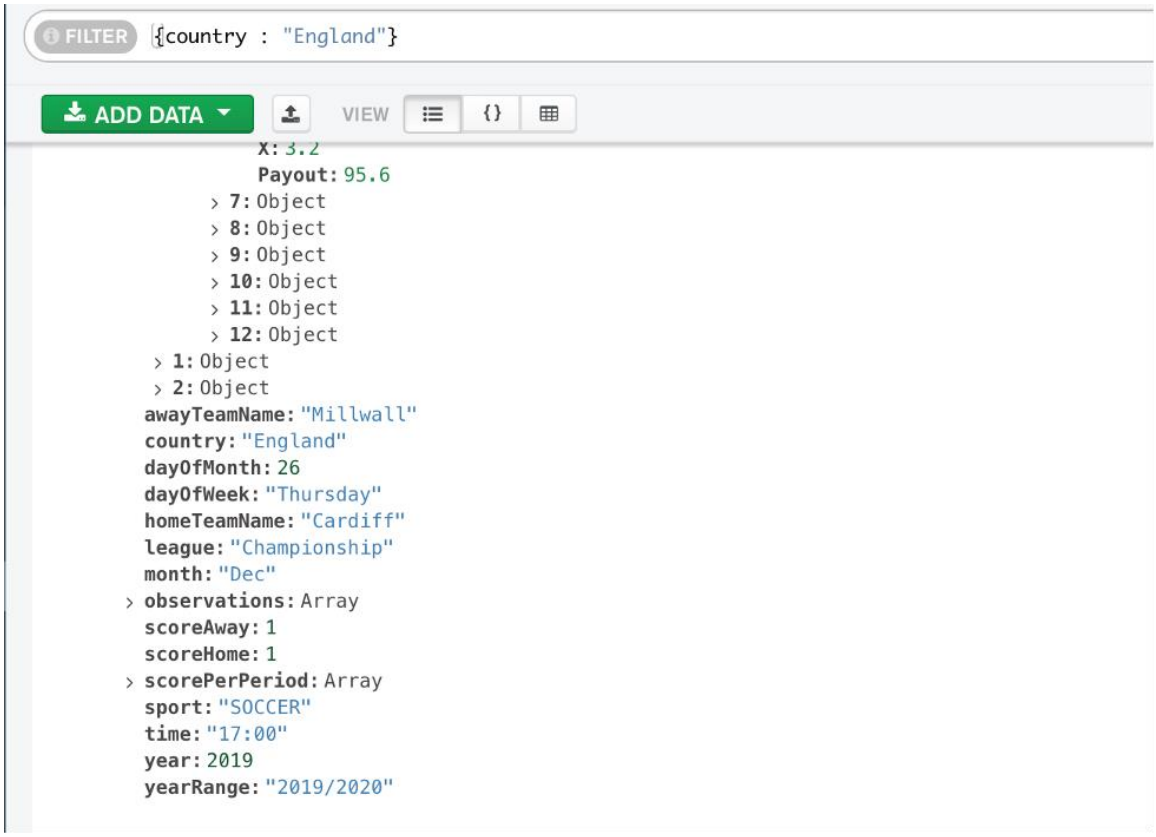

FILTER
{country : "Greece"}
ADD DATA
VIEW
{}
_id: "https://www.oddsportal.com/soccer/greece/super-league-2018-2019/smyrni..."
1X2: Array
0: Object
Full Time: Array
0: Object
1: 8.1
2: 1.45
Bookmakers: "1xBet"
X: 3.78
Payout: 92.8
1: Object
1: 7.4
2: 1.47
Bookmakers: "Asianodds"
X: 3.9
Payout: 93.3
2: Object
1: 6.71
2: 1.47
Bookmakers: "bet-at-home"
X: 3.7
Payout: 90.9
3: Object
1: 8.5
2: 1.45
Bookmakers: "bet365"
X: 4
Payout: 94.6
4: Object
5: Object
6: Object
7: Object
8: Object
9: Object

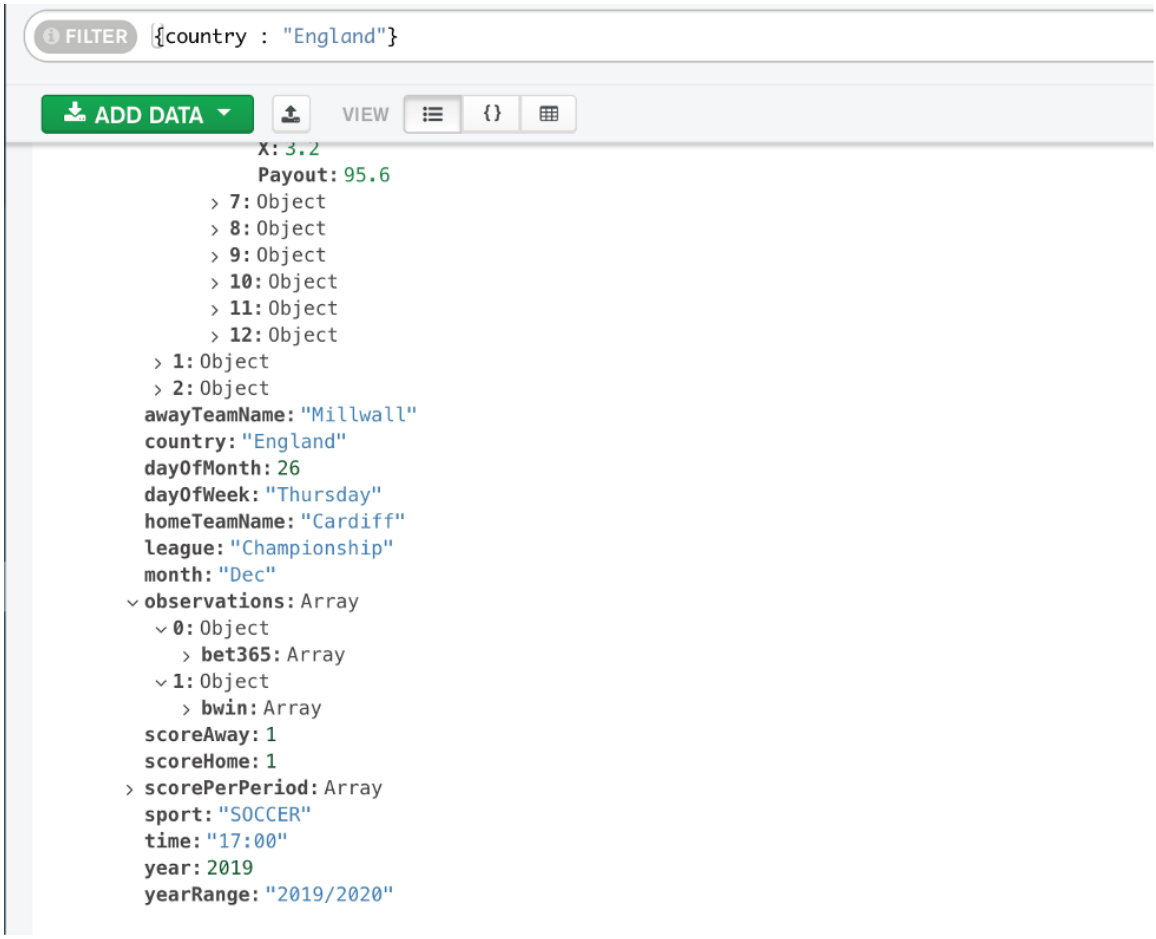

FILTER {country : "Greece"}
ADD DATA
VIEW
2: 1.45
Bookmakers: "bet365"
X: 4
Payout: 94.6
4: Object
5: Object
1: 8.5
2: 1.48
Bookmakers: "bwin"
X: 3.9
Payout: 95.3
6: Object
7: Object
8: Object
9: Object
10: Object
1: Object
1st Half: Array
2: Object
2nd Half: Array
awayTeamName: "Panathinaikos"
country: "Greece"
dayOfMonth: 1
dayOfWeek: "Monday"
homeTeamName: "Smyrnis"
league: "Super League"
month: "Apr"
scoreAway: 3
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "19:00"
year: 2019
yearRange: "2018/2019"

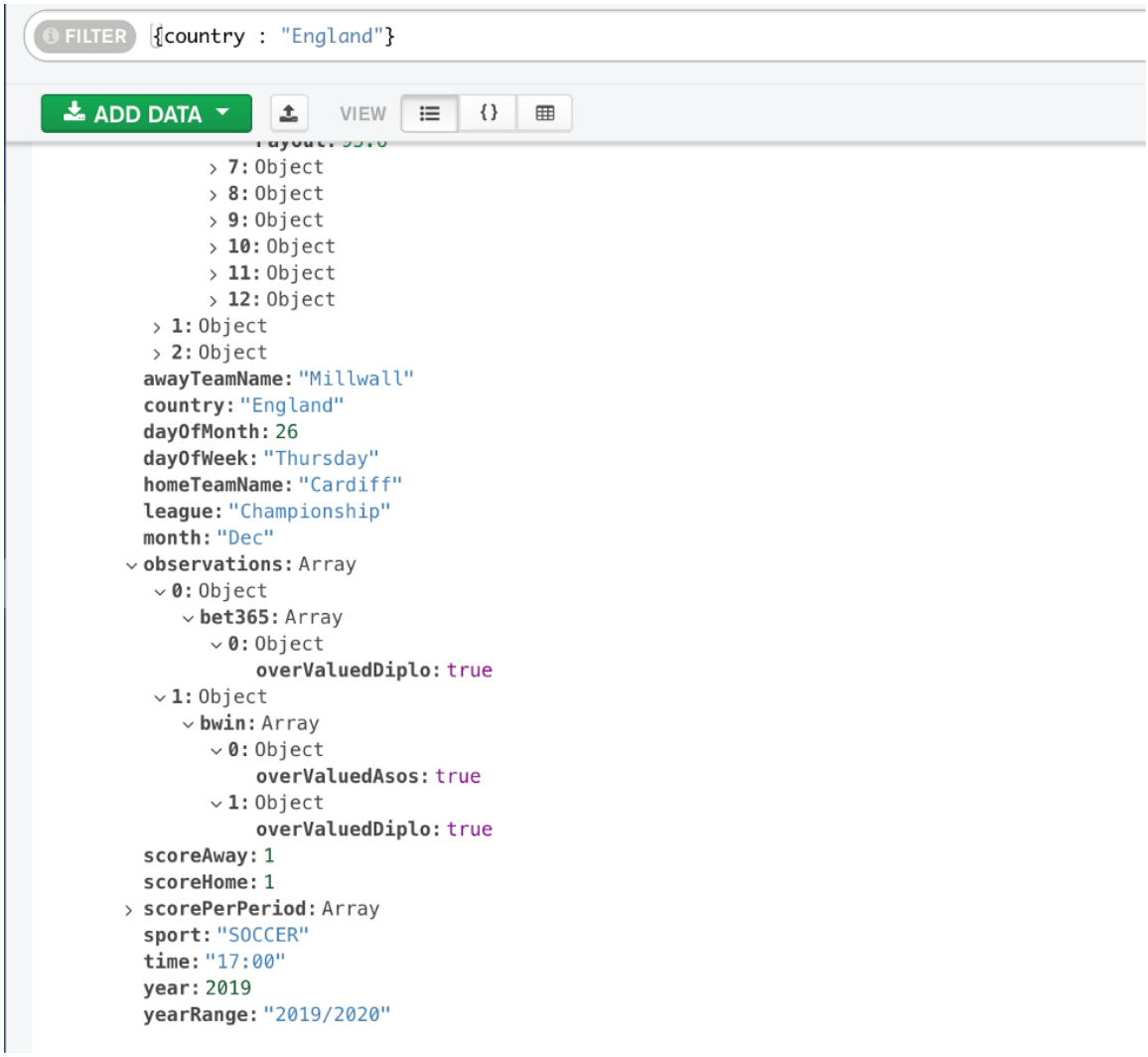

FILTER
{country : "Greece"}
ADD DATA
VIEW
8: Object
9: Object
10: Object
1: Object
1st Half: Array
2: Object
2nd Half: Array
awayTeamName: "Panathinaikos"
country: "Greece"
dayOfMonth: 1
dayOfWeek: "Monday"
homeTeamName: "Smyrnis"
league: "Super League"
month: "Apr"
scoreAway: 3
scoreHome: 1
scorePerPeriod: Array
0: Object
period: 1
scoreHome: 0
1: Object
period: 1
scoreAway: 1
2: Object
period: 2
scoreHome: 1
3: Object
period: 2
scoreAway: 2
sport: "SOCCER"
time: "19:00"
year: 2019
yearRange: "2018/2019"

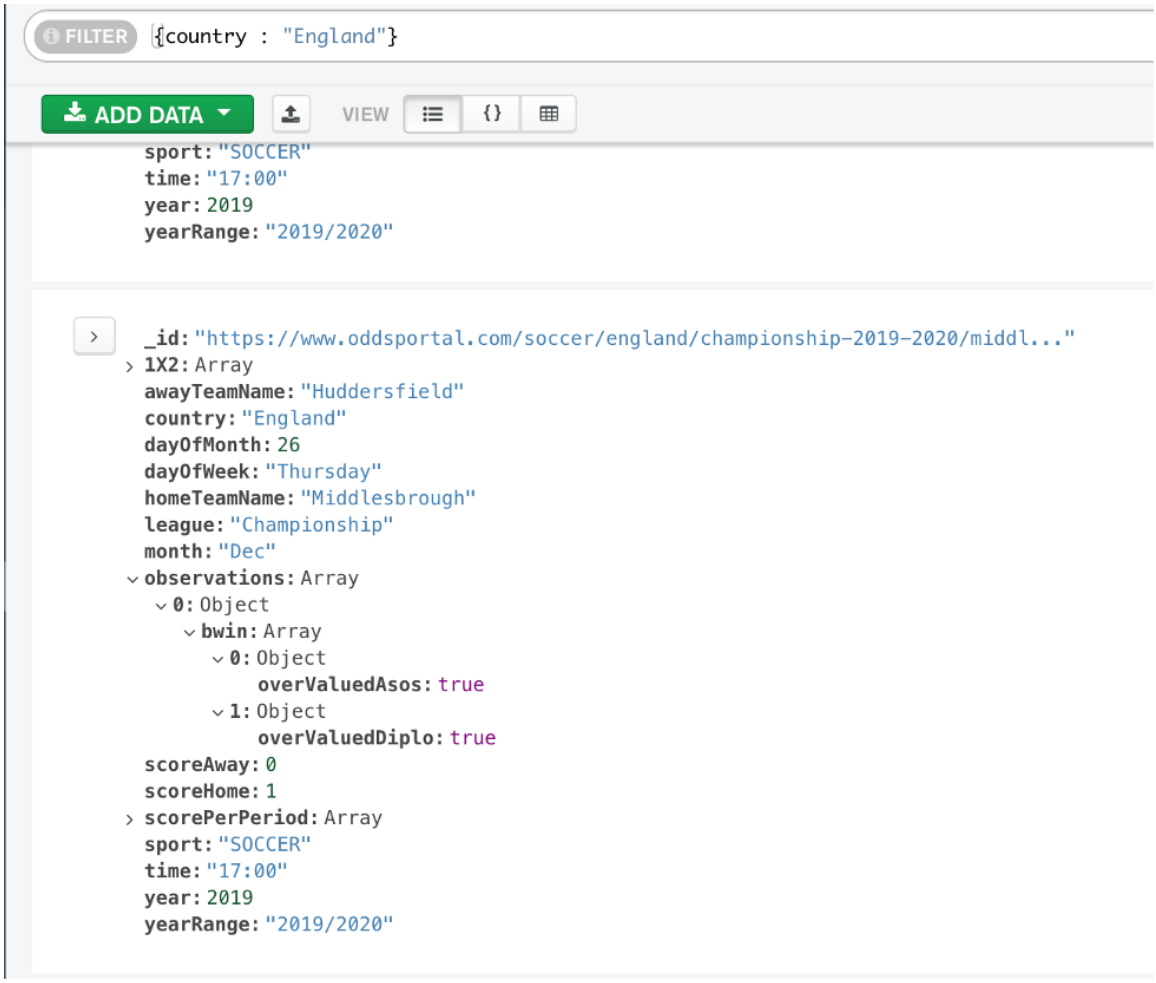
Upcoming Soccer Matches & 2019 Selected Data (Testing Data)


FILTER
{country : "England"}
ADD DATA
VIEW
_id: "https://www.oddsportal.com/soccer/england/championship-2019-2020/cardi..."
1X2: Array
0: Object
Full Time: Array
0: Object
1: Object
2: Object
3: Object
1: 2.37
2: 3.1
Bookmakers: "bet365"
X: 3.25
Payout: 95
4: Object
5: Object
6: Object
1: 2.35
2: 3.25
Bookmakers: "bwin"
X: 3.2
Payout: 95.6
7: Object
8: Object
9: Object
10: Object
11: Object
12: Object
1: Object
2: Object
awayTeamName: "Millwall"
country: "England"
dayOfMonth: 26


FILTER
{country : "England"}
ADD DATA
VIEW
X: 3.2
Payout: 95.6
7: Object
8: Object
9: Object
10: Object
11: Object
12: Object
1: Object
2: Object
awayTeamName: "Millwall"
country: "England"
dayOfMonth: 26
dayOfWeek: "Thursday"
homeTeamName: "Cardiff"
league: "Championship"
month: "Dec"
observations: Array
scoreAway: 1
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "17:00"
year: 2019
yearRange: "2019/2020"


FILTER
{country : "England"}
ADD DATA
VIEW
X: 3.2
Payout: 95.6
7: Object
8: Object
9: Object
10: Object
11: Object
12: Object
1: Object
2: Object
awayTeamName: "Millwall"
country: "England"
dayOfMonth: 26
dayOfWeek: "Thursday"
homeTeamName: "Cardiff"
league: "Championship"
month: "Dec"
observations: Array
0: Object
bet365: Array
1: Object
bwin: Array
scoreAway: 1
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "17:00"
year: 2019
yearRange: "2019/2020"


FILTER
{country : "England"}
ADD DATA
VIEW
7: Object
8: Object
9: Object
10: Object
11: Object
12: Object
1: Object
2: Object
awayTeamName: "Millwall"
country: "England"
dayOfMonth: 26
dayOfWeek: "Thursday"
homeTeamName: "Cardiff"
league: "Championship"
month: "Dec"
observations: Array
0: Object
bet365: Array
0: Object
overValuedDiplo: true
1: Object
bwin: Array
0: Object
overValuedAsos: true
1: Object
overValuedDiplo: true
scoreAway: 1
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "17:00"
year: 2019
yearRange: "2019/2020"


FILTER
{country : "England"}
ADD DATA
VIEW
sport: "SOCCER"
time: "17:00"
year: 2019
yearRange: "2019/2020"
_id: "https://www.oddsportal.com/soccer/england/championship-2019-2020/middl..."
1X2: Array
awayTeamName: "Huddersfield"
country: "England"
dayOfMonth: 26
dayOfWeek: "Thursday"
homeTeamName: "Middlesbrough"
league: "Championship"
month: "Dec"
observations: Array
0: Object
bwin: Array
0: Object
overValuedAsos: true
1: Object
overValuedDiplo: true
scoreAway: 0
scoreHome: 1
scorePerPeriod: Array
sport: "SOCCER"
time: "17:00"
year: 2019
yearRange: "2019/2020"

Machine Learning Results & Findings

```
/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Sweden, league: Allsvenskan, and provider: bet365
OverValued Asos: Draws -> 25.0%, Home -> 75.0%, Away -> 0.0%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: France, league: Ligue 2, and provider: bet365
UnderValued Diplo: Draws -> 31.25%, Home -> 50.0%, Away -> 18.75%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Belgium, league: Jupiler League, and provider: bet365
UnderValued Diplo: Draws -> 31.71%, Home -> 46.34%, Away -> 21.95%

/*********************************************************************************************************/
```

```
/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Costa Rica, league: Primera Division, and provider: bet365
OverValued Asos: Draws -> 21.74%, Home -> 65.22%, Away -> 13.04%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Czech Republic, league: 1. Liga, and provider: bet365
OverValued Asos: Draws -> 0.0%, Home -> 75.0%, Away -> 25.0%
Overvalued Both: Draws -> 10.0%, Home -> 75.0%, Away -> 15.0%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Czech Republic, league: Division 2, and provider: bet365
UnderValued Diplo: Draws -> 25.0%, Home -> 54.17%, Away -> 20.83%

/*********************************************************************************************************/
```

```
/*********************************************************************************************/

Statistical Data for the year: 2019, country: Japan, league: J1 League, and provider: bet365
UnderValued Asos: Draws -> 23.53%, Home -> 11.76%, Away -> 64.71%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Mexico, league: Liga MX, and provider: bet365
UnderValued Diplo: Draws -> 31.25%, Home -> 59.38%, Away -> 9.38%

/*********************************************************************************************/
```

```
/*********************************************************************************************/

Statistical Data for the year: 2019, country: Netherlands, league: Eerste Divisie, and provider: bet365
UnderValued Diplo: Draws -> 20.0%, Home -> 65.0%, Away -> 15.0%
Overvalued Both: Draws -> 26.67%, Home -> 61.9%, Away -> 11.43%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Netherlands, league: Eredivisie, and provider: bet365
UnderValued Diplo: Draws -> 21.21%, Home -> 57.58%, Away -> 21.21%
Overvalued Both: Draws -> 16.98%, Home -> 62.26%, Away -> 20.75%

/*********************************************************************************************/
```

```
/*********************************************************************************************/

Statistical Data for the year: 2019, country: Norway, league: OBOS-ligaen, and provider: bet365
OverValued Asos: Draws -> 23.08%, Home -> 61.54%, Away -> 15.38%
UnderValued Diplo: Draws -> 27.03%, Home -> 51.35%, Away -> 21.62%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Poland, league: Ekstraklasa, and provider: bet365
UnderValued Diplo: Draws -> 20.0%, Home -> 60.0%, Away -> 20.0%

/*********************************************************************************************/
```

```
/*********************************************************************************************/

Statistical Data for the year: 2019, country: Serbia, league: Super Liga, and provider: bet365
Overvalued Both: Draws -> 23.08%, Home -> 63.46%, Away -> 13.46%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Spain, league: LaLiga, and provider: bet365
UnderValued Diplo: Draws -> 28.89%, Home -> 55.56%, Away -> 15.56%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Switzerland, league: Challenge League, and provider: bet365
UnderValued Diplo: Draws -> 27.78%, Home -> 55.56%, Away -> 16.67%
Overvalued Both: Draws -> 12.0%, Home -> 60.0%, Away -> 28.0%

/*********************************************************************************************/
```

```
/*********************************************************************************************/

Statistical Data for the year: 2019, country: Uruguay, league: Primera Division, and provider: bet365
UnderValued Diplo: Draws -> 30.3%, Home -> 51.52%, Away -> 18.18%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Wales, league: Cymru Premier, and provider: bet365
Overvalued Both: Draws -> 5.26%, Home -> 68.42%, Away -> 26.32%

/*********************************************************************************************/


/*********************************************************************************************/

Statistical Data for the year: 2019, country: Greece, league: Super League, and provider: bwin
OverValued Asos: Draws -> 25.0%, Home -> 66.67%, Away -> 8.33%
Overvalued Both: Draws -> 15.79%, Home -> 63.16%, Away -> 21.05%

/*********************************************************************************************/
```

```
/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Turkey, league: Super Lig, and provider: bwin
Undervalued Asos, Overvalued Diplo: Draws -> 34.78%, Home -> 21.74%, Away -> 43.48%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Uruguay, league: Primera Division, and provider: bwin
UnderValued Diplo: Draws -> 31.25%, Home -> 43.75%, Away -> 25.0%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Venezuela, league: Primera Division, and provider: bwin
Undervalued Asos, Overvalued Diplo: Draws -> 56.52%, Home -> 17.39%, Away -> 26.09%

/*********************************************************************************************************/
```

```
/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Netherlands, league: Eredivisie, and provider: bwin
OverValued Asos: Draws -> 11.63%, Home -> 69.77%, Away -> 18.6%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Poland, league: Ekstraklasa, and provider: bwin
UnderValued Diplo: Draws -> 15.79%, Home -> 78.95%, Away -> 5.26%

/*********************************************************************************************************/
```

```
/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Chile, league: Primera Division, and provider: bwin
UnderValued Diplo: Draws -> 20.83%, Home -> 58.33%, Away -> 20.83%

/*********************************************************************************************************/


/*********************************************************************************************************/

Statistical Data for the year: 2019, country: Cyprus, league: First Division, and provider: bwin
OverValued Asos: Draws -> 12.5%, Home -> 68.75%, Away -> 18.75%

/*********************************************************************************************************/
```

Tools & Technologies Used

Editors Used:

- InteliJ IDEA
- Anaconda (Spyder)
- Visual Studio Code

Programming Languages:

- Python
- Java

Libraries:

- Jsoup
- Selenium Web Automation
- Mongo DB

Third Party Applications & Drivers:

- Mongo Compass
- Chrome x86 Web Driver (Mac OS & Windows)
- Phantom JS

Source Code

```
    if(machineLearning.online) {
        //set up the webdriver
        machineLearning.driver = WebDriverFunctions.setUpWebDriver();
        //Call the fucation to check for done matches and update all the databases accordingly
        machineLearning.UpdateStateOfUpcomingMatches();
        machineLearning.testCollectedHistoricalMatches(machineLearning.oddsPortalMLTrainedCompletedMatchesDB);
    }else{

        //get each country in the trained database and their tournaments respectively and run the statistics class
        ArrayList<String> countriesAlreadyTrainedFor = MongoDBFunctions.getCountries(machineLearning.oddsPortalMLMatchesDB, machineLearning.currentSport);
        ArrayList<String> remainingCountries = MongoDBFunctions.getCountries(machineLearning.oddsPortalDB, machineLearning.currentSport);

        //train and test the ML agent with matches from 2019 from the vian countries
        for(String country : remainingCountries) {

            if (!countriesAlreadyTrainedFor.contains(country)) {
                if (!Arrays.asList(machineLearning.countriesToExcule).contains(country)) {
                    System.out.println("To Train -> " + country);
                    //machineLearning.TrainMLAgentWithHistoricalMatches(country, machineLearning.year);
                }else{
                    System.out.println("Excluding this country due to limitations: " + country);
                }

            }else{
                System.out.println("Country data and matches already trained the ML : " + country);
            }
        }

        //Test the trained data
        machineLearning.testCollectedHistoricalMatches(machineLearning.oddsPortalMLMatchesDB);

    }
    //Terminate all active connections and call System Exit (0)
    HelperFunctions.terminateAndCloseConnections(machineLearning.client,  driver: null);
}//end of main
```

```
The ONLINE training tool for the machine learning agent

//region The OFFLINE training tool for the machine learning agent
void TrainMLAgentWithHistoricalMatches(String givenCountry, int givenYear) {

    //Connect to a collection
    MongoCollection<Document> collection = oddsPortalDB.getCollection(currentSport);
    FindIterable<Document> findIterable;

    //Get the documemtns (mathces) for which the prediction should happen (year and country)
    findIterable = collection.find(Filters.and(
            Filters.eq( fieldName: "year", givenYear),
            Filters.eq( fieldName: "country", givenCountry)
    ));

    findIterable.noCursorTimeout(true);

    //Concurrent Multithreading to check several matches in one iteration of the for loop by
    //performing multithreading
```

```java
    //Concurrent Multithreading to check several matches in one iteration of the for loop by
    //performing multithreading

    final ExecutorService executor = Executors.newFixedThreadPool(threadPoolSize); // it's just an arbitrary number
    final List<Future<?>> futures = new ArrayList<>();

    for (Document match : findIterable) {
        //each one of the doc is a document ( an upcoming match )
        try {
            Future<?> future = executor.submit(() -> {
            OddsErrorPrediction(match, oddsPortalDB , oddsPortalMLMatchesDB,  historicalData: true);
            });
            futures.add(future);
        } catch (Exception e) {
            System.out.println("Failed to check match from the historical matches odds portal database...");
        }
    }

    try {
        for (Future<?> future : futures) {
            future.get(); // do anything you need, e.g. isDone(), ...
        }
    } catch (InterruptedException | ExecutionException e) {
        e.printStackTrace();
    }

}
```

```java
public void OddsErrorPrediction(Document currentMatch, MongoDatabase inputDB , MongoDatabase outputDB , boolean historicalData) {

    String country, league, mainBetType = "1X2" , subBetType = "Full Time";
    ArrayList<Document> observations = new ArrayList<>();
    List<String> providersVisited = new ArrayList<>();

    try {
        //first get the matche's country
        country = currentMatch.get("country").toString();
        league = currentMatch.get("league").toString();
        //first get the bet eg: 1X2 and then get the Sub bet Array list holding the documents (Providers with their odds)
        // List<Document> subBetTypeProviders = (List<Document>) currentMatch.get(mainBetType + "." + subBetType);
        //run the test to find any odd errors for all the requested providers
        ArrayList<Document> mainBet = new ArrayList<>();
        ArrayList<Document> subBetTypeProvidersList = new ArrayList<>();

        try {
            mainBet = (ArrayList<Document>) currentMatch.get(mainBetType);
        }catch (Exception ignore){return;}
        //THe 1X2 contains a list of documents, and this list of documents contains one to three lists of documents with keys:
        // Full Time, 1st Half, 2nd Half
        //Lets get the Full Time
        for (Document subBet : mainBet) {
            //now check if full time exists, and if so get it :
            //Full time is going to be of type list of documents
            try {
                subBetTypeProvidersList = (ArrayList<Document>) subBet.get(subBetType);
                break;
            } catch (Exception ignore) { }
        }

        if (subBetTypeProvidersList == null) { return; }
```

```
    //Now we need to check if the providerRetrievedName matches with the one in our database
    if(Arrays.asList(errorPredictionProviders).contains(providerRetrievedName) && !providersVisited.contains(providerRetrievedName)){

        providersVisited.add(providerRetrievedName);

        try {
            asos = provider.getDouble( key: "1");
            diplo = provider.getDouble( key: "2");

        }catch (Exception ignore){continue;}

        //Create a document list for the provider to hold all of the observations made:
        //here we are either checking for bet 365 or bwin
        //we need to go into the given collection and retrieve the data
        //Here we check the asos value for errors and add the value result to the document
        //System.out.println(provider);
        ArrayList<Document> providerObservations =  checkForFaultyOdds(inputDB, mongoDBCollectionName: "SOCCER", providerRetrievedName, country, league, asos, diplo, historicalData );

        if(!providerObservations.isEmpty()){
            //this match proves to cause risk because of it's odds, thus we need to print it on the screen
            //check if the matches's odds are
            Document providerObservationsDocument = new Document();
            providerObservationsDocument.append(providerRetrievedName, providerObservations);
            observations.add(providerObservationsDocument);

        }else{
            System.out.println("The match value odds seem to be ok for the provider: " + providerRetrievedName);
        }

    }
}//end of period for loop
```

```
            //Now insert to mongo db
            if(!observations.isEmpty()){
                currentMatch.append("observations", observations);
                //Insert the match into the database as to be stored as an error prone match
                MongoDBFunctions.insertMatchToMongo( currentSport ,currentMatch, currentMatch.get("_id").toString(), outputDB );

            }

        }catch (Exception e){
            //We could not get all the neccessary information from the match
            System.out.println("***** Could not perform check for the odd values *****");
        }
    }

    ArrayList<Document> checkForFaultyOdds (MongoDatabase DB, String mongoDBCollectionName, String matchProvider, String country, String league ,  double matchAsos, double matchDiplo, boolean historicalData ){

        Boolean [] booleansToRunFor = {true, false};
        double minBettingValue, maxBettingValue;
        String asosOrDiploString, overValuedString;
        int decimalPlaces = 3;
        ArrayList<Document> providerObservations = new ArrayList<>();
```

```
            //These are BSON documents used to query the data and find which team won the game
            Document home = Document.parse(" { $gt: [ \"$scoreHome\" , \"$scoreAway\"] } ");
            Document away = Document.parse(" { $lt: [ \"$scoreHome\" , \"$scoreAway\"] } ");
            //Get the current match country and league
            //Connect to a collection
            MongoCollection<Document> collection = DB.getCollection(mongoDBCollectionName);
            Bson filter, homeFilter, awayFilter;
            String filterForOdds = "{  \"" + asosOrDiploString + "\"  : { $gte:  " + minBettingValue + "  , $lt: " + maxBettingValue + "  }, Bookmakers:  \"" + matchProvider + "\" }";

            //Get its documents
            if (historicalData) {
                filter = Filters.and(Filters.lt( fieldName: "year", year),
                        Filters.eq( fieldName: "league", league),
                        Filters.eq( fieldName: "country", country),
                        Filters.elemMatch( fieldName: "1X2.Full Time", Document.parse(filterForOdds)));

                homeFilter = Filters.and(Filters.lt( fieldName: "year", year),
                        Filters.eq( fieldName: "league", league),
                        Filters.eq( fieldName: "country", country),
                        Filters.expr(home),
                        Filters.elemMatch( fieldName: "1X2.Full Time", Document.parse(filterForOdds)));

                awayFilter = Filters.and(Filters.lt( fieldName: "year", year),
                        Filters.eq( fieldName: "league", league),
                        Filters.eq( fieldName: "country", country),
                        Filters.expr(away),
                        Filters.elemMatch( fieldName: "1X2.Full Time", Document.parse(filterForOdds)));
```

```
        //find the statistics and related percentages required for the given odd, in addition to calculating and allowing some room for statistical, probability errors
        double meanOfBettingValues = round( value: (minBettingValue + maxBettingValue) / 2, decimalPlaces);
        double percentage = round( value: 1.0 / meanOfBettingValues, decimalPlaces);
        double statisticalErrorRange = round( value: (percentage * statisticalRoomForError) / 100.0 , decimalPlaces);
        double minPercentageWithError = round( value: percentage - statisticalErrorRange, decimalPlaces);
        double maxPercentageWithError = round( value: percentage + statisticalErrorRange, decimalPlaces);
        //System.out.println("minPercentageWithError : "+ minPercentageWithError);
        //System.out.println("maxPercentageWithError : "+ maxPercentageWithError);

        //find the percentage of success for the given matches (How many out of the returned matches were validated
        double percentageOfSuccess = round( value: (double) matchesValidatingHypothesis / matchesBetweenGivenOdds, decimalPlaces);
        //System.out.println(percentageOfSuccess);
        //if the percentage of success is under the minimum percentage error then it means the value is undervalued, else it is overrated
        //insert the identified odds into the observations Database in order to create an alert system for
        if (percentageOfSuccess < minPercentageWithError) {
            providerObservations.add(new Document(overValuedString, true));
            //System.out.println(new Document(overValuedString, true));
        } else if (percentageOfSuccess > maxPercentageWithError) {
            providerObservations.add(new Document(overValuedString, false));
            //System.out.println(new Document(overValuedString, false));

        }

    }

    return providerObservations;
}
```

```
//region TESTING MACHINE LEARNING TOOL:
void testCollectedHistoricalMatches(MongoDatabase DB) {

    //get each country in the trained database and their tournaments respectively and run the statistics class
    ArrayList<String> countries = MongoDBFunctions.getCountries(oddsPortalMLMatchesDB, currentSport);

    for (String providerName : errorPredictionProviders) {
        //Connect to a collection
        MongoCollection<Document> collection = DB.getCollection(currentSport);

        for (String country : countries) {
            ArrayList<String> tournaments = MongoDBFunctions.getTournaments(oddsPortalMLMatchesDB, currentSport, country);
            for (String tournament : tournaments) {
                //Reset the counters
                StatisticalObservations(country, tournament ,providerName, collection);
            }
        }//end of countries loop
    }//end of outer for loop
}
```

```
void StatisticalObservations(String country, String league, String provider,  MongoCollection collection ) {

    //set the desired maximum number of decimals for the representation of the percentages
    int decimalPlaces = 2;

    //region filters and declarations
    Bson draw = Filters.expr(Document.parse(" { $eq: [ \"$scoreHome\" , \"$scoreAway\"] } "));
    Bson homeWins = Filters.expr(Document.parse(" { $gt: [ \"$scoreHome\" , \"$scoreAway\"] } "));
    Bson awayWins = Filters.expr(Document.parse(" { $lt: [ \"$scoreHome\" , \"$scoreAway\"] } "));
    Bson countryFilter = Filters.eq( fieldName: "country", country);
    Bson leagueFilter = Filters.eq( fieldName: "league", league);
    //Existance Bson Filters
    Bson bothExist = Filters.and(Filters.exists( fieldName: "observations." + provider + ".overValuedAsos",  exists: true),
    Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: true));

    Bson onlyAsosExists = Filters.and(Filters.exists( fieldName: "observations." + provider + ".overValuedAsos",  exists: true),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: false));

    Bson onlyDiploExists = Filters.and(Filters.exists( fieldName: "observations." + provider + ".overValuedAsos",  exists: false),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: true));

    Bson bothOverValued = Filters.and(Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: true),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: true));
    //
    Bson bothUnderValued = Filters.and(Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: false),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: false));
    //
    Bson overValuedAsosUnderValuedDiplo = Filters.and(Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: true),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: false));
    //
    Bson underValuedAsosOverValuedDiplo = Filters.and(Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: false),
            Filters.exists( fieldName: "observations." + provider + ".overValuedDiplo",  exists: true));
    //
    Bson overValuedAsos = Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: true);
    //
    Bson underValuedAsos = Filters.eq( fieldName: "observations." + provider + ".overValuedAsos",  value: false);
    //
    Bson overValuedDiplo = Filters.eq( fieldName: "observations." + provider + ".overValuedDiplo",  value: true);
    //
    Bson underValuedDiplo = Filters.eq( fieldName: "observations." + provider + ".overValuedDiplo",  value: false);


    //endregion
```

```
    //region Percentages Calculations
    double overValuedAsosHomePercentage = ((double)overvaluedAsosHome / (double)totalOvervaluedAsos) * 100;
    double overValuedAsosAwayPercentage = ((double)overvaluedAsosAway / (double)totalOvervaluedAsos) * 100 ;
    double overValuedAsosDrawPercentage = ((double)overvaluedAsosDraw / (double)totalOvervaluedAsos) * 100 ;

    double underValuedAsosHomePercentage = ((double)undervaluedAsosHome / (double)totalUndervaluedAsos) * 100;
    double underValuedAsosAwayPercentage = ((double)undervaluedAsosAway / (double)totalUndervaluedAsos) * 100;
    double underValuedAsosDrawPercentage = ((double)undervaluedAsosDraw / (double)totalUndervaluedAsos) * 100 ;

    double overValuedDiploHomePercentage = ((double)overvaluedDiploHome / (double)totalOvervaluedDiplo) * 100;
    double overValuedDiploAwayPercentage = ((double)overvaluedDiploAway / (double)totalOvervaluedDiplo) * 100;
    double overValuedDiploDrawPercentage = ((double)overvaluedDiploDraw / (double)totalOvervaluedDiplo) * 100;

    double undervaluedDiploHomePercentage = ((double)undervaluedDiploHome / (double)totalUndervaluedDiplo) * 100;
    double undervaluedDiploAwayPercentage = ((double)undervaluedDiploAway / (double)totalUndervaluedDiplo) * 100;
    double undervaluedDiploDrawPercentage = ((double)undervaluedDiploDraw / (double)totalUndervaluedDiplo) * 100;

    double overvaluedAsosUndervaluedDiploHomePercentage = ((double)overvaluedAsosUndervaluedDiploHome / (double)totalOvervaluedAsosUndervaluedDiplo) * 100;
    double overvaluedAsosUndervaluedDiploAwayPercentage = ((double)overvaluedAsosUndervaluedDiploAway / (double)totalOvervaluedAsosUndervaluedDiplo) * 100;
    double overvaluedAsosUndervaluedDiploDrawPercentage = ((double)overvaluedAsosUndervaluedDiploDraw / (double)totalOvervaluedAsosUndervaluedDiplo) * 100 ;

    double undervaluedAsosOvervaluedDiploHomePercentage = ((double)undervaluedAsosOvervaluedDiploHome / (double)totalUndervaluedAsosOvervaluedDiplo) * 100;
    double undervaluedAsosOvervaluedDiploAwayPercentage = ((double)undervaluedAsosOvervaluedDiploAway / (double)totalUndervaluedAsosOvervaluedDiplo) * 100;
    double undervaluedAsosOvervaluedDiploDrawPercentage = ((double)undervaluedAsosOvervaluedDiploDraw / (double)totalUndervaluedAsosOvervaluedDiplo) * 100 ;

    double bothOvervaluedHomePercentage = ((double)bothOvervaluedHome / (double)totalBothOvervalued) * 100;
    double bothOvervaluedAwayPercentage = ((double)bothOvervaluedAway / (double)totalBothOvervalued) * 100;
    double bothOvervaluedDrawPercentage = ((double)bothOvervaluedDraw / (double)totalBothOvervalued) * 100 ;

    double bothUndervaluedHomePercentage = ((double)bothUndervaluedHome / (double)totalBothUndervalued) * 100;
    double bothUndervaluedAwayPercentage = ((double)bothUndervaluedAway / (double)totalBothUndervalued) * 100;
    double bothUndervaluedDrawPercentage = ((double)bothUndervaluedDraw / (double)totalBothUndervalued) * 100;
```

```
if(shouldPrintData){
    System.out.println("\n/*****************************************************************************************/\n");
    System.out.println("Statistical Data for the year: " + year + ", country: " + country + ", league: " + league + ", and provider: " + provider);
}
//Here print the results for each one of the providers:

if(totalOvervaluedAsos > minimumMatchesThresholdStatistics && overValuedAsosHomePercentage >= significantOverValuedThreshold ) {
    //Here print the results for each one of the providers:
    System.out.println("OverValued Asos: Draws -> " + round(overValuedAsosDrawPercentage, decimalPlaces) + "%" +
            ", Home -> " + round(overValuedAsosHomePercentage, decimalPlaces) + "%" +
            ", Away -> " +  round(overValuedAsosAwayPercentage, decimalPlaces) + "%" );
}

if(totalUndervaluedAsos > minimumMatchesThresholdStatistics && underValuedAsosHomePercentage <= significantUnderValuedThreshold) {
    //Here print the results for each one of the providers:
    System.out.println("UnderValued Asos: Draws -> "  + round(underValuedAsosDrawPercentage, decimalPlaces) + "%" +
            ", Home -> " + round(underValuedAsosHomePercentage, decimalPlaces ) + "%" +
            ", Away -> " + round(underValuedAsosAwayPercentage , decimalPlaces) + "%");
}
```

```
//region MONGODB SET UP & FUNCTIONS

public static MongoClient setUpMongoClient () {

    return MongoClients.create("mongodb://localhost:27017");
}


public static MongoDatabase setUpMongoDB(String DatabaseName, MongoClient client) {

    try {
        //First get all the databseses and prompt user to connect to one
        // Creating a Mongo client
        //Connect to the DB

        return client.getDatabase(DatabaseName);

    }catch (Exception ex) {
        ex.printStackTrace();
        return null;
    }

}

//endregion
```

```java
static void insertMatchToMongo(String currentCollection , org.bson.Document currentMatch, String uniqueID, MongoDatabase DB ) {
    MongoCollection collection = DB.getCollection(currentCollection);
    //Using update_one () method for single update.
    UpdateOptions options = new UpdateOptions().upsert(true);

    //Create a query which will search the match we want to replace:
    org.bson.Document query = new org.bson.Document();
    query.append("_id", uniqueID);

    org.bson.Document update = new org.bson.Document();
    update.append("$set", currentMatch);

    try {
        collection.updateOne(query, update, options);
        //stoiximanDB[currentSport].update(currentMatch, currentMatch, upsert = True)
        Timestamp timestamp = new Timestamp(System.currentTimeMillis());
        System.out.println(uniqueID + " was updated at: " + DeclarationsAndConstants.sdf.format(timestamp));

    }catch (Exception e){
        //could not update the document so lets replace it :
       System.out.println(e + "\nCould not insert or update the current match..");
    }
}
```

```java
static void updateOddsStructure (String currentCollection , org.bson.Document currentMatch, String uniqueID, MongoDatabase DB){

    try {

        MongoCollection collection = DB.getCollection(currentCollection);
        //Using update_one () method for single update.
        UpdateOptions options = new UpdateOptions().upsert(true);
        //Create a query which will search the match we want to replace:
        org.bson.Document query = new org.bson.Document();
        query.append("_id", uniqueID);

        System.out.println(currentMatch.toJson());

        try {
            ArrayList<ArrayList<Document>> EnaXDiplo = (ArrayList<ArrayList<Document>>) currentMatch.get("1X2");
            ArrayList<Document> EnaXDiploUpdated = new ArrayList<>();
            //THe 1X2 contains a list of documents, and this list of documents contains one to three lists of documents with keys:
            // Full Time, 1st Half, 2nd Half
            //Lets get the Full Time
            for (ArrayList<Document> periods : EnaXDiplo) {

                for (Document periodDoc : periods) {

                    //now check if full time exists, and if so get it :
                    //Full time is going to be of type list of documents
                    if (periodDoc.containsKey("Full Time")) {
                        ArrayList<Document> fullTimeProvidersList = (ArrayList<Document>) periodDoc.get("Full Time");
                        Document newPeriodDoc = new Document("Full Time",fullTimeProvidersList);
                        EnaXDiploUpdated.add(newPeriodDoc);
                    }
```

```
                    if (periodDoc.containsKey("1st Half")) {
                        ArrayList<Document> fullTimeProvidersList = (ArrayList<Document>) periodDoc.get("1st Half");
                        Document newPeriodDoc = new Document("1st Half",fullTimeProvidersList);
                        EnaXDiploUpdated.add(newPeriodDoc);
                    }


                    if (periodDoc.containsKey("2nd Half")) {
                        ArrayList<Document> fullTimeProvidersList = (ArrayList<Document>) periodDoc.get("2nd Half");
                        Document newPeriodDoc = new Document("2nd Half",fullTimeProvidersList);
                        EnaXDiploUpdated.add(newPeriodDoc);
                    }

                    currentMatch.remove( key: "1X2");
                    currentMatch.append("1X2",EnaXDiploUpdated );


                }
            }//emd of period for loop
        }catch (Exception e){
            System.out.println("Could not turn Full Time to FullTime");
        }

        System.out.println(currentMatch.toJson());

        org.bson.Document update = new org.bson.Document();
        update.append("$set", currentMatch);

        collection.updateOne(query, update, options);
        Timestamp timestamp = new Timestamp(System.currentTimeMillis());
        System.out.println(uniqueID + " was updated at: " + DeclarationsAndConstants.sdf.format(timestamp));




    }catch (Exception e){
        //could not update the document so lets replace it :
        System.out.println(e + "\nCould not insert or update the current match..");
    }
}
```

```java
static ArrayList<String> getTournaments (MongoDatabase DB, String mongoDBCollectionName, String country){

    ArrayList<String> tournaments = new ArrayList<>();
    //Connect to a collection
    MongoCollection<Document> collection = DB.getCollection(mongoDBCollectionName);
    FindIterable<org.bson.Document> findIterable;
    findIterable = collection.find(Filters.eq( fieldName: "country", country));

    for (Document doc : findIterable) {
        //check if this specific match ended with the home team or the away team winning
        if (doc.containsKey("league")) {
            if (!doc.getString( key: "league").isEmpty() && !tournaments.contains(doc.getString( key: "league"))) {
                tournaments.add(doc.getString( key: "league"));
            }
        }
    }

    return tournaments;
}
//endregion
```

```java
public class OddsPortal {

    private ChromeDriver oddsPortalDriver;
    private MongoDatabase oddsPortalDB;
    private MongoClient oddsPortalClient;
    public static boolean shouldRun = true;
    List<HashMap> allCollectedSports = new ArrayList<>();

    public static void main(String[] args) {

        //create a class instance
        OddsPortal oddsPortal = new OddsPortal();
        //Driver code
        //create the driver object.
        //Set Up Mongo DB
        oddsPortal.oddsPortalClient = MongoDBFunctions.setUpMongoClient();
        oddsPortal.oddsPortalDB = MongoDBFunctions.setUpMongoDB( DatabaseName: "oddsPortalDB", oddsPortal.oddsPortalClient);
        //set up the webdriver
        oddsPortal.oddsPortalDriver = WebDriverFunctions.setUpWebDriver();

        while (shouldRun) {

            //Now empty the all collected match list before repeating the actions of collecting the mathces
            oddsPortal.emptyHashMapList(oddsPortal.allCollectedSports);
            //call this function to get the user's input choice for which sport to gather data for
            oddsPortal.getUserInputForSportChoice();
            oddsPortal.goToTournamentGameResults();

        }

        //Outside of the loop run
        //Terminate all active connections and call System Exit (0)
        HelperFunctions.terminateAndCloseConnections(oddsPortal.oddsPortalClient, oddsPortal.oddsPortalDriver);


    }
```

```
//region User's Input

private void getUserInputForSportChoice() {...}

private void getSportCountriesAndTournaments(WebElement sport) {...}

private List<WebElement> getInputAndSpecificCountry(List<WebElement> countries) {...}

private List<WebElement> getInputAndSpecificTournament(List<WebElement> tournaments) {...}

private List<WebElement> getInputAndSpecificTournamentYearRange(List<WebElement> years) {...}

//endregion
```

```
private void goToTournamentGameResults() {....}

//endregion

//region Secondary Classes
public static void runTerminationClass () {
    //Start the second class which will delete all the inactive matches from the Mongo DB
    OddsPortalTerminator oddsPortalTerminator = new OddsPortalTerminator();
    Thread t = new Thread(oddsPortalTerminator);
    t.start();
}

//endregion

//region Clear Arrays And Lists
private void emptyHashMapList(List<HashMap> list) { list.clear(); }
//endregion
```

```
private void getGameInfo(String href, String sportName, String countryName, String leagueName, String yearRange) {

    try {

        //press the year range button and load the page
        //open the href in another tab
        oddsPortalDriver.get(href);

        WebElement waitElementTemp = oddsPortalDriver.findElement(By.id("col-left"));
        WebDriverFunctions.waitForElementVisibility(oddsPortalDriver, waitElementTemp);

    /*
    We also need to check if there are multiple pages available for the specific year range option and
    iterate through as to gather all of the available data
    #We will start from page 1, repeat for 15 - 20 pages and if there are no more data we will break numberOfPages =
    np.arange(1, 50, 1)

    */
```

```
String urlPage = href + "#/page/" + page_number + "/";
oddsPortalDriver.get(urlPage);

WebElement waitElementTemp1 = oddsPortalDriver.findElement(By.id("tournamentTable"));
WebDriverFunctions.waitForElementVisibility(oddsPortalDriver, waitElementTemp1);

//JSOUP & Selenium
// convert page to generated HTML and convert to document
Document doc = Jsoup.parse(oddsPortalDriver.getPageSource());

//Now we gather the data
//Now gather all the games shown in the live sections with their attributes(teams - time - score - odds - values)
Element tournamentTable = doc.selectFirst("div#tournamentTable");
Element table = tournamentTable.selectFirst("table#tournamentTable");

//Check now whether there is the no data available displayed on the table which means that the page is empty,
//Thus either there is no data available or the page range is not valid
Element noDataAvailable = table.selectFirst("td#emptyMsg");

if (noDataAvailable == null) {

    //This means that the no data available box was not found so we proceed with gathering the games from thetable
    Elements tableRows = table.select( cssQuery: "tr");

    //Create a list which will hold all the available matches shown on the table, their href so we can visit them one by one
    List<String> tableMatchesHREFLinks = new ArrayList<>();
```

```
for (Element row : tableRows) {

    if (row.selectFirst("td.table-participant") != null) {

        Element link = row.selectFirst("a");

        if (link != null) {
            //We got the link of the match, thus we can now visit the match page history and gather all of it 's data
            String refinedMatchUrl = StringsProcessing.replaceInPlayOddsMatchUrl(DeclarationsAndConstants.oddsPortalMatchUrlPrefix + link.attr( attributeKey: "href"));
            tableMatchesHREFLinks.add(refinedMatchUrl);
            //tableMatchesHREFLinks.add(DeclarationsAndConstants.oddsPortalMatchUrlPrefix + link.attr("href"));
        }
    }
}//end of for loop for each row
```

```
                for (String matchLink : tableMatchesHREFLinks) {

                    //System.out.println("THE COLLECTED MATCH LINK IS " + matchUrlLink);
                    //We got the link of the match, thus we can now visit the match page history and gather all of it 's data
                    //Create a current match with it's identifier being the matchLink (Unique)
                    org.bson.Document currentMatch = new org.bson.Document("_id", matchLink);

                    currentMatch.append("sport", sportName);
                    //First assign the sport and the league to the match:
                    currentMatch.append("League", leagueName);
                    //Give the current match it 's country:
                    currentMatch.append("country", countryName);
                    //Give the match a year Range
                    currentMatch.append("yearRange", yearRange);
                    //Call this method to get all the info and return the current match
                    currentMatch = visitMatchesAndGatherData(matchLink, currentMatch);

                    if (currentMatch != null) {
                        collectedMatchesForSport++;
                        //Final step is to insert the match into the mongo DB
                        //here we will check if we are validating the match, if not we add the match to our collection
                        MongoDBFunctions.insertMatchToMongo(sportName, currentMatch, matchLink, oddsPortalDB);
                    }
                }

                //When done Close the tab with match Url to not waste ram
                oddsPortalDriver.close();
                //Switch back to the first tab with URL A
                oddsPortalDriver.switchTo().window(tabs.get(0));
```

```
private org.bson.Document visitMatchesAndGatherData(String matchLink, org.bson.Document currentMatch){...}

private void getMatchHeaderInfo(org.bson.Document currentMatch) {...}

private void getMatchOdds(org.bson.Document currentMatch){...}

private void retrieveSubBetTypes (List<WebElement> activeSubBetTypes, String mainBetText, org.bson.Document currentMatch) {...}

private List<org.bson.Document> retrieveAndExpandBetTypeTables( String subBetString) {...}

private List<org.bson.Document> retrieveOddsFromTable (Element oddTable ) {...}
```

```
public static String noviBetMatchUrlPrefix = "https://www.novibet.gr/en/live-betting";
public static String oddsPortalUrl = "https://www.oddsportal.com";
public static String oddsPortalMatchUrlPrefix = "https://www.oddsportal.com";
//To collectAll Available Bets
public static boolean oddsPortalShouldGatherAllBetTypes = false, oddsPortalShouldGatherAllSubBetTypes = false;
//Arrays.asList("1X2", "Home/Away", "Over/Under");
public static List<String> oddsPortalBetTypesToCollect = Collections.singletonList("1X2");
public static List<String> oddsPortalSubBetTypesToCollect = Collections.singletonList("Full Time");

public static int webElementsDelay = 15;
public static int sleepTime = 4;
public static int smallDelay = 2;
public static int maxPageLimit = 100;
public static int maxPageLimitForUpcomingMatches = 2;
```

```
    public static void pauseForSeconds (int seconds) {
        try {
            Thread.sleep( l: seconds * 1000);//Sample: Thread.sleep(1000); 1 second sleep
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
    }

    public static void terminateAndCloseConnections(MongoClient client, ChromeDriver driver) {
        //Close the driver and the mongo DB and exit app
        //Finally close or disconnect the driver
        WebDriverFunctions.driverDisconnect(driver);
        MongoDBFunctions.closeMongoClient(client);
        System.exit( status: 0);
    }
```

```java
public class SystemAwakeRegulator {

    public static void main(String[] args) throws Exception {
        Robot robot = new Robot();
        Random random = new Random();
        // this code will keep on moving mouse pointer to a random location every minute
        while (true) {
            robot.delay( ms: 1000 * 60);
            int x = random.nextInt() % 640;
            int y = random.nextInt() % 480;
            robot.mouseMove(x, y);

        }
    }

}
```

```java
public static String cleanUpString(String stringToClean) {...}

public static List<Integer> splitInputIntoIntegerList(String inputString){...}

public static String replaceInPlayOddsMatchUrl(String url) {...}

public static String cleanUpScoreString(String stringToStrip) { return stringToStrip.replaceAll( regex: "[^\\d-:]",  replacement: ""); }

public static String[] splitStringBy(String stringToSplit, String splitter) {...}

public static String filteredStringRemoveParentheses (String toProcess) {...}

public static double stringToDouble (String toConvert){...}

public static int stringToInt (String toConvert){...}

public static String stringToMongoDBField (String toProcess){...}
```

Bibliography

Apostolou, Konstantinos & Tjortjis, Christos. (2019). Sports Analytics algorithms for performance prediction. 1-4. 10.1109/IISA.2019.8900754.

Astous, A, and M Gaspero. “Heuristic and Analytic Processing in Online Sports Betting.” *Journal of Gambling Studies*, vol. 31, no. 2, 2015, pp. 455–470.

Bar-Eli, Michael, et al. “PREDICTIONS AND BETTING” *Judgment, Decision-Making and Success in Sport*. Wiley, 2011.

Cantinotti M, et al. “Sports Betting: Can Gamblers Beat Randomness?” *Psychology of Addictive Behaviors: Journal of the Society of Psychologists in Addictive Behaviors*, vol. 18, no. 2, 2004, pp. 143–7.

D. Prasitio, D. Harlili, Predicting football match results with logistic regression, in: Proceedings of the 2016 International Conference on Advanced Informatics: Concepts, Theory and Application (ICAICTA), 16–19 Aug. 2016, Penang, Malaysia, 2016.

Gainsbury S.M, and Russell A. “Betting Patterns for Sports and Races: A Longitudinal Analysis of Online Wagering in Australia.” Journal of Gambling Studies, vol. 31, no. 1, 2015, pp. 17–32., doi:10.1007/s10899-013-9415-4.=OVIC&xid=3b366a56.

Glanz, James, and Agustin Armendariz. “Legal Sports Betting Intensifies a Cat-And-Mouse Data Game.” New York Times, vol. 167, no. 58012, 2018.

Hubáček Ondřej, et al. “Exploiting Sports-Betting Market Using Machine Learning.” *International Journal of Forecasting*, vol. 35, no. 2, 2019, pp. 783–796., doi:10.1016/j.ijforecast.2019.01.001.

I.H. Witten, E. Frank, M.A. Hall Data Mining: Practical Machine Learning Tools and Techniques Morgan Kaufmann, Burlington, MA (2011)

James, Richard J. E, et al. “Understanding the Psychology of Mobile Gambling: A Behavioural Synthesis.” *British Journal of Psychology*, vol. 108, no. 3, 2017, pp. 608–625., doi:10.1111/bjop.12226.

Jsoup Java HTML Parser, with the Best of HTML5 DOM Methods and CSS Selectors., 2021, https://www.jsoup.org/.

Levitt, 2004 Levitt S.D. Why are gambling markets organized so differently from financial markets? The Economic Journal, 114 (495) (2004), pp. 223-246

Lopez-Gonzalez H, and Griffiths M.D. “Understanding the Convergence of Markets in Online Sports Betting.” International Review for the Sociology of Sport, vol. 53, no. 7, 2018, pp. 807–823., doi:10.1177/1012690216680602.

Mathworks (n.d.). machinelearning_supervisedunsupervised.png. Retrieved from https://de.mathworks.com/help/stats/machinelearning_supervisedunsupervised.png.

MongoDB “The Most Popular Database for Modern Apps.”, 2021, www.mongodb.com/.

Salian, Isha. “NVIDIA Blog: Supervised Vs. Unsupervised Learning.” The Official NVIDIA Website, 2 Aug. 2018, www.blogs.nvidia.com/blog/2018/08/02/supervised-unsupervised-learning/.

Schuessler, Olena, et al. Parallel Training of Artificial Neural Networks Using Multithreaded and Multicore Cpus, 2011, pp. 70–79.

Sportradar “Sportradar's Clients - over 800 Companies Worldwide.”, 2021, www.sportradar.com/about-us/our-clients/.

Winston, Wayne L. “*Freakonomics* meets the bookmaker.” *Mathletics: How Gamblers, Managers, and Sports Enthusiasts Use Mathematics in Baseball, Basketball, and Football: With a New Epilogue by the Author*. Princeton University Press, 2012. 262-265. ProQuest Ebook Central, http://ebookcentral.proquest.com/lib/acggr/detail.action?docID=832770.