Zillow Home Value Prediction

Sumanth Chinnamudiam
Fundamentals of Data Science
(CS 6780)
Department of Computer Science
Atlanta, GA, United States
send2sumanthcm@gmail.com

Manish Kolla
Fundamentals of Data Science
(CS 4740)
Department of Computer Science
Atlanta, GA, United States
maneeshkolla@gmail.com

Hemanth Gorrepati
Fundamentals of Data Science
(CS 6780)
Department of Computer Science
Atlanta, GA, United States
ghemanth2001@gmail.com

Abstract— **With the increase in house prices year to year causing several factors of the economy such as inflation, higher interest rates, increased expense of raw materials. We have decided to build a machine learning model which is resistant to most of the market trends using sophisticated measures of imputation and model making to decrease the error as much as possible. Some of the models we have experimented with our data are Random Forest and Linear Regression. For the data imputation, we have expermed through various methods which include replacing with the median of the state prices, KNN imputation, forward/backward filling.**

Keywords: *Prediction, Random Forest, Time Series, KNN Imputation, mean square error, absolute error, K Fold Cross Validation, forward/backward filling*

Introduction

The rapidly rising cost of housing in the United States has significant implications for the economy, driving inflation, higher interest rates, and increased expenses across various sectors. To better understand these dynamics and predict future trends, we employed data science methodologies on a comprehensive dataset from Zillow, a leading real estate platform. Through thorough data exploration and visualization, we identified intriguing patterns, including unexpected house price drops in specific states during certain periods. This sparked our interest in delving deeper and developing a model for house price prediction. Under the insightful guidance of Professor Rafael, we implemented sophisticated data processing techniques, including effective null value replacement strategies. This meticulous approach ensured the accuracy and reliability of our analysis. Our research is poised to contribute valuable insights into the complex dynamics of the US housing market, enabling informed decision-making for individuals, businesses, and policymakers alike.

Code Link:
https://colab.research.google.com/drive/1Z2Gb1Qsr7VTAcUg0Cw1QFr3bXej3hbk4?usp=sharing

Google Drive (Data):
https://drive.google.com/drive/folders/11Ap-y5FHtPeJnoGu4oDddtiJq9kcamgI?usp=drive_link

GitHub Link:
https://github.com/manishkolla/Zillow-Home-Value-Prediction

Literature Survey:

N. N. Ghosalkar and S. N. Dhage 2018 [1], Real Estate Price value using Linear Regression are using simple Linear Regression technique to give the price value for the houses. Through this paper they have tried to have best fitting line (relationship) between the factors of the real estate taken into consideration and used various mathematical techniques like MSE (Mean Squared Error), RMSE(Root Mean Squared Error)

Sangani et al. [2], where they examined the effectiveness of several machine learning models and techniques, to decrease the error of price estimation done by Zillow, a real estate listing website. The models chosen were linear regression and gradient boosting models. They used the property data to train these models, with which they made predictions for other properties.

M. Jain, H. Rajput, N. Garg and P. Chawla 2020 [3] is also a house price prediction system using some techniques. In this they have used the simple process of machine learning from data cleaning, visualization, pre-processing and using k-fold cross validation for the output results. Finally, they have displayed the graph that shows close resemblance with actual price and the predicted price showing decent accuracy through their working model.

To predict the cost of resale homes, P. Durganjali[4] suggested using classification algorithms. The selling price of a property is predicted in this study using a variety of classification methods, including Linear regression, Decision Tree, K-Means, and Random Forest. A home's price is influenced by its physical attributes, its geographic location, and even the state of the economy. Here, they apply these techniques, use RMSE as the performance matrix for different datasets, and find the best accurate model that predicts better results.

Life Cycle of this Project

Throughout our project we have worked on following the CRISP-DM life cycle using the following phases

1. Business Understanding (Understanding the final deliverables)
2. Data Understanding (Data Availability, Features understanding and selection)
3. Data Processing (handling null values, molding the data, and merging them)
4. Modeling (Implementation of several ML Algorithms on the data)
5. Evaluation (Measuring the MAE, R2, MSE)
6. Deployment (Implementing a prediction function with the help of ensembling)

Business Understanding:

Helping individuals who are planning to purchase a house in any state across the United States with a certain number of bedrooms and Metro Area, and RegionID to predict the future house prices including the factor of inflation effects.

Data Understanding:

1. All the datasets have same exact schema of columns and formatting.
2. Each dataset has the following columns: 'RegionID','SizeRank','RegionName', 'RegionType', 'StateName', 'State', 'Metro', 'StateCodeFIPS', 'MunicipalCodeFIPS', and the prices for each month starting from January 2000 till September 2023 which is 285 months in total.
3. All the rows are unique and there are no duplicate entries.
4. The only categorical feature with null values is Metro which is approximately equal to 30%, the best imputation measure is to replace the null values with zero and non null values as 1, because if the certain area comes under a metro area then it is categorized as 1 else 0.

The sizes of the datasets are:

1. Dataset 1 with 1-bedroom houses:
   - Rows: 1490, Columns: 294
   - 108 columns with null values greater than 50%
     - The years with those null values are from 2000-2008
   - 4 columns with null values greater than 65%
     - The months of these null values from Jan 2000 till April 2000.
2. Dataset 2 with 2-bedroom houses:
   - Rows: 2532, Columns: 294
   - 107 columns with null values greater than 50%
     - The years with those null values are from 2000-2008
   - 0 columns with null values greater than 65%
3. Dataset 3 with 3-bedroom houses:
   - Rows: 2805, Columns: 294
   - 107 columns with null values greater than 50%
     - The years with those null values are from 2000-2008
   - 0 columns with null values greater than 65%
4. Dataset 4 with 4-bedroom houses:
   - Rows: 2522, Columns: 294
   - 98 columns with null values greater than 50%
     - The years with those null values are from 2000-2008
   - 0 columns with null values greater than 65%
5. Dataset 5 with 5-bedroom houses:
   - Rows: 1815, Columns: 294
   - 77 columns with null values greater than 50%
     - The years with those null values are from 2000-2006
   - 0 columns with null values greater than 65%

Quality Check: Each individual dataset has all the 51 US states, and the merged dataset has 1684 unique counties.

Data Preprocessing:

Data Merging:
Merging all the 5 datasets into a unified single dataset with the addition of a new feature to each individual dataset which features "Bedroom" which contains the number of the numbers. For instance, using value 1 for 1 bedroom dataset, and 2 for 2 bedroom dataset and so on. After adding this new feature to each of the individual datasets then concatenate them to a single dataset with dimensions (11164, 295). The merged dataset has 102 columns with null values greater than 50% and those columns are ranging from 2000 to 2008.

Drop the features:
We're following the drop of any feature data preprocessing method as we see that there are more than 60% of the missing values present in the columns. So we have removed the columns with missing values greater than 60% that is from dates 1/31/2000 to 1/31/2009 which helped us to implement other data preprocessing techniques which makes the data clean and train the machine learning model.

Missing Indicator feature:
We have a feature called Metro which provides the information of the house whether it is in the metro area or not and we found missing values in this feature, so we're using the missing indicator feature method by representing binary values 0 and 1. If the given metro feature is not null we're replacing the value with binary value 1 and if the feature value is null then we're replacing it with binary 0.

Background for Data imputation:
Some of the initial imputation methods we came up with are

1. Replacing the null values with the median data of the state with the same number of bedrooms.
2. KNN Imputation using uniform and distance measure.
3. Using a simple imputer module using median of the column
4. Using forward and backward filling in each region id/row.

The problem we faced with median of the state data is, the data is being skewed and the linear regression and random forest models are yielding a higher mean absolute error of over $100,000. Another issue with this approach is, all the regions in the same state are being imputed with the same value regardless of their geographic standard. For instance, let's imagine there are two regions of a state which have null values, one with higher cost of living and another with least cost of living, this median imputation replaces both the nulls with the same value which is not accurate to either of them.

The problem with KNN are, it relies heavily on the distance between points. Outliers can significantly skew the distance calculations and lead to inaccurate imputations. KNN Imputation tends to favor the majority class when imputing missing values. This can bias the imputed values towards common values, which may not accurately reflect the true distribution of the data. Furthermore, computing the nearest neighbor for unscaled data is very computationally huge.

The problem with the simple imputer is, that it is replacing the missing values with the median/mean of the column which includes a wider distribution of states and regions and does not give a right metric of replacement.

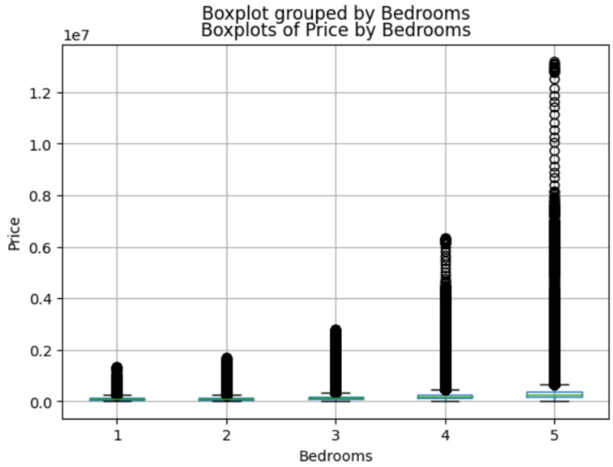
The final approach we have decided on which is not as well 100% accurate but at least we were able to convince ourself with this approach of backward and forward filling, which fills the missing values if its closest possible value either by forward or backward search. This works by using instance-based imputation which recursively scans each row for null values and replaces the missing with its closest value to the date. However, this is one of the accurate measures it has a potential drawback of computation time being higher computation time. But this is the most accurate model we were able to deploy and also the mean absolute error with this model, we were able to see a positive decrease in the error rate for ML models.

Melting the Data:
Melting data is a crucial data preprocessing technique in data science, which is converting a wide-format dataset into a long-format dataset. When working with datasets where information for each observation is dispersed across several columns, this restructuring is especially helpful. Because each row in the melted format usually represents a distinct combination of variables, it is easier to apply to specific analyses.

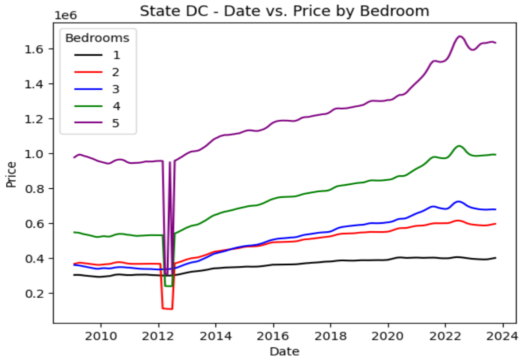
For the model to run efficiently, we have decided to use this melting data technique to an easier way for the model to be efficient, which is removing most of the data columns and changing the dimension to vertical. Focusing on the RegionID, State, Bedrooms, and the prices from 2009 to 2023.

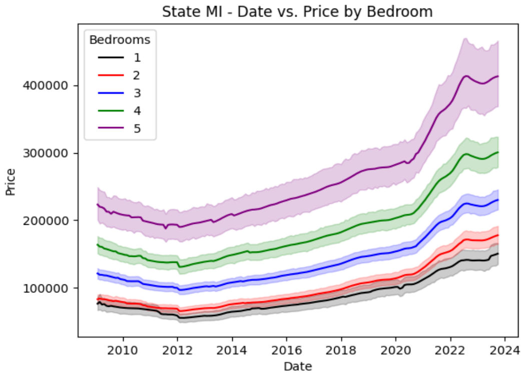
Label Encoding:
Label encoding is a preprocessing method that's especially helpful for categorical features. Every distinct category or label in the "state" feature is given a distinct number value in the context of label encoding. In addition to making categorical data representation easier to understand, this transformation also makes it compatible with machine learning techniques that need numerical input.
In this project we're label encoding the state feature to make the categorical data into numerical data.

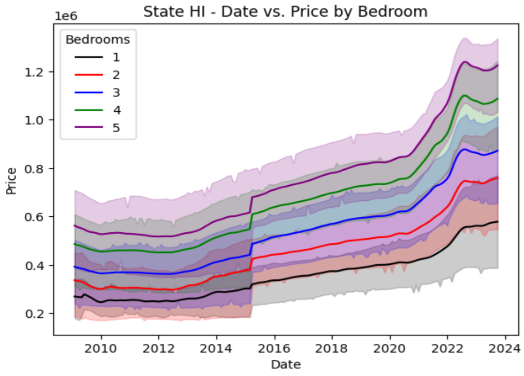
Outliers:
We've box plotted the data based on the number of the bedrooms and price of the house and found that there are more outliers present in the boxplot as the price of the house vary from county to county and from state to state, so we're not removing the outliers nor clamping the outliers with the upper bound values of the box plot as the model will be trained differently and predict incorrect values.

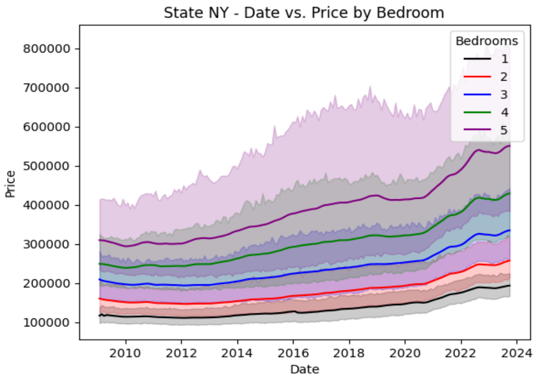


Line Plots
Initial Data Analysis plotted with the prices vs the years differentiated with the number of bedrooms:









Model Selection and Evaluation:

Models Used:

1. Train Test Random Split (80:20)
   a. Linear Regression
   b. Random Forest
   c. Random Forest with K Fold Cross Validation
2. Time Series Split (using 2021 as dividing factor)
   a. Linear Regression
   b. Random Forest
   c. Random Forest with K Fold Cross Validation

Evaluation methods for the below models:

Mean Squared Error:

MSE is the most used and very simple metric with a little bit of change in mean absolute error. Mean squared error states that finding the squared difference between actual and predicted value.

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

It gives a measure of the average squared difference between the expected and actual values and penalizes larger errors more severely. Better model performance is indicated by lower MSE values.

R Squared Error:

R2 score is a metric that tells the performance of your model, not the loss in an absolute sense that how many wells did your model perform.

In contrast, MAE and MSE depend on the context as we have seen whereas the R2 score is independent of context.

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

RSS= Sum of Squares of Residues

TSS= Total Sum of Squares

Mean Absolute Error:

MAE is a very simple metric which calculates the absolute difference between actual and predicted values.

The average absolute deviation between the expected and actual values is given. Compared to MSE, MAE is less susceptible to outliers, and lower MAE values denote superior model performance.

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

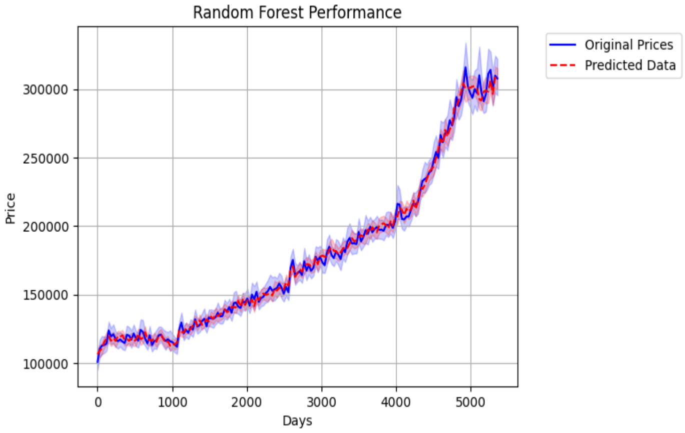
Machine Learning Models with Training Testing Random Split:

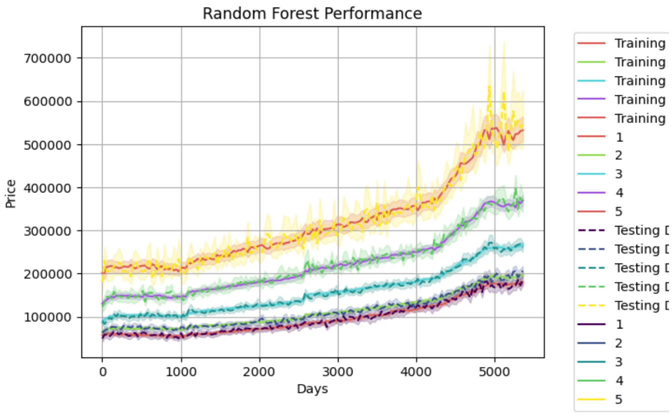
Linear Regression with Training Testing Random Split:

The first machine learning model we chose to implement for the house price prediction is linear regression and we followed training and testing data split which is used to evaluate how this linear regression model performs with unknown data. Two subsets of the dataset are used in the process: one is used to train the model and the other to evaluate its predicted accuracy. We have used the train_test_split function for this, dividing the data into training and testing sets by random shuffling. The evaluation Metrics like Mean Squared Error (MSE) and R-squared (R²), and Mean Absolute Error (MAE). are used to assess the model's performance in this project, after it has been trained on the training set by linear regression. Adjusting the split ratio with the test_size parameter allows for flexibility in balancing the trade-off between training and testing data.

Random forest with Training Testing Random split:

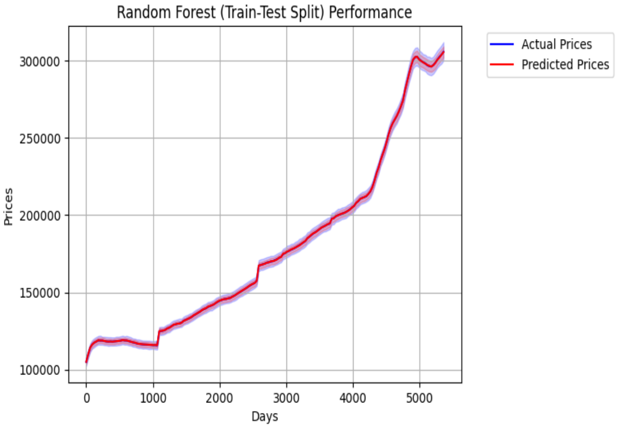
Random Forest using a testing-training data split to two subsets: a testing set that is used to assess the Random Forest model's performance on unobserved data, and a training set that is used to construct the model. This is often accomplished by using the train_test_split function, which partitions the data at random. During the training phase, the Random Forest algorithm generates an ensemble of decision trees, each of which makes predictions on its own. The predictive accuracy of the model or model evaluation is then evaluated on the testing set using measures like R-squared (R²), Mean Squared Error (MSE), and Mean Absolute Error (MAE). The power of Random Forest resides in its capacity to manage intricate relationships and identify non-linear patterns in the data. We've increased the model's performance by adjusting hyperparameters such as the number of trees (n_estimators). This approach offers a thorough assessment of the Random Forest's capacity to provide precise predictions and generalize new data.

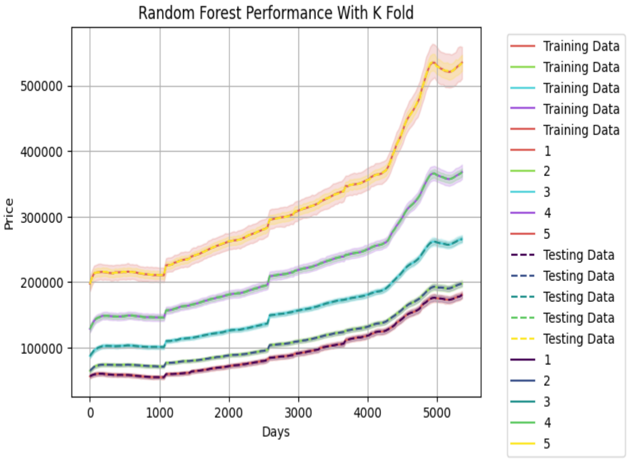




Random Forest with K Fold:

A thorough method for evaluating a Random Forest is to use Random Forest together with k-fold cross-validation and a random split for training and testing. Using methods like train_test_split, the dataset is initially randomly divided into training and testing sets in this process. The training set is then split into k subsets, or folds, using k-fold cross-validation. Using k-1 folds for training and the remaining fold for validation, the Random Forest model is trained k times. In order to guarantee that every fold functions as a training and validation set, this operation is done k times.





Time Series Split:

A specific type of cross-validation used in data science for models trained on time-ordered data, like time series, is called time series split. By addressing the temporal dependencies that are present in time series data, it confirms a linear ordering of training and testing sets. The main goal is to assess the model's performance on data that follows the training set, simulating real-world situations in which models are used to forecast observations to come.

Machine Learning Models with Time Series Split (2021):

Linear Regression with Time Series Split:

Rather than employing a random split of data for training and testing, we opted for a time series-based split using the year 2021 as the dividing point. Data prior to 2021 was exclusively used for training purposes, while data after 2021 comprised the test set. This approach ensured a temporally consistent evaluation of the model's performance. Notably, the data split was implemented through a custom function rather than relying on existing libraries.

Once the training and testing sets were established, we implemented a linear regression model. To evaluate the model's predictive accuracy, we utilized metrics such as R-squared (R^2), Mean Squared Error (MSE), and Mean Absolute Error (MAE) on both the training and testing datasets.

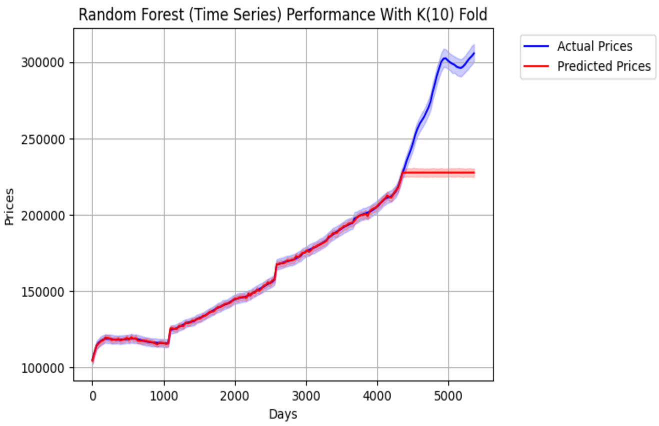
Random Forest with Time Series Split

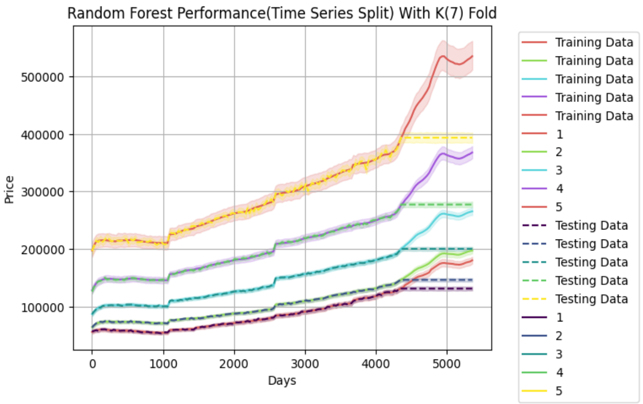
Just like Linear Regression, we have also deployed the same methodology of splitting for the Random Forest using 125 estimators and built a model using these testing and training sets. Here are some of the model predictions and original values. However, there is a potential drawback which we faced in this approach. After a certain number of days the model is typically predicting a constant value for all the rest of the date which is one of the interesting findings we have discovered. Due to certain time restraints for this project we were unable to further deep dive into and improve the model. But this is one of the things we are focusing on to improve in the future.

Random Forest with Time Series Split w/ K Fold (7 Folds)

Similar to the previous Random Forest model, this model utilizes the time series split technique for training and testing. Additionally, we implemented K-fold cross-validation with 7 folds to partition the data, further enhancing the model's performance. This strategy led to a significant improvement in accuracy, as evidenced by higher R^2 and lower Mean Absolute Error (MAE) compared to the model without K-fold cross-validation.

Some of the visualizations are provided below, depicting the original prices alongside the predicted prices generated by the K-fold cross-validated model.





Evaluation:

Here is the summary of all the Machine Learning Algorithms and its accuracy in terms of Mean Absolute Error and r2. According to our observation Random Forest with K fold cross validation using the train test split has attained a lower MAE and a higher r2.

Method	**Linear Regression**	**Random Forest**	**Random Forest with K Fold**
With train test split (80:20) (5-fold)	MAE: 87722.54, r2: 0.21	MAE: 67058.19, r2: 0.38	MAE: 63592.3 r2: 0.45
With Time series split (2021) (7-fold)	MAE: 113293.51, r2:0.09	MAE: 90213.48, r2: 0.34	MAE: 61666.07, r2: 0.37

Summary of Evaluation:

Best Model: Random Forest with K Fold using the Train Test Split

Average Mode: Random Forest with K Fold using the Time Series Split

Worst Model: Linear Regression with Time Split

Model Ensembling

Our models encountered limitations in achieving higher accuracy due to several data deficiencies. These limitations included the absence of crucial categorical features such as zip codes, house area, and location (city vs. suburban). This lack of information hindered the development of a more accurate model. The only meaningful features available were state, number of bedrooms, and metropolitan area. Building a higher accuracy model with only these four features was not feasible.

However, we optimized our model for efficiency within the limitations of the available data. This optimization resulted in a maximum accuracy of 45%. To improve prediction accuracy, we employed an ensemble approach. This approach involved generating predictions from both linear regression and random forest models with K-fold cross-validation. By averaging the predictions from all models and incorporating an inflation factor, we gave equal weight to each model and utilized their combined predictive power. This approach ensured that no potentially accurate model was disregarded.

Conclusion:

Among the implemented models, Random Forest with K-fold cross-validation yielded the highest accuracy, achieving an error rate reduction of over 50% compared to the Linear Regression model and a 20% reduction compared to Random Forest without K-fold cross-validation. This performance was observed both when the data was split using the standard training/testing split and the time series split. Notably, for the time series split, the error reduction achieved by the K-fold cross-validated Random Forest model surpassed 50% compared to both the Linear Regression model and the Random Forest model without K-fold cross-validation.

Another promising approach for this type of data involves running separate models for each house type and region ID. While this approach has been implemented in a prototype model, it poses a potential risk of overfitting due to the uniqueness of region IDs within the data. Further investigation is required to mitigate this risk and optimize the model's performance.

Overall, the findings suggest that Random Forest with K-fold cross-validation offers a robust and accurate approach for predicting housing prices. Exploring the potential of using region-specific models warrants further research to balance accuracy with the risk of overfitting.

Future Works:

We are dedicated to continuously improving our model's performance. To address the issue of the constant predictions in our K-fold cross-validation Random Forest model, we plan to conduct an in-depth investigation into the underlying causes. This investigation will involve analyzing factors such as feature selection, parameter tuning, and potential overfitting. By addressing these issues, we aim to significantly enhance the model's ability to generate accurate and varied predictions.

Furthermore, we intend to augment the dimensionality of the dataset by incorporating additional relevant features. This will provide the model with a richer and more comprehensive data landscape, potentially leading to improved predictive power. Additionally, we plan to explore the use of more sophisticated feature engineering techniques to extract latent information and further enhance the model's performance.

Enhancing Prediction Accuracy with Future Inflation Rates:

Currently, our model utilizes a standard inflation rate. However, we envision a future where the model leverages the power of predicted inflation rates for each specific prediction year. This will be achieved by utilizing the Consumer Price Index (CPI) values from the past few years to predict future inflation rates. By incorporating these predicted inflation rates into the model, we anticipate a significant increase in prediction accuracy. This approach will enable the model to dynamically adjust its predictions to account for the evolving economic landscapes, leading to more realistic and reliable outcomes.

Overall, our continuous research efforts and planned enhancements are directed towards achieving a robust and highly accurate model capable of providing valuable insights for future price predictions.

References

[1] R. Agrawal, “Evaluation Metrics for Your Regression Model,” *Analytics Vidhya*, May 19, 2021. https://www.analyticsvidhya.com/blog/2021/05/know-the-best-evaluation-metrics-for-your-regression-model/

[2] A. Kumar, “Random Forest for prediction,” *Medium*, Jun. 22, 2020. https://towardsdatascience.com/random-forest-ca80e56224c1

[3] D. Sangani, K. Erickson and M. A. Hasan, ”Predicting Zillow Estimation Error Using Linear Regression and Gradient Boosting,” 2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), Orlando, FL, 2017, pp. 530-534, doi: 10.1109/MASS.2017.88

[4] “Real estate value prediction using linear regression,” Real Estate Value Prediction Using Linear Regression, https://www.semanticscholar.org/paper/Real-Estate-Value-Prediction-Using-Linear-Ghosalkar-Dhage/f2308f0a4f0981801b518b9ca2152bcb4c797ad7 (accessed Dec. 12, 2023).

[5] A. Bhagat, M. Gosavi, A. Shaahsane, N. Mishra, and Prof. A. Nerurkar, “House price prediction using machine learning,” SSRN, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4413863 (accessed Dec. 11, 2023).

[6] J. Brownlee, “KNN imputation for missing values in machine learning,” MachineLearningMastery.com, https://machinelearningmastery.com/knn-imputation-for-missing-values-in-machine-learning/ (accessed Dec. 11, 2023).

[7] House resale price prediction using classification algorithms | IEEE ..., https://ieeexplore.ieee.org/document/8882842 (accessed Dec. 12, 2023).