Hacking Global Health

Ebury DS Team
enrique.colin@ebury.com
inigo.cortajarena@ebury.com
vicente.laiseca@ebury.com
antonio.malpica@ebury.com
pedro.morales@ebury.com

Contents

Getting started:

- Data set with 17370 ultrasound measurements.
- We just focus on pre-birth measurements that contain all ultrasound information and target weight.
- **Duplicated measurements** (very close in time) **for the same baby are removed** to avoid training leakage bias.
- The data set results in **6841 measurements**.

Validation methodology

- Conventional train/test split and Cross-validation strategies are based on independent identically distributed data hypothesis (i.i.d.).
- Since some babies have several measurements, the previous hypothesis is not fulfilled (dependent samples).
- Therefore, babies with samples in the validation fold should not be present in the training set and vice versa; otherwise the model would potentially overfit certain subjects (avoided using ***Group K-Fold***)
- This ensures that a model trained on specific groups (i.e. babies) generalises well to new subjects.

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html#sklearn.model_selection.GroupKFold

Modelling

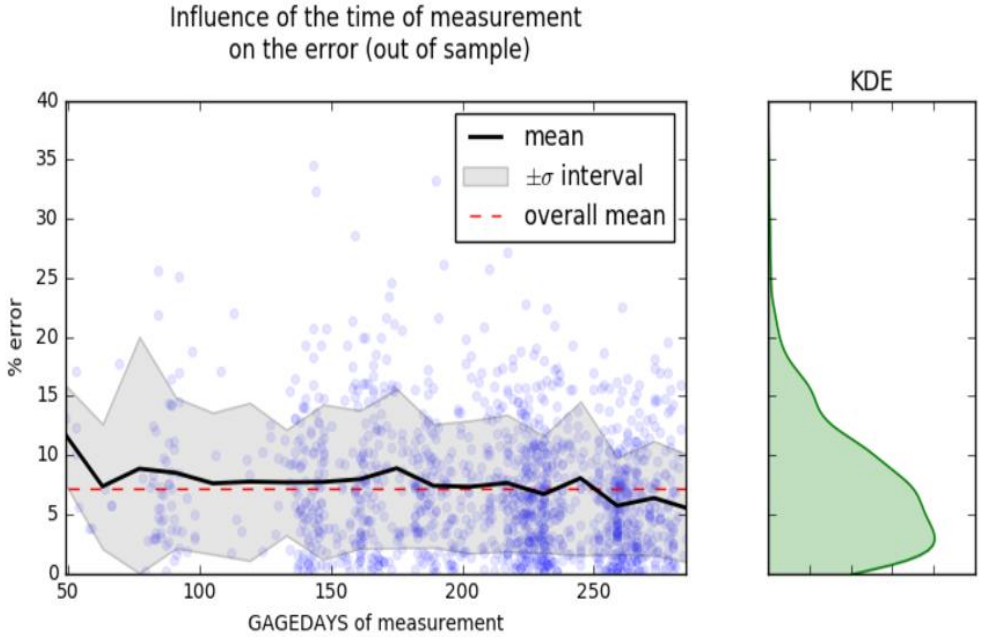
- Objective: approximate the unknown function $\mathcal{F}(\bar{x}) = y$, where $\mathcal{F}$ is a function that models the relation between input features and baby weight.

- For this purpose, we use an ***ensemble of boosted trees***. The XGBoost Python library is used, in particular the **XGBRegressor** model. Link to original paper and slides.

- The % error on the test set and cross-validation folds are computed and the influence of the time of measurement on this error is depicted.

First model

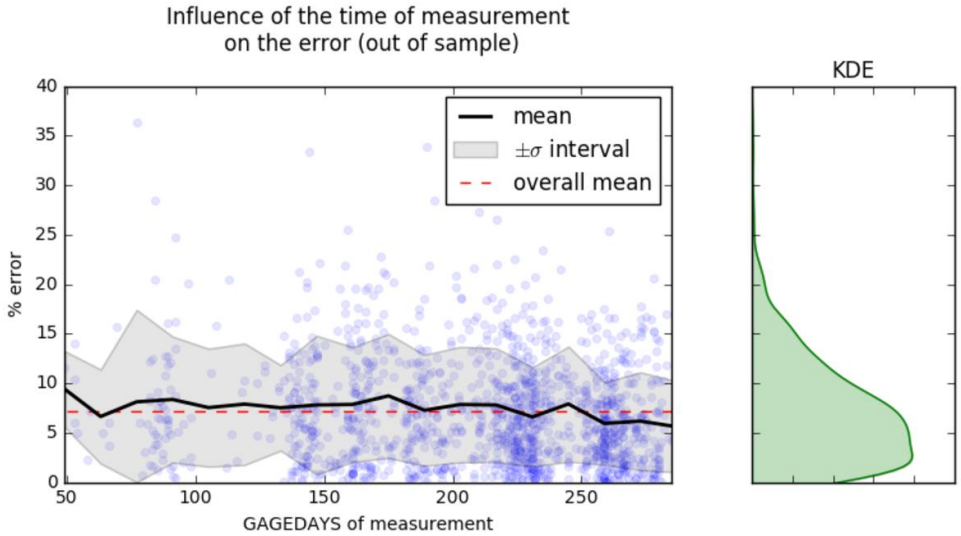
- Uses default variables as a benchmark $\mathcal{F}$*(ultrasound measurements, GAGEDAYS, SEXN, PARITY, GRAVIDA) = BWT_40* (***weight***)
- Test mean absolute error: 0.23215 kg.
- Mean absolute percentage error: 0.0718 (7,18%)
- Script: benchmark.ipynb



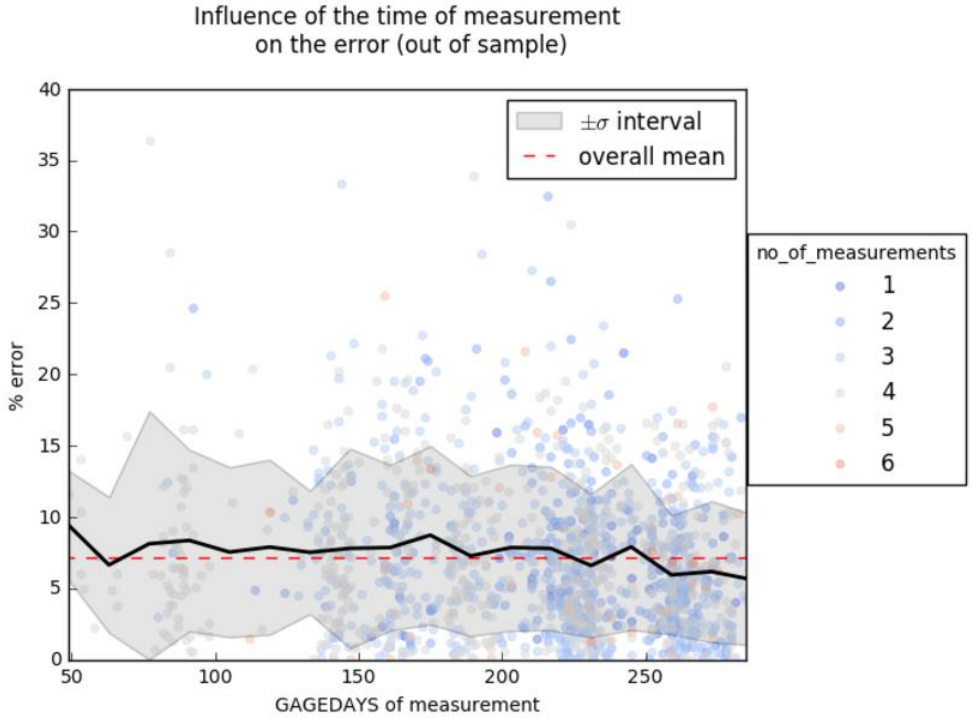
Second model

- Several rows per baby, each row contains the current measurement together with the previous measurement as a new variable.
- If there is not previous measurement, it is filled with NA. ***XGBoost deals with missing values*** (sparse aware).
- Test mean absolute error: 0.23169 kg ***(not much improvement)***
- Mean absolute percentage error: 0.0717
- Script: xgb_n_row_per_subj.ipynb

- *$\mathcal{F}$(ultrasound measurements, previous measurements, GAGEDAYS, SEXN, PARITY, GRAVIDA) = BWT_40*



Second model



- Error is independent of number of measurements on a certain baby.
- Including previous measurements doesn't seem to improve the model.
- The ensemble learns dependency between time of measurement (GAGEDAYS) – good for real life application. ***No need for previous measurements.***

Third model – FINAL MODEL

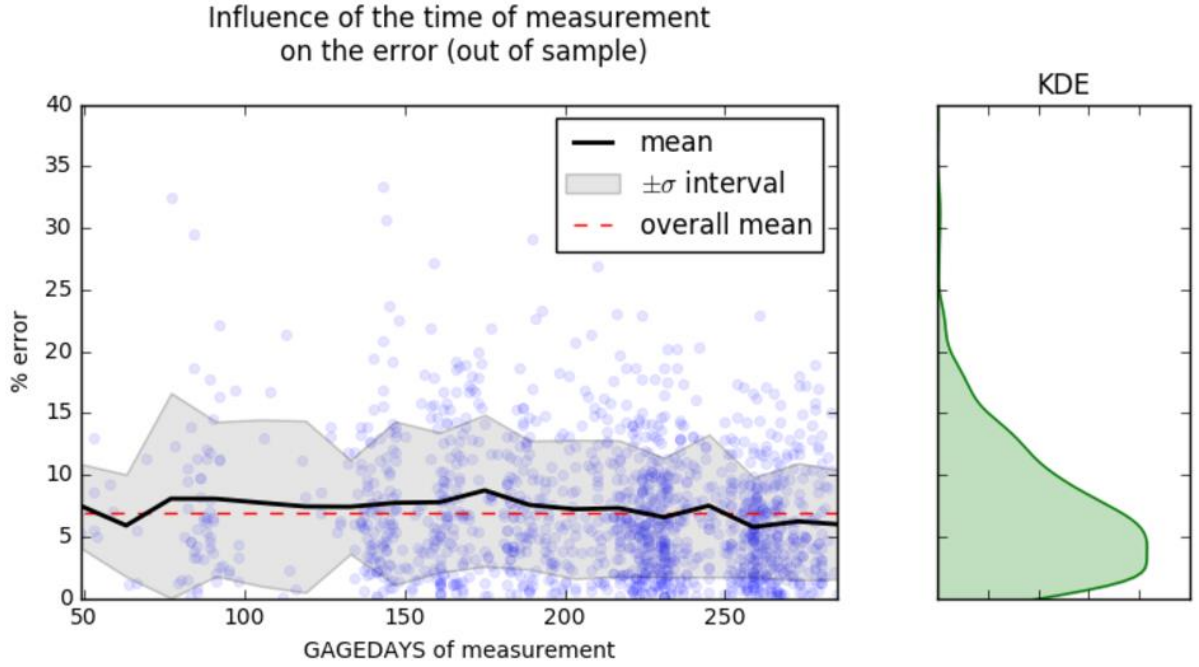
1. ***We engineer more features*** based on geometrical ratios, time-scaled variables, interactions, etc.
2. **We use these features + original ones to approximate the function $\mathcal{F}$.**
3. **We try to predict log of output (like current models do).**

$$\mathcal{F}(\bar{x}) = \log(y)$$

- Test mean absolute error: 0.2277 kg.
- Mean absolute percentage error: 0.0704 (7,04%)
- Script: **xgb_feat_eng.ipynb**
- **Lower error variance overall** (especially for low GAGEDAYS)**.**

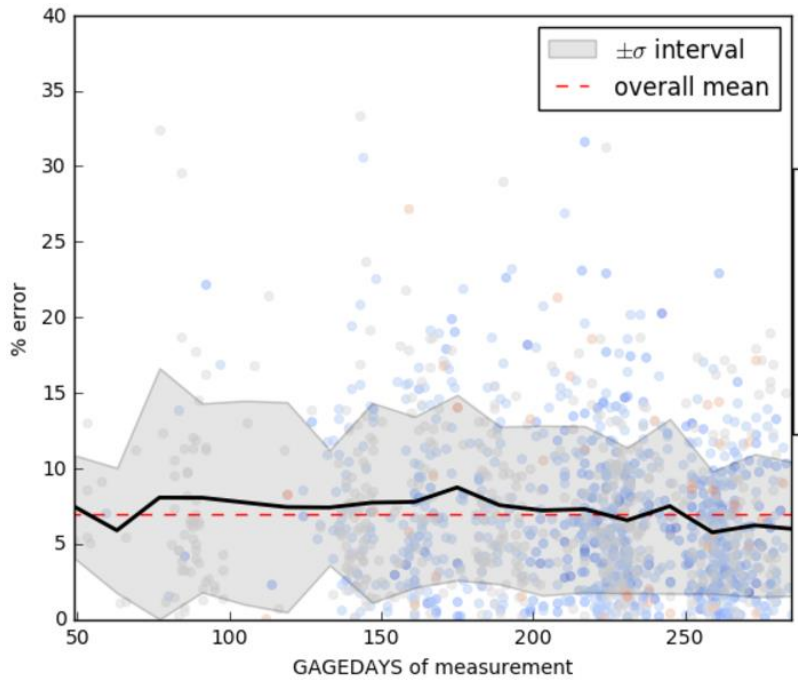
Third model – FINAL MODEL

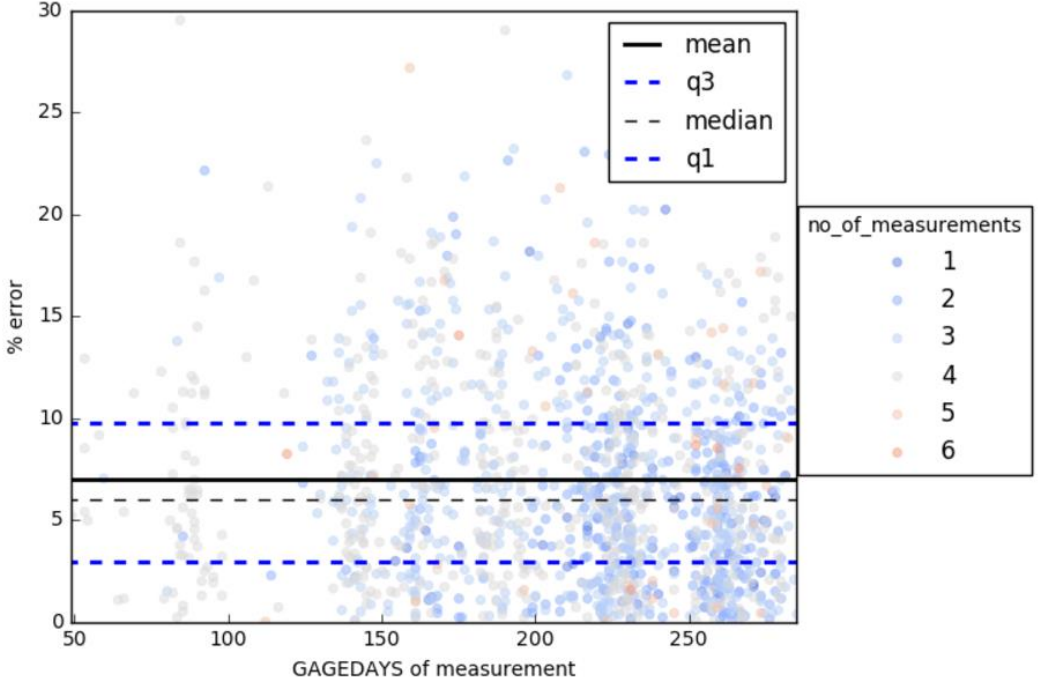
- Lower error variance for earlier measurements (high density below 10%).
- Lower overall mean error (7%).



Third model – FINAL MODEL

- The third quartile regardless the time measurement is below 10% error.





Conclusions

- Summary of errors:

		Model 1	Model 2	Model 3
Cross-validation	Mean absolute error (kg)	0.2333	0.2335	0.2321
Test	Mean absolute error (kg)	0.2321	0.2316	0.2277
	Mean absolute percentage error (%)	7.18	7.17	7.04

- Model 3 (with feature engineering) improves the error, and reduces variance for early measures.

- Using group cross validation is a key action, otherwise the model would overfit certain subject instances and would generalise poorly.

Further actions

- Include the effect of time derivatives of the derived features.
- Extend dataset, would allow to find the optimal spot on the learning curve.
- Add auxiliary features, related to family history, parent information, relevant and easily obtainable genetic data, for instance, merging with other datasets.