

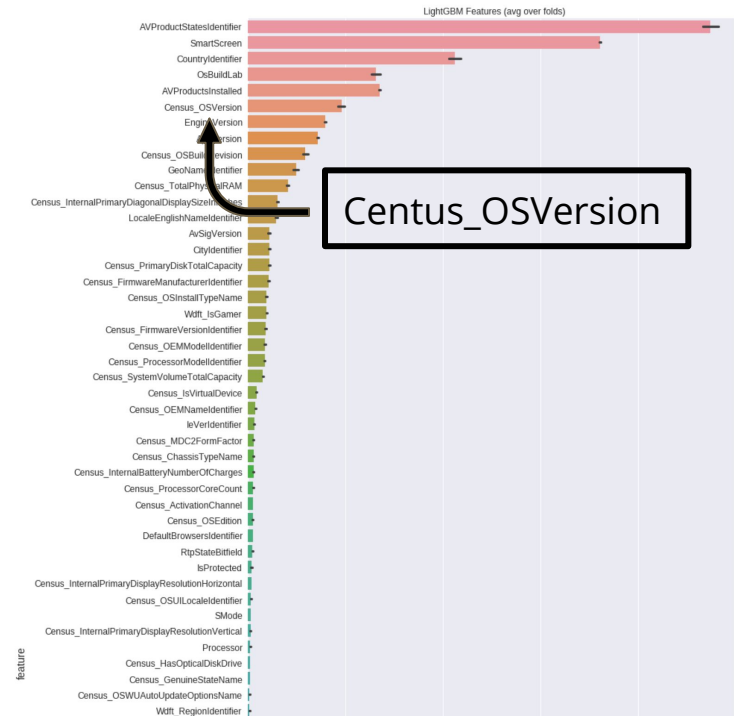
LOFO Importance

— Rafah El-Khatib —

Ahmet Erdem, Stephane Collot, Eva Van Weel

Example: Kaggle's Microsoft Malware Prediction

- Predict if a machine will soon be hit with malware
- Seasonal feature: OS version
 - Split out of time
 - Training data: OS version was very predictive
 - Test data: What about new OS versions?
- Tree model importances:
 - Consider tree structure in training phase
 - Give high importance to **OS version**



Feature Importance measures out there

- Gini: mean decrease in impurity
- Gain: average training loss reduction
- Split: #splits involving a feature
- Shapley values/Saabas score

Do not generalise
well to test sets

Overestimate
importance to
granular features

Model-
dependent

- Permutation: permute values in test set
and compare performance

Overestimates
importance to
correlated features

LOFO (Leave One Feature Out) Importance

- We want our models to generalise well to unseen **test sets**
- LOFO algorithm:
 - Get baseline performance with all features
 - Remove features one by one, retrain the model, evaluate CV performance
- For **any sklearn-friendly model** (with LightGBM as the default)
- For **any performance metric**
- **Any validation scheme**
- Can even get negative LOFO importances when a feature is harmful

Kaggle's Microsoft Malware Prediction, *Revisited*

<https://www.kaggle.com/divrikwicky/lofo-importance>

Install LOFO and try it out!

`pip install git+https://github.com/aerdem4/lofo-importance`

Caveats:

- Slow although we parallelise!
- So far still misleading when it comes to correlated features (as most other importance types)

Future Work:

- Flag correlated features, leave-n-features-out