ReadMe

1. Run the Preprocessor.py file first, then SplitTrainingData.py, then run the main code (HousePricing.py)
2. I split the training file into two files due to memory errors when running it. If the whole training file can be run instead then just change the training\_split.csv input to the training.csv in the read in section of the code.

# Linear Regression Model:

Looking at the lease duration, of which there are two in the data set: freehold and leasehold. So it made sense to turn the two types into 1's and 0s using scikit learns label encoder.

## SplitTrainingData.py & Preprocessing.py:

Before using the files in my model I preprocessed the given dataset, splitting it into training and testing data. Training data using the rows, or observations before 2015 and the testing data being the observations during 2015. The training data set was still quite large after this, so to save time when processing the final model I made another parser to split the training file into two again. Taking every second row and writing it to another training file. Both these parsing tasks could have been done in one module, however I wanted to have the choice of using them separately.

## HousePricing.py:

Holds all code that takes in the data, splits data into the needed features matrix and response/target vector then models data, training and fitting it and finally evaluating the model using a variance score.

### Taking in data:

Used pandas to create a panda data frame so I could specify my own column names for the file for use with the features matrix and target vector.

Features vector has one feature since I could view it as on and off, or one or zero easily. To see a correlation between the lease type and the price. Of course the other factors come into play, this was just a simple view of it.

Target vector is just the price of the house as needed.

### Modelling Process:

Scikit Learn has a four step modelling process since it uses interfaces for the model types, which is fantastic! Object orientated languages make life so much better…

The four steps are simple, import the model, instantiate or create the model object then fit the model and make predictions based on the test data.

For linear regression is one of the simplest models, widely used. It fits a linear function to a set of data points where the function is:

Y = B0 + B1\*X1 + B2\*X2 … Bn\*Xn

Y is the target variable and X1,X2..Xn are the predictor variables and B1..Bn are the coefficients that multiply predictor variable as well as B0 being constant.

The sum of squares simply represents a variation from the mean values, used this for prediction.

### Evaluation:

Variance score is the difference between the predicted score and the actual score. Calculating the variance of the score set. Where the best possible score is 1.0, and lower values are worse.

There are other ways of evaluating the models such as confusion matrices, but I am not sure if they are applicable here.