Predicting house prices in King County

1 Introduction

Buying your own home can be one of the biggest decisions of your life. The price of an apartment is affected by several factors and the purchase decision is rarely easy. In this project, I try to facilitate this process and aim to predict the house prices for houses with different features in King County.

We aim to make predictions about the price based on some simple parameters. Our model should learn from the data and be able to predict the price of the house with a new combination of features. The predictions could be done using different models such as linear regression and logistic regression.

The structure of this report is as follows: the machine learning problem is explained in more detail in the problem formulation section. In the methods section, the feature selection and used machine learning models are described. More detailed information on data processing is also provided in the methods section. In the result section the results are compared and there is a summary of the results and the report in the conclusion section.

2 Problem formulation

The data points contain information about houses at different locations in King County (USA) between May 2014 and May 2015. There are 16 different features. There is a lot of basic information such as the number of bedrooms, floors, and bathrooms. 0.5 bathrooms means that there is a toilet in the room without a shower. Building year, possible last renovation, and zip code are also included. There is also information about square footage of land and interior living space. Interior living space is divided into above square footage and basement square footage. Above square footage means the square footage above ground level and basement square footage the opposite. Waterfront is described with a dummy variable for whether there is a water view or not. Other features of the apartment are described on different scales. The view is described with an index from 1 to 4 of how good the view is. Condition is an index from 1 to 5 and grade of the apartment from 1 to 13. Grade 1-3 has poor construction and design, 7 is average and from 11 to 13 is high-level quality. There is also included the average square footage of interior and land of the 15 nearest neighbours.

2.1 Summary of the problem

Label: the price of the house. **Features**: number of bedrooms, number of bathrooms, square footage of the apartments interior living space, square footage of the land space, number of floors, whether there is a water view or not, view(0-4), condition(1-5), grade(1-13), the square footage of interior living space that is above ground level, the square footage of interior living space that is below ground level, building year, year of the house's last renovation, zip code, the square footage of interior living space that for the nearest 15 neighbours and the square footage of land of the nearest 15 neighbours.

3 Methods

3.1 Dataset

For the project, I found the data from the Kaggle [1].

The dataset includes data for 21 614 apartments in different locations in King County, so we have enough data to solve this problem with machine learning methods. We use price as a label and 21 other columns could be used as a feature. To get an idea of the correlation of different properties, we make a heat map which is an easy way to compare features. Based on the heat map, number of bedrooms, number of bathrooms, grade, the square footage of interior living space that is above ground level and the square footage of interior living space for the nearest 15 neighbour correlates the most with the price. Some features are dropped because they are not relevant to the problem. The id, date of sale, latitude, and longitude was removed from the dataset because we do not want to study their effect on price. If we wanted to include the date, we should have data from the longer term. We also get a good understanding of the apartment's location from the postal code and thus we do not need the exact location of the apartment. There were no missing features or labels in the data set. All the features were scaled between 0 and 1 because it makes it easier to compare different features. After comparing the correlations in the heatmap and deleting the extra features we are left with 16 features.

Datapoints were split into training (80%), validation (10%), and test (10%) sets. The validation set is a separate section of the dataset that is used to evaluate the model trained on the training set to get a little sense of how the model works. In this case for example we can compare the success of the training process with different training sizes with the validation set. We can evaluate the final model performance of the chosen model with the test set. We want to keep the training set as big as possible because the more samples we have in the training set the better opportunity the algorithm has to understand the dependencies of the features.

3.2 Linear regression model

The first model I used was the linear regression model. We try first a linear model because if thinking with common sense, there could be a relatively strong correlation between these different features and price. I believe that it is good to start with a linear model if you are not quite sure what model to try first.

In the linear regression model, we assume that there is a linear relationship between continuous variable y and one or more independent variables X. The model takes the form:

$$h(x) = w_0 + w_1x_1 + \cdots + w_nx_n$$

where y is the predicted label, x’s are the values of different labels, w_0 is a constant term and w_n‘s are weights. We use linear regression to search optimal weights w_0 and w_n from a linear hypothesis space. As a loss function, we use mean squared error. We try to minimize the error by trying different weights w_n. Mean squared error is a convenient way to determine how “good” a model is. Also, according to the literature, it is smart to use mean squared error with linear regression [2]. Mean squared error is defined as:

$$MSE = \frac{1}{m}\sum\nolimits_{i=1}^{m}\left(y^{(i)} - h_w(x^i)\right)^2$$

Sklearn.linear_model.LinearRegression was used to do the linear regression in Python [3].

4 Results

5 Conclusion

6 Appendices

The code can be found from Git:

https://github.com/Valdde/Machine-Learning.git

References

[1] https://www.kaggle.com/harlfoxem/housesalesprediction

[2] https://vitalflux.com/mean-square-error-r-squared-which-one-to-use/

[3] https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html#sklearn.linear_model.LinearRegression