A practical Implementation of Machine Learning Methods for Price Imputation

13th May 2024

Lucien MAY, Claude LAMBORAY,

Botir RADJABOV, Yu-Lin HUANG

STATEC

Agenda

1. Methodology – Price Statistics

- **Data source:** Transaction data or web scraped data with prices and characteristics
- **The set of items develops over time**
  In the comparison periods $t1$ and $t2$, we have:
  - Matched items $M_{t1,t2} = N_{t1} \cap N_{t2}$
  - New items $N_{t1,t2} = N_{t2} \backslash N_{t1}$
  - Disappearing items $D_{t1,t2} = N_{t1} \backslash N_{t2}$
- **Index on matched items:** Törnqvist index

$$I_T^{t1,t2} = \prod_{i \in M_{t1,t2}} \left(\frac{p_i^{t2}}{p_i^{t1}}\right)^{0.5*\left(\frac{e_i^{t1}}{\sum_{j \in M_{t1,t2}} e_j^{t1}} + \frac{e_i^{t2}}{\sum_{j \in M_{t1,t2}} e_j^{t2}}\right)}$$

- **Dynamics product assortments:**
  a matched model approach that only takes into account products available in the two comparison periods may not be fully satisfactory (relaunches, shrinkflation, life-cycle pricing)

1. Methodology – Price Statistics

- Our approach to the dynamic item universe is to **replace the missing prices with imputed prices** in order to obtain a full data set for the two comparison periods
- We denote an imputed price for item *i* in period *t* by $\hat{p}_i^t$
- This leads us to the **single imputation Törnqvist** price index

$$I_{IT}^{t1,t2} = \prod_{i \in M_{t1,t2}} \left(\frac{p_i^{t2}}{p_i^{t1}}\right)^{0.5*\left(\frac{e_i^{t1}}{\sum_{j \in N_{t1}} e_j^{t1}} + \frac{e_i^{t2}}{\sum_{j \in N_t} e_j^{t2}}\right)} \prod_{i \in D_{t1,t2}} \left(\frac{\hat{p}_i^{t2}}{p_i^{t1}}\right)^{0.5*\left(\frac{e_i^{t1}}{\sum_{j \in N_{t1}} e_j^{t1}}\right)} \prod_{i \in N_{t1,t2}} \left(\frac{p_i^{t2}}{\hat{p}_i^{t1}}\right)^{0.5*\left(\frac{e_i^{t2}}{\sum_{j \in N_{t2}} e_j^{t2}}\right)}$$

1. Methodology – Price Statistics

- We use a **multilateral method** instead of a bilateral price index: **Imputation GEKS** (or imputation CCDI)

$$I_I^{t1,t2} = \prod_{k \in T} \left(I_{IT}^{t1,k} * I_{IT}^{k,t2}\right)^{\frac{1}{|T|}}$$

- **Webscraped data:** expenditures are not available; we use the **imputation Jevons** and imputation Jevons variant of the GEKS index.

2. Methodology – Machine Learning

- **Impute missing prices using ML models on pooled data**

$$\ln\left(p_i^t\right) = f\,(z_i\,, t) + \epsilon_i^t \qquad \forall i \in N_t, \forall\, t \in [1, .., T]$$

- **Standard log linear model** with time dummies as a **benchmark model**

$$f_{LIN}(z_i, t) = \alpha + \sum_k z_i^k \beta^k + \sum_t d_i^t \gamma^t$$

….compared to two common tree-based ML methods:

Random Forest and **XGBoost**

Data driven approach

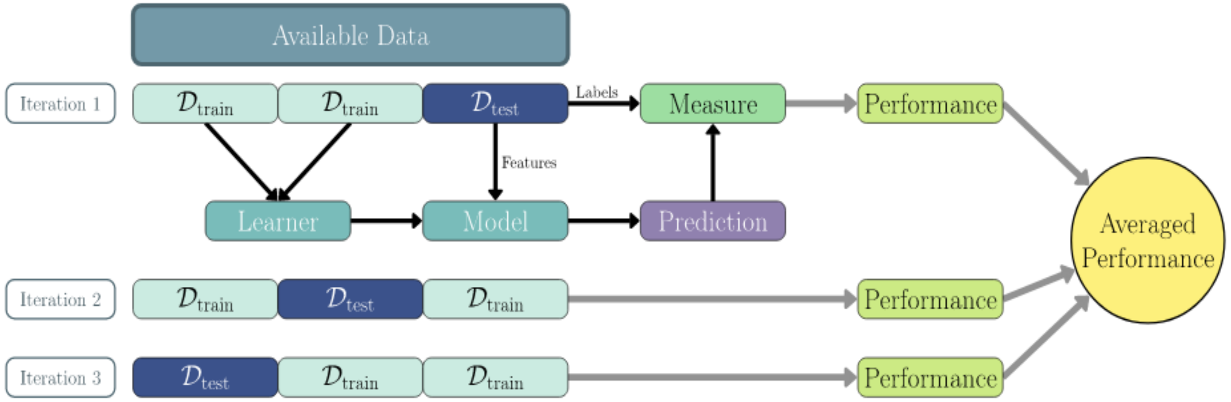
2. Methodology – Machine Learning

- How to measure the **performance of a given model**?

- We **split data randomly** into
  - training set
  - test set

$$RMSE = \sqrt{\sum_{(i,t)\in TEST} \frac{1}{n_{TEST}} \left(\ln(p_i^t) - \hat{f}_{TRAIN}(z_i^t, t)\right)^2}$$

- Accuracy of predictions on test set measured by **Root Mean Square Error (RMSE)**
- Split repeated several times using **Cross-Validation** → Mean of RMSEs



(Bischl et al., 2024)

2. Methodology – Machine Learning

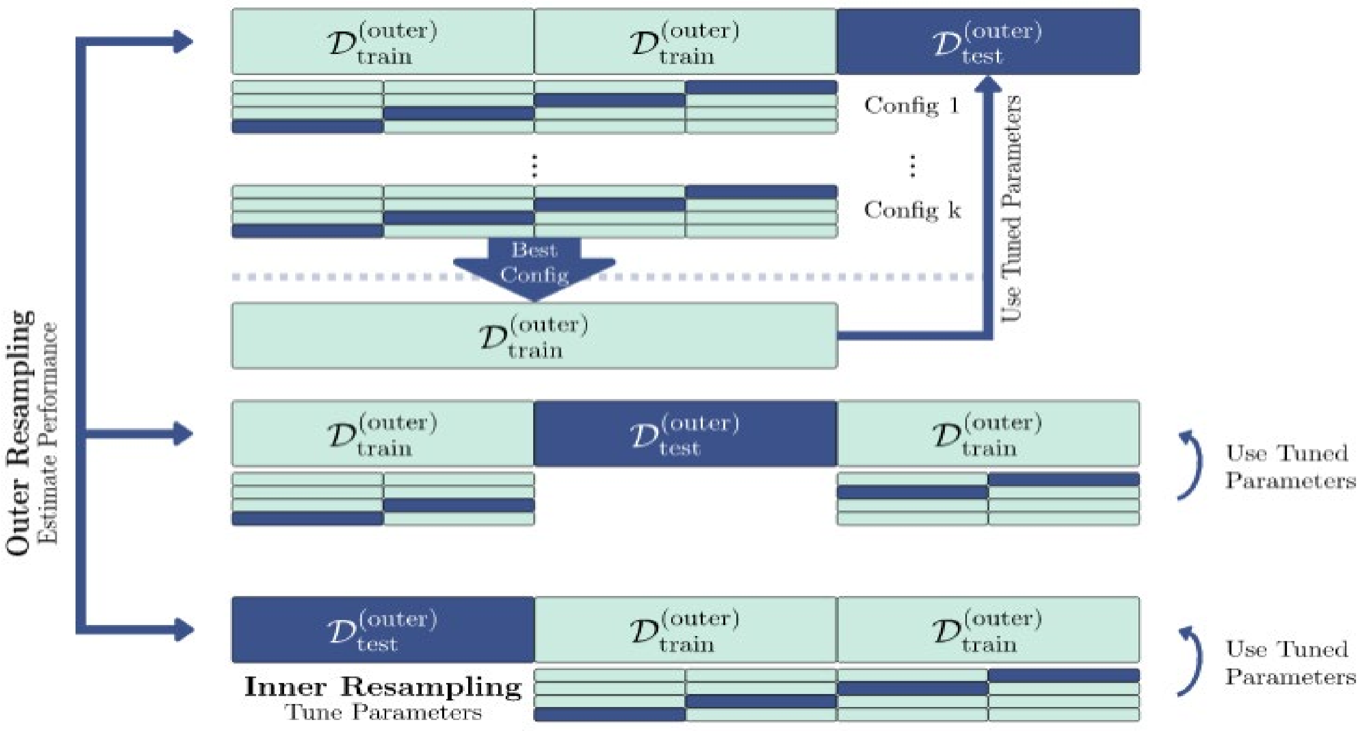
- ML models depend on **hyperparameters**

Method	Hyperparameters
Random Forest	mtry, num_trees, replace, sample_fraction
XGBoost	alpha, lambda, eta, nrounds, max_depth, subsample, colsample_bylevel, colsample_bytree

- We want to find **the best hyperparameters** for optimal prediction performance
- We can use **Cross-Validation** for hyperparameter optimization; the performance of a given set of hyperparameters is then measured by their average RMSE

2. Methodology – Machine Learning

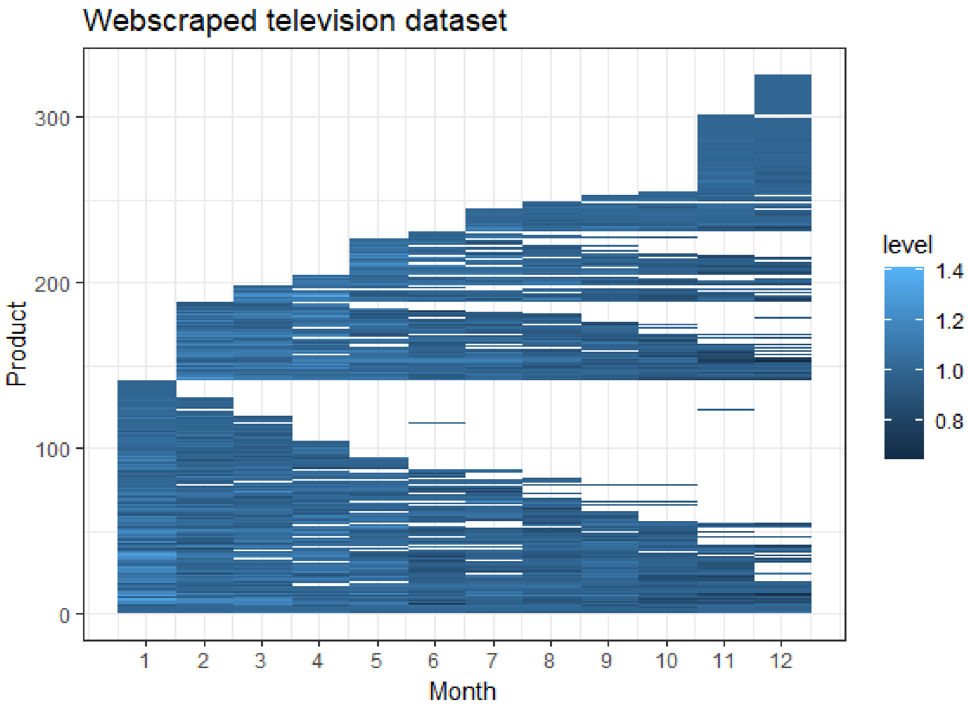
- We use **nested Cross-Validation** to combine the tuning process of the hyperparameters and the performance evaluation of the model itself.

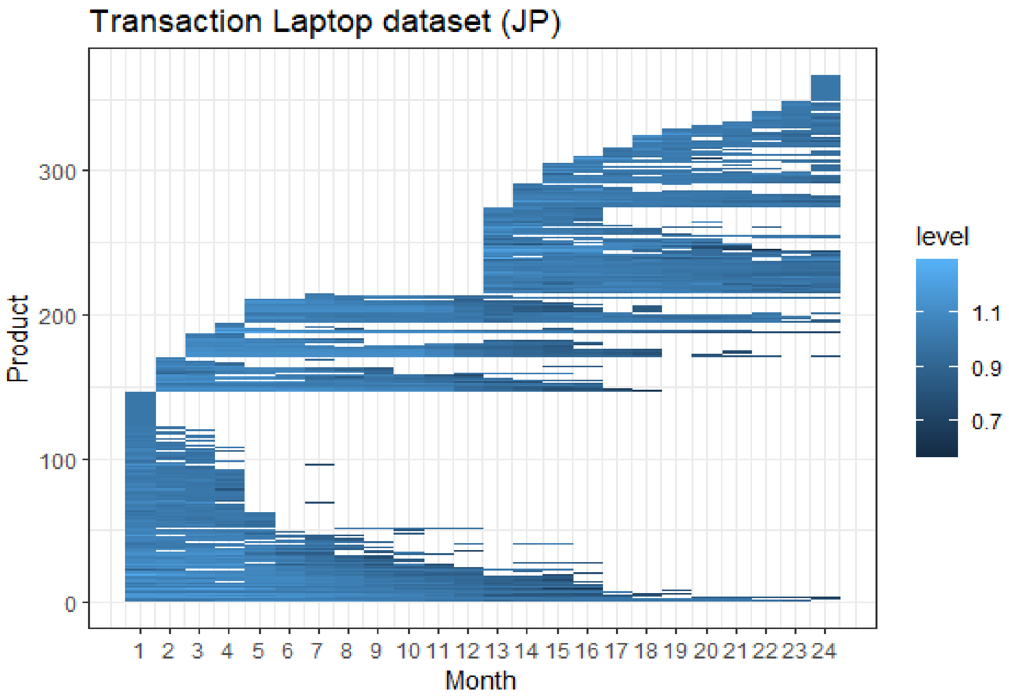


(Bischl et al., 2024)

3. Results

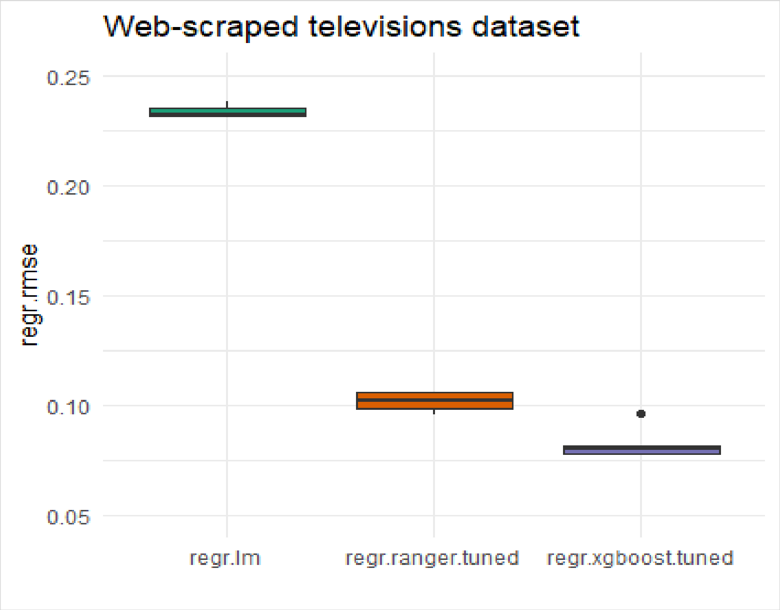
Two datasets with **high item churn** and **technological change**

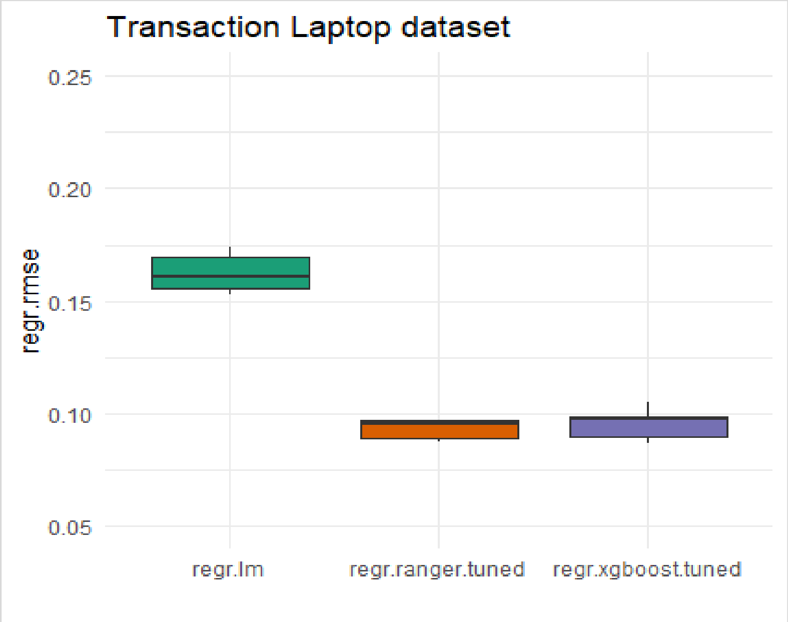




3. Results

- Nested Cross-Validation: 5 folds (outer layer – for model performance), 10 folds (inner layer – for hyperparameter optimization)
- **ML methods (Random Forest, XGBoost) perform better on unseen data than linear regression : lower RMSE**



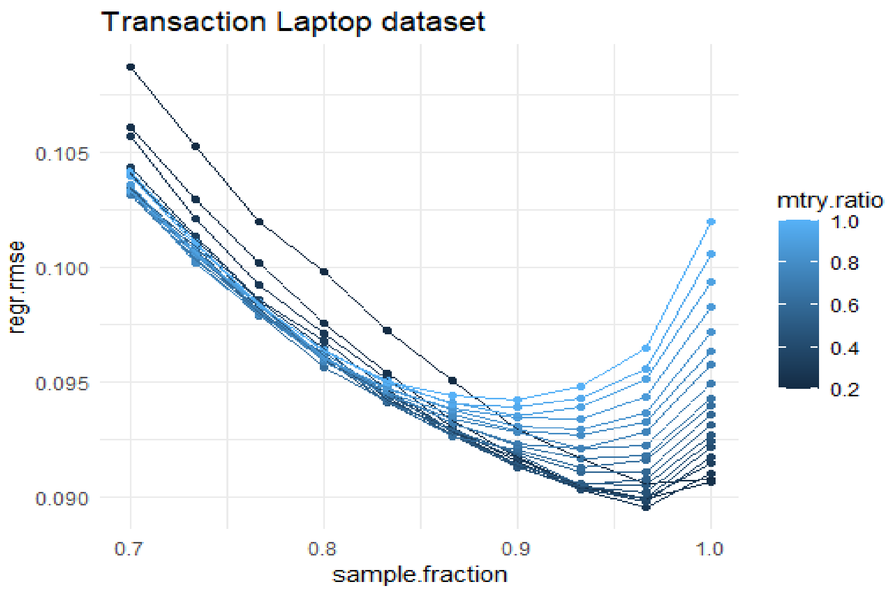


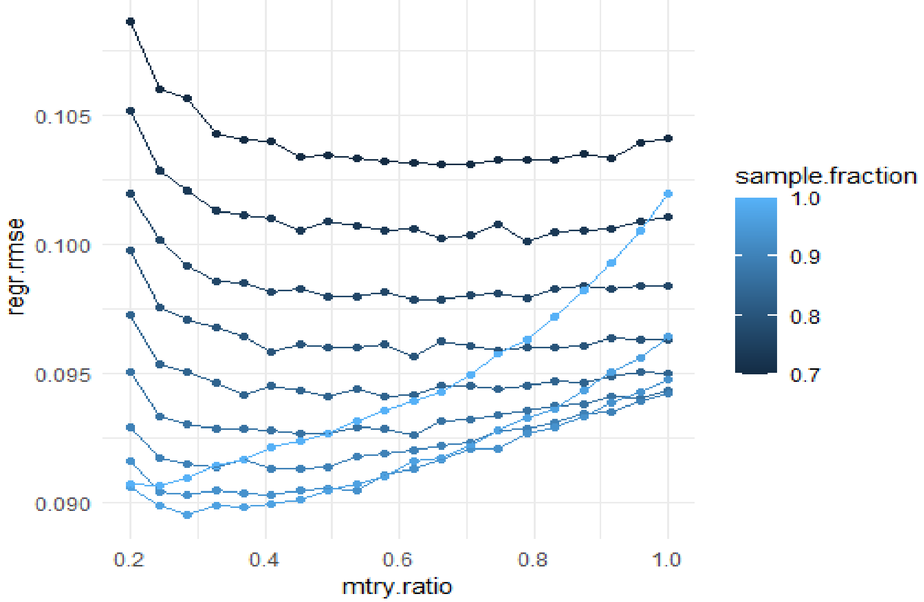
→ *How to best design the (nested) resampling scheme ? (e.g. use time as stratification?)*

→ *How to encode time in the ML methods ?*

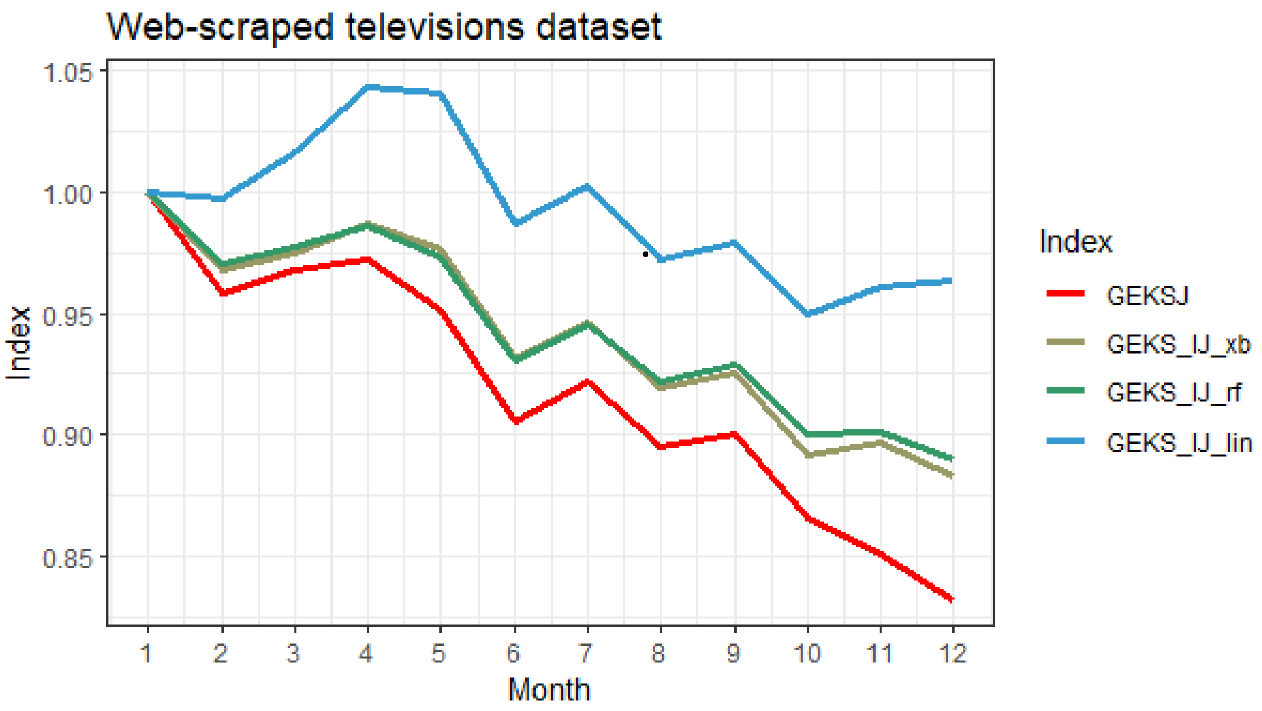
3. Results

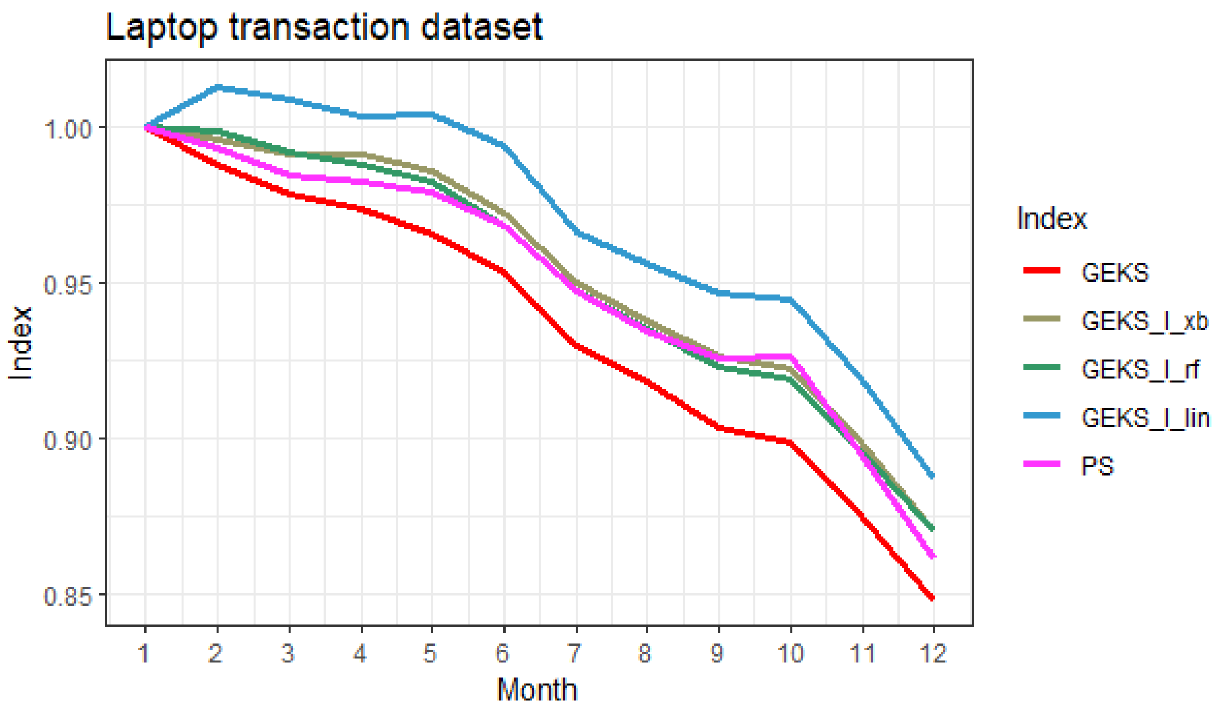
- **Hyperparameter optimization** affects model performance crucially
- Two step approach: first **Bayesian optimizer**, and then fine tuning via **grid search**
- Illustration of grid search for Random Forest





3. Results

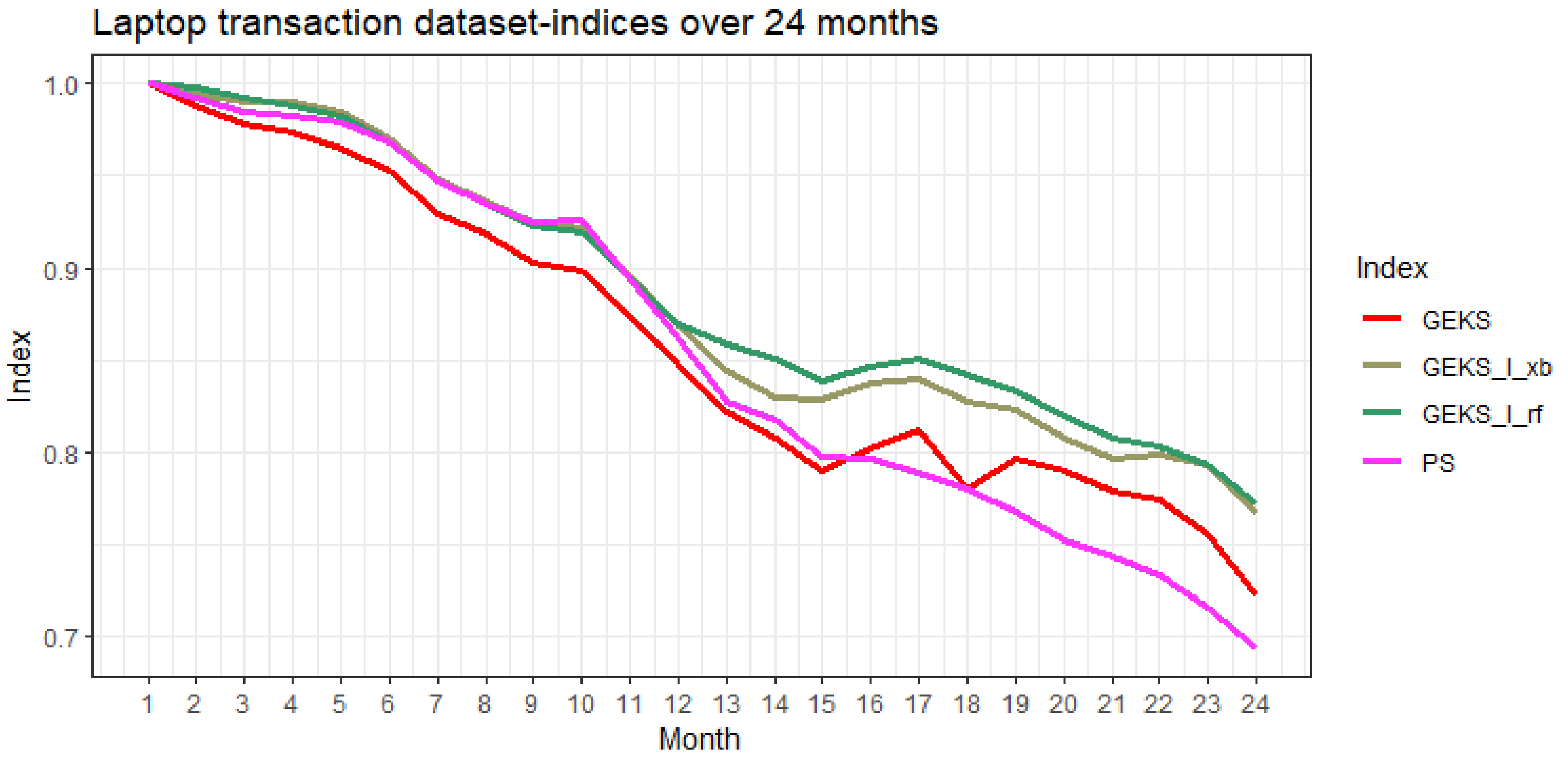




PS = Predicted Share Similarity Linked Index of Diewert E., Shimizu C. (2023). *Scanner Data, Product Churn and Quality Adjustment.* Paper presented at the UNECE meeting of the Group of Experts on Consumer Price, Geneva, included for comparison

3. Results

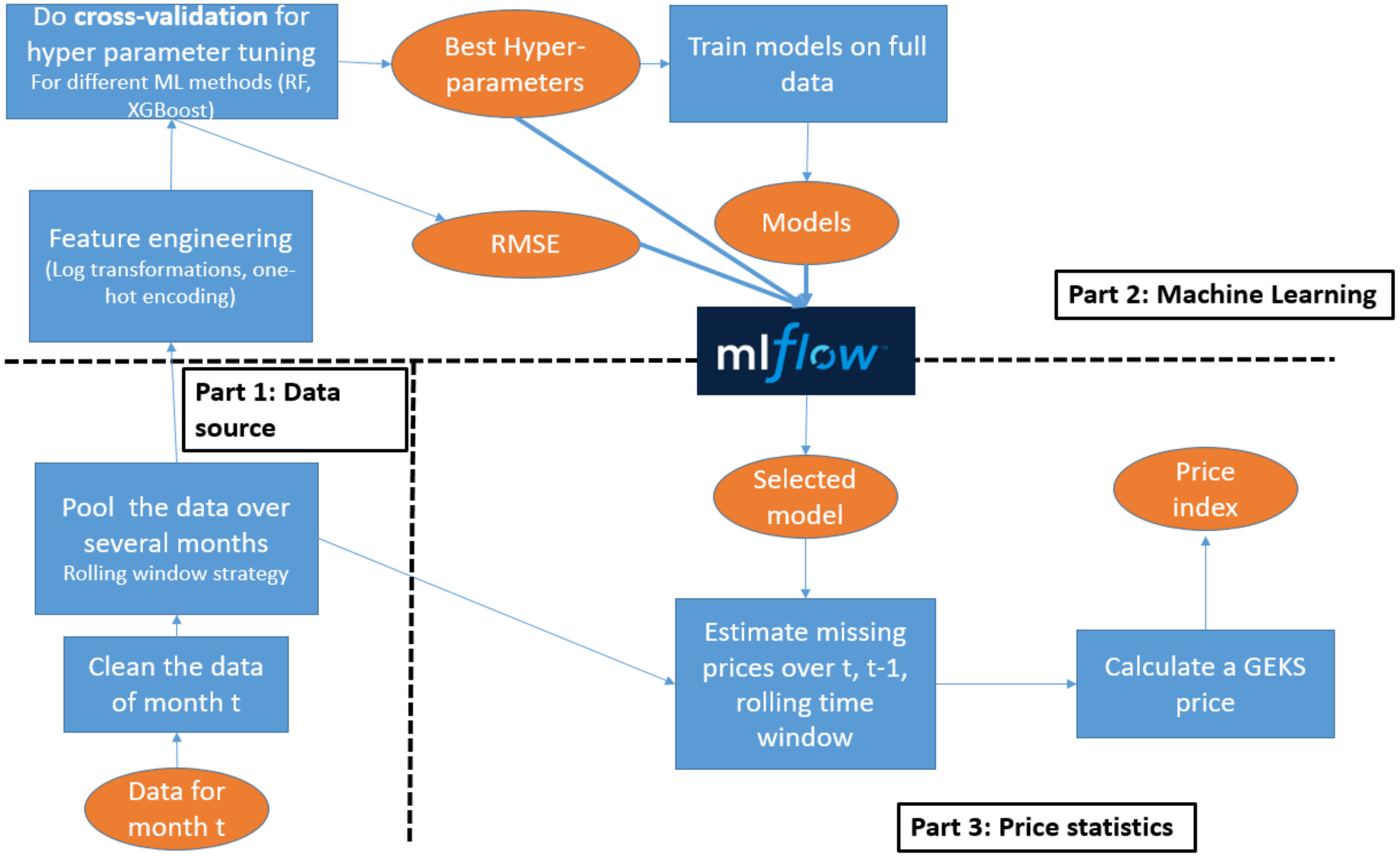
12-month rolling GEKS indices



PS = Predicted Share Similarity Linked Index of Diewert E., Shimizu C. (2023). *Scanner Data, Product Churn and Quality Adjustment.* Paper presented at the UNECE meeting of the Group of Experts on Consumer Price, Geneva, included for comparison

4. Process

ML based imputation methodology in practice – **division of the process** into three parts



Conclusions and open issues

- ML methods can be used for making **price imputations** that can be integrated in the price indices for items that are missing.
- **Performance** of the ML methods has been evaluated on two datasets →XGBoost and Random Forest can significantly lower the RMSE compared to a standard linear regression
- **Choice of the imputation method** also has **an impact on the final prices indices**
- We prefer Random Forest over XGBoost as the hyperparameters can be more easily optimized for the former than for the latter
- The **entry cost** to Machine Learning was reduced by relying on existing tools (e.g. MLR3 collection of R packages)
- We have suggested a **process** for using ML methods in a production environment.

Conclusions and open issues

Open issues:

- Choice of the **re-sampling strategy** (stratifications, etc.) in the context of price statistics that take product and time dimensions into account
- Treatment/encoding of **time variable** in ML methods
- **Best time window** over which ML should be trained and how should these time windows be adjusted when data of the last period becomes available.

- Relationship between the imputations for an individual price and the corresponding price index. For example, how does the **bias-variance trade-off** of the ML method propagates to the price indices?

Any Questions?

STATEC

Institut national de la statistique
et des études économiques

Thank you!

13, rue Erasme
L-1468 Luxembourg

(+352) 247-84219

info@statec.etat.lu

statistiques.public.lu

@Statec Luxembourg

/STATEC

@STATEC

Statec Luxembourg