T-504-ITML

Project 1

HÁSKÓLINN Í REYKJAVÍK

REYKJAVÍK UNIVERSITY

Haukur Húni Árnason
Óðinn Hjaltason Schiöth
Björn Breki Þorbjörnsson

Introduction

This paper's main goal is to generate a good model and a process for predicting final selling prices in icelandic króna on the Icelandic real estate market. Supervised learning models will be used for regression.

The data set comes from the Official Icelandic Register and is used in determining the official property evaluation for the year 2021. This dataset is very suitable for this experiment due to the fact that there are no missing values in this dataset, the dataset has little or no errors and alot of preprocessing has been done.

We decided to use this dataset to assist both real estate agents and potential sellers to evaluate the value of their real estate, this will hopefully give them an accurate estimate and save them time and or money.

Process

Exploratory analyses on dataset

When exploring a dataset for the problem at hand for determining Icelandic Real Estate prices. There were few datasets that were evaluated and analysed. The best quality dataset that was explored was the dataset used in this project. The data set is the dataset used for determining the official property value for the year 2021 and comes from the Icelandic official Registry.

The dataset is split into 4 categories:

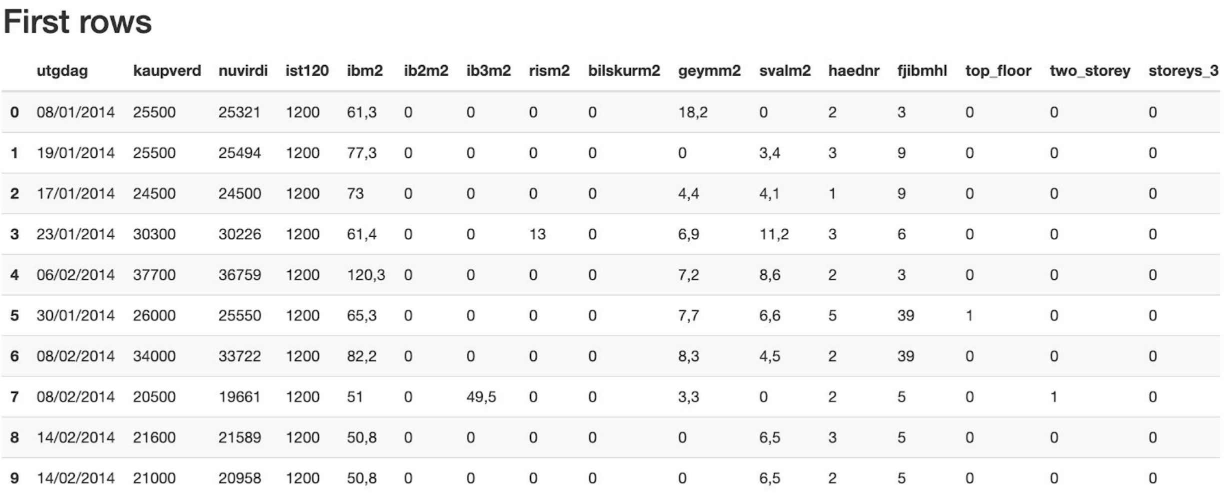
1. Detached houses within the Capital Region
2. Attached houses within the Capital Region
3. Detached houses outside the Capital Region
4. Attached houses outside Capital Region

For the sake of simplicity, the scope of the project and after exploration of the data, a decision was made to use 2 of the categories in the dataset to generate and train the model.

The categories chosen were attached houses within the capital region and attached houses outside of the capital region. The reason for this is that the two categories had the same features so no imputation needed to be done and were a bigger dataset with more instances.

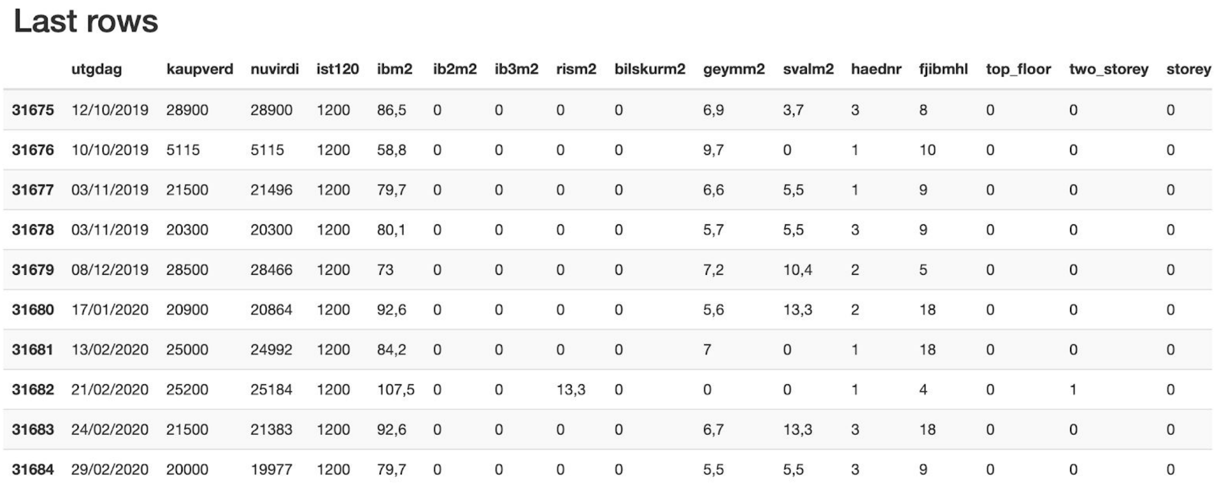
Exploratory analysis results

On the pictures below the first and last rows are examined for improving understanding of the dataset and its features. From the table it shows that the features have different ranges and that needs to be handled during preprocessing depending on the model that will be picked.
It is also clear that this model needs to be a regressor if the model is supposed to predict price.



First rows

	utgdag	kaupverd	nuvirdi	ist120	ibm2	ib2m2	ib3m2	rism2	bilskurm2	geymm2	svalm2	haednr	fjibmhl	top_floor	two_storey	storeys_3
0	08/01/2014	25500	25321	1200	61,3	0	0	0	0	18,2	0	2	3	0	0	0
1	19/01/2014	25500	25494	1200	77,3	0	0	0	0	0	3,4	3	9	0	0	0
2	17/01/2014	24500	24500	1200	73	0	0	0	0	4,4	4,1	1	9	0	0	0
3	23/01/2014	30300	30226	1200	61,4	0	0	13	0	6,9	11,2	3	6	0	0	0
4	06/02/2014	37700	36759	1200	120,3	0	0	0	0	7,2	8,6	2	3	0	0	0
5	30/01/2014	26000	25550	1200	65,3	0	0	0	0	7,7	6,6	5	39	1	0	0
6	08/02/2014	34000	33722	1200	82,2	0	0	0	0	8,3	4,5	2	39	0	0	0
7	08/02/2014	20500	19661	1200	51	0	49,5	0	0	3,3	0	2	5	0	1	0
8	14/02/2014	21600	21589	1200	50,8	0	0	0	0	0	6,5	3	5	0	0	0
9	14/02/2014	21000	20958	1200	50,8	0	0	0	0	0	6,5	2	5	0	0	0



Last rows

	utgdag	kaupverd	nuvirdi	ist120	ibm2	ib2m2	ib3m2	rism2	bilskurm2	geymm2	svalm2	haednr	fjibmhl	top_floor	two_storey	storey
31675	12/10/2019	28900	28900	1200	86,5	0	0	0	0	6,9	3,7	3	8	0	0	0
31676	10/10/2019	5115	5115	1200	58,8	0	0	0	0	9,7	0	1	10	0	0	0
31677	03/11/2019	21500	21496	1200	79,7	0	0	0	0	6,6	5,5	1	9	0	0	0
31678	03/11/2019	20300	20300	1200	80,1	0	0	0	0	5,7	5,5	3	9	0	0	0
31679	08/12/2019	28500	28466	1200	73	0	0	0	0	7,2	10,4	2	5	0	0	0
31680	17/01/2020	20900	20864	1200	92,6	0	0	0	0	5,6	13,3	2	18	0	0	0
31681	13/02/2020	25000	24992	1200	84,2	0	0	0	0	7	0	1	18	0	0	0
31682	21/02/2020	25200	25184	1200	107,5	0	0	13,3	0	0	0	1	4	0	1	0
31683	24/02/2020	21500	21383	1200	92,6	0	0	0	0	6,7	13,3	3	18	0	0	0
31684	29/02/2020	20000	19977	1200	79,7	0	0	0	0	5,5	5,5	3	9	0	0	0

The National registry provided documentation on the dataset. Here below are the feature descriptions on the dataset. Here it shows that “nuvirdi” or “kaupverd” are the values that needs to be predicted.

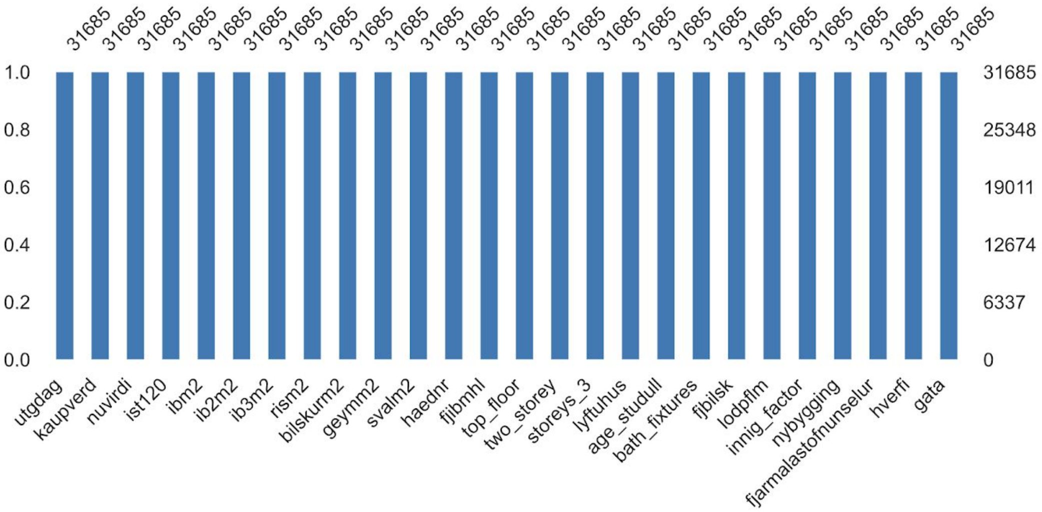
Attribute	Description
utgdag	útgáfudagur kaupsamnings (issue date of the purchase agreement)
kaupverd	Kaupverð (Purchase price)
nuvirdi	núvirði (staðgreiðsluvirði kaupsamnings) (Curent value)
ist120	flokkun eignar (sjá skýrslu um fasteignamat) ist120 (Class of asset)
ibm2	Íbúðarflatarmál (Apartment area)
ib2m2	íbúðarflatarmál í kjallara (Apartment area in basement)
ib3m2	íbúðarflatarmál í risi (Apartment area in the attic)
rism2	flatarmál í risi þar sem salarhæð er undir 1,8m (Apartment area in the attic with roof height less than 1,8m)
bilskurm2	Bílskúrsflatarmál (Garage area)
geymm2	Geymsluflatarmál (Storage area)
svalm2	Svalaflatarmál (Balcony area)
haednr	númer hæðar sem matshluti er staðsettur (Floor of the asset)
fjibmhl	fjöldi íbúða í húsi (matshluta) (Number of apartments in asset)
top_floor	er matshluti á efstu hæð (Is the asset on the top floor)
Two_Storey	tvær hæðir (Two stories)
Storeys_3	þrjár hæðir (Three stories)
lyftuhus	er lyfta í húsi eða ekki (is there an elevator or not)
age_studull	Aldursstuðull (age coefficient)
bath_fixtures	fjöldi baðkara, sturta, salerna deilt með íbúðarflatarmáli x100 (Number of bathtubs, showers, toilets divided by the apartment area x100)
fjbilsk	fjöldi bílastæða (Number of parking spots)
lodpflm	hlutdeild eignar í lóðarflatarmáli (Shared assets in property area)
innig_factor	leiðrétting vegna skoðunar inni (Corrections from inside inspection)
nybygging	er mastshluti nýbygging (Is the asset newly built)
fjarmalastofnunselur	fjármálastofnun er seljandi (Is fjármálastofnun the seller)
hverfi	matssvæði (sjá á http://www.skra.is) (Neighborhood)
gata	undirmatssvæði (sjá á http://www.skra.is) (Street)

Datatypes of dataset:

It is important to explore the datatypes of the dataset for both determining what preprocessing needs to be done and for picking the correct model. The table below shows that most of the features are numerical values. The only non numerical feature is the “utgdag” which is a string object that represents a date. When preprocessing it is clear that some feature engineering needs to be done on the “utgdag” feature.

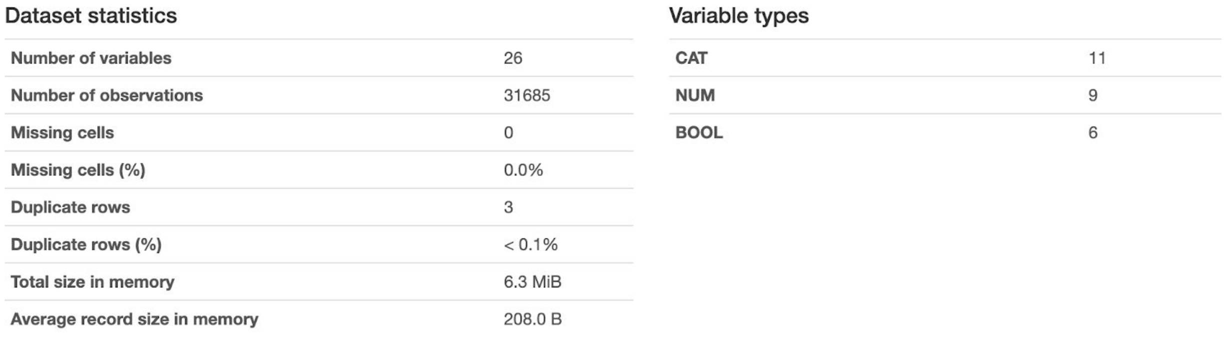
Attribute	DataType	Attribute	DataType
utgdag	object	top_floor	int64
kaupverd	int64	two_storey	int64
nuvirdi	int64	storeys_3	int64
ist120	int64	lyftuhus	int64
ibm2	float64	age_studull	int64
ib2m2	float64	bath_fixtures	int64
ib3m2	float64	fjbilsk	int64
rism2	float64	lodpflm	float64
bilskurm2	float64	innig_factor	float64
geymm2	float64	nybygging	int64
svalm2	float64	fjarmalastofnunselur	int64
haednr	int64	hverfi	int64
fjibmhl	int64	gata	int64

Below is a plot on the feature value count in the dataset. This bar plot shows that there are no values in this dataset that are missing, so no imputation needs to be done.



Statistics for the dataset show that we have 26 variables and the set has 31685 instances. It also shows how many categorical, numerical and boolean variables there are in the dataset.

Statistics for this dataset:



Dataset statistics

Number of variables	26
Number of observations	31685
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	3
Duplicate rows (%)	< 0.1%
Total size in memory	6.3 MiB
Average record size in memory	208.0 B

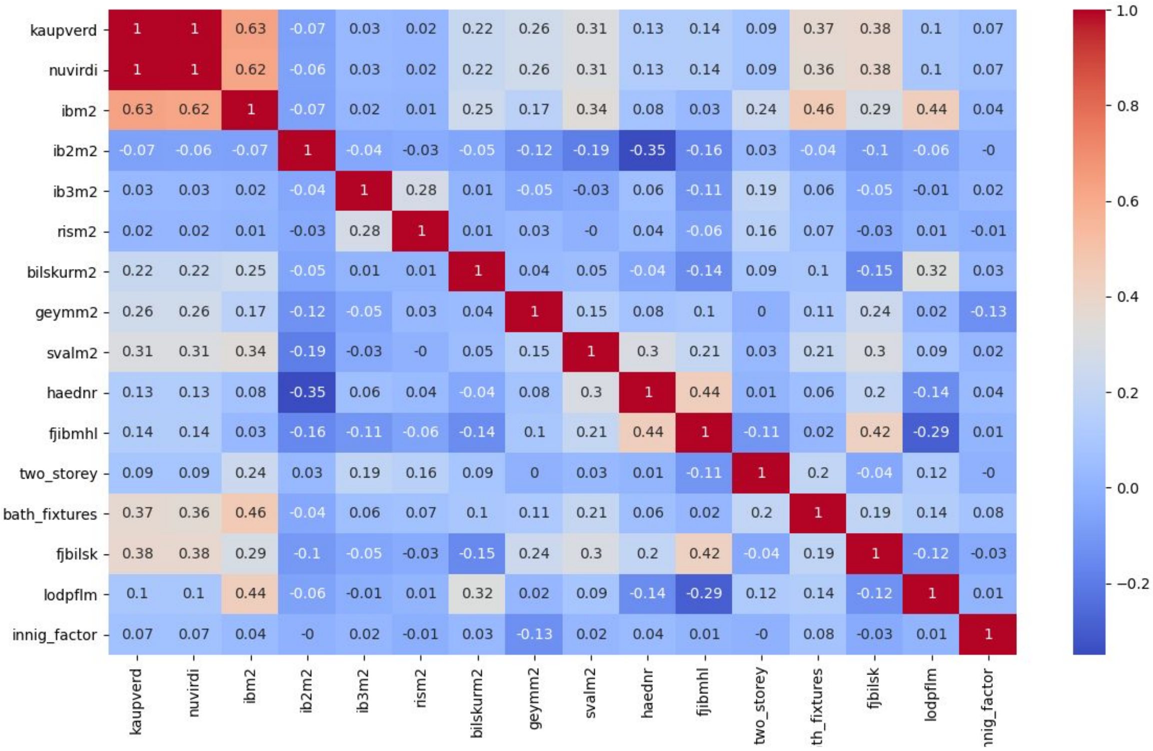
Variable types

CAT	11
NUM	9
BOOL	6

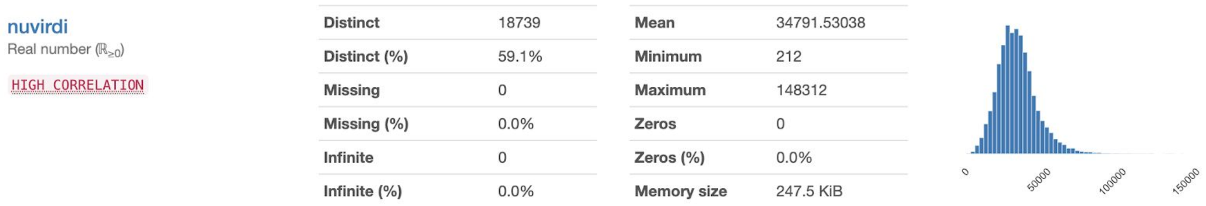
Correlation of variables is determined by correlation heatmap below. The correlation matrix shows that the attributes “kaupverd” and “nuvirdi” have total positive correlation and for the sake of training a model it is important to only choose one of the attributes to predict and the other can not be in the training and testing dataset because “nuvirdi” is an extracted attribute calculated from “kaupverd”.

The variable “ibm2” has the highest correlation with the “kaupverd” and “nuvirdi” this is expected and shows that the apartment area has a big effect on the price.
Other variables that have high correlation with the two variables are “bath_fixture” and “fj_bilsk” and the “bilskur_m2”, “geym_m2” and “sval_m2”, which does make sense, since they’re desirable features to have.

When evaluating the data it is clear that a lot of preprocessing has already been done on the dataset. There is a lot of feature engineering that has been done to weigh qualities that cause a price of real estate to be higher with ratios. For other attributes a coefficient is created from the attribute.

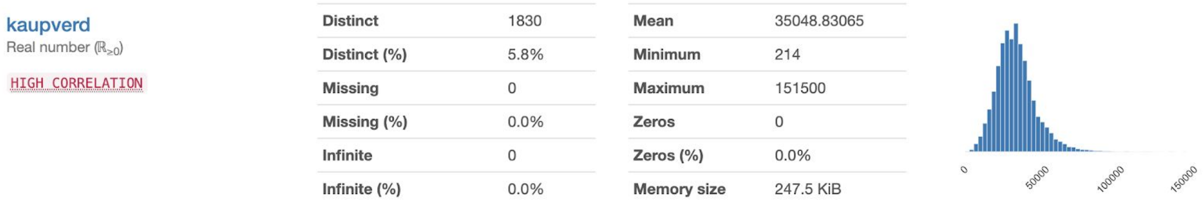


The variable to predict is either “nuvirdi” or “kaupverd” below are some statistics on the two variables. There is also a possibility for making a prediction on an extracted feature like the square meter price of an apartment.



nuvirdi
Real number ($\mathbb{R}_{\geq 0}$)
HIGH CORRELATION

Distinct	18739	**Mean**	34791.53038
Distinct (%)	59.1%	**Minimum**	212
Missing	0	**Maximum**	148312
Missing (%)	0.0%	**Zeros**	0
Infinite	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%	**Memory size**	247.5 KiB



kaupverd
Real number ($\mathbb{R}_{\geq 0}$)
HIGH CORRELATION

Distinct	1830	**Mean**	35048.83065
Distinct (%)	5.8%	**Minimum**	214
Missing	0	**Maximum**	151500
Missing (%)	0.0%	**Zeros**	0
Infinite	0	**Zeros (%)**	0.0%
Infinite (%)	0.0%	**Memory size**	247.5 KiB

Exploring models that may be suitable for the problem

The first thing to consider when choosing a model is that for this problem the model chosen needs to be a regressor and use supervised learning. Due to the fact that the prediction is numerical it has infinitely many possibilities therefore we need a regressor for predicting.

Few regression models were chosen based on their characteristics. Their pros and cons were evaluated.

Random forest regressor
Decision tree regressor
K neighbors regressor
Linear regressor : large number of features and low training data.
Support Vector regressor

Random Forest Regressor:

Pros:

- Because a random forest regressor is a collection of decision trees that picks the average/majority votes it should give a more accurate and generalized result then decision tree.
- Doesn’t really care about outliers

Con:

- Can easily overfit the model

Linear Regression

Pros:

- Easy to implement
- Fast training

Cons:

- Assumes the input residuals to be normally distributed
- Sensitive to outliers

SVR

Pros:

- Handles both linear and non-linear problems
- Handles outliers better than LR

Con:

- Is outperformed by kNN and Random forest with our dataset

Decision Tree Regressor

Pros:

- Not much preprocessing needed
- Can provide an understandable explanation for the prediction

Cons:

- High chance to overfit and could grow to be very complex
- Loses information when handling continuous variables

K Neighbors Regressor

Pro:

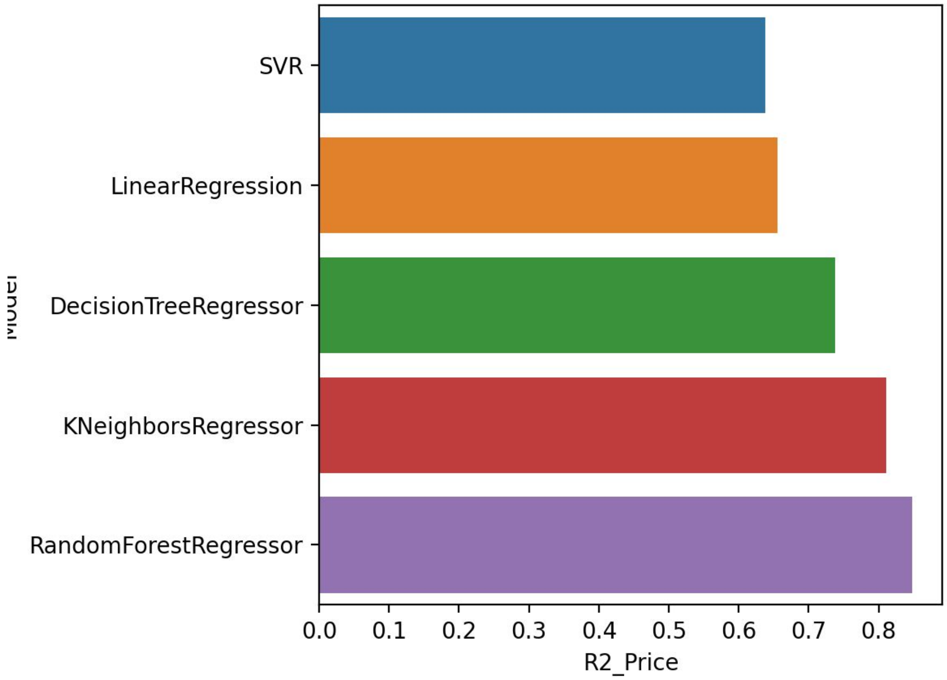
- Simple with few hyperparameters

Con:

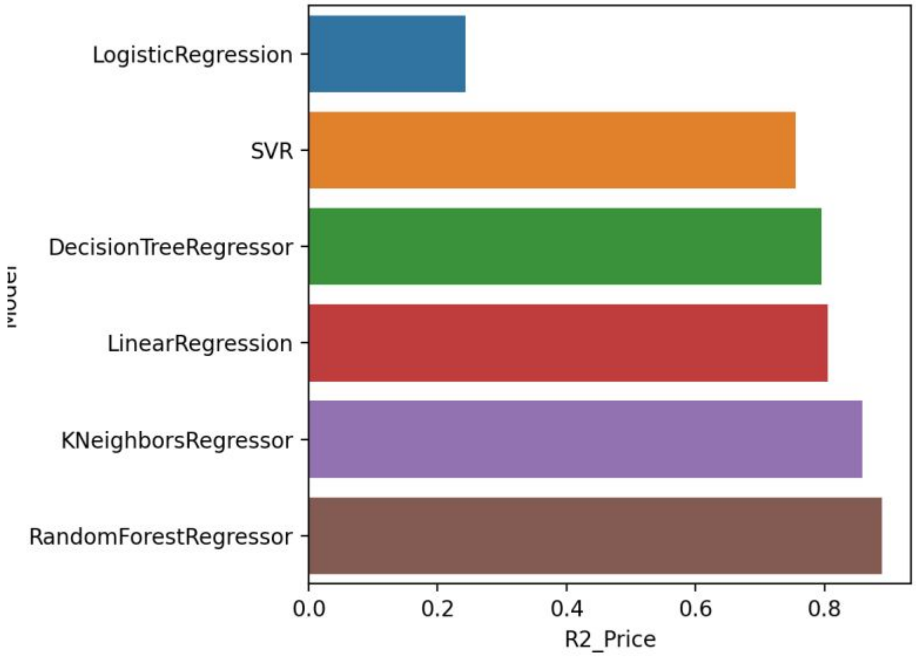
- Not that great with outliers

When we ran a test on the models without tuning the hyperparameters there were 2 standout models, we ran the test on both the current value and the value per area. These models were random forest regressor and K Neighbors regressor. The pros and cons were carefully evaluated along with the results of our testing.

Prediction on fermetraverð (Price by area):



Prediction on núvirði (Current Price):



Preprocessing the dataset

From evaluation on the dataset There is a lot of feature engineering and preprocessing that has already been done to weigh qualities that cause a price of real estate to be higher with ratios. For other attributes a coefficient is created from the attribute.

The two models that were chosen to have different requirements when it comes to preprocessing the dataset.

Most regression models require outliers to be removed before training a model. Random Forest Regressor is pretty immune to outliers.
These are the scaling method tried on the dataset:
[StandardScaler(), RobustScaler(), MinMaxScaler()],

The dataset was split in training and testing 0.75 was training and 0.25 testing. Because our dataset isn’t very large so we decided to have our test size little larger than usual to compensate for that.

Testing regressors
The regressors we chose were Random Forest Regressor and kNearest Regressor, we tested both of them extensively. Tuning and training took significantly longer with the RFR but it turned out to be better suited for this dataset.

Analysing hyperparameters

Random forest regressor hyperparameters

When picking hyperparameters to tune we tried to choose mainly parameters that could have a large impact or help us generalize the outcome.
The random forest regressor has quite a few hyperparameters that needed to be calculated.

The parameters used for tuning:
N_estimators: number of trees generated.
Max_features: number of features to consider in every split
Max_depth: maximum number of levels in tree
Min_sample_split: minimum number of samples required to split a node
Min_sample_leaf: minimum number of samples required at each leaf node

The selected hyperparameters and the values used for tuning:
n_estimators: [10, 34, 73, 127, 196, 280, 378, 491, 619, 762, 919, 1091, 1278, 1480, 1696, 1927, 2173, 2434, 2709, 3000]
max_features: ['auto', 'sqrt', 'log2']
max_depth: [10, 16, 24, 34, 46, 59, 74, 90, 108, 128, 150, None]
min_sample_split: [2, 3, 4, 5, 7, 9, 11, 14, 17, 20]
min_samples_leaf: [1, 2, 3, 4, 5, 7, 9, 11, 14, 17, 20]

All the hyperparameters were chosen with a mix of trial and error on the data and theory.

K nearest neighbors regression hyperparameters

For K nearest neighbors regression there were mainly two hyper parameters that needed to be tuned. The parameters were the “k” in k-nearest neighbor and what formula is used in the calculation of distance between data points.
The selected hyperparameters and the values used for tuning:

k = [1, 2, 3, 4, 5, 6, 7, 8, 10, 15, 25, 40, 60, 100]
metric = ["minkowski",
"manhattan",
"Euclidean"]

Generating the pipeline and tuning hyperparameters

For the Random forest regressor randomized search with cross validation was used for hyper parameter tuning because of the fact that grid search with the selected parameters was too computationally expensive. Random search gave good enough results when tuning the parameters. We ran a random search with 400 iteration and 10 fold validation and took in total 6 hours to run.

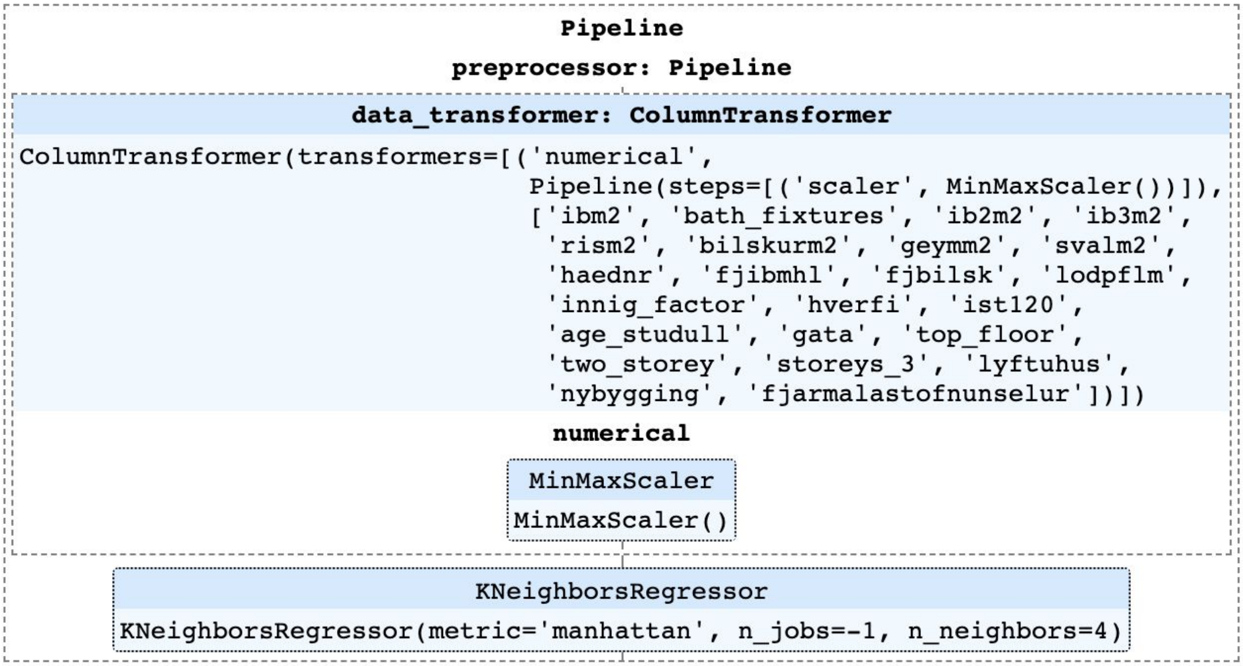
For the k-nearest neighbour regression grid search with cross validation was used for hyperparameter tuning. This resulted in 126 candidates and totalling in 1260 fits. The computational time was not long and therefore possible to try a lot of values for parameters.

10-fold cross validation was used to not overfit the training models.

K-nearest model was tested with 2019 dataset in the pipeline but did not perform well enough

```
Model Performance for KNR
TEST SET
==============
R2 score =             0.8014486878290485
Accuracy =              88.83%.
Mean absolute error = 4363.535011355034
RMSE =                 6389.155528175782
Average Error =        4363.535011355034

TRAIN SET
==============
R2 score =             0.8787354030513259
Accuracy =              91.61%.
Mean absolute error = 3223.714925561443
RMSE=                  4789.782779971526
Average Error =        3223.714925561443
```



```
Pipeline
preprocessor: Pipeline
data_transformer: ColumnTransformer
ColumnTransformer(transformers=[('numerical',
                                 Pipeline(steps=[('scaler', MinMaxScaler())]),
                                 ['ibm2', 'bath_fixtures', 'ib2m2', 'ib3m2',
                                  'rism2', 'bilskurm2', 'geymm2', 'svalm2',
                                  'haednr', 'fjibmhl', 'fjbilsk', 'lodpflm',
                                  'innig_factor', 'hverfi', 'ist120',
                                  'age_studull', 'gata', 'top_floor',
                                  'two_storey', 'storeys_3', 'lyftuhus',
                                  'nybygging', 'fjarmalastofnunselur'])])
numerical
MinMaxScaler
MinMaxScaler()
KNeighborsRegressor
KNeighborsRegressor(metric='manhattan', n_jobs=-1, n_neighbors=4)
```

Best params {'preprocessor__data_transformer__numerical__scaler': MinMaxScaler(), 'regressor__metric': 'manhattan', 'regressor__n_jobs': -1, 'regressor__n_neighbors': 4}

Scoring and error analyses, performance metric

Multiple performance metrics were introduced to the training. The best result gave the Coefficient of the Determination and was used in both the grid search and the randomized search for scoring.

All of the Performance and error metrics that come here later were used in the training and model analyses but what was discovered was that the R2 performance metric was best suited when tuning parameters.

1. Mean Squared Error(MSE):

2. Root-Mean-Squared-Error(RMSE):

3. Mean-Absolute-Error(MAE):

4. R^2 or Coefficient of Determination:

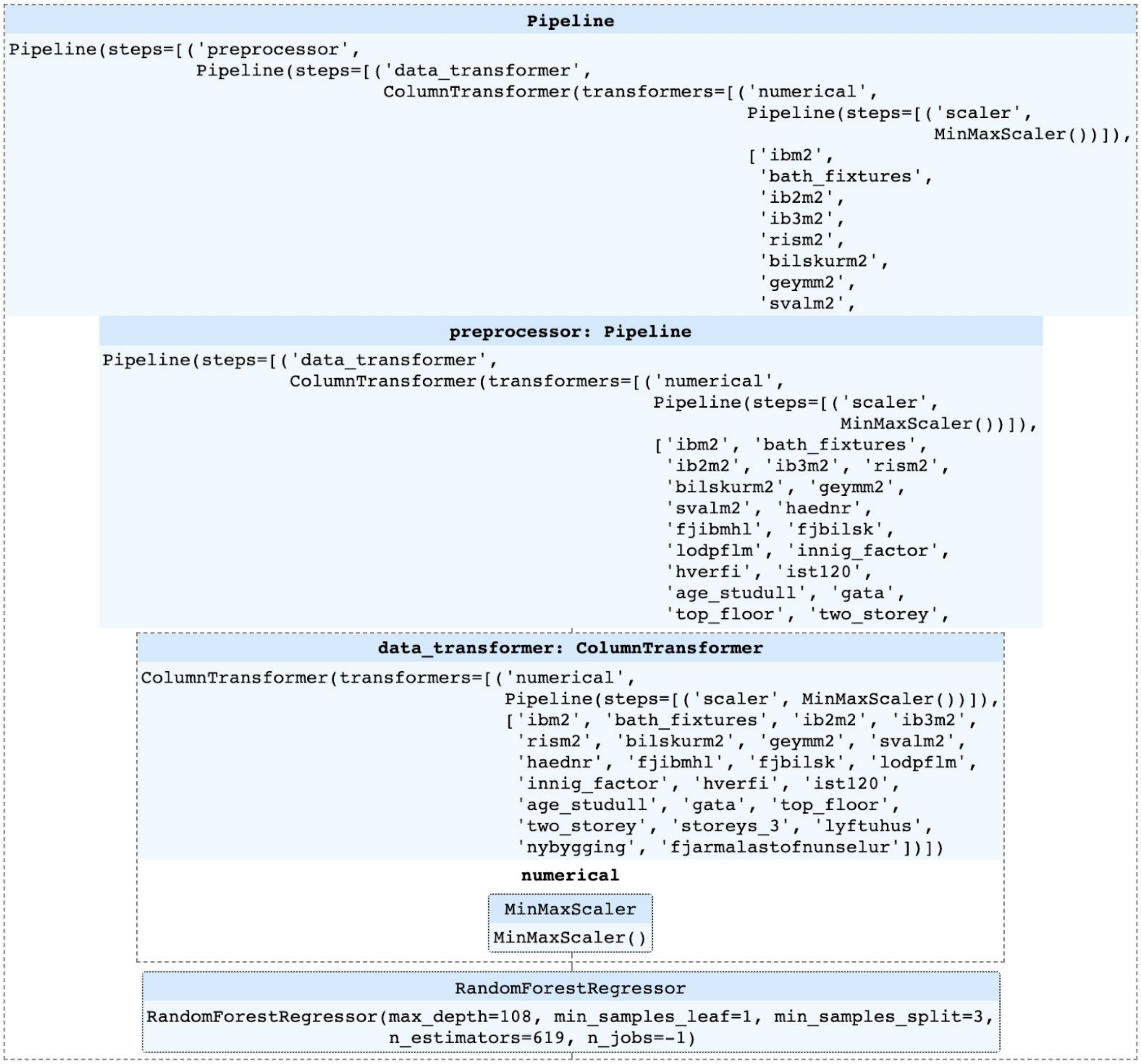
Final model and results

Results from running model on dataset filtered on the year 2019

```
Best params {'regressor__n_jobs': -1, 'regressor__n_estimators': 619,
'regressor__min_samples_split': 3, 'regressor__min_samples_leaf': 1,
'regressor__max_features': 'auto', 'regressor__max_depth': 108,
'preprocessor__data_transformer__numerical__scaler': MinMaxScaler()}

TEST SET
==============
R2 score =              0.9023732599732811
Accuracy =              91.80%.
Mean absolute error = 3065.185300520732
RMSE =                  4480.138840224952
Average Error =         3065.185300520732

TRAINING SET
==============
R2 score =              0.9766566115734981
Accuracy =              96.24%.
Mean absolute error = 1383.67292291246
RMSE =                  2101.5065734064174
Average Error =         1383.67292291246
```



Pipeline

```
Pipeline(steps=[('preprocessor',
                 Pipeline(steps=[('data_transformer',
                                  ColumnTransformer(transformers=[('numerical',
                                                                   Pipeline(steps=[('scaler',
                                                                                    MinMaxScaler())]),
                                                                   ['ibm2',
                                                                    'bath_fixtures',
                                                                    'ib2m2',
                                                                    'ib3m2',
                                                                    'rism2',
                                                                    'bilskurm2',
                                                                    'geymm2',
                                                                    'svalm2',
```

preprocessor: Pipeline

```
Pipeline(steps=[('data_transformer',
                 ColumnTransformer(transformers=[('numerical',
                                                  Pipeline(steps=[('scaler',
                                                                   MinMaxScaler())]),
                                                  ['ibm2', 'bath_fixtures',
                                                   'ib2m2', 'ib3m2', 'rism2',
                                                   'bilskurm2', 'geymm2',
                                                   'svalm2', 'haednr',
                                                   'fjibmhl', 'fjbilsk',
                                                   'lodpflm', 'innig_factor',
                                                   'hverfi', 'ist120',
                                                   'age_studull', 'gata',
                                                   'top_floor', 'two_storey',
```

data_transformer: ColumnTransformer

```
ColumnTransformer(transformers=[('numerical',
                                 Pipeline(steps=[('scaler', MinMaxScaler())]),
                                 ['ibm2', 'bath_fixtures', 'ib2m2', 'ib3m2',
                                  'rism2', 'bilskurm2', 'geymm2', 'svalm2',
                                  'haednr', 'fjibmhl', 'fjbilsk', 'lodpflm',
                                  'innig_factor', 'hverfi', 'ist120',
                                  'age_studull', 'gata', 'top_floor',
                                  'two_storey', 'storeys_3', 'lyftuhus',
                                  'nybygging', 'fjarmalastofnunselur'])])
```

numerical

MinMaxScaler

```
MinMaxScaler()
```

RandomForestRegressor

```
RandomForestRegressor(max_depth=108, min_samples_leaf=1, min_samples_split=3,
                      n_estimators=619, n_jobs=-1)
```

Results from running on all data in dataset

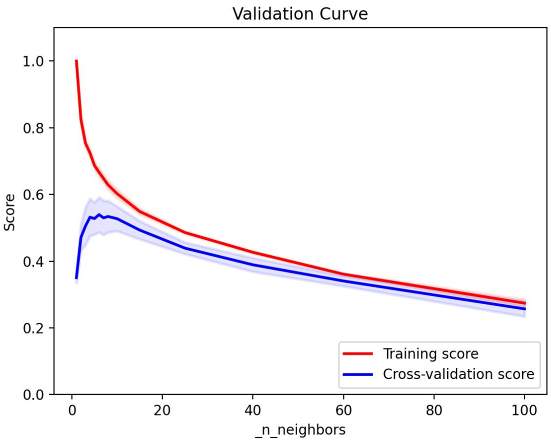
TEST SET
==============
R2 score = 0.9213233416455373
Accuracy = 90.43%.
Mean absolute error = 2547.475443988679
RMSE=   3815.0088484428984
Average Error =   2547.475443988679

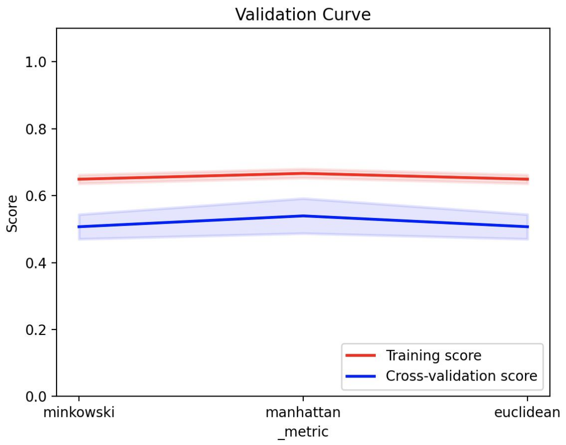
TRAIN SET
==============
R2 score = 0.9882711852855559
Accuracy =   96.86%.
Mean absolute error = 971.4052858324939
RMSE =   1453.2998419737614
Average Error =   971.4052858324939

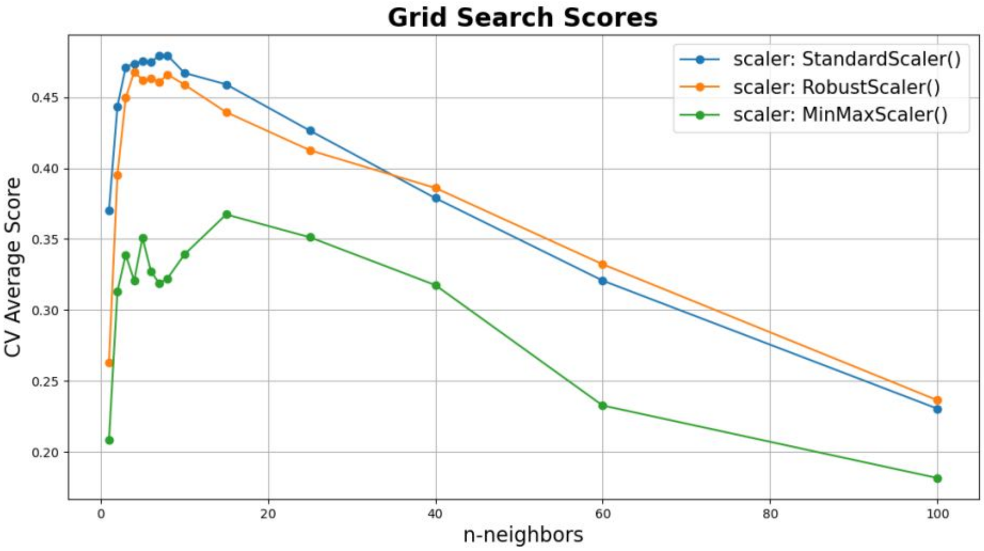
Features:

Best params {'n_estimators': 264, 'min_samples_split': 3, 'min_samples_leaf': 1, 'max_features': 'auto', 'max_depth': 41}

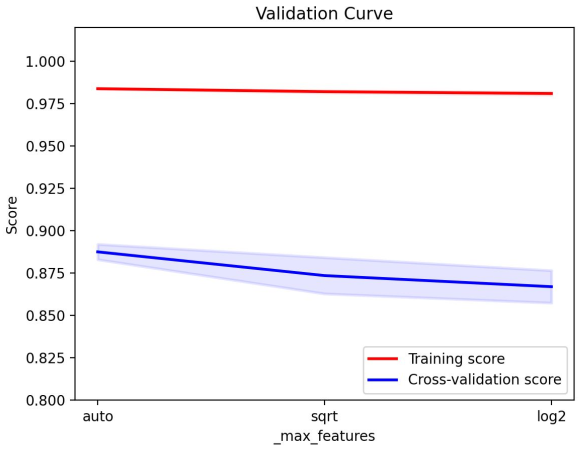
K neighbor parameters:

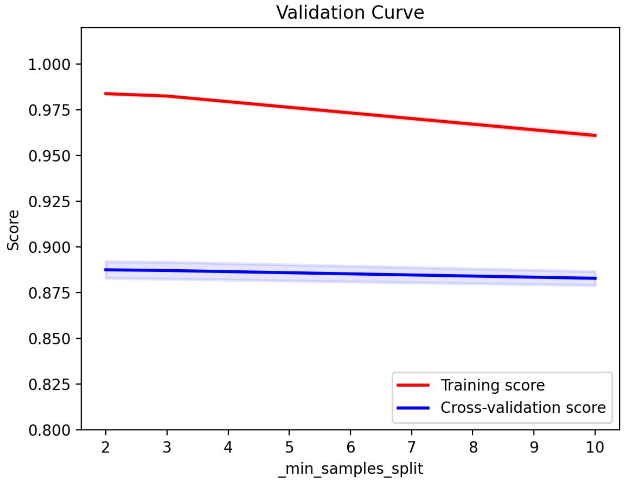


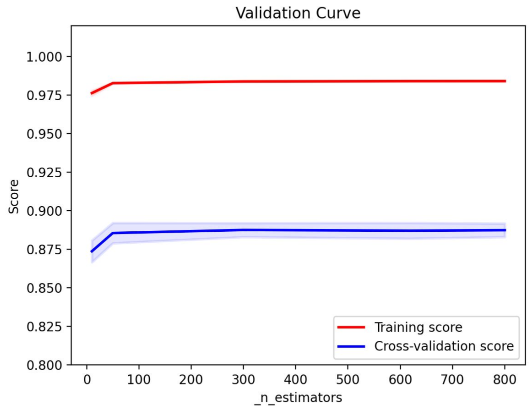


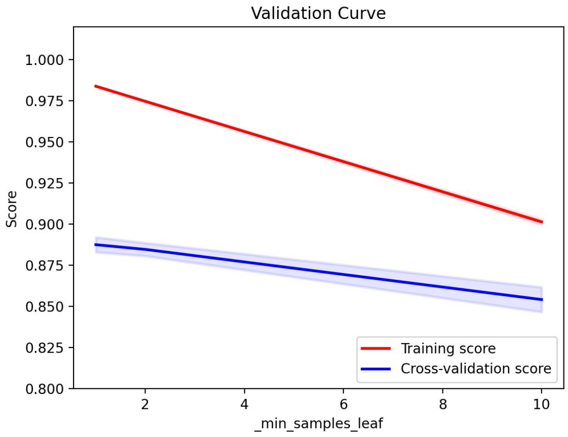


Random Forest parameters:









Conclusions

After doing this experiment it is clear that the best model for predicting real estate prices current value using this dataset is the RandomForestRegressor model. Using 10-fold cross validation there is not a lot of overfitting on the dataset and the model generalises well.

In the experiment the pipeline and training model were only used on a dataset filtered on the year 2019 which provided an R^2 score of 0.90, accuracy 91.80% and Mean absolute error of 3065185 ISK on the test set.
When the pipeline and training model is run on the whole dataset it provides the R^2 score of 0.92, accuracy 90.43% and Mean absolute error of 2547475 ISK on the test set.

It was a challenge that the dataset is time sensitive but when tested on the whole data it proved that the training model generalised well.

The best parameters from the results for data from 2019:

max depth of the tree is 108, if too deep it could overfit that is why cross validation is used.
min samples leaf is 1, if cross validation would not be used this would overfit to training data.
min samples split is 3
max_features: auto
n estimator 619

The best parameters from the results for all data:
max depth of tree is 41
min samples leaf is 1,
min samples split is 3
max_features: auto
n estimator 264

The reason why these values were chosen where to generalize and reduce overfitting. This is shown in the validation curves for the parameters in the Results chapter, the validation curve supports the parameters chosen.

Future Work

Our future plans for this project is to research more parameters and values, perform a grid search on further values to fine tune the project even more, other preprocessing methods might also be viable.
After that we might also look into different models, other more complex models that are beyond the scope of this project might suit the dataset better, hopefully that will help us yield a better more consistent result.

Future plans also include expanding the dataset into other types of property or assets and to expand it to other countries, that would require a major rework of the whole project and could be a huge undertaking.

More future plans include tuning the k-fold more, we only tested it a very limited amount and it could yield better results, also looking at our results further, analyze them and look at how everything generalizes and possibly make other decisions in our process. If we do another similar project in the future, we will definitely plan more ahead and not go straight to the solution, that way we might understand everything better earlier on, make less mistakes and come to a better solution.