Real estate price prediction - linear regression vs. random forest regression

1 Introduction

This report presents a project that was implemented for the Aalto University Machine Learning course. The goal of the machine learning project was to predict property prices of Melbourne's housing market. Two regression models were used to obtain the predictions: linear regression and random forest regression.

Chapter 2 presents a formulation for the machine learning problem and justifies the choice of data points, features and labels. Chapter 3 explains how the regression analysis was conducted for both models. Chapter 4 presents the results of the analysis.

2 Problem Formulation

Properties in Melbourne's housing market represent the data points of the analysis. All the column names of the data set are explained in table 1. The price of a property was chosen as the label of the ML problem. The list of features is as follows (using the notation of table 1): `Rooms`, `Distance`, `Bedroom`, `Bathroom`, `Car`, `LandSize`, `BuildingArea`, `YearBuilt`, `Latitude` and `Longitude`.

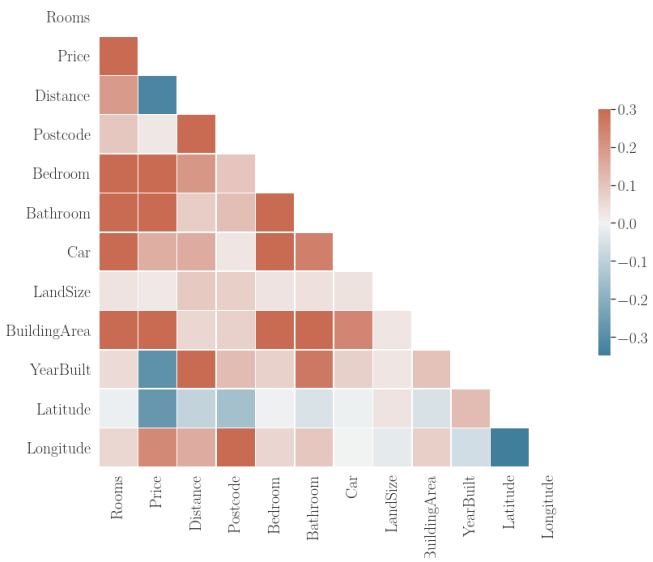

Figure 1: Correlation matrix plot for the numeric columns.

Figure 1 represents a heat map of the numeric columns in the data set. According to the heat map, `Rooms`, `Bedroom`, `Bathroom` and `BuildingArea` correlate positively with the price of the property. Hence, they were included in the feature set. In addition, the heat map suggests that `YearBuilt` and `Distance` have quite a strong negative correlation with the price which is why they were included in the feature set, too. According to the heat map, latitude and longitude also correlate with the price and they turned out to have a great impact on the regression accuracy. That's why, they were chosen as features, too.

Furthermore, `LandSize` was used as a feature, because larger lots tend to be more expensive when comparing properties in the same city. In addition, more expensive properties tend to have more car spots which is why `Car` was included in the feature set.

3 Methods

The regression analysis was based on a data set of Melbourne's housing market prices that can be found on Kaggle. The data set contains 34857 rows and 21 columns. The rows in the data set represent properties that have been for sale on the market. Not all properties have necessarily been sold for the price mentioned in the data set, however. For example, for some properties the bidding price did not reach the seller's reserve price and the property was not sold. The columns represent different characteristics that the properties

Column name	Description
`Suburb`	Suburb of the property
`Address`	Address of the property
`Rooms`	Number of rooms
`Price`	Price in Australian dollars
`Method`	e.g. property sold, withdrawn prior to auction...
`Type`	e.g. house, townhouse, cottage...
`SellerG`	Real estate agent
`Date`	Date sold
`Distance`	Distance from Melbourne's Central Business District in kilometers
`RegionName`	General Region (West, North West, North, North east, etc)
`PropertyCount`	Number of properties that exist in the suburb
`Bedroom`	Number of bedrooms
`Bathroom`	Number of bathrooms
`Car`	Number of car spots
`LandSize`	Land size in square meters
`BuildingArea`	Building area in square meters
`YearBuilt`	Year the house was built
`CouncilArea`	Governing council for the area
`Latitude`	-
`Longitude`	-

Table 1: The columns of the data set.

have, for example the number of bedrooms or bathrooms.

When preprocessing the data, properties that contained clearly unrealistic data were removed from the data set. For example, properties with a lot size or building area of less than 10 m^2 were removed. In addition, this analysis only considers properties that were actually sold for the price mentioned in the data set. Furthermore, propertied built before year 1900 were excluded from the data set. After these operations, the data set contained 5175 properties. The large drop in the number of properties results from the fact that quite many properties weren't actually sold because the bidding price didn't reach the reserve price. Also, the final sale price of quite many properties was not disclosed. Next, properties with missing values for any of the numeric column values were excluded from the regression analysis. After removing those rows, the data set contained 4555 properties. Figure 2 visualizes the numeric columns of the data set after preprocessing the data. The figure suggests that there is a lot of deviation in the data set so obtaining good predictions might be difficult.

`scikit-learn` library's `LinearRegression` and `RandomForestRegressor` models were used for solving the ML problem. Linear regression finds such a line that minimizes the average error between the predicted label values and the correct label values. In turn, random forests utilize multiple decision trees in predicting labels. According to the documentation of `scikit-learn`, a decision tree can be seen as a piecewise constant approximation. In `scikit-learn`, the default loss function for both linear regression and random forest regression is mean squared error. Mean squared error has the following form:

$$\text{MSE}(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=1}^{n} (y - \hat{y})^2,$$

where n is the number of data points.

I used `scikit-learn` library's `train_test_split` for dividing the data set to separate train, validation and test sets. `train_test_split` divides the data set with a single random split into two sets. That's why, `train_test_split` was used twice. Of all the data points, the train set contained 60%, the validation set 20% and the test set 20%.

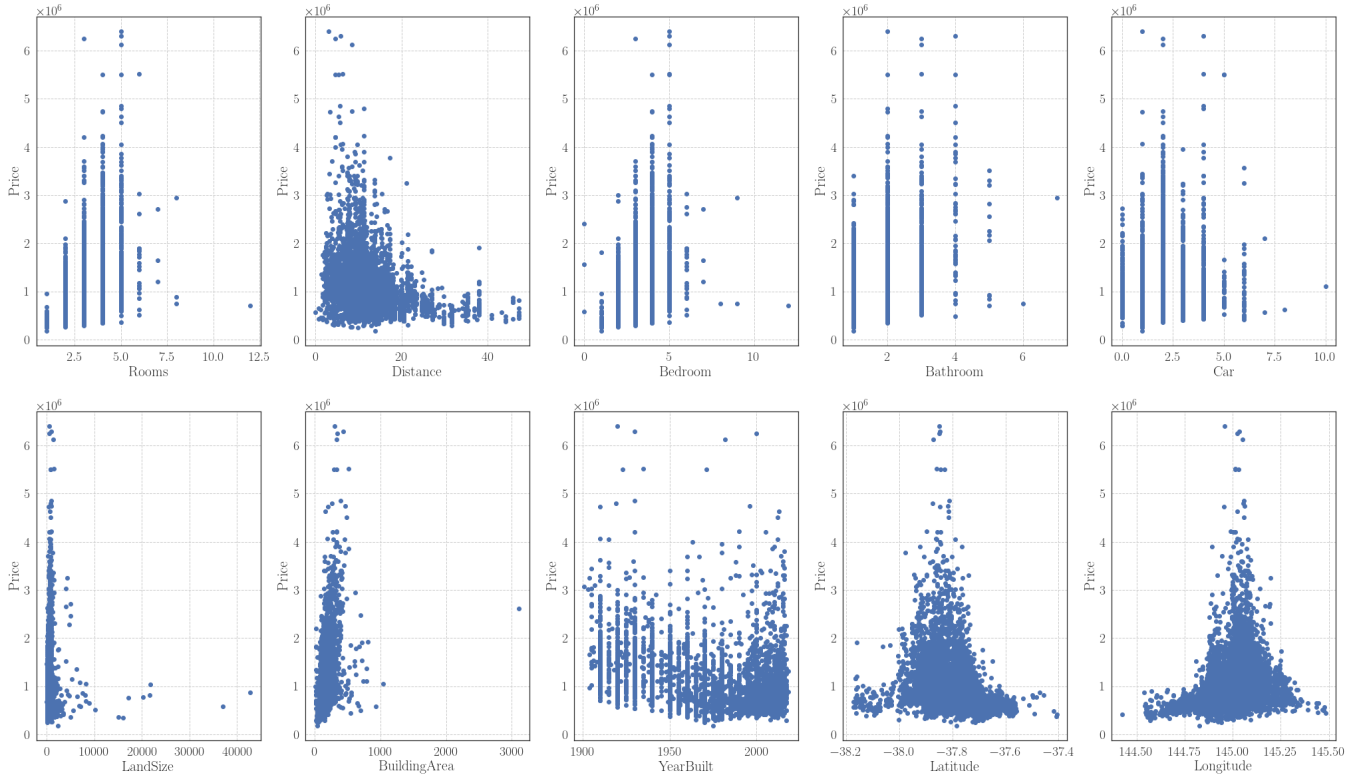

Figure 2: Scatter plots of the numeric columns against property price.

4 Results

The results of both models were assessed by using median absolute error (MAE). MAE has the following form:

$$\text{MAE}(y, \hat{y}) = \frac{1}{n} \cdot \sum_{i=1}^{n} |(y - \hat{y})|.$$

The prediction accuracy of both models was good for most of the properties. However, root mean squared error is not tolerant against even a small number of large errors. That's why, median absolute error gives a better understanding of the prediction accuracy in this case and hence, it was chosen as the evaluation metric.

Both linear and random forest regression yielded good results. The left plot in figure 3 presents the median absolute errors that were obtained with linear regression for the train and validation sets. The median absolute error for the validation set was approximately 193300 (Australian) dollars. The median property price in the validation set was \$903000 resulting in a median error of 21.4%. That is quite a large relative error. However, as can be seen from figure 2, there is a lot of variation in the data set. Moreover, none of the features in the figure seem have a linear enough a relationship with the price to obtain excellent results with linear regression. R^2 score can also be used as an evaluation metric when analysing the prediction accuracy. Linear regression obtained an R^2 score of 0.56 which is pretty decent.

The right plot in figure 3 presents the median absolute errors that were obtained with random forest regression for the train and validation sets. MAE for the validation set was approximately \$111000 resulting in a median prediction error of 12.2%. The R^2 score for random forest regression was 0.79 for the validation set, which is a great score.

Because the validation error was smaller for random forest regression than for linear regression, random forest regression was chosen as the better model for the ML problem. The test error for random forest regression was approximately \$111200.

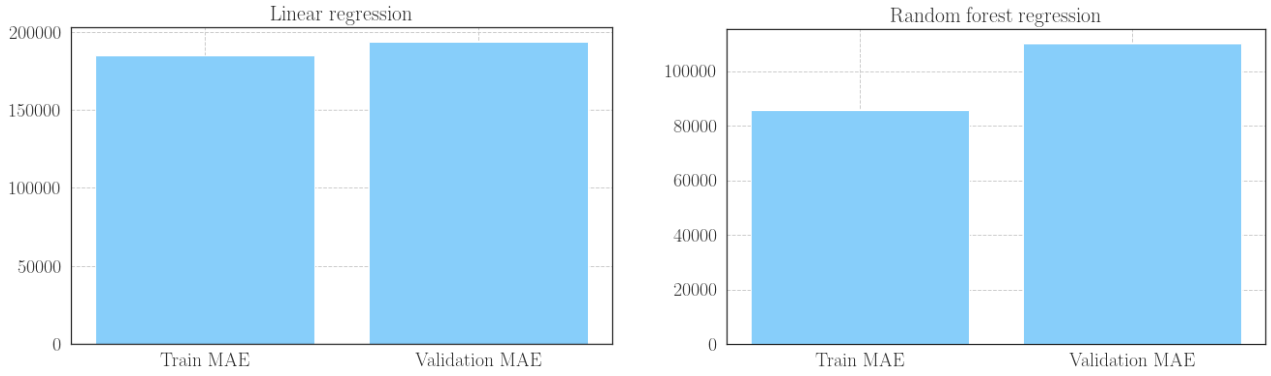

Figure 3: Median absolute errors for both models and for both train and validation sets.

The left plot in figure 4 shows the median absolute errors in price in an increasing order. As can be seen from the plot, the prediction accuracy of random forest regression was good for most of the properties. Also, the middle plot in the figure shows that the predictions were good since most of the scatter points are quite close to the x-axis. The middle and right plots visualize that the prediction errors were largest for the most expensive properties.

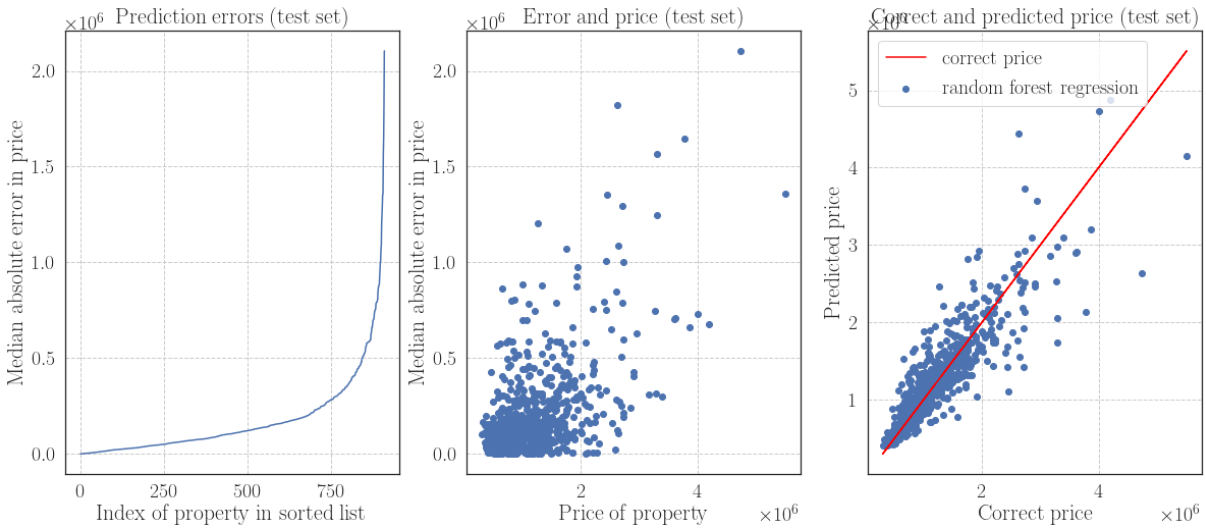

Figure 4: Prediction error was large only for a small subset of the properties (left). The error was larger for more expensive properties (middle). Predicted price plotted against correct price (right).

5 Conclusions

Figure 3 hints that the random forest regression model would contain a bit more overfitting compared to linear regression. Random forests utilise multiple decision trees and decision trees tend to overfit the training data quite easily, particularly with a large feature set. When using the default parameters of the `scikit-learn` library's `RandomForestRegressor`, the ratio $\frac{\text{validation MAE}}{\text{train MAE}}$ was approximately 2.8, which suggests overfitting. To overcome the overfitting, I tried tuning the parameters of `RandomForestRegressor` according to a Stackoverflow answer. I first tried increasing the `n_estimators` parameter from the default 100 to up to 400, which results in using more trees. I also tried reducing the `max_features` parameter that indicates the number of features to consider when looking for the best split. I tried the parameter values 5, 4 and 3. However, tuning `n_estimators` and `max_features` had quite a small impact. Next, I tuned the `max_depth` parameter which determines the maximum depth of the trees. I started with a depth of 5 and gradually increased it to 9 which turned out the be a good compromise between the overfitting problem

and the validation error. The right plot in figure 3 and the train, validation and test errors mentioned previously where obtained with the parameter values `n_estimators`=200, `max_features`=3 and `max_depth`=9.

According to The Guardian, house prices started dropping in Melbourne after the 2017 price peak. The data set I used consists of properties that were sold between 2016 and 2018. Therefore, some of the deviation in the data set could be due to these sudden changes in Melbourne's housing market. Collecting more data points from a longer period of time could help in getting more accurate results. Because of the high R^2 score, random forest regression turned out to be a great model for this problem. However, it would be interesting to see how well an artificial neural network could cope with the large variation in the data set.