Sweet, sweet Data Science

*Predicting factory demand of a bakery*

**By**

Miguel Ruiz Nogues

Illán Lois Bermejo

[ÍNDICE DE CONTENIDOS: 2](#_Toc29212305)

[Introduction 3](#_Toc29212306)

[Raw Data 3](#_Toc29212307)

[Tech-suite 3](#_Toc29212308)

[Preditive models used 3](#_Toc29212309)

[Project re-execution guidelines 4](#_Toc29212310)

[Main issues encountered 4](#_Toc29212311)

[Results 4](#_Toc29212312)

[Conclusions 4](#_Toc29212313)

# Introduction

El proyecto consiste en predecir los pedidos a un obrador de pastelería por parte de las tiendas de la misma cadena.

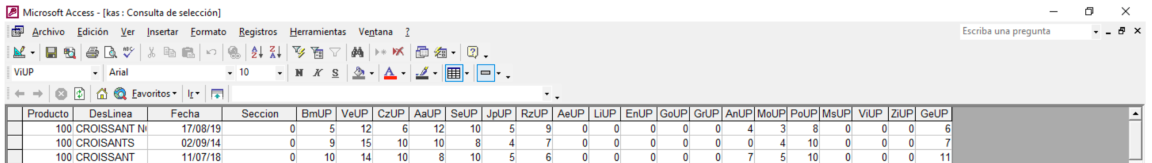
La idea de este proyecto surgió debido a la necesidad de dicho obrador de poder mejorar sus predicciones para poder planificar mejor las cargas de trabajo y compras de materias primas.

En la actualidad, los métodos utilizados se basaban en el conocimiento experto de sus empleados, sin utilización de series históricas a pesar de almacenarlos.

Nuestro objetivo es, por lo tanto, optimizar las predicciones de la demanda.

# Raw data

1. Name\_of\_file: Histórico de pedidos al obrado desde 01/01/2008 hasta 30/09/2019. El archivo contenía 23 columnas y 1607908 filas.



*Pantallazo proporcionado por el cliente con los nombres de las columns.*

1. Localización de las tiendas stores.csv
2. Otras fuentes de datos:
   1. Partidos de futbol
   2. Dias Festivos

# Tech-suite

* **Main tech**: Python, Tableau & Github.
* **Standard tech \***: Pandas, Numpy, Seaborn, Matplotlib, pending complete,
* **Non-standard-tech:\***: Scikit-learn, Git Large Files, pending complete,
* **Supporting tech**: Jira, Slack, Webex, Scikit-learn, virtual enviroments, pair programming, etc.

*\* Full list of libraries included in the repository under*

# Preditive models used

* **Exponential Smoothing:** rationale for trying
* **Arima:** rationale for trying
* **Sarima:** rationale for trying
* **Prophet:** rationale for trying
* **Random Forest:** rationale for trying

# Methodology & re-execution guidelines

In a nutshell what we did is, after obtaining the data from a real client, we cleaned for 10 products, forecasted for one of them, and created a visualization based on the 10 products.

But of course, describing the project in this two lines is like describing google as a this simplified text-box and a button that searches webs based on that button… Simple, right?

Here is a more detailed view of the steps we took:

* **Data acquisition**
* **Data cleaning & preparation**
* **Analysis**
  + Dickey-Fuller
* **Front-end & visualization**

**RE-EXECUTION**

The easiest way to re-execute is to execute the notebooks in the following order:

1. …

Note that note all the notebooks are part of the base-line, there are some notebooks that where yoused in parallel to gather some information.

Include all notebook names

# Results and main issues encountered

1. Results of the models.
2. Time spent…

**MAIN ISSUES**

* Data cleaning
  1. Descriptions were manually introduced – we solved by creating a dictionary of words included in the online catalogue, and using fuzzywizzy normalizing product description by product description, and word by word.
  2. Relationship N-N product\_descriptions and product\_id. In other words, one
  3. One product id corresponds to N products.
* Data reality:
  1. After cleaning the data, we observed a big number of anomalities, such as orders of thousands of croissants in one day, or 0 orders from a product for months. Meetings were required to understand the reality of the data. – we solved by asking the client for each of the abnormalities.
* Adding Modin to speed up Pandas
* Prophet bug with matplotlib.
* Integrating tableau with TabPy.
* Blocking master on Git:

# Conclusions & Improvements

1. Improvements… Additional work.
2. Rationale for the decisions.

* **Conclusion 1:** A data scientist really spends 80% of his/her time cleaning data.
* **Conclusion 3:**
* **Improvement 1:**

Random forest requires