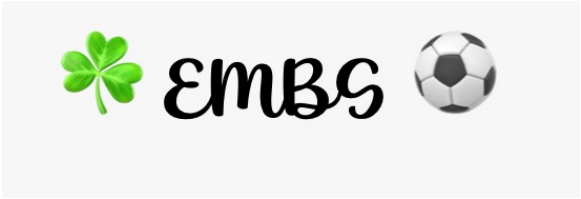

Easy Money Betting Soccer

Fabián Orduña[*]
Jorge García[†]
Miguel Lerdo[‡]
Nelson Gil[§]

May 16th, 2022

Contents

[*]fabian.orduna@itam.mx
[†]jgarc401@itam.mx
[‡]miguel.lerdodetejada@itam.mx
[§]nelsongilvargas@gmail.com

Problem Definition

Soccer is the most popular sport globally, gathering 3.5 billion fans, followed by cricket with 2.5 billion fans [1]. Many of these fans like to be aware of the odds for future games of their teams, especially with upcoming football classics like FC Barcelona vs Real Madrid, Manchester United vs Liverpool or America vs Chivas in the Mexican league. Another proportion of these followers are looking for betting tips. This data product objective is to predict football match results (win, loss or draw) of the Premier League teams. The prediction could be used for informative means or as betting tips.

Traditional bets are done based on odds that bookmakers publish. Another approach of predictions is based on machine learning models, as we do in this project. We propose a model based on accumulated goals by teams until their previous game and their local playing status. Among the compared classification models are: logistic regression, KNN, support vector machine, random forest, gradient boosting and perceptron. Data is updated and the model is retrained weekly, picking the best performing one.

System Design

The system design is easy to understand and is efficient. The different steps of the system were chosen and developed to guarantee the best end-to-end process.

Getting information. The information is getting from API-FOOTBAL, the API was found in RapidApi where has a lot of popularity. The API has little latency, and the information is constantly updating, in some cases the information is updated by day and even by hours.

PostgreSQL. Is a powerful, open source object-relational database system that is known for reliability, feature robustness, and performance. PostgreSQL is becoming the preferred database for more and more enterprises. It is currently ranked #4 in popularity amongst hundreds of databases worldwide according to the DB-Engines Ranking [2]. For this reason, we decide to use PostgreSQL as a data base storage.

In our project, PostgreSQL is used to storage the last information necessary to retain the model. On the other hand, this tool keeps the results of a prediction that is used to compare the performance of our model each week when the journey was closed and understand how pretty good are our predictions.

Google Cloud Storage. Everything that we generate is on google cloud storage. The data, the models, the scripts that we use to do the setup of the project are in a bucket on google cloud.

Airflow. The leading open source tool used today for scheduling, orchestrating and monitoring workflow [3]. Airflow updates the information, retrain and delivery a new model on a google cloud storage.

A brief explanation of the workflow:

- Each new week and airflow DAG ask for a request to update the PostgreSQL data table with the new information related to the result of the last available journey.
- After update, at the next day whit the last information an airflow DAG takes the new information and retrain five different models. The process compares the performance of each one and picks the best model for the week.
- The model was overwriting in a cloud storage. This model will use for the next journey to predict the results.
- The predictions are made on the batch process with a DAG, but on the other hand we create a tool to review predictions via online with the las updated model.

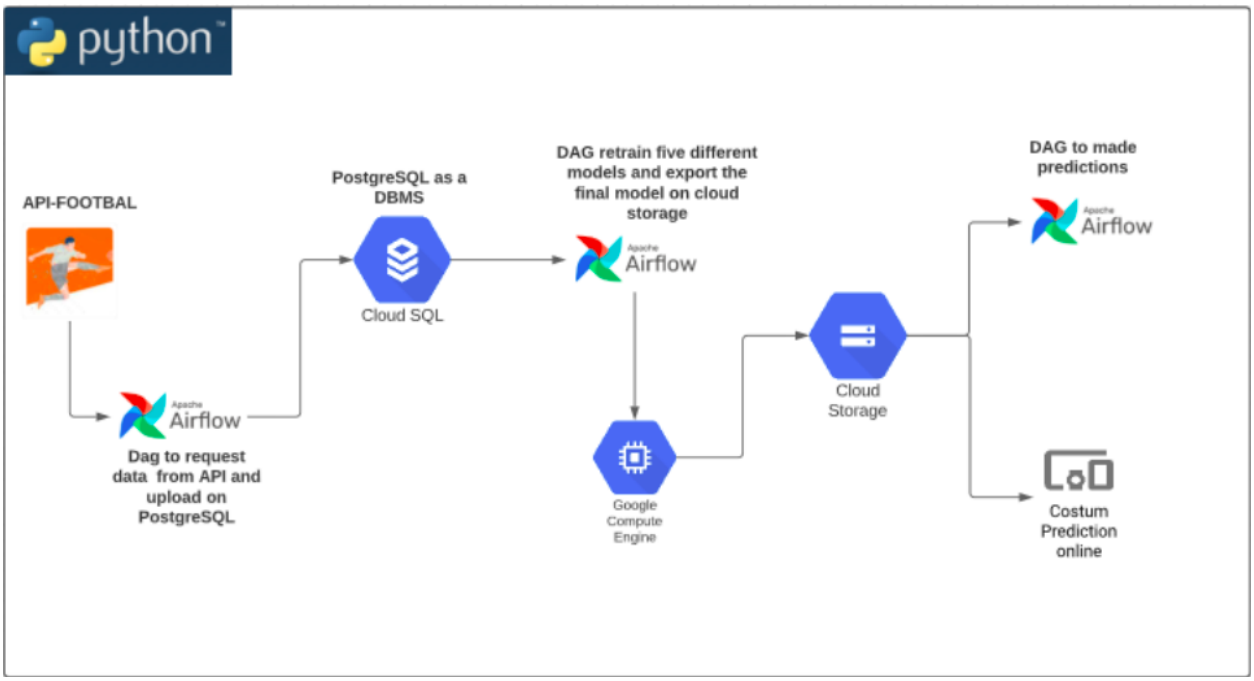

Figure 1: The workflow diagram.

Data Source, Ingestion and Feature Engineering

Based on the problem scope, we're focusing on the Premier League. The source of data for our model is the Soccer API. This provides different endpoint which includes matches, players, stats, and more information related to the soccer world.

We retrieve, each week, the different matches results and we store the data in Google PostgreSQL. We get that information because it can be used to generate historical data which provides important information when the modeling process ocurr, and of course since we want to predict the results, we needed to incorporate somehow data with that info.

After the information is stored in the database (this happens to keep track of the different seasons matches results) the transformation of the data starts, the feature engineering process. The model uses the team names, local and visit, it also uses the acumulated scored, received and the difference of golas before the prediction happens and, it also uses the number matches that both teams played before the prediction is needed. To achieve this, a sorting by match date is required and then, for each new match, the addition of the scored, received is done, and finally the difference between both is performed.

Machine Learning Development

Since we want to provide an effective mechanism that can help the customers to take their decisions with more confidence, we developed a model to have success.

The process involves using the transformed data by the feature engineering.

To enable a production model, we went through different scenarios, different tests and applied our knowledge to provide something with quality.

At the beginning, we centered our efforts in learning about the different approaches and the state of the art for this kind of problems. That let us gain more clarity and we confirmed and discarded previous thoughts.

One of the main paths we discovered was documented in Dixon and Coles, 1997 [4] and it says that a Poisson bivariate regression with score matches could be a good approach; however, there were not good results related to these models and we decided, after comparing with following models' performance, that we should drop those solutions.

To develop a good solution, the use of different machine learning algorithms was needed.

First, we tried with a logistic regression algorithm, but not only with one configuration, we proved 40 different configurations based on the hiperparams.

The use of the KNN model was involved, and we train different models changing different params such as the number of neighbors taken in count to make the decision, in total we tested 60 models with this algorithm.

We also tried with the SVM algorithm, and we tested 14 different configurations. With the random forest classifier we tested 324 models, 72 with a gradient boosting classifier and 7 with perceptron.

For the above algorithms and configurations, we build a pipeline to help with the flow of the training. During the pipeline process first, the initial categorical data is transformed, that means that the name of the teams is encoded with a OneHot encoding (also known as dummy classifiers).

After the transformation, the information is ready to be provided to the training process. For that, a Grid search is used and for each algorithm it required the different params that want to be tested in order to get the best model based on accuracy. The grid search also performs, for each set of parameters a cross validation at 80-20 and we picked the best model.

After we choose the best model for each algorithm, we compare the winners and get the global winner model. This last one, is the one that performs the predictions.

Model Evaluation

We believe that to evaluate the model performance, transparency is the key. Being this way with our clients is a core value for us. We save our predictions in the same table that stores the results, so we aim to print every week's predictions and after the games are played we will update such table with the actual results. This is the clearest way to inform our clients of our model's weekly performance. It is not needed a profound knowledge of statistics or concepts like 'accuracy' or 'recall' to judge whether the model did a good job. The client can compute a pseudo-accuracy metric –number of games predicted correctly divided by the total number of games played that weekend- just by logging in our webpage.

We also use some other, more technical, measures to test our model's predictive power. Our DAGs select the model which exhibits the lowest proportion of false positives. Accuracy is a secondary metric for us, since it can be a bit tricky to evaluate a model's performance based in it's accuracy. Also, false positives are the biggest concern in betting markets. For our business goals, a Support Vector Machine has been the best model so far but we might choose an alternative model if needed.

Our scripts also print a table with the distinct models that we trained and it's performance metrics. If requested, we can provide it to the interested clients together with a brief technical description of the metrics.

We face a time-cost tradeoff. Whereas training a model in VertexAI is easier and more efficient, it is costlier than writing a script that tries different models/hyperparameters. We plan to also include predictions from a VertexAI endpoint. If it vastly outperforms our hand-written model in the metrics most important to us, we might switch to a VertexAI trained model. We plan to add a column to our 'predictions' table which will feature the endpoint's predictions. This way, clients can know why we might charge more by our services.

Finally, how accurate our predictions are really depends of the odds of a soccer match. The closer the odds, the harder it is to predict a match 's result. We can also test our model's performance against the betting odds. The more accurate our predictions for close matches, the more we can trust our model. Depending

on performance, we may also include pre match odds as a predictive variable in a future version of the data product.

We decided to start with a simple model. To test it's limitations, we will also check feature importance and based on it, we will test new features to add complexity to the model. It is important that the new variables contribute with information that was not already used to predict. In those lines, an EDA with simple correlation matrixes will helps us to determine which variables can helps us reach better results. As always, we are open to comments and tips.

Model Serving

The model produces weekly predictions of the games that are scheduled to be played. Therefore, these are batch predictions that the airflow obtains automatically. These predictions are available in an interactive website in which the user can pick the games he or she is more interested. The website only shows the predictions for the matches that the user selected.

We are working with three Airflow scripts: one that does the ETL, a second that does the feature engineering and obtains batch predictions, and a third one that retrains the model. These three scripts are automated to be triggered weekly. The only manual components were the creation of the bucket, but this task is not required to be repeated.

We function as service providers. Our users are soccer fans interested in knowing future match results. We are responsible of maintaining and improving our product.

In case the dag fails, our endpoint will still be able to produce predictions with the latest endpoint. The problem is that the predictions will be done with outdated data, losing the forecast potential.

Reflection

A data product, nowadays, is not just a machine learning model in production. There are lot of different elements behind a data product. Starting with a data extraction process, till and endpoint or a system available for clients. We discovered that there are many processes needed to provide a high-quality solution and, according to the case of use, the solution can change.

To build a solution is required to generate an architecture according to the project needs. Every solution can be developed in-house, but also, the new and incredible technology can be used. It does not matter if it is an Amazon or Azure or Google solution, that depends on the team that is developing the solution. In this case, we worked with Google Cloud Platform (GCP) since that was the suggested tool during this course.

First of all, the checkpoints allowed us to take advantage of the tools offered in Google Cloud Plattform. Each new checkpoint was useful to build a complete application. The team's dynamic was great, each of us has a different profile and everyone contributed an important part into the project.

Using GCP was a smart decision, it is state-of-the-art architecture, has a lot of functionalities that we use and many others that we would like to explore in the near future. At the end we really felt comfortable working in the cloud, which we believe was the primary goal of the class.

We present a vanilla model, in the sense that we didn't exploit all the features of the Football API. We explored other variables that have more information and could help to scale up our predictive power. For example, following [5] we found that using the pre match betting odds as predictive variables further increases our forecasting power. Also, player's ratings were strongly correlated with a team's statistics. In the future, we aim to include a combination of the aforementioned variables that captures the most information available.

Finally, if we have more resources, we would like to work in a complete online implementation in order to sell "bet services" that is the initial purpose of this project.

Broader Impacts

First of all, in order to avoid facing responsibility for our client's loses, we added a disclaimer indicating that the information we provide does not represent financial advice. Then, we are not subject of the SEC's regulations regarding financial advisors and underperforming clients can't sue us for their bad betting decisions.

It could also be the case that certain users bet more of what they can afford. We indicate that clients that feel their relationship with betting isn't healthy should seek out for help. Certain organizations like GamblersAnonymus help people to regain control of their lives and recover from gambling addiction. As an entertainment company, we do not condemn gambling. Next we present an extract of the disclaimer:

While we discuss betting on our website and offer suggestions, we do this entirely for editorial purposes and hobby enthusiasm. We do not conduct or promote gambling. Nor do we claim to provide any knowledge as to the local laws regarding sports betting in the jurisdiction of our readers. It is the responsibility of all visitors to our website or subscribers to our services to check local laws in their own area before placing bets.

We are an entertainment company that provides information for recreational purposes only. Readers and subscribers should understand any and all responsibility for betting decisions rests with them. We do not condone gambling. If you believe you have a problem with gambling, we highly encourage you to seek support from organizations such as GamblersAnoymous.

EMBS can not be held responsible for the decisions of our readers or subscribers when off our site. Our sports betting information is for editorial purposes only. By visiting our website you confirm that you do not hold EMBS responsible for your gambling decisions or any losses that may follow from that personal decision and activity when off our site.

Every data needed to access our virtual machines in the cloud is secure. Hackers could be trying to access our resources for illegal or free riding activities, e. g. mining crypto. Our credentials and bucket names are not available in our Github repo.

References

- 1, Soccer fans.
- 2, What is PostgreSQL?
- 3, Airflow orchestration.
- 4, Dixon, M. J., & Coles, S. G. (1997). Modelling Association Football Scores and Inefficiencies in the Football Betting Market. Journal of the Royal Statistical Society. Series C (Applied Statistics), 46(2), 265–280.
- 5, Markwick, D. Predicting Goals Using the Winning Odd, Mar 7, 2022.