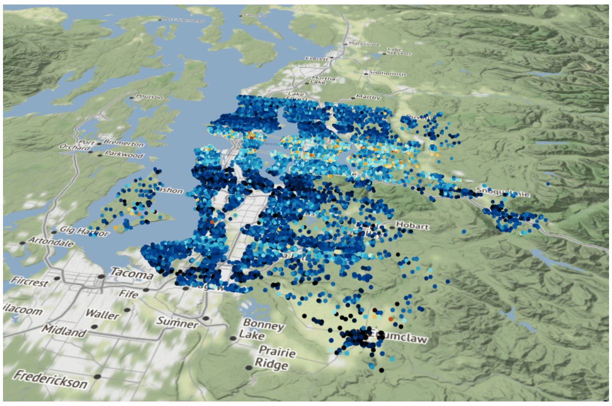
UNIVERSITY OF POTSDAM
HASSO-PLATTNER-INSTITUTE

Predicting HousePrices - a ML-Approach

Student name: *Lukas Laskowski*

Course: *Introduction into Data Science* – Professor: *Prof. Dr. de Melo*
Due date: *February 05th, 2020*



Disclaimer: The code for each graph shown in this homework can be seen in the metrics-folder.

Motivation. HousePrices can massively vary depending on several factors, for example: location, size of the house or social environment. In my homework I will predict HousePrices based on information over the specific house and its surroundings.

1. Data Preparation

1.1. Data Collection. I collected two datasets from different sources to develop the Machine-Learning Model:

- `Kaggle.com`: I collected the base dataset from a datascience-website. It includes information about properties and prices of 21614 house in KingCounty, Washington. (see kc_house_data.csv in my submission)

- `unitedstateszipcodes.org`: My intention is to extend this base dataset with information about the location of each house. For that I use a dataset which provides information about many areas, identified by the ZIP-code. As there exists no real dataset, I had to crawl it from a website with the python-module *Beautiful Soup*: First of all, I collected all distinct ZIP-codes from the dataset1. Now I requested the html-file from *unitedstateszipcodes.org*, parsed the file and added columns like the median household income or the age distribution for each ZIP-code to my main dataset.

1.2. Data Joining. I now iterate over the rows of the house-dataset and add for each house the data about the social environment with the correct ZIP-code (see out.csv in my submission).

2. Data Splitting

Dataset	#records	#unique ZIP-codes	$\emptyset price$	$\emptyset Median - Household - Income$
x-train	16209	70 (100%)	-	$80969
x-test	5404	70 (100%)	-	$81520
y-train	16209	-	$537425	-
y-test	5404	-	$548075	-
total	21613	70	$540088	$81107

Table 1: Metrics to measure the differences between test & train datasets

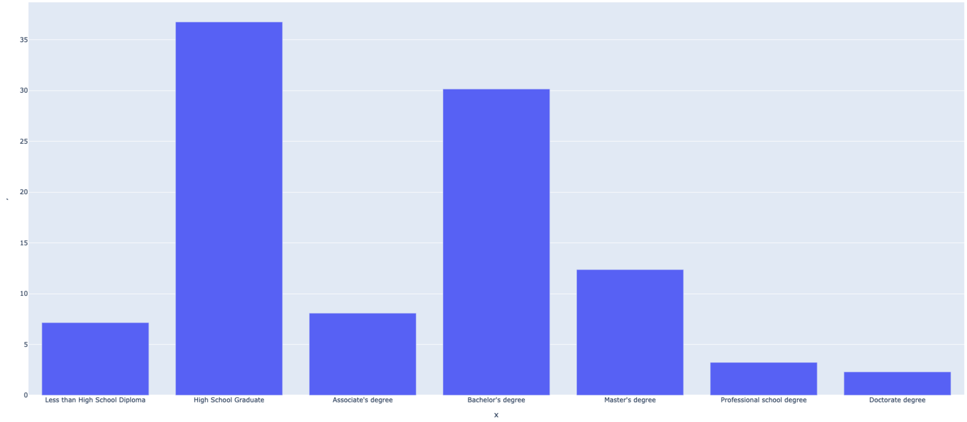


Figure 1: Average educational qualification: x-train

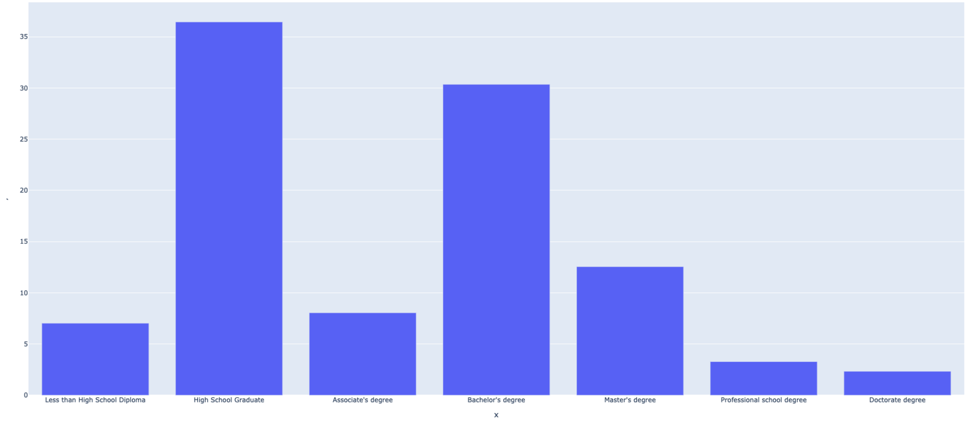


Figure 2: Average educational qualification: x-test

Results. I splitted the data randomly into training and test data (see code below). The test-size is 25% and training-size 75% The table above shows some of the differences between train & test dataset at some key metrics. It gets clear that each dataset slightly differs but there are no huge differences. The bar chart above visualises the average of the educational qualification for the training dataset and for the test dataset. In this case the exact numbers are not very important but the differences between the test and training dataset.

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(X, y, random_state=1)
```

3. Feature Engineering

In this section I will discuss the data, select the attributes as features with the highest influence on the price and develop new features which might help to improve the accuracy of the machine learning model.

3.1. Preprocessing. *df.info()* shows an overview over the data, to get an insight in the different datatypes and the existence of non-null values. My dataset does not contain any null-values, so there is no need to handle them.

3.2. Data exploration. First of all I wanted to get an overall understanding of the data. How do some features correlate to the price? Which are not useful?

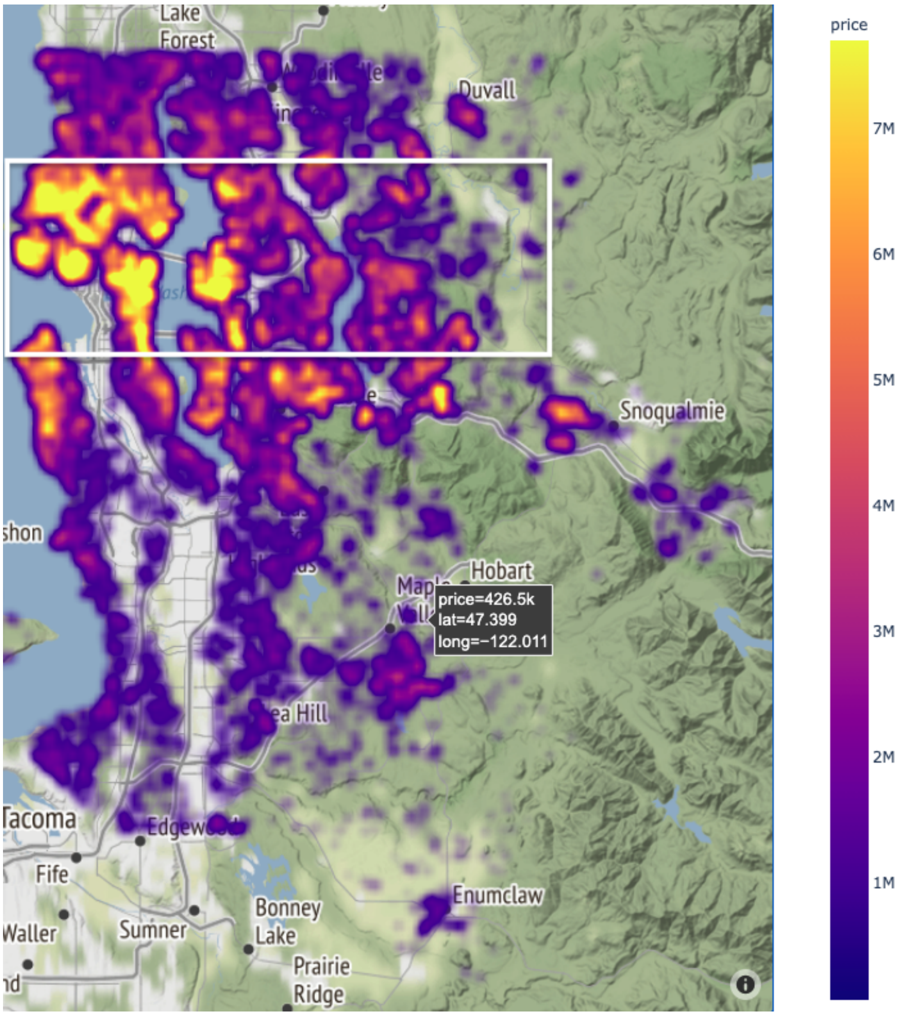


Figure 3: Houseprices depending on their location

HousePrices correlating to the position of the houses. To get an initial feeling, I mapped the data on a map of Seattle, KingCounty (see Figure 3). It shows the houseprices in dependancy to their location. The yellow color symbolises high prices, the dark purple color symbolises low prices. You can see a "belt" (white box) at the top of the image. It seems like a high latitude might correlate to high prices. To ensure this hypothesis, I pairplot the price with the latitude:

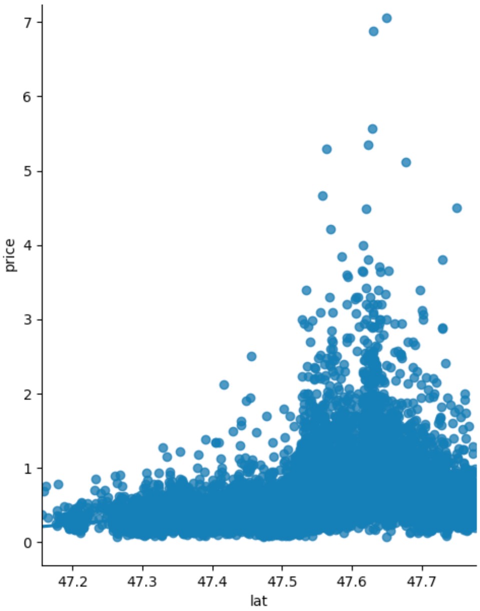


Figure 4: Correlation between price and latitude.

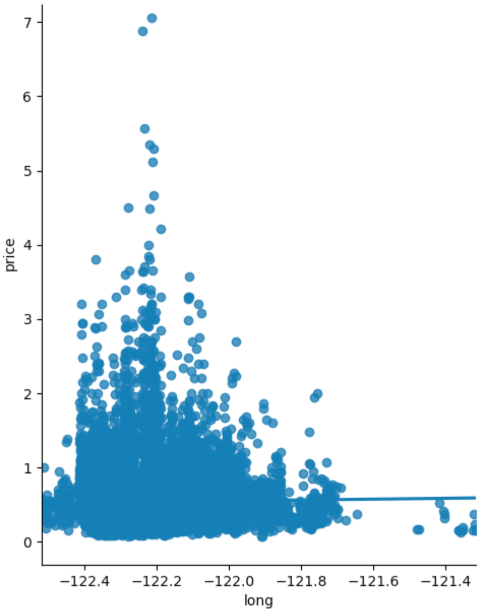


Figure 5: Correlation between price and longitude.

It gets clear that the latitude slightly influences the price. If the specific house is more in the north, it is more likely to be more expensive. The longitude shows an equivalent behaviour: The more western, the higher is the price of a house. These insights correlate to the map (Figure 3): The north-west of KingCounty is the location with the highest prices. This might be a good feature to fit the ML-model. I will come back to it later on.

Further insights. I further plotted nearly every attribute in correlation to the target-attribute: *price*. This should give a better feeling for the feature engineering process. Figures 6-11 show the most interesting results of this analysis. The information of Figures 6-8 come from the crawled dataset containing information about each ZIP-code in the US. These plots show that there are differences in the houseprices for each area/ZIP-code. For example one can see that regions with a higher median income, have higher house prices. Furthermore Figures 7, 8 show that the educational qualification influence the house price market. If there is a higher educational qualification in average, the house prices are higher, too. This might be an interesting feature to improve the accuracy of the model. Figures 9, 10 and 11 include attributes from the main dataset. The correlation between the price and the grading of a house is shown in Figure 9. Figure 10 shows the strong correlation between the size of the living space and the price of a house. Figure 11 shows that not only regional attributes of houses but also house-specific attributes like environmental properties (e.g. waterfront) influence the price of a house.

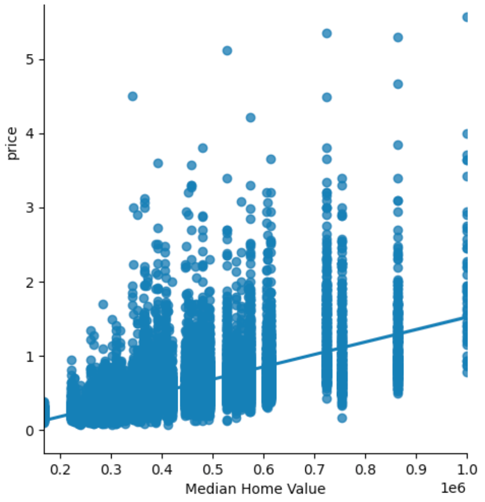


Figure 6: Median home value

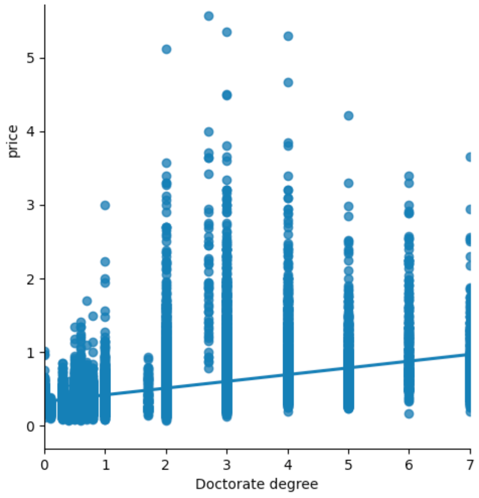


Figure 7: Proportion of Doctorates

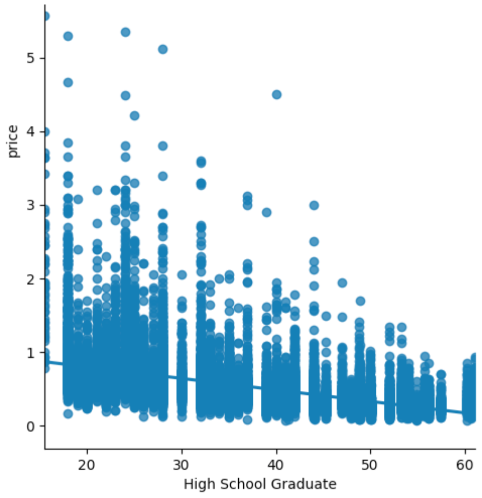


Figure 8: High-school-graduates

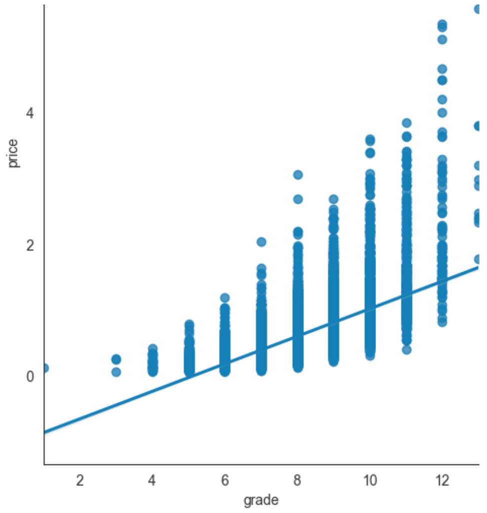


Figure 9: Grading of houses

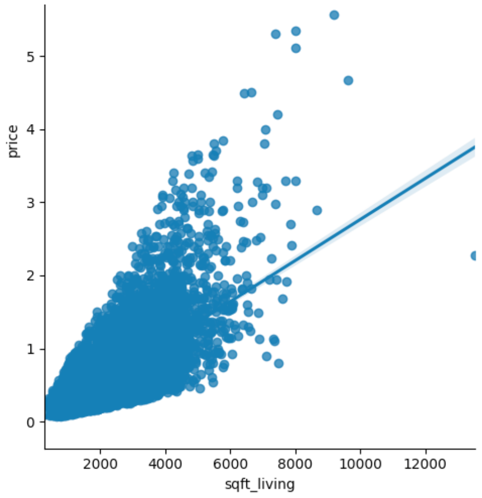


Figure 10: Size of living space

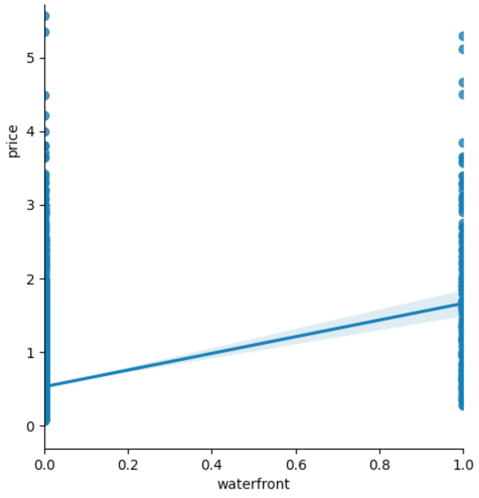


Figure 11: Waterfront

3.3. Correlation of features. I want to investigate which features I should use for my model now. Features that have mainly the same meaning, that are highly correlated to other features or that do not have any statistical relationship to the target variable *price* will not help to improve my ML-model. This means that I can remove them. To recognize them, I will print a correlation map which shows which features correlate very strongly to each other and which features do not correlate.

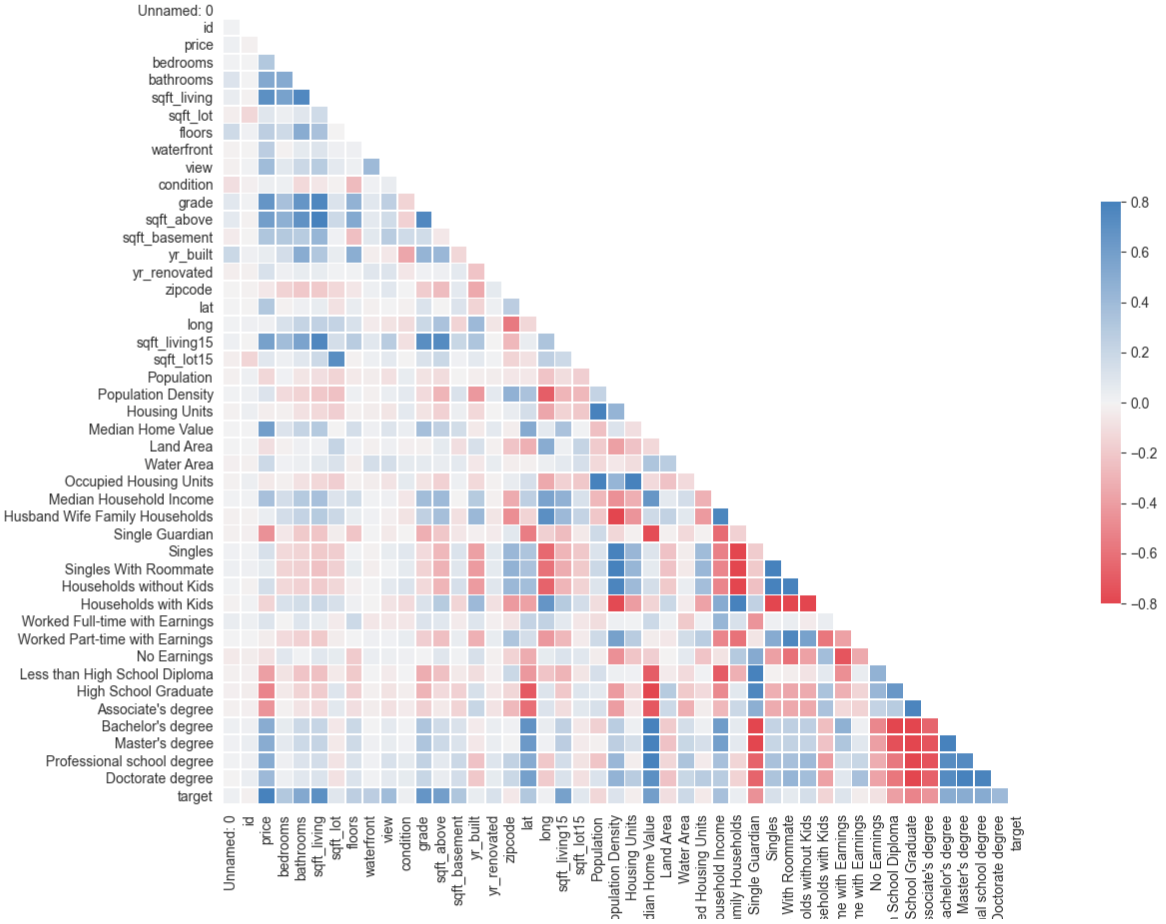


Figure 12: Correlation map calculated with the pearson score

This figure measures the correlation between two features. With that graphic you can see which features have the same message and which features do not influence the price. But some features correlate strongly to the price. Some of them belong to the main dataset and some of them belong to the ZIPcode-dataset. The attributes *bedrooms, bathrooms, sqft_living, grade, sqft_above, median home value, median household income and educational qualification (containing every attribute belonging to education)* seem to influence the price of a house the most. There are also some correlations between features, like the correlation *sqft_above* and *sqft_living*. This might decrease the accuracy of some ML-models. That's why I will choose only one attribute for each tuple of strong correlating attributes.

3.4. Hypothesis based on visualisations. In this section I will describe some of the features I will use and why I think these will increase the model-accuracy.

Hypothesis1: Educational Score. The correlation plots and pair plots pointed out that the educational qualification influences the house price (see Figure 7 & 8). Up to now there are 7 categories (like *doctorate degree*). I will merge them to one *EducationalScore*. I tried different configurations to get a good correlation between this score and the HousePrices.

$$
\begin{gathered}
EducationalScore = \\
3 * (2 * DoctorateDegree + 1.5 * ProfessionalSchoolDegree + MastersDegree) \\
+ 1.5 * (BachelorsDegree + AssociatesDegree) \\
+ 1 * (LessThanHighSchoolDiploma + HighSchoolGraduate)
\end{gathered}
$$

This leads to the following pair plot between the score and house prices. You can see that the correlation between educational qualification and house price still exists.

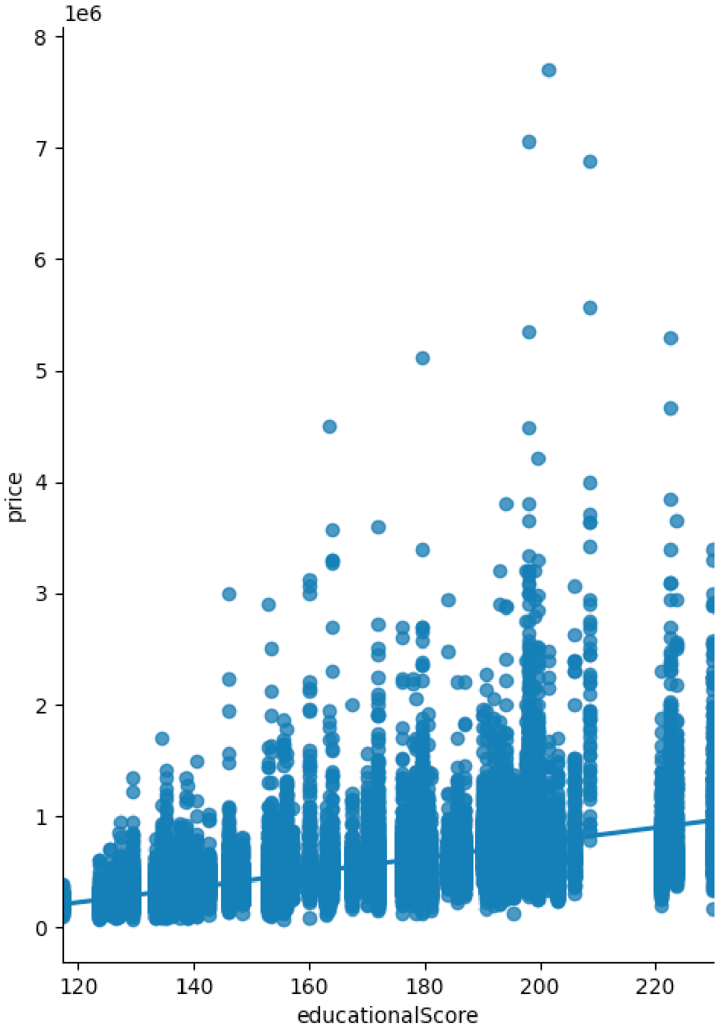


Figure 13: Correlation between the house prices and the educational score

Hypothesis2: LocationScore. As I have shown in section 3.2, the house prices differ regarding the longitude and latitude. I will test whether the accuracy of my model will change when using the location information and when not using the location information. For that I create a score which will describe the environment of every house.

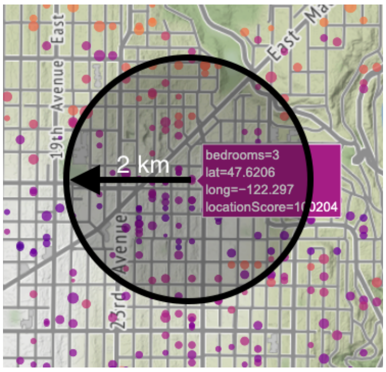


Figure 14: Houseprices depending on their location

My algorithm collects for house X every other house which is located in a radius of max. 2km to the house X. I use these houses to calculate a score which consists of several differently weighted averaged attributes. The algorithm iterates over the houses sorrunding house X and then uses attributes like the average amount of bedrooms, the average of the eductionalScore and so on. The calculated score will represent the "environmental value" for each house.

$$LocationScore =$$

$$(2 * \varnothing(waterfront) + 0.5)$$

$$* 3 * \varnothing(sqft\_living) * \varnothing(bedrooms) * \varnothing(grade) * \varnothing(educationalScore)$$

I calculate this score for each house. You can see below (Figure 15) that the LocationScore has a correlation to the price: The higher the locationScore, the higher the price gets. Furthermore I wanted to crosscheck my insights from Figure 3 (prices visualised in a map). You can see the same "belt" in the northern middle of the map.

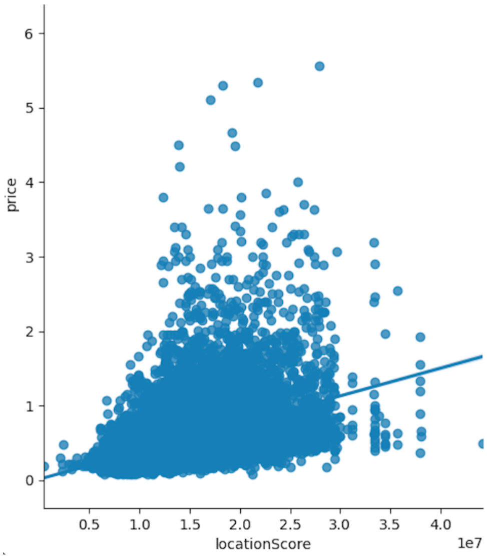


Figure 15: Correlation between the locationScore and the price.

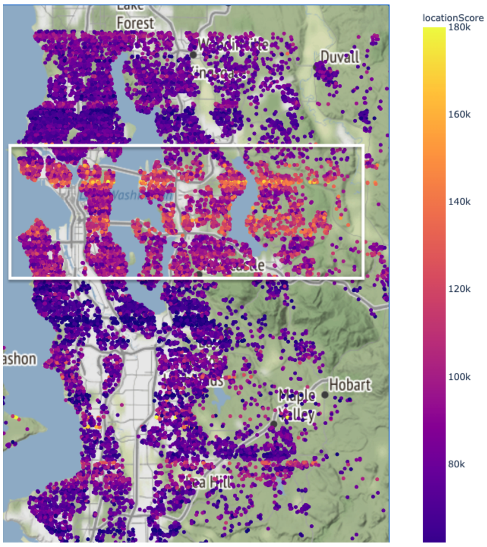


Figure 16: LocationScore of each house mapped on a map of Seattle

4. Developing the machine learning model

4.1. Approach. I will compare multiple machine learning models trained with different parameters and different attribute sets. I will then evaluate the differences between each model. I will train my models on different feature sets to see differences in the results. I will evaluate the influence of the crawled dataset including information about every ZIP-code. Furthermore I will take a look at the effectiveness of my calculated scores.
Used feature sets:

- *everyAttribute*: This feature set includes every attribute from both datasets and both calculated scores.
- *only MainDataset*: This feature set includes only attributes from the main dataset (information about each house). It does not include the calculated scores.
- *MainDataset + Scores*: This feature set is equal to the feature set above, but also includes the *locationScore* and *educationalScore*.
- *selected Features*: This feature set includes my hypothesis of the most relevant attributes. I got the insights to decide on relevant or not from my data exploration (Section 3).

4.2. Simple Linear Regression. First of all, I will train a simple LinearRegression-Model. I run this model on different feature sets, to see the differences in each one.

FeatureSet	Root-mean-square-error	R^2-score
everyAttribute	192956.71	76.7%
only MainDataset	236167.12	65.1%
MainDataset + Scores	211493.90	72.0%
selected Features	196415.87	75.9%

Table 2: Metrics of a simple linear regression Machine Learning model with differently selected features.

You can see a large differences between the different feature sets. Especially the feature sets which do only include attributes from the main dataset do not perform very well. It get's clear that when you use the main dataset and the calculated scores as a feature set the performance increases noticeable (RMSE decreases from 236.167 down to 211.493). This shows that the house price can be predicted more precisely when taking information from the sorrundings of a house into account. This seems to support my previously established hypotheses.

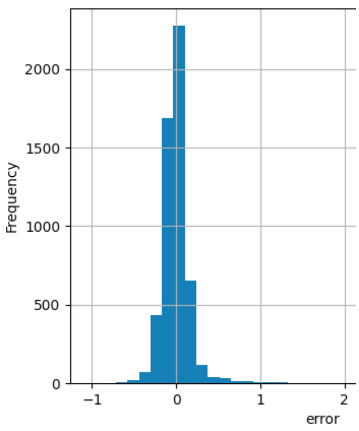


Figure 17: features from main dataset and scores

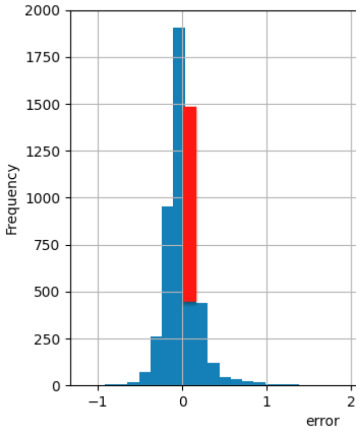


Figure 18: features from main dataset

Figure 17 & 18 show the effect of adding the calculated *locationScore* and *educationalScore* in more detail. When using only features from the main dataset, the model predicts many prices lower than they are in real (see red bar in Figure 18). Figure 17 does not have this problem because it includes information about the sorrounding environment of a house. If a house is in a more expensive area, it is more likely to have a higher price.
I will use k-fold crossvalidation to recognize overfitting:

```
model = LinearRegression()
scores = cross_val_score(model, x_train, y_train, cv=5, scoring='r2')
print(scores.mean())
print("std: ", scores.std())
```

Linear regression results in a average r^2-score of 79.9%. The low standard deviation 0.01 indicates that the model does not overfit. Furthermore you can see that the test & train $r^2 - score$ below does not strongly differ from the crossvalidation-score (mean).

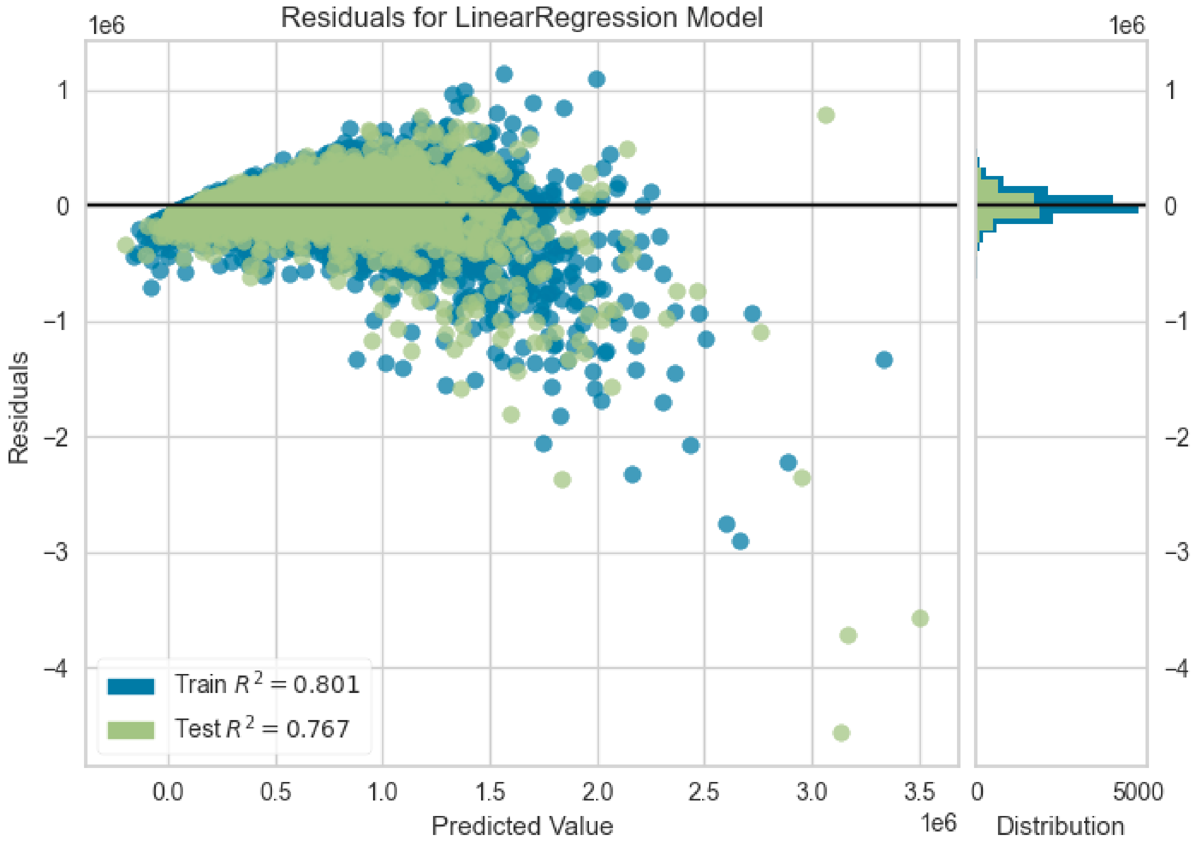


Figure 19: This shows the the variance of the error (residuals expresses the differences between the observed value and predicted value of the target variable).

I now plot a residual plot which should give a better insight into the error of my model. One can see the same insight as from Figure 17& 18. When the model has to predict high house prices, my LinearRegression-model predicts values that are too low. We further see that a linear model does not seem to be the best fitting model. As you can see it my model has in the residual plot a slight curve to the down which often indicates that a nonlinear model [1] could lead to more precise predictions. It seems like there is no linear but a nonlinear relationship between the features and the houseprices.

Because of that I will use a polynomial regression model as my next model:

4.3. Polynomial Regression. I will further train a LinearRegression model but the curve will be a higher order polynomial curve. To do so, I create polynomial features which include all the polynomial combinations of the selected features. Then I train my model with a LinearRegression-model, based on the new created features:

```
#polynomial features
features_polynomial = PolynomialFeatures(degree=2)
x_train = features_polynomial.fit_transform(x_train)
x_test = features_polynomial.fit_transform(x_test)
model = linear_model.LinearRegression()
```

[1] Interpreting residual plots

This model trained on the different feature sets has the following results:

FeatureSet	Root-mean-square-errors	R^2-score
everyAttribute	137056.60	88.3%
only MainDataset	188828.44	77.7%
MainDataset + Scores	156624.20	84.7%
selected Features	132012.77	89.1%

Table 3: Metrics of a polynomial Regression with differently selected features.

It gets clear that the error could be reduced compared to use a simple LinearRegression-model. This supports my hypothesis that a nonlinear-model does match a lot better than the simple LinearRegression. We can further see that information about the location of the house helps a lot and improves the $R^2 - score$ by approximately 10%. I receive the highest $R^2 - score$ by using a manually selected feature set. This feature set includes a combination of both datasets (I selected it from the insight I got during the data exploration, especially Figure 12) and the *locationScore* and *educationalScore*. Furthermore I removed some features which were correlating to other features in the feature set (so representing the same informatiom) and I removed features which did not had any influence on the house price (like *yr_built*).
Let's analyse this model a little bit more in detail: First of all, I will check for possible overfitting: I run a very similar code for crossvalidation as above, this time I received got an average $r^2 - score$ of 88.1%. You can see the detailed behaviour below. The $r^2 - score$ does not differ very strongly.

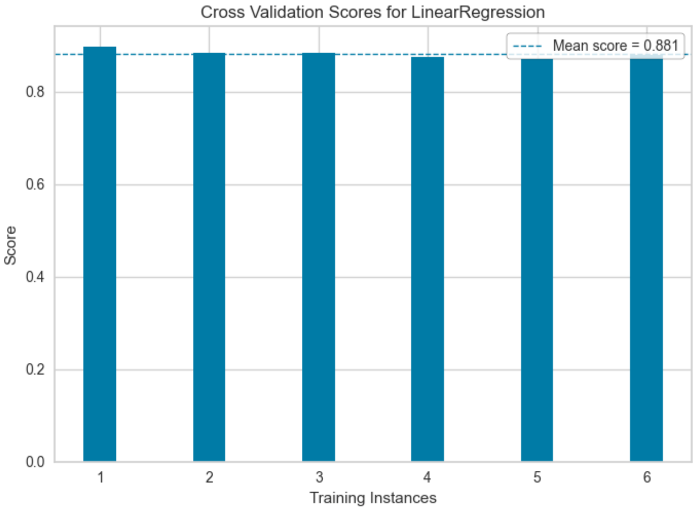


Figure 20: $r^2 - score$ for each run

I selected a polynomial degree of 2. With a higher degree, I got a much higher gap between my test and train score which usually shows that my model does not generalize very well. I further tested other regressors (e.g.: DecisionTreeRegressor) but none of these performed better than the *LinearRegressor* (combined with polynomial regression).

5. Summary

Let me summarize my work:

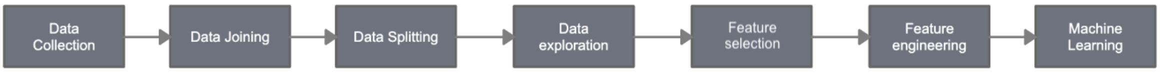


Figure 21: From raw data to a prediction

I started downloading my dataset and exploring it. After doing this, I continued with looking for a dataset which could add further information. Since there were no suitable data available, I had to crawl the geographical data from a website (& parsing the html file) and join it with my main dataset via the zipcode.
The next step was to analyse the dataset and extract the most correlating features to predict the price of one house very precisely. In the following, I created own features to add an important information: Data about the direct environment of one house. The *locationScore* described the "value" of the environment of one house which directly correlates to the house price: A house is in an environment with a higher *locationScore* (meaning in a "better" area), is often more expensive as an area with a lower *locationscore*.
After finishing the preprocessing steps, I started developing the ML-models. Figure 22 shows my first ML model. This is only a simple linear-regression-model, trained on the main dataset without any local information. You can see that the higher the house-price is the higher the error of the model gets, too (this causes the "drift to the right" on Figure 22). I now tried to improve this model by adding the new crawled dataset (containing information about each ZIP-code) and adding new features like the *locationScore* & *educationalScore*. By changing my model to a polynomial regression model and using the new features I could improve the $r^2 - score$ by approximately 24% to 89%. Furthermore I could lower the root-mean-square-error from 236.167 to 132.012. This is an reduction by approximately 45%.

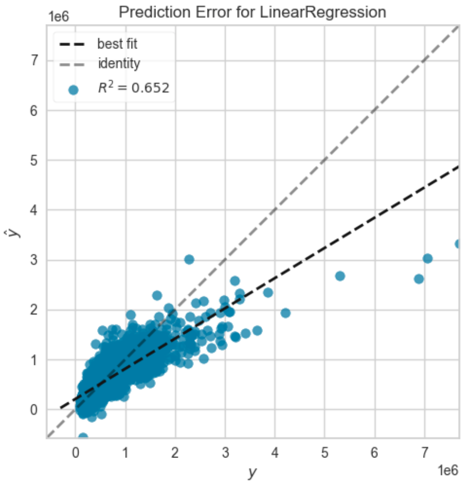


Figure 22: only maindataset as feature and linear regression

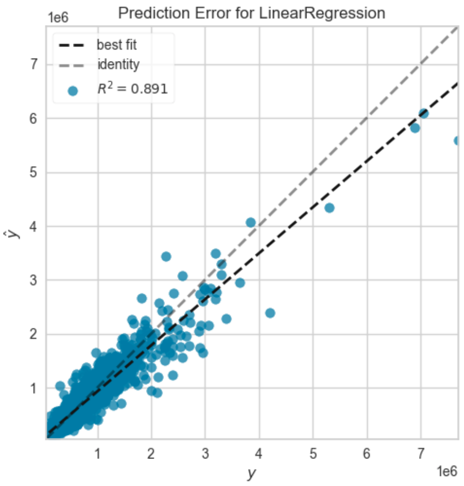


Figure 23: specific selection of features and polynomial regression

My model continues tending slightly to predict lower prices as in real world. This might be adjusted by handling the outliers and continue training the hyperparameters. This could make the model even more accurate.

In summary, it can be ascertained that the added ZIP-code dataset included important information about each house. It shows that aerial information is very important because they can influence the house price the same as the house price is influenced by specific house data like the amount of bathrooms.