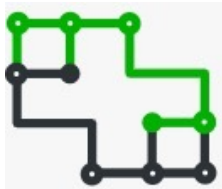


Challenge Data

ENS | PSL

COLLÈGE DE FRANCE 1530

INSTITUT Louis Bachelier

Real estate price prediction – Challenge by Institut Louis Bachelier

Paul-Emile Dugnat –

Doctorant CIFRE AXA-IM – Telecom ParisTech

Presentation of the Challenge

- Goal of the challenge is to **predict houses prices** using :
  - Classic **features** likes location, surface, land, property type, number of bedrooms, etc.
  - **Images** of the houses (between 1 and 6 images)

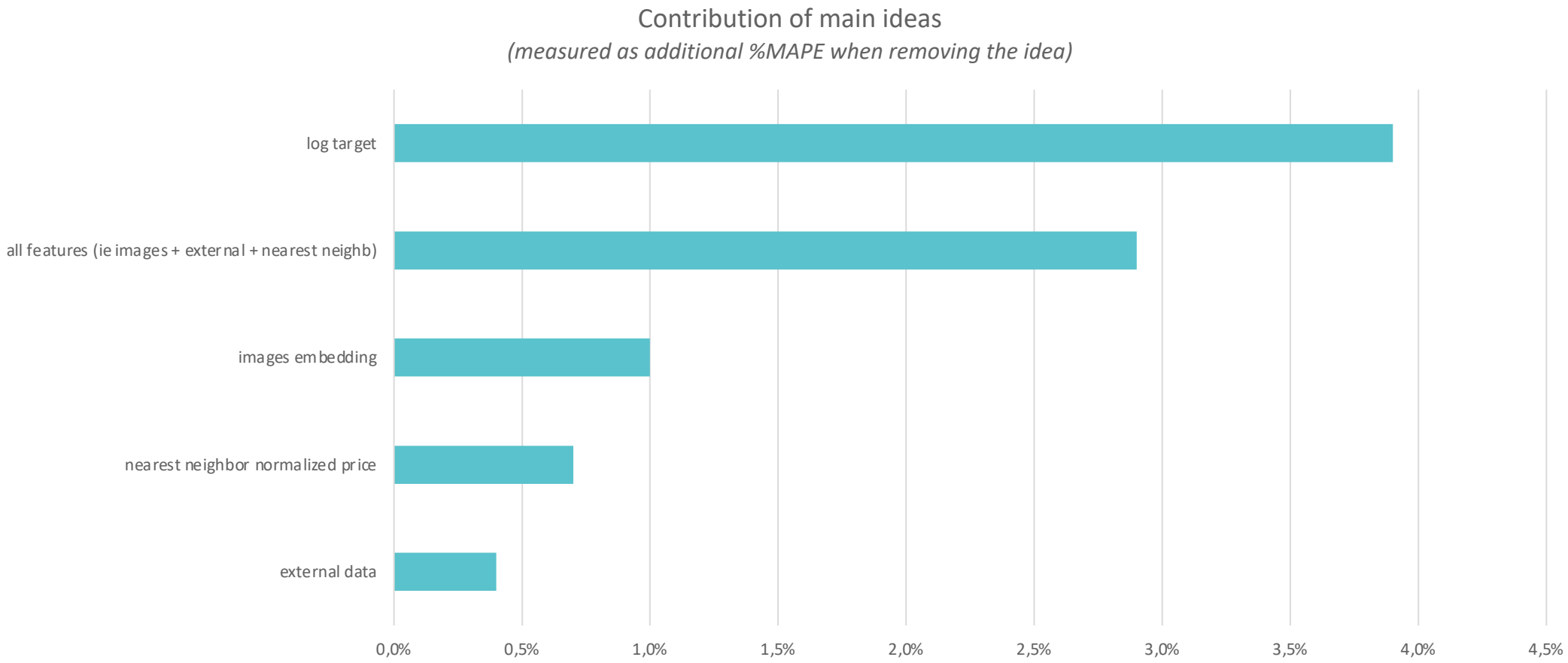
- **Specificities**:
  - Metric is **MAPE** (so we want to optimize for relative error)
  - Use information from tabular and image data

- Code and current slides are available at https://github.com/pedugnat/ilb-data-challenge-2022

Overview of the solution: Feature Engineering

- **Classic feature engineering**
  - Simple ratios & interactions between features: size of rooms (ie size / number of rooms), size over landside, number of room per type of property, number of rooms per city, size vs mean city or department size, etc.
- **Magic feature**
  - Most important single feature: custom sklearn transformer computing the mean price per square meter in a given radius from the house (ie feature is: mean price per square meter for all houses closer than *n* meters from current house)
  - Computed on train set, using latitude and longitude features, for values of *n* ranging from very low values (useful in cities) to pretty large values (useful in less dense areas)
- **Images embedding**
  - The spirit of the challenge was to see how incorporating features of images could help price prediction
  - Embed all images using classic ResNet/EfficientNet, then predict the price of the property using image embeddings
  - Use image price prediction as a feature for modeling
- **External data**
  - Demande de Valeur Foncière open data: compute mean price per square meter for the city
  - INSEE open data on revenues
  - Not super useful overall but it makes sense to use them

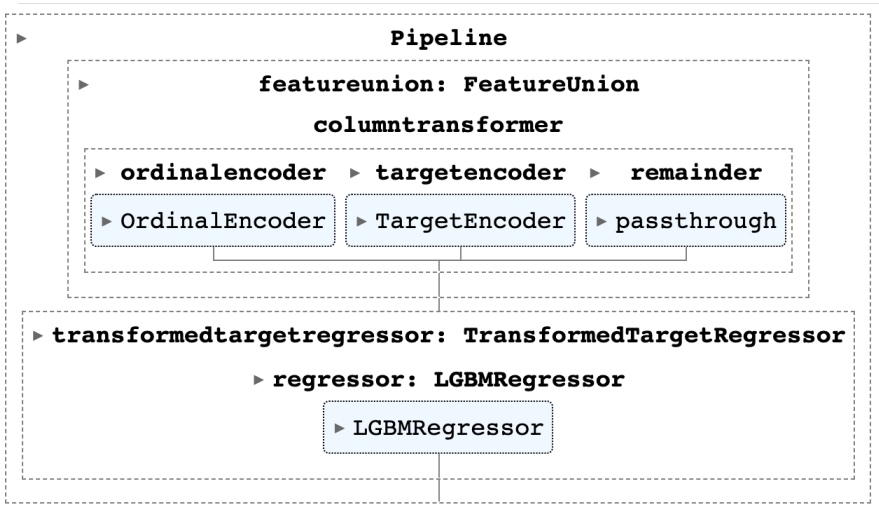
CONTRIBUTION OF IDEAS



Overview of the solution: Modeling

- **Modeling**
  - Model was a simple **LGBM** (LGBMRegressor) with (manual) parameters tuning
  - Part of the magic: predict **log target** rather than target directly (using sklearn TransformedTargetRegressor), probably because of the metric (MAPE)
  - LGBM proved to be much more effective than XGBoost and sklearn implementation, probably because of the number of categorical variables
  - Used **sklearn pipelines** capabilities all along to keep code simple and modular, although it comes with some drawbacks (less inspection possibilities for the models/less interpretability, but that would not be very complicated to reimplement)
  - No ensembling/stacking of models (bring very little benefit to the final score)

- **Validation setting**
  - 10 folds, using sklearn cross_validate function
  - Folds are run in parallel
  - Takes ~10 minutes to fit on a laptop (so rather quick)



WHAT CAN WE REMEMBER FROM THIS CHALLENGE?

- **General principles**
  - Understand the metrics (MAPE is very specific)
  - Visualize the data to understand the target variable, the features, missing values, etc.
  - Understand the data, its nature, what it means in the “real world” (normalizing by square meters was really helpful in our case)
  - Several “orthogonal” good ideas are needed to have a good model

- **About this challenge**
  - A lot of different sources of information : numerical & categorical features, images, external data
  - Validation was super reliable, which makes it pleasant to work on
  - Overall, very enjoyable challenge, so a big thanks to the organizers!