FINAL MASTER THESIS

MASTER IN DATA SCIENCE

SPANISH LA LIGA PREDICTION

Author: PABLO FERNÁNDEZ MATUS

Madrid, JULY 2021

INDEX

INTRODUCING THE PROJECT

Here starts the final project of the Master in Data Science at KSCHOOL, done by Pablo Fernández Matus. The name of this project is *Spanish La Liga Prediction* and it is realized during the first half of year 2021.

During this project work will be carried out with data related to soccer matches results, from which will be extracted relevant information in order to build a prediction model. The type of model, and what it will consist on are decisions that would be made as the content of the data being worked on becomes better understood.

The project is divided into different notebooks following the steps taken for the data processing, algorithm execution and data analysis. This final step is performed via Tableau dashboards operating as the front-end of the project.

All the notebooks that make up this project can be found in the following repository, including the Tableau file that contains the Front-end.

https://github.com/PabloMatus6/Spanish-LaLiga-Round-Prediction

In terms of software or licenses required to run this program, it is sufficient to have access to Jupyter Notebook for the code. To view the front-end in Tableau, you will need a Tableau account or request the free trial offered by Tableau desktop.

DATA INFORMATION

The data to be worked with belong to the BeSoccer API.

This project focuses on the top two categories of Spanish football, which are called La Liga Santander and Liga Smart Bank, first and second respectively. In this project we are going to deep into the results that have been seen in these two leagues in the last 6 years, it was considered that it would be helpful to have older seasons but currently the API only allows requests for the last six seasons. This factor will condition some of the decisions made during the project. Finally, these 6 seasons allow us to obtain a total of approximately 5000 rows.

The variables that are going to be handled within this dataset correspond to data relating to the results and statistics of the teams accumulated throughout a season. Work will be carried out with data obtained from two requests to the API, one of them will correspond to the results of the matches and the other to the information of the position in the table of a team as well as statistics accumulated so far in the league.

The meaning of each of the variables returned by these requests can be found in the API documentation on the API website. Although some of them are intuitive, a breakdown is attached below.

The output variables corresponding to the request for matches results:

[id] => Match identifier
[year] => Season
[group] => Group
[total_group] => Total groups
[round] => Match day
[home] => Home team
[visitor] => Visiting team
[league_id] => League identifier
[team1] => Home team identifier for the season of the match [team1] => Home team identifier for the match season
[team2] => Visiting team identifier for the season of the match [team2] => Visiting team identifier for the match season
[dteam1] => Unique identifier of the home team
[dteam2] => Unique identifier of the away team
[numc] => Number of comments in RF
[no_hour] => Info for RF
[local_abbr] => Local Team Short Name
[visitor_abbr] => Short name of visitor team
[competition_name] => Competition name
[playoffs] => With value 1 is a playoffs competition
[group_code] => Group
[coef] => Match importance coefficient
[local_shield] => Home team shield

[visitor_shield] => Visitor_shield] => Visitor shield
[extraTxt] => Extra information schedule] => Date and time of the match [date] => Match day
[date] => Match day
[hour] => Match hour
[minute] => Minute of the match
[local_goals] => Goals home team
[visitor_goals] => Away team goals
[result] => Result
[live_minute] => Minute of the match when it is live
[status] => -1: Not started, 0: Live, 1: Finished, 2: Postponed, 3: Overtime, 4: Penalty, 5: Halftime
[channels] => Array

```
[id] => Identificador del partido
[year] => Temporada
[group] => Grupo
[total_group] => Total grupos
[round] => Jornada
[local] => Equipo local
[visitor] => Equipo visitante
[league_id] => Identificador de liga
[team1] => Identificador del equipo local para la temporada del partido
[team2] => Identificador del equipo visitante para la temporada del partido
[dteam1] => Identificador único del equipo local
[dteam2] => Identificador único del equipo visitante
[numc] => Número de comentarios en RF
[no_hour] => Info para RF
[local_abbr] => Nombre corto del equipo local
[visitor_abbr] => Nombre corto del equipo visitante
[competition_name] => Nombre de la competición
[playoffs] => Con valor 1 es una competición de playoffs
[group_code] => Grupo
[coef] => Coeficiente de importancia del partido
[local_shield] => Escudo equipo local
[visitor_shield] => Escudo equipo visitante
[extraTxt] => Información extra
[schedule] => Fecha y hora del partido
[date] => Día del partido
[hour] => Hora del partido
[minute] => Minuto del partidos
[local_goals] => Goles equipo local
[visitor_goals] => Goles equipo visitante
[result] => Resultado
[live_minute] => Minuto del partido cuando está en directo
```

The output variables corresponding to the request for standings:

[id] => General equipment identifier

[group] => Group
[conference] => RF Info
[team] => Team name
[points] => Points
[wins] => Victories
[draws] => Draws
[losses => Losses
[shield] => Shield
[basealias] => Team Aliases
[gf] => Goals for
[ga] => Goals against
[avg] => Goal Difference
[mark] => Identifier for legend
[round] => Matchday
[pos] => Position
[form] => Result streak w=win, d=defeat, e=draw
[direction] => RF info

```
[id] => Identificador general del equipo
[group] => Grupo
[conference] => Info de RF
[team] => Nombre del equipo
[points] => Puntos
[wins] => Victorias
[draws] => Empates
[losses] => Derrotas
[shield] => Escudo
[basealias] => Alias del equipo
[gf] => Goles a favor
[ga] => Goles en contra
[avg] => Diferencia de goles
[mark] => Identificador para leyenda
[round] => Jornada
[pos] => Posición
[form] => Racha de resultados w=victoria, d=derrota, e=empate
[direction] => Info de RF
```

The aim of the work is not only to make a prediction of match results as reliable as possible, improving on the initial 33%, but also to draw conclusions that allow us to gauge the importance that playing at home may or may not have for a team. On the other hand, it should be noted, as it may influence or have influenced the data contained in the project, that during the last season, 2020-21, and the second half of the previous season 2019-20, there were no fans present in the stadiums in Spain due to the COVID-19 pandemic. The factor of public presence will also be taken into account when analysing results.

USER MANUAL

If the project is downloaded from GitHub, it is recommended to save all notebooks in the same folder in the local directory. It will also be necessary that the CSV files to be downloaded during the execution of the project are saved in the same folder. This will facilitate the execution of the code.

When this project was carried out, the requests made to the API contained match information and statistics for the seasons between 2015-16 and 2020-21.

However, when this project was being completed, the API removed the data from the 2015-2016 season, so the process of requesting data from the API cannot be done in the same way as it was done at the beginning of this project. Anyway, it is reflected how these requests were made in notebooks 01 and 02. What was done in these notebooks is replaced by a CSV data file that is indicated how to download at the beginning of notebook 03, but it is important that this file is stored locally in the same folder in which the notebook is saved. It has been decided to keep these notebooks because they are very useful to understand where the data comes from and the meaning of most of them.

Important: if you want to replicate this project with the same seasons, these notebooks (01 & 02) should not be executed, you should start directly from notebook 03.

If you want to execute this project with a different range of seasons, the code of notebooks 01 and 02 is perfectly valid, you will have to change in the requests and in the for loops the range of seasons that you want to request. For this case, a version of the project has been added (*All-in-one* version) which the code of all notebooks is merged into one, and which removes the 2015-16 season. (See section *All-in-one* version of this document).

Although it is recommended to compare the requests made on notebooks 01 and 02 with the All-in-one version in order to replicate the model by changing the year variable, a brief explanation will follow.

The value in the request that should be modified is the variable 'year'. However, despite having this name, it does not symbolize a calendar year but a season, specifically, the year represents the second part of the season to which it refers (check *Data information*). This other way, if this variable has as value the year 2016 the two in it correspond to the season 2015-16, in the same way it will work for other seasons.

If this project is being carried out years later it is important to know that this API only allows to make requests referring to the last 5 seasons from the moment the request is made, therefore, in this case, if you want to use the same seasons (15-16 to 20-21) you should go directly to notebook 03, otherwise you must take into account that request value ‘year’ must be adjusted accordingly. To specify the changes that would have to be made if the project is to be replicated with other seasons, this change must be made in all the code cells that contain information relating to the year variable in notebooks 01 and 02, and it will also be necessary to make this change in the plots that refer to or filter data

according to these variables in notebook 04, although in this case it would only be able to display the data correctly, not for the correct execution of the code. However, we recommend once again, in this case, to use the *All-in-one* version.

As already mentioned, in the case of wanting to replicate this project with the same data, the recommendation is to read notebooks 01 and 02 without executing them and start executing the code from notebook 03.

Once you have chosen one of the two possible ways to replicate the project, you should know that the notebooks are not independent, each notebook starts from the last point of the previous one, so it is important to start in order and run the notebooks one by one. As you run these notebooks, CSV files will be saved in the local folder in which the notebook is saved, so you will not have to make any changes to the code.

The GitHub repository also contains a folder, which contains two Tableau files. One of them is a front-end of this project and the second one is a small visualization of the prediction model and its final results corresponding to the end of this project. To run these files, it is not necessary to have the data as they are stored in the file itself.

If at any time there is a problem obtaining any of the files used in the project, in order to continue running the code, you can download any of them (including the data used for Tableau in .xls) from the following link:

https://drive.google.com/drive/folders/1ruyw5XHLtJQklH8xDZlydWsO9fkt8ITY?usp=sharing

In this document we will try to explain as well as possible the content of the data and the process they have gone throughout the project, however, in case you want to know more information about the content of the data or have any doubts about it, you can refer to the API documentation.

ALL-IN-ONE VERSION

An *all-in-one* version of the project is added, in which all notebooks are collected into one.

To run this version of the project you don't need to download any previous file, you just need to run the whole notebook, from start to finish.

This version solves the problem of the request to data of 2015-16 season with the API by removing this season from the request. The results are different from the original project, and this version lacks comments and conclusions about the results obtained (different results). The only purpose of adding this version to the repository is to facilitate the replicability of the project. Thanks to this notebook it is possible to appreciate the changes to be made with respect to the original version if the interval of seasons to be used is different.

NOTEBOOKS BRIEF SUMMARY

Although they should not be executed, we will comment on what has been done in notebooks 01 and 02. In the **first notebook, 01**, a request is made to return information about the results of the matches of round. To do this, we have to specify three variables, one referring to the league division, another referring to the season, and finally, we have to specify the round. It should be remembered that the leagues on which this project works are the first two Spanish professional football divisions. For this reason, a loop is established to make the request for these two leagues, their complete seasons, for the last 6 years.

The Spanish Second Division, La Liga Smart Bank, is made up of 22 teams instead of 20, which means that it has fout more rounds than the first one, La Liga Santander. These rounds were not added to the project because of the possibility of confusing the model. About the second request, also made in this notebook, it returns information regarding the position in the league table and accumulated statistics of a team for a specific round.

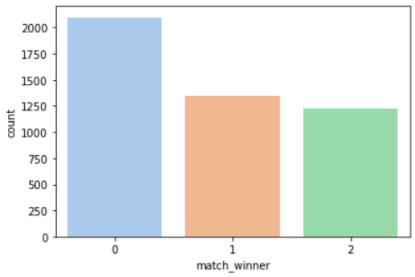
The second **notebook, 02**, after a small cleaning of the data, consists of performing a merge and joining the two requests of the first notebook in a single data frame. To do this, it is necessary to make a new request to the API and add two columns needed to create the keys that will be used to join both requests.

These first two notebooks correspond to the phase considered as data collection.

In the third **notebook, 03**, the execution of the code begins. First of all, as already mentioned, the necessary data must be downloaded from the link attached to Google Drive. Once saved, in the same local folder as the notebook, the complete notebook can be executed without any problem. In the notebook, the preparation of the data continues, including data cleaning prior to modelling and to plotting, in order to see its contents. The target variable to be predicted is also created and contains the result of the match, depending on whether the home or away team won, or the match ended in a draw.

In the **notebook 04**, the numerical content of the data is further elaborated and relevant information found in the data is plotted. To do this, the seaborn library is used. Some of the graphs and conclusions obtained are shown below.

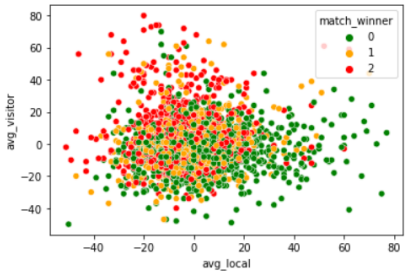
```
In [7]: sns.countplot(data = plotting_data, x="match_winner", palette='pastel')
Out[7]: <matplotlib.axes._subplots.AxesSubplot at 0x21db9024250>
```



As can be seen in both histograms(the content is the same), the most abundant category is Local win (0). There is not a big difference between the number of draws (1) and visitor wins (2), with the number of draws being slightly higher. Therefore, from this graph alone, it can already be seen that there is a tendency for the home team to win.

Another example:

```
In [10]: sns.scatterplot(data=plotting_data, x="avg_local", y="avg_visitor", hue="match_winner", palette=['green','orange','red'])
Out[10]: <matplotlib.axes._subplots.AxesSubplot at 0x21db93ad250>
```



It is observed that in addition to a large concentration of matches around an avg in an interval of (-20, +20) for both home and away, there is a certain dispersion of points for matches that pit a team with a very high avg against a very low avg. For these scattered points, the wins correspond to the team with a much higher avg. That is, for very high avg of the home team versus low avg of the away team, a majority of points correspond to a home win. Conversely, the same happens when the high avg corresponds to the visiting team and it faces a home team with a low avg, which means a very high percentage of wins for the visiting team.

It can be concluded from this that a large avg difference between two teams is decisive for the victory of the team with the higher average, regardless of whether they play at home or away. Although it should be mentioned that when this happens at home, the reliability is even higher, with a few defeats (small percentage) of the team with high avg when playing away.

In **notebook 05**, modelling begins. The idea is to make a first model testing with some classifiers, and see what results are obtained. The same process is applied to each of the classifiers that are tested at this point. A train test split is performed and then a scaling of the data. It is decided to do it in this order because if the scaling is done before the split, it could fall into the error of providing the test set with information related to the train one, that would give an advantage to the model. In addition, for each model we obtain a classification report with information on the type of failures and successes according to target groups within the categorical variable and also according to the metrics used, f1-score being the one that encompasses all this content and on which we will evaluate the models. A confusion matrix is also attached that indicates the false positives, false negatives and hits for each of the classes of the output variable.

In **notebook 06**, the aim is to improve the model by creating new variables, most of them obtained from variables previously deleted, and also from operations carried out with

variables already contained in the models that have been tested. Throughout this notebook, each of the new variables created is explained and the objective is to improve the results obtained in the models of the previous notebook.

During **notebook 07**, the models of notebook 05 are applied to the new data frame containing the new variables created in notebook 06. In addition, new classifiers are incorporated in order to have a wider range of results, the same process as in notebook 05 is repeated, and the results obtained, possible reasons, and conclusions are evaluated at the end.

Finally, in **notebook 08**, after having the results of the models, some checks are carried out with other classifiers that had not been used, the results are analysed and conclusions are obtained.

RESULTS

In addition to being commented on in the notebooks, we would like to comment in this document the results obtained, possible ways of improvement, as well as achievements.

Firstly, the variable we have worked on, and which we have tried to predict, is categorical and has three possible outcomes: home win, draw and visitor win. In principle, the probability of guessing the result of a match without taking into account any extra factor is one third, that is, a 33.33% chance of being right. We start from this value, which should not be difficult to improve. However, in addition to the result obtained by the model, the intention of this project is to extract conclusions from the data through plots made either in the notebook or via story in Tableau.

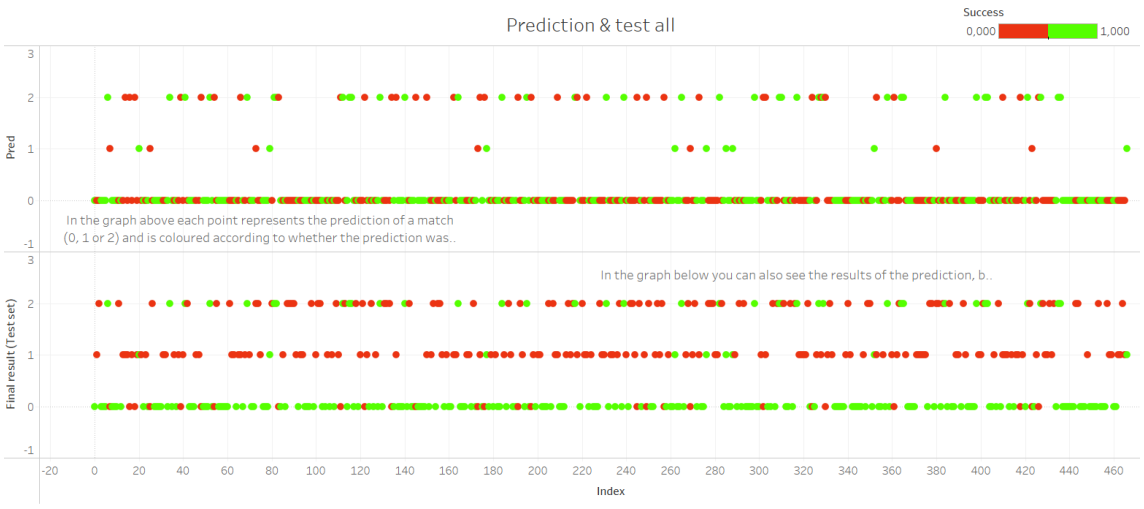
After making some initial models with the non-featured data, i.e. without creating new, more complex variables, the first conclusion is already drawn. The model that obtains the best overall results is the least sophisticated and obtains 51% accuracy in the prediction of the winning team of the match. On the other hand, the different results obtained for the 3 categories by each model are also studied, which makes it easier to draw conclusions about the difficulty of predicting each of them. It is also interesting to know what type of classifier we should use if we want to be more accurate in a specific category. Clearly, the easiest category to predict is the home win and the most complicated is the draw.

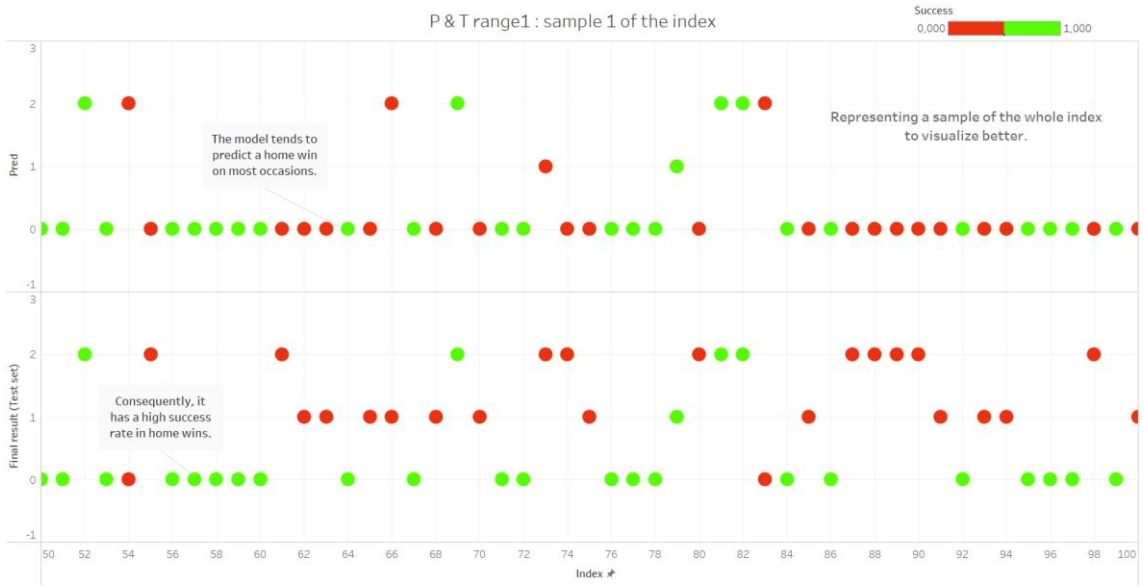
The fact that the simplest classifier is the one that obtains the best results can initially be considered simply a coincidence, since there is not a very big difference with respect to any other model, but it can also be caused by having a model with a limited data extension, which makes it difficult for the model to find complex patterns in the data. However, after the data feature engineering phase we will have a more sophisticated model, or at least its variables will be, and we will be able to see how it reacts to these classifiers and some others.

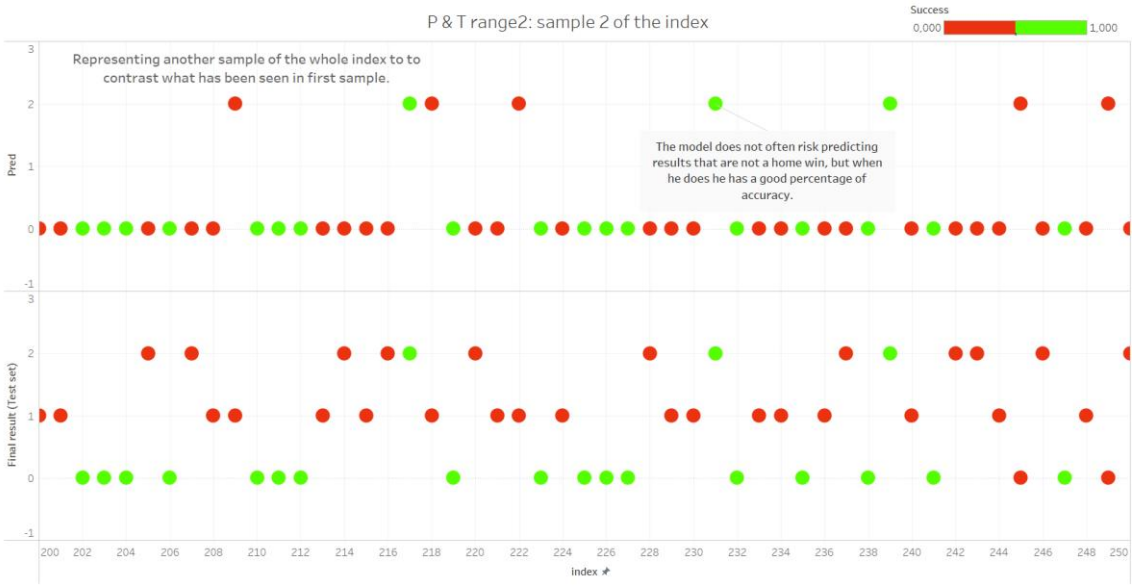
After featuring, we replicate what we have done with the unfeatured data. Some unused classifiers are also added at this point. The results obtained do not improve on those achieved in the first approach. In the search for answers to this event, the conclusion is reached that, given the limited data, an excess of information or information that is too precise may have been produced with the addition of new variables. This could mean that the model has received more information but in the form of noise, or simply that this information has caused overfitting, although this is more difficult to verify and even more so in the case of single categories that are so marked.

Accordingly, it is decided to apply the remaining classifiers to the first dataframe as well. In this case, none of them improves on Logistic Regression, which is the model with which we obtained the best results at the beginning and which is also less complex than the latter classifiers. This might confirm the suspicion that, since the database is not very large, it works better with simpler models. It has been approved throughout for both datasets by increasing and decreasing the test set in case it could be beneficial, and was not an influential factor in the prediction result.

Graphs representing the hits and misses of the model as a function of the prediction made, and as a function of what the final result was, have been obtained using Tableau. This is the information that was available in the confusion matrix, making it possible to better understand the results in terms of false positives and false negatives, but thanks to the visualization, facilitating their analysis. Two extracts from the total Index of the test set are attached and some conclusions drawn from these data are commented on.







On the other hand, also in Tableau, the content of the data itself is analyzed, beyond the model. This analysis allows interesting conclusions to be drawn, which are also shown below with some images. This visualization in Tableau is translated into a front end that also allows the user to interact with the data.