0. Introduction:

This project is aimed to put some light upon the problem of predicting which of the incoming projects and their budgets are accurate scheduling the end of the construction and its resources. The initial issue to solve is to get valid data of real constructions with their delay reported.

Of course, large construction companies have huge lists of observations of this kind. But in this sector local circumstances are highly relevant, like the socioeconomic moment or the location of each construction process, as they affect to viability, prices and HHRR. So, even for these companies, having big "clean" data doesn't mean that this data will be helpful without expert data preprocessing.

1. User manual for the frontend.

The GUI application (graphical user interface) is in the "Interface" file.

*** For WINDOWS users:**

Execute or launch "Contadex.pyw". If it doesn't work, check:

- TKinter installation.
- Ensure Python 3 is executing the extension ".pyw".
- If none of this works, try to python-launch "Constadex.py" from a console with python available.

*** For UBUNTU users:**

Launch "Constadex_ubuntu.py" form a console.

2. Raw data description:

As an Expert Model, the relevant raw data is provided by the Data Scientist. This is an strategic decision that helps to use the scarce data from the field effectively as testing data.

Taking into account that the Data Scientist on command for this study is an Architect and works as Project Manager in the construction sector, we expect that his experience is valuable for creating a rich and expert dataset with observations of good and bad constructions characteristics in terms of its delay.

The method used for creating this Train Dataset is a controlled normal distribution (using "numpy.random"). Variables are controlled by restricting the "centre" of the distribution and its standard deviation. Of course, every normal distribution captures an intuition of "good" or "bad" characteristics in terms of project planning.

D**ata proportions:**

- Train Data: 3530 observations
- Test Data:223 observations

Variables:

- “built_area”: Construction area, summing floors in case of a multi storey building.
- “modul_price”: Construction rough cost by square meter.
- "weeks_duration": Total duration of the construction project in weeks.
- “weeks_delay”: Total delay of the construction project in weeks.
- "typology": Typology of the building. We have simplified this categorical variable in COMMERCIAL, COLLECTIVE, DETACHED and OTHERS.

3. Methodology:

Data compilation, plotting and expert analysis using:

- **Jupyter (IPython).**
- **Matplotlib.**
- **Plotly.**
- **Pyplot.**
- **Seaborn.**

For predictive analysis we have used three different tools that, used together, will help to both reduce overfitting and take advantage of local fenomena. Supervised Machine Learning Algorithms used (by SciKit Learn):

- **Logistic Regression.**
- **Random Forest.**
- **K-Neighbors.**

User interface developed specifically for this project with:

- **TKinter.**

4. Summary of main results.

The concept "True delay" depends on the delays and the duration, assigning a threshold based in the minimum proportion of 0.15 times the total duration of the construction project to consider it a TRUE delay. So, the threshold is put on a new boolean variable “DELAYED”, the one used as target.

We have increased accuracy by 2% over the most accurate algorithm alone (68.6% acc Random Forest) by giving each of the algorithms the right of flagging the project as a “possible delayed project”. But this strategy obviously tend to overfit the model, reducing its robustness.

We have trained a ML Ensemble model to detect Delays in a construction only with some previous conditions of the construction contract. As the Train Dataset have more proportion of “DELAYED” observations, this machine will tend to over detect false positives. In this case, that behavior is not casual but intentionally programmed by the Data Scientist:

1. Statistically, in Spain most construction projects have delays. It is completely natural that the testing dataset express that reality. So, if we forget about false positives we will take advantage of local singular conditions of the model without overfitting.
2. In practice, it is profitable to force the algorithm to be very propense to predict delays anyway. Only putting an alarm upon construction managers we can make them to give extra effort to find those 20% of the causes of those 80% of the possible delays.
3. Even if the user of the model suffer the bad luck of having a false positive in his project (which is a construction project that wrongly activates the alarm of possible delays), it will have little repercussion in the project in relation with the possible cost of a true delay.

For the graphical user interface we have put a conservative threshold in the voting script so the program could generalise better. It is ment to avoid one algorithm can flag a project as a “possible delay” by itself. So two of them must agree to flag the input project as a “possible delay. This improves the model’s robustness at expense of its accuracy.

5. Conclusions.

This study and the resulting tool would be helpful for a “second opinion” in management auditions.

Due to the changing socio-economic variables (material and human resources prices and fluctuations in the building market), **the data has a short-term validity**. So it is strongly advised to have a maintenance plan for this kind of models. The maintenance should be driven by an expert in Data Science with experience in the construction field.