Sogang University

Introduction to Machine Learning

Final Project: Housing Price Prediciton

Name: Sanghyeon Kim

Student ID: 20181605

Due Date: 12/09/2020

**Objective**

The main goal of the project is to find the parametrized model that best fit the data, i.e., to find an optimized model that is as close as possible to the model that produced the data. In order to do so, we need to define our model class and the number of parameters it should have. Also, we must look at our data to extract important and relevant characteristics, which can be altered if necessary. Then, our model can be trained with the data. In this project, we are attempting to predict the median house price of each block using the training data. The steps taken to find the models will be explained in the following.

**Data Analysis and Modification**

**\*Graphs and other visualizations of data can also be referred to the IPython Notebook**

The training data contained 10 labeled columns and one unlabeled column. The unlabeled column is discarded because it is unclear what its values indicated. The Pandas library allows for easy formatting and manipulation of tabular data. The first step taken is to check for any data that is not numerical. Any non-numerical data should be converted before they are used to train our model. In the data, the column labeled ‘ocean\_proximity’ has string values, which categorizes how close each block is to the ocean. Proximity to the ocean is assumed to have a correlation with median house prices. The reasoning behind it is because the data is exclusively based on the population of California and California has most of its large cities closer to the ocean. The categories of the column have five unique values (‘<1H OCEAN’, ‘INLAND’, ‘NEAR OCEAN’, ‘NEAR BAY’, ‘ISLAND’). The string values are converted into integer values of increasing order according to the decreasing proximity to the ocean. Thus, the closer to the ocean, the smaller its integer values. The values range from 0 to 3 and the string values ‘NEAR OCEAN’ and ‘NEAR BAY’ are assigned the same integer value. Now the data fully consists of numerical values.

The next step is to find any missing values in the data set. The data contain 158 missing values in the ‘total\_bedrooms’ column. The instances of data with the missing values either have the option to be removed or to be filled with values according to the distribution of ‘total\_bedrooms’. Before applying any method to solve the problem, the distribution of data with missing values in ‘total\_bedrooms’ is compared with that of the rest of the data to see if there is an underlying difference between them. The distribution of each data is visualized as histograms with 30 bins using the matplotlib library. Looking at the distribution of data with non-empty values, we see that the ‘longitude’ is bimodal, ‘population’ is positively skewed, ‘latitude’ is also bimodal like ‘longitude’, ‘households’ is positively skewed, ‘housing\_median\_age’ is symmetric, ‘median\_income’ is positively skewed with range of values not as dramatic as other skewed categories, ‘total\_rooms’ is positively skewed, ‘ocean\_proximity’ is categorized into three integers, ‘total\_bedrooms’ is positively skewed, ‘target’ is weakly positively skewed. Having observed these characteristics, the distribution of 158 instances with empty values is compared to it. The distribution of the data with empty values does not have any unique characteristics relative to the distribution of data with non-empty values. Therefore, the instances with empty entries in the ‘total\_bedrooms’ is dropped from the data.

|  |  |
| --- | --- |
| 그림 1: Non-empty | 그림 2: Empty |

Now that the numerical data is modified to be fully utilized in training the model, the correlation between each column or from now on will be called the variable. The correlation coefficient matrix between each variable is calculated in order to check if there is any linear relationship between each pair of variables. There are few things to note from the correlation matrix. First, the ‘longitude’ and ‘latitude’ are negatively correlated with the coefficient of -0.92. The ‘total\_rooms’, ‘total\_bedrooms’, ‘population’ are all mutually positively correlated with values ranging between 0.86 and 0.98. However, the most important correlation between variables is that of ‘target’ and the rest of the variables. The goal is to predict the ‘target’, which is the median house price, with other variables. Looking at the last column or the last row of the correlation matrix, it can be seen that ‘ocean\_proximity’ is negatively correlated with ‘target’. Also, ‘median\_income’ is positively correlated with ‘target’. Because the analysis of the data is completed, the data is attempted to be fitted with a linear model.

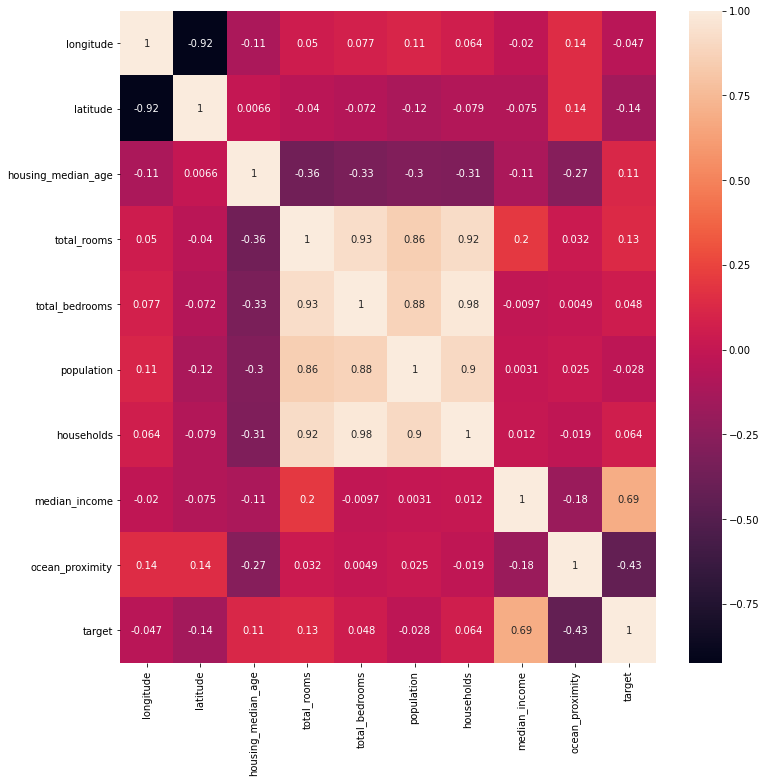


그림 3: Correlation Matrix

**Training/Optimizing the Model**

The linear model assumes that all features are independent of each other. However, in the first attempt of fitting the linear regression model with the data, no data is discarded despite some of the variables other the ‘target’ are correlated to each other. The linear model uses all variables except ‘target’ as the features for optimization of the parameters. After the model has been trained, we must evaluate it using test data. However, while the optimization is being done, there is a need to evaluate whether the model is not overfitting the data. Thus, estimating the generalization error through cross validation is needed. Therefore, 25% of the total data is set aside as the validation data and the other 75% of the data is used as train data. The linear regression model is trained, and the performance of the model is measured with RMSE. The value of the RMSE of linear regression model using all features is 70608.6 rounded to the first decimal value.

Next, the linear regression is done by only using the features that are linearly independent of each other. The features, ‘total\_bedrooms’, ‘population’, ‘households’ are dropped and the feature, ‘total\_rooms’ is kept because the ‘total\_rooms’ has the largest absolute correlation coefficient with respect to the ‘target’ compared to the other three variables. Using the reduced feature set to train the model, the value of the RMSE of the linear regression is 74140.5. The performance measure does not differ much from the first attempt at using all the features to train the model. Thus, it has been seen that the linear model performs well with a smaller feature set, which leads to a model with less parameters to optimize, when the features are linearly independent.

Now, an attempt at improving the model performance will be made by performing transformation on some of the features. From the analysis of the data, it has been observed that some of the features are very positively skewed resulting in a distribution that is far from normal. From the correlation matrix, it has been seen that the relationship between the features and the target are not necessarily linear. In linear regression, we not only assume that the features are linearly independent, but also the distribution of the features follow a normal distribution (normality of errors) and variance is kept constant. Moreover, if there exists a non-linear relationship between the features and the target, there is a need to transform the variables so that the effective relationship is closer to linear.

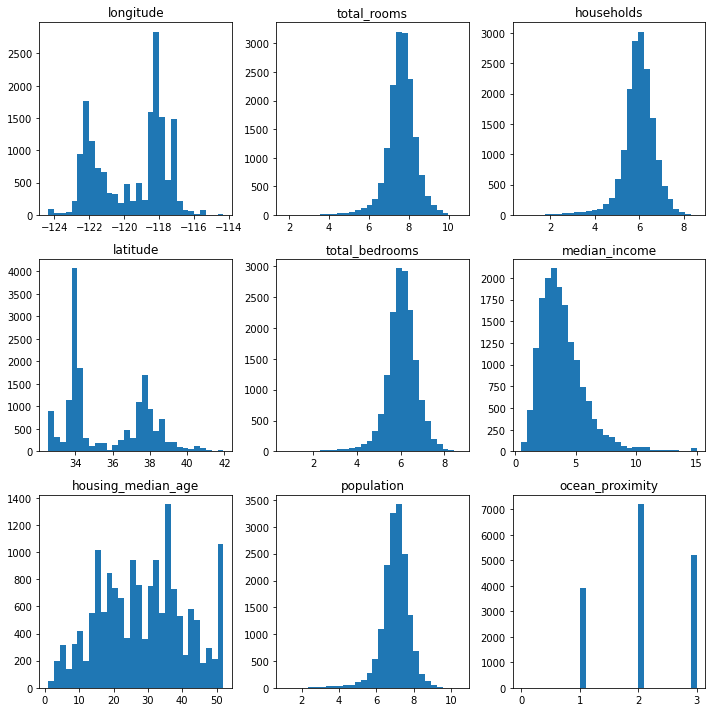


그림 : Log Transformation

To achieve this, the log transformation of the features is implemented. The features, ‘households’, ‘total\_bedrooms’, ‘total\_rooms’, ‘population’, are log transformed and the distribution of those features are presented in figure 3.

The data that were significantly right skewed are now closer to a normal distribution. Using these transformed feature set, the linear model is trained to get a performance uplift of about 3.3%. The log transformation was possible because the features applied consisted of positive data. Applying log transform on longitude would cause an error. The transformation also allowed the range of values to shrink by scaling the data, allowing the model to deal with smaller values.

**Limitations of the Linear Model**

The linear model is generally easier to interpret and implement than any other class of models. However, the assumptions of linear independence of features and their linear relationship to the target, must be satisfied to maximize the performance of the model. When these underlying assumptions on our data is false, the model will perform poorly on a given data. Using the log transformation, the possible multiplicative relationship between the features and the target is assessed, but it is not enough to model the data to its full extent.

**Use of Non-linear Models**

Because of the limitations with the linear models, the data is now fitted with two types of polynomial regression models. First with the polynomial regression, then with XGBoost. The polynomial regression is done by getting a polynomial feature set. All possible polynomial combinations of the features of degree less than or equal to 2 is calculated. These new features are then used to train the polynomial model so that the coefficients of the polynomial that best fits the data is found. The performance of the model measured in RMSE is found to be 60238.6. So far, general regression models have been implemented to best represent the data. However, decision tree-based algorithms have recently been used to solve many regression problems. XGBoost is an ensemble tree method that trains multiple decision tree algorithms in parallel and makes the final prediction based on the predictions and confidence of those decision trees. The model performs regularization and cross-validation inherently. Using XGB regression, the model achieves the best RMSE performance measure of 55825.9.

**Limitations of the Non-linear Models**

Compared to linear models, non-linear models can fit data with more complex relationship. However, as the complexity of the model increases the model is prone to overfitting the data. For example, using the polynomial regression, as the degree of the polynomial increases the model can optimize well to the training data. However, when the parameters are optimized to encapsulate all the training data, the generalization performance of the model diminishes. Therefore, methods such as cross validation and regularization are needed to monitor and prevent overfitting.

**Model Saved**

Various models are trained, but the linear model with logarithmic transformation was saved for simple saving and loading of model as well as for maximizing feature modification.