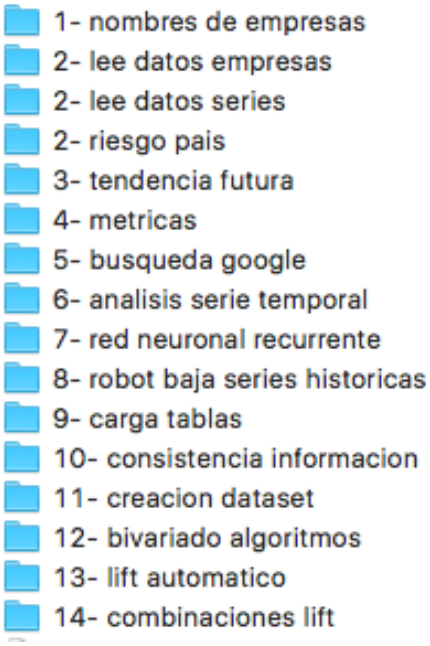
Detail of algorithms - Thesis project

Used tools: Python, R, SQL, Excel, Sublime.



1-nombres empresas

I get the names of the companies in the stock market and they are copied to a csv file "**nombre_empresas.csv"**

2- lee datos empresas

In **scraping.py** I download quotes of the stock market from a particular website.

2- lee datos series

In **scraping.py** I download series of the central bank to analyze the correlation with quotes (interest rates, monetary reserves, bonds).

In **metricas_bcra.csv** we can see the result of the joined metrics by day.

2- riesgo país

Data source of country risk to download.

3- tendencia futura

In **tendencia.py** I create a regression that adjusts to future values to take the trend in coming days, so I obtain a continuous target to analyze.

4- metricas

In "**creación de métricas.py"** I create metrics based on the history of quotations closings.

5- búsqueda google

I create a robot that performs a search by company, I used cromedriver to simulate human behavior.

It recovers every day in the past the pages in which the company was named. For each page it iterate and makes a copy paste of everything that it finds in the display (if we only capture the source code we lose valuable information that is why chromedriver is used and also you avoid several regular expression coding).

Once the text is copied, a bag of words is made to analyzing how the social context influences the fluctuations of the company's stocks. Unwanted pages are removed by performing a matching by regular expressions taking into account the context of the searched word.

6-Analisis serie temporal

In **descomposicion.R** the time series is broken down into trend, seasonality and random components.

In **lag_overPartition.sql** we have the creation of lags and trends by SQL.

In **series.py** it graphic the series.

The goal is to find correlations between time series by varying the lag of them.

7- red neuronal recurrente

In "prob_dataset_pampa_petb/**iterador.py"** I created a parameter iterator to train several recurrent neural networks and keep the one that best fits to the problem.

8- robot baja series históricas

Historical download of three different data sources to ensure data consistency.the robot goes over different web pages simulating human behavior using scraping and webdriver.

For example in "bajada de invertironline**acciones_Arg.py"** we can see the downloading of information from the webpage invertironline.

9- carga tablas

Load the sources downloaded in the previous point into a MySQL database. There are several tables, making the upload manually would take a long time so I automated the creation of tables with Python.

For example in "investing/**python crea script carga.py"** for each data source corresponding to the quotation of a stock, a corresponding script is created for the upload to MySQL. You can see the result of the query in **salida_script.txt** ready to execute it in Mysql.

10- consistencia información

Consistency checks from previous sources in SQL queries.

11- creacion dataset

Create metrics with new dataset in "**creacion de metricas.py"** observing history.

12- bivariado algoritmos

Algorithm that iterates creating combinations taken from two variables. For each set of two variables, an SVM transformation is applied to find a new dimension that can separate the problem (rbf, sigmoid, linear). The new variable contains probability information that can be treated continuously in another algorithm.

You can see the test code in **"transformacion de variables.py"**

13- lift automatico

In "**Cruce Variables - COBRANZAS.xlsx"** I create a metric to select variables taking into account the lift in each subset respect to the total share to evaluate the force of discrimination.

In **lift.py** the previous logic is replicated to automate it in python.

14- combinaciones lift

The goal of this script is to find sets that represent a lift greater than the total set to identify action groups, deviations, or anomalies.

For example if we are in a bank and the default rate increases but we do not know why it is due. The goal of this algorithm is to perform combinations of variables and calculate the default rate in each subset to identify if there is growth or deviation from the global metric in a subset.

In **prueba_transformaco5.py** you can see the tests, there are performance problems by number of variables and hardware.

15- transformacion correlacion

In “**en masa11.py**” I do an analysis by correlations to eliminate variables with little information when we have many variables.

Categorical variables are subjected to a transformation by mean difference so that they can adjust with information from the target. It is also done a transformation by creating dummy variables.

Having these transformations of categorical variables and adding the continuous ones, we can apply correlations with the target. If the absolute value of the correlation with the target is greater than 0.2, the variable is saved in a list.

Then with all the variables in that list, it’s made the correlation between each other to eliminate the collinear ones. After this process we have a set of variables for a regression avoiding collinearity.