Automated valuation of commercial property

Agenda

- Introduction & Motivation
- Workflow of this project
- Data Extraction
- Data Preprocessing
- Explanation of ML models used
- Evaluation of models
- Results & Discussion
- Conclusion & Outlook

Introduction & Motivation

Main challenges in valuation of commercial property

- Lack of good reference objects.
- Uniqueness of each individual object.
- High sensitivity to market fluctuations.
- Require long and intensive market analysis.

My approach for automated valuation of commercial property

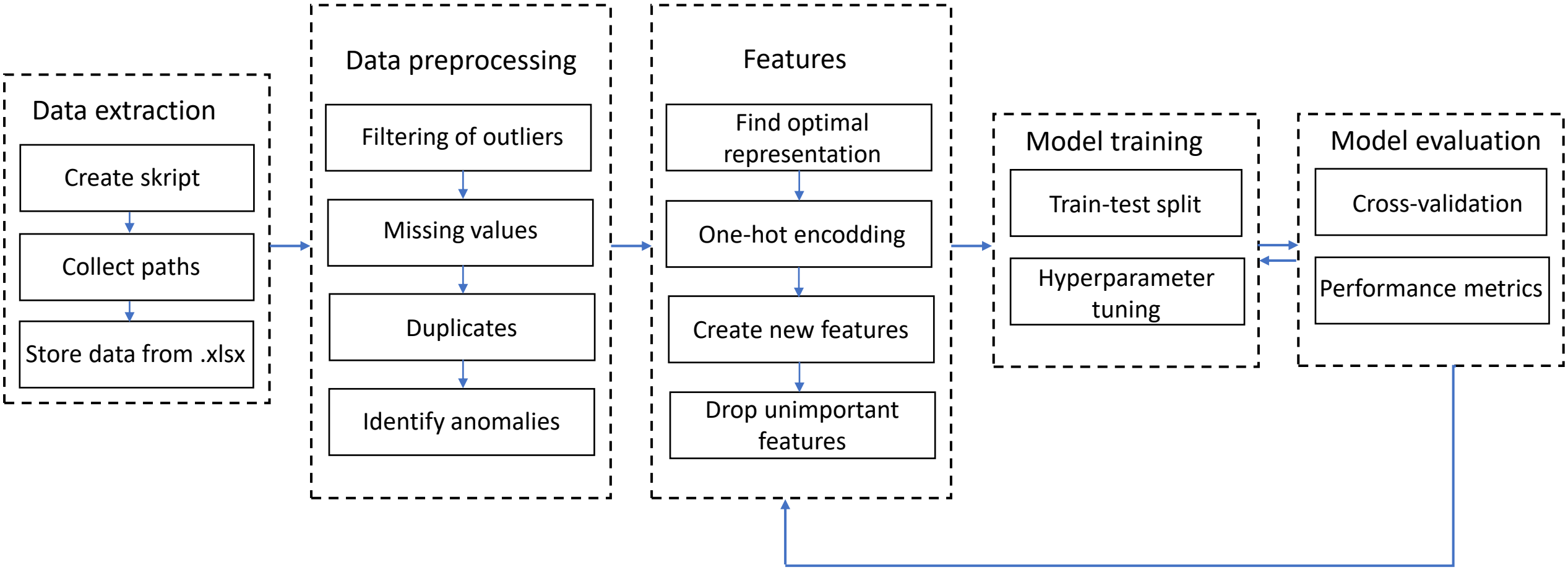
- Use in-house collected data for ML application.
- Use long-term commercial valuation experience for proper data preparation.
- Apply most used for housing prices ML models.
- *Future: Integrate it in the current valuation tool.*

Literature review

- Housing Price Prediction Using ML Algorithms in COVID-19 Times (Raul-Tomas Mora-Garcia, November 2022).
- Using ML to Evaluate Real Estate Prices Using Location Big Data (Walter Coleman, May 2022).
- Housing Price Prediction via Improved ML Techniques (Quang Truong, July 2020).
- House Price Prediction using Random Forest Machine Learning Technique (Abigail Bola Adetunji, February 2022).

Workflow in this project

Workflow


Data extraction
Create skript
Collect paths
Store data from .xlsx
Data preprocessing
Filtering of outliers
Missing values
Duplicates
Identify anomalies
Features
Find optimal representation
One-hot encodding
Create new features
Drop unimportant features
Model training
Train-test split
Hyperparameter tuning
Model evaluation
Cross-validation
Performance metrics

Data Extraction

Data Extraction

- Store paths to the all .xlsx files in a dictionary
- Use city name as a key and list of paths to all valuations in this city
- Iterate through each list of paths and extract needed features from .xlsx in the separate list
- List of features -> one object

```python
def data_reader(excelPaths):
    dataset = []
    for key, value in excelPaths.items():
        for path in value[0]:
            x = pd.read_excel(path, sheet_name=['Mieterliste', 'Internationales Verfahren DCF'], header=None)
            #Time of the last modification in seconds
            ti_m = os.path.getmtime(path)
            #Time of the last modification standart
            m_ti = time.ctime(ti_m)
            features = collect_data(x)
            features.insert(0, key)
            features.append(ti_m)
            features.append(m_ti)
            dataset.append(features)
        df = pd.DataFrame(dataset, columns =['City', 'Address', 'LivingArea', 'CommercialArea',
                                             'ANCR', 'DCF_Price', 'AMR', 'MaintenanceReserve', 'Capex',
                                             'LM_Timecode', 'LM_date'])
        writeFiles(df, key)
    return dataset
```

```python
def collect_data(excelFile):
    infos = []
    infos.append(str(excelFile['Internationales Verfahren DCF'].loc[1, 3]))          #Adresse
    infos.append(excelFile['Mieterliste'].loc[26, 8])                                #Wohnfläche
    infos.append(excelFile['Mieterliste'].loc[27, 8])                                #Gewerbefläche
    infos.append(excelFile['Internationales Verfahren DCF'].loc[82, 3])              #JNKM
    infos.append(excelFile['Internationales Verfahren DCF'].loc[12, 8])              #DCF Preis
    infos.append(excelFile['Internationales Verfahren DCF'].loc[26, 8])              #m²-Miete Marktüblich
    infos.append(excelFile['Internationales Verfahren DCF'].loc[48, 8])              #Instandhaltungsrücklage
    infos.append(excelFile['Internationales Verfahren DCF'].loc[50, 8])              #Capex
    return infos
```

Features used in this project

Feature names	Feature description
City	Name of the city (location)
Address	Adress of the object
LivingArea	Living area of the house
CommercialArea	Commercial area of the house
ANCR	Annual net cold rent
AMR	Annual market rent
MaintenanceReserve	Annual Maintenance Reserve (per sqm)
Capex	Annual Capital Expenditures (per sqm)
LM_Timecode	Exact date and time of valuation (as timecode)
LM_date	Date of valuation (DD-MM-YY format)
Market_sqm	Market sqm-price of MFHs in this city
Vacancy rate	The percentage of not-rented flats in the city
DCF_Price	Valuation price (target variable)

Data Preprocessing

Data Preprocessing

- Limit the project for Top-100 largest cities
- Add additional feature „Factor“ to simplify the process of finding outliers
- **Filter out outliers:**
- By living area 250 m² ≤ Area ≤ 2.000 m²
- By DCF-Price 140K ≤ Price ≤ 3,5M €
- By AMR 15K ≤ AMR ≤ 250K € (for the whole area p.a.)

Data Preprocessing

- Drop objects with missing values (approx. 1% of dataset)
- Filter out objects outside of market Factor range
  ($8 \leq \text{Factor} \leq 40$)
- Remove duplicates (only completely identical objects)
- Limit for multi-family houses + residential and commercial buildings with less than 15% of commercial area
- Add new features (market sqm & vacancy rate)

Data Preprocessing (One-hot encodding)

	City	LivingArea	CommercialArea	ANCA	DCF_Price	AMP	MaintenanceReserve	Capex	LM_Timecode	Province	Market_sqm	Leerstandsquote
0	Berlin	306.98	0.00	119.711903	2678.339140	94.92	9.700	9.300	8	Berlin	2500	0.8
1	Berlin	549.98	0.00	115.592567	3093.446560	86.88	9.000	6.500	8	Berlin	2500	0.8
2	Berlin	1057.80	53.35	133.237367	2883.439300	138.00	10.500	7.000	5	Berlin	2500	0.8
3	Berlin	426.38	0.00	97.715653	2292.734181	109.80	9.742	9.334	20	Berlin	2500	0.8
4	Berlin	570.00	0.00	70.168421	1753.663158	109.80	9.742	9.334	20	Berlin	2500	0.8

	LivingArea	CommercialArea	City_Berlin	City_Hamburg	2_2023	2_2022	Province_Hessen	Province_Berlin
0	306.98	0.00	1.0	0.0	0.0	0.0	0.0	1.0
1	549.98	0.00	1.0	0.0	0.0	0.0	0.0	1.0
2	1057.80	53.35	1.0	0.0	0.0	0.0	0.0	1.0
3	426.38	0.00	1.0	0.0	0.0	0.0	0.0	1.0
4	570.00	0.00	1.0	0.0	0.0	0.0	0.0	1.0

Explanation of ML models used

Baseline models

- A baseline model is essentially a simple model that acts as a reference in a machine learning project.

- Its main function is to contextualize the results of trained models.

- **Simple linear regression (with AMR only)** is used as baseline in this project

[5] A. Nair, 'Baseline Models: Your Guide For Model Building'. [Online]. Available: https://towardsdatascience.com/baseline-models-your-guide-for-model-building-1ec3aa244b8d

Linear models: Simple linear regression

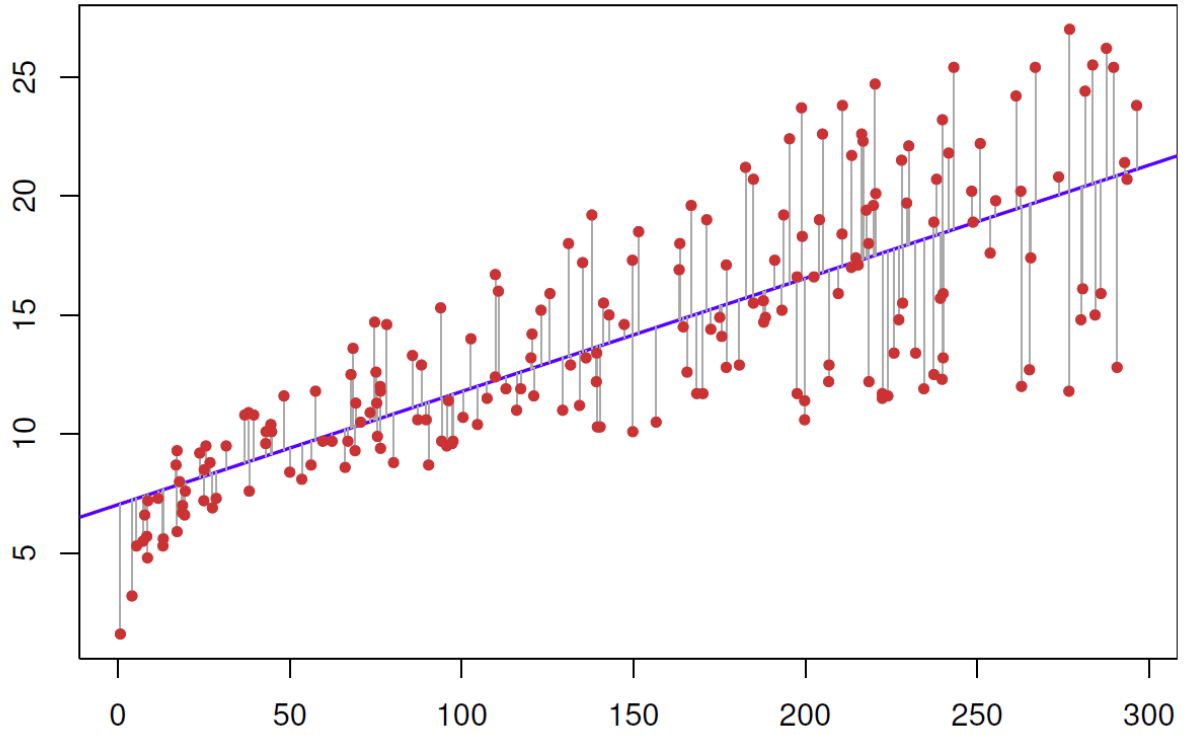
$$y_i = b_0 + b_1 x_{i1} + \cdots + b_{k-1} x_{ik-1} + \epsilon_i$$

$$\boldsymbol{Y} = \boldsymbol{XB} + \epsilon$$

$$\boldsymbol{min}_{\boldsymbol{b}} \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \left(y_i - \sum_{\boldsymbol{j=0}}^{\boldsymbol{k-1}} b_j x_{ij} \right)^{\boldsymbol{2}}$$

$$\widehat{\boldsymbol{B}} = \left(\boldsymbol{X^t X}\right)^{-1} \boldsymbol{X^t Y}$$

$$\widehat{\boldsymbol{Y}} := \boldsymbol{X}\widehat{\boldsymbol{B}}$$



[6] L. E. Melkumova and S. Ya. Shatskikh, 'Comparing Ridge and LASSO estimators for data analysis', Procedia Engineering, vol. 201, pp. 746–755, 2017, doi: 10.1016/j.proeng.2017.09.615.
[7] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, vol. 103. in Springer Texts in Statistics, vol. 103. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-7138-7.

Linear models: Lasso & Ridge regression

Lasso regression

$$\min_b \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k-1} b_j x_{ij} \right)^2 + \lambda \|B\|$$

Lasso regression:

- Shrink the coefficients
- Completely nullify some of them (feature selection)

Ridge regression

$$\min_b \sum_{i=1}^{n} \left(y_i - \sum_{j=0}^{k-1} b_j x_{ij} \right)^2 + \lambda \|B\|^2$$

Ridge regression:

- Shrink the coefficients

Regularization: More stable solutions

[6] L. E. Melkumova and S. Ya. Shatskikh, 'Comparing Ridge and LASSO estimators for data analysis', Procedia Engineering, vol. 201, pp. 746–755, 2017, doi: 10.1016/j.proeng.2017.09.615.

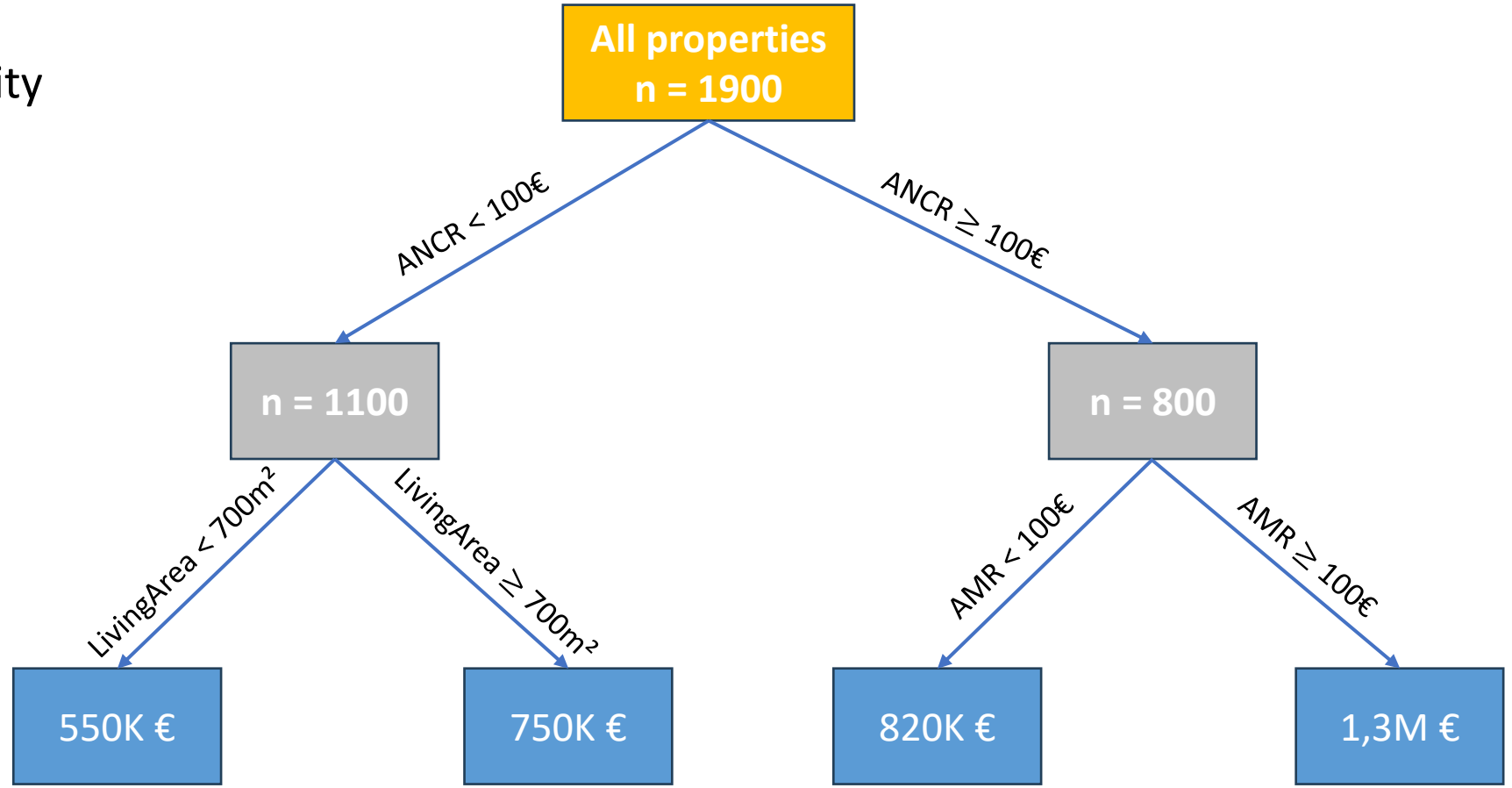
Decision Tree

Pros:

- Able to deal with non-linearity
- Easy to interpret
- Fast trainable algorithm

Contra:

- Tend to overfit the data
- Non-robust (sensitive to small changes in the data)



[8] Adopted from: 'Algorithm Selection for Machine Learning', 14.07.2022. [Online]. Available: https://elitedatascience.com/algorithm-selection

[9] A. Amro, M. Al-Akhras, K. E. Hindi, M. Habib, and B. A. Shawar, 'Instance Reduction for Avoiding Overfitting in Decision Trees', Journal of Intelligent Systems, vol. 30, no. 1, pp. 438–459, Jan. 2021, doi: 10.1515/jisys-2020-0061.

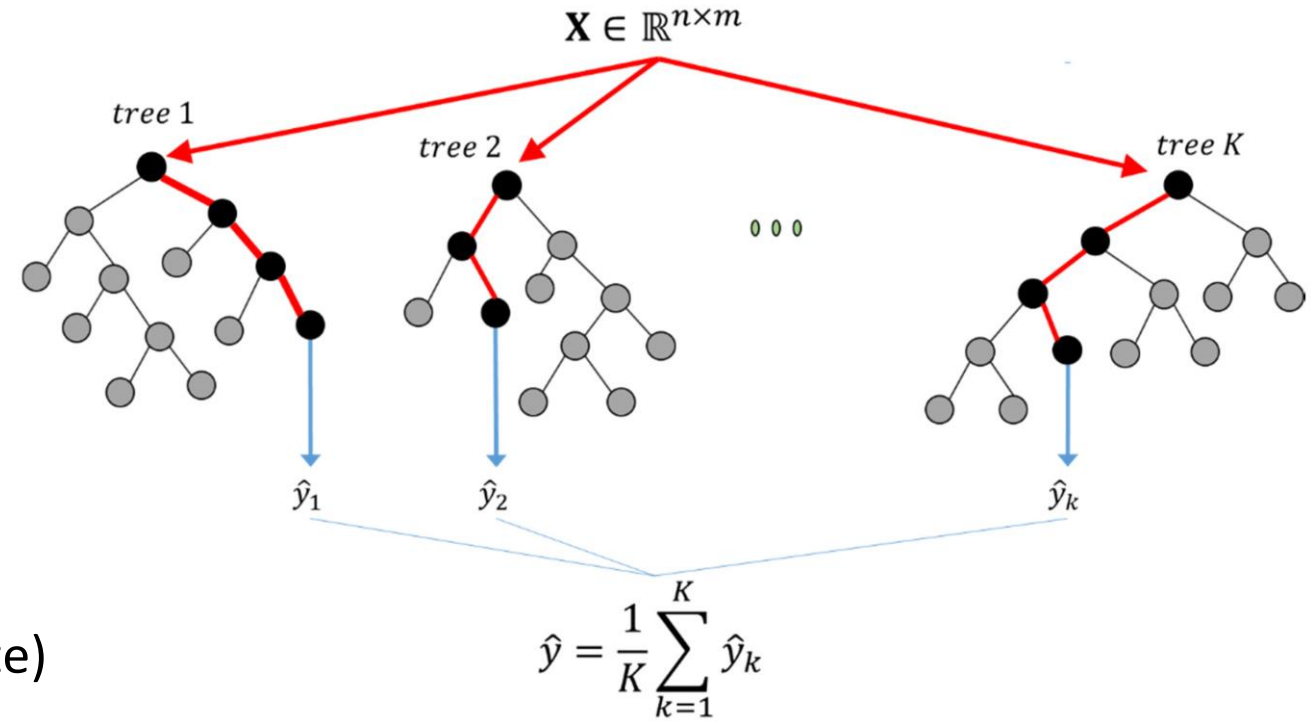
Ensemble models: Random forest

Pros:

- More generalized solution
- Robust to outliers
- Lower risc of overfitting
- Able to deal with non-linearity
- High predictive accuracy

Contra:

- Black box algorithm (difficult to interprate)
- Longer training time compared to DT



[10] C. Aldrich, 'Process Variable Importance Analysis by Use of Random Forests in a Shapley Regression Framework', Minerals, vol. 10, no. 5, p. 420, May 2020, doi: 10.3390/min10050420.

[11] M. Aria, C. Cuccurullo, and A. Gnasso, 'A comparison among interpretative proposals for Random Forests', Machine Learning with Applications, vol. 6, p. 100094, Dec. 2021, doi: 10.1016/j.mlwa.2021.100094.

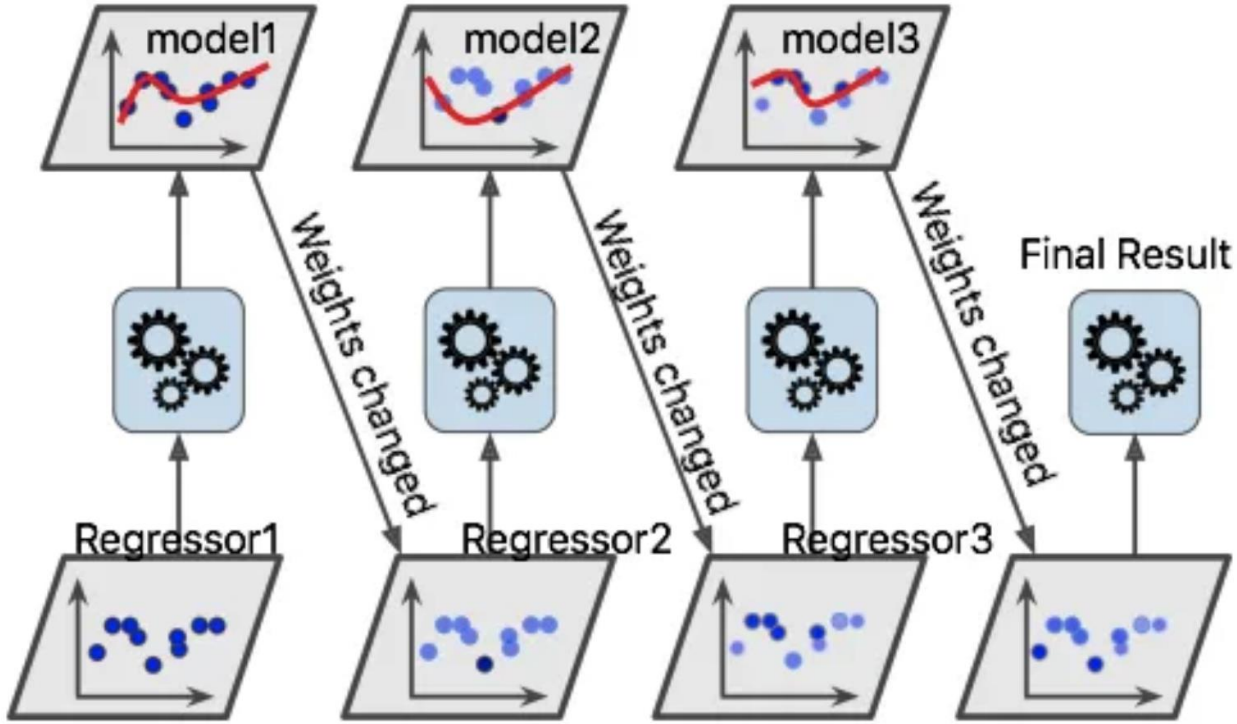
Ensemble models: AdaBoost

Pros:

- Less hyperparameters
- Often more accurate for difficult objects
- Robust to multi-collinearity
- Able to deal with non-linearity

Contra:

- Slower than Random Forest
- Very sensitive to outliers an noise data



[12] R. Madan, 'Gradient boosting Vs AdaBoosting — Simplest explanation of how to do boosting using Visuals and Python Code'. [Online]. Available: https://medium.com/@madanflies/gradient-boosting-vs-adaboosting-simplest-explanation-of-how-to-do-boosting-using-visuals-and-d5939133b435
[13] G. Shanmugasundar, M. Vanitha, R. Čep, V. Kumar, K. Kalita, and M. Ramachandran, 'A Comparative Study of Linear, Random Forest and AdaBoost Regressions for Modeling Non-Traditional Machining', Processes, vol. 9, no. 11, p. 2015, Nov. 2021, doi: 10.3390/pr9112015.

Ensemble models: LightGBM

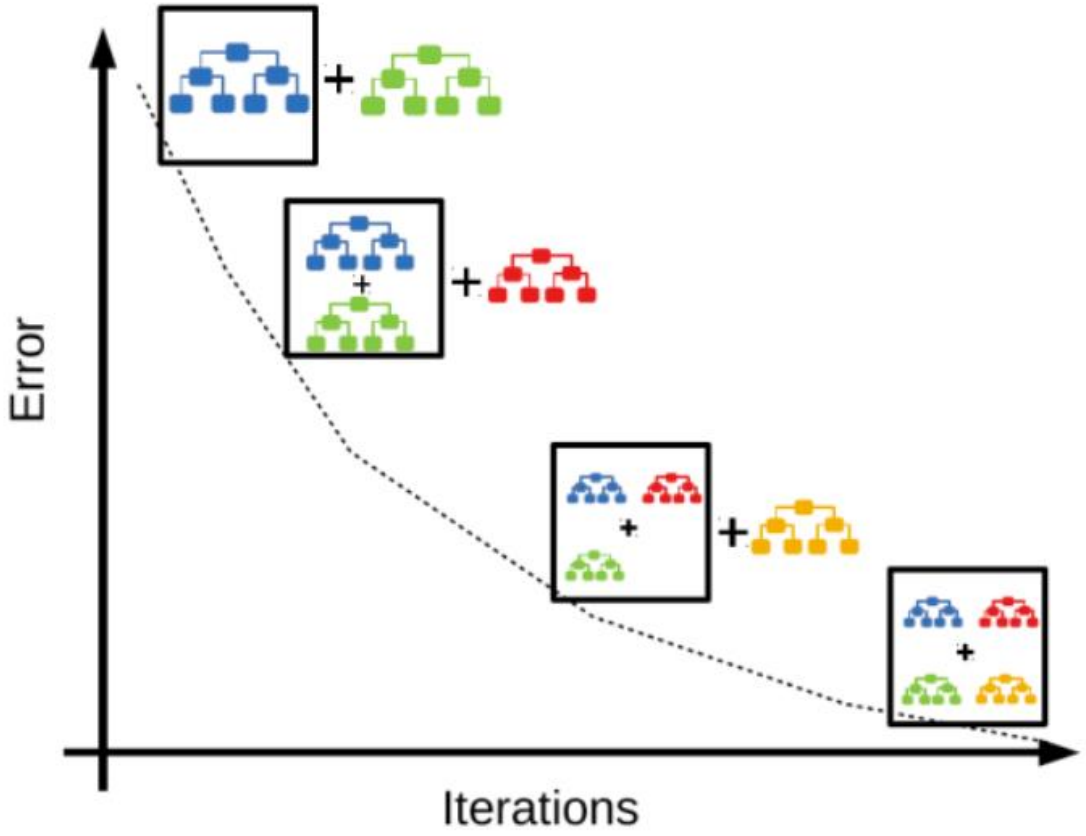
Pros:

- Often faster than other Boosting algorithms
- Can handle categorical features

  (also integrated in implementation)
- Often higher performance
- Large space for optimization

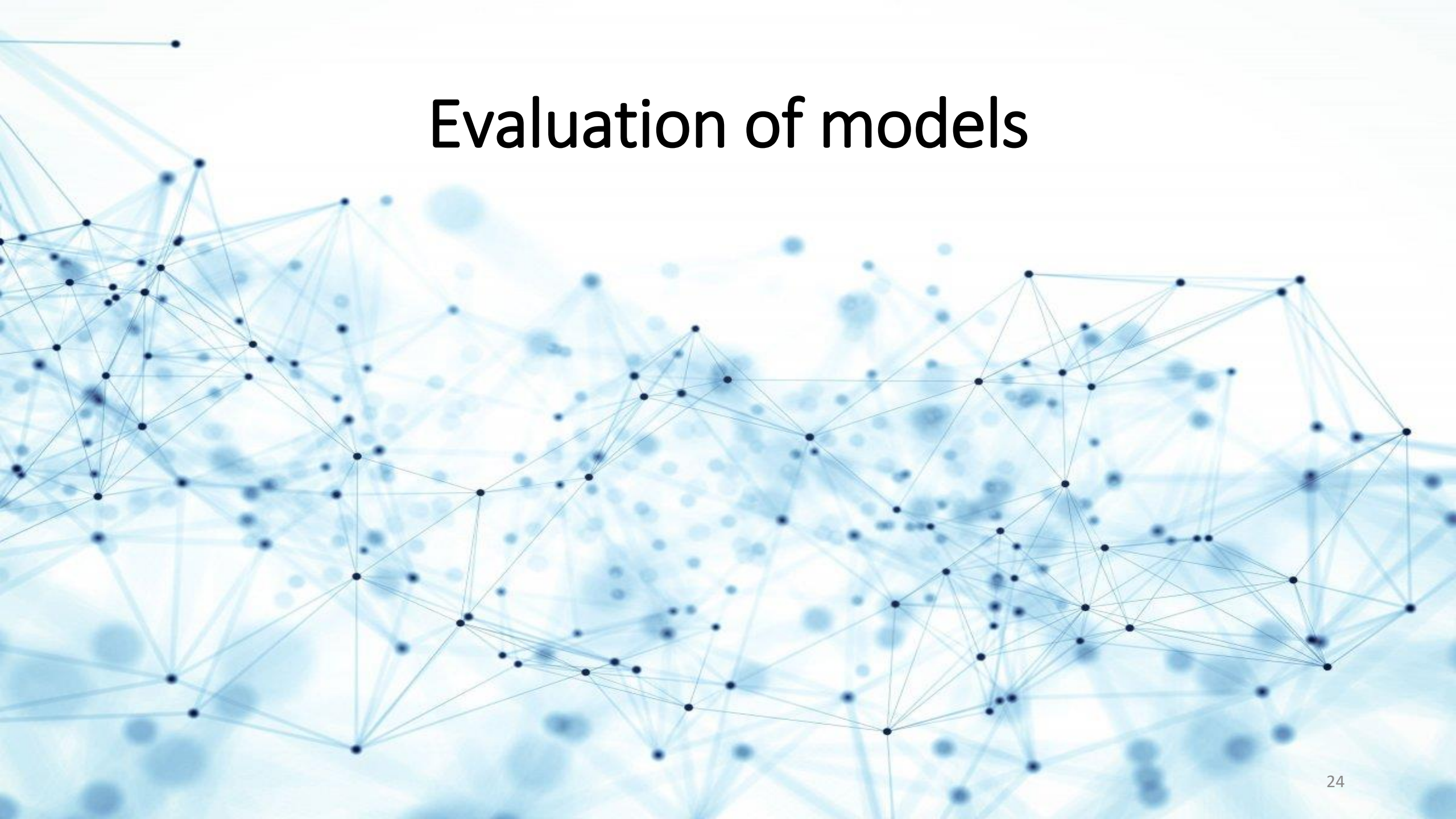
Contra:

- Difficult to optimize
- Tend to overfit the data



[14] S. Saha, 'XGBoost vs LightGBM: How Are They Different'. [Online]. Available: https://neptune.ai/blog/xgboost-vs-lightgbm
[15] F. Wang, H. Cheng, H. Dai, and H. Han, 'Freeway Short-Term Travel Time Prediction Based on LightGBM Algorithm', IOP Conf. Ser.: Earth Environ. Sci., vol. 638, no. 1, p. 012029, Feb. 2021, doi: 10.1088/1755-1315/638/1/012029.

Evaluation of models



Train-test split

- Split the entire dataset on two sub-samples: train data and test data
- Used to control the model performance on the new unseen data and to control overfitting
- 80% of the data for training & 20% for the testing

Cross validation

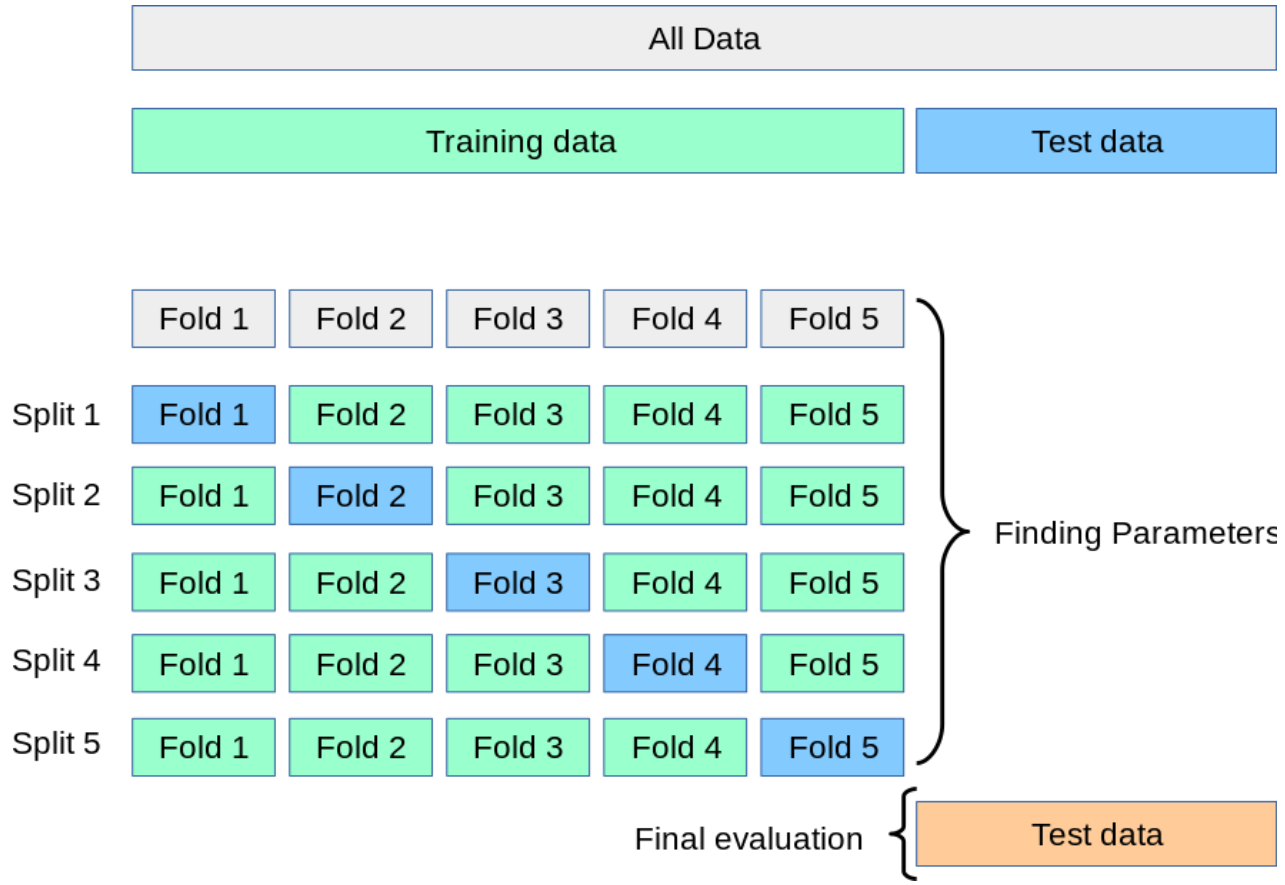


Figure 1. *Visualization of the process of hyperparameter tuning using 5-fold cross-validation.*

[16] 'Scikit-learn Machine Learning in Python'. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html

Evaluation metrics

Mean **A**bsolute **E**rror $$\boldsymbol{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i(x_i)|$$

Median **A**bsolute **E**rror $$\boldsymbol{MDAE} = \boldsymbol{med}\,|y_i - \hat{y}_i(x_i)|$$

Mean **S**quared **E**rror $$\boldsymbol{MSE} = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - \hat{y}_i(x_i)\right)^2$$

- **Median** metrics: Less influenced by outliers
- **Mean** metrics: Good to control large error minimization

[17] M. Steurer, R. J. Hill, and N. Pfeifer, 'Metrics for evaluating the performance of machine learning based automated valuation models', Journal of Property Research, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.

Evaluation metrics

Mean **A**bsolute **P**ercentage **E**rror $$\boldsymbol{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|1 - \frac{\widehat{y_i}(x_i)}{y_i}\right|$$

Median **A**bsolute **P**ercentage **E**rror $$\boldsymbol{MDAPE} = \boldsymbol{med}\left|1 - \frac{\widehat{y_i}(x_i)}{y_i}\right|$$

- More informative metrics for the price prediction tasks
- Metrics that take into account differences in price categories

[17] M. Steurer, R. J. Hill, and N. Pfeifer, 'Metrics for evaluating the performance of machine learning based automated valuation models', Journal of Property Research, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.

Evaluation metrics

Coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y_i}(x_i)\right)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Other metrics

$$\mathbf{5 - 10 - 20\% - \textit{error interval}}$$

- **R^2** shows how much variance is explained by the model

[17] M. Steurer, R. J. Hill, and N. Pfeifer, 'Metrics for evaluating the performance of machine learning based automated valuation models', Journal of Property Research, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.

Feature importance

- **Features** impact the target variable **differently strong**.
- **Feature importance** is a measure of how important this particular feature for the prediction.
- Different ways to measure feature importance dependent on the model.
- **Decision Tree** and **Random Forest**: Normalized total reduction of the MSE after all splits.
- **LightGBM**: How often this feature was used for the split.

[18] M. Saarela and S. Jauhiainen, 'Comparison of feature importance measures as explanations for classification models', SN Appl. Sci., vol. 3, no. 2, p. 272, Feb. 2021, doi: 10.1007/s42452-021-04148-9.

Results & Discussion

Experiment 1

Model	MDPAE_Train	MDPAE_Test	20%_Train	20%_Test
Baseline	20.81	20.25	0.48	0.49
LinReg	10.15	10.66	0.8	0.78
Lasso	9.96	11.05	0.8	0.76
Ridge	10.06	10.6	0.8	0.77
DecTree	11.44	12.84	0.75	0.7
Random Forest	10.61	11.55	0.8	0.74
AdaBoost	14.43	14.84	0.64	0.63
LightGBM	8.58	10.01	0.87	0.81

Table 1. *MDAPE and 20% on both train and test data for the dataset without filtering of objects with anomalies, for the whole house price and with all cities independent on the number of objects. Although the LightGBM ensemble model is the most powerful for these datasets, all linear models show almost comparable performance and outperform other ensemble methods as well as the decision tree model.*

Experiment 2

Model	MDPAE_Train	MDPAE_Test	20%_Train	20%_Test
Baseline	15.68	15.51	0.59	0.6
LinReg	8.57	9.32	0.86	0.83
Lasso	8.41	9.32	0.86	0.85
Ridge	8.61	8.85	0.86	0.84
DecTree	10.59	12.13	0.78	0.74
Random Forest	9.84	10.87	0.82	0.78
AdaBoost	12.4	12.61	0.72	0.69
LightGBM	8.43	9.21	0.88	0.84

Table 2. *MDAPE and 20% on both train and test data for the dataset without filtering of objects with anomalies and with all cities independent on the number of objects, but for the sqm-house price. While the ensemble models show only a slight improvement in quality, the linear models have become noticeably more powerful.*

Experiment 3

Model	MDPAE_Train	MDPAE_Test	20%_Train	20%_Test
Baseline	15.5	16.13	0.61	0.59
LinReg	8.03	8.18	0.89	0.87
Lasso	8.01	8.81	0.89	0.86
Ridge	8.02	8.73	0.89	0.87
DecTree	10.34	11.0	0.81	0.77
Random Forest	9.36	11.39	0.85	0.78
AdaBoost	12.29	12.11	0.72	0.72
LightGBM	7.77	8.65	0.9	0.86

Table 3. *MDAPE and 20% on both train and test data for the dataset. Here the objects with anomalies were filtered out. All cities independent on the number of objects were left in the dataset, ANCR, AMR and DCF-price are represented for sqm. After this filtering, the quality of most of the models increased.*

Experiment 4

Model	MDPAE_Train	MDPAE_Test	20%_Train	20%_Test
Baseline	15.5	15.46	0.61	0.61
LinReg	8.04	8.74	0.89	0.87
Lasso	8.09	8.58	0.89	0.86
Ridge	8.1	9.03	0.89	0.86
DecTree	10.07	11.45	0.8	0.75
Random Forest	9.62	10.68	0.84	0.8
AdaBoost	12.0	12.42	0.72	0.71
LightGBM	7.95	9.04	0.89	0.86

Table 4. *MDAPE and 20% on both train and test data for the dataset. Here the objects with anomalies were filtered out. Only the cities with more than three objects were left in the dataset, ANCR, AMR and DCF-price are represented for sqm. This filtering didn't bring any significant improvement.*

Feature Importance

Features DecTree	Feat_Imp DecTree	Features RandFor	Feat_Imp RandFor	Features LightGBM	Feat_Imp LightGBM
ANCR	0.822	ANCR	0.714	ANCR	204
Market_sqm	0.156	AMR	0.153	AMR	117
AMR	0.017	Market_sqm	0.122	City	66
Vacancy rate	0.003	Vacancy rate	0.004	Market_sqm	52
Capex	0.001	Capex	0.003	LM_Timecode	51
LivingArea	0.001	MaintenanceReserve	0.002	LivingArea	40
City_Offenbach am Main	0.0	LivingArea	0.002	Province	20
City_Mülheim an der Ruhr	0.0	Province_Sachsen	0.0	Vacancy rate	20
City_München	0.0	Province_Berlin	0.0	Capex	16
City_Münster	0.0	City_Berlin	0.0	MaintenanceReserve	11

Table 5. *Feature importances of 10 most important features for Decision Tree, Random Forest and LightGBM models. Features importances in LightGBM model are shown as the number of times they were used for the split. And for decision tree and random forest models the importance of a feature is shown as the normalized total reduction of the MSE brought by that feature after all splits.*

Experiment 5

Model	MDPAE_Train	MDPAE_Test	20%_Train	20%_Test
Baseline	15.47	15.83	0.61	0.59
LinReg	11.45	13.82	0.74	0.68
Lasso	11.85	12.0	0.73	0.72
Ridge	11.69	12.87	0.73	0.7
DecTree	13.1	13.62	0.69	0.65
Random Forest	12.18	13.81	0.72	0.68
AdaBoost	16.31	17.0	0.58	0.57
LightGBM	11.31	12.58	0.75	0.69

Table 6. *MDAPE and 20%-error interval on both train and test data for the dataset. Here the objects with anomalies were filtered out. All cities independent on the number of objects were left in the dataset, AMR and DCF-price are represented for sqm, but ANCR is removed.*

Most wrong predictions vs targets

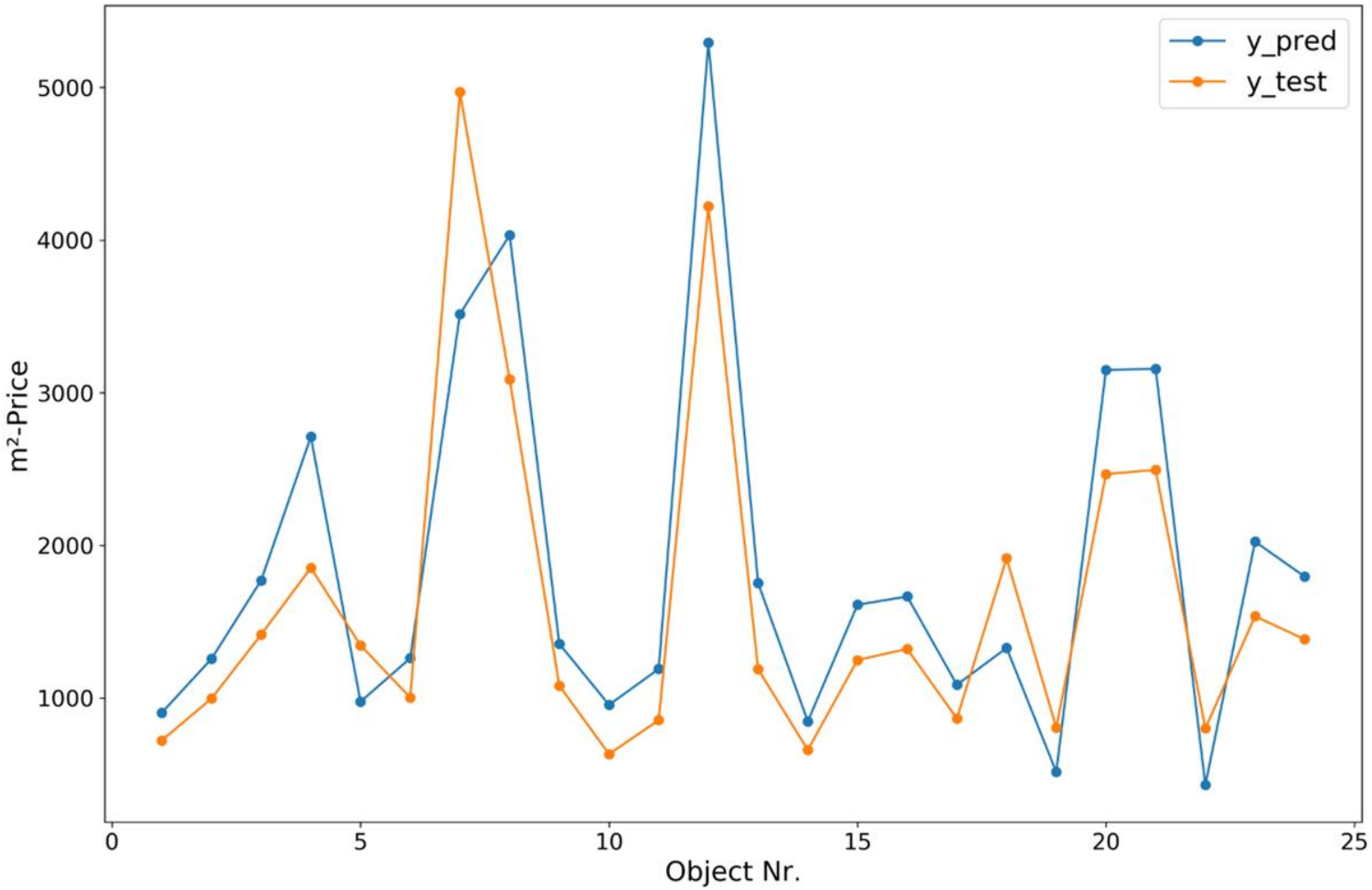


Figure 2. *Comparison of prediction prices of linear regression model to real targets for the objects with the largest prediction error of more than 25%. In most of the cases the model tended to overestimate the price.*

Residuals distribution linear regression

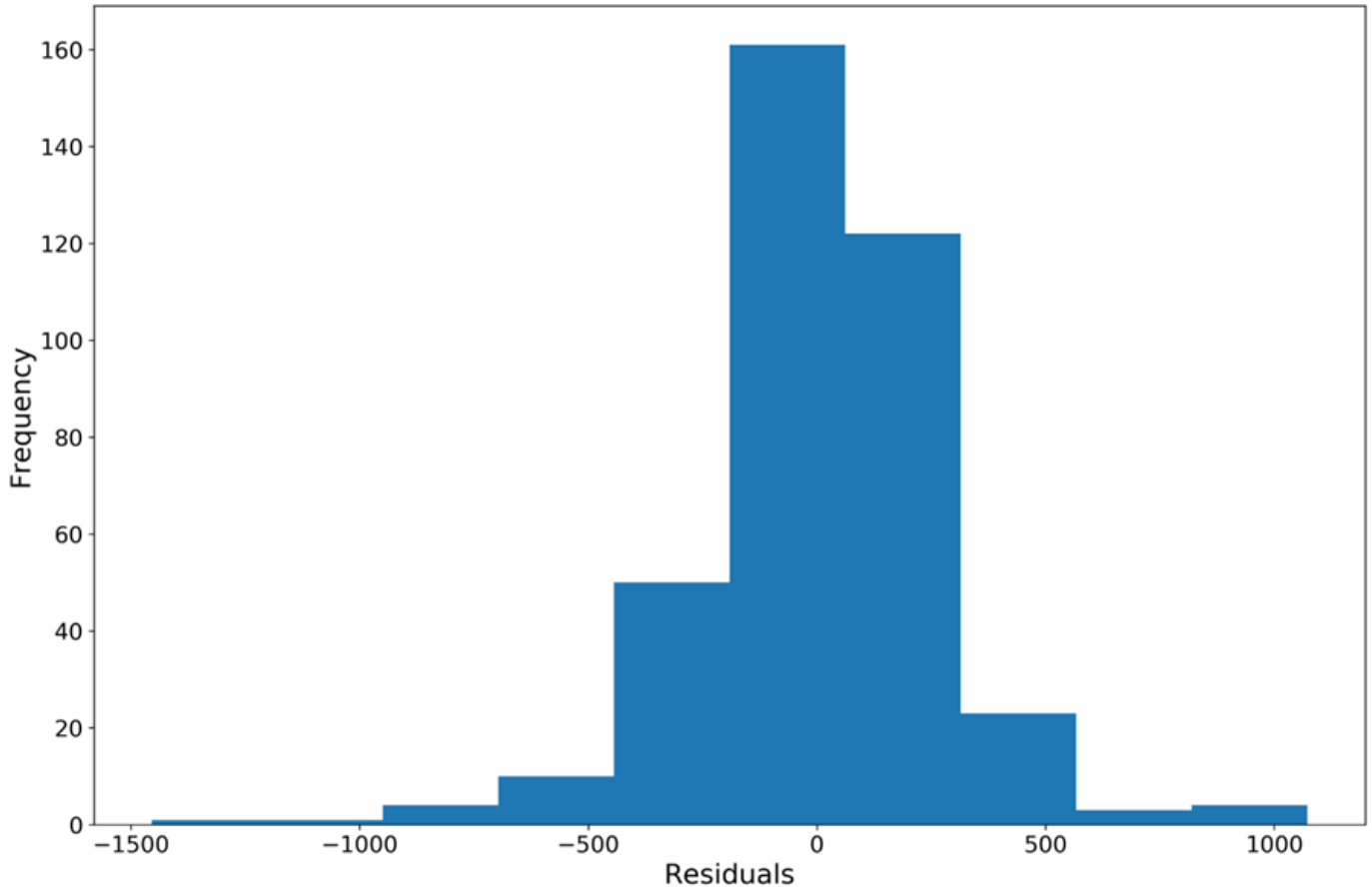


Figure 3. *Residuals distribution for the linear regression model. Except for a few outliers the residuals are as expected for this model normally distributed around 0.*

Coefficients of Lasso and linear regression

Features	Coefficients LassoReg	Coefficients LinReg
LivingArea	-0.03	-0.0
CommercialArea	-0.76	-0.3
ANCR	19.13	18.8
AMR	6.4	5.78
MaintenanceReserve	-38.33	-30.38
Capex	-21.16	-24.58
Market_sqm	0.29	0.29
Vacancy rate	-5.55	-11.23
2_2023	-62.18	-73.28
1_2023	-44.65	-49.39
4_2022	27.74	51.67
3_2022	74.77	88.1
2_2022	192.95	210.81
1_2022	205.77	227.56
4_2021	175.12	149.32
3_2021	174.37	172.91
2_2021	146.99	140.53
1_2021	135.14	124.58
4_2020	60.54	42.71
3_2020	0.89	23.48
2_2020	1.7	35.83
1_2020	-0.0	7.88
4_2019	-10.46	-56.41

Table 7. *Comparison of coefficients of Lasso and simple linear regression. The coefficients reflect well the expected "natural" dependencies between the features and target.*

Test on completely new data

City	LivingArea	CommercialArea	ANCR	y_true	y_pred
Frankfurt am Main	494.15	0.0	10.94	3056.0	3372.68
Köln	354.0	0.0	9.32	2260.0	2685.01
Frankfurt am Main	700.0	35.0	8.65	2871.0	2840.28
Duisburg	416.0	0.0	4.86	913.0	736.69
Dresden	252.0	0.0	8.26	1786.0	2105.53
Essen	288.9	0.0	9.73	1765.0	2019.47
Duisburg	393.0	0.0	8.4	1425.0	1541.57
Nürnberg	430.0	0.0	5.87	1700.0	1712.88
Dortmund	335.0	0.0	5.28	1045.0	1011.1
Hannover	572.0	0.0	7.01	1713.0	1732.06
Leipzig	842.2	0.0	7.38	1876.0	1889.11

Table 8. *Several examples from the new test dataset collected in the period between July and August 2023. The table provides basic information about the property's characteristics, location, as well as the information about predicted and real prices. Note that for the purpose of better visualisation and understandability, the ANCR is presented here in sqm and per month, rather than per year as was used during the training.*

Data	MDAPE	20%_Interval
Test Original	8.81	0.86
Test July-August	8.95	0.92

Table 9. *Comparison of the results for predictions for initial test data and for the new test data based on MDAPE and 5%-error-interval metrics.*

Conclusion & Outlook

Conclusion

- ML **can** be **successfully** applied for **commercial** valuations
- Linear models **outperformed** most of the ensemble models
- **Proper data preprocessing** and feature engineering lead to **significant improvement** of performance
- Only **few features** are the **most important** for the target
- The amount of availible **data** is still **very limited**

Outlook

- **Further** and more intensive **hyperparameter tuning**
- Grow the **database**
- Add **new features**: Location within the city, more accurate representation of house condition
- **Extend** the project on **other cities** (outside of Top-100)
- **Extend** the project for **all types** of commercial properties
- Integrate the results in the curret valuation tool

Thank you for attention!

Appendix

Outliers based on Factor

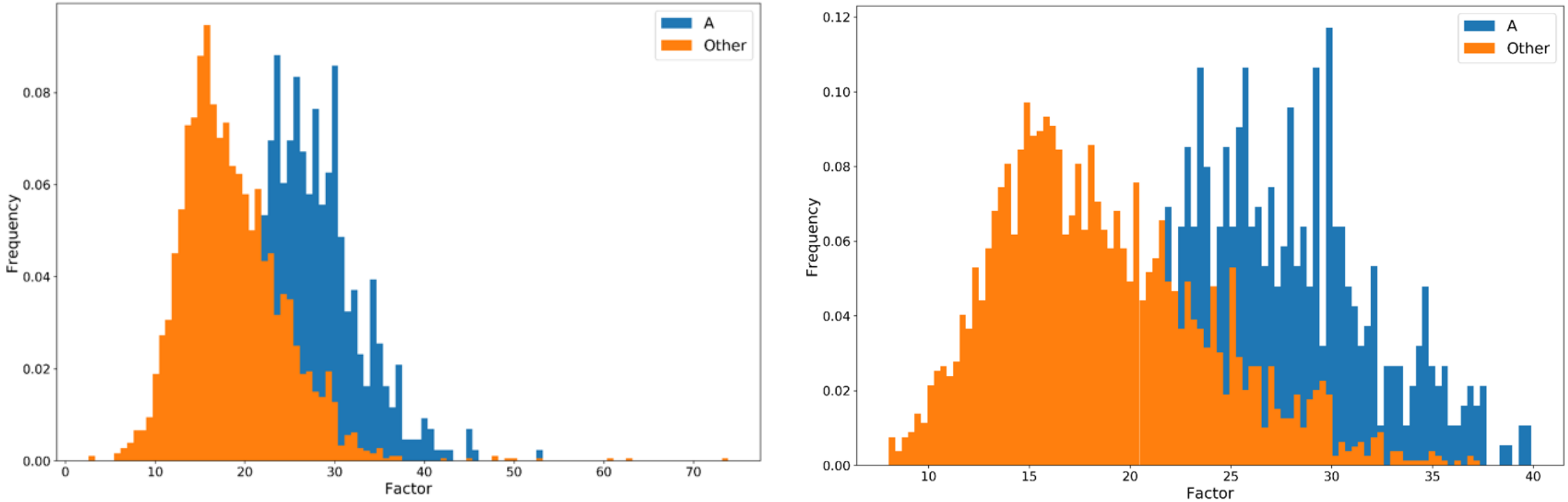


Figure 4. *Factor distribution of all objects in the dataset with outliers (above) and after filtering of outliers (below). The city class is colour-coded by orange for B, C and D class cities and blue for A class cities.*

Other metrics

Model	10%_Train	10%_Test	MAE_Train	MAE_Test	R2_Train	R2_Test
Baseline	0.33	0.35	357.76	373.46	0.72	0.72
LinReg	0.59	0.59	179.11	195.01	0.93	0.92
Lasso	0.6	0.55	179.55	204.91	0.93	0.92
Ridge	0.6	0.56	180.77	190.36	0.93	0.92
DecTree	0.49	0.46	229.3	250.92	0.9	0.87
Random Forest	0.53	0.44	208.11	254.18	0.91	0.87
AdaBoost	0.42	0.43	254.34	266.08	0.9	0.88
LightGBM	0.61	0.56	178.94	205.45	0.93	0.9

Table 10. *Some other metrics for the best pre-processed dataset.*

Random Forest algorithm

1) Generate randomly sampled subset of training data B.
2) Fit to each set B a decision tree, by using a random subset of size n’ of features for splits.
3) At each split store the empirical risk reduction for the corresponding split variable.
4) Return the final prediction as the average of the predictions of all the trees in the forest.

[19] G. Bontempi, (2021) Handbook Statistical foundations of machine learning, ULB, (2nd ed.).

AdaBoost algorithm

1) Initialize all object weights w_i with 1.
2) Pick N objects from the training set with the probability $p_i = w_i / \sum_{i=1}^{N} w_i$.
3) Construct a weak estimator for this training set (in this project Decision tree with depth of 3).
4) For every object in the training set make a prediction and calculate the individual loss function L_i.
5) Calculate an average loss $\bar{L}$ of all objects.
6) Calculate a measure of confidence of this estimator by the following formula: $\beta = \bar{L} / (1 - \bar{L})$.
7) Update weights of the objects by $w_i = w_i \cdot \beta^{(1-L_i)}$.
8) Repeat it for T estimators in sequence.
9) Return the final prediction of an ensemble as the weighted median prediction of all estimators.

[20] Drucker H. Improving regressors using boosting techniques. Proceedings of the 14th International Conference on Machine Learning; July 1997; Nashville, Tenn, USA.

List of references

- [1] R.-T. Mora-Garcia, M.-F. Cespedes-Lopez, and V. R. Perez-Sanchez, ‘Housing Price Prediction Using Machine Learning Algorithms in COVID-19 Times’, Land, vol. 11, no. 11, p. 2100, Nov. 2022, doi: 10.3390/land11112100.
- [2] W. Coleman, B. Johann, N. Pasternak, J. Vellayan, N. Foutz, and H. Shakeri, ‘Using Machine Learning to Evaluate Real Estate Prices Using Location Big Data’, 2022, doi: 10.48550/ARXIV.2205.01180.
- [3] Q. Truong, M. Nguyen, H. Dang, and B. Mei, ‘Housing Price Prediction via Improved Machine Learning Techniques’, Procedia Computer Science, vol. 174, pp. 433–442, 2020, doi: 10.1016/j.procs.2020.06.111.
- [4] A. B. Adetunji, O. N. Akande, F. A. Ajala, O. Oyewo, Y. F. Akande, and G. Oluwadara, ‘House Price Prediction using Random Forest Machine Learning Technique’, Procedia Computer Science, vol. 199, pp. 806–813, 2022, doi: 10.1016/j.procs.2022.01.100.
- [5] A. Nair, ‘Baseline Models: Your Guide For Model Building’. [Online]. Available: https://towardsdatascience.com/baseline-models-your-guide-for-model-building-1ec3aa244b8d
- [6] L. E. Melkumova and S. Ya. Shatskikh, ‘Comparing Ridge and LASSO estimators for data analysis’, Procedia Engineering, vol. 201, pp. 746–755, 2017, doi: 10.1016/j.proeng.2017.09.615.
- [7] G. James, D. Witten, T. Hastie, and R. Tibshirani, An Introduction to Statistical Learning, vol. 103. in Springer Texts in Statistics, vol. 103. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-7138-7.
- [8] Adopted from: ‘Algorithm Selection for Machine Learning’, 14.07.2022. [Online]. Available: https://elitedatascience.com/algorithm-selection
- [9] A. Amro, M. Al-Akhras, K. E. Hindi, M. Habib, and B. A. Shawar, ‘Instance Reduction for Avoiding Overfitting in Decision Trees’, Journal of Intelligent Systems, vol. 30, no. 1, pp. 438–459, Jan. 2021, doi: 10.1515/jisys-2020-0061.
- [10] C. Aldrich, ‘Process Variable Importance Analysis by Use of Random Forests in a Shapley Regression Framework’, Minerals, vol. 10, no. 5, p. 420, May 2020, doi: 10.3390/min10050420.

List of references

- [11] M. Aria, C. Cuccurullo, and A. Gnasso, ‘A comparison among interpretative proposals for Random Forests’, Machine Learning with Applications, vol. 6, p. 100094, Dec. 2021, doi: 10.1016/j.mlwa.2021.100094.
- [12] R. Madan, ‘Gradient boosting Vs AdaBoosting — Simplest explanation of how to do boosting using Visuals and Python Code’. [Online]. Available: https://medium.com/@madanflies/gradient-boosting-vs-adaboosting-simplest-explanation-of-how-to-do-boosting-using-visuals-and-d5939133b435
- [13] G. Shanmugasundar, M. Vanitha, R. Čep, V. Kumar, K. Kalita, and M. Ramachandran, ‘A Comparative Study of Linear, Random Forest and AdaBoost Regressions for Modeling Non-Traditional Machining’, Processes, vol. 9, no. 11, p. 2015, Nov. 2021, doi: 10.3390/pr9112015.
- [14] S. Saha, ‘XGBoost vs LightGBM: How Are They Different’. [Online]. Available: https://neptune.ai/blog/xgboost-vs-lightgbm
- [15] F. Wang, H. Cheng, H. Dai, and H. Han, ‘Freeway Short-Term Travel Time Prediction Based on LightGBM Algorithm’, IOP Conf. Ser.: Earth Environ. Sci., vol. 638, no. 1, p. 012029, Feb. 2021, doi: 10.1088/1755-1315/638/1/012029.
- [16] ‘Scikit-learn Machine Learning in Python’. [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html
- [17] M. Steurer, R. J. Hill, and N. Pfeifer, ‘Metrics for evaluating the performance of machine learning based automated valuation models’, Journal of Property Research, vol. 38, no. 2, pp. 99–129, Apr. 2021, doi: 10.1080/09599916.2020.1858937.
- [18] M. Saarela and S. Jauhiainen, ‘Comparison of feature importance measures as explanations for classification models’, SN Appl. Sci., vol. 3, no. 2, p. 272, Feb. 2021, doi: 10.1007/s42452-021-04148-9.
- [19] G. Bontempi, (2021) Handbook Statistical foundations of machine learning, ULB, (2nd ed.).
- [20] Drucker H. Improving regressors using boosting techniques. Proceedings of the 14th International Conference on Machine Learning; July 1997; Nashville, Tenn, USA.