# XYZ Assignment: Machine Learning Summer 2017 Internship House Pricing Prediction

###### Pre- Processing

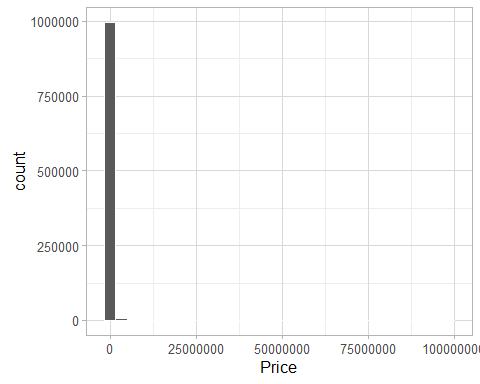
This is a report concerning the XYZ assignment about Machine Learning Summer 2017 Intern position. It is about UK house pricing prediction. Since we need to predict the house prices, we chose to implement a multiple linear regression model. However, as the dataset contains only categorical features related to property type, lease duration and location, we assume that our fit-model won't be able to perform well. Parameters such as location, property type and lease duration are taken into consideration when setting a selling price, but there are also other practical conditions such as No of bedrooms, garage etc., that in this particular dataset aren't provided.

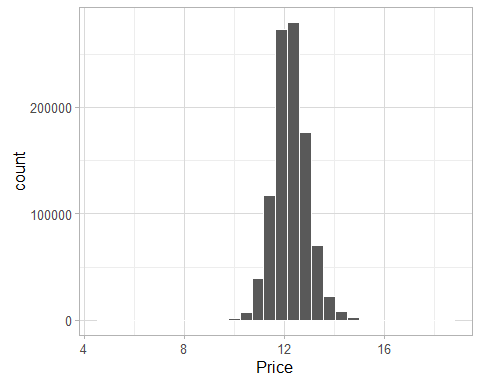
To begin with, we will do some data pre-processing to bring the initial dataset and variables in the desirable format. First, we import only the 4 variables that are of interest ("Price", "Property\_type", "Lease\_duration", "Town\_city"). Then we set names to the columns and keep just the year value for the "Date" variable. We chose to create a new dataset, containing only the records of year 2015, as it was impossible to run a 1.6Gb sized dataset (PS: I even used the trial AzureML and couldn't run linear regression algorithm, because of memory exhaustion). After that, we delete the column "Date" as it is no longer needed.

Furthermore, in order to save some training time, we decided to recode all the factor variables levels into numeric variables. It's easier, thus faster, for a training algorithm to read and process numeric values than character ones. Especially, for factor "Town\_city", we recoded it into 'London, 'Non-London', after spotting that London is by far, the level with the highest frequency, which means the most desirable city to reside (*Table 1*). Moreover, after checking the "Price" distribution by a histogram plot (*Figure 1*), we noticed that it is right-skewed, so performed a log-normalization technique, then plotted again to check whether the distribution came closer to normal (*Figure 2*). Finally, we checked for missing values and proceeded to the split training-test bootstrap method. The initial dataset was divided into training and test subsamples by 80% and 20% amount respectively.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LONDON | MANCHESTER | BRISTOL | BIRMINGHAM | NOTTINGHAM |
| 76.496 | 16.486 | 16317 | 14286 | 13124 |
| LEEDS | **LIVERPOOL** | **SHEFFIELD** | **LEICESTER** | **SOUTHAMPTON** |
| 11202 | 10064 | 9490 | 8934 | 8091 |

*Table 1. Town-City Frequency Table*

*Figure 1. Histogram of Price variable*

*Figure 2. Histogram of Price variable after log-normalization*

###### Modeling

###### Multiple Linear Regression

|  |  |  |
| --- | --- | --- |
| **Coefficients** |  |  |
|  | **Estimate** | **Std. Error** |
| **(Intercept)** | 12.342224 | 0.003038 |
| Property\_type2 | -0.436577 | 0.003405 |
| Property\_type3 | 0.199128 | 0.005213 |
| Property\_type4 | -0.487585 | 0.001902 |
| Property\_type5 | -0.666568 | 0.001846 |
| Lease\_duration1 | 0.310084 | 0.002834 |
| Town\_city1 | 1.162051 | 0.002663 |

­

|  |
| --- |
| **R-Squared**: 0.2984 |

At a first point of view, we conclude that our model is of low performance as the R-squared = 0.2984 is really low.

From this output, we can determine that the intercept is 12.34 and also the coefficients for the Flats/Maisonettes (Property\_type2), Other (Property\_type3), Semi-Detached (Property\_type4), Terraced (Property\_type5),  
Freehold(Lease\_duration1), London (Town\_city1) are -0.44, 0.2, -0.49, -0.67, 0.31, 1.16 respectively.

Therefore, the complete regression equation is:   
  
House Pricing *= 12.34 - 0.44 \* Flats/Maisonettes + 0.2 \* Other – 0.49 \* Semi-Detached – 0.67 \* Terraced + 0.31 \* Freehold + 1.16 \* London.*

This equation tells us that the predicted house pricing for UK in the year 2015 will increase by 0.2. 0.31 and 1.16 for every one percent increase in the Other Property type, Freehold tenure duration and London rate respectively. Also, it will be decreased by 0.44, 0.49 and 0.67 for every one percent increase in the Flats/Maisonettes Property type, Semi-Detached Property type and Terraced Property type rate respectively.

One last thing to note is that one level of each factor will not appear in the output. This is because when fitting a regression with categorical variables, one level must be left out in order to avoid overfitting ([dummy variable trap](http://en.wikipedia.org/wiki/Dummy_variable_(statistics))). In our model, Detached Property type, Leasehold tenure duration and non-London towns/cities are left out of the summary but still taken into account. The intercept gives the mean value of price for the ‘invisible’ levels. Thus, the coefficient for Flats/Maisonettes means that the mean price for a Flat/Maisonette will be 0.44(Pounds? / Euros?) smaller than the price for a Detached, regardless of the Lease duration and Town/City.

###### Diagnostic Plots

###### *Figure 3. Diagnostic Plots*

Observing the 4 diagnostic plots (*Figure 3*), we conclude that the assumptions of normality of residuals (*Normal Q-Q plot*) and homoscedasticity of variance (*Residuals vs Fitted, Scale Location vs Fitted*) are getting violated. Given that the standard assumptions are violated, the coefficients obtained are not reliable and should not be used.

###### Evaluation Metric

As a final move after scoring our model, we apply an evaluation method.  
For linear regression, a suitable error metric method is Root Mean Square Error.   
It gives the standard deviation of the model prediction error.   
In this model RMSE is 0.59, a pretty high one, which means that our model isn’t good. In general, the lower the RMSE, the better for model performance.

###### Conclusion

As expected the multiple linear regression model didn't perform well. This happened not only because the dataset lacks of necessary information but also because all our independent features are categorical. In this case, an Ensemble model like Random Forest might perform a bit better. Random Forests build multiple Decision Trees and correct the overfitting problem. Or even a more accurate and faster Boosting model like xGBoost.