

Housing Market Crash Prediction Using Machine Learning and Historical Data

A Project Report

Presented to

Chris Pollett

Thomas Austin

Mike Wu

Department of Computer Science

San José State University

In Partial Fulfillment

Of the Requirements for the Class

CS 298

By

Parnika De

May, 2020

Abstract

The 2008 housing crisis was caused by faulty banking policies and the use of credit derivatives of mortgages for investment purposes. In this project, we look into datasets that are the markers to a typical housing crisis. Using those data sets we build three machine learning techniques which are, Linear regression, Hidden Markov Model, and Long Short-Term Memory. After building the model we did a comparative study to show the prediction done by each model. The linear regression model did not predict a housing crisis, instead, it showed that house prices would be rising steadily and the R-squared score of the model is 0.76. The Hidden Markov Model predicted a fall in the house prices and the R-squared score for this model is 0.706. Lastly, the Long Short-Term Memory showed that the house price would fall briefly but would stabilize after that. Also, fall is not as sharp as what was predicted by the HMM model. The R-squared score for this model is 0.9, which is the highest among all other models. Although the R-squared score doesn't say how accurate a model it definitely says how closely a model fits the data. From our model R-square score the model that best fits the data was LSTM. As the dataset used in all the models are the same therefore it is safe to say the prediction made by LSTM is better than the other ones.

Index Terms — Subprime mortgage, credit derivatives, linear regression, hidden markov model, long short-term memory.

LIST OF FIGURES AND TABLES

Figure 1: Pictorial description of Mortgage backed Securities

Figure 2: Division of Tranches

Figure 3: Date format of different datasets that are used

Figure 4: Pre-processed dataset with changed Date format

Figure 5: Code snippet for Simple Linear Regression Calculation

Figure 6: Code snippet for calculation of Linear Regression using Least Squares method

Figure 7: Code snippet of using Sci-kit Learn to train and fit model and then make prediction

Figure 8: Hidden Markov Model

Figure 9: The state transition matrix

Figure 10: Observation matrix

Figure 11: Result of 30 observations of ring size

Figure 12: Code snippet of using Sci-kit Learn to train and fit data into HMM model

Figure 13: An LSTM network

Figure 14: Forget gate

Figure 15: Input Gate

Figure 16: Updating old state into new state

Figure 17: Output gate

Figure 18: Code snippet showing division of training and testing data and data conversion

Figure 19: Code snippet showing the building of LSTM model from Sequential model

Figure 20: Code snippet showing the fitting of LSTM model

Figure 21: Code snippet showing prediction for next 12 months

Figure 22: Plot of Price vs Date

Figure 23: Plot of Date vs Interest Rate

Figure 24: Plot of Date vs Houses Sold

Figure 25: Actual prices vs Predicted prices from Multiple linear regression

Figure 26: Price prediction by the regression model

Figure 27: House Prices actual vs predicted by HMM

Figure 28: Zoomed in graph to show the prediction

Figure 29: Graph of training, testing, and prediction using LSTM

Figure 30: Graph show extended Prediction by LSTM

Figure 31: Zoomed in graph showing extended Prediction by LSTM

Table 1: Sci-kit learn HMM model attribute description

Table 2: Model Comparison

Note: In this project all the figures are mine except Figure 13 – Figure 17. These are used as reference from [15] with permission from the author.

CONTENTS

1. Introduction	5
2. Background	7
2.1 Mortgage-backed Security	8
2.2 Causes of Housing Crisis	9
2.3 Prediction of Housing Crisis	11
3. Experimental Design	13
3.1 Datasets	13
3.2 Machine Learning Models	15
3.2.1 Linear Regression	16
3.2.2 Hidden Markov Model	20
3.2.3 Long Short-Term Memory	25
4. Results and Discussion	32
5. Conclusion and Future Work	41
6. References	42

1. INTRODUCTION

The United States of America has had many recessions in the past. The total number of recessions seen by the US is about 47, both major and minor. The 2008 recession was caused by faulty banking policies, mainly the sub-prime mortgage policy and selling the subprime mortgage securities in the market. Sub-prime mortgages are those mortgages that are given to people who do not qualify for prime mortgages. There can be many reasons for these viz. having low credit score, or, not being able to put a certain amount of down payment towards the house etc. This crisis caused a major ripple effect on the banks and the people. This crisis shut down many banks like Bear and Stearns and Lehmann Brothers. But it mostly affected people in the US, it drove many people homeless, jobless, and cashless for a long time. Crises like this can be avoided if we are aware of a bubble that would burst. Econometric and intelligent techniques can help us predict a bubble. Intelligent techniques especially those that analyze times series data can be used to predict the housing market crisis very accurately [1]. In this project we use econometric and intelligent techniques to predict the next housing crisis. Previous works that have been done in this area used Logistic regression and Back-propagation Neural Network (BPNN) [1][2]. Housing market prediction using Hidden Markov Model and Long Short-Term Memory has not been done yet, therefore in this project we would be using these techniques along with addition of Linear regression.

The 2008 housing crisis devastated the American economy. But before a recession comes there are markers to show trends that all housing recessions follow. The factors that led us to the 2008 recession [2]:

- 1) Inflated housing prices, that created a housing bubble

- 2) Relaxed banking policies that led to the high borrowing rate
- 3) Relaxed overall financial regulation i.e., how poorly the regulating bodies worked
- 4) Policies developed by banks to give more subprime mortgages

The mortgages were made more lucrative when the Federal Reserve Bank reduced the interest rates extremely low for short-term loans (ARM), along with easy availability of subprime mortgages. There were more and more people buying houses. As a result, the house prices started going up very quickly as there was a lot of demand for houses and the supply was not that high. Also, people thought that the housing market is the pillar for investment as the housing market had never crashed before. But everything changed in 2007-2009. The sub-prime loans were a huge risk the banks were taking and it all backfired when a lot of people started defaulting. The problem aggravated more when the banks started to take their houses and sell them in the already slow market. The house prices which were the highest a year ago, reached the rock-bottom. There were more houses for sale than there were buyers to buy. In this project we will look into few elements that are related to housing market to predict for the next year.

Now we discuss the organization of this report. In the next chapters we will look into the background of the financial institutions and how the change from the norm caused one of the biggest financial crises in the history of the US housing market. In Chapter 3 we will look in the algorithms that we have used to predict the housing market. Next in Chapter 4 we will look into the results from the machine learning models that we have built. Chapter 5 is the conclusion and future work.

2. BACKGROUND

In order to understand the financial crisis and how the banks played a major role in that we will be looking into the background of those policies. In this section we will also see how machine learning models can be used to help us predict crises like that of 2008 in advance by using relevant datasets.

The elements that were controlled by the banks in the US were the major elements that contributed to the financial recession of 2008. During this period the very thriving housing market was affected very badly which in turn affected the global economy. To understand how the banking system created havoc in the US, it is necessary to look at how the housing market was in the pre-recession period. The housing market was doing well before the 2008 recession. People could get sub-prime mortgages without any substantial credit score so more people could now afford houses. Therefore, there was an initial boom in the housing market. Housing prices were rising, as the market was very competitive. There were more buyers than there was a supply of houses. People (investors) could also invest in the housing market even without buying a house through Mortgage-Backed Securities (MBS). An MBS is a type of asset-based derivative security that derives its value from the underlying asset, the mortgages.

In the old days, there was no concept of these securities that were tied to mortgages. Buying houses did not have too many layers under them. If people had money, they would buy a house all cash and if they did not then they would have to get a mortgage from a bank to buy their house. These banks or credit unions had very strict lending rules and it was almost impossible for people with low credit history to get mortgages. But as the risks were low there were low mortgages that were given out and also the interest that was earned by the banks was also very low. This was pre-1970's.

In the 1970s the dollar value inflated rapidly as the then President of the US declared that the US dollar would not be tied to the gold standards going forward. This policy led banks to lose all the assets that they had as they were not being able to match the interest that was paid by the money market. This resulted in losing the deposits they had to give out loans to people. So, the banks were not being able to make profits. To help the banks from this bad situation the Congress then passed an act that could give the banks the liberty to raise interest rates on mortgages and also to lower the quality of the mortgage to make short-term profits.

During the early 2000s after the dot-com crisis, it was thought that the housing market was the sturdiest market as the housing prices increased throughout this crisis. Therefore, people started investing more money in the housing market. Investors who were not buying houses were investing in the housing market through MBS. The investors of MBS receive periodic payments just like other bonds.

2.1 MORTGAGE BACKED SECURITY

“Mortgage-backed securities (MBS) are debt obligations that represent claims to the cash flows from pools of mortgage loans, most commonly on residential property. Mortgage loans are purchased from banks, mortgage companies, and other originators and then assembled into pools by a governmental, quasi-governmental, or private entity. The entity then issues securities that represent claims on the principal and interest payments made by borrowers on the loans in the pool, a process known as securitization.”

-US Securities and Exchange Commission

A bank first makes a mortgage and it then sells those mortgages to investment banks to make more money and that money is used to make more mortgages. The investment banks then bundle this pool of mortgages into securities. These securities are called mortgage-backed

securities. MBSs are investments that are backed by the mortgages. After making them into securities it is then sold to investors. The investors receive their regular payments when people buying mortgage loans pay towards their monthly mortgages.

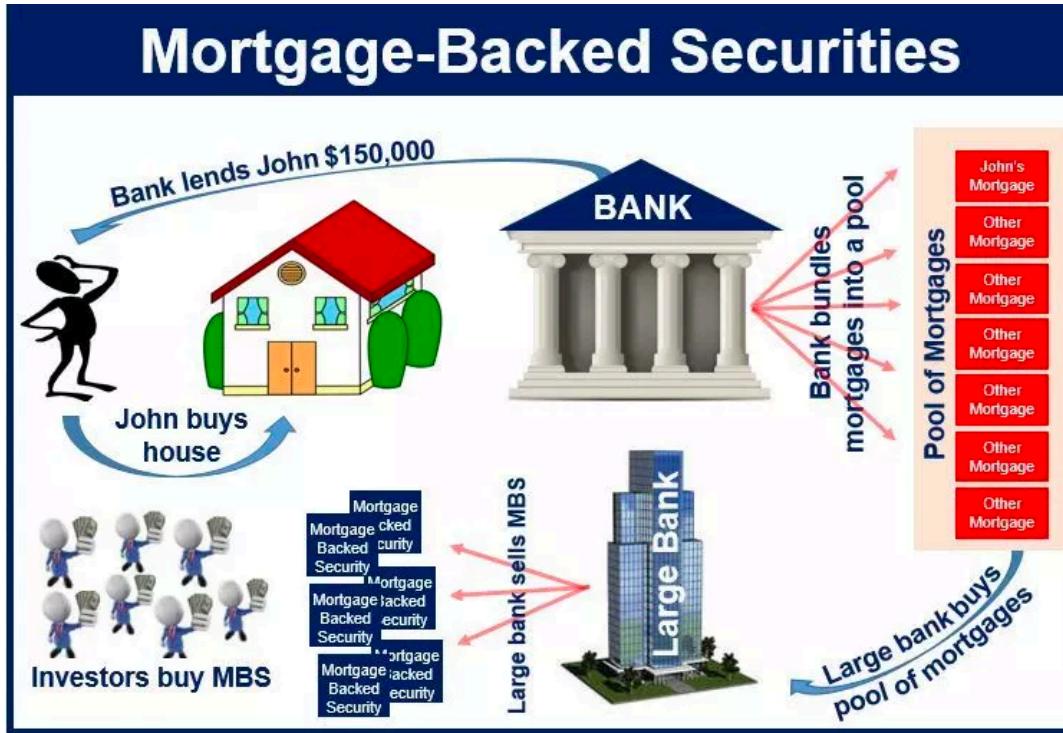


Fig 1: Pictorial description of Mortgage backed Securities

2.2 CAUSES OF HOUSING CRISIS

In the 2000s the MBS investments started getting sophisticated. Investment banks started slicing MBS's into tranches. The banks were also becoming greedy and they started giving out low-quality mortgages to people with bad credit scores. As banks were giving out more sub-prime mortgages than prime mortgages, MBSs mainly consisted of sub-prime mortgages. The quality of these subprime loans has been constantly deteriorating every year since 2001 [5]. The problem started when the banks gave out too many of these low-quality sub-prime mortgages to people who had a bad credit history. The tranches that had these MBSs of the sub-prime mortgages had the

chance of most percent to gain as these were given out at a very high-interest rate. Everything works fine if mortgagees pay their mortgages on time. But the problem starts when people start defaulting. Then the investors start losing their money. During the 2008 recession many people together defaulted on their mortgages. Therefore, the investors lost money and also the banks lost money from the mortgage non-payment.

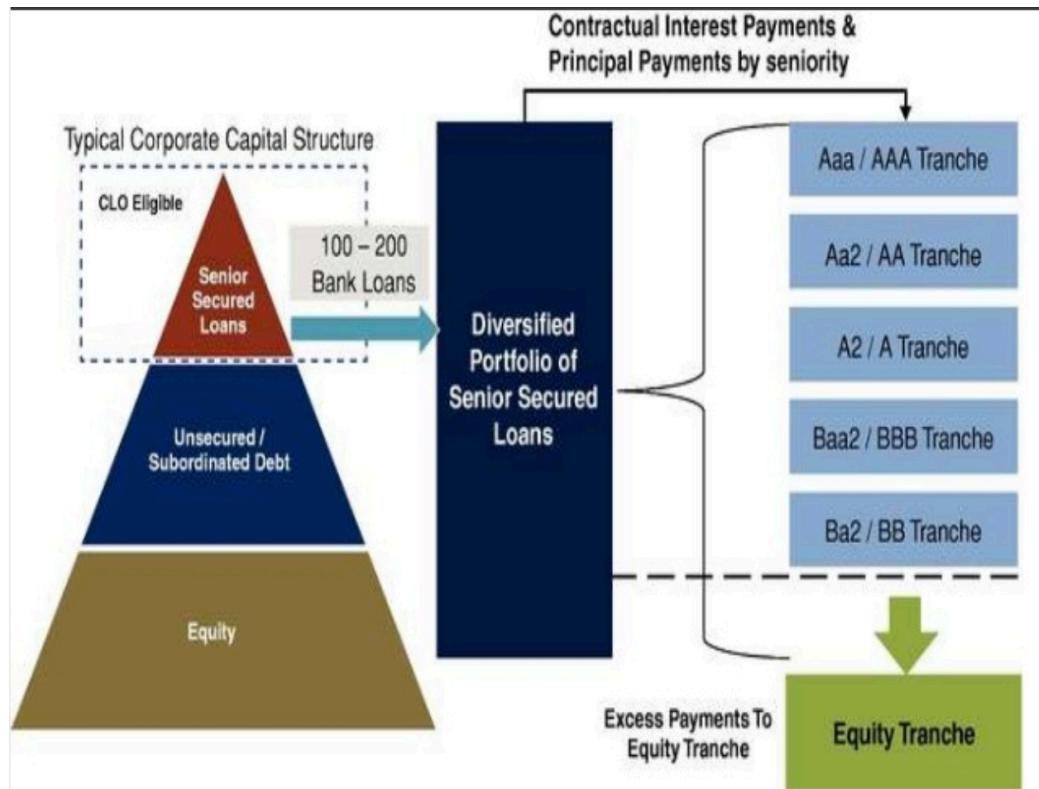


Fig 2: Division of Tranches

Derivative securities and their complexities also attributed to the collapse of the housing market. Securities such as MBS were split and then repackaged into tranches, then they were again split and repackaged. A tranche is a portion of something, here it means a security that can be split into several smaller pieces. The splitting and repackaging happened several times over and after that, rating agencies like S&P and Moody's Analytics gave ratings to these

tranches. Therefore, securitization of the mortgages into MBS and making a complex financial entity can also be blamed for the 2008 crisis [5], [6], [7].

The effects of the crisis were a drop in the housing prices, a sharp rise in unemployment, and the decline in overall GDP. The housing prices dropped by 32% from 2006 prices and homeownership also dropped from its peak at 69% to about 66%. About 61,000 businesses filed for bankruptcies which resulted in sharp decline in employment, 8.8 million people lost their jobs which resulted the long-term unemployment to rise to 45% in 2010 [3]. During this period the US saw the worst GDP decline since 1939, it dropped by 4.7% [3]. Everything described above made common people suffer more than anyone else.

2.3 PREDICTION OF HOUSING CRISES

Studies have shown that various econometric techniques can be helpful in predicting such housing crises. If these crises can be predicted such, then measures can be taken to prevent or lessen the impact of the crisis. Techniques for predicting can range from simple statistical techniques to more complex deep learning ones. Simple statistical techniques like linear regression can be used to predict banking or financial crisis. The working of this simple technique is different than the more complex machine learning techniques as the complex ones require more computation to produce more intelligent results. In this project we make use of the following techniques:

1. Hidden Markov Model (HMM)
2. Long short-term Memory (LSTM)
3. Linear Regression

HMM is used when there is one observable state present and using that observable state the states that are hidden could be predicted. [3] and [4] both uses HMM to predict stock market analysis for time series data but takes different approaches. [3] uses a higher order HMM with dimensionality reduction to predict the next day price for S&P 500 index. In a slightly different approach [4] uses a more complex form of first order HMM to predict airline stock prices by recognizing pattern and interpolating them to predict a bit farther than just one day. These two techniques can be merged together to build an HMM model that could predict housing prices ahead in the future.

LSTM is a more complex machine learning technique where prediction can be made even when there are gaps between important events. [5] provides us with an approach where this gap can be avoided while predicting a particular event in time. On the other hand, [6] describes an enhanced version of LSTM model to predict all types of time series data (TSD). According to [6] TSD clustering, classifying and forecasting is the new trend that can solve a lot of complex problems including recession forecast. Using the concepts discussed in these papers, a model can be developed to predict housing market crisis since there is a time gap between one recession to another.

A linear regression model is a statistical modelling technique that can be used on historical data to find the market trend. [10] talks about different linear regression analysis with GDP growth to predict economic crisis. It also focuses on short term GDP growth which takes into account three previous quarters to predict the growth for the next quarter. This model can be extended to be used on housing dataset to predict housing crisis. It can be inferred from this that regression analysis can be done even when there is not enough data available.

3. EXPERIMENTAL DESIGN

The objective of this research is to examine the historical housing prices, mortgage rate and, number of houses sold datasets and predict using machine learning techniques and deep learning techniques whether we are nearing another housing crisis. For example, the average house prices for single-family homes in San Francisco soared to about 1.4 million dollars by the end of 2017 and it continued rising till 2018. After which the prices fell a bit in early 2019. The goal is to analyze the data for house prices and mortgage rates and, use a learning technique that would predict whether there is a housing crisis in the near future or if the market is just correcting itself. In the next sections we discuss about the dataset and machine learning techniques that we used in this project.

3.1 DATASETS

The datasets that will be used in this project are a combination of a few datasets which had some federal data like the mortgage rates and state data for the house prices. For the total number of houses that were sold, we crawled a website to get data. The dataset that we will be using are:

1. Mortgage rate [12]
2. Housing price [11]
3. Total number of houses sold [13]

We merged these datasets using Python data analysis library Pandas. The datasets had data from 1990 to 2020 and the interval was a month. There were various data formats for each dataset so in order to merge them on the date, we converted all the date into the same format. The pictures below show the different date formats that were in each dataset.

1	Date	CA	Alameda	1	observation_date	MORTGAGE30US
2	Jan-90	\$194,952	\$226,149	2	1971-04-02	7.33
3	Feb-90	\$196,273	\$219,306	3	1971-04-09	7.31
4	Mar-90	\$194,856	\$225,162	4	1971-04-16	7.31
5	Apr-90	\$196,111	\$229,333	5	1971-04-23	7.31
6	May-90	\$195,281	\$232,291	6	1971-04-30	7.29
7	Jun-90	\$194,410	\$231,250	7	1971-05-07	7.38
8	Jul-90	\$193,088	\$232,916	8	1971-05-14	7.42
				9	1971-05-21	7.44
				10	1971-05-28	7.46
				11	1971-06-04	7.52
				12	1971-06-11	7.52
				13	1971-06-18	7.54
				14	1971-06-25	7.54

1	Month	Year	house_sold
2	Jan.	2020	764000
3	Dec.	2019	708000
4	Nov.	2019	692000
5	Oct.	2019	707000
6	Sept.	2019	725000
7	Aug.	2019	708000
8	July	2019	660000
9	June	2019	729000

Fig 3: Date format of different datasets that are used

We changed the date format for all the dataset into “yyyy/mm/dd time” added another column of the period for easy reading of data. For the price, we took only the aggregated house prices in all of CA and not each county as that would have complicated the dataset and the analysis would have been homogenous. The interest rate dataset had a lot more data starting from the 1970’s but when it joined with the primary dataset of the house price and the resulting dataset had the data that fell in the intersecting period.

ID	Date	month	year	price	rate	total_house_sold	period
1	1990-01-01 00:00:00.000000	1	1990	194952	10.05	620000	1/1990
2	1990-02-01 00:00:00.000000	2	1990	196273	10.31	591000	2/1990
3	1990-03-01 00:00:00.000000	3	1990	194856	10.34	574000	3/1990
4	1990-04-01 00:00:00.000000	4	1990	196111	10.56	542000	4/1990
5	1990-05-01 00:00:00.000000	5	1990	195281	10.67	534000	5/1990
6	1990-06-01 00:00:00.000000	6	1990	194410	10.29	545000	6/1990
7	1990-07-01 00:00:00.000000	7	1990	193088	10.11	542000	7/1990
8	1990-08-01 00:00:00.000000	8	1990	192180	10.29	528000	8/1990
9	1990-09-01 00:00:00.000000	9	1990	189979	10.22	496000	9/1990
10	1990-10-01 00:00:00.000000	10	1990	187630	10.24	465000	10/1990
11	1990-11-01 00:00:00.000000	11	1990	192020	10.13	493000	11/1990
12	1990-12-01 00:00:00.000000	12	1990	190375	9.81	464000	12/1990
13	1991-01-01 00:00:00.000000	1	1991	192054	9.75	401000	1/1991
14	1991-02-01 00:00:00.000000	2	1991	194806	9.56	482000	2/1991
15	1991-03-01 00:00:00.000000	3	1991	202666	9.59	507000	3/1991

Fig 4: Pre-processed dataset with changed Date format

3.2 MACHINE LEARNING MODELS

The machine learning models used to do this project range from simple to complex deep learning techniques. We would look into each of the techniques and compare each of their results. For this project we have used three machine learning algorithms on the preprocessed data that was described in the previous section.

- 1) Linear Regression
- 2) Hidden Markov Model (HMM)
- 3) Long Short-Term Memory (LSTM)

In the next sub sections, we will be describing each method and how we have used them to make prediction about the housing prices for another year i.e. 12 months.

3.2.1 LINEAR REGRESSION

Linear regression is a supervised learning technique that models linear relationship between the dependent or scalar and the independent or explanatory variables. If there is one independent variable, then the modelling technique is called simple linear regression. In this case the scalar variable is dependent on only one explanatory variable. The form of simple linear regression model is

$$y = b_0 + b_1 \cdot x_1$$

When there is more than one explanatory variable for a scalar then it is called multiple or multivariate linear regression. In this model the scalar variable has its value dependent on more than one explanatory variable. The form of multiple linear regression model is

$$y = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + b_3 \cdot x_3 + \dots + b_n \cdot x_n$$

In the above equations, ‘y’ is the dependent variable ‘ x_1 ’, ‘ x_2 ’, ‘ x_n ’ are the independent variables. The error term is the constant b_0 , it is also called the y-intercept and ‘ b_1 ’, ‘ b_2 ’, ‘ b_n ’ are the weights of the independent variable.

In this project we have used both simple and multiple linear regression. For both the model the dependent variable is the house price and the independent variable is date for the simple linear regression model. We started with simple linear regression to understand the dynamics of the house price related to time and the time has affected the housing market and coded the algorithm instead of using sci-kit learn (used later for multiple regression). The dataset used for this is the housing price dataset where it contains house prices for all counties of California and average of all California house price [11].

Steps to code Linear Regression Model:

Step 1: For each (x, y) , calculate x^2 and (xy)

Step 2: Sum all $(x, y, x^2 \text{ and } xy)$ which would give us $\sum x$, $\sum y$,

$\sum x^2$, and, $\sum xy$

Step 3: Calculate slope b_1 :

$$b_1 = n \sum xy - \sum x \sum y \quad n \sum x^2 - \sum x^2$$

where n is the number of points

Step 4: Calculate the y-intercept b :

$$b_0 = \sum x - b_1 \sum x n$$

Step 5: Substitute all the value in the simple linear regression equation

$$y = b_0 + b_1 * x$$

The above method of finding linear regression is called the least squares method. This method aims at minimizing the sum of squares errors or residuals. The squared sums are calculated for each x and y values which are the inputs and the outputs. A training rate is used to scale the factors in order to minimize the error values. This is repeated until the least sum squares of errors is achieved and there is no more improvement possible.

```
numer, denom = 0, 0
for i in range(n):
    numer += (X[i] - mean_x) * (Y[i] - mean_y)
    denom += (X[i] - mean_x) ** 2

m = numer / denom
c = mean_y - (m * mean_x)
```

Fig 5: Code snippet for calculation of Slope and Error for Linear Regression

```

max_x = n + 12
x = np.linspace(0, max_x, 1000)
y = c + m * x

ss_t = 0
ss_r = 0
for i in range(n):
    y_pred = c + m * X[i]
    ss_t += (Y[i] - mean_y) ** 2
    ss_r += (Y[i] - y_pred) ** 2
r2 = 1 - (ss_r / ss_t)
print("R-squared: {}".format(r2))

```

Fig 6: Code snippet for calculation of Linear Regression using Least Squares method

The code snippets show the steps involved in finding the Least Squares Regression model for housing price. After finding the best fit line we can extrapolate the line to as far we want to make prediction.

For simple linear regression we employed the algorithm on the whole housing price dataset [10] to see how the line is for the thirty years of historical data. After that, we extracted data for last 2 years to compare the housing inflation from 2018. Along with finding the best fit line we have also calculated the Root Means Square Error (RMSE) score which typically lies between 0 and 1. RMSE measures the difference between the predicted values and the observed values. A RMSE score of 1 suggests that the predicted values and the observed are very similar and a score of 0 suggests that there is no correlation between the predicted and the observed values. An RMSE score of 1 is very unlikely as it means that both the data are same which in reality is never going to happen. A model with RMSE of 0 means that the prediction is very different from the observed which should never be the case, as this would mean that the model has not been trained well.

Next we used multiple linear regression to predict the housing prices. In this part the dependent values are still the housing prices, but the independent values are date, mortgage rates

and the total number of houses that were sold during that period. Multiple linear regression was coded using the Python Sci-kit Learn library. In this, the dataset was divided into training and testing set, with 20% of the data being in the testing set. Then we fit the data into the model, two separate models were created to see the relationship between the actual observed data and the predicted data. After the separate models were created, we calculated the RMSE score to see the error value in the model and the R^2 goodness of fit to see how well the model fits the data. The code snippet below shows how the above steps were coded. The result from the model will be discussed in Chapter 4.

```
X = df[['Date', 'rate', 'total_house_sold']]
Y = df['price']

lin_reg = LinearRegression()
lin_reg.fit(X, Y)
lin_reg.predict(X)
print(lin_reg.score(X, Y))

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2,
                                                    random_state=0)

linreg = LinearRegression()
linreg.fit(X_train, Y_train)
Y_pred = linreg.predict(X_test)
Y_train = linreg.predict(X_train)

rmse = (np.sqrt(metrics.mean_squared_error(Y_test, Y_pred)))
r2 = metrics.r2_score(Y_test, Y_pred)
```

Fig 7: Code snippet of using Sci-kit Learn to train and fit model and then make prediction

3.2.2 *HIDDEN MARKOV MODEL*

Linear Regression was built to understand the dynamics of the Housing Market, it is not as intelligent as the other modelling techniques. Hidden Markov Model (HMM) is a statistical modelling technique that derives its name from Russian mathematician Andrey Markov, who invented Markov chains. HMM is a collection of Markov chains which gives us the probability of the next sequence depending on the present states. In a Markov model the past and future doesn't have relationship given that the present state is known. HMM is used to predict the probability of a hidden state given that there are observed states and probabilities of the transition from one state to another. In CS 297 we experimented using these two steps:

1. Coded the HMM with the example from Prof. Stamps paper on HMM to understand the working of HMM [14]
2. Used HMM from Sci-kit Learn to code and model the data from housing dataset [11][12][13]

The HMM algorithm would determine the annual temperature (Hot, Cold) given the observation of ring size of tree growth (Small, Medium, Large). To start with, we will have the state transition matrix which would give us the probability of if the temperature is hot one year what is the probability that the temperature would be hot or cold next year. Similarly, if the temperature is cold this year the probability of it being cold or hot next year. This is called the state transition matrix. We will also have an observation matrix that would give us the probability of the temperature being hot or cold given the ring size of the tree whether it is small, medium or large.

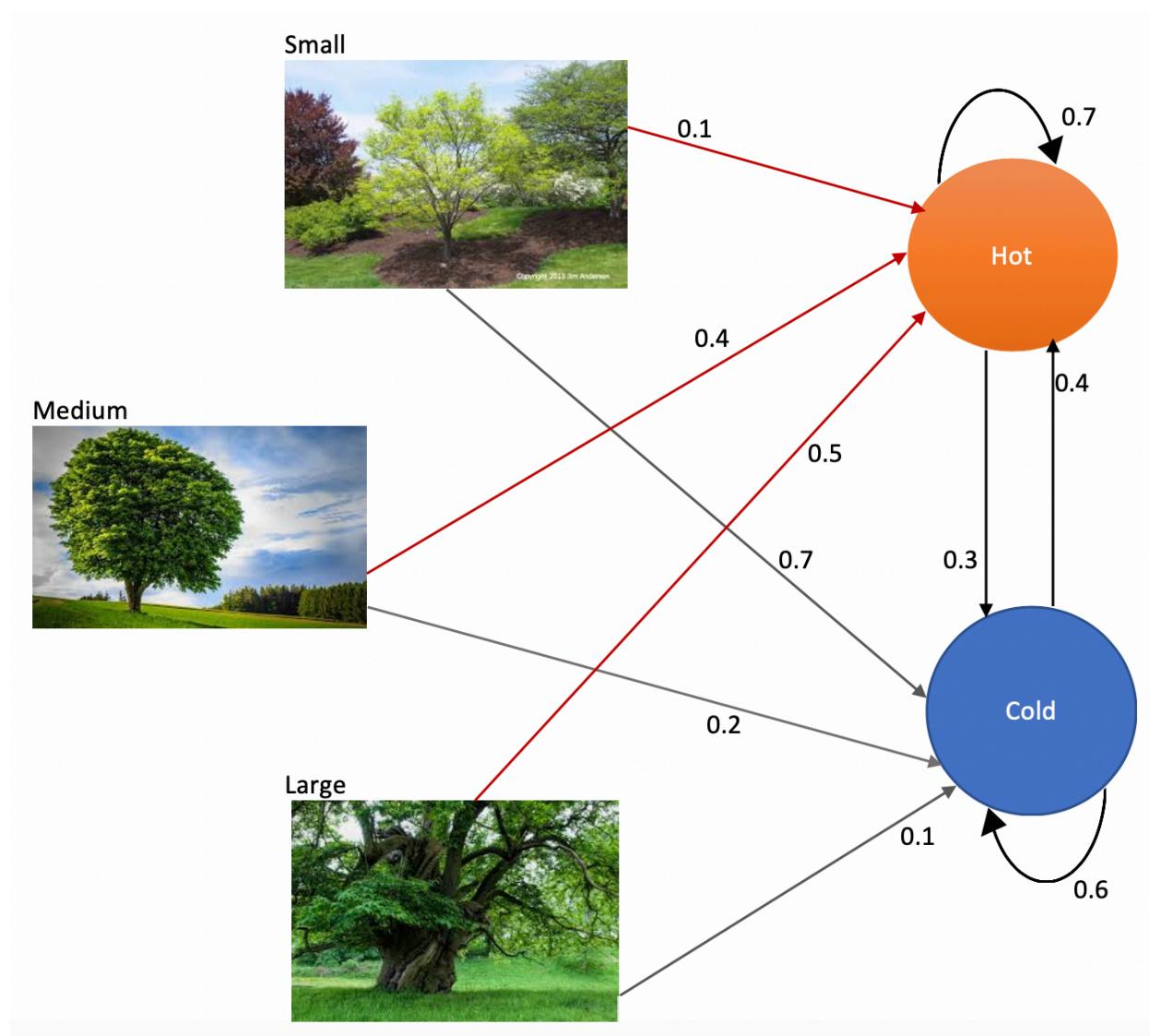


Fig 8: Hidden Markov Model

The transition from the present step to the next step is a Markov chain that follows a Markov transition matrix. A Markov transition matrix is an $n * n$ matrix that has probabilities of transitioning from one state to another, for our example it's from Hot to Cold or vice versa. However, the actual state is not known they are hidden as we cannot directly observe the temperature whether it is Hot or Cold in the previous state. For this problem the state transition

matrix is same as that of Prof Stamp's paper [14] and it's also shown in the figure above.

$$\begin{array}{cc} & \begin{matrix} H & C \end{matrix} \\ \begin{matrix} H \\ C \end{matrix} & \left[\begin{matrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{matrix} \right] \end{array}$$

Fig 9: The state transition matrix

Although we don't know the hidden states, we have a provision to make prediction through the observation emission matrix. An observation matrix is an $n * m$ matrix which gives us the probability of state happening given that observation. In this case as well we would be referencing this from Prof Stamp's paper [14].

$$\begin{array}{ccc} & S & M & L \\ \begin{matrix} H \\ C \end{matrix} & \left[\begin{matrix} 0.1 & 0.4 & 0.5 \\ 0.7 & 0.2 & 0.1 \end{matrix} \right] \end{array}$$

Fig 10: Observation matrix

Along with the state transition matrix(A) and observation emission matrix(B) we would also need an initial state distribution matrix (π) which would let us start the calculation for the next steps. All these matrices are row stochastic meaning that if we add up a row it's always equal to 1. Next we have a sequence of observations from which we predict the temperature. The observations are denoted by $O = O_0, O_1, O_2, \dots, O_{T-1}$

To compute the hidden states, we have to solve three questions and these questions are discussed by Prof. Stamp paper [14]. The coding for the above problems was done in python. To solve the above problems described in the paper we have first prepared the data. After preparing the data, we have calculated the alpha pass or the forward algorithm by multiplying the A matrix

with the probabilities of occurrence. After calculating the alpha-pass, we calculated the beta-pass or the backward algorithm. The backward algorithm is calculated by starting the matrix backwards. The calculation is similar to the alpha pass the only difference is the starting point of the matrix. After the calculation of these two matrices we would use them to calculate the gamma and di-gamma. Di-gammas are used to find the best fit values for the model. To calculate di-gammas, we get all the values before gamma from alpha matrix and all the values after gamma from the beta matrix. Then subtract beta matrix from alpha matrix to get the di-gamma. After the di-gammas are found we scale the HMM model and update the original matrix with values that are calculated. Therefore, after each calculation there is an updated A, B and Pi values. We would use these updated values in our calculation of next steps.

	Observation	Best_Path
0	M	H
1	M	H
2	L	H
3	M	H
4	S	C
5	S	C
6	S	C
7	M	H
8	L	H
9	M	H
10	L	H
11	L	H
12	S	C
13	S	C
14	S	C
15	M	H
16	L	H
17	L	H
18	M	H
19	M	H
20	M	H
21	L	H
22	L	H
23	L	H
24	L	H
25	M	H
26	S	C
27	M	H
28	M	H
29	L	H

Fig 11: Result of 30 observations of ring size

After coding the HMM algorithm and getting to know how HMM works we started the next part of the problem. We used the HMM from hmmlearn.hmm module Sci-kit learn to apply it to the housing dataset [15]. For this we have added a percentage difference in price to the housing dataset to build the model. The data used to build this model is a column stack of diff_percentages, prices, num_of_houses_sold, rate. After that we used GausianHMM to build the model.

```
diff_percentages = 100.0 * np.diff(prices) / prices[:-1]
diff_percentages = np.append([0], diff_percentages)

data = np.column_stack([diff_percentages, prices, num_of_houses_sold, rate])

hmm = GaussianHMM(n_components=15, covariance_type='tied', n_iter=100000,
                  algorithm='viterbi', random_state=False)

hmm.fit(data)
pred_count = 12
```

Fig 12: Code snippet of using Sci-kit Learn to train and fit data into HMM model

Sci-kit learn has some attributes we need to pass in to build the model. They are shown below:

```
sklearn.hmm.GaussianHMM(n_components=1, covariance_type='diag', startprob=None,
transmat=None, startprob_prior=None, transmat_prior=None, means_prior=None,
means_weight=0, covars_prior=0.01, covars_weight=1, n_iter = None, algorithm = 'viterbi')
```

For building HMM model for this project we have used the following values for the attributes. We have added random_state = False to get repeatable result.

```
hmm.GaussianHMM(n_components=15, covariance_type='tied', , n_iter = 10000, algorithm =
'viterbi', random_state=False)
```

After the model was built with the attributes defined, the data was fit into the model and prediction was made for the next 12 months. The result from the model will be discussed in the Chapter 4.

Table 1: Sci-kit learn HMM model attribute description

Gaussian HMM attributes	Meaning
<i>n_components = 15</i>	The number of states of the HMM model
<i>covariance_type = ‘tied’</i>	All components share the same general covariance matrix
<i>n_iter = 10000</i>	The number of backward and forward run while training the model
<i>algorithm = ‘viterbi’</i>	The algorithm used inside the HMM model
<i>random_state = False</i>	Whether to used random variable as seed or not

3.2.3 LONG SHORT-TERM MEMORY

Long short-term memory (LSTM) is a type of recurrent neural network (RNN) that is mostly used in the field of deep learning. Although its architecture is similar to that of RNN, LSTMs have feedback connections and not just feed-forward connection. RNNs had a major problem of vanishing gradient which led to the popularity of LSTM. The main cause of vanishing gradient was that information had to travel sequentially through all cells before getting to the present cells. Travelling for long distances easily corrupted data by getting multiplied by garbage values (small number < 0). LSTMs also have the ability to learn from long term dependencies, which was a problem for traditional RNNs. LSTMs help preserve the error that can be backpropagated through time and layers. By maintaining a more constant error, they allow recurrent nets to continue to learn over many time steps. Also, not being sensitive to gap-length makes LSTM superior than RNNs and Hidden Markov Models. LSTMs are well-suited for

classifying, processing and making predictions on time series data, since there can be gaps of unknown duration between important events in a time series.

An LSTM network typically has a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The cell keeps track of the dependencies between all the elements of the input sequence. Next the input gate checks the amount of new information flow into the cell. Then the forget gate controls how long the information can stay in the cell. Finally, the output gate checks the amount to which the values in the cell are used to compute the final output to the next cell. There are connections in and out of the LSTM gates. The weights of these connections, which need to be learned during training, determine how the gates operate.

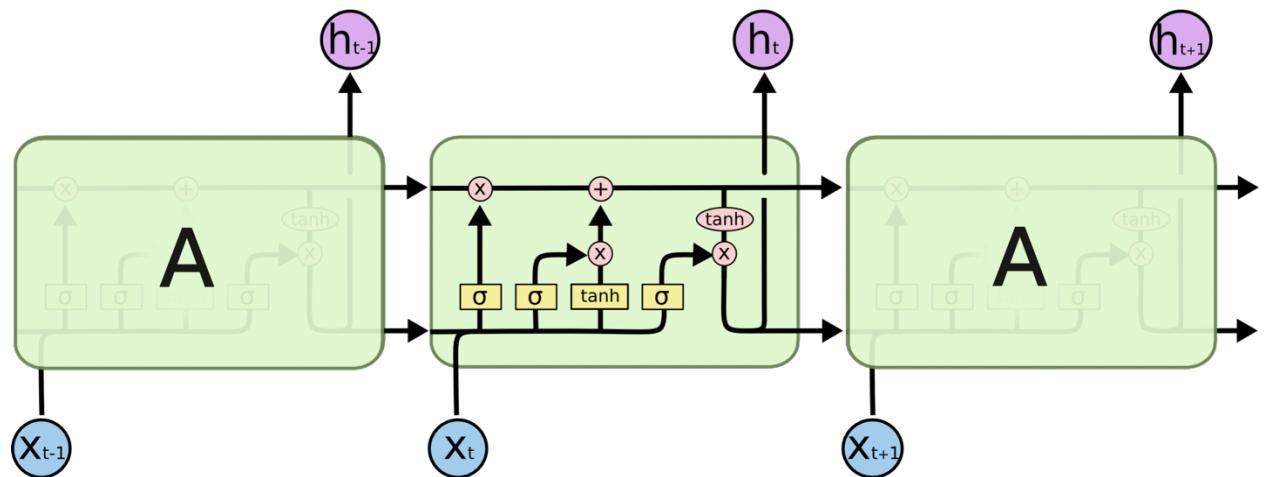


Fig 13: An LSTM network

LSTM working step by step:

Step 1: LSTM has to decide on what information is going to stay in the cell state and what information needs to be dumped. This decision is made by the forget gate or the sigmoid layer. It

looks at h_{t-1} and x_t of the LSTM, and outputs a number between 0 and 1 for each number in the cell state C_{t-1} . 1 represents “keep all the information” while a 0 represents “don't keep any of the information”

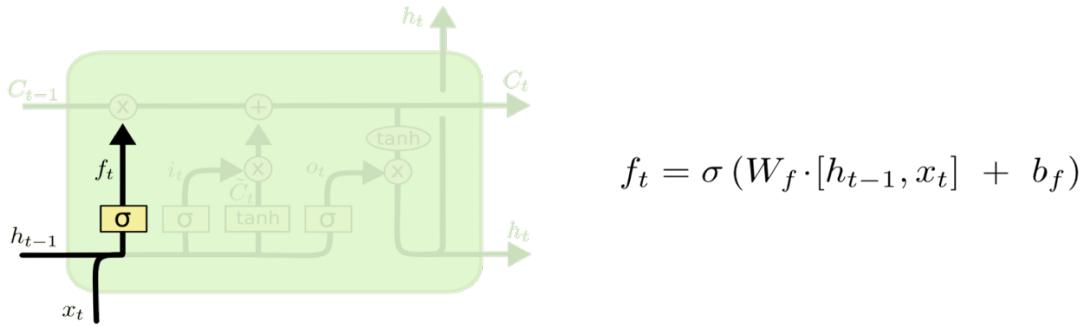


Fig 14: Forget gate

Step 2: The next layer of LSTM decides what information is to be stored in the cell state of LSTM network. This is done in two parts. First the input gate layer decides what information/values needs to be updated. Then the tanh layer creates \tilde{C}_t a candidate vector, that is added to the state.

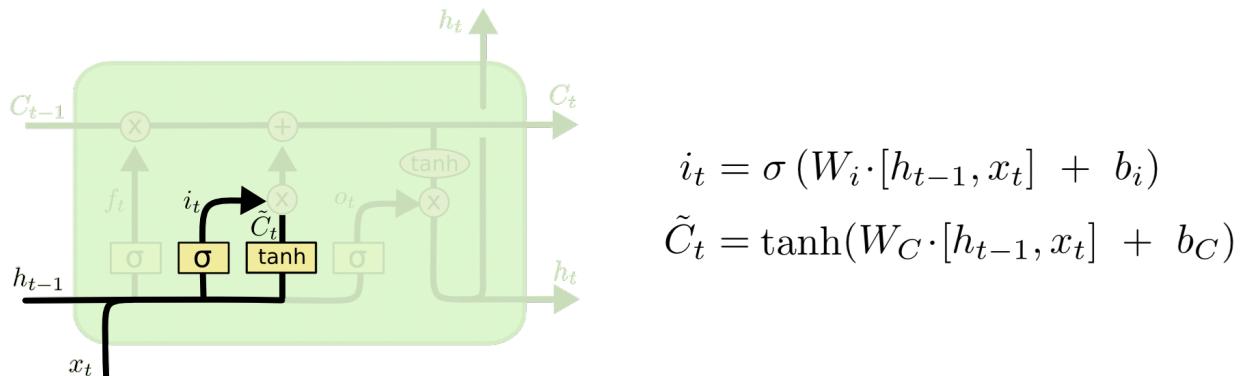


Fig 15: Input Gate

Step 3: Next the old state of the cell C_{t-1} is updated to the new cell state C_t . The previous steps gave us all the essential parameters to this. The old state is multiplied by the output of the forget

layer and then it is added to the value we get from multiplying the input layer value to the candidate vector. The figure below demonstrates that.

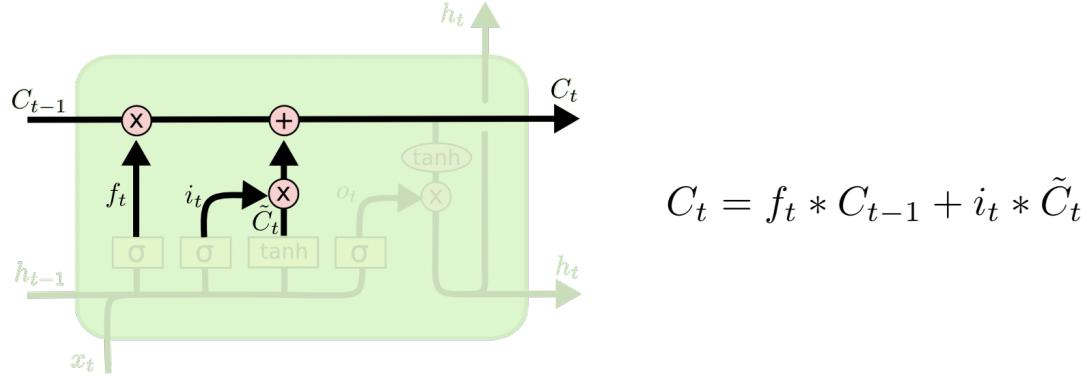


Fig 16: Updating old state into new state

Step 4: Finally, the output layer outputs the value of the current cell state to the next cell. This is also done in two steps firstly; the sigmoid layer decides what parts of the cell state is going to the output layer. Then, the cell state is sent through the tanh layer (to push the values to be between -1 and 1) and is multiplied by the output of the sigmoid layer.

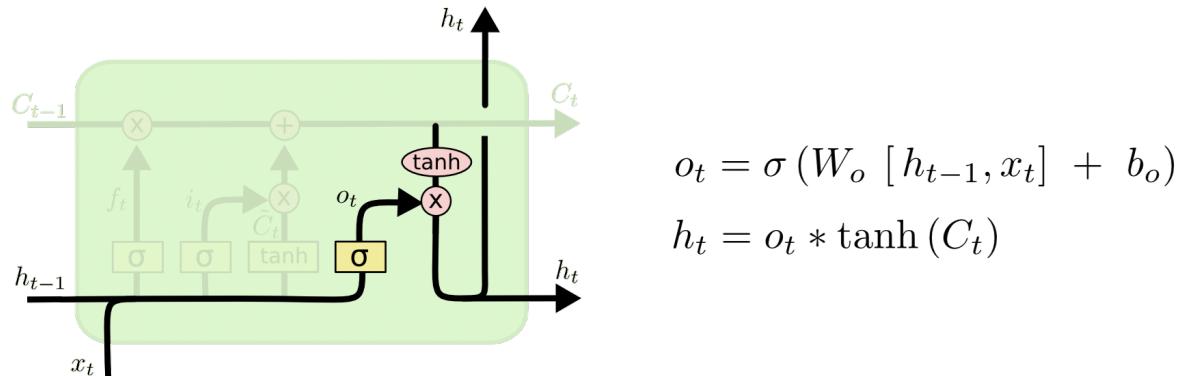


Fig 17: Output gate

There are many variants of LSTM, but those details are not required for this project as we would be using Keras to build and model the LSTM network. Next we will discuss about building an LSTM model using the housing dataset for prediction.

The LSTM model for this project was build using a neural network library Keras which runs on TensorFlow backend. This deep learning library is written in Python and it is useful for

our project as the whole coding for this project is done in Python. Keras is new therefore to learn the dynamics and working of keras we first used it on a small dataset to do prediction before jumping directly to the project data. After we successfully understood to concepts needed for this project, we went ahead to use keras on the project dataset. For the project we divided the dataset into training data and testing data with testing data being 15% of the whole dataset. Then we processed the training and testing data by feeding it to the time series generator of Keras sequence generator.

```
house_price_train = house_price[:train_test_split]
house_price_test = house_price[train_test_split:]

date_train = df['Date'][:train_test_split]
date_test = df['Date'][train_test_split:]

look_back = 5

train_generator = TimeseriesGenerator(house_price_train, house_price_train,
                                      length=look_back, batch_size=20)
test_generator = TimeseriesGenerator(house_price_test, house_price_test,
                                     length=look_back, batch_size=2)
```

Fig 18: Code snippet showing division of training and testing data and data conversion for LSTM

To build the LSTM network a Sequential model from keras was chosen and to that model LSTM network was added with number of hidden nodes being 50 within the LSTM cell and input shape of 5X1. The input shape describes what is the input to the first layer of the network. The weights that are given to initial Keras network is uniformly divided within each layer which is given by *init='uniform'*.

```
model = Sequential()
model.add(LSTM(50, activation='relu', input_shape=(look_back, 1),
              init='uniform'))
model.add(Dense(1))
model.compile(loss='mean_squared_error', optimizer='adam')
```

Fig 19: Code snippet showing the building of LSTM model from Sequential model

Next we define the size of epoch for the neural network. For this the number of epochs is 1000. An epoch is one cycle through the full training data. Just one epoch would lead to underfitting as it would not learn much from the training data. Also, too many epochs would cause overfitting as it would learn too much from the training data that might not generalize the testing data. Next we fit the model to number of epochs and the processed training data.

```
num_epochs = 1000
model.fit_generator(train_generator, epochs=num_epochs, verbose=2,
                    shuffle=False)
```

Fig 20: Code snippet showing the fitting of LSTM model

After building the model and training the model we use the testing data to see how well the model is behaving and then make prediction. After that extend the prediction to the 12 months to see one-year prediction of the housing market.

```
prediction = model.predict_generator(test_generator)
house_price_train = house_price_train.reshape((-1))
house_price_test = house_price_test.reshape((-1))
prediction = prediction.reshape((-1))
```

```
num_prediction = 12
prediction_list = house_price[-look_back:]

for _ in range(num_prediction):
    x = prediction_list[-look_back:]
    x = x.reshape((1, look_back, 1))
    out = model.predict(x)[0][0]
    prediction_list = np.append(prediction_list, out)
prediction_list = prediction_list[look_back - 1:]
print('Prediction list')
print(prediction_list)
```

Fig 21: Code snippet showing prediction for next 12 months

After this we do some data processing to plot the graph of model and the prediction. The result of the model will be described in Chapter 4.

4. RESULTS AND DISCUSSION

In this chapter we will discuss about all the results that we got from the machine learning models and compare their results with each other. First, we will start with few scatter plots to understand the dataset and how everything is related to the house prices.

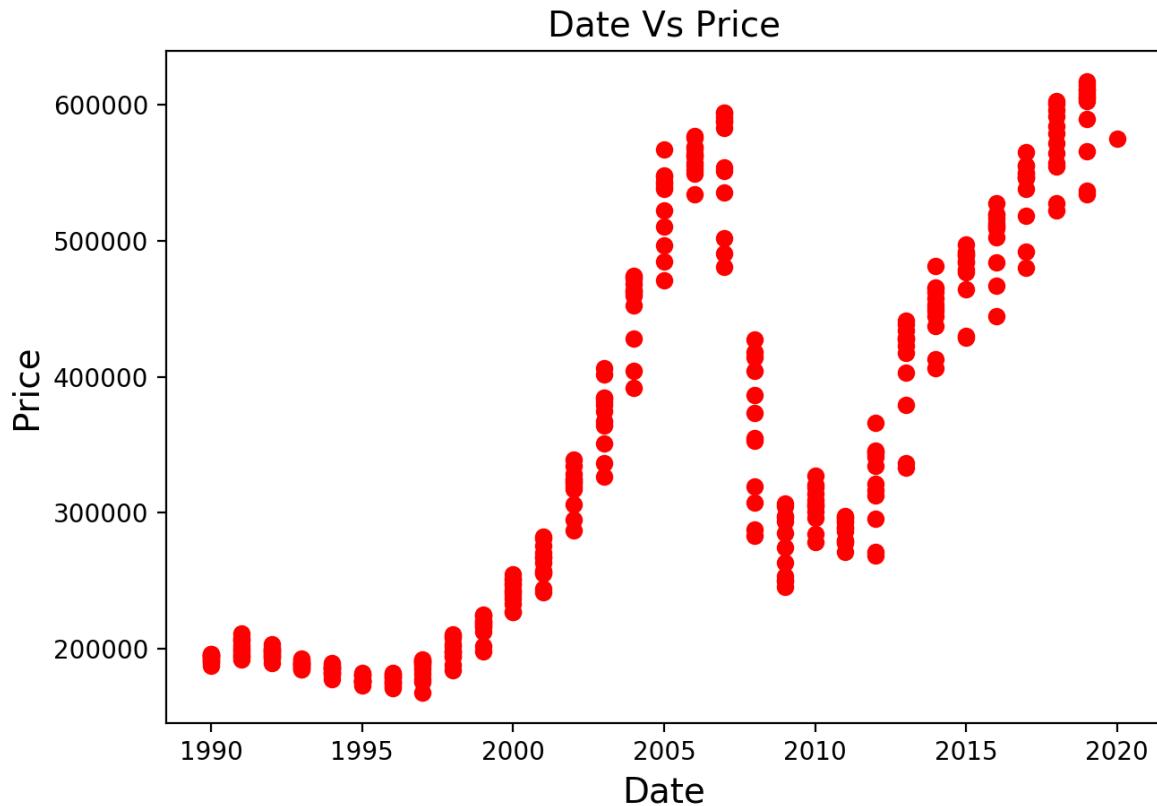


Fig 22: Plot of Price vs Date

The figure above shows how price have increased or decreased over time. We can see that the price had increased in 2005-2006 and troughed in 2008-2010. After that the price had increased steadily over the years with some seasonal variance.

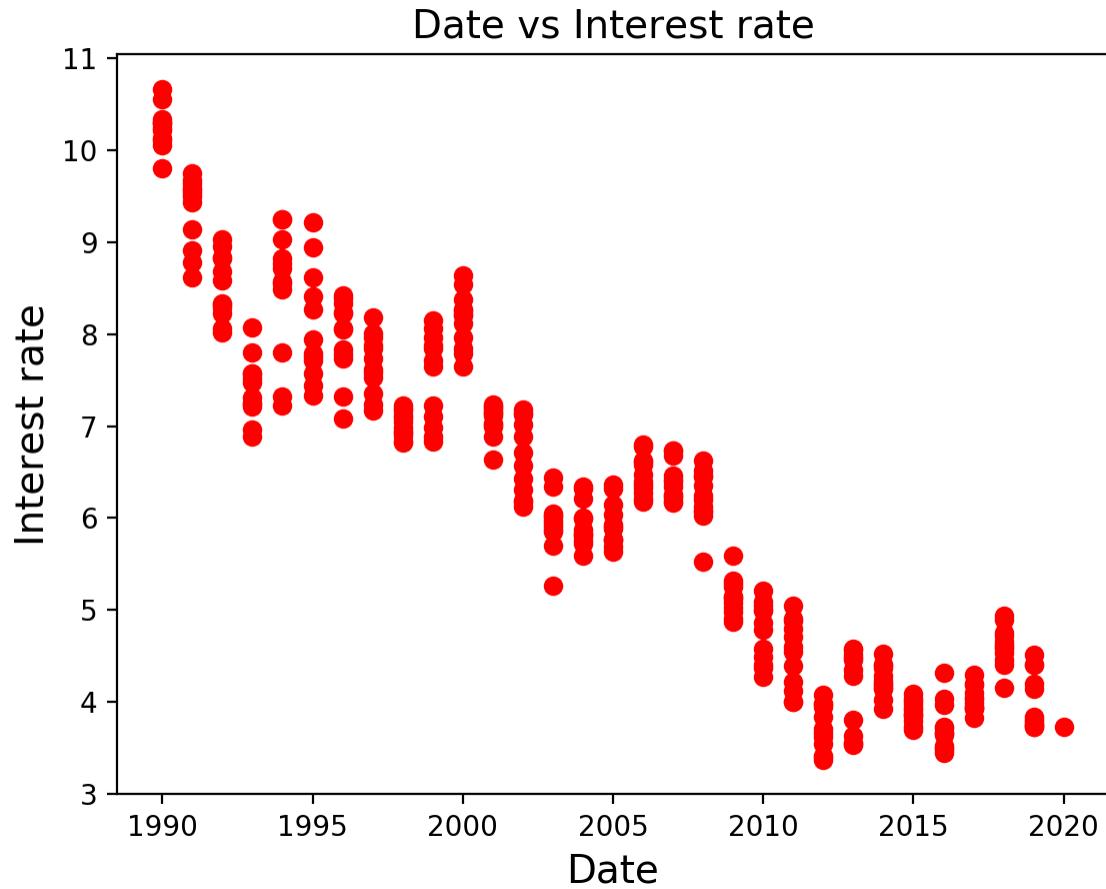


Fig 23: Plot of Date vs Interest Rate

This figure shows that the interest rate has decreased over time. But we can see that there was a spike in the interest rate between 2006-2008, which is around the time when giving out sub-prime mortgages were at it's highest. After 2008 there has been a sharp decline of mortgage interest rates between 2009-2012. This is because of the effort made by the federal government to bring back the economy by slashing interest rates.

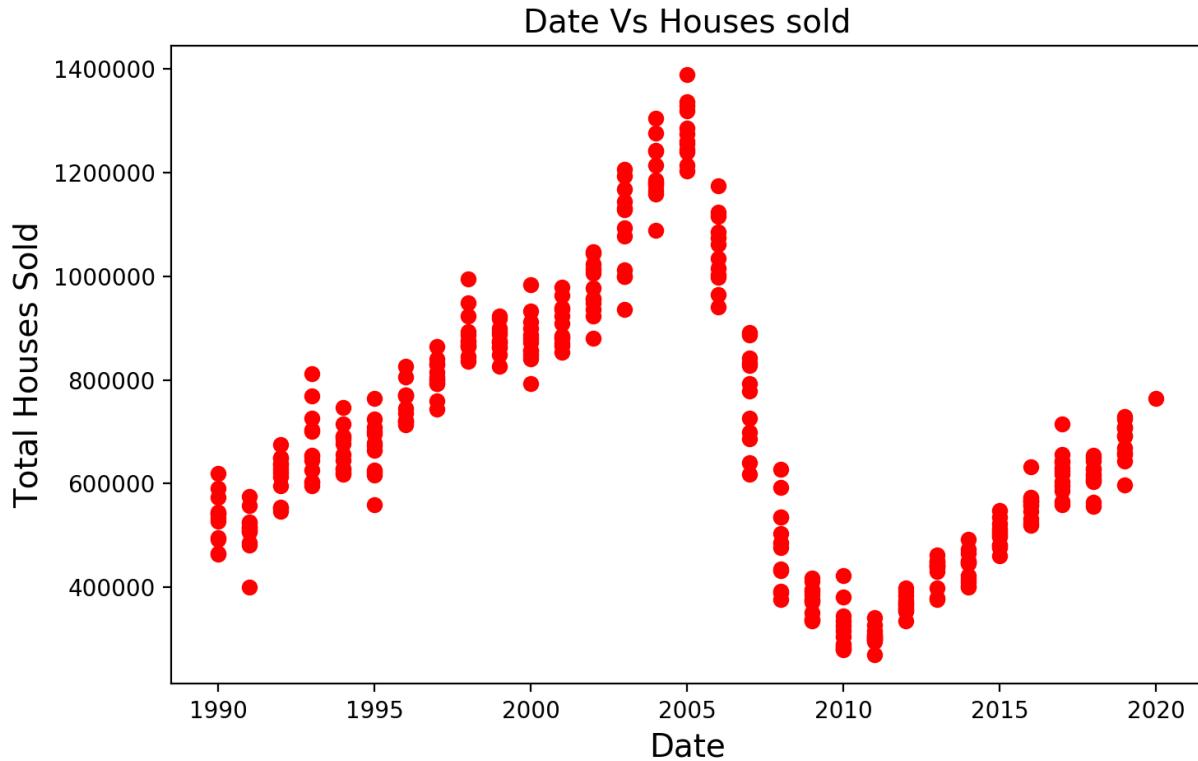


Fig 24: Plot of Date vs Houses Sold

This figure shows the plot between date and total number of houses sold. The number of houses sold peaked in 2005 which might be because of easy mortgage policies developed by banks. These mortgage policies eventually led to the financial crisis of 2008-2009. Also, we can see from the graph the houses sold troughed in 2008-2010 which was the time if the financial crisis.

From the above graphs we can say that the prices of the houses in 2005-2006 were high because the number of houses sold were also high. This also means that the supply of houses during that time was low and the demand was high. This supply and demand triangle inverted during and after the recession of 2008-2009 where there were more houses available than there were buyers as a result prices dropped. This is one of the many reasons that led to the recession.

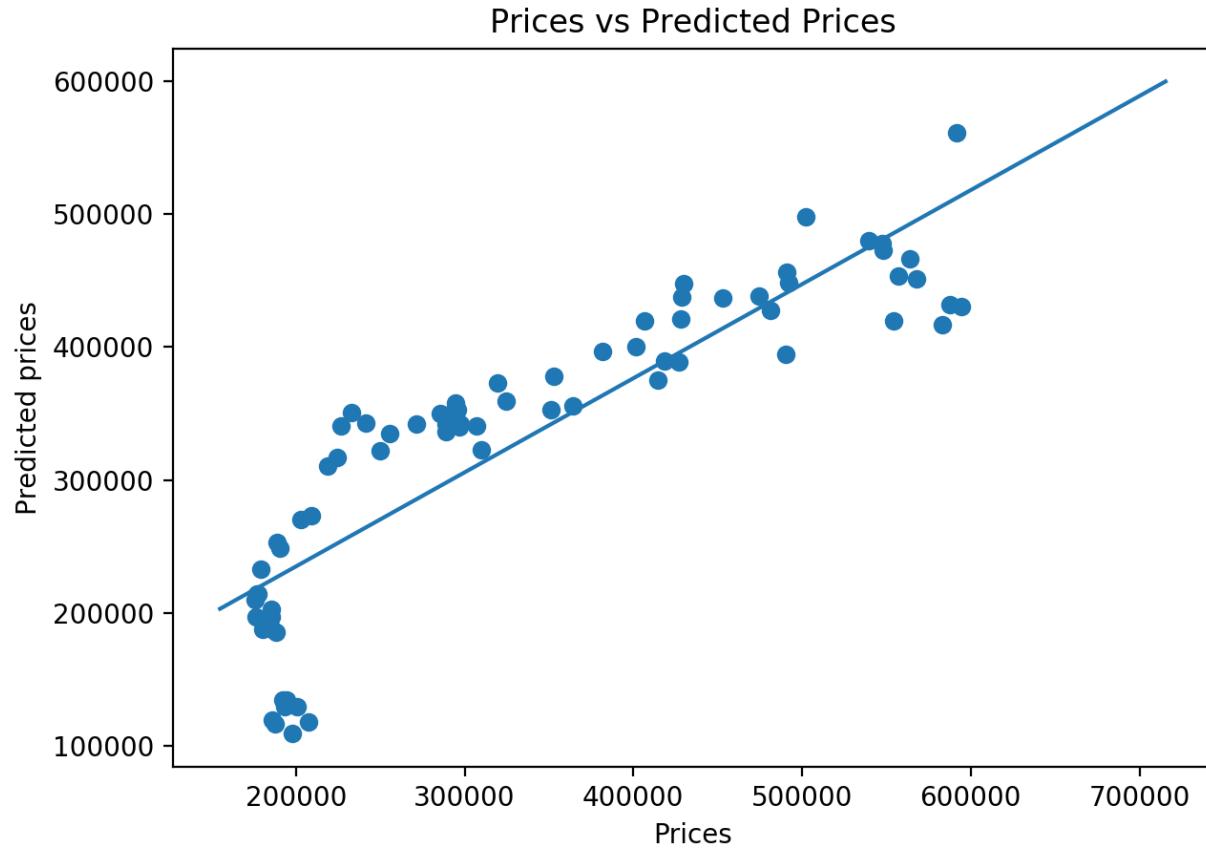


Fig 25: Actual prices vs Predicted prices from Multiple linear regression

The above figure shows the graph between actual prices and the prices predicted by the Regression model. The ideal graph would be diagonal line with a slope of 1 and all the points on the line. The slope of the line we are getting is about 0.7 and the predicted values fall close to the line. This means the model is pretty good at predicting the values, however there is room for improvement.

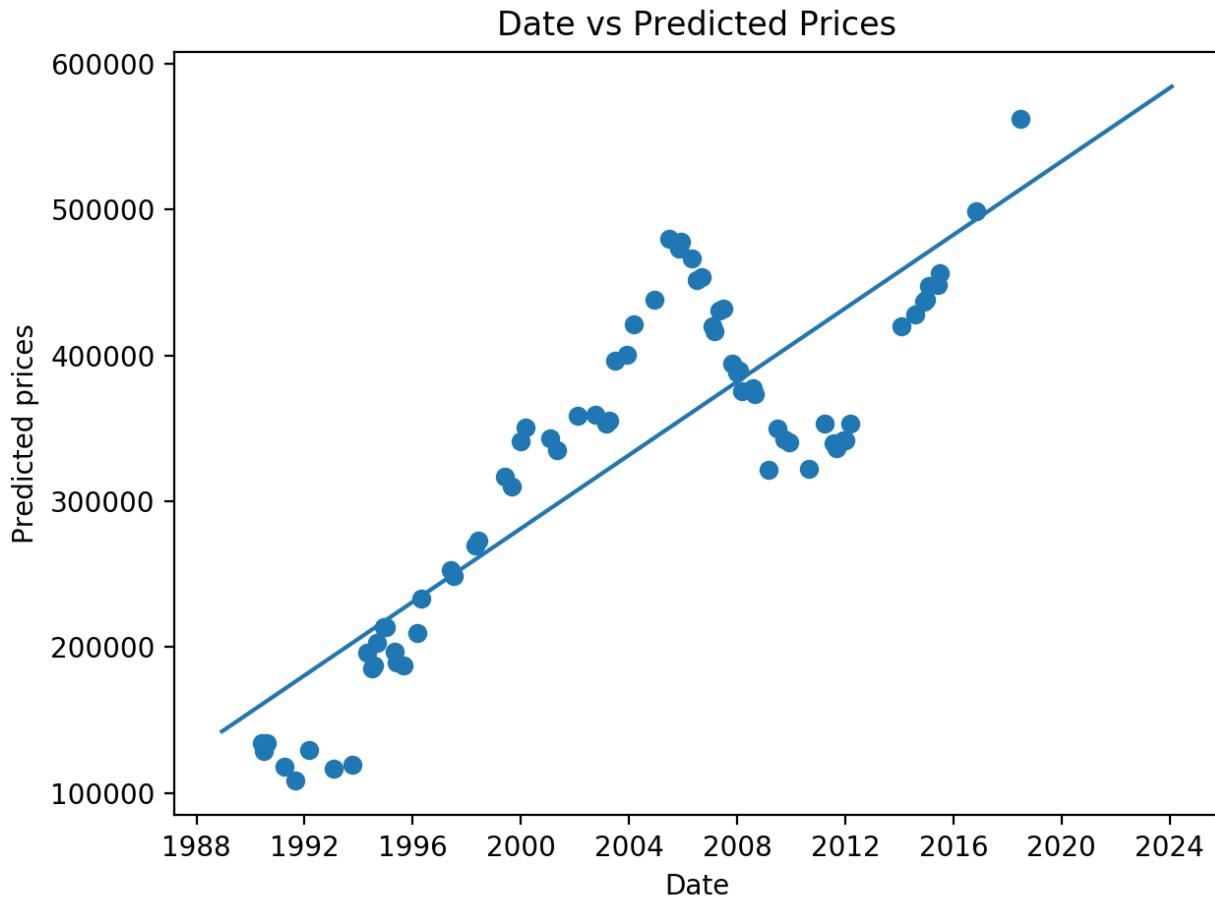


Fig 26: Price prediction by the regression model

The prediction made by the linear regression model does not show any fall in prices as the prices for the future keeps rising which can be seen by the overall positive slope. So, this model does not predict any recession in the housing market. The model also shows that the houses prices during 2004-2007 was way above than the ideal slope of the line therefore the ideal situation to avoid a crisis should be that the prices stay near the regression line.

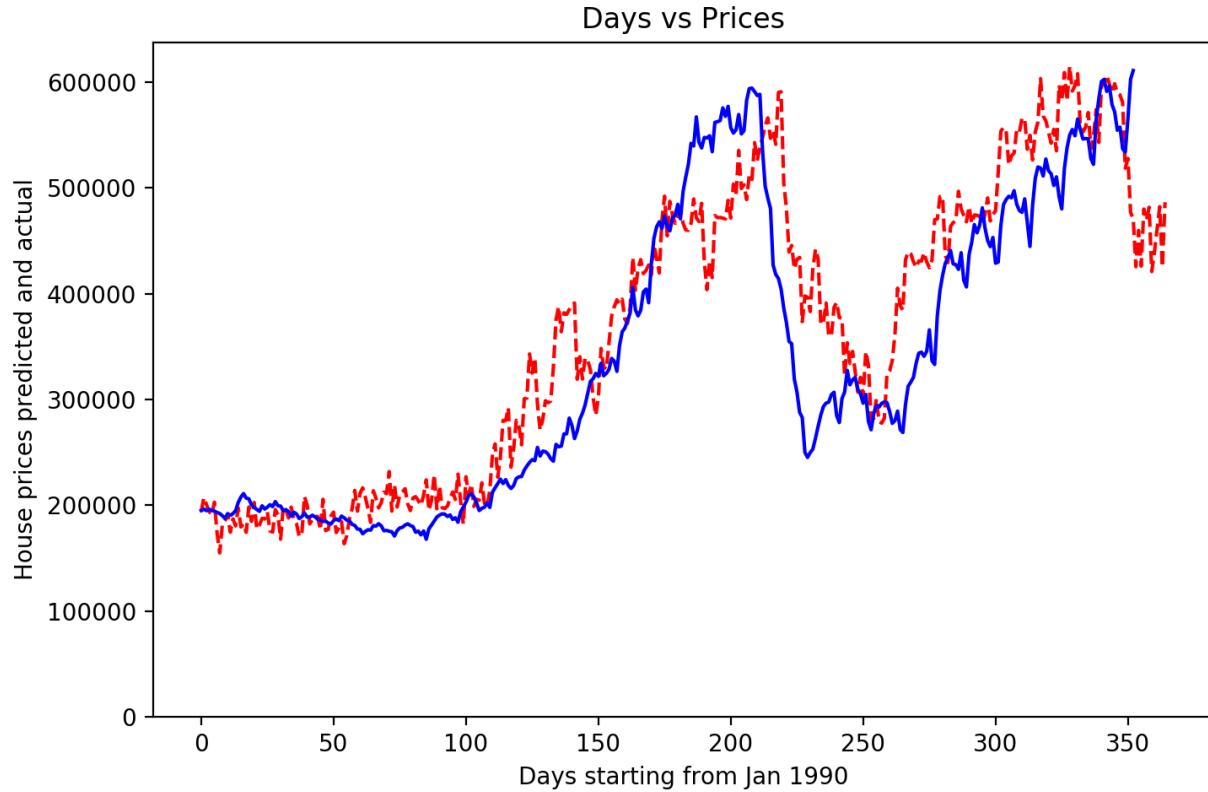


Fig 27: House Prices actual vs predicted by HMM

The next model prediction we are looking at is by Hidden Markov Model (HMM). The graph above is showing a drop in the house prices for the next coming year. The drop in the price is oscillating back and forth but is definitely lower than the current prices of the housing market. The prices in the prediction for a year is between \$425,000 and \$500000 which is \$100000 drop from the average house price in currently in California. The next plot shows the zoomed in version of the plot above to show the price prediction by the model.

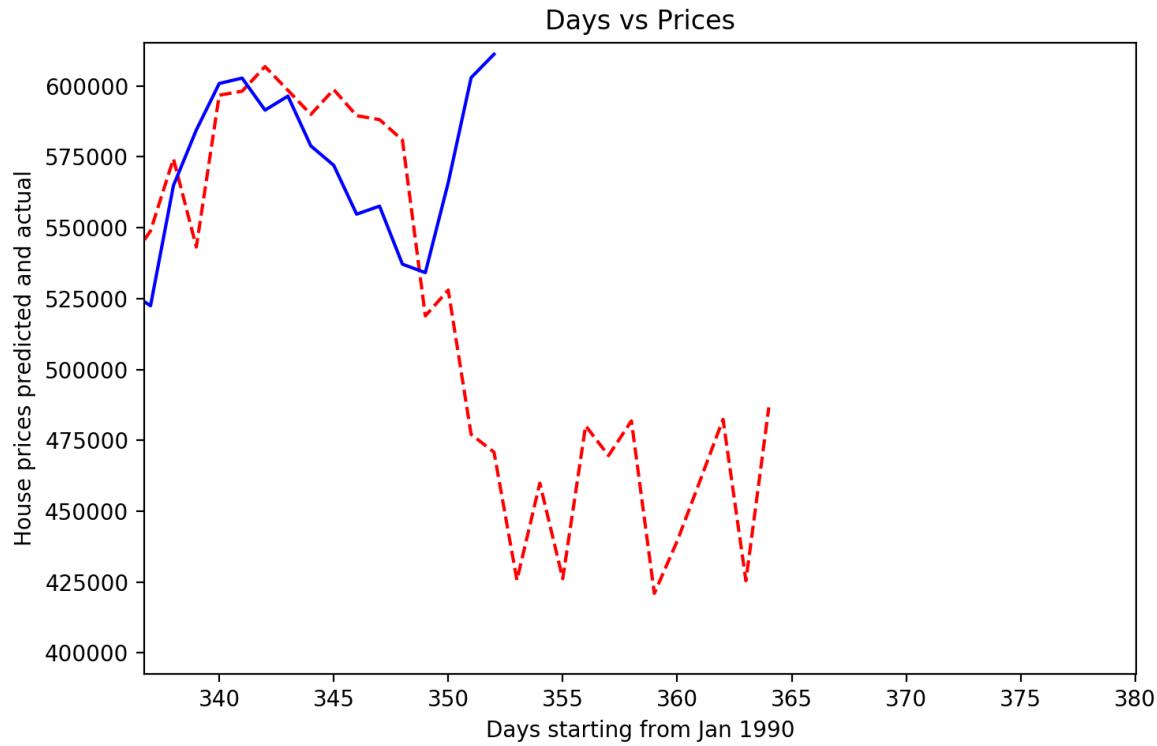


Fig 28: Zoomed in graph to show the prediction

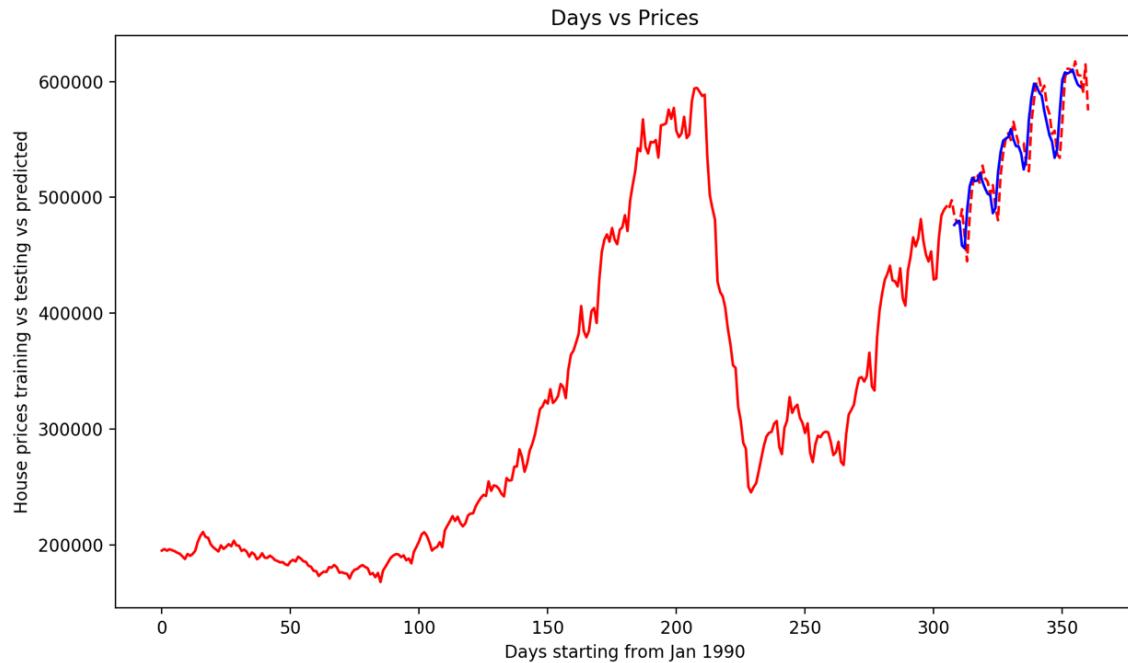


Fig 29: Graph of training, testing, and prediction using LSTM

The graph above shows the plot of the training data (represented by solid red line), testing data (represented by dashed red line) and, prediction made after training and testing (represented by solid blue line). From the graph we can see that the prediction made is very close to the testing data which is what we aim for. By far the prediction by our LSTM was the closest to that of the testing data.

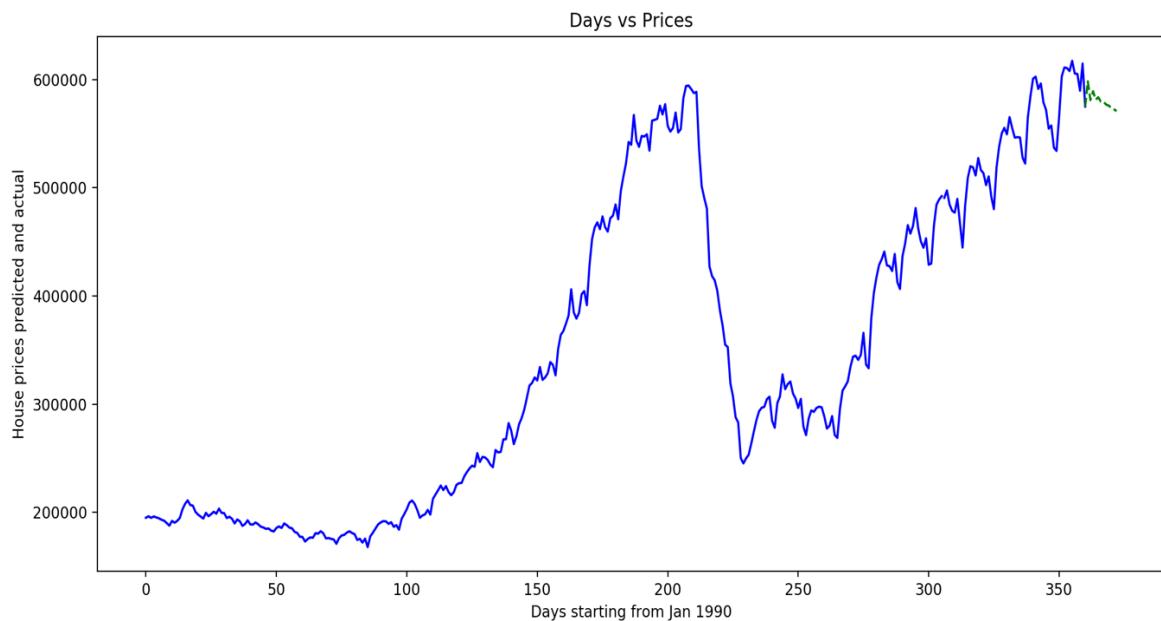


Fig 30: Graph show extended Prediction by LSTM

In the above graph we extend the prediction by for the next twelve months (represented by dashed green line). The prediction for the next one year shows that there will be a fall in the house prices. The fall is not as steep as that of 2008. In the next graph we will look into the zoomed in part of the prediction to see prediction prices.

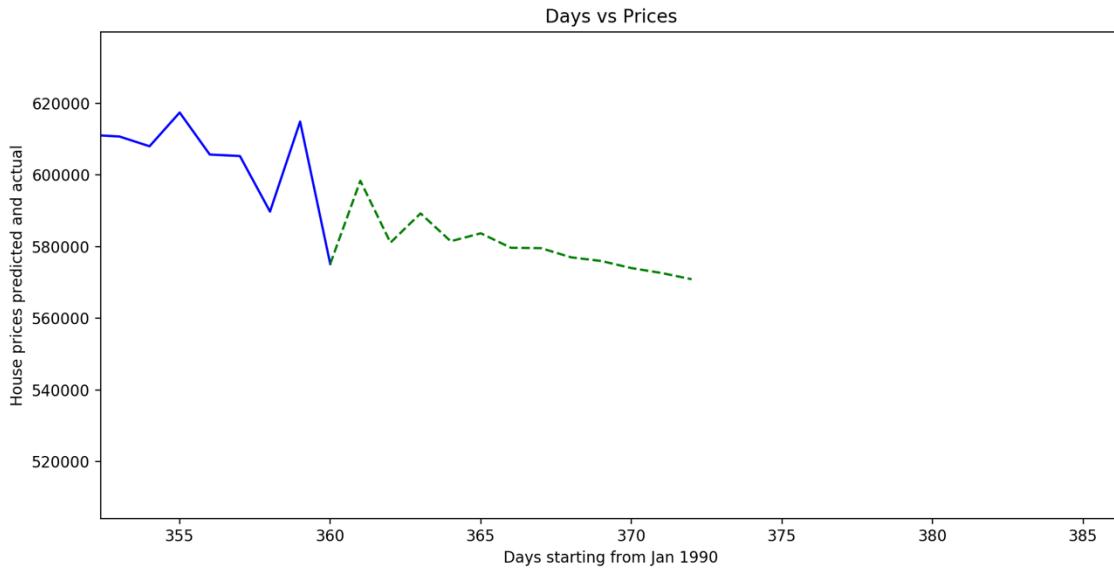


Fig 31: Zoomed in graph showing extended Prediction by LSTM

From the zoomed in graph we can see that the price lingers around \$570,000 – \$600,000 which is not a lot reduction from the current price of housing market of California. But if we look into the seasonal variation then we can see that during the summer the prices normally rises. Therefore, we can say according to the model the prices are going to fall.

Table 2: Model Comparison

<i>Model Name</i>	<i>Prediction</i>	<i>Time to train</i>	<i>Efficiency</i>	<i>R-Square Score</i>
Linear Regression	House prices will eventually rise	Low	Medium	0.76
HMM	House prices will fall	Low	Medium-Low	0.706
LSTM	House prices will fall slightly	High	High	0.92

5. CONCLUSION AND FUTURE WORK

Market recession and housing market crisis are closely tied together and have a huge impact on economy. The techniques discussed can help to forecast the housing prices for the future. From all the graphs and prediction model, we can foresee that there will be a fall in the house prices for the next year. Also, with everything going on now with coronavirus and stock market behaving in an unpredicted manner it's very unlikely to say how long will the price go on diving low. But it won't be as bad as that of 2008 because the banks this time around are taking every precaution to prevent a crisis like that of 2008. In this project we have built models using intelligent techniques these models can be extended to do more intelligent prediction with more data sets. \

We can use these models along with coronavirus data to make more knowledgeable prediction about how the coronavirus would affect the housing market in the future when the coronavirus data is available to us.

REFERENCES

- [1] Y. Demyanyk and I. Hasan, "Financial crises and bank failures: A review of prediction methods", *Omega*, vol. 38, issue 5, pp.315-324, 2010.
- [2] E.J. Schoen, "The 2007–2009 Financial Crisis: An Erosion of Ethics: A Case Study", *J. Bus. Ethics*, vol. 147, pp. 805-830, Dec 2017.
- [3] M. Zhang and K. Xu, "High order Hidden Markov Model for trend prediction in financial time series", *Physica A: Stat. Mech. and its Appl.*, vol. 517, pp.1-12, 2019.
- [4] M.R. Hasan and B. Nath, "Stock market forecasting using Hidden Markov Model: A New Approach", *5th Intl. Conf. on Intel. Sys. Design and Appl., IEEE*, 2006.
- [5] F.A. Gers, D. Eck, J. Schmidhuber, "Applying LSTM to time series predictable through Time-Window approaches", *Perspectives in Neural Comput., Springer*, vol. 1, pp. 193- 200, 2002.
- [6] Y. Hu, X.Sun, X. Nie, Y. Lweand L. Liu, "An Enhanced LSTM for Trend Following of Time Series", *IEEEAccess*, IEEE, 2019.
- [7] Y. Demyanyk, "Quick exits of subprime mortgages" *Fed. Res. Bank of St. Louis Rev.*, vol. 92, 2008.
- [8] M.G. Crouhy, R.A. Jarrow and S.M. Turnbull, "The Subprime Credit Crisis of 2007", *J. of Deriv*, pp. 81-110, 2008.
- [9] E.P. Davis, D. Karim, "Could early warning systems have helped to predict the sub-prime crisis?", *Ntl. Inst. Econ. Rev.*, vol. 206, pp. 35–47, 2008.
- [10] R.Nyman and P.Ormerod, "Predicting economic recessions using machine learning algorithms", Dec 2016.
- [11] Housing price dataset: <https://www.car.org/marketdata/data/housingdata>
- [12] Mortgage interest rate dataset: <https://fred.stlouisfed.org/series/MORTGAGE30US>
- [13] Total houses sold dataset: https://ycharts.com/indicators/new_homes_sold_in_the_us
- [14] M.Stamp, "A Revealing Introduction to Hidden Markov Models", Oct 2018.
- [15] C.Olah, "Understanding LSTM Networks", Aug 2015.