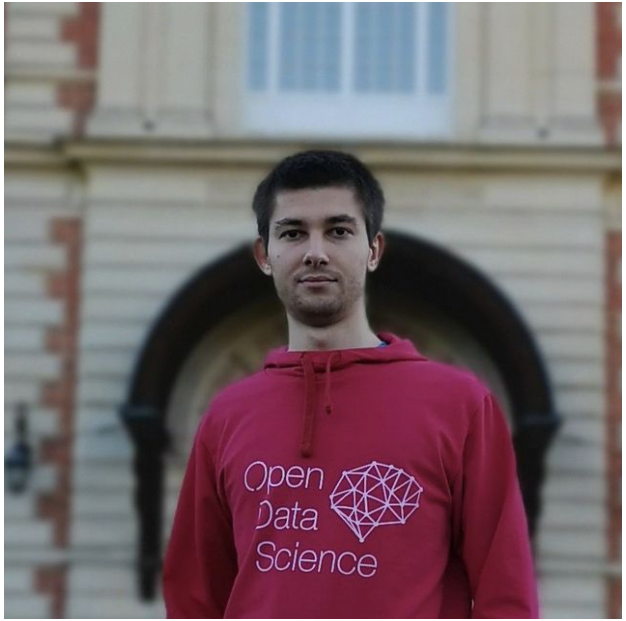
AutoSeries

Bio



Denis Vorotyntsev
Sr Data Scientist @ Oura

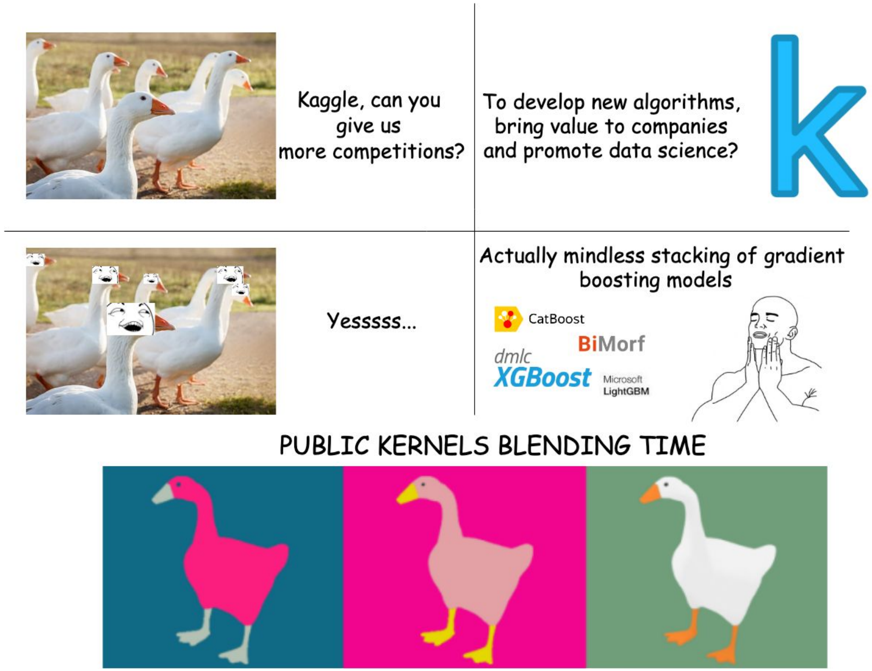
- Various project with time-series data: classification, clustering, regression, etc

Research Scientist @ VTT Research Center of Finland

- Anomaly detection in steel manufacturing

Linkedin

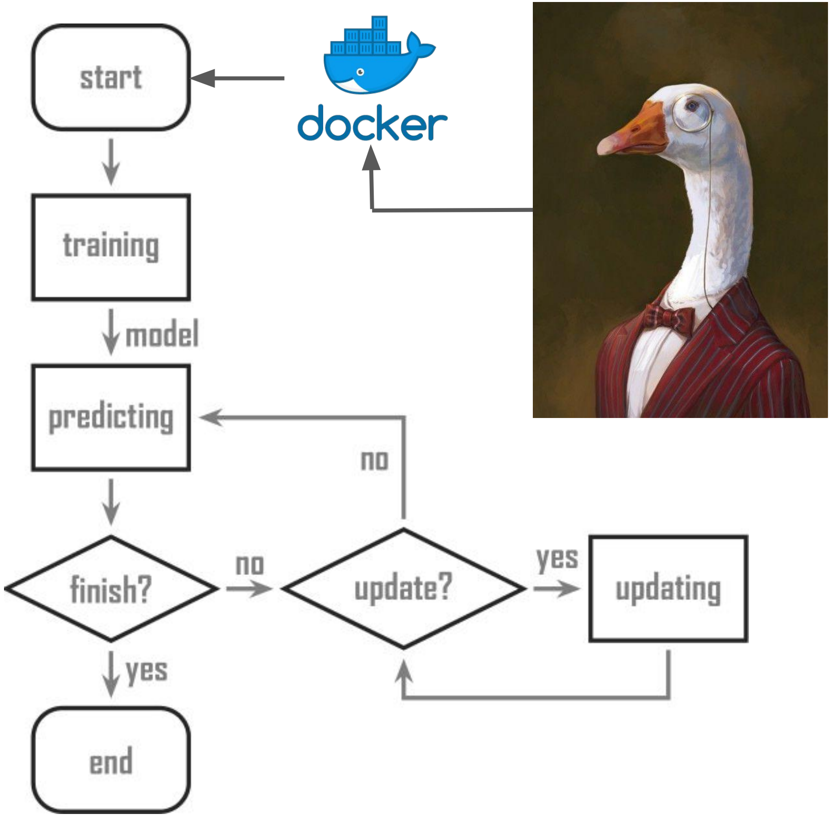
Current State of ML Competitions



Write code, train and predict locally ⟶ submit answers

- Single dataset ⟶ deep dive into problem;
- Test data is available;
- Domain understanding: sophisticated feature engineering, advanced models;
- Time inefficient: train as many models as you wish; stacking & blending.

AutoML Competitions



Write code locally ⟶ submit code

- New, unseen data in test;
- Data from many domains ⟶ solution should be general;
- Strict time limits. Model has not fit in a given time ⟶ you'll get the worst score.

AutoSeries

- [AutoSeries]() - 10th competition in AutoML series organized by 4Paradigm and ChaLearn
- Time series regression
- Ten datasets (five public and five private) from different domains: air quality, sales, work presence, city traffic, etc
- Submit code, constraints: 16 Gb RAM, 4 CPU, no GPU
- Average rank (among all participants) of RMSE obtained on the five datasets
- 45 participants in the feedback phase, 12 in the final phase
- Results: 1st place
  - Code
  - Blog post

Task Example

Example: retail sales prediction.

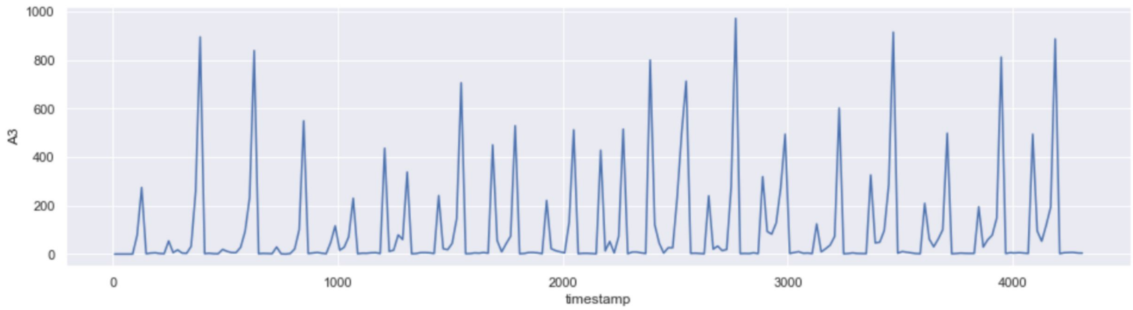
A1 - timestamp. Must be in data, single, sorted.

A2 - primary ID (shop ID). We could have none (single shop) or multiple primary ID (shop & product type) in data.

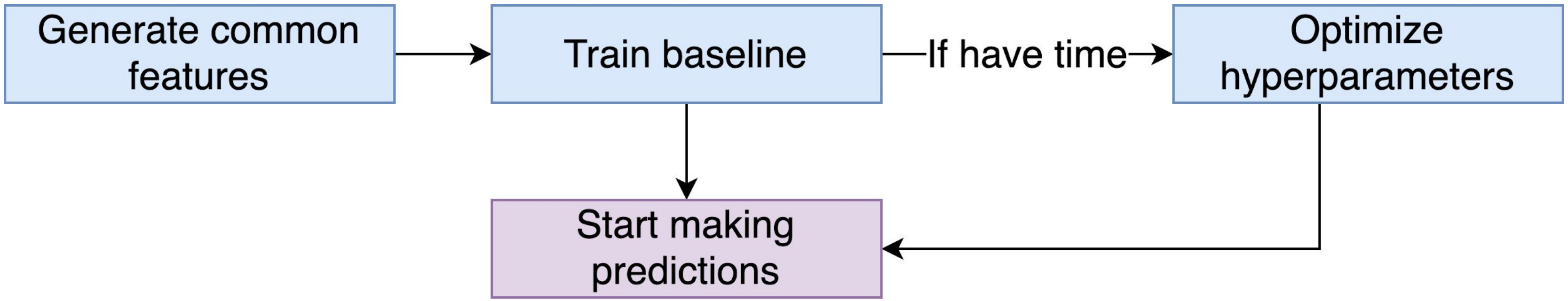
A3 - target (number of sales).

	A1	A2	A3
0	883612800	-6608418032804380965	0.0
1	883612800	3055235448505306399	0.0
2	883612800	3729226436453271103	0.0
3	883612800	6960584904140561905	0.0
4	883612800	1143350366519272165	0.0
5	883612800	-8223253218484014081	0.0
6	883612800	5404213741217308375	0.0
7	883612800	8613428591356018513	0.0
8	883612800	7698141674612140154	0.0
9	883612800	-808100555186871340	0.0

```
---
time_budget:
    train: 1000
    predict: 1000
    update: 1000
    save: 1000
    load: 1000

schema:
    "A1": "timestamp"
    "A2": "str"
    "A3": "num"

is_multivariate: True
is_relative_time: False

primary_timestamp: "A1"
primary_id: ["A2"]
label: "A3"
```



Overview of the Final Pipeline

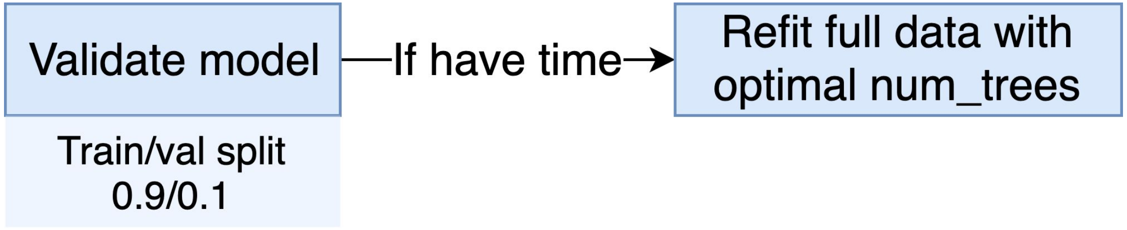


Common Features

- Numerical operations of pair of numerical features
  - Determine important features: fit random forest ⟶ top 3 important (gini)
  - num1 + num2, num1 * num2, num1 / num2, num1 - num2
- Time-based features: year, month, day of year, weekday, hour. Treated as numerical. Other options:
  - As category (works well sometimes);
  - Turn into embeddings (worked well in [Cold Start Energy Predictions]);
- Shift and diff features for target and important numerical features
  - x(t-lag), lag = 1, 2, 3, 5, 7;
  - x(t-1) - x(t-n), n = 2, 3, 4, 6, 8;
- Each category is replaced with category + ID
  - df[cat_col] = df[cat_col].astype("str") + "_" + df["timeseries_id"].astype("str")

Validation & Baseline

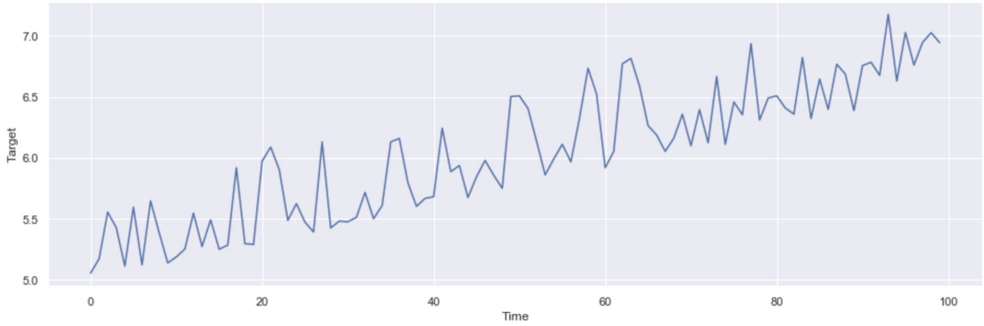
1. LightGBM model, Catboost encoder for categories (Category Encoders), target “as is”
2. Refit model using full data



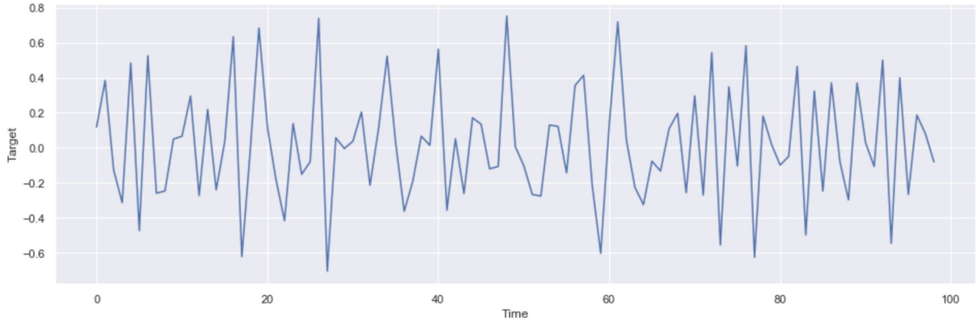
Benchmarking Categorical Encoders

Optimize Main Parameters

1. Transform target
   a. Keep “as is”
   b. Difference
2. Transform categorical columns
   a. pd.Categorical (OHE)
   b. Catboost encoder



new_target(t) = target(t) - target(t-1)



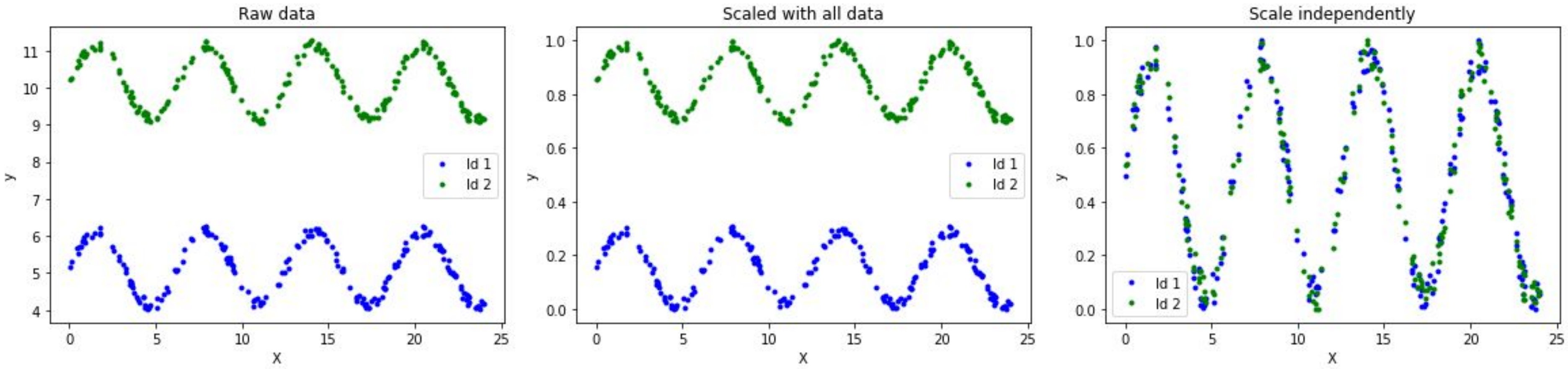
Features Selection & Hyperparameters Optimization

1. Select features: refit on top-n% (10, 20, 50, 75%) most important (“gini”)
2. Optimize hyperparameters - RandomizedGridSearch

```
param_grid = ParameterGrid({
    "learning_rate": [0.05],
    "n_estimators": [1000],
    "num_leaves": [15, 31, 63, 127, 255],
    "min_child_samples": [3, 20, 50, 150],

    "subsample_freq": [1, 5, 25, 50],
    "colsample_bytree": [1.0, 0.8, 0.6],
    "subsample": [1.0, 0.8, 0.6],

    "lambda_l2": [0, 0.1, 1, 10],
    "random_state": [2020]
})
```

What didn’t Work

- Single model for each time-series ID or target scaling
- Catboost (too slow), Linear Models (inaccurate)
- Target transformations: power, Box-Cox, log transform
- Stacking & Blending with different seeds



Cold Start Energy Predictions

Results

	#	User	Entries	Date of Last Entry	<Rank> ▲
ods.ai	1	rekcahd	27	12/13/19	2.0000
	2	DeepBlueAI	23	12/30/19	2.4000
ods.ai	3	DenisVorotyntsev	42	12/30/19	3.0000
	4	DeepWisdom	20	12/30/19	4.0000
	5	Kon	84	12/30/19	4.8000
	6	qijiaheng	8	12/13/19	8.2000
	7	bingo	33	12/18/19	8.4000
	8	lishuqiao	38	12/30/19	8.8000
	9	Reeed	39	12/30/19	9.4000
	10	Jie_Zhang	20	12/30/19	10.4000

Team	Avg Rank	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Dataset 5
DenisVorotyntsev	1.8	1	2	1	2	3
DeepBlueAI	3.6	2	3	5	3	5
DeepWisdom	4.2	5	4	7	4	1
Kon	5	3	8	3	9	2
bingo	5.6	4	5	2	6	11
rekcahd	5.8	9	1	9	1	9
Jie_Zhang	6.8	8	11	4	7	4

AutoSeries Challenge - Feedback Phase

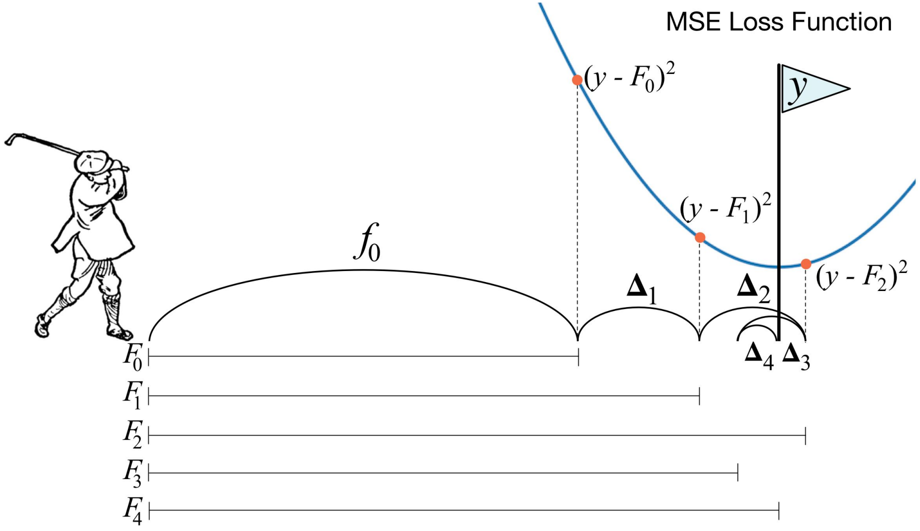
DeepBlueAI Solution (2nd)

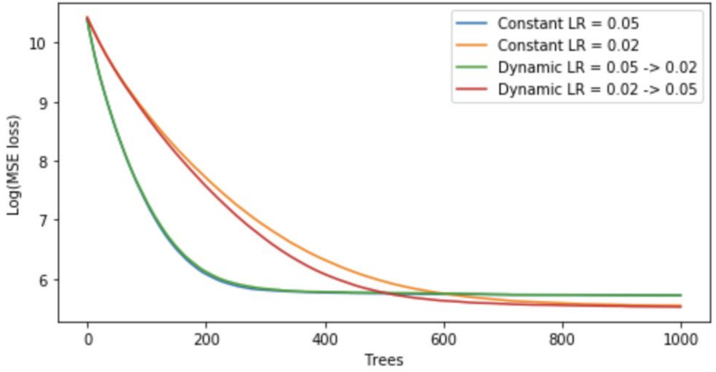
1. Additional features - Previous target values, lag=1. Probably made a bug, I opened issue;
2. Time features - 'year' (unique values>1), 'month'(>11), 'day'(>27), 'hour'(>23), 'weekday'(>6), 'minute'(>4);
3. Target scaling: Min = mean - 6*std, max = mean + 6*std;
4. Category - pd.Categorical;
5. LightGBM and Linear Model blend with coefficients
    a. LightGBM - optimize (meta learning) subsample, num trees and lr (dynamic lr);
    b. LR - sklearn.feature_selection.SelectPercentile for Feature Selection;
    c. Make prediction for validation data;
    d. *final_pred = pred_a * a + pred_b * (1-a)*, *a* is hyperparameter;
6. Number of updates = 5 (constant).

Code

Dynamic LR

LR strategy	Num rounds	MSE loss
Constant 0.05	1624	299
Constant 0.02	2000*	237
0.05 → 0.02	1430	299
0.02 → 0.05	1994*	235 (1400 iter - 241)





* Did not meet early stopping

Picture: [How to explain gradient boosting]

[Experiment Code]

DeepWisdom Solution (3rd)

1. Categories - pd.Categorical;
2. Time features - 'year', 'month', 'day', 'hour', 'weekday';
3. New features:
    a. DeltaFeatures: num(t-1) - num(t-2), (num(t-1) - num(t-2)) / (num(t-2) + eps);
    b. “LagFeatures”: mean, std, max, min for last 3, 7, 14, 30 periods;
4. Explore stage
    a. Bayesian hyperparameters optimization on subset of data for LightGBM;
    b. Feature selection for linear models;
5. Models: LightGBM, Ridge, Lasso;
6. Adjust blend coefficients during inference;
7. Number of updates = update time / train time (without explore stage).

[Code]

Comments & questions