Machine Learning Engineer Nanodegree

Capstone Project

Alison Gaby
May 25th, 2018

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.

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.

```sql
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
```

From 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.

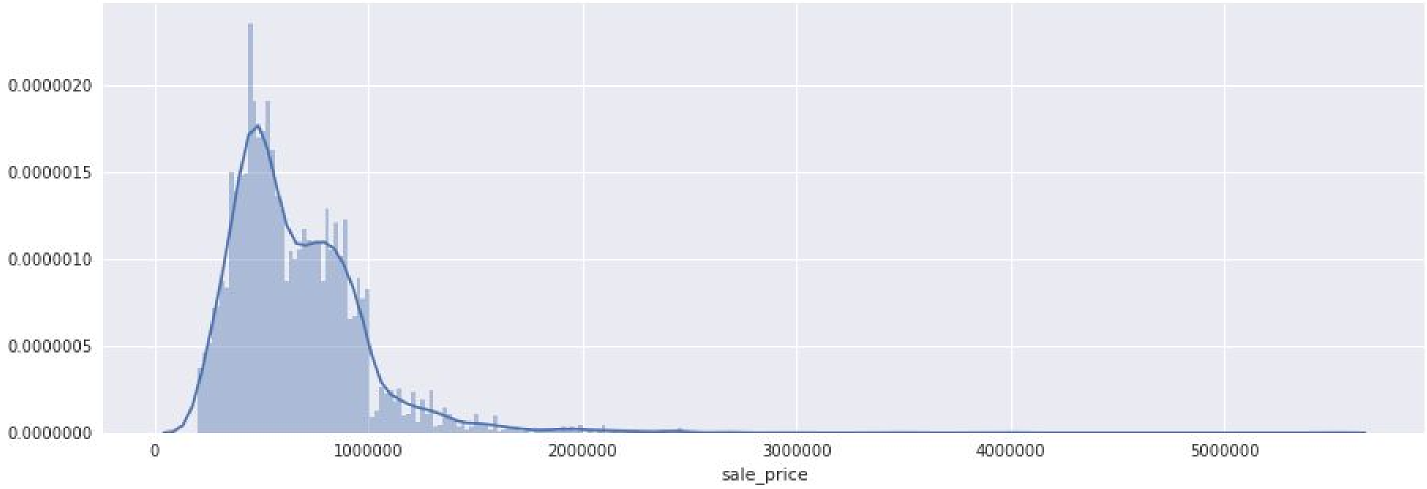
Exploratory Visualization

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

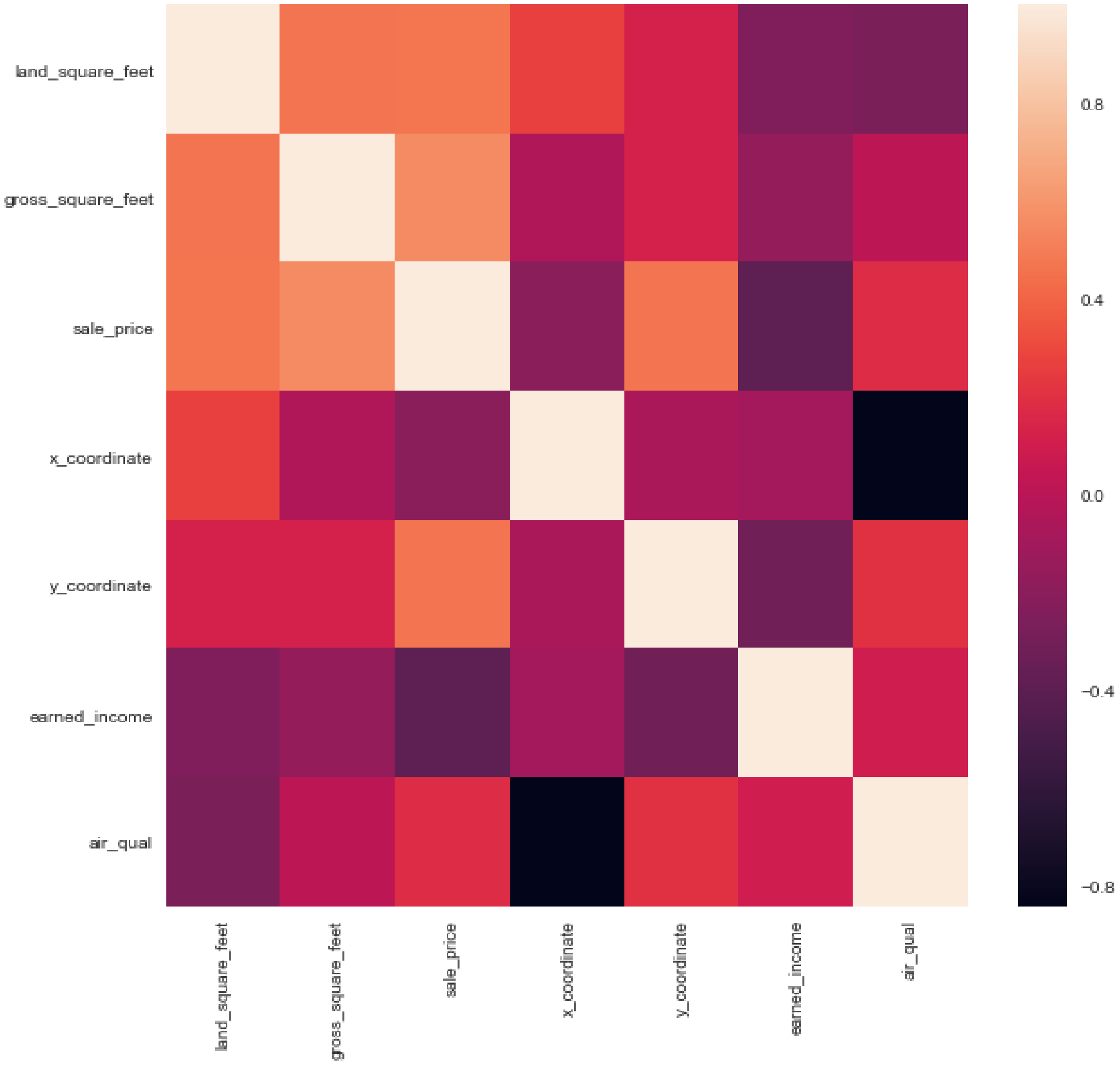


To get an idea of pairwise relationships between features, let’s take a look at a “pair plot”, showing comparisons between each feature (see next page for plot).

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).

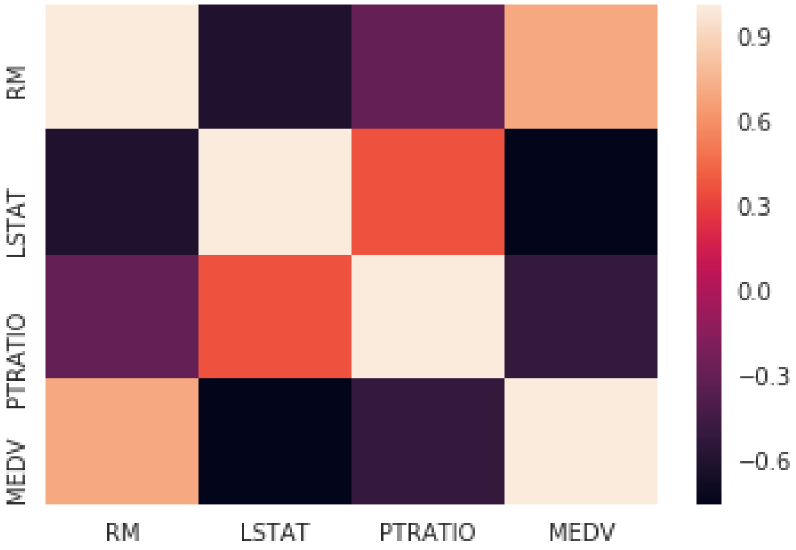
To get an even better idea of how these features relate, let’s look at a heatmap showing how our non-categorical features correlate.

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. (Fig. 4 is provided for overall reference).

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* *)*

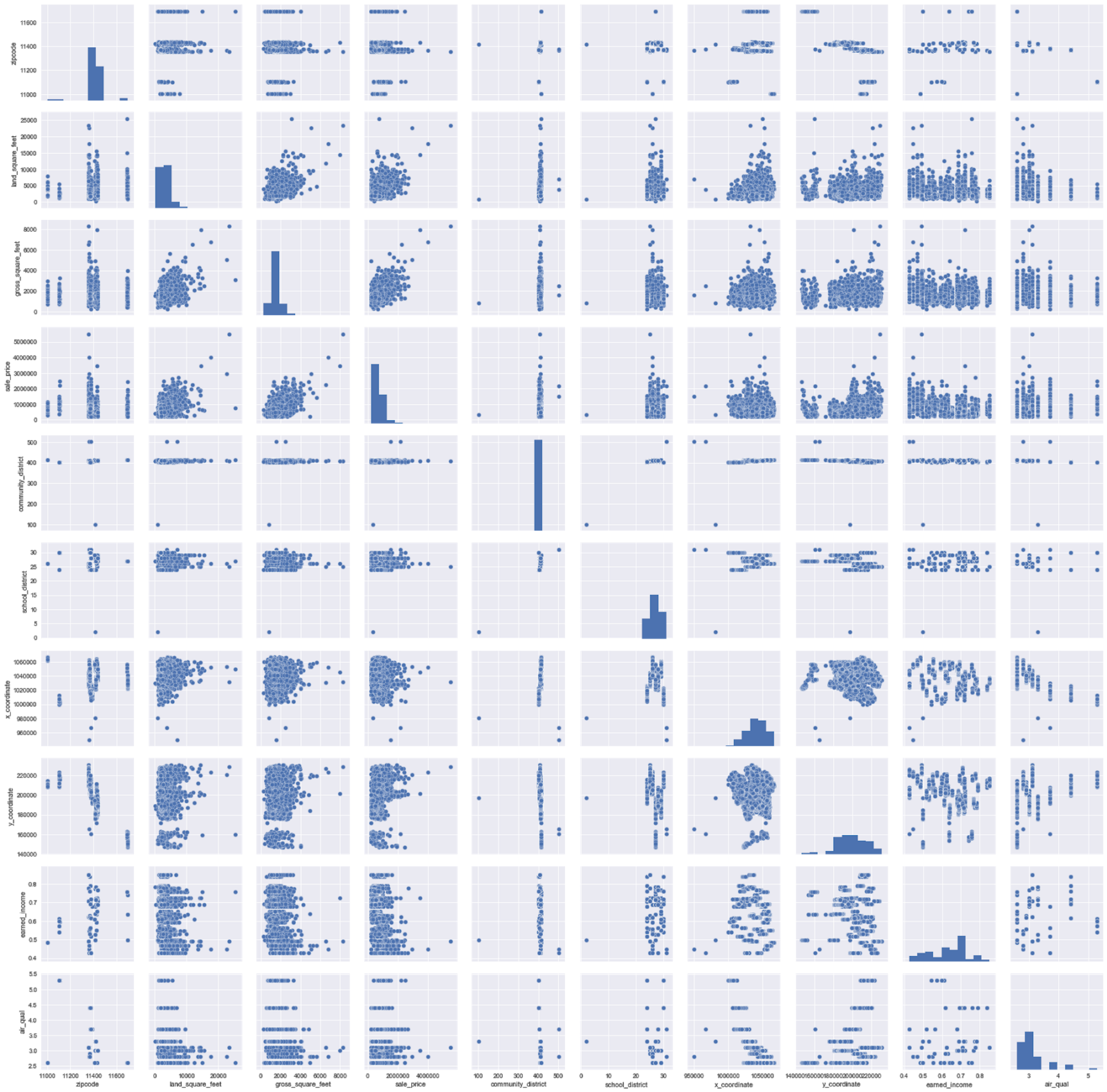


Fig. 5 - Sale Price vs. Earned Income reflecting a negative correlation

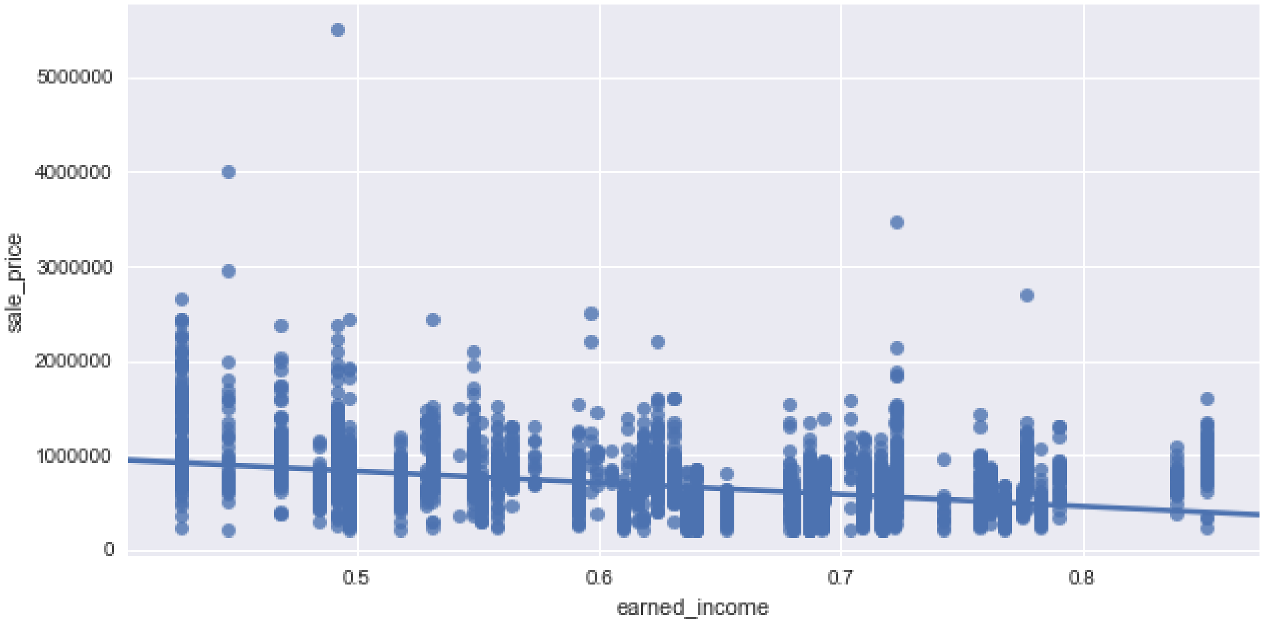
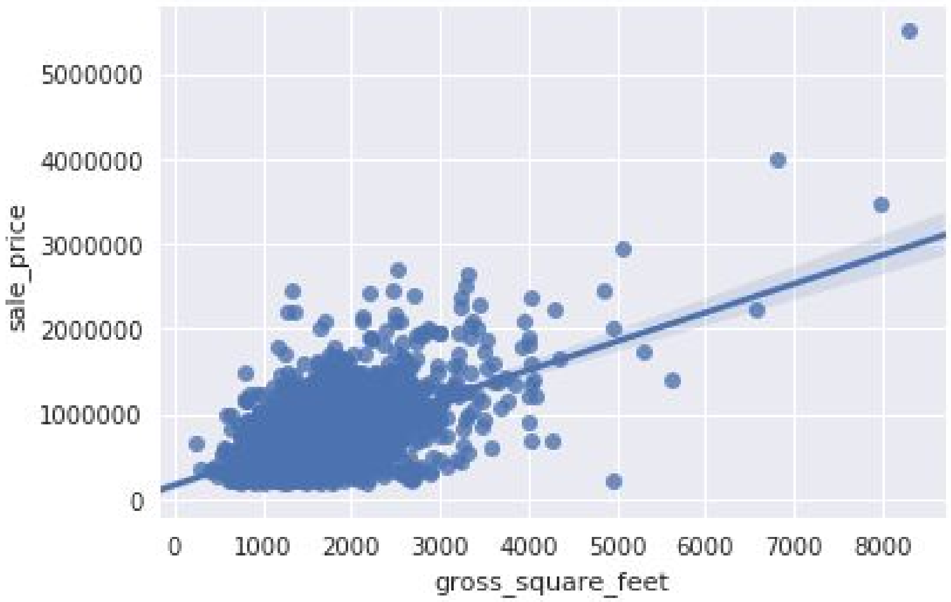


Fig. 5 - Sale Price vs. Gross Square Feet reflecting a positive correlation



Algorithms and Techniques

In order to approach this problem, we've examined our Queens 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.

As the distribution of our target values is roughly normally distributed and the original project with similar data is also not scaled, we will leave our dataset as-is.

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?

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

Are the algorithms you will use, including any default variables/parameters in the project clearly defined?
Are the techniques to be used thoroughly discussed and justified?
Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?

Benchmark

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. 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 methods 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 to max_depth. 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's

In order to improve accuracy, let us fit our data to an ensemble method called AdaBoost. How does this accuracy improve? We will then adjust to account for overfitting.

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.

Fig. 7 - Boston Housing Set baseline results

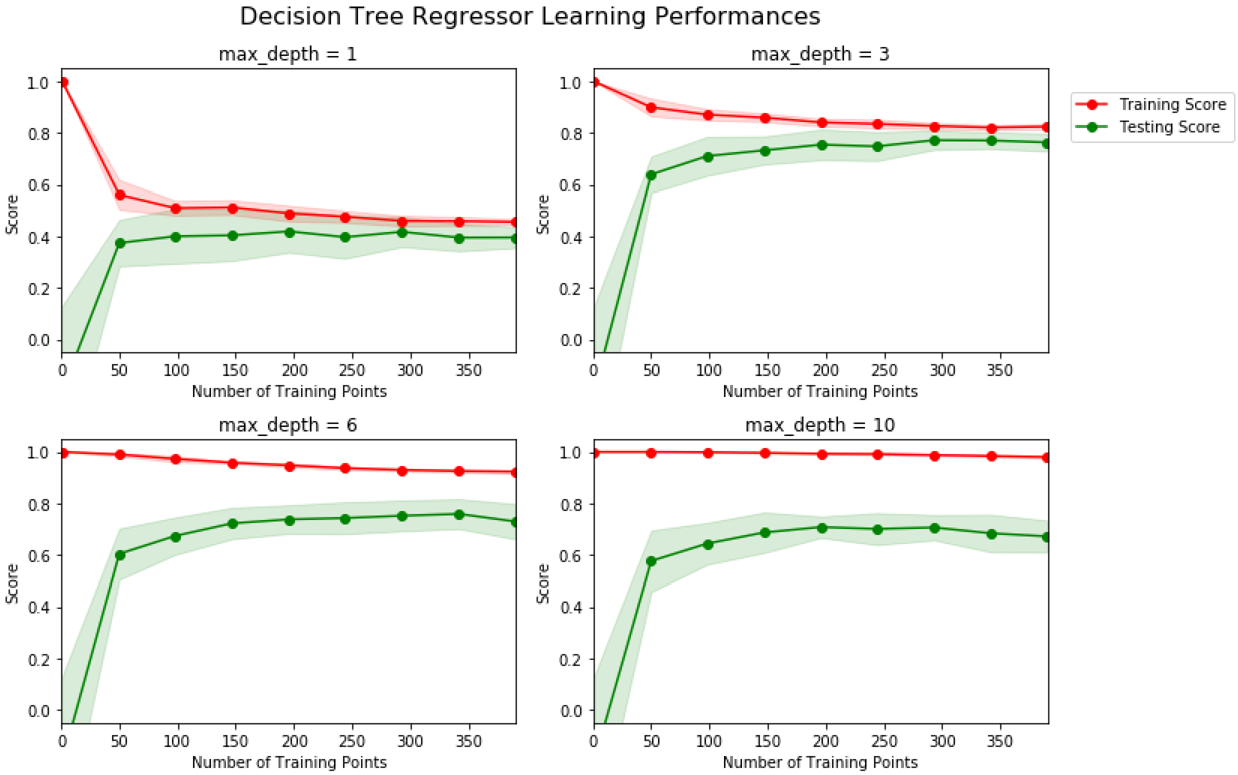
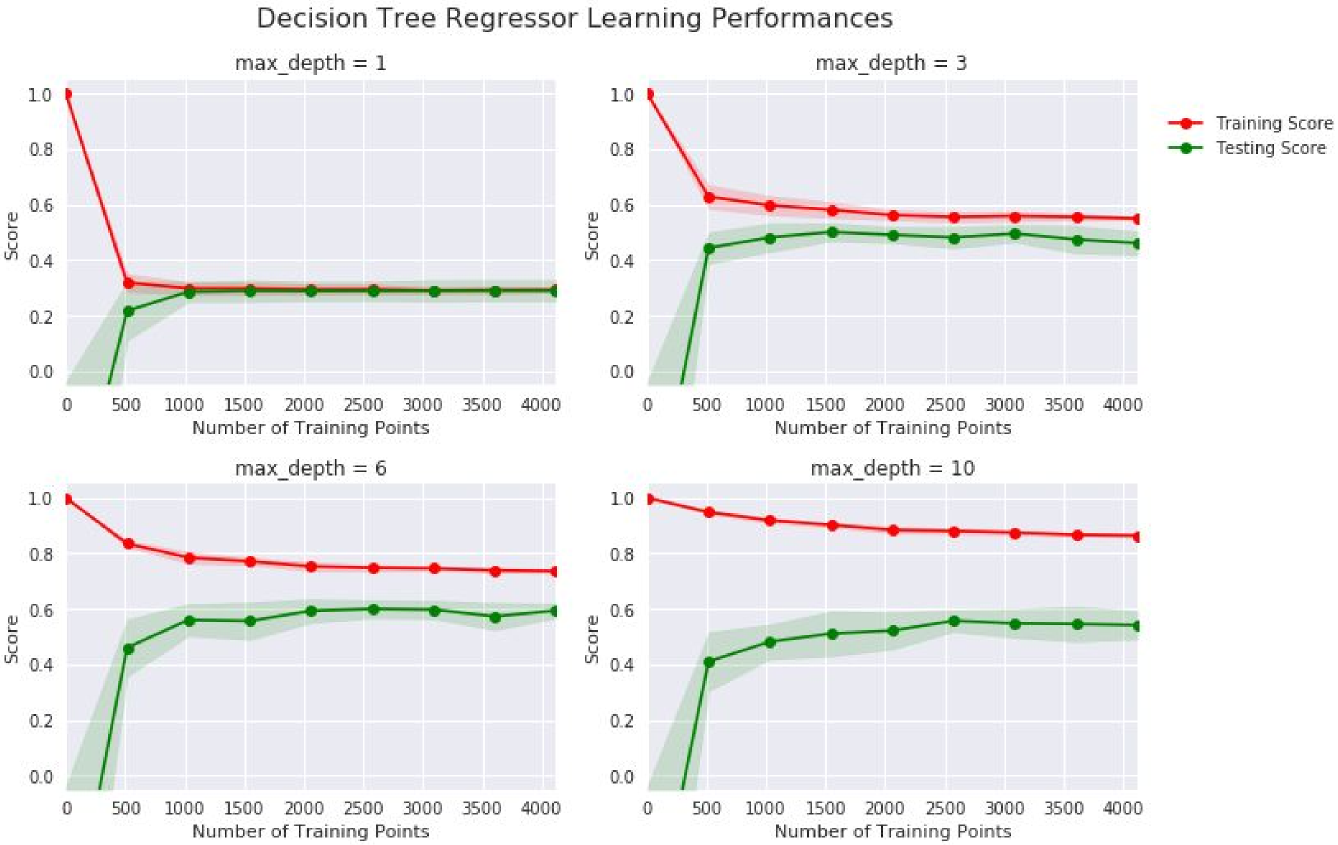


Fig. 8. - Queens, NY Property Sales



III. Methodology

(approx. 3-5 pages)

Data Preprocessing

As all of our data was relatively normally distributed, 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 this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

- *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
- *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
- *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- *Has an initial solution been found and clearly reported?*
- *Is the process of improvement clearly documented, such as what techniques were used?*
- *Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?
- Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?
- Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?
- Can results found from the model be trusted?

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- Are the final results found stronger than the benchmark result reported earlier?
- Have you thoroughly analyzed and discussed the final solution?
- Is the final solution significant enough to have solved the problem?

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- Have you visualized a relevant or important quality about the problem, dataset, input data, or results?
- Is the visualization thoroughly analyzed and discussed?

- If a plot is provided, are the axes, title, and datum clearly defined?

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- Have you thoroughly summarized the entire process you used for this project?
- Were there any interesting aspects of the project?
- Were there any difficult aspects of the project?
- Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- Are there further improvements that could be made on the algorithms or techniques you used in this project?
- Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?
- If you used your final solution as the new benchmark, do you think an even better solution exists?

Before submitting, ask yourself. . .

- Does the project report you’ve written follow a well-organized structure similar to that of the project template?
- Is each section (particularly Analysis and Methodology) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proofread your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?