NYC Property Price

ML Regression Models in PySpark

SCS 3252-020 Big Data Management Systems and Tools

March 31, 2020

Kiran Sohi

INTRODUCTION

New York City is a premium location for real estate purchase in the world. Properties in New York have always been in demand. It is a city where people from all walks of life live together, work together and enjoy New York. New York also has great historical significance since the day the Dutch surrendered the colony, then called New Amsterdam, to the English who renamed it New York after the Duke of York.[1] New York is also a famous travel destination that includes such sights as the Brooklyn Bridge, Time Square and Broadway theatres.[2] The restaurants are superb and the people are lively.

The purpose of this analysis is to determine if rolling sales dataset provided by the New York City Department of Finance on kaggle can help predict the value of a property using a year's worth of raw transactions.[3] Note that the dataset only encompasses one year of property sales (2016-2017) and requires a significant amount of data wrangling before ML regression models can be constructed.

OBJECTIVES

The purpose of this analysis is to:

1. Review the data provided in the dataset by NYC Department of Finance
2. Wrangle data to build a dataset with completed values that are normally distributed and usable for modelling
3. Determine which features will be important for a regression model analysis and ensure proper normalization and treatment for modelling
4. Construct several regression models to improve the predictive model for NYC properties based on the dataset provided

Data Cleaning

Dataset used in this analysis was downloaded from kaggle and was reviewed in Microsoft Excel. The column names were all capital letters and long which is an impediment to coding. Hence, the column names were amended to be shorter and lower case. The dataset was uploaded into databricks using the Data > Add Data method. The script for upload created by databricks was used to upload the file into the databricks notebook for review. A dataframe with inferred schema and first row as header was created for use in the notebook.

A schema review of the dataframe provided data type for the following data columns:

[1] https://www.softschools.com/facts/13_colonies/new_york_colony_facts/2043/
[2] http://www.aviewoncities.com/nyc/nycattractions.htm
[3] https://www.kaggle.com/new-york-city/nyc-property-sales

- |--_c0: integer (nullable = true)
- |-- borough: integer (nullable = true)
- |-- neighborhood: string (nullable = true)
- |-- blg_class: string (nullable = true)
- |-- tax_class: string (nullable = true)
- |-- block: integer (nullable = true)
- |-- lot: integer (nullable = true)
- |-- ease_ment: string (nullable = true)
- |-- bldg_class: string (nullable = true)
- |-- address: string (nullable = true)
- |-- apt: string (nullable = true)
- |-- zip: integer (nullable = true)
- |-- res_unts: integer (nullable = true)
- |-- com_unts: integer (nullable = true)
- |-- tot_unts: integer (nullable = true)
- |-- land_sqft: string (nullable = true)
- |-- gross_sqft: string (nullable = true)
- |-- yr_built: integer (nullable = true)
- |-- tax_class_sale: integer (nullable = true)
- |-- bldg_class_sale: string (nullable = true)
- |-- price: string (nullable = true)
- |-- sale_date: timestamp (nullable = true)

Based on the Data Description provided on kaggle, the dataset required wrangling to ensure that it can be used for regression modelling. A basic statistical describe() command was run on the dataframe to provide a general statistical overview of the dataset. There were 84,548 rows in the data set for NY properties. According to kaggle's data summary, *ease_ment* column did not contain any data and the first column, _c0, was a general index and could safely be deleted from the dataset. Some of the features in the dataset were uploaded with incorrect data type inferred by databricks and were updated. For example, price appears as string data type but should be an integer. All duplicate rows in the dataset, if any, were deleted to ensure the data rows remained unique for model development.

Based on the data description, it was clear that the price column needed to be reviewed and cleaned. There were several rows that had a price of $0 and generally low values that represented property transfer from parent to child as opposed to an actual sale of the property. This is important because price is our label or y value for the models. Having a value of $0 or otherwise low value would train the models improperly for future price predictions. The price column also has several outlier values for property sale. These outliers reflected the sale of entire buildings and the prices were very high, in billions of dollars. These data were also deleted to remove outliers so as not to skew the regression model being generated.

Other data[4] in the dataset that needed to be further reviewed and explored included:

[4] https://www1.nyc.gov/assets/finance/downloads/pdf/07pdf/glossary_rsf071607.pdf

- “borough” – name of the borough in which the property is located
- “tax_class” – every property in the city is assigned to one of the four classes based on the use of property
- “blg_class” – building class category represents the type and size of property
- “res_unts” – number of residential units in the building
- “com_unts” - number of commercial units in the building
- “tot_unts” – number total units in the building
- “land_sqft”- land area of the property listed in square feet
- “gross_sqft” - total area of all the floors of building as measured from the exterior surfaces of the outside walls of the building
- “yr_built” – year the structure on the property was built
- “tax_class” – every property in the city is assigned to one of the four classes based on the use of property

(See link in footnote 4 below for more detailed description of the data listed above.)

These data points were selected for further review and analysis because these features are associated with the price of the property for sale. This was a judgement call based on review of the Data Description and all the features in the dataset.

FINDINGS
Data Analysis

“borough”

“borough” was originally listed in categorical form with values 1, 2, 3, 4 and 5. This variable was updated to string and the respective names were inserted: Manhattan, Bronx, Brooklyn, Queens and Staten Island.

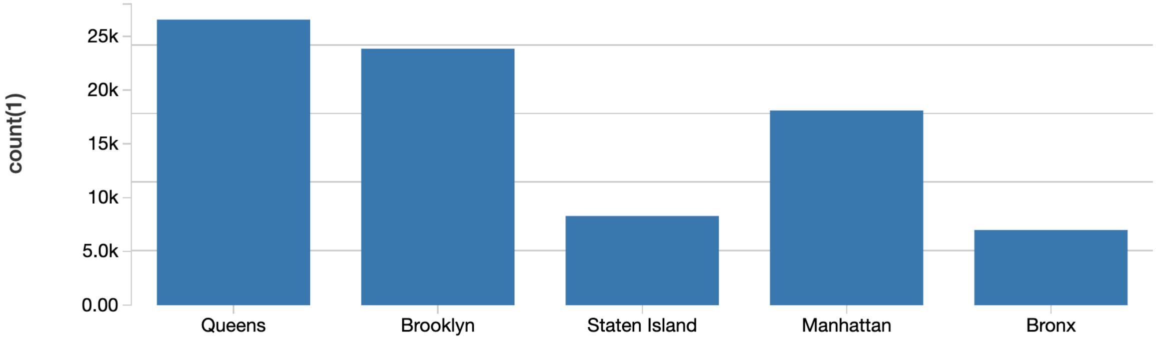


Figure 1: Histogram of properties in the dataset by Borough

Borough of Queens has the majority of the property sales in New York followed by Brooklyn and Manhattan.

“tax_class”

“tax_class” variable was also a categorical variable and identifies the type of dwelling the property is.

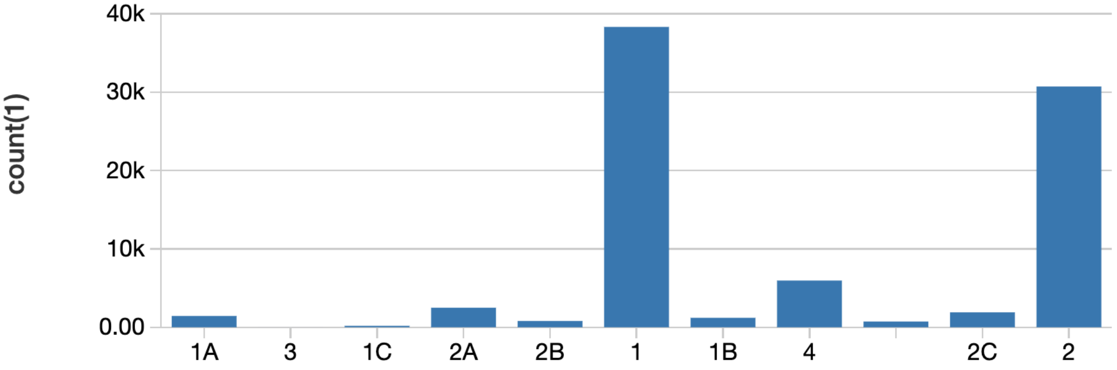


Figure 2: Histogram of properties in the dataset by Tax Class

Majority of the dataset consists of Class 1 and 2 properties. Class 1 properties include mostly residential properties of up to 2 units, example one-, two-, three-family homes as well as small stores or offices with one or two attached apartments. Class 2 includes all other property that is primarily residential like co-ops and condominiums.

“sale_date”

Since the New York dataset is rolling sales data for years 2016 and 2017, a time series plot of the data was reviewed.

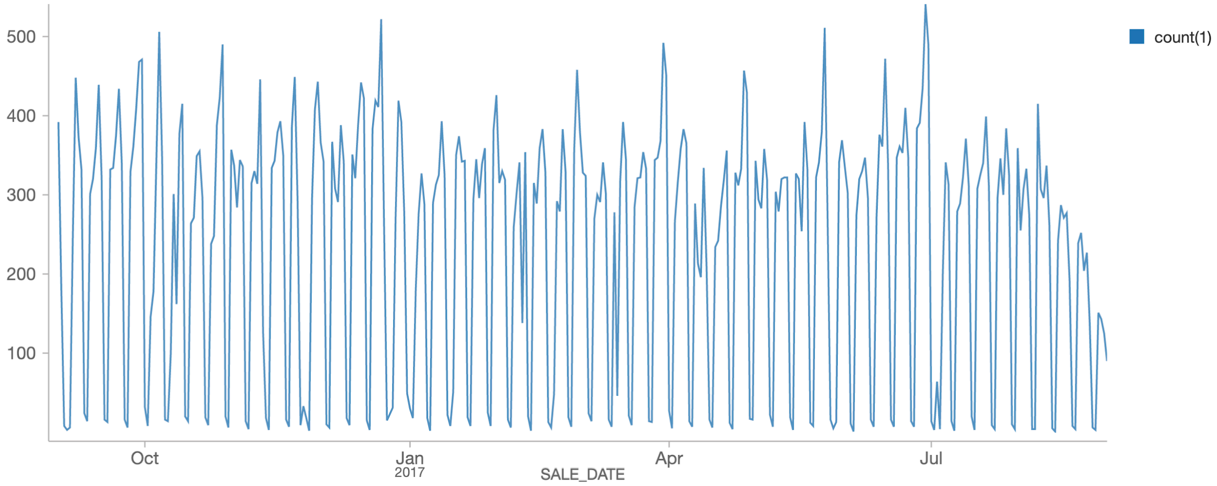
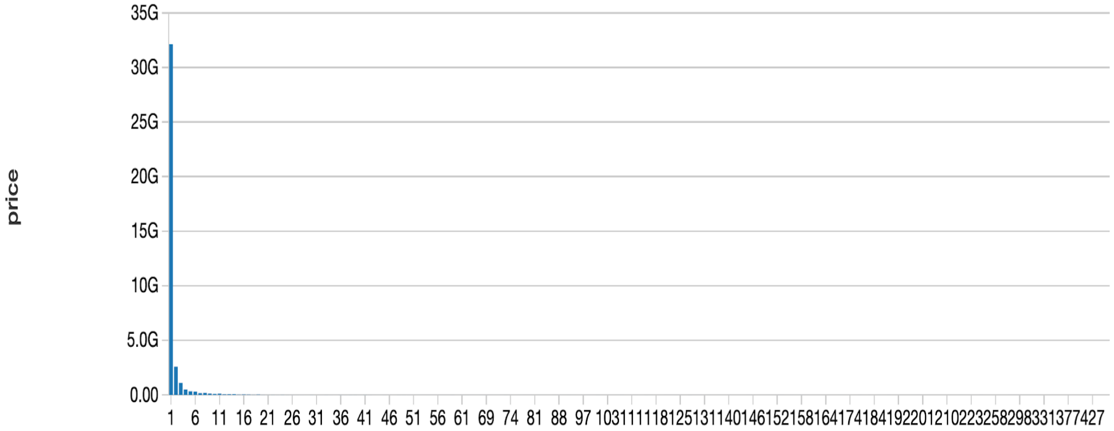


Figure 3: Time Series graph of properties sold in New York for 2016 to 2017

There is a weekly pattern in the sale of properties in New York. On Sunday, there are no property sales in New York. Sales pick up as the week progresses and peak on Wednesday, Thursday and Friday. This analysis was based on a visual review of the plot above.

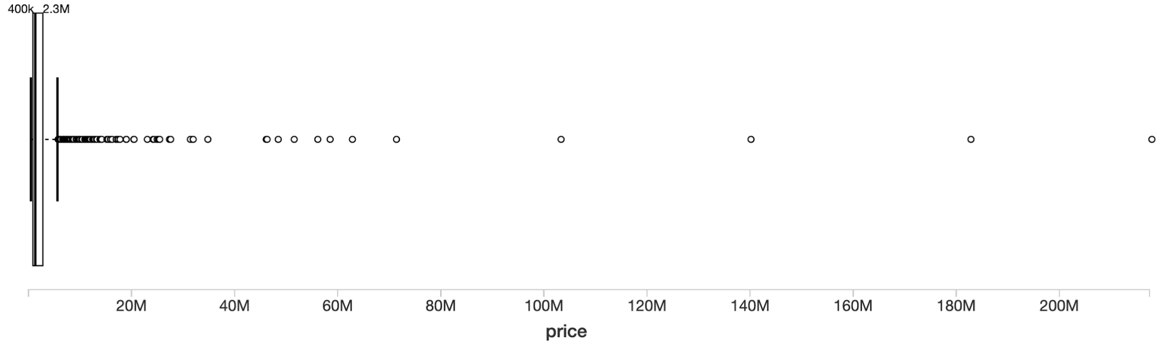
“price”

“price” column is the label for our anticipated regression models and so will need to be complete and meaningful to ensure that the models are provided with appropriate training and test data.



Aggregated (by sum) in the backend.

Figure 4: Bar graph of price variable showing large counts of $0 or low values



Showing sample based on the first 1000 rows.

Figure 5: Box plot of price variable showing skewed dataset towards $0

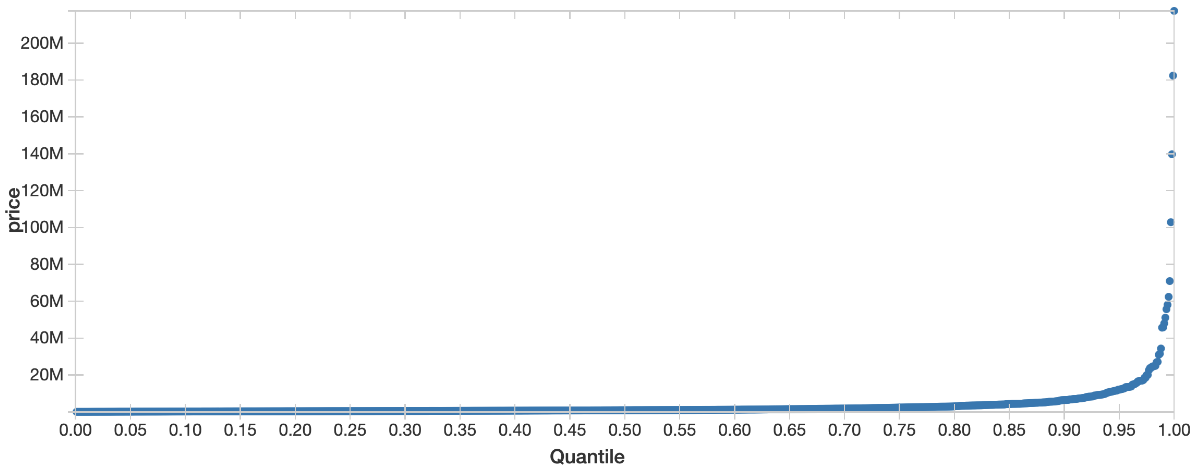
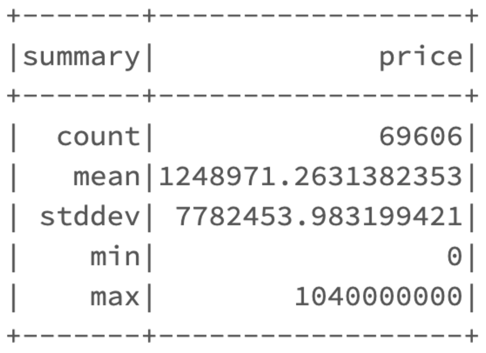


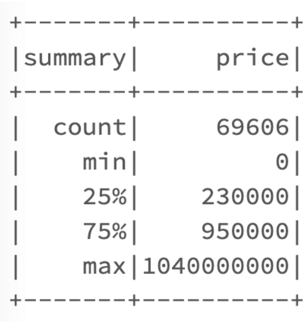
Figure 6: A quantile graph of price showing outlier values

As a start, approximately 14,177 rows containing null values for “price” were removed from the dataset. A summary of the data provided insight into the distribution.

Table 1: Summary statistics for “price”



```
+-------+------------------+
|summary|             price|
+-------+------------------+
|  count|             69606|
|   mean|1248971.2631382353|
| stddev| 7782453.983199421|
|    min|                 0|
|    max|        1040000000|
+-------+------------------+
```



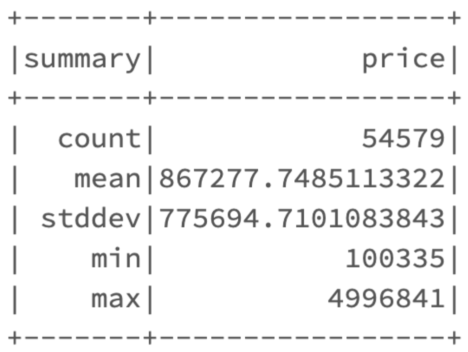
```
+-------+----------+
|summary|     price|
+-------+----------+
|  count|     69606|
|    min|         0|
|    25%|    230000|
|    75%|    950000|
|    max|1040000000|
+-------+----------+
```

As can be seen in Table 1, the second summary table, 50% of the prices fall between $230K and $950K. “price” column contains a lot of sale prices for properties that are close to or equal to $0. This created a dataset that is skewed significantly towards $0 values. These values were expected as it was noted in the data description that a significant portion of the sales in the dataset were property transfer of deeds between parents and children and so were irrelevant. As a result, this data was removed from the dataset.

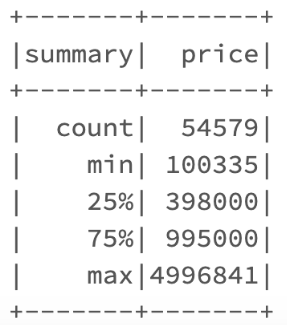
On the higher side of the scale, there were also some extreme outliers for the “price” column. These values were also adjusted to allow for generally a normally distributed dataset.

All data for “price” below $100K and above $5M were removed from the dataset providing the following statistical summary:

Table 2: Summary statistics for “price” after removal of extreme data values



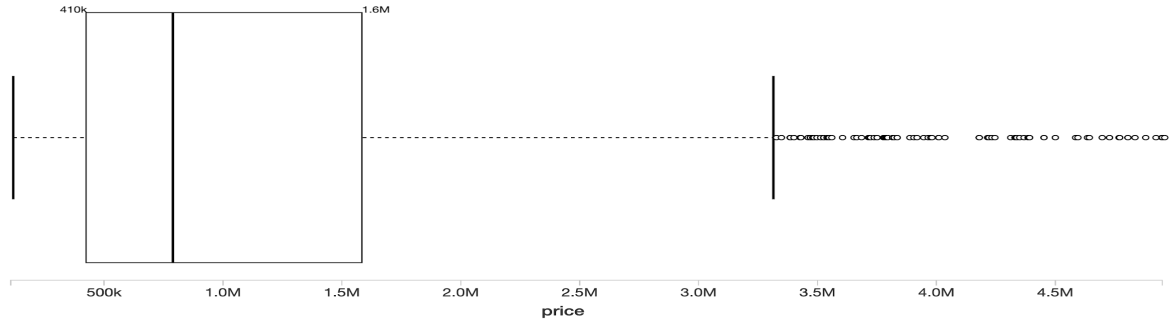
```
+-------+-----------------+
|summary|            price|
+-------+-----------------+
|  count|            54579|
|   mean|867277.7485113322|
| stddev|775694.7101083843|
|    min|           100335|
|    max|          4996841|
+-------+-----------------+
```



```
+-------+-------+
|summary|  price|
+-------+-------+
|  count|  54579|
|    min| 100335|
|    25%| 398000|
|    75%| 995000|
|    max|4996841|
+-------+-------+
```

The summary statistics show that the dataset now contains 54,479 rows and 50% of the data falls within $100K and $995K “price” value.

The box plot in Figure 7 below indicates that the “price” column is distributed normally although there are still some outliers on the higher end. As the goal is to preserve the dataset as much as possible, these outliers were left in. Also, outliers can provide additional information as part of the modelling process and so these outliers were preserved.



Showing sample based on the first 1000 rows.

Figure 7: Box plot of price variable after removal of low and extremely high outliers

“yr_built”

The analysis for the “yr_built” column indicated that most of the buildings in New York were constructed between 1910 and 1960 as shown in Table 3 below. It is likely that these buildings have been renovated and upgraded over the years. There were 3,692 buildings that had no information in the column. These rows were removed from the dataset.

Table 3: Top 5 years for building construction in New York

Year	Buildings Constructed
1910	2,131

1920	3,802
1925	2,760
1930	3,176
1940	1,607
1950	2,072
1960	1,756

Also, the “yr_built” did not seem to be very associative when it comes to the “price” of the property. So a new variable “age_built” was created to identify the age of the property being sold. Total dataset for regression modelling now included 50,887 rows.

“gross_sqft” *&* ***“land_sqft”***

After filtering for null values, it was discovered that there were 17,790 rows with null “gross_sqft” values and 17,641 rows with null “land_sqft” values. Rather than removing these rows from the dataset, the data rows were preserved. A decision was made to use *imputation estimator* to complete these missing values as part of the modelling process, thereby filling in the null values.

“res_unts”*,* ***“com_unts”*** *&* ***“tot_unts”***

In reviewing the boxplots for “res_unts”, “com_unts” and “tot_unts” variables, there seemed to be extreme outlier values in all three data columns. Also, the data seemed to be skewed towards 0 value. And, for all rows, “tot_unts” needed to equal to “res_unts” + “com_unts”. So rows where this calculation was incorrect were removed from the dataset.

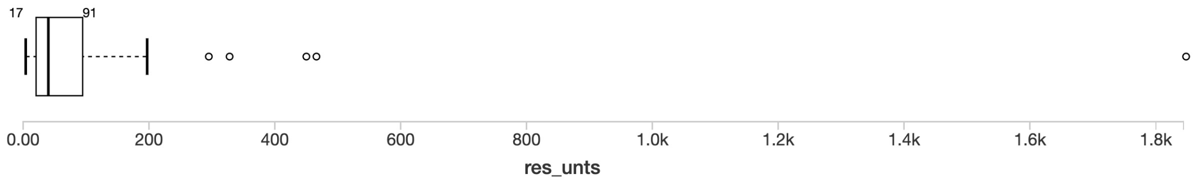


Figure 8: Box plot for “res_unts”

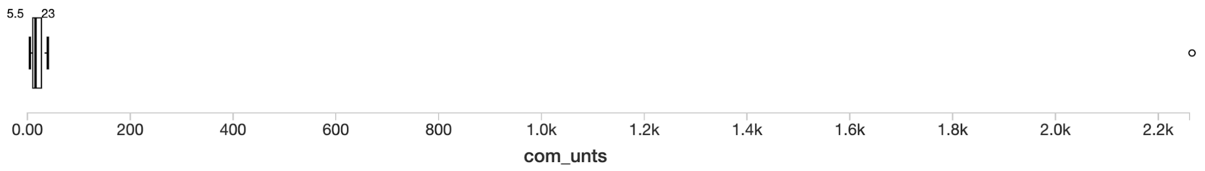


Figure 9: Box plot for “com_unts”

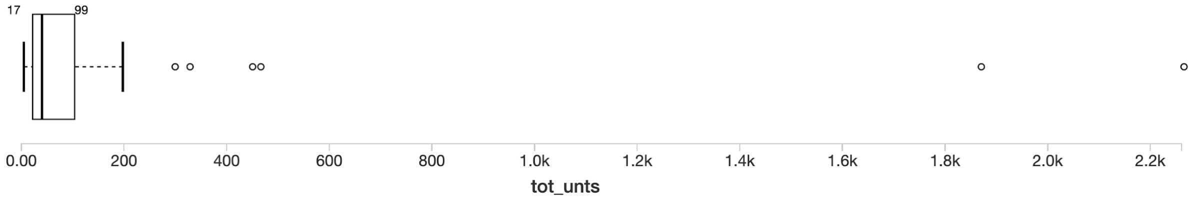


Figure 10: Box plot for “tot_unts”

After removing the data, 50,575 rows were part of the dataset and the “tot_unts” column seems to have a better distribution generally as shown in Figure 11.

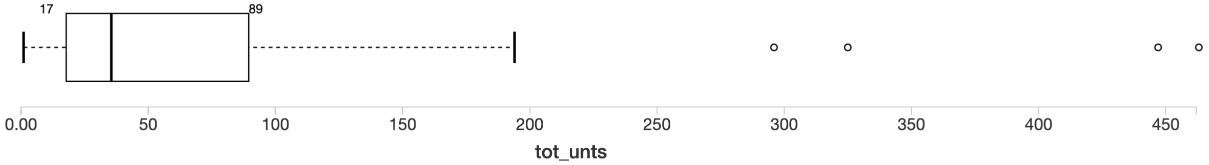


Figure 11: Box plot for “tot_unts” after removing erroneous data

“blg_class”

“blg_class” is a categorical value that shows the Building Class Category with no significant changes necessary. All values in the dataset are complete and useful for regression modelling.

Correlation Analysis

The correlation graph for variables “age_built”, “price”, “tax_class”, “gross_sqft” and “tot_unts” indicates that “price” is weakly correlated with “age_built”, “gross_sqft” and “tot_unts”. This implies that features that are important for regression analysis include “age_built”, “gross_sqft” and “tot_unts”. Remainder of the features are categorical and the correlation is therefore not sufficient for evaluation.

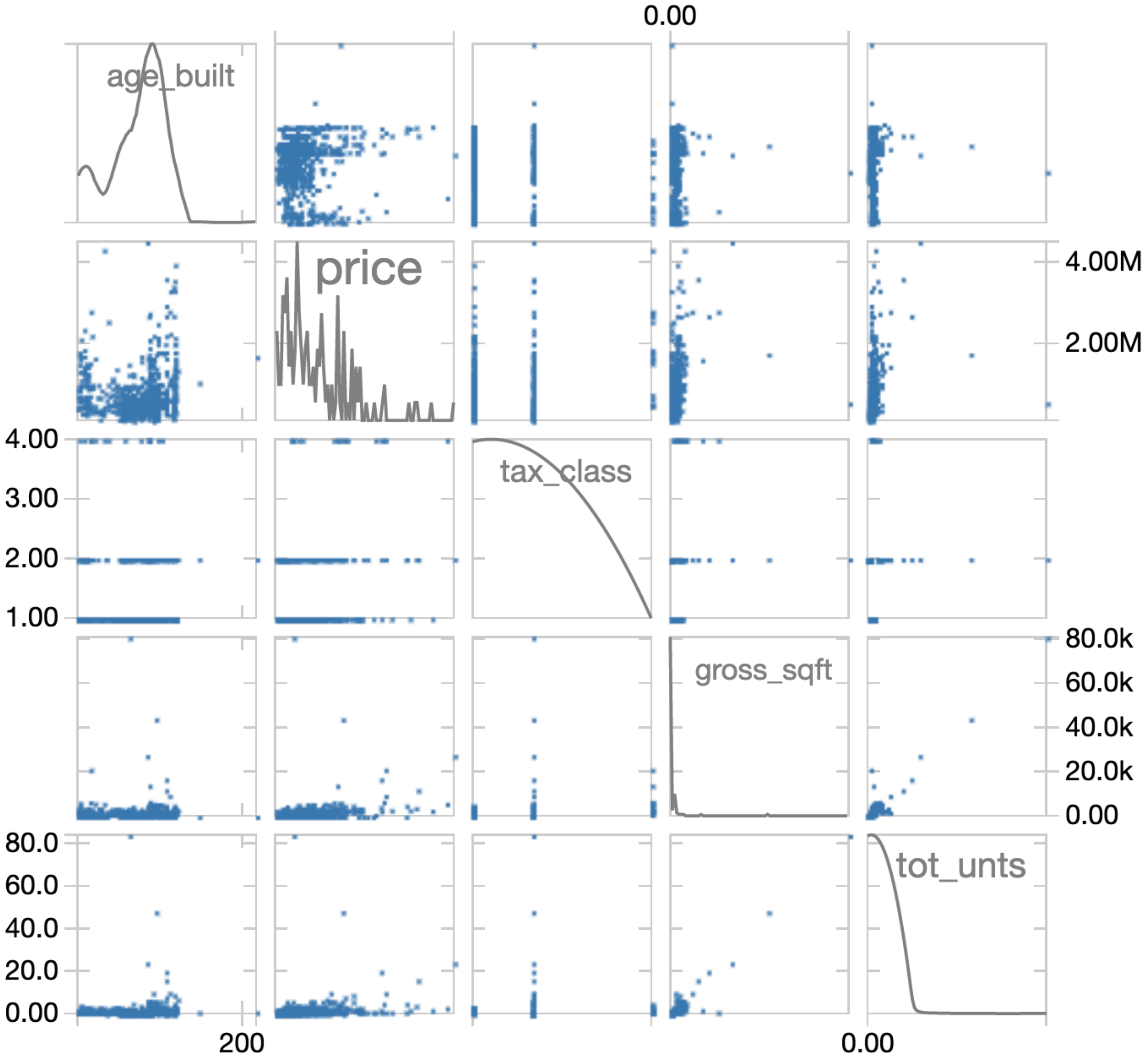


Figure 12: Correlation graph for “age_built”, “price”, “tax_class”, “gross_sqft” and “tot_unts” variables

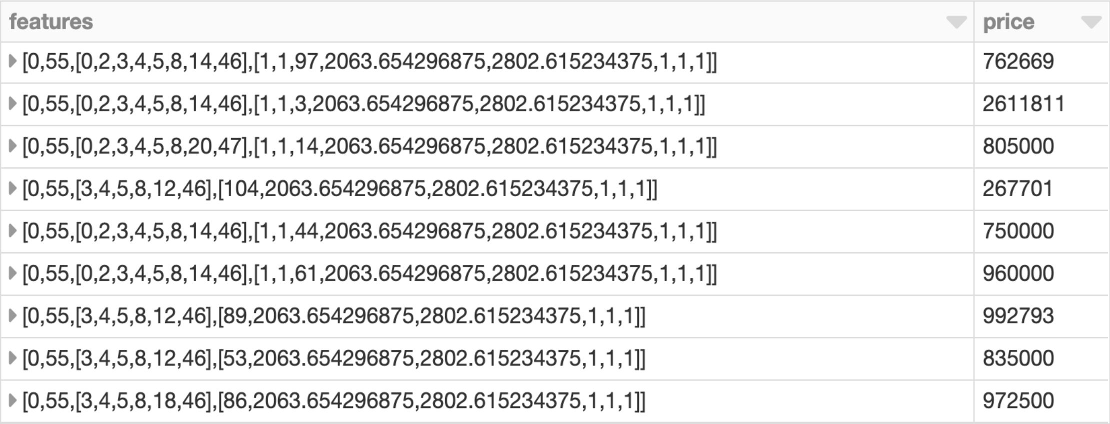
Feature Engineering

The features to be used in the regression model were prepared for model training. As mentioned earlier “gross_sqft” and “land_sqft” were “*imputed*” to ensure that the null values were filled for regression analysis.

Following this, the string features “borough”, “blg_class” and “tax_class” were transformed using StringIndexer and then OneHotEncoder to allow the model for better training of the regression models.

All non-essential columns that were not associative like “neighborhood”, “block”, “lot”, “address”, “apt”, “zip”, etc. were dropped from the feature set.

The features were then converted to vectors using the VectorAssembler method and a new dataset was created with only the vector column named “features” and “price”, the predictive value.



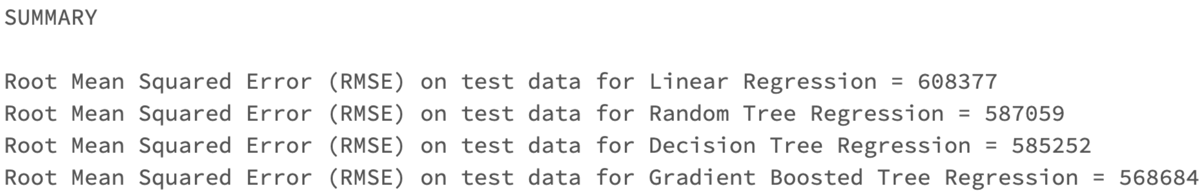
features	price
▸ [0,55,[0,2,3,4,5,8,14,46],[1,1,97,2063.654296875,2802.615234375,1,1,1]]	762669
▸ [0,55,[0,2,3,4,5,8,14,46],[1,1,3,2063.654296875,2802.615234375,1,1,1]]	2611811
▸ [0,55,[0,2,3,4,5,8,20,47],[1,1,14,2063.654296875,2802.615234375,1,1,1]]	805000
▸ [0,55,[3,4,5,8,12,46],[104,2063.654296875,2802.615234375,1,1,1]]	267701
▸ [0,55,[0,2,3,4,5,8,14,46],[1,1,44,2063.654296875,2802.615234375,1,1,1]]	750000
▸ [0,55,[0,2,3,4,5,8,14,46],[1,1,61,2063.654296875,2802.615234375,1,1,1]]	960000
▸ [0,55,[3,4,5,8,12,46],[89,2063.654296875,2802.615234375,1,1,1]]	992793
▸ [0,55,[3,4,5,8,12,46],[53,2063.654296875,2802.615234375,1,1,1]]	835000
▸ [0,55,[3,4,5,8,18,46],[86,2063.654296875,2802.615234375,1,1,1]]	972500

Finally, the dataset was split into train and test for use in regression models.

Regression Models

Four different statistical regression models were created to produce predictive values for price. These models were Linear Regression, Random Tree Regression, Decision Tree Regression and Gradient-Boosted Tree Regression.

The following Root Mean Square Error Summary for each model were obtained:



```
SUMMARY

Root Mean Squared Error (RMSE) on test data for Linear Regression = 608377
Root Mean Squared Error (RMSE) on test data for Random Tree Regression = 587059
Root Mean Squared Error (RMSE) on test data for Decision Tree Regression = 585252
Root Mean Squared Error (RMSE) on test data for Gradient Boosted Tree Regression = 568684
```

CONCLUSION

The ML regression models for New York properties produced predictions that have a high degree of error. However, improvement was seen in Root Mean Square Error (RMSE) based on usage of different (better) statistical models: from Linear Regression: $603K to Gradient

Boosted Tree Regression: $586K. Linear regression R^2 was 0.343199 which means that 34% of the variation in the price data is due to variation in the features data.

If there is to be improvement in the model, more data and other types of models should be generated in future iterations. Following are some opportunities to improve the model for future iterations:

- Improvement in the data fields being collected, perhaps additional criteria needs to be identified that help customers buy property in New York boroughs.
- A lot of data cleansing was required, so perhaps the data can be more cautiously collected or the collection can be automated. It might also make sense to separate the data for residential properties from the commercial properties which would help separate the different types of sales price within the dataset.
- Further data wrangling may definitely be required.
- Other types of statistical models can also be tried to improve the modelling results, like classification modelling using KNN, Decision Tree and even use of Neural Networks may be needed.

In conclusion, regardless of the high RMSE in the models, New York will always remain a city where real estate is well worth owning, if you can afford it.