Machine Learning Engineer Nanodegree

Capstone Project

Alison Gaby
August 8th, 2018

Housing in Northeastern Flushing, Queens, NY

I. Definition

Project Overview

For my project, I will predict housing prices in Queens, NY. using the New York City Property Sales dataset generously hosted and curated by Enigma Public. This project was inspired by the Boston Housing project that was a part of Udacity's Machine Learning Nanodegree, in which students estimated the prices of housing in a suburb of Boston, MA in 1978 using supervised learning techniques. The project raised several questions regarding the applicability of using supervised learning techniques to other geographical areas, or to areas where there is more variability in housing affordability or population density.

For reference, my submission for the Boston Housing Project can be found here: https://nbviewer.jupyter.org/github/aog5/machine-learning/blob/master/projects/boston_housing/boston_housing-submission.ipynb#

Datasets:

The New York City Property Sales Dataset - Based on Rolling Sales Data provided by the City of New York's Department of Finance's collection of property listings that sold in the last twelve-month period. The original dataset contains the following - Total Records: 60,295, total fields: 63.

Of the 63 fields in the dataset, the below columns have been included in this project:

- zipcode
- land_square_feet
- gross_square_feet
- sale_price
- community_district
- school_district
- floor_area
- total_buildings
- floor_area_residential
- maximum_allowable_residential_far
- x_coordinate
- y_coordinate

Annualized Sales - This is a collection of yearly sales information of properties sold in New York City between the years 2005 to 2016. (Only data from the years 2013-2016 was used).

This dataset includes the following attributes, grouped by neighborhood, and type of home:

- NEIGHBORHOOD
- TYPE OF HOME
- NUMBER OF SALES
- LOWEST SALE PRICE
- AVERAGE SALE PRICE
- MEDIAN SALE PRICE
- HIGHEST SALE PRICE

SOI Tax Stats - Individual Income Tax Statistics - 2015 ZIP Code Data (SOI) - Earned Income Credit by Zip Code - This was calculated using data extracted from this dataset, and mapped to the existing housing data.

Neighborhood Air Quality Concentrations: Sulfur Dioxide (SO2) - this data was extracted from the following linked dataset: (http://a816-dohbesp.nyc.gov/IndicatorPublic/VisualizationData.aspx?id=2026,719b87,122,Summarize) and mapped from each UHF 42 section to each zip code, then mapped to the original housing data. (This particular dataset was included to test the suggestion that Sulfur Dioxide levels signalled disparities between neighborhoods. Were they economic disparities? https://www1.nyc.gov/assets/doh/downloads/pdf/epi/databrief88.pdf)

Problem Statement

Estimating housing prices can be a difficult problem. There are multiple factors one must take into account beyond taking into account many different variables. In our project using the Boston Housing set, we explored using different regression methods to get accurate housing price predictions. With that particular data set, we were able to get fairly accurate results, but does this model “travel”? Can we use this on data from a different local? Let us explore.

Metrics

As an evaluation metric, I’ve chosen the to use the R2 method or the Coefficient of Determination. As an evaluation method for regression, we will use this method for testing how accurately our model is by how close it reaches 1 (out of values 0 to 1, with 1 being the most accurate possible). As our model is a regression, I’ve chosen this method for accuracy as our method is based on how accurately we can predict roughly normally distributedsale_price data (normally distributed data is assumed when using the R2 method for scoring).

II. Analysis

Data Exploration

To begin our analysis, let us take a quick look at our dataset. The dataset I'm using for this project originally has 60,295 rows and 63 columns. As we're only interested in data from one borough and one type of home (Single Family Homes), I've used the following SQL query to select only the most relevant data, leaving us with 5219 rows representing sales prices, and 13 columns from which to select the most meaningful features.

```
SELECT borough_code,
    neighborhood,
    zipcode,
    land_square_feet,
    gross_square_feet,
    sale_price,
    community_district,
    school_district,
    floor_area_total_building,
    floor_area_residential,
    maximum_allowable_residential_far,
    x_coordinate,
    y_coordinate
FROM enigma_nychousing.nyc_housing
WHERE sale_price > 200000
 AND tax_class_at_present_code LIKE '1%'
 AND building_class_category_code_definition LIKE 'ONE%'
 AND borough_code = 4
```

At a cursory glance, we can see there are a few columns with null values or `NaN`. Are these features informative?

The column labelled `maximum_allowable_residential_far` doesn't give us very much information, as the homes selected are already designated as being from Tax Class 1, which means the property meets the following criteria:

> *"Includes most residential property of up to three units (such as one-, two-, and three-family homes and small stores or offices with one or two attached apartments), vacant land that is zoned for residential use, and most condominiums that are not more than three stories (1-3 UNIT RESIDENCES)"*

I believe we can safely drop this column from our dataset, as it is redundant. Additionally, the columns labelled `floor_area_total_building` and `floor_area_residential` contain information already taken into account by the column `gross_square_feet`, and, as we've already selected for properties in Queens, the column `borough_code` is no longer necessary. Let's go ahead and remove these columns from our dataset.

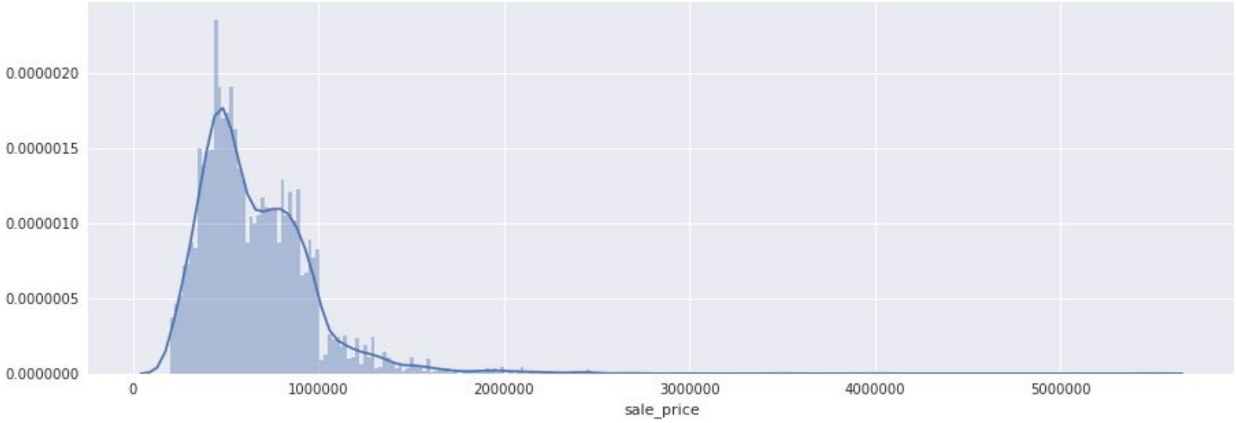
Exploratory Visualization

To get a sense of our dataset, let's look at our features relationships to each other. The below "Pair Grid" (figure shows how each of the columns relates to the others. We see a few potentially meaningful pairwise relationships.

Let's look further.

First, let's take a look at our target values for sale_price. How are they distributed? As it appears that our target is roughly normally distributed, but skewed to the left, with a long right tail, I believe we are limited in what scaling we can do, without excluding important information.

Fig. 1 - Distribution of Sales Prices

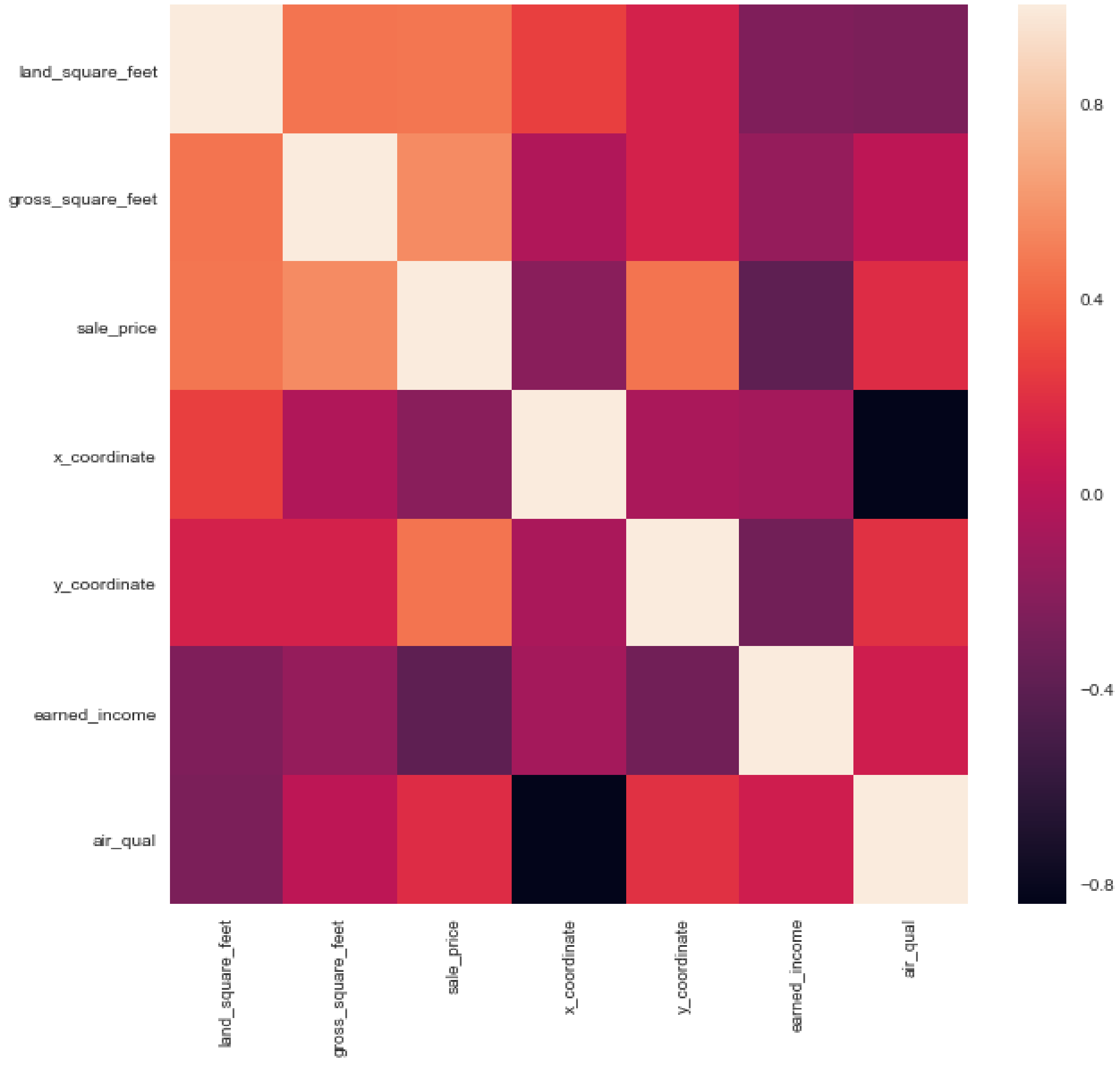


To get an idea of pairwise relationships between features, let's take a look at a "pair plot", showing comparisons between each feature (see next page for plot).

At a glance, we can see there are a few interesting relationships, particularly regarding the properties for gross and land square feet, which seems inline with the results from the Boston Housing project (# of rooms - `RM` - was considered one of the highly predictive features).

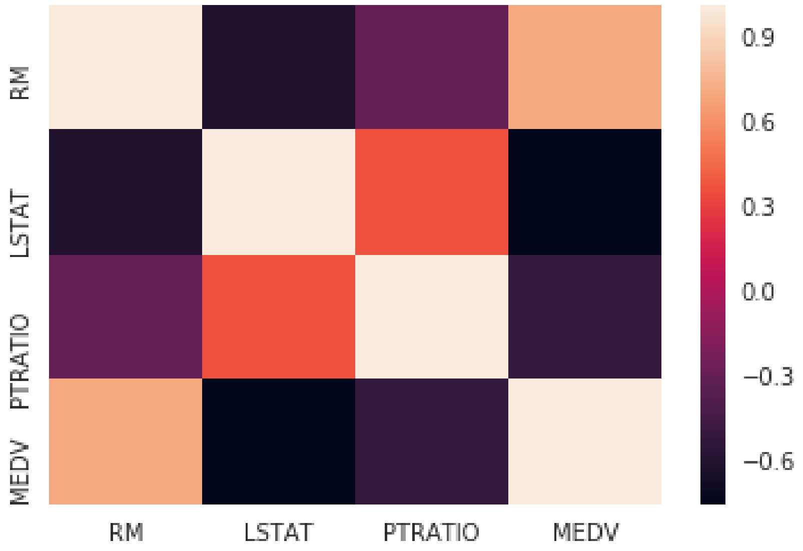
To get an even better idea of how these features relate, let's look at a heatmap showing how our non-categorical features correlate.

Fig. 2 - Headmap of all non-categorical features



From our heatmap, we get a general idea of how features correlate. More specifically, if we look at the column for `sale_price`, we can see that the highest correlations are somewhat predictably related to the size of the property (`gross_square_feet` and `land_square_feet`, respective of their correlations), their geographic location (`x_coordinate` is negatively correlated almost as much as `y_coordinate` is positively correlated, although I suspect this set of correlations is unique to this particular dataset), and the percentage of residents eligible to receive Earned Income Credit on there yearly tax return (`earned_income` - the closest data I could find to proxy the percentage of low income residents in the Boston Housing dataset - `LSTAT``).

Fig. 3 - Headmap of all features in Boston Housing Project dataset



In comparing Fig. 2 and Fig. 3 correlations, we can clearly see that the relationships between `sale_price` and `gross_square_feet` closely mirror those of `MEDV` (median value) and `LSTAT`. Let's look at those relationships more closely in Figures 5 & 6.

<u>ATTRIBUTE</u>	<u>Description</u> [1]
MEDV	Median value of owner-occupied homes in $1000's
PTRATIO	pupil-teacher ratio by town
LSTAT	% lower status of the population
RM	average number of rooms per dwelling

(Fig. 4 is provided for overall reference).

[1] http://www.lsi.upc.edu/~belanche/Docencia/mineria/Practiques/Boston.dat

Fig. 4 - Pair plot of all features in Queens, NY dataset
(For easier viewing, the pair plot graph referenced can be viewed here:
https://drive.google.com/open?id=1Ar50g6PfFdpDiGDLin44YiBc7ZArm-_E)

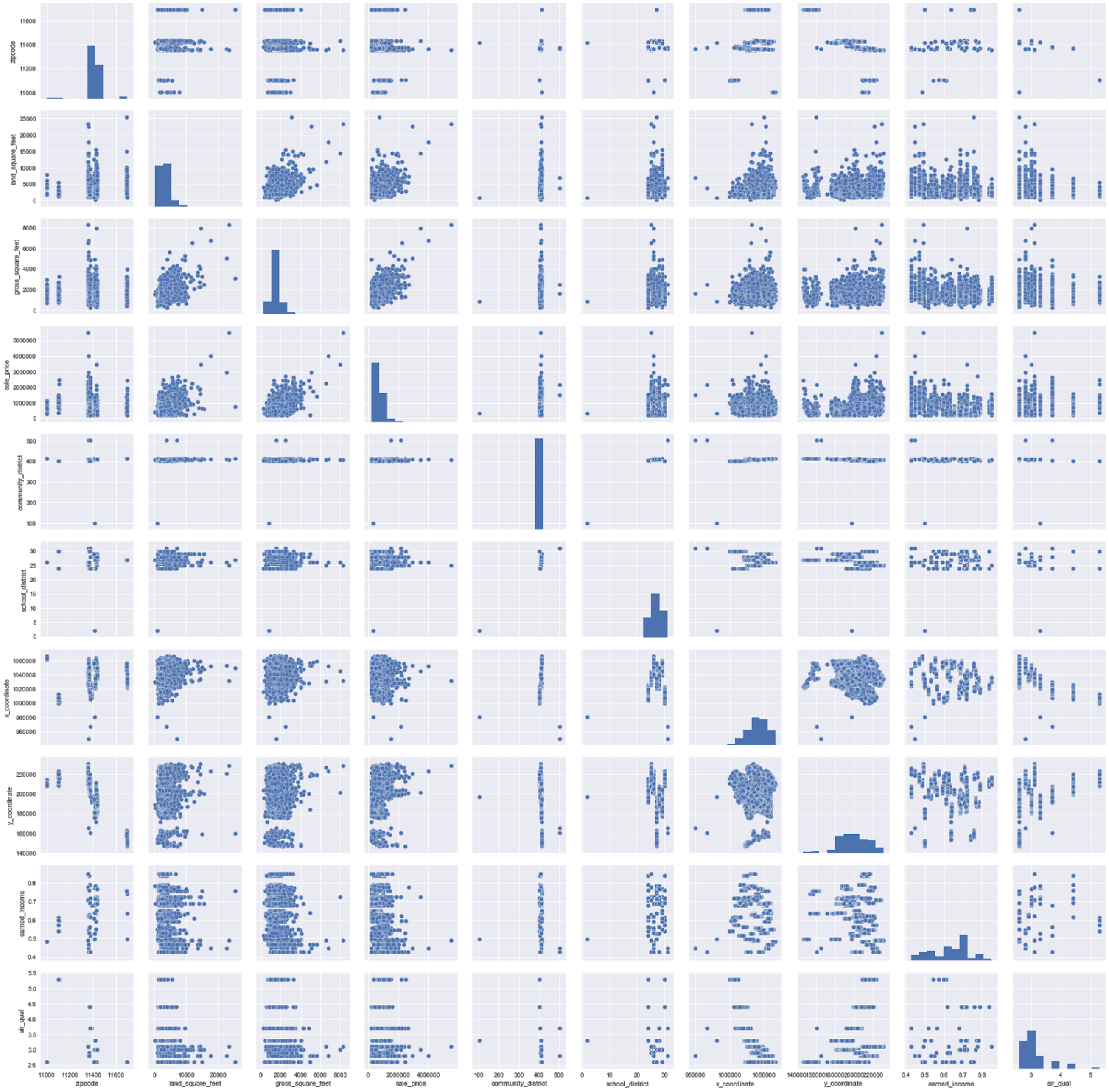


Fig. 5 - Sale Price vs. Earned Income reflecting a negative correlation

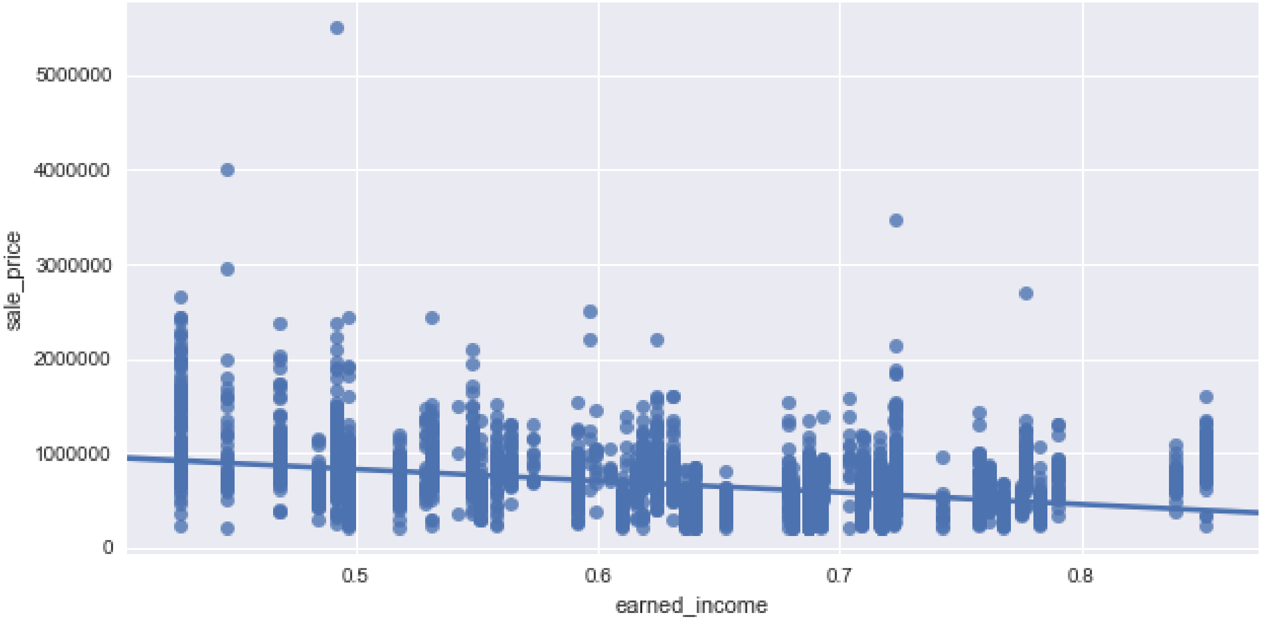
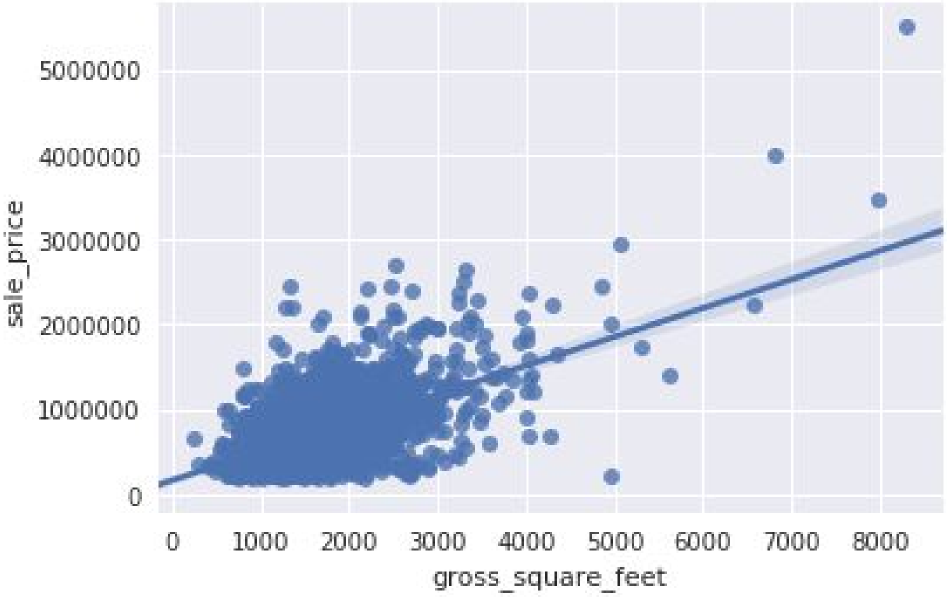


Fig. 5 - Sale Price vs. Gross Square Feet reflecting a positive correlation



Algorithms and Techniques

In order to approach this problem, we've examined our New York City Property Sales dataset, and determined which features are most predictive, which happen to align with features that were most predictive in the Boston Housing Set.

The particular supervised learning algorithm used in the Boston Housing project is the Decision Tree Regressor. While the choice for using this particular algorithm was not discussed in the project, decision trees can be particularly useful when estimating housing prices as they can accept categorical and numerical data types (such as zip codes and percentage of low income residents), does not make any statistical assumptions about the data, and may yield results that are more easily explained as they somewhat mirror the human decision-making process.[2]

Benchmark

As a benchmark, we will fit our data to a simple decision tree regression to test its accuracy. How does it compare to the accuracy of the Boston Housing project's model?

As we can see in the figures below, the Boston Housing Set visualized in Fig. 7 appears to yield the most accurate results with a decision tree of max_depth 3, minimizing both the variance and the bias.

It appears the Decision Tree for our Queens dataset is most accurate at a depth of 1. In this case, this is telling of a model with is not picking up on nuances in our dataset and overfitting to the data. In order to find the optimal depth of a decision tree, our Boston Housing Project uses a Grid Search method with the ShuffleSplit method for cross-validation. GridSearch is a method that allows us to test our model using a variety of parameters, searching for the combination of parameters yielding the most accurate results. Cross-validation is a technique which allows for us to consider variations in our data by splitting our data into equal portions (10 "splits" in this case), and select random permutations for training and testing sets.

Using the same GridSearch method as our Boston Housing Project, the decision tree of depth 7 represents the most accurate model. I believe our first visualization is a little misleading, as there don't seem to be any noticeable trends from max_depth of 1 to max_depth of 10. When following along with the Boston Housing Projects methods, our visualizations would lead us to believe that a tree with a depth of 1 is the most accurate, while our GridSearch and cross-validation leads us to believe that the optimal depth for a decision tree is 7.

Let us compare our results in this project with that of that Boston dataset. Are our results the same? Have they improved? If not, I will propose different methods to test in future experiments.

[2] https://en.wikipedia.org/wiki/Decision_tree_learning; https://dzone.com/articles/decision-trees-vs-clustering-algorithms-vs-linear

Fig. 7 - Boston Housing Set baseline results

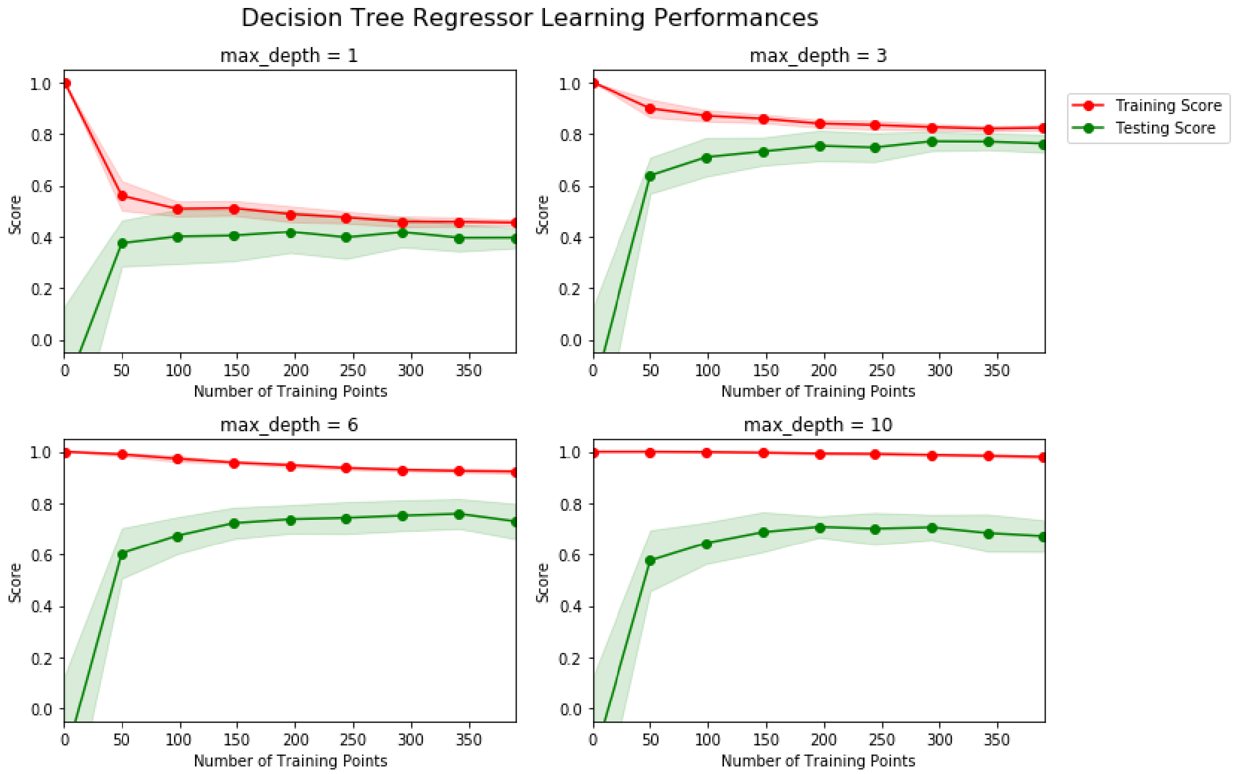
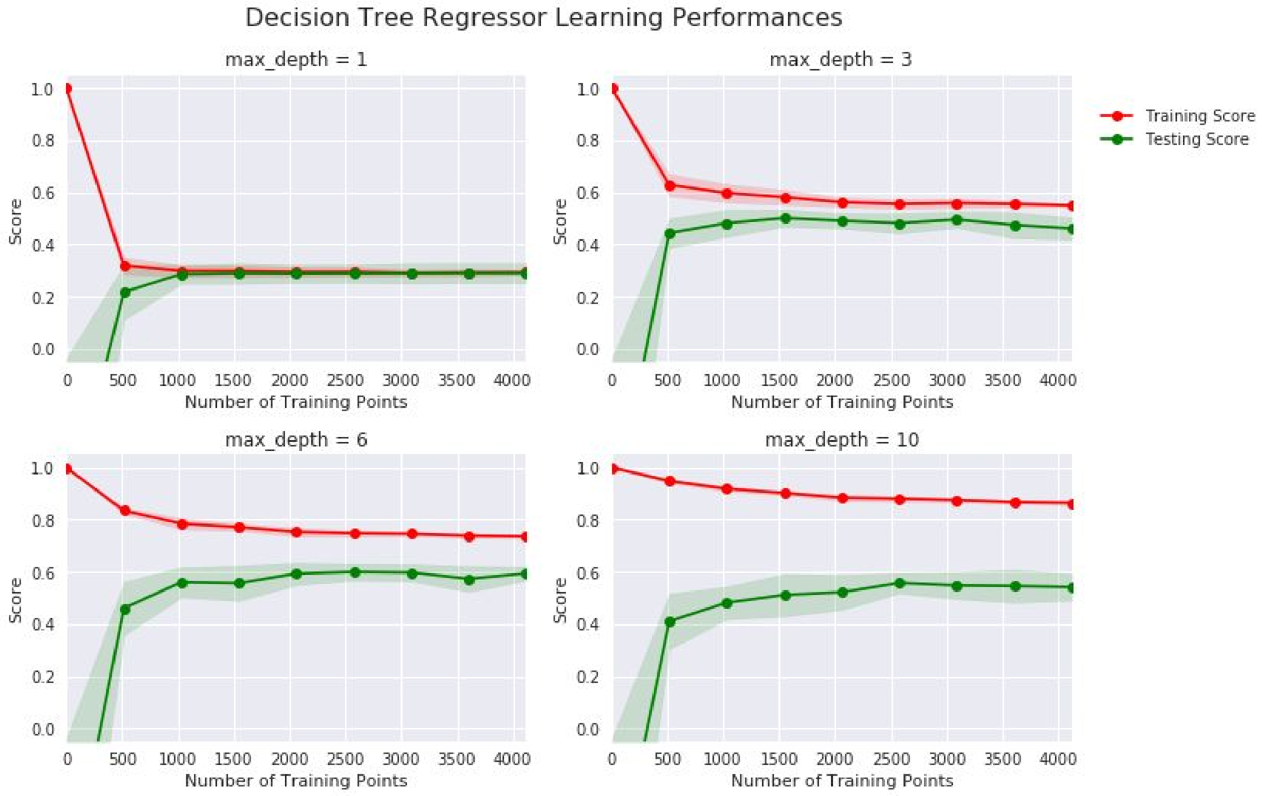


Fig. 8. - Queens, NY Property Sales



III. Methodology

Data Preprocessing

As all of our data was relatively normally distributed, and will be used with a decision tree, no scaling or preprocessing was necessary. Also, as the project off of which this analysis was based did not preprocess any data as part of it's analysis, it would be best to leave the data as it is.

Implementation

In order to implement this project, I wanted to ensure that the algorithms implemented mirrored those in the Boston Housing Project to ensure that a one-to-one comparison could be made. As the methods for preprocessing and selecting data for the Boston Housing Dataset was not documented as a part of the project, I carefully reviewed the complete New York City Property Sales data found here: https://public.enigma.com/datasets/new-york-city-property-sales/fd6efa37-2dcd-4294-8795-a0e6044f15b4

Once selecting the attributes deemed most relevant and unique, I initially used an exported CSV of only the relevant data. This method proved helpful, but as the project and my analysis continued, I had questions that could only be answered by revisiting the original dataset. As a solution, I imported the entire dataset into Google Cloud Platform's BigQuery (Google Cloud Platform's data warehouse), and then imported those tables into Mode Analytics (https://about.modeanalytics.com/). I could then run and save SQL queries, such as the one referenced in the Data Analysis section on page 4., which I could then refer to in a "python notebook", and if I had questions, such as "how normally distributed is this dataset?" or "is this data consistent with historical data and truly representative?", I could run another SQL query and test my hypothesis!

Next, I mapped my data for percentage of earned income and air quality to their zip codes, adding that to our dataset, and reviewed for consistency.

Once I fully trusted the data I was working with, I further refined the data to remove any null values or misleading data (for instance, data for two zip codes in an adjacent county were included, as it appears they have identical names to towns in Queens: https://en.wikipedia.org/wiki/New_Hyde_Park,_New_York#Geography)

Now that we have all the features we need for implementing our analysis, I'll go ahead and select our most important features as determined by our tested correlations.

Now that our data is fully refined, I implemented scikit-learn’s train test split model to divide our data into testing and training data. I then implemented a decision tree regressive in a method identical to that in the Boston Housing project in order to maintain consistent results.

Comparing the Decision Tree using the Boston Housing Projects ‘’ function, we have a way to visually compare our two model's performances (see figures 7 and 8).

While the visualization seemed helpful for our Boston Housing dataset, the visualization for the Queens, NY dataset seems much less helpful. From what we can see, there doesn’t seem to be any particular trends or patterns indication the most appropriate tree depth.

Next, I implemented the GridSearch method in order to truly find the best model given the maximum depth range 1-10. The resulting “best estimator” in this case was 6.

In addition to the benchmark provided by the Boston Housing project, I wanted to get a general idea of the range of predictive ability of other algorithms not included in the original project. I implemented a linear regression to test the lowest possible rate. To test a possible model improvement, I tested the AdaBoost regression method. While I did not explore the results of these models as far as I could have, it was helpful to see how my decision tree model faired.

Refinement

During the model's implementation, there were points where the model could be adjusted, for instance, adjusting the parameters of the decision tree, or perhaps creating more splits for the implementation of ShuffleSplit for cross-validation, but I believe any adjustments of the model would render the comparison between the two projects invalid.

IV. Results

Model Evaluation and Validation

The decision tree regressor of depth of 6, on this dataset, seems to be the most accurate according to our implementation of GridSearch. Compared the implementation on the Boston Housing Project dataset, this method (selecting the best parameters for a decision tree using GridSearch) seems to provide valuable results. Were this algorithm to be used in a context where high-precision were required, as is customers were using the predictions provided by this model for housing purchases, I would definitely suggest re-evaluating the model choice. Perhaps using a more granular approach, with several different model's for predictions of schooling quality, transportation options, weather, neighborhood history, or demographic shift that are used for a more inclusive model might be more appropriate.

I believe the results of this model’s predictions are appropriate and feel this model generalizes well to unseen data, provided that the data is of Housing Prices in Queens, New York. Results seem to reflect what I've seen in my interactions with the local housing market.

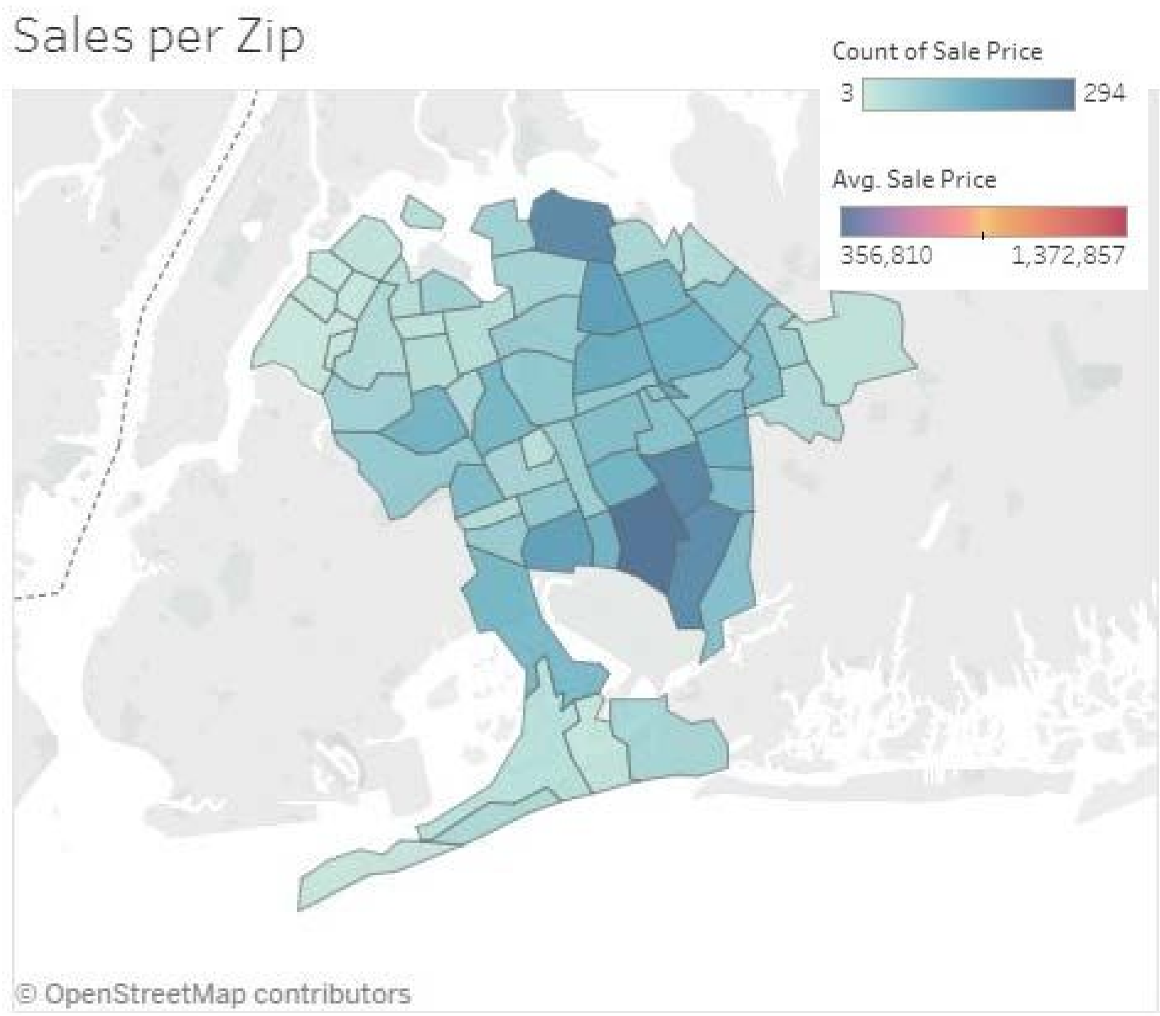
Justification

The problem solved in this project was to see if the model and methods used in the Boston housing Project would provide valuable predictions when real data from a different location was taken into account. While I believe the methods used in the Boston Housing Project could potentially provide accurate results, I believe the accuracy of the results is dependent on accurately selecting features for analysis. Since the dataset was not continuously updated and only reflects one location, only approximations to the original features can be made. I can clearly state that the model does "travel", with the caveat that datasets do not.

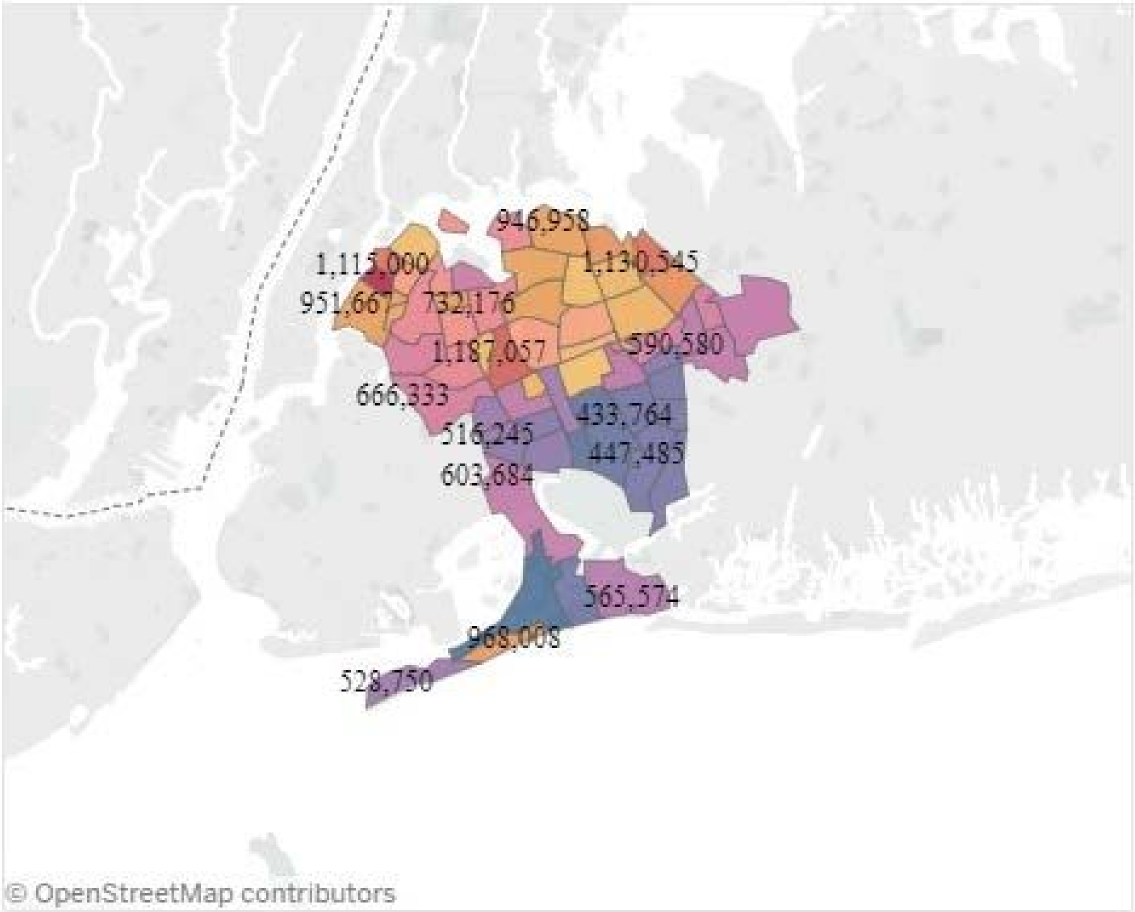
V. Conclusion

Free-Form Visualization

(The below visualizations can be found here: https://public.tableau.com/views/QueensHousingSales/Dashboard1?:embed=y&:display_count=yes)



Average Sale per Zip Code



Another potential feature of this dataset is the neighborhood or zip code “churn”. How long do people stay in their homes? Do sales follow a certain pattern?

If we were to predict upcoming housing sales in addition to their costs, I think the availability of houses in a certain neighborhood would certainly aid in our predictions. In the maps above, we can see the average sale price per zip code in addition to the number of sales per zip code. Given the small number of sales per zip code, we would need to introduce additional historical data and plot the ratio between sales and sale price within a given year to see if there are any notable trends. If so, I think it would make an ideal feature for a more sophisticated model.

Reflection

Looking back and considering the process taken in this project, I believe more time should have been taken learning the “ins-and-outs” of preprocessing data and feature engineering. While I don’t believe the time taken to select the features was well-spent, I believe a shorter, and more iterative approach, testing the predictability of different features, would yield the best results faster.

Streamlining my workflow for an iterative approach to building machine learning algorithms has been the most difficult part of this project. While difficult and time consuming, I feel it will lead to much quicker and more optimal results for future projects.

Improvement

The accuracy of this project's model could be improved by more clearly defining the relationship between `land_square_feet` and `gross_square_feet`. While there is a clear correlation between `sale_price` and each of the aforementioned features, they are clearly not independent. An exploration of the ratio between them might provide more insights.

In addition, the use of geographical coordinates limits the model's predictive accuracy to Queens, NY. Perhaps, in future iterations of housing models, geographical data could be used, but in a different way (not a feature, but as a way to define boundaries for certain models to be considered).