House Price Prediction using Zillow Dataset

Project Report

Arun Venkatesh
Department of CS
UCR
Riverside, California
avenk013@ucr.edu

Hsi-Min Chou
Department of CS
UCR
Riverside, California
hchou026@ucr.edu

Can Niu
Department of CE
UCR
Riverside, California
cniu005@ucr.edu

Fedah Alanazi
Department of CS
UCR
Riverside, California
falaan005@ucr.edu

ABSTRACT

Real estate has always played an important role in influencing the economy of a place. With real estate fluctuating rapidly subject to several factors, the economy can be badly affected. COVID-19 had only exacerbated this situation. Real Estate is at a loss due to this pandemic and people are wary of investing due to its fickle nature. This would have been easier if there were models to predict the otherwise volatile industry. We try to achieve this via our project.

KEYWORDS

Data mining, Linear Regression, XGBoost, k-nearest neighbours, Random Forest, Data Preprocessing

ACM Reference Format:
Arun Venkatesh, Hsi-Min Chou, Can Niu, Fedah Alanazi. 2021. House Price Prediction using Zillow Dataset. In. ACM, New York, NY, USA, 13 pages.

1 INTRODUCTION

In this project, we make an attempt to predict the house prices based on the square feet bought across regions. From the predicted values, we analyze various parameters such as variation of house prices with the area and compare how the regions fare with respect to one another. With this work, we hope to provide to users a way to predict the ever fluctuating real estate and make wise investment decisions.

We are using Kaggle Dataset on house prices to train and test our models. As a uniform step, we are training our model on the US data set and trying to test it across multiple countries' datasets. To achieve a stable model, we are taking multiple models into consideration and analyzing the results. Using these results, we are providing an efficient model to predict house prices across regions.

The models we are analyzing for the scope of this project are Linear Regression, KNN clustering, Random Forest Model and XGBoost Model.

Arun Venkatesh has worked on the Linear Regression model. In this analysis, he would be predicting prices based on the area in square feet. Based on his predictions, he would further analyze which regions have a stark increase in price as square feet increases. Such regions raise a red flag of being unstable and should be avoided.

Hsi-Min Chou has worked on the Random Forest Algorithm. In this project, she would predict prices based on year and county area. Through the prediction, she can get further analysis about prices increasing as the year

passes by. In addition, understanding different regions obtain different house prices.

Can Niu has worked on the XGboost Algorithm. In this project, she is working on predicting the house prices using Machine learning algorithms XGBoost considering factors such as city regions and time of the year in the county. Through the prediction, she can get further know of US house prices as the time goes by and even the US economy. Understanding the important features that influence the change of house prices in the US . In addition, She makes a simple comparison between American housing prices and those in her own home country.

Fedah Alanazi has worked on the K-Nearest-Neighbours regression algorithm. In our project , the machine learning algorithm (k-NN) was used to predict the house prices from a set of features in the Kaggle Dataset. This algorithm was selected from an assessment of previous research, and the intent was to compare their relative performance at this task. So, I predicted the House Prices using the trained dataset to understand the House types, Region, and Year.

2 RELATED WORKS

There has been a long history of research in the area of house prediction using various parameters and models. They all aim to perform and conclude on a variety of areas such as a comparison study or enhancing an existing data mining or a machine learning model(s). Some of the existing related works in this area are as follows:

[1] Using Linear Regression, Boost regression, Forest Regression and Neural Networks, this research paper predicts house prices in the state of Mumbai, India. The main idea on the conclusion of this paper is that the results that are produced is the weighted mean of all the techniques used in the paper. The paper has an approach that proves that by considering the weighted mean of all the techniques used, minimum error and maximum accuracy can be achieved rather than doing it separately. They are achieving the result by passing the results of the individual models into a neural network framework. Even though this model seems to give reasonable results, it has concluded future works by testing and enhancing its efficiency by simulating it using large datasets.

[2] This paper works on a combination of machine learning/ data mining methods such as Random Forest, XGBoost, Light Gradient BoostingMachine(LightGBM) alongside complex machine models like Hybrid regression and Stacked Generalization. Its main objective is to use both traditional and advanced machine learning models and find the differences between them in terms of performance and accuracy. The paper predicts house prices using the above models using varying parameters such as the learning rate, train-test split proportion and maximum depth(for tree based models). It concludes by saying that all the models, irrespective of the complexity, have their own pros and cons and should be decided based on the end result that one is trying to achieve.

[3] Using data mining techniques such as Extra Tree, Support Vector Machine, K-Nearest Neighbors, Naive Bayes, Logistic Regression, Voting Classifier and Stochastic Gradient Descent, this research paper is similar to [1], where it combines all the the models using blending function, combines them and produces the weighted average as the final result. It considers various features such as the square feet area of a house, its internal features such as electricity, water and gas. To predict the

accuracy and sensitivity of the model(s), it uses the concept of confusion matrix on the dataset. The paper concludes with a statistical inference on the algorithms by segregating them on the accuracy rate, precision rate, sensitivity rate and specificity rate. Looking at the tabulation, Voting Classifier seems to perform better on the dataset considered.

[4] This paper uses regression techniques such as multiple linear regression, Lasso and Ridge regression, support vector regression, and XGBoost to predict the house prices in Bangalore, India. The paper considers 9 different sets of features to train and predict on the price and test the accuracy of the models. It concludes on a neutral standing that accuracy is dependent on the dataset considered, the noise it has and all models show similar efficiency for the dataset considered,

3 PROPOSED METHOD(S)

The US dataset that we took had multiple parameters. In each model as described in the introduction, we train on the US dataset and use the same model on our local region dataset. We select features which we see fit for our models. Having predicted our data, we compare the inflation in each region to find the best places in the region we can invest in.

3.1 Linear Regression

This is one of the simple models for data prediction. This model uses two variables and predicts one variable based on the other. The coefficient/ the weights to be allotted to the feature is calculated and the best coefficient is selected.
This model was chosen because upon analyzing the dataset we found that the price is linearly dependent on the area(in square feet) over the years. A linear regression model is defined as follows:

$$Y = \theta X + C$$

Where θ is the coefficient/ the weight assigned to the feature X and C is a constant.
In this model, the Y is the price of the unit and X is the square feet area of the unit.

Based on our research, we found that Linear Regression is a potential contender for House price prediction. Since we wanted to implement a model from scratch, we have chosen the Linear Regression model. In this implementation, we have implemented the train and test function from scratch in contrast to using predefined methods.

3.2 Data Preprocessing(Linear Regression)

The US dataset had multiple features for which data was not reliable. For the purpose of this project,
(i) We cleaned our dataset so that we could focus on the features that we wanted. This includes the price information categorized bedroom wise.
(ii) This dataset did not have the squarefeet information for the bedrooms. As a preprocessing step, we have calculated the square feet area information using the price of each bedroom and the price per sqft detail in that region.
(iii) The last preprocessing that was done on this data is normalization. Both the price and the area can range widely. In order to make our results readable, we have normalized the price and the area information before we started our training and prediction.
The preprocessed data is shown in Fig 1..

	Date	Year	RegionName	ZHVIPerSqft_AllHomes	ZHVI_1bedroom	ZHVI_2bedroom	ZHVI_3bedroom	ZHVI_4bedroom	ZHVI_5BedroomOrMore	Square_Feet_1bedroom	
200254	2017-01-31	2017	10300	95.0	0.131527	0.088046	0.074646	0.060126	0.047808	0.424186	
200255	2017-01-31	2017	10420	89.0	0.040558	0.049851	0.064122	0.120021	0.069602	0.119568	
200258	2017-01-31	2017	10540	145.0	0.140887	0.130208	0.131187	0.139476	0.094223	0.205635	
200259	2017-01-31	2017	10580	134.0	0.149589	0.140253	0.122536	0.145757	0.089916	0.267356	
200263	2017-01-31	2017	10740	96.0	0.133169	0.127480	0.114332	0.145297	0.120802	0.422798	
...	...	...	...	...	...	...	...	...	...	...	...
204798	2017-05-31	2017	49620	112.0	0.112315	0.106151	0.096406	0.130591	0.030973	0.259187	
204799	2017-05-31	2017	49660	61.0	0.016420	0.012525	0.025328	0.046875	0.029581	0.164078	
204800	2017-05-31	2017	49700	154.0	0.210016	0.154514	0.170606	0.174403	0.127371	0.332058	
204802	2017-05-31	2017	49780	81.0	0.071921	0.043279	0.057879	0.055453	0.048156	0.291665	
204804	2017-05-31	2017	United_States	126.0	0.192447	0.140129	0.123696	0.187347	0.164129	0.418762	

1785 rows × 14 columns

Fig 1 Preprocessed Data Sample

3.3 Data Visualization

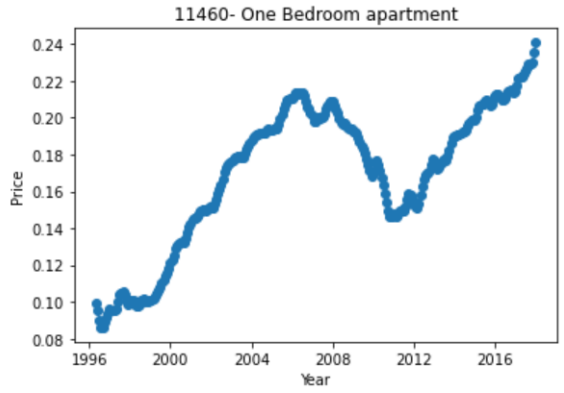


Fig 2 Area vs Price for the metro Ann Arbor,MI

With the data we got from the preprocessing, we took a particular region and visualized the relationship between year and price for a given sq ft range using a histogram..

Fig 2. shows the relationship between Year and

Price for the metro Ann Arbor, MI. We ran this visualization for all the metros and all the apartment types. We found that the relationship between Year and Price is linear in most of the cases. Based on this, the dataset qualifies for a linear regression analysis.

3.4 Methodology

In this project, we have developed the Linear regression model from scratch. We have implemented the train and predict functions on our own which are described in the following sections.

3.5 Training

Since the dataset is relatively small, we decided to split the data as 90-10. We are training on 90 percent of the data and testing on the rest 10 percent.

In the training process, upon adequate pre-checks to ensure our input(X) and output(Y) are of the same dimensions, we are using the least squares approach to find the coefficient(θ). Least squares is a standard approach to solve more equations than unknown, frequently known as overdetermined systems. In simple terms, the data we have is of the form,

$\theta_1 . X_1 = Y_1$

$\theta_2 . X_2 = Y_2$ and so on

We are expressing this in the form of a matrix multiplication to satisfy the equation $\theta.X = Y$. With this, we perform the least squares on it to identify the θ value.

3.6 Testing

Using the above generated θ coefficient, we predict the Y values of the test dataset(the remaining 20%) as a dot product of θ and X. We improve our θ value by calculating the Mean Square Error of the predicted values and based on that we do the following analysis.

With the predicted values, we then identify the highest inflation in the prices over the years for each region. Using this data, we rank the regions as the top 5 regions with highest inflation and 5 regions with least inflation. This extrapolation of data will help in identifying which regions are worthy of an investment. We have also calculated our accuracy on this prediction. We discuss these results in the "Experimental Evaluation" section of this report. Fig. 3 shows the predicted values for Elkhart in Indiana, US.

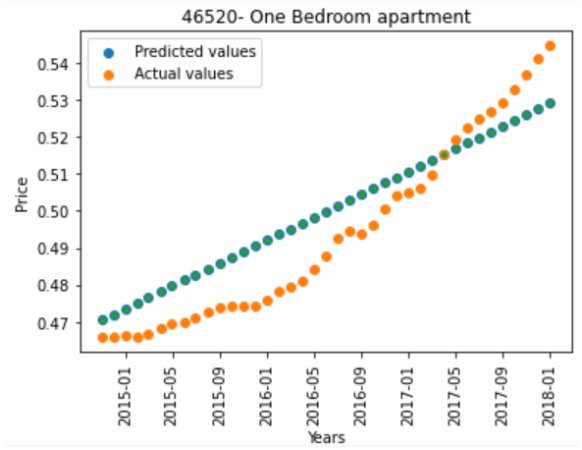


Fig 3 Predicted values based on Linear regression for Region Code - 46520

We have also used the same model on the Indian dataset to do a similar analysis. Fig 4 shows one such prediction graph for the location Delhi in India.

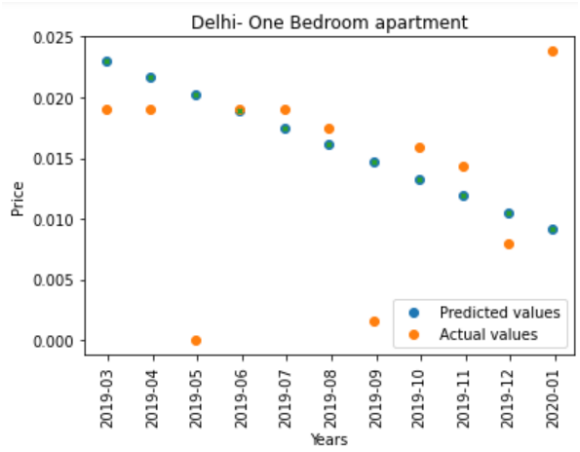


Fig 4 Predicted values based on Linear regression for Region - 'Delhi'

3.7 Data Preprocessing (Random Forest & XGBoost)

There are lots of datasets in the US house price file that we chose from Kaggle. In order to make the prediction more specific, we chose two datasets, which were County_time_series.csv and CountyCrossWalk_Zillow.csv, to work with our implementation.

(i) The first step was to read those two datasets into the model. Since we considered "Year" as our training and testing condition and the dataset showed only "Date", we added a row named "Year" to make the input data reach our requirement.

(ii) Secondly, we renamed some row names so it was able to merge two datasets into a new one. Additionally, we removed some rows (or features) that were not considered, so the data would be clean and easy to read.

(iii) Last but not least, it is important to deal with missing values, which might cause the model to stop or influence the accuracy. To solve this problem, we inserted the median values of the rows instead of removing them.

3.8 Data Visualization

Now we can finally use those preprocessed data to implement the model. In Fig 5, we can see

clearly from the chart presenting the house prices increase along with the year.

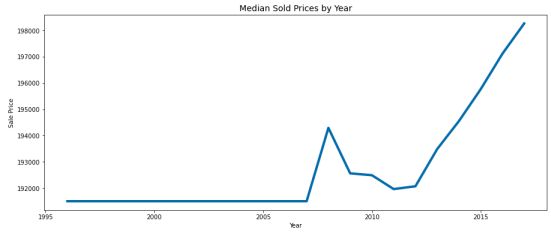


Fig 5 Median Sale Prices by Year

We can also get the information from Fig 6, which showed different tiers of real estate properties, also named Zillow Home Value Indices. ZHVI are the most accurate and family measures of residential real estate prices in the United States.

Through these charts, we can fully understand the trend of the US house price in the past few years from 1996 to 2018.

Fig 6 Real Estate Properties for Different Tiers in USA

3.9 Random Forest

Random forest is a supervised learning algorithm, a combination of multiple decision trees. To simplify, there are two kinds of machine learning algorithms, one is one-leader style and another is everyone-voted style. The most important parameter that we need to care about for a random forest is "n_estimators", which is the meaning of how many decision trees we want to create. The more trees we use, the better accuracy we predict, but the longer time it takes.

The reason I chose this algorithm is because I found it is often used for classification and regression tasks. For example, banks apply this model in order to find loyal customers, moreover, it can also be used for predicting stock behavior, which is similar to predicting house prices.

By using the Random Forest Algorithm to implement the training and testing function, we can get the prediction of future house prices and also learn the concept of the model. Furthermore, we can come up with some new ideas and improve accuracy by taking this algorithm as the foundation.

3.10 XGBoost

Based on the example given in Tianqi's paper, the XGBoost algorithm uses weak learners to form a strong learner, which means that each model will depend on the error of previous predictions. Weak learners are added sequentially until no further improvements can be made. This method matches the XGBoost tree to the residuals, and then uses a splitting technique to determine how well the clustering performs, and selects the best threshold for each feature. We follow the loss function for regression:

$$Obj = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j+\lambda} + \gamma T$$

The reason I chose this algorithm is because I found XGBoost is a widely used machine learning algorithm for the real-world problem, and it can give us pretty satisfying performances in most cases. So I want to try these algorithms in this project.

3.11 Training & Testing

We first shuffled the data before splitting into train and test and getting prepared for it. And then, we split the data into 'training' and 'testing'.

The test data is 20% of the total data and the train data is 80%. Then, we began to train the XGBoost model.

To implement the XGBoost algorithm , the most important step is to tune the parameters. In our case, the tuning parameters are included as below Fig 5.

```
xgboost = XGBRegressor(objective ='reg:squarederror',
                       colsample_bytree = 0.3,
                       learning_rate = 0.1,
                       max_depth = 5,
                       alpha = 10,
                       random_state=60,
                       n_estimators = 100)
```

Fig 5 XGBoost model with parameters

We have also used the same model on the Beijing dataset to do a similar analysis.

3.12 K-Nearest Neighbours

As we know, the KNN algorithm can be applied for either classification or regression problems. The KNN algorithm uses 'feature similarity' to predict the values of any new data points, which means the unique point is assigned a value based on how closely it resembles the points in the training set.

The reason that I chose the KNN algorithm is that to get better performance and more accurate estimation, I think feature-weighted KNN will be closer to the actual value. Abidoye's research[11] and our data picked some features to train the linear regression model. Though I think we still have some valuable features, due to the current progress of the project and the time limit, we have to make a compromise. We will use linear regression to find weights for each feature to get a better measurement of house similarity.

Actually, the choice of K is a factor of accuracy, When k=1 the result is the nearest neighbor value. Either too few or too many neighbors all will have an influence on the prediction.

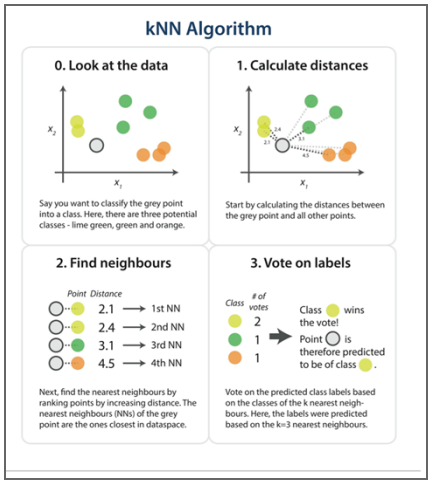


Fig 7 K-Nearest Neighbours working

First, we determine the K- value number, and then we compare the distance between classes' points. After that, see which class has more points inside the K area. Finally, we can decide that the target value is related to the class with more points and closest distance than the other. It's preferable to choose an odd number for K-value input number to avoid equality between classes that causes blurring of the classification.

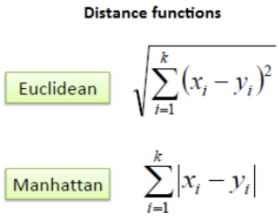
In the KNN technique, we will use X to present a *feature* (aka. predictor, attribute) and y to present the *target* (aka. label, class) that we are deciding to predict. KNN happens in the supervised learning family of algorithms. In other words, we are given a marked dataset that contains training observations (x, y) and would like to take the relationship between x And y. In addition, our goal is to learn a function $h: X \rightarrow Y$, so that given an unseen observation x $h(x)$ can easily predict the corresponding output y.

Calculating the distance between the current point and each training point that we have. There are several methods to calculate that distance, such as the most common methods – Euclidian, Manhattan (for continuous), and Hamming distance (for definite).[12]

1. **Euclidean Distance:** Euclidean distance is calculated as the square root of the

sum of the squared differences between a new point (x) and an existing point (y).

2. **Manhattan Distance**: This is the distance between real vectors using the sum of their absolute difference.



3. **Hamming Distance**: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

$$D_H = \sum_{i=1}^{k}|x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

$$x \neq y \Rightarrow D = 1$$

3.13 Data Visualization

In our data that we got from the preprocessing, we took a particular region and visualized the relationship between area and price using a histogram. In Fig 8 below, I have displayed few rows of our dataset without preprocessing

	Date	RegionName	DaysOnZillow_AllHomes	InventorySeasonallyAdjusted_AllHomes	InventoryRaw_AllHomes	MedianListingPricePerSqft_1Bedroom	MedianListingPricePerSqft_
0	1996-04-30	10001	NaN	NaN	NaN	NaN	
1	1996-04-30	10003	NaN	NaN	NaN	NaN	
2	1996-04-30	1003	NaN	NaN	NaN	NaN	
3	1996-04-30	1009	NaN	NaN	NaN	NaN	
4	1996-04-30	1017	NaN	NaN	NaN	NaN	

5 rows × 82 columns

Fig 8. Dataset without Preprocessing

After that I put all the data ranging from date '2000-01-01' taken along with the RegionName and House Prices as you can see in Fig 9 below.

	Date	RegionName	ZHVI_AllHomes
68575	2000-01-31	10001	98800.0
68576	2000-01-31	10003	128400.0
68577	2000-01-31	1001	91900.0
68578	2000-01-31	1003	128900.0
68579	2000-01-31	1009	102700.0

Fig 9 Displaying the data range based on some features

.

In Fig 10, I merged the datasets of two files Crosswalk and County_time_Series that have a common column, which is 'FIPS'/'RegionName', and both of them can be merged on that basis.

	Date	FIPS	AverageHousePrice	CountyName	StateName
0	2000-01-31	10001	98800.0	Kent	Delaware
1	2000-02-29	10001	99200.0	Kent	Delaware
2	2000-03-31	10001	99700.0	Kent	Delaware
3	2000-04-30	10001	100100.0	Kent	Delaware
4	2000-05-31	10001	100300.0	Kent	Delaware

Fig 10 Merged Columns in the Dataset

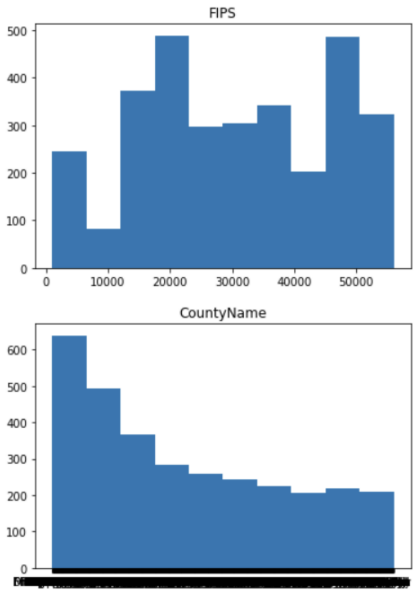


Fig 11 Histograms for all features in the crosswalk dataset

3.14 Methodology

In our project, we have tried implementing the K-nearest neighbours model from scratch. We have implemented the train and predict functions on our own. However, for some reason we could not get the results and the performance of this method.

4 EXPERIMENTAL EVALUATION

4.1 Linear Regression

Mean Square Error(MSE) is a good indicator of the quality of prediction. RMSE is a statistical measure which represents the standard deviation of data. In our analysis we check for MSE, RMSE, and MAE.

MSE: 0.0009350991902480893
MAE: 0.02537112000147549
RMSE: 0.030579391593818367

	Date	Actual	Predicted
0	2014-10-31	0.515311	0.535931
1	2014-11-30	0.518688	0.537627
2	2014-12-31	0.522966	0.539380
3	2015-01-31	0.526343	0.541133
4	2015-02-28	0.527769	0.542716
5	2015-03-31	0.528970	0.544469
6	2015-04-30	0.531747	0.546165
7	2015-05-31	0.535425	0.547917
8	2015-06-30	0.538127	0.549614
9	2015-07-31	0.540078	0.551366
10	2015-08-31	0.542330	0.553119
11	2015-09-30	0.544581	0.554815
12	2015-10-31	0.546082	0.556568
13	2015-11-30	0.546683	0.558264
14	2015-12-31	0.547733	0.560017
15	2016-01-31	0.549460	0.561770
16	2016-02-29	0.550811	0.563409
17	2016-03-31	0.551261	0.565162
18	2016-04-30	0.553287	0.566858
19	2016-05-31	0.558466	0.568611
20	2016-06-30	0.564770	0.570307

Table 1 Predicted and Actual Normalized values for the region 46520(Elkhart, Indiana) - US Dataset

We further predicted the regions with highest inflation rates for both the US(top 10) and India(top 2 owing to small data size). Below are our accuracy rates as calculated by the program code.

	1 BR	2 BR	3 BR	4 BR
US	60%	70%	70%	60%
India	50%	100%	50%	50%

4.2 Random Forest

In my implementation by using the Random Forest Algorithm, I chose several regions to make the results look more particular. For the accuracy of the prediction, I decided to apply the regression scoring parameter, which is "Root Mean Squared Logarithmic Error", as my evaluation standard. In Fig 12 , we can see that the actual prices were shown in blue dots and the predicted prices were shown in orange dots. Through the prediction without tuning, we got

the result of RMSLE is approximately 0.021142.

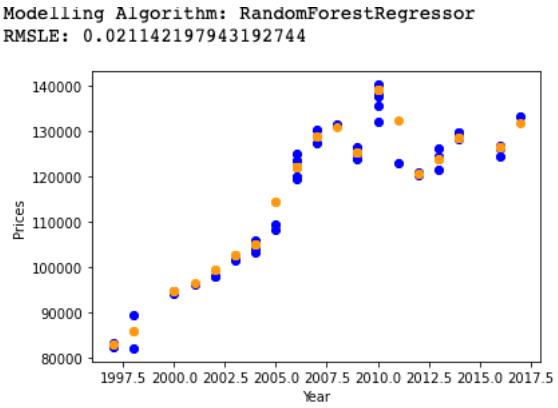


Fig 12 Predicted Prices and Actual Prices without Tuning

In order to understand the concept of a different number of decision trees might lead to different accuracy of prediction. I tuned the training data and set "n_estiamtors" to 200/500/800. After that, I printed the plots and analyzed the effects. Green dots were predicted prices after tuning while orange dots were the data we predicted before. Figures 13 and 14 below showed that the best prediction was when decision trees are equal to 200, RMSLE = 0.02010. However, the accuracy decreased when we created more trees.

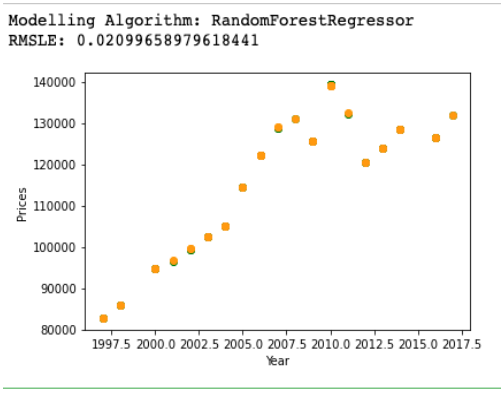


Fig 13 Predicted Prices w/o tuning VS Predicted Prices with 200 Decision Tree

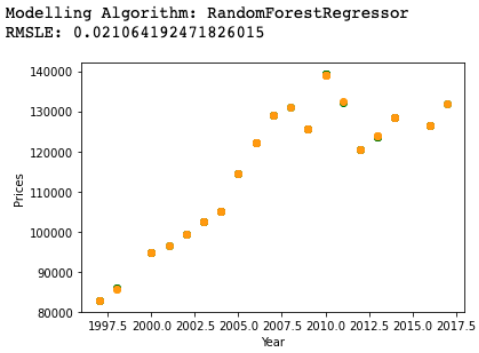


Fig 14 Predicted Prices w/o tuning VS Predicted Prices with 500 Decision Tree

4.3 XGBoost

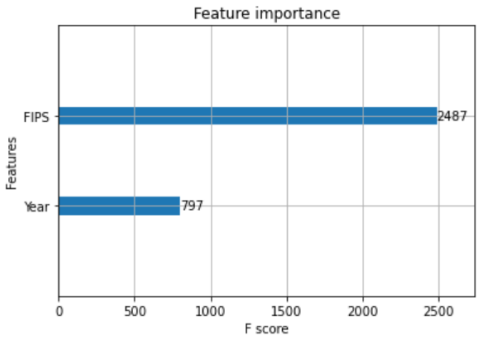


Fig 15 Feature Importance in US prices

From Fig 15, we can see the importance of each feature in the US house prices. FIPS is 0.7922696 and Year is 0.20773046. The FIPS has a more important effect on the change in US house prices than Years.

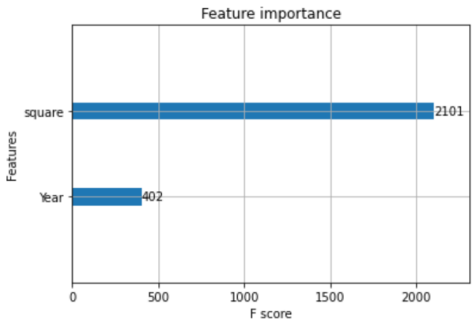


Fig 16 Feature importance in Beijing prices

From Fig 16, we can see the importance of each feature in the Beijing house prices. Square is

0.7124744 and Year is 0.28752556. We can get the conclusion that the square meter of a house is the most important factor in Beijing's house prices.

The result for US house prices predict				
	YEAR	FIPS	Model predicted price	The actual prices
288168	1998	48313	79967.375	76900
430774	2011	22037	114497.39	110900
509124	2013	45011	138378.42	119600
262372	2001	47097	79819.52	58400
176148	2015	36079	241297.58	302600
108655	2002	25003	220501.53	151800
505100	2014	19161	111631.92	119600
47917	2009	17057	131450.03	119600
433936	2015	19121	114890.34	119600
374442	2000	19017	80345.45	40900

Fig 17 XGBoost Result - US Dataset

The result for Beijing house prices predict			
	YEAR	Model predicted price	The actual prices
189659	2016	565308.4	704615.3846
215843	2016	779313.9	492307.6923
255809	2014	611264.56	861538.4615
234202	2015	398566.8	361538.4615
34255	2016	748554.6	692307.6923
208447	2012	850095.6	653846.1538
42599	2017	793755.06	715384.6154
23003	2016	508271.12	792307.6923
196041	2012	202844.27	293076.9231
115722	2016	901504.06	461538.4615

Fig 18 XGBoost Result - Beijing Dataset

After I ran my code, we can see the result from the above two Figures(17 and 18). The XGBoost model runs very fast. The model predicted prices are not exactly the same as the actual price, but they are close. There are also some results that are far from the real value. So I tried to tune the model parameters. I changed the max_depth and n_estimators, but the predicted results did not change too much. The RMSLE of this Modelling Algorithm is 0.20937668911938556.

Average price of US	Average price of Beijing
135966.27	535615.24
Minimal price of US	**Minimal price of Beijing**
46690.6	398011.78
Maximum price of US	**Maximum price of US**
810487.3	6120454

Fig 19 XGBoost Result comparison

From the above Fig 19 , we can see that the average house price of the US is $135966.27 and the average price of Beijing is $535615.24. Beijing's house price is much higher than the average of the US. Yes, the house price in my home country has been very high since 2008. The minimal price is almost the same. The maximum price in the US is $810487.3, which is higher than Beijing's maximum price.

4.4 K-Nearest Neighbours

I have been trying to test the k-NN with different values. For example, the value of the k number of neighbors to use for the algorithm compared with some data and prediction for some entries. In addition, I have the result of the correlation between a house price and FIPS which is (-0.09189268298777031).
Also, the prediction of Interest Rates in the neighborhood_time_series data set was (1695038, 76) by using the .shape() function.
Regrettably, the other results of the KNN model like test , train, and measuring its performance were unable to run in order to give us the best prediction with the current setup for the k-NN algorithm when k is ranging between any numbers.

5 DISCUSSIONS & CONCLUSIONS

5.1 Linear Regression

From the above evaluations, for linear regression, we have trained a model with a large dataset such as the US and used the same model

for predictions on a smaller dataset such as that of India's. While doing so, we took various performance parameters such as the MSE, RMSE and MAE into consideration and found that our predictions are close to the actual values. We also further extended our concept to identify the inflation rates over years in these two countries. We found that for the US we achieved a maximum accuracy of 70%. This shows that for this dataset, we could get only average performance. On exploring further, we found that though most of it has a linear relationship between price and year it is not strictly linear. Hence linear regression fails a little in these scenarios. Moving along, we would like to explore Logistic regression as an improvement to this and achieve a higher accuracy rate.

5.2 Random Forest

The Random Forest Algorithm can be an appropriate model on predicting our project. The accuracy of the prediction is high and we can get better results if we tune the data. However, the improvement decreases as the number of trees increases. In simpler words, there are redundant decision trees in the model and hence they will not help improve the performance.

There are a couple of ways to improve the model. In the project, we've already tuned the parameters by increasing and decreasing the number of estimators. In the future, we can also specify the maximum depth of the trees and the maximum number of features to be included at each node split.

5.3 XGBoost

In our project, exploratory analysis and data visualization play important roles. Because of these two steps, we can have a better understanding of features and finish the initial feature selection.

During the XGboost algorithm implementation step, we try to use different indicators to evaluate model performances. But the accuracy of the prediction is not very high, this may be because this model has a lot of parameters which makes tuning parameters complicated. There are some enhancements we can make in the future. Try to further look into the parameters, tuning it and get better results. In addition, we can include more information for prediction. For example, the supply and demand that affected the price, the inflation rate, the unemployment rate, even the seasonality can be another important feature. By including more relative features, I believe model performance will be improved.

5.4 K-Nearest Neighbours

The k-nearest neighbors (KNN) algorithm is an uncomplicated-to-implement supervised machine learning algorithm that can solve both classification and regression problems. The KNN algorithm assumes that similar things are near to each other. However, k-NN scales unsuccessfully with more observations and features. Meanwhile, we used this model to predict the values for the entire properties dataset, and it doesn't work as we want. As a next step, we hope to use more techniques, more experiences to establish this k-NN model within production and test it on the Kaggle dataset.

There are some suggestions to do in the future. Firstly, having more property features can be applied in this project, such as providing pictures and house descriptions to get a more precise prediction of the target. Further, using more combinations of complex metrics can be used to give more deeper conclusions.

ACKNOWLEDGEMENTS

We thank Prof. Papalexakis for all his lectures, his guidance, accommodation, and support for making us complete this project successfully

REFERENCES

[1] Ayush Varma, Abhijit Sarma, Sagar Doshi, Rohini Nair, House Price Prediction Using Machine Learning And Neural Networks, (2018)

[2] Quang Truong, Minh Nguyen, Hy Dang, Bo Mei, Housing Price Prediction via Improved Machine Learning Techniques, (2019)

[3] Maharshi Modi, Ayush Sharma, Dr. P. Madhavan, Applied Research On House Price Prediction Using Diverse Machine Learning Techniques, (2020)

[4] Manasa J, Radha Gupta, Narahari N S, Machine Learning based Predicting House Prices using Regression Techniques, (2020)

[5] Examensarbete Inom Teknik, Grundnivå, 15 Hp Stockholm, Sverige Zillow Prize: Competing to Improve the Zestimate|Data, (2019)

[6] Shujia Zhao, Implementation and Study of K-Nearest-Neighbour,(2018)

[7] Niklas Donges, The Random Forest Algorithm: A Complete Guide to the Random Forest Algorithm | Built in, (2019)

[8] Tony Yiu, Understanding Random Forest - How the Algorithm Works and Why it is so Effective, (2019)

[9] Tianqi Chen and Carlos Guestrin, XGBoost: A Scalable Tree Boosting system, (2016)

[10] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, Mu Li, Junyuan Xie, Min Lin, Yifeng Geng, and Yutian Lee, XGBoost: Extreme Gradient Boosting, (2019)

[11] Rotimi Boluwatife Abidoye, Factors That Influence Real Estate Project Investment: Professionals' Standpoint, (2016)