MASTER'S FINAL PROJECT

AUTOMATIZATION OF MARKET RESEARCH DEPARTMENT:

MARKET RESEARCH HUB™

LLUÍS PELLEJÀ SOLDEVILA

MASTER IN DATA SCIENCE 2022/20223

INDEX

1. INTRODUCTION

Welcome to my final master's thesis. In this document I am going to explain my project in the clearest possible way, so that any data scientist can replicate it without the complications that I have had when finishing it.

This project started as a small application that has evolved according to the needs of the company's Market Research department. These new needs have allowed the app to be very complete today, having many more modules than it had at the beginning, and solving problems in more than one department in the company.

I will separate this document into two parts, the first is where I explain the reasons and context why the development of this application was necessary, and why it made sense to do so.

In the second part I will explain the notebooks I have used to make the tests and the tests to make the machine learning model reach its maximum potential and above all I want to explain the data I have used to analyze how to make this model in the best possible way.

The third part of the document is the explanation of how the app works so that anyone with the same problem we had, can develop it or replicate it without facing the problems I had when doing it.

Having made this brief introduction, we can start with the Master's thesis report.

2. OBJECTIVES

This project was born with the main objective of saving and optimizing time for the Market Research department, in the following section I explain why it was necessary.

Once the project started, several problems arose that could be solved by adding more modules to the application, so the objective was not only to optimize time but also to help us make the best decisions about the sales and SDR departments.

The objectives that this application had to fulfill are the following:

- To optimize time in the search for SDR contacts.
- To increase the volume of contacts without the need to increase the number of staff.
- To be able to measure and analyze the metrics of the SDR team in real time and in the same tab.
- Be able to listen to the most important calls of each SDR in a quick way
- To be able to show updated metrics to the SDRs with access control so that they do not see sensitive information.

In short, the main purpose of this application was to automate processes so that the Market Research department could dedicate its efforts to strategic reasons and leave aside the more operational and manual actions with the main objective of providing more value to the company.

3. CONTEXT

In this part of the dissertation report, I would like to give a bit of context as to why it was important to do this project, as I believe that this way the problem it solves and, above all, the value it brings will be better understood.

First of all we will separate the app modules into three parts:

- The duplicate check
- The one for generating company info
- The part of the KPIs and calls

Duplicate Check:
In the section on checking duplicates, what the programme does, roughly speaking and without going into detail, is that it examines the companies that we pass it to check and checks whether they are clients or are in "pilot test" within the ERP[1].

Also checks whether they are in the CRM[2] (Pipedrive), in the case that they are in the CRM and have won or open business, the app understands that it is a company that we cannot call as it was already won in its day, but if the deal is in the CRM and is lost and that more than 90 days have passed, it can try to call it again.

Also in the module to check duplicates you can merge in one click the duplicate companies in the CRM, this part is explained in more detail in section 5 a), where I only talk about this module in question.

Before we had the application, what we used to do was to generate an excel file with all the clients every week, to keep it updated and with an excel formula we crossed it with the companies we wanted to prospect and we kept deleting them.

Once the clients were deleted what we did was that we went company by company checking if in the CRM they were open, won or lost, in the case of lost clients we had to look at the annotations, the reason and the time they had been lost, which meant a lot of time.

With the application we not only save the time of doing it by hand, but while the programme is working we can do other things that add more value.

With the adoption of the programme we have gone from taking 30-40 seconds per company to being able to check each company in 2 seconds, which means that we can now make 20 times the contacts that we made by hand and make them better because the human factor is no longer present, so the contacts that reach sales are checked and are of quality.

---

[1] **ERP**: is the system used by the company to store internal customer information, incidents, invoicing and everything to do with them.

[2] **CRM:** is where the sales department has all the contacts they are prospecting, it helps us to organize calls, meetings and have all the information of all the prospects in one place. In our case we use Pipedrive.

Generate info context:
In the module of generating the information two parts are executed, the first one is the same as the previous one, this was done because sometimes we only want to check if we have repeats or not and others we want to see if there are repeats and those that remain we want to know their information, for that reason we put these two modules. The second part, as I said now, is to generate the contact information of the companies.

Also in the module to generate the information you can find the e-mails of the contact persons that you have in the bases, through the link of the company or their name and surname (with the link of the company), about this functionality I will talk more in detail in section 5 b).

This second part of the module is also complicated because what it does is to collect all available contact information such as: postal address, telephone numbers, e-mails, website, contact persons, invoicing and employees through the companies' VAT number. All this information is stored in an excel file and the user can download it in CSV format.

Previously this information was extracted by hand from databases or the internet, doing the companies one by one and copying and pasting the information into an excel spreadsheet. This meant that the department could only do this, as you can imagine it was a tedious and mentally tiring job.

In addition, before the programme, we used to find the fleet size[3] by searching the websites, which made it even more time consuming to generate a lot of contacts in a short period of time.

With the implementation of the app we have gone from being able to manage 150-200 contacts a week to being able to do it in 1 hour, so now no matter how many contacts there are or how many sales people the company has, we can supply the sales department without the need to incorporate more people to make databases manually and without sense.

KPI and SDR calls:
The KPIs of the SDR and sales department are controlled by Market Research, this is because market research has a broader global view than sales so it can see beyond each profile and analyze it properly.

In the application there are different modules to evaluate the KPIs but in the end they all lead to the same conclusions. In the most important KPI module we measure the SDR indicators individually, collectively and save the previous month's KPIs for comparison in the same app.

[3] The fleet size of the companies is of interest to us as Movertis is dedicated to commercialize fleet management software to companies that have vehicles, these companies are billed more or less depending on the number of vehicles they have, so it is important to segment the companies by the size of turnover they are likely to carry.

KPIs come from two main sources: CRM (pipedrive), the virtual switchboard[4] (Aircall). Both softwares stores the department's data and sends it to the app to display it in the most convenient way for us.

Previously, in order to see the same KPIs I had to run a report in Pipedrive, which took about 15 seconds to load and then I had to filter by user if I wanted to see the segmented indicators or filter by team if I wanted to see them grouped.

In the case of Aircall what I was doing was filtering the calls and then profile by profile counting them by hand and then seeing the approximate average durations and calls made by hour.

With the app this process becomes a matter of clicking a button and seeing it in a matter of 5-10 seconds, filtered, sorted and with even more data applied to our use case.

With this implementation we can make decisions in real time as the app shows us the KPIs updated every minute and a half.

We have also incorporated a section where with just one click you can listen to calls that have lasted more than 5 minutes and see in a matter of seconds where you can improve to gain more customers.

Conclusion of the context

As I have explained, making this application not only solved the issue of my TFM but in the company where I am this application, despite not being very complex, has solved the life of a whole team of Market research that has gone from being almost all the working day chopping excels and doing work by hand, to automate it and focus on what really matters, the sales strategy and training of all SDR profiles, in order to generate more business opportunities and the company to grow faster.

[4] The virtual switchboard is the programme from which all the SDR profiles make the calls, this allows us to centralize all the telephones in the same place and above all to centralize all the data generated by the telephone, so that the extraction, cleaning and analysis of the data becomes much simpler.

4. DATA USED

In this part of my master's thesis report I will break down the data I have worked with and the two test notebooks I have used in order to get the most out of one of the fundamental parts of the information generation module, the machine learning model.

To do this we will make a small explanation of the notebooks and especially of the data that I have used to do it. I will also present the datasets that you can use to test the modules, since obtaining data in this project is a bit peculiar.

a. NOTEBOOKS USED

In this project I have used two notebooks to make tests and trials in order to get the most out of the training data, the notebooks can be found in the repository in the notebooks folder or in this GitHub link, the notebooks are as follows:

- analysis_of_input_table.ipynb
- model_tests_fleet_size.ipynb

Both notebooks have their own particular purpose and were both used to determine the Kedro[5] pipelines (one for the preparation of the table for input into the model, and the other to determine which model was the most suitable for the use case on the table).

Analysis of input table notebook:
Although the notebook explains the reasons for each action, I believe it is necessary to do so in a synthesized form in this document and if any doubts remain, it can be consulted for more details.

The purpose of this notebook is to analyze the dataset we have of all the historical companies (with all their information) that we have surveyed since January 2022.

From this file we want to draw several conclusions, first of all, which are the variables that most affect the fleet size, since, if it is the value we want to predict, we need to know the independent variables that affect the value of the dependent variable, in this case the fleet size.

In this case we make different assumptions: the sector of the company, the annual sales and the employees will be important and almost certain independent variables.

At the same time I want to analyze the dataset to see if the variables: has an email (yes or no), has a website (yes or no), what type of phone it has (mobile or landline) and what part of the territory the company is from are important enough independent variables to add to the machine learning model, the notebook explains the reasons why I thought at the time that they might influence our dependent variable.

[5] **Kedro:** It is the technology I used to create the project, thanks to it I have been able to keep my folder system as clean and tidy as possible, I have also used it to generate the Pipelines of the transformations that were made in the tables in all the processes.

The conclusion of this notebook is that the variables that are best related to fleet size are the expected variables (sector, annual sales and employees), and it concludes that the other variables we have tested cannot be completely ruled out even though they do not have the same relationship with the dependent variable.

In short, the notebook concludes that the models should be tested with all variables to understand which one works best and with which variable. Another conclusion I drew from the notebook was that the sector variable had to be narrowed down to 5 possibilities[6], as the ones that fall under the label "other" are those for which there is not as much data to make reliable predictions.

Model test fleet size:
In this notebook what we want to compare are different things, the first one is to see which model is the best to predict the type of dependent variable we have, I also want to check which dependent variables are better to test these models and also compare the machine learning models with the logic[7] that we applied before the app was developed.

The models we are going to test are two types of models: regression models and classification models because the dependent variable can be expressed in two ways: as a numerical value and as a range between two values (this is how we express it in the CRM), this is the example:

Range[8]	Numerical values[9]	Range	Numerical values
1 - 5	2.5	**51 - 100**	75
6 - 10	7.5	**101 - 200**	250
11 - 20	15	**201 - 500**	325
21 - 30	25	**> 500**	500
31 - 50	40		

Table 1: relationship between fleet size ranges and their numerical values
Lluís Pellejà 2023

[6] The final sectors I decided to incorporate were: Transport, Coaches, Distribution and Technical Services. The others were included in the same sector due to the lack of data with which to train the machine learning model.

[7] Before developing the machine learning model and before automating the process in the company we took into account the turnover when assigning fleet sizes, this logic is explained in the notebook but, what we did was that we assigned in each sector a value of turnover per vehicle and it used to go well, to understand how well or bad the models go we will compare it to our "handmade" model.

[8] The fleet size ranges allow us to easily classify companies into different ranges and thus understand what size of company we are targeting.

[9] To represent the fleet size as a numerical value, we take both values of the range and apply an average to them. I understand that it will not be the exact fleet value of the companies but it will give us an approximate idea of what range the analyzed company is in.

In the notebook what we try to test is first of all which type of model is best for the types of data we have, what we see when testing the regression models is that despite passing the fleet sizes to numerical values, they don't work quite right.

This result is due to the fact that the independent variables and the dependent variable do not have a strict linear relationship, so in a first set of tests we already see that the classification models will be the best positioned to handle this type of data.

After testing the classification models I saw that they were the correct models, I simply passed the encoder to the dependent variable of the training set and it matched perfectly, so now that I had found the type of model I needed I had to test the three models that define classification models par excellence: Decision tree, K-nearest neighbors and the Random Forest.

Once I was clear about the models, I prepared the tests with all the combinations of variables that we decided on in the previous notebook and I ended up deciding on the Random Forest, as it was the one that showed the best performance in equal hyperparameters.

Once the machine learning model was chosen I had to find the optimal hyperparameters for my data, for this I used the GridSearchCV library to make the necessary iterations to find the hyperparameters that best fit my data and made the best prediction so I could choose the hyperparameters for my model.

In order to get my doubts out of the way and explore all the options, I tested the GridSearchCV library in all the classification models and it turned out that in the Random Forest model it was still the one that scored the highest, so I decided to use this one.

As an advancement to the testing since we have been using this technique to predict fleet size we have received less complaints from the SDR department on this part so after 3 months of implementing this model on fleet size I can say that it is working very well.

The conclusions we draw from this notebook are that the best decision based on the data and now on the results was to choose the Random Forest model as we did at the time, the notebook helped us make that decision and it was the right one.

b. DATA TO ANALYZE THE INDEPENDENT AND DEPENDENT VARIABLES AND TRAIN THE MODEL.

The data to analyze the independent variables and the dependent variable is the file called *raw_input_table.xlsx*[10]. This file contains all the companies prospected by the SDR team since January 2022, with a total of 5,051 company records with the information filled in and most importantly the fleet size.

[10] The shared file is confidential and can only be seen by the people with whom it has been shared, and its dissemination is completely forbidden.

This data set has 27 columns but we are not going to use all of them, we will keep the following:

Column name[11]	Value type	Column name	Value type
Name	string	Fleet size	string
Sector	string	Region	string
CNAE code[12]	int	Phone	int
Annual sales	int	Mail	string
Employees	int	Website	string
Mean fleet size	int		

Table 2: columns of file raw_input_table.xlsx and their value type
Lluís Pellejà 2023

In the file we find everything we need to analyze the impact of all the independent variables and to reach interesting conclusions. This file keeps growing week by week as we add more companies to the database, so the model is trained every week with more data.

The aim is to reach a point where all sectors can be independent of each other and where the base is so broad that there is very little wrong with the prediction. The company believes that we will be able to reach the target by the end of this year, which is why the model is constantly changing and evolving on a weekly basis.

c. DATA TO CHECK THE FUNCTIONING OF EACH MODULE

The files I have shared to test the two modules of the app are focused on companies in the sectors that interest us. You will see that I have prepared several files so that you can see that the app is agile and all the commands work.

Duplicate comprobation file:
To test the duplicate management module I have added a file called "*check_duplicates.xlsx*[13]", in this file you will find 15 companies, which when passing them through the duplicate check should be 11 (tested on 07/10/2023, as the result may vary depending on what we have imported into the CRM during the last few days).

[11] In the file, columns are written in Spanish, this is due the fact that the file is used by the company and everybody has to understand it.

[12] CNAE codes are in Spain the number code to differentiate each sector and activity from others that could be similar.

[13] The shared file is confidential and can only be seen by the people with whom it has been shared, and its dissemination is completely forbidden.

In the file of the good companies there should be two extra columns, one with the reason for the loss of the companies that have been imported, and the days that they have been lost.

The process for testing the information generation module is very simple:

1. *You enter the app*
2. *In the side panel select "Check for duplicates".*
3. *Drag the file over the importer*
4. *Click on the button "Start scanning".*
5. *Wait for the results to be displayed and ready to download*

Generate the information file:
To test the module to generate the information of the companies I have attached a file called "*generate _the_information.xlsx*[14]" This file is 14 companies that we want to prospect so, we need to see if there are any repeated and if they are not we want to contact them so we will need their information.

The process for testing the information generation module is very simple:

1. *You enter the app*
2. *In the side panel select "Generate information".*
3. *Drag the file over the importer*
4. *Click on the button "Start scanning".*
5. *Wait for the results to be displayed*[15] *and ready to download*

Generate mail with name, last name and link:
To test the functionality to generate the mails of people with their name, surname and link of the company where they work, I have added a file called "*get_mails_name_lastname_link.xlsx*[16]", with this file you can test the module without problems.

You will see that it does not have all the mails of the people, it is the most common, but this module helps us to improve the contactability in companies that are very hermetic.

The process for testing the information generation module is very simple:

1. *You enter the app*
2. *In the side panel select "Generate information".*
3. *Select the 2nd tab*
4. *Choose the first option*
5. *Drag the file over the importer*

[14] The shared file is confidential and can only be seen by the people with whom it has been shared, and its dissemination is completely forbidden.

[15] The results only show the companies that can be prospected, so if any of the companies have been imported into the CRM during the course of these weeks, they will not appear in the results.

[16] The shared file is confidential and can only be seen by the people with whom it has been shared, and its dissemination is completely forbidden.

6. *Click on the button "start generating mails".*
7. *Wait for the results to be displayed and ready to download*

Generate mail with link:
To test this module I have added a document called "*get_mail_link.xlsx*[17]" that contains the urls of different companies, so that with the module you can find the mails of the most important people in the company.

This is the module that is less used because it returns all the data mixed and needs to be polished but it is good to see the size of the company and the people who work there, or get an idea of what pattern their mails follow.

The process for testing the information generation module is very simple:

1. *You enter the app*
2. *In the side panel select "Generate information".*
3. *Select the 2nd tab*
4. *Choose the second option*
5. *Drag the file over the importer*
6. *Click on the button "start generating mails".*
7. *Wait for the results to be displayed and ready to download*

Get all the mails from a link:
To test this module you only need one link, as it is focused on searching the mails from a single link and sorting them in the same application so you can copy the one you are most interested in, some suggestions are the ones in the previous file, as it will show you the results sorted in a table.

The process for testing the information generation module is very simple:

1. *You enter the app*
2. *In the side panel select "Generate information".*
3. *Select the 3rd tab*
4. *Write the URL in the textbox*
5. *Click enter.*
6. *Wait for the results to be displayed*

[17] The shared file is confidential and can only be seen by the people with whom it has been shared, and its dissemination is completely forbidden.

5. THE FUNCTIONING OF THE APP

- *a. DUPLICATE COMPROBATION*
  - i. Objective
  - ii. Functioning
  - iii. Final output
- *b. GENERATE COMPANY INFO*
  - i. Objective
  - ii. Functioning
  - iii. Final output
- *c. KPI AND DASHBOARDS MODULES*

6. CONCLUSION