# CS 490 Master’s Project

# Housing Price Prediction in New York, NY

# Submitted By: LATHA SARADHA

# Email ID: lsaradha@neiu.edu

# Student ID: 660146

# Semester: August 2020

**Table of Contents**

[**Abstract** 3](#_Toc47437309)

[**Introduction** 4](#_Toc47437310)

[**Data** **Preparation** 5](#_Toc47437311)

[1. Description 5](#_Toc47437312)

[2. Data Cleaning and Pre-processing 8](#_Toc47437313)

[3. Data Analysis 9](#_Toc47437314)

[4. Data Standardisation 14](#_Toc47437315)

[**Why Machine Learning Models?** 15](#_Toc47437316)

[Model Evaluation Metrics 16](#_Toc47437317)

[**Prediction Models** 20](#_Toc47437318)

[**1.** **Linear Regression** 20](#_Toc47437319)

[Python Implementation 22](#_Toc47437320)

[**2.** **SVR Regression** 25](#_Toc47437321)

[Python Implementation 28](#_Toc47437322)

[**3.** **K- Nearest Neighbor** 30](#_Toc47437323)

[Python Implementation 31](#_Toc47437324)

[**Tree-Based Models** 33](#_Toc47437325)

[**1.** **Random Forest Regression** 33](#_Toc47437326)

[Python Implementation 33](#_Toc47437327)

[**2.** **XGBoost Regression** 35](#_Toc47437328)

[Python Implementation 35](#_Toc47437329)

[**Conclusion and Future Work** 37](#_Toc47437330)

[**Appendix** 38](#_Toc47437331)

[**References** 39](#_Toc47437332)

# **Abstract**

In this project, an analysis of real-life housing price prediction is conducted based on the data collected from New York City. Along with the data collected from 'Redfin' regarding the price and features of a house, external data sets are incorporated, which includes information about the surrounding area. Combining all these features, data visualization is done to the cleaned and pre-processed data to identify the trends and patterns. Important features are studied, identified, and selected to attain dimensionality reduction. Following this, different machine learning models are applied to the datasets. For each model, the performance metrics are analyzed. It is concluded that the "Random Regression Model" predicts the house price with the most accuracy and the fewest errors among the selected models.

Keywords: Statistics, Supervised Model, Overfitting, Data Set, Variable

# **Introduction**

After six years of steady house price growth, the U.S. housing market is moderating. In the United States, house sales have grown by 34% in the last decade and reached a record high of 5.51 million units last year (Gao, et al., 2019). Despite a record bull market, the housing market in the U.S. could enter a recession in 2020, according to Zillow[[1]](#footnote-1). House price prediction (Kain & Quigley, 1970), (Schulz & Werwatz, 2004) has therefore gained extensive attention because the prediction outcomes can help various real estate stakeholders make more well-informed decisions.

All three major categories of stakeholders (buyers, sellers, and real estate agents) in the real estate market could benefit from house price prediction. For example, buyers could use house price prediction to search for candidate houses that match their financial capabilities and desires. Similarly, house owners could keep monitoring the market and seeking the best opportunity for selling their houses. Moreover, real estate sales agents also rely on house price prediction to help customers better understand market trends. The accuracy of prediction has become an important criterion. Various other industries like the insurance industry, public development authorities, and law enforcement authorities could also benefit from the house price prediction process.

Most of the house price prediction projects are based on the dataset from Boston housing. It is considered a standard set of data to try different concepts in statistical learning. This project is to find a similar model for the New York City data and find a suitable and more accurate model for price prediction. Instead of using a readily available data set, during the course of this project a new dataset was collected, analyzed, cleaned, and prepared. External features which influence the price of a house was also considered and incorporated in the study. Along with the house features, geographical features around the house such as distance to a retail store, medical facility centre, schools, etc are also evaluated.

The loaded data sets are cleaned and analyzed and missing values are dropped or assigned values accordingly. The correlation plot and relation between different features are calculated.

As the last step, different machine learning models are applied, accuracy and errors are calculated. It is important to compare the performance of multiple different machine learning algorithms consistently. Different models are compared and analyzed using several factors such as accuracy, root mean square error, etc.

The Research Question is which Machine Learning Model will be the most optimal for housing price prediction in New York City, based on a new real-life dataset with consideration of geographical features around the house.

# **Data** **Preparation**

## Description

In this project data collected from different sources are combined to create a new dataset. The house features are collected from Redfin Corp, which is a real estate brokerage (RedFin, n.d.)The only filter used to retrieve data is setting the location as "New York, NY, USA". The range of house prices is between 0 $ and 2.75 M $. The dataset was started with 19173 records. All the datasets were exported from the Red Fin website as CSV files. There was a total of 72 CSV files. Using our application, the CSV files are imported and combined.

The following are the features retained from this dataset after removing the unwanted features.

Property Type - *Condo/Co-op, Single Family Residential, etc.*

City - *Name of the city where the house resides*

Zip or Postal Code - *Zip code where the house resides*

Price -*the listing price of the house*

Beds - *Number of beds*

Baths - *Number of baths*

Location - *The community where the house belongs*

Year Built - *The Year of built of house*

Square Feet - *Total Square feet area of the house*

Latitude - *Latitude*

Longitude - *Longitude*

Age - *Derived feature from the year built of house*

City (numeric) - *The numeric equivalent of CITY name*

The decision to purchase/sell a house is embedded within a set of economic and socio-cultural processes. A house’s price is not only dependent upon its features but also the factors such as location, age, condition, local market, crime rate, population, proximity to school, stores, public transport, etc. (Levy, K.C. Lee, & Murphy , 2008). In this project, additional datasets include School, Crime data set, retail food stores, NYPD (New York City Police Department) complaints recorded for the year 2019, New York City substations, New York City population, New York City health care facilities, School Safety report. All the distance calculations are done using Manhattan Distance – “*a form of geometry in which the*[*distance*](https://en.wikipedia.org/wiki/Distance)*between two points is the sum of the absolute differences of their Cartesian coordinates.*” (Taxicab geometry, n.d.)

House dataset will not consist of the information about the school rating or nearby school data, thus additional datasets are used for the same. Three datasets are used to retrieve the information about school rating and location. 2009-10 School Progress Report, School Locations,2010 – 2016 School Safety Report. School progress report data set consists of details about the “overall grade” for the school for the year 2009-10. Schools are only compared to other schools in the same School Level. (2009-2010 School Progress Reports - All Schools, n.d.). Due to the absence of coordinate details of the location of the school, another dataset that provides the same is merged based on the “ATS System Code” a unique identifier for each school. (School Locations 2017-2018, n.d.). The Third dataset which provides information about the crime incidents that occur in New York City public schools (2010-2016-School-Safety-Report , n.d.), is used to identify the number of complaints that have been recorded around a school within a distance of 0.1-mile radius. School records that were not present in any of the three datasets were rejected.

Each school has a rating of A-F, where A being the best and F to be the worse. After combining the school dataset with house data set, for each house record, Number of Level A-F schools, the total number of schools, and the total number of complaints occurred in those schools is calculated. The distance from a school to the house was set to a 3.0-mile radius while calculating.

NYPD complaints dataset is merged with the main dataset and unimportant features are eliminated. This dataset includes all valid felony, misdemeanour, and violation crimes reported to the New York City Police Department (NYPD) for the year 2019. (NYPD-Complaint-Data-Current-Year-To-Date, n.d.) . The total number of complaints was calculated using the distance calculation between the occurrence of the incident and house location within the radius of 1.0 mile.

New York City Population by Neighbourhood dataset is combined using the “Zip code” of the neighbourhood as a merging condition (New York Population Density Zip Code Rank, n.d.). Two different features are retrieved from this merging, population, and people per square mile.

Population - *population of the community*

People/Sq. Mile – *population of the community per square mile of land area*

NYC (New York City) Health hospitals and patient care locations dataset was combined with the house data. This is a list of the public hospitals, skilled nursing facilities, and some of the community-based health centres that are part of the NYC Health + Hospitals system as of 2011. (NYC Health + Hospitals patient care locations - 2011, n.d.). The feature retrieved from this dataset is the total number of hospitals from each house within a 5.0-mile radius.

Total Num of Hospitals – *total number of hospitals, patient care locations near the*

*house.*

NYC Subways Stations dataset (NYC Subways Stations, n.d.) is combined with the main data set and two features are interpreted. Manhattan distance from each house to the closest subway station and the number of subways stations within a 1-mile radius was calculated. NYC retails stores dataset, which includes the listing of all retail food stores which are licensed by the Department of Agriculture and Markets (Retail Food Stores, n.d.), provides three properties, Number of retail stores from each house within a 1-mile radius, the distance from each house to the nearest store, and the number of retail stores available in the corresponding zip code as that of house.

## Data Cleaning and Pre-processing

The main data set of house features are cleaned by removing the duplicate records. The records with no details for “beds”, “baths” were removed. The records with missing values for the “square feet” column was filled with arithmetic mean value calculated by grouping the beds and baths. Those records with missing “square feet” values after the calculation are removed. In this project, five major categories of houses were considered, 'Condo/Co-op', 'Single Family Residential', 'Multi-Family (2-4 Unit)', 'Townhouse', and 'Multi-Family (5+ Unit)'.

To get rid of incorrect data, “Zip code” details are filtered with the length being equals to 5. “City” feature is converted to lower case to find the appropriate numeric representation as a sum of character’s ASCII (ASCII, n.d.) values, which is represented as “City Numeric”. Year built feature from the housing dataset was converted to “Age” As the housing data includes all the available properties in the market at the time of access, we set the current year to 2021 and computed the age of each property.

While combining the additional datasets, the features which are represented as numeric values are identified. Irrelevant features are removed while merging the main data set with additional data sets. In the school dataset, those records with missing values for “2009-2010 Overall Grade” were removed.

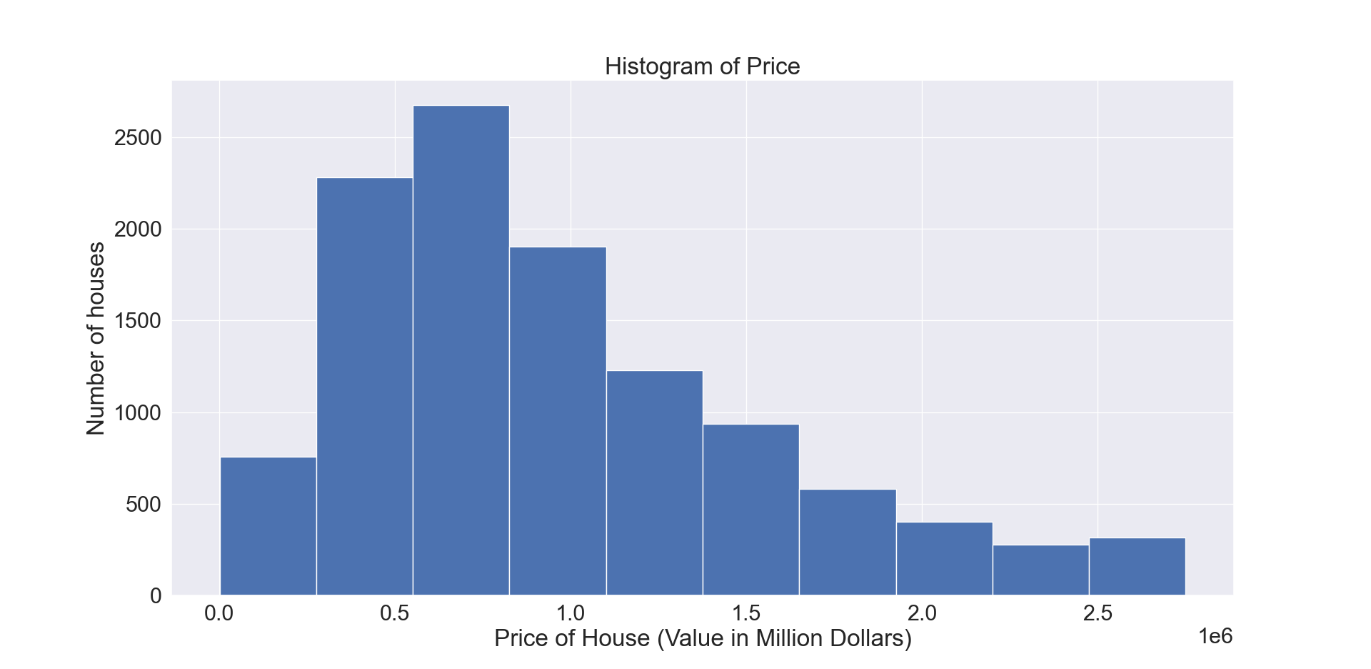
The total number of records, after combining additional datasets concludes to “11364”.

## Data Analysis

Before applying the Machine-Learning models, data analysis of the combined dataset with 11364 records and 33 variables is conducted. Out of the 33 variables, there are 27 numeric variables. The histogram of house prices is provided below.

X-Axis – Price of the house in terms of Million Dollars.

Y-Axis – Number of houses



From the histogram, it can be interpreted that there are a greater number of houses in the range between 250$ to 1M$. As the price value of the house increases the count is decreasing gradually.

The distribution of house prices based on the number of bedrooms (NoB) is represented in the graph below (Figure 2). As the NoB increases gradually from 0 to 15, the house prices also increase but beyond 15, the NoB has no impact on the price of a house. When the NoB is between 3 and 6, the house price appears to be at a higher value, i.e. in the range of 1.5 to 2.75 M$. This indicates that the NoB in a house has an impact on the house prices when the NoB is in the range of 0 to 7.

The distribution of house prices based on the number of baths (NoBa) is represented in the graph below (Figure 3). As the NoBa increases gradually from 0 to 6, the house prices also increase, but beyond 6, the NoBa have an unsymmetrical impact on the price of a house. When the NoBa is between 1 and 4, the house price appears to be at a higher value, i.e. in the range of 1.5 to 2.75 M$. This indicates that the NoBa in a house has an impact on the house prices when the NoBa is in the range of 0 to 6.

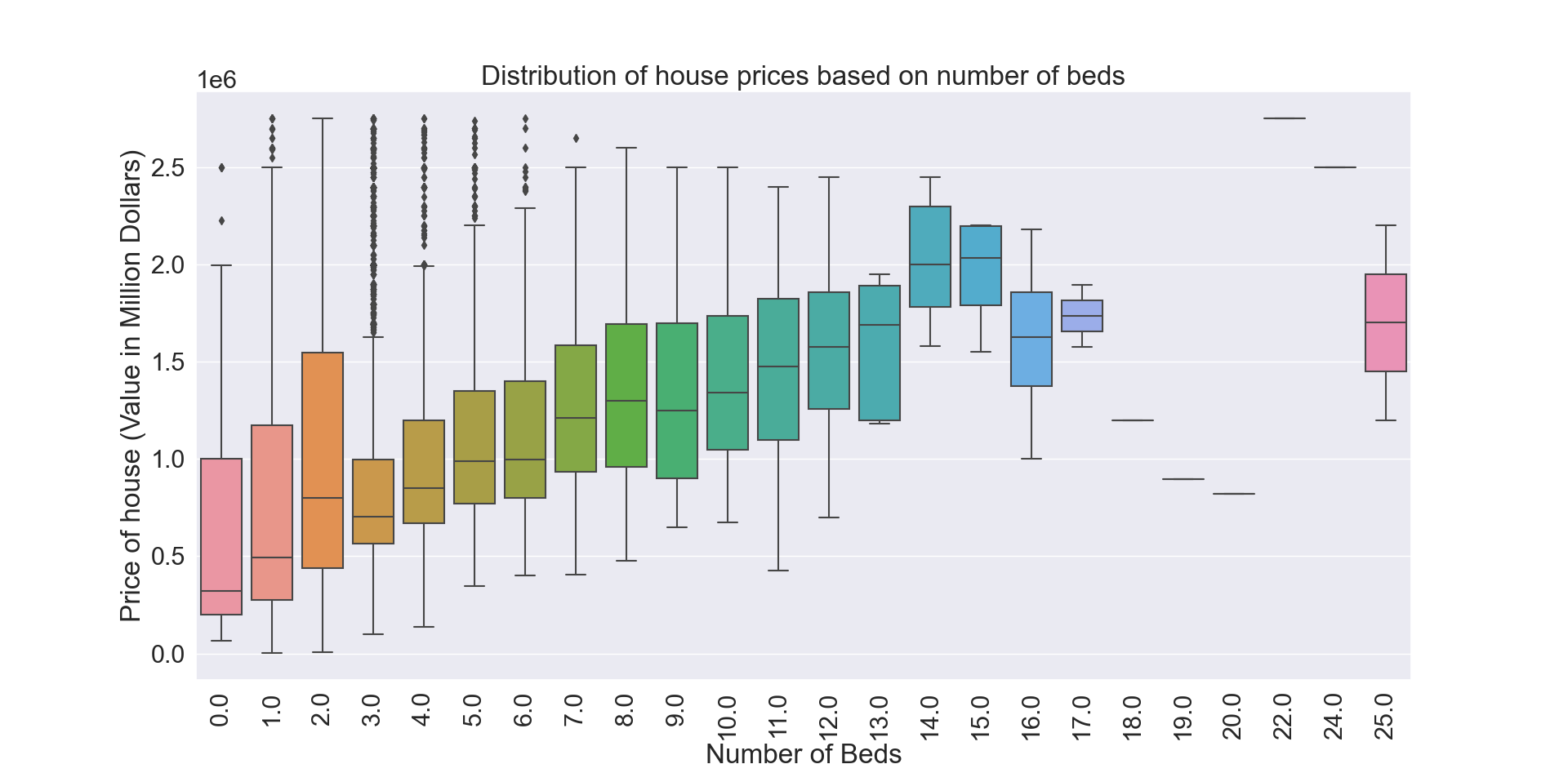


Figure 2

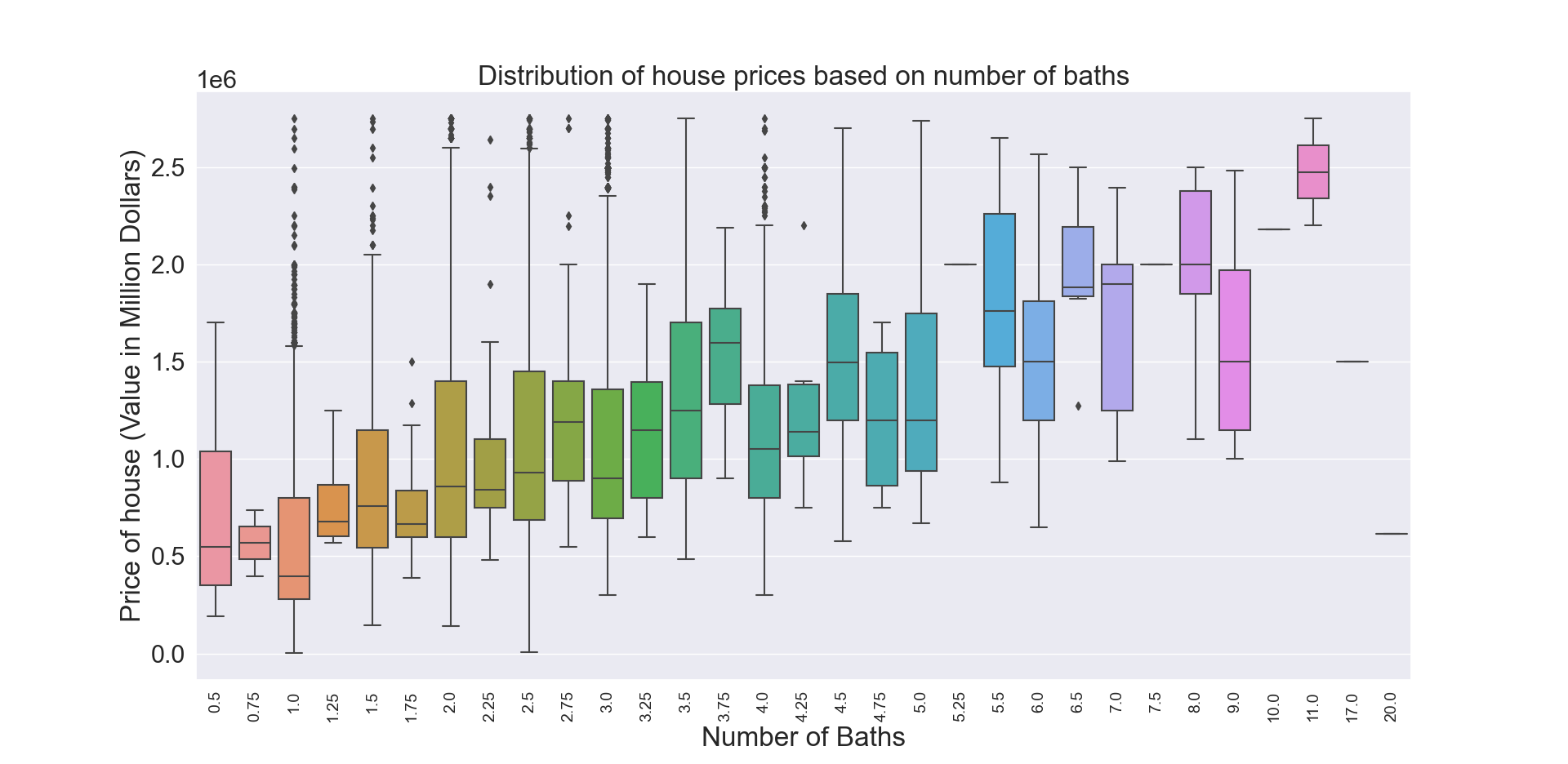
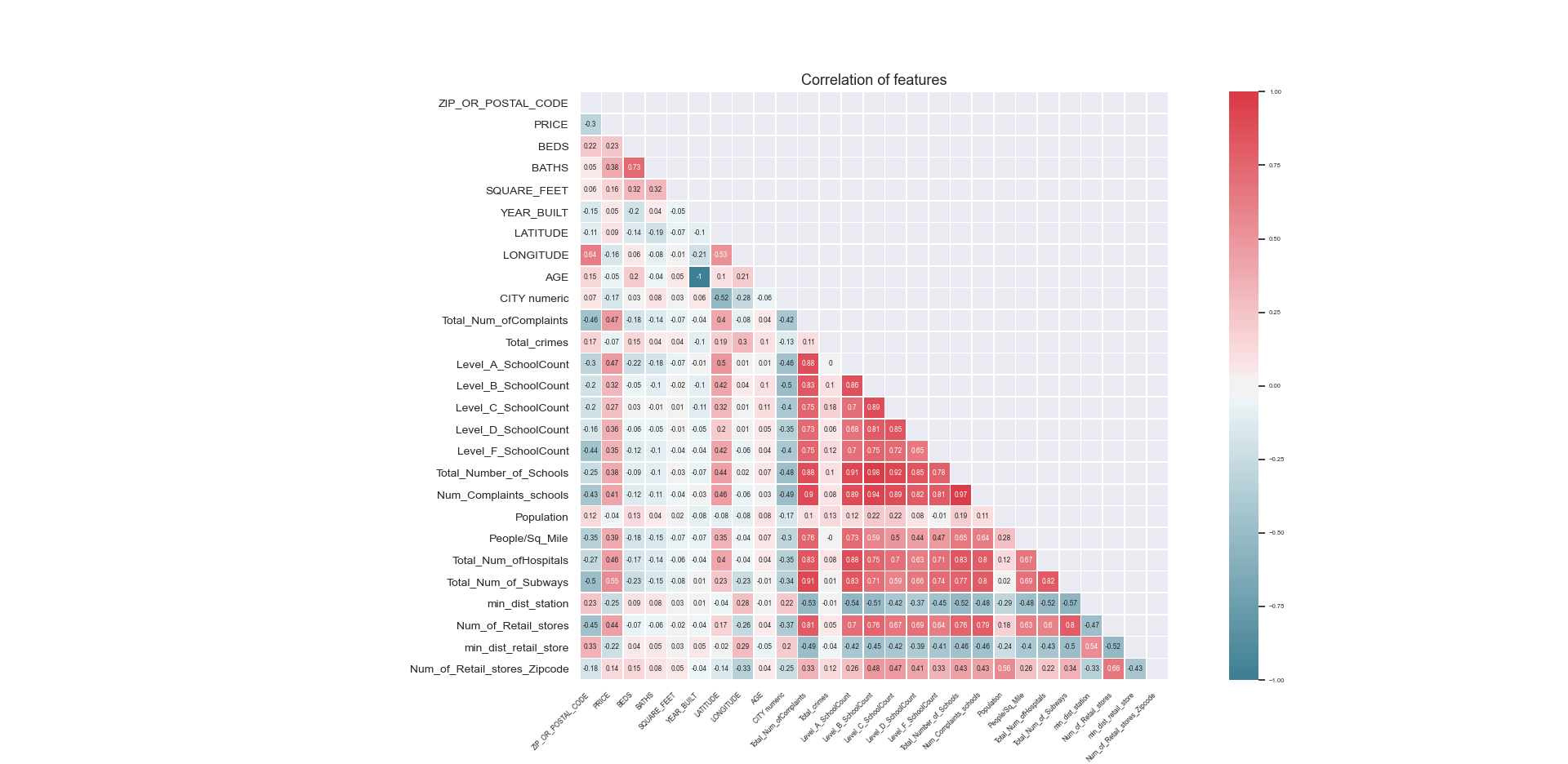


Figure 3

To evaluate the relation between features and the relation between features wit house of prices, the Pearson's correlation coefficient is applied. This is a statistic that measures the linear correlation between two variables X and Y. It has a value between +1 and -1, where 1 is a total positive linear correlation, 0 is no linear correlation, and -1 is a total negative linear correlation. (Pearson\_correlation\_coefficient, n.d.). The correlation coefficient is represented as *rxy*. Given paired data {(x1, y1), …... (xn, yn)} consisting of n pairs, rxy is defined as below equation:

In our dataset, each feature will be paired with another feature to find the paired data and the correlation coefficient is calculated. In the figure below the correlation of all the 27 numeric variables is shown.

gene

The following are the findings analysing the correlation of different features in the combined dataset.

1. The features with a significant correlation to the ‘Price of house’ are ‘Total Number of Complaints’, ‘Level A school count’, ‘Total Number of Subways’, ‘Number of Retail Stores’, ‘Total Number of Hospitals. This indicates that the price of the house is dependent on external features like proximity to school, hospitals, and public transport stations.
2. The feature ‘Total Number of complaints’ are positively correlated with high value to ‘Total Number of Subways’, ‘Total Number of Schools’, and ‘Total Number of Hospitals’. This implies that subways, schools, hospitals are more crowded places, which in turn increases the crime incidents near to those places.
3. The variables “Minimum distance to station” and “Minimum distance to retail store” are related to the location of the house. Therefore, the correlation between these variables and other variables are insignificant.
4. The features “Total Crimes”, “Age of the house”, “Minimum distance to station”, and “Minimum distance to retail store” have a negative correlation to the price of the house. This implies that when one value increases the price of the house decreases. That is people prefer houses with fewer crimes, recently built, nearer to a retail store and subway station. But the feature “people per square mile” is positively correlated to the price of the house. This implies that in the areas nearer to downtown or city, the people per square mile is higher and therefore the price of the house is also at a higher rate.

## Data Standardisation

In the combined dataset every feature has varying degrees of magnitude, range, and units. To normalize the range of independent variables or features of data feature scaling is implemented. Feature Scaling or Standardization is a step of data Pre-Processing which is applied to independent variables or features of data. It helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm. In this project, the Standardization technique where the values are centred around the mean with a unit standard deviation is used. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation (How and where to apply feature scaling, n.d.).

The general method of calculation is to determine the distribution mean and standard deviation for each feature. Next, we subtract the mean from each feature. Then we divide the values (mean is already subtracted) of each feature by its standard deviation (Feature Scaling, n.d.). The equation is as given below.

# **Why Machine Learning Models?**

Machine Learning provides smart alternatives to analyzing vast volumes of data. By developing fast and efficient algorithms and data-driven models for real-time processing of data, Machine Learning can produce accurate results and analysis.

While training a model is a key step, how the model generalizes on unseen data is an equally important aspect that should be considered in every machine learning pipeline. To identify whether the model actually works and, consequently, if the predictions can be trusted or not.

Methods for evaluating a model’s performance are divided into 2 categories: namely, holdout and Cross-validation. Both methods use a test set (i.e. data not seen by the model) to evaluate model performance. To evaluate the built model the trained dataset is not used. This is because the model will simply remember the whole training set, and will therefore always predict the correct label for any point in the training set.

Single fold validation is a technique that involves partitioning the original observation dataset into a training set, used to train the model, and an independent set used to evaluate the analysis.

In this project after applying the standardization to the combined house and additional dataset, it is split into two sections X and Y. where X is the numeric values removing the Price of the house and Y is the Price column. The X and Y dataset is split into two subsets, one for training the machine learning model and another for testing the model.

## Model Evaluation Metrics

Model evaluation metrics are required to quantify model performance. In this project following are the model evaluation metrics used.

1. Coefficient of Determination (r2)

This is a statistic that will provide information about the goodness of fit of a model i.e. describing how well a statistical model fits a set of observations.  In regression, the R2, coefficient of determination is a statistical measure of how well the regression predictions approximate the real data points (Coefficient\_of\_determination, n.d.). An R2 of 1 indicates that the regression predictions perfectly fit the data. The best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get an R^2 score of 0.0 (sklearn.metrics.r2\_score., n.d.). The equation for the coefficient of determination is as provided below. If  is the predicted value of the *i*-th sample and  is the corresponding true value for total *n* samples, the estimated R2 is defined as:

Where

Scores of all outputs are averaged with uniform weight. If an array of shape (outputs,) is passed, then its entries are interpreted as weights and a weighted average is returned.

1. Adjusted Coefficient of Determination (r2)

The adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. This shows how well the data points fit a curve or line but adjusts for the number of terms in a model. The adjusted R-squared increases only if the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected by chance. The adjusted R-squared can be negative, but it’s usually not.  It is always lower than the R-squared. The formula for the Adjusted R-squared is as below.

Where R is the Coefficient of determination, *n* is the number of points or records in the sample; *k* is the number of independent regressors, i.e. the number of variables in the model excluding the constant.

1. Mean Absolute Error (MAE)

The Mean absolute error is a measure of errors between paired observations expressing the same phenomenon. It is one of the ways of comparing forecasts with their eventual outcomes (Mean Absolute Error, n.d.).  MAE is conceptually simpler and more interpretable than RMSE. MAE does not require the use of squares or square roots. MAE is calculated as below equation:

Where yi is the predicted value and xi is the true value and ‘n’ being the number of records.

In this project, the true value is considered as the actual sale value of the house retrieved from Redfin.

1. Mean Squared Error (MSE)

 The mean squared error (MSE) or mean squared deviation (MSD) of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value (Mean Squared Error, n.d.). The MSE is a measure of the quality of an estimator—it is always non-negative, and values closer to zero are better. MSE is calculated as below equation:

Where n- sample data points, Yi is the true value of i-th sample, being the predicted value for the i-th sample.

1. Root Mean Squared Error (RMSE)

The Root Mean Squared Error (RMSE) represents the square root of the second sample moment of the differences between predicted values and observed values (Root-mean-square\_Error, n.d.). RMSE is always non-negative, and a value of 0 (rarely achieved in practice) would indicate a perfect fit to the data. RMSE is the square root of the average of squared errors. The effect of each error on RMSE is proportional to the size of the squared error; thus, larger errors have a disproportionately large effect on RMSE.

1. Accuracy

The accuracy variable is calculated by converting the Coefficient of Determination (r2) to a percentage value.

1. Percentage Error

The percentage error metric is a measure by how much the predicted value is distant from the true value. The error value is calculated by fixing a percentage difference of true value is allowed. The absolute difference between the true and predicted value is calculated. The count of records with the absolute difference greater than the percentage of true value is calculated. The count is divided by the total number of records to convert as a percentage. The higher value of percentage indicates that the prediction is having greater deviation from true value i.e., the model’s accuracy is low.

Yi is the true value of i-th sample, being the predicted value for the i-th sample.

1. Percentage Error with Sigmoid Function

This metric is an alternative method to calculate the percentage error metric. In the ‘Percentage Error’ metric, it can be noticed that the percentage value is a fixed value for all the records, which is independent of the price of the house. In this ‘percentage error with sigmoid function’ calculation method, the percentage for each record will vary depending upon the true price value. The reason for this metric usage is explained with an example below.

For a house price of “2.5M$’ if the percentage deviation allowed is 10%; the absolute difference of true value and the predicted value is allowed to fall in the range of “2.25M $” to “2.75M$”. This indicates that there is a quarter of a Million-dollar difference and in many cases, this deviated value is equivalent to the true price of another house itself.

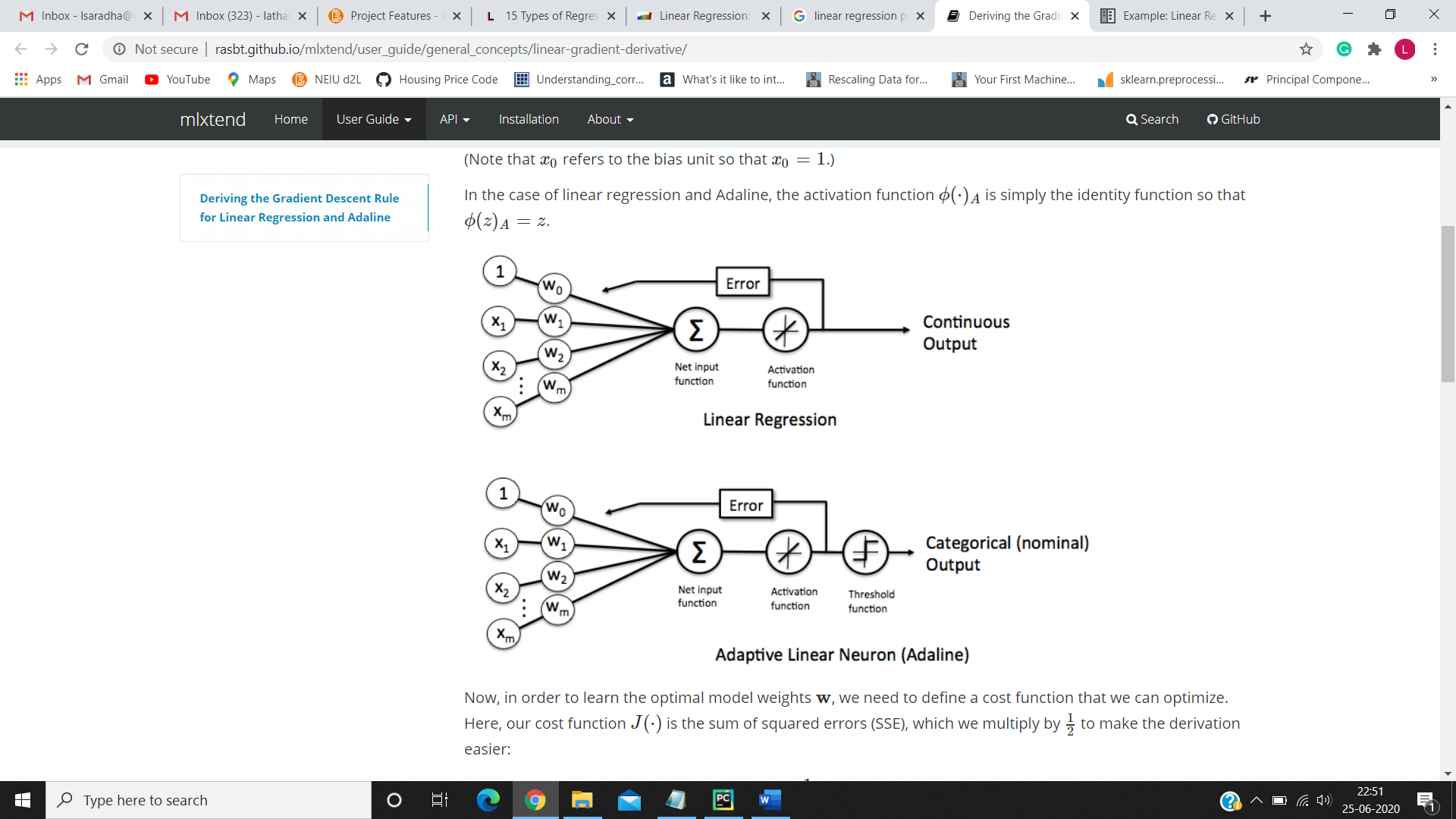
Therefore, to avoid a fixed percentage deviation for every record, the sigmoid function is used to calculate the percentage deviation for each record which will be depending upon its true price value. For high prices, a small percentage deviation is calculated and for small prices, a higher percentage deviation is computed. The lower value of percentage error indicates that the prediction is having lesser deviation from true value i.e., the model’s accuracy is high.

Yi is the true value of i-th sample, being the predicted value for the i-th sample.

# **Prediction Models**

## **Linear Regression**

Linear regression is a linear approach to modeling the relationship between a dependent variable (the price of the house) and one or more [independent variables](https://en.wikipedia.org/wiki/Independent_variable)(other numeric variables) (Linear Regression, n.d.). For more than one explanatory variable, the process is called multiple linear regression. In this project, the multiple linear regression is implemented. The equation for multiple linear regression is listed below.



Where y is the dependent variable to be estimated, and X are the independent variables and is the error term.’s are the regression coefficients.

In this project, ‘y’ is the Price of the house to be estimated and X is denoted by the different features.

By comparing the model’s prediction with true value, the model performance is compared to find the best model based on minimum error. Different types of linear Regression are calculated. -Ridge, Lasso, Positive Lasso, and Elastic Net.

The cost function is used to define and measure the error of the model. The cost function of Ridge Regression is provided below. One of the hyperparameters of Ridge Regression is ‘alpha’.

A new term is added from the normal linear regression, this is the penalty. λ, the penalty, is denoted by the alpha parameter in the ridge function. So, by changing the values of alpha, the penalty term is controlled. Higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients is reduced.

The cost function of Lasso Regression (Least Absolute Shrinkage and Selection Operator) is provided below. Mathematics behind lasso regression is quite similar to that of ridge only difference being instead of adding squares of theta, we will add the absolute value of W.

λ is the hypermeter, whose value is equal to the alpha in the Lasso function.

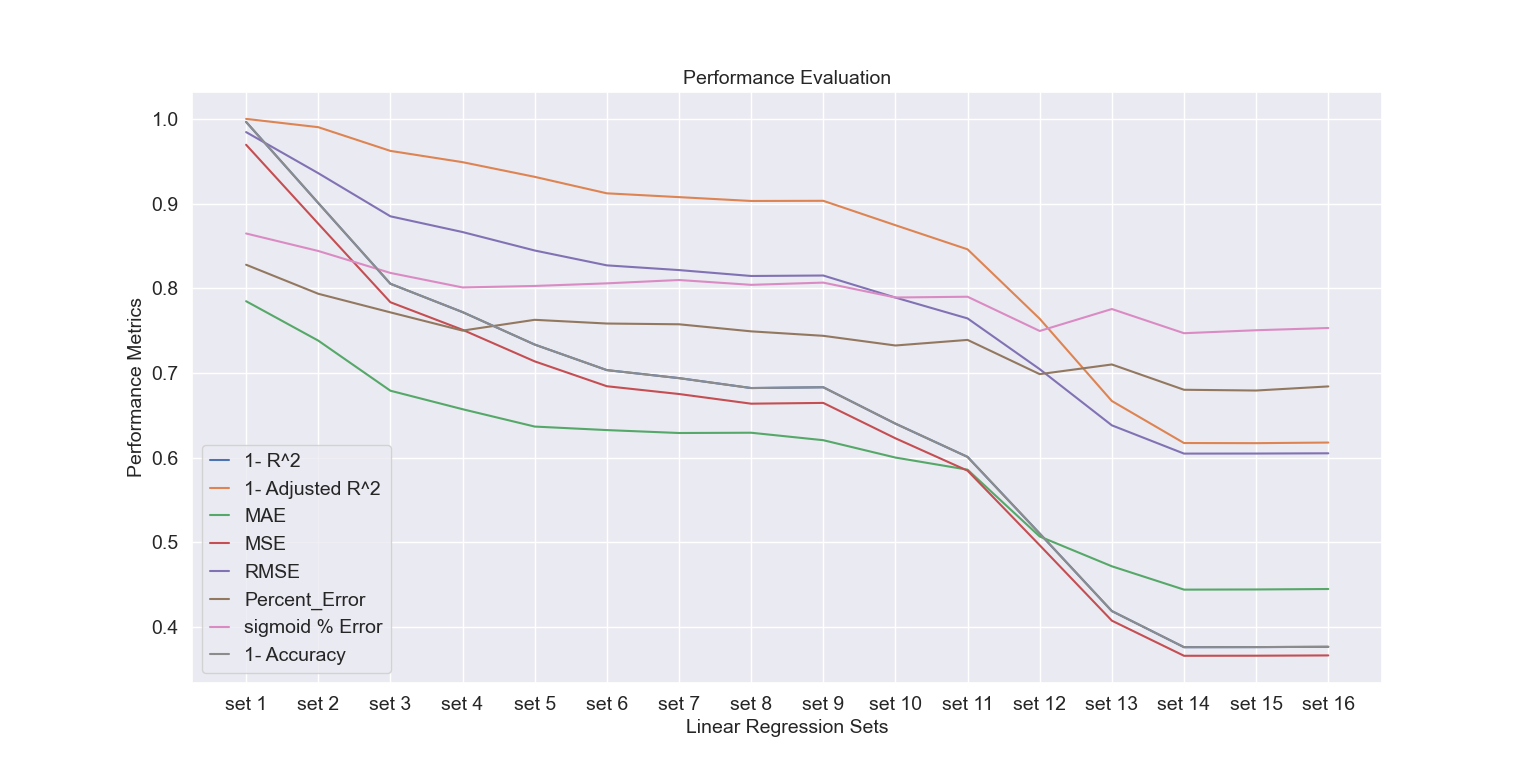
The cost function of Positive Lasso Regression is provided below. It is similar to as lasso provided above. The only difference is all the coefficients wi is positive.

The cost function of Elastic Net Regression is provided below. It is similar to as lasso provided above. There is an additional term to the elastic net, which is a factor of alpha.

### Python Implementation

The training dataset is fit using multiple linear regression. Using the test dataset, the model is predicted and performance metrics are evaluated and provided below. To evaluate the performance results and to find out which combination of features yields the best result, all the combination of numeric variables is carried out. For each combination of the feature variables, the performance metrics of Linear Regression was calculated. Out of the 351 combinations possible from the 26 numeric variables, the following graph represents a comparison between some combinations.

The X-axis indicates the ‘Feature Set’, which is the combination of variables. The Y-Axis indicates the value from 0 to 1. Since the ideal combination of variables must have lower values of errors such as RMSE, MSE, MAE, Percentage Error, and Percentage error with sigmoid function; the R2 score and adjusted R2 score is converted to (1- R2 score) and (1- adjusted R2 score) respectively. This will formulate the intention to minimize all the metric values. Feature Sets used for the graphical representation are as below.



The sets with column lists are provided in Appendix as Table I. From the graph it is understandable that set 16 is having the least error and most accuracy.

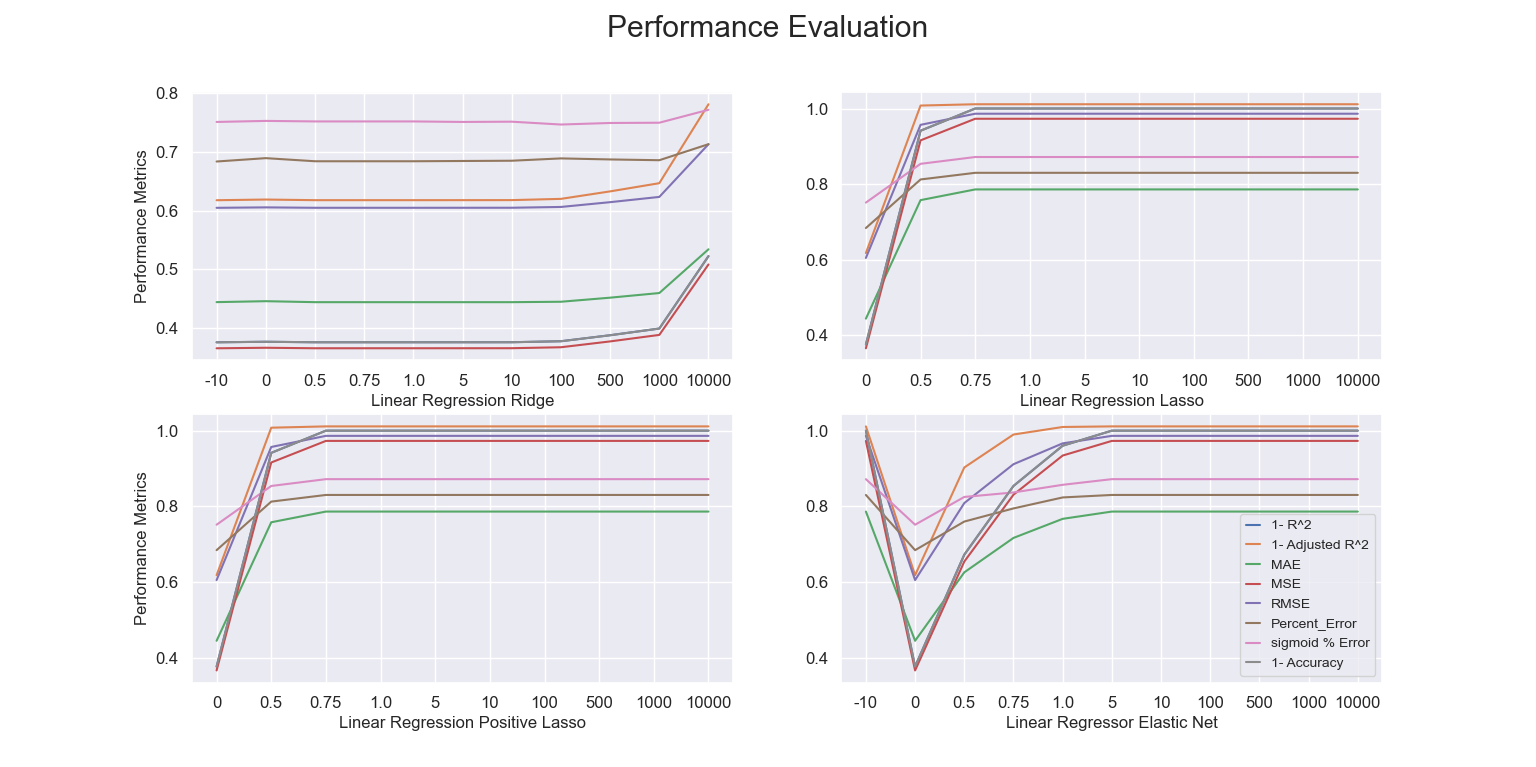
From the table at Appendix ([Table I](#_Appendix)), set 16 indicates the columns list as ['BEDS', 'BATHS', 'SQUARE\_FEET', 'YEAR\_BUILT', 'LATITUDE', 'LONGITUDE', 'AGE', 'CITY numeric', 'Total Numb of Complaints', 'Total crimes', 'Level A School Count', 'Level B School Count', 'Level C School Count', 'Level D School Count', 'Level F School Count', 'Total Number of Schools', 'Numb of Complaints schools', 'Population', 'People/Sq. Mile', 'Total Numb of Hospitals', 'Total Numb of Subways', 'min dist. station', 'Numb of Retail stores', 'min dist. retail store', 'Numb of Retail stores Zip code'].

The linear regression model was applied to the combined data set without standardization. The price values are converted to logarithm to the base of e and fit with the model and predicted. The below table provides a relation between the performance metrics with and without standardization.

|  |  |  |
| --- | --- | --- |
| **Metric** | **With Standardization** | **Without Standardization (logarithmic value)** |
| R2 | 0.6241 | 0.5969 |
| Adjusted R2 | 0.3829 | 0.349 |
| MAE | 0.444 | 0.3156 |
| MSE | 0.3657 | 0.176 |
| RMSE | 0.6047 | 0.419 |
| Percentage Error | 0.67927 | 0.687 |
| % error with sigmoid function | 0.7514 | 0.730 |

In the table, it can be identified that R2 and adjusted R2 is not much affected by the standardization. But the metrics MAE and MSE are reduced for the model without standardization. This is due to the calculation of MSE, MAE, and RMSE. In the error metric calculations, the factor of ‘n’, which is the number of records/data points is included, but there is no factor of ‘n’ in the calculation of R2 and Adjusted R2.

The below graph describes the performance difference between different types of linear regression such as Ridge, Lasso, Positive Lasso, and Elastic Net with different values of alpha.



The graph indicates that when alpha is increasing or decreasing from zero, the errors are increasing. This concludes that regularization has less or no effect on this dataset.

## **SVR Regression**

SVR or Support Vector Machine is a linear model for classification and regression problems. Support-vector machines (SVMs, also known as support-vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis (Support Vector Machine, n.d.)

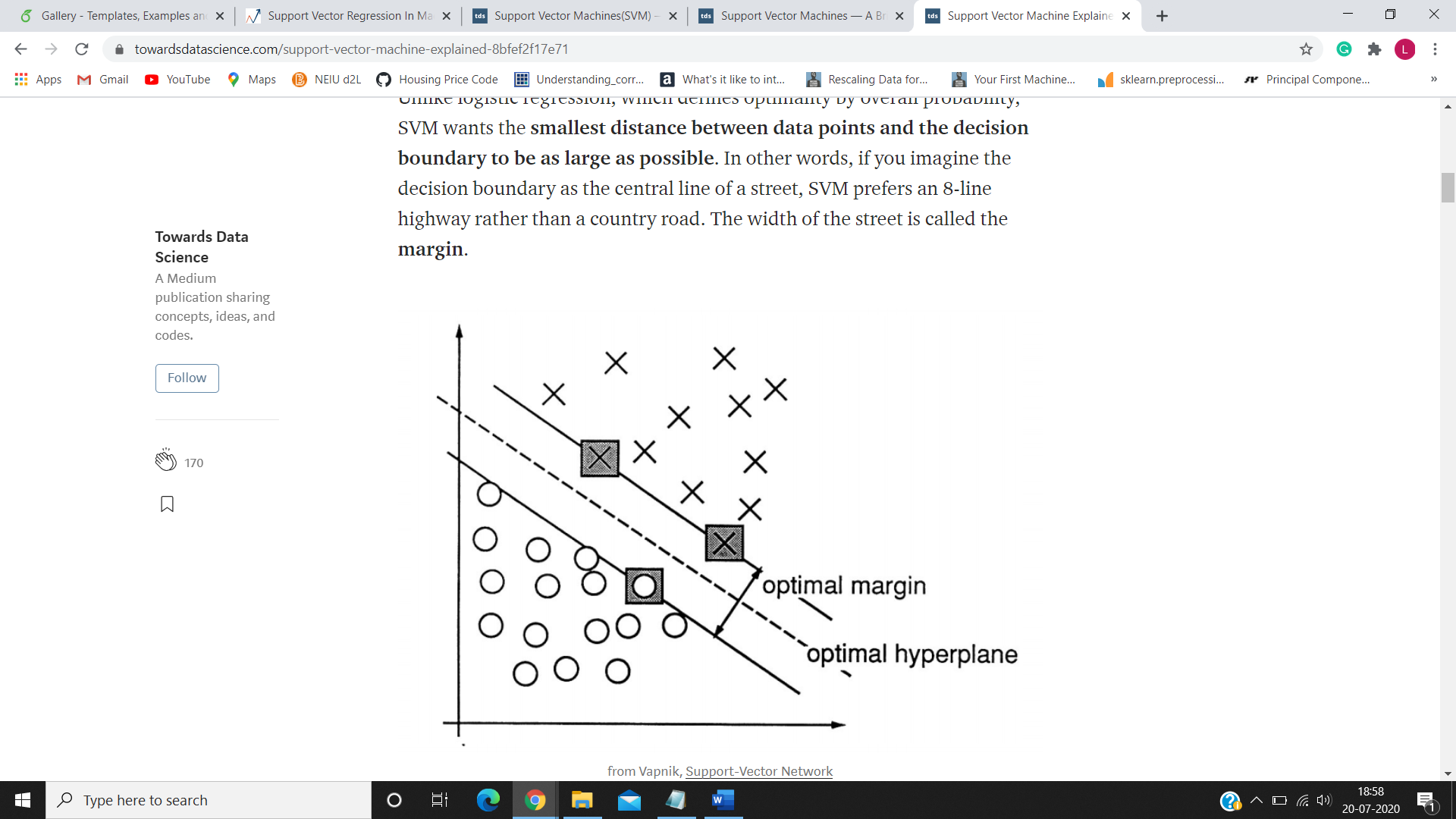
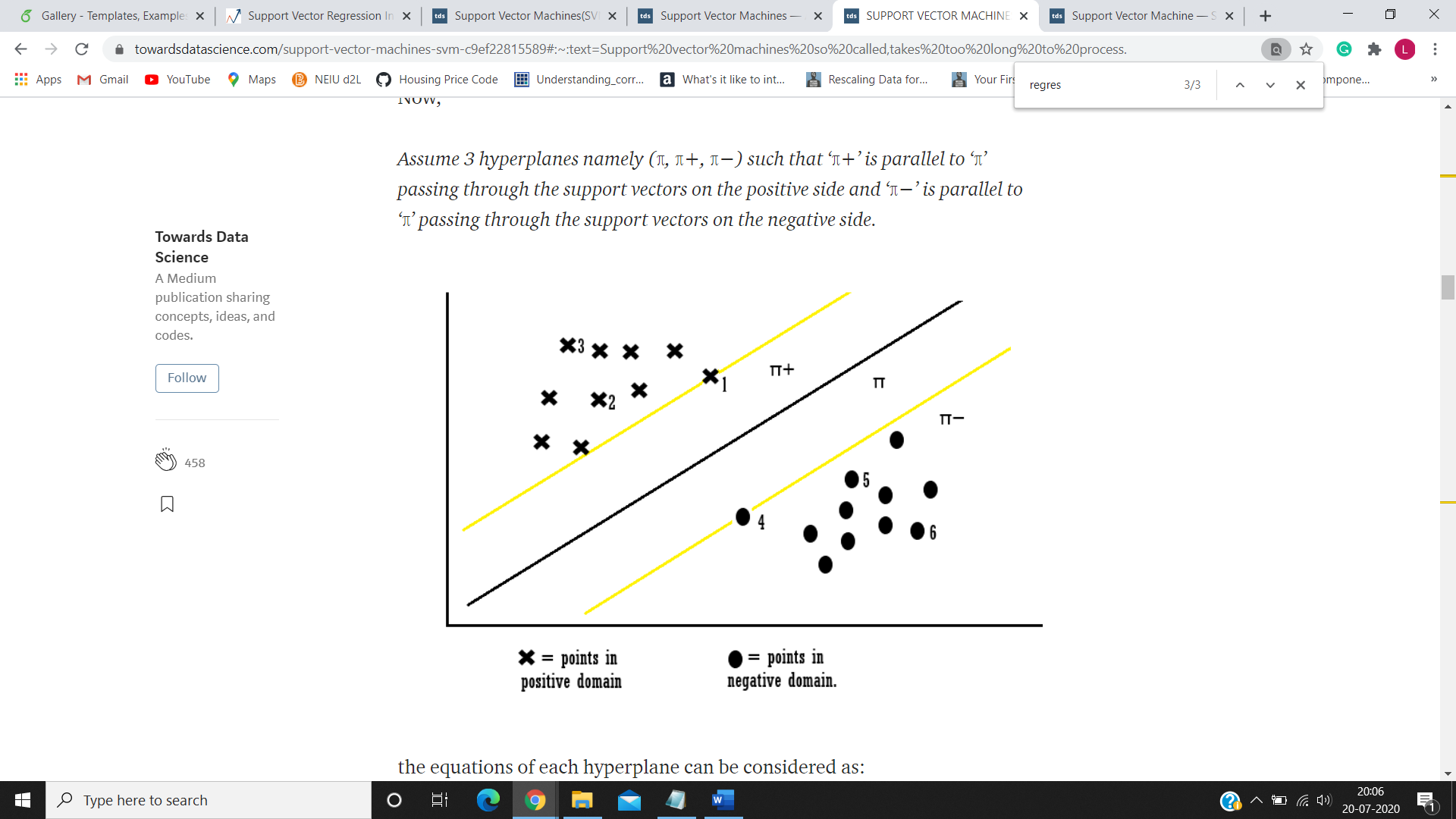
The main objective of SVM is to find the optimal hyperplane, a plane that linearly divides the n-dimensional data points into two components by maximizing the margin. In the case of regression, a margin of tolerance (epsilon) is set in approximation to the SVM (Support Vector Machine Towards Data Science, n.d.).

At first, an approximation of what SVMs do is to find a separating line (or hyperplane) between data of two classes. This hyperplane is a linear separator for any dimension; it could be a line (2D), plane (3D), and hyperplane (4D+). SVM wants the smallest distance between data points and the decision boundary to be as large as possible. The points closest to the hyperplane are called the *support vector points* and the distance of the vectors from the hyperplane is called the *margins*. The following figure represents the margin and data points.

The equation of the hyperplane in the ‘n’ dimension is given by,

=

Where b is the biased term (w0) and X being the variables.



Assume 3 hyperplanes namely (π, π+, π−) such that ‘π+’ is parallel to ‘π’ passing through the support vectors on the positive side and ‘π−’ is parallel to ‘π’ passing through the support vectors on the negative side. the equations of each hyperplane can be considered as:

For each example,

*if Yi (WT \* Xi +b) ≥ 1: then Xi is correctly classified*

*else: Xi is incorrectly classified.*

So, this type of SVM is called hard margin SVM (since it has very strict constraints to correctly classify every datapoint).

To skip a few outliers and to provide consideration to real-life scenarios, a new Slack variable (ξ) which is called Xi is introduced. This is a Soft Margin.

if ξi= 0, the points can be considered as correctly classified.

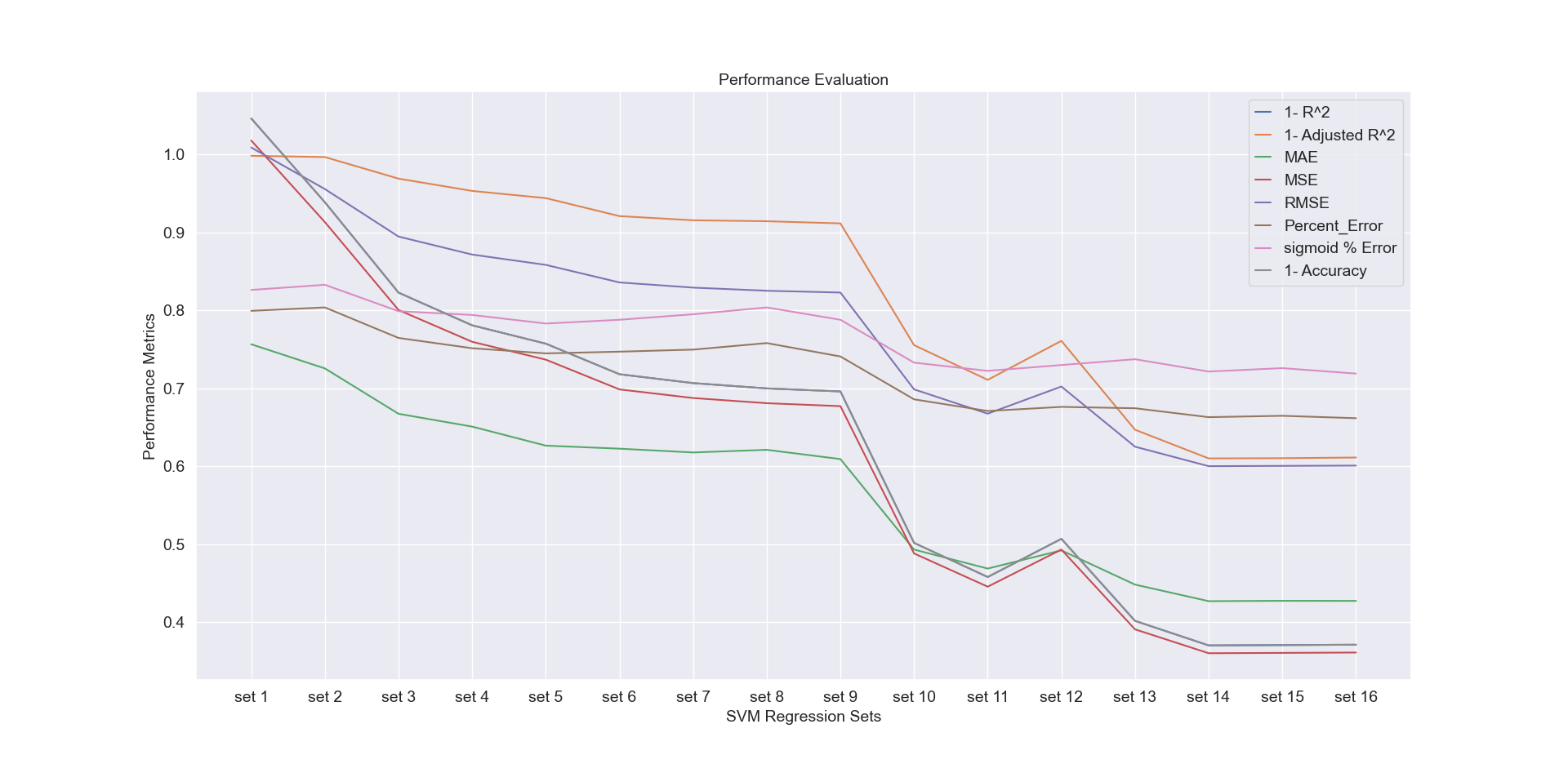
else: ξi> 0, Incorrectly classified points.

The degree of tolerance is defined as the tolerance(soft) to be given whenfinding the decision boundary is an important hyper-parameter for the SVM (both linear and nonlinear solutions)*.*Bigger the value, the more penalty SVM gets when it makes misclassification.

In this project, SVM is used with linear kernel and the penalty term is set to be 1.0.

### Python Implementation

The results of the SVM algorithm with column sets are provided below. The graph indicates that the SVM model has a similar result to that of Linear Regression. But the accuracy is higher in SVM than in Linear. Sets 14,15 and 16 have lease error and highest accuracy.

****

The below table provides a relation between the performance metrics of SVM Regression with and without standardization. When R2 value is compared, with standardizing the data points, accuracy is increasing. SVM tries to maximize the distance between the separating plane and the support vectors. If one feature (i.e. one dimension in this space) has very large values, it will dominate the other features when calculating the distance. This issue can be avoided using standardization.

|  |  |  |
| --- | --- | --- |
| **Metric** | **With Standardization** | **Without Standardization (logarithmic value)** |
| R2 | 0.629499 | 0.515269 |
| Adjusted R2 | 0.389552 | 0.25832 |
| MAE | 0.427072 | 0.333097 |
| MSE | 0.360464 | 0.211772 |
| RMSE | 0.600386 | 0.460187 |
| Percentage Error | 0.66476 | 0.675319 |
| % error with sigmoid function | 0.721073 | 0.736912 |

## **K- Nearest Neighbor**

K nearest neighbor is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). k-nearest neighbor is used for both regression and classification problems and in this algorithm, there is no training process, the entire data set is used for predicting/classifying new data (K Nearest Neighbours, n.d.).

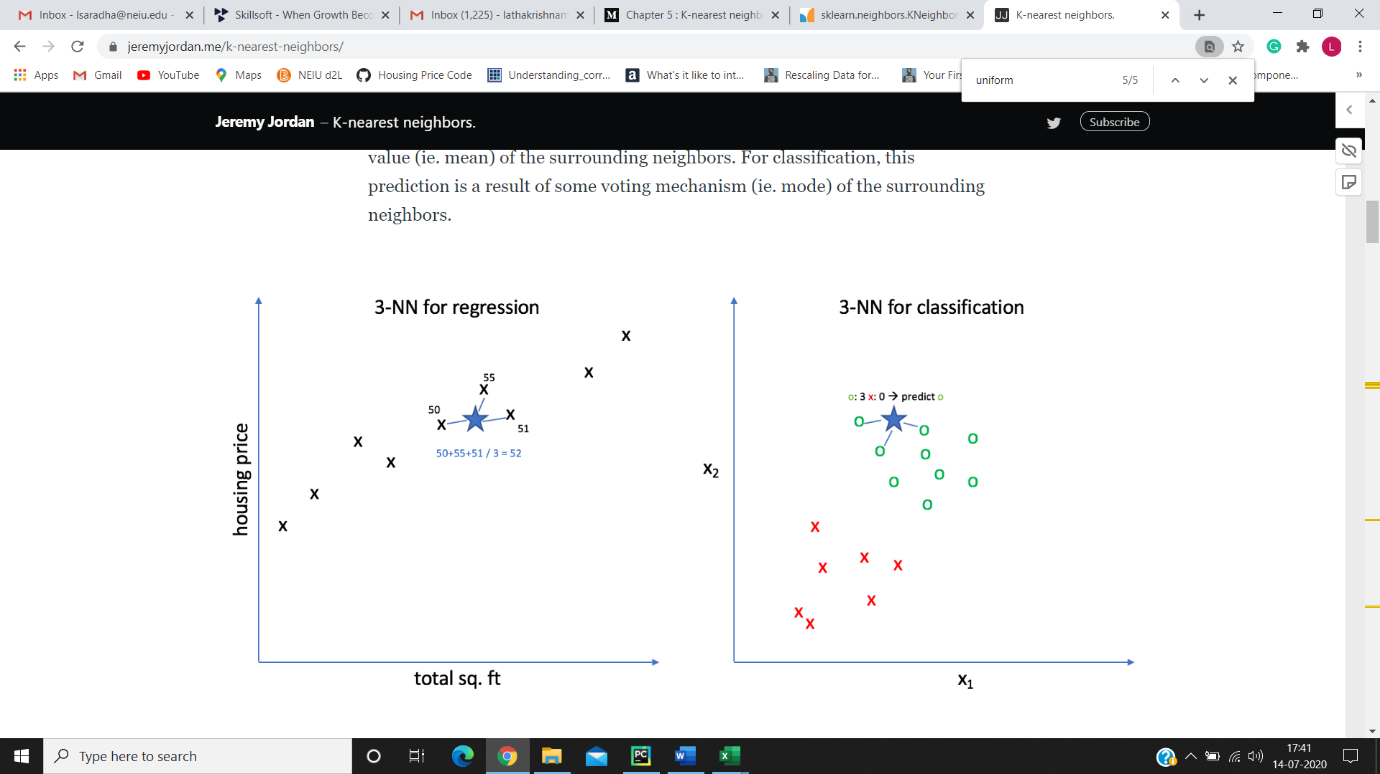
When a new data point is given, it calculates the distance from the new data point to all other points in the data-set. Then depending on the *K*value, it identifies the nearest neighbor(‘s) in the data set. If *K*=1 then it takes the minimum distance of all points and classifies as the same class of the minimum distance data point. If k>1 then it takes a list of K minimum distances of all data points. For regression, it takes the average of all values in the list.

While using distance as a measure of similarity, we're implicitly constraining our model to weight all features equally; distance in the x1 dimension is on the same scale as distance in the xn dimension. In this project, all points in each neighbourhood are weighted equally and the number of neighbors is set to 6.

To find the k-nearest neighbors, some measure of "closeness" is defined (in other words, distance). There are many ways such as Euclidean distance, Manhattan Distance can be used as a measure of the distance between two points. The default distance is Minkowski distance, which is the generalization of both Euclidean and Manhattan distance (K-nearest\_neighbors\_algorithm, n.d.). The equation is provided below.

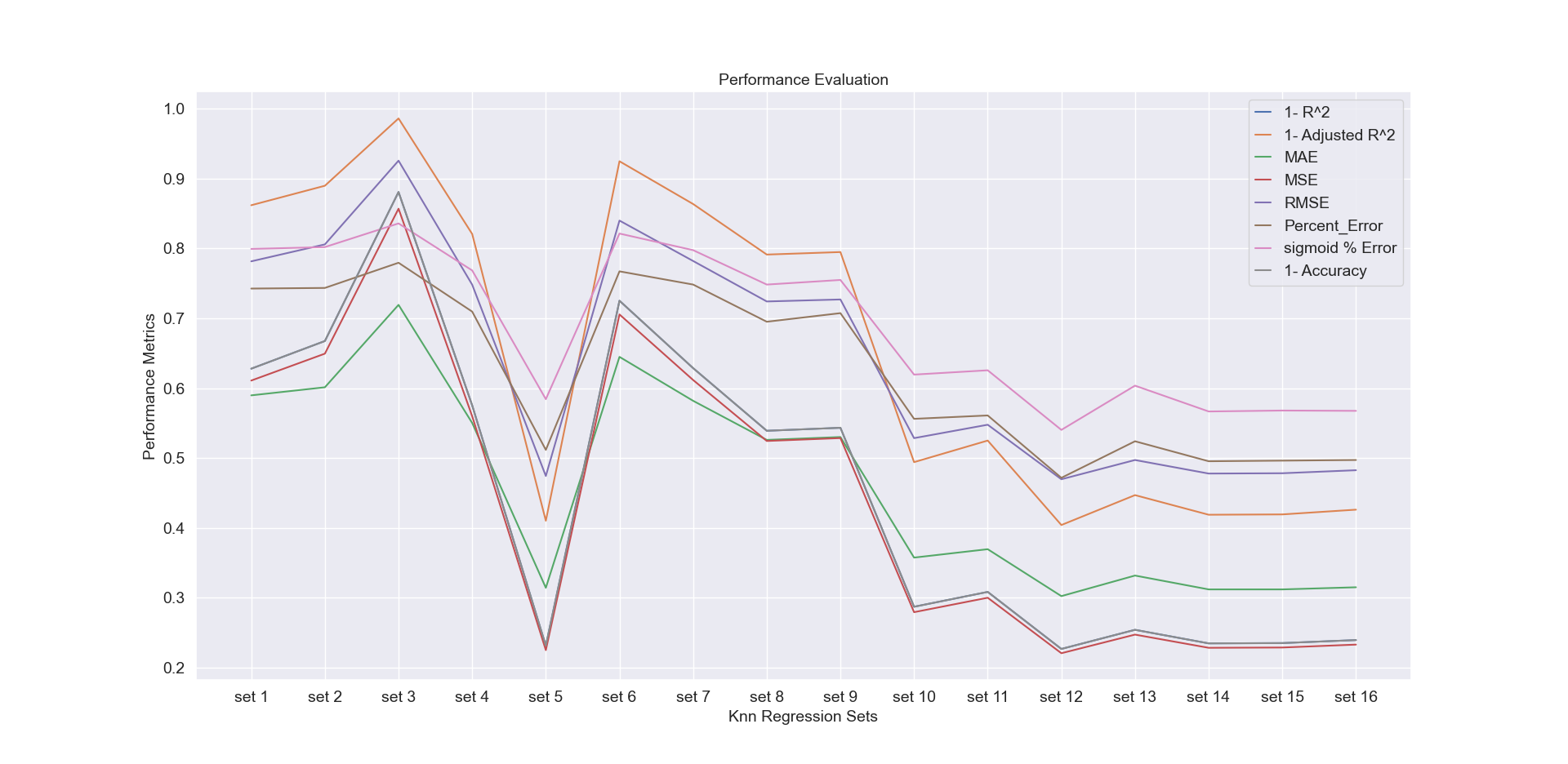
n- Number of dimensions. x- data point from the set and y- data point to be predicted. With q=2, the equation will be converted to Euclidean Distance.

Find the sample picture of KNN Regression with k=3.



### Python Implementation

The results of the K-nearest neighbor algorithm with column sets are provided below.



The Below table provides a relation between the performance metrics of KNN with and without standardization. It indicates that standardization has minimal effect on the Knn model.

|  |  |  |
| --- | --- | --- |
| **Metric** | **With Standardization** | **Without Standardization (logarithmic value)** |
| R2 | 0.781225 | 0.78107 |
| Adjusted R2 | 0.608071 | 0.609555 |
| MAE | 0.2996 | 0.220068 |
| MSE | 0.212848 | 0.095648 |
| RMSE | 0.461354 | 0.309269 |
| Percentage Error | 0.472943 | 0.530136 |
| % error with sigmoid function | 0.542455 | 0.600088 |

## **Tree-Based Models**

## **Random Forest Regression**

Random forest is a Supervised Learning algorithm that uses the ensemble learning method, a method where multiple learning algorithms combined to obtain predictive performance (Ensemble Learning, n.d.), for classification and regression. Random Forest is operated by constructing a multitude of decision trees at training time and outputting the mean prediction of the individual trees (Random Forest, n.d.). A random forest is a meta-estimator (i.e. it combines the result of multiple predictions) which aggregates many decision trees, with some helpful modifications:

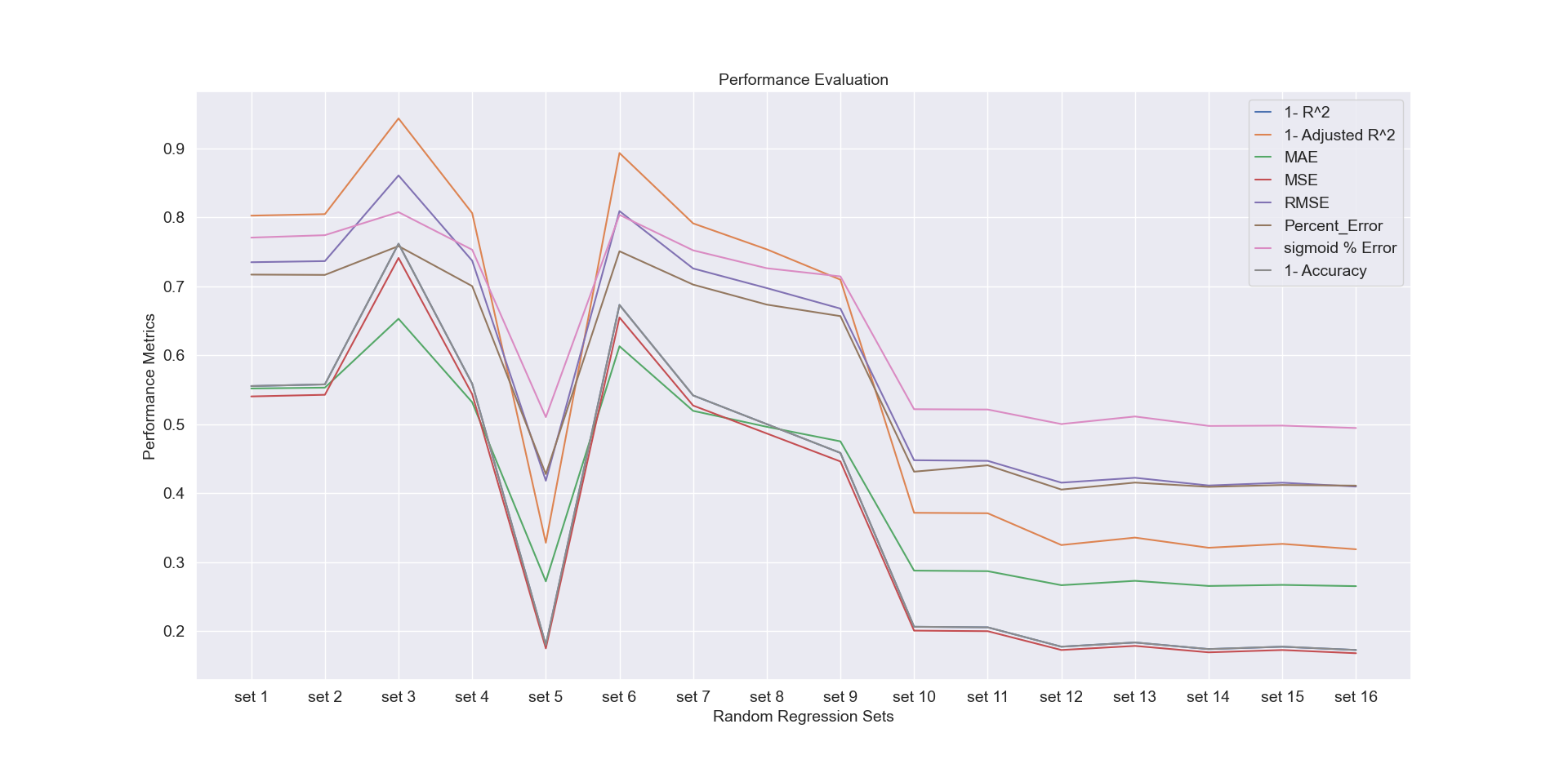
1. The number of features that can be split on at each node is limited to some percentage of the total (which is known as the **hyperparameter**). This ensures that the ensemble model does not rely too heavily on any individual feature, and makes fair use of all potentially predictive features.
2. Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents **overfitting**.

The above modifications help prevent the trees from being too highly correlated. When using the Random Forest Algorithm to solve regression problems, the mean squared error is used (MSE) to determine how data branches from each node.

### Python Implementation

The steps are applied as in Linear Regression. The results are provided below. The sets with column lists provided in Appendix as Table I are used.

The graph indicates that set 5,14,15,16 have the least error and highest accuracy. The performance of set 14,15,16 is similar to that of Linear Regression. But set 5, with columns list as ['BATHS', 'SQUARE\_FEET', 'YEAR\_BUILT', 'LATITUDE', 'LONGITUDE', 'AGE', 'CITY numeric'], have least error similar to the Set 16 with all the features includes. This indicates that with the house features alone, without involving the environmental properties, Random Forest Regression provides better performance in predicting the prices.

The below table provides a relation between the performance metrics of Random Forest Regression with and without standardization.

|  |  |  |
| --- | --- | --- |
| **Metric** | **With Standardization** | **Without Standardization (logarithmic value)** |
| R2 | 0.827317 | 0.876788 |
| Adjusted R2 | 0.680943 | 0.766391 |
| MAE | 0.263154 | 0.163672 |
| MSE | 0.168005 | 0.05383 |
| RMSE | 0.409884 | 0.232013 |
| Percentage Error | 0.400352 | 0.403432 |
| % error with sigmoid function | 0.485262 | 0.490101 |

The table indicates that the R2 increases by 5% when Random Forest regression is applied to the dataset without standardization. This is because Random Forest is a tree-based model and hence it does not require feature scaling.

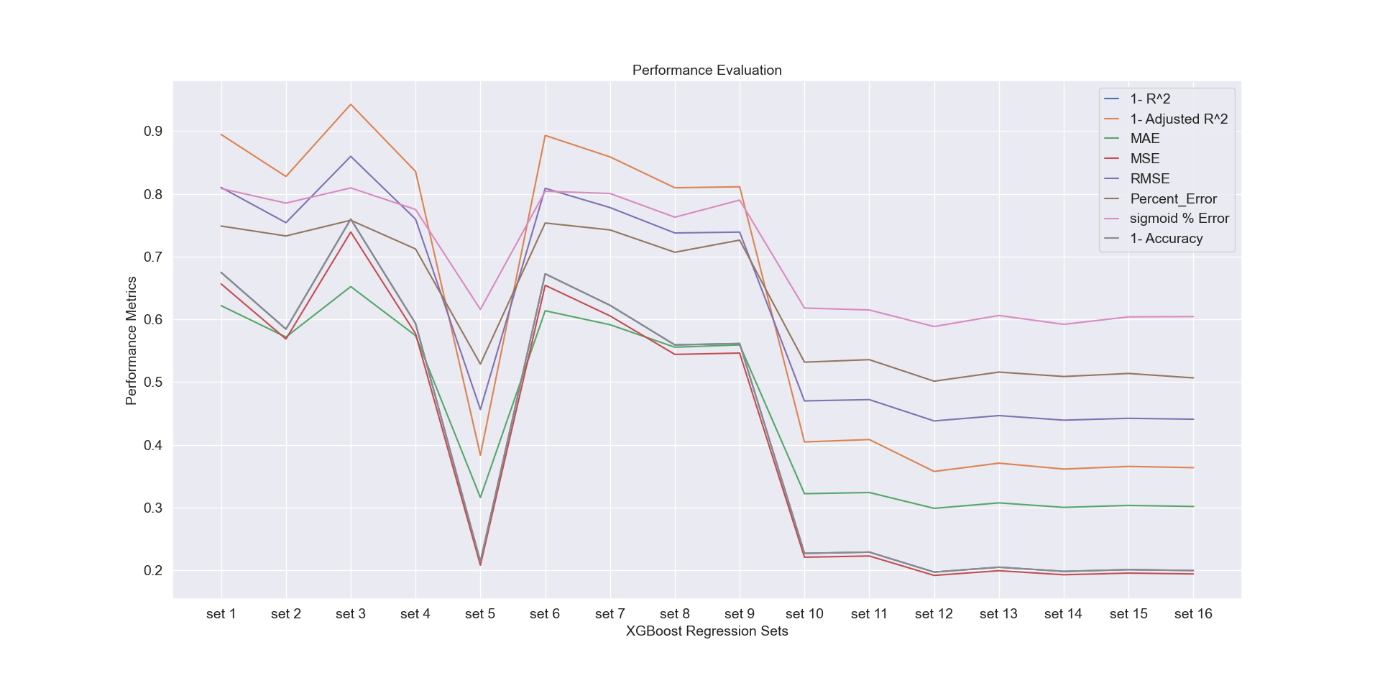
## **XGBoost Regression**

XGBoost is an implementation of gradient boosted decision trees designed for speed and performance. XGBoost stands for e**X**treme **G**radient **B**oosting (xgboost-applied-machine-learning, n.d.). When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leaves that contains a continuous score. XGBoost is used for supervised learning problems, where the training data is used (with multiple features) 𝑥𝑖 to predict a target variable 𝑦𝑖, in this project the predictor variable being the “Price of the house”.

The objective function used is the squared error. Therefore, with each iteration, the error is minimized.

### Python Implementation

The results of the XG Boost algorithm are provided below. The sets with column lists provided in Appendix as Table I are used. The graph indicates that the results of XGBoost are almost similar to the Random Forest with a small deviation of value in accuracy and error. The graph indicates that set 5,14,15,16 have the least error and highest accuracy.



The below table provides a relation between the performance metrics of XGBoost Regression with and without standardization. XGBoost Regression is a tree-based model, therefore the feature scaling can be avoided. Because of this reason, the R2 value increase by 2% without standardization.

|  |  |  |
| --- | --- | --- |
| **Metric** | **With Standardization** | **Without Standardization (logarithmic value)** |
| R2 | 0.801873 | 0.829056 |
| Adjusted R2 | 0.642812 | 0.683714 |
| MAE | 0.297877 | 0.194043 |
| MSE | 0.192759 | 0.074683 |
| RMSE | 0.439044 | 0.273282 |
| Percentage Error | 0.50022 | 0.49846 |
| % error with sigmoid function | 0.58557 | 0.560493 |

# **Conclusion and Future Work**

Data collected from different sources such as Redfin, facilities available in each locality, NYPD reports, etc are combined to create a new data set for prediction Modelling. The key insights from the data analysis are that the factors that most influence the price of the house include the square footage of the house, the number of baths in the house, environmental conditions such as the number of hospitals, subways, and schools near the house. Data standardization is applied to the dataset to normalize the features on the same scale.

Machine Learning provides smart alternatives to analyzing vast volumes of data. By developing fast and efficient algorithms and data-driven models for real-time processing of data, Machine Learning can produce accurate results and analysis. To evaluate different ML models, performance metrics are considered and evaluated for each model. In this project ML models used are Linear Regression, Random Forest, XGBoost, SVM, and KNN. Out of the different models applied throughout the project, Random Forest Regression Model achieves good performance with the least error and most accuracy.

As for future improvement, principal component analysis (PCA) can be performed to reduce the features implementing feature importance to each model. The model performance can be measured with and without applying PCA. Additional future work could apply filter conditions as per the user’s area and locality, thus improving the speed and performance of the model.

# **Appendix**

|  |  |
| --- | --- |
| **Set** | **Columns List** |
| 1 | ['Population'] |
| 2 | ['ZIP\_OR\_POSTAL\_CODE'] |
| 3 | ['Total Numb of Hospitals'] |
| 4 | ['Total Numb. Of Complaints’, ‘Total crimes’, ‘Level A School Count'] |
| 5 | ['BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric'] |
| 6 | ['Total Numb of Subways'] |
| 7 | ['Total\_Num\_of\_Subways','min\_dist\_station','Num\_of\_Retail\_stores','min\_dist\_retail\_store'] |
| 8 | ['Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount','Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population','People/Sq\_Mile','Total\_Num\_ofHospitals'] |
| 9 | ['Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population','People/Sq\_Mile','Total\_Num\_ofHospitals','Total\_Num\_of\_Subways','min\_dist\_station','Num\_of\_Retail\_stores','min\_dist\_retail\_store'] |
| 10 | ['SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount','Level\_C\_SchoolCount'] |
| 11 | ['SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount','Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools'] |
| 12 | ['ZIP\_OR\_POSTAL\_CODE','BEDS','BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount'] |
| 13 | ['BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount','Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population'] |
| 14 | ['ZIP\_OR\_POSTAL\_CODE','BEDS','BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount','Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population','People/Sq\_Mile','Total\_Num\_ofHospitals','Total\_Num\_of\_Subways','min\_dist\_station','Num\_of\_Retail\_stores','min\_dist\_retail\_store'] |
| 15 | ['BEDS','BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount', 'Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population','People/Sq\_Mile','Total\_Num\_ofHospitals', 'Total\_Num\_of\_Subways','min\_dist\_station','Num\_of\_Retail\_stores','min\_dist\_retail\_store'] |
| 16 | ['BEDS','BATHS','SQUARE\_FEET','YEAR\_BUILT','LATITUDE','LONGITUDE','AGE','CITYnumeric','Total\_Num\_ofComplaints','Total\_crimes','Level\_A\_SchoolCount','Level\_B\_SchoolCount', 'Level\_C\_SchoolCount','Level\_D\_SchoolCount','Level\_F\_SchoolCount','Total\_Number\_of\_Schools','Num\_Complaints\_schools','Population','People/Sq\_Mile','Total\_Num\_ofHospitals', 'Total\_Num\_of\_Subways','min\_dist\_station','Num\_of\_Retail\_stores','min\_dist\_retail\_store','Num\_of\_Retail\_stores\_Zipcode'] |

# **References**

*2009-2010 School Progress Reports - All Schools*. (n.d.). Retrieved from NYC Open Data: https://data.cityofnewyork.us/Education/2009-2010-School-Progress-Reports-All-Schools/ffnc-f3aa

*2010-2016-School-Safety-Report* . (n.d.). Retrieved from City of New York: https://data.cityofnewyork.us/Education/2010-2016-School-Safety-Report/qybk-bjjc/data

*ASCII*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/ASCII

*Coefficient\_of\_determination*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Coefficient\_of\_determination

*Ensemble Learning*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Ensemble\_learning

*Feature Scaling*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Feature\_scaling

Gao, G., Bao, Z., Cao, J., Qin, A., Sellis, T., Wu, Z., . . . Fellow. (2019, Jan). Location-Centered House Price Prediction: A Multi-Task Learning Approach. *ArXiv*.

*How and where to apply feature scaling*. (n.d.). Retrieved from Geeks for Geeks: https://www.geeksforgeeks.org/python-how-and-where-to-apply-feature-scaling/

*K Nearest Neighbours*. (n.d.). Retrieved from www.saedsayad.com/: https://www.saedsayad.com/k\_nearest\_neighbors.htm

Kain, J., & Quigley, J. (1970). Measuring the value of housing quality. *Journal of the American Statistical Association, 65*(330), 532–548.

*K-nearest\_neighbors\_algorithm*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/K-nearest\_neighbors\_algorithm

Levy, D., K.C. Lee, C., & Murphy , L. (2008). Influences and Emotions: Exploring Family Decision-making Processes when Buying a House. *Housing Studies, 23*, 271-289. doi:https://doi.org/10.1080/02673030801893164

*Linear Regression*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Linear\_regression

*Mean Absolute Error*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Mean\_absolute\_error

*Mean Squared Error*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Mean\_squared\_error

*New York Population Density Zip Code Rank*. (n.d.). Retrieved from USA.com: http://zipatlas.com/us/ny/zip-code-comparison/population-density.htm

*NYC Health + Hospitals patient care locations - 2011*. (n.d.). Retrieved from NYC Open Data: https://data.cityofnewyork.us/Health/NYC-Health-Hospitals-patient-care-locations-2011/f7b6-v6v3

*NYC Subways Stations*. (n.d.). Retrieved from mta.info: http://web.mta.info/developers/data/nyct/subway/Stations.csv

*NYPD-Complaint-Data-Current-Year-To-Date*. (n.d.). Retrieved from NYC Open Data: https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Current-Year-To-Date-/5uac-w243

*Pearson\_correlation\_coefficient*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Pearson\_correlation\_coefficient

*Random Forest*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Random\_forest

*RedFin*. (n.d.). Retrieved from https://www.redfin.com/

*Retail Food Stores*. (n.d.). Retrieved from data.ny.gov: https://data.ny.gov/Economic-Development/Retail-Food-Stores/9a8c-vfzj/data

*Root-mean-square\_Error*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Root-mean-square\_deviation

*School Locations 2017-2018*. (n.d.). Retrieved from NYC Open Data: https://data.cityofnewyork.us/Education/2017-2018-School-Locations/p6h4-mpyy

Schulz , R., & Werwatz, A. (2004). A state space model for berlin house prices -Estimation and economic interpretation. *The Journal of Real Estate Finance and Economics, 28*(1), 37–57.

*sklearn.metrics.r2\_score.* (n.d.). Retrieved from scikitlearn: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2\_score.html

*Support Vector Machine*. (n.d.). Retrieved from Towards Data Science: https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

*Support Vector Machine Towards Data Science*. (n.d.). Retrieved from Towards Data Science: https://towardsdatascience.com/support-vector-machines-svm-c9ef22815589

*Taxicab geometry*. (n.d.). Retrieved from Wikipedia: https://en.wikipedia.org/wiki/Taxicab\_geometry

*xgboost-applied-machine-learning*. (n.d.). Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

1. This project does not take into account the events of COVID-19 that occurred during 2019-2020. [↑](#footnote-ref-1)