National Research University
Higher School of Economics
Faculty of Computer Science

Sberbank Russian Housing Market

Presenters:

MAJID SOHRABI

BAYAN ALI HENDAWI

24/12/2021

Content

1. Overview.
2. Data Description.
3. Preprocessing Data.
4. Feature Engineering.
5. Feature Selection.
6. Models.
7. Results.

Content

Overview

Sberbank Russian Housing Market.

Apr 27, 2017 – Jun 23, 2017

predict the price of houses in Moscow.

Content

1. Motivation.
2. Data Description.
3. Preprocessing Data.
4. Feature Engineering.
5. Feature Selection.
6. Models.
7. Results.

Data Description

The dataset contains information about houses and consists of two dataset:

- train dataset 30471 rows x 291 columns
- test dataset: 7662 rows x 291 columns

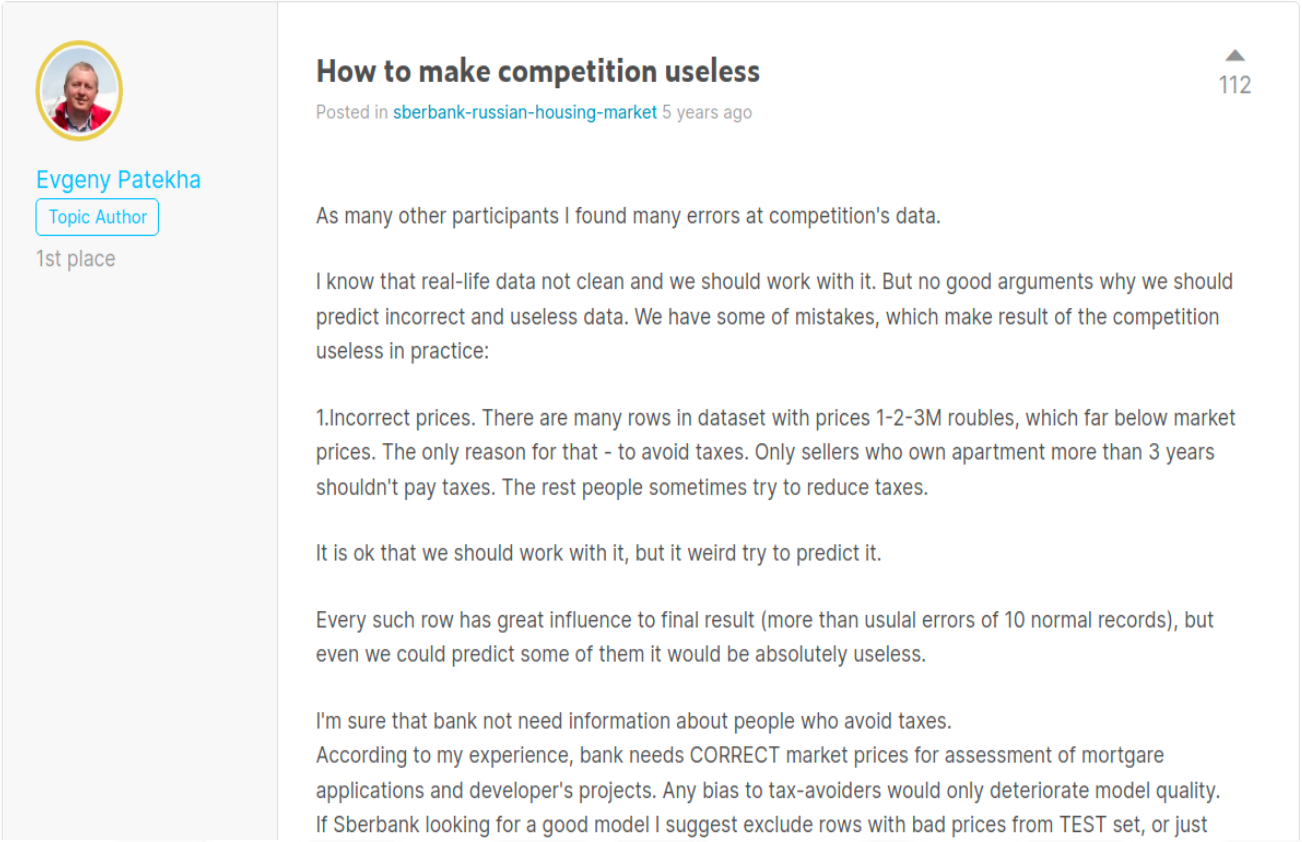
- Train: August 2011 – June 2015
- Test: July 2015 – May 2016

But …?

Data Description

A lot of missing values:

- 261026 / 8897532 in the training data set (~3%)
- 47138 / 2229642 in the testing data set (~2%)



Evgeny Patekha

Topic Author

1st place

How to make competition useless

112

Posted in sberbank-russian-housing-market 5 years ago

As many other participants I found many errors at competition's data.

I know that real-life data not clean and we should work with it. But no good arguments why we should predict incorrect and useless data. We have some of mistakes, which make result of the competition useless in practice:

1.Incorrect prices. There are many rows in dataset with prices 1-2-3M roubles, which far below market prices. The only reason for that - to avoid taxes. Only sellers who own apartment more than 3 years shouldn't pay taxes. The rest people sometimes try to reduce taxes.

It is ok that we should work with it, but it weird try to predict it.

Every such row has great influence to final result (more than usual errors of 10 normal records), but even we could predict some of them it would be absolutely useless.

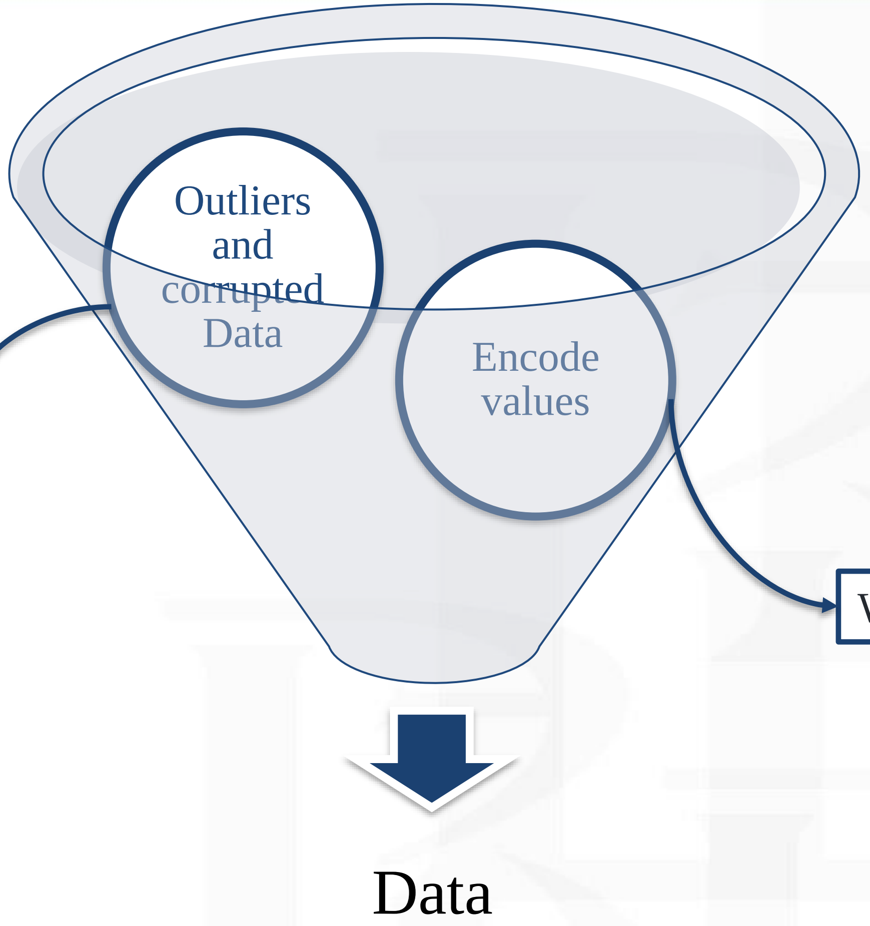
I'm sure that bank not need information about people who avoid taxes.
According to my experience, bank needs CORRECT market prices for assessment of mortgare applications and developer's projects. Any bias to tax-avoiders would only deteriorate model quality.
If Sberbank looking for a good model I suggest exclude rows with bad prices from TEST set, or just

Content

Preprocessing Data

First Preprocessing

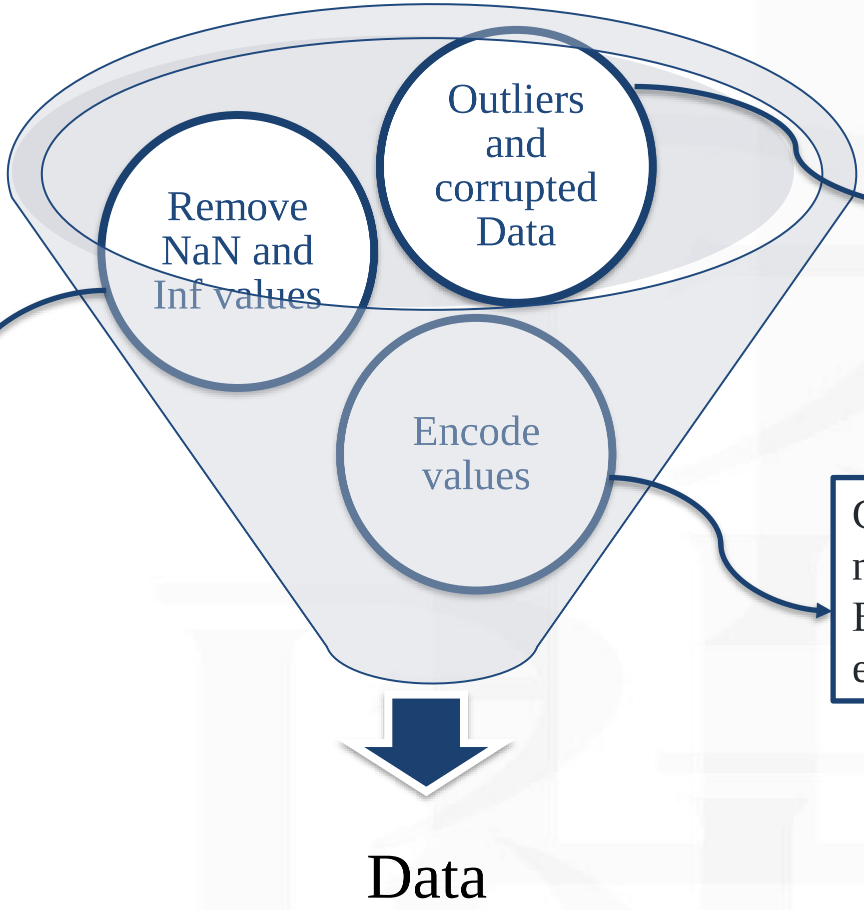
- life square > full square → life square = NaN.
- kitchen square > life square → kitchen square = NaN.
- floor number = 0 → floor number = NaN.
- floor number > max floor → floor number = NaN.
- kitchen square = 1970.0 should be for build year.



We use LabelEncoder().

Preprocessing Data

Second Preprocessing



Same as before

- NaN→ mean of column.
- Inf → maximum of column.
- -inf → minimum of column.

Categorical features encoded manually:
Example: Two values → 0,1 & etc.

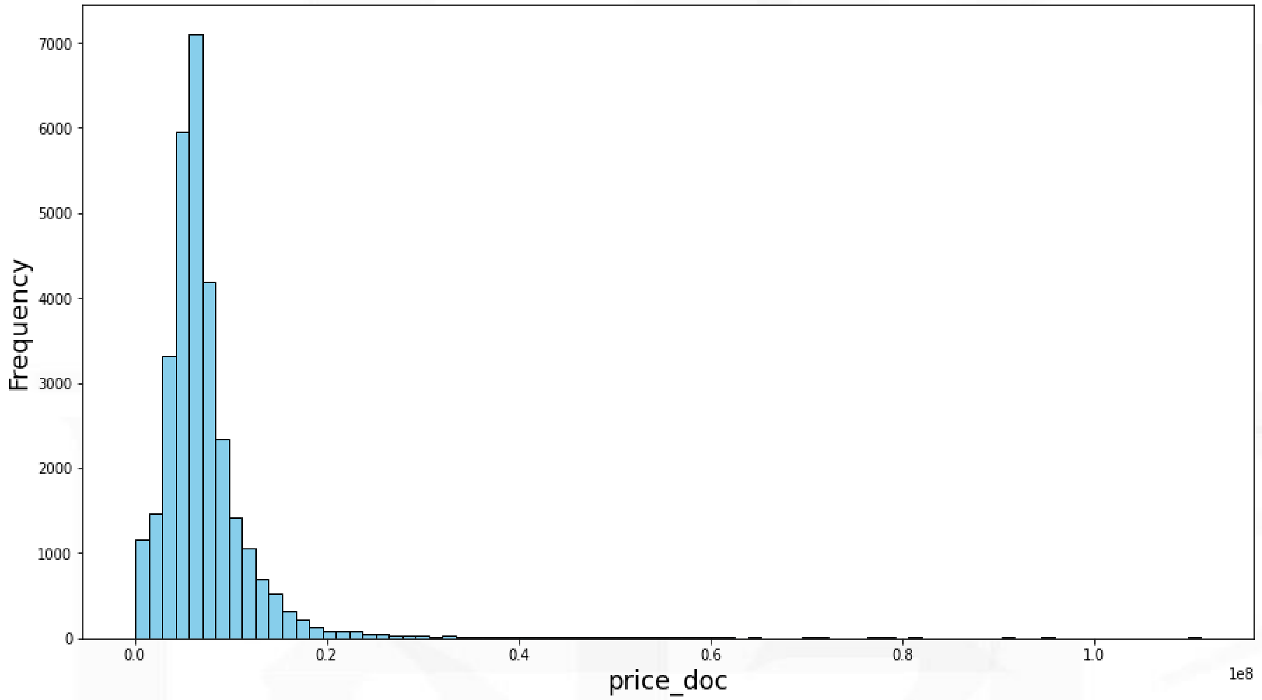
Preprocessing Data

Second Preprocessing

Removing the outliers:

- 4M > Prices > 100M
- 5055 samples (~16.5%)

- 3M > Prices > 100M
- 2724 samples (~9%)



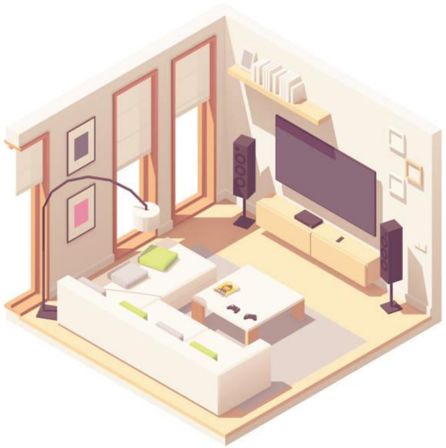
Content

1. Overview.
2. Data Description.
3. Preprocessing Data.
4. Feature Engineering.
5. Feature Selection.
6. Models.
7. Results.

Feature Engineering

We add some features of engineering:

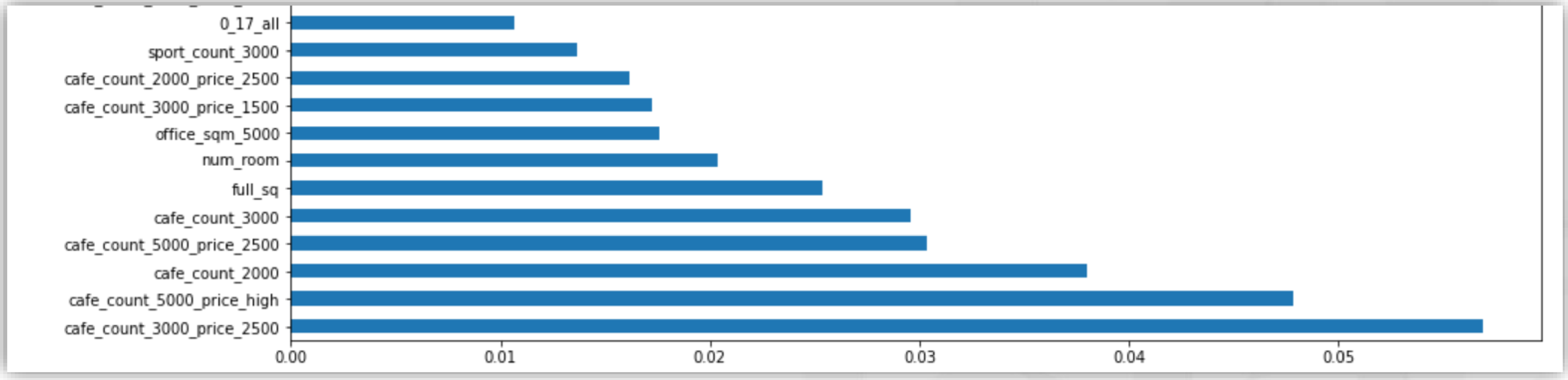
- Relative floor = number of floor / max floor.
- Relative kitchen square = kitchen square / full square.
- Room size = life square / number of rooms.



Content

Feature Selection

We extract the important features by XGBRegressor model and selected the most 12 important features for experiments.

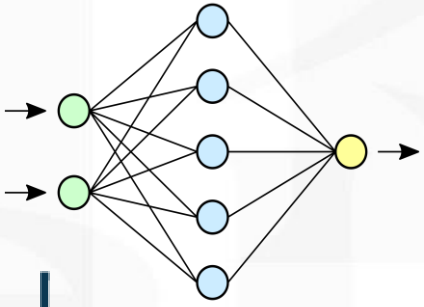


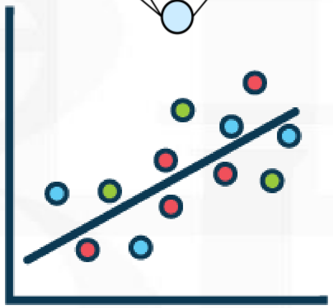
Content

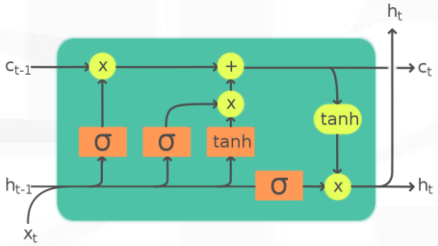
Models

The models which we apply:

1. XGBRegressor: But first, we use GridSearchCV to search over specified parameter values to find the best parameters. (Tuning stage).
2. Random forest.
3. Linear Regression.
4. LightGBM.
5. Neural Network:
    1. Without LSTM.
    2. With LSTM.







Models

1. **XGBRegressor parameters:**
    - learning_rate = 0.03
    - max_depth = 5
    - min_child_weight = 4
    - n_estimators = 500
    - n_thread = 4
    - objective = ‘reg:linear’
    - subsample = 0.7

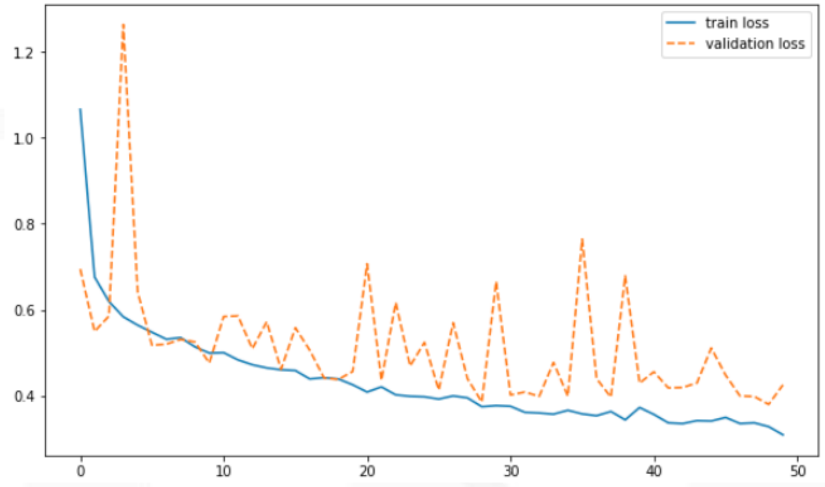
2. **Neural Network (W/O LSTM):**
    - 4 Dense layers (293, 128, 64, 1) with ‘relu’ activation.
    - 3 BatchNormalization.
    - Adam optimizer.

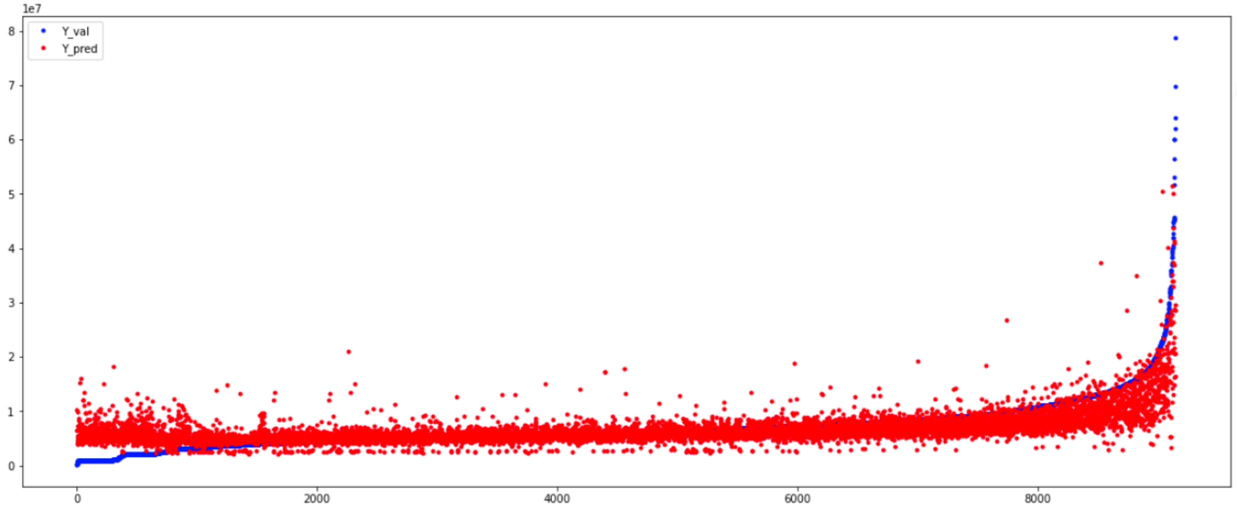
3. **Neural Network (With LSTM):**
    - 3 Dense layers (293, 10, 1) with ‘relu’ activation.
    - 3 LSTM layers (128, 64, 32).
    - BatchNormalization.
    - GlobalAveragePooling1D.
    - Adam optimizer.

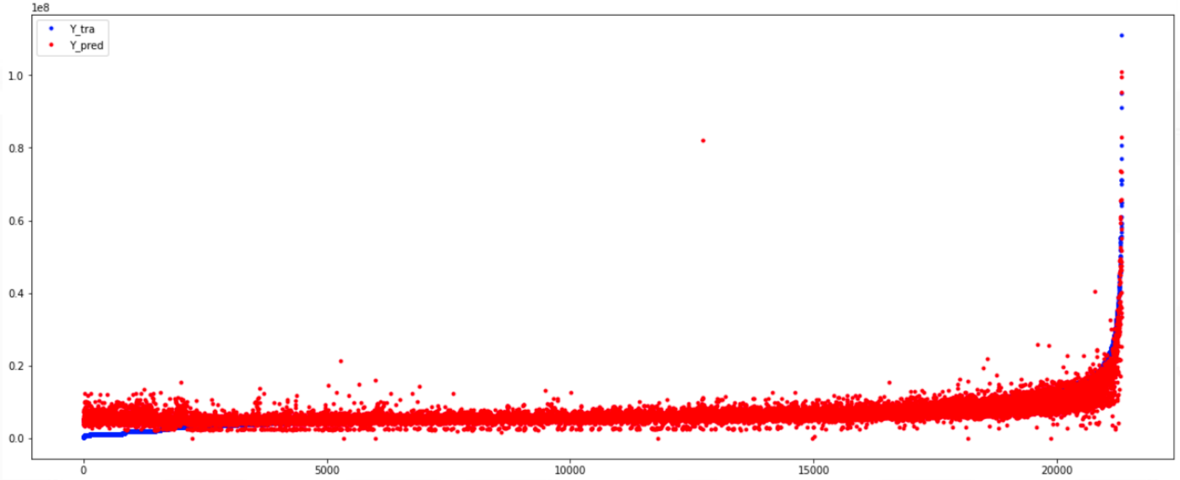
Content

Results

1. Neural Network (W/O LSTM).
2. 50 epochs.
3. True targets vs. prediction for training & validation.

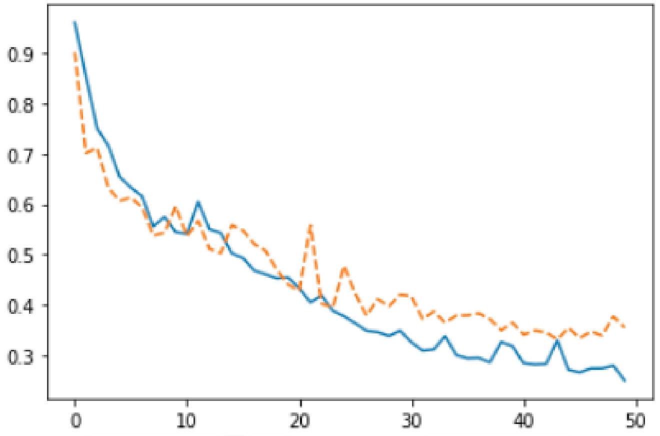


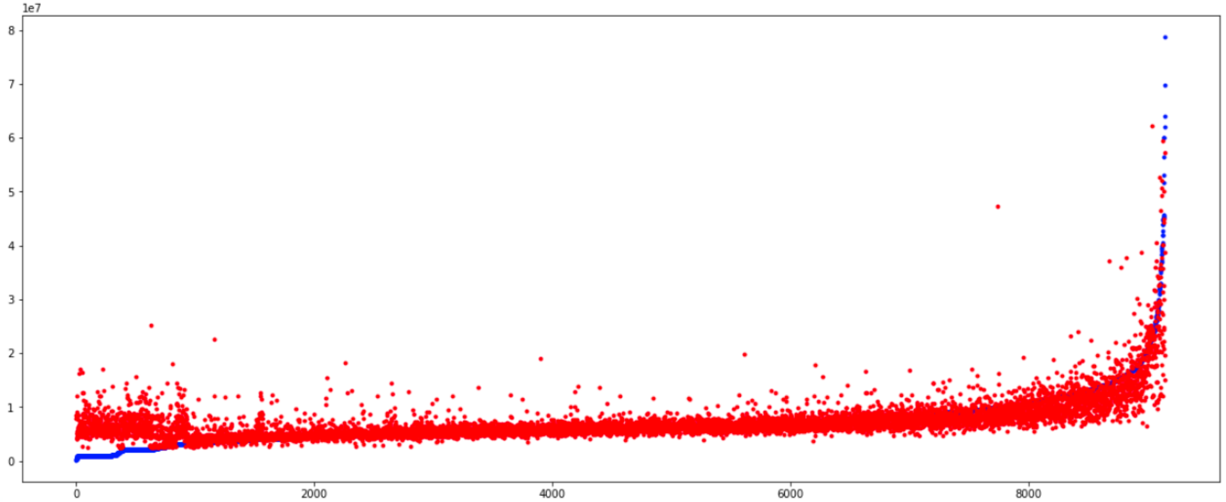


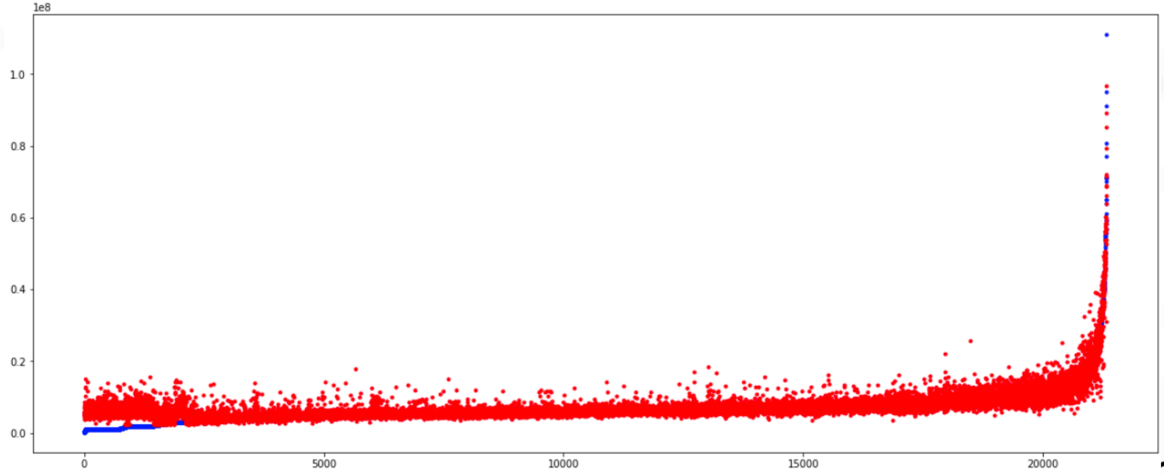


Results

1. Neural Network (With LSTM).
2. 50 epochs.
3. True targets vs. prediction for training & validation.







Results

We implemented the models through many trials depending on the preprocessing.

The results are shown in the figure.
The best result for XGBRegressor.

Our Rank
Public: 1200/3266
Private: 77/3266

Overview Data Code Discussion Leaderboard Rules Team My Submissions Late Sub

lgbModel.csv 4 minutes ago by BayanHendawi LGB Model	0.31953	0.31549
xgb2.csv 6 days ago by BayanHendawi Trial 3 (XGB2)	0.31847	0.31611
xgb1.csv 6 days ago by BayanHendawi Trial 3 (XGB1)	0.31906	0.31523
rfModel.csv 6 days ago by BayanHendawi Trial 3 (RF)	1.21213	1.21799
nnModel.csv 6 days ago by BayanHendawi Trial 3 (NN)	5.96900	6.08614
lstmModel.csv 6 days ago by BayanHendawi Trial 3 (LSTM)	1.00796	1.00669
Linear_model.csv 6 days ago by BayanHendawi Trial 3 (Linear)	2.48282	2.51866

We are ready for your questions.