Property Evaluation in New York City

****

Data Science Capstone

ISQS 5381 Summer II 2019

Professor Fred Davis

August 4, 2019

Prepared by:

Arjuna Menon, Jaimie Capps,

Marcin Grzechowiak, Mikaela Pisani, Rogelio Valdez

Table of Contents

[*Executive Summary*](#_a1hdmrk1crw5)***2***

[*Business Understanding*](#_xoecq31k2ta)***3***

[*Business Objectives and Success Criteria*](#_pc59qca1246w) *3*

[*Data Mining Objectives and Success Criteria*](#_nnz7tbmtk7b4) *4*

[*Data Preparation*](#_ynxew2w2milb)***8***

[*Modeling*](#_ugyuw061ce7a)***11***

[*Evaluation*](#_4tqi70671s8g)***27***

[*Deployment*](#_mdzq7xcnp4vx)***29***

[*SCRUM Backlog/Sprints*](#_8onjg9ux1j83) ***30***

[*Appendix*](#_vrr0xlrd0g2)***33***

# 

# Executive Summary

# 

In this data analysis, “Property Evaluation in New York City,” the intention is to predict the land value and total property value for any given tax lot in New York City. In order to understand the impact of the assessed land and assessed property of the tax lot on its market value, research was conducted to determine the relationship between these variables. Therefore, after predicting the assessed value of the land/property, the market value might be calculated according to the tax class. The prediction models provided in this analysis can be useful as a business service. For example, if a company was looking to expand infrastructure in New York City they could use the information from the predictions to decide if they should buy land and build on it or buy land with a building already on the property. In order to provide this service, a business model in the form of an online platform would be implemented. The online platform will give the user access to filters for multiple variables and receive a visualization about the potential land/property values that align with their preferences.

When considering variables, research was conducted in order to evaluate previous methods of model creation and which variables were chosen for prediction as a reference. One article found titled, “Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data,” considered variables such as the number of full bathrooms, fireplaces, garage spaces, bedrooms, and more. While those variables might be important for predicting, at the same time they are very difficult and expensive to gather. In order to find the number of bedrooms a residence has, there comes a need to ask the owner questions about home details and at this point, it is up to the owner if they would like to disclose that information. However, the idea of this project is to predict the value of the property using data which are much easier and cheaper to access, for example, using data that are gathered for tax purposes and that are freely available online. If the model can be built based on that information the business' cost could be significantly reduced, and as a result, the profit would be increased.

This report concentrates on comparison between different machine learning prediction techniques in order to evaluate which model performs the best. The critical measurement in the analysis is RMSE. The advantages of root mean square error is its interpretability - it gives the error value in the same units as target variables, in dollars. Expressing errors in dollars makes the results easily understandable for people who are not fluent in machine learning analysis.

# 

# Business Understanding

#### ***Business Objectives and Success Criteria***

Imagine a world without Zillow and its “Zestimate” tool. How would you determine what the potential cost of a property should be or what you expect it to be? In this analysis, we analyze data from New York City to find out what exactly the necessary predictors are when asking this question and how we can implement this into an online platform that can be utilized by its users. This analysis requires to delve deeper into how we define each variable and what we consider to be influential factors based on the results of tests and modeling we conduct.

The audience, or clientele, are companies and consumers seeking, or selling, property for either business or residential use. The implemented strategy, as mentioned earlier, is to create an online platform where the client can input his or her preferences and the platform will populate with the potential listings corresponding to those preferences. Likewise, there is a need to keep the client’s interests in mind during this process. The expected benefits associated with this analysis, with the seller in mind, is to save time and increase trust in a model to predict the potential value that can be obtained by selling that piece of land or property. For the buyer, the expected benefits include the ability to weigh their options and evaluate whether or not they should buy a certain piece of land or property. The success will be defined by the ability to randomly select a lot and predict the land value or property value associated with that lot.

The evaluation and census data used in this analysis was a result of active data capture. Active data capture is manual data transfer that is initiated by an individual’s decision to provide the data. We integrated data fusion into our analysis when we fused the census data and the tax lot data together. Monetization of the data is one of the primary driving factors for creating a dashboard for users. This entire analysis is based on the assumption that the client is risk-averse. This opens up many opportunities for providing a service that in a way, gives the client comfort in their process of deciding where to buy or sell their property. In an effort to evaluate the return on investment, we can potentially use a heatmap tool after the dashboard has been in use for a while to determine the property value across all areas of NYC. Investors can utilize this feature by knowing where in the city they will likely capitalize on the most positive rate of return.

#### ***Data Mining Objectives and Success Criteria***

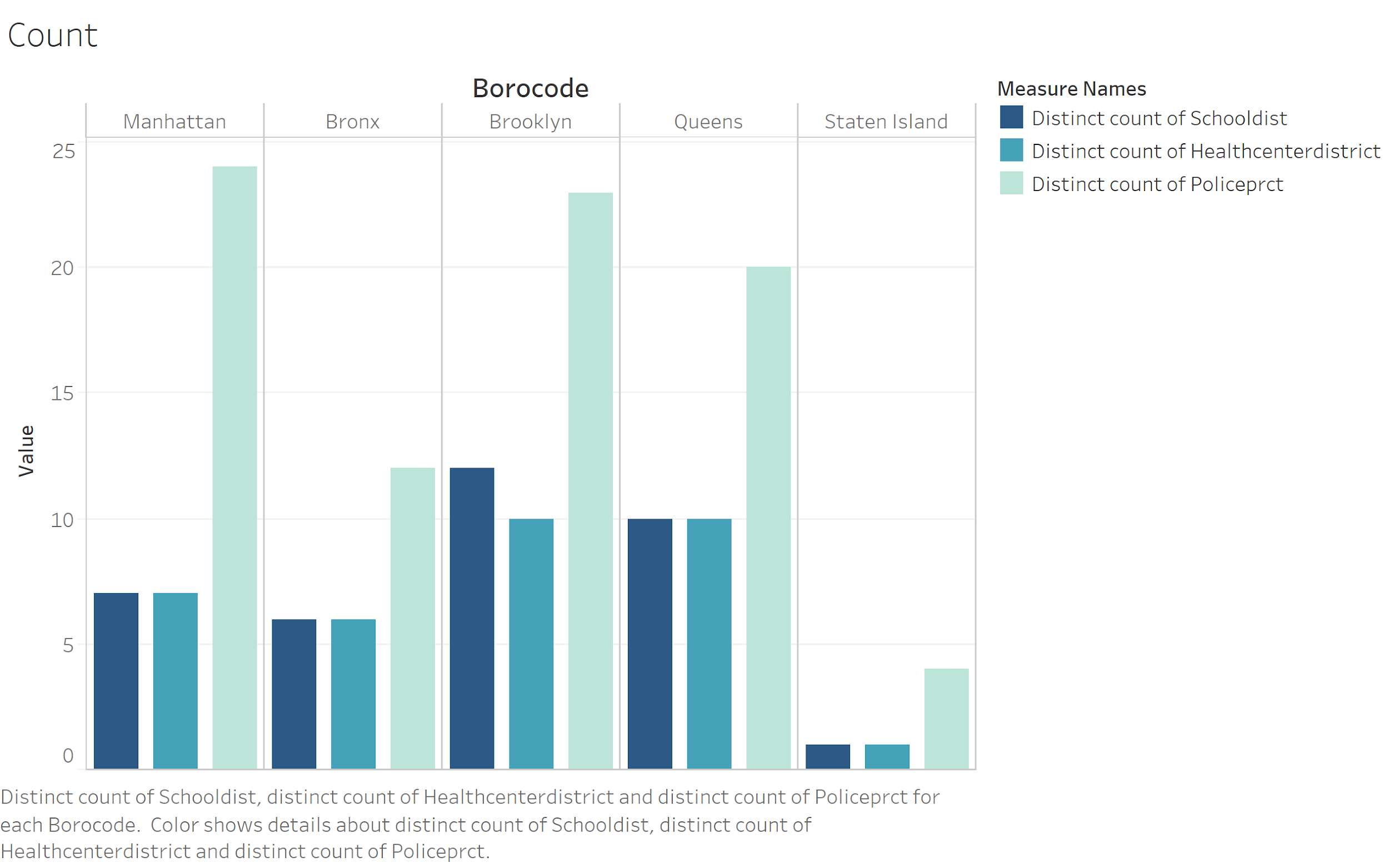
Two target variables were defined in the project, assessland and assesstot. The first describes the assessed value of the land without any buildings present, while the latter describes total assessed value of the property. Since those variables are strictly related to the market value, the results of this analysis might be widely used for many business needs. The most important criteria for the model evaluation is RMSE. The error allows to compare models in terms of their errors described in dollars. Expressing error in dollars make model interpretation easy to understand for people who are not fluent in machine learning jargon. Moreover, the results can be easily and quickly transformed into business decisions. Additionally, different visualization techniques are used as well as R-Squared value in order to check for overfitting the model. When the histogram of differences between predicted value and the true value is significantly different, that means that model is prone to overfitting. Furthermore, R-squared values is being used to assess if model is overfitted. When the value of the R-squared is high for training set but low for the test set, it indicates overfitting problem. The success criteria are defined by the lowest RMSE as well as avoiding overfitting the model. The model which best fulfills those criteria will be chosen as the final one.

***Data Understanding***

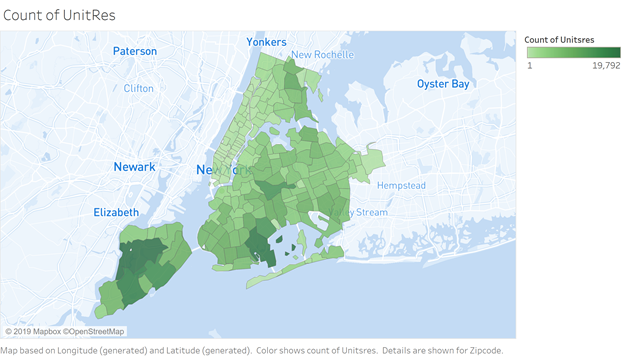
The data from the analysis was found from the City of New York Primary Land Use Tax Lot Output (PLUTO) 2018 and later combined with the census data on the zip code in the same year. The original data set can be found at the website New York City Open Data[[1]](#footnote-0) and the census data can be found on the United States Census Bureau website[[2]](#footnote-1). A list of all of the variables can be located in the appendix of this document. From here the defined variables along with their alias name, datatype, appropriate name, issue associated and the solution can be found.

In order to better understand the data, various views were created and variables were chosen that were most useful for visualizations. Zip code and borough code variables helped plot the data on a map of New York City which became useful combined with different measures of the dataset. Instances of variables for measurement included median income, count of residential units, count of police precincts, count of health center districts and count of school districts per borough.

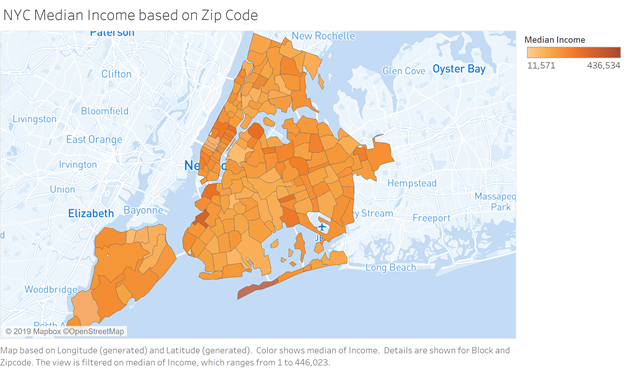
For visualizations the application Tableau was chosen, this allowed different inputs such as measures and dimensions in order to plot on various graphs. For example, the bar graph figure created below split the data into the five different borough codes in New York City. Once split into the boroughs a measure was created using a distinct count of the variables “schooldist”, “healthcenterdistrict”, and “policeprct.” Once graphed it became easier to view the number of police precincts, health center districts, and school districts within each borough. As a result, it can be seen that all of the boroughs contain more police precincts than school districts and health center districts.



The dataset contained zip codes which were used as a dimension to plot on a map graph. From here the measure was created using the count of the variable “unitsres,” which is the sum of residential units in all buildings on the tax lot. The graph provided the results of a map of New York City with darker colors reflecting areas where the number of residential units were the largest. According to the data and referring to the map the darkest green is located in Staten Island area.



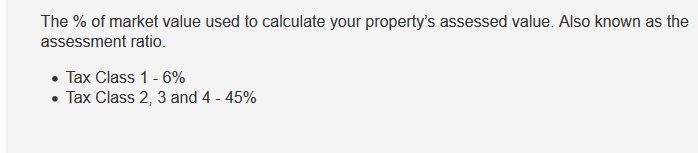
The same map was created above but instead of the count of residential units, the measure of median income was used. In this case, the largest median incomes are sporadic with darker areas around Manhattan and closer to the oceanfront around Brooklyn. For these areas, it would make sense that they would have larger median incomes because the cost to live in the most popular city or an oceanfront would require more income.



# Data Preparation

In order to keep the data cleaning process concise and easy to understand, all the steps taken are explained in bullet points below. Detailed explanation and reasoning for deleting variables which would not be useful for prediction were explained in the Appendix A.

* All variables that have a majority of NA values were removed (mostly greater than 90%). Also, variables that contained only one value in addition to zeroes, such as rpaddate, were removed.
* The variables block and lot were removed since lots repeat within blocks and blocks repeat within borough. Hence, these variables would need to be combined to have unique values and this provides redundant location information present in other variables and basically, it would act only as an identifier.
* The missing value of the zipcode variable were assigned values by matching them with a shapefile in Python and QGIS.
* One of the zipcodes corresponds to an airport. It was removed because it has an extremely high assesstot value that might affect the prediction. This outlier was found via Tableau.
* Variables that must be converted to factors, and later dummies, are identified. Some of these variables have too many levels that provide too much detail, and these were transformed into variables with fewer levels.
* For variables containing an acceptable number of NAs, these missing values were replaced with a random value of that variable for the corresponding zip code if that variable was non-numeric. For numeric variables, these missing values were replaced with the median value of that variable for the corresponding zip code.
* If a particular zip code contained only NAs for a column, that column’s NA values were replaced with either a random value or the median of that variable for the corresponding borough.
* Outliers that have values more than three standard deviations from the mean in at least one or more of the numerical variables were removed in Python.
* All numeric variables, except for the target variables assessland and assesstot, were scaled and the factors were converted to (n-1) dummies.
* Stratified sampling based on zipcode was done on the dataset in Python to obtain a sample that adequately represented the population.
* This sample was used for all models except neural network.
* For predicting assessland, we remove all building related variables and assesstot while we consider all variables except for assessland for predicting assesstot.
* In the case of neural network, principal component analysis was performed on the sample disregarding building related variables for predicting assessland. 170 principal components were chosen that explained around 93% of the variation in the entire dataset.
* For predicting assesstot, all the variables except for assessland were considered. 170 principal components were chosen that explained around 95% of the variation in the entire dataset.
* Finally, the market value of assessland and assesstot can be calculated as follow:

  
Source:<https://www1.nyc.gov/site/finance/taxes/definitions-of-property-assessment-terms.page>

* The tax class is defined by the amount of residential units that the lot has. If it has 1,2, or 3 residential units, it is classified as class 1 (6%). Otherwise the percentage would be 45% (for 0 residential units and above 3 residential units).

***Merging multiple data sources:***

* Merge pluto with census data: The census dataset has information about the average income for families and non-families by zip code. This data was merged by zipcode with the original dataset.
* Missing values in the mean income for families and non-families were replaced by the median income for families and non-families respectively, for the corresponding zip code.
* Any NAs still remaining were replaced by the median value of income for families and non-families of the corresponding borough. This is in the cases of entire zip codes containing only NAs for the mean income for families or non-families.
* In order to better represent reality, the mean income was then replaced with values from a triangular distribution with a minimum, maximum and most likely values for each zip code, rather than having a repetition of the same average income value for each zip code.
* Also, income was assigned a value of zero in the case of zero building area since the absence of a building should prohibit any income being generated in that tax lot. Furthermore, those tax lots having a landuse value greater than 5 were assigned 0 income since they refer to tax lots that do not contain families residing permanently in these buildings.

# Modeling

As stated at the very beginning, the objective of the project is to predict the value of the land as well as total value of the property. In order to keep this report short and neat, only the process for predicting assessed land value will be described in detail with plots and descriptions. In the case of variable describing the assessed value of the property (assesstot) it is only described the process for the best model, and the rest of the plots are presented in Appendix in order to present the process of analysis.

1. **Assessing the value of the land (Assessland variable)**

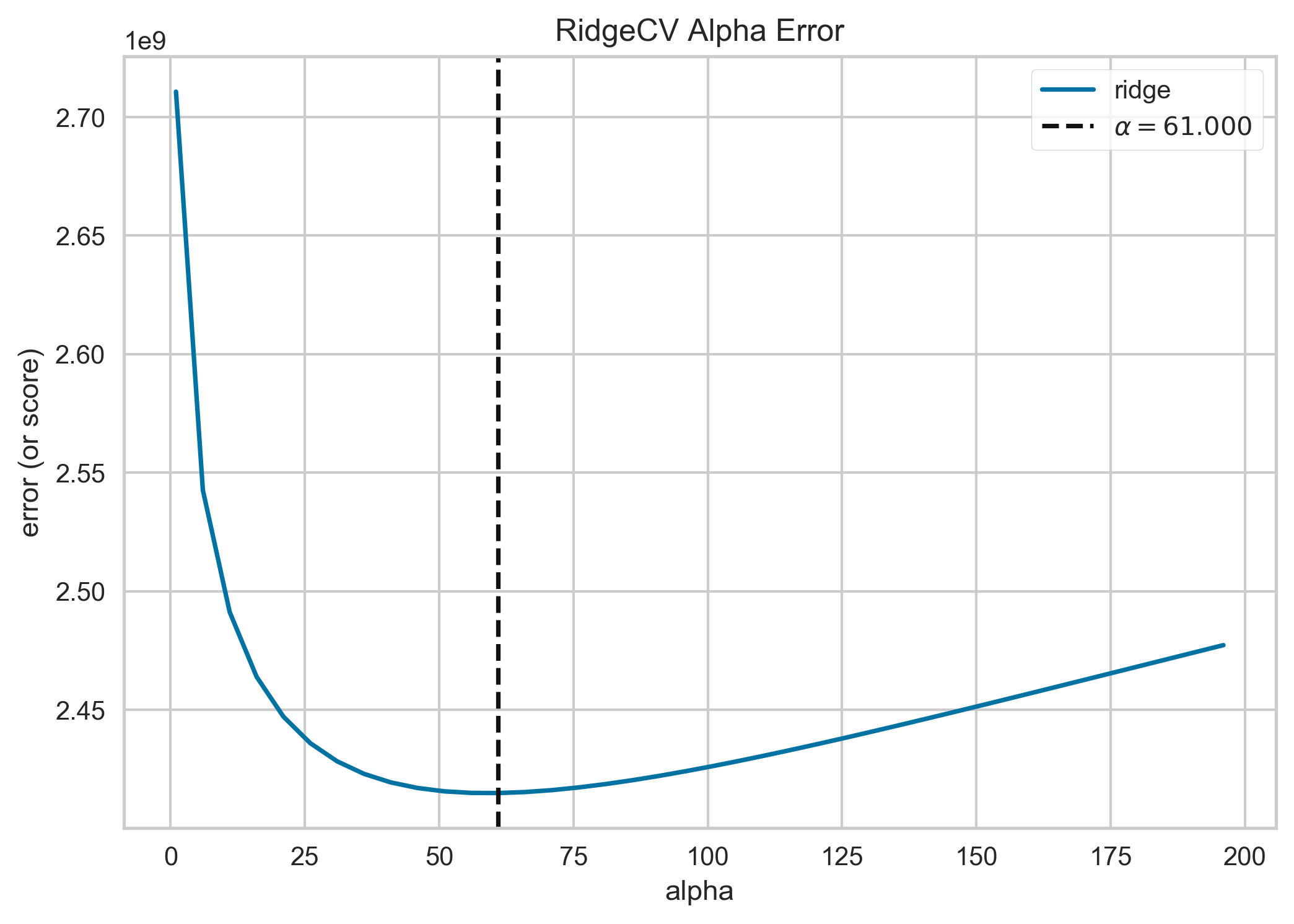
In order to predict a value for the target variable, assessland, five models were used. The process of tuning the models is described below. The used models are as follows:

1. Ridge Regression
2. Lasso Regression
3. KNN
4. Random Forest
5. Neural Network

These models are explained in detail below.

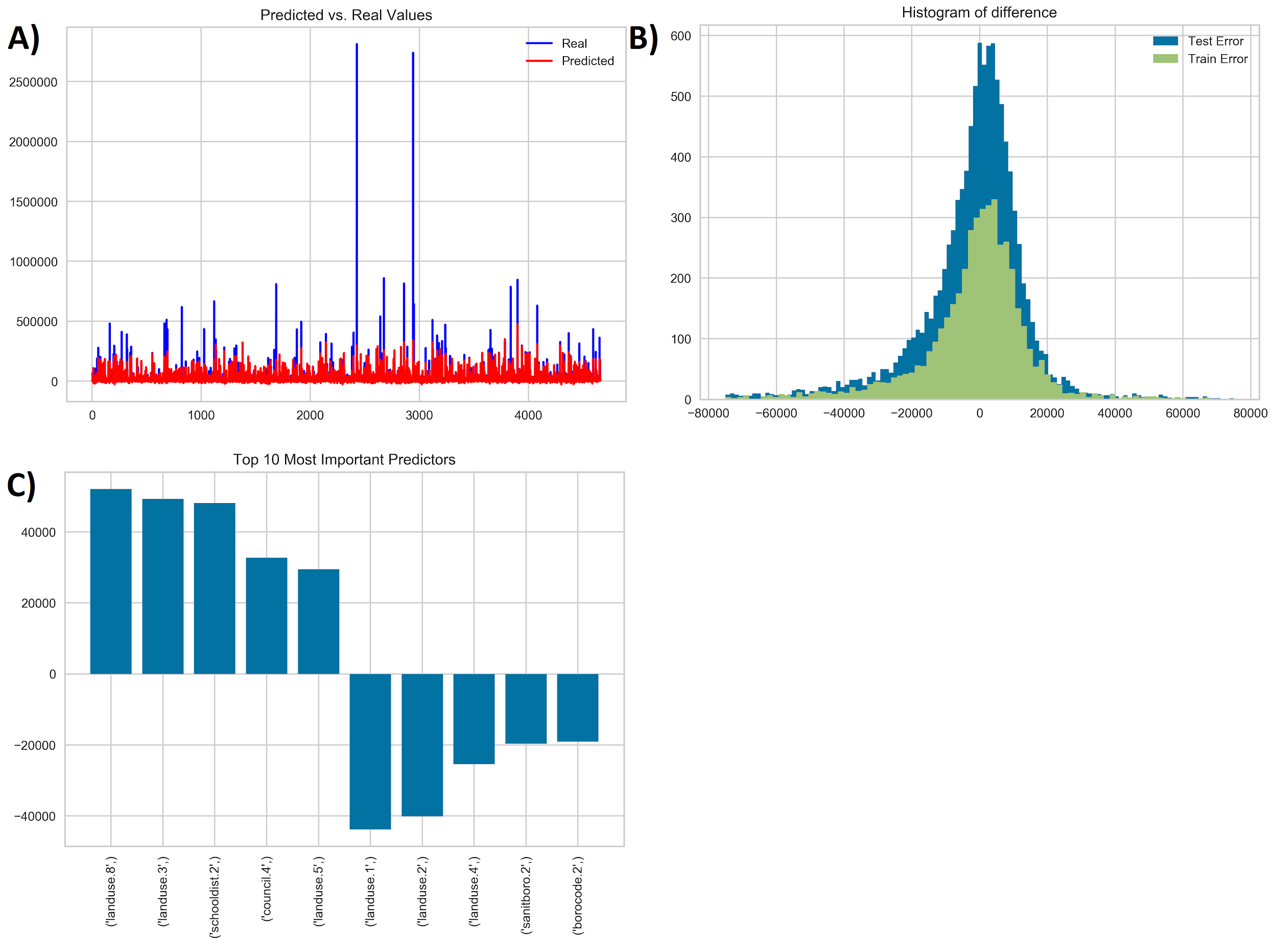
1. Ridge Regression
   1. Tuning process

In order to find the best value of the parameter alpha, which describes the penalty level for variables that account for a low prediction power, cross validation was used. The figure below shows that as alpha goes up, the error goes down. This means that the penalty level for predictors should be relatively high and the cross validation yields the best value for alpha equal to 61.



* 1. Model performance

In order to better understand the model behavior, several plots and measurements are presented below.



The graph (A) compares true value of assessland and predicted one. It can be seen that blue lines represent real value of the land while red values describe predicted value. The simple regression model performs quite good when it comes to predict small values of the prices, however when it comes to higher values in general Ridge Regression underestimate assessland.

Another presented measure is histogram of difference (B). Data for both training and testing set shows difference between true value and predicted value of assessland. The plot tells that on average differences are around zero. Both differences, in training and testing sets, have similar distribution, which means that there is no overfitting problem in this case. The issue is that, while most of the errors are around zero, in some cases the model makes big mistakes, however they are rather rare.

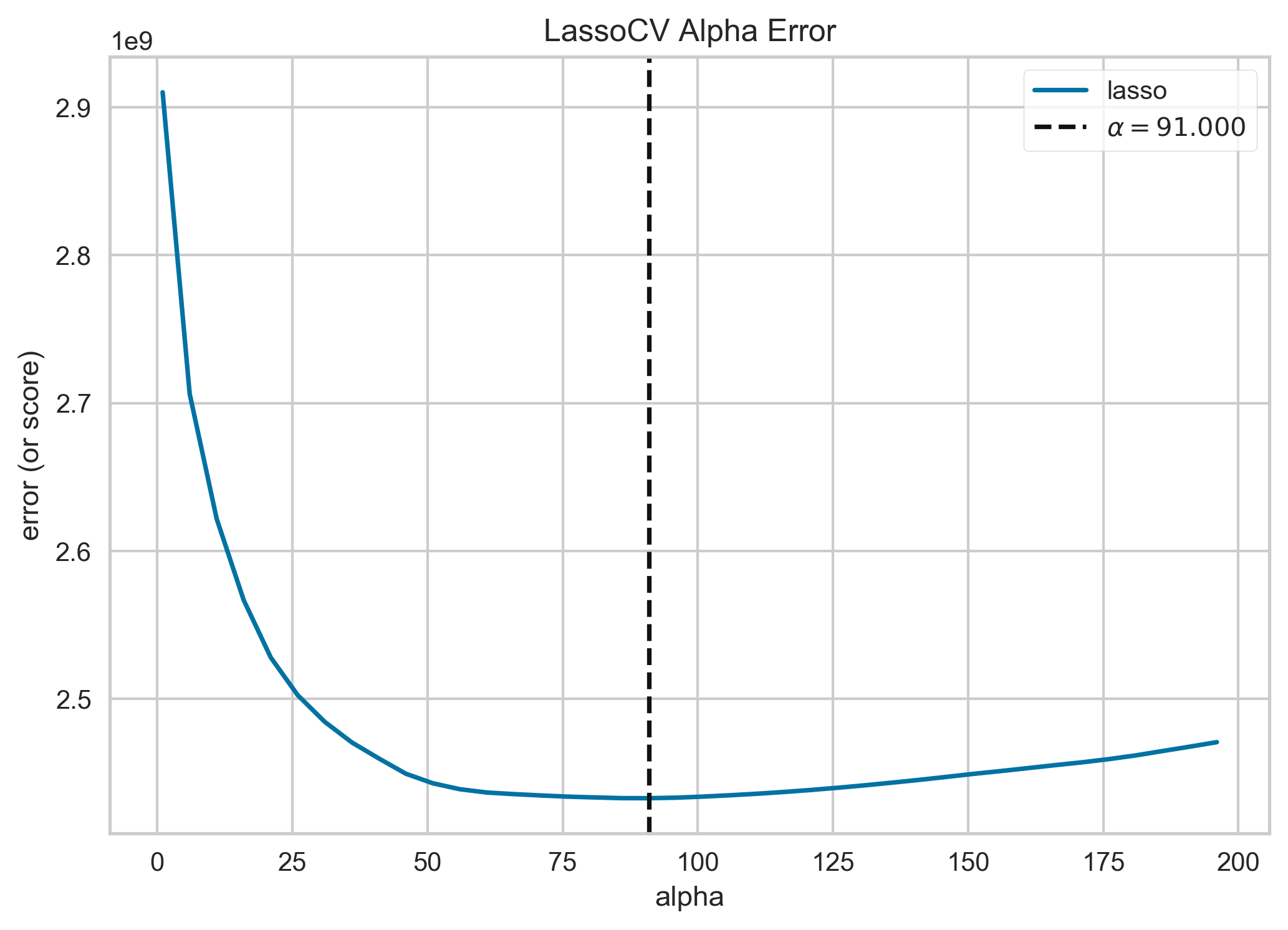
The advantage of regression is that for each predictor used in the model, it estimates its coefficient. In the graph (C), 10 the best predictors, both with positive and negative coefficients are presented. The variables which presence increases the price of the land are, for example, Landuse 8, 3 and 5. According to the documentation, those lands refers to: Public Facilities & Institutions, Multi-Family Elevator Buildings and Commercial & Office Buildings. However, variables which presence decrease the land value are for example, Landuse 1, 2 and 4. Those variables represents: One & Two Family Buildings, Multi-Family Walk-Up Buildings, Mixed Residential & Commercial Buildings. As a result it can be concluded that lands which are related to public institutions, lands with tall buildings where elevator is needed as well as lads where offices are placed are higher priced.

2. Lasso Regression

The idea of the Lasso Regression is very similar to Ridge Regression, however Lasso penalizes predictors more harshly. When the variables is not useful as a predictor its coefficient will be reduced to zero. Because the data presented in this project contains more than thousands of dummy variables it is reasonable to use Lasso Regression.

a) Tuning process

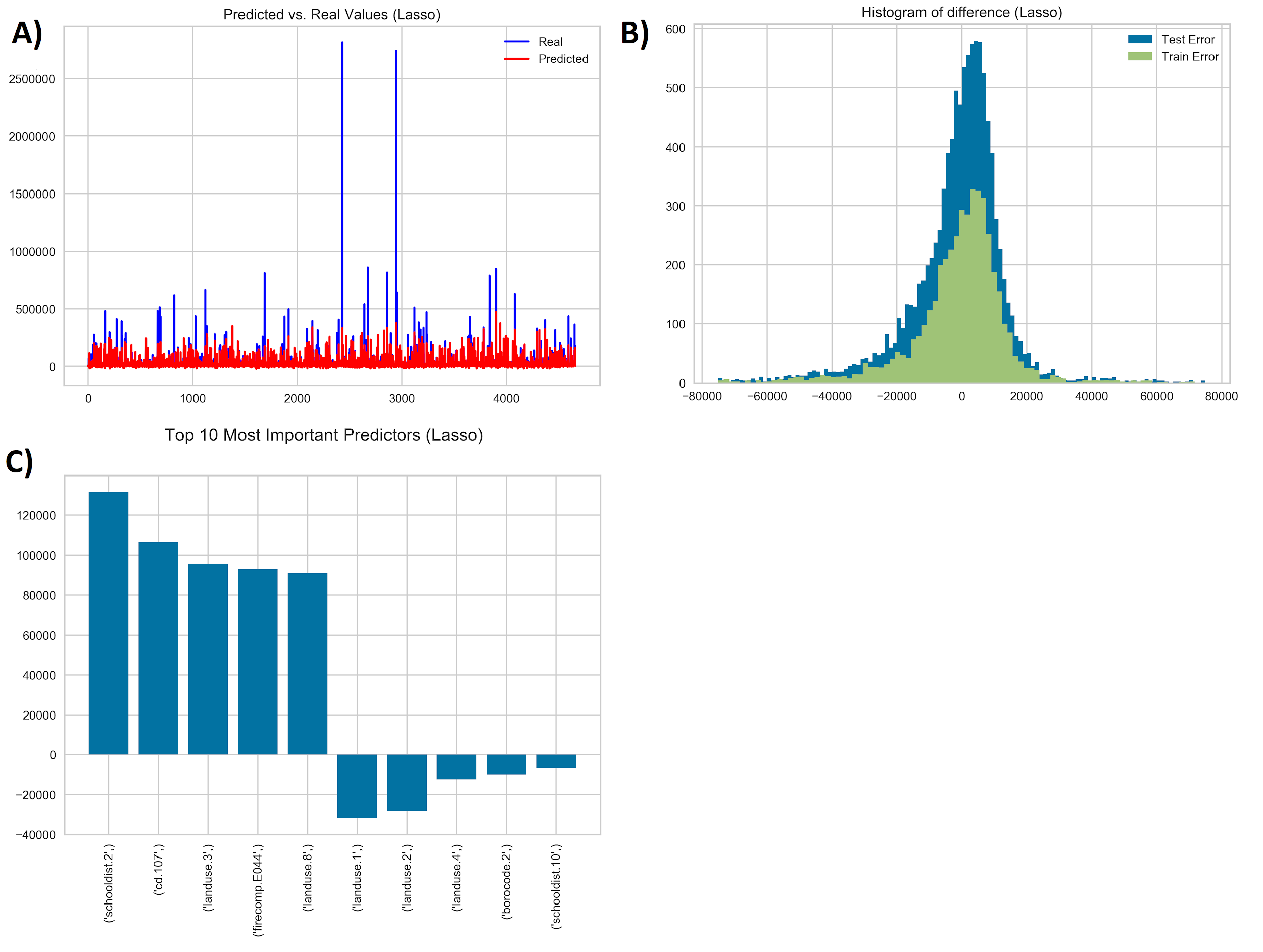
Similarly to the previous Regression problem, the alpha value was determined by Cross Validation technique. The graph is presented below.



The process of cross validating the data decided optimal alpha to be 91, thus this value was utilized in the model.

b) Model performance

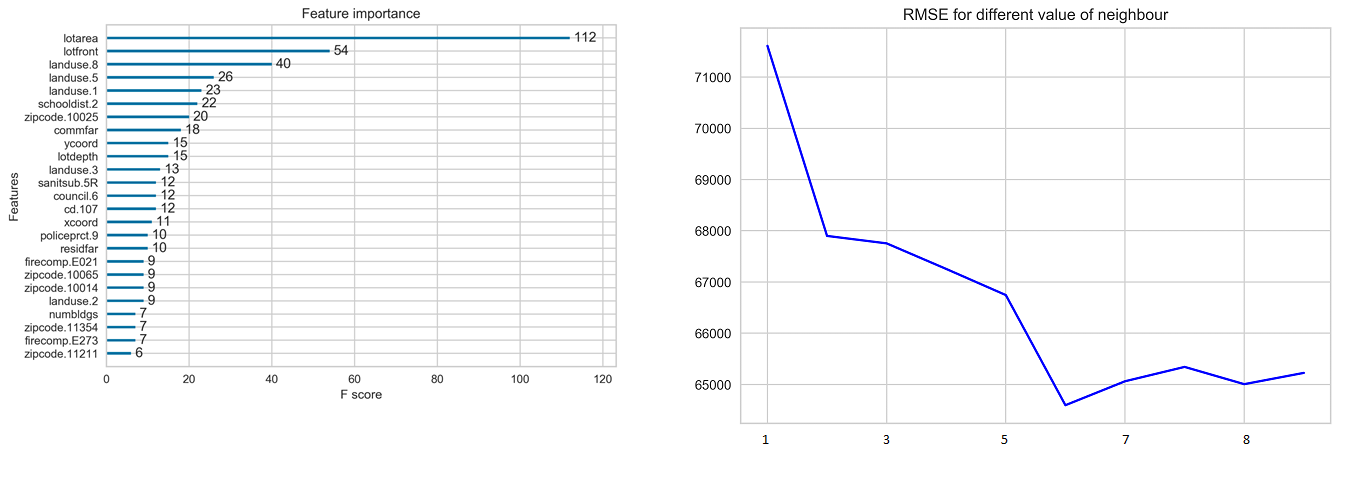
While comparing the prediction with real values of land value the results are very similar to Ridge Regression. The model has difficulties to predict values of the land which are higher than most of the data (A). From the histogram of difference (B) it can be intereferd that model does not overfit the data and most of the errors are placed around zero. The coefficients values displayed on plot (C) are very similar to those for Ridge Regression. Additionally, Lasso regression penalized 1027 variables and reduce their coefficients to zero.



3. KNN

1. Model Tuning

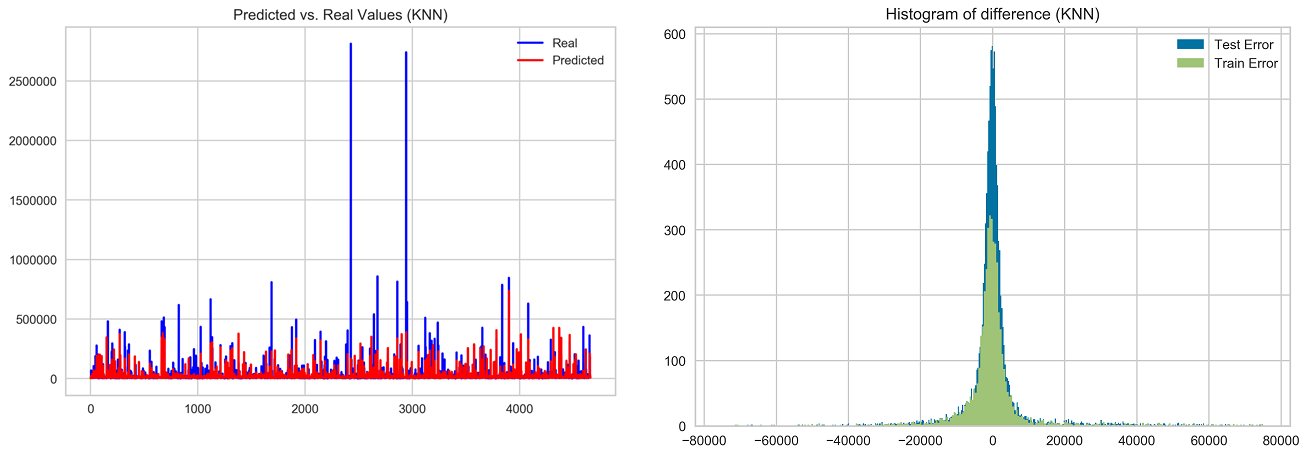
The algorithm uses euclidean distance formula to find the most similar row for prediction. In contrast to previous algorithms KNN does not have implemented any feature selection options. Thus, in order to improve the model performance, gradient boosting - XGBoost technique were used to determine the best predictors. Additionally, in order to get the best result of prediction, the model was tested on different number of neighbours. Results can be seen below.



Graph on the left shows the top 20 most important variables for prediction used in KNN. As it can be seen among selected variables are also those which were chosen by Ridge and Lasso regression, for example, Landuse 8,5 and 1. The XGBoost algorithm chooses ‘lotarea’ as the most important variable for predicting price of the land. Graph on the right side shows that the lowest error is presented when number of neighbours are equal to 6.

b) Model performance

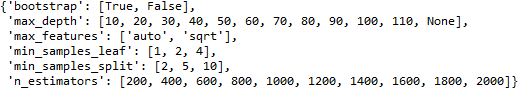
Histogram of difference shows that distribution of difference of error for both train and test are very similar. However, the graph on the left shows that predicted values do not align well with real values.



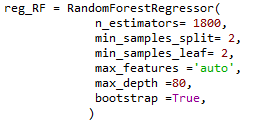
4. Random Forest

1. Model Tuning

In order to determine the best parameters for Random Forest, cross-validated search over parameter settings were used. Below the list of different setting for each parameter is presented.

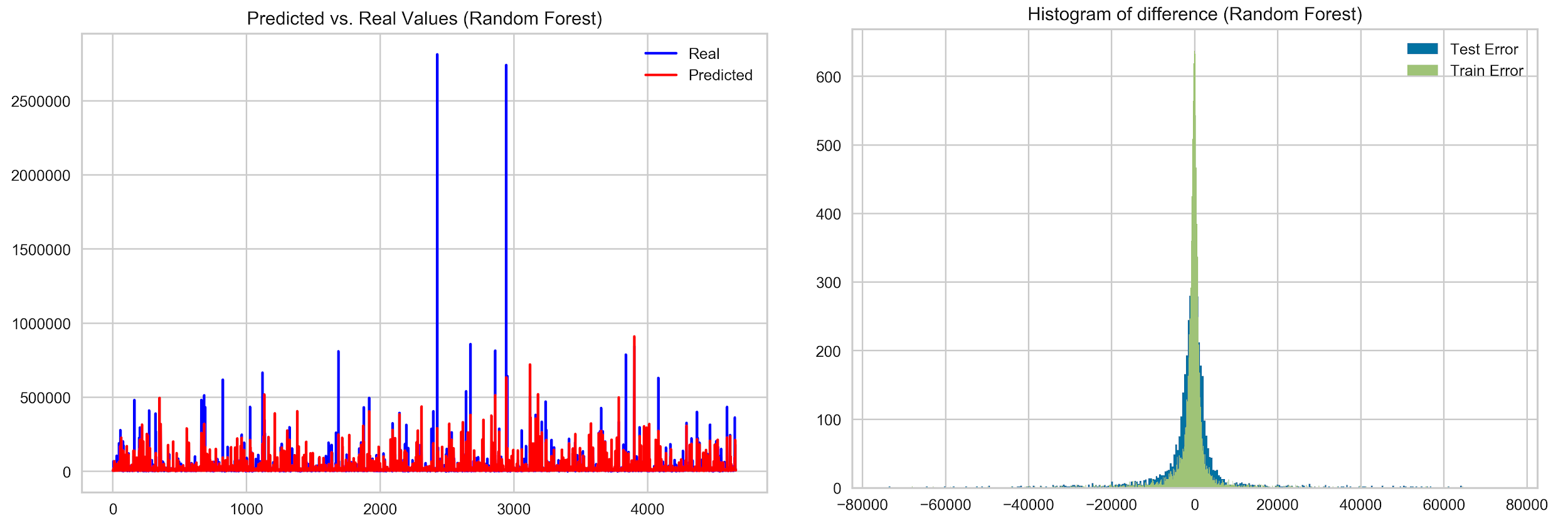


It can be seen that there are 32 different parameters to check, so the total number of combinations is equal 1024. However, in order to improve the process, the function RandomizedSearchCV was used. The function does not try all possibilities but randomly choose number of specified settings given by the user, in this case 100 different parameters were checked. Best parameters are presented below.



b) Model performance

The performance of the model is described by the graphs below. Random Forest still has problems with predicting very big values. However, while comparing histograms across models which were presented before, it can be seen that the histogram for the random forest is very narrow. This means that in general difference between predicted value and real value is very close to zero.



5. Neural Network

1. Model tuning

In order to model a neural network, several parameters should be defined. The discussed parameters are listed below. An automatic test was performed in order to check the performance and choose the best combination of the different parameters discussed below.

Amount of layers: With two layers the model does not predict well. This might be because the data is complex and it has many variables. Therefore, it was decided to try with just three layers, one input layer, one hidden layer and one output layer. From this testing with different amount of nodes for the hidden layer was conducted.

*Amount of nodes:*

* Input layer: For the input layer the same amount of predictors are used
* Hidden layer: In order to determine which amount of nodes should be tested, some rules of thumb were followed.
  + 1 hidden layer with (Number of inputs + outputs) \* (2/3) nodes
  + A typical recommendation is that the number of weights should be no more than 1/30 of the number of training cases. Several number of nodes were tried, decreasing the number of nodes in each trial.
* Output layer: The result for the prediction should be only one value, therefore only one node for the output layer is used.

*Activation function:* With the purpose of introducing non-linearity into the output of a neuron, an non-linear activation function is used in the hidden layer.

As the problem is a regression predictive problem, the recommended activation function is relu, which generates an output to be 1 if the value is greater than 0 or 1, if not, this makes the activation sparse and efficient. This is recommended when you do not know the nature of the function that is being predicted. This activation function should only be used within hidden layers of a neural network model.

*Optimizer*: In order to minimize the loss during the training process, an optimizer is used during the compile phase. Several optimizers were tried with the objective function to compare the results obtained. The optimizers used are listed below:

* RMSprop: Root Mean Square Propagation. It utilizes the magnitude of the recent gradient descents to normalize the gradient. This optimizer is usually a good choice for recurrent neural networks. Learning rate gets adjusted automatically, and it is based on division by the average of the exponential decay of squared gradients.
* Adagrad: performs larger updates for infrequent parameters and smaller updates for frequent parameters.
* Adam: Adam can be viewed as a combination of Adagrad, which works well on sparse gradients and RMSprop which works well in online and nonstationary settings.
* AdaDelta: Adadelta is an extension of Adagrad and it also tries to reduce the learning rate.

b) Model performance

The following metrics and graphs were used to evaluate the models’ performance.

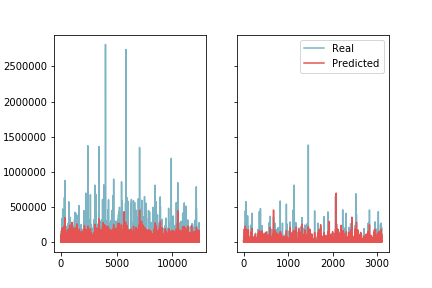
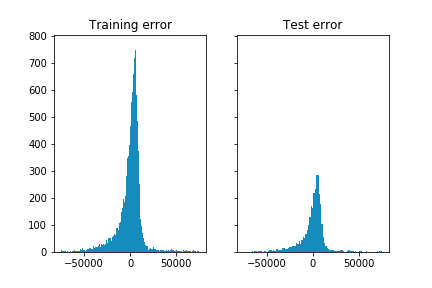
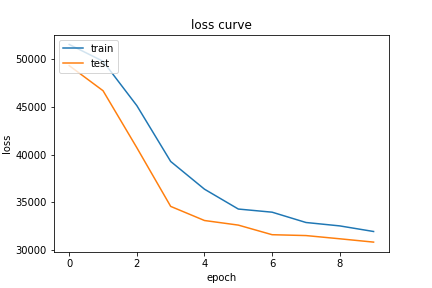
* Learning Curve: It is a graph that compares the performance of the model on the training and testing data over a varying number of training instances. It is based on the loss function, which is the objective function the model is trying to minimize. In each iteration the loss is calculated for the training and the validation set. Looking at this plot, it can be determined if the model is overfitting. It is expected that the training error is less than validation but among the iterations the validation error gets closer to the training.
* Errors: the metrics MSE, R-square and RMSE errors for each model are provided. Choosing among them the one that has less R-square error and maximum RMSE. In addition, the error calculated by actual value minus predicted values is provided. In order to visualize the error, the actual values vs predicted values are plotted overlayed.

Several problems were faced. Firstly, training for a neural network with all the variables as input was performed, this made the model complicated and it took a long time to process. Therefore, it was decided to perform a neural network over the results from PCA variance ratio. Where 93.4% of the data is explained in the case of assessland and 95% in the case of assestot. In the first attempt, a sample of the data was taken based on the proportion per zip code; applying first PCA and then executing the model over PCA results. A total of twenty models were performed changing the amount of nodes for hidden layers. Secondly, the results for the sample case presented a problem, the error for the testing set was lower than the training. This might be because of several reasons with the way the sample is built, such as the variance of the data or the proportion of the classes in each set. It is difficult to determine which is the specific reason because of the fact that the data contains many categorical variables. To resolve the issue, PCA was applied to all the data and then the models were performed. In this case, the loss curve for assesstot looks as expected, where the testing loss is higher than the testing. However, for assessland the problem persisted. Although other parameters were configured in order to improve the model, such as batch size, the amount of iterations, regulations functions and the percentage for validation set, the problem was not resolved. Thus, further investigation should be done to fit the model.

Either for the prediction of assessland and assestot the same process was followed. Twenty models were performed for the result of PCA of the sample. After, the best models were selected and performed over the results of PCA for the full dataset.

First attempt:

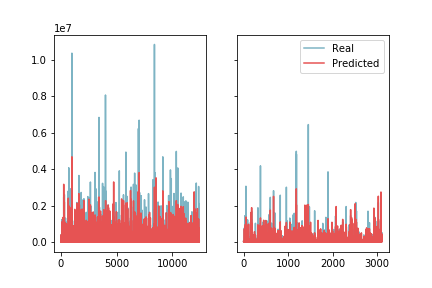
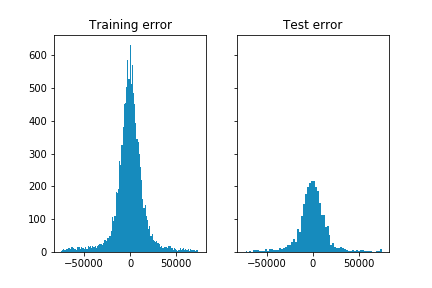
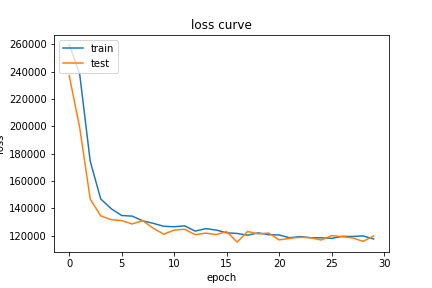
* Assessland : The best model was the one that had 170 input nodes, 51 nodes for the hidden layer and used the optimizer rmsprop with 10 iterations. The results are provided below.



|  |  |  |
| --- | --- | --- |
| **Error** | **Training** | **Test** |
| mse | 2654085536.567 | 1826452058.415 |
| RMSE | 51517.818 | 42737.01 |
| R2 | 0.431 | 0.458 |
| error | 3756.8431480694837 | 3206.323461536716 |

Although in the loss for test is lower than testing, looking at the overall error based on the subtraction of predicted versus actual values, it can be seen that the average R-squared value is higher for the training set than the testing, as expected. In addition, the difference between both sets is low. However, as it can be seen in the graph, the model does not predict well for higher values.

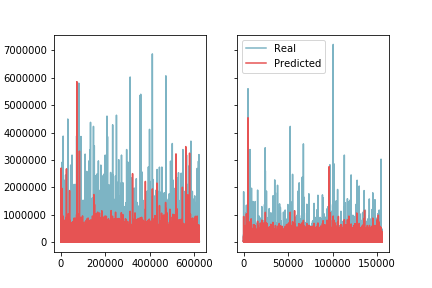
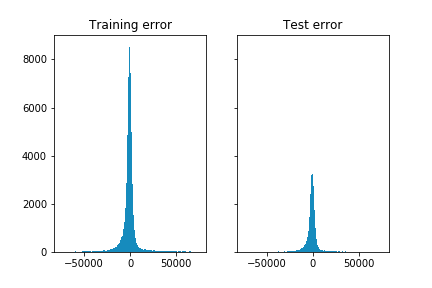
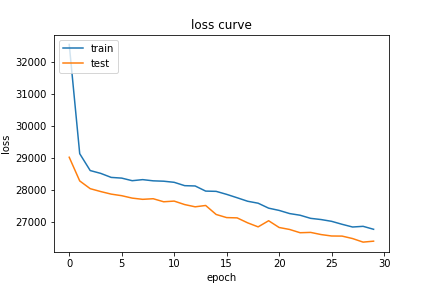
* Assesstot: The best model in this case was the one that had 170 input nodes, 114 nodes for the hidden layer and used the optimizer adam with 30 iterations. The prediction results are similar to the case for assessland, but the problem about the loss value is presented but in a lower level, since the difference between the errors is smaller. In addition, the overall RMSE error for test is lower than training as well as R-squared error is higher for training, as expected.



|  |  |  |
| --- | --- | --- |
| **Error** | **Training** | **Test** |
| mse | 34571967236.57 | 29076248801.285 |
| RMSE | 185935.385 | 170517.591 |
| R2 | 0.694 | 0.691 |
| error | 7579.095 | 7600.385 |

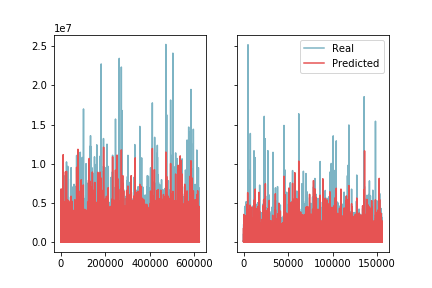
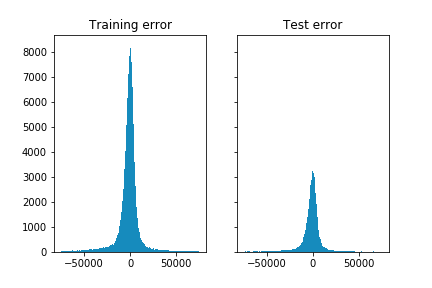
Second attempt: Here are provided the results for after performing PCA for all data.

* Assessland: For assessland the problem persisted. Therefore, this model would not be used for the prediction because it is not accurate.



|  |  |  |
| --- | --- | --- |
| **Error** | **Training** | **Test** |
| mse | 2389213693.381 | 2038678307.96 |
| RMSE | 48879.584 | 45151.725 |
| R2 | 0.544 | 0.601 |
| error | 1278.283 | 1275.985 |

Assesstot: For assesstot the problem was resolved. The loss is higher for the test as expected, having a higher RMSE value and a lower R-square.



|  |  |  |
| --- | --- | --- |
| **Error** | **Training** | **Test** |
| mse | 25735610540.596 | 29741303103.831 |
| RMSE | 160423.223 | 172456.67 |
| R2 | 0.772 | 0.75 |
| error | 5444.63 | 5902.59 |

1. **Assessing the total value of the property (Assesstot variable)**

In order to keep the report concise, the result of the analysis will be displayed in the Appendix in the form of graphs only, thus the process of analysis can be tracked by the reader. Only the final model for assesstot variable will be described in more details.

NN goes here…………………...

**Here everything about assestot (final model etc)**

# 

# Evaluation

The results of all models are presented in the summary table below, where RMSE and R square measures are compared across models. At first the models for assessland will be compared.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model for Assessland** | **RMSE train** | **RMSE test** | **R squared (train)** | **R squared (test)** |
| **Ridge Regression** | 39028 | 62829 | 0.57 | 0.37 |
| **Lasso Regression** | 39508 | 62622 | 0.56 | 0.37 |
| **KNN** | 36621 | 64593 | 0.62 | 0.33 |
| **Random Forest** | 19502 | 59076 | 0.89 | 0.44 |

It can be seen from the table for assessing the land value that the best performance, the lowest RMSE, is for Random Forest. On average the model is wrong around $59,000. While this value might seem to be very high, it should be kept in mind that values of the lands within New York are very often in hundreds of millions of dollars. Looking from this perspective $59,000 might not look that high.

The second table shows comparison between models regarding assesstot value.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model for Assesstot** | **RMSE train** | **RMSE test** | **R squared (train)** | **R squared (test)** |
| **Ridge Regression** | 141806 | 155380 | 0.80 | 0.80 |
| **Lasso Regression** | 144454 | 155796 | 0.79 | 0.79 |
| **KNN** | 196144 | 238272 | 0.63 | 0.53 |
| **Random Forest** | 77950 | 184690 | 0.94 | 0.71 |
| **Neural Network** | 160423 | 172456 | 0.772 | 0.75 |

In the case of prediction of assessing total value of the property the best performance is shown by the Neural Network model. On average the model is incorrect around $172,456 based on the validation set.

As a result of analysis, it is recommended to use models for initial evaluation of the property value. Businesses can make use of models on the beginning of their deciding process, in order to consider among different alternatives. However, it is not recommended to make a final and crucial decisions based on the models’ predictions. It should be kept in mind that there is a possibility to attach new data for the existing model in order to improve its prediction accuracy.

Additional value can be added by updating the models with data from property’s conditions, for instance if it is ready for business or needs renovations according to the article “Estimating commercial property prices: an application of cokriging with housing prices as ancillary information”.

Potential clients might be interested in taking into account the results of this prediction to base their decisions complementing with other sources. Support the decisions with models might help to reduce the bias generated by assumptions or gut feeling that buyers and sellers have.

# Deployment

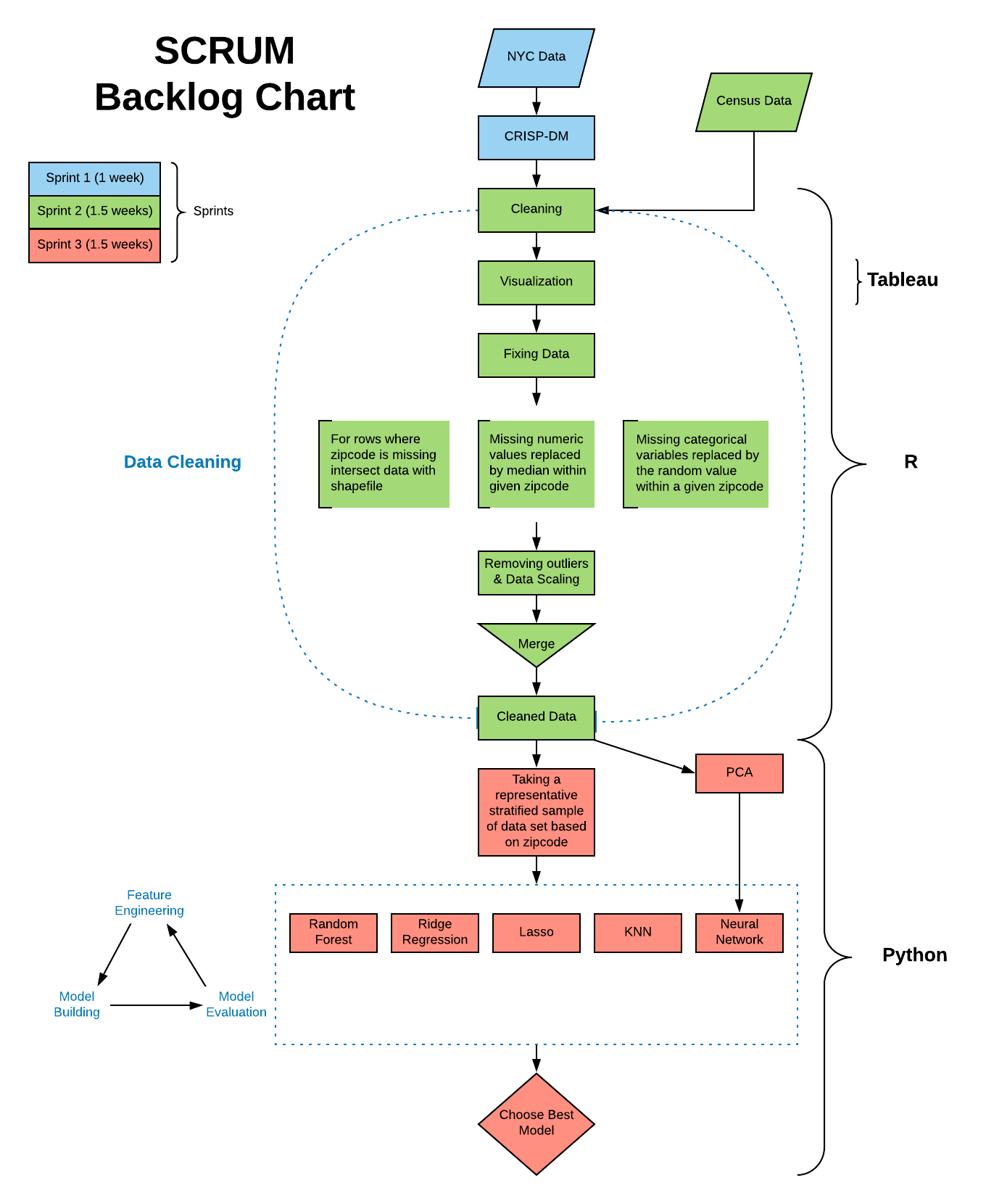
Out of the several algorithms that were tested, it was decided from the results that the best choice is the random forest regressor for predicting assessland and neural network for predicting assesstot. However, these models give a general idea about the properties in NYC, but it is not enough information to make an important business decision. In order for the models to be accurate and effective, more information would need to be considered instead of just free publicly available data. For deployment it would be noted to include easy update options for the system to account for additional data sources that are not publicly available.

In order to effectively deploy the models, there would need to be necessary guidelines to ensure the projects success. Areas such as storage, documentation, configuration, flexibility, environments, platforms, and ease of use would need to be discussed. Storage infrastructure, or a contract with a cloud computing service like Amazon Web Services, would be crucial to hold the vast amounts of data collected from different sources. When considering between the choices, the decision has to be based on cost, velocity, and volume to have the most effective outcome. In the case of storage infrastructure, it may be more cost effective to outsource to a third-party service like AWS instead of deploying in-house infrastructure. Considering all three, the best strategy is to have enough storage with a large volume and the ability to pull the data quickly. All of the code used for the model should be well documented for the ability to transfer to future developers who need to update or fix the code.

The system should be flexibly configurable to change parameters and add predictors in an easy and efficient. When referring to changing parameters and adding predictors, there would first need to be a managed environment that can account for testing before the actual deployment. Implementing a managed testing environment would allow for mistakes without affecting the model that is already in place. A hosting platform that guarantees high availability and ease of use for the clients would be crucial for the success of the deployment. When researching which hosting platforms to use the client has to be kept in mind to account for an easy and user friendly outcome.

# SCRUM Backlog/Sprints

This analysis has utilized the SCRUM backlog/sprint process in an attempt to simplify the project tasks. It has been divided into three sprints. The first sprint involves the identification of the data and the CRISP-DM, which gives a layout of what the approach should be throughout the analysis which have been followed throughout this report. The second sprint involves all of the aspects of cleaning and merging the data. The third and final sprint is dominated by the modeling processes to be considered and evaluated. The visual diagram is shown on the next page.

******

# Appendix

***Table 1. NYC Data description***

|  |  |
| --- | --- |
| **References** |  |
| *Variables to Keep* | *Variables Deleted* |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column Alias** | **Datatype** | **Column Name** | **Issue** | **Description** | **Solution** |
| borough | Categorical | Borough | There were some erroneous levels present for this factor. | Borough. | The boroughs of interest are 5, so the rows that have other values were dropped. This column was dropped at the end since we have borocode which has numeric values for each borough, unlike this column which has text. |
| block | Categorical | Block | It has 13,968 levels and repeats across boroughs. | Block. | Combining this variable with borocode and lot would make it almost just an identifier. Also, combining it with only borocode would provide too many levels that provide redundant location information that can be gleaned from columns such as firecomp. Hence, this variable was dropped. |
| lot | Categorical | Lot | It has 2,521 levels and repeats across blocks. | Lot. | Combining this variable with borocode and lot would make it almost just an identifier. Also, combining it with only block would provide too many levels that provide redundant location information that can be gleaned from columns such as firecomp. Hence, this variable was dropped. |
| cd | Categorical | Community District | 11 Nas | Community district. It has 76 levels. | Filled NAs with random cd values of the corresponding zip codes. |
| ct2010 | Categorical | Census Tract | Merged zip code with census data, so this column is not needed. | 2010 census tract. | Deleted. |
| cb2010 | Categorical | Census Block | Merged zip code with census data, so this column is not needed. | 2010 census block. | Deleted. |
| schooldist | Categorical | School District | 78 NAs | School district. It has 32 levels. | NAs filled with random schooldist values of the corresponding zip codes. |
| council | Categorical | City Council District | 11 NAs | City council district. It has 51 levels. | Filled NAs with random council values within the same zip codes. |
| zipcode | Categorical | Zip code | 20,535 Nas | Zip code. It has 216 levels. | NAs filled with values after matching them to a shapefile in QGIS. |
| firecomp | Categorical | Fire Company | 102 blank values. | Fire company. It has 348 levels. | Blanks replaced with random firecomp values within the same zip codes. |
| policeprct | Categorical | Police Precinct | 83 NAs | Police precinct. It has 82 levels. | Filled NAs with random policeprct values within the same zip codes. |
| healtharea | Categorical | Health Area | 82 NAs | Health area. It has 228 levels. | NAs filled with random healtharea values of the corresponding zip codes. |
| sanitboro | Categorical | Sanitation District Boro | 259 NAs | Borough of the sanitation district that services that tax lot. It has 5 levels. | Filled NAs with random sanitboro values within the same zip codes. |
| sanitdist | Categorical | Sanitation District Number | 259 NAs | Sanitation district that services that tax lot. It has 27 levels. | Filled NAs with random sanitdistrict values within the same zip codes. When the corresponding zip code had only NAs for sanitdistrict, these NAs were replaced with random sanitdistrict values of the corresponding boroughs. |
| sanitsub | Categorical | Sanitation Subsection | 404 blank values. | Subsection of the sanitation district that services that tax lot. It has 62 levels. | Blanks replaced with random sanitsub values within the same zip codes. When the corresponding zip code had only NAs for sanitsub, these NAs were replaced with random sanitsub values of the corresponding boroughs. |
| address | Text | Address | Provides redundant location information. | Address. | Deleted. |
| zonedist1 | Categorical | Zoning District 1 | 967 blank values. | Zoning district classification. It has 5 levels. | Blanks replaced with random zonedist1 values of the corresponding zip codes. Originally had many levels that provided unnecessary details. Replaced with levels R, C, M, MR, and BPC for residential, commercial, manufacturing, mixed-manufacturing and residential, and Battery Park City respectively. |
| zonedist2 | Categorical | Zoning District 2 | 837565 blanks. | Zoning classification occupying 2nd greatest percentage of the tax lot's area. | Deleted since greater than 90% contains blanks. |
| zonedist3 | Categorical | Zoning District 3 | 857164 blanks. | Zoning classification occupying 3rd greatest percentage of the tax lot's area. | Deleted since greater than 90% contains blanks. |
| zonedist4 | Categorical | Zoning District 4 | 857340 blanks. | Zoning classification occupying 4th greatest percentage of the tax lot's area. | Deleted since greater than 90% contains blanks. |
| overlay1 | Categorical | Commercial Overlay 1 | 782706 blanks. | Commercial overlay. | Deleted since greater than 90% contains blanks. |
| overlay2 | Categorical | Commercial Overlay 2 | 857191 blanks. | Commercial overlay occupying 2nd greatest percentage of the tax lot's area. | Deleted since greater than 90% contains blanks. |
| spdist1 | Categorical | Special Purpose District 1 | 756231 blanks. | Special district. It has 2 levels. | Blank means there is no special purpose. Hence, the column is transformed to 1/0 (special purpose/no special purpose). |
| spdist2 | Categorical | Special Purpose District 2 | 857283 blanks. | Special purpose district occupying 2nd greatest percentage of the tax lot's area. | Deleted since greater than 99% contains blanks. |
| spdist3 | Categorical | Special Purpose District 3 | 857353 blanks. | Special purpose district occupying 3rd greatest percentage of the tax lot's area. | Deleted since greater than 99% contains blanks. |
| ltdheight | Numeric | Limited Height District | 854317 blanks. | Limited height district. | Blank conveys no height limit and hence, this column is transformed to a binary variable 1/0 (Limit/No limit). |
| splitzone | Categorical | Split Boundary Indicator | 836456 'N' value. | Indicates whether a tax lot is split between multiple zoning features. | Deleted since greater than 97% of the tax lots are not split between zones. |
| bldgclass | Categorical | Building Class | 7943 NAs | Describes major use of structures on the tax lot. | Removed since this information can be inferred from the landuse variable. |
| landuse | Categorical | Land Use Category | 2359 NAs | Land use. It has 12 levels. | NAs replaced with 0 wherein 0 stands for unknown category. |
| easements | Numeric | Number of Easements | 853598 0s | Number of unique easements on the tax lot. | Deleted since greater that 99% of the tax lots have 0 easements. |
| ownertype | Categorical | Type of Ownership Code | 823448 blanks. | Type of ownership. | Deleted since greater than 95% is blank. |
| ownername | Text | Owner Name | Unnecessary information. | Name of the tax lot owner. | Deleted since it is not useful for prediction here. |
| lotarea | Numeric | Lot Area | 3153 0s and 144 NAs. | Total area. | Zeroes and NAs replaced with the medians for the corresponding zip codes. When the corresponding zip code has only NAs for lotarea, these NAs were replaced with random lotarea values of the corresponding boroughs. |
| bldgarea | Numeric | Total Building Floor Area | No issues. | Total building floor area. | No issues. |
| comarea | Numeric | Commercial Floor Area | 700290 0s and 47716 NAs. | Area allocated for commercial use. | Deleted since nearly 90% is either 0 or NA. |
| resarea | Numeric | Residential Floor Area | 50940 0s and 47716 NAs. | Area allocated for residential use. | Deleted since it doesn't make sense to replace with the medians in cases where landuse variable might categorize it as non-residential. |
| officearea | Numeric | Office Floor Area | 783667 0s and 47716 NAs. | Area allocated for office use. | Deleted since greater than 95% is either 0 or NA. |
| retailarea | Numeric | Retail Floor Area | 745750 0s and 47716 NAs. | Area allocated for retail use. | Deleted since 90% is either 0 or NA. |
| garagearea | Numeric | Garage Floor Area | 799825 0s and 47716 NAs. | Area allocated for garage use. | Deleted since greater than 98% is either 0 or NA. |
| strgearea | Numeric | Storage Floor Area | 801927 0s and 47716 NAs. | Area allocated for storage use. | Deleted since greater than 99% is either 0 or NA. |
| factryarea | Numeric | Factory Floor Area | 802538 0s and 47716 NAs. | Area allocated for factory, warehouse or loft use. | Deleted since greater than 99% is either 0 or NA. |
| otherarea | Numeric | Other Floor Area | 790856 0s and 47716 NAs. | Area allocated for other uses. | Deleted since greater than 97% is either 0 or NA. |
| areasource | Categorical | Total Building Floor Area Source Code | Unrequired information. | Methodology used to determine the tax lot's total building floor area. | Deleted since this data is not required for prediction. |
| numbldgs | Numeric | Number of Buildings | 144 NAs | Number of buildings. | Replaced NAs with medians of corresponding zip codes. |
| numfloors | Numeric | Number of Floors | 144 NAs | Number of floors for the tallest building on the tax lot. | Replaced NAs with medians of corresponding zip codes. |
| unitsres | Numeric | Residential Units | No issues. | Sum of residential units in all buildings on the tax lot. | No issues. |
| unitstotal | Numeric | Total Units | No issues. | Sum of residential and non-residential units in all buildings on the tax lot. | No issues. |
| lotfront | Numeric | Lot Frontage | 144 NAs | Frontage. | Replaced NAs with medians of corresponding zip codes. |
| lotdepth | Numeric | Lot Depth | 1928 NAs | Depth. | Replaced NAs with medians of corresponding zip codes. |
| bldgfront | Numeric | Building Frontage | 144 NAs | Building frontage along the street. | Replaced NAs with medians of corresponding zip codes. |
| bldgdepth | Numeric | Building Depth | 144 NAs | Building depth. | Replaced NAs with medians of corresponding zip codes. |
| ext | Categorical | Extension Code | 507555 blanks. | Extension on the lot or garage. It has 2 levels. | It originally had 4 levels that were converted to 2 levels. Zeros for blanks (no extensions) and 1 for extensions (E,EG, and G). |
| proxcode | Categorical | Proximity Code | Too many levels that provide extra details. | Physical relationship of the building to neighboring buildings. It has 3 levels. | Originally had 4 levels. Semi-attached and attached converted to 2 for attached. 1 stands for detached and 0 for unavailable. |
| irrlotcode | Categorical | Irregular Lot | Blank is a level with no counts. | Irregularly shaped or not. | Blank level dropped. |
| lottype | Categorical | Lot Type | No issues. | Location in relationship to another tax lot or the water. | No issues. |
| bsmtcode | Categorical | Basement Type/Grade | Extra information. | Describes building's basement. | Deleted since it provides details that would be hard to use as a meaningful predictor. |
| assessland | Numeric | Assessed Land Value | 3489 0s | Assessed land value. | As this is a target variable, all values equal to 0 were deleted. |
| assesstot | Numeric | Assessed Total Value | No issues. | Assessed total value. | No issues. |
| exemptland | Numeric | Exempt Land Value | Extra information. | Exempt land value. | Deleted since tax exemption related to factors other than the location alone (such as the business present, …). |
| exempttot | Numeric | Exempt Total Value | Extra information. | Exempt total value. | Deleted since tax exemption related to factors other than the location alone (such as the business present, …). |
| yearbuilt | Numeric | Year Built | 39614 0s and 1 value as 2040. | Year construction of the building was completed. | Row containing the year 2040 deleted. 0s replaced by mode value, which is the year 1920. |
| yearalter1 | Numeric | Year Altered 1 | 763568 0s | If there was only one alteration, this is the date that alteration began. If there were many alterations, this is the date the second most recent alteration began. | If year2 is empty, then take year 1 value. If both are empty, put 0. Here, 0 means there wasn't any modification/renovation. These 2 columns were deleted and instead last\_modif was created. |
| yearalter2 | Numeric | Year Altered 2 | 838572 0s | The year the most recent alteration began. It is 0 if there was only one alteration. |
| histdist | Text | Historic District Name | 823561 blanks. | Historic district. | Deleted since greater than 96% is blanks. |
| landmark | Text | Landmark Status | 852457 blanks | Indicates whether the tax lot contains an individual landmark building, an interior landmark building, or both. | Deleted since greater than 99% is blanks. |
| builtfar | Numeric | Built Floor Area Ratio | Extra information since we can calculate this value from other variables. | Total building floor area divided by the tax lot area. | Deleted. |
| residfar | Numeric | Maximum allowable residential FAR | No issues. | Maximum allowable residential floor area ratio. | No issues. |
| commfar | Numeric | Maximum allowable commercial FAR | No issues. | Maximum allowable commercial floor area ratio. | No issues. |
| facilfar | Numeric | Maximum allowable community facility FAR | No issues. | Maximum allowable community facility floor area ratio. | No issues. |
| borocode | Categorical | Boro Code | No issues. | Borough. It has 5 levels. | No issues. |
| bbl | Categorical | Borough, Tax Block & Lot | Extra information. | Concatenation of borough code, tax block, and tax lot. | Deleted because it acts nearly similar to an identifier. |
| condono | Categorical | Condominium Number | 845526 0s and 39 NAs | Condominium number assigned to the complex. | Deleted since greater than 99% is 0s or NAs. |
| tract2010 | Categorical | Census Tract 2 | Merged zip code with census data, so this column is not needed. | 2010 census tract. | Deleted. |
| xcoord | Numeric | X Coordinate | 77 NAs | X coordinate location of the tax lot. | These 77 rows were dropped. Based on X and Y coordinates, the missing values in zip code were found. |
| ycoord | Numeric | Y Coordinate | No issues. | Y coordinate location of the tax lot. | No issues. Based on X and Y coordinates, the missing values in zip code were found. |
| zonemap | Categorical | Zoning Map | Extra information. | Department of City Planning Zoning Map Number associated with the tax lot's coordinates. | Deleted since this is not useful for prediction. |
| zmcode | Categorical | Zoning Map Code | Extra information. | Indicates whether a tax lot is on the border of 2 or more zoning maps. | Deleted since this is not useful for prediction. |
| sanborn | Categorical | Sanborn Map | Extra information. | Sanborn Map Company number associated with the tax block and lot. | Deleted since this is not useful for prediction. |
| taxmap | Categorical | Tax Map | Extra information. | Department of Finance paper tax map volume number associated with the tax block and lot. | Deleted since this is not useful for prediction. |
| edesignum | Categorical | E-Designation Number | Too many levels (105) that provide extra details. | Indicates the presence of an environmental requirement pertaining to potential hazardous materials. | Converted to 2 levels 0 and 1 wherein 0 indicates the absence of potential hazardous materials. |
| appbbl | Categorical | Apportionment bbl | Extra information. | Originating borough, block, and lot from the apportionment prior to the merge, split, or property's conversion to a condominium. | Deleted since it is not useful for prediction. |
| appdate | Numeric | Apportionment date | Extra information. | Date of apportionment. | Deleted since it is not useful for prediction. |
| mappluto\_f | Categorical | Map pluto ID | 853863 NAs | Map pluto ID. | Deleted since it contains only NAs. |
| plutomapid | Categorical | Pluto - Dtm Base Map Indicator | Extra information. | Indicates whether the tax lot is in the Pluto file or MapPluto file. | Deleted since it is not useful for prediction. |
| version | Text | Version Number | Only 1 level, and it is extra information. | Version number for this release of Pluto. | Deleted since it is not useful for prediction. |
| healthcenterdistrict | Categorical | Health center district | 74 NAs | Health center district. | Filled NAs with random healthcenterdistrict values within the same zip codes. |
| firm07\_flag | Categorical | 2007 Flood insurance rate map indicator | Old since more recent 2015 data available. | Indicates whether a tax lot falls within the 1% annual chance floodplain. | Deleted. |
| pfirm15\_flag | Categorical | 2015 Preliminary flood insurance rate map indicator | 789461 NAs | Indicates whether a tax lot falls within the 1% annual chance floodplain. | NAs replaced with 0s wherein 0 means there is no 1% chance of flooding. |
| rpaddate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| dcasdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| zoningdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| landmkdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| basempdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| masdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| polidate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |
| edesigdate | Date | Erroneous column not in PDF. | Only 1 date value. | Erroneous column not in PDF. | Deleted. |

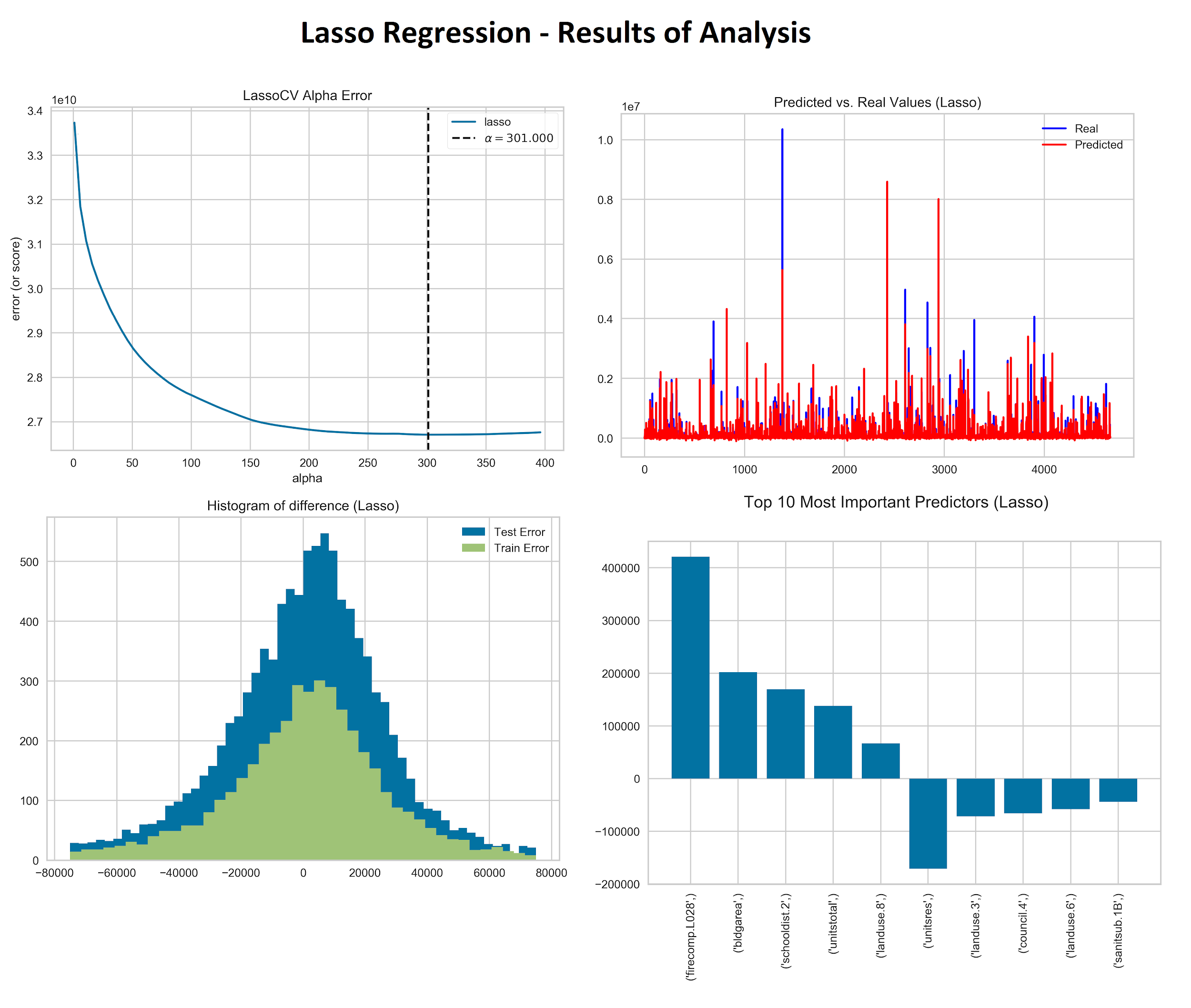
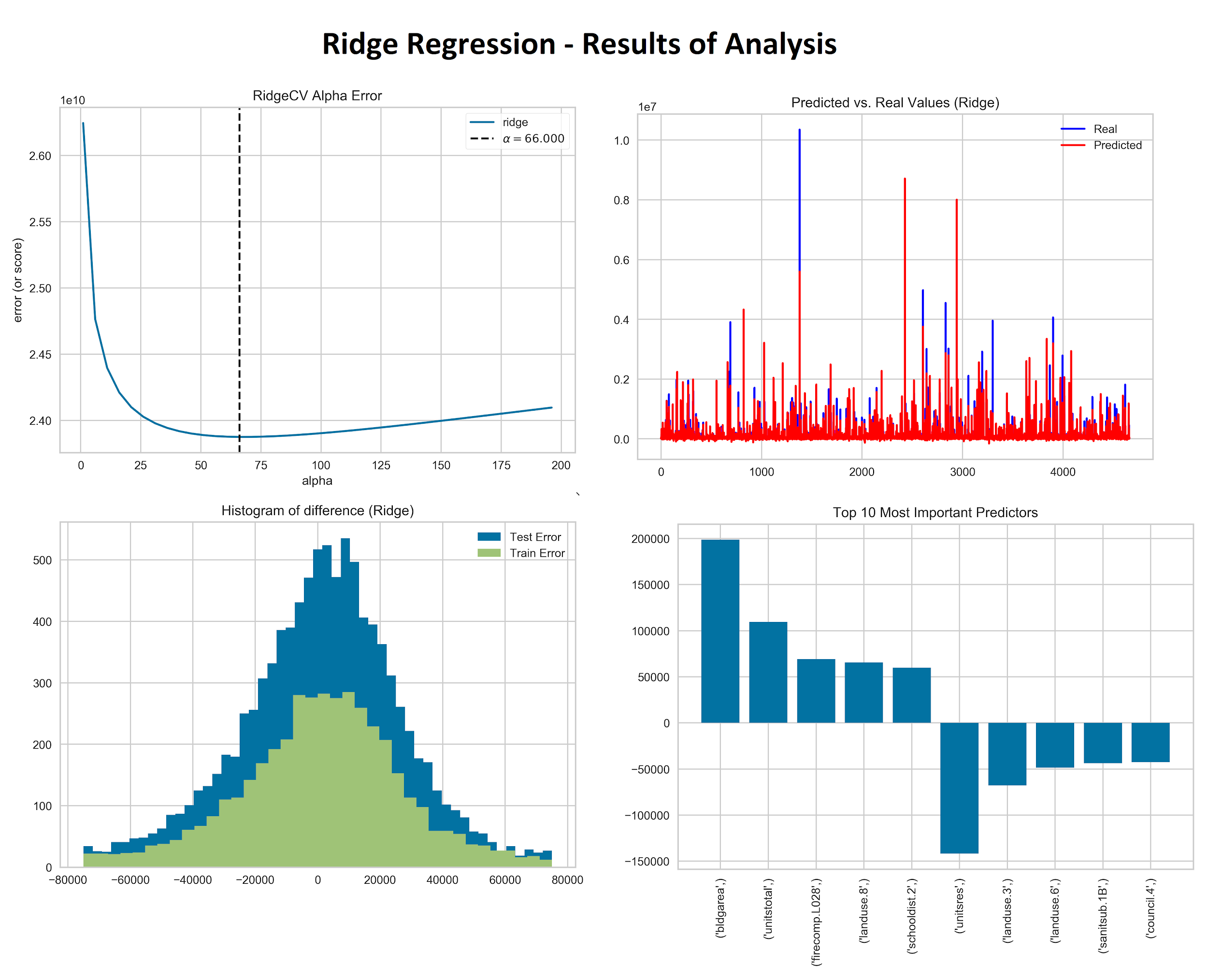
Transformed Variable:

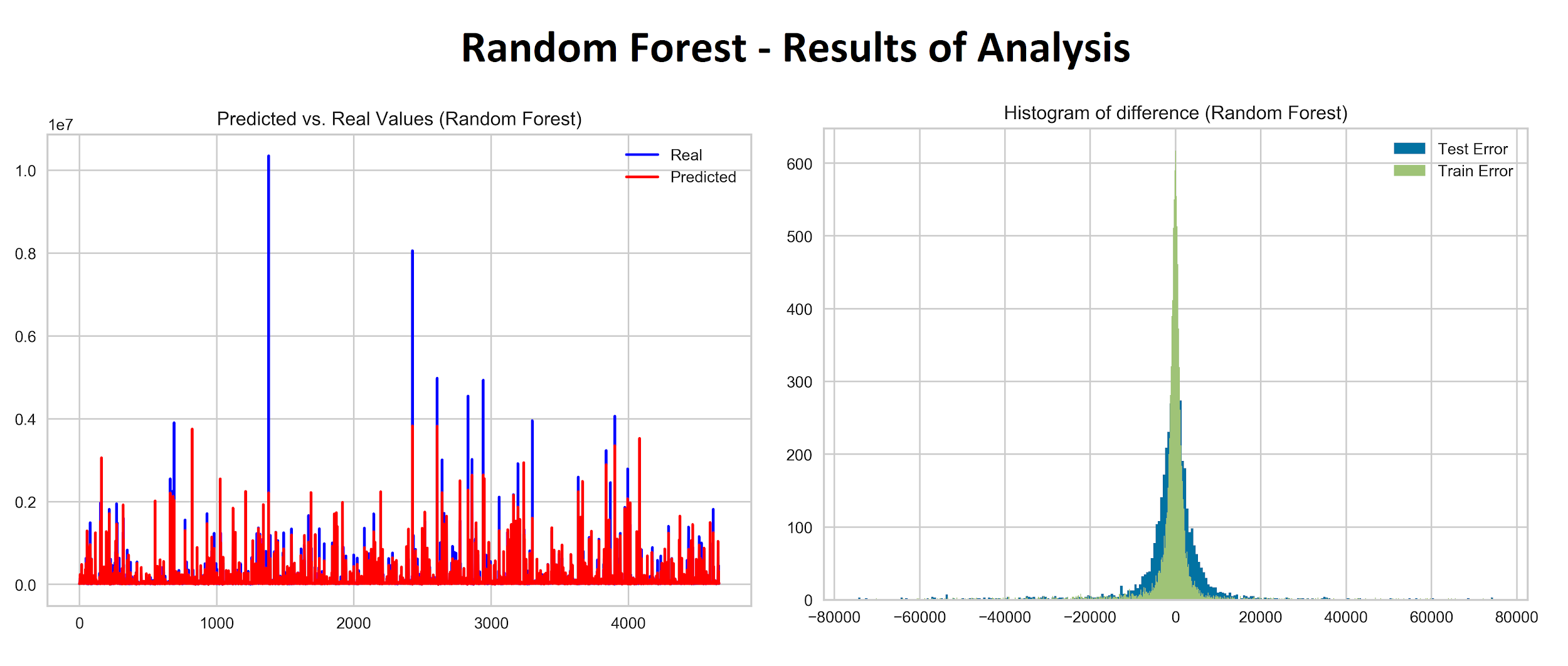
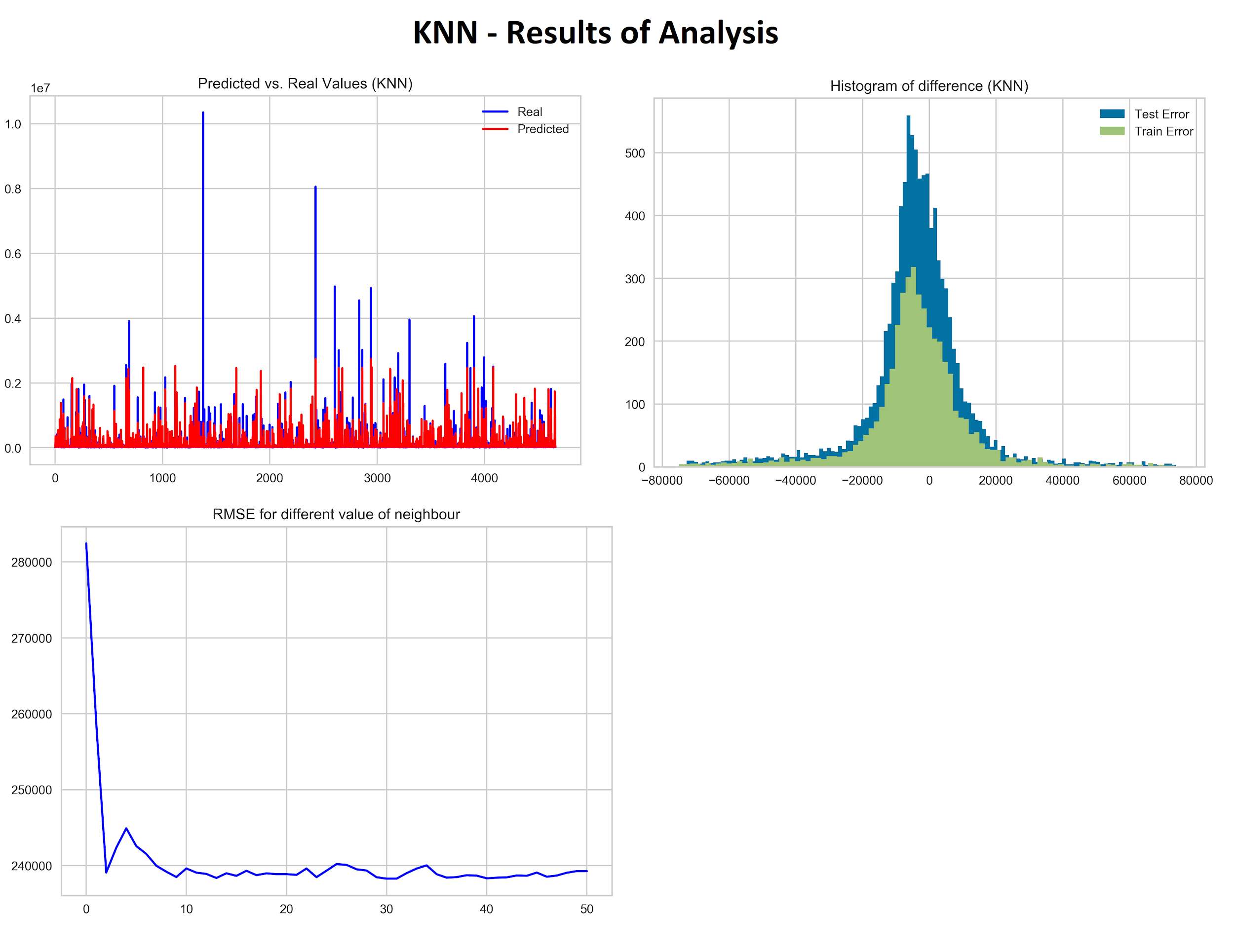
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| yearalter | Numeric | Year altered | No issues. | Year the most recent alteration began. | Newly created from columns yearalter1 and yearalter2. |

Below is the data types and columns used from census data for 2010 merged to the original dataset by zip code.

***Table 2. Census data***

|  |  |  |
| --- | --- | --- |
| Column | Datatype | Description |
| mean\_income\_fam | Numeric | Estimate of the mean of the total income of families. |
| mean\_income\_non | Numeric | Estimate of the mean of the total income of non-families. |
| zipcode | Categorical | Zip code. |

**Figure *2. Process of Analysis for Variable ‘Assesstot’***

1. <https://data.cityofnewyork.us/City-Government/Primary-Land-Use-Tax-Lot-Output-PLUTO-/xuk2-nczf> [↑](#footnote-ref-0)
2. [https://factfinder.census.gov/faces/tables](https://factfinder.census.gov/faces/tableservices/jsf/pages/productview.xhtml?pid=ACS_10_5YR_S1901&prodType=table) [↑](#footnote-ref-1)