Sberbank Data Science Journey: AutoML

2nd place solution

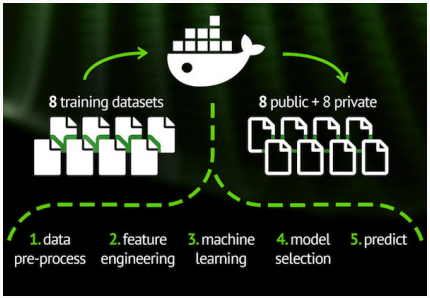
Anton Klenitskiy

November 24, 2018

About me

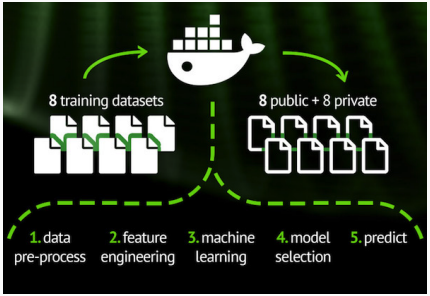
- PhD in theoretical physics (MSU)
- DS/ML during last several years
- Currently Research Engineer at IBM

Formulation



- 3 regression tasks (RMSE)
- 5 binary classification tasks (ROC AUC)
- Anonymized features, but with known types
- Size vary a lot - from several hundreds to million of rows
- Score for each task - rank between 0 and 1 (0 - baseline, 1 - top solution)

Formulation



- Submit code, run in docker
- Resources limit - 12GB RAM, 4 CPU, 1GB hard drive
- Time limit - from 5 to 30 minutes

```
{
    "image": "sberbank/python",
    "entry_points": {
        "train_classification": "python train.py --mode classification --train-csv {train_csv} --model-dir {model_dir}",
        "train_regression": "python train.py --mode regression --train-csv {train_csv} --model-dir {model_dir}",
        "predict": "python predict.py --test-csv {test_csv} --prediction-csv {prediction_csv} --model-dir {model_dir}"
    }
}
```

Data

	rows	columns	number	string	datetime	id
check_1_r	365	42	39	0	1	0
check_2_r	13958	11	6	2	0	0
check_3_r	250000	43	39	0	1	1
check_4_c	114130	143	138	0	3	0
check_5_c	467485	17	14	0	1	0
check_6_c	108814	114	112	0	0	0
check_7_c	92091	774	765	4	2	1
check_8_c	143525	878	753	32	91	0

Leak

Time series task - lags of target as features

datetime_0	id_0	target	number_23	number_24	number_25
2016-09-12	101	28600.0	100.0	0.0	32500.0
2016-09-13	101	14400.0	28600.0	100.0	0.0
2016-09-14	101	9500.0	14400.0	28600.0	100.0
2016-09-15	101	107000.0	9500.0	14400.0	28600.0
2016-09-16	101	118600.0	107000.0	9500.0	14400.0

Fun times on the LB after public leaky solution with high score
But leak didn't work on private datasets - thanks to organizers and docker format

Validation

Validation is tricky

- Impossible to reconstruct competition metric locally
- Trust your CV?

  But private consists of completely different datasets
- Public leaderboard climbing?

  The same objection

What else?

- CV + LB + common sense

Memory and time issues

- A lot of competitors had problems with biggest test dataset. Naive solution failed due to memory limit (probably).

  My way out:
  - read small part of data, define data types for all columns
  - read entire data with float32 instead of float64
  - parse datetime columns while reading
  - feature selection based on LightGBM feature importance

- For the safety one should check if time limit is not exceeded

  (or hope for good luck)

Preprocessing

Basic preprocessing from public scripts

- Extract datetime features (day of week, day of month,..)
- Count encoding of categorical features
- Drop constant features
- Didn't use scaling and filling missing values

More preprocessing

- For each datetime feature add *is_holiday* flag
  Helped a lot on train, didn't help on public, ? on private
- Mean target encoding
  Helped a lot on train, didn't help on public, ? on private

Machine learning approach

- LightGBM - fast and accurate
- Hyperopt for parameter tuning
  After each step check if time limit is not exceeded, then continue
- Single train-validation split (for speed, time limits were quite restrictive)

Two ways of dealing with hyperopt:

- Find best set of parameters (may be on subset of data), then retrain on all data, including validation set
- Train on all data (excluding validation set), remember best models and blend them

Blending

- During hyperopt iterations remember all models that were trained
- Choose 5 best models in the end
- Blend them using stepwise blending
- No stacking, too complicated and high risk of overfitting

Stepwise blending

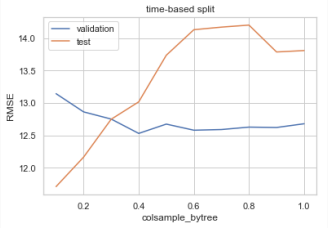
from

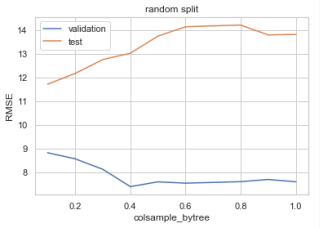
Caruana, R., Niculescu-Mizil, A., Crew, G., & Ksikes, A. (2004). Ensemble selection from libraries of models.

Stepwise forward selection, add models by one with replacement with equal weights.

Some observations

1. Completely different dependency on parameters for validation set and test set (Dataset 1)





2. Two identical solutions, but with different number of hyperopt iterations

iterations	50	100
score	4.37	3.81

More tuning is worse? Test may be quite different from train for some tasks? Or problem with time series validation?

Special cases

Very small dataset

- Don't use target encoding
- Don't use parameter optimization at all
- Just run several LightGBM, XGBoost, Random Forest and Extra Trees models with random parameters and average them

Very big dataset

- Don't use target encoding (for safety from memory problems)
- Simple feature selection from LightGBM feature importance to reduce number of features and hence size of the dataset

Other things tried

- XGBoost
- Ensemble of XGBoost and LightGBM
- H2O AutoML
- Hyperband/Successive Halving
- Would be useful to make time series validation, but how?

Final submissions

1. Vanilla, plain hyperopt without blending

   Public score - 4.65, private score - 5.16 (26th place)

2. With blending, holidays and mean target

   Public score - 4.27, private score - 6.48 (2nd place)

Final thoughts

- Nice docker format of the competition
- Don't overfit to public leaderboard
- May be very small dataset is not good for competition

  Impact of noise and random chance
- Hard to analyze what worked and what didn't

  We know only combined score for all tasks, private score only for final submissions
- `https://github.com/antklen/sdsj2018_solution`

Thank you for your attention