MASTER'S IN DATA SCIENCE

2021-2022

Final Master's Thesis

SPAIN'S MEDICINE ADMISSIONS APP

Author: Miguel Monedero Rubio

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Context summary

Historically, Medicine has always been one of the most demanded careers to study in Spain. Nowadays, it's still one of the hardest careers to get in, due to the high demand of people wanting to study this career in a public university (because of economic pricing, big difference in comparison to a private university, plus reputational reasons) and the very few available slots for students.

To be able to study this career in Spain, you need to achieve a certain average score in "Selectividad". "Selectividad" is the popular name given to the Spanish University Admission Tests ("Evaluación de Acceso a la Universidad", E.v.A.U.), a non-compulsory exam taken by students after secondary school. Students must take six 90-minute written exams over three days in June/July, consisting of common and specific subjects taken in "Bachillerato" (the last two non-compulsory years of secondary education). "Selectividad" exams are set by the Public Universities of each autonomous community and allow students access to the Spanish university system.

Once you do "Selectividad" the universities publish the score you need in order to enter their university, this score is a number ranging from 0 to 14. They publish these scores in the web pages corresponding to each university, meaning that applicants need to consult these scores in each of the university web pages, making it hard to have a holistic view. There are a few web pages that consolidate this information and put it in one single place to have centralized platform. However, these web pages show just a table containing the scores from the different universities and do not show any visualizations or filters in order to facilitate students consult this information. In addition, many students suffer uncertainty every year because of not knowing what the scores would be the next year (to see if they could finally be admitted in a university to study this career), therefore, one of the objectives will be to predict the next year scores (2022).

1.2 Objective

1.2.1 Objective of this project

There are two objectives for this project:

1) **Predict next year scores (2022):** data source has data ranging from 2010 to 2021.
2) **Create a centralized and accessible place, in this case it will be a web app**, to consult the different Medicine admission grades from all the universities in Spain, in a visual and helpful manner. This app will help people save time by not having to go to each of the universities web pages to consult the different scores. Also, it will help guide the user in order to make the decision that works best for them.

1.2.2 Objective of this document

The objective of this document is to explain the end-to-end process of creating this project, going into detail on the challenges found and the decisions made along the way.

1.3 Repository

The repository where you can find this project is:

- **https://github.com/MiguelMonederoRubio/Spain_Medicine_Admissions_App**

This repository has three different folders:

- **final_master_thesis:** folder in which this document is stored.
- **notebook:** jupyter notebook (Spain_Medicine_Admissions_App.ipynb) containing all the code from this project except for the front-end.
- **streamlit:** file (myapp.py) containing code corresponding to the front-end application built on streamlit.
    - To execute this file just run the following command on your terminal, entering your directory:
        - streamlit run "your directory" /myapp.py

Please see in figure 1 a diagram showcasing the workflow followed to build this project:

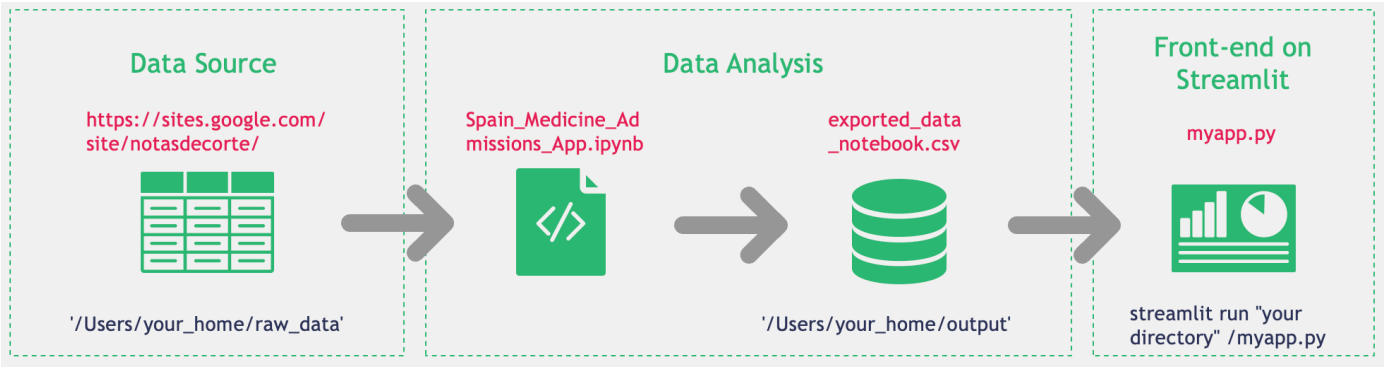


Figure 1: Workflow diagram

To execute the two coding files (Spain_Medicine_Admissions_App.ipynb and myapp.py), you would need to create two folders within your home directory:

1) "raw_data", being your directory '/Users/your_home/raw_data': here you would store the different csv files. Details of the data source will be explained on section 2 (data obtention). Enter highlighted directory in Spain_Medicine_Admissions_App.ipynb.
2) "output", being your directory '/Users/your_home/output': the output csv coming from Spain_Medicine_Admissions_App.ipynb will be stored here. Enter highlighted directory in myapp.py.

2. DATA OBTENTION

In this section we will be describing the source from which we got the data to do this project.

2.1 Web page

The source of data is coming from a web page called "**ACCESO A LAS FACULTADES DE MEDICINA DE ESPAÑA Y NOTAS DE CORTE"** that you can find in the following link:

- https://sites.google.com/site/notasdecorte/

The data we are using is found on the left side of the web page *("Notas de corte de medicina year/year+1")*, see figure 2:

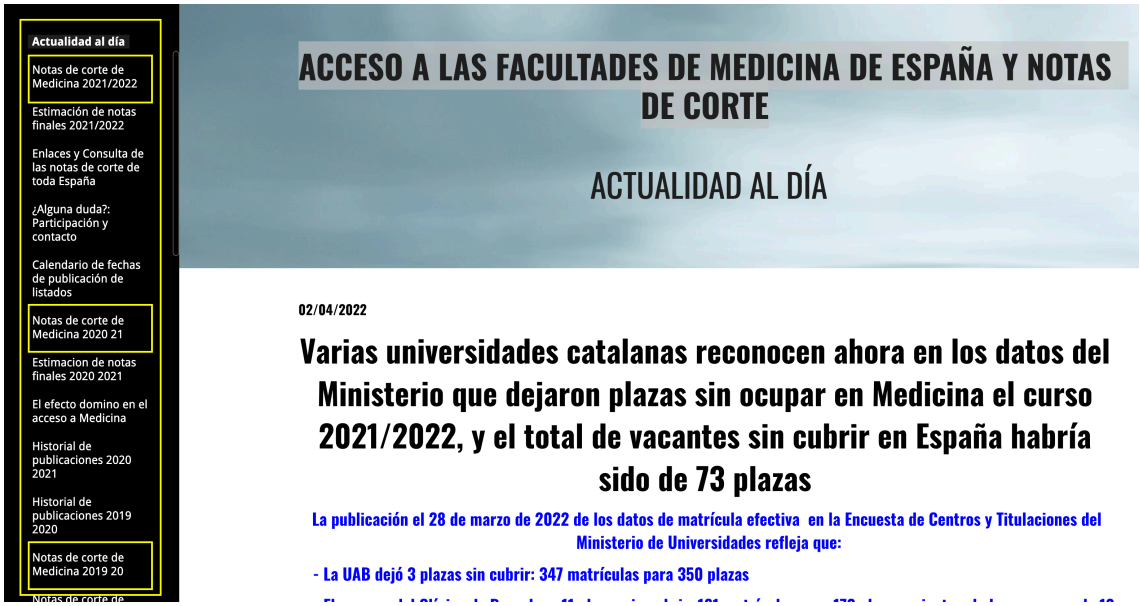


Figure 2: Web page from which data was taken

Data used for this project is going from the period of year 2010-2011 till 2021-2022, 12 years:

1) 2010/2011
2) 2011/2012
3) 2012/2013
4) 2013/2014
5) 2014/2015
6) 2015/2016
7) 2016/2017
8) 2017/2018
9) 2018/2019
10) 2019/2020
11) 2020/2021
12) 2021/2022

Each year has its own excel file. For simplicity, we have saved the 12 files in csv format. Example of data that can be found in year 2010-2011 can be seen in figure 3:

Notas de corte de Medicina	Última fecha de admisión (Fecha de la nota final)	Nota de corte final de Medicina	1ª Lista	Sobre 10	2ª Lista	Sobre 10	3ª Lista	Sobre 10	4ª Lista	Sobre 10	5ª Lista	Sobre 10
Universidad de Lérida	8/11/2010	11,04	11,654	8,324	11,296	8,069	11,249	8,035	11,21	8,007	11,04	7,886
Universidad de Gerona	8/11/2010	11,07	11,61	8,293	11,305	8,075	11,259	8,042	11,22	8,014	11,07	7,907
Universidad Rovira i Virgili (Reus)	8/11/2010	11,142	11,708	8,363	11,334	8,096	11,33	8,093	11,22	8,014	11,142	7,959
Universidad de Zaragoza - Campus de Huesca	7/12/2010	11,19	12,059	8,614	11,975	8,554	11,88	8,486	11,767	8,405	11,713	8,366
Universidad Autónoma de Barcelona	8/11/2010	11,195	11,59	8,279	11,34	8,1	11,304	8,074	11,304	8,074	11,195	7,996
Universidad de Valladolid	15/11/2010	11,204	12,29	8,779	11,996	8,569	11,768	8,406	11,504	8,217	11,484	8,203
Universidad de Cantabria (Santander)	29/10/2010	11,212	12,592	8,994	12,34	8,814	12,17	8,693	12,035	8,596	11,722	8,373
Universidad de Santiago de Compostela	19/10/2010	11,23	11,8	8,429	11,578	8,27	11,47	8,193	11,41	8,15	11,35	8,107
Universidad de Extremadura (Badajoz)	20/10/2010	11,236	12,5	8,929	12,21	8,721	12,05	8,607	11,975	8,554	11,332	8,094
Universidad de Cádiz	6/11/2010	11,286	12,12	8,657	11,975	8,554	11,757	8,398	11,552	8,251	11,469	8,192
Universidad Pompeu Fabra (Barcelona)	1/10/2010	11,328	11,976	8,554	11,536	8,24	11,5	8,214	11,328	8,091	11,328	8,091
Universidad de Barcelona	1/10/2010	11,35	12,04	8,6	11,631	8,308	11,524	8,231	11,35	8,107	11,35	8,107
Universidad de Salamanca	20/10/2010	11,37	12,564	8,974	12,23	8,736	12,106	8,647	11,672	8,337	11,595	8,282
Universidad de Zaragoza - Campus de Zaragoza	24/11/2010 (Actualizada 12/1/11)	11,39	12,185	8,704	11,966	8,547	11,9	8,5	11,824	8,446	11,791	8,422
Universidad de CLM- Campus de Ciudad Real	18/10/2010	11,478	12,343	8,816	11,971	8,551	11,885	8,489	11,478	8,199		
Universidad de Córdoba	6/11/2010	11,496	12,295	8,782	12,136	8,669	11,841	8,458	11,623	8,302	11,626	8,304
Universidad del País Vasco (Lejona)	24/07/2010	11,503	11,922	8,516	11,726	8,376	11,503	8,216	11,503	8,216	11,503	8,216
Universidad de CLM- Campus de Albacete	3/11/2010	11,53	12,246	8,747	12,082	8,63	12	8,571	11,53	8,236		
Universidad de Sevilla	23/10/2020	11,559	12,272	8,766	12,075	8,625	11,848	8,343	11,692	8,351	11,6	8,286
Universidad Miguel Hernández (San Juan de Alicante)	15/10/2010	11,584	12,216	8,726	11,972	8,551	11,584	8,274				
Universidad de Alcalá	3/11/2010	11,631	11,862	8,473	11,862	8,473	11,79	8,421	11,77	8,407	11,77	8,407
Universidad de Málaga	2/10/2010	11,645	12,225	8,732	12,05	8,607	11,84	8,457	11,73	8,379	11,66	8,329
Universidad de Murcia	3/11/2010	11,666	12,388	8,849	12,19	8,707	11,89	8,493	11,758	8,399	11,666	8,333
Universidad de La Laguna	Dato 11/10 13.00h	11,68	12,2	8,714	12,1	8,643	12	8,571	11,9	8,5	11,68	8,343
Universidad Rey Juan Carlos - Campus de Alcorcón	Dato 31/10 11.46h	11,7	11,867	8,476	11,807	8,433	11,7	8,357				
Universidad de Valencia	4/10/2010	11,732	12,247	8,748	12,049	8,606	11,992	8,566	11,936	8,526	11,804	8,431
Universidad Autónoma de Madrid	Dato 22/10 (Tablón de la facultad)	11,84	12,036	8,597	11,956	8,54	11,84	8,457				
Universidad de Las Palmas	29/09/2010	11,849	12,312	8,794	12,074	8,624	11,979	8,556	11,849	8,464		
Universidad de Granada	18/09/2010	11,959	12,503	8,931	12,392	8,851	12,095	8,639	11,959	8,542	11,959	8,542
Universidad Complutense de Madrid	Dato 2/11 (Envío de carta de admisión)	12,022	12,189	8,706	12,15	8,679	12,06	8,614				
Universidad de Oviedo	27/10/2010	12,088	12,579	8,985	12,418	8,87	12,293	8,781	12,243	8,745	12,199	8,714

Figure 3: Medicine admission scores for year 2010-2011

The columns shown above are the following:

1) **"Notas de corte de Medicina":** names of the different public universities.
2) **"Última fecha admisión (Fecha de la nota final)":** date in which the last score (we will call this score the final grade) is published.
3) **"Nota de corte final de Medicina":** last score published (final grade), meaning it's the lowest score in order to be admitted to that university. This score is ranging between 0-14.
4) **"1ª Lista":** first score published, therefore the highest score. This score is ranging between 0-14.
5) **"Sobre 10":** here they are showcasing the score corresponding to an Nª List but in range 0-10 instead of 0-14. Since this is rarely used, for the purpose of the project we will be analyzing the scores ranging between 0-14.

3. DATA QUALITY

In this section we will be describing the different data quality challenges we faced creating this project and how we solved them.

3.1 Data quality challenges

Since the data source is coming from 12 different excel files, we need to consider the following:

- **Column names containing scores:** make sure the columns containing the scores are consistent, and in case they are not, create a standardized column name. Also, make sure they are all a certain type of data, in this case, float.
- **University names**: make sure they are consistent throughout the different years, and in case they are not, create a standardized name.
- **Null values**: records that have null values across all the columns should be deleted.
- **Duplicated values:** make sure we do not have any duplicated values.

3.1.1 Column names containing scores

We have created 2 new columns called “1_list” and “final_grade” to have a standard name since there were consistency issues between the different data sources. Example:

The column “final_grade” has different names depending on the year. A few different names found that refer to the same data are:

- “Nota de corte final de Medicina”
- “Nota de corte final”
- “Nota de corte actual”

3.1.2 University names

We have taken the names of the last year (2021) as reference, and we realized 2 things:

1) **There are 13 universities which have a different name than the ones in 2021 data and need to be mapped**

   To solve this, we have used SequenceMatcher:

   - from difflib import SequenceMatcher

   SequenceMatcher can be used for comparing pairs of input sequences.

   On figure 4, you can see on the left-hand side, the 13 universities that have a different name than on the right-hand side (list of university names from 2021, that we are using as a reference). As you can see, it’s not 100% accurate but it is still helpful to map most of the names in a quick manner.

not_matched_uni	ratio_x	list_of_uni_names_2021	ratio_y
universidad autónoma de madrid	0.825397	universidad autónoma de barcelona	0.825397
universidad de barcelona	0.857143	universidad de barcelona-clínico	0.857143
universidad de cantabria (santander)	0.825397	u. de cantabria (santander)	0.825397
universidad de clm- campus de albacete	0.676056	universidad de las islas baleares	0.676056
universidad de clm- campus de ciudad real	0.648649	universidad complutense de madrid	0.648649
universidad de extremadura (badajoz)	0.825397	u. de extremadura (badajoz)	0.825397
universidad de jaume i	0.926829	universidad jaume i	0.926829
universidad de santiago de compostela	0.830769	u. de santiago de compostela	0.830769
universidad de zaragoza - campus de huesca	0.750000	u. de zaragoza (campus huesca)	0.75
universidad de zaragoza - campus de zaragoza	0.763158	u. de zaragoza (campus zaragoza)	0.763158
universidad miguel hernández (san juan de alic...	0.847826	u. miguel hernández (s. juan de alicante)	0.847826
universidad pompeu fabra (barcelona)	0.825397	u. pompeu fabra (barcelona)	0.825397
universidad rey juan carlos - campus de alcorcón	0.628571	urjc - campus alcorcón	0.628571

Figure 4: SequenceMatcher ratios

2) There are different number of universities per year

As you can see on figure 5, there are different number of universities per year. This is due to the fact that some universities have started teaching Medicine and have added this degree in their university program.

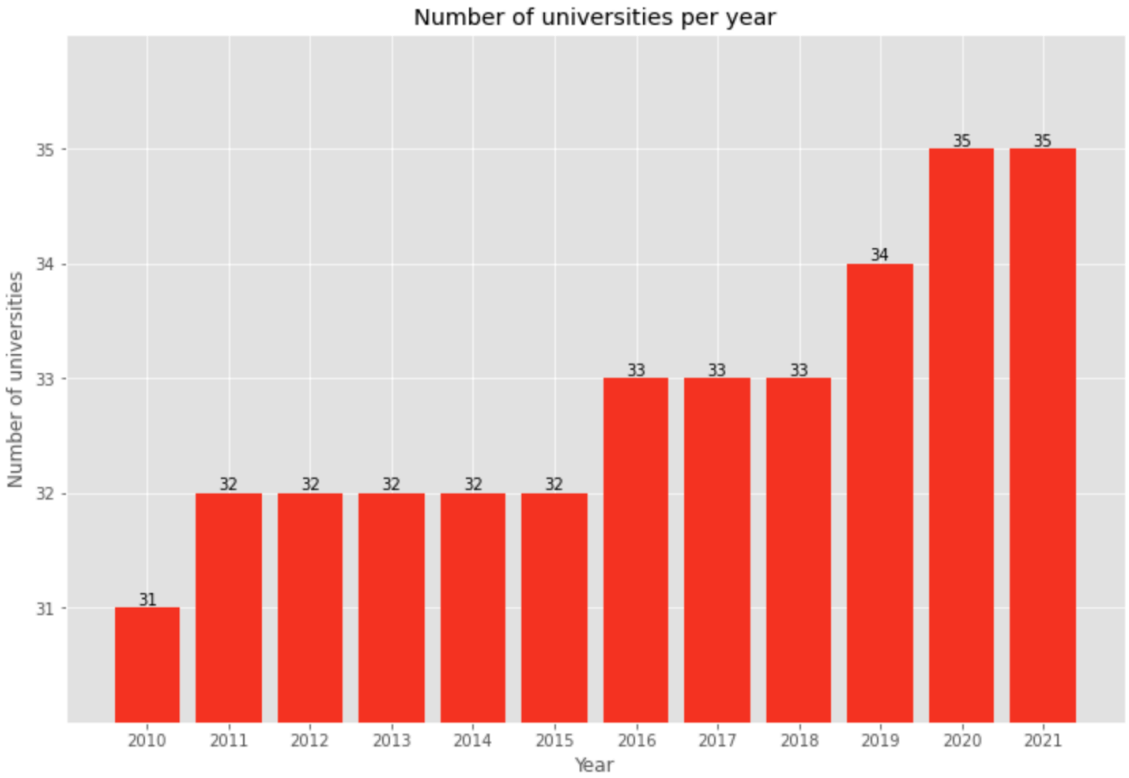


Figure 5: Number of universities per year

Analyzing the data, the names of the universities that have started teaching Medicine in the years above are:

- 2011: universidad jaume i
- 2016: universidad de las islas baleares
- 2019: universidad pública de navarra
- 2020: universidad de barcelona-bellvitge

3.1.3 Null values

We have done a quality check to make sure there aren't any nulls. Please see code on figure 6:

Check for nulls

```
# quality check: nulls

nulls_1_list = dfs_unified[dfs_unified['1_list'].isnull()].shape[0]
nulls_final_grade = dfs_unified[dfs_unified['final_grade'].isnull()].shape[0]
nulls_university = dfs_unified[dfs_unified['university'].isnull()].shape[0]

if nulls_1_list or nulls_final_grade or nulls_university != 0:
    sys.exit()
else:
    print('column 1_list has', nulls_1_list, 'null rows')
    print('column final_grade has', nulls_final_grade, 'null rows')
    print('column university has', nulls_university, 'null rows')
```

✓ 0.1s

```
column 1_list has 0 null rows
column final_grade has 0 null rows
column university has 0 null rows
```

Figure 6: Nulls quality check

3.1.4 Duplicated values

We have also done a quality check to make sure there aren't any duplicated values. Please see code on figure 7:

Check for duplicates

```
if dfs_unified[dfs_unified.duplicated() == True].shape[0] != 0:
    dfs_unified.drop_duplicates()
else:
    print('There are no duplicates')
```

✓ 0.5s

```
There are no duplicates
```

Figure 7: Duplicates quality check

4. FEATURE ENGINEERING

In this section we will be describing the different features that have been added in order to be considered in the correlation matrix described in section 5 (Modelling). Also, we will be defining the columns that we had already created before this section. The way we are going to do these 2 tasks is by creating a table schema of all the columns.

4.1 Table schema

Columns we previously had before this section:

- **year:** Indicates the year from which the data is. Values range from 2010 to 2021. In section 5 (Modelling) we will be predicting the scores for 2022 and will be adding them to the existing table, therefore values will range from 2010 to 2022
- **university:** String containing the university name
- **1_list:** Float containing the first score published. This is the highest score to be admitted at a university
- **final_grade:** Float containing the last score published. This is the lowest score to be admitted at a university

Columns added:

- **since_2010:** Indicates whether the university has been teaching the medicine career since 2010 or not. In case it does, value of this column is “True”. If not, the value will be the year from which that university has been teaching the career
- **city:** String that contains the city in Spain where the university is
- **CCAA:** String that contains Spain’s autonomous community corresponding to the university
- **diff_1_list:** Float that is the difference of the “1_list” column between the current year and the previous year. For year 2010, this column’s value is NaN since we don’t have data from the previous year (2009)
- **diff_final_grade:** Float that is the difference of the “final_grade” column between the current year and the previous year. For year 2010, this column’s value is NaN since we don’t have data from the previous year (2009)
- **diff_1_list_final_grade:** Float that is the difference between the “1_list” and “final_grade” columns for the current year
- **latitude:** Float containing the latitude corresponding to the location of the university. Values have been extracted from this web page: https://www.mapcoordinates.net/es
- **longitude:** Float containing the longitude corresponding to the location of the university. Values have been extracted from this web page: https://www.mapcoordinates.net/es

- **covid_format_change:** Due to COVID-19 the format of the tests from “Selectividad” changed, since they considered students weren’t as prepared as usual and had to make a change in the way the tests were done, to make it easier for the students. Before COVID-19, the exams had two options, option A or option B, you had to do all the questions that were on the option that you chose. In 2020, they changed this by letting the student choose the questions that they want to do, doesn’t matter if they are in option A or B. You can notice this difference in the “1_list” scores that went up an average of 0.408 in 2020 in comparison to 2019, when before 2020 this average score was of 0.07. Please see figure 8:

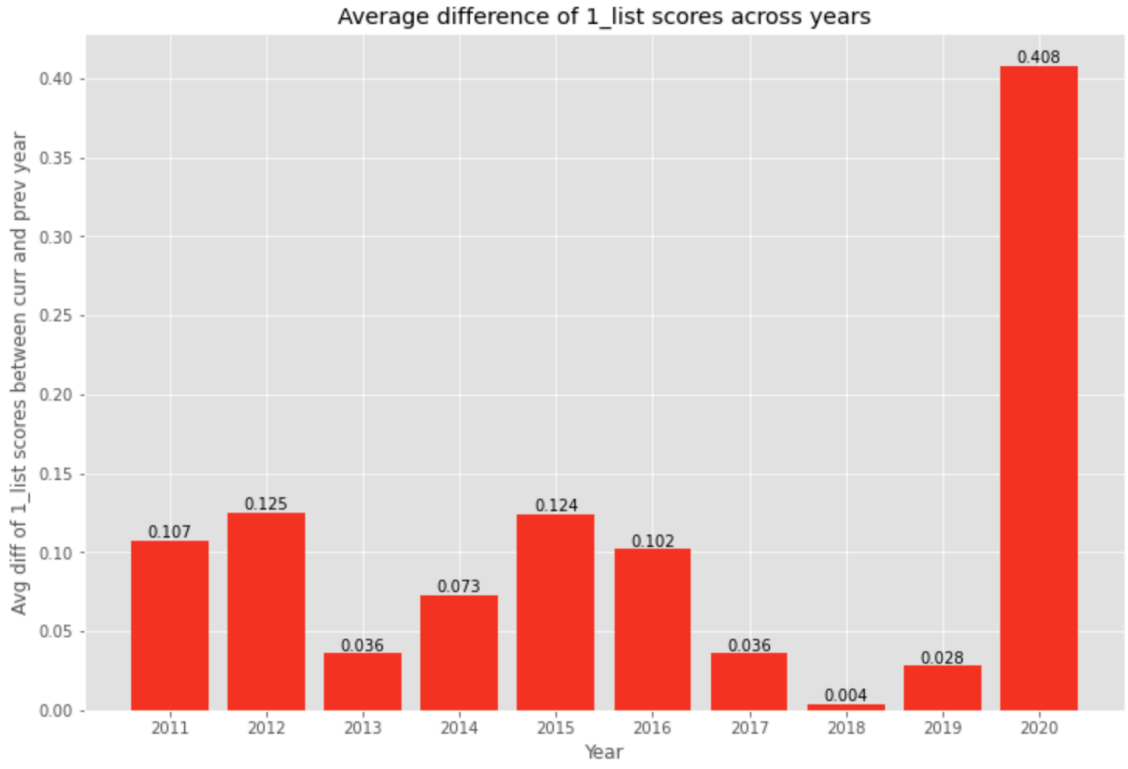


Figure 8: Average difference of 1_list scores before 2020 and in 2020

Therefore, it’s relevant to create a variable that reflects this change. This field has two different values, 1 if the year is >= 2020 and 0 if it’s < 2020

5. MODELLING

In this section we will be predicting the “1_list” scores for 2022. In order to do so, we will need to analyze which models and variables to use. As the output variable “1_list” is a number, we need to solve a regression problem. Therefore, models that we will be analyzing are Linear Regression, as well as ARIMA.

5.1 Correlation matrix

To analyze the linear correlation between variables, we have created a correlation matrix. Please see figure 9:

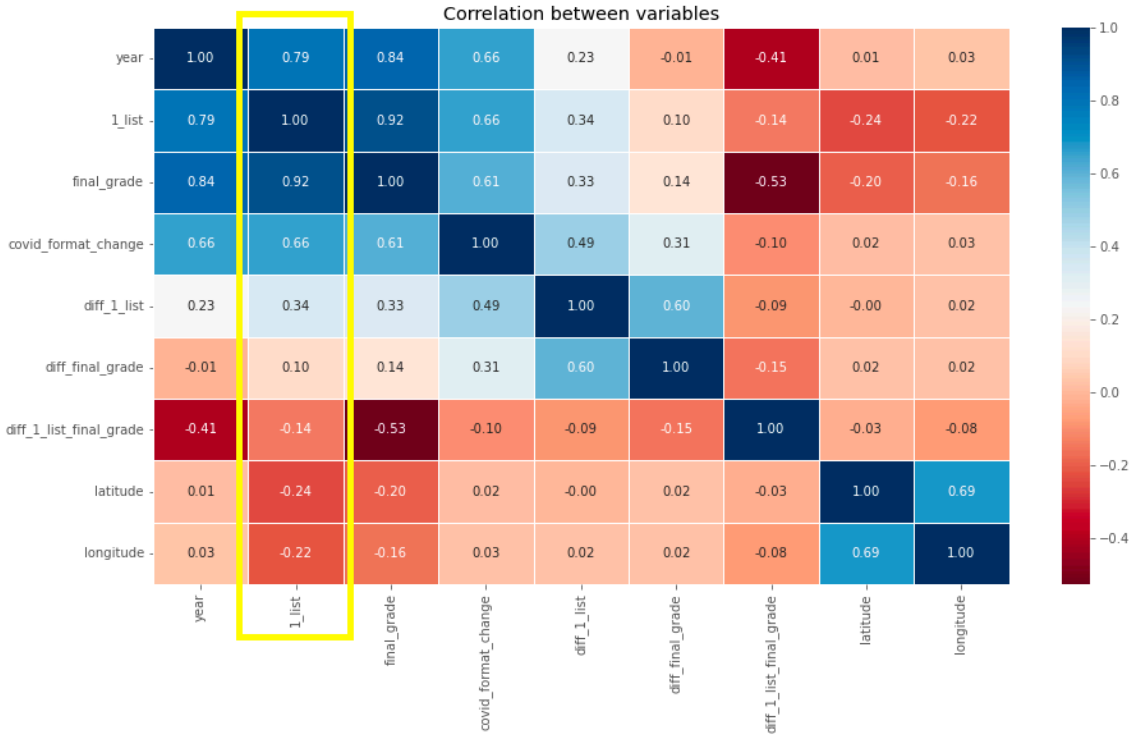


Figure 9: Correlation matrix

As we can see in the figure above, highlighting the output variable “1_list” we can see that the variables with higher linear correlation are:

- **year**: 0.79
- **final_grade**: 0.92
- **covid_format_change**: 0.66
- **diff_1_list**: 0.34

We will be using these variables in the models below to analyze which combination has the lowest error.

5.2 Linear Regression

In this section we will be using as training data the years 2010 and 2012-2020, as validation the year 2011 and the year 2021 as test data. The reasoning behind choosing 2021 as test data is that as we have explained before, the scores incremented considerably do to the change in the test format (due to COVID-19) in 2020. This way we could train the model considering this year and predict more efficiently the scores for 2021 (tests this year also had the format applied due to COVID-19).

In the next sections we will be showcasing the results obtained by using different variables in the model.

5.2.1 Variables: ['final_grade']

First model training was done with the variable "final_grade", which was the one that we saw above that had the highest correlation (0.92), with the target variable. Please see in figure 10 the results obtained predicting the test data:

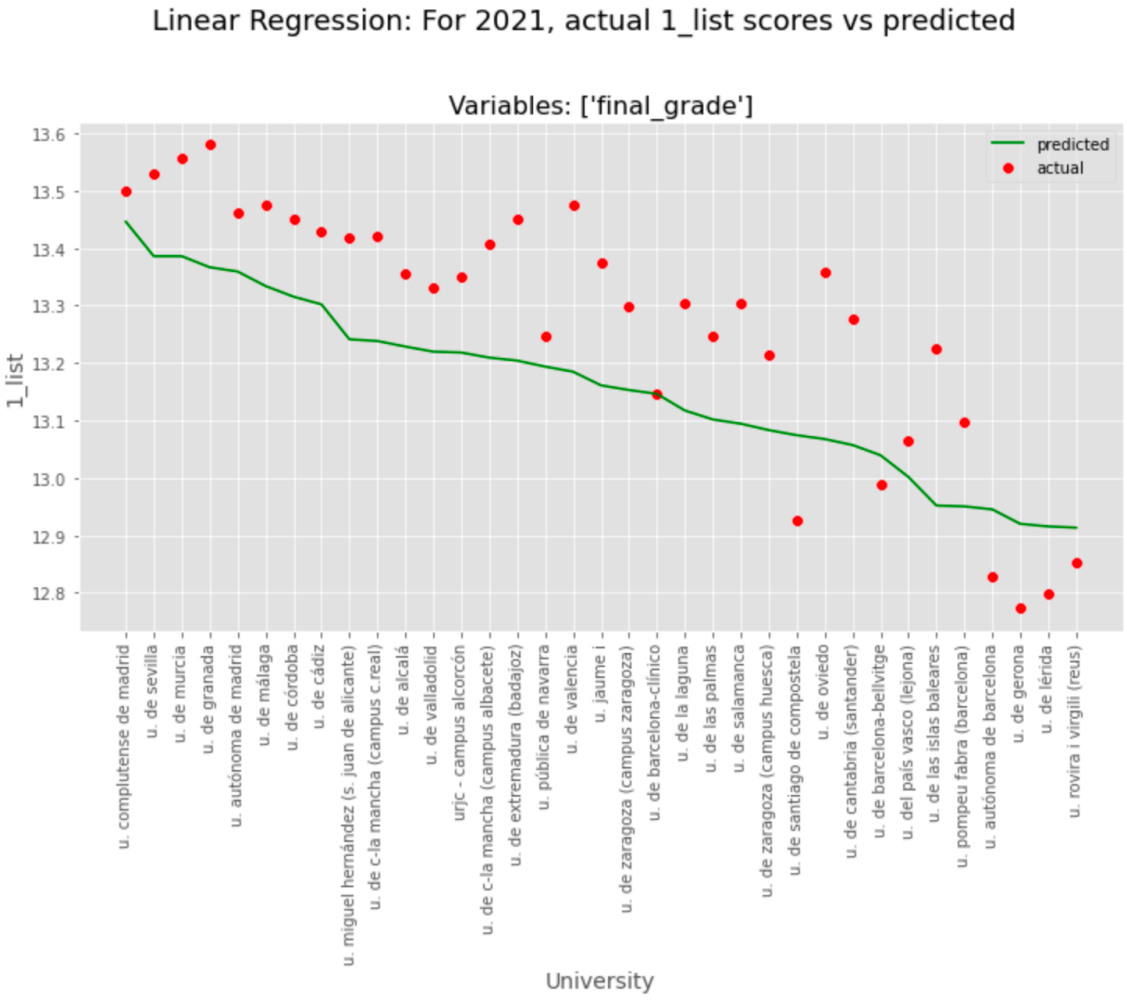


Figure 10: Linear Regression – variables: final_grade

Analyzing the graph above, we can see that as a base model, the model has predicted a similar trend than the actual data, although a bit lower scores than expected.

5.2.2 Variables: [‘final_grade’, ‘year’]

Second model training was done with the variables “final_grade” and “year”, which were the two variables with highest correlation (0.92 and 0.79, respectively). Please see in figure 11 the results obtained predicting the test data:

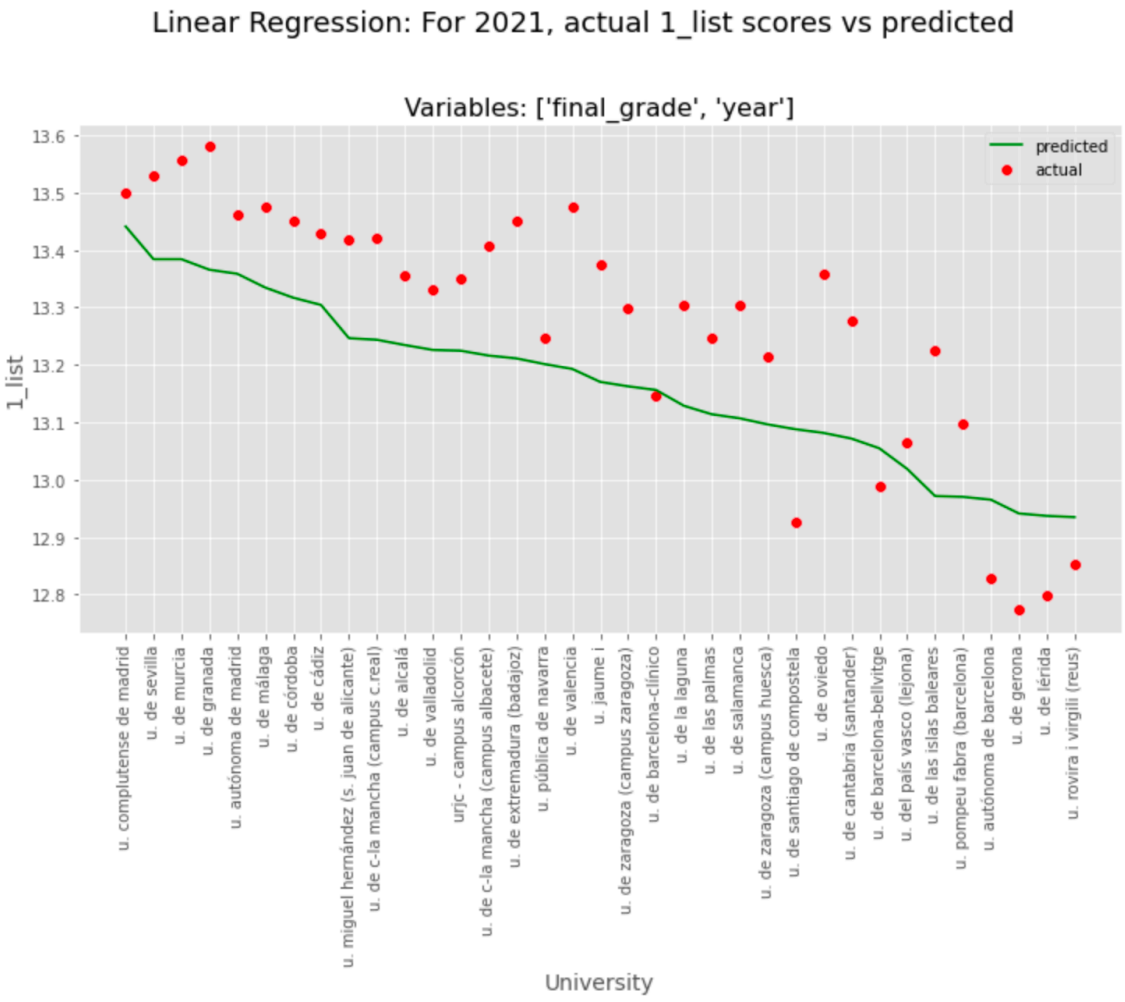


Figure 11: Linear Regression – variables: final_grade, year

Analyzing the graph above, we can see that the prediction is very similar to the previous one where we used only the variable “final_grade”. There is still room for improvement.

5.2.3 Variables: ['final_grade', 'year', 'covid_format_change']

Third model training was done with the variables "final_grade", "year" and "covid_format_change", with the following correlations: 0.92, 0.79 and 0.66, respectively. Please see in figure 12 the results obtained predicting the test data:

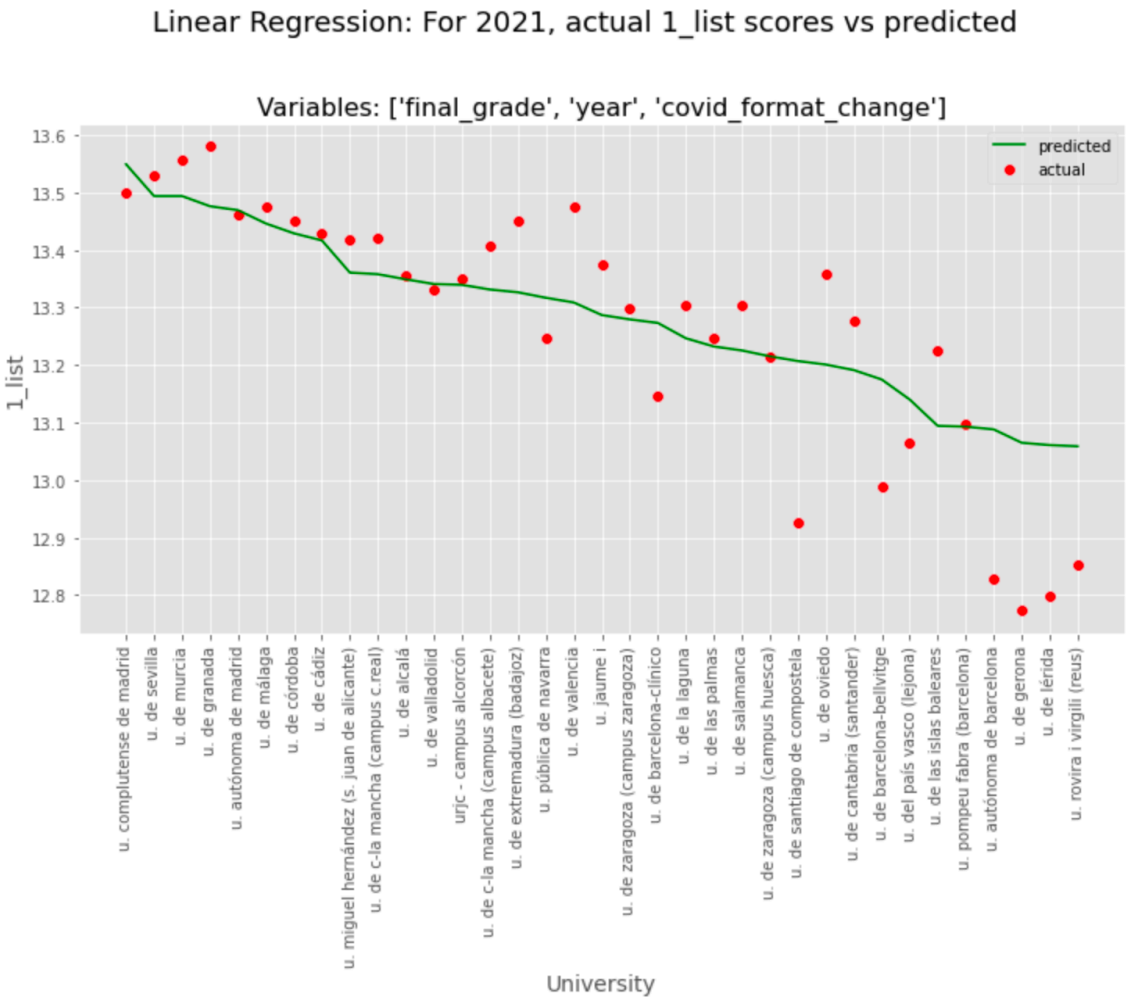


Figure 12: Linear Regression – variables: final_grade, year, covid_format_change

Analyzing the graph above, there is a significant difference between this prediction and the 2 previous ones. In general terms, it seems as there is less error. We will analyze this in detail in the section where we calculate the metrics (MAE, MSE, RMSE) obtained.

5.2.4 Variables: ['final_grade', 'year', 'covid_format_change', 'diff_1_list']

The fourth and last model training (for Linear Regression) was done with the variables "final_grade", "year", "covid_format_change" and "diff_1_list", with the following correlations: 0.92, 0.79, 0.66 and 0.34, respectively. Please see in figure 13 the results obtained predicting the test data:

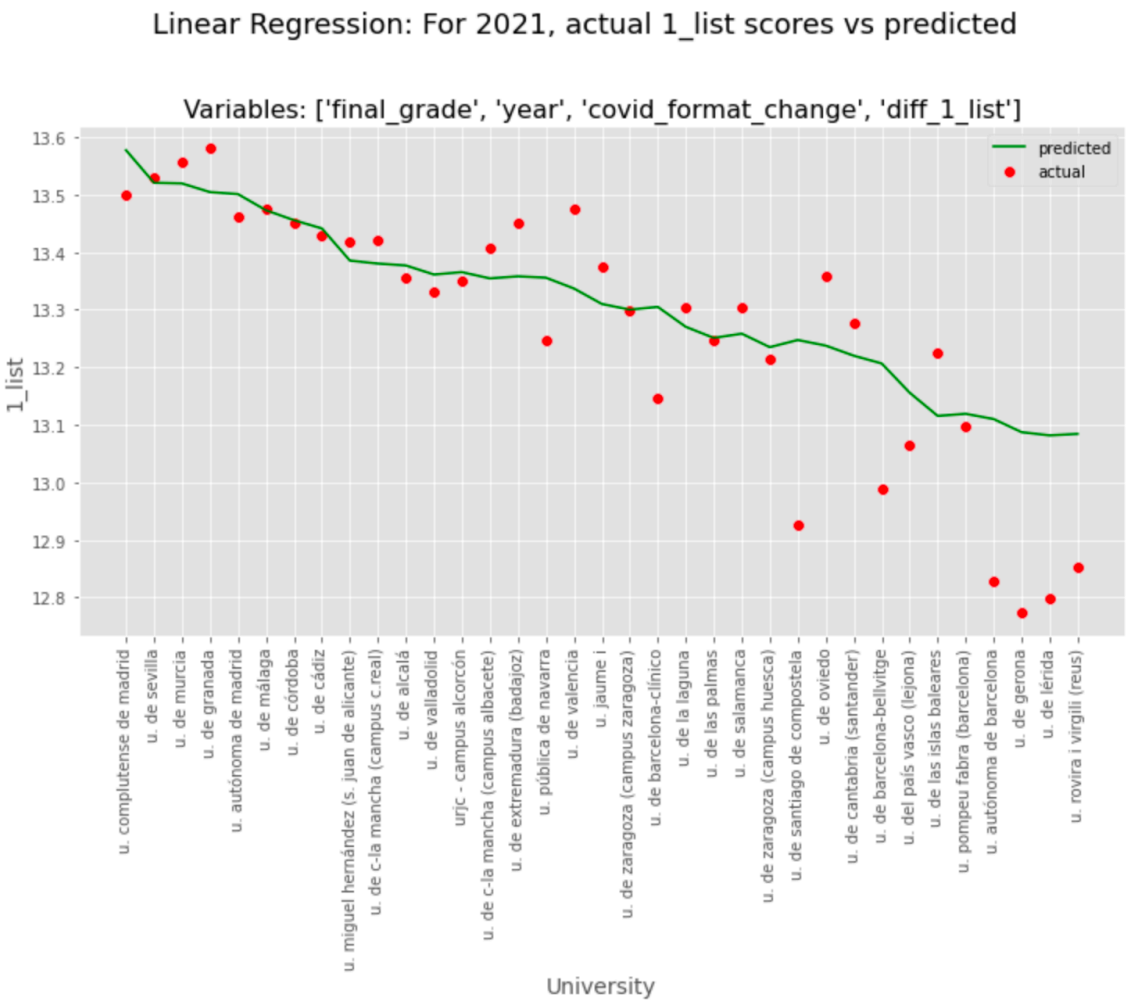


Figure 13: Linear Regression – variables: final_grade, year, covid_format_change, diff_1_list

Results seem to be similar to the ones obtained with the 3 variables: final_grade, year, covid_format_change. Error might be slightly higher due to the fact that this time we have trained the model with a variable with a not very high linear correlation (diff_1_list: 0.34). We will analyze this in detail in the metrics section.

5.3 Linear Regression Per University

In this section we will be doing the same split of training-test-validation data that we did before but in this case, we will be doing a Linear Regression model per university. Therefore, we will be creating 35 different models.

In the next sections we will be showcasing the results obtained by using different variables in the model.

5.3.1 Variables: [‘final_grade’]

First iteration training the 35 models was done with the variable “final_grade”. Please see in figure 14 the results obtained predicting the test data:

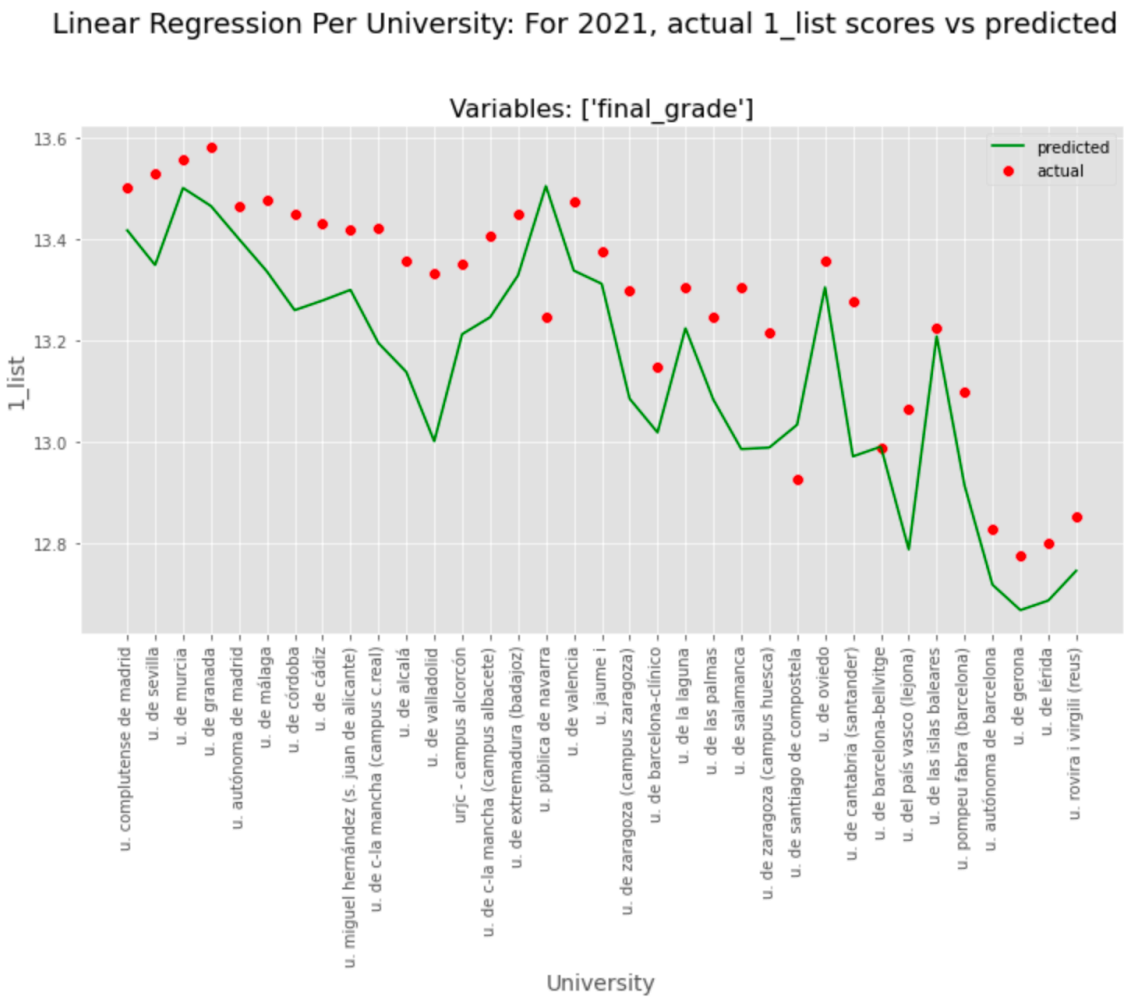


Figure 14: Linear Regression Per University – variables: final_grade

Analyzing the graph above, we see that the prediction is not almost a “straight” line as we were seeing before, but rather a more targeted prediction because in this case we are doing one model per university. Overall, the predictions seem a bit lower than the actual score, there is still room for improvement.

5.3.2 Variables: [‘final_grade’, ‘year’]

Second iteration training the 35 models was done with the variables “final_grade” and “year”. Please see in figure 15 the results obtained predicting the test data:

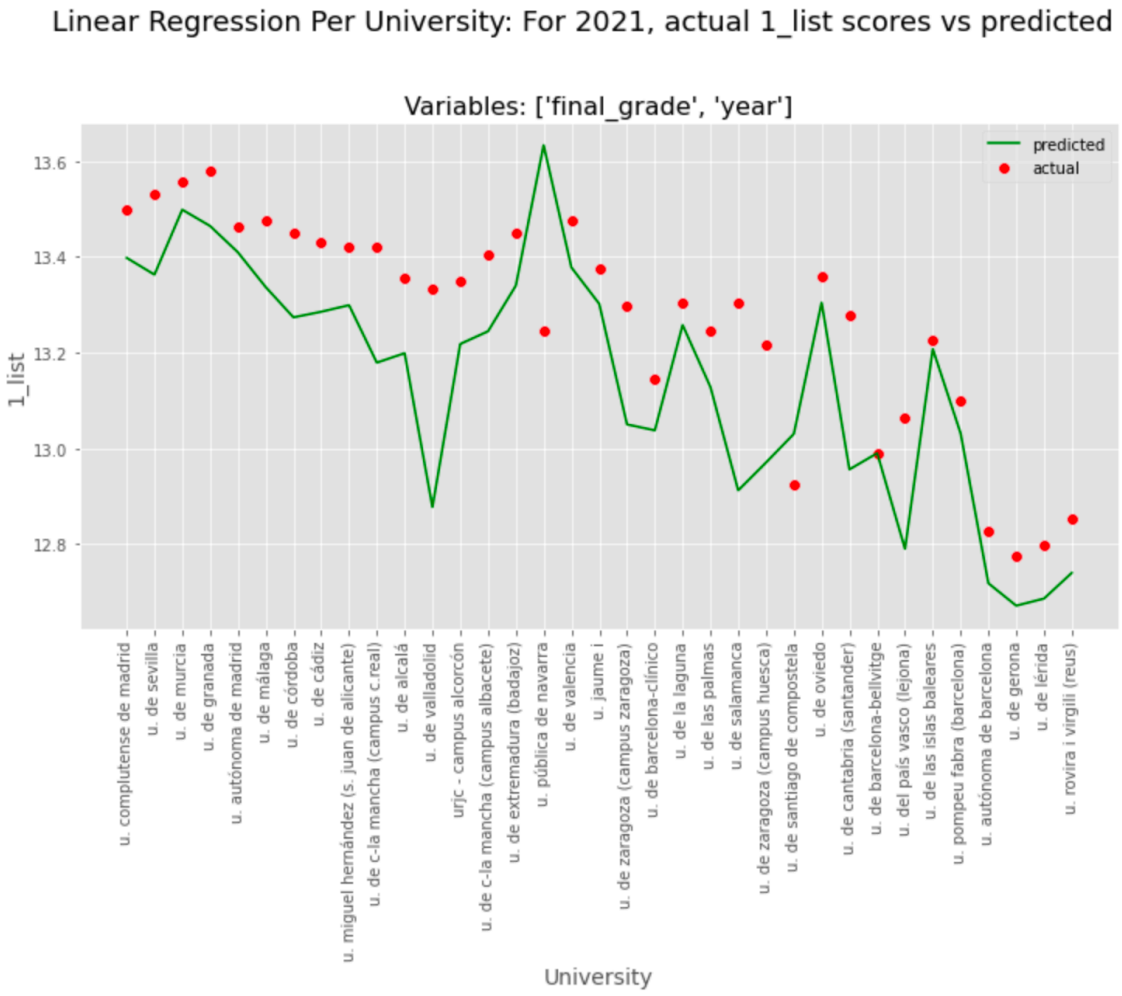


Figure 15: Linear Regression Per University – variables: final_grade, year

Similar trend as the previous scenario seen with only the variable “final_grade”. In general terms, predicted scores seem lower than actual ones.

5.3.3 Variables: [‘final_grade’, ‘year’, ‘covid_format_change’]

Third iteration training the 35 models was done with the variables “final_grade”, “year” and “covid_format_change”. Please see in figure 16 the results obtained predicting the test data:

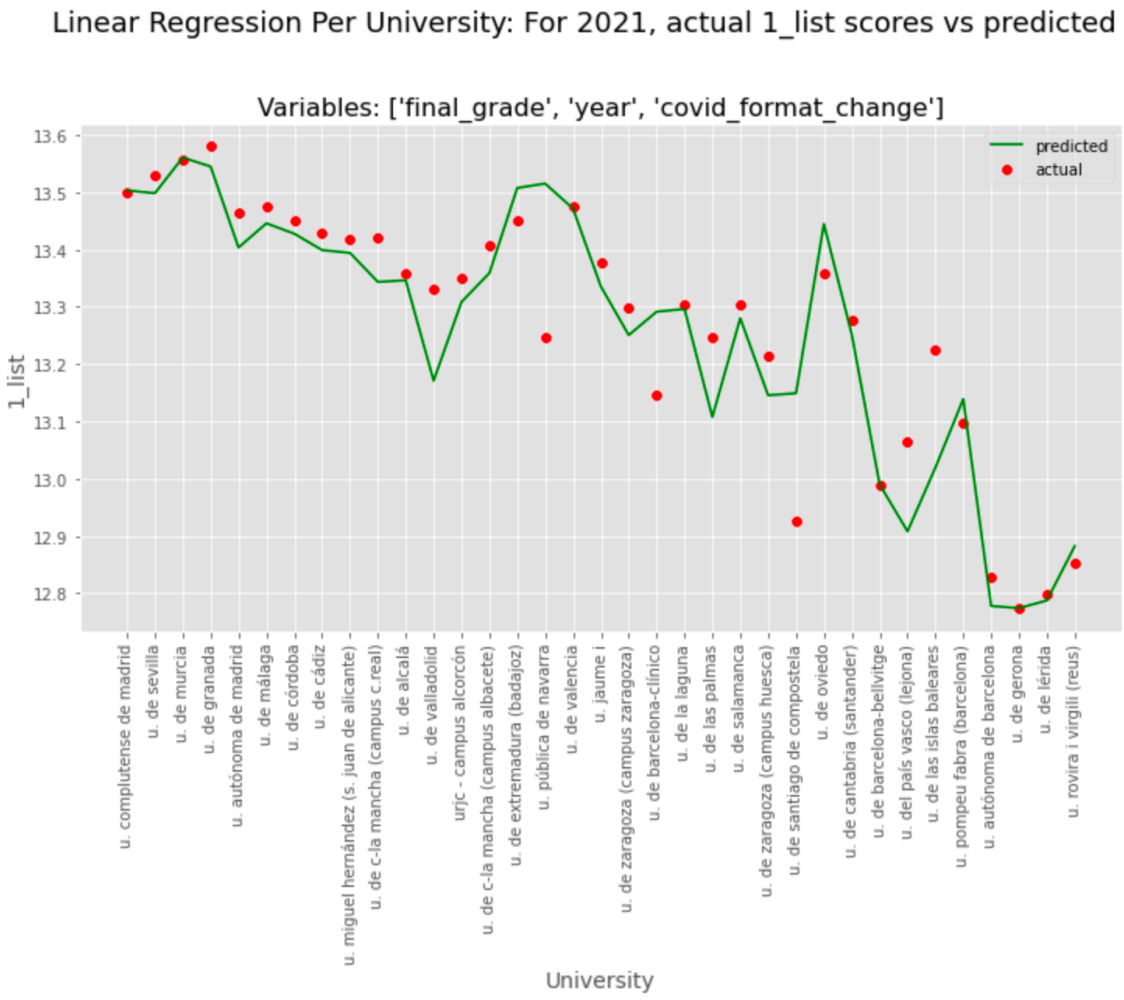


Figure 16: Linear Regression Per University – variables: final_grade, year, covid_format_change

Analyzing the graph above, we can see that the predicted scores are noticeably more accurate than the first two iterations. The predicted scores don’t seem as low as in the other iterations in comparison to the actual ones.

In the metrics section we will most likely see this combination (Linear Regression per university models using variables final_grade, year and covid_format_change) as one of the ones, or the one, with less error.

5.3.4 Variables: ['final_grade', 'year', 'covid_format_change', 'diff_1_list'

Fourth and last iteration training the 35 models was done with the variables "final_grade", "year", "covid_format_change" and "diff_1_list". Please see in figure 17 the results obtained predicting the test data:

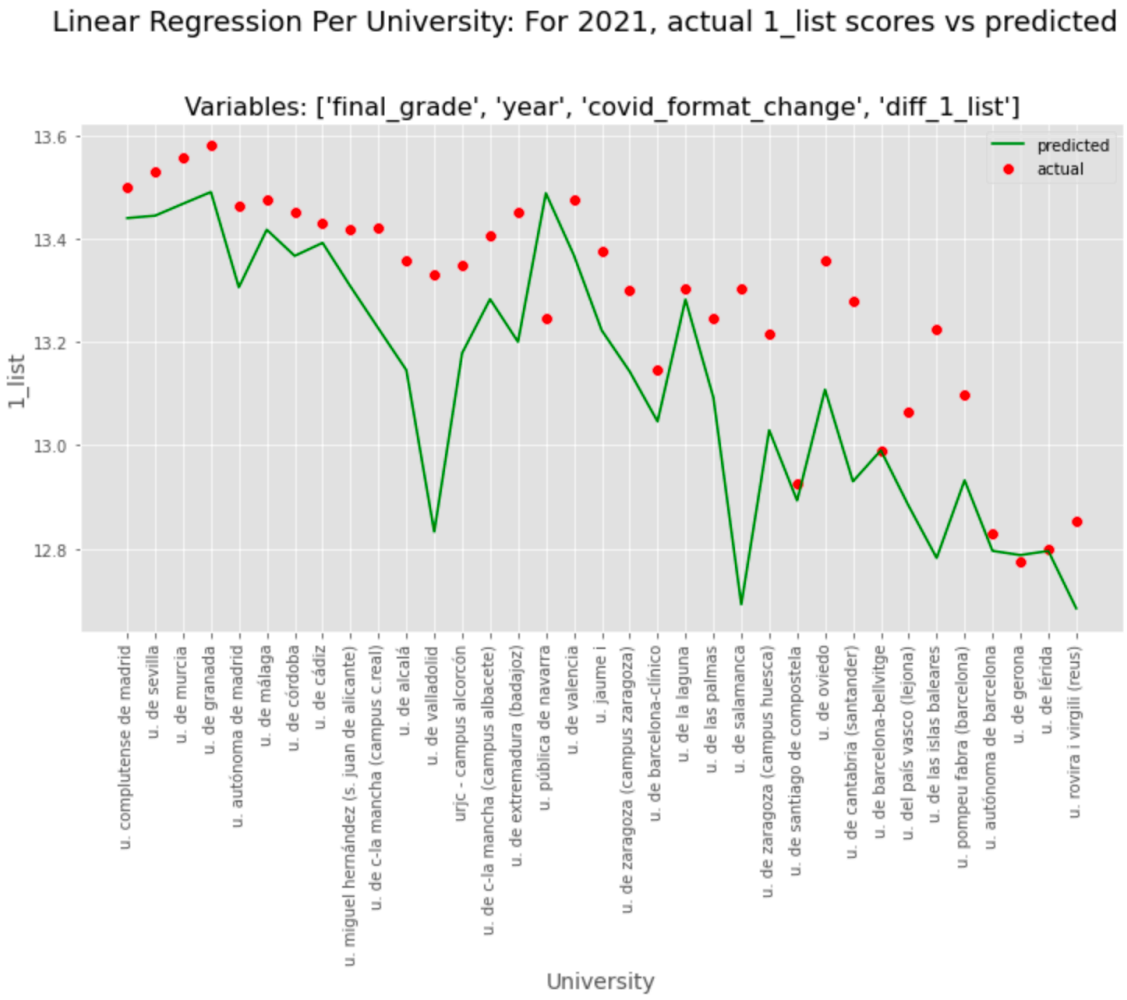


Figure 17: Linear Regression Per University – variables: final_grade, year, covid_format_change, diff_1_list

Since in this case we are introducing a variable with low linear correlation (diff_1_list: 0.34) we could have expected that the model prediction would be worse than without this variable and indeed, looking at the graph above we see that the predicted scores are not as accurate and are overall quite below the actual scores.

5.4 ARIMA Per University

In this section we will be doing the ARIMA model instead of Linear Regression, per university. Therefore, we will be creating 35 different models. Same split of training-test-validation data that we did before. Please see in figure 18 the results obtained predicting the test data:

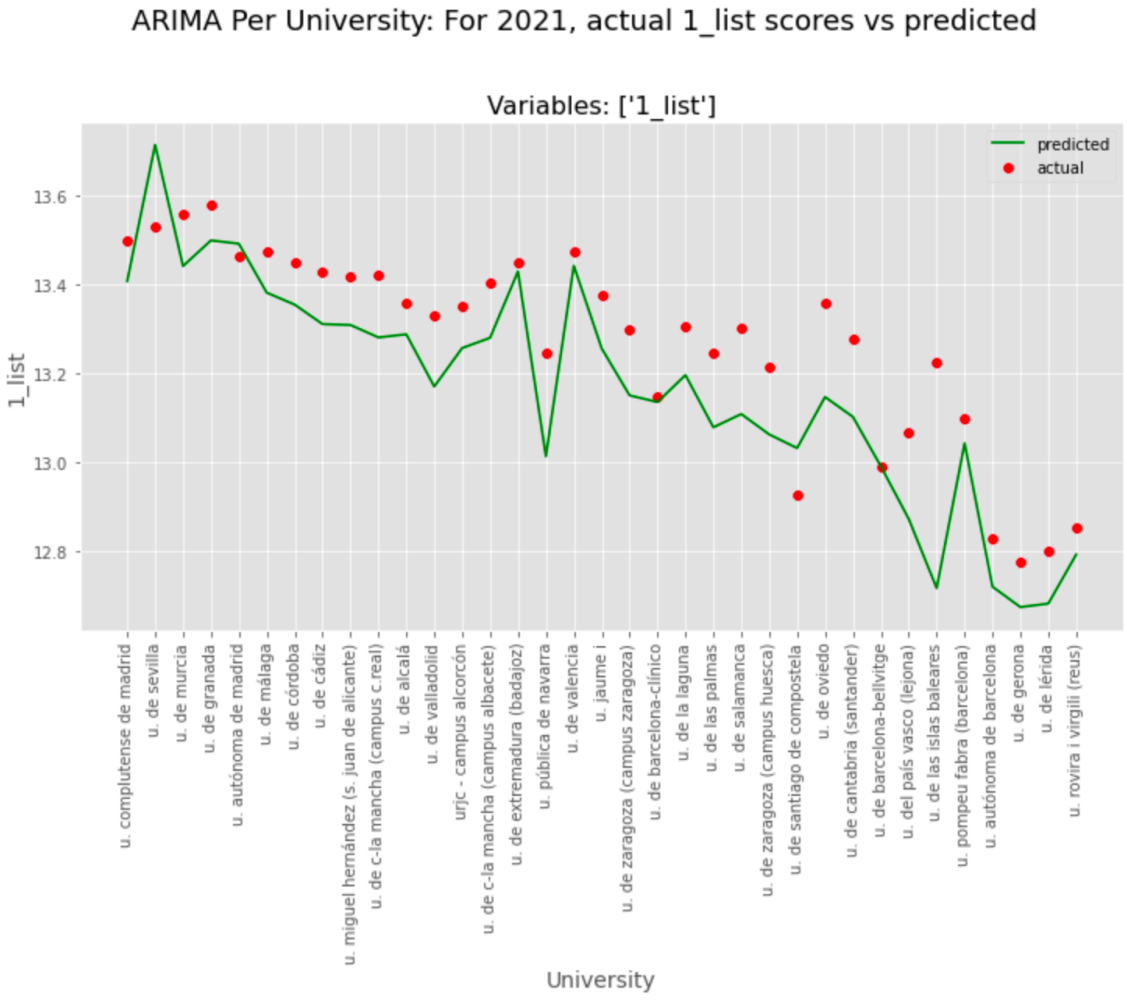


Figure 18: ARIMA per university – variables: 1_list

We have used the variable “1_list” in the 35 ARIMA models and we can see that the predicted scores are quite below the actual ones.

5.5 Metrics

The metrics we are going to use to evaluate the different models showcased are MSE (Mean Squared Error), RMSE (Root Mean Squared Error) and MAE (Mean Absolute Error). In figure 19, we can see the scores of these metrics for the different combination of models and variables analyzed in previous sections:

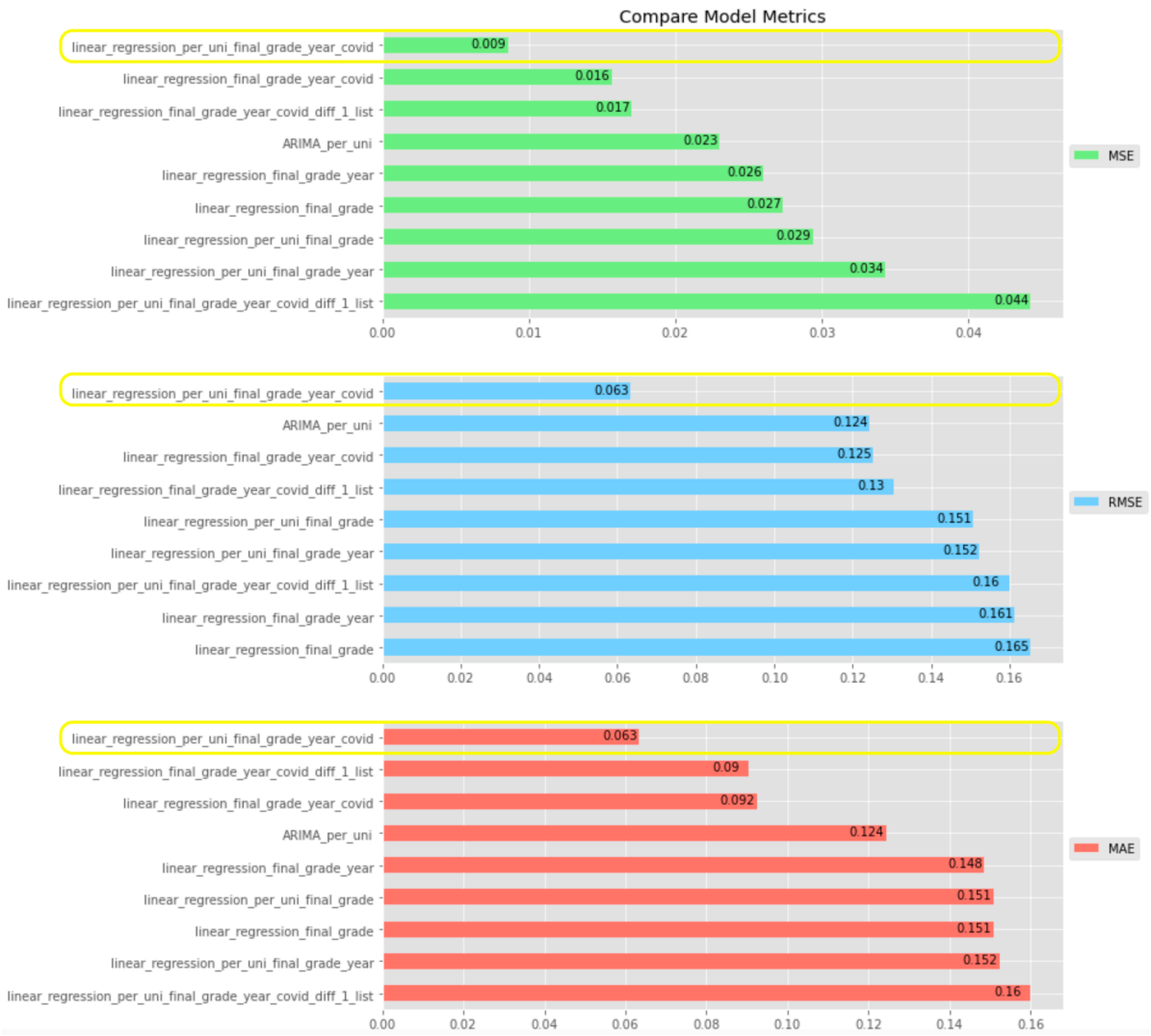


Figure 19: Comparison of Model Metrics

Analyzing the graph above we can see that the model-variables combination with less MSE, RMSE and MAE is the Linear Regression Per University using the variables “final_grade”, “year” and “covid_format_change”. Therefore, we will be using this combination to predict the 2022 “1_list” scores.

5.6 Check if there is overfitting

Before predicting the 2022 “1_list” scores, let’s check if in the model configuration that we are going to use there is overfitting. To check this, we are going to do the following:

- graph of predicted vs actual 1_list values for validation data (year: 2011)
- plot and compare metrics of the validation data and the test data, they should be similar

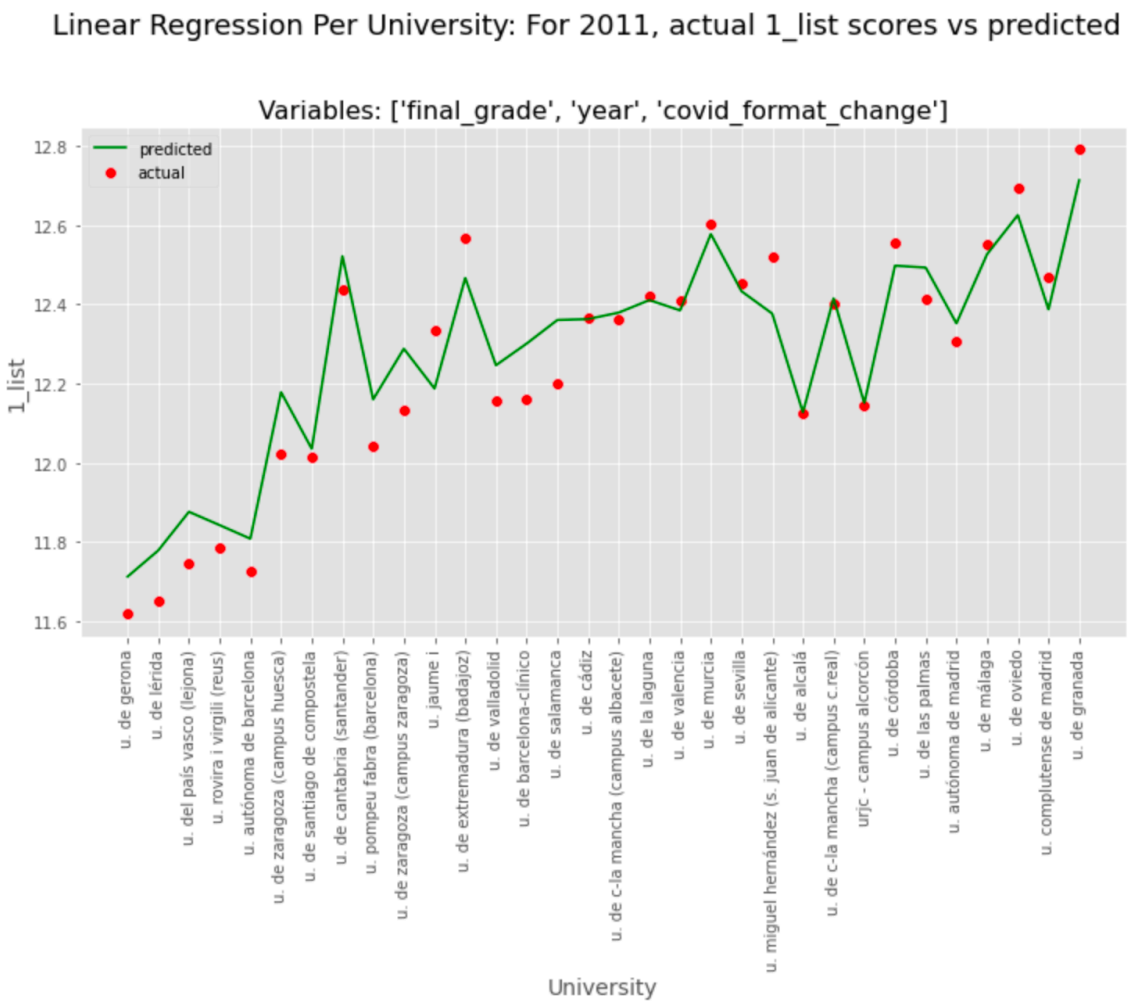


Figure 20: year 2011 - Linear Regression Per University – variables: final_grade, year, covid_format_change

Figure 21: Metrics highlighting test vs validation

Analyzing figure 21, we can see that the metrics between the validation and test data are very similar, as we expected. Taking this into account, plus analyzing the model predicted values in figure 20, we can confirm there isn't overfitting.

5.7 Predict next year (2022) 1_list scores

As discussed in the previous section, we will be doing a Linear Regression Per University, training the models with the variables “final_grade”, “year” and “covid_format_change”, since with this combination we observed it had the least MSE, RMSE and MAE out of all the other combinations analyzed.

To predict the next year (2022) **“1_list”** scores we input as variables the following data:

- **“final_grade”:** “final_grade” of previous year (2021)
- **“year”:** 2022
- **“covid_format_change”:** 1.0, as they will still be using the same test format used since COVID-19 in 2020.

To get the next year (2022) **“final_grade”** scores we calculated them the following way: for every university, we took the predicted “1_list” score (2022) and subtracted the difference between the “1_list” and “final_grade” that the university had the previous year (2021). The formula would like this:

“final_grade” (2022) = “1_list” (2022) – diff_1_list_final_grade (2021)

6. FRONT-END

The front-end has been done on streamlit, to use it you will need to run the “myapp.py” file. In this section we will be analyzing the different data visualizations that are in it.

6.1 Objective

The objectives of the front-end are to make a **simple and interactive interface**, so that a user could understand and start using right away, and making it **insightful** by showcasing not only the grades from the different universities but also providing additional information such as location of the universities, historical scores, filters to see if you could be admitted given a certain score, etc.

6.2 Filters and visualizations

There are many graphs and functionalities created to fulfill the objectives mentioned above.

6.2.1 Filters / Selectors

In figure 22 you can see highlighted the filters/selectors section. There are 4 different options:

- **“Select year”**: to showcase data from the year selected. Values range from 2010 to 2022.
- **“Select score”**: to choose between showcasing the “1_list” scores or the “final_grade” ones.
- **“Filter by CCAA (optional)”**: you can filter by the autonomous communities of Spain, in case you are interested in a particular one. This filter is optional, in case you don’t select a specific community the default value is “All” which will show data from all of them.
- **“Filter by score (optional) (e.g., 12.5)”**: you can filter by a score. This is very helpful if you already know your score, or can sense the score you will get, and want to see the different universities you could be admitted to. This filter is optional.

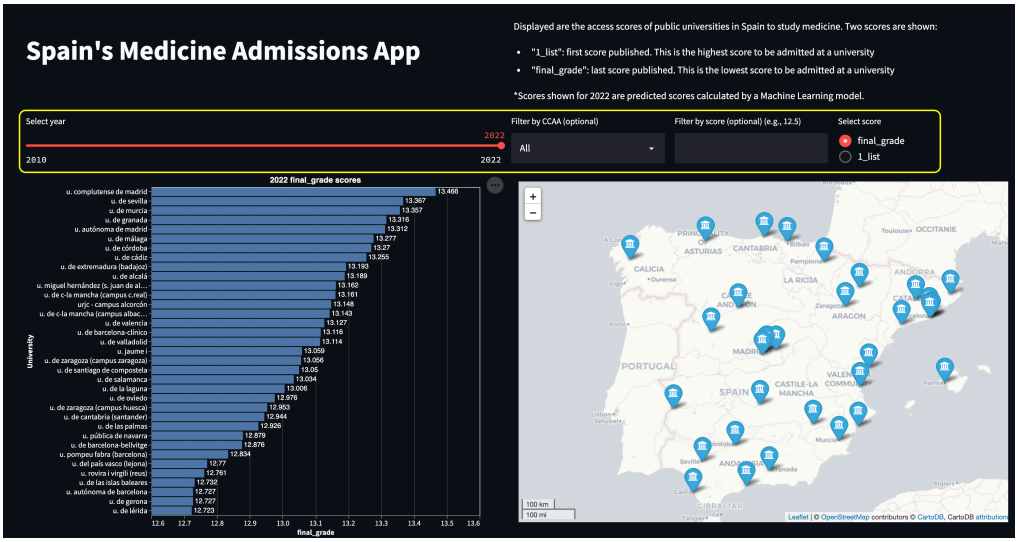


Figure 22: Top part of front-end with focus on filters / selectors

6.2.2 Bar chart with 1_list / final_grade scores by university

On the top left-hand side of the app, we can find a bar chart showcasing the selected score ("1_list" or "final_grade") and the selected year, in the filter section, for the different universities. Please see figure 23, showcasing the bar chart with the year 2022 selected and "final_grade" score selected:

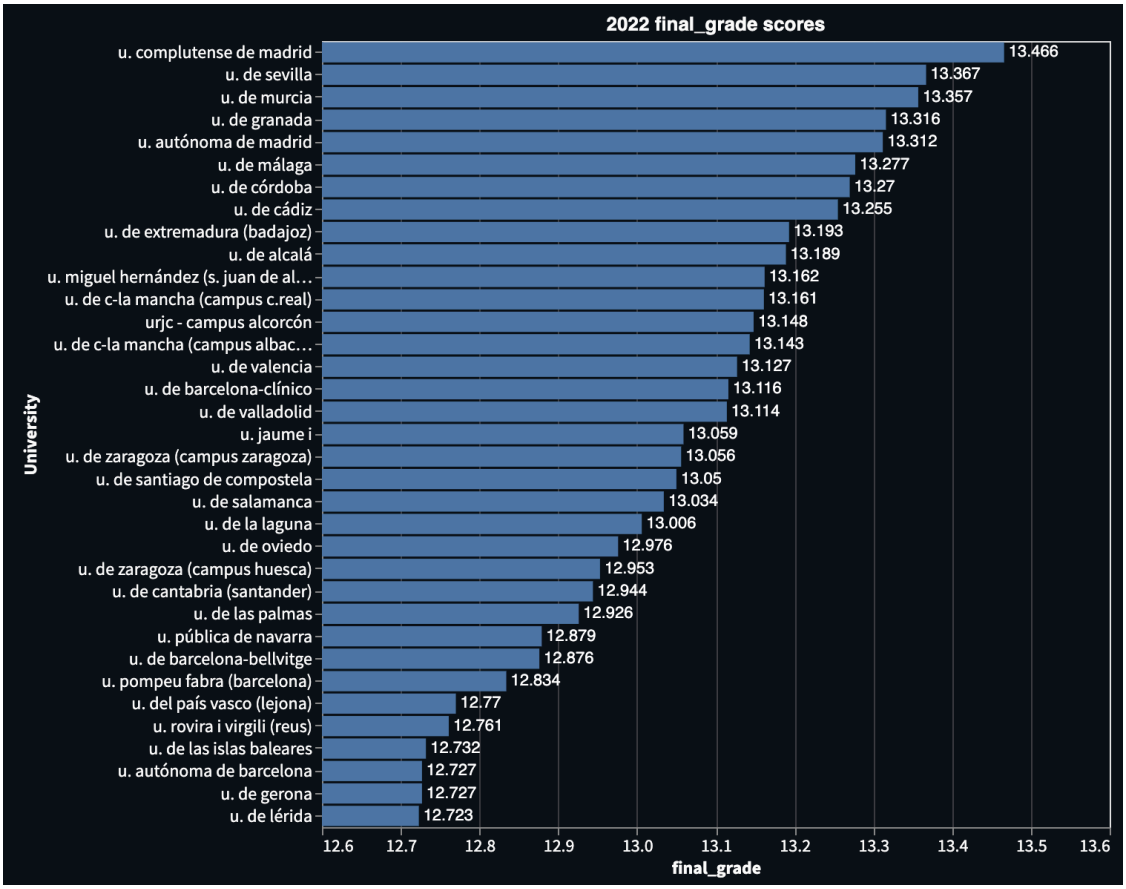


Figure 23: Bar chart of scores per university

If we would like to filter by a certain score, to see what universities we would be admitted in given that score, we can do so, it marks in orange the ones you would be admitted in. Let's say we have a score of 13, which universities would I be able to be admitted:

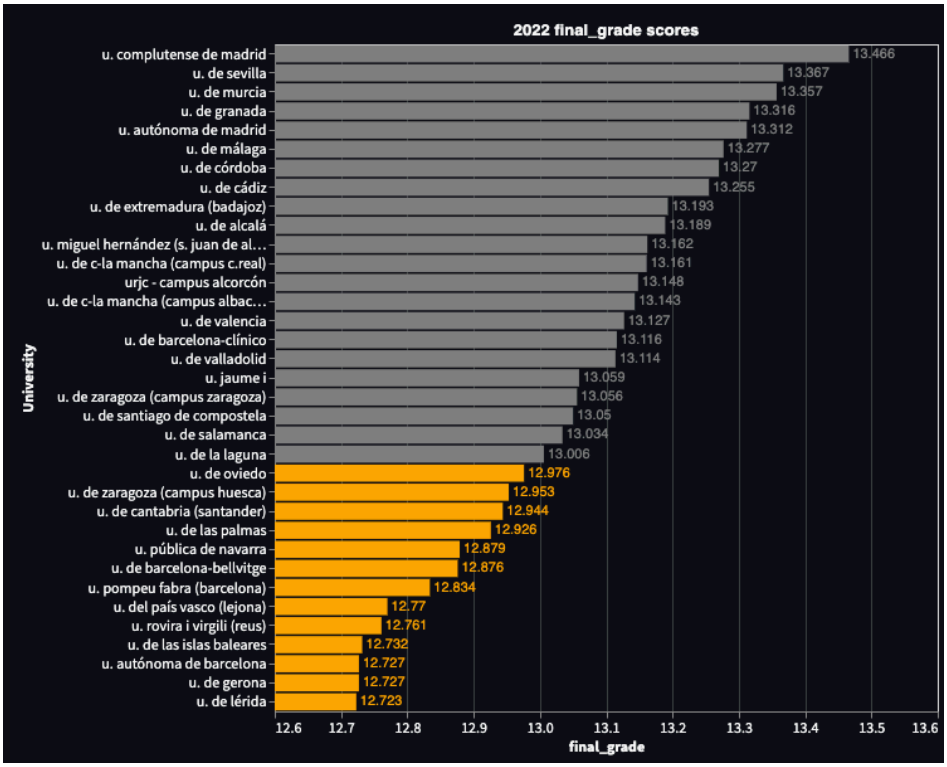


Figure 24: Bar chart of scores per university filtered by score

You can also filter by a CCAA, in case you are interested in a certain region of Spain. If we are interested in studying in Cataluña, it will only highlight the universities from that CCAA and that are under our score of 13 (previously filtered):

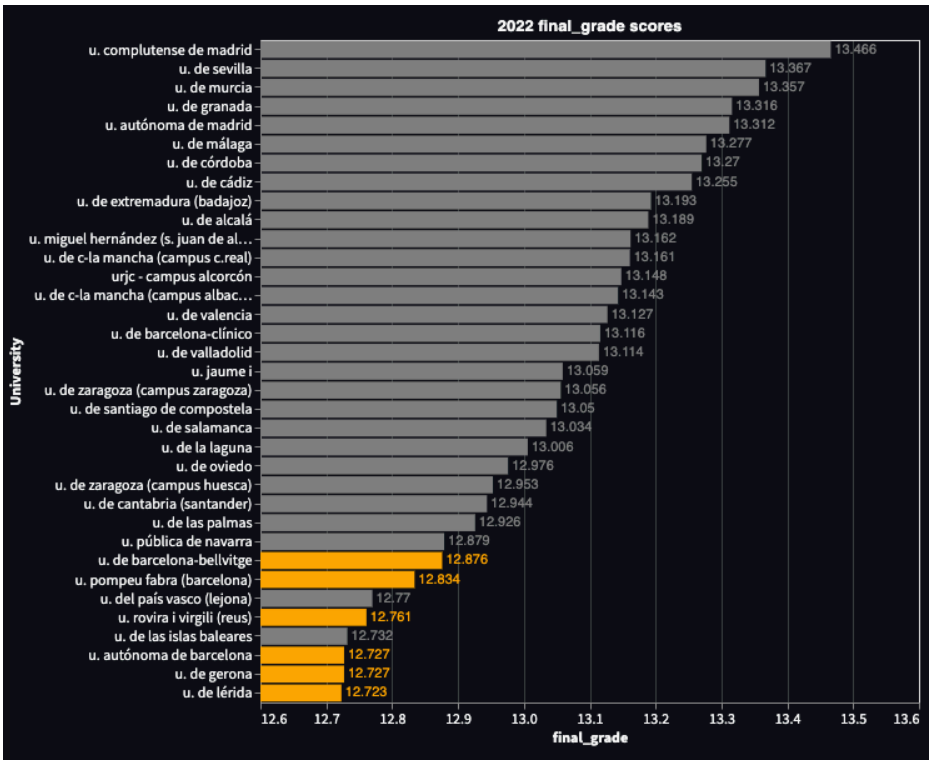


Figure 25: Bar chart of scores per university filtered by score and CCAA

6.2.3 Map

If we would like to know the location of the universities, there is a map of Spain with the exact latitude and longitude of each of the universities. This map has several functionalities:

- **Click and drag to move through the map**
- **Hover over functionality:** showcasing the university name and the score
- **Click on universities:** helpful to compare scores (within the map) using it together with the hover over functionality
- **Zoom functionality:** to deep dive or zoom out on different parts of the map
- **Filter coloring:** CCAA and score filters affect the map by coloring in orange the options selected/admitted and in grey the ones that aren't

Without applying filters, the map would like in figure 26:

Figure 26: Map with locations of universities

If you would like to filter by a score, you can do so, same as the bar chart, it will mark in orange the universities you would be admitted to. If we filter by any score, e.g., 13.3, we get the following:

Figure 27: Map with locations of universities filtered by score

Additionally, if you would like to select a certain CCAA, it will zoom on that CCAA. For example, if we select Madrid, we will see the following:

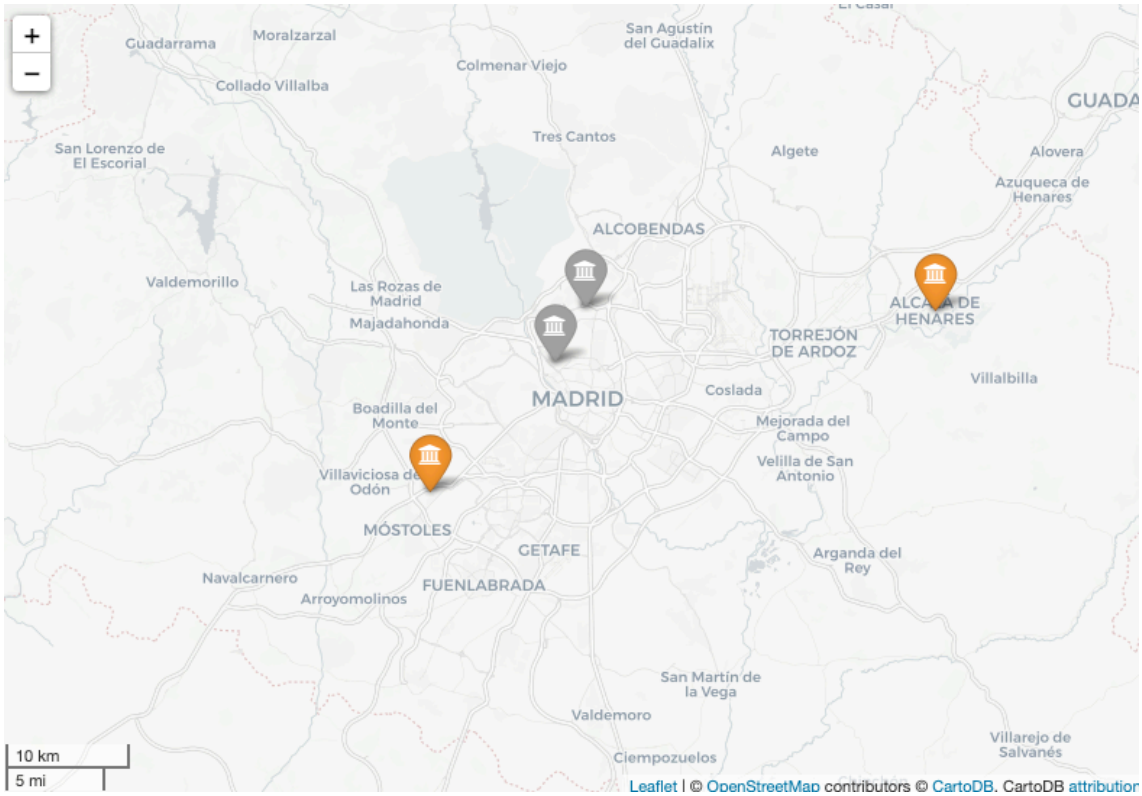


Figure 28: Map with locations of universities filtered by score and CCAA

6.2.4 Bar chart with average scores by CCAA

If you would like to know the average scores by CCAA, there is a bar chart that shows this information. Please see figure 29:

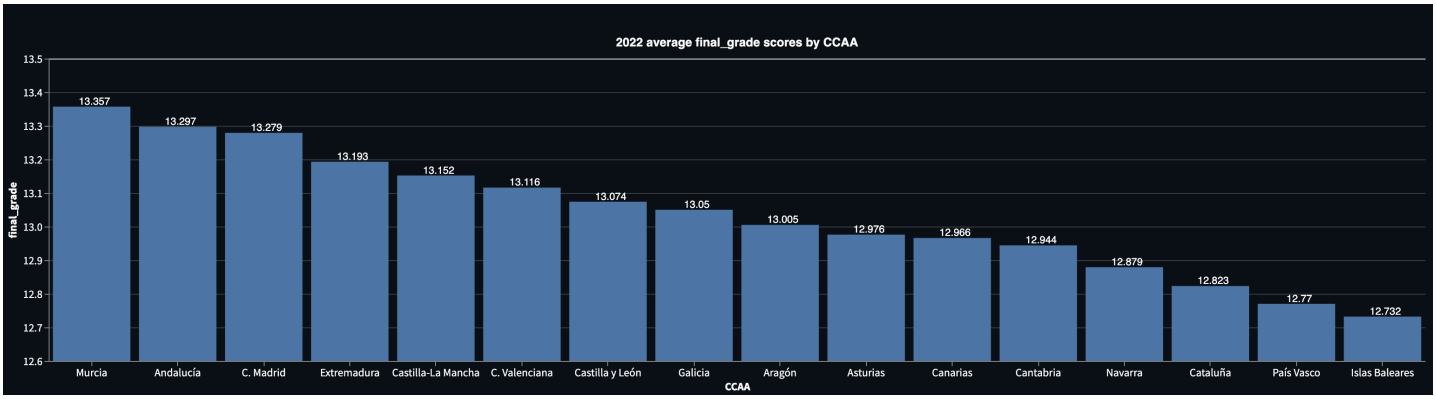


Figure 29: average scores by CCAA

When you select a CCAA, it highlights that CCAA in the bar chart. Same methodology as explained earlier in the other bar chart.

6.2.5 Difference between 1_list and final_grade by university and by CCAA

Some people get admitted to an university they applied for but they decide to switch to another university, making the access score of the university they are leaving go down. Below are 2 bar charts showcasing the difference between the “1_list” and the “final_grade” scores by university and by CCAA.

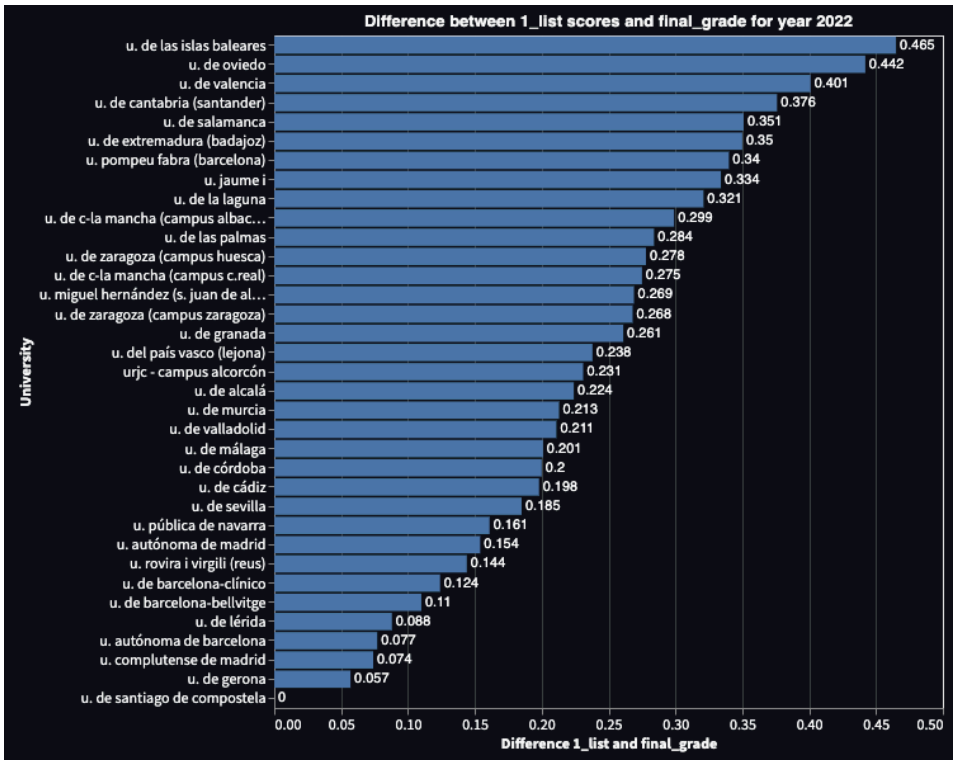


Figure 30: Difference between 1_list and final_grade by university

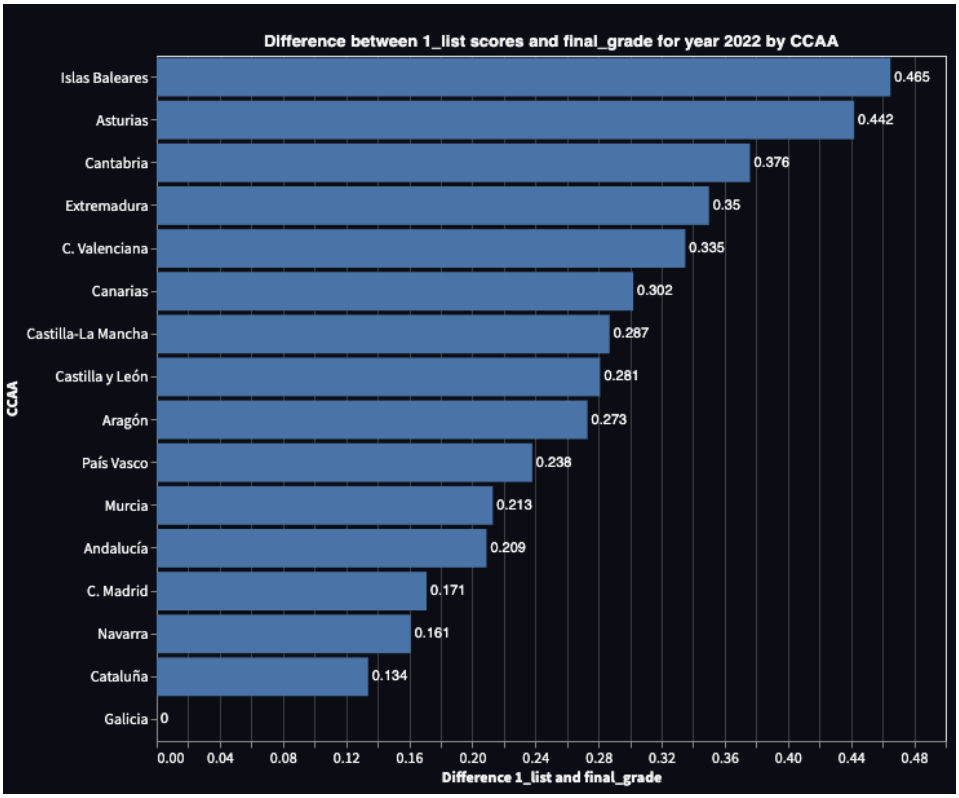


Figure 31: Difference between 1_list and final_grade by CCAA

As well as in previous visualizations, a colorway is created to highlight the filters chosen.

6.2.6 Historical scores and correlation heatmap

Lastly, there is a line graph showcasing the historical results for the university that you choose. Results showed are going from 2010 to 2022, see figure 32. Also, next to this graph there is a correlation heatmap between the score chosen and the variable year, in case you would like to know how linear the historical scores have been.

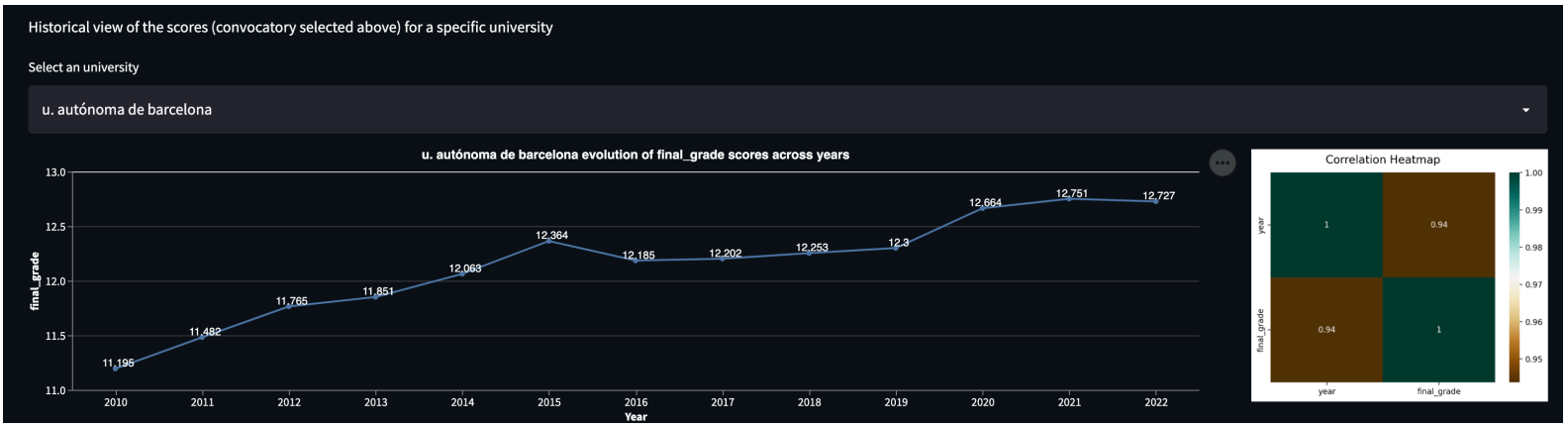
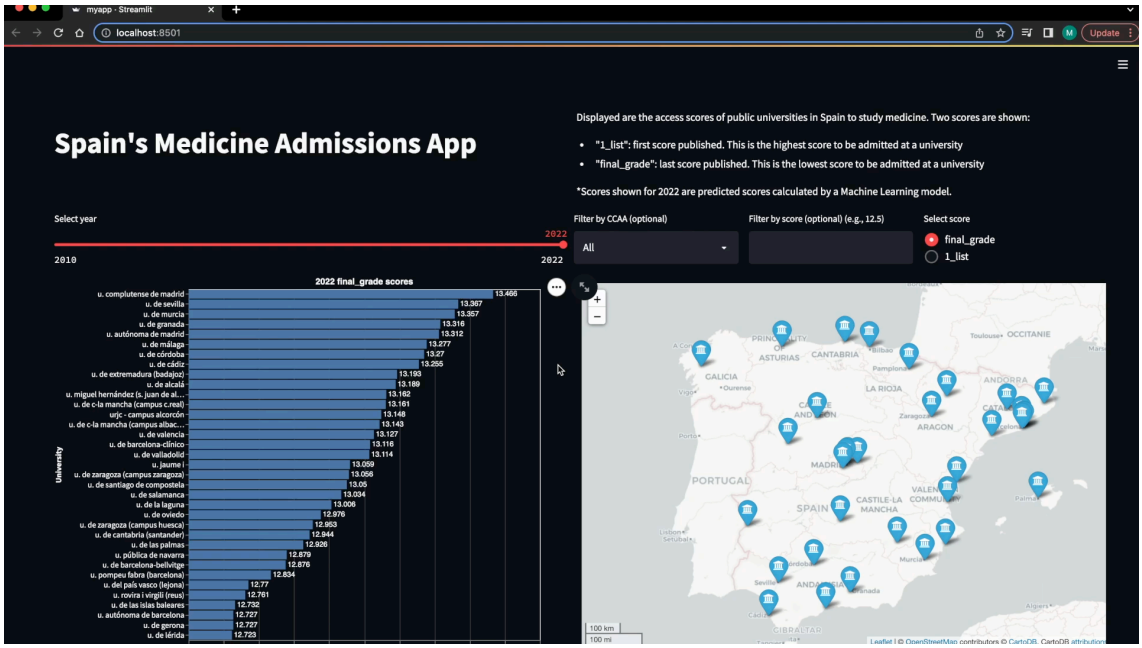


Figure 32: Historical scores and correlation heatmap for university selected

6.3 Video using front-end

In the following video you can get a feeling of the different functionalities and visualizations that have been mentioned above. Please double click on it to start visualizing it.



Link to video on YouTube: https://youtu.be/hWlolS7YmA4

7. CONCLUSION

In conclusion, if we go back to the 2 objectives we described in the beginning of this document, we could see that we have completed them successfully:

1) **Predict next year scores (2022):** we have done this by analyzing different model configurations and ended choosing a Linear Regression Per University with variables: “year”, “final_grade” and “covid_format_change” since it had the best metrics.
2) **Create a centralized and accessible place, a web app**, to consult the different Medicine admission grades from all the universities in Spain, in a visual and helpful manner: we have done this by building a front-end on streamlit.

All in all, this project will be helpful for students to avoid uncertainty of not knowing whether the next year they will be able to be admitted to a certain university or not and visualize all the different possibilities that they have.

As next steps, I would like to share this project with the data community, to get feedback on it and most likely on the next version it will be deployed on Google Cloud for people to be able to use it in a more friendly manner without having to run any code.