

# **Kaggle Sprint: Real-Estate Price Estimation**

PhDc, Valentas Gružas

PhD, Andrius Kriščiūnas

# Speakers

## **Valentas Gružas**

KTU PhD student in operation research in logistics specializing in agent-based modelling and machine learning.

## **Andrius Kriščiūnas**

KTU Informatics faculty lecturer & researcher specializing in optimization and approximation techniques.

Workshop focuses on basic approach and not state of the art solutions, we focus on research not production. Thus, more advanced techniques to place in TOP Kaggle places will not be presented.

# Agenda

- Introduction to real estate price estimation
- Understanding the algorithms
- Exploratory analysis of the dataset
- Features engineering
- Main algorithms usage: XGBoost, LightGBM
- Result comparison
- Useful resources
- Questions

# Goal

- How to decide for how much to sell a real estate object?
- How to evaluate tax for real estate?
- What size of a loan and what size of interests can be provided?
- Etc.

Usually expert evaluation and object comparison are used to answer the questions, and not data science approaches.

# Real estate evaluation


Real estate evaluations being used by investors, government, banks, citizens:

- To plan their households;
- To plan infrastructure projects;
- To plan bank loans and interest schemes;
- Etc.

The current methodology of real estate price evaluation usually uses expert evaluation, because:

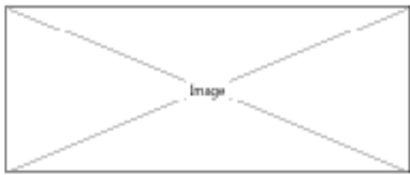
- It is difficult to integrate multiple sources of data
- The majority of experts does not comprehends mathematical evaluation methods

# R&D project RE market estimation system in LT

 Duomenų šaltiniai Analizės metodai Ekspertinis vertinimas Rodikliai ir jų palyginimas [Prisijungti](#) LT | EN

## Apie projektą

Fusce auctor orci vel magna efficitur condimentum. Sed ut elit sit amet tortor euismod dictum a ac erat. Duis fringilla eget odio sed ultrices. Proin eleifend consequat cursus. Morbi mauris elit, gravida eget elit ac, laoreet lobortis nisi. Fusce auctor nec risus at commodo. Integer ut magna arcu. Nulla dignissim tellus nec tellus eleifend pellentesque. Vivamus volutpat suscipit ultricies. Mauris vitae nisi ut orci pulvinar tristique. Nunc vehicula sapien sit amet odio euismod, et tempor odio vulputate. Quisque id magna at velit laoreet pellentesque.



## Preliminarus objekto vertės nustatymas


Objekto tipas  
Sklypai

Paskirtis  
Pramonės

Plotas (arais)

Papildomi duomenys

Vertės nustatymas

Vieta (Plinas adresas arba koordinatės)  
  


Daugiau informacijos

Duomenų tiekėjai

Naudingos nuorodos

Kontaktai

Orci vel magna efficitur  
Cras rutrum vestibulum  
interdum et malesuada

Aenean ultrices  
Nam fringilla  
Morbi eu

Mauris porta  
Sed gravida felis

email@email.am  
Pagalba

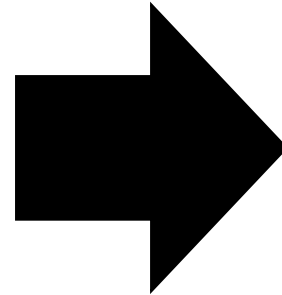
Facebook

Twitter


Google+

Facebook

Copyright © 2019. All rights reserved.



Matomas tik prisijungusiam vartotojui

 Duomenų šaltiniai Analizės metodai Ekspertinis vertinimas Rodikliai ir jų palyginimas [Atsijungti](#)


## Objekto informacija

Objekto tipas	Sklypas
Paskirtis	Pramonės

Ariamos žemės plotas 80 a.

Miško plotas 10 a.

Našumo balas 12



## Objektas 1 ( xx % )

Objekto tipas	Sklypas
Paskirtis	Pramonės

Ariamos žemės plotas 120 a.

plačiau

Sandorio metai: 2015 Sandorio vertė: 2014 EUR

Pataisos koeficientai:

Laikas	Vieta	Būklė	Kita
- 15 %	+ 10%	- 5 %	0 %

Komentaras

## Objektas 2 ( xx % )

Objekto tipas	Sklypas
Paskirtis	Pramonės

Ariamos žemės plotas 120 a.

plačiau

Sandorio metai: 2015 Sandorio vertė: 2014 EUR

Pataisos koeficientai:

Laikas	Vieta	Būklė	Kita
- 15 %	+ 10%	- 5 %	0 %

Komentaras

Pridėti objektą

Pataisos koef. įtaka	Laikas	Vieta	Būklė
	33.3%	33.3%	33.3%

Išsaugoti

Vertės skaičiavimas

Daugiau informacijos

Duomenų tiekėjai

Naudingos nuorodos

Kontaktai

Orci vel magna efficitur  
Cras rutrum vestibulum  
interdum et malesuada

Aenean ultrices  
Nam fringilla  
Morbi eu

Mauris porta  
Sed gravida felis

email@email.am  
Pagalba

Facebook

Twitter

Google+

Facebook

Copyright © 2019. All rights reserved.

# Data sources to improve estimation

- Real estate demand - <https://www.aruodas.lt/>
- Climate - [www.windguru.com](http://www.windguru.com)
- Commodity prices - [www.indexmundi.com](http://www.indexmundi.com)
- Natural resources - [www.usgs.gov](http://www.usgs.gov)
- Macro indicators – <https://www.worldbank.org/>, <https://ec.europa.eu/Eurostat>,  
<http://www.oecd.org/>, <http://data.un.org/>
- Ortophotograpy - <https://github.com/chrieke/awesome-satellite-imagery-datasets>,  
[www.copernicus.eu](http://www.copernicus.eu)
- Transport infrastructure statistics - <https://github.com/graphhopper/open-traffic-collection>
- Assocications - <https://www.fefac.eu/>
- Import, Export - <https://comtrade.un.org/>
- Google trends - <https://trends.google.com>
- Stock prices - [www.nasdaq.com](http://www.nasdaq.com)
- Unstructured data (TF-IDF) – reports, news

# Main issue

## **Different methodologies and revisions:**

- The Global Industry Classification Standard (GICS)
- Classification of Economic Activities (EVRK)
- Harmonized Commodity Description and Coding Systems
- Common classification of territorial units for statistics (NUTS)
- And so on



# Data archives for research & learning

- UCI Machine Learning Repository - <https://archive.ics.uci.edu>
- Kaggle - [www.kaggle.com](http://www.kaggle.com)
- Other?

# Which data can you use?

Which data you have before making the decision?

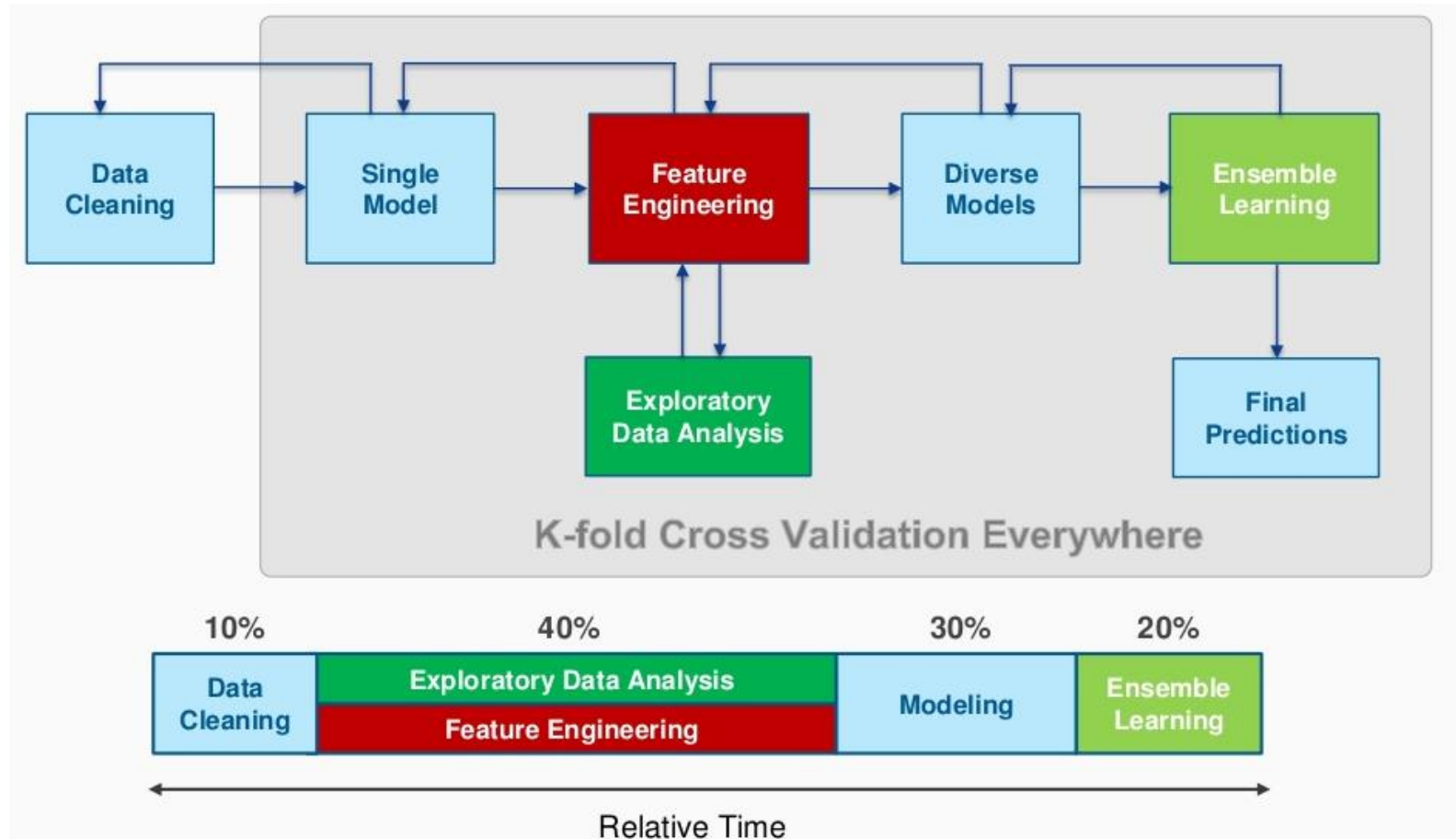


Source: <https://www.oreilly.com/ideas/drivetrain-approach-data-products>

# Kaggle competition

- Kaggle is an online community of data scientists and machine learners, that provides competitions to win prizes and develop algorithms for companies.
- Kaggle is a great tool to learn machine learning and have a benchmarking to compare your model.
- Data source used: Sberbank Russian Housing Market

# ML implementation process



# **INTRODUCTION TO MATHEMATICS BEHIND THE ML ALGORITHMS**

PhD, Andrius Kriščiūnas

# **LOAD LIBRARIES AND DATA**

# Libraries

1. Run anaconda prompt
  2. cd to directory
  3. pip install -r requirements.txt
- numpy
  - panda
  - matplotlib
  - seaborn
  - sklearn
  - xgboost
  - lightgbm
  - datetime
  - shap
  - pickle

**EXPLORE DATA**



# Exploratory analysis of the dataset

- Dependencies of target variables and additional variables
- Distribution of data
- Dependencies between target variable and additional variables
- Missing values
- Outlier detection
- Data mistake fixing

# **FEATURE ENGINEERING**

# Feature engineering

- Transform data to improve dependency, scale data, fit closer to normal distribution (e.g. log,  $1/x$  etc.)
- Identify main variable influence to target variable (e.g. correlation analysis)
- Remove similar variables (multicollinearity)
- Merge similar variables (e.g. Principal component analysis)
- Add other variables based on experience and creativity

# Work automation

## Pipeline

Sequentially apply a list of transforms and a final estimator. Intermediate steps of the pipeline must be 'transforms', that is, they must implement fit and transform methods.

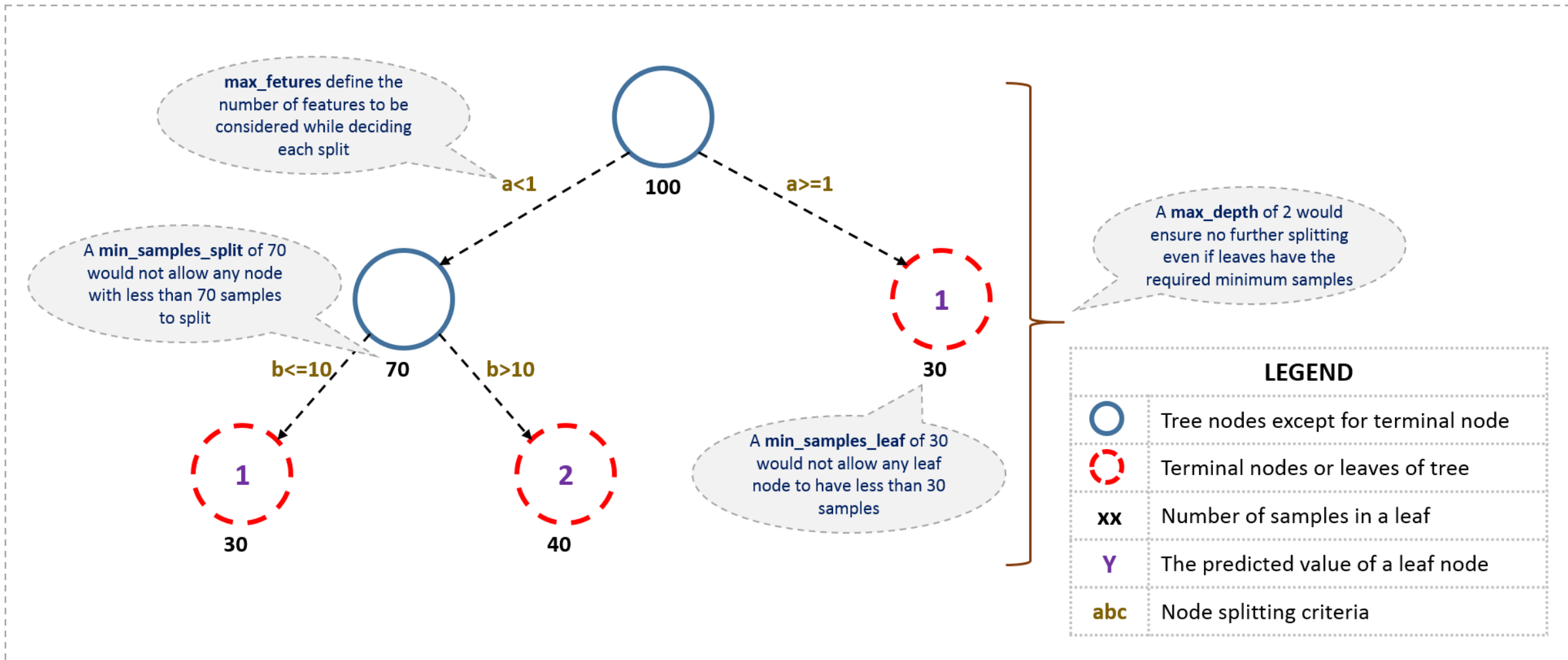
E.g. Sklearn pipeline

# **PARAMETER SELECTION**

# XGBoost and LightGBM

- **XGBoost** is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable.
- **LightGBM** is a gradient boosting framework that uses tree based learning algorithms.
- **Common parameters:**
  - **objective** - specify the learning task and the corresponding learning objective;
  - **eval\_metric** - evaluation metrics for validation data
  - **learning\_rate** - The parameter controls the magnitude of this change in the estimates.
  - **max\_depth** This indicates how deep the built tree can be.
  - **n\_estimators** represents the number of trees in the forest

# Decision tree structure



# XGBoost parameters

- **min\_samples\_split** represents the minimum number of samples required to split an internal node
- **min\_samples\_leaf** - min\_samples\_leaf is The minimum number of samples required to be at a leaf node
- **max\_features** represents the number of features to consider when looking for the best split
- **Subsample** - subsample ratio of the training instance
- **colsample\_bytree** - subsample ratio of columns when constructing each tree.
- **min\_child\_weight** - minimum sum of instance weight (hessian) needed in a child.
- **Gamma** - minimum loss reduction required to make a further partition on a leaf node of the tree.

**Source:** <https://xgboost-clone.readthedocs.io/en/latest/parameter.html>



# LightGBM parameters

**sub\_feature** - will randomly select part of features on each iteration if feature\_fraction smaller than 1.0

**num\_leaves** - max number of leaves in one tree

**min\_data** - minimal number of data in one leaf.

**max\_bin** - max number of bins that feature values will be bucketed in

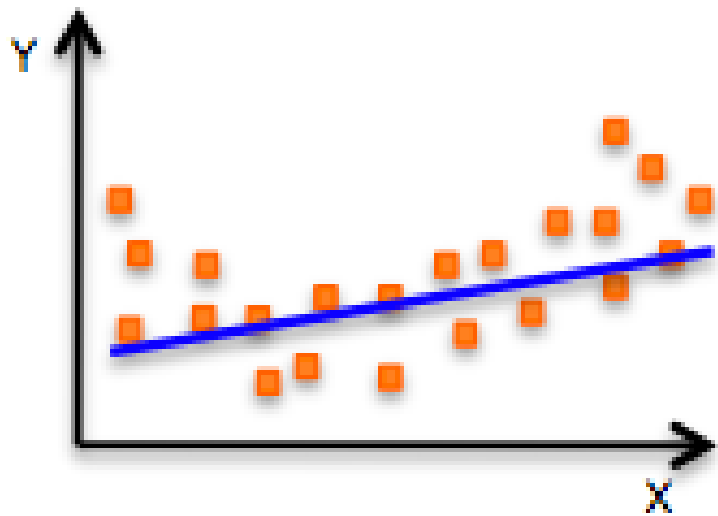
**bagging\_freq** - frequency for bagging

**Source:** <https://lightgbm.readthedocs.io/en/latest/Parameters.html>

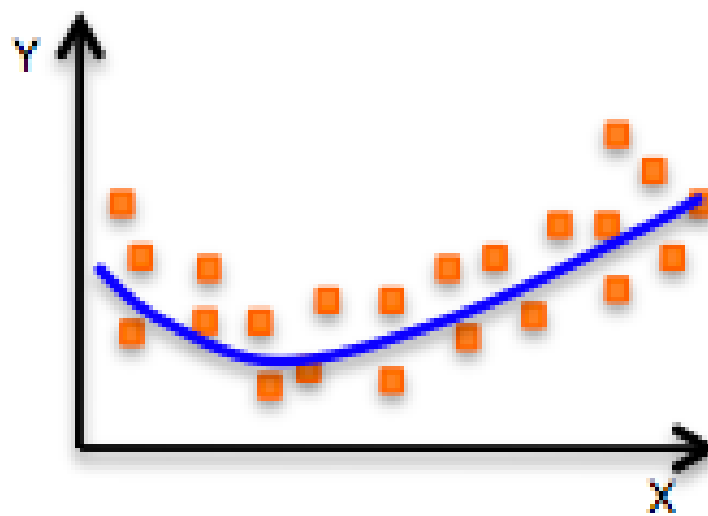
# Read more

- **Bias VS. Variance** - <https://becominghuman.ai/machine-learning-bias-vs-variance-641f924e6c57>
- **XGBoost** - <https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/>
- **LightGBM** - <https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc>
- **Comparison** - <https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db>

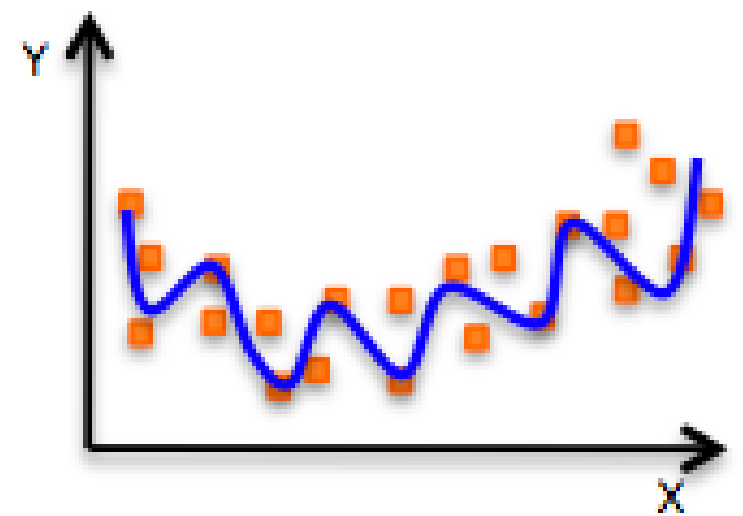
# Model must be able to generalize



Underfitting



Just right!

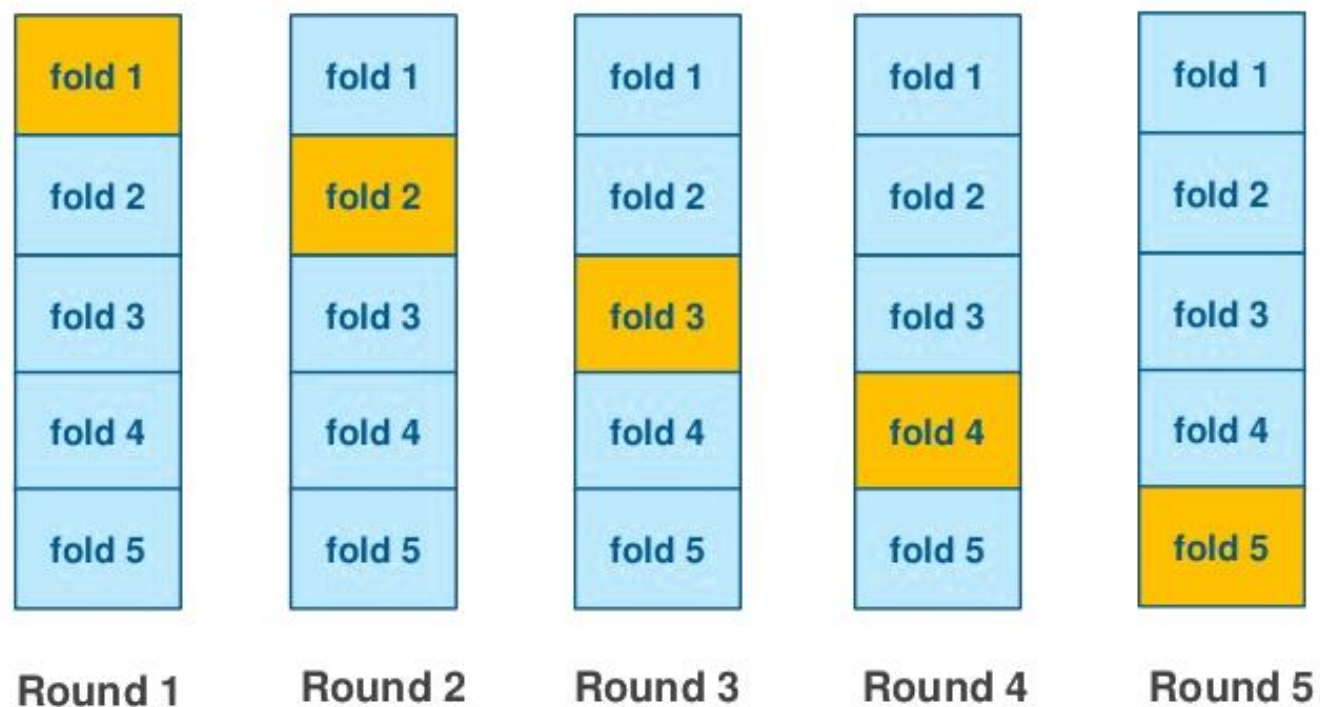


overfitting

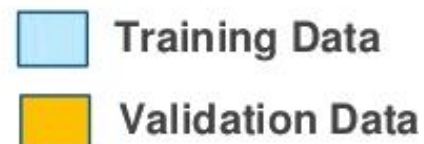
# Choosing the parameters

- Understanding the data
- Understanding the parameter influence to the output
- Validating the model
- Hyper parameter optimization E.g. Sklearn Gridsearchcv
- Cross validation e.g. Sklearn cross\_validate

# Cross validation



*score(CV) = the average of evaluation scores from each fold*  
*You can also repeat the process many times!*

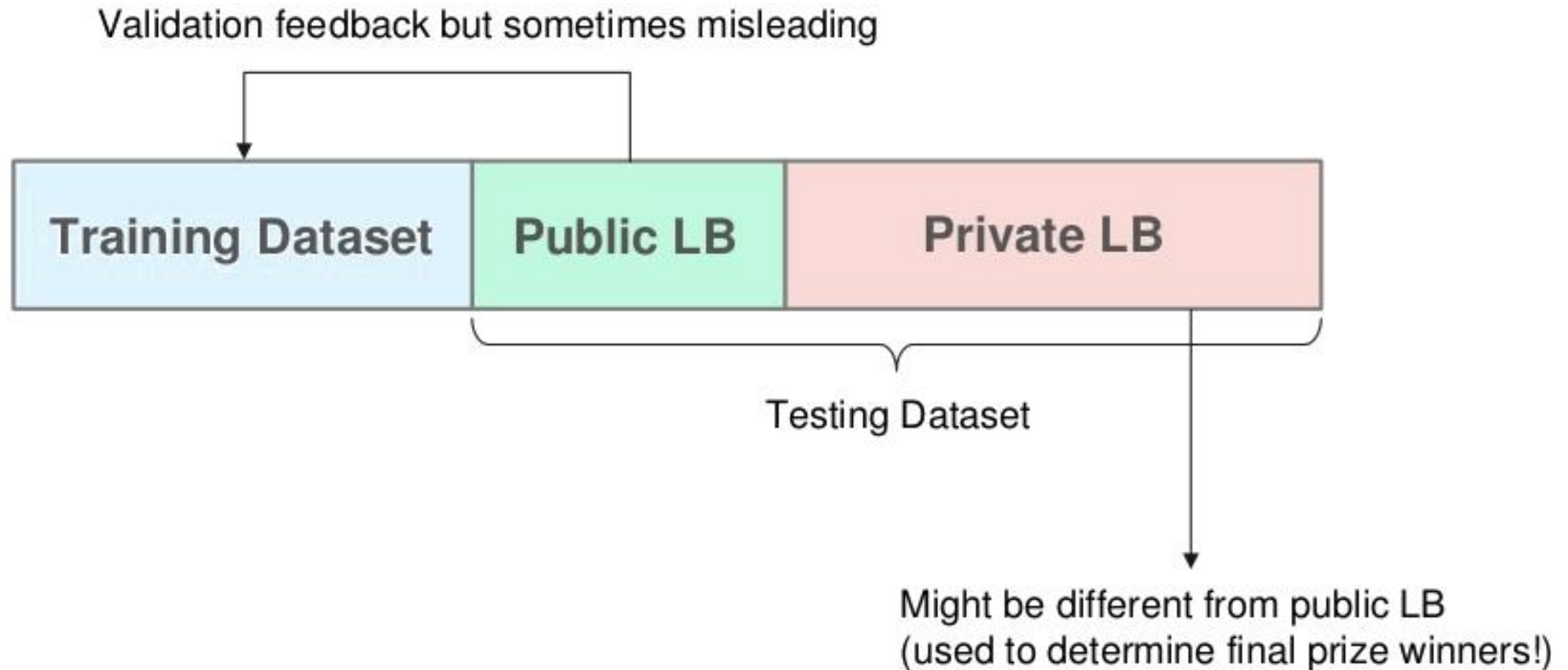


# Model validation

- **Training Set**
  - this data set is used to adjust the weights
- **Validation Set**
  - this data set is used to minimize overfitting.
- **Testing Set**
  - this data set is used only for testing the final solution in order to confirm the actual predictive power of the network.

**Important!** Kaggle: public vs private leader boards

# Why validation is important?



# **FEATURE IMPORTANCE**



# Feature importance

- Move back to step 1: Data exploration
- Use model feature importance libraries e.g. sklearn feature\_selection SelectFromModel, Shap interpreter
- Feature importance determination uses different approaches, e.g. R2

# Additional resources

- Basic courses for excel and so on, <https://www.lynda.com/>
- Deep learning course , <https://www.fast.ai/>
- Dive into Deep Learning, <https://d2l.ai/>
- Agent-based modelling, complexity approach, <https://www.complexityexplorer.org/>
- Micromasters programmes, <https://www.edx.org/>
  - Program in Supply Chain Management, Massachusetts Institute of Technology
  - Business and Economics for a Circular Economy, Wageningen University & Research
- Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.
- Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems* (pp. 3146-3154).
- Zhou, Z. H., & Feng, J. (2017). Deep forest: Towards an alternative to deep neural networks. *arXiv preprint arXiv:1702.08835*.
- Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363*.

# Working

## **1. Estimate the given real estate price:**

- Use file in data folder - New customer request.xlsx

## **2. Improve the RMSE metric and submit to kaggle:**

- use LightGBM notebook (LightGBM implementation)
- Implement XGBoost (Implement XGboost, Step by step)