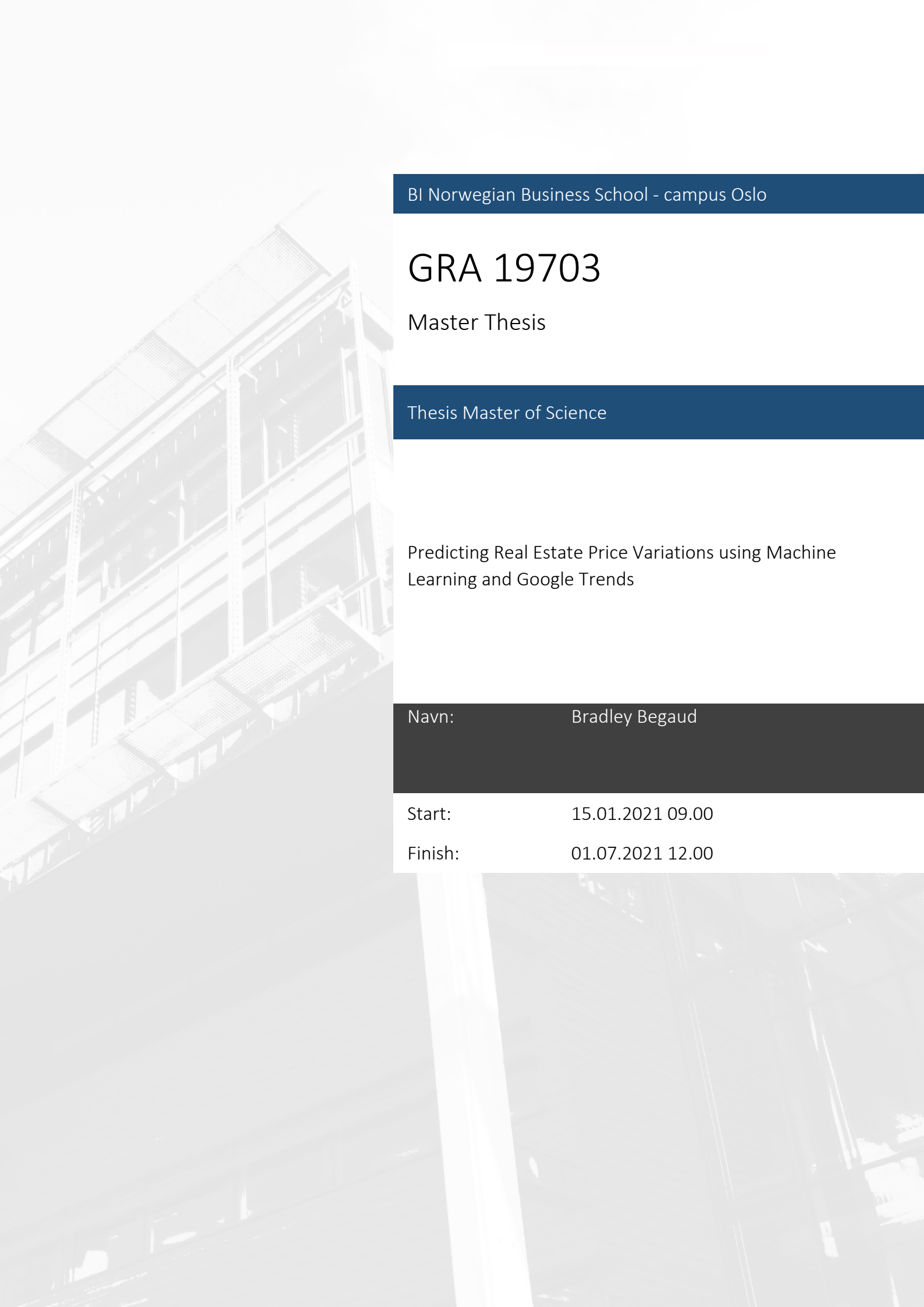


BI Norwegian Business School - campus Oslo

GRA 19703

Master Thesis

Thesis Master of Science

Predicting Real Estate Price Variations using Machine Learning and Google Trends

Navn:	Bradley Begaud

Start:	15.01.2021 09.00
Finish:	01.07.2021 12.00

Thesis Final Report

Predicting Real Estate Price Variations using Machine Learning and Google Trends

Supervisor: Paulo Giordani **BI Norwegian Business School**

Program: Double Degree Master

Abstract

The goal of this paper is to create a modern model via the use of machine learning (such as support vector regression, regression tree and neural networks) and google trends to predict real estate price variations. The model should achieve significant predictive capabilities in monthly variations and should be both interpretable and not overly complex. There is major interest in being able to predict real estate prices and many articles have been published on the subject. Most traditional models use economic data which are usually published quarterly or annually and thus are not very efficient for short term predicting. As an investor, real estate has always been an asset class of interest for its performance, diversifying effect on a portfolio and its interest to a short or long term investor. The interest in the subject goes beyond investors as it is one of the most important costs for a regular family. These models will use as inputs various variables that effect either directly or indirectly prices in real estate. We will focus on the Miami metropolitan area or the Miami-Fort Lauderdale-Pompano Beach area. The US market was chosen because it provides the best access to reliable and consistent data. Our model will also focus on predicting single family house prices which are very popular in the US.

Table of Contents

1. Introduction

Real estate holds a special place as one of the oldest investments known to man. It occupies a place of great importance in our daily lives, where it is linked to the most basic of our needs, the need for shelter. Moreover, we also use it for agricultural production, a place for our work and of course as an investment. Historically, it has often been considered a safe haven asset, especially in the US. This explains some of the shock people had during the recession of 2008 where real estate prices crashed, ending most of the belief in its safe haven status.

Currently, the real estate industry is experiencing another major shock, that of the Coronavirus epidemic and the resulting economic recession in most of the world. Like many other fields or assets, the epidemic has impacted severely the real estate industry with trends like working from home encouraging companies to reduce their offices and real estate space. This puts downwards pressure on real estate prices with even some major companies like Twitter and Facebook expecting that after the pandemic it will encourage most of their workforce to work from home.

All things being considered, being able to predict real estate price variations would prove to be very important. As much for an individual buying his first house, an investor profiting on capital gains or a company looking for new offices. Econometric models have been used in the past to predict real estate prices giving relatively accurate forecasts. However, with the current digital revolution, increasing hard drive capacities, internet speed, databases and processing speed many professionals look towards more advanced models using machine learning and artificial intelligence who can have higher predictive capacities.

In our case, we represent the view of the investor, the investor who with modern tools is looking for the highest accuracy in predicting real estate prices in a faster paced world, filled with data who aims to satisfy his investment/portfolio goals.

2. Literature Review

It is very important to point out that there are many studies about the prediction of real estate prices, however the vast majority predict precise nominal prices and not the variation in prices. As mentioned previously macro-economic data is the basis of most linear real estate predicting models. This data would then be used in autoregressive mathematical models to find trends and aim to predict future prices or observe the extent of the correlation. *Baffoe-Bonnie.* (1998) analyzed the impact of 4 macro-economic aggregates, mortgage rates, consumer price index (CPI), changes in employment and money supply on the prices of real estate and its cycles. Already, it showed that there was a strong correlation and that real estate prices were sensitive to macro-economic data, however the conclusion was that by themselves they were insufficient to explain fluctuations in real estate data and construction levels. *Goetzmann and Rouwenhorst*. (2000) show the high correlation between international real estate and the importance of Gross national product (GNP) on real estate prices and suggests that fundamental economic variables play a very strong role in real estate internationally.

There are more than just economic variables that have been used to predict real estate prices and even completely alternative methodologies, *Dubin*.(1998) uses a technique called kriging whereby using the real estate listings from the neighborhood and adding them to an ordinary least square regression model, effectively adding neighborhood data to get a more precise prediction. This achieved a better fit than previous OLS regressions models. Consumer confidence is another important indicator, but it is a more psychological indicator, it has been noted to be especially important in consumer spending. *Ludvingson*.(2004) shows how increases in consumer confidence can result in increases in consumer spending in the near future. It is no surprise that this indicator has also been tested for its correlation to real estate prices. *Meulen, Micheli, Schmidt*.(2014) have shown that consumer confidence has some forecasting ability for real estate prices on the German market. More so, out of all their explanatory variables the number of new building permits proved to be very successful at predicting real estate prices. Finally, consumer confidence alone was not

a good predictor and needed to be accompanied by macroeconomic variables. However we note that this study was performed over a short period of 7 years from 2007 to 2013 and we will use consumer confidence over a longer period of time in ours.

All these methods give us insight into possible models that can be applied to have some precision in predicting real estate prices, but with our new digital era more modern methods have appeared and even old methods have been revisited with increased efficiency. All of this is fueled by progress in artificial intelligence, machine learning, big data, alternative data sources and in general increased computer speed and prowess. Artificial intelligence has found a wide range of uses from self-driving vehicles, intelligent weaponry, precision agriculture to better manage crops and animals, medicine in recognizing skin cancer for example, portfolio management and predicting real estate prices. *Tabales, Ocerin, Carmona.* (2013) have shown that artificial neural networks (ANN) yielded significantly better results at predicting real estate prices in Spanish cities than traditional econometric models. *Ceh, Kilibarda, Lisec, Bajat.* (2018) showed that random forest method (method used in algorithm based models and machine learning) was superior to the classic multiple linear regression even in a small market such as Ljubljana, capital of Slovenia. An example of an older method being revisited with modern tools is with *Nadai,Lepri.* (2018). They have shown the economic value in neighborhoods when appraising the price of real estate property by using tools such as of Google Street View, which use a modern camera/photography system to estimate the traffic and security level in Italian neighborhoods. *Pai, Wang.* (2020) has compared different models in machine learning when predicting real estate prices. Four different models were used such as classification and regression tree (CART), least squares support vector regression (LSSVR), general regression neural networks (GRNN) and finally backpropagation neural networks (BPNN). The study concluded that all four models yielded good results when predicting real estate prices on the Taiwanese market. However, the least square support vector regression proved to be the most effective and outperformed the others in their study. We can observe that some of these studies have shown better forecasting ability whether through their use of modern data or

modern tools to analyze it. Our model will aim at both.

Lastly we can talk about google trends, a tool that has shown promising results in a diverse range of applications. It shows the amount of search queries on a certain topic and has shown much use in predicting trends/events for business purposes at zero cost. As far back as 2011, *Gawlik, Kabaria,* Kaur.(2011), showed that google trends showed very strong results with low test and training error in predicting upcoming tourist destinations for consumers. More recently *Roman, Martinez, Cruz.*(2020) showed that google trends was a strong predictor for presidential elections in the US and Canada. In the context of predicting real estate we have *Wu, Brynjolfsson*.(2015). Their study finds that there is a very strong correlation between google trends and the number of real estate sales, one of their models shows that for a 1% increase in search queries for the term "real estate agencies" results in 16550 home sales in the next quarter. The study also finds correlation between search queries and the real estate price index but it is not as strong as the correlation to sales.

3. Data Collection

Our study will have as our dependent variable (y) the log return of the average real estate price of a family home in the Miami metropolitan area at t+1. We will have a total of 13 independent variables (X) which will be set at time t. Thus, the 1 month lag for the purpose of the monthly prediction model. The timeframe will be from January 2004 to February 2020; this gives us 194 observations for each variable. The US was chosen for the wide availability of data on its market and the real estate data was taken from Zillow. As mentioned previously the real estate market is very regional in nature and this is will result in different results depending on the area. Miami was chosen as it has been a real estate hot spot in the US, the metropolitan population is significant (almost 6.2 million people as of January 2019, FRED Statistics), it is also a significant business hub and the capital of Latin culture in the US. The single family home was chosen because it is among the most common and popular types of homes in the US and holds a symbolical value as a pillar of the American dream.

To fulfill our goal of creating a predictive model adapted for our digital age we have focused on monthly data, so that the data may be updated quickly and lead to higher efficiency in short term predicting. We will for some variables, use quarterly data. In these cases the data considered is believed to have a significant impact on real estate prices. Consumer sentiment will also be included in the variables. This variable has shown promise in predicting real estate prices in a previous study. We will also have some variables which are real estate specific. Finally, to further contribute to our goal we are using google trends, a tool that provides monthly data since 2004 and shows a standardized index of search queries (i.e. how many times a specific term was looked up on the search engine on a basis of 100 being the highest). As we have seen there is prior evidence that a google trend has some predictability in real estate prices. The study will analyze various data from the Timeline of 2004 until early 2020.

3.1 List of Explanatory Variables

Macroeconomic data represents an important data source since it is cyclical and even an individual with the intention of buying a real estate asset/home can be impeded

under poor macroeconomic conditions. This represents the bulk of our independent variables with 8 macro-economic variables:

Unemployment rate (%) being the first variable, this is a key variable with most governments/central banks aiming to have a very low unemployment rate. This is important for the welfare of the general population and economic welfare of a nation. A low unemployment means more people earning working revenue, stimulates consumption and savings and thus purchasing power and the ability for people to take a mortgage and buy a home, impacting real estate prices. We will focus more specifically on the unemployment rate in the Miami Metropolitan area as we expect the impact to be noticeable quicker than the national unemployment rate. This variable is published monthly from the economic data of the Federal Reserve Bank of St Louis or FRED. We will use the base form (percentage) of the data in our model.

The Fixed Mortgage Rate 30 years is the second, this variable is an average of rates across the US, since individual mortgage rates also depend on the commercial banks and the credit score of the client in question. The fixed rate loans represent the vast majority of all mortgage loans taken by consumers to finance their acquisition of a real estate asset. This is due to the higher certainty against floating rates/adjustable mortgages which can change as frequently as the federal funds rate. This data is published monthly and taken from the FRED. The base form (percentage) of the data will be used in our model.

Consumer Price Index for All Urban Consumers will be used as proxy for inflation, this gives a number with the base index of 100 occurring in 1983, the higher the number the more expensive general prices are and takes into account popular items consumed by the general population, rent, food, shelter, clothes, fuel, electricity etc. This concerns only urban consumers but is a very good representative since it accounts (according to FRED) for roughly 88% of the US population and Miami is an urban center. Inflation is included in our study since historically, gold and real estate assets have been considered great hedges over inflation. More so, we expect that an investor expecting higher inflation would invest in real estate assets increasing their price. This data is published monthly and taken from the FRED. The base form of the data will be used in our model.

Custom Affordability Index: We created a custom affordability index for our study which takes the average price of a Single Family Home at time t and divides it by the real disposable Income in billions of USD at time t. Real disposable income represents the income households have after taxes and benefits have been accounted for. It is published monthly and taken from the FRED. The Real estate data is from Zillow and is the same as in our dependent variable but taken at time t. We believe this variable could indicate the creation of a real estate bubble when prices increase much faster than real income. In our model we use the natural logarithm of this custom index.

Currency in Circulation: This represents the total amounts of billions of USD circulating in the economy and is a direct result of the monetary policy of the FED which can reduce or expand the monetary supply. An increase in currency in circulation can impact directly the prices of real estate through direct buying or indirectly through increased inflation due to this monetary policy. This data is published monthly and taken from the FRED. It is used in the form of log returns on a 12 month period.

Nominal GDP (US): The GDP is the one of the most known economic metrics and represents the amount of goods/services produced within a territory multiplied by their currency value. Here we have taken the nominal GDP because we already have an inflation proxy in our variables. An increase in GDP is expected to make real estate prices appreciate. This data is published quarterly and taken from the FRED. Lastly, we will be using the log returns on a 12 month period.

S&P500 returns: The S&P500 is currently the best representative of the US equity market by taking 500 of the largest US companies from diverse industries. We expect positive stock returns to have an indirect effect on real estate as gains in the stock market could be used to purchase/diversify into real estate assets. This data is obtained from the Wharton Research Data Services (WRDS), the original data is daily, but it was modified in monthly values for the purpose of this model. Just like our y variable we will be using the log returns formula for the S&P500 variable.

After the macro economic variables we have some independent real estate variables.

The number of new permits authorized for private real estate units: The title of this

variable is self-explanatory as to its meaning and concerns the Miami Metropolitan Area. The macroeconomic variables up to now are expected to explain the demand side of real estate transactions, however this variable concerns the supply side. We expect this variable to have downward pressure on real estate prices since an increase in the amount of building units will increase the real estate supply. The importance of this variable was shown in the same study for consumer confidence (*Meulen, Micheli, Schmidt,2014*) which said it was the most important variable to predict real estate prices when studied with consumer confidence. It is published monthly by the FRED and is used in its base form here.

All Employees: Construction in Miami-Fort Lauderdale: This variable gives the number of people employed in the construction sector in the Miami Metropolitan area. Although not all construction is related to real estate, we expect this variable could have an indirect effect since an increase in real estate demand would increase the number of workers and number of homes built. Thus, would affect the supply side of the real estate market. This data is published monthly by the FRED and used in base form in our model.

Delinquency Rates on Real Estate Loans: This variable returns the percentage of borrowers who are delinquent on their loans i.e. have missed two consecutive payments on their mortgage. This is not to be confused with default which is when the borrower has been late for 270 days. The rise in this metric preceded the 2008 financial crisis and the crash of the US real estate market. Therefore, we believe it to be an important variable which could predict downward trends on the real estate market. The data is published quarterly from the FRED and used in base form in our model.

Net Percentage of Domestic Banks Tightening Standards for Commercial Real Estate Loans: For our last real estate specific variable, this one measures the net percentage of banks that are restricting or making it harder for applicants to obtain their loans. Even though this does not concern the single family homes, the different sub branches of real estate are correlated and we believe that a higher restriction on commercial real estate loans would indirectly impact the prices of single family homes as it would be a further sign of economic hardship ahead. The release of this

data is by the FRED and quarterly. However, this data was discontinued in 2013 and replaced with three different sub-categories which represent the entire commercial real estate loans market, thus for 2014 and upwards we have taken the average of those 3 sub categories. This is possible since their values are very close to each other. The data is used in its base form in our models.

Subsequently, we have the independent variable of consumer confidence. Consumer confidence is identified as the degree of confidence or optimism in the business climate, the US economy and the personal finances of the American public at the time. The most common index is from the University of Michigan which is the one we will be using. The higher the number the more optimistic US consumers are, thus we expect that a high number will have upwards pressure on real estate prices. It's effectiveness has already been discussed in detail in the literature review. We will use this data in its base form.

Google trends will be our final variable, which is an online tool that gives us the search queries for a particular term or topic. The tool offers us the possibility to compare different search queries and aggregate them on the basis of the main one. For our study we have taken the aggregate of 5 search possibilities which correspond to the main parts in the Miami Metropolitan area, those being:

- Real estate Miami (the main search query)
- Real estate for sale Miami
- real estate fort Lauderdale
- real estate Hialeah
- real estate west palm beach

We expect that a higher number of search queries will generally put upward pressure on real estate prices in the short term. Nonetheless, this might not be as easy as it sounds since we know an increase in search terms means that there is increased interest in the subject however it tells us nothing of the transaction the searcher is planning. In other terms he could be willing to search this particular term to buy a house in the designated area or to sell his house. Thus we could expect a situation

where we have a heightened number of search queries but a future price that goes down because the majority of interested parties were aiming to sell their real estate good.

Other variables could have been added to increase the precision of the model, like Demographics data and crime. However these were not available in monthly/quarterly frequencies and thus contradict our aim to create a model efficient for short term predicting. The data will be used in its base form in our models.

Dependent Variable	**Frequency**	**Data Base**
Single Family Homes Returns - Miami Metropolitan Area	Monthly	Zillow
Explanatory Variables		
Macroeconomic Variables		
Unemployment (Miami Metropolitan Area)	Monthly	FRED
Fixed Mortgage Rate 30 Years (US)	Monthly	FRED
Custom Affordability Variable (Log(House Price/Real Disposable Income))	Monthly	FRED
University of Michigan: Consumer Confidence (US)	Monthly	FRED
Consumer Price Index - Urban Consumers: All Items in U.S. City Average	Monthly	FRED
Currency in Circulation (US)	Yearly	FRED
GDP Growth Nominal (US)	Yearly	FRED
S&P500 Returns	Monthly	WRDS
Real Estate Variables		
New Private Housing Units Building Permits (Miami Metropolitan Area)	Monthly	FRED
All Employees: Construction in Miami-Fort Lauderdale	Monthly	FRED
Delinquency Rates on Real Estate Loans (US)	Quarterly	FRED
Domestic Banks Tightening Standards - Commercial Real Estate Loans	Quarterly	FRED
Google Trends		
Search Queries	Monthly	Google Trends

Table 1: Variables and their Characteristics

Dependent Variable	**Form**
Single Family Homes Returns - Miami Metropolitan Area	Log returns - Monthly
Explanatory Variables	
Macroeconomic Variables	
Unemployment (Miami Metropolitan Area)	Base - Percentage
Fixed Mortgage Rate 30 Years (US)	Base - Percentage
Custom Affordability Variable (Log(House Price/Real Disposable Income))	Log returns (Custom Index)
University of Michigan: Consumer Confidence (US)	Base - Nominal
Consumer Price Index - Urban Consumers: All Items in U.S. City Average	Base - Nominal
Currency in Circulation (US)	Log Returns - Yearly
GDP Growth Nominal (US)	Log Returns - Yearly
S&P500 Returns	Log returns - Monthly
Real Estate Variables	
New Private Housing Units Building Permits (Miami Metropolitan Area)	Base - Nominal
All Employees: Construction in Miami-Fort Lauderdale	Base - Nominal
Delinquency Rates on Real Estate Loans (US)	Base - Percentage
Domestic Banks Tightening Standards - Commercial Real Estate Loans	Base - Percentage
Google Trends	
Search Queries	Base - Normalized Index

Table 1 Continued: Variables and their Characteristics

4. Methodology

In our goal to create a modern model for predicting real estate prices, we will use the gathered data and use different machine learning models to obtain the best predictions possible in a monthly timeframe. This model also has the aim of being as straightforward as possible, thus avoiding being over complicated. Ultimately, if the predictions of one or more models underperform, we may use machine learning techniques such as bagging to try and improve the performance. Python will be used for its simplicity and wide range of applications within financial work and data. This language has quickly risen to being one of the most popular in many fields including economics and finance. Resources on its use, as well as libraries providing extra resources/functions/models are widely available. This is important, as we will be using many functions and models from the Scikit-learn library which is the library used for machine learning in Python. It's also important to note that we believe that if this study is successful, the model could be applied practically, but with a secondary model predicting real estate prices at a micro level. Concretely, this model is aimed to predict the general trend in at a macro level, for a whole metropolitan area. The secondary model would evaluate characteristics at a micro level, such as the neighborhood, size of the house, number of rooms and etc. The model used in *Pai, Wang.* (2020) is a good example of such.

4.1 Data Considerations

As already mentioned previously, there is a 1 month time difference in the y and X variables, this is because these models are intended purely for predicting the future variation. In the modelling phase, the data will always be separated into 2 sets for each model, a training set to "train" the model and a test set where it will aim at predicting the y values. Subsequently, some of the data was altered from its base form using log growth and log returns. The Natural Logarithm of a particular data is used when we want to normalize the data, thus in basic form the data does not follow (or as much) a Gaussian distribution which works best with OLS and machine learning models in general. The distribution of the data was examined graphically in its base form, using the natural logarithm so that most data fits as closely as possible

to a normal distribution and take into account consistency in our approach. The distributions of all the explanatory variables are available in appendix 1. Just like our y variable, taking the S&P500 log returns is appropriate, since it gives the advantage mentioned above. Additionally, it also provides benefit in the form of the additivity property in mathematics, where log returns can be added together and then returned to their base form to find the exact growth. For the currency in Circulation and GDP growth the log returns formula was used as well, however on a longer span on time of 12 months as we believe sustained variations would impact the real estate market more significantly. Here is the formula for log returns:

$$Log\ Returns = LN(\frac{V_{t+1}}{V_t})$$

LN: Natural Logarithm

V: Value

4.2 Machine Learning Considerations

There is an important distinction in the world of machine learning where tasks are classified as supervised, unsupervised or reinforcement. Supervised models are named as such because they will be guided through most of the process. The person behind the models will add labeled data which is why they are less complex than unsupervised models. There are two main types of algorithms used in Supervised Learning, classification and regression. Classification algorithms are used for predicting actual numbers (integers) or probabilities of a specific event, while Regression is used for continuous numbers. In Unsupervised learning, data will not be labeled and the human intervention is lesser. The machine learns by itself as there is no training for the model. It achieves this through adapting and trying to find patterns. Finally, reinforcement learning utilizes the notion of rewards and uses algorithms to maximize these rewards in the forms or goals. As such they are more useful for predicting outcomes in dynamic environments. The machine learning models we will be using are all supervised models. This is due to multiple factors, supervised learning will be more efficient than unsupervised when the data is labelled and input/output are clearly defined. Moreover, they are quicker to implement than

unsupervised models who can take hours or days to find the patterns in the data. Lastly, supervised models are generally less complex which fits with our goal of straightforwardness. Concerning the sub categories of supervised learning models, we will be using Regression type algorithms since the predicted y value is in the form of a continuous number and thus classification types will not work in our case.

4.3 The Machine Learning Models

We will compare the results in 3 different machine learning models and the classic linear regression model (OLS). They are the following:

- Ordinary Least Square regression
- Support Vector Machine (Regression)
- Random Forest (Regressor)
- Multilayer Perceptron Neural Networks (Regressor)

Ordinary Least Square Regression: The OLS regression or commonly known as linear regression (multiple, depending on the number of variables) is one of the most used models in regression analysis. Even though it is not a machine learning model per se it can be considered as such with the addition of machine learning features like a training/testing set for example. Moreover, its popularity in regression analysis leads us to also use it as an added measure of comparison with the other models. The OLS regression aims at finding the relationship in a linear way between the explanatory variables and y. The assumed relationship is the following:

$$y_i = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \varepsilon_i$$

y: Dependent Variable

β_0: Intercept

β_n: Slope coefficient

n: Number of Independent Variables

X: Independent Variables

It finds this linear relationship using the least square method and minimizing the sum of squared residuals.

Support Vector Regression: Our first proper machine learning model is the Support Vector Regression (SVR), the regression version of the Support Vector Machine (SVM). In general, whether for classification or regression the Support Vector method will use the data to construct a hyperplane, the number of dimensions depending on the number of variables. The data is plotted on this hyperplane and the model then uses a specific Kernel function to create the SVM and find the patterns in the data, whether for classification or regression purposes. There are multiple kernel functions available and the results may differ substantially depending on which one is used. We will discuss this more in detail in the choice of model/parameter section. While finding the patterns for accurate predictions the algorithm will aim to minimize error. This can be shown mathematically as solving:

$$Min\frac{1}{2}||w||$$

Under the constraint of:

$$y_i - wx_i - b \leq \varepsilon$$

And:

$$wx_i + b - y_i \leq \varepsilon$$

w: Normal Vector
x: Training Sample
y: Target Value
Ɛ: Margin of error tolerance

Random Forest Regression

This is another common machine learning technique and part of the ensemble method. Therefore, random forest includes the use of bagging within its model, which helps to increase the accuracy and reduce risks of overfitting in the overall model. Random forest can be used for both classification and regression type problems and works by constructing multiple decision trees in contrast to the CART method which creates only one decision tree. Furthermore, owing to it being an ensemble method and with the regression version, it will take the average output of all the trees for its predictions. The algorithm uses the residual sum of squares and the following equation when determining if split will result in a sufficient decrease of the impurity:

$$\frac{N_t}{N} * (Gini\ Impurity - \frac{N_t Right}{N_t} * Right\ Impurity - \frac{N_t Left}{N_t} * Left\ Impurity$$

N: total number of samples

N_t: Number of samples at time t

Right/Left: Right and Left Nodes on the decision trees

Multilayer Perceptron Neural Networks

Finally, we have the multilayer perceptron neural network. Neural Networks depending on the data and model can be either supervised or unsupervised but in our case, just like the others, this model is part of supervised learning. A multilayer perceptron (MLP) as its name suggests is a form of neural networks that can have multiple layers of neurons. An MLP model will have at the very least 3 layers of data, the input layer, the hidden layer and output layer. It uses a backpropagation algorithm which sends forward the data of the inputs, through the hidden layers (using an activation function to find the patterns) and then gives an output values. After this, errors in the output are sent backwards towards the input layer. During this process the weights of input data will be modified as to reduce the errors. A significant advantage is its ability to handle both linear and non-linear problem sets and this

method finds usage in regressions, predictions and classifications tasks. Mathematically, the MLP will depend on multiple factors including the activation function. However, the activation function for a model with a one-level hidden layer is:

$$f(x) = g(\sum_{j=0}^{M} W_{kj}\, g\left(\sum_{i=0}^{d} W_{ji}\, x_i\right))$$

x_i: Input

W_{ji}: Input Layer Weights

W_{kj}: Hidden Layer Weight

4.4 Choice of Models

All the models used in our study were chosen for a specific purpose. Already mentioned previously, was that all of them were supervised and this was chosen because the data is labelled, and in this case, supervised machine learning models are more effective than unsupervised. Individually, the OLS regression is used in the majority of regression based studies and provides relatively good predictive capabilities. Thus, it forms a base model that can be used to compare the performance of the other models to. The support vector machine in its regression form has been used it many predictive studies and has a reputation for strong predictive capabilities. Moreover, it is more time consuming to implement on large data sets, but the size of our data here makes it much easier. Finally, it is a model that is less prone to outliers in the data since it focuses on the data close to the decision boundary. The random forest model was an obvious choice since it provides advantages over the normal decision tree model. It does this by generating multiple decision trees, thus increasing predictive accuracy and uses bagging which reduces the risk of overfitting. Lastly, we have the multilayer perceptron neural network, which was chosen because it has a strong capacity at solving non-linear problems, meaning that if there are non-linear patterns in our data, the model would be able to pick up on them.

5. Results

5.1 Choosing Parameters

The parameters of every model are an important consideration and can significantly change results. Globally, to achieve our goals we decided to choose the parameters which gave us the best results, but keeping in mind the bias/variance tradeoff, and/or made rational sense. For all the models the train/test split was made with the ratio of 80/20 and the random state was equal to zero so that we have the same values in the training sample and test sample. There can be usually a dozen or more parameters for each machine learning model, for the ones we don't mention this means that they were kept in default mode. Individually, the OLS regression didn't require parameter selection other than what was common to every other model. In the case of the SVR before choosing the parameters, the SVM model due to its method of creating a hyperplane, requires the use of scaling the data (making them all fit on the same scale). The main parameters to choose are the kernel, Epsilon (the size of tube in the hyperplane where no penalties are given), penalty parameter C and the Max number of iterations. Arguably, the first 2 will yield the biggest differences in results. The choice of Kernel goes to the Kernel that best fits the data, for us this was the radial basis function (RBF), the penalty parameter was kept in default. In general a lower Epsilon and a higher number of iterations will result in a higher precision, but this should be done without reducing the accuracy of the test set and thus have overfitting. This resulted in an Epsilon of 0.0009 and a maximum number of iterations of 300. The default settings on the Random Forest Regression are very effective for most problems and in our case required no modification. The most important parameter for predictive precision is the number of decision trees in our model. The default number is 100 trees, an increase of the number of trees will generally increase precision to a certain point and then stagnate past this point. In our model 100 trees achieved substantial precision, more than 100 proved to yield no gains and less than 100 the precision would gradually decrease. Other important parameters are to allow bootstrapping, criterion for splits, max number of splits and the minimum impurity for splits to occur. Lastly, the multilayer perceptron is also a model where scaling is

required since the MLP method is sensitive to feature scaling. It is also the model that has the most parameters, yet some of them are only used depending on which solver is chosen. Arguably, the most important parameters to choose, from a predictive point of view, will be the number of hidden layers, the number of neurons for each layer, the activation function (the function used in the hidden layer to determine the output), the solver (used for weight optimization of the parameters). Additionally, the maximum number of iterations and the alpha (penalty parameter) are important in our case. When the data is not too complex, it is generally agreed that one hidden layer is sufficient, which is what we used. For the number of neurons per layer the answer is also similar where the number of neurons will increase per the size of the data and its complexity, we achieved the best results with 500 neurons. The activation function that best matched our data was the logistic function and the limted BFGS (lbfgs) was the solver chosen due to being more effective on smaller datasets like ours. Finally, the max number of iterations is 200 and the alpha equal to 1e-10, a higher number of iterations and smaller alphas will increase the precision.

5.2 Result Metrics

To evaluate the performance of our models, we will be using different metrics. As we are not trying to explain the precise effect of the different explanatory variables, our result metric will focus on the predictive power, accuracy and overall explanatory power of the model. Thus we have the following metrics and their formulas:

R-Squared: $R^2 = 1 - \frac{RSS}{TSS}$

RSS: Sum of Squared Residuals

TSS: Total Sum of of Squares

Adjusted R-Squared: Adjusted $R^2 = 1 - \frac{(1-R^2)(N-1)}{N-p-1}$

p:Number of Predictors

n: Total Sample Size

Mean Squared Error (MSE): $MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - Predicted\ Y\ values_i)^2$

n: Total Sample Size

y_i: Observed Values

Root Mean Squared Error

(RMSE): $\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - Predicted\ Y\ values_i)^2}$

Mean Absolute Error(MAE): $MAE = \frac{\sum_{i=1}^{n}|Predicted\ Y\ values_i - y_i|}{n}$

n: Total Sample Size

y_i: Observed Values

Mean Absolute Error Percentage (MAPE):

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Predicted\ Y\ values_i - y_i}{y_i}\right|$$

Variation Hit Rate (VHR):

$$VHR = \frac{1}{n}\sum_{i=1}^{n} Correctly\ Predicted\ Direction\ of\ Variations$$

5.3 Model Performance

Below we have provided a table with the results of all our models using the metrics we have mentioned previously. Graphical result comparison can be found in appendix 2.

	OLS Regression	Support Vector Regression	Random Forest Regressor
R-Squared	0.83260	0.96665	0.96555
Adjusted R-Squared	0.74555	0.94930	0.94764
Mean Squared Error	0.00002	0.00000	0.00000
Root Mean Squared Error	0.00435	0.00194	0.00197
Mean Absolute Error	0.00346	0.00156	0.00150
Mean Absolute Percentage Error	0.71099	0.39942	0.40697
Variation Hit Rate	0.92308	0.92308	0.97436

Table 2: Model Performance and different metrics

	MLP Regressor	MLP Regressor Bagged
R-Squared	0.80004	0.80015
Adjusted R-Squared	0.69606	0.69622
Mean Squared Error	0.00002	0.00002
Root Mean Squared Error	0.00475	0.00475
Mean Absolute Error	0.00400	0.00364
Mean Absolute Percentage Error	0.95013	0.98378
Variation Hit Rate	0.92308	0.89744

Table 2 Continued: Model Performance and different metrics

Our first comparison metric is the R squared, which is used here to see the quality of the fit of the data in the model and the explanatory power of the global model. We can see some large differences between the models. The OLS regression, our base model, has a high R squared, but is significantly lower than the SVR and Random Forest which both have R squares over 0.95. Surprisingly, the MLP whether with bagging or in its normal state performed less well on this metric than the basic OLS. The MLP was the only model where we believed the results were too low and decided to use bagging which is known for generally improving the performance of the model as well as a good tool used in cases of overfitting. However, bagging resulted in very little change to the model and even resulted in a slight drop in the VHR. The adjusted R squared is a less important metric for our use, it works by giving a penalty to added variables, a major decrease in the adjusted R squared relative to the R square suggests that for the concerned models, the added explanatory variables don't add much more to the precision. Furthermore, it can be a sign of an issue of the model with multicollinearity, which we believe is present between some of our variables but will be discussed later. The OLS and both MLP models

experienced this significant drop in the adjusted R squared, which would suggest that if we were to use those models we would need further research and probably either remove some independent variables or replace them. However, the adjusted R-squares of both the SVR and Random Forest Regressor had little variations. This shows that most of our variables produced little penalties in the fit. Mean Squared Error or MSE was calculated for the purpose of having the Root Mean Squared Error or RMSE, along with the Mean Absolute Error, which are 3 of the most common metrics used to evaluate machine learning models. Although, they all have the goal of calculating the error size of the predicted y versus the real y value, they do it in a different manner. The MSE and RMSE will give higher penalties to outliers and MAE will not have this sort of penalty. The smaller those metrics are, the better it is. We believed it would be better to use the 3 of them since in our dataset we did have outliers during the 2008 crisis period which severely impacted the real estate sector. Relative to the predicted y values, the RMSE, MSE and MAE for the OLS regression and both MLP models they are high, again the SVR and RFR here have much better results, both nearing 0.0015 versus 0.0035 for the others in the MAE, thus twice as high of an error. Arguably, the most important metrics for us will be the Mean Absolute Percentage Error (MAPE) and the Variation hit rate (VHR). The MAPE is just the MAE divided by the average predicted y value and to the reader not aware of the exact y values is easier to interpret and easier to compare to other models. Finally, the VHR is simply the number of times the model predicted correctly the direction of the variation (prices appreciated or depreciation) divided by the size of the test set. The results for the MAPE naturally replicate those for the MAE where the SVR and RFR performed much better than others and here have a MAPE of roughly 40%, while the OLS stands at over 70% and both MLP models are around 90%! According to *Lewis.*(1982) a MAPE below 20% gives good predictions and one beneath 10% gives highly accurate predictions. Hence considering these numbers, none of our models have sufficient accuracy to be classified as good predictions. This is true even for our SVR and RFR, who have superior performances across all metrics to the OLS and MLP models. Furthermore, some of the studies we have covered in our literature review also obtain very low MAPE when predicting prices, *Pai.*(2020) get less than

1% MAPE in some of his machine learning models. *Ceh, Kilibarda, Lisec, Bajat*. (2018), obtained roughly a MAPE of 7% using his Random Forest multiple in Ljubljana and *D Sun*.(2014), achieved less than 20% in some of his SVR models. Yet, all these studies mentioned either try to predict the nominal price of real estate or its simple log form. Additionally, even if they use smaller timeframes they are predicting using a micro model usually looking at physical characteristics of the house, neighborhood statistics, crime rates which gives them much more data since it is taken from transaction data sources. Taking into consideration all of this we believe it explains in part why our model doesn't yield the same accuracy as the others and that it would be higher if we were predicting nominal prices. However, since the start we have explained that we intended this model to be used along with one of those micro models. Moreover, we believe that even though the predicting accuracy of the variation is not perfect, the predicting of the direction of the variation (shown by the VHR) is very high for all models. All models, achieved roughly 90% VHR or higher with the Random Forest performing best with over 97% and only 1 error in the test set out of a sample size of 39. Overall, other than the MLP models all our other machine learning models performed much better than our base model the OLS regression in most or all metrics. Even so, they are not performing as well as anticipated in the accuracy of their predictions and practically we don't recommend the use of any on its own. But, we are confident to say that we have partially achieved our goal since the SVR and RFR performed best and are accurate enough in our view to be used in combination with a real estate predicting micro model. They would perform best in timing the real estate market since they have a high accuracy in predicting the direction of the variations.

5.4 Potential Issues with the Models

One of the most common issues with machine learnings models is with underfitting/overfitting. This is linked to the bias/variance tradeoff, where our ideal is having a model with low variance and low bias. Practically, at some point if we try to reduce one of those excessively, it will generally increase the other, leading to overfitting or underfitting. Underfitting is when a model will have low variance but very high bias, thus it fails to recognize patterns in the data and learn much from it,

leading to poor precision. Overfitting is the opposite, the model will have high variance, but very low bias, therefore it will be highly accurate but it will be so accurate that it captures noise in the data and when used with a different set of data will perform with much less precision. The easiest method to check for those problems is comparing the precision of the fit in the test dataset with that of the training dataset.

	OLS Regression	**Support Vector Regression**	**Random Forest Regressor**
Train Score	0.91803	0.99225	0.99623
Test Score	0.83260	0.96665	0.96555

Table 3: Model training/data sets and their precision scores (R Squared)

	MLP Regressor	**MLP Regressor Bagged**
Train Score	0.83759	0.85675
Test Score	0.80004	0.80015

Table 3 Continued: Model training/data sets and their precision scores (R Squared)

With R squares this high, all over 0.8 and some above 0.95, underfitting is not an issue in our models. We can see that other than the OLS regression, our machine learning models very likely don't have an overfitting problem, with differences in train/test scores contained within the 0.03-0.06 interval. Even for the OLS, the gap between training and test sets is less than 0.1. This suggests that there is likely some overfitting but not to a high degree, further research would be needed to determine to what degree overfitting is present in the OLS regression. However, our focus here is on the machine learning models.

When looking at regression analysis, a very important matter is multicollinearity between variables. This is because in some models, like the OLS, multicollinearity can have a negative impact on the model when explanatory variables are too closely

correlated and can be predicted to a high degree by each other. This can lead to high discrepancies in R-squared values versus adjusted R-Square values. We expect a large number of our variables to have such a relationship, as economic variables have high correlations to each other. Moreover, multicollinearity can make it much harder to interpret impacts of different explanatory variables on the dependent variable. However, most machine learning models are very good at picking up on multicollinearity and usually in the worst case scenario, the addition of a new variable with a very high correlation to other explanatory variables will result in no gain in precision but no penalty on the predictability prowess of the model. Additionally, since our research does not focus on the explanatory power of a single variable, but rather on using modern tools and modern data to obtain the highest accuracy in predictions, multicollinearity is not a problem for this study.

6. Discussion

6.1 Limitations and Covid

In the results, we have discussed the limitations and shortcomings of the predictive performance of our models. Yet, as mentioned previously we have achieved partially our objective with the SVR and RFR models as they have very high VHR. Therefore, we believe they can be used effectively in combination with a secondary model. Nevertheless, Covid was a black swan event for financial markets in general including the real estate sector. Many, extraordinary measures were taken by governments around the world to curb the spread of the virus as well as the economic downfall. Some of these measures like the overwhelming influx of currency into circulation in many countries including the US, could be captured by our model, the tightening of commercial real estate loans is also another example. However, many unique measures like relocation of remote workers to rural areas, closing of businesses like bars/clubs, increased backing of the US government on certain loans, legal restrictions on removing tenants who could not afford to pay their rent, have impacted the real estate sector in ways never seen before and would almost certainly not be captured by our model. Thus we would expect the accuracy of our models to drop significantly. All of this is expected, even though our time frame goes through the 2008 financial crisis which has some outliers present. We expect a separate study on the impact of Covid on the real estate market would be necessary to create models capable of predicting real estate prices in this special period.

6.2 Specifics of the Real Estate Market

Another important perspective to discuss are the specifics of the real estate market. The real estate market is quite different from other financial markets. Firstly, it is private and not public, meaning information is not as widely available as assets in other markets, like fixed income or equities. Our models used data that is relatively easy to find, but for some of the data used in the micro models discussed in the literature review, this is much harder and requires more work. For some smaller areas and/or emerging markets, this data might be extremely difficult to find. Other issues include liquidity which is lower since real estate assets are private and not public.

Sentimental value also occurs, certain real estate assets have sentimental value to their owners and owing to its individual nature would be very hard to measure. Globally, the real estate market is not as efficient as other markets like the Forex, equity or bond markets and this can add further difficulty in predictions.

6.3 Future Research

We have discussed already a few topics where future research could be done like a study on the effects of Covid on the real estate markets and changing some of our explanatory variables in the hope of increasing the predictive accuracy. Finding data for the prediction of trends in the real estate market could make a significant impact in creating a prediction model for Covid. Our study was focused on the predictive power of our models and not on the individual independent variables, further research into their impact on real estate fluctuations is possible. Another possible route of improvement for someone aiming at improving the predictive power of our models would be to use the variance inflation factor (VIF). The goal would be to replace the variables with the highest multicollinearity with new ones. As previously discussed, multicollinearity is not an issue in general for predictive accuracy in machine learning. However, if they don't yield any penalty, they also don't yield much benefit and also make the impact of other variables harder to interpret. Lastly, there are also ways of trying to improve the precision of the models and/or expanding our approach in the machine learning field. This includes utilizing other machine learning models like regular decision trees, k Nearest Neighbor, other forms of Neural Network and Utilizing other techniques than bagging to improve the performance of the models like boosting or stacking.

7. Conclusion

Globally, our study aimed at achieving a high accuracy in predicting monthly real estate variations using as example the Miami Metropolitan Area. All of this was to be achieved utilizing modern tools such as machine learning and modern data such as google trends. We used 13 different explanatory variables in the period from 2004 to early 2020 and 3 different machine learning models in addition to the OLS regression, used as a base for comparison. We obtained mixed results as the SVR and RFR achieved much better results than the OLS, while the MLP both in its simple state and using bagging achieved a similar performance to the OLS. But comparisons with other studies are complicated since the vast majority aim at predicting the nominal prices of real estate assets and not the variations. The explanatory power of our models, in the form of the R Squared, was high or very high depending on the model. Yet, the predictive power of our models didn't achieve a high level of predictive accuracy in terms of the precision of the variation. Lastly, the models achieved a very high VHR, which indicates that they are quite effective at predicting the direction of the variation. In this regard, we believe they would be useful practically when applied in combination with a model that predicts real estate prices using *micro* data like physical characteristics of the asset or neighborhood statistics. Finally, there are other limitations to take into consideration, the main one being the applicability of such a model during the Covid period. This is followed by characteristics which are specific to the real estate market and future research to be possible in these areas.

8. References and Related Works

- Pai, P.F, Wang, W.C 2020, '*Using Machine Learning Models and Actual Transaction Data for Predicting Real Estate Price'*, Department of Information Management, National Chi Nan University, DOI: 10.3390/app10175832
- Wu, L, Brynjolfsson, E 2015, *The Future of Prediction: How Google Searches Foreshadow Housing Prices and Sales*, University of Chicago Press, pp. 89–118.
- Bahia, I.S.H 2013, *A Data Mining Model by Using ANN for Predicting Real Estate Market: Comparative Study*, International Journal of Intelligence Science, pp.162-169, DOI:10.4236/ijis.2013.34017.
- Tabales, N 2013, *Artificial Neural Networks for Predicting Real Estate Prices,* Revista De ,Metodos Cuantitativos Para la Economia y la Empresa, pp. 29-44.
- Sun, D, Du, Y, Xu, W, Zuo, M, Zhang, C, Zhou, J 2014, *Combining Online News Articles and Web Search to Predict the Fluctuation of Real Estate Market in Big Data Context,* Pacific Asia Journal of the Association for Information Systems, Vol. 6, pp.19-37, DOI: 10.17705/1pais.06403.
- Meulen, P 2014, *Forecasting real estate prices in Germany: the role of consumer confidence*, Journal of Property Research, Vol. 31, pp. 244-263, DOI: 10.1080/09599916.2014.940059.
- Salnikov, VA, Mikheeva, V.A 2018, *Models for Predicting Prices in the Moscow Residential Real Estate Market*, Institute of Economic Forecasting, Russian Academy of Sciences, Vol.29, pp.94-101 DOI: 10.1134/S1075700718010136.
- Ludvingson, S.C 2004, *Consumer Confidence and Consumer Spending,* Journal of Economic Perspectives, Vol.2, pp.29-50.
- Gawlik, E, Kabaria, H, Kaur, S 2011, *Predicting tourism trends with Google Insights.*

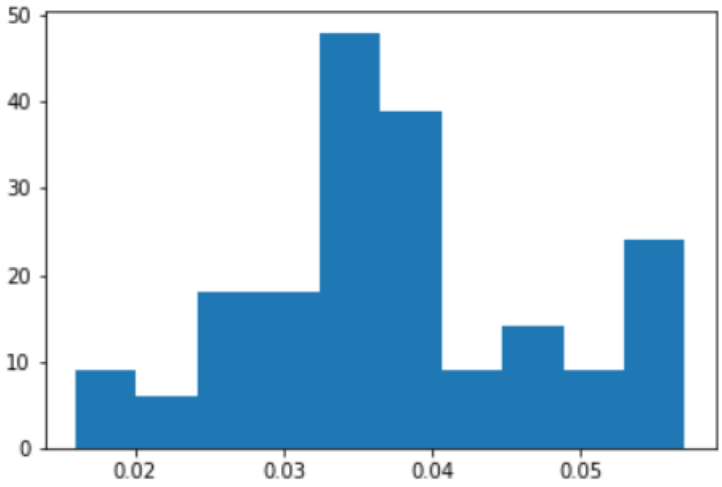
- Prado-Román, C, Gómez-Martínez, R, Orden-Cruz, C 2020, *Google Trends as a Predictor of Presidential Elections: The United State Versus Canada*, SAGE, American Behavioural Scientist, Vol. 65, pp. 666-680, DOI: 10.1177/0002764220975067.
- Swamynathan, M 2017, *Mastering Machine Learning with Python in Six Step: A Practical Implementation Guide to Predictive Data Analytics Using Python*, Apress, New York.
- Sarkar, D, Bali, R, Sharma, T 2017, *Practical Machine Learning with Python: A Problem-Solver's Guide to Building Real-World Intelligent Systems*, Apress, New York.
- Lewis, C.D 1982, *Industrial and Business Forecasting Methods; A Practical Guide to Exponential Smoothing and Curve Fitting*, Butterworths Publishing, London.
- Wharton Research Data Services, n.d, *(Daily) S&P 500 Index*, Viewed 21 May 2021, https://wrds-www.wharton.upenn.edu/pages/get-data/compustat-capital-iq-standard-poors/.
- Zillow, n.d, *Housing Data*, Viewed 1 May 2021, https://www.zillow.com/research/data/.
- Federal Research Economic Data, n.d, Viewed 20 June 2021, https://fred.stlouisfed.org/.
- Du Boisberranger, J, Van den Bossche, J, Estève, L, Fan, T.J, Gramfort, A, Grisel, O, Halchenko, Y, Hug, N, Jalali, A, Lemaitre, G, Lorentzen, C, Metzen, J.H, Mueller, A, Niculae, V, Nothman, J, Qin, H, Thirion, B, Dupré, la Tour T, Varoquaux, G, Varoquaux, N, Yurchak, R, n.d, *Scikit-Learn Machine Learning in Python*, Viewed from 5 June – 30 June 2021, https://scikit-learn.org/.

Appendices

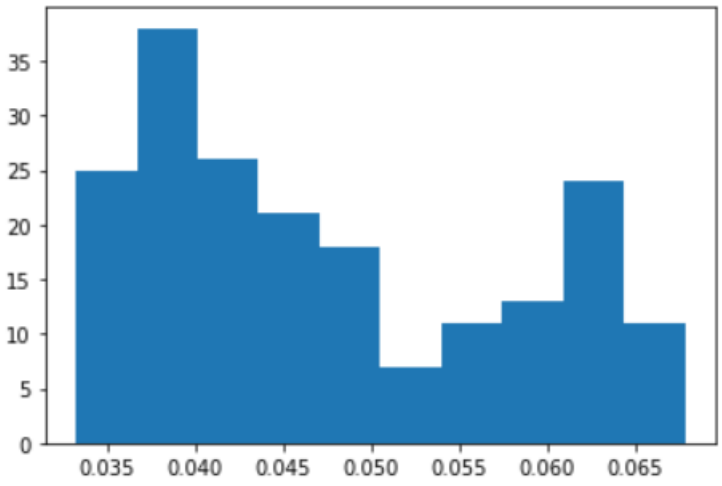
Appendix 1: Histogram Distributions of Explanatory Variables in the model

For all the histograms, the Y axis reprensents frequency and the X axis values.

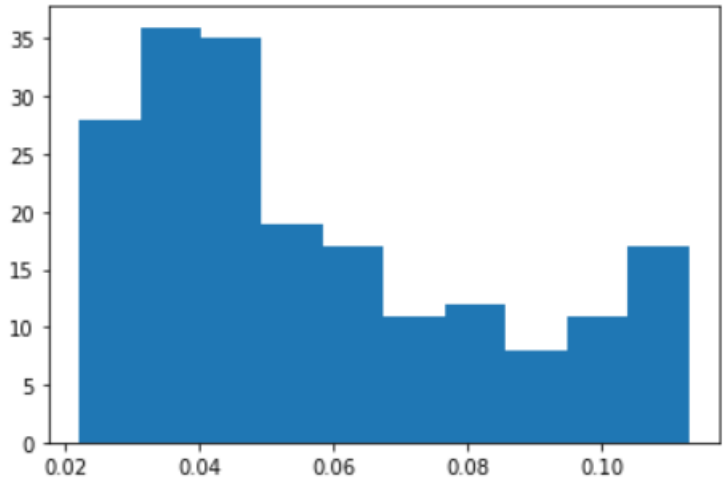
Graphical Distribution of GDP (US) with Log Returns (12 months) – X1



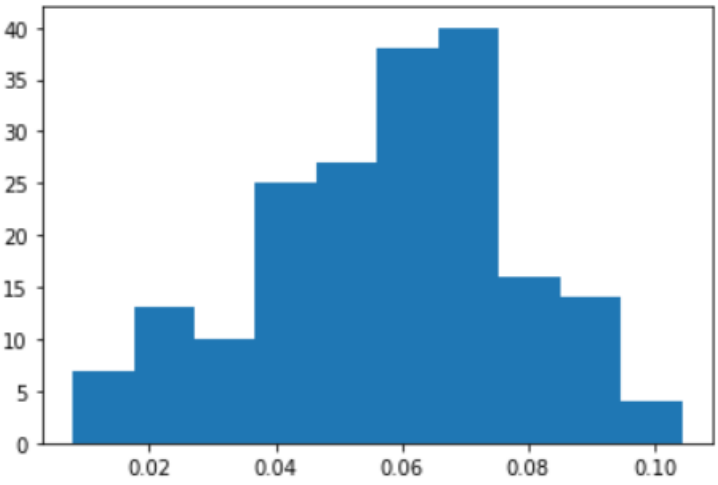
Graphical Distribution of 30 year Fixed Mortgage (%) – X2



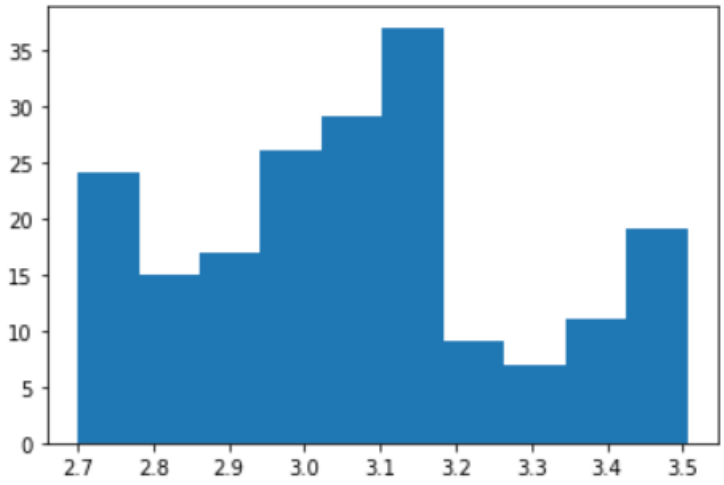
Graphical Distribution of Unemployement - Miami Metropolitan area (%) – X3



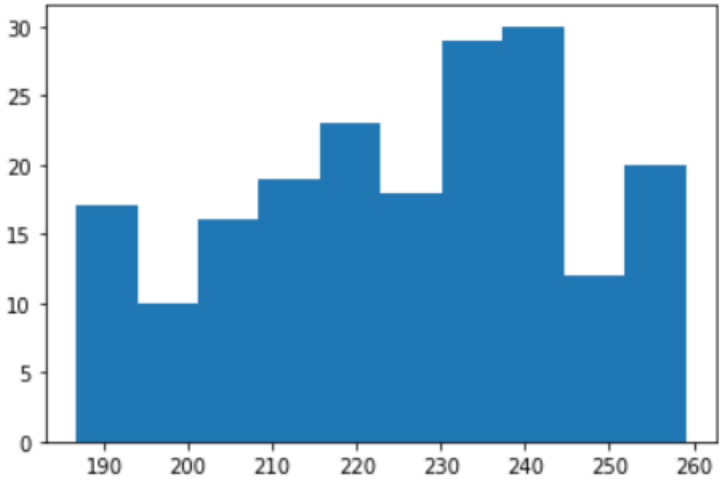
Graphical Distribution of Currency in Circulation (US) with Log Returns (12 months) – X4



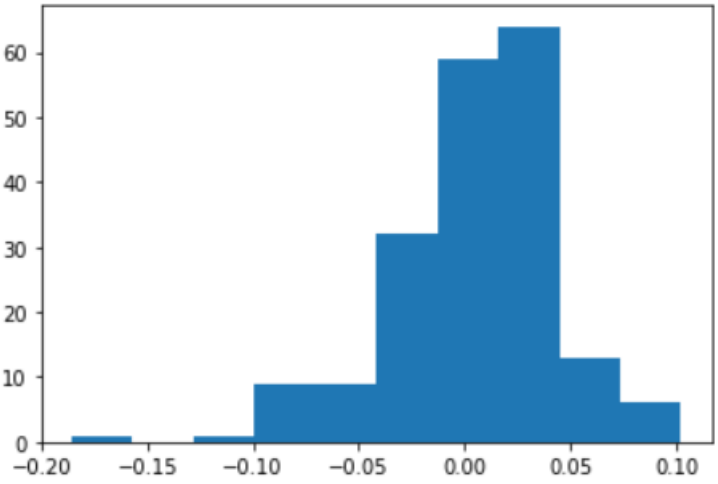
Graphical Distribution of Custom Affordability Index (Log(Single Family Home/Real Disposable Income)) – X5



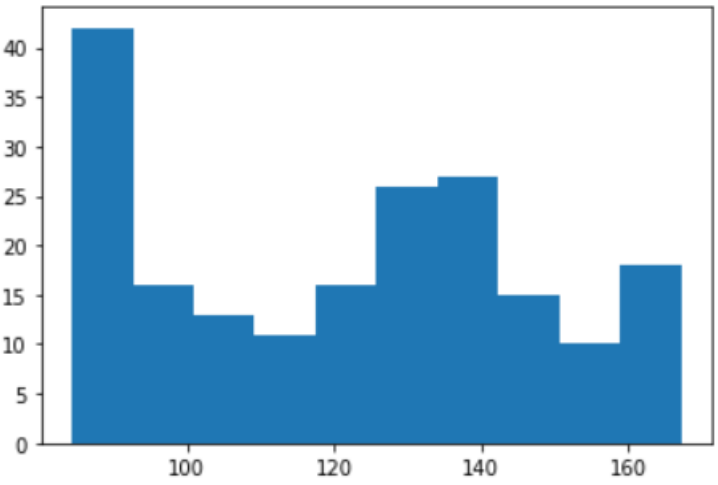
Graphical Distribution of Inflation Proxy - Urban Consumers – X6



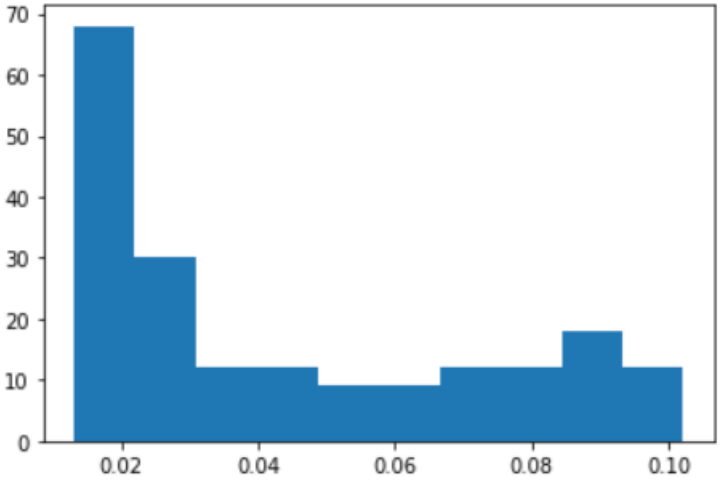
Graphical Distribution of S&P 500 returns - Log Returns– X7



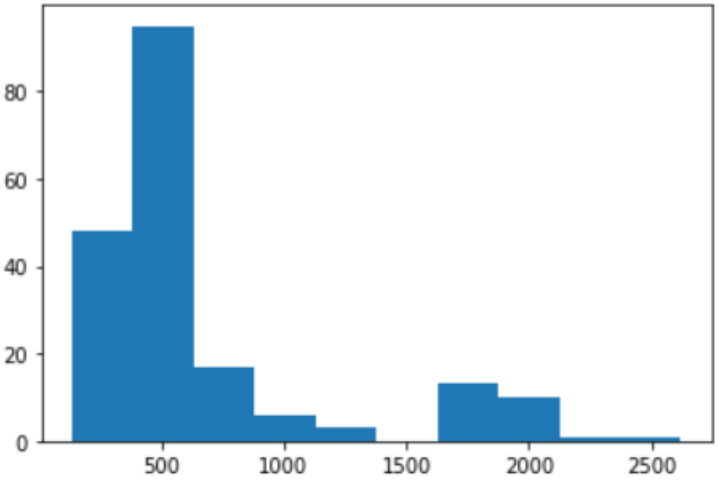
Graphical Distribution of All Employees: Construction Sector Miami Metropolitan Area – X8



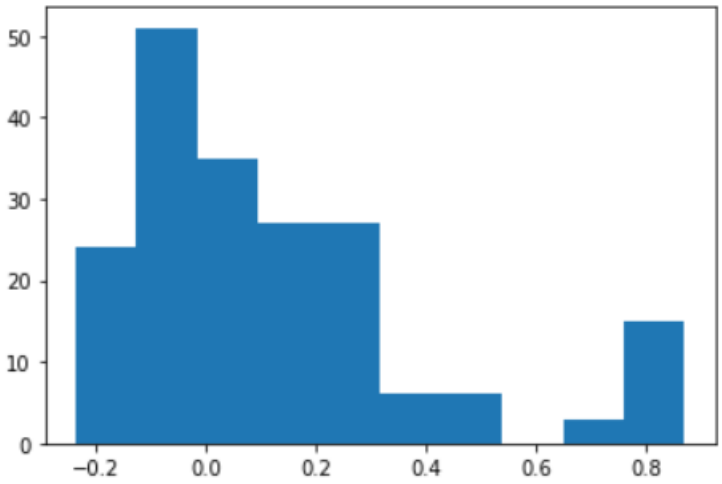
Graphical Distribution of Delinquency Rates on Real Estate Loans US (%) – X9



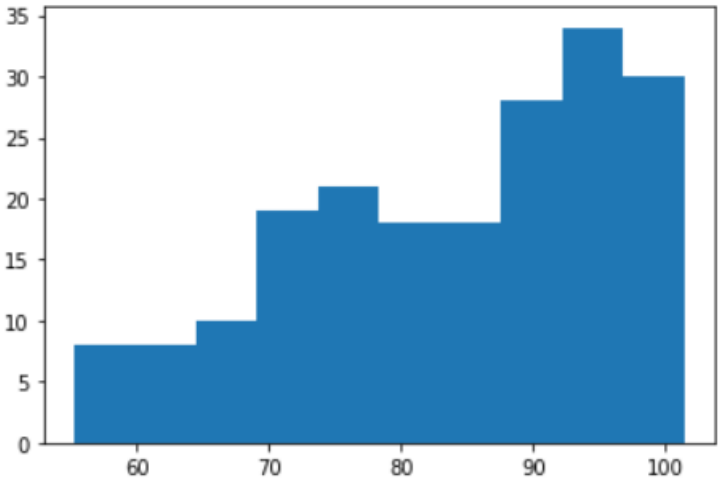
Graphical Distribution of New Private Housing Units Authorized by Building Permits – Miami Metropolitan Area – X10



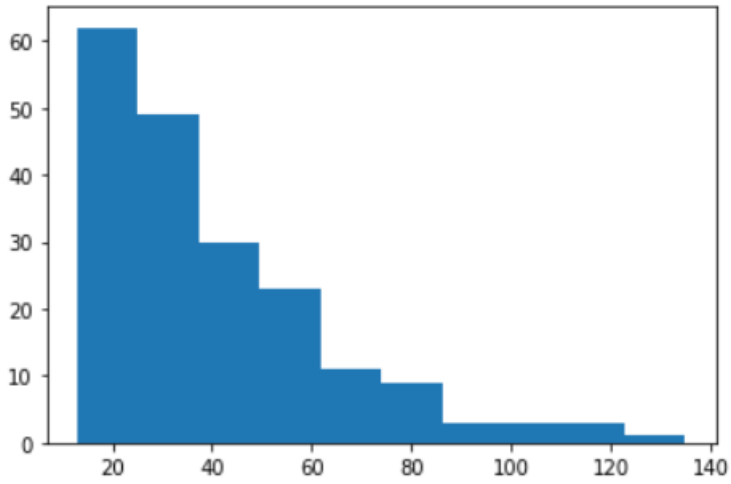
Graphical Distribution of the Net percentage of Domestic Banks Tightening Standards - Commercial Real Estate Loans (US) – X11



Graphical Distribution of University of Michigan Consumer Confidence – X12

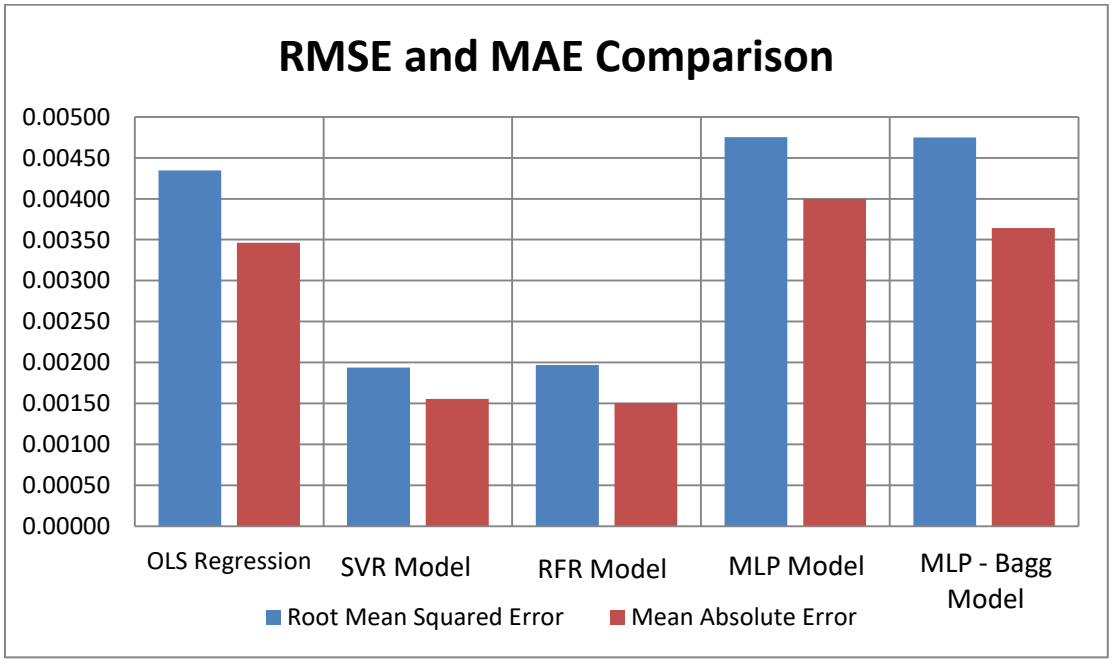


Graphical Distribution of the Google Trend Search Queries – X13

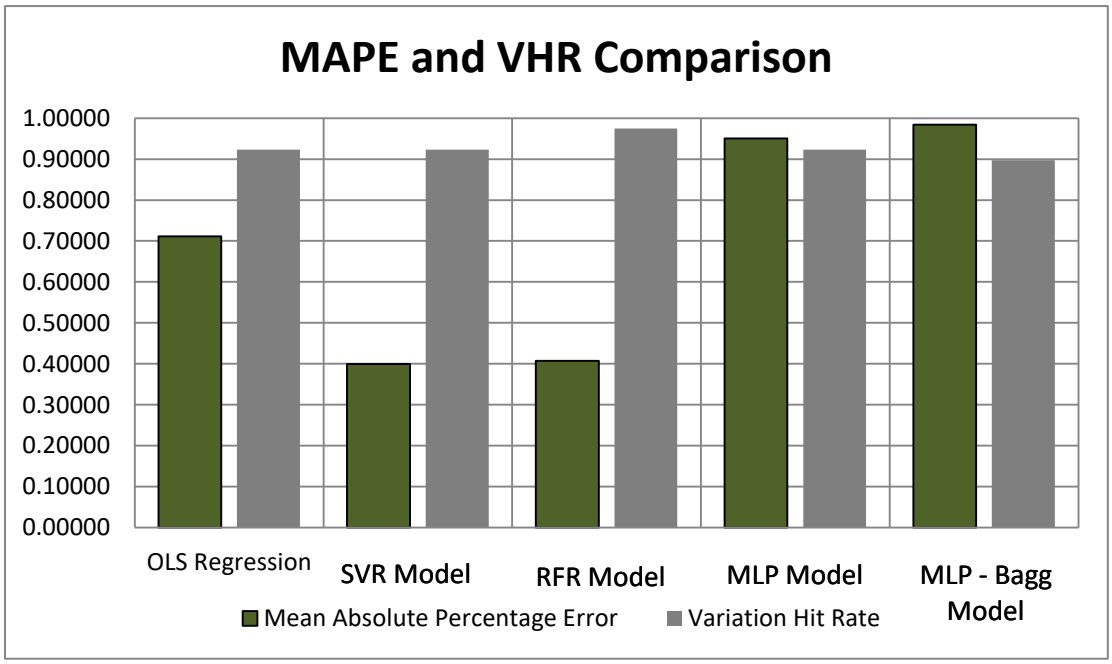


Appendix 2: Result Metric Comparisons

Comparison of Root Mean Squared Error and Mean Absolute Error across Models



Comparison of the Mean Absolute Percentage Error and Variation Hit Rate across Models



Python Code

```
#Importing all the necessary libraries, functions, models we will use

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as colors
import statsmodels.api as sm
import math
from sklearn.utils import resample
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import scale
from sklearn.svm import SVR
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import confusion_matrix, classification_report, accuracy_score
```

```
from sklearn.decomposition import PCA
from sklearn import metrics
from sklearn import tree
from sklearn import linear_model
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, mean_squared_log_error,
mean_absolute_error
from sklearn.ensemble import BaggingRegressor

#Importing the labelled data from prepared Excel files

df=pd.read_excel(r'C:\Users\Bradley Begaud\Documents\Devoir\4 - BI Business
School\Thesis\Thesis Data - Predictive Data.xlsx')
df2=pd.read_excel(r'C:\Users\Bradley Begaud\Documents\Devoir\4 - BI Business
School\Thesis\Data - Base Form.xlsx')

#Graphical Analysis of the Distributions
#Showing the LN version of the data when it is kept in its base form

plt.hist(df2.X1)

plt.hist(df.X1)

plt.hist(df2.X2)

log_X2=np.log(df.X2)
plt.hist(log_X2)

plt.hist(df2.X3)

log_X3=np.log(df.X3)
plt.hist(log_X3)

plt.hist(df2.X4)
```

```
plt.hist(df.X4)

plt.hist(df2.X5)

plt.hist(df.X5)

plt.hist(df2.X6)

log_X6=np.log(df2.X6)
plt.hist(log_X6)

plt.hist(df2.X7)

plt.hist(df.X7)

plt.hist(df2.X8)

log_X8=np.log(df2.X8)
plt.hist(log_X8)

plt.hist(df2.X9)

log_X9=np.log(df2.X9)
plt.hist(log_X9)

plt.hist(df2.X10)

log_X10=np.log(df2.X10)
plt.hist(log_X10)

plt.hist(df2.X11)

plt.hist(df2.X12)

log_X12=np.log(df2.X12)
plt.hist(log_X12)

plt.hist(df2.X13)
```

```
log_X13=np.log(df2.X13)
plt.hist(log_X13)

X=df[['X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13']]
y=df.y

#OLS Regression Model

reg=linear_model.LinearRegression()

#Divinding the model into training and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

#fitting the model

reg.fit(X_train,y_train)

print(reg.coef_)
print(reg.intercept_)

#Calculating the R2 and Adjusted R2

print("Training set score: %f" % reg.score(X_train, y_train))
print("Test set score: %f" % reg.score(X_test, y_test))

adj_R2_train=1-(1-(reg.score(X_train,y_train)))*(155-1)/(155-13-1)
print(adj_R2_train)
adj_R2_test=1-(1-(reg.score(X_test,y_test)))*(39-1)/(39-13-1)
print(adj_R2_test)

#Calculating Predicted y for the model

y_predicted_OLS=reg.predict(X_test)

print(y_test)

print(y_predicted_OLS)
```

```
#Calculating all our other result metrics except VHR which is done by Excel

print("MSE:"+str(mean_squared_error(y_predicted_OLS,y_test)))
print("RMSE:"+str(np.sqrt(mean_squared_error(y_predicted_OLS,y_test))))
print("MAE:"+str(mean_absolute_error(y_predicted_OLS,y_test)))
print("MAPE:"+str(np.mean(np.abs((y_test-y_predicted_OLS)/y_test))))

#Support Vector Regression Model

#Usage of scaling for the SVR

sc =StandardScaler()
sc.fit(X)
X = sc.transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

svr = SVR(kernel='rbf', epsilon=0.0009, max_iter=300)
svr.fit(X_train, y_train)

print("Training set score: %f" % svr.score(X_train, y_train))
print("Test set score: %f" % svr.score(X_test, y_test))

adj_R2_train=1-(1-(svr.score(X_train,y_train)))*(155-1)/(155-13-1)
print(adj_R2_train)
adj_R2_test=1-(1-(svr.score(X_test,y_test)))*(39-1)/(39-13-1)
print(adj_R2_test)

y_predicted_svr=svr.predict(X_test)

print(y_predicted_svr)

print("MSE:"+str(mean_squared_error(y_predicted_svr,y_test)))
print("RMSE:"+str(np.sqrt(mean_squared_error(y_predicted_svr,y_test))))
print("MAE:"+str(mean_absolute_error(y_predicted_svr,y_test)))
print("MAPE:"+str(np.mean(np.abs((y_test-y_predicted_svr)/y_test))))
```

```
#Random Forest Regressor Model

X=df[['X1','X2','X3','X4','X5','X6','X7','X8','X9','X10','X11','X12','X13']]
y=df.y

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

RFR_model=RandomForestRegressor(n_estimators=100)

RFR_model.fit(X_train, y_train)

print("Training set score: %f" % RFR_model.score(X_train, y_train))
print("Test set score: %f" % RFR_model.score(X_test, y_test))

adj_R2_train=1-(1-(RFR_model.score(X_train,y_train)))*(155-1)/(155-13-1)
print(adj_R2_train)
adj_R2_test=1-(1-(RFR_model.score(X_test,y_test)))*(39-1)/(39-13-1)
print(adj_R2_test)

y_predicted_RFR=RFR_model.predict(X_test)

print(y_predicted_RFR)

print("MSE:"+str(mean_squared_error(y_predicted_RFR,y_test)))
print("RMSE:"+str(np.sqrt(mean_squared_error(y_predicted_RFR,y_test))))
print("MAE:"+str(mean_absolute_error(y_predicted_RFR,y_test)))
print("MAPE:"+str(np.mean(np.abs((y_test-y_predicted_RFR)/y_test))))

#Multilayer Perceptron Neural Network Model

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

scaler = StandardScaler()
scaler.fit(X_train)
```

```
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

mlpr_model=MLPRegressor(solver='lbfgs', hidden_layer_sizes=(500,),
max_iter=200, activation='logistic', alpha=1e-10, random_state=0)

mlpr_model.fit(X_train_scaled,y_train)

print("Training set score: %f" % mlpr_model.score(X_train_scaled, y_train))
print("Test set score: %f" % mlpr_model.score(X_test_scaled, y_test))

adj_R2_train=1-(1-(mlpr_model.score(X_train_scaled,y_train)))*(155-1)/(155-13-1)
print(adj_R2_train)
adj_R2_test=1-(1-(mlpr_model.score(X_test_scaled,y_test)))*(39-1)/(39-13-1)
print(adj_R2_test)

y_predicted_mlpr=mlpr_model.predict(X_test_scaled)

print(y_predicted_mlpr)

print("MSE:"+str(mean_squared_error(y_predicted_mlpr,y_test)))
print("RMSE:"+str(np.sqrt(mean_squared_error(y_predicted_mlpr,y_test))))
print("MAE:"+str(mean_absolute_error(y_predicted_mlpr,y_test)))
print("MAPE:"+str(np.mean(np.abs((y_test-y_predicted_mlpr)/y_test))))

#Multilayer Perceptron Neural Network Model with Bagging

mlpr_model_bag=BaggingRegressor(base_estimator=mlpr_model,
random_state=0).fit(X_train_scaled,y_train)

print("Training set score: %f" % mlpr_model_bag.score(X_train_scaled, y_train))
print("Test set score: %f" % mlpr_model_bag.score(X_test_scaled, y_test))

adj_R2_train_bag=1-(1-(mlpr_model_bag.score(X_train_scaled,y_train)))*(155-
1)/(155-13-1)
print(adj_R2_train_bag)
adj_R2_test_bag=1-(1-(mlpr_model_bag.score(X_test_scaled,y_test)))*(39-1)/(39-
```

13-1)

```
print(adj_R2_test_bag)

y_predicted_mlprbag=mlpr_model_bag.predict(X_test_scaled)

print(y_predicted_mlprbag)

print("MSE:"+str(mean_squared_error(y_predicted_mlprbag,y_test)))
print("RMSE:"+str(np.sqrt(mean_squared_error(y_predicted_mlprbag,y_test))))
print("MAE:"+str(mean_absolute_error(y_predicted_mlprbag,y_test)))
print("MAPE:"+str(np.mean(np.abs((y_test-y_predicted_mlprbag)/y_test))))

#The End
```