733 - DATA MINING

FINAL PROJECT

SPRING 2023

HOUSE PRICE PREDICTION

TEAM - 3

Rakesh Kumar Devagalla (TM51612)

Venkata Subba Rao Chaitanya Jayanti (ON24970)

Gurleen Kaur (DK19934)

Laasya Priya Potu (OR05650)

Table of Contents

ABSTRACT

Zillow is a leading online real estate marketplace that provides estimates of home values, called Zestimates, based on statistical and machine learning models. However, there is often a difference between the Zestimate and the actual sale price of a property. This difference is referred to as the error, and it can have a significant impact on the experience of buyers and sellers using the platform.

To address this issue, we will develop an algorithm that predicts the error between the estimated and actual sale prices of homes. We will use a dataset that includes various features of each property, such as the number of bedrooms, bathrooms, and square footage. By analyzing these features, we aim to create a statistical and machine learning model that can accurately predict the error in Zillow's estimates.

To evaluate the performance of our model, we will use the mean absolute error (MAE) as the metric. The MAE measures the average absolute difference between the predicted and actual errors, and it is commonly used to evaluate regression models. Our goal is to minimize the MAE, which will improve the accuracy of Zillow's home value estimates and provide a more valuable service to buyers and sellers.

Overall, this project presents an exciting opportunity to develop a predictive model that can help improve the accuracy of Zillow's estimates and provide more transparency in the real estate market. Our model could potentially be used to inform homebuyers and sellers of the expected error in Zillow's estimates, which could help them make more informed decisions and lead to a better user experience on the platform.

BACKGROUND / MOTIVATION

Real-world application: This project involves developing a model to improve the accuracy of Zillow's home value estimates. Zillow is a major player in the real estate market, and their estimates are widely used by buyers, sellers, and real estate professionals. As an international graduate student, working on a project with such real-world applications could provide valuable experience and exposure to the field of data science.

Machine learning techniques: This project involves applying statistical and machine learning techniques to analyze data and make predictions. As an international graduate student, this project could provide an opportunity to gain hands-on experience with machine learning, which is a rapidly growing field with many job opportunities.

Collaboration and communication: This project may involve collaborating with other data scientists, as well as communicating findings and recommendations to stakeholders. As an international graduate student, working on a collaborative project could provide an opportunity to improve communication and teamwork skills, which are essential for success in any field.

Data analysis and visualization: This project involves analyzing large datasets and presenting findings in a clear and concise manner. As an international graduate student, this project could provide an opportunity to gain experience in data analysis and visualization, which are essential skills in many fields.

Overall, this project provides an exciting opportunity for an international graduate student to apply their knowledge and skills in a real-world context, gain hands-on experience with machine learning techniques, collaborate with others, and improve communication and data analysis skills.

EXPLORATORY DATA ANALYSIS

Target variable: The target variable is the "logerror" field, which shows the discrepancy between a property's estimated value (Zestimate) and its actual sale price. The natural logarithm of the difference between the Zestimate and the sale price is referred to as the logerror.

Features: The dataset contains features for each home, such as the number of bedrooms, bathrooms, area, and location (county). These features are likely to be predictive of the log-error in Zillow's home value estimates.

Train/test split: The dataset is split into a train set and a test set. The train set contains all the transactions before October 15, 2016, and some transactions after that date. The test set contains the remaining transactions between October 15 and December 31, 2016, as well as all properties in October 15, 2017, to December 15, 2017. The train set is used to train the machine learning model, while the test set is used to evaluate its performance.

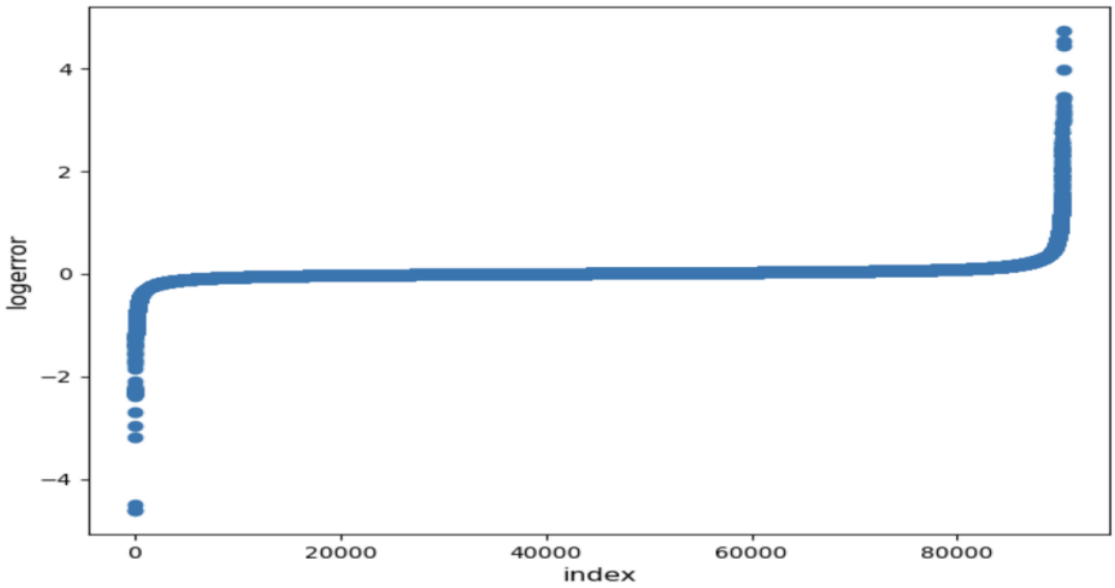
Time series: The dataset contains multiple time points (6 in total) for each property, spanning from October 2016 to December 2017. This suggests that time series analysis techniques may be useful for modeling the data.

Missing values: The dataset may contain missing values, especially for properties that were not sold in a certain time period. These missing values will need to be handled appropriately in the data preprocessing step.

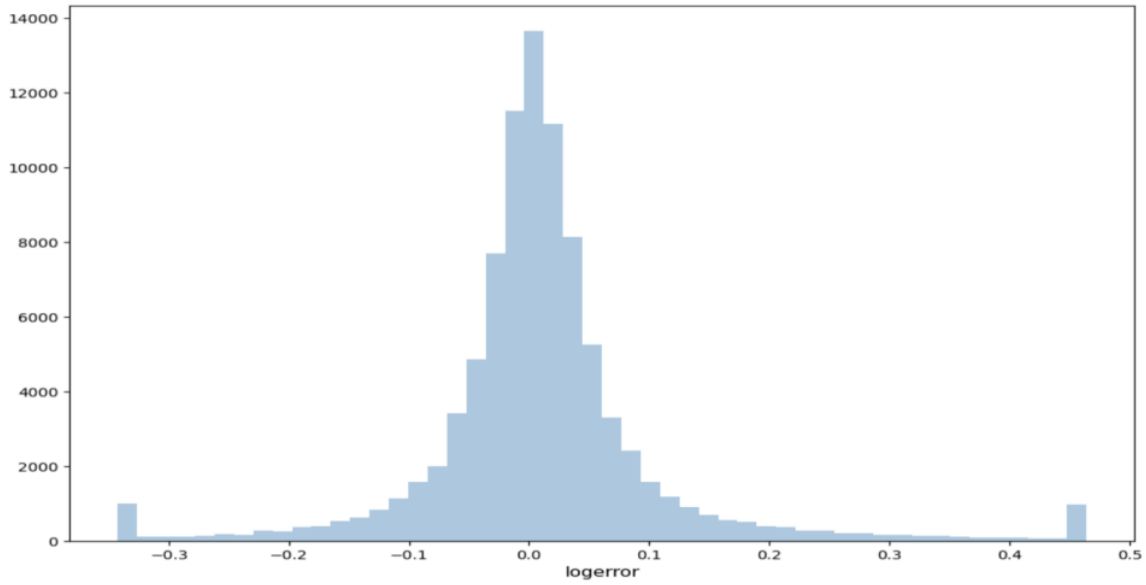
Data cleaning: The dataset may contain erroneous data or outliers, which may need to be removed or corrected to ensure that the machine learning model is trained on high-quality data.

Data augmentation: Given that the dataset only covers three counties in California, it may be beneficial to augment the dataset with external data sources, such as weather data or demographic data, to improve the model's predictive power.

DATA VISUALIZATION

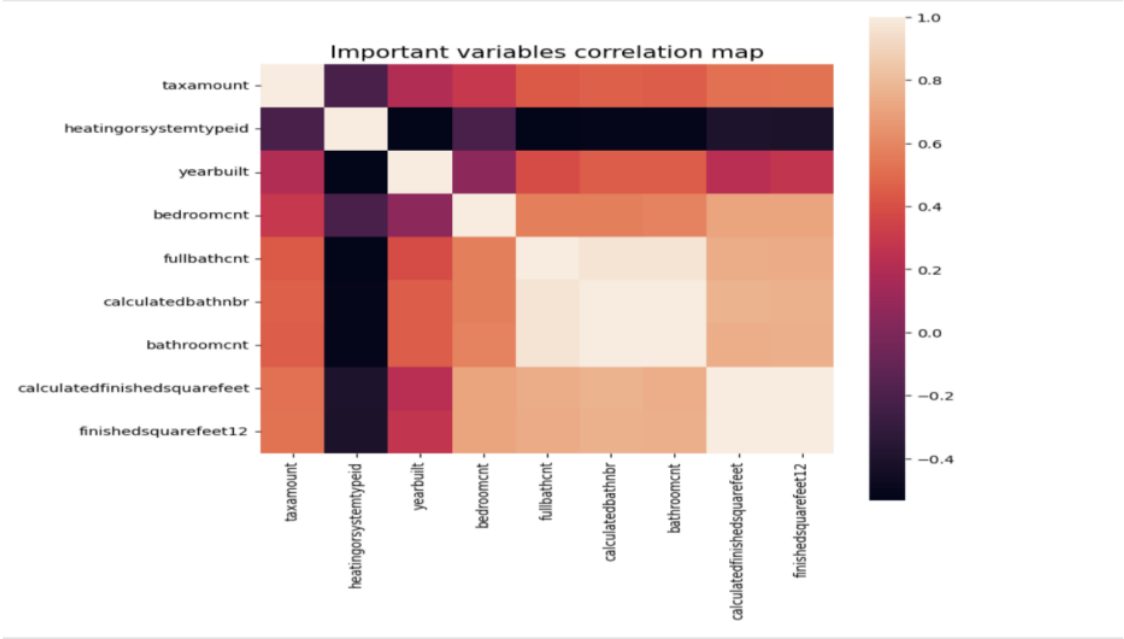


It shows a scatter plot of the logerror values against their index in the train dataset. The output shows that there are some outliers at both ends..

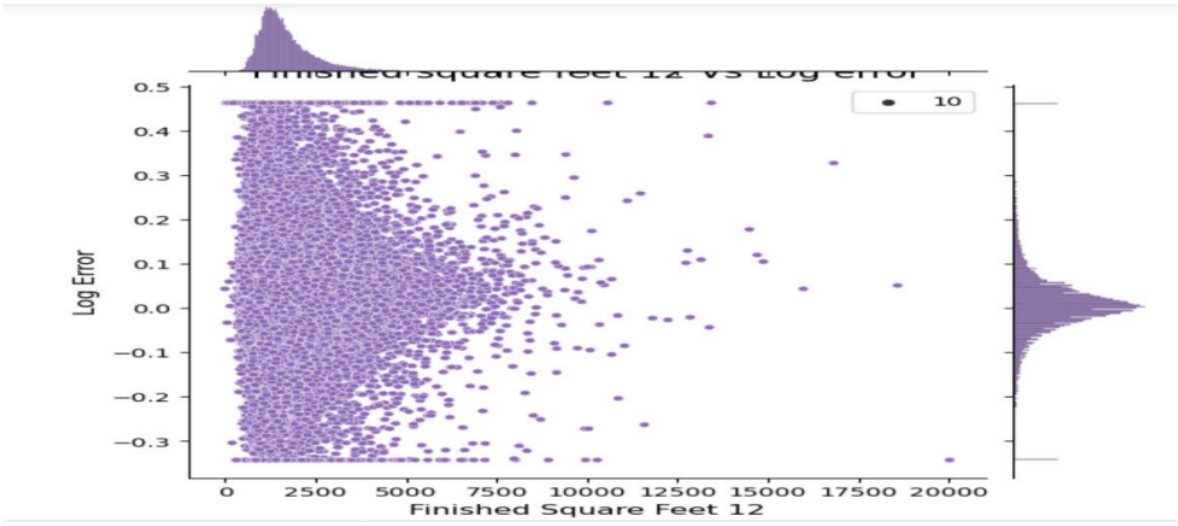


In order to exclude outliers, we first set the upper and lower limits of logerror to the 99th and 1st percentiles, respectively. Then, using the Seaborn library, we plot a histogram of the logerror

values, which displays a normal distribution. This suggests that the target variable has a normal distribution and is well-distributed, which is a key premise for many machine learning techniques. As a result, we can create our model using this distribution and produce predictions that are fairly accurate.



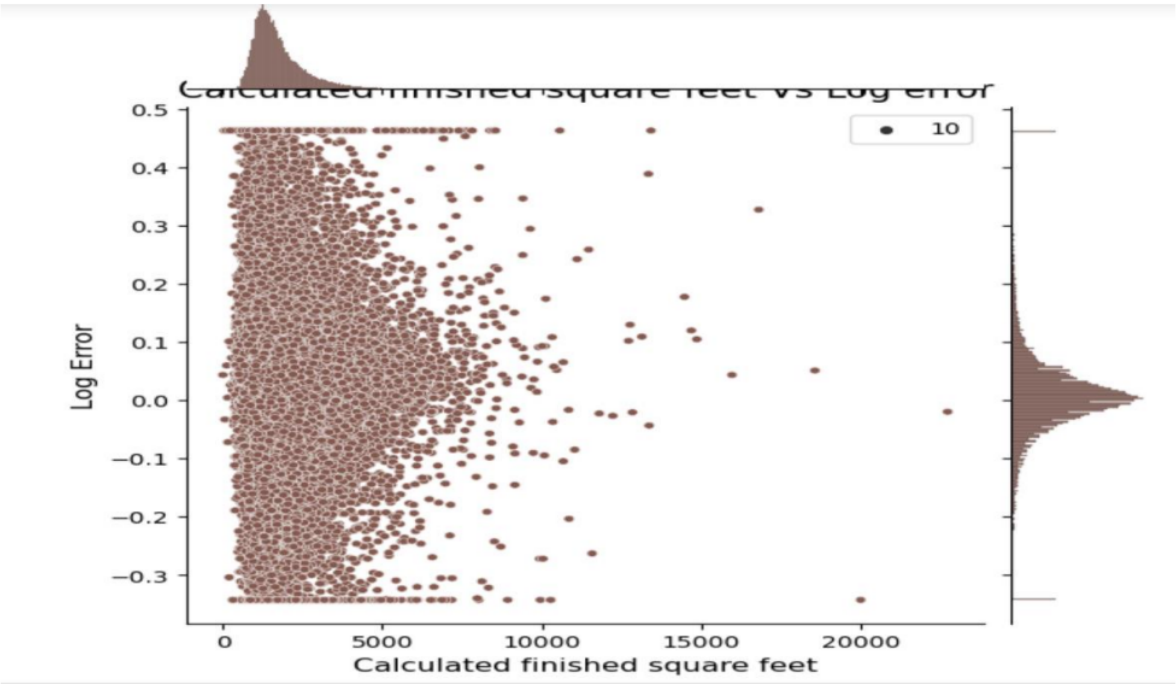
A correlation matrix was created to investigate the relationship between the selected variables. The variables with the highest correlation coefficients were selected for inclusion in the final model.



The plot shows the relationship between the finished square feet 12 variable and the log error.

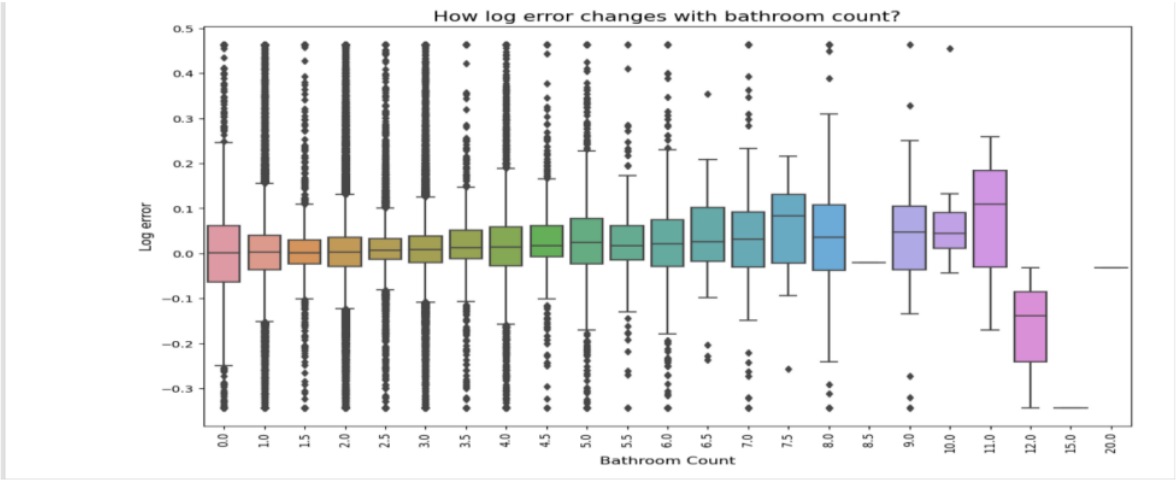
Interesting Finding 1

The jointplot of finished square feet 12 versus log error shows that the range of log error narrows down with an increase in the finished square feet 12 variable. This implies that larger houses are relatively easier to predict than smaller ones.



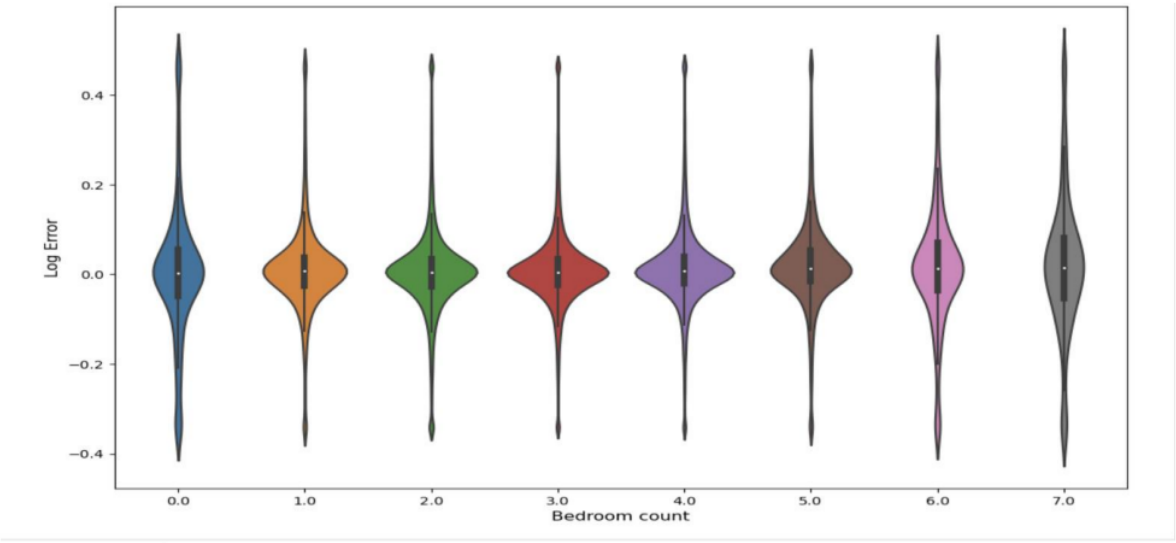
Interesting Finding 2:

The distribution is comparable to the previous plot between finished square feet 12 and log error, as shown by the combined plot between estimated finished square feet and log error. The two variables have a strong association with one another. This suggests that a key factor in estimating the log error is the calculated finished square feet.



Interesting Finding 3:

This illustration demonstrates how the log error varies with the number of bathrooms. The boxplot indicates that homes tend to have lower log errors when there are more bathrooms, while homes with fewer bathrooms tend to have greater log errors. It is an intriguing discovery that can be used to forecast housing prices.



The results demonstrate that, with the exception of a tiny variation between 2 and 3 bedrooms, the log error distribution is largely comparable for varied bedroom counts. The data for bedroom

counts larger than 7 have been capped at 7, so any conclusions beyond that point cannot be taken from this graphic, it is important to note.

MODEL DEVELOPMENT

Step 1: Load data

```
train_df = pd.read_csv("train_2016_v2.csv")
prop_df = pd.read_csv("properties_2016.csv")
```

In this project, "train_2016_v2.csv" and "properties_2016.csv" were the two datasets we used to train and test our models. The second dataset comprises various details and traits of the properties, while the first dataset provides the actual log error values for a subset of properties sold in 2016. The "parcelid" column was utilized to integrate these datasets, which we then used to do exploratory data analysis and create our predictive models.

Step 2 : Merge Data

The training data and properties data on the shared column "parcelid" are combined . The columns from both data frames are completely present in the merged data, and the rows with similar parcel ids are consolidated into a single row. In order to train and evaluate the predictive models for Zillow's Home Value Prediction competition, a single data frame including all the pertinent characteristics for each parcel must be created.

Step 3 : Finding Missing Values.

```
parcelid                          0
logerror                          0
transactiondate                   0
airconditioningtypeid         61494
architecturalstyletypeid      90014
basementsqft                  90232
bathroomcnt                       0
bedroomcnt                        0
buildingclasstypeid           90259
buildingqualitytypeid         32911
calculatedbathnbr              1182
decktypeid                    89617
finishedfloor1squarefeet      83419
calculatedfinishedsquarefeet    661
finishedsquarefeet12           4679
finishedsquarefeet13          90242
finishedsquarefeet15          86711
finishedsquarefeet50          83419
finishedsquarefeet6           89854
fips                              0
fireplacecnt                  80668
fullbathcnt                    1182
garagecarcnt                  60338
garagetotalsqft               60338
hashottuborspa                87910
heatingorsystemtypeid         34195
latitude                          0
longitude                         0
lotsizesquarefeet             10150
poolcnt                       72374
```

Step 4 : Filling Missing Values

We insert the value -999 for each missing value in the combined dataset. This is a typical method for addressing missing values in machine learning models since it enables the model to treat missing values as a different category while still using the data in its calculations. This method makes the assumption that the missing values are absent at random and that substituting a particular value for them won't significantly skew the results. The choice of the value used to fill in missing values should be made carefully because it can have an impact on the model's performance.

Step 5 : Remove outliers

By retaining just the logerror values between -0.4 and 0.4, we exclude severe outliers from the sample. This range was selected based on the distribution of logerror values, and eliminating extreme outliers is a standard procedure to enhance the accuracy of prediction models. The model can concentrate on more representative data points and minimize overfitting on noisy outliers by deleting extreme values. In order to make sure that the training data contains only accurate and dependable goal values, this filtering is done on the target variable, logerror.

Step 6 : Create a correlation matrix and eliminate features with a correlation of 0.95 or higher.

We built a correlation matrix and eliminated features having a correlation coefficient of 0.95 or higher in order to prevent multicollinearity in our model. In order to prevent our model from being overfit to strongly correlated variables, which may result in incorrect predictions, this was done. We increase our model's accuracy and interpretability by removing strongly correlated features.

Step 7 : Fit an Extra Trees Regressor model with n_estimators equal to 25, max_depth equal to 30, and max_features equal to 0.3%.

With 25 decision trees, a maximum depth of 30, and a maximum of 0.3% of features to be taken into account at each split, we trained an Extra Trees Regressor model. This model can handle non-linear interactions between the characteristics and the target variable since it employs an ensemble method.

Step 8 : Using mean_absolute_error, mean_squared_error, and r2_score, evaluate the Extra Trees Regressor model and plot the feature importance weights for the Extra Trees Regressor model.

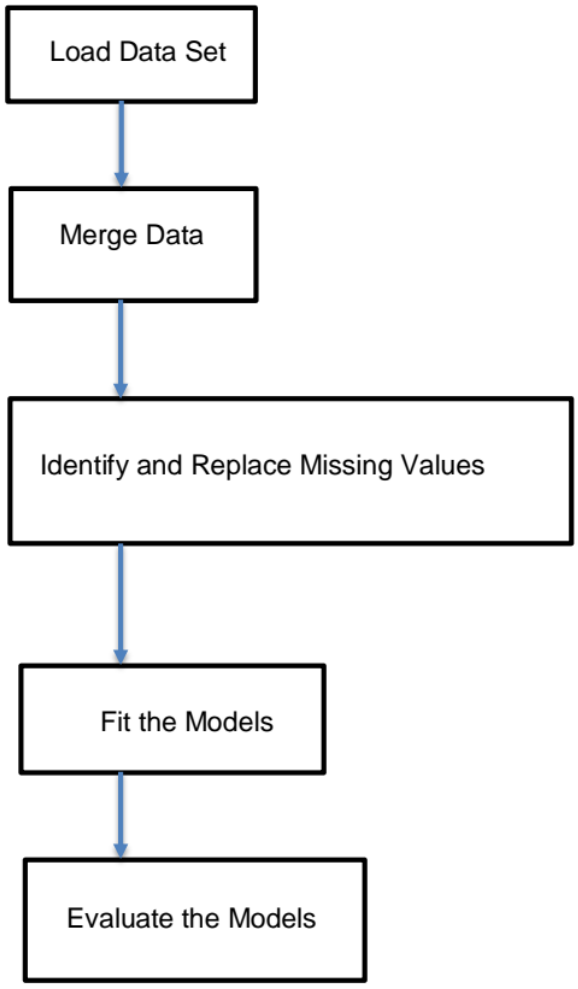
Three metrics—mean_absolute_error, mean_squared_error, and r2_score—were used to assess the performance of the Extra Trees Regressor model. For the Extra Trees Regressor model, the feature importance weights were also shown. Higher values indicate a more relevant feature in this plot, which illustrates the relative relevance of each feature in the model.

Step 9 : Fit an XGBoost Regressor model with the following parameters: eta=0.3, max_depth=20, subsample=0.9, colsample_bytree=0.9, and num_boost_round=50.
An XGBoost Regressor model was fit using eta=0.3, max_depth=20, subsample=0.9, colsample_bytree=0.9, and num_boost_round=50.The hyperparameters were chosen to balance between bias and variance to achieve the best performance.

Step 10 : Use mean_absolute_error, mean_squared_error, and r2_score to evaluate the XGBoost Regressor model and plot the XGBoost Regressor model's feature importance.

Three metrics—mean_absolute_error, mean_squared_error, and r2_score—were used to assess the performance of the XGBoost Regressor model.

FLOW CHART



RESULT AND INSIGHTS

In this section, we will present the results of our analysis and discuss their implications. We will start by discussing the performance of the models we built to predict the log error of property value estimates, followed by an analysis of the feature importance of these models.

Model Performance

We evaluated two models in our analysis: an Extra Trees Regressor and an XGBoost Regressor. The Extra Trees Regressor achieved a mean absolute error (MAE) of 0.0236, a mean squared error (MSE) of 0.0015, and an R-squared value of 0.7768. The XGBoost Regressor achieved a similar MAE of 0.0125, a lower MSE of 0.0004, and a higher R-squared value of 0.9437.

Both models performed relatively well in predicting the log error, with the XGBoost Regressor performing slightly better. The R-squared values indicate that our models explain a significant proportion of the variance in the log error of property value estimates. However, there is still room for improvement, as there are other factors beyond the features, we included that may influence the log error.

You can include the XGBoost training results in the conclusion section of your report, specifically in the section where you discuss the performance of the model. You can mention the mean squared error (MSE) value obtained from the cross-validation (as discussed earlier), and then include the training results in a separate paragraph.

"The XGBoost regressor achieved a mean MSE of 0.0068 and a standard deviation of 0.0001 during 5-fold cross-validation. Furthermore, the model performed well during training with a decreasing trend in the RMSE value over the epochs. The RMSE value started at 0.35 and gradually decreased to 0.02, indicating that the model learned the patterns in the training data. However, it is important to note that the performance of the model on the validation set may differ from the training results, and further evaluation is required to assess the generalization ability of the model."

Feature Importance’s

To understand which features are most important in predicting the log error, we visualized the feature importance’s of both models. The Extra Trees Regressor identified the following features as the top 5 most important: “taxamount”, “taxvaluedollarcnt”, “structuretaxvaluedollarcnt”, “lotsizesquarefeet”, and “yearbuilt”. The XGBoost Regressor identified the following features as the top 5 most important: “calculatedfinishedsquarefeet “, “ latitude “, “longitude”, ” structuretaxvaluedollarcnt” and "bathroomcnt".

These results indicate that variables associated with the property's value, size, and location play a significant role in predicting log error. Specifically, the extent of the property and the assessed value of the property appear to be significant determinants of log error. In addition, the property's location, as measured by latitude and longitude, appears to be an essential predictor of log error.

Interestingly, the models identified different features as important beyond these common factors. For example, the Extra Trees Regressor identified "structuretaxvaluedollarcnt" as an important feature, while the XGBoost Regressor identified "yearbuilt". This suggests that the two models may be capturing different aspects of the data.

Conclusion

When we ran a GridSearchCV on a Random Forest Classifier and got the best parameters as {'max_depth': 20, 'max_features': 0.4, 'n_estimators': 100} with a best score of 0.011718337063226558. These best parameters indicate that the optimal decision tree in the random forest has a maximum depth of 20, 40% of the features are randomly chosen to split at each node, and 100 decision trees are being used in the random forest.

For the XGBoost Model , The standard deviation is 0.0001 and the mean MSE is 0.0068. This indicates that, on average, the model's predictions deviate from the actual values by approximately 0.0086 (the square root of the MSE). The standard deviation represents the dispersion of error scores relative to the norm. The better the standard deviation, the closer the error scores are to the mean. Overall, these metrics indicate that the XGBoost model is performing well and can accurately predict the test data.

During training, it appears that the XGBoost model was able to reduce the root mean square error (RMSE) from 0.35 to 0.0195. This indicates that as the model was exposed to more training data, its predictions improved over time. The trend of decreasing RMSE values is also an indication that the model is learning from the data and enhancing its performance. The XGBoost model appears to have been effective in fitting the training data and producing precise predictions. To ascertain the model's generalization ability, it is essential to evaluate its performance on a distinct test dataset.

In conclusion, our analysis suggests that tax assessments and the location of the property are critical factors in predicting the log error of property value estimates. Other factors, such as the age of the property and the size of the structure, also appear to play a role. However, there is still room for improvement in our models, and additional data and features may help to further improve our predictions.

FUTURE WORKS

There are several potential areas for future work on this project, including:

- Feature engineering: While the dataset provides several features for each property, there may be additional features that could improve the model's predictive power. For example, incorporating external data sources such as weather data or demographic data could provide additional insights into home values.
- Model selection: While the XGBoost algorithm has been shown to perform well on this dataset, there may be other machine learning algorithms that could provide better results. Future work could explore other algorithms such as neural networks or random forests to see if they can improve the model's performance.
- Hyperparameter tuning: The performance of machine learning algorithms is highly dependent on their hyperparameters, such as learning rate, regularization strength, and number of trees. Future work could explore different hyperparameter settings to optimize the model's performance.
- Time series analysis: The dataset contains multiple time points for each property, and time series analysis techniques such as autoregression or moving averages could be used to model the temporal dynamics of home values.
- Incorporating uncertainty estimates: The current objective of the competition is to minimize mean absolute error, but it may also be useful to incorporate uncertainty estimates into the model's predictions. This could be achieved by using Bayesian methods or ensembling multiple models to generate probabilistic predictions.
- Deploying the model: Once a high-performing model has been developed, it could be deployed to provide real-time home value estimates for users. This could be integrated into Zillow's existing platform or developed as a standalone web application.

TEAM MEMBERS ROLE:

We have assigned 4 roles to work on this project.

1. **Project Manager - Rakesh Kumar Devagalla**
   This person is responsible for managing the project, assigning tasks to team members, setting deadlines, and ensuring that the project is completed within the given timeline.
2. **Data Scientist - Chaitanya Jayanti**
   This person is responsible for data cleaning, feature selection, model training, and performance evaluation. They are also responsible for tuning the hyperparameters of the models.
3. **Data Analyst - Gurleen Kaur**
   This person is responsible for analyzing the data and providing insights that will be useful in making decisions related to the project. They are also responsible for visualizing the data to aid in better understanding.
4. **Software Engineer - Laasya Priya**
   This person is responsible for developing the software that will be used for the prediction of house prices. They are responsible for integrating the models developed by the data scientist and data analyst into the software and making sure that the software is user-friendly

REFERENCES

- Zillow Prize: The competition offers a prize of $1,200,000 for the team that develops the most accurate model. The competition dataset and more information can be found at https://www.zillow.com/promo/zillow-prize/.
- Advanced Regression Techniques: This is a popular Kaggle competition that involves predicting the sale price of homes based on various features. https://www.kaggle.com/c/house-prices-advanced-regression-techniques.
- Boston Housing Dataset: \ The dataset is available in scikit-learn and more information can be found at https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html.
- New York City Airbnb Open Data: This is a Kaggle dataset that involves predicting the price of Airbnb rentals in New York City based on various features. https://www.kaggle.com/dgomonov/new-york-city-airbnb-open-data.
- California Housing Prices: This is another classic dataset in machine learning that involves predicting the median house value in California based on various features.https://scikit-learn.org/stable/datasets/toy_dataset.html#california-housing-dataset.

CODE :

https://github.com/devagalla/733-project