THE PENNSYLVANIA STATE UNIVERSITY
SCHREYER HONORS COLLEGE

DEPARTMENT OF INFORMATION SCIENCES AND TECHNOLOGY

A COMPARATIVE ANALYSIS ON THE ABILITY OF MACHINE LEARNING MODELS TO CREATE A HOUSING PRICE INDEX USING INTEGRATED NEW YORK CITY HOUSING AND SCHOOL PERFORMANCE DATA

WILLIAM GAULT
SPRING 2020

A thesis
submitted in partial fulfillment
of the requirements
for a baccalaureate degree
in Applied Data Sciences
with honors in Applied Data Sciences

Reviewed and approved* by the following:

John Yen
Professor of Information Sciences and Technology
Thesis Supervisor and Honors Adviser

Suhang Wang
Assistant Professor of Information Sciences and Technology
Faculty Reader

* Electronic approvals are on file.

ABSTRACT

This thesis introduces data from the New York City Department of Finance and New York City Department of Education to compare the traditional methods of creating a Housing Price Index (H.P.I.) with a machine learning-based approach that incorporates the quality of school district each property belongs to, and compare their benefits and limitations. The thesis describes the key components of the process of creating an H.P.I. using machine learning methods: including data integration and feature engineering. The thesis also describes the assumptions made as well as the thought process behind each decision. It also describes the techniques used in integrating the two datasets, and my approach in experimenting and comparing the prediction quality of various machine learning-based models. For each model, I compare the performance of both a base-line model and a model that integrates the information about the performance of elementary school that each property is assigned to decide if the inclusion of school performance data has a positive effect on the ability of different machine learning models to predict home sale prices in New York City.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgements

I would like to thank Dr. Yen for his continued support since the very beginning of my undergraduate years. He has allowed me to thoroughly explore and learn from a field in which I had minimal prior knowledge of. I could not think of a better mentor throughout my years here at Penn State. It has been a joy and an honor working and learning from Dr. Yen not only throughout the thesis writing process, but throughout my entire time as an undergraduate student. Thank you also to Dr. Wang and his oversight of my final paper. I greatly appreciate the advice and knowledge he has given me throughout this process.

Chapter 1

Introduction

As 2020 concludes, more and more “Millennials” and “Generation Z” will begin looking for homes of their own. Often these new homeowners have young families and a low budget, making affordable housing difficult to find. New York City, for example, is one of the most expensive housing markets in the country. However, NYC is the headquarters for many companies, making it essential for young employees to relocate there. It is important for these young professionals to find affordable housing while also getting the best education possible for their children.

This paper utilizes the idea that young families are in search of affordable housing. By focusing on New York City, this paper describes the process in which machine learning methods were implemented to create a Housing Price Index (H.P.I.) for this particular group of people. By combining both previous home sale data from the five boroughs of New York (Brooklyn, Bronx, Manhattan, Queens, Staten Island) with public school performance data, I attempt to train multiple regression models reflecting the desires of young families: cheap housing, and quality schooling. The goal is to minimize R-Squared score and create a reliable location-centric model to use in future home purchase decisions.

Beginning with a literature review and background information on classic H.P.I.’s, this paper will transition into data collection, preprocessing and integration. Feature engineering, model selection, and model performance will follow. Finally, the models and results will be compared before discussing conclusions and future work that can be done in this field.

Previous Work

Using regression to predict housing prices is not a new concept. However, as with all markets, prices fluctuate, making it incredibly difficult to create an accurate model. Park and Bae (2014) tested whether machine learning models could predict if a house would close above or below its listing price. Using C4.5, RIPPER, Naïve Bayes, and Adaboost they found RIPPER to be the most effective model, and concluded that mortgage lenders and financial institutions could employ machine learning based models for better real estate appraisal.

Plakandaras, et al., (2015) found an effective means of combining Ensemble Empirical Mode Decomposition and Support Vector Regression into a hybrid machine learning approach. Compared to Random Walk, Bayesian Autoregressive, and Bayesian Vector Autoregressive models, the hybrid outperformed all competing models. This model tested if it could predict the entire US housing market's Schiller Index score based on a variety of features including GDP per capita, population, and inflation rate. The results were promising leading to a conclusion that the model could be used to forecast the business cycle of the US economy.

Barr, et al., (2017) utilized Cross-sectional Gradient Boosted Regression Trees to create an H.P.I. using new data features such as days on the market. They found that while the model used was effective, it was far more effective when the data was broken into property level rather than Zip Code. This added a layer to the data that was not reflected in the base data of sale price and zip code. They also concluded that traditional methods cannot offer reliable results as uncertainty about how a property sold in the past compared to its surroundings is essential in housing valuation.

All of the methodologies mentioned showed promising results. They all conclude that their models perform better than what is the base metric in each case. However, each has adapted a different means of performance. The models of Park and Bae (2014) predicts whether the final sale price falls above or below the listing price. This application would be used in more of a bank setting rather than on the individual purchase side as it acts similarly to a classification problem. Plakandaras, et al., (2015)

apply their model to an existing index in an attempt to predict the trends in the US economy as a whole. While very effective, this application is broad scale and not geared towards the individual buyer. It also is not creating its own index score but rather attempting to predict the score that a property would receive in an established index. Again, banks can use this application to predict potential recession in the economy. Finally, Barr, et al., (2017) use Gradient Boosted Trees to measure error in predicting individual property sale prices arguing that trees naturally group like-properties together. This is very similar to my research, but utilizes properties in Los Angeles rather than New York. Also, the models tested are different and their means of data integration are not consistent with the techniques used in this paper.

My research is geared to a more specialized application. Directed towards young families looking to purchase a home in New York City, my models attempt to utilize general property dimension, location, and schooling data to predict property sale price. It is often said that "location is key" and by creating my models I am testing to what extent location in regards to public school district can alter the price of single-family homes in New York City and how accurately this can be predicted.

Chapter 2

Standard Housing Price Indices

Before computer models and machine learning were relevant in the world of economics, H.P.I.s were created using basic mathematical strategies. While these indices can be affective, they are not always the most accurate and have inherent flaws. This section describes two of the earliest H.P.I.s, median sales index and repeat sales index, and their use-cases in the world of real estate. It then moves on to discuss potential limitations of each and why a machine learning approach is likely to be more accurate in predicting home values.

Median Sales Index

The median sales index is very simple to implement and utilize. The necessary data is time of sale and dollar amount. Each quarter, the median sale price of homes sold is calculated and used to create a price index. This is then reported by the National Association of Realtors among other organizations. Of course, this number can be modified in order to be more specific based on neighborhood, time, type of home, etc. However, it is often reported as a measure for all homes without any specialization making the number potentially misleading and difficult to interpret. The median sales index algorithm is explained below:

1. Establish and calculate the median sale price for the homes sold in the same time period.
2. Divide the index vector by the median sale price in time period 1, converting to an index.
3. Multiply the entire vector by 100 so the first period of the index equals 100.

Repeat Sales Index

The repeat sales index, while not entire complex, is more intricate than the median sales index. Introduced by Baily, Muth, and Nourse (1963), this index requires that a home must be sold at least twice allowing for a comparison of that house over time. This is intended to capture and control prices differences that may arise from construction, newer, and older homes. While the median sales index did not require any statistical modeling, the repeat sales index utilizes linear regression. More recent applications of the index account for the fact that price volatility is greater the longer time a house goes without being sold again. This algorithm is explained below (Barr, et al., (2017)):

1. Regress the logarithm of the sales prices at the second time period minus the logarithm of the sales prices at the first time period against the dummy variables for each time period.
2. Regress the squared residuals from the results in step 1 against the number of time periods between sales.
3. Repeat step 1 using the reciprocal of the square root of the results in step 2 as weights.
4. The estimated coefficients are exponentiated to for the index. The value of the index at the first time period is 1 as it is the base period for the index.
5. Multiply vector by 1 so the base period of the index equals 100.

Limitations of Standard H.P.I.

While these standard indices may seem like a good choice to evaluate housing price, there are actually many flaws that can lead to poor judgement. The median sales index faces the main issue that the composition of sales are not uniform over time (Barr, et al., (2017)). This can be caused by fluctuations of supply, i.e. less expensive homes are being sold. This doesn't accurately represent a price drop, but rather misrepresents a supply change in the market. This can make the index difficult to understand and interpret.

Another issue with these indices is that the variance in the market is not completely represented. Instead, the most represented areas are those with high turnover. The more often a home is sold, the more weight it will hold in the index calculation. Additionally, if an area is highly represented in the data, and the prices in this area are more likely to change, price fluctuations in other markets will not be accounted for. This misrepresentation can cause the index to sway one way or another, throwing off judgment and index applicability.

One of the biggest problems with these indices are the misrepresentation of locational data within the index. Wealthier neighborhoods are more likely to see larger growth than more impoverished areas. This is cannot possibly be accounted for in either the median sale index or the repeat sale index because the input data does not include locational data.

Modern technology has allowed for more techniques in an attempt to accurately predict home prices in the housing market. Machine learning can be applied to utilize statistical methods that require large input data. These statistical methods allow for more robust models that are likely to be better indication of home price. More input data means a more accurate result that can be implanted and tested in similar housing regions.

Barr, et al., (2017) suggest a machine learning model called a gradient boosted home price index. Decision trees are formed utilizing input such as Zip code, metropolitan area, etc. The results suggest that the gradient boosted trees greatly outperform both the median sale index and the repeat sales index. While these conclusions are promising, the gradient boosted regression trees were applied to large areas in the Los Angeles Metropolitan Area. My research attempts to target a specific group, young families, looking to move into the vastly different New York City. By incorporating school district data, I am able to reflect the importance of location in regards to school choices families make when deciding on affordably priced housing.

Chapter 3

Integrating School Performance Data with Real Estate Data

Many previous projects have been carried out to see how accurately machine learning methods can predict housing prices. However, in order to make this project novel, I decided to test the models with data that is not commonly utilized. This data was school district rating data, and being that the targeted home buyers are young families, the school data being tested is elementary school rating. Humans have an uncanny ability to analyze lots of data at the same time. It is not explicitly stated in the brain, but subconsciously many data points are combined together when trying to price a home. Machine learning methods, however, do not have this ability to arbitrarily analyze data, so each decision made was in the interest of creating the best possible model for the targeted user group. This section describes the decisions I made when creating the dataset to be used in training and testing the models.

Identifying Suitable School Ranking Data

The city of New York has many different departments and committees responsible for managing community affairs, finances, and education, among others. The NYC Department of Education is responsible for the Public Schools. According to the NYC DOE website, the department decides curriculum, school districts, school standards, and school testing. I found that this department releases a "Snapshot" of schools' yearly performance meant to summarize how the school is doing in regards to the standards set out in the beginning of the year. Naturally, this would be a great place to begin data collection on school information and rating. However, there was the glaring issue that this data only ranged back to 2014. My dataset was from 2010 onward leaving a large gap in the available information to base my rating system on. I originally thought it may be a good idea to generalize the available data to the unaccounted-for years, but this would be irresponsible and inaccurate. Requirements, facilities, and faculty members change over time which may mean a school could have drastically increased or

decreased in performance. A second issue with this data is that all of the rankings are categorical variables. For example, the "Rigorous Instruction Rating" can range from "Not Meeting Expectations" to "Exceeding Target." The rating criteria was not available making these kinds of classification particularly challenging to interpret. The NYC Public School Snapshot was not a viable option.

The next idea I had was to visit a public website of "School Rankings." This option gave a good representation of what the public thought of each school as a survey system was implemented to collect public feedback. Again, however, the responses were broken down into arbitrary categorical answers. The data was also not ideal for processing as there was no downloadable format and the site would require scraping. Finally, this option is susceptible to response bias. People answering the survey are more likely to be either extremely please or upset with a particular school. This can drive responses to the extreme and not capture an accurate representation of how the public feels or how the school actually performs.

The last and ultimately final data source analyzed was school testing scores. Test result scores measure the students' performance on standardized tests that are distributed by the state. This is a measure of how much the students learned and provides a baseline for where the average student should be competent. This data is numerical and is consistent for all schools across the city. There is no bias and gives an accurate representation of performance. The main issue found with this data is that state testing begins in the third grade. This is not useful for students aged Kindergarten through second grade. However, in a home pricing scenario, this issue can be justified. Families looking to move somewhere are likely to stay in that home for at least a few years meaning a third-grade student is likely to be living in the same home they were in during kindergarten. Also, there aren't likely to be too many discrepancies between overall quality among grades. The school as an entity is going to carry an overall quality with it, not individual grades within the school. Standardized testing data offered a numerical value to evaluate each school's overall performance while also limiting bias and creating a constant measure across each school and district. This was the data used to evaluate school performance in the overall dataset.

Data Integration and School District Scoring

With all of the school testing data available online, I was now able to download the scores into its own file. Again, I had to download each year individually, clean the data, then integrate it back into one total dataset containing all boroughs and all years. Next came an important decision as I had to decide how to evaluate the scoring data. A common approach to continuous numbers as features is to group them into categories (Salcedo-Bernal, et al., 2016). This seemed like a viable option as the NYC Department of Finance had their "School Performance Snapshot" containing categorical variable such as "Exceeds Expectations" and "Approaching Proficient." The reason I did not use this approach, however, was that I truly wanted to differentiate each school. By bucketing schools together into categories this makes it impossible for the model to consider how much a few test points can affect home price. For example, a school scoring 15 may be bracketed as "Average" along with a school that scored a 40. These are just arbitrary numbers, but there is no way to identify how much the 25-point different may have caused a home price to rise. Yes, grouping the numbers into categories would have made training and testing easier and faster, but the interpretation of the results would have been limited.

This decision drove me to include the test scoring for each school as a continuous, numerical feature. There was still a major decision that needed to be made. Each test consisted of two parts: Mathematics and English and Language Arts. I first began by considering each exam section and what it tells about the student population. Mathematics can be considered a universal language understandable by all ethnicities and nationalities. English Language Arts (ELA), is much more challenging for those who do not consider English as their first language. The thought was that proficient levels of ELA would be more challenging to achieve for a diverse group of students than all English speakers. This would be reflective of the school as it is common for certain demographics to be in a community together (Walks, 2014). Meaning, a school is more likely to be made up of either many non-English speakers or many English speaker. A truly diverse student population is less likely. Additionally, the current emphasis on STEM studies lead me to believe that Math scores carried more weight in judging school performance.

While this may be the case, it does not mean that I can consider this in the original problem statement. My goal was to find out how school performance affects housing prices not how Math scores affect home prices. Therefore, I needed a way to quantify each into one complete score. The most straightforward way was to average the two. I took the mean of each grade in each school for a particular and averaged both the ELA and Math scores together. The result was one final continuous number that represented the actual test performance throughout all grades of each individual school.

Location-Based Integration into the Larger Data Set

The final, and most difficult, part of school performance data inclusion was the actual integration into the housing price dataset. Remember, the original data being used in the models was housing price data for sales in each of the five New York City boroughs. While this dataset included fields such as "Address", "ZIP Code", and "Neighborhood", school designation is determined based on New York City "School Districts." When beginning this project, I was under the incorrect assumption that "Neighborhood" classification would be enough to determine where a student would attend school. However, each school is actually determined by school districts that are created by the NYC Department of Education. These districts are not only made of overlapping neighborhoods, but the neighborhood names are not always consistent as the housing data being used was provided by the NYC Department of Finance. The integration based on neighborhood was not a viable option.

The next idea I had was to use Address in order to identify which school district the home fell within. I was unable to find any data capable of mapping address within a school district, but I was able to collect a dataset consisting of all school district boundaries in longitudinal and latitudinal coordinates. While promising, each address was still in the form of street number and zip code. If I was able to collect the coordinates of each address, I would then be able to map them to their respective school districts. Searching many methods lead me to one common path: using an online mapping API to determine and

retrieve address coordinates. Google offers access to the Google Maps API, but there is a cost associated with it. GeoPy, however, is a free and open source library capable of geocoding different addresses. This seemed like the perfect option for geocoding each address in my dataset before integrating school performance data. Just when I thought I had found a solution, I faced another setback. In order to prevent over use and potentially malicious attacks, the GeoPy request servers were limited to a maximum of one request per second. My dataset was far too large to be sending individual requests to the server as completion of the code could have taken days. In the interest of time, I decided there must be another solution to my integration problem.

After digging around and searching for different options I stumbled upon yet another New York City government agency: the NYC Department of City Planning (DCP). Responsible for setting the framework of the City's land use, I quickly found that there was much more than just neighborhood and Zip code classifications. Community district, council district, and census tract are all locational classifications that were available for use and potential integration purposes. The one that I was most interested in was the Neighborhood Tabulation Area (NTA). NTAs were created by the DCP to project populations at a small area level. In other words, NTAs are a kind of neighborhood in themselves. The most exciting aspect about this discovery was that deep in the NYC Department of Education files, I was able to find a dataset for school locations. Each school name, address, and, most importantly, NTA was listed in this dataset. Also, each neighborhood in the housing dataset was consistent with the NTA names. While some preprocessing was necessary to accurately map the neighborhood names together, I was able to add the field "School Performance" to the school location data before integrating this information into the housing dataset based on NTA.

One other aspect of the final data is that it must be encoded to work with the given models. This means that categorical variables are given numerical values in a binary matrix. Utilizing school performance as a feature in the data was optimal because a feature like "Neighborhood" has over 150 unique instances. These must all be encoded and adds far too many additional features to the data set.

Since school performance is representative of location, I was able to remove neighborhood completely. This sped up model training and also made the results more easily interpretable. The final dataset used in the model training and evaluation consisted of the features: Borough, Tax Class at Present, Land Square Feet, Gross Square Feet, Year Built, Year Sold, Quarter Sold, School Performance, and Sale Price.

This process is the perfect example of what a true data science project takes. Human judgement and consistency are required to create the optimal dataset for model fitting and testing. The original plan was faulty, forcing me to change my approach and adapt. Decisions were made in order to compile a usable dataset and I was finally able to move on to fitting, testing, and comparing my models.

Chapter 4

Predictive Modeling of Home Values

The prediction of housing prices is a regression problem in nature. This is ultimately much more difficult than other machine learning problems such as classification. Trying to predict a specific, continuous number, naturally, is harder to do than predicting a bucket. In this section I will describe the models used in trying to get the most accurate regression results. I will describe the data used in the model and the steps taken to create a model that can accurately predict home sale price in New York City.

Feature Engineering

It is common belief that machine learning can be done completely autonomously. However, human intervention is essential to an accurate and reliable model. Feature engineering is the step in which the data scientist cleans, preprocesses, and alters the input data in order to have the best results when fed into the model. It can be incredibly laborious and often requires a great deal of thought in create the best dataset possible for training, testing, and ultimately creating the model.

Data Description and Selection

The data used in this project comes from the New York City Department of Finance dataset. This detailed dataset gave information on the sales of homes in the five boroughs of New York City (Brooklyn, Bronx, Manhattan, Queens, Staten Island). The data provided is from 2003 until 2019. Being that 2020 is not currently available, this data was almost completely up-to-date including the most recent, complete year. The following chart is a description of the fields and what each of them means. It is incredibly important for the data scientist to know what each feature means as many are encoded and may not have any applicability to the particular project.

Feature	Description
Borough	Number relating to borough: 1. Manhattan 2. Bronx 3. Brooklyn 4. Queens 5. Staten Island
Neighborhood	Neighborhood name where property is found
Building Class Category	Description of building class
Tax Class at Present	Building Tax Classification: 1. Most residential property of up to three units and most condominiums that are not more than three stories 2. All other property that is not in class 1 and is primarily residential 3. Most utility property 4. All commercial and industrial properties
Block	City block number
Lot	City lot number
Easement	Easement contingent to property (if applicable)
Building Class at Present	Symbol for building class
Address	Unit Address
Apartment Number	Apartment number (if applicable)

ZIP Code	Unit ZIP Code
Residential Units	Residential units found in property
Commercial Units	Commercial units found in property
Total Units	Total units found in property
Land Square Feet	Land (property) square footage
Gross Square Feet	Total square footage of building
Year Built	Year unit was constructed
Tax Class at Time of Sale	Tax class when unit was sold
Building Class at Time of Sale	Building class when unit was sold
Sale Price	Price paid when unit was purchased
Sale Date	Exact date of sale

Table 1. Data Features and Descriptions

The data available ranged from 2003 to 2019, but, while extensive, not all of this data was necessary or appropriate. In reassessing my initial problem statement, it is important to understand how difficult it is to predict housing prices even in a perfect market. However, perfect "regular" markets make prediction easier and more applicable as irregularities are uncommon and challenging to predict in themselves. I decided that the best approach would be to use data only from 2010 until 2019. This was to account for the housing crisis of 2008 when home prices and ownership plummeted. In 2010, the housing market was still recovering, but was almost clear of the previous market crisis. By limiting the data to these 9 years, a "normal" market became the grounds for my model training and testing.

Preprocessing and Cleaning

Data never comes in the necessary form for processing and model fitting. Often it is messy and must be cleaned before it can be useful in a machine learning environment. This was the case with this

project as each borough's data had to be downloaded individually by year. Upon initial inspection, the columns for the all of the years were consistent, but the column headers were not all the same. This required manipulation of the column names as concatenation would be difficult without complete consistency. For example, the 2018 files contained the column name "Tax Class as of Present." This has the same meaning as "Tax Class at Present", but it still needed to be changed.

Next came the issue of unidentifiable characters. When utilizing different models, they can only understand some symbols. This was an issue because the monetary values in the dataset were formatted as such: $1,500. Both the comma and the money sign would raise errors when trying to pass these values into later models. Iterating through each field, I removed problem characters that were identifiable. In addition to the comma and money sign, some zeros were denoted with '-' and had to be converted to the number "0". Extra whitespace was removed from each feature column as well as blank instances that were found throughout the data. Once this was all complete, I was able to concatenate each individual borough's data to consist of every year before combining all borough data into one large dataset for training and testing.

As mentioned previously, understanding the data and its features is essential to creating a model that is a solution of the actual problem being solved. This dataset contained information on non-residential properties. Since this model is intended for young families in search of a residence, information on non-residential properties is not useful. Iterating through the data I removed all sales of properties that were not tagged with the Tax Class of 1 or 2 at the time of sale. This removed most utility properties and all commercial and industrial properties. Next, it was noticeable that some properties sold for $0 or suspiciously low prices such as $100. A deeper look into the data description provided information that some properties can be handed down to relatives. While this is not a sale, it is a property exchange and is therefore recorded in the dataset. Again, I iterated through the data and removed all properties that sold for less than $1,000. This was to remove filler values that were put as placeholders.

An important aspect of this dataset to understand is that it does not contain information on the age of the home buyer. Since the target audience of this model is young families, I had to use my own intuition and judgement to narrow down what I believed to be properties desirable for this demographic. This poses a potential challenge to the problem because there is no concrete metric for evaluating this feature of the data. It is based on personal judgement which makes it harder to generalize.

A commonality in home value fluctuation is at what point in the year the property sold. This raw data contained the feature Sale Date, but it was in the form of datetime. Simplifying this feature would make interpretation easier for the model as well as the human. I created two more features: year sold and quarter sold. By categorizing these values from the date, the price fluctuations over time were more easily identifiable as home prices trend upward year to year and properties tend to be more expensive in the spring and summer quarters. I then removed Sale Date as a feature because the most important aspects of the date were simplified and included in the new dataset.

The last step before any models could be tested was the removal or alteration of any additional, unnecessary data. Below is a chart of the features that were removed or altered in any way and a description of each:

Feature	Alteration	Explanation
Tax Class at Present and Tax Class at Time of Sale	Removed	These were already used to remove all non-residential properties from the data. They no longer carried and significance as all properties in the dataset were now residential.
Building Class Category	Removed	This was just a longer description of Building Class at Present. They offered the same information just made it easier for the human to understand. It is unnecessary to repeat this information in the training and testing of the model.
Apartment Number	Removed	Too specific for each apartment. Not used for determining school district and it is incredibly difficult to relate to other properties.

Address	Removed	Too specific to each sale. Not generalizable and not useful in a machine learning model.

Table 2. Removed Features

The final data set was made up of 22,252 home sales with the features: Borough, Land Square Feet, Gross Square Feet, Year Built, Building Class at Time of Sale, Year Sold, and Quarter Sold.

Linear Regression

Linear regression is one of the basic regression models used in machine learning problems. It has a simple theory but can be altered to fit many different applications. In its most basic form, a linear regression model attempts to model the relationship between two variables by fitting a linear equation . Linear regression utilizing two variables, an independent variable (x) and dependent variable (y), is represented by the equation:

$$y = mx + b$$

Where m is the slope of the line and b is the intercept of the y axis. The same idea can then be extended to a problem that contains multiple independent variables, which is the case with this project. This multivariate linear regression problem contains many different variables that are trying to explain and predict the housing price. The equation is as follows:

$$y = b_0 + m_1 b_1 + m_2 b_2 + \cdots + m_n b_n$$

Again, y is the dependent variable trying to be predicted and m is the slope. However, in multiple dimensions, instead of a line or a plane (3-dimensions), we have many features resulting in what is referred to as a hyperplane.

Essentially, during linear regression, the model is fitting many different lines to the data points and the one that is the best fit is used as the linear regression equation. This is determined by the cost function also known as the Mean Squared Error (MSE):

$$minimize \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Where n is the number of training data samples. The difference between the predicted value ($pred_i$) and the true value (y_i) is the error difference. By squaring these values, summing them over all the data points, and dividing them by the number of data points, we are able to create the MSE, and minimizing this value is what results in the best fit linear regression equation.

The final and most challenging aspect of linear regression is iterating through the different lines in order to find the one minimizing the MSE. Any person can start with arbitrary numbers and continue changing them by guessing and checking. This is computationally expensive and unnecessary as the process is optimized through gradient descent. Gradient descent involves taking partial derivatives (gradients) from the cost function and working our way to a minima. Below is the equation for updating the terms based on gradient descent:

$$b_0 = b_0 - \alpha * \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i)$$

$$b_1 = b_1 - \alpha * \frac{2}{n}\sum_{i=1}^{n}(pred_i - y_i) * m_i$$

The partial derivatives are the gradients and they are used to change the terms (b_1, b_0). Alpha (α) is a learning rate that must be explicitly stated during model formation. This determines the step size of the descent. The larger the learning rate the faster the terms converge on a minima. However, there is a chance of over-shooting the minima. A smaller learning rate means smaller step sizes which is more likely to get closer to the minima, but this descent takes longer.

Linear regression is a good model to begin testing with because it assumes that there is a linear relationship between all of the features and the sale price of a house. While this is most likely not the case, the performance of the linear regression model will provide a baseline for comparison with the other models.

Neural Network

Much of the talk up to this point has been about machine learning application. While machine learning is very affective, there reaches a point where better results are desired in more complex problems. This is where deep learning comes in to help. Deep learning is a bit more challenging to explain as it functions like a Blackbox. There are hidden layers used in predictions that are not seen by the human. However, this lack of explainability is justified by increased prediction and accuracy. Neural Networks are a staple of deep learning as they form the basis for many applications. A Neural Network functions much like any other machine learning model in that it is trying to make the best prediction from a set of inputs.

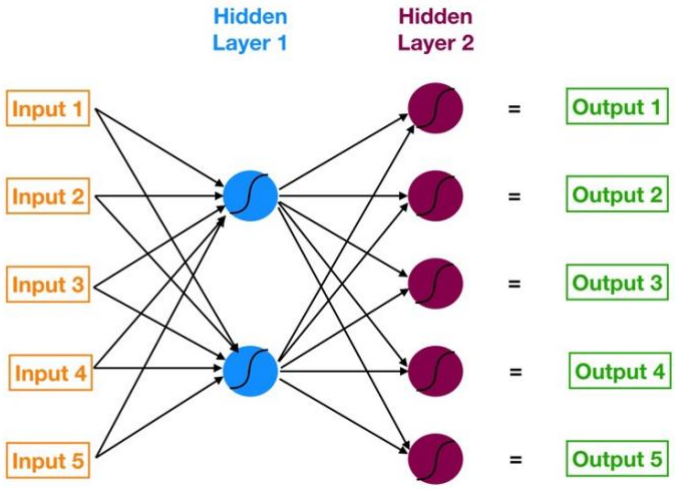


Figure 1. Neural Network Process Flow

In the example above there are five inputs and two hidden layers. Each layer consists of neurons (like the human brain) which are each responsible for a mathematical decision. Hidden Layer 1 consists of two neurons while Hidden Layer 2 consists of five neurons. Additionally, each neuron and its input carry a weight to be considered in the transformation. This weight essentially determines how the input is transformed before it reaches and is passed to the next neuron. The simplest way to describe this entire

process is that each neuron acts like its own smaller model and each input is a feature in the dataset that gets transformed by a weight. Each layer is a stack of models which later feeds into another stack of models further into the Neural Network. The goal is to find the set of weights and neuron biases that minimize the cost function.

Similar to the linear regression model, a Neural Network must also be trained in a similar manner. A cost function must be defined before applying gradient descent to minimize that cost. Although this is a neural network application, regression is the underlying problem, so the cost function used is Mean Squared Error as above. Since this is a neural network with many different neurons and input vectors, gradient descent is incredibly challenging. Back propagation can be used to limit the error or the neural net. Working backwards through the neural network, the error of each neuron is calculated. If a neuron has a much greater error than the rest, those respective weights and biases are tweaked in order to limit the error of that particular neuron. This is repeated over and over to minimize the error essentially replicating gradient descent through back propagation.

Chapter 5

Model Evaluation and Comparison

Now that model selection was finalized, the data was cleaned, and the models were completely understood I was able to move on to fitting and testing my models. Sklearn is a machine learning library with many different models that are easy to implement. I decided to use this library as it is optimal for easy, quick, interpretable results. My first iteration through the linear regression model fitting and testing showed that the prices within my data were very unbalanced and spread across a wide range. This poses a problem for the models because if unbalanced data is used to train, the model predictions will be weighted more heavily towards the common training data. In order to mitigate this problem, I bucketed the prices into three ranges. The lowest bucket contained houses that sold for under $500,000, the middle bucket between $500,000 and $1 million, and the highest was houses that sold above $1 million. Throughout the remainder of the project, whatever action I took on one subset, I also had to take on the rest of the subsets. The breakdown of the price buckets can be seen below:

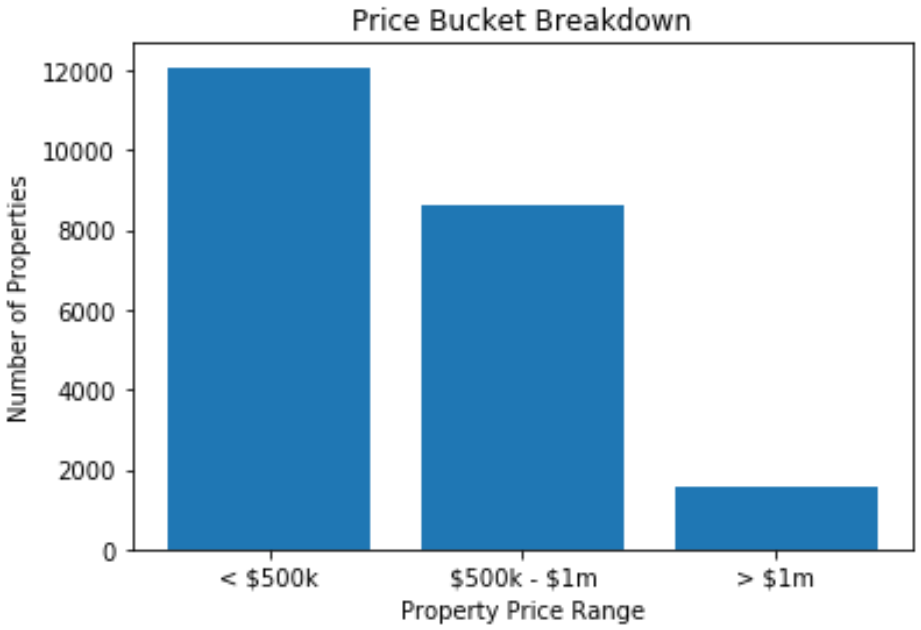


Figure 2. Housing Price Bucket Breakdown

Data Splitting and Cross Validation

The first step in any machine learning process is to split the data into training data and testing data sets. At its core, a model is trained using the training data which consists of the features and the dependent variable (home price). Once this model is fit to this training data subset, it can then be used to predict the outcome for the testing data set which the model has not yet seen. Once the output is calculated, it is then compared to the ground truth of the testing data, which is, in this case, the actual sale price in the testing data. From this comparison, an accuracy score is calculated allowing interpretation of how well the trained model is able to predict future outcomes.

To split the data for training and testing, there are many different methods. The simplest is a basic split based on a partition size. The data could be split 80:20 where 80% of the data is used for training and 20% is used for testing. This is effective but often not the most robust as the order of the data may cause bias during training. One way to mitigate this issue is to randomize the order of the data and perform K-Fold cross validation. This method breaks the data into a number of "folds", i.e. ten-fold splits the data into 9 training folds and 1 testing fold. These are then passed into the model for training and testing in order to produce an accuracy score for that model. Each fold is iterated through and a new accuracy score is calculated. At the end of the iterations, these accuracy scores are averaged to achieve a mean accuracy score for the model after cross validation is complete. Below is a good visual representation of how cross validation is carried out:

Figure 3. K-Fold Cross Validation Process

Since each fold generates a different model, there are different accuracy scores for each. This means there needs to be a way to evaluate the overall performance of each fold. I decided to take the average of each R_2 score in order to evaluate my models. This gives a good summary of how the model performed on average across each fold. Additionally, I used 10-fold cross-validation, meaning there were 10 different models generated for each larger machine-learning method, and their accuracy scores were average to create one comprehensive accuracy measure.

R_2 score is a statistical measure of how close the data are to the fitted regression line. It is the percentage of the response variable variation that is explained by a model: Explained Variation/Total Variation. The closer R_2 is to 1.0 indicates how well the model explains the variability of the response data around its mean. Therefore, the closer to 1.0 the R_2 is, the better the model is performing.

Linear Regression Results

I began by passing in housing data into the linear regression model. This data did not include school performance score as it was to serve as my baseline model for comparison. There were some parameters that were passed to the model that must be explained. First, I normalized all of the data based on a predetermined normalization measure in the model call. This function takes the data before

regression and normalizes it by subtracting the mean and dividing by the l2-norm. I also used 10-fold cross validation and shuffled the data before training the models. Below are the results of the linear regression models trained and tested on all the different price-based sub categories.

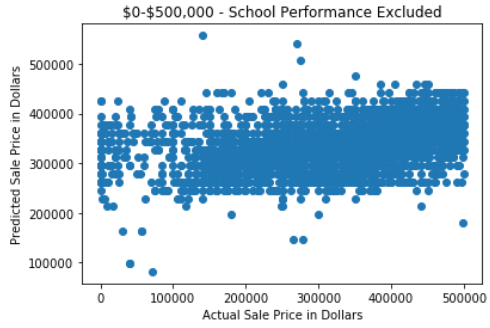
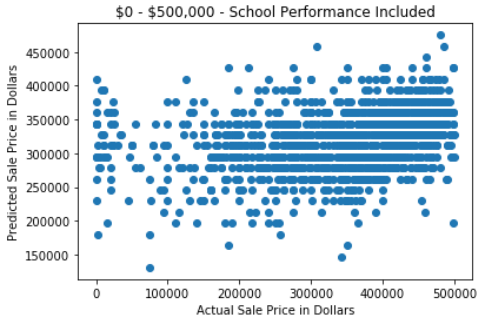
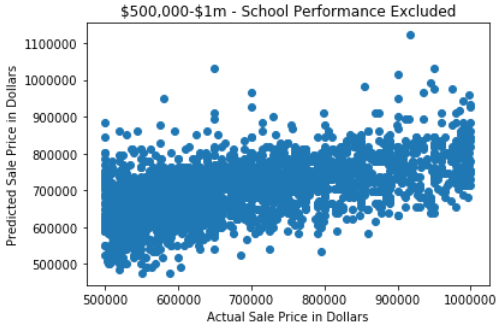
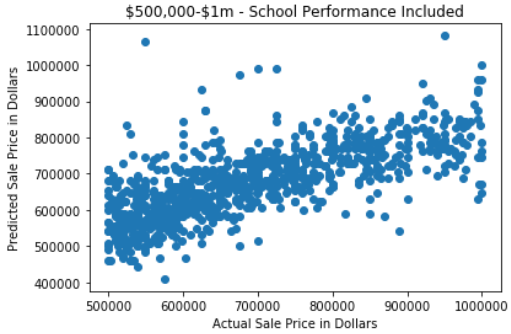
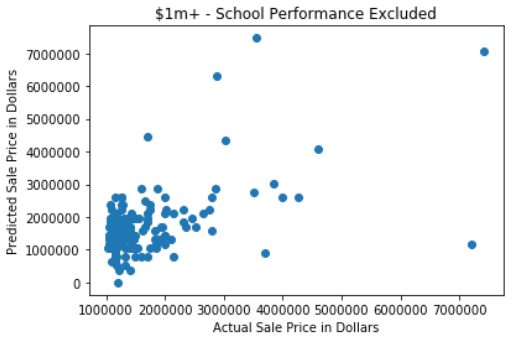
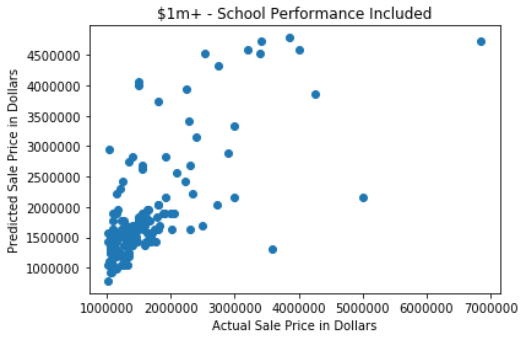
	Housing Only Data	**School Performance Included**
$0 - $500k		
$500,000 - $1m		
$1m+		

Figure 4. Linear Regression Performance Visualization

	Housing Only Data (R^2 Score)	School Performance Included (R^2 Score)
$0 - $500k	0.149335	0.205234
$500,000 - $1m	0.306137	0.32468
$1m+	0.124267	0.14276

Table 3. Linear Regression Accuracy Scores (R^2)

An additional metric that is often used during model evaluation is Mean-Squared-Error (MSE). MSE is an effective and easily interpretable metric for regression problems as well as being simple. Essentially, MSE measures error by taking the distances of the points to the regression line and squaring them:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

In this case, Y_i is the actual value of the home sale price and $\hat{Y}_i$ is the predicted value of the home sale price. The higher the value of MSE, the worse the model performs because that means there is more error. So, unlike R^2, a lower MSE is desired. Below is a chart of the MSE for each of the home price buckets for each data set:

	Housing Only Data (MSE)	School Performance Included (MSE)
$0 - $500k	12934.5792	11679.5594
$500,000 - $1m	10258.3692	9688.47921
$1m+	13462.6979	13183.7993

Table 4. Linear Regression MSE

Neural Network Results

Now that the results from the linear regression model were visible, I needed to run the Neural Network test to see if this model was a better fit for the housing and school dataset prediction. After altering parameters based on performance, I found that the optimal hidden layer size was 10. Additionally, the best performance came when 5 nodes were included in each of the 10 hidden layers. Below are the performance scores for different hidden layers sizes with 5 nodes in each hidden layer that were tested on the middle price bucket with the school performance data integrated:

Hidden Layers	6 Layers	7 Layers	8 Layers	9 Layers	10 Layers
R^2 **Score**	0.31435	0.32267	0.38283	0.38274	0.42847
MSE	18379.3829	14394.5933	15283.5672	11562.6529	9189.43857

Table 5. Neural Network Hidden Layer Performance

As you can see, 10 hidden layers with 5 nodes preformed the best. Additionally, the activation function in the hidden layer was 'relu'. This stands for 'rectified linear unit function' in which the function act as such: f(x) = max(0, x). The solver used for weight optimization was 'lbfgs.' This represents an optimizer in the family of quasi-Newton methods. Finally, set the maximum iterations to 1000. This parameter is used as a stopping point if the solver does not iterate to convergence. The results of this model are below:

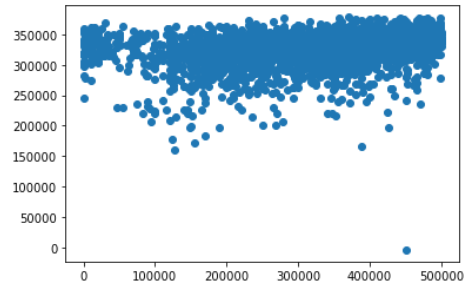
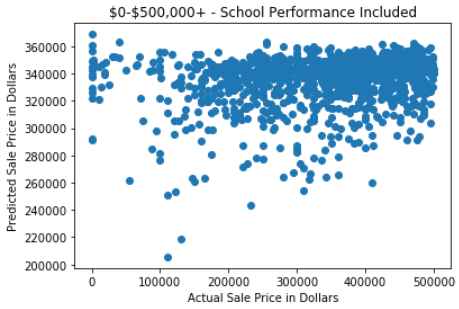
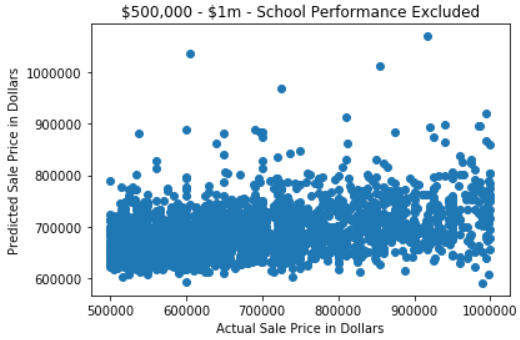
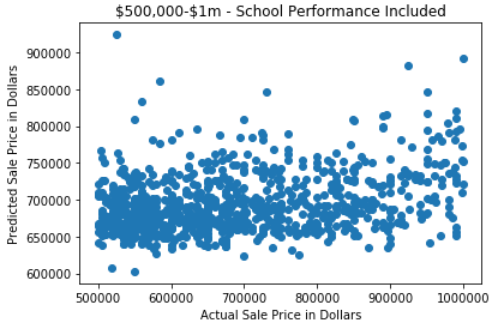
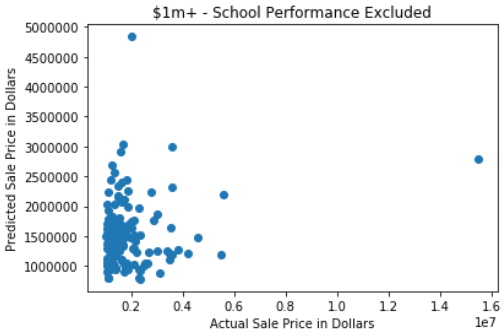
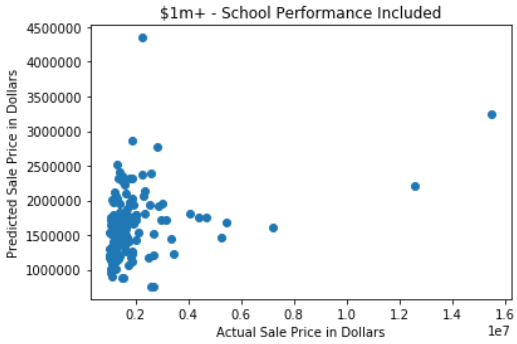
	Housing Only Data	School Performance Included
$0 - $500k		
$500,000 - $1m		
$1m+		

Figure 5. Neural Network Performance Visualization

	Housing Only Data (R^2 Score)	School Performance Included (R^2 Score)
$0 - $500k	0.137493	0.187351
$500,000 - $1m	0.39385	0.42847
$1m+	0.043895	0.079039

Table 6. Neural Network Accuracy Scores (R^2)

	Housing Only Data (MSE)	**School Performance Included (MSE)**
$0 - $500k	12233.5792	11448.9503
$500,000 - $1m	10027.7886	9189.43857
$1m+	13215.452	12875.4938

Table 7. Neural Network MSE

Chapter 6

Discussion and Future Work

Comparing the results of the models above, I found some interesting patterns and discoveries. First, the best and most consistent results of all the models came from the subset of sale prices between $500,000-$1 million. These results are reflected in both the Mean-Squared-Error and the R_2 score. This is not entirely surprising since this subset contained the most data points. Usually, the more data there is during training, the better the prediction is going to be as training is more robust. This can also be observed inversely as the worst predictions came from the over $1 million subset. This subset contained the least amount of data and therefore the models struggled to train and accurately predict sale price. Additionally, this subset had the potential for the largest range of sale prices as prices could have been anything over $1 million. This is itself can cause an issue from data balance. The poor performance can be seen especially in the Linear Regression model with school data excluded as the model had a very difficult time predicting sale prices reflected by a poor R2 score and a high MSE.

While the inclusion of school performance data seemed to boost accuracy scores, they did not show extremely significant improvements. Yes, the accuracy got “better,” but there are a few potential issues with this conclusion. The dimensionality of the data is relatively low as the number of features is

only 9 with school performance included. Higher dimensionality often results in more robust and accurate models. Also, reflecting on the data, it is not the ideal data for assessing home pricing. A lot of home value comes from the attributes of the home itself on an individual level. This dataset did not include information on aspects such as number of bedrooms, number of bathrooms, air conditioning, etc. These tend to have a greater effect on price and would hold more weight than Tax Class and Year Sold.

Finally, the neural network seems to have produced better, more consistent predictions. Iteration after iteration, the neural network had higher R_2 scores and lower MSE than the linear regression models. This is does not come as much of a surprise since it was unlikely that the sale price and all of its features displayed a linear relationship. Instead, the neural network was able to learn a model that was not linear, making the model more accurate in future predictions. One thing to be aware of, however, is overfitting. Too many nodes and hidden layers can result in a model that is extremely accurate and well fit to the training data. However, the purpose of a predictive model is to accurately predict unknown values. An overfit model is not generalizable and would not be able to accurately predict data values outside of the testing set.

Model Coefficient Evaluation and Interpretation

After analyzing the accuracy metrics and understanding what may have been changed or how well each price bucket predicted the unknown values, it is important to understand how each feature affects the dependent variable. For this problem it is particularly important because we are trying to evaluate how school performance affects home price. While there are many features, seeing the effect of the school performance variable on the home price variable can give great insight into how important this factor truly is. Below are the tables of the coefficients of each model run, including each fold:

Linear Regression $0-$500,000 Housing Data Only

	Land Square Ft.	Gross Square Ft.	Year Built	Year Sold	Borough	Building Class at Time of Sale	Quarter Sold
Fold 1	4.71831	18.7354	250.779	8.678	-3e+17	-1e+18	-1e+17
Fold 2	5.1599	19.088	250.069	8.7595	2e+17	-1e+18	1e+18
Fold 3	4.5434	20.4502	207.53	9.17307	-3e+17	2e+18	1e+18
Fold 4	5.6991	19.469	227.233	8.994	4e+18	5e+17	3e+17
Fold 5	5.3844	20.6716	214.276	7.748	3e+18	-1e+18	-1e+18
Fold 6	5.1716	19.788	271.444	8.666	9e+16	3e+16	2e+15
Fold 7	4.378	17.768	203.9981	8.2537	1e+18	2e+18	3e+17
Fold 8	5.364	18.953	235.9324	8.73683	2e+18	1e+18	2e+16
Fold 9	4.4876	17.9191	259.526	8.8243	2e+18	1e+18	2e+18
Fold 10	5.5922	18.2474	201.4217	8.5391	-1e+18	2e+18	-1e+18

Table 8. Linear Regression Lowest Bucket Feature Coefficients No School Data

Linear Regression $500,000-$1m Housing Data Only

	Land Square Ft.	Gross Square Ft.	Year Built	Year Sold	Borough	Building Class at Time of Sale	Quarter Sold
Fold 1	21.0166	38.7065	22.4387	22.2218	-1e+17	1e+18	1e+18
Fold 2	24.2847	38.2788	24.3891	22.371	3e+17	1e+18	-1e+17
Fold 3	20.4782	36.2843	22.3328	23.1546	-1e+17	-1e+18	1e+17
Fold 4	21.7388	37.0302	22.9002	21.5678	2e+18	-1e+17	2e+18
Fold 5	21.6823	36.2223	23.7912	20.1873	-1e+18	-1e+18	1e+18
Fold 6	21.3726	36.4788	24.7702	19.2887	3e+17	2e+17	-2e+17
Fold 7	23.4561	39.2918	23.0049	22.9366	1e+18	3e+18	-3e+18
Fold 8	22.7836	34.9485	25.1892	21.2456	1e+18	1e+18	1e+17
Fold 9	24.9275	38.3444	23.5063	22.3343	1e+18	2e+18	2e+18
Fold 10	23.4326	38.9289	22.3998	22.3887	-1e+18	-2e+18	1e+18

Table 9. Linear Regression Middle Bucket Feature Coefficients No School Data

Linear Regression $1m+ Housing Data Only

	Land Square Ft.	Gross Square Ft.	Year Built	Year Sold	Borough	Building Class at Time of Sale	Quarter Sold
Fold 1	29.899	550.8464	69.325	34.598	-6e+17	-1e+20	7e+18
Fold 2	28.3847	558.2839	38.608	24.407	-3+18	-1e+19	3e+18
Fold 3	28.9255	555.273	12.5884	23.0046	2e+19	7e+19	-1e+20
Fold 4	27.4709	551.2992	80.1536	0.89880	5e+19	-4e+19	4e+17
Fold 5	29.1226	552.3458	75.536	30.9105	-2e+19	3e+19	8e+18
Fold 6	25.8359	556.3033	31.6022	18.2133	3e+18	-6e+19	-2e+18
Fold 7	26.7793	570.2275	48.925	17.2415	-5e+18	2e+17	-3e+18
Fold 8	27.3392	571.2933	84.7903	27.1522	-2e+19	-6e+19	1e+19
Fold 9	28.1235	559.5587	68.452	35.5169	-5e+19	1+18	2e+19
Fold 10	28.9499	561.2288	102.713	22.3794	-6e+19	8e+19	5e+18

Table 10. Linear Regression Highest Bucket Feature Coefficients No School Data

The above tables give great insight into what features the model is utilizing most heavily during fitting. Unsurprisingly, in all of the price buckets, “Land Square Feet” and “Gross Square Feet” have positive coefficients in each fold. This is indicative that there is a positive correlation between the amount of land a property has and its price. The table shows that the highest price bucket has the greatest difference between the two. This suggests that people who are buying more expensive homes favor large home structures with a lot of interior area (Gross Square Feet), more than they favor a large yard (Land Square Feet). “Year Built” and “Year Sold” also show positive coefficients for each price bucket. Since these are quantitative features, this trend suggests that newer houses are more expensive and the housing market increases over time. “Year Sold” varies quite a bit for the most expensive housing bucket. While this may be due to any number of reasons, the most likely reason is due to a lack of data. The model had trouble grasping the effect of “Year Sold” on the price of the house.

"Borough", "Building Class at Time of Sale", and "Quarter Sold" stand out because the values are extremely sporadic across all folds in all of the price buckets. This may be the effect of having these two features as categorical variables. After one-hot-encoding, features are created in a smaller matrix. This matrix is then filled with binary values determining whether or not that instance belongs to that category. Another thought is that the data within those variables is unbalanced. This would make it difficult for the model to accurately capture the effects of these features on the price of the home. One model may include many instances of a particular Borough. This may mean the importance is boosted. However, the model in another fold may contain many instances of a different Borough. This will cause sporadic results that are difficult to interpret.

Linear Regression $0-$500,000 School Performance Data Integrated

	Land Square Ft.	**Gross Square Ft.**	**Year Built**	**Year Sold**	**Borough**	**Building Class at Time of Sale**	**Quarter Sold**	**School Performance**
Fold 1	7.2025	20.5002	349.9642	9.1826	3e+18	-2e+18	-6e+17	2637.85
Fold 2	7.8121	18.76899	365.253	8.6542	1e+18	4e+17	-1e+17	2700.583
Fold 3	8.0255	18.2538	340.6938	8.2836	3e+17	2e+18	-1e+17	2578.8676
Fold 4	7.9958	17.6157	398.9824	9.1233	3e+17	5e+17	-6e+18	2595.815
Fold 5	7.9229	18.349	398.1163	8.1122	-3e+18	1e+18	4e+17	2607.98753
Fold 6	7.40990	20.947	461.7773	7.3582	-3e+16	-6e+18	1e+18	2612.389
Fold 7	6.6393	21.1722	387.9933	9.2227	2e+18	3e+19	3e+17	2693.4296
Fold 8	7.3883	19.7457	315.635	8.7783	-2e+18	-2e+16	-2e+16	2597.8113
Fold 9	7.615	19.164	416.0115	8.8223	1e+18	-3e+18	3e+18	2658.50376
Fold 10	8.287	18.878	369.7679	8.5699	-1e+18	3e+18	-1e+18	2589.15645

Table 11. Linear Regression Lowest Bucket Feature Coefficients School Data Integrated

Linear Regression $500,000-$1m School Performance Data Integrated

	Land Square Ft.	Gross Square Ft.	Year Built	Year Sold	Borough	Building Class at Time of Sale	Quarter Sold	School Performance
Fold 1	22.0853	57.91127	154.6	1871.907	3e+18	-3e+18	1e+17	4021.4079
Fold 2	22.7811	56.9016	138.125	1646.2339	1e+17	2e+18	2e+17	3999.3316
Fold 3	23.9147	57.23528	85.545	1361.1066	-2e+17	1e+18	-1e+17	3938.4022
Fold 4	23.1974	53.565	117.742	1864.6533	3e+17	-1e+17	-2e+18	4120.6599
Fold 5	22.3776	56.0344	162.143	1407.8666	-3e+18	1e+17	-1e+17	3946.7371
Fold 6	21.9325	55.6238	97.4603	1597.663	-3e+18	2e+18	2e+18	4053.47622
Fold 7	21.75051	56.3011	74.008	1163.4855	2e+17	3e+19	-1e+17	3939.4245
Fold 8	22.7229	56.937	165.703	1984.5571	-1e+17	-2e+17	-1e+18	3992.90978
Fold 9	22.79502	57.9294	161.645	1472.555	1e+18	-2e+18	2e+18	4026.6301
Fold 10	22.37882	55.68927	121.075	1772.0524	-2e+18	3e+18	1e+18	3920.8774

Table 12. Linear Regression Middle Bucket Feature Coefficients School Data Integrated

Linear Regression $1m+ School Performance Data Integrated

	Land Square Ft.	Gross Square Ft.	Year Built	Year Sold	Borough	Building Class at Time of Sale	Quarter Sold	School Performance
Fold 1	18.68457	508.888	588.953	4597.904	-2e+18	-2e+17	2e+17	20105.439
Fold 2	19.0547	519.9903	486.793	5802.470	1e+18	2e+17	1e+17	17557.8199
Fold 3	22.128	509.618	398.3548	4046.3403	-3e+17	1e+17	1e+17	16777.459
Fold 4	20.0660	510.1964	490.233	4420.858	3e+18	-1e+18	-2e+17	17925.452
Fold 5	19.8933	516.0043	300.1231	5706.442	-2e+18	1e+17	-1e+18	18681.44
Fold 6	15.8787	489.727	528.509	4793.9143	-1e+18	2e+17	2e+19	17184.529
Fold 7	17.426	506.645	402.378	5595.535	1e+17	1e+17	-1e+18	17899.998
Fold 8	18.247	502.01	546.919	4572.94953	-1e+18	-1e+18	-2e+18	16415.8658
Fold 9	13.9622	486.6546	391.3331	4292.4957	1e+18	-2e+19	1e+18	13531.0628
Fold 10	14.523	498.9258	539.1138	5138.217	-2e+17	-2e+18	-1e+18	17905.38

Table 13. Linear Regression Highest Bucket Feature Coefficients School Data Integrated

These tables are essential to the underlying question we are asking. The school performance coefficient shows us the effect that school performance has on the price of the home. While the accuracy metrics showed us the overall effects of including school performance as a feature, we are now able to see how much this feature truly comes in to play. By referencing the tables above, it is clear that school performance has a very large, positive effect on the home sale price. For all three price buckets, the "School Performance" feature is by far the most influential in determining the home price. This indicates that homeowners value school quality very heavily in their decision on how much to pay for a home. However, while the data suggests these conclusions, there are many factors that indicate these results are not necessarily what they seem. "School Performance" was integrated from a combination of different locational factors. Once the score was included in the dataset, the other locational features were removed. Therefore, "School Performance" not only indicates how well a school performs, but also is a

representation of location. Location is often considered to be one of the most important features in determining home price. The higher the school performance, the more likely its surrounding area is to be affluent and higher priced in general. While these numbers suggest that school performance has a massive impact on home price, it may simply reflect the effects of location on home price. Further work must be done to include locational features in order to differentiate school rating influence on home price from location influence on home price.

It is important to note that these coefficient charts are only for the Linear Regression model. While it would be very insightful to see the feature importances of the Neural Network, it is extremely difficult. As mentioned before, Neural Networks act as a black box. Interpretability is sacrificed for performance accuracy. It is very challenging to track these coefficients between hidden layers and different nodes making this task difficult. Furthermore, the library used in this scenario (Sklearn) does not offer a method for displaying these coefficients. Therefore, in the interest of time and feasibility I was only able to analyze the feature importances resulting from the Linear Regression Model.

Future Changes

This project allowed for insight to be gathered through a process of data cleaning, data integration, model training, and model testing. While the interpretation of the results is very valuable, looking back, there are steps that could have been taken to increase the accuracy and robustness of the model.

The first difference would be the data used in the process. The data I utilized was from the NYC Department of Finance Rolling Sales Data. This provided information on land size, lot size, tax class, etc. As mentioned before, however, this data set did not include information on actual home features such as bedrooms, bathrooms, and air conditioning. Additionally, this dataset contained information on non-

residential properties like warehouses and office buildings. The ideal dataset for this project would have been residential only properties in NYC with information on home features

The next potential issue I identified had to do with the calculation of school performance score. My method utilized student testing scores to create a system of quantifying school performance. However, school quality is based on much more than just testing. Special programs and administrator quality are all aspects that can affect if a family will want to send their child to a particular school. Incorporating all of this data into a comprehensive score would create a better reflection of the actual school quality rather than just its testing results.

Finally, two is a relatively small amount for comparing model performance. The more models the better, as the goal is to find the model with the best prediction ability. However, in the interest of time, two was the amount that was feasible for this project. A deep learning application, while more complex and much more difficult to interpret, may have proved to have the best accuracy. Going forward with this research, testing more models will allow for better insight and conclusions to be drawn.

Bibliography

Bailey, Martin J., et al. “A Regression Method for Real Estate Price Index Construction.” *Journal of the American Statistical Association*, vol. 58, no. 304, 1963, pp. 933–942., doi:10.1080/01621459.1963.10480679.

Barr, Joseph R., et al. “Home Price Index: A Machine Learning Methodology.” *International Journal of Semantic Computing*, vol. 11, no. 01, Mar. 2017, pp. 111–133., doi:10.1142/s1793351x17500015.

Diamond, Phil, and Hideo Tanaka. “Fuzzy Regression Analysis.” *The Handbooks of Fuzzy Sets Series Fuzzy Sets in Decision Analysis, Operations Research and Statistics*, 1998, pp. 349–387., doi:10.1007/978-1-4615-5645-9_11.

“NYC Resources: Agencies: City of New York.” *NYC Resources | Agencies | City of New York*, www1.nyc.gov/nyc-resources/agencies.page.

Park, Byeonghwa, and Jae Kwon Bae. “Using Machine Learning Algorithms for Housing Price Prediction: The Case of Fairfax County, Virginia Housing Data.” *Expert Systems with Applications*, vol. 42, no. 6, 15 Apr. 2015, pp. 2849–3296.

Plakandaras, Vasilios, et al. “Forecasting the U.S. Real House Price Index .” *Economic Modelling*, vol. 45, Feb. 2015, pp. 1–290.

Rosaen, Karl. “K-Fold Cross-Validation.” *Karlrosaen.com*, 20 June 2016, karlrosaen.com/ml/learning-log/2016-06-20/.

Salcedo-Bernal, A., et al. “Clinical Data Analysis: An Opportunity to Compare Machine Learning Methods.” *Procedia Computer Science*, vol. 100, 6 Oct. 2016, pp. 731–738., doi:10.1016/j.procs.2016.09.218.

Walks, Alan. “Gated Communities, Neighbourhood Selection and Segregation: the Residential Preferences and Demographics of Gated Community Residents in Canada.” *Town Planning Review*, vol. 85, no. 1, 2014, pp. 39–66., doi:10.3828/tpr.2014.5.

Yiu, Tony. “Neural Network with Two Hidden Layers.” *Towards Data Science*, 2 June 2019, towardsdatascience.com/understanding-neural-networks-19020b758230.

Academic Vita

William Joseph Gault

EDUCATION

Pennsylvania State University — **University Park, PA**
Schreyer Honors College — **Class of 2020**

- Bachelor of Science in Applied Data Sciences
- College of Information Sciences and Technology

WORK EXPERIENCE

Goldman Sachs Group, Inc. – *Data Analyst Intern* — May 2019 – August 2019

- Create Python Test Environment for Department Application Suite
- Use Machine Learning to Classify Employee Application Feedback
- Write Algorithm for analyzing PDF documents following OCR
- Python

Ford Motor Company – *Software Engineering Intern* — Summer 2018

- Designed, built, and implemented financial tracking application for Project Managers
- Worked with teams in India to coordinate efforts in manufacturing operations
- Python, VBA, and Java Development

Independent Technology Consulting – *Self Employment, Skillet.ai* — March 2018- May 2020

- Work with startup companies to help develop their products
- Analyze and visualize large cyber-attack data on a global scale
- Python, D3.js, Angular

RELEVANT EXPERIENCES

Penn State College of IST Alternative Spring Break — March 2017

- Selected to participate in week-long program in Seattle, WA
- Spoke with and toured large global companies such as Amazon, Microsoft, Google, Boeing, and Kaiser Permanente
- Gained insight to company cultures and life in the Pacific North West

Confucius Institute Chinese Summer Bridge Program — May 2015

- Studied abroad in Beijing and Changchun, China becoming immersed in the Chinese culture and language
- Obtained proficiency in Mandarin Chinese speaking, reading, and writing
- Home-stay with a Chinese family learning their everyday routines and lifestyle

West Point Summer Leaders Experience — May 2015

- Developed leadership skills through military based courses at the United States Military Academy
- Participated in Electrical Engineering, Computer Science, and Military Robotics seminars
- Spoke with high ranking officers gaining advice and learning about their experiences

ACTIVITIES/LEADERSHIP

Nittany Data Labs – *Co-Founder, Director of Projects, Treasurer* — January 2017 – May 2020

- Oversee and advise all corporate and capstone projects being completed by our project teams.
- Complete various projects covering data visualization, predictive modeling, and business intelligence
- Work as liaison between corporate sponsors, executive board, and club members
- Help students build their knowledge of data science concepts through our training program
- Travel to speak with companies and gather new machine learning and artificial intelligence projects
- Work to gain club funding while managing and delegating already existing funds

Human Trafficking Research - *Machine Learning* — August 2017 – September 2019

- Web scrape open source websites to gather data on human trafficking advertisements
- Work with predictive models to analyze trends in Pennsylvania human trafficking data
- Write natural language processing script to predict the sentiment and classify advertisements
- Use multiple other machine learning methods and algorithms to build a classification model related to human trafficking
- Work with Pennsylvania State Police and FBI officials while carrying out research

Member of Acacia Fraternity - *Consigliere* — November 2016 – May 2020

- Work as the president's right-hand-man in advising and running the chapter
- Devote time to the community in both service and philanthropic efforts including Dance Marathon
- Second highest overall GPA of all Penn State fraternities