OTTO VON GUERICKE UNIVERSITÄT MAGDEBURG

WW

FACULTY OF ECONOMICS AND MANAGEMENT

## Chair for Innovation and Financial Management

## Prof. Dr. Elmar Lukas

### Master's Thesis

# *Determinants of house price: Multiple comparisons of machine learning regression algorithms*

Submission date: 02.11.2020

Name: Dang Tran Khanh

Email address: khanh.dang@st.ovgu.de

Matriculation number: 220678

Semester of study: 7

Study program: Operations Research and Business Analytics

# I Table of contents

# II List of abbreviations

| | |
|---|---|
| AdaBoost | Adaptive Boosting |
| ANNs | Artificial neural networks |
| CAPM | Capital Asset Pricing Model |
| CV | Cross-validation |
| GD | Gradient descent |
| IG | Information Gain |
| LASSO | Least Absolute Shrinkage and Selection Operator |
| ML | Machine-Learning |
| MPT | Modern Portfolio Theory |
| MSE | Mean Squared Error |
| MSP | Maximum Sharpe ratio Portfolio |
| MVP | Min Variance Portfolio |
| RANSAC | Random Sample Consensus |
| SFS | Sequential Forward Selection |
| SSE | Sum of Squared Errors |
| SSR | Residual Sum of Squares |
| SST | Total Sum of Squares |
| SVM | Support Vector Machine |
| XGBoost | Extreme Gradient Boost |

# III List of symbols

| | |
|---|---|
| $1$ | Vector of ones |
| C | Hyperparameter for tuning the model |
| $d$ | Degree of the polynomial regression |
| $I$ | Impurity function |
| k | Number of independent variables |
| $n$ | Number of samples in the dataset |
| $m$ | Number of dimensions |
| $r$ | The vector of expected rates of return |
| $R^2$ | Coefficient of determination (or R-squared scores) |
| $R_{adj}^2$ | The adjusted coefficient of determination (or Adjusted r-squared scores) |
| $x$ | Explanatory variable or Matrix of explanatory variables |
| $y$ | Target variable (or dependent variable) |
| $w_j$ | Weight coefficient of explanatory variable *j* |
| $x_j$ | Explanatory variable (or independent variable) or feature *j* |
| $\hat{y}_i$ | Predicted target variable by sample *i* |
| $w^T$ | Inverse matrix of the weights matrix |
| $D_P$ | Subset of training samples |
| $D_{left}$ | The subsets of training samples at the left |
| $D_{right}$ | The subsets of training samples at the right |
| $N_{left}$ | Number of samples in the left child nodes |
| $N_{right}$ | Number of samples in the right child nodes |
| $y^{(i)}$ | The true value of the independent variable at sample *i* |
| $\hat{y}_t$ | Estinamted *y* at node *t* |
| $w_{kj}$ | Weights of neuron *k* at layer *j* |
| $b_k$ | Bias in a neuron calculation |

| | |
|---|---|
| $u_k$ | Sum of input in a neuron by input variables |
| $\varphi$ | Activation function |
| $y_k$ | Output signal |
| $\varepsilon$ | The maximum error (or Margin) |
| $\xi$ | Deviation from margin |
| $J$ | Cos function |
| $\lambda$ | Hyperparameter for tuning the model |
| $r_P$ | The target return of the portfolio |
| $r_f$ | The risk-free rate |
| $r_p$ | Expected return |
| $\sigma_p$ | The standard deviation of the portfolio |

# IV List of figures

# V List of tables

# 1 Introduction

The world's real estate has witnessed considerable changes during recent years. The annual report by MSCI for the year 2019 has shown that the size of the professionally managed real estate investment market increased from $8.9 trillion from the previous year to $9.6 trillion.

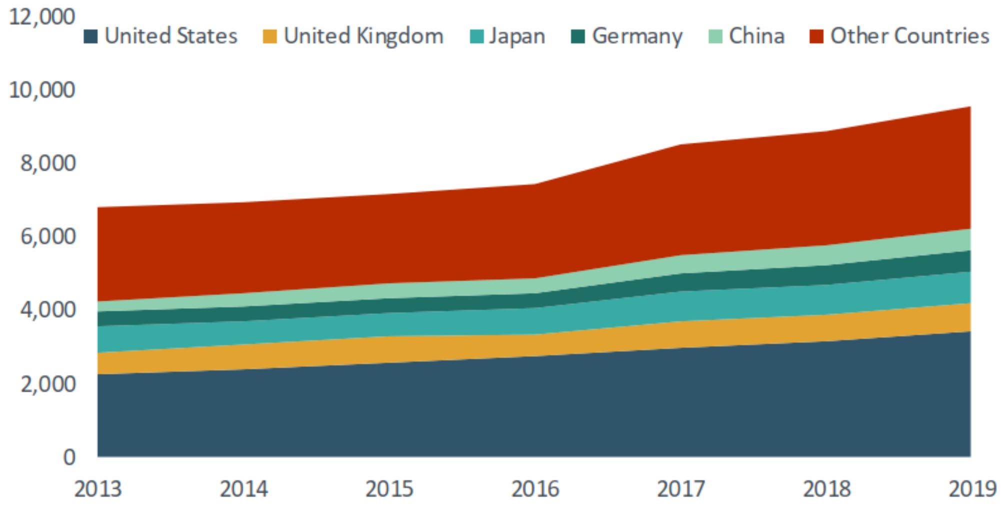


*Figure 1: Change in global market size estimate and for 5 largest countries, USD billion. (Teuben, 2020)*

As illustrated in Figure 1, the global market of professional managed real estate investment saw gradual growth during the period, in every leading country. If non-professionally managed real estate is taken into account, the numbers would be much larger. More precise valuation models powered by up-to-date data could bring more asymmetric material information, that ultimately helps both buyers and sellers, as well as investors in their decision-making process. As a result, the liquidity of the market as a whole would be advanced and the percentage of the global real estate market which is professionally managed would also be increased.

There are numerous factors, that drive the prices as well as the development of the real estate market. Among them, the most known are demographics, interest rate, the economy, and the government's policies. Technology also contributes to the development of the real estate industry. New development in technology changes the way buildings, houses are designed, built, and sold.

When it comes to real estate valuation, there are three most common approaches: Comparable sales approach, Capitalization of income approach, and Replace cost new value approach. Capitalization of income approach values a real estate asset based on expected discounted cash flow from rents and the resale of the asset. Replace cost new value approach estimates the current cost of constructing a comparable property using current material, regarding current design standards. This thesis is focusing on Comparable sales approach that values a real estate based on a comparable one that has been sold recently. The characteristics could be its size, location, number of bedrooms, security, and other features that can affect the value of a property as a whole (Internal Revenue Service, 2020). By using data of actual transactions to build and to compare multiple Machine-Learning (ML) regression models, the best fit model would be chosen and then deploy to build a web-based application for production. With the help of the ML regression models, we could answer what-if questions by just changing the input information of the asset. This flexibility could not be found in the two other above-mentioned approaches.

For further prospects, ML expected to shed new light on real estate valuation. However, well structured and up-to-date real estate data is still a luxury. Blockchain technology, with the help of smart contracts, promises to provide robust real-time data that might fuel more precise ML models for determinants of real estate prices (Mihnea, 2019).

# 2 Theoretical framework

It is not overestimated to state that regression is one of the most important and the most used methods in scientific research. In regression models, a continuous target variable is predicted by using one or various independent variables. In ML, regression is categorized as a supervised technique, which tries to learn labeled training data and establish a model, and then the model will be used to map unseen or future data. Regression is widely used in industry as well because of its application, such as giving insight to relations between variables, evaluating trends, or making forecasts (Sebastian, Vahid (2017, p. 41)).

To differentiate regression techniques, data scientists consider the number of independent variables used and the nature of the relationship between a dependent variable and independent variables (Rohith, 2018). In the following section, some of the most well-known regression techniques would be presented before being used on the data set for comparisons.

## 2.1 Linear regression

Linear regression attempts to establish a linear relationship between one dependent variable and one or more independent variables. A linear regression model that contains only one independent variable ( also known as an explanatory variable) is called Simple linear regression. The model that the dependent variable is explained by more than one independent variable is called Multiple linear regression (Jason, 2020).

### 2.1.1 Simple linear regression

As stated before, Simple linear regression aims to model the relationship between a single feature (independent variable) with a continuous response (dependent variable). The mathematic formula of Simple linear regression is defined as follow:

$$y = w_0 + w_1 x$$

In the formula, the weight $w_0$ represents the y-axis intercept and $w_1$ is the weight coefficient of the independent variable x. The regression process tries to determine the intercept and coefficient by training the model with data. Then the model could be used to predict the values of y given new values of explanatory variable x, which might not be included in the training data.

If the data points are posted on the two-dimension graph, the regression could be understood as finding the best-fitting straight line (regression line) for the given data points (Sebastian, Vahid (2017, p. 450-451)). The distances between the training data points and the corresponding points in the regression line are the errors. The training process attempts to minimize the errors, thus minimizing the cost function. Mean squared error (MSE) is the most commonly used cost function. MSE is calculated by averaging the squared differences between the estimated values and the actual value.

$$Minimize\, \frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

With each value set of coefficient, the value of the cost function is calculated for comparison. Gradient descent (GD) approach is commonly used to initiate and update the coefficient set. In a nutshell, it will generate a random set of coefficients, compute the gradient cost function of these values, update and recalculate the cots function. The process ends when a minimum value of cost function is found (Apoorva, 2018).

### 2.1.2 Multiple linear regression

It is often that the target variable's value depends not only on one independent variable but on various features. In these cases, Multiple linear regression is deployed. The formula for Multiple linear regression could be obtained by generalized the previous formula of  Simple linear regression.

$$y = w_0x_0 + w_1x_1 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i = w^Tx$$

In this formula, $w_0$ is y-axis intercept, $x_0$ = 1. The number of independent variables m is also the number of dimensions of the model. The methods for determinants of coefficients as well as the cost function calculation are the same as of Simple linear regression (Sebastian, Vahid (2017, p. 452)).

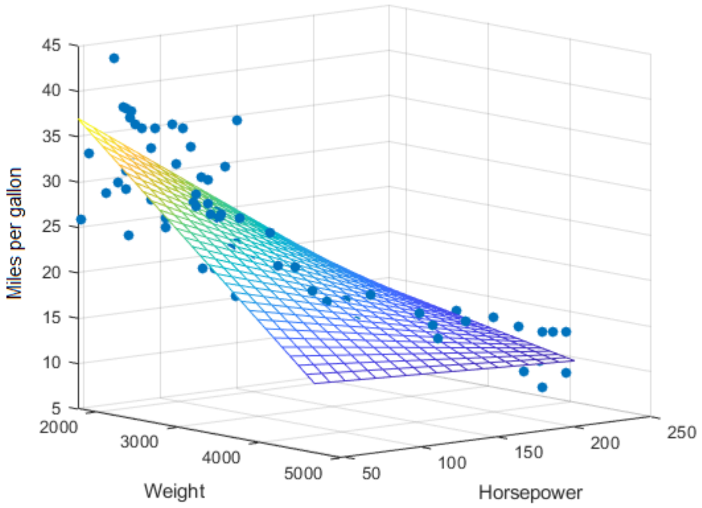


*Figure 2: Example of Multiple linear regression (3 dimensions) (Mathworks, 2020)*

The performance of a model on training dataset could be increased by increasing its complexity, in this case, is the number of dimensions. However, the performance of a model on an independent test set could be better at first and then get worse when we increase its complexity. This is well-known in ML as “overfitting” and “underfitting”, the terms would be used throughout the remaining sections. Overfitting happens when a model is too complex so that it performs well on training data but does not generalize well on test data.  Opposite with overfitting is underfitting, which happens when the model is too simple that could not adequately capture the pattern on the training set, as a result of having a low performance on test data (Sebastian, Vahid (2017, p. 134)).

### 2.1.3 Polynomial regression

In the previous section, the relationship between the dependent and independent variables is assumed to be linear. In case this assumption does not hold, as often happens in real life, the linear regression model will not portrait well the pattern of training data.  To take the non-linear effect into account, polynomial terms are added. The general formula for Polynomial regression is as follow:

$$y = w_0 + w_1 x + w_2 x^2 + \cdots + w_d x^d$$

In the formula, *d* denotes the degree of the polynomial regression.

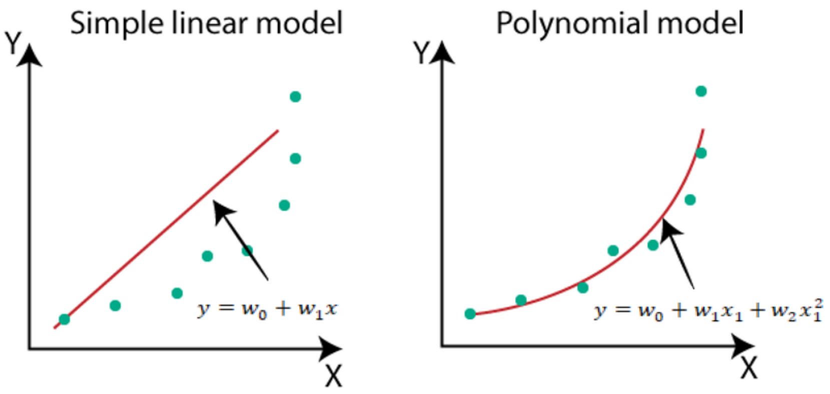


*Figure 3: Simple and Polynomial regression (Pratik, 2020)*

In practice, after making polynomial transformation on training data, a linear regression model is employed on the transformed data. In that sense, although the non-linear feature is added, Polynomial regression is still considered a Multiple linear regression model (Sebastian, Vahid (2017, p. 483)).

## 2.2 Random forest regression

A Random forest contains multiple decision trees that are capable of performing both classification and regression. This method employs a technique called Bootstrap Aggregation, which samples different datasets from the original dataset with replacement and trains decision trees on those samples (Krishni, 2018). In the previous sections, the linear regression methods attempt to establish a global linear relationship between independent variables and a dependent variable that best fits the training dataset. In contrast, a Random forest is considered some of piecewise linear functions. That means it divides sample space into smaller manageable regions. The determinant of output is based on the aggregation of multiple decision trees, not on a single one (Sebastian, Vahid (2017, p. 490)).

### 2.2.1 Decision tree regression

To have a better understanding of Random forest regression, it is necessary to take a look at how a decision tree is constructed. Forming a decision tree requires splitting its nodes until the leaves are pure or stopping criterion is satisfied. Entropy is defined as a measure to identify which attribute should be used to split that maximizes the Information Gain (IG). The formula for IG is as follow:

$$IG(D_P, x_i) = I(D_P) - \frac{N_{left}}{N_P} I\big(D_{left}\big) - \frac{N_{right}}{N_P} I\big(D_{right}\big)$$

In the formula, $x$ is the feature by which the current node is split. $N_P$ is the number of samples in the previous node, $I$ is the impurity function, $D_P$ is the subset of training samples at the previous node, $D_{left}$ and $D_{right}$ are the subsets of training samples at the left and the right node after the split. The best split is the one that maximizes IG or minimizes the impurity of the nodes after the split. For the regression measurement of continuous variables, we need an impurity calculation as the errors of estimations. The most popular, as stated earlier, is MSE.

$$I_{(t)} = MSE_{(t)} = \frac{1}{N_t} \sum_{i \in D_t} \left(y^{(i)} - \hat{y}_t\right)^2$$

The number of training sample at node *t* is denoted as $N_t$, while $D_t$ is the training subset. The true value of the dependent variable is $y^{(i)}$, the estimated one is $\hat{y}_t$, which is the sample mean:

$$\hat{y}_t = \frac{1}{N_t} \sum_{i \in D_t} y^{(i)}$$

In other words, the splitting attribute is selected in a tendency to reduce the variance of child nodes compare to the parent node as much as possible (Sebastian, Vahid (2017, p. 492)).

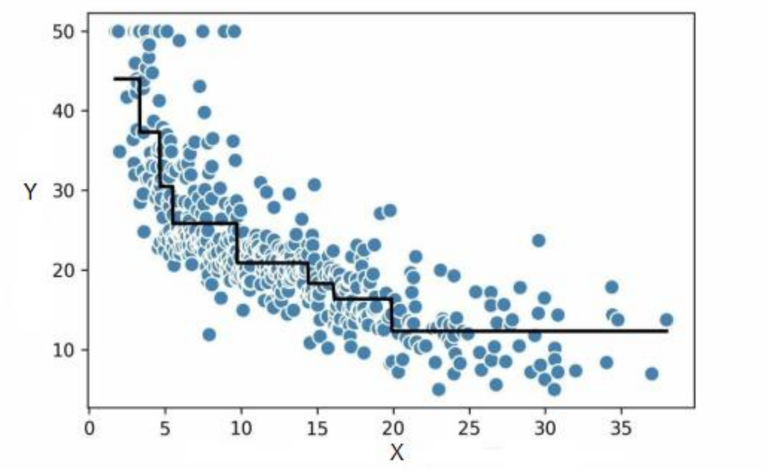


*Figure 4: Decision tree regression (Sebastian, Vahid (2017, p. 494))*

If the splits continue until the leaves are pure, it is often that the model will be overfitting. That’s why stopping conditions are needed. One popular condition is the depth of the tree.

There has been no systematic way to choose the optimal depth of a decision tree. In practice, comparing the training score and test score could be help full to find a reasonable depth for a tree.

Another solution to improve the quality of a decision tree after it is built is to perform a tree-pruning step, that could help reduce its size. Pruning trims the branches in a way that eventually increases the generalization capability of the decision tree (Pang, et al., 2013).

### 2.2.2 Random forest regression

As above mentioned, that combination of decision trees forms a random forest. As each subset is randomly sampled, a random forest is less overfitting than a decision tree since the variation is reduced. Although achieving great improvement compare to decision tree regression, random forest regression still suffers from higher overfitting compare to other regression methods. However, it enjoys the advantage of less sensitivity to outlier and less requirement of parameter tuning. When using random forest regression, only the number of random trees is required (Sebastian, Vahid (2017, p. 495)).

## 2.3 Neural Network Regression

Inspired by how the human brain works, Artificial neural networks (ANNs) are designed by mimicking the functioning of our brain to solve complex problems (Ata, 2015). It can be said that artificial neural networks have been one of the hottest areas in academic research as well as in applications by big tech enterprises (Sebastian, Vahid (2017, p. 542)). ANNs are a composition of units, which are called neurons. Each neuron could make simple mathematical decisions. Together, they could solve a very complex problem (Missinglink.ai, 2020). Haykin, in his book "Neural Networks: A Comprehensive Foundation.", presented formulation for a neuron $k$ as:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

$$y_k = \varphi(u_k + b_k)$$

Here, $w_{kj}$ denotes the weights of neuron $k$, $x_j$ represent input values. Bias, as $b_k$, functions like the coefficient of the y-axis to adjust the input before it goes through the activation function, which is denoted as $\varphi$. The neuron produces its output signal $y_k$ (Haykin, (1999, p.33)).

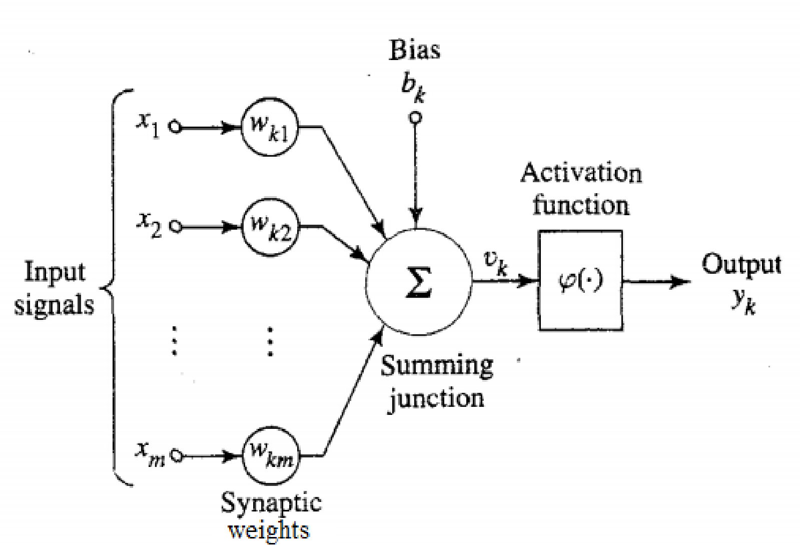


*Figure 5: Nonlinear model of a neuron (Haykin (1999, p. 33))*

ANNs could contain multiple layers, the output of the previous layer will be the input of the current layers. After the signal outputs are produced, the weights are updated to reduce the errors (Erdi, et al., 2016). The most popular method to train ANNs is Backpropagation using the GD approach. ANNs could be used to make predictions as well as classifications, depending on which kind of activation functions are used. There are several types of activation functions namely: Step function, Linear function, Sigmoid function, Tanh function, ReLu function... To make regression, in this thesis the Linear activation function would be used to produce continuous output (Avinash, 2017).

## 2.4 Support Vector Regression

Support Vector Machine (SVM) is widely used in machine learning and is well-known for its classification use. Other machine learning algorithms attempt to minimize classification or regression errors. SVM, in another way, tries to maximize the margin, which is defined as the distance between the separating hyperplane (decision boundary) and the training samples that are closest to this hyperplane. Those closest training samples are called Support vectors. A model with a larger margin would yield a better generalization. In a step further, a slack variable (denoted as ξ ) was introduced by Vladimir Vapnik in 1995, which gave flexibility for the model (Sebastian, Vahid (2017, p. 138 -143)). The objective function is as follow:

$$Minimize: \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}|\xi_i|$$

$$Constraints: |y_i - w_i x_i| \leq \varepsilon + |\xi_{\,i}|$$

Here, w is the weights vector, $\varepsilon$ is the maximum error or the margin. Every point that falls out of the margin, which has an error larger than $\varepsilon$, would have ξ as its deviation from the margin. By tuning the hyperparameter C, we could adjust how an outside point is punished. If C is increased, the model is more tolerant with the points outside the margin and vice versa. In practice, a Grid search could be performed to find C so that the model would produce a line that better fit training data (Tom, 2020).

## 2.5 Random Sample Consensus algorithm

Some ML techniques are prone to outliers, such as Linear regression. A small subset of outliers could greatly impact regression results. In this section, a robust method of regression is introduced to help eliminate outliers, which is called the Random Sample Consensus (RANSAC) algorithm. RANSAC algorithm first trains the model on a group of randomly selected data points, which are called inliers. Then all other data points would be used to test the trained model. A tolerance is set by the user to identify which data points are considered inliers. The model then is trained on all inliers and the errors are estimated based on the original inliers. If the algorithm does not meet the conditions to terminate, it would pick another random sample to be inliers, and the process repeats. RANSAC algorithm terminates when the performance meets the user's objective threshold or it reaches a certain number of iterations.

As stated before, the RANSAC algorithm helps to reduce the effect of outliers on a regression model. One disadvantage of this algorithm is that users have to choose an appropriate performance threshold for stopping. It could be problematic difficult (Sebastian, Vahid (2017, p. 473-476)).

## 2.6 Regularized methods

A commonly used approach to remedy the affection of overfitting is regularization. The method works by adding additional information to a model. The aim is to shrink the parameters of the model by penalizing its complexity. For linear regression, the most popular regularization approaches are Least Absolute Shrinkage and Selection Operator (LASSO), Ridge Regression, and Elastic Net (Sebastian, Vahid (2017, p. 481)).

### 2.6.1 Ridge regression

In addition to the cost function of normal linear regression, the squared sum of the weights is added to be penalized. The aim is to shink the weights of the model to increase its generalization. The squared sum of the weights is called L2 penalty.

$$J(w)_{Ridge} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda\|w\|_2^2$$

$$With : L2 \ = \lambda\|w\|_2^2 = \lambda\sum_{j=1}^{m} w_j^2$$

By increasing or decreasing the hyperparameter $\lambda$, users could strengthen or weaken the model's regularization (Sebastian, Vahid (2017, p. 481)).

### 2.6.2. LASSO regression

The below figure is the comparison of how Lasso and Ridge's regularization works.

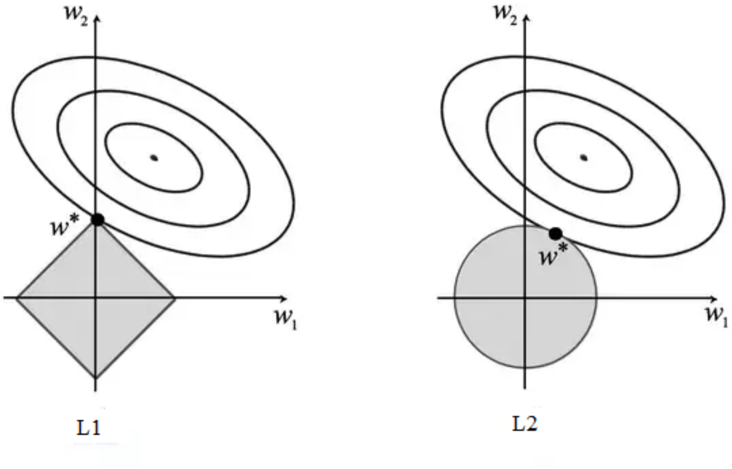


*Figure 6: Contours of the error and constrain functions for the Lasso (L1) and Ridge (L2) regression (Vijay, 2019).*

LASSO regression is a spare model compare to Ridge regression. In this model, certain weights could be shrunken to zero due to regularization strength. Because of this advantage, LASSO could be used as a supervised technique for feature selection (Sebastian, Vahid (2017, p. 482)).

$$J(w)_{LASSO} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda\|w\|_1$$

$$With\!: L1 = \lambda\|w\|_1 = \ \lambda\sum_{i=1}^{m}\left|w_j\right|$$

### 2.6.3 Elastic net regression

A solution in between of LASSO and Ridge regression is Elastic net. This regression technique has both L1 and L2 penalty. So it could bring sparsity to the model, meanwhile helps to deal with some problems of LASSO, for example, the number of selected variables.

$$J(w)_{ElasticNet} = \sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2 + \lambda_1 \sum_{j=1}^{m} w_j^2 + \lambda_2 \sum_{i=1}^{m}\left|w_j\right|$$

In practice, the ratio of L1/L2 will be adjusted, so that Elastic net could become LASSO, Ridge, or a compromised model in between (Sebastian, Vahid (2017, p. 482- 483)).

## 2.7 Boosting Algorithms

Boosting is a technique that converts multiple weak learners into a more robust rule (Sunil, 2015). Boosting algorithms are often outperform many other peers. In this section, some of the boosting algorithms that have gained their popularity are presented.

### 2.7.1 Adaptive Boosting Algorithms

Adaptive Boosting (AdaBoost) attempts to train a chain of weak learners on different training data with different weights. AdaBoost first makes predictions on original dataset and gives equal weights for each prediction. If the estimated value is close (or equal) to the real value by using the first learner, the algorithm would then give higher weights to estimations that are far from the target variable's real values. The process continues until it reaches a predetermined accuracy target or a maximum number of iterations. User could adjust the number of weak learners, learning rate, and base estimators to gain better performance (Sunil, 2015).

### 2.7.2 Gradient Boosting Algorithms

Gradient Boosting fits various models in order. The new models themself, through the GD method, minimize the overall loss function. The purpose of the algorithm is to minimize the cost function of the whole ensemble through building " new base learners which can be maximally correlated with negative gradient of the loss function, associated with the whole ensemble " (Sunil, 2015). The steps below are how Gradient Boosting works in brief:

*Step 1*: The mean is considered to be the prediction of all variables.

*Step 2*: Determine errors of each observation (deviation from the mean of the latest prediction).

*Step 3*: Find the attribute that can split the errors perfectly and calculate the value for the split. This prediction is considered to be the latest one.

*Step 4*: Determine errors of each observation as in step 2, but for both sides of the split.

*Step 5*: Repeat step 3 and 4 until the accuracy is maximized (or the cost function is minimized).

*Step 6*: The final model is established by taking the weighted average of all the classifiers (Tavish, 2015).

### 2.7.3 Extreme Gradient Boost Algorithms

Extreme Gradient Boost Algorithms (XGBoost) is an ensemble learning technique that uses a GD boosting framework. However, through extreme optimization and mathematical strengthening, XGBoost obtains significant improvement compare to GD boosting framework in some cases. Some system optimization is parallelized implementation for sequential tree building, tree pruning using 'max_depth' parameter, hardware optimization by allocating internal buffers to store gradient statistics for cache awareness. Some algorithmic enhancements are: using both LASSO and Ridge for regularization**,** learning the best missing value of sparse features for inputs, deploying "the distributed weighted Quantile Sketch algorithm to effectively find the optimal split points", adding " built-in cross-validation method at each iteration " (Vishal, 2019).

## 2.8 Regression modeling

From the diagram below in Figure 7, how a typical predictive learning model works are showed. Then some of the important steps, that are used in this thesis, should be presented in more detail in later sub-sections.

The first phase in the process is Data processing, which attempts to get data into shape for deploying to ML models. Raw data is often collected in a form that is rarely optimized to use as immediate input for models. Datasets might contain missing values or a large number of outliers. Data cleansing steps are needed to remove extreme outliers and to replace (normally with average values) or remove missing values. Then, by using various techniques, such as standardization, normalization, dimensional reduction techniques are deployed to bring data into the same scale and reducing the needed-computational effort. That makes the model more precise and reduces running time. To be confident that ML models would perform well on new data, the dataset is randomly split into a training set and a test set. Those two sets are used in the next phases to train and to evaluate the models. After the valuation, the final model is determined and then used for predictions (Sebastian, Vahid (2017, p. 54 - 57)).

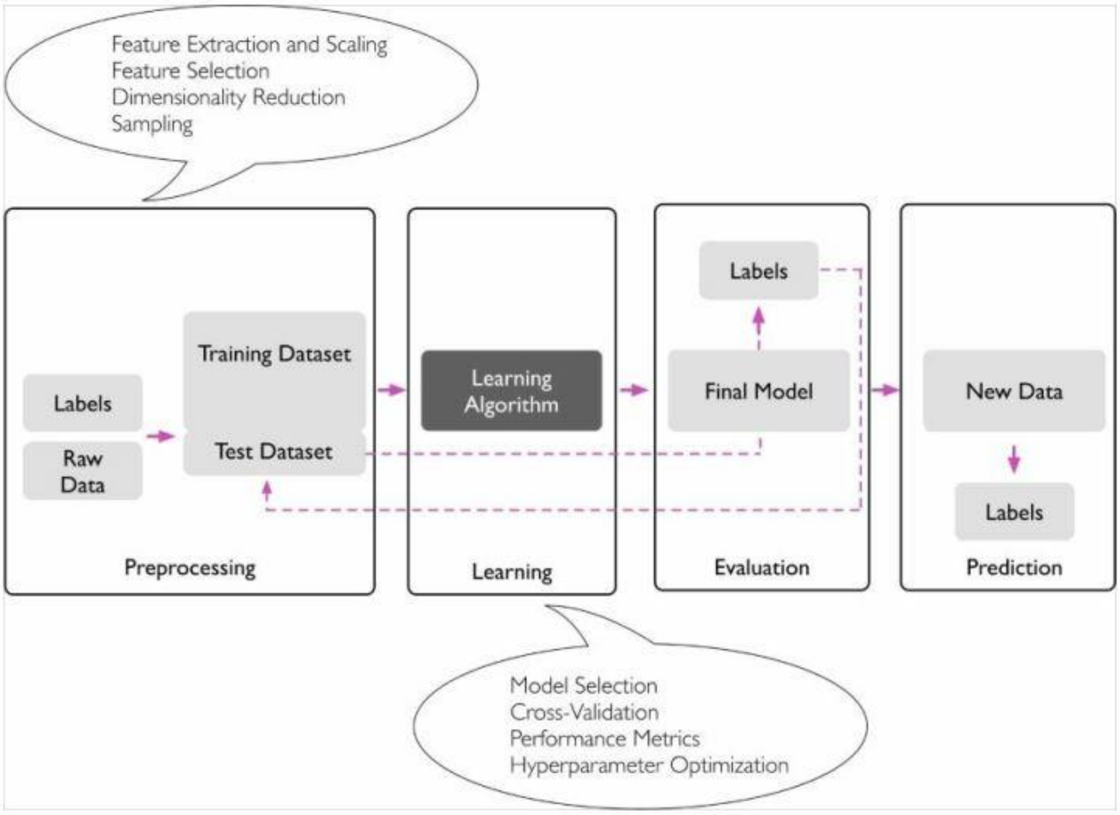


*Figure 7: Process of a predictive Machine learning model (Sebastian, Vahid (2017, p. 54))*

### 2.8.1 Train-test split

The dataset will be split into a training dataset and a testing dataset. It should be noted that the split should be performed before feature scaling. Otherwise, when the dataset is standardized, it will learn the sample mean ($\mu$) and standard variation ($\sigma$) of the whole dataset. As a result, the model is fit using partly information from the test set. This eventually harms the testing accuracy.

### 2.8.2 Feature scaling

In the above-mentioned regression techniques, there are many techniques benefitted from feature scaling. In this thesis, standardization method would be used to gives our data the property of a standard normal distribution. That means every feature is zero-centered and has a standard deviation of one. The formula to standardize a feature is as follow:

$$x_j' = \frac{x_j - \mu_j}{\sigma_j}$$

Here $\mu_j$ is the sample mean of feature $j$th and $\sigma_j$ is its standardiviation. $x_j$ is the vector of all feature $j$th's $n$ training values.

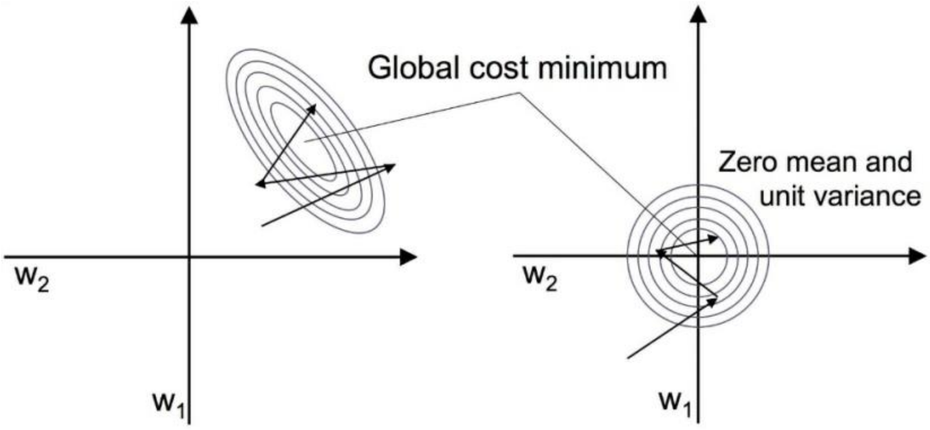


*Figure 8: The optimization process of normal data and standardized data (Sebastian, Vahid (2017, p.98))*

Once the dataset is standardized, the optimizer through GD would find the optimal cost function value faster because there are now fewer steps to run through. Standardization helps to preserve useful information on outliers while reducing the negative effect caused by them.

After having the regression prediction prices, "`inverse_transform`" function should be used to transform the result into the original scale in dollars (Sebastian, Vahid (2017, p. 97 - 98)).

### 2.8.3 Performance metrics

In this section, some performance metrics are presented. They would be used to determine which techniques perform the best on the given dataset.

#### 2.8.3.1 Mean squared error

Training a model on a dataset requires a Loss function to measure how good the model at fitting the data. In that sense, choosing the appropriate Loss function is crucial in data mining in general and in regression to be specific. As stated before, MSE is the most used measurement of a regression model's performance. It tackles the disadvantages of Sum of Squared Errors (SSE), which will increase while raising the size of the dataset. To calculate MSE, we simply take the average of SSE (Chayan, 2019):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

### 2.8.3.2 Coefficient of determination

Coefficient of determination ($R^2$) is an important performance measure for a regression model. $R^2$ represent the proportion of the dependent variable's variation that is explained by the regression model.

$$R^2 = 1 - \frac{SSE}{SST}$$

SSE as defined earlier is Sum of Squared Errors, which is understood as the variation dependent variable that is unexplained by the model. SST is the total variation of the dependent variable, the total sum of squares.

$$SSE = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$$

$$SST = \sum_{i=1}^{n} \left(y_i - \mu_y\right)^2$$

Here $\mu_y$ is the average of the actual values of the independent variable. The value of r-squared for a training model is between 0 and 1. For testing, r-squared could be negative. R-squared preferable value is close to 1 (Sebastian, Vahid (2017, p. 478- 480)).

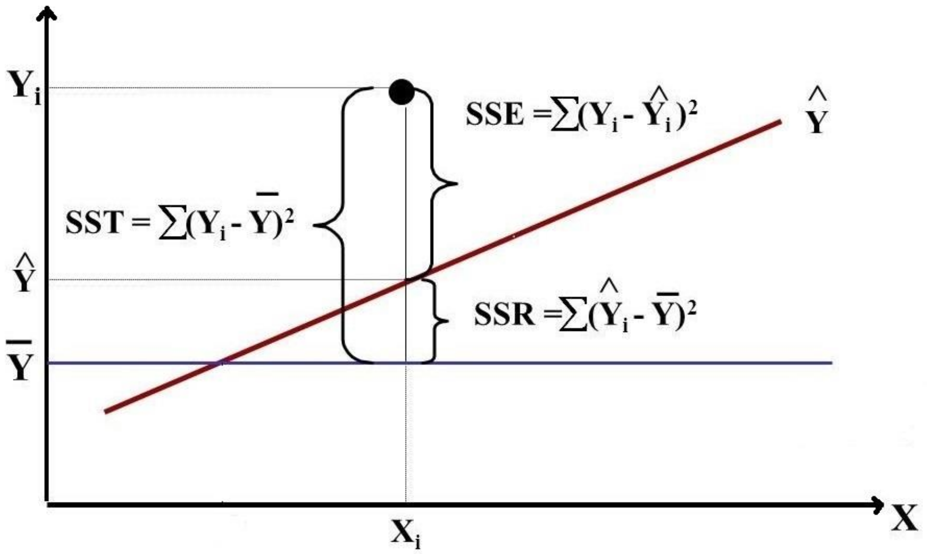

$Y_i$
SSE =∑(Y$_i$ - $\hat{Y}_i$)²
$\hat{Y}$
SST = ∑(Y$_i$ - $\bar{Y}$)²
$\hat{Y}$
SSR =∑($\hat{Y}_i$ - $\bar{Y}$)²
$\bar{Y}$
$X_i$
X

*Figure 9: Coefficient of determination's components (Baranidharan, 2019)*

To make the measurement more helpful, the adjusted r-squared would be used to penalize the complexity of the model. It would take into account the explanatory variables that are added to the model but are not useful for explaining the target variable.

$$R_{adj}^2 = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

Here, n is the number of samples, k is the number of attributes of the models. If attributes (explanatory variables) are added to the model but there is no significant change in $R^2$ , then $R^2_{adj}$ will decrease (Aishwarya, 2019).

### 2.8.3.3 Cross-validation score

Cross-validation (CV) is often used to measure how a machine learning model performs on a new dataset. Especially when the training data is limited, CV could help re-sampling the dataset. One popular approach is to split the dataset into a training and a testing set. The model will be fit and evaluate on the training set, tested on the test set (Hold-out method). The common practice is 70% for training, 30% for testing. This approach might be biased if the distribution of the training and testing set is not similar, which usually happens when the sample is small. Another approach is K-folds cross-validation.

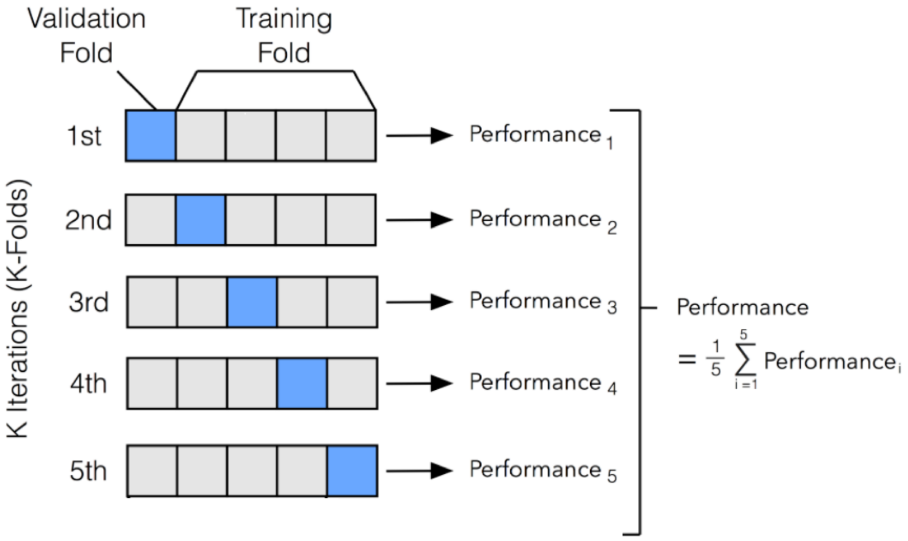


*Figure 10: K-folds cross-validation (Ethen, 2020)*

The dataset first is split into k folds (very common k=10), the k-1 folds are used to train the model and the remaining is used for testing. Repeat the process k times so that each of the k folds is used for testing exactly one time. The K-folds CV score is obtained by taking the average of the k test scores. This process helps to reduce the biased compare to single data split only (Sanjay, 2018).

### 2.8.4 Tuning the models

The above-mentioned regression algorithms are used to train and to test on datasets. The models' hyperparameters are then tuned so that they could yield the best performances. The results, which are calculated as performance metrics, are put in comparison. The best model would be chosen to build up the web application.

One popular approach for tuning machine learning models is Grid-search, which could find an optimal combination of hyperparameter values through a “brute-force exhaustive search paradigm”. In this paradigm, a list of values for different hyperparameters is specified. The computer would fit the model with each combination of hyperparameter values and find the optimal one. Although Grid-search could yield the optimal set of parameters, it is computationally expensive. An alternative approach is Randomized-search, which draws parameter combinations from sampling distribution with a specified budget (Sebastian, Vahid (2017, p. 314)). In this thesis, Randomized-search is used to tune the models.

### 2.9 Modern Portfolio Theory and Sharpe ratio.

After using the ML regression models to evaluate a portfolio of real estate assets, we then combine the real estate portfolio (as an alternative asset) with other publicly traded assets (for example stocks, ETFs, bonds, currencies) to form a diversified portfolio. So it is necessary to briefly introduce the Modern Portfolio Theory (MPT) and Sharpe ratio.

#### 2.9.1 Modern Portfolio Theory

Harry Markowitz was the first one to inspire MTP in his paper "Portfolio Selection" in 1952. MPT focuses on how a risk-averse investor should construct a portfolio, which maximizes the expected return given a level of market risk. MPT could also be used to minimize the risk of a portfolio given a level of expected return. The optimization problem is as follow:

$$\min_{w} \quad w^T \Sigma w$$

$$\text{st: } w^T r = r_P$$

$$w^T 1 = 100\%$$

Here *w* denotes the vector of weights, *Σ* is the covariance matrix, *r* the vector of expected rates of return, $r_P$ is the target return of the portfolio, and *1* is a vector of ones. The efficient frontier graphically portraits portfolios that maximize return given levels of risk or minimize risk given a level of return. The measure of risk is the standard deviation of individual securities. The optimal portfolio, that lies on the efficient frontier, is the portfolio that has a balance between risk and return. The risk-seeking investors decide to invest in the right end of the frontier while the risk-averse investors tend to invest in the left end of the curve.

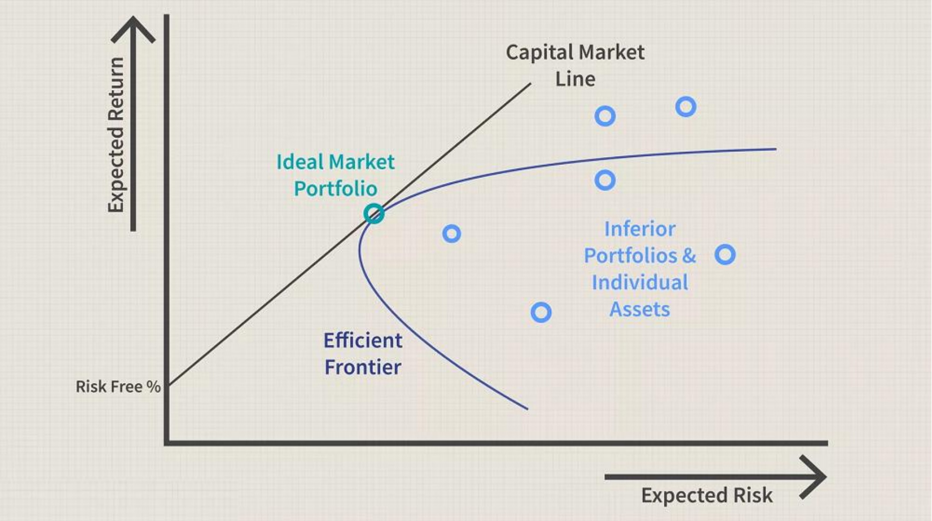


*Figure 11: Efficient frontier (Will, 2020)*

Some critics of the theory that its assumption might not properly represent reality. For example, the distributions of asset returns are normal while in reality, some assets might have a leptokurtic distribution or heavy-tailed distribution (James, 2020).

### 2.9.2 Sharpe ratio

The Sharpe ratio has been long used by investors to assess the risk-adjusted return. To calculate the Sharpe ration, we subtract the risk-free rate $(r_f)$ from return (or expected return) of the portfolio $(r_p)$ and divided by the standard deviation of the portfolio excess return $(\sigma_p)$.

$$Sharpe\ Ratio = \frac{r_p - r_f}{\sigma_p}$$

Investors prefer a higher Sharpe ratio portfolio to a lower one. In plain words, one portfolio with a higher return and higher risk is not always better than a portfolio with a lower return and lower risk. The Sharpe ratio helps to determine which portfolio performs better given a unit of extra risk taken (Marshall, 2020).

## 3 Dataset exploration.

In this section, a close look at our dataset will be presented. Otherwise, some important questions should be raised, so that we could determine how to manipulate the data and design regression models for the next section. These tasks are very important because a researcher can squeeze the most out of a data set only if he has a comprehensive understanding of it.

### 3.1 Overview of the dataset.

Our dataset contains 21,613 actual house sales transactions of King County, Washington, The United States between May 2014 and May 2015. Each transaction (item) includes the details of 19 attributes, along with its price and Id. There is no missing value in the whole dataset. The attributes (features) are described as in Table 1 (Abdallah, et al., 2017):

| Attributes | Descriptions |
|---|---|
| Id | Unique ID for each transaction |
| Date | The date when the transaction was made |
| Price | The price of the house at the transaction |
| Bedrooms | Number of bedrooms |
| Bathrooms | Number of bathrooms, where .5 account for a room with a toilet but no shower |
| Sqft_living | Square footage of the apartments interior living space |
| Sqft_lot | Square footage of the land space |
| Floors | Number of floors |
| Waterfront | A dummy variable for whether the apartment has a front view of a water area or not. |
| View | An index from 0 to 4 of how good the view of the property was |
| Condition | An index from 1 to 5 on the condition of the apartment |
| Grade | An index from 1 to 13, at which 1-3 describes a poor construction and design condition, 7 has the average quality and 11-13 have a high quality. |
| Sqft_above | The square footage of the interior housing space that is above ground level |
| Sqft_basement | The square footage of the interior housing space that is below ground level |
| Yr_built | The year when the house was initially built |
| Yr_renovated | The year of the house’s last renovation |
| Zipcode | The zip code area where the house is located |
| Lat | Lattitude |
| Long | Longitude |
| Sqft_living15 | The square footage of interior housing living space for the nearest 15 neighbors |
| Sqft_lot15 | The square footage of land lots for the nearest 15 neighbors |

*Table 1: Overview of the dataset attributes (Abdallah, et al., 2017)*

### 3.2. Attributes exploration.

The relationship between each attribute and the target variable (Price) should be analyzed elaborately. The correlation matrix which contains the correlation coefficient of various attributes is presented below. We pay special attention to the correlation between the target attributes and others. Because some features are in nominal scales, to make use of these variables, four new attributes are introduced: “zipcode_up”, “sales_time_up”, “age”, “age_rnv”. The detail of these new attributes will be presented in section 4.1 Feature transformation.

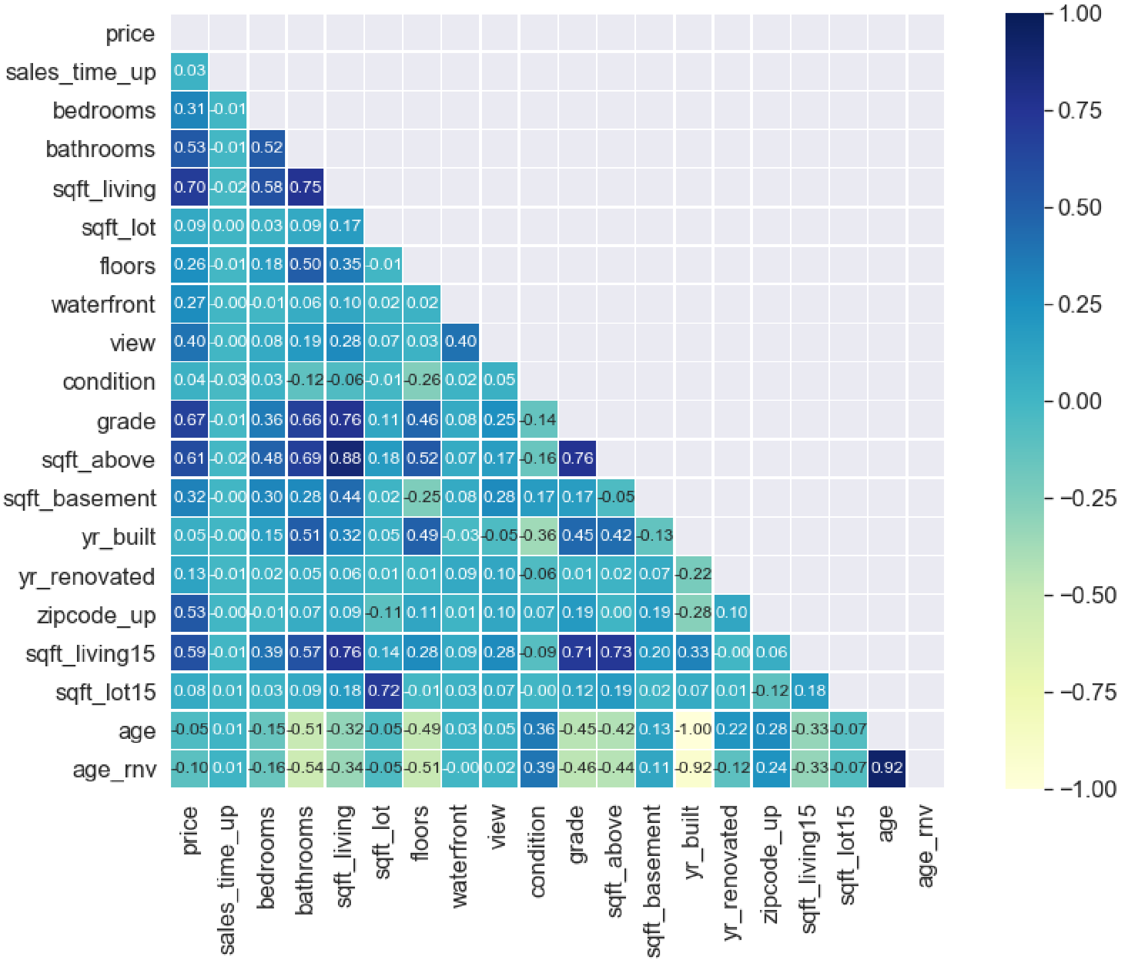


*Figure 12: Correlation matrix (own illustration)*

As illustrated in Figure 12, Square footage of living, Square footage of living upstairs, Square footage of living of 15 neighbors has a very strong correlation with the target variable - sale price. The correlation coefficients with price are, in turn, 0.7, 0.61, and 0.59. Moreover, they have a strong correlation among themselves (Square footage of living has correlation coefficients with Square footage of living upstairs, Square footage of living of 15 neighbors, in turn, of 0.88 and 0.76) so they should be grouped to make further analysis because one of them might represent the explanatory power of all three variables.

The same situation applies to the Number of bedrooms and the Number of bathrooms, Grade and Condition, and View and Waterfront. The details of how each group of attribute affect price would be presented in the next sections.

### 3.2.1 Knowing the target variable.

The target variable, which our model tries to predict, here is the price of houses. As illustrated in Figure 13 and the detail from the dataset, the number of houses that worth less than 1 million US dollar accounts for more than 80% of the total number of transactions. The minimum price is 75 thousand while the maximum price is 7,7 million dollars. The mean price is 540 thousand and the median is 450 thousand with a standard deviation of 36,7 thousand dollars. The distribution is not a typical normal but a positive skew with a bigger right tail. There is a small percent of houses that worth more than 2 million dollars. They could have exceptional features and could be considered as outliers. But they could contain useful information and for that reason, they would not be eliminated from the data set.

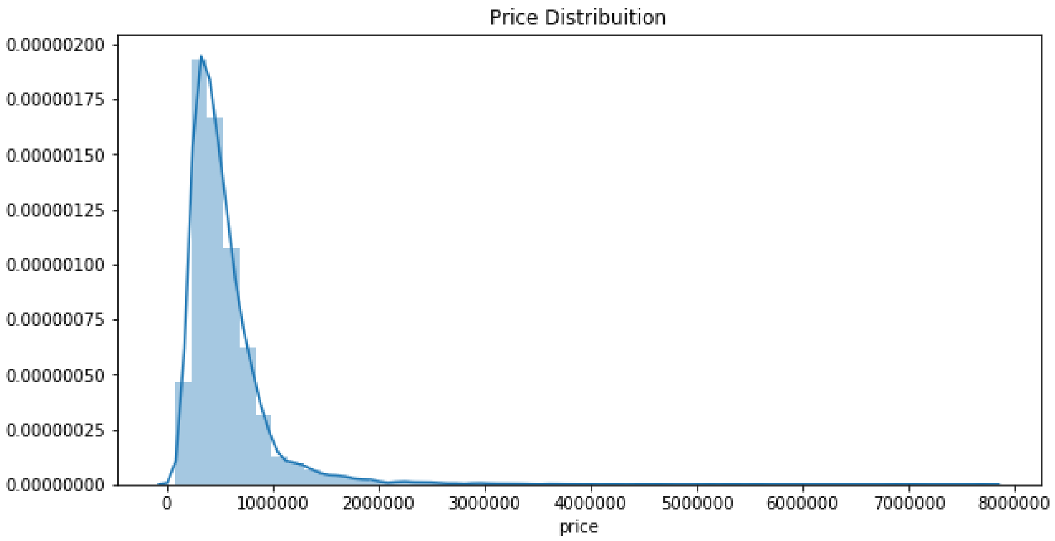


*Figure 13: Price distribution ( recreated referring to (Leonardo, 2018 ))*

### 3.2.2 Square footage of living

The square footage of living is positively correlated with Price, that means a bigger house is supposed to have a higher price compared to a smaller house, assuming that other features are the same. This relationship is also confirmed in Figure 12. The correlation coefficient between Price and sqft_living is 0.7, the highest among all attributes. As shown in Figure 14, more than 95% of houses are under 4000 square footage of living. It is reasonable that some exceptional big houses have the highest values. Some smaller houses could have higher prices due to differences among other aspects. The upper sub-graph shows that the distribution of house square footage of living is a positive skew distribution with a bigger right tail, the same as of the target variable.

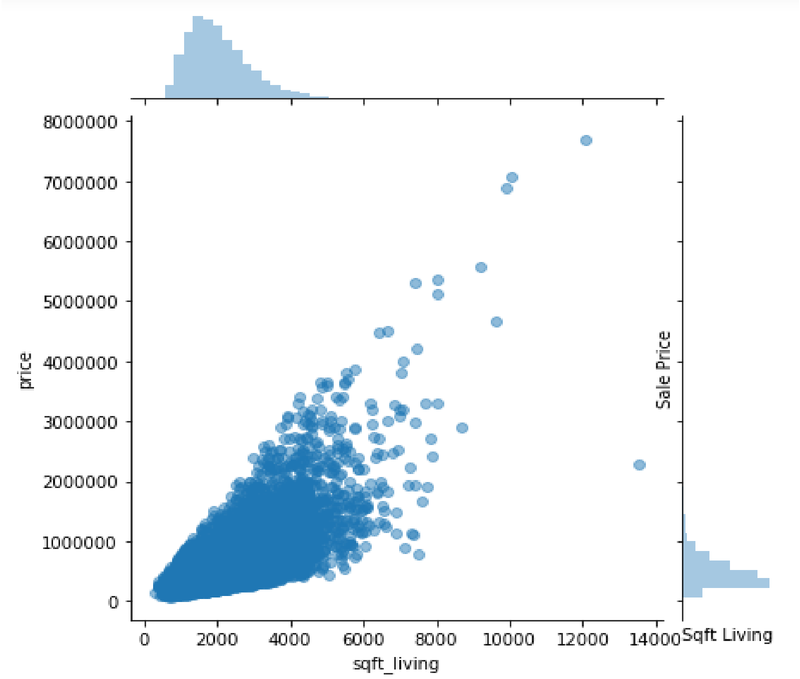


*Figure 14: Price and square footage of living (recreated referring to (Leonardo, 2018 ))*

### 3.2.3 Condition and grade.

While the majority of houses have conditions of three, four, and five, the houses with the condition of three accounts for more than 60% of the total dataset. The condition of houses in this area is average and above average, the houses in low condition for sales are few.

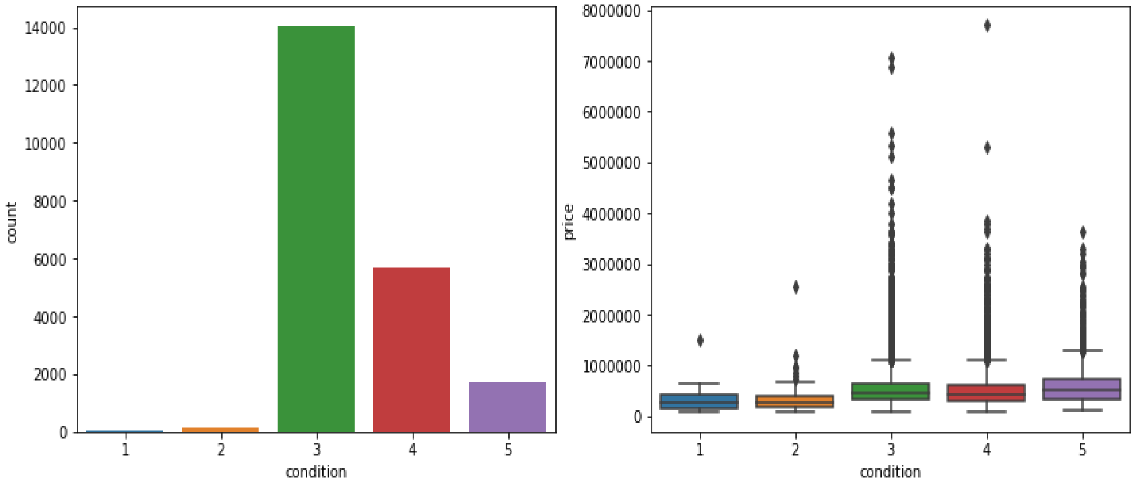


*Figure 15: Condition and Price (recreated referring to (Leonardo, 2018 ))*

Overall, the houses with higher conditions have a higher price compared to the lower condition houses with the same in other features, but the correlation is not strong. Figure 12 shows that the correlation coefficient of condition and price is just 0.04.

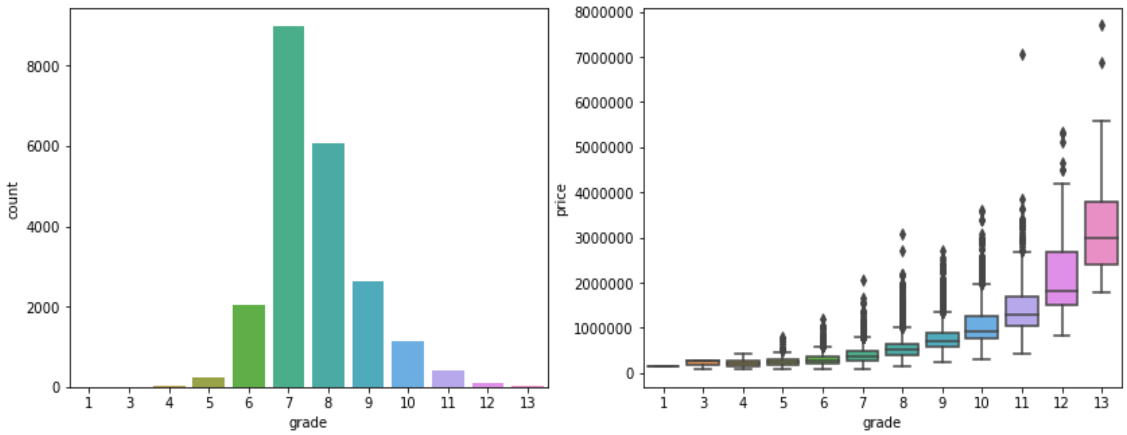


*Figure 16: Grade and Price (recreated referring to (Leonardo, 2018 ))*

When it comes to grade, the situation is different. Grade has a stronger correlation with price as we can see in Figure 16, the price increase significantly in the houses with a higher grade. This is also confirmed in Figure 12. The correlation coefficient of grade and price is 0.67, the second-highest among all attributes. Due to this reason, “grade” should be one of the most important independent variables.

### 3.2.4 Number of bedrooms, bathrooms

As shown in Figure 12, the number of bedrooms and bathrooms positively correlated with the price of a house. The correlation coefficients to prices, in turn, are 0.31 and 0.53. This trend is also can be seen in Figure 17. House price increases significantly and positively with the number of bathrooms.

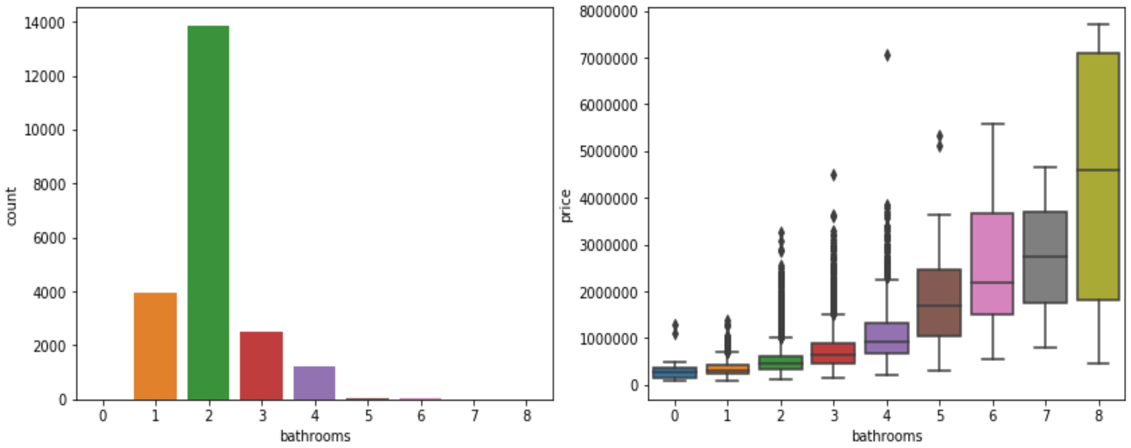


*Figure 17: Number of bathrooms and Price (recreated referring to (Leonardo, 2018 ))*

The trend is the same but does not strong when it comes to the number of bedrooms. Both attributes suffer from numerous outliers, it is explained by various dots outside the Mix-Max range. Some of the outliers are extreme outliers which locate far away from the range. There are a few houses that have 0 or .5 or .75 bathrooms. Those are denoted for bathrooms that lack some equipment or small bathrooms.

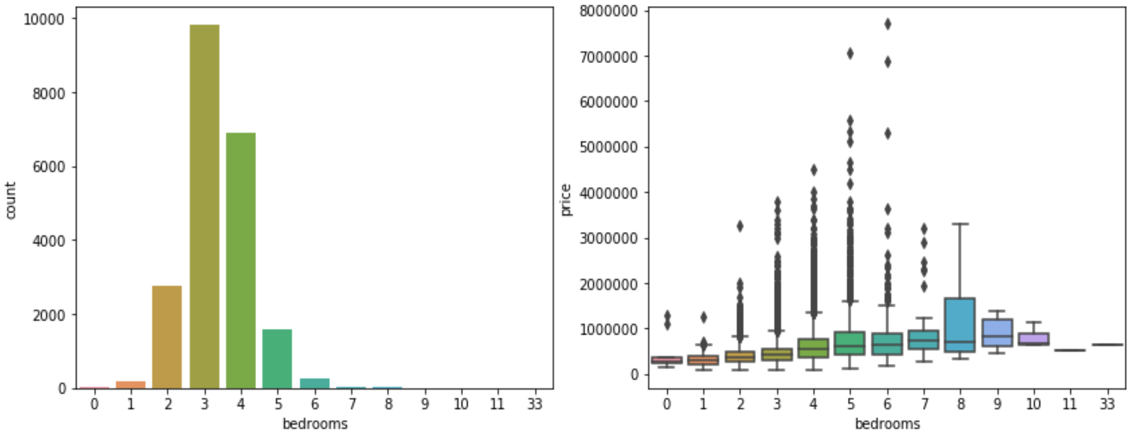


*Figure 18: Number of bedrooms and Price (recreated referring to (Leonardo, 2018 ))*

### 3.2.5 View and number of floors, waterfront

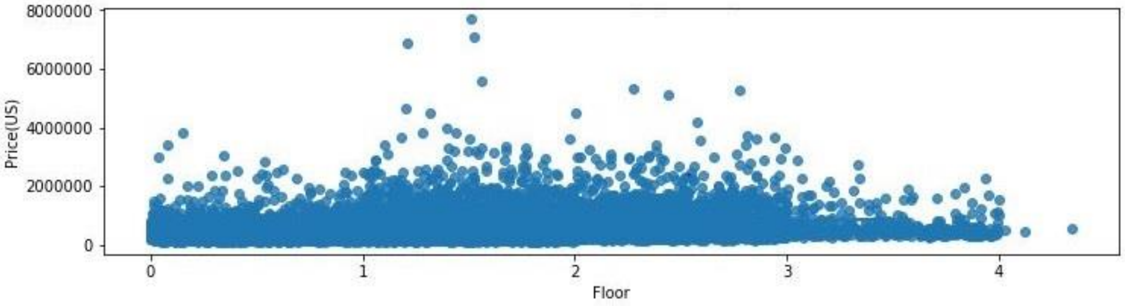


*Figure 19: Number of number of floors and Price (recreated referring to (Leonardo, 2018 ))*

There is a small correlation between the number of floors and house prices as the coefficient is just 0.26. The same trend happened with the waterfront attribute (0.27). View has a stronger correlation with house price (0.4).

### 3.2.6. Year built and Year of last renovation.

As shown in Figure 12, Year built and Year of last renovation have a loose connection with the price of a house. In general, older houses have cheaper prices. The coefficient for “age” (Year sales - Year built) and “age_rnv” (Year sales - Year of last renovation) with Price, in turn, are -0.05 and -0.1.

### 3.2.7 Other attributes.

At this point, we have an overall understanding of the main factors that decide the house price namely: “Sqft_living”, “Grade”, “Bathrooms”, “View”…Between the explanatory attributes, some have strong correlations with each other. As stated before, they're Number of bathrooms with Number of bedrooms, Square footage of living, Square footage above with Square footage of 15 neighbors, Condition and Grade, Waterfront and View.

However, there are two features that we have to make some basic data processing steps to see a clear relationship with the target variable. They are the position ("Zipcode", "Longitude", and "Lattitude") and time of transactions.

# 4. Data manipulation and regression modeling

As mentioned previously, to make use of several important features and turn them into independent variables, we need to make some necessary adjustments. After obtaining all the needed explanatory variables, all regression techniques that are introduced in Section 2: Theoretical framework would be used to build various regression models. The results then will be assessed and compared through the performance metrics. The model with the highest performance will be selected to implement for production through a web application.

## 4.1 Feature transformation

In this section, a house geographical position attribute and the momentum effect at a certain time point will be extracted by transforming original features.

### 4.1.1 Zipcode

In the data set, we are provided with three features that decide the geographical position of a house. They are "Zipcode", "Longitude", and "Lattitude". If we directly employ these attributes as they are, the correlation with Price will be insignificant. That is because the computer perceives them as float type data. If we use each zip code as a nomial type variable (using the "One hot code" package of scikit-learn to encode nomial type variable into sparse matrix form), our model will have 69 additional dummy variables. It would increase the complexity of the model and would eventually be punished when calculating accuracy.

The solution is to calculate the unit price of each square footage sale in each zipcode area and use it as a variable, named as "zipcode_up".

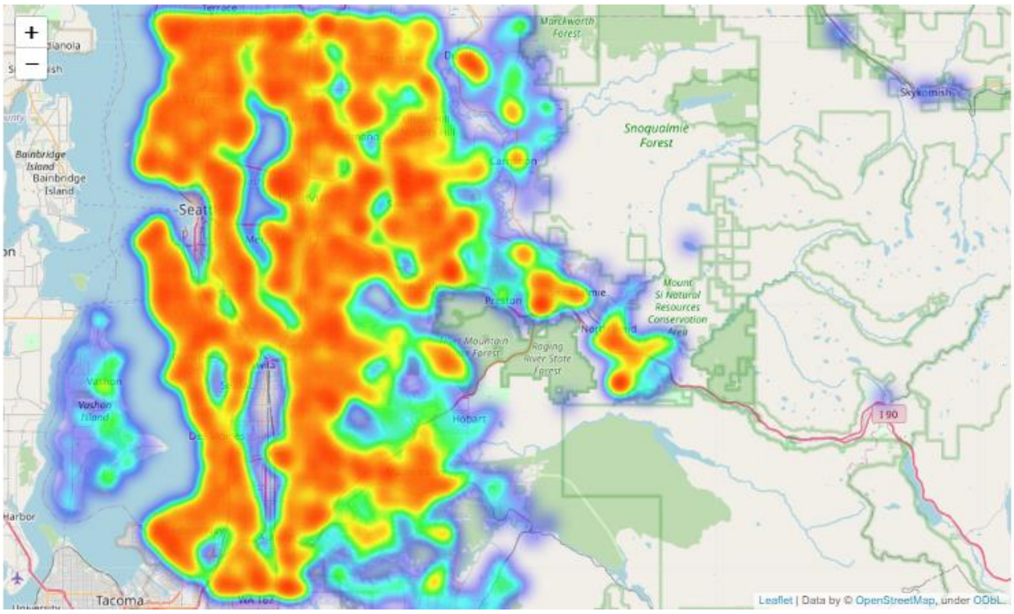

+
−
Bainbridge Island
Seat
Vashon Island
Tacoma
Snoqualmie Forest
Mount Si Natural Resources Conservation Area
Raging River State Forest
Preston
Hobart
Skykomish
I 90
Leaflet | Data by © OpenStreetMap, under ODbL

*Figure 20: Heat map by number of transactions (recreated referring to (Burhan, 2019))*

As illustrated in Figure 12, “zipcode_up” has a coefficient of 0.53 with house price. It is understandable since zipcode unit price is calculated from the prices of houses in a zip code area. Zipcode unit price makes the connection between price and zip code more visible.

### 4.1.2 Date of transactions

As stated before, transactions were made during May 2014 and May 2015. To make use of the momentum effect, the time of transactions is illustrated as in Figure 21.

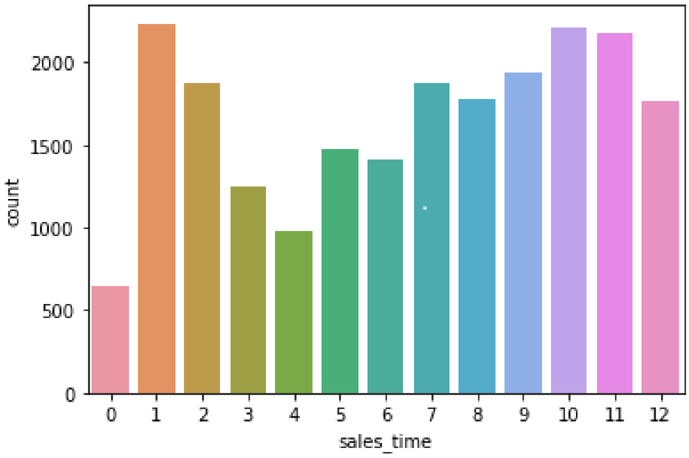


*Figure 21: Number of transactions by time (own illustration)*

The transactions that happened in May 2015 corresponds to $t = t_0$. Those happened in May 2014, as a result, have $t = t_{12}$. Because the dataset is collected in mid-May 2015, it just covers the transaction until that date, so the number for that month ($t = t_0$) is exceptionally low. One distinct feature we can see from the charts that around Christmas and New year, there were fewer transactions that are made, just around 1000 transactions. This is understandable since people did not want to move to a new house during the time, as well as they had to spend a portion of money shopping. The most active time of the market fell between March and October.

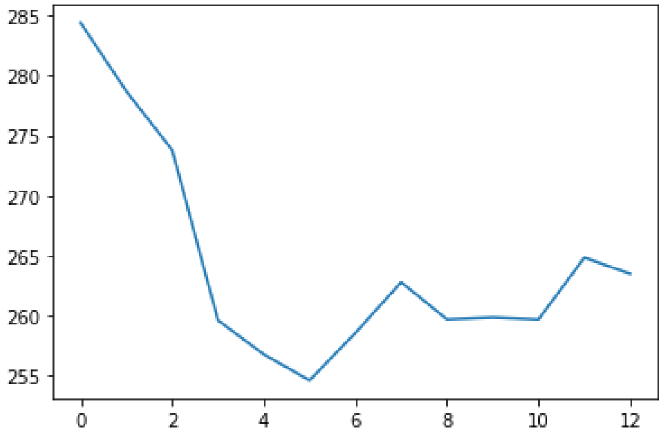


*Figure 22: Absolute sales price per square footage (dollar) by time (own illustration)*

The correlation between transaction time and the price is low for this particular dataset. For broader consideration, the timing could play an important role in determining the price of a house. Prices could reflect the sentiment of the market at a particular moment or the trend of the market in a period.

In this period and this area, the price of real estate generally (price per square footage sale) increased as illustrated in the below chart (t= 0 is the latest month).

### 4.2 Feature selection

As mentioned before, increasing the complexity of the algorithm would potentially reduce the accuracy of a model due to overfitting. Feature selection comes to extract a subset of features of the original feature set. The aim is to automatically reduce the number of dimensions of the model, reducing the computational effort needed or generalization errors.

Sequential feature selection algorithms allow us to search through and determine sub-features that most relevant to the target variable. They are a family of greedy algorithms, which take the local optimal solution at each step of the combinatorial search process that often arrives at a locally optimal. Greedy algorithms are in contrast to Exhaustive search algorithms, which search through all possible combinations to find global optimal. Sometimes it is infeasible or computationally expensive to find a globally optimal, then Greedy algorithms come to play (Sebastian, Vahid (2017, p. 214 - 215)).

In this thesis, Sequential Forward Selection (SFS) would be deployed. SFS starts with no feature, then tries to add a feature that most improves the performance of the model.

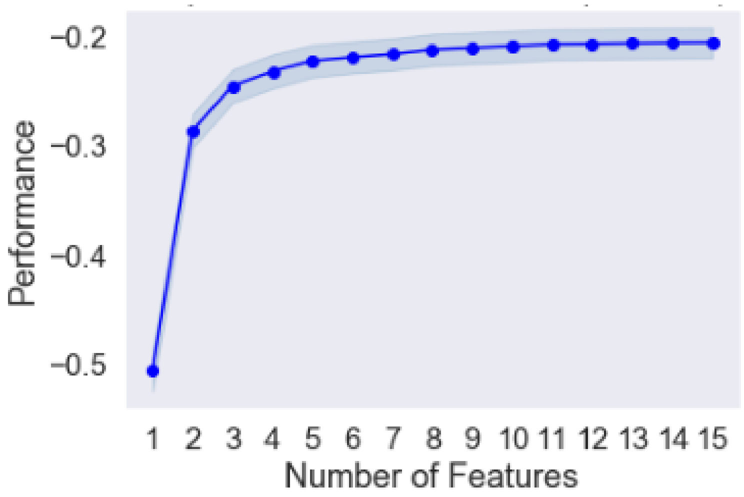


*Figure 23: Sequential Forward Selection with Standard errors (own illustration)*

The process continues until adding a feature does not improve the performance of the model or a maximum number of k-features is reached (Saurav, 2016). As illustrated in Figure 23, the performance of our model (using Random forest regression) on the training data set increases when we add more features to the model. However, the pace of increase decays as later features are added.

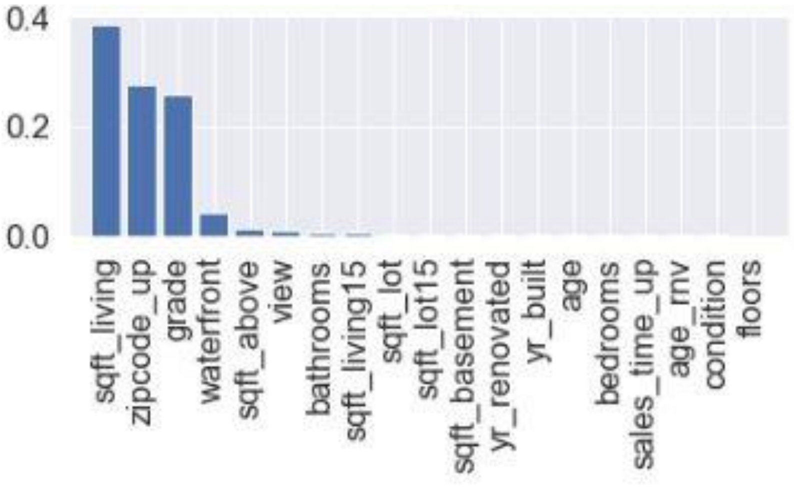


*Figure 24: Feature Importance (own illustration)*

As shown in Figure 12, some features are strongly correlated with each other. Thus, there is a little descriptive effect when adding one feature while the other was already added (both features have a strong correlation with the target feature and with each other). If we only focus on predictive performance, the features that are selected might not have strong practical explanatory meaning (Sebastian, Vahid (2017, p. 223)).

Square footage of living contributes almost 40% of the explanatory ability of the model. While the second and the third place belong to Zipcode unit price and Grade of the houses. The three top features account for 92% explanatory ability for the target variable. Fortunately, in this model, the three highest-ranked features are convincing. First, the price of a house is determined depending on how large it is, which is featured by “sqft_living”. Second, the house price is affected by where it locates, which is describe by “zipcode_up”. Third, a house with a better “grade”, which stands for the quality of design and construction, would yield a better price.

The three top features would be used to build a prediction model that is for private customers on the web application. A robust model with more variables would be used for corporate customers to evaluate their real estate portfolio and to combine a real estate portfolio with other publicly traded assets to form an optimal portfolio.

# 5. Regression results

In this section, the results of regression models would be presented. Then comparations and analysis of the results are made to chose the most suitable model for further usage.

## 5.1 Simple linear regression.

| | **MSE_std** | | $R^2$ | | **Adjusted** $R^2$ | | **Cross-validation** | |
|---|---|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test | Train | Test |
| LinearRegression 3 varibles | 0.272 | 0.278 | 0.728 | 0.729 | 0.728 | 0.729 | 0.73 (+/- 0.03) | 0.73 (+/- 0.03) |
| LinearRegression 13 varibles | 0.210 | 0.229 | 0.790 | 0.777 | 0.789 | 0.776 | 0.79 (+/- 0.02) | 0.77 (+/- 0.02) |
| LinearRegression 19 varibles | 0.203 | 0.219 | 0.797 | 0.786 | 0.796 | 0.785 | 0.80 (+/- 0.02) | 0.78 (+/- 0.02) |

*Table 2: Simple linear regression results (own illustration)*

First, some Simple linear regressions are conducted to examine how the model performs when we increase the number of explanatory features. As illustrated in Table 2, the performance increases when more variables are used. The model using 3 independent variables yield 0.729 r-squared scores for testing, the adjusted r-squared is the same (after being rounded). There is very little difference between r-squared and adjusted r-squared scores for all the three regressions. As presented in Section 2 adjusted r-squared is calculated as:

$$R^2_{adj} = 1 - \left[\frac{(1 - R^2)(n - 1)}{n - k - 1}\right]$$

The dataset includes 19 attributes so that the maximum value of k is 19. Here n equal to 21,613, which overwhelms the number of attributes k. (n-1)/(n-k-1) would become closer to 1, so that $R^2_{adj} \approx 1 - (1 - R^2) = R^2$. General speaking, if we have an adequately large informative dataset, the negative effect of the increasing complexity of the model would be reduced. In other words, to be able to use complex models, more data should be collected correspondingly.

As we increase the number of dimensions, the performance of the model also increased. This suggests that we could use the maximum number of attributes to predict the target variable. For that reason, all 19 attributes would be used to build a more precise model in web application.

### 5.2 Three variables model.

As stated in the section 4.3 Feature selection, three independent variables, “sqft_living”- square footage of living, “zipcode_up”- zipcode unit price, “grade” - the quality of design and construction, account for 92% explanatory ability for the target variable. Because of that, we would build a light model for private customers, who prefer a fast and convenient valuation tool. After making regression models and tuning them, the results are presented below:

| **3 variables model** | **MSE_std** | | **$R^2$** | | **Cross-validation** | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| LinearRegression | 0.272 | 0.278 | 0.728 | 0.729 | 0.73 (+/- 0.03) | 0.73 (+/- 0.03) |
| Ransac Regression | 0.272 | 0.279 | 0.728 | 0.728 | 0.73 (+/- 0.04) | 0.73 (+/- 0.04) |
| Ridge regression | 0.272 | 0.278 | 0.728 | 0.729 | 0.73 (+/- 0.03) | 0.73 (+/- 0.03) |
| Lasso regression | 0.272 | 0.278 | 0.728 | 0.729 | 0.73 (+/- 0.03) | 0.73 (+/- 0.03) |
| Quadratic polynomial regression | 0.184 | 0.169 | 0.816 | 0.835 | 0.81 (+/- 0.02) | 0.83 (+/- 0.02) |

| | | | | | | |
|---|---|---|---|---|---|---|
| Cubic polynomial regression | 0.179 | 0.167 | 0.821 | 0.837 | 0.81 (+/- 0.02) | 0.83 (+/- 0.02) |
| Random forest regression | 0.040 | 0.208 | 0.960 | 0.797 | 0.77 (+/- 0.03) | 0.80 (+/- 0.03) |
| Neural network regression | 0.202 | 0.196 | 0.798 | 0.809 | -0.23 (+/- 0.08) | -0.26 (+/- 0.08) |
| AdaBoost Regression | 0.278 | 0.281 | 0.722 | 0.726 | 0.70 (+/- 0.03) | 0.74 (+/- 0.03) |
| GradientBoosting Regression | 0.099 | 0.186 | 0.901 | 0.819 | 0.79 (+/- 0.02) | 0.81 (+/- 0.02) |
| XGBoost Regression | 0.118 | 0.172 | 0.882 | 0.832 | 0.80 (+/- 0.01) | 0.82 (+/- 0.01) |

*Table 3: Three variables multiple models regression results (own illustration)*

There is no big difference between the results of LinearRegression, Ransac Regression, Ridge regression, Lasso regression. The adjusted r-squared test scores are around 0.729, which is not bad. When polynomial factors are added, the results increase significantly.

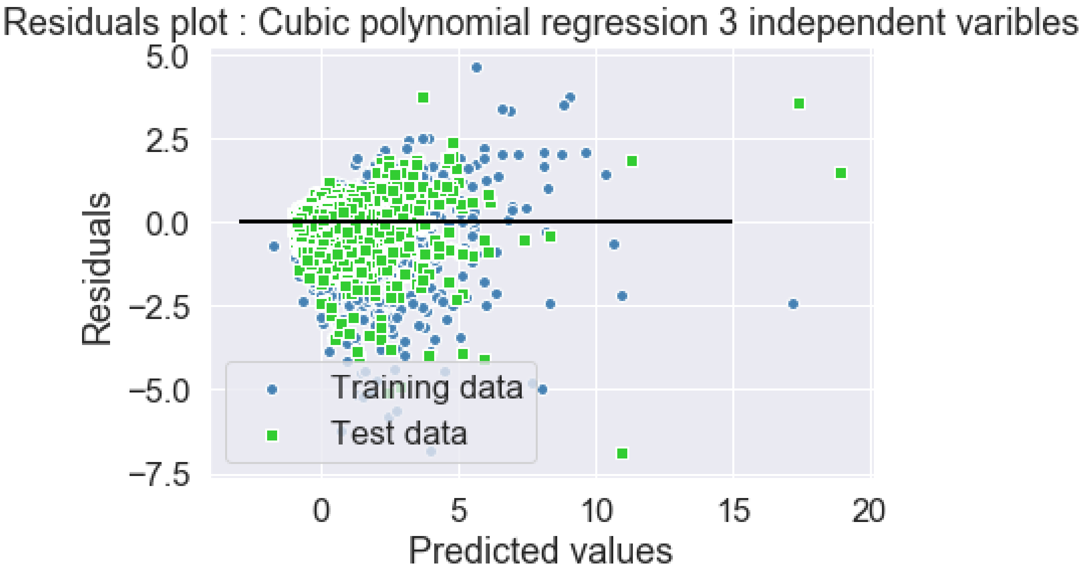


*Figure 25: Residuals plot of Cubic polynomial regression 3 independent variables (own illustration)*

The adjusted r-squared test score of Quadratic polynomial regression is 0.835, standardized MSE is 0.169. Cubic polynomial regression yields the best result among all with the adjusted r-squared test score of 0.837, standardized MSE is 0.167. The residuals plot of Cubic polynomial regression for the 3-variables model shows that the residuals are not close to randomly scattered around the zero-line, indicating that the model could not capture total explanatory information from the dataset.

Random forest regression yields the best training score with an adjusted r-squared of 0.960, standardized training MSE of 0.040. But it suffers from overfitting so that testing scores are significantly lower at 0.797 adjusted testing r-squared and standardized testing MSE 0.208. Neural network regression performs adequate results with adjusted testing r-squared of 0.809. But Neural network regression suffers from high uncertainty of results. This leads to the consequence that the testing cross-validation score is negative at -0.26 (+/- 0.08). The best possible r-squared score is 1.0, and “it can be negative because the model can be arbitrarily worse” (Scikit-learn document, 2020). If the scores are highly uncertain, they could be positive and negative at times of cross-validation and cancel each other.

When it comes to boosting algorithm, AdaBoost Regression’s performance is worse than its peers. GradientBoosting Regression, XGBoost Regression performed quite good with adjusted testing r-squared, in turn, are 0.819 and 0.832. These are better than that of Neural network regression but slightly worse than that of polynomial regression.

Because of having the best performance, the Quadratic polynomial regression would be chosen for deployment on the web application for the 3-variable model.

### 5.3 Nineteen variables model.

As the accuracy for models increases when the number of independent variables increases (for this particular dataset), all nineteen explanatory variables would be used to build a better model for house price prediction. The results are presented in the table below:

| **19 variables model** | **MSE_std** | | $\boldsymbol{R^2}$ | | **Cross-validation** | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| LinearRegression | 0.203 | 0.219 | 0.797 | 0.786 | 0.80 (+/- 0.02) | 0.78 (+/- 0.02) |
| Ransac Regression | 0.205 | 0.221 | 0.795 | 0.784 | 0.80 (+/- 0.03) | 0.78 (+/- 0.03) |
| Ridge regression | 0.203 | 0.219 | 0.797 | 0.786 | 0.80 (+/- 0.02) | 0.78 (+/- 0.02) |
| Lasso regression | 0.203 | 0.219 | 0.797 | 0.786 | 0.80 (+/- 0.02) | 0.78 (+/- 0.02) |
| Quadratic polynomial regression | 0.098 | 0.103 | 0.902 | 0.899 | 0.89 (+/- 0.02) | 0.89 (+/- 0.02) |

| | | | | | | |
|---|---|---|---|---|---|---|
| Cubic polynomial regression | 0.062 | +∞ | 0.938 | -∞ | -∞ | -∞ |
| Random forest regression | 0.018 | 0.110 | 0.982 | 0.892 | 0.87 (+/- 0.01) | 0.88 (+/- 0.01) |
| Neural network regression | 0.146 | 0.155 | 0.853 | 0.848 | -0.28 (+/- 0.42) | -0.65 (+/- 0.42) |
| AdaBoost Regression | 0.256 | 0.277 | 0.744 | 0.729 | 0.72 (+/- 0.03) | 0.74 (+/- 0.03) |
| GradientBoosting Regression | 0.037 | 0.096 | 0.963 | 0.906 | 0.89 (+/- 0.02) | 0.90 (+/- 0.02) |
| XGBoost Regression | 0.049 | 0.100 | 0.951 | 0.903 | 0.89 (+/- 0.02) | 0.89 (+/- 0.02) |

*Table 4: Nineteen variables multiple models regression results (own illustration)*

As the same in the three variables model, there is no big difference between the results of Linear Regression, Ransac Regression, Ridge regression, Lasso regression. The adjusted r-squared test scores are around 0.786, standardized MSEs are approximately 0.219. The adjusted r-squared test score of Quadratic polynomial regression is 0.899, standardized MSE is 0.103. Quadratic polynomial regression's result is the third-highest when 19 variables are used. Cubic polynomial regression is not the champion anymore. The model explodes as the test score are worst although the training scores are among the highest. That is a typical example of overfitting. Neural network regression still performs adequate results with adjusted testing r-squared of 0.848 and still suffers from high uncertainty of results with a cross-validation score of -0.65 (+/- 0.42). When it comes to process metadata such as images, ANNs could be the most preferred, but for tabular data regression analysis, it does not always prevail.

The new champion is GradientBoosting Regression with an adjusted r-squared of 0.963 for training and 0.906 for testing, a standardized MSE of 0.096, and a cross-validation score of 0.90 (+/- 0.02). Following closely is XGBoost Regression with an adjusted r-squared of 0.903. Random forest regression performs pretty well but still suffers from overfitting when the training and testing adjusted r-squared are 0.982 and 0.892 respectively.

The residuals plot in Figure 26 shows that although the model still suffers from non-random residuals scattered around the centerline, it has been advanced significantly when we compare the residuals plot of 19 variables and the 3 variables' plot. Additionally, the model still contains some outliers, which vary far away from the centerline.

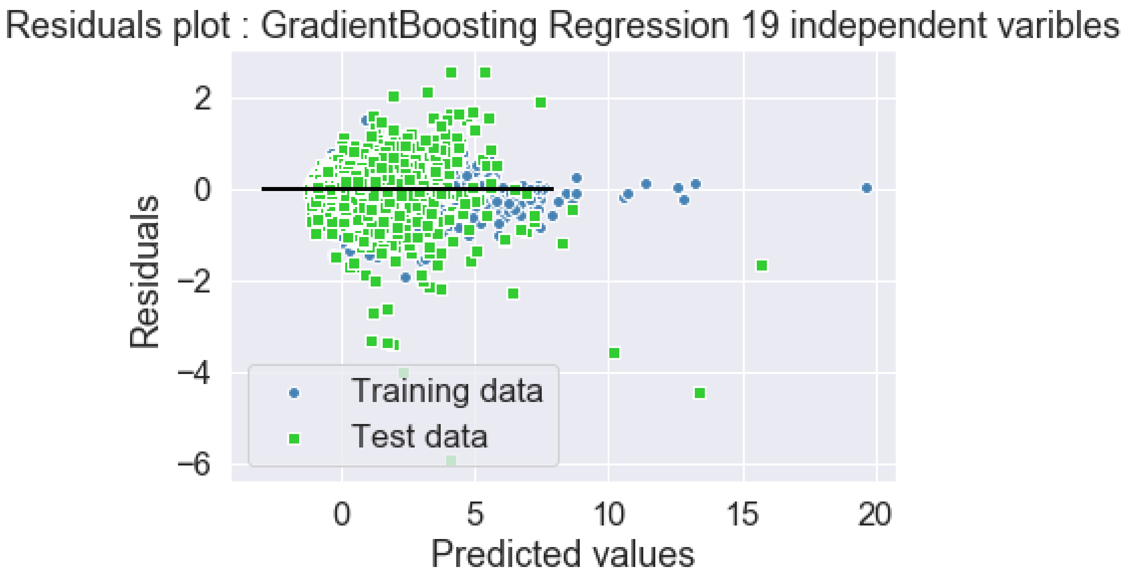


*Figure 26: Residuals plot of GradientBoosting Regression 19 independent variables (own illustration)*

Because of its best performance, the GradientBoosting Regression would be used as a complex model on the web application.

# 6. Web application deployment of the models

After the regression models are chosen, they would be deployed to build a web application. In this section, the structure of the web application would be presented along with some sample results.

## 6.1 The web application structure

The models first would be trained and the regression results are stored in pickle (.pkl) files. Pickle is a Python package that allows programmers to store the trained model's results. So that they can be called to make predictions without training the model again. To use the web application, users access the website and input data. Then the data will be transfer to Python for calculation through Flask. The regression models are retrieved for calculation, the results then are given back to the web app for users. The web application could be accessed at https://agentkaren.herokuapp.com/ (preferred to be accessed on PC or Laptop).

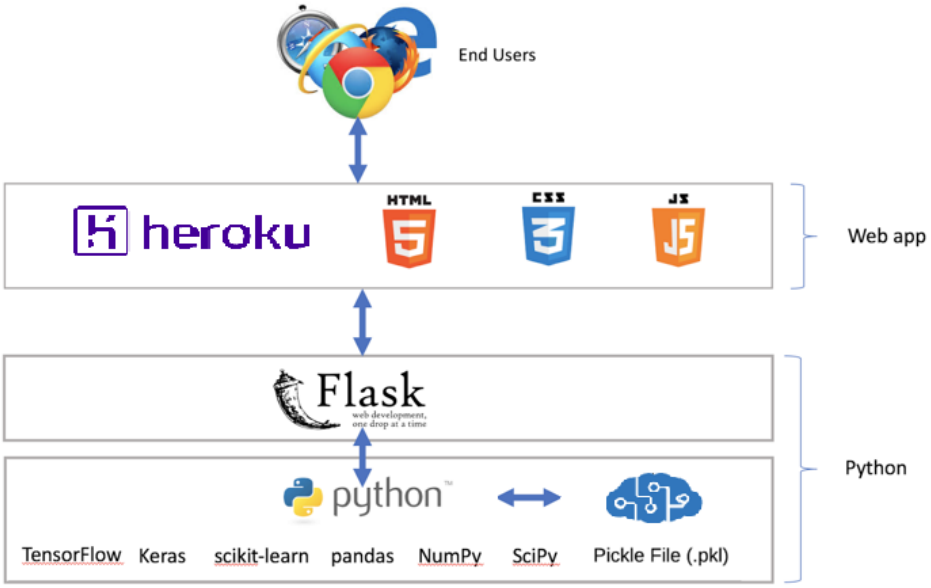
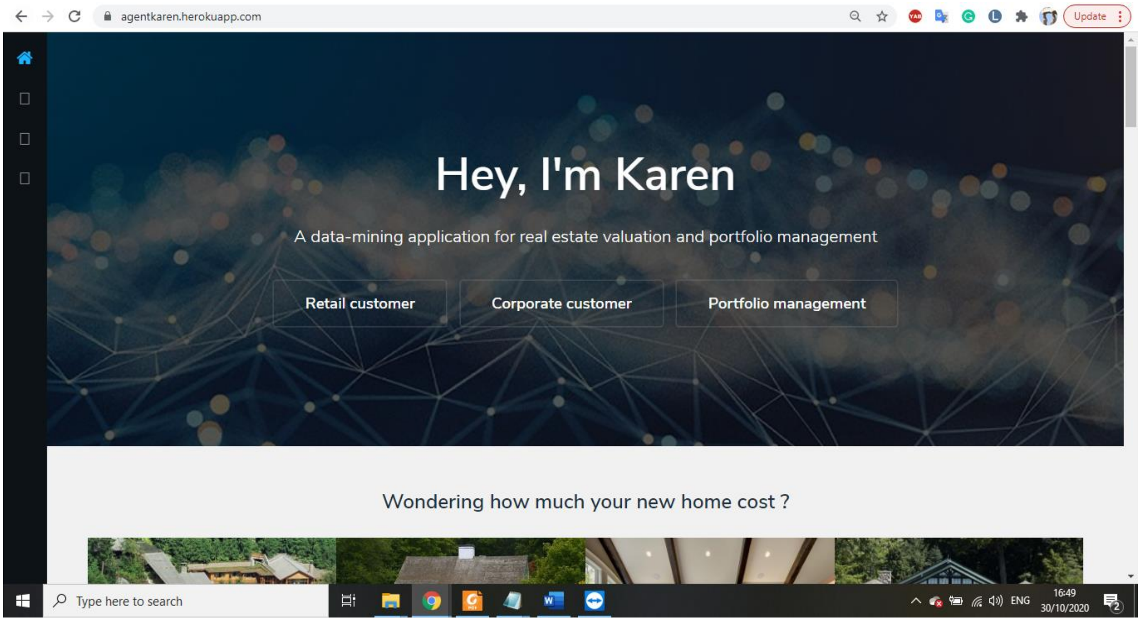


*Figure 27: Web application structure (Shivas, 2017)*



Hey, I'm Karen

A data-mining application for real estate valuation and portfolio management

Retail customer

Corporate customer

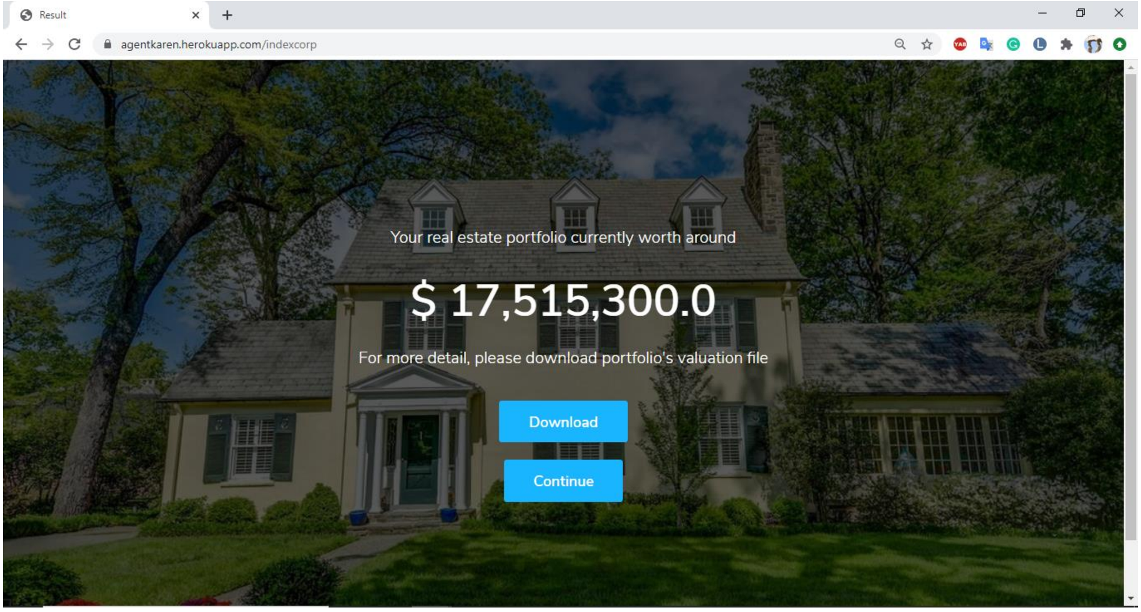
Portfolio management

Wondering how much your new home cost ?

*Figure 28: Web application Home page (own illustration)*

## 6.2 House price prediction and portfolio management results

At both models, 3-variables and 19-variables, after making predictions, the values of houses would be displayed. The 3-variables model is used for retail customers, who want quick and simple predictions. In this model, users input 3 values direct on the website and click submit for the result. For 19-varibles model of corporate customers, users input the data to a sample excel file and upload the file to the system. The results would be used to create a result file, which users could download to see all the detail about the valuation of each house.



Your real estate portfolio currently worth around

$ 17,515,300.0

For more detail, please download portfolio's valuation file

*Figure 29: Result page of the Valuation feature (own illustration)*

When it comes to the portfolio management feature, users also input the data into an excel file the data of the house along with tickers of publicly traded security. The app will retrieve the data from yahoo_finance for the security stock price and the house data would be used to evaluate the real estate portfolio. Because the period of the real estate data set is between May 2014 and May 2015, the stock price data should be available in this time frame so that the app could run properly. In other words, the companies' stocks or selected securities have to be traded in the period.

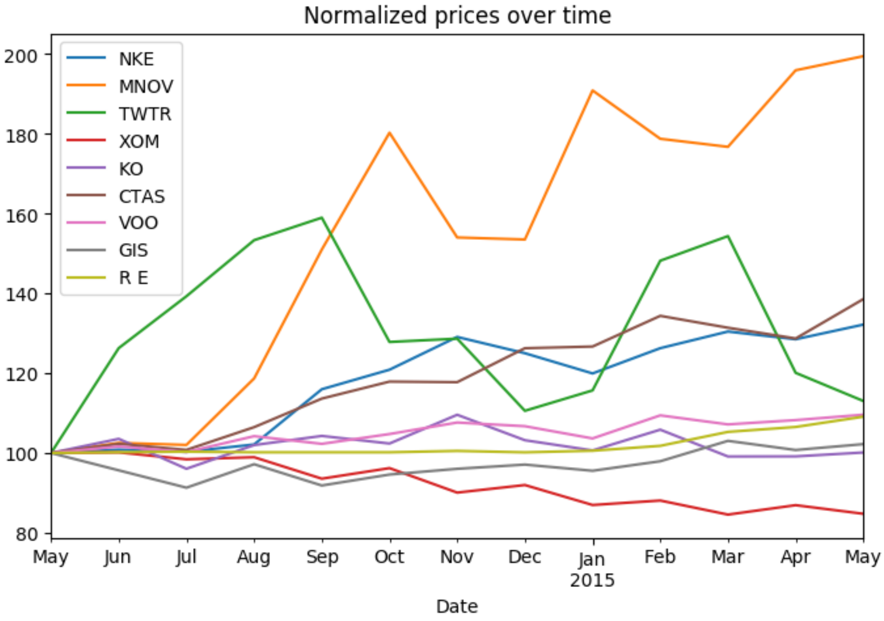


*Figure 30: Normalized prices (own illustration)*

The valuation of real estate portfolio through time would be determined by changing the time of sales (see section 4.1.2 Date of transactions), assuming all other aspects stay the same. The normalized prices are calculated and displayed in Figure 30 (real estate portfolio is denoted as R E).

Since the real estate prices are calculated monthly because of the low liquidity, the other assets' prices are also taken monthly. The asset prices are normalized while the log returns and covariance are then annualized so that they are not so different as if the prices would have taken daily.

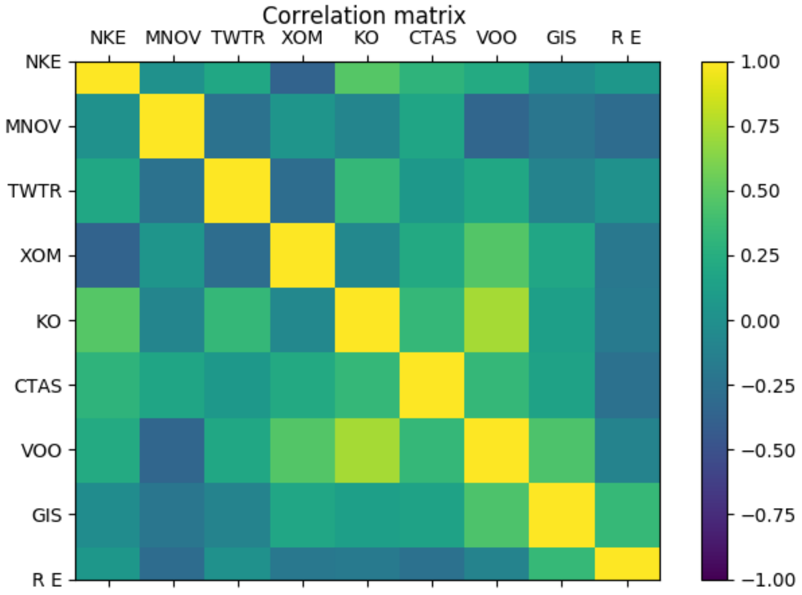
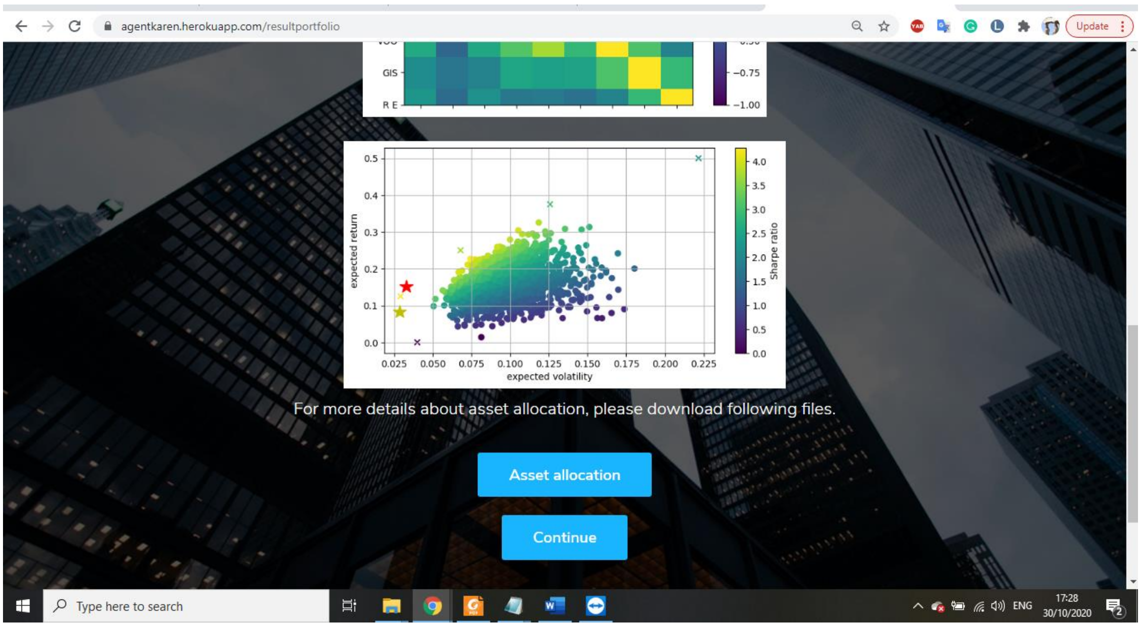


*Figure 31: Correlation matrix (own illustration)*



For more details about asset allocation, please download following files.

*Figure 32: Result page of the Portfolio management feature (own illustration)*

Then the application will generate a plot of the random-weight portfolio (assuming short-selling is prohibited), the Min Variance Portfolio (MVP) (the yellow star), and the

Maximum Sharpe ratio portfolio (MSP) (the Red star). The efficient frontier is the line that connects the “x” marks. In this Figure, the expected return is bounded that is not allowed to be significantly less than zero. Otherwise, the portfolio would witnesses losses, which should be avoided.

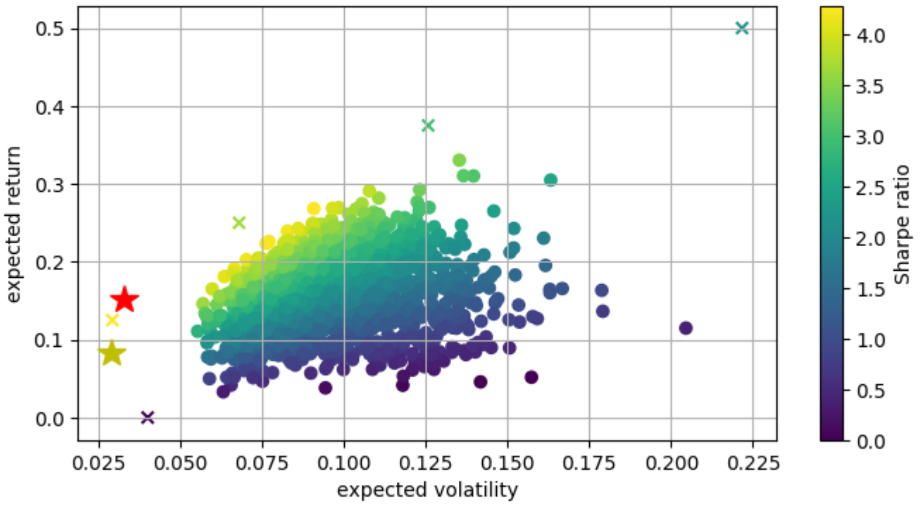
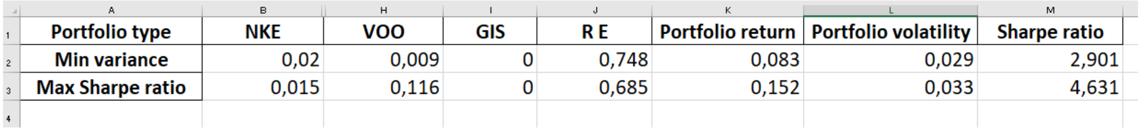


*Figure 33: Portfolio optimization with real estate (own illustration)*

The optimization process takes a significant amount of time, which could make the web app crashed. For that reason, the frontier is marked by “x” at only some representative points.

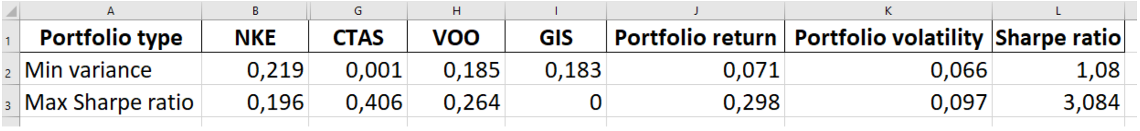


| Portfolio type | NKE | VOO | GIS | R E | Portfolio return | Portfolio volatility | Sharpe ratio |
|---|---|---|---|---|---|---|---|
| Min variance | 0,02 | 0,009 | 0 | 0,748 | 0,083 | 0,029 | 2,901 |
| Max Sharpe ratio | 0,015 | 0,116 | 0 | 0,685 | 0,152 | 0,033 | 4,631 |

*Figure 34: Asset allocations with real estate (own illustration)*

The asset allocations, portfolio return, and Sharpe ratio of the MVP and the MSP are then stored in an excel file for downloading as in Figure 34. The feature that allows investors to input a level of risk or return and maximize or minimize the other would be available in the future releases of the app.

One aspect that worths mentioning is the effect of including real estate in the total portfolio. Figure 35 and 36 shows how the optimization looks like without real estate.



| Portfolio type | NKE | CTAS | VOO | GIS | Portfolio return | Portfolio volatility | Sharpe ratio |
|---|---|---|---|---|---|---|---|
| Min variance | 0,219 | 0,001 | 0,185 | 0,183 | 0,071 | 0,066 | 1,08 |
| Max Sharpe ratio | 0,196 | 0,406 | 0,264 | 0 | 0,298 | 0,097 | 3,084 |

*Figure 35: Asset allocations without Real estate (own illustration)*

The MVP without real estate could reach minimum volatility of 0.066. If we include real estate the number is 0.029 as in Figure 34. Sharpe ratio for MSP including real estate is 4,631, while that of MSP excluding real estate is 3.084. It is clear that including real estate could lead to significant diversification (lower volatility and higher Sharpe ratio). The difference in the Efficient frontier is as Figure 36 compares to Figure 33.

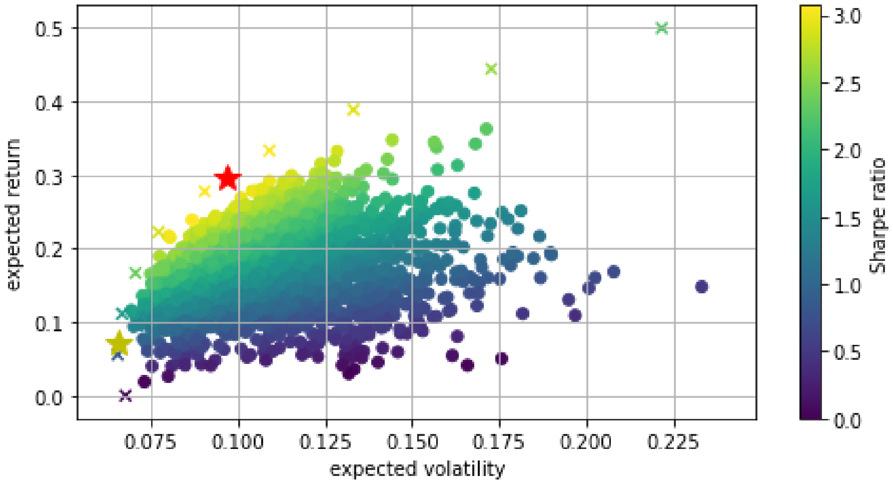


*Figure 36: Portfolio optimization without Real estate (own illustration)*

## 7. Summary of the analysis and recommendations

In this section, the findings from the above analysis would be briefly summarized. Then a conclusion and some put-forward recommendations would be presented.

### 7.1 Summary of analysis

In this thesis, various regression techniques have been explored, ranging from Simple linear regression to more advanced techniques such as ANNs regression and Boosting algorithms. From the modeling point of view, on a robust and informative dataset, there is little difference in performances of LinearRegression, Ransac Regression, Ridge Regression, and Lasso regression. Random forest regression performs well on the training dataset, but it suffers from overfitting so that testing scores are significantly worse than the training score. Polynomial seems to be the best choice for a low dimension model. Higher degree polynomial would suffer heavily from overfitting and those models could explore when many independent variables are employed. Neural network regression performs adequately well on the dataset. The Boosting algorithm, especially GradientBoosting and XGBoost yields the top results, both on low and high dimension models.

From the features point of view, by transforming some ordinal features (such as zip code and date of sales) we could refine more descriptive information compare to directly deploy the features to the model. As shown from features selection, the position, the size, and the condition of design and construction account for 92% of the house price on this dataset. Many attributes are positively correlated with house prices, such as the number of bedrooms, the number of bathrooms, the houses' view, the number of floors. There are also attributes that negatively correlated with house prices, for example, age and age from the last renovation.

Real estate has been a long time an alternative investment instrument that helps investors to diversify their portfolio. By deploying a robust valuation model, it is more convenient to combine real estate and other publicly traded security for diversification purposes.

## 7.2 Recommendations

In the above-mentioned regression, the relationship between the target variable and the explanatory variables is sharp and exactly measured. In reality, it is not the case and the relationship is somehow vague and non-exactly measured. Another situation that the dataset is too small or when the normal distribution cannot be verified. That's when Fuzzy logic comes to play. For further research, Fuzzy logic could be used to establish a Possibilistic Regression Model, which "minimizes the fuzziness of the model by minimizing the total spreads of its fuzzy coefficient" (Shapiro, 2006).

Blockchain is a promising technology, which helps store all real estate transactions in realtime and securely. In the future, if real estates are registered and managed using blockchain technology, it would be a robust data ledger for advanced regression models to mine.

# Bibliography

Abdallah A., Sree I., Pawan., Sakshi S., Karpagam T.V., (2017) King County House Prices Prediction Model, URL: https://www.slideshare.net/PawanShivhare1/predicting-king-county-house-prices Last visit on: 31.08.2020

Aishwarya (2019), R-Squared vs Adjusted R-Squared, URL: https://medium.com/analytics-vidhya/r-squared-vs-adjusted-r-squared-a3ebc565677b Last visit on: 03.09.2020

Apoorva (2018) Regression in Machine Learning. URL: https://medium.com/datadriveninvestor/regression-in-machine-learning-296caae933ec Last visit on: 24.08.2020.

Ata (2015), Artificial neural networks applications in wind energy systems: a review, Renew. Sustain. Energy Rev. 49 (2015) 534– 562.

Avinash (2017), Understanding Activation Functions in Neural Networks, URL: https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0 Last visit on: 27.08.2020

Baranidharan (2019), Linear Regression, URL: https://www.linkedin.com/pulse/linear-regression-baranidharan-rajan Last visit on: 22.09.2020

Burhan (2019), Predicting House Prices, URL: https://www.kaggle.com/burhanykiyakoglu/predicting-house-prices Last visit on: 28.09.2020

Chayan (2019), Regression — Why Mean Square Error?, URL: https://towardsdatascience.com/https-medium-com-chayankathuria-regression-why-mean-square-error-a8cad2a1c96f Last visit on: 02.09.2020

Erdi T., Kadir A., Mehmet B., (2016) Comparison of linear regression and artificial neural network model of a diesel engine fueled with biodiesel-alcohol mixtures, Alexandria Engineering Journal (2016) 55, 3081–3089.

Ethen (2020), Model selection, machine learning, URL: http://ethen8181.github.io/machine-learning/model_selection/model_selection.html Last visit on: 22.09.2020

Haykin (1999), Neural Networks: A Comprehensive Foundation, Prentice Hall, Ontario.
Internal Revenue Service (2020) Determining the Value of Donated Property. URL https://www.irs.gov/publications/p561#en_US_201911_publink1000258001 Last visit on: 17.08.2020.

James (2020), Modern Portfolio Theory (MPT), URL: https://www.investopedia.com/terms/m/modernportfoliotheory.asp Last visit on: 20.09.2020

Jason (2020), Linear Regression for Machine Learning, URL: https://machinelearningmastery.com/linear-regression-for-machine-learning Last visit on: 29.09.2020

Krishni (2018) A Beginners Guide to Random Forest Regression. URL: https://medium.com/datadriveninvestor/random-forest-regression-9871bc9a25eb Last visit on: 25.08.2020.

Leonardo (2018), Predicting House Prices [XGB/RF/Bagging-Reg Pipe], URL: https://www.kaggle.com/kabure/predicting-house-prices-xgb-rf-bagging-reg-pipe Last visit on: 06.09.2020

Marshall (2020), Sharpe Ratio, URL: https://www.investopedia.com/terms/s/sharperatio.asp Last visit on: 22.09.2020
Mathworks (2020), Estimate Multiple Linear Regression Coefficients, URL: https://de.mathworks.com/help/stats/regress.html Last visit on: 20.09.2020

Mihnea (2019) Machine-Learning Real Estate Valuation: Not Only a Data Affair. URL: https://towardsdatascience.com/machine-learning-real-estate-valuation-not-only-a-data-affair-99d36c92d263 Last visit on: 17.08.2020.

Missinglink.ai, (2020) Neural Networks for Regression (Part 1)—Overkill or Opportunity? URL: https://missinglink.ai/guides/neural-network-concepts/neural-networks-regression-part-1-overkill-opportunity/ Last visit on: 27.08.2020

Pang, N.T., Michael, S., Vipin, K., (2013), Introduction to data mining, Second ed., Pearson Education Limited.

Pratik (2020), Complete Guide On Linear Regression Vs. Polynomial Regression With Implementation In Python, URL: https://medium.com/datadriveninvestor/complete-guide-on-linear-regression-vs-polynomial-regression-with-implementation-in-python-964c64c28aa8 Last visit on: 20.09.2020

Rohith (2018) Introduction to Machine Learning Algorithms: Linear Regression. URL: https://towardsdatascience.com/introduction-to-machine-learning-algorithms-linear-regression-14c4e325882a Last visit on: 24.08.2020.

Sanjay (2018), Why and how to Cross Validate a Model? URL: https://towardsdatascience.com/why-and-how-to-cross-validate-a-model-d6424b45261f Last visit on: 03.09.2020

Saurav (2016), Introduction to Feature Selection methods with an example (or how to select the right variables?), URL: https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ Last visit on: 13.09.2020

Scikit-learn document (2020), $R^2$ score, the coefficient of determination, URL: https://scikit-learn.org/stable/modules/model_evaluation.html#the-scoring-parameter-defining-model-evaluation-rules Last visit on: 16.09.2020

Sebastian, Vahid (2017) Python Machine Learning. Second ed., Packt.

Shapiro (2006), Fuzzy regression model, URL: https://www.soa.org/globalassets/assets/files/static-pages/research/arch/2006/arch06v40n1-ii.pdf Last visit on: 20.09.2020

Shivas (2017), Integrating a Machine Learning Model into a Web app, URL: https://github.com/shivasj/Integrating-a-Machine-Learning-Model-into-a-Web-app Last visit on: 20.09.2020

Sunil (2015), Quick Introduction to Boosting Algorithms in Machine Learning, URL: https://www.analyticsvidhya.com/blog/2015/11/quick-introduction-boosting-algorithms-machine-learning/ Last visit on: 03.09.2020

Tavish (2015), Getting smart with Machine Learning – AdaBoost and Gradient Boost, URL:. https://www.analyticsvidhya.com/blog/2015/05/boosting-algorithms-simplified/ Last visit on: 05.09.2020

Teuben, B., Neshat, R., (2020) MSCI's Annual update on the size of the professional managed global real estate investment market, 2020 realease, MSCI.

Tom (2020), An Introduction to Support Vector Regression (SVR), URL: https://towardsdatascience.com/ an-introduction-to-support-vector-regression-svr-a3ebc1672c2 Last visit on: 31.08.2020

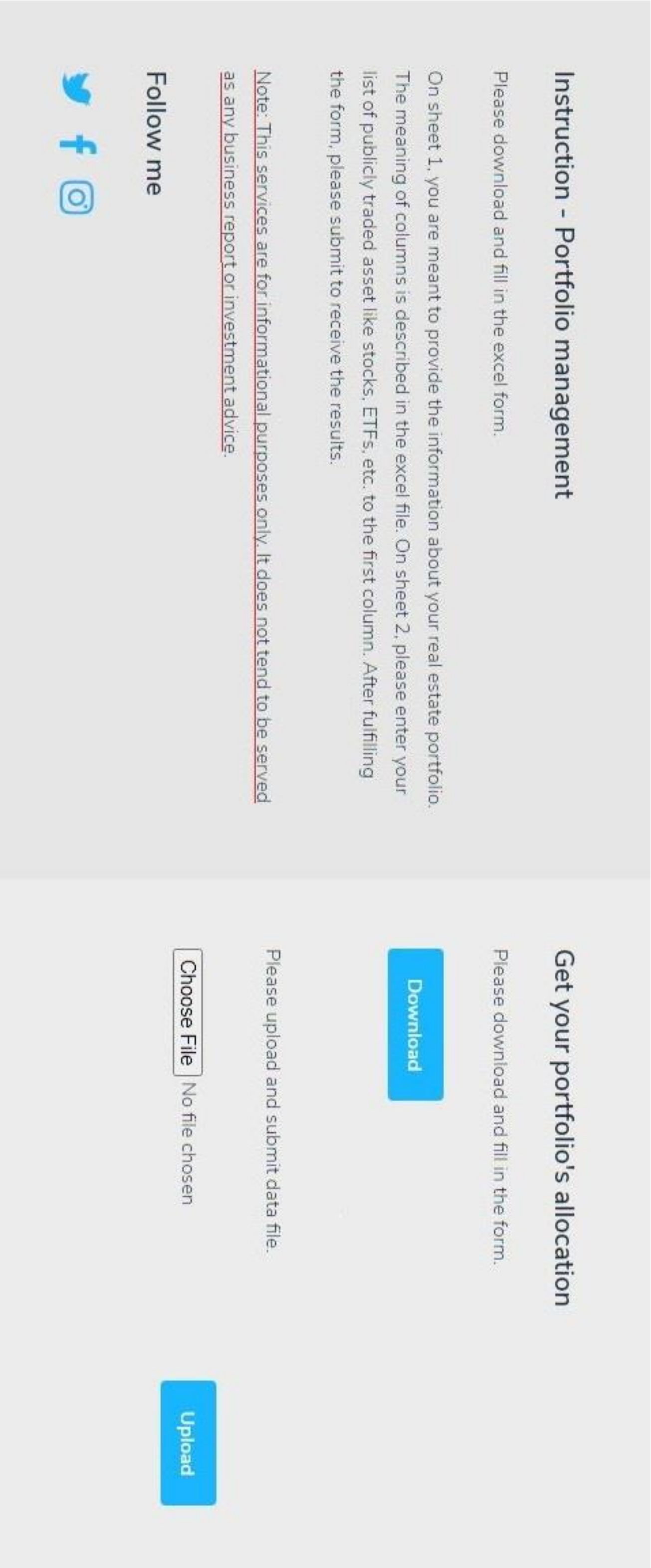
Vijay (2019), L1-L2 Regularization, URL: https://medium.com/@pavanmeduri1_55193/l1-l2-regularization-409039dd111a Last visit on: 30.10.2020

Vishal (2019), XGBoost Algorithm: Long May She Reign!, URL: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d Last visit on: 06.09.2020

Will (2020), Capital Asset Pricing Model (CAPM), URL: https://www.investopedia.com/terms/c/capm.asp Last visit on: 22.09.2020

# Appendix

Below is one of the instruction forms on the web application. In this one, the note of the disclaimer is added for legal protection.



## Get your portfolio's allocation

Please download and fill in the form.

Please upload and submit data file.

## Instruction - Portfolio management

Please download and fill in the excel form.

On sheet 1, you are meant to provide the information about your real estate portfolio. The meaning of columns is described in the excel file. On sheet 2, please enter your list of publicly traded asset like stocks, ETFs, etc. to the first column. After fulfilling the form, please submit to receive the results.

Note: This services are for informational purposes only. It does not tend to be served as any business report or investment advice.

*Figure 37: The instruction form including the disclaimer of the web application (own illustration)*