House Price Prediction

and

Collinearity with Stock Market Indexes

Fundamentals of Analytics and Discovery Informatics

16:137:50:01

Clarence Li

Holly Sickler

Zachary Sisco

Introduction

The goal of this study is to develop a model that both produces accurate home valuations as well as understands a home value's relationship with market conditions. This model would provide both buyers and sellers with the tools necessary to not only strategically price their properties, but also know when to act on their properties. The importance of understanding home value now and future conditions in terms of helping to protect financial investments of both buyers and sellers can be demonstrated by the fact that, "by 2009, homes built between 2005 and 2007 — the height of the housing boom — were more likely to have negative equity than units built before or after that period" ("Negative Equity in the United States: HUD USER").

This project also stresses the importance of feature selection and feature engineering during the data preprocessing stage. There are so many different factors that influence the price of a property as well as its supply and demand, so it is important to understand and evaluate each data attribute before running predictive models on them. There is also a possibility that some features are weighted more heavily than others which would give rise to skewness within the data, so scaling and normalization techniques are required to correct such skewness to improve the statistical integrity of the data used.

This project seeks to analyze various data sources which would be divided into two parts: house price prediction and the housing market's collinearity with stock market indexes. For the first part of the project, real estate sales data from Boulder, Colorado (Boulder County Open Data, 2020) from 2016 to 2019 was collected, resulting in a dataset containing roughly 25,000 instances and 37 total attributes. In the second portion, historical house prices were collected from a variety of cities provided by Zillow (Zillow, 2020) joined in a dataset with 297 instances and 915 total attributes. Historical stock market data was collected from a variety of market indexes provided by the Wall Street Journal (Wall Street Journal, 2020), forming a dataset with 7,759 instances and 19 total attributes. Both the Zillow and Wall Street Journal datasets were merged in order to find any possible relationships between the two.

In order to execute this project's approach to predictive modeling, supervised learning methods were applied that include a variety of regression and classification techniques. The models used for regression were a Dummy Regressor as the baseline, Multiple Linear Regression, Ridge Regression, Lasso Regression, Regression Tree, Random Forest Regression, and Support Vector Regression. The classification models used in this project were K-Nearest Neighbor and Support Vector Classifier. Also, a few forecasting techniques were implemented such as AutoRegression and Prophet package algorithms. All of these techniques will be described in greater detail later in the paper.

Background & Motivation

Buying a house is usually the most monumental purchase in a person's life. It takes years of hard work to build enough credit and savings to be qualified for a mortgage. When taking such a significant step you want to be sure that you are paying the right price, and that you are buying at the right time. The same is true for when you decide to sell a house: price and timing are paramount. The ability to strategically buy and sell houses is of great interest to both real estate brokers and homeowners, as well as a new wave of tech-heavy real estate startups such as Compass. With the introduction of Big Data to the real estate market, machine learning has become commonplace to better manage the buying and selling process.

Using predictive models for determining property sale prices in cities like Boulder, Colorado is still a challenging task even though there are many literature review papers on the subject matter. When attempting to price a house, these algorithms typically consider relevant house attributes. As described by M. Jain et. al., there are three components that impact property values: states of being, thought, and territory (M. Jain et. al., 2020). States of being are constraints within a house such as the range of the house, the amount of rooms, the availability of a kitchen or parking space, and the age of the house. Thought is a conceptual view provided by the architects who can influence potential buyers via the style, type, and condition of the house. Territory is basically the costs of the land and surrounding areas. With so many different types of parameters that factor into the price of a house, each parameter must be taken into consideration when selecting the right house attributes for running model predictions. But as mentioned previously, factors larger than house attributes play a role when deciding to buy/sell your house.

The timing of a property purchase or sale plays a critical role in the valuation of the property, and many basic housing valuation algorithms do not take market conditions into account. Based on literature review, the correlation between financial market conditions and housing market conditions is unknown. We aim to uncover the following concept: the correlation between real estate property prices and financial market. Gaining this knowledge has the potential to greatly impact the home buying and selling processes by providing end-users with the ability to better time their purchases or sales.

Literature Review

In order to prepare for this analysis, a thorough review of relevant literature was undertaken. Of the literature reviewed, titles include "Housing Price Prediction Via Improved Machine Learning Techniques" (Q.Truong, et. al., 2020), "Machine Learning based Predicting House Prices using Regression Techniques" (J. Manasa, et. al, 2020), and "House Prices Prediction with Machine Learning Algorithms" (C. Fan, et a;, 2018). Along with these articles, four additional pieces were considered: "Predicting Sales Prices of Houses Using Regression Methods of Machine Learning" (P.A. Viktorovich, et. al, 2018), "Real estate markets and the macroeconomy: A dynamic coherence framework" (R. Bouchouicha and Z.Ftiti, 2012), "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data" (B. Park, & J.K. Bae, 2015), and "House Price Prediction Using Machine Learning and Neural Networks" (A. Varma et. al 2018). The first three articles listed will now be discussed in greater detail.

The first article, "Housing Price Prediction Via Improved Machine Learning Techniques", was a research study completed by a team of researchers at Texas Christian University in 2020. The use of machine learning algorithms to better predict pricing has become common, but the researchers note that most predictions involve comparing the performance of one model to another. To improve this process, the researchers propose a Stacked Generalization approach which combines multiple algorithms to optimize the predicted value. With data in hand, the researchers preprocessed the data through removal, feature engineering, and outlier removal. Before moving onto the model selection phase, the research team made sure to complete some exploratory analysis on their data to find implicit trends and patterns. After the exploratory analysis, the researchers moved onto testing different algorithms using the sklearn package and using the Root Mean Squared Logarithmic Error (RMSLE) function to evaluate their efficacies. The models that the researchers implemented were Random Forest, Extreme Gradient Boosting (XGB), Light Gradient Boosting Machine (LGBM), Hybrid Regression (using 33.33% Random Forest, 33.33% XGB, and 33.33% LGBM), and Stacked Generalization, where the prediction of Random Forest and LGBM were used as the features for XGB. It was found that Random Forest had the best performance on the training set, while Stacked Generalization performed the best on the test set. Random Forest may result in the lowest training error but it's prone to overfitting and time complexity. XGB and LGBM have decent accuracy, but their time complexities are the best. Due to Hybrid Regression's generalization, it performs better than the previous three models but also contains time complexities due to Random Forest's inclusion. Lastly, the Stacked Generalization has the best accuracy results, but it also has the worst time complexity. In conclusion, the researchers determined that each of these models have their own pros and cons and the choice of either model should depend on what the end goal is.

In a similar approach, another study attempts to analyze housing price prediction with a focus on various regression techniques (J. Manasa, et. al., 2020). The dataset used in this study

was taken from a publicly-available hackathon platform that described the 2016 housing market in the city of Bengaluru. Rather than relying on the price values alone, forecasting property sales in cities like Bengaluru depend on a number of interdependent factors that could affect the price itself such as the amenities within the house, the location of the house, and the surrounding environments. The Bengaluru dataset contained around 1480 records and just 9 features for training and test sets that describe the property such as price, size, availability, location, and other attributes. The learning method of choice utilized in this study were comparative analysis of predictive regression models consisting of Ordinary Least Squares (OLS) model, Lasso and Ridge regression models, Support Vector Regression (SVR) model, and XGBoost regression model. The authors also outlined their general data pre-processing procedures that we will mimic in our project. The first step in the data cleaning was to convert any categorical features into numeric, nominal values with a label encoder (or a one-hot encoder) in order to fit a linear regression. The second step was to locate and impute missing values with appropriate values (for example, null numeric values cleaned with median values and null categorical values cleaned with mode values (P. A. Viktorovich, et. al, 2018)). Additionally, unnecessary and uncorrelated column features that did not add value to the regression model were dropped. The third step involved correcting the shape of the presented data to closely match a Gaussian distribution of the data. Finally, the cleaned data were split into training sets and testing sets. After the data processing and model implementation, the performance of each model was measured using the coefficient of determination metric (r^2), Root Means Square Error (RMSE), and Root Mean Squared Logarithmic Error (RMSLE). The results of this study suggested the need for a better characterization of housing price data with a lot more features than the ones analyzed within the study. They also noted that data collected from an urban city should not be applicable to a rural community due to uneven distribution of feature prices where rural areas are generally much higher than urban areas.

In the third article, “House Prices Prediction with Machine Learning Algorithms" (C. Fan, et a;, 2018), researchers built a model to predict home prices in Ames, Iowa by evaluating several different models in a manner like those identified in the previous articles, however these researchers ended up using an ensemble of machine learning models. The data was split into a training set and a testing set and ultimately the RMSLE was used to evaluate the model. This was the selected evaluation method to ensure that it was not disproportionately impacted by the homes in the dataset with a larger value. To begin with understanding and cleaning the data, visualization was used and assisted with dropping any outliers which were not representative of the distribution displayed in the data. The method of dropping rows containing missing data was not used to handle missing values in this dataset as the risk of potentially losing data that could be very important exists, especially in this case given the generally large number of missing values. Instead, three types of interpolation were used with the goal of establishing an accurate model despite the missing values. However, it is noted within the article that interpolation tends to be a time-consuming strategy for solving the problem of missing values. The feature

engineering used by the researchers included one-hot encoding for categorical data so that it can be used in models requiring numeric values, as well as feature selection using several methods such as RFE. Lasso linear regression, Ridge linear regression, SVR, Random Forest, and Extreme Gradient Boosting (XGB) models were considered by the researchers and both 10-fold cross validation and average MSE were used to determine performance of each of these models. After testing these several models, the ones with the lowest MSE (XGB, Lasso, and Ridge) were then turned into a prediction model containing a factor for consideration of different weights for each of the models used to create the prediction model. To finalize the combination model, the correct weights for each component of the model were determined using the evaluation metric of RMSLE. Effectively the weight of the Ridge linear regression was determined best as it became closer to 0, leaving Lasso linear regression and XGB as the combination model. The researchers noted that the use of combination methods produced a more accurate result however future work to improve the model could include the use of non-linear combination forecasting (C. Fan, et a;, 2018).

Approach

Obtain and Clean the Data

The data necessary for this project was collected from a variety of sources. For the price prediction portion of this project, it was imperative to collect data that was both robust and relevant to the home buying process. Initially, a robust dataset from NYC Open Data that contained roughly 340,000 instances was collected. However, after the initial analysis of the data, it was found that the features of this dataset were largely irrelevant for the purposes of predicting price. Once this dataset was dropped, open data for other US cities were examined and ultimately Boulder County, Colorado was chosen. The Boulder County dataset contains over 25,000 instances with 37 features, both numerical and categorical, that greatly describe the three real estate valuation components discussed earlier in this report. For the state of being component, this dataset has attributes that count the number of bedrooms and bathrooms, the amount of square footage, and an attribute that allows us to derive the age of the house. For the thought component, this dataset has an attribute that describes the architectural style of the real estate. Lastly, regarding the territory component, the dataset contains features that describe the monetary value of the building as well as the land that the building is on. Due to the large size of the dataset as well as its relevance to real estate valuation, this data was deemed suitable for further analysis.

The data collection for the second portion of the model was broken into two parts: historical stock market data and historical home value data. In order to obtain historical data for the stock market, the Market Data webpage of the Wall Street Journal was visited where historical data for most stocks and indices was available. In order to get a big-picture look at the stock market, some of its largest indices were chosen for analysis, including the Dow Jones Industrial Average, the S&P 500, the Nasdaq Composite, and the CBOE Volatility Index. The first three indices were chosen for their ability to largely describe overall market conditions, while the latter index was chosen with the thought of analyzing the effect of stock market volatility on the housing market. The dataset contains historical open, high, low, and close values for the indices dating from 1990 to mid-October 2020. The historical home value data was collected from Zillow. This dataset contains seasonally adjusted, smoothed measures of typical home values in various regions across the US dating back to 1996. This data describes home values that fall within the 35th to 65th percentile range for a given region, thereby providing a well-rounded representation of the greater housing market.

Once the data was collected, it was time to start preprocessing. For the price prediction model, the first step in cleaning the data was gathering the multiple datasets together into one dataframe. Boulder County broke its real estate sales data up by year of sale, and sales data for years 2016-2019 was collected. These four datasets were assigned to respective data frames and then these data frames were appended to create the master dataframe (Figure 1, Appendix). As

mentioned above this data frame consisted of 37 features, many of which were able to be dropped due to their irrelevance to the price prediction task. These dropped features include the following: Account Number, Parcel Number, Property Address, Multiple Buildings, Account Type, Building1 Description, Owner Name, Care Of, Mailing Address 1 and 2, Grantor 1, Grantee 1, Reception Number, Subname, and Extra Feature Value. These features were eliminated as they were either largely null or provided no tangible value to the price of the real estate. The next step included eliminating any instances whose State value was not Colorado as these instances were erroneous. These initial steps helped to greatly reduce the dimensionality of our dataset as well as eliminate over one thousand useless instances. Once this was completed, the next step was to scan the dataset for null values. Only one null value was found in the Location City feature, and this null value was replaced with the correct city according to other instances sharing the same Neighborhood Code value. This cleaning would prove to be futile, though, as the Location City feature was eventually dropped along with the City and Neighborhood Code features. With no nulls remaining in the dataset, the Building1 Design feature was then analyzed. It quickly became apparent that this feature contained many redundant values, so the feature was cleaned so as to eliminate redundancy through consolidation and grammar correction. Similarly, the Zip Code feature needed cleaning in order to remove unnecessary characters. The next step of cleaning involved converting the Sale Price, Land Value, and Building Value features from string type to integer type. This step was crucial as string values are not able to be used for regression analysis, and these features were of utmost importance. The next step was to look for ways to further reduce the dimensionality of our dataset. To do this, the various features describing the square footage of the property were summed together to create a new Total SQFT feature. This feature was then utilized to drop any instances from the data that contained square footage equal to 0. Additionally, the features counting the number of bathrooms were summed to create a Total Baths feature. Lastly, a House Age attribute was created by subtracting the Year Built feature from 2020. The creation of these new attributes allowed for the removal of 11 now-useless attributes, dropping our feature count from 37 to 10 in total.

The historical stock market and home values data for the second portion of the project required far less cleaning. The file containing the stock price data contained a few null columns from the source file, so these were dropped. It was also decided that only market open and close values were of importance so the high and low values were dropped from the data. With the stock data cleaned, the next step was to move onto cleaning the historical Zillow data. This dataset contained historical values for over 900 cities across the US, but only the 4 largest cities by population were kept, with Boulder as an add-on, leaving the dataset with historical data for New York, Los Angeles, Chicago, Dallas, and Boulder. Boulder was the only city that required cleaning as a large portion of the city's values were null. These values were imputed with the mean home value. With these two datasets cleaned, it was time to merge the two together to create a master data frame containing historical data for both stock prices and home values. This

merge was completed using an inner join, utilizing the Date column as the common feature. The stock market data contained dates that ran from Monday to Friday, excluding holidays, while the Zillow dataset only contained month end dates. After the merge, the master dataframe contained 210 instances and the Date attribute was able to be removed. Due to the wide range in values between the features, it was apparent that normalization was required. This data transformation was completed using the MinMaxScaler available from sklearn which converted all values to a value between 0 and 1. This transformation would prevent drastically unequal weights from being assigned to features during future regression analysis.

Evaluation

Before beginning to run regression analyses on the housing data, exploratory data analysis (EDA) was undertaken. The importance of exploring the data before tackling analyses cannot be understated; this process can greatly facilitate the discovery of noise and outliers in the data, as well as erroneous values that were missed in the initial round of data cleaning. To begin the evaluation of the Boulder data, a simple description of the dataframe was executed as can be seen in Figure 1 below.

```
bld_df.describe()
```

	BEDROOMS	ABOVE_GROUND_SQFT	SALE_PRICE	LAND_VALUE	BLDG_VALUE	TOTAL_SQFT	TOTAL_BATHS	HOUSE_AGE_YRS
count	23569.000000	23569.000000	2.356900e+04	2.356900e+04	2.356900e+04	23569.000000	23569.000000	23569.000000
mean	3.222156	1789.008401	6.035427e+05	1.644334e+05	3.422690e+05	3014.079893	2.770376	33.418304
std	2.645822	1714.755401	5.360944e+05	2.317764e+05	3.018887e+05	2333.707392	2.655063	24.886207
min	0.000000	1.000000	1.760000e+02	0.000000e+00	0.000000e+00	1.000000	0.000000	1.000000
25%	2.000000	1144.000000	3.575000e+05	5.300000e+04	2.090000e+05	1830.000000	2.000000	16.000000
50%	3.000000	1588.000000	4.925000e+05	9.900000e+04	2.988000e+05	2710.000000	3.000000	29.000000
75%	4.000000	2235.000000	6.950000e+05	1.950000e+05	4.114950e+05	3904.000000	3.000000	47.000000
max	370.000000	230384.000000	4.570000e+07	1.767540e+07	2.651320e+07	250184.000000	370.000000	146.000000

Figure 1: Boulder Dataframe Description

This description revealed two things: it seems that there are some values that still need to be removed, such as instances with 0 bathrooms, and there is a very wide range of values between the different features. These issues will have to be corrected before completing analyses. To further evaluate the Boulder data, plots showing the distribution of the feature values were created, as seen in Figure 2w.

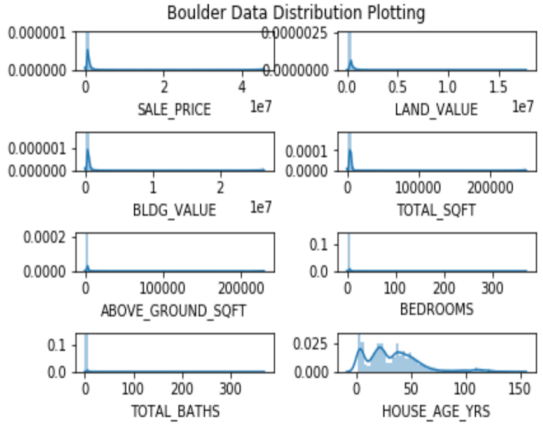

Figure 2: Boulder Data Distribution

The apparent positive skew on all of the numerical features exposed another issue with the data that will need to be addressed before running certain regressions, such as Lasso and SVR, as well as classification algorithms like KNN. Now that the general outlay of the data is understood, the next step was to discover any trends in the data such as feature correlation, which is displayed through a heatmap in Figure 3 below.

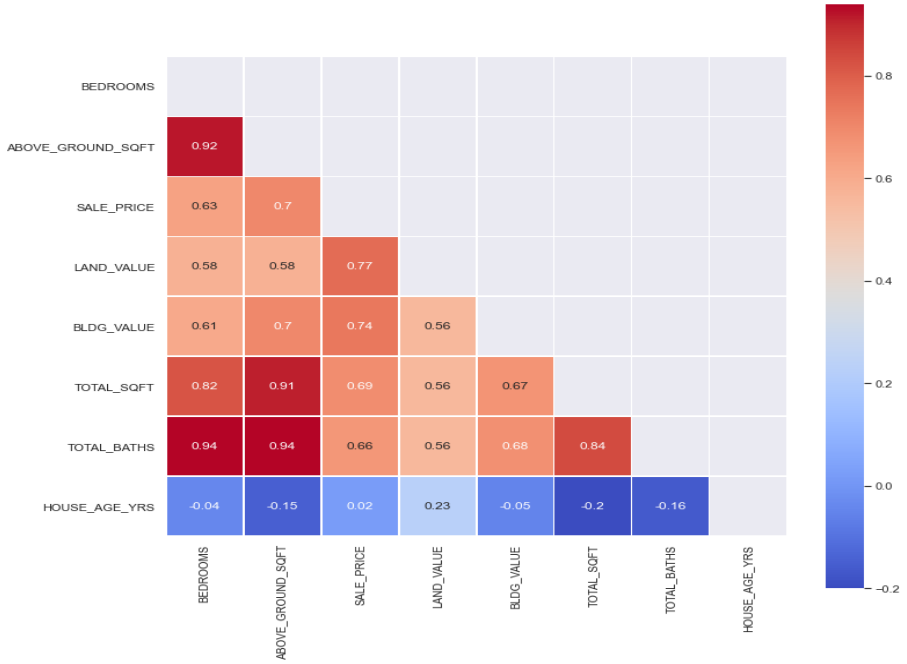

Figure 3: Boulder Dataframe Feature Correlation

This heat map revealed some important information by showing that certain features, particularly Land Value, Above Ground SQFT, and Building Value are most highly correlated to our target variable of Sale Price. The next logical step was to begin the process of outlier removal, which began by looking at Sale Price values. The scatter plot of Sale Price vs House Age can be seen in Figure 4 below, along with outliers that could be removed. It was easy to miss at first, but a closer look at the scatter plot reveals a single sale price of over $40,000,000. This instance is greatly removed from the majority of the instances so it was eliminated from the data, resulting in the scatter plot shown in Figure 5.

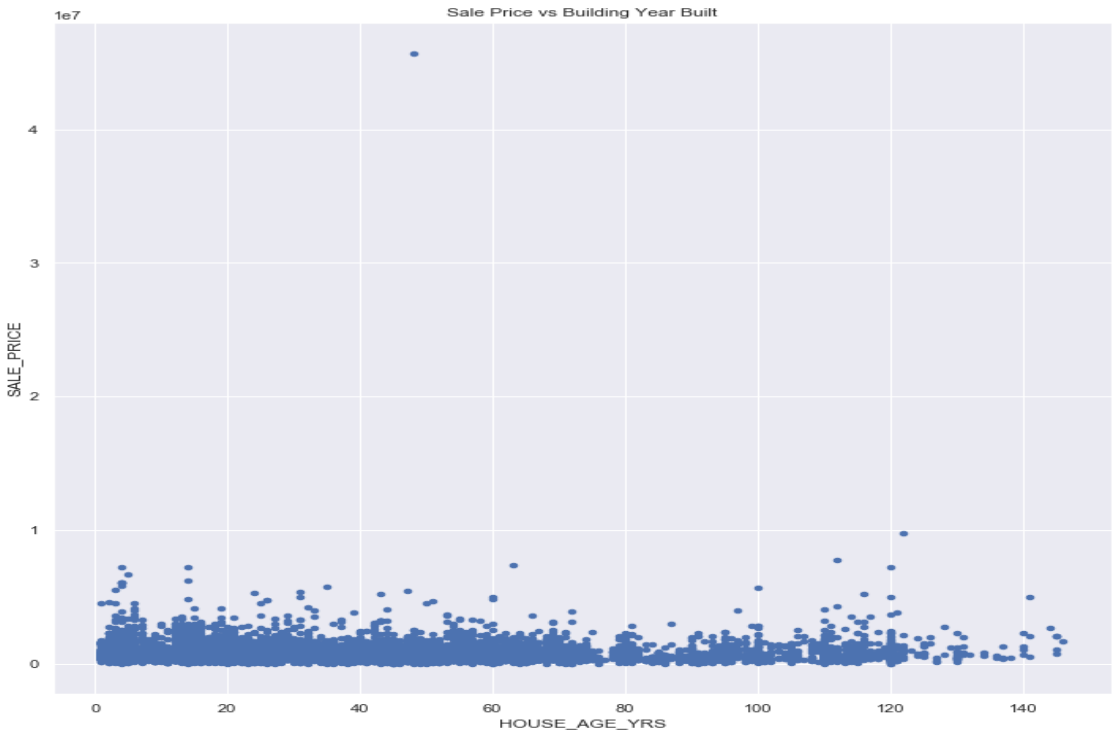

Figure 4: Sale Price vs House Age Scatter

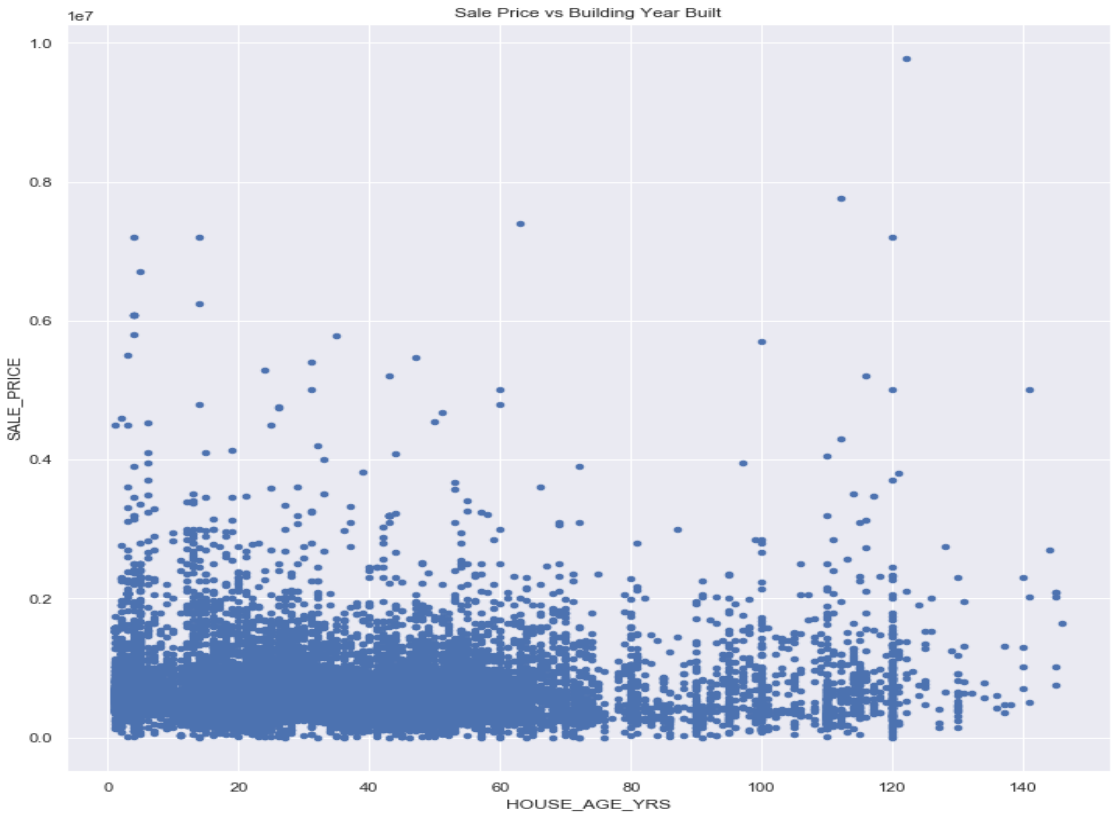

Figure 5: Sale Price vs House Age Scatter, Outlier Removed

This scatter plot revealed that even with extreme outliers removed, there are still plenty of other outliers and noise lurking in the dataset. From this point, additional data cleaning and outlier removal attempts were made. Data cleaning attempts included applying a logarithmic function to the Sale Price feature in order to create a more normal distribution, as well as One-Hot Encoding (OHE) of categorical values. Additionally, the use of a MinMaxScaler normalization method was applied to the data in an attempt to equalize weights among features. Some of the outlier removal attempts included excluding instances with Sale Price values over the 75th percentile mark of $1,200,000, excluding instances with Sale Price values outside of the interquartile range, and also removing outliers from this IQR dataset. Once these edits were made, linear regressions were performed on the various datasets to test the efficacy of cleaning. Using the coefficient of determination, R^2, as the metric it was found that none of cleaning methods above outperformed the regression run on the native dataset, which was considered the baseline at this point. Knowing this, it was determined that a more thorough and effective data cleaning was necessary.

The dataset excluding the $40,000,000 instance was used as a starting point for this next phase of cleaning. The first issue to be dealt with was the large amount of categorical values that were being encoded in our dataset. Through the OHE of the Building Design and Zip Code features, the dataset's feature count increased from 10 to over 200, representing a drastic increase in dimensionality. To reduce this dimensionality, the Building Design values were cleaned to group the values into one of three categories (Single Family Residence, Attached Single Family Residence, and Condo) rather than the original twenty. Once this was completed, a deeper look at the numerical features was done in order to remove erroneous values and outliers. First, instances listed with 0 bathrooms and 0 bedrooms were removed, as residential units are required to have at least 1 bathroom and 0 bedroom studios had already previously been removed. It was also found that instances with over 6 beds and 6 baths were an insignificant portion of the dataset, so they were removed to avoid high variation. To remove outliers from the Above Ground SQFT feature, instances with less than 400 SQFT and more than 8000 SQFT were eliminated as they insignificantly contributed to the dataset. The same tactic was applied to the Sale Price feature, eliminating instances with less than $80,000 price and more than $2,000,000 price. Furthermore, instances with Building Value and Land Value prices equal to 0 were removed if the instances did not fall into the 'Condo' category of house. Generally, a residential home will have both building and land value, however depending on the condo association rules, a condo owner may only own the inside of their unit leaving them with no building or land value. Additionally, outliers with Building Values of over $800,000 and Land Values over $1,000,000 were removed. Lastly, to further reduce the dimensionality of our dataset, the Zip Code attribute was dropped as the OHE of this feature would result in roughly 200 features total. Given that the dataset being used is from one single region of one state, it was reasoned that dropping the Zip Code attribute would not have a severe impact. The newly cleaned dataset was normalized with MinMaxScaler, and then explored to see the layout of the data. The purpose of the

MinMaxScaler normalization is to rescale all the data values into a predetermined range between 0 and 1. By doing so, all the features within the DataFrame will now have roughly equal weights when implemented in our machine learning models. The pitfall of this normalization technique is that it is sensitive to outliers and does not correct skewness as desired. As can be seen in Figure 6 below, the cleaned data still had a slightly positive skew in its features. To combat this skew, the dataset was then transformed using a Yeo-Johnson Power Transformer method, the results of which can be seen in Figure 7.

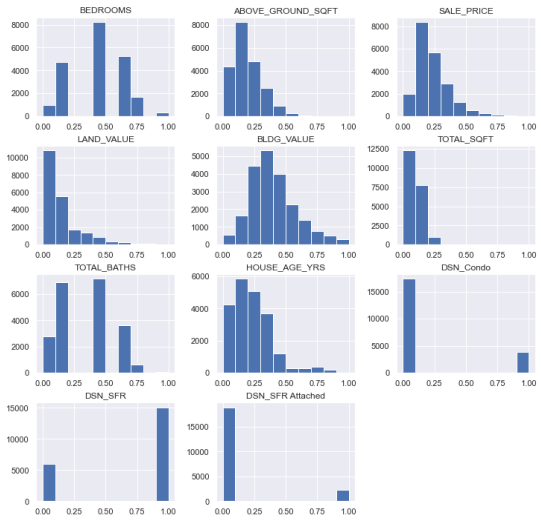

Figure 6: Data Distribution of Cleaned Dataset

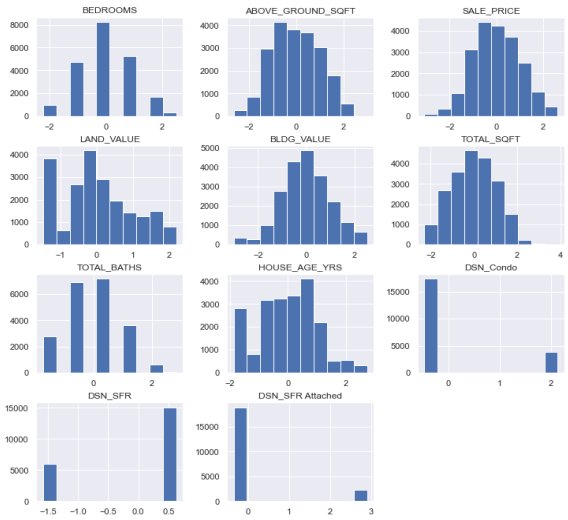

Figure 7: Data Distribution After Yeo-Johnson PowerTransform

The Power Transformer normalization technique helps to correct the skewness of the data by forcing a Gaussian-like distribution at the 0 mean, unit variance. This PowerTransformer technique will become necessary when advanced models such as the ones listed earlier are run.

For the second portion of the project, due to the cleanliness of the Zillow and stock data, far less additional cleaning was necessary. Once the two datasets were properly merged, the only cleaning that was completed was dropping the ‘Open’ values for the stock indices. Because the ‘Open’ and ‘Close’ values were indistinguishable from each other on a time series graph, the ‘Open’ was deemed unnecessary. With this cleaning complete, the next step was to explore the data to uncover any patterns or relationships between the stock and housing values. As can be seen in Figure 8, there does not appear to be any strong relationship between housing values and stock market values. Although the markets both have a general upward trend, none of the housing market lines are strongly tied to any index line. The housing values seem to rise at around the same rate as the financial markets. One interesting time point is the period leading up to the 2008/2009 financial crisis and housing market crash. The CBOE VIX index has a strong peak, followed by a large decrease in housing values.

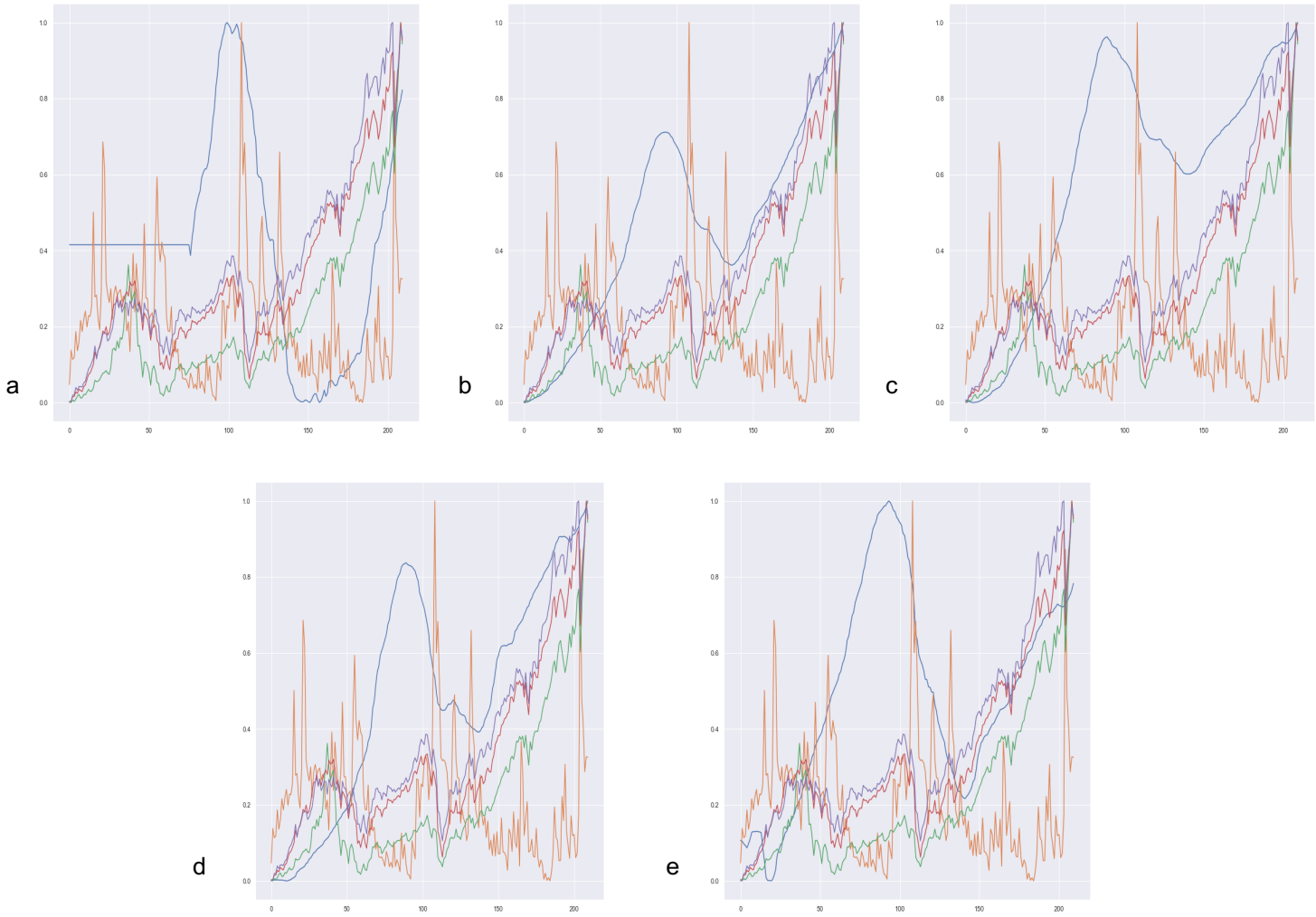

Figure 8: Cities (blue line) vs Stock Indices; a: Boulder, b: NYC, c: LA, d: Chicago, e: Dallas-Fort Worth

With this exploration complete, the correlation matrix between the two markets was then analyzed, as can be seen in Figure 9 below. Through the correlation matrix, it became clear that the strongest correlations are between the DOW Jones Industrial Average and the New York City and Chicago housing markets, with a correlation coefficient of 0.8 and 0.77, respectively. With this knowledge, the DOW index was focused on for the remainder of the model.

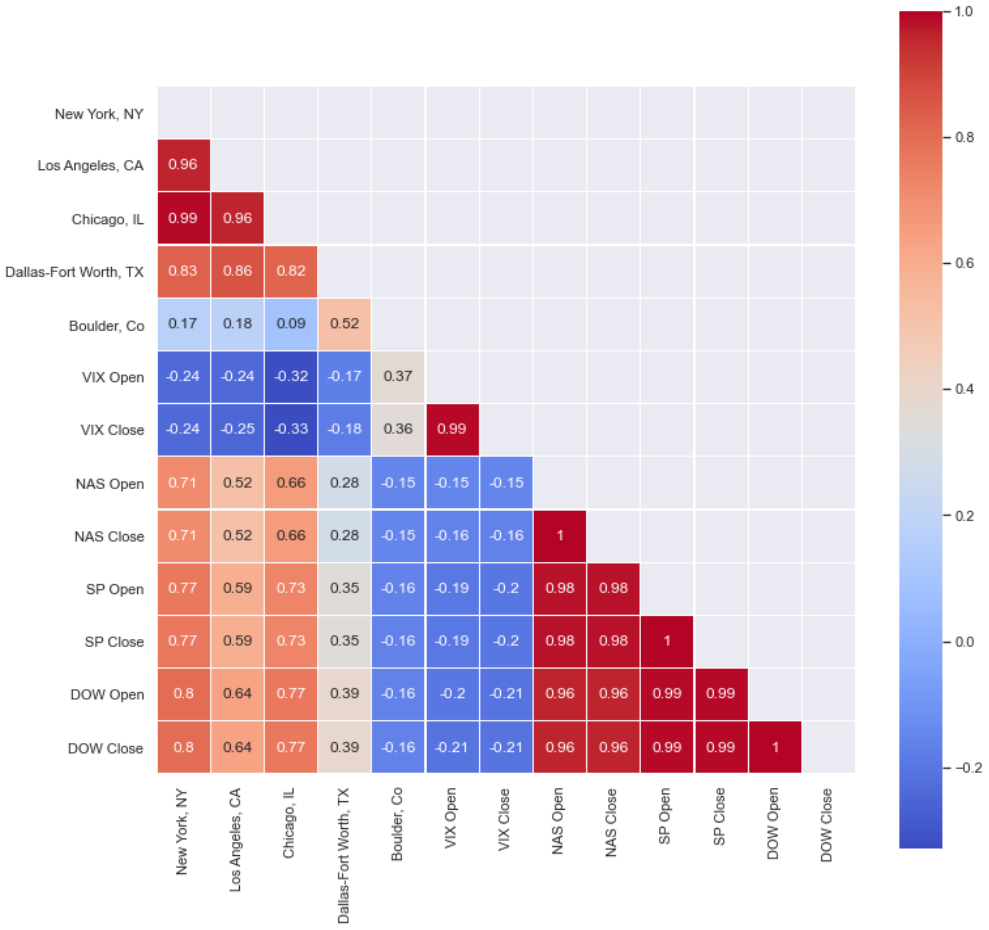

Figure 9: Correlation Matrix Between Housing and Financial Markets

Model Implementation

After all of the final data cleaning and manipulation for an equally-weighted Gaussian distribution of the data features and target, the machine learning models can finally be implemented and evaluated for performance comparison. To begin, the dataset was split into separate X and Y variables with X being all the feature attributes and Y being only the target variable, Sale Price. For the train/test split, instead of simply using the general 80-20 method, a 10-fold cross validation method was used as well for better train/test splitting. This is because the cross-validation method allows for training multiple train/test splits instead of just one time in the 80-20 method. For example, when the 80-20 train/test split method was utilized previously, there was high variation within the scoring results. Sometimes a model will perform amazingly well while other times (when re-running the same model), the model performed horribly. As for the 10-fold cross validation method, each fold (or split) is iteratively tested among the other nine folds, and the results are generally consistent throughout.

After establishing the train/test splitting method, a baseline linear regression model was implemented with a Dummy Regressor module from the sklearn library. The Dummy Regressor is a generic regression baseline for use to compare against the real regression models. For the Dummy baseline model, the strategy parameter was set to "mean" which predicts the mean of the training set, and was compared against a multiple linear regression model. Subsequent regression models such as Ridge Regression, Lasso Regression, Regression Tree, Support Vector Regression, and Random Forest Regressor were also used for performance metric comparisons. These regression models were ultimately chosen due to the fact that they are all commonly seen in many machine learning research papers. For the ridge and lasso models, the alpha parameters were selected based upon the one that performs the best in terms of accuracy. An alpha parameter test list was created and iterated through to check for the most optimized value. The same concept was applied into the tree model for the max depth parameter and the support vector model for the epsilon parameter. After all the regression models were completed, their metrics were compared against each other to see which models would be the best predictor of the target variable Sale Price.

Aside from the regression modeling, a few additional algorithms were implemented to test for possible classification models. At the very beginning of the home buying or selling process, an exact price for buying or selling isn't always in mind. Rather, an ideal price range is usually determined. The goal of implementing classification algorithms is to give the end-user an idea of what price range their property falls under. Not only is this implementation a unique approach to the house price prediction problem, but it has the potential to make a stressful process a little bit less complex. In this model, a new column titled "House Class" was added to the cleaned Boulder dataframe described earlier, and properties were classified into one of six classes. The class designation was determined by the sale price of the property. Because the majority of the properties were under $1,000,000, classes 1-5 were designated to houses with sale prices under that mark, starting at our $80,000 minimum and increasing by roughly $184,000 for each class. Class 6 was designated for all properties with sale prices over the $1,000,000 mark. When beginning to implement the classification algorithms, a Dummy Classifier was used as the baseline, with the Dummy predicting classes uniformly at random. With the baseline covered, the next classification algorithm to be used was the K-Nearest Neighbors (KNN) algorithm with the neighbor count set equal to 5. This algorithm was chosen for a couple of reasons. Firstly, the algorithm is relatively easy to explain to non-technical stakeholders. Secondly, the large size of the Boulder dataset provided the algorithm with plenty of neighbors to check over, thereby allowing the algorithm to hopefully make precise decisions. After KNN was implemented, the next step was to implement the Support Vector Classifier (SVC) algorithm with all hyperparameters set to their default state. The SVC model was chosen for its popularity among recommendation systems as well as its reputation for being wildly successful at classification tasks.

For the second portion of this study, attribute and target variables were defined with the attribute being the DOW and for each city in the cleaned dataset the home values were each considered the target. A standard test train split was established using an 80% training size and 20% test size. Then a baseline linear regression model was implemented for each of the cities with a Dummy Regressor as well as a standard multiple linear regression. The linear regression was completed in large part to further visualize the relationship between home values and the DOW.

Following that, two forecasting models were executed for the stock data, specifically the DOW attribute. For implementation of these models the merged stock and Zillow dataset was not used, rather the cleaned stock dataset was used as a result of the size of the datasets. The merged dataset has significantly less stock data available than the stock dataset alone due to the merge on date as previously mentioned. The first forecasting model executed was AutoRegression from statsmodels.tsa.ar_model. The additional forecasting model implemented was the Prophet model from fbprophet. The Prophet model required the use of the date along with the stock values, which required the use of the stock data that was cleaned but did not have the date attribute dropped. The Prophet model was also completed for the NYC and Chicago data from the merged stock and Zillow dataset for comparison to the Prophet stock forecast.

Results

Regression Models

The evaluation metrics used to compare each regression model were the coefficient of determination (r^2), mean squared error (MSE), and root mean squared error (RMSE). Unlike the evaluation metrics provided within the literature review papers, the root mean squared log error (RMSLE) was not utilized for the comparative analysis in the models due to the sole reason that the normalized data contains negative values. In order for RMSLE to be applied properly, the log function does not work on negative or zero data values. Given that the cleaned Boulder house dataset was normalized with the Yeo-Johnson Power Transformer technique, all the data values were converted and scattered among a zero-mean gaussian-like distribution which would include negative values. Therefore, the RMSLE evaluation metric was ultimately discarded.

From the table shown in Figure 10, all of the real regression models outperform the dummy regression baseline which was highly expected. The normalized Linear Regression, Ridge Regression, and Lasso Regression models all performed relatively similarly with r^2, MSE, and RMSE scores of 0.73, 0.27, and 0.51 respectively. The Regression Tree model performed better with metric scores of 0.83, 0.19, and 0.44 respectively. The top regression model performances were displayed in the Support Vector Regression with metric scores of 0.84, 0.16, and 0.40, and the Random Forest Regression with metric scores of 0.97, 0.17, and 0.41 respectively.

	r2	mse	rmse
Baseline Linear Regression	0.000000	1.004920	1.002119
Linear Regression Normalized	0.736415	0.271402	0.519886
Ridge Regression	0.736443	0.271293	0.519789
Lasso Regression	0.736414	0.271356	0.519846
Regression Tree	0.837724	0.199796	0.445400
Support Vector Regression	0.847450	0.165054	0.404303
Random Forest Regression	0.971457	0.175955	0.418050

Figure 10: Regression Metrics

A SALE_PRICE plot comparing the target test values and the target predicted values is displayed in Figure 11 below.

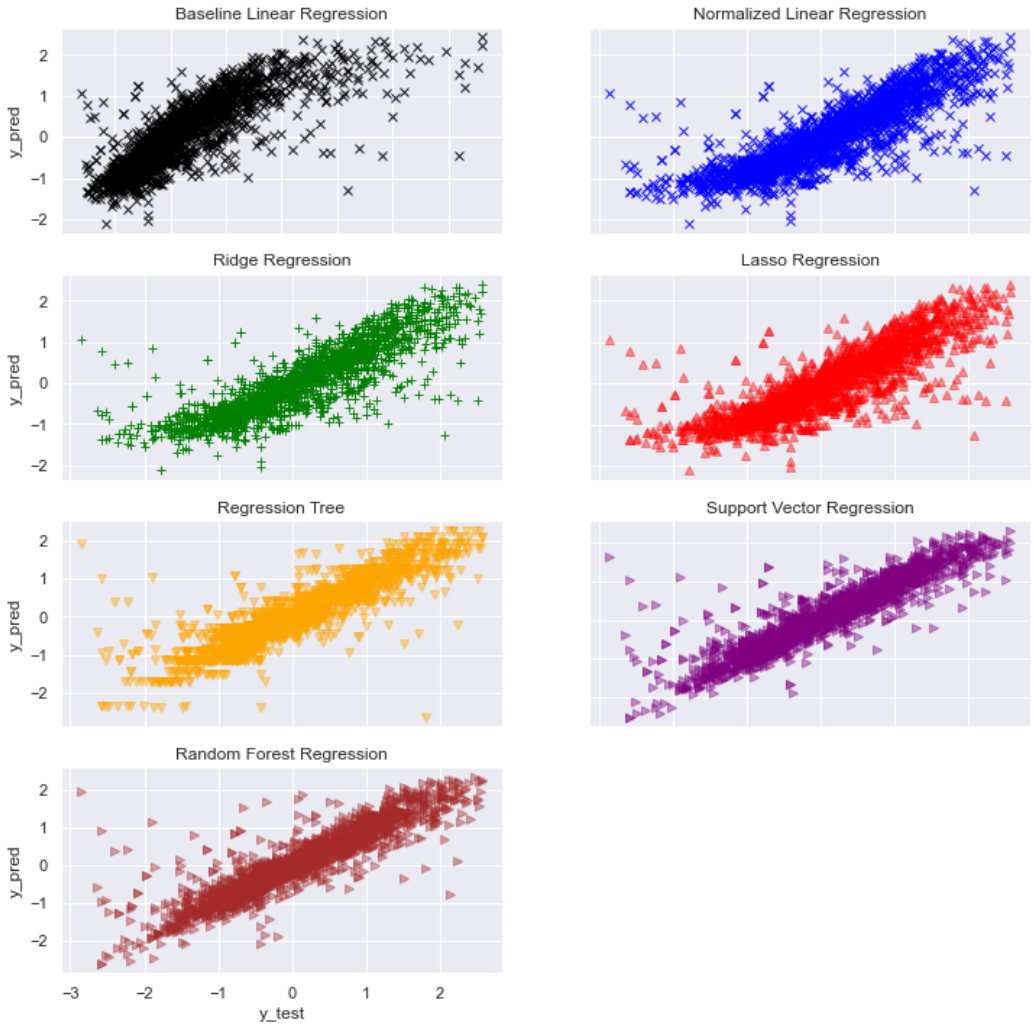

Figure 11: Regression Plots of Test Values vs Prediction Values

As for the second portion of the project, the same evaluation metrics were used for the Baseline and Linear Regression. As seen in Figure 12**,** in all five of the cities the Linear Regression models outperformed the baseline model.

Model	r2	mse	rmse
NYC Baseline	0.000000	0.081810	0.286025
NYC Linear Regression	0.685805	0.021713	0.147352
CHI Baseline	0.000000	0.074414	0.272789
CHI Linear Regression	0.500173	0.036191	0.190239
LA Baseline	0.000000	0.111340	0.333677
LA Linear Regression	0.369018	0.070030	0.264633
BOU Baseline	0.000000	0.076191	0.276027
BOU Linear Regression	0.028161	0.071262	0.266950
DFW Baseline	0.000000	0.092973	0.304915
DFW Linear Regression	0.090189	0.083990	0.289811

Figure 12: Regression Metrics for Portion 2

Classification Models

The evaluation metrics used to compare each classification model were accuracy, f1 score, precision, and recall. Additionally, a confusion matrix was created for each of the algorithms in order to get a closer look at the algorithm's predictive abilities. From the table shown in Figure 13, all of the real classification models outperform the baseline classification as expected.

	accuracy	f1	precision	recall
Dummy Classification	0.166145	0.189240	0.254107	0.169131
KNN	0.900707	0.837372	0.840084	0.838191
Support Vector Classification	0.965748	0.955019	0.955539	0.955037

Figure 13: Classification Metrics

The accuracy, f1, precision, and recall scores for the K-Nearest Neighbors model were 0.90, 0.83, 0.84, and 0.83 respectively. The evaluation metric scores for the Support Vector Machine model were 0.96, 0.95, 0.95, and 0.95 respectively. The confusion matrix for the Dummy Classifier can be seen below in Figure 14. The confusion matrices for the KNN and SVC classifiers can also be seen below in Figures 15 and 16, respectively.

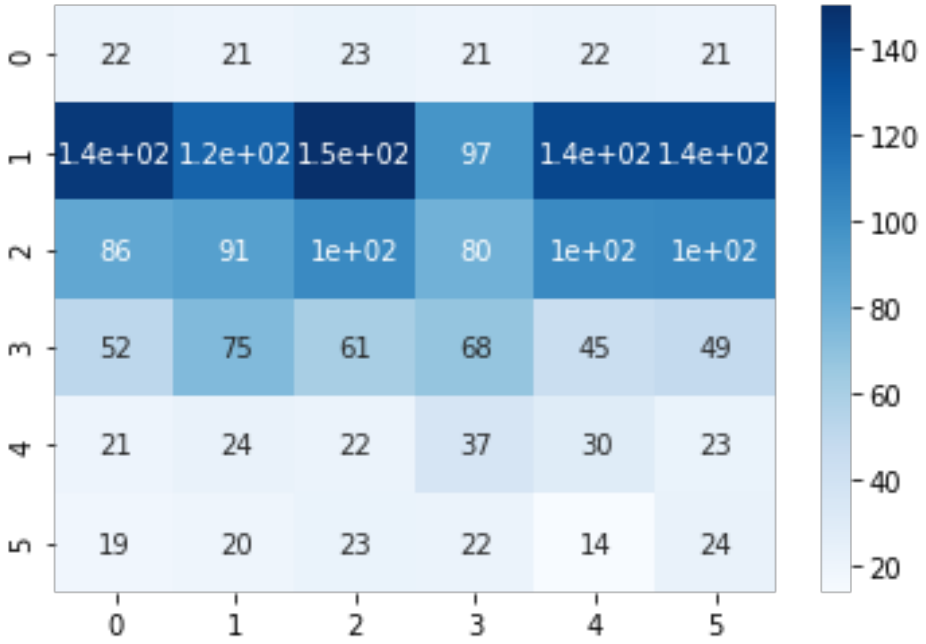

Figure 14: Dummy Classifier Confusion Matrix

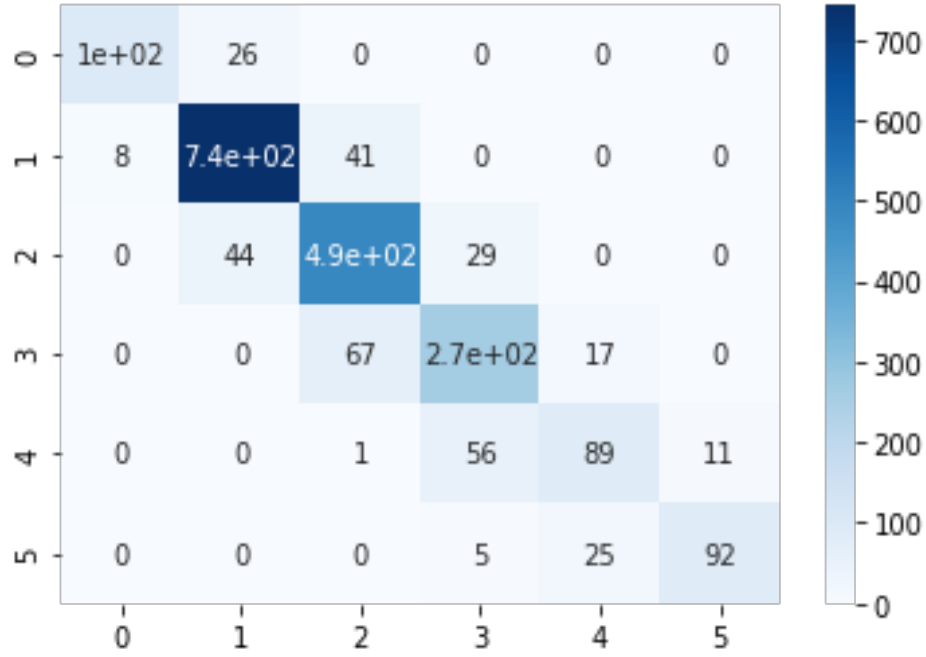

Figure 15: KNN Confusion Matrix

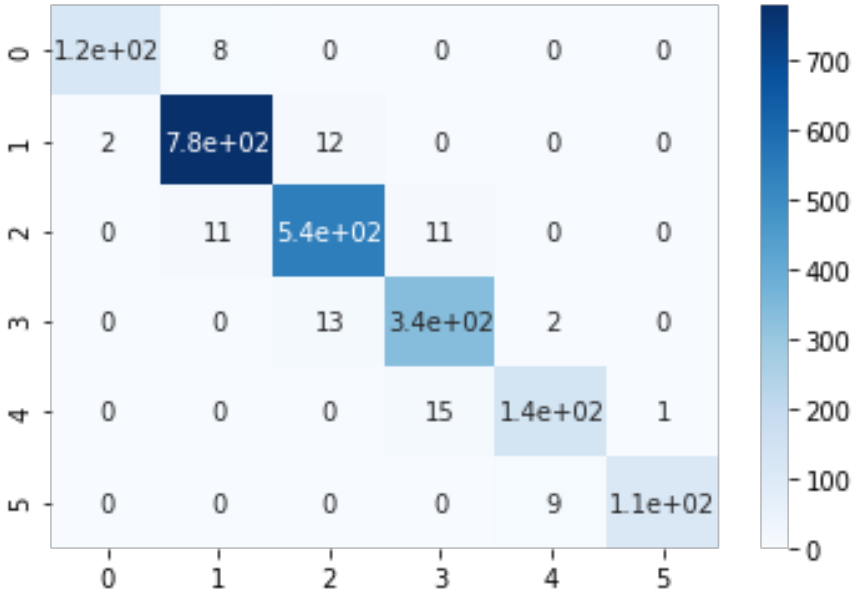

Figure 16: SVC Confusion Matrix

Forecasting Models

Autoregression and Prophet were the two forecasting models used for the second portion of the project. The evaluation metrics used for this portion were again the r^2, MSE, and RMSE. For the Prophet model, evaluation metrics were only calculated based on the training data. The plot of the Prophet model does also show a visual representation of the confidence interval of the prediction which will be shown later in this section. Evaluation metrics for the AutoRegression and Prophet models can be seen in Figure 17 below.

```
Test R^2: -6.650
Test RMSE: 15806.401
Test MSE: 249842322.142
```

```
R^2: -2.750
RMSE: 12576.599
MSE: 158170841.307
```

Figure 17: Evalutation Metrics for AutoRegression (left) and Prophet (right) Forecasting

A sample output from the AutoRegression can be seen in Figure 18 below.

```
Coefficients: [-0.64019873  0.99649509 -0.02388337  0.01095299  0.02695765 -0.04269057
  0.03045665 -0.02354334  0.03699267 -0.00756306  0.01627527 -0.07549351
  0.10115842 -0.0246341  -0.02422609 -0.01085161  0.01123536  0.00274868
 -0.01355807  0.00306199 -0.00627318 -0.00153928 -0.00696627  0.03137767
  0.00808206 -0.06340826  0.02450101  0.0548628  -0.02777371 -0.00299555]
predicted=2805.199045, expected=28494.200000
predicted=2806.084666, expected=28514.000000
predicted=2804.277509, expected=28679.810000
predicted=2798.072160, expected=28837.520000
predicted=2800.232931, expected=28586.900000
predicted=2799.320863, expected=28425.510000
predicted=2793.693076, expected=28303.460000
predicted=2794.337052, expected=27772.760000
predicted=2788.453525, expected=28148.640000
predicted=2785.492695, expected=27682.810000
predicted=2786.796031, expected=27816.900000
predicted=2782.013712, expected=27781.700000
predicted=2777.448229, expected=27452.660000
predicted=2777.413165, expected=27584.060000
predicted=2773.020667, expected=27173.960000
predicted=2771.719526, expected=26815.440000
predicted=2771.198880, expected=26763.130000
predicted=2764.971367, expected=27288.180000
```

Figure 18: AutoRegression Sample Output

Figure 19 below shows the plot of the forecast from the AutoRegression model with the red line being the prediction, and Figure 20 shows the plot of the forecast from the Prophet model.

Figure 19: AutoRegression Forecast Plot

Figure 20: Prophet Forecast Plot

Discussion

The best regression model that can be utilized to predict real estate prices is the Support Vector Regression algorithm with a r^2 value of 0.84 and RMSE of 0.40. Although relatively close to the scores of the Random Forest Regression algorithm, it is suspected that the Random Forest model may have an overfitting situation due to how its r^2 value of 0.97 is statistically close to the fitted regression line. The SVR and Random Forest model took much longer to run than the next best performing regression models Lasso Regression and Ridge Regression.

The best classification model that can be utilized to predict building designs is the Support Vector Classifier algorithm with high evaluation scores throughout each evaluation metric. Support Vector Machine has an accuracy score of 0.96 and f1, precision, and recall scores of 0.95 as opposed to K-Nearest Neighbor scores of 0.90 for accuracy, 0.83 for f1 and recall, and 0.84 for precision. Although KNN still performed admirably, it simply can't match up to SVC's classification abilities. As can be seen in Figures 15 and 16, the confusion matrix for the SVC classifier consistently logs a higher number of true positives compared to the KNN. The confusion matrices are built with classification counts from just one of the 10 folds that the algorithm was implemented on. When taking this 10X multiple into account, Figure 16 shows that the SVC classifier averages over 400 true positives more for each class when compared to KNN. As can also be seen in Figures 15 and 16, the SVC also accrued a lower count of false positives. The KNN accrued 1,240 false positives while the SVC accrued only 340, while again taking the 10X multiple into account. If this performance is taken as the average performance for all of the 10 folds, then this large difference in false positive count is something that cannot be ignored. The superiority of the SVC shines again when counting false negatives; the SVC has far fewer false negatives compared to the KNN. Lastly, the SVC also outperforms the KNN in terms of true negatives. All of these results, the classification metrics and the correlation matrices, declare the SVC as the most effective classification model without a doubt. Both classification models did outshine the dummy classification baseline. This was to be expected as the Dummy Classifier was guessing classes uniformly at random, although there was a higher number of Class 2 houses compared to other classes. These results display the potential for SVC to be used as a model for estimating the price range that a property can fall into.

The regression models used in this project were chosen to mimic some of the models used within the literature research reviews, namely Ridge and Lasso Regressions, Support Vector Regression, and Random Forest. As for the best performing model, each of the regression models performed differently compared to those seen in literature. To reiterate, Q. Truong's paper had Random Forest as the best performer (Q. Truong et. al., 2020) while J. Manasa's paper showed that Support Vector Regression performed better than both Ridge Regression and Lasso Regression (J. Manasa, et. al., 2020). However, in C. Fan's paper, the best performing model was the Ridge Regression model. Even though this machine learning project focused on Boulder,

Colorado, and those three literature review papers had different geographical data, one of the major differences between each study is how the data was preprocessed in the feature engineering procedures.

For example, in the Beijing dataset (Q. Truong et. al., 2020), some room attributes that describe the number of kitchens, bathrooms, and drawing rooms were removed while in the Boulder dataset, the attributes that describe the number of bedrooms and bathrooms were kept. Additionally, the Beijing dataset was thoroughly cleaned via the Interquartile Range (IQR) technique to remove outliers. Although the IQR technique was attempted within the Boulder dataset, it was abandoned due to its removal of almost half of the entire dataset. In another example, the Bengaluru dataset (J. Manasa et. al., 2020) used the label encoder to convert categorical values between 0 and n-1 number of unique values, and applied a log transformation to normalize the dataset. However, in the Boulder dataset, the one-hot encoder technique was used to convert categorical values into only 0s or 1s. It was assumed that one-hot encoder was a better encoder than the label encoder because label encoding works better for ordinal data whereas one-hot encoder works better for binary classifications: yes/no or true/false values. Additionally, although log transformation was attempted initially in the Boulder dataset, MinMaxScaler and PowerTransformer prove to be better normalization techniques to generate equal weighing of selected features and Gaussian-like distributions of the data.

In the data analysis for determining the relationship between housing prices and stock market indices, it was found that there are housing markets that do have some correlation with the stock market. The home values in different cities have different relationships with the stock market and through the cities we examined it was shown that the more densely populated cities have a higher correlation with the stock market than the other cities. From our analysis New York City and Chicago notably have the highest correlation of the five examined cities--Los Angeles, Dallas-Fort Worth, and Boulder--and it was speculated that it may be due to the fact that New York City and Chicago actually have their own stock exchanges. While the regression models could not alone predict the home value trends, they helped to further visualize and understand the relationship between the stock market and home values that was identified for certain cities within the correlation matrix. Through these relationships, we determined it was worthwhile to work on a simple forecasting model for stocks. If we were able to create a model to accurately predict the trend in stock prices in the coming years, then this could potentially be the start to understanding how the home values in the highly correlated cities would trend in those coming years. While the Prophet model did outperform the AutoRegression model, objectively, neither of the forecasting models showed good performance.

This second portion of the project sought to provide insight into how the external market impacted the value of homes, so that could be coupled with the static home price prediction model from part one which focused solely on the home attributes. At the start, it was decided that

we would proceed with using the stock market as the external market factor we would examine. After our analysis, we can determine that the stock market does not appear to be the best macroeconomic feature for prediction of home price trends in most cities.

Conclusion and Future Work

This project investigates different machine learning algorithms for house price prediction as well as attempting to discover any relationship between house prices and stock market indexes. Regression models used in the house price prediction were linear regression, Ridge Regression, Lasso Regression, Regression Tree, Support Vector Regression, and Random Forest Regression. The coefficient of determination, mean squared error, and root mean square error were the evaluation metrics for the comparative analyses and resulted in the Support Vector Regression performing the best with Random Forest Regression being a close second. Classification models used in the house building design class prediction were K-Nearest Neighbors and Support Vector Classifier in which the classification evaluation metrics--accuracy, f1 score, precision, and recall--determined that the Support Vector Classifier also performed extremely well. For the second portion of the project a baseline regression and linear regression were preliminarily performed. In part 2, which looked at an analysis in the determination of any house price and stock market index relationship, AutoRegression and Prophet models were used to attempt some level of forecasting capability, however, it was deemed statistically insignificant due to limited data, low correlation in certain city markets, and poor performance.

Future research on some topics may be conducted to further investigate all the models implemented within this project. One of the biggest roadblocks in machine learning is the proper procedures in accurately preprocessing the dataset before the model implementation. Most of the outlier-cleaning techniques used in this project were manually removing extreme outliers that had very low instances, and removing categorical values based on domain knowledge in the housing area. Techniques similar to the interquartile range removal technique can be applied in future works simply because they are effective, however, IQR is highly sensitive to outliers as seen in this project.

Other topics that can be explored further include feature engineering such as encoding, scaling, and normalizing techniques. As such, the feature engineering techniques used in this project were One-Hot Encoding, MinMaxScaler, and PowerTransformer. There are other ways to encode categorical values into numerical values for regression modeling such as Label Encoding, Target Encoding, Dummy Encoding, Hash Encoding, and more. Perhaps the future works can include implementing the target encoding technique because it is a Bayesian-like encoding technique where the mean of the target variable for each category is calculated (similar to a Group By statement), and each mean replaces each categorical variable. For feature scaling and normalization techniques, more research is needed to implement new strategies such as the Robust Scaler with the interquartile range cleaning or a Maximum Absolute Scaler that rescales numerical values between ranges of -1 to +1 instead of using MinMaxScaler which rescales to between 0 and 1.

Additional, future research can be done to improve the machine learning algorithms by optimizing the parameters of the models. In order to avoid model overfitting, or even underfitting, it is important to find a good level of accuracy within the alpha parameter for the ridge and lasso regressions, the epsilon parameter for the support vector models, the max depth parameter for the tree models, and k-value parameter for the KNN clustering model.

Future work for the second portion of this project could include creation of a better stock forecasting model to be used to gather insight on the trend of home values in the future. Of course, as shown within the report, this would only prove beneficial in certain city markets. Given that information, future work should put more of a focus on identifying which macroeconomic and market factors would provide more accurate insight on predicting the movement trend of home values in the future. Then, using any significant macroeconomic features identified it would be beneficial to build a better model for forecasting the movement of the home value predicted in the first portion of the project.

Once a model is created that accounts for the macroeconomic and market factors effectively, the model(s) from part one of the project should be combined with that model. This would then give the full picture on the value of the home in this moment and the value of the home in the coming years. This type of combination model would allow buyers, sellers, and startup real estate investment firms to make very educated choices about when to buy and sell their home(s).

Time Logs

Zachary Sisco:

- **10/12**:1-2:30pm:1.5 Hours: Searching for topic ideas
- **10/13**:1:30-2:30pm:1.5 Hours: Searching for topic ideas
- **10/13**:9:30-10pm::.5 Hours: Searching for topic ideas
- **10/14**:3-4pm:1 Hour: Searching for topic ideas
- **10/16**:2:30-6:30:4 Hours: Data and literature acquisition
- **10/16**:6:30-8pm:1.5 Hours: Literature review
- **10/16**:8-9:30pm:1.5 Hours: Begin writing the proposal
- **10/17**:9-1am:4 Hours: Writing the proposal
- **10/18**:11-12:30pm:1.5 Hours: Writing the proposal
- **11/26**:11am-1pm:2 Hours: cleaning the stock data file
- **11/27:**12pm-2pm:2 Hours:analyzing nyc data, finding it to be unsuitable
- **11/28:**11am-12:30pm:1.5 Hours:looking for new housing data to be used for price prediction model
- **11/28**:1:30pm-3pm:1.5 Hours: Compiling new housing data
- **11/29**:10am-2pm:4 Hours:Cleaning housing data
- **11/30**:9pm-10:30pm:1.5 Hours: Cleaning data
- **12/1**: 12pm-2pm:2 Hours: Cleaning data
- **12/3**: 1pm-530pm: 4.5 Hours: Cleaning data and beginning analyses
- **12/4**: 130pm-5pm: 3.5 Hours: Cleaning data and testing analyses
- **12/5**: 4pm-1030pm: 6.5 Hours: Model evaluations
- **12/6**: 12:30pm-8pm: 7.5 Hours: trying to get the damn models to beat the baseline
- **12/7**: 1:30pm-4pm: 2.5 Hours: running models
- **12/7**: 7pm-9pm: 2 Hours: running models
- **12/7**: 9pm-10pm: 1 Hour: presentation discussion,
- **12/9:** 8pm-1045pm: 2.75 Hours: writing the final report
- **12/10:** 8pm-1045pm: 2.75 Hours: writing the final report
- **12/12:** 10:30am-1:30pm, 4pm-5pm: 4 hours: writing final report
- **12/13:** 630pm-1030pm: 4 Hours: Writing final report

Holly Sickler:

- **October 6**: 9:30PM to 10PM: 0.5 hours: Forming group, discussing topics, and searching topics.
- **October 11**: 12:30AM to 1AM: 0.5 hours: Locating articles on topics.
- **October 12**: 6PM to 7PM: 1 hour: Determining if data will be available for the potential topics. Considering choosing a new topic idea due to accessibility of data.
- **October 13**: 12:30PM to 1PM & 6PM to 6:30PM: 1 hour: Research and discussion for potential topic.
- **October 15**: 7AM to 7:30AM & 7:30PM to 8PM & 11:30PM to 11:59PM: 1.5 hours: Topic finalization, identification of potential methods, and browsing articles as potential articles for the literature review.
- **October 16**: 3:30PM to 4PM & 11:30PM to 11:59PM: 1 hour: Exploring business need/problem statement/introduction content, picking articles for literature review out of articles identified, and formatting presentation materials.
- **October 17**: 12:15PM to 2:15PM: 2 hours: Investigating additional data sources, evaluating current data sources, working to better understand the problem statement, as well as literature review.
- **October 17**: 3:15PM to 6PM: 2.75 hours: Literature review.
- **October 17**: 6:45PM to 8PM & 10PM to 11:15PM: 2.5 hours: Writing Proposal and PowerPoint Slides, as well as solidifying approach and methodology.
- **October 18**: 5:30AM to 7:30AM & 11AM to 1PM: 4 hours: Writing Proposal and PowerPoint Slides.
- **December 1:** 5:10PM to 6:10PM: 1 hour: Starting Final Report and Presentation documents/slides, as well as looking at data.
- **December 3**: 4:30PM to 8:30PM & 8:35PM to 9:35PM: 5 hours: Exploratory analysis of all of the data, group meeting, and Part 2 models.
- **December 4**: 11:35AM to 2:35PM & 4:00PM to 5PM & : 4 hours: Part 2 models.

- **December 5**: 3:15PM to 5:45PM & 6:15PM to 8:30PM & 11:44PM to 11:59PM: 5 hours: Part 1 data exploration, Part 1 data cleaning, and Part 2 evaluation.
- **December 6**: 12AM to 2AM & 12:30PM to 5PM: 6.5 hours: Part 1 data cleaning and models.
- **December 7**: 7:15PM to 8PM & 8:45PM to 11:00PM: 3 hours: Presentation content and preparation.
- **December 12**: 2:30PM to 6:45PM & 9:45PM to 11:59PM: 6.5 hours: Cleaning up and reviewing the Jupyter notebook. Writing the report and preparing for submission.
- **December 13**: 1AM to 1:30AM & 1:30PM to 2:05PM & 2:50PM to 3:45PM & 6PM to 7PM: 3 hours : Writing the report and preparing for submission.

Clarence Li:

- **October 6:** 9:30PM-10:00PM: 0.5 hour: Brainstorming of machine learning ideas and group topic discussions.
- **October 9**: 5:00PM-6:00PM: 1 hour: Literature research and finding potential data sources.
- **October 12**: 6:00PM-7:00 PM: 1 hour: Brainstorming of machine learning ideas and group topic discussions.
- **October 13**: 12:30PM-1:30 PM, 10:30PM-11:30PM: 2 hours: Brainstorming of machine learning ideas and group topic discussions. Literature research and finding potential data sources.
- **October 14**: 5:00PM-6:00PM, 11:00PM-11:30PM: 1.5 hours: Literature research and finding potential data sources. Brainstorming another topic due to previous topic having limited and unattainable data (access requirements).
- **October 15**: 7:00PM-8:30PM: 1.5 hours: Brainstorming final topic, literature research, and collection of data sources.
- **October 16**: 10:00AM-11:00AM, 2:00PM-3:00PM, 4:00PM-5:00PM, 9:00PM-11:PM: 5 hours: Brainstorming final topic, literature research, and collection of data sources. Summarizing selected literature research for reference. Proposal design and writing.
- **October 17**: 10:00AM-11:00AM, 4:00PM-5:00PM, 11:30PM-1:30AM: 4 hours: Summarizing literature research. Proposal design and writing.
- **October 18**: 8:00AM-9:00AM, 12:00PM-1:00PM: 2 hours: Proposal design and writing.
- **November 29:** 5:00-6:00PM, 10:00-11:00PM: 2 hours: Data cleaning.
- **December 1**: 5:00-6:00PM: 1 hour: Data cleaning.
- **December 2**: 2:00-4:00PM, 8:00-11:00PM: 5 hours: Data cleaning. Model implementation. Group discussion.
- **December 3**: 5:00-6:00PM, 8:00-10:00PM: 3 hours: Data cleaning. Model implementation. Group discussion.
- **December 4**: 3:00PM-6:00PM, 8:30PM - 9:30PM: 4 hours: Data cleaning. Model implementation. Group discussion.
- **December 5**: 2:00PM-6:00PM: 4 hours: Data cleaning. Model implementation. Group discussion.
- **December 6**: 2:00PM-6:00PM: 4 hours: Data cleaning. Model implementation. Evaluation metrics.
- **December 7**: 1:00PM-3:00PM, 6:00PM-11:00PM: 7 hours: Model implementation. Evaluation metrics. Presentation slides. Group discussion.
- **December 9**: 4:00PM-6:00PM: 2 hours: Final coding touch-ups.
- **December 10**: 3:00PM-4:30PM, 11:30PM-1:00AM: 3 hours: Final coding touch-ups.
- **December 11**: 4:00PM-6:00PM, 8:00PM-10:00PM: 4 hours: :Final coding touch-ups. Final report writing.
- **December 12**: 2:00PM-4:00PM, 10:00PM-12:00AM: 4 hours: Final report writing.
- **December 13**: 12:00PM-1:00PM, 4:30PM-5:30PM: 2 hours: Final report writing.

References

A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT),* Coimbatore, 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.

Boulder County Open Data, 2020. Boulder County, "Lists of Recent Sales by Property Type and Time Frame", https://www.bouldercounty.org/property-and-land/assessor/sales/recent/.

B. Park, & J.K. Bae, "Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data," *Elsevier Expert Systems with Applications, 42*(6), 2015, pp. 2928-2934, doi:10.1016/j.eswa.2014.11.040.

Brownlee, J., (2020, August 14), Autoregression Models for Time Series Forecasting With Python, Retrieved December 3, 2020, from https://machinelearningmastery.com/autoregression-models-time-series-forecasting-python/.

C. Fan, Z. Cui, & X. Zhong, "House Prices Prediction with Machine Learning Algorithms," *Proceedings of the 2018 10th International Conference on Machine Learning and Computing - ICMLC 2018,* 2018, pp. 6-10. doi:10.1145/3195106.3195133.

J. Manasa, R. Gupta and N. S. Narahari, "Machine Learning based Predicting House Prices using Regression Techniques," *2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA)*, Bangalore, India, 2020, pp. 624-630, doi: 10.1109/ICIMIA48430.2020.9074952.

M. Jain, H. Rajput, N. Garg and P. Chawla, "Prediction of House Pricing using Machine Learning with Python," *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Coimbatore, India, 2020, pp. 570-574, doi: 10.1109/ICESC48915.2020.9155839.

"Negative Equity in the United States: HUD USER", Retrieved October 18, 2020, from https://www.huduser.gov/portal/pdredge/pdr_edge_research_072012.html.

P. A. Viktorovich, P. V. Aleksandrovich, K. I. Leopoldovich and P. I. Vasilevna, "Predicting Sales Prices of the Houses Using Regression Methods of Machine Learning," *2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)*, Vladivostok, 2018, pp. 1-5, doi: 10.1109/RPC.2018.8482191.

Population Density for U.S. Cities Statistics, Retrieved December 04, 2020, from https://www.governing.com/gov-data/population-density-land-area-cities-map.html.

Q. Truong, M. Nguyen, H.Dang, and B.Mei, “Housing Price Prediction via Improved Machine Learning Techniques,” *Elsevier Procedia Computer Science, 174*, 2020, pp.433-442, doi:10.1016/j.procs.2020.06.111.

Quick Start, Retrieved December 3, 2020, Retrieved from https://facebook.github.io/prophet/docs/quick_start.html.

R. Bouchouicha and Z.Ftiti, “Real estate markets and the macroeconomy: A dynamic coherence framework,” *Elsevier Economic Modelling, 29*(5), 2012, pp. 1820-1829, doi:10.1016/j.econmod.2012.05.034.

Wall Street Journal, 2020, “WSJ | Markets.” Dow Jones & Company, Inc., https://www.wsj.com/market-data?mod=nav_top_subsection.

Zillow. 2020, “Housing Data”, Zillow Research, https://www.zillow.com/research/data/.

Appendix

	Neighborhood Code	Account #	PARCELNB	PROPERTY_ADDRESS	LOCCITY	SUBNAME	MULTIPLE_BLDGS	ACCOUNT_TYPE	BLDG1_DESCRIF
0	135	M8724018	146321300001	5000 BUTTE ST 187	BOULDER	VISTA VILLAGE - MHP BOV	NO	MANUFACTURED HOME	MANUFACT HOU IMPROVEM
1	135	M9000082	146322300014	5505 VALMONT RD 276	UNINCORPORATED	SAN LAZARO - MHP BOV	NO	MANUFACTURED HOME	MANUFACT HOU IMPROVEM
2	203	M8724746	120534203001	729 17TH AVE 31	LONGMONT	WESTON MANOR - MHP LG	NO	MANUFACTURED HOME	MANUFACT HOU IMPROVEM
3	460	M9200147	157502400007	11990 SOUTH BOULDER RD 221	LAFAYETTE	BOULDER RIDGE MHP FKA COTTONWOOD VILLAGE - LA	YES	MANUFACTURED HOME	MANUFACT HOU IMPROVEM
4	115	R0002870	146319327003	3025 17TH ST	BOULDER	SILVER MAPLE	NO	RESIDENTIAL	SINGLE FAI

Figure 1: Section of Boulder County Dataset

Row	Date	New York, NY	Los Angeles-Long Beach-Anaheim, CA	Chicago, IL	Dallas-Fort Worth, TX	Philadelphia, PA	Houston, TX	Washington, DC	Miami-Fort Lauderdale, FL	Atlanta, GA	Boston, MA	San Francisco, CA	Detroit, MI	Riverside, CA	Phoen
0	1/31/96	107630	187842	183929	164647	114406	121233	110773	178034	109403	120291	168008	246464	105560	1213
1	2/29/96	107657	187403	184185	164345	114471	121078	110849	177811	109533	120519	168003	245634	105940	1208
2	3/31/96	107707	187125	184205	163946	114634	120937	110858	177681	109670	120740	168176	245234	106307	1205
3	4/30/96	107834	186592	184312	163493	114962	120693	111007	177407	109917	121229	168453	244494	107095	1198
4	5/31/96	107977	186274	184286	162886	115314	120527	111148	177288	110132	121711	168821	244171	107902	1193

Figure 2: Zillow Historical Dataset

Date	VIX Open	VIX High	VIX Low	VIX Close	Unnamed: 5	NAS Open	NAS High	NAS Low	NAS Close	Unnamed: 10	SP Open	SP High	SP Low	SP Close	Unnamed: 15	DOW Open
10/15/20	27.10	29.06	26.82	26.97	NaN	11559.88	11740.68	11559.10	11713.87	NaN	3453.72	3489.08	3440.89	3483.34	NaN	28323.40
10/14/20	25.72	27.23	25.53	26.40	NaN	11889.07	11939.92	11714.35	11768.73	NaN	3515.47	3527.94	3480.55	3488.67	NaN	28731.30
10/13/20	25.67	26.93	25.16	26.07	NaN	11901.76	11946.98	11821.83	11863.90	NaN	3534.01	3534.01	3500.86	3511.93	NaN	28764.95
10/12/20	25.65	25.65	24.14	25.07	NaN	11732.33	11965.54	11704.13	11876.26	NaN	3500.02	3549.85	3499.61	3534.22	NaN	28671.12
10/9/20	26.20	26.22	24.03	25.00	NaN	11487.60	11581.23	11476.66	11579.94	NaN	3459.67	3482.34	3458.07	3477.13	NaN	28533.61

Figure 3: Historical Stock Price Data