NYC Property Sale Prediction

APRIL 5, 2019

COURSE PROJECT

3252 – Big Data Management Systems & Tools

Authored by: Mohammad Ali

UNIVERSITY OF
TORONTO

Table of Contents

Introduction

This report was produced as a project requirement for the course “3252 Big Data Management Systems & Tools” at the University of Toronto, School of continuing studies. A real dataset of New York City property sale was downloaded from Kaggle - the subject of this data analysis is to generate prediction models using spark framework in community cloud’s databricks with pysharp language.

Objective

The objective of this course project is to design and implement a sales prediction model by performing a regression analysis utilizing all sales information provided by NYC property data.

Data Overview

This dataset is a record of every building or building unit (apartment, etc.) sold in the New York City property market over a 12-month period. The inspiration is to discover about New York City real estate by looking at a year's worth of raw transaction records to spot trends in the market or build a model that predicts sale value in the future.

The data source contains 84.5K records with 21 features including target feature “Sale Price” for the period from September 2016 to September 2017.

Data Preparation

The dataset had several issues with the records that needed to be addressed prior to analysis. The dataset contains about 08 categorical features and 09 numerical features. There were 1358 duplicate records were removed.

Fixing Schema:

The data load from CSV file were fixed as string data type, later the scheme of whole data was fixed. [Schema].

Fixing Null Records:

The 21% sale price records were null values, which were removed from the dataset. The land square feet were 39% and gross square feet were 41% were applied mean value. The pie chart distribution for null values in appendix [Missing Records].

Sale Price Correlation:

A general correlation analysis was conducted to better understanding the relationship between features and how they may influence the regression optimization process. The following heat map [Figure 1] illustrates the overall correlation between all the subject features.

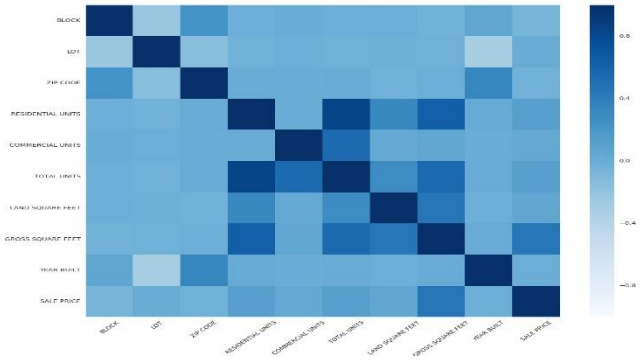


Figure 1 Represents the correlation of different features with Sale Price

Data Analysis

The most important data analysis is the target variable “Sale Price”, If we look at [Figure 2] the box plot for “Sale Price” in whole data set.

Figure 2 Box Plot (Sale Price)

Figure 3 Sale Price Distribution Rightly Skewed.

There are outliers which is because of very high prices in the dataset. The data is filter as “Sales Price” greater than 100,000 and less than 5,000,000, which has removed 15028 records.

Skewness of Sale Price:

The Sale Price is highly right skewed [Figure 3], it was applied log transform it so that it gives better results. Now it is symmetry and thus it is normalised [Figure 4]. The skewness of Sale Price was 2.34 down to 0.1989 in normalized form.

Figure 4 Normalized Sales Price

Square Feet Analysis:

The box plot to analysis Gross and Land square feet data in [Figure 5]. As we noticed in these box plots that there are almost no records more than 10,000, so have filter those records which removed 541 from whole dataset.

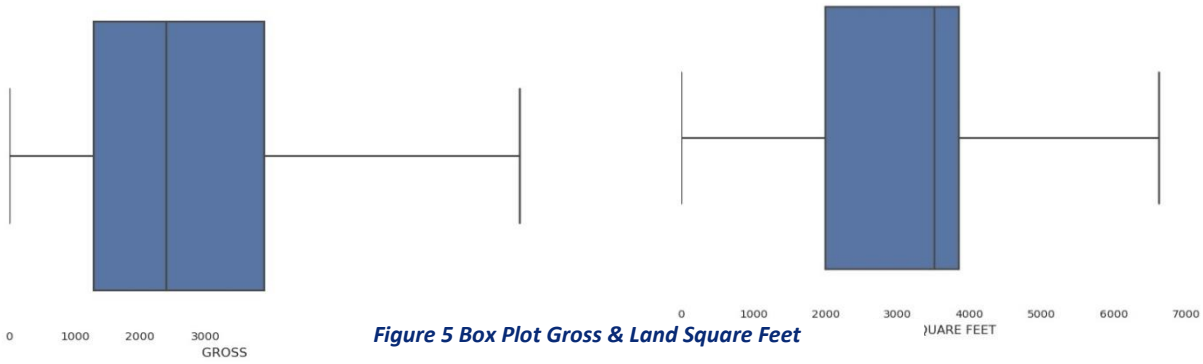


Figure 5 Box Plot Gross & Land Square Feet

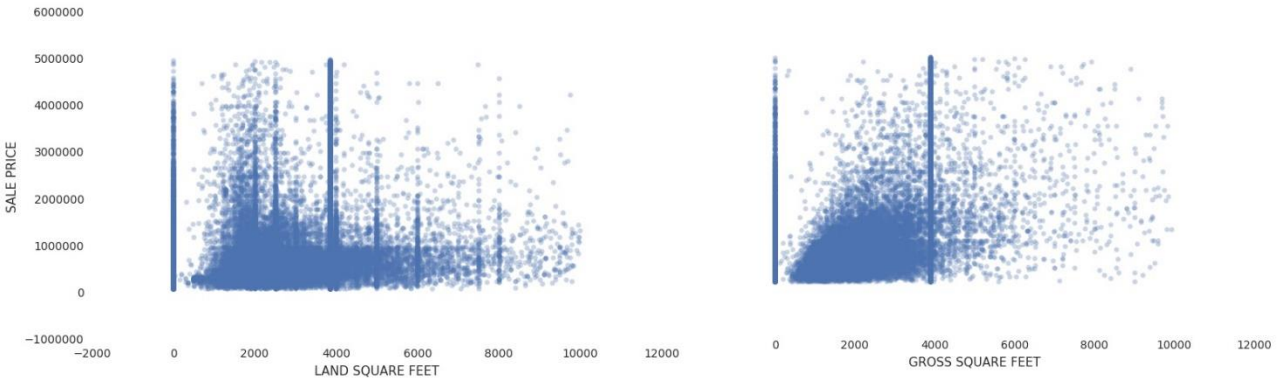


Figure 6 Gross & Land Square Feet Regression

Total Units, Commercial Units, Residential Units Analysis:

Reference to box plots in appendices [Residential Units], [Commercial Units] and [Total Units], there are some outlier mostly with total units. The total units 0 in dataset are 15493 and one outlier with 2261 units with one sale record is removed from dataset.

Top Prices in Categorical Variables:

The top price in building class category are Rental apartments with average price $3.56m [Top Building Class Category]. The top price in tax class at present category is residential property tax class with average price $2.06m [Top Tax Class at Present Category]. The top neighborhood is Civic Center with average price is $3.1m [Top Neighborhood]. The top location is (Borough) Manhattan with average price is $1.7m followed by Brooklyn with average price $1m [Top Borough].

Feature Engineering

The whole process of feature engineering involved is to normalize numeric feature through standard scaler then index & encode to one hot vector to categorical features and at the end assemble to one feature vector of 38342 final dataset.

Normalization – Standard Scaler:

The Standard Scaler is to transforms a dataset of vectors (rows), normalizing each feature to have unit standard deviation and zero mean. The standard scaler helps to assumes data is normally distributed within each feature and scale relative to one another to all numeric features.

Index Categorial Features:

String Indexer encodes a column of string labels/categories to a column of indices. The ordering of the indices is done based on popularity and the range is [0, numOfLabels). The spark's assembly feature 'StringIndexer' has been used to apply index to following string categorical features [String Indexer Example].

BOROUGH, NEIGHBORHOOD, BUILDING CLASS CATEGORY, TAX CLASS AT PRESENT, BUILDING CLASS AT PRESENT, TAX CLASS AT TIME OF SALE and BUILDING CLASS AT TIME OF SALE

Encode to One Hot Vectors:

One hot encoder maps the label indices to a binary vector representation with at the most a single one-value. The spark's assembly feature 'OneHotEncoderEstimator' has been used to apply encoding to all

index columns, converted from string to numerical index at pervious process [One Hot Encoder Example].

The representation of one hot vector has 4 different components. The first component which is a 0 indicates that it is a sparse vector. The second component talks about the size of the vector. The third component talks about the indices where the vector is populated while the fourth component talks about what values these are. This truncates the vector and is efficient when you have large vector representations.

Assemble to a Feature Vector:

Vector assembler's job is to combine the raw features and features generated from various transforms into a single feature vector. It accepts boolean, numerical and vector type inputs. The spark's feature 'VectorAssembler' has been used to assemble features into one vector. Following features are NOT included with target variable "Sale Price":

['SALE DATE', 'ADDRESS', 'APARTMENT NUMBER', 'SALE PRICE']

Spark's MLLib standardizes APIs for machine learning algorithms to make it easier to combine multiple algorithms into a single pipeline, or workflow. The pipeline has been used to stream line the process.

Train/Test Split:

The random sample of 70/30 has been used to split train and test data of **26784** for train and **11558** for test. Later apply caching for faster performance.

Predictive Modeling

The regression models have been used to predict the price of target variable "Sale Prices". Why use regression analysis? The reason is Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (s) (predictor). This technique is used for forecasting, time series, price prediction modelling and finding the causal effect relationship between the variables.

Linear Regression:

The linear regression was used with parameters "ElasticNet mixing parameter: 0.5", "Regularization parameter: 0.3", and "Max number of iterations: 10" and the result evaluate from spark's MLLib "RegressionMetrics". The RMSE was **0.8607** and R-Squared **0.2653** [Linear Regression Result].

Random Forest Regressor:

The random forest regressor was used with parameter grid with params "Maximum depth of the tree: [5,10]" and "Number of trees to train: [20,60]" and model was built through spark's ml Pipeline stages with cross validation of multiple folds to get best model. The result evaluates from spark's MLLib "RegressionMetrics". The RMSE was **0.5959** and R-Squared **0.6478** [Random Forest Regressor Result].

Gradient-Boosted Tree Regression:

The gradient-boosted tree regression was used with params like "Max number of iterations: [5,10]" and model was built through spark's ml Pipeline stages. The result evaluates from spark's MLLib library "RegressionMetrics". The RMSE was **0.7056** and R-Squared **0.5062** [Gradient-Boosted Tree Regression Result].

Decision Tree Regression:

The decision tree regression was used with default params and the result evaluate from spark's MLLib library "RegressionMetrics". The RMSE was **0.7022** and R-Squared **0.5110** [Decision Tree Regression Result].

Conclusion

In conclusion, only 45% data has been used and lot of data has been removed due to "Sale Price" was null and other factors. The **Random Forest Regressor** works best for this dataset with ***RMSE** score of **0.5959** and ***R-Square** of **0.6478** [Random Forest Regressor Result].

The "Sale Price" null values data (21%) which was removed, can be use as test data to predict "Sale Price" from this model.

The top result at Kaggle using Sklearn result was RMSE 0.58.

References

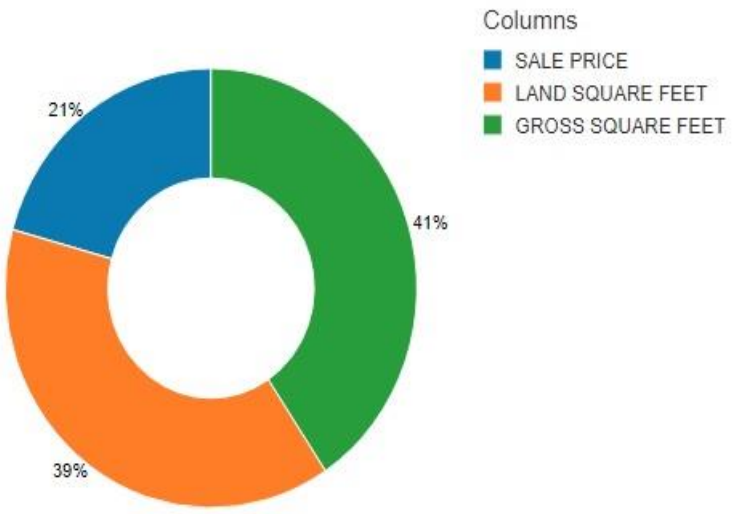
1) https://databricks-prod-cloudfront.cloud.databricks.com/public/4027ec902e239c93eaaa8714f173bcfc/4465375683188438/295089399769471/5192511032894936/latest.html
2) https://www.kaggle.com/new-york-city/nyc-property-sales
3) https://spark.apache.org/docs/2.2.0/ml-classification-regression.html
4) https://community.cloud.databricks.com

Appendices

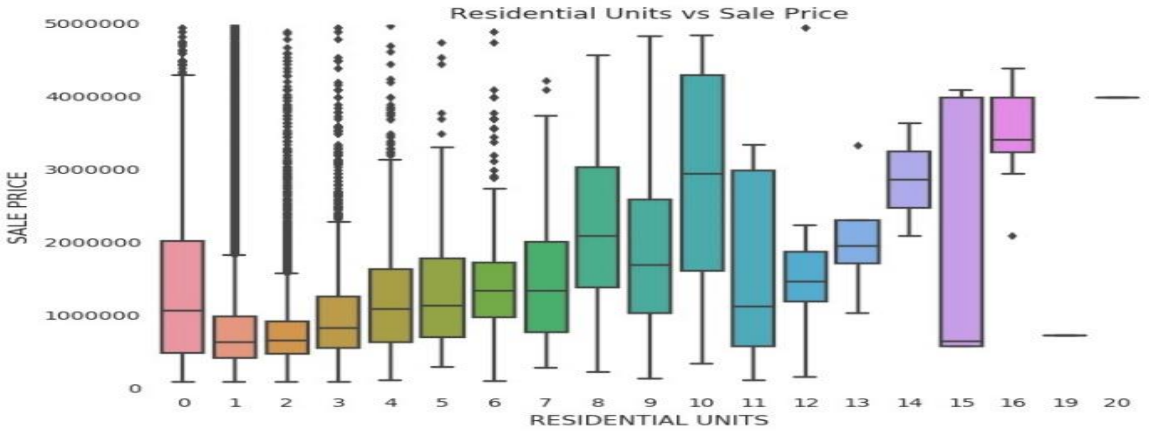
Schema:

#	Variable	Data Type	Categorical
1	BOROUGH	String	Yes
2	NEIGHBORHOOD	String	Yes
3	BUILDING CLASS CATEGORY	String	Yes
4	TAX CLASS AT PRESENT	String	Yes
5	BUILDING CLASS AT TIME OF SALE	String	Yes
6	BLOCK	Numeric	
7	LOT	Numeric	
8	ZIP CODE	Numeric	
9	RESIDENTIAL UNITS	Numeric	
10	COMMERCIAL UNITS	Numeric	
11	TOTAL UNITS	Numeric	
12	LAND SQUARE FEET	Numeric	
13	GROSS SQUARE FEET	Numeric	
14	YEAR BUILT	Numeric	
15	TAX CLASS AT TIME OF SALE	String	Yes
16	BUILDING CLASS AT TIME OF SALE	String	Yes
17	SALE PRICE	Long (Numeric)	Target Variable

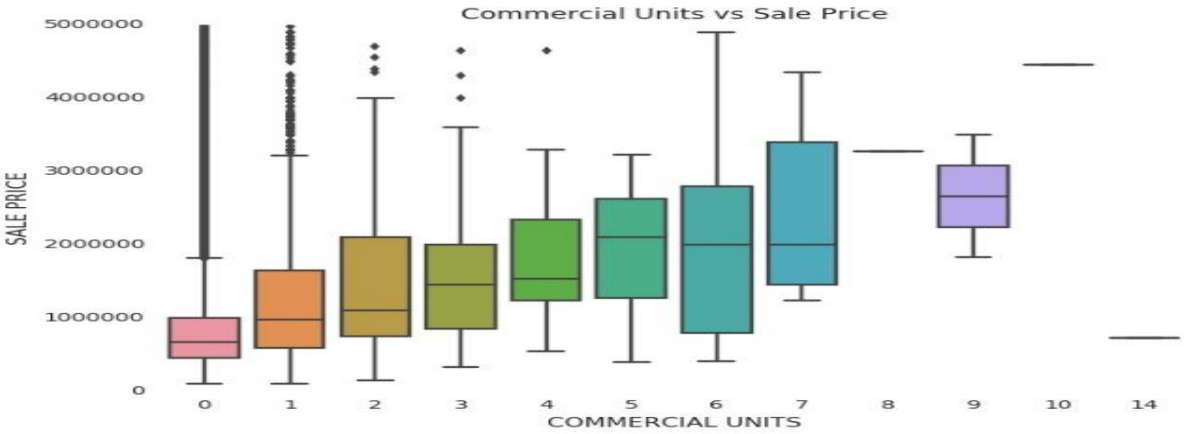
Missing Records:



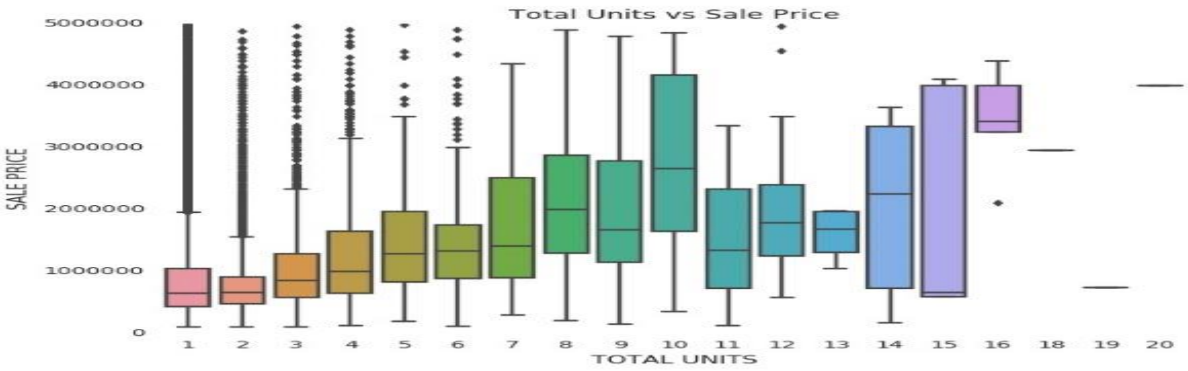
Residential Units:



Commercial Units:



Total Units:



Top Building Class Category:

BUILDING CLASS	AVERAGE PRICE
08 RENTALS - ELEVATOR APARTMENTS	$3,565,000
38 ASYLUMS AND HOMES	$2,983,029
23 LOFT BUILDINGS	$2,881,250
26 OTHER HOTELS	$2,135,000
16 CONDOS - 2-10 UNIT WITH COMMERCIAL UNIT	$1,862,794
46 CONDO STORE BUILDINGS	$1,792,828
32 HOSPITAL AND HEALTH FACILITIES	$1,743,333
30 WAREHOUSES	$1,720,859
14 RENTALS - 4-10 UNIT	$1,662,410
27 FACTORIES	$1,647,886
43 CONDO OFFICE BUILDINGS	$1,623,524

Top Tax Class at Present Category:

TAX CLASS AT PRESENT	AVERAGE PRICE
2B	$2,064,155
4	$1,398,685
2A	$1,355,044
2	$1,313,432
1C	$1,242,359
2C	$1,110,688

Top Neighborhood:

NEIGHBORHOOD	AVERAGE PRICE
CIVIC CENTER	$3,160,668
JAVITS CENTER	$2,878,750
BROOKLYN HEIGHTS	$2,743,342
LITTLE ITALY	$2,658,866
SOHO	$2,639,805
FLATIRON	$2,599,035
TRIBECA	$2,298,968
GREENWICH VILLAGE-CENTRAL	$2,237,906
UPPER WEST SIDE (96-116)	$2,112,789
CARROLL GARDENS	$2,095,826
GREENWICH VILLAGE-WEST	$2,076,147
GRAMERCY	$2,038,991
MIDTOWN EAST	$2,037,596

Top Borough:

BOROUGH	AVERAGE PRICE
1 (Manhattan)	$1,764,065
3 (Brooklyn)	$1,070,449
4 (Queens)	$712,735
2 (Bronx)	$492,759
5 (Staten Island)	$489,850

String Indexer Example:

Column Name	Original Value	New Column	New Value
BOROUGH	ALPHABET CITY	BOROUGH_index	2
NEIGHBORHOOD	BAYCHESTER	NEIGHBORHOOD_index	44

One Hot Encoder Example:

Column Name	Original Value	New Column	New Value
BOROUGH_index	2	BOROUGH_vec	[0,246,[44],[1]]
NEIGHBORHOOD_index	44	NEIGHBORHOOD_vec	[0,4,[2],[1]]

Linear Regression Result:

Regression metrics	Value
MSE (Mean Squared Error)	0.7408
MAE (Mean Absolute Error)	0.6661
RMSE (Root Mean Squared Error)	0.8607
R-Squared	0.2653
Explained Variance	0.0893

Random Forest Regressor Result:

Regression metrics	Value
MSE (Mean Squared Error)	0.3551
MAE (Mean Absolute Error)	0.4279
RMSE (Root Mean Squared Error)	0.5959
R-Squared	0.6478
Explained Variance	0.5520

Gradient-Boosted Tree Regression Result:

Regression metrics	Value
MSE (Mean Squared Error)	0.4979
MAE (Mean Absolute Error)	0.5293
RMSE (Root Mean Squared Error)	0.7065
R-Squared	0.5062
Explained Variance	0.3585

Decision Tree Regression Result:

Regression metrics	Value
MSE (Mean Squared Error)	0.4930
MAE (Mean Absolute Error)	0.5229
RMSE (Root Mean Squared Error)	0.7022
R-Squared	0.5110
Explained Variance	0.5147