Home Price Evaluator: Determining Market Value

Final Report of DSC148-WI25

By Diego Arevalo, Diego Osborn, Khoi Phan, and Sardor Sobirov

Introduction

In today's competitive housing market, overpaying for a home is an all-too-common pitfall—one that can lead to missed opportunities and costly investment mistakes. Accurately determining a home's expected price and value could be a crucial aspect in real estate, benefiting both sellers and buyers. This project aims to build a model that predicts home prices and assess their value, helping to identify if a property's price is reasonable. https://house-evaluation.vercel.app/ This live demo will also provide detailed information about an entered property and its surrounding area including estimated price, the property's value, and nearest local amenities.

1 Datasets

1.1 Identify Our Datasets

The **USA Real Estate** dataset, available in three parts (first_half.csv, second_half.csv, and third_half.csv) originates from a Kaggle dataset[1]. This dataset provides detailed information on real estate listings across the United States, categorized by state and zip code. It consists of 2,226,382 entries, with each row representing an individual property listing, and 12 columns containing various attributes of the property, such as the number of bedrooms, bathrooms, and other relevant features.

Other datasets were also used to create context for the market of a property's area (we're basing it on zip code). One dataset used was the **ZIP Code Tabulation Areas** Dataset (stored as national-zcta-data.csv)[2], which contains geographic identifier codes, names, area measurements, and representative latitude and longitude coordinates for zip codes based on the 2020 Census tabulation blocks.

Another one is the **US Population Density by Zip Code, City, and State** Dataset (stored as zip_population.xlsx)[3], which contains 2010 census data for every zip code in the United States, including the corresponding City, State, and latitude/longitude coordinates for the center of the zip code.

Another one used was the **Offenses Known to Law Enforcement** Dataset (stored as fbi-crime.csv)[4], which contains the volume of violent crime and property crime, as reported by 12 months of complete offense data for 2015, as reported by city and town law enforcement agencies that contributed data to the Uniform Crime Reporting Program.

Another dataset used was the **Institutional Characteristics** Dataset (stored as hd2023.csv)[5], which contains detailed information for every institution in the 2023-24 IPEDS universe, including name, address, city, state, zip code, various URL links (e.g., homepage, admissions, financial aid, net price calculator), institutional characteristics (e.g., control, level, highest degree offered, Carnegie classifications), geographical location variables (e.g., county, congressional districts, statistical areas, urbanization degree), and identifies active institutions participating in Title IV federal financial aid

[1] USA Real Estate dataset

[2] ZIP Code Tabulation Areas

[3] US Population Density by Zip Code, City, and State

[4] Offenses Known to Law Enforcement

[5] Institutional Characteristics

programs.

The last dataset used was the **Property Taxes by State** Dataset (stored as propertytax-by-state.csv)[6], which provides state-level property tax information, including effective tax rates for owner-occupied housing in 2022 and 2023, as well as state rankings based on tax rates, offering insights into regional tax burdens.

1.2 Perform EDA

Data Cleaning

Row drop: We first began cleaning by dropping rows with missing 'zip_code' because we felt that it couldn't be reliably imputed and where the 'state' wasn't a US state or DC, e.g., Guam or Puerto Rico. We also made sure every feature was its correct type. We then began the imputing side of things after a good chunk of the original dataset had missing values. We filled in missing values using hierarchical imputation by first using the median zip code values, then median state values, and median global values as a final fallback. Specifically to handle the missing values of columns 'city' and 'house_size', we imputed first using zip code, however for city we then filled in any remaining missing values with state mode values, and dropping the remaining rows, and for 'house_size', we filled in any missing values with median grouped city, state values, and then filled in median state values as a final fallback. We created a binary feature 'first_sale', which was created for homes with missing 'prev_sold_date' values, marking homes that had never been sold before. Temporal data in 'prev_sold_date' was adjusted to account for future dates beyond 2024. Categorical features, such as 'brokered_by' and 'street', were cleaned by filling missing values with 'Not Specified'. Any duplicates were then dropped. After cleaning up the original USA Real Estate dataset, we then moved onto merging the other datasets onto the original dataset. We merged the dataframe 'zipcode_land' (ZIP Code Tabulation Areas Dataset) onto ‘realtor_data‘ to include ‘land area‘ (in sq. miles), ‘water area‘ (in sq. miles), and latitude and longitude for each zip code, merged the dataframe 'zipcode_pop' (US Population Density by Zip Code, City, and State Dataset) onto 'realtor_data' to include 'population' and 'density' for each zip code, and merged the 'crime_data' dataframe (Offenses Known to Law Enforcement Dataset) onto 'realtor_data' to include crime statistics for each ‘city_state‘ (a combinationation of city and state as a temp identifier), and merged the dataframe 'zipcode_pop' onto 'realtor_data' to include 'population' and 'density' for each zip code. We then got rid of any additional columns from the merges that were unnecessary and filtered 'population' to only keep rows properties where the city has a population above 0. We then had to impute missing values again due to the merges, so we did this with the median for each (‘city‘, ‘state‘) group for the ‘land_area‘, ‘water_area‘, ‘latitude‘, and ‘longitude‘ features, and dropped the remaining rows where ‘latitude‘ and ‘longitude‘ were missing, as these corresponded to non-existent areas, such as out of county, louisiana or out of county, california, or remote locations such as Naukati Bay, Alaska, or Kasaan, Alaska. We then merged the ‘uni_data‘ dataset (Institutional Characteristics Dataset) onto ‘realtor_data‘ by grouping by zip code and computing the nearest university once for each unique zip code using the haversine distance formula:

See appendix formula no. 1

and creating new features of characteristics of the nearest university to that zip code. We also created two more new features 'median_zip_price', which represented the median price of homes in the zip code (helped compare listing prices to the market norm), and 'inventory_count', which represented the number of homes available for sale (low inventory → higher prices) by zip code. To deal with the remaining missing values we decided to calculate the nearest zip code for each unique zip code using the haversine distance formula again and creating the features 'nearest_zip' and ‘nearest_zip_distance‘ (sq miles). We then also created new features 'total_crime' that sums up total ‘violent_crime‘

[6] Property Taxes by State

numbers and ‘property_crime‘ numbers for a given zip code to calculate ‘crime_rate‘ which is based on the formula:

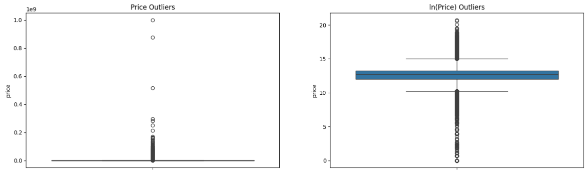
$$\frac{\text{\# of total crimes reported}}{\text{population}} \times 1,000$$

meaning crime rate per 1,000 inhabitants. Our last step was merging the dataframe ‘property_taxes‘ (Property Taxes by State Dataset) to include property taxes for each state. We also created a new feature ’house_category’, which represents the type of home a property is e.g., tiny home, investment property, family home, ranch home, luxury home, luxury estate, and other. This was done through geodemographic segmentation, which selected relevant features, enabling nuanced, region-specific property classification. The derivation came from zip code-level aggregations to classify homes based on local real estate characteristics, e.g., price, house size, crime rates, and property taxes, acreage, etc. Properties labeled Luxury Estates typically exceeded local norms in price, size, and acreage. Properties were labeled Luxury Homes if they met the high-price and large-size criteria, but not necessarily acreage. Properties labeled Ranch Homes depended on having substantial land size relative to local norms. Other categories included Family Homes, which were defined by mid-range prices, adequate bedrooms, and bathrooms; Tiny Homes were identified by having small size and being affordable; and Investment Properties were characterized by high crime, elevated property taxes, or unusually small sizes.

Basic Statistics

After looking at the summary statistics of the numeric features, there were some extreme outliers that had to be dealt with. We saw that max ‘price‘ is 1×10^9, which is $1 billion, however, this is not realistic because, as of February 2024, the most expensive home in the United States is Gordon Pointe in Naples, Florida, which is listed for $295 million dollars[2][7].

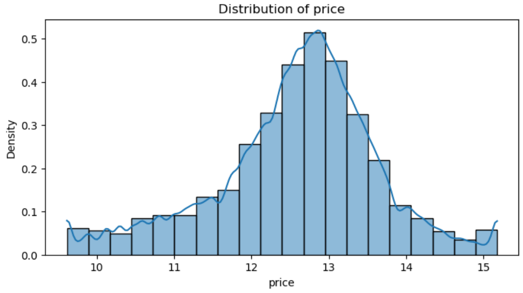
Figure 1: Box plots of Price (pre- and -post log transform)



Therefore we filtered ‘realtor_data‘ to not include the outliers in ’price’ that are greater than $295 million dollars.[2] We can also see that the min is 0.0, meaning the home was free, however, this is not realistic and to avoid data entry errors, the lowest threshold was set at $10,000. We did this same process for many other numeric features that had extreme outliers, e.g., ’bed’, ’bath’, and ’house_size’. For the remaining columns that had relatively realistic outliers, we winsorized them to limit any extreme values by capping them at certain thresholds (we did at a 0.01% threshold level). After attempting to handle the extreme outliers, we ended up logarizing all of the numeric features that had a skew score above 1 and that weren’t ’bed’, ’bath’, ’property_tax’, ’crime_rate’, ’nearest_zip_distance’, and ’total_crime’.

Findings in EDA

1) **Figure 2: Histogram of (log) Price**

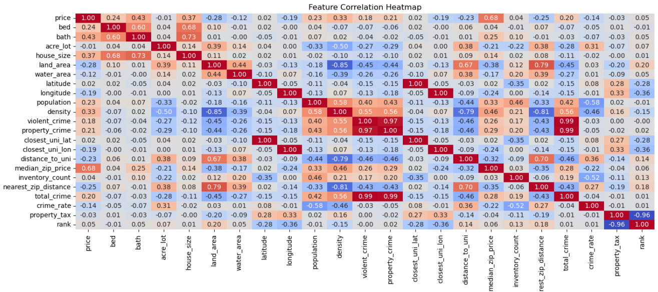


After logarizing our target feature ’price’, we can see that the distribution has gotten closer to resembling a normal curve compared to the box

[7] Most expensive home in the US is on the market

plot from figure 1, where we can see that the box plot of price pre-transformation is not normal.

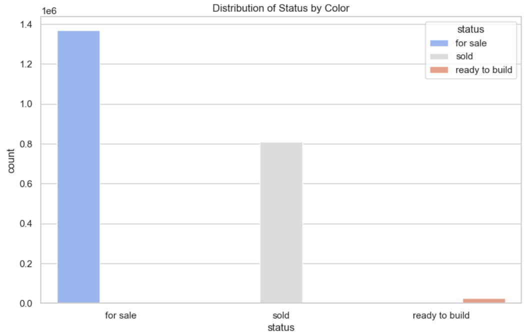
2) **Figure 2: Correlation Heatmap**


Feature Correlation Heatmap

	price	bed	bath	acre_lot	house_size	land_area	water_area	latitude	longitude	population	density	violent_crime	property_crime	closest_uni_lat	closest_uni_lon	distance_to_uni	median_zip_price	inventory_count	nearest_zip_distance	total_crime	crime_rate	property_tax	rank
price	1.00	0.24	0.43	-0.01	0.37	-0.28	-0.12	0.02	-0.19	0.23	0.33	0.18	0.21	0.02	-0.19	-0.23	0.68	0.04	-0.25	0.20	-0.14	-0.03	0.05
bed	0.24	1.00	0.60	0.04	0.68	0.10	-0.01	0.02	-0.00	0.04	-0.07	-0.07	-0.06	0.02	-0.00	0.06	0.04	-0.01	0.07	-0.07	-0.05	0.01	-0.01
bath	0.43	0.60	1.00	0.04	0.73	0.01	-0.00	-0.05	-0.01	0.07	0.02	-0.04	-0.02	-0.05	-0.01	0.01	0.25	0.10	-0.01	-0.03	-0.07	-0.03	0.05
acre_lot	-0.01	0.04	0.04	1.00	0.14	0.39	0.14	0.04	0.00	-0.33	-0.50	-0.27	-0.29	0.04	0.00	0.38	-0.21	-0.22	0.38	-0.28	0.31	-0.07	0.07
house_size	0.37	0.68	0.73	0.14	1.00	0.11	0.02	0.02	0.01	-0.02	-0.10	-0.12	-0.10	0.02	0.01	0.09	0.14	0.02	0.08	-0.11	-0.02	-0.00	0.01
land_area	-0.28	0.10	0.01	0.39	0.11	1.00	0.44	-0.03	-0.13	-0.18	-0.85	-0.45	-0.44	-0.03	-0.13	0.67	-0.38	0.12	0.79	-0.45	0.03	-0.20	0.20
water_area	-0.12	-0.01	-0.00	0.14	0.02	0.44	1.00	-0.10	0.07	-0.16	-0.39	-0.26	-0.26	-0.10	0.07	0.38	-0.17	0.20	0.39	-0.27	0.01	-0.09	0.05
latitude	0.02	0.02	-0.05	0.04	0.02	-0.03	-0.10	1.00	-0.05	-0.11	-0.04	-0.15	-0.15	1.00	-0.05	-0.03	0.02	-0.35	0.02	-0.15	0.08	0.28	-0.28
longitude	-0.19	-0.00	-0.01	0.00	0.01	-0.13	0.07	-0.05	1.00	-0.13	0.07	-0.13	-0.18	-0.05	1.00	-0.09	-0.24	0.00	-0.14	-0.15	-0.01	0.33	-0.36
population	0.23	0.04	0.07	-0.33	-0.02	-0.18	-0.16	-0.11	-0.13	1.00	0.58	0.40	0.43	-0.11	-0.13	-0.44	0.33	0.46	-0.33	0.42	-0.58	0.02	-0.01
density	0.33	-0.07	0.02	-0.50	-0.10	-0.85	-0.39	-0.04	0.07	0.58	1.00	0.55	0.56	-0.04	0.07	-0.79	0.46	0.21	-0.81	0.56	-0.46	0.16	-0.15
violent_crime	0.18	-0.07	-0.04	-0.27	-0.12	-0.45	-0.26	-0.15	-0.13	0.40	0.55	1.00	0.97	-0.15	-0.13	-0.46	0.26	0.17	-0.43	0.99	-0.03	0.00	-0.00
property_crime	0.21	-0.06	-0.02	-0.29	-0.10	-0.44	-0.26	-0.15	-0.18	0.43	0.56	0.97	1.00	-0.15	-0.18	-0.46	0.29	0.20	-0.43	0.99	-0.05	-0.02	0.02
closest_uni_lat	0.02	0.02	-0.05	0.04	0.02	-0.03	-0.10	1.00	-0.05	-0.11	-0.04	-0.15	-0.15	1.00	-0.05	-0.03	0.02	-0.35	0.02	-0.15	0.08	0.27	-0.28
closest_uni_lon	-0.19	-0.00	-0.01	0.00	0.01	-0.13	0.07	-0.05	1.00	-0.13	0.07	-0.13	-0.18	-0.05	1.00	-0.09	-0.24	0.00	-0.14	-0.15	-0.01	0.33	-0.36
distance_to_uni	-0.23	0.06	0.01	0.38	0.09	0.67	0.38	-0.03	-0.09	-0.44	-0.79	-0.46	-0.46	-0.03	-0.09	1.00	-0.32	-0.09	0.70	-0.46	0.36	-0.14	0.14
median_zip_price	0.68	0.04	0.25	-0.21	0.14	-0.38	-0.17	0.02	-0.24	0.33	0.46	0.26	0.29	0.02	-0.24	-0.32	1.00	0.03	-0.35	0.28	-0.22	-0.04	0.06
inventory_count	0.04	-0.01	0.10	-0.22	0.02	0.12	0.20	-0.35	0.00	0.46	0.21	0.17	0.20	-0.35	0.00	-0.09	0.03	1.00	-0.06	0.19	-0.52	-0.11	0.13
nearest_zip_distance	-0.25	0.07	-0.01	0.38	0.08	0.79	0.39	0.02	-0.14	-0.33	-0.81	-0.43	-0.43	0.02	-0.14	0.70	-0.35	-0.06	1.00	-0.43	0.27	-0.19	0.18
total_crime	0.20	-0.07	-0.03	-0.28	-0.11	-0.45	-0.27	-0.15	-0.15	0.42	0.56	0.99	0.99	-0.15	-0.15	-0.46	0.28	0.19	-0.43	1.00	-0.04	-0.01	0.01
crime_rate	-0.14	-0.05	-0.07	0.31	-0.02	0.03	0.01	0.08	-0.01	-0.58	-0.46	-0.03	-0.05	0.08	-0.01	0.36	-0.22	-0.52	0.27	-0.04	1.00	-0.01	0.01
property_tax	-0.03	0.01	-0.03	-0.07	-0.00	-0.20	-0.09	0.28	0.33	0.02	0.16	0.00	-0.02	0.27	0.33	-0.14	-0.04	-0.11	-0.19	-0.01	-0.01	1.00	-0.96
rank	0.05	-0.01	0.05	0.07	0.01	0.20	0.05	-0.28	-0.36	-0.01	-0.15	-0.00	0.02	-0.28	-0.36	0.14	0.06	0.13	0.18	0.01	0.01	-0.96	1.00

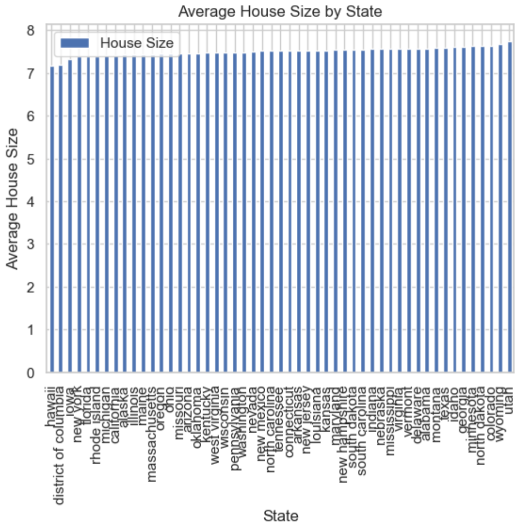
From the correlation heatmap, we can see that a majority of the numeric features are not very correlated, however, there are a few hotspots that are highly correlated such as 'population', 'density', 'violent_crime', and 'property_crime'.

3) **Figure 3: Distribution of Status**

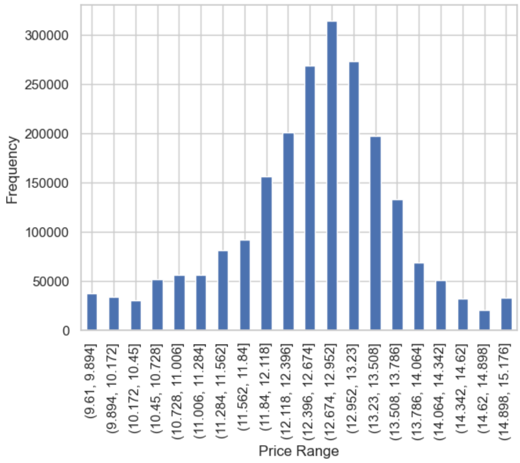


From the count plot, we can see that there are very few properties that are of the status 'ready to build', meaning these homes are to be constructed, and that around $\frac{2}{3}$ of the homes are for sale.

4) **Figure 4: Average House Size by State**
From this bar chart, we can see that Utah, Wyoming, and Colorado all have the highest average house size, meaning that there most likely is some correlation with the location of the state and average house size, as these 3 states are all in the mid-west.

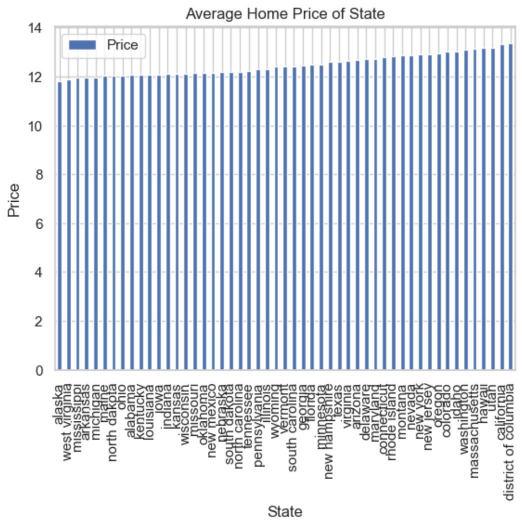


5) **Figure 5: Price Range Frequency**



From this bar chart, we can see that a majority of the home prices fall within the (12.674, 12.952] bin. Since these are log-transformed prices, this would equate to a bin of around $319336.28 to $421679.15.

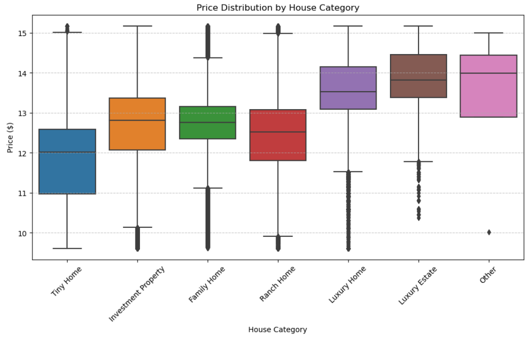
6) **Figure 6: Average Price of State**



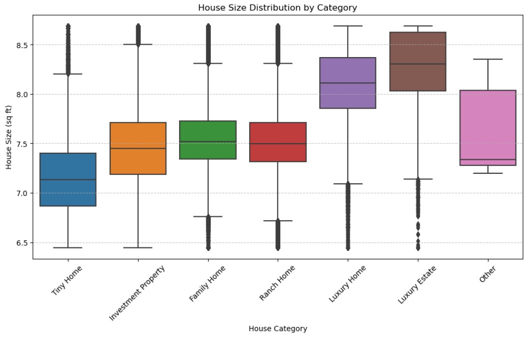
From this bar chart, we can see that the states with the highest average price are DC, California, Utah, and Hawaii. This makes sense since DC is a city, and prices of an urban dense population are typically higher. California home prices are notoriously higher because of hot spots such as San Francisco, Los Angeles, and San Diego. Utah has a high average home price because of recent high demand and limited housing supply. Hawaii having a high average home price makes sense due to scenery and high demand since that state is a 'beautiful' tropical island.

7) **Figure 7: Price Distribution by House Category**

It is expected that Luxury Estates command the highest prices, as they include expansive properties in prime locations. Ranch Homes appear to have a broad price distribution, which could be due to differences in land value across regions. Investment Properties show high variation, suggesting the category includes both affordable rental units and high-value investment assets. The Other category remains inconsistent, indicating that it might contain unclassified properties such as mobile homes or mixed-use spaces.



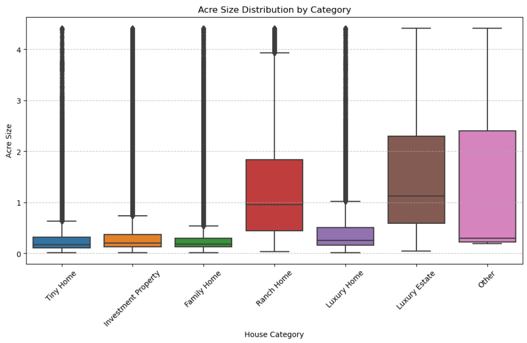
8) **Figure 8: House Size Distribution by Category**



Family Homes having larger median sizes than Tiny Homes aligns with expectations, as they are designed for larger households. The wide range in Ranch Home sizes suggests that some may be large, multi-acre properties, while others could be smaller homes with significant land. Investment Properties showing variation in size suggests a mix of rental units, duplexes, and larger income-generating properties. The relatively small size distribution of Tiny Homes reflects a niche market targeting affordability and compact living.

9) **Figure 9: Acre Size Distribution by Category**

Ranch Homes having large acre sizes but mid-tier prices aligns with expectations, as these homes prioritize land over house size. Luxury Estates,



which have both large acre sizes and high prices, appear correctly classified. The presence of large-acre properties in the Other category may indicate incomplete classifications or edge cases. The relatively small land size of Tiny Homes and Investment Properties is expected, as these properties are typically located in dense urban areas or are compact rental units.

2 Predictive Task

2.1 Identify a Predictive Task

Our objective is to **predict house prices using a comprehensive array of features**, including factors such as zip code, house size, and the number of bathrooms, among others. To achieve this, we aim to develop a regression model trained on an extensive dataset comprising millions of observations from across the United States. In constructing our final predictive model, we will evaluate its performance against six baseline models, iteratively improving upon them to create a more accurate and robust regression model.

2.2 Evaluation

The primary metric used to compare our baseline models and the final model is the **Root Mean Squared Error (RMSE)**. RMSE is a commonly used evaluation metric for regression problems, providing a clear indication of model performance. It calculates the square root of the average squared differences between predicted and actual values, making it highly sensitive to large errors. This sensitivity is particularly beneficial when we want to penalize significant prediction errors more heavily.

2.3 Preprocessing Features

For our baseline models, we decided to start with the seven numerical features most correlated with price: **median_zip_price, house_size, acre_lot, bath, latitude, longitude, and crime_rate**. We used these exact features for all baseline models to standardize the comparison and make it easier to observe future improvements in terms of adding or subtracting features in the creation of our final model. We also split our dataset into training and test sets in order to properly evaluate our baseline models on unseen data.

2.4 Baseline Models

The first baseline model is a basic **Linear Regression**. The main purpose of this model is to analyze and quantify the linear effect of each of the seven features most correlated with price. While a simple linear regression model like this one is not going to be the best performing model, it is an important setting stone for more advanced models with more features. Additionally, it is the model least likely to overfit to the training set.

OLS linear regression can help us determine the importance of our features through their weights, but sometimes these weights can be detrimental when it comes to predicting unseen data and lead to higher variance. To prevent overfitting, we can introduce penalties in the regression and reduce the importance of the weights by shrinking them towards 0. Two techniques to achieve this are **Ridge and Lasso Regression**. Therefore, we also created two more regression models as a baseline point of reference.

One of the key disadvantages of linear regression is its failure to capture nonlinear relationships, thus producing biased estimates. Because of this, we created a **Random Forest Regression** model. A random forest model can effectively capture valuable trends and relationships through the use of ensemble learning methods, which train several models through a series of decision trees on random samples of the training set and return an averaged result in which the trees "vote" for the best model. One disadvantage is that it can lead to overfitting, but we limited max_depth to counteract it. Random forest regression can provide additional input on what features accurately predict price non-linearly.

For tabular data, **XGB Regression** and **LightGBM** are known to be some of the best-performing models. Like with random forest regression, both of these algorithms use ensemble learning methods, with the addition of gradient boosting frameworks that iteratively improve predictions made by individual decision trees (or "weak learners"), with the additional techniques like regularization. In a way, both of these models combine previous simpler models into a faster and more accurate single model. XGB Regression tends to perform better overall, but lightGBM is more memory efficient and tends to be faster for large datasets like this one.

2.5 Baseline Results

The ordinary linear regression model exhibited the poorest performance, with an RMSE of 0.71. Both the ridge and lasso regression models also recorded an RMSE of 0.71, indicating that any potential improvements over the ordinary linear regression model were minimal. The random forest regression model performed slightly better, achieving an RMSE of 0.64. As anticipated, the most proficient models were XGB Regression, which delivered the lowest RMSE of 0.57, and LightGBM, with an RMSE of 0.62, underscoring their superior predictive accuracy compared to the other models tested.

3 Model

3.1 Price Prediction Model

The final model selected for the predictive task is **Light Gradient Boosting Machine (LightGBM)**. Since our final dataset has over two million entries, LightGBM was chosen due to its superior balance between predictive accuracy and computational efficiency, making it well-suited for handling such large dataset with numerous features. Another reason we chose LightGBM is because of its ability to handle categorical variables effectively and mitigate overfitting through built-in regularization techniques, making it a robust choice for real estate price prediction.

3.2 Features Preprocessing

Instead of just using the seven most correlated numerical features like our baseline models, we decided to use all the available features as well engineered ones. Thus, we need to preprocess our features before training the model.

Date Feature Extraction

The prev_sold_date column, which is a datetime feature, is decomposed into three separate numerical features:

- prev_sold_year (year of the previous sale)
- prev_sold_month (month of the previous sale)
- prev_sold_day (day of the previous sale)

The original prev_sold_date column is then removed, as it has been replaced by more useful numerical representations.

Encoding Categorical Variables

In order to make categorial features useful for the model, we used LabelEncoder from sklearn.preprocessing to transform categorical values into numerical values.

For example, if a column contains ['Apartment', 'House', 'Condo'], LabelEncoder will convert it to [0, 1, 2]

3.3 Model Optimization

In order to make sure that our final model performs better than all the baseline models while avoiding overfitting, we optimized it by hyperparameter tuning and splitting the data into a test and training set.

Hyperparameter Tuning

Accounting for efficiency, given that we have a large dataset, we deicded to used RandomSearchCV from sklearn.model_selection over GridSearchCV in order to find the best combination of hyperparameters for our model.

The parameters that we used for the model are:

- **objective**: regression
- **metric**: rmse
- **boosting_type**: gbdt
- **num_leaves**: 100
- **max_depth**: 20
- **learning_rate**: 0.1
- **lambda_l1**: 1.0
- **lambda_l2**: 1.0
- **feature_fraction**: 1
- **bagging_fraction**: 0.8

Since we are predicting home prices (a continuous variable), we need a regression model and we are evaluating the model using RMSE as discussed above.

We went with GBDT (Gradient Boosted Decision Trees) as the boosting type because it works well for structured/tabular data, learning from previous mistakes by iteratively reducing errors. Additionally, GBDT generally offers the best performance for structured regression tasks.

To controls the complexity of the model by determining how many decision leaves a single tree can have, we tuned num_leaves to make sure that we don't risk overfitting the model by capturing too many patterns. Additionally, we also tuned max_depth to limit how deep each tree can grow, preventing excessive complexity.

In order to make sure that our model can generalize data well, we tuned learning_rate to control how much the model adjusts weights in each iteration.

Furthermore, to encourage sparsity in feature importance, control large weights, reducing variance and improving generalization, we tuned L1 Regularization and L2 Regularization.

Lastly, we also tuned feature_fraction and bagging_fraction to prevent overfitting by ensuring the model doesn't become too dependent on a specific subset of training data and making sure that the model use all available features as feature selection was already optimized during preprocessing.

3.4 Mapping Valuation

We took the difference between the expected price from our optimized LightGBM model and the listed price, and based our valuations on this difference. If the difference was within one standard deviation of the overall price difference, we classified the property as **properly valued**. If the difference was less than negative one standard deviation, we classified the property as **undervalued**. Conversely, if the difference exceeded one standard deviation, we classified the property as **overvalued**. Based on our valuations, nearly 79% of homes were classified as "Properly Valued." However, approximately 10.5% were deemed "Undervalued," potentially indicating good investment opportunities. This distribution suggests that while most homes are priced fairly, a significant portion still deviates from their expected value, emphasizing the importance of accurate valuation in real estate decision-making.

4 Literature

Implementing GIS in Real Estate Price Prediction and Mass Valuation: The Case Study of Nicosia District. By Charambolos Yiorkas

and Thomas Dimopoulos
https://www.researchgate.net/publication
Yiorkas and Dimopoulos wanted to produce better real estate predictions for a small dataset of 1341 properties in Cyprus. The main purpose of this report is to prove that geospatial features such such as proximity to schools or universities have a significant effect on property prices. Price predictions for properties in Cyprus were usually done by only looking at the physical properties of the house, such as its size, age, and number of bedrooms and bathrooms. The two main models that were performed in this research are Ordinary Least Squares Regression (as a baseline model) and Geographically Weighted Regression (as the final model). While we didn't implement geospatial analytical techniques in our models unlike in this report, this served as inspiration for us to include features that were outside of our original dataset and related with the geography and environment of the property. The main evaluation metric used was R-squared instead of RMSE, and the Geographically Weighted Regression model proved to be far more accurate, with an R-squared value of 0.79 versus 0.55 for the OLS model, thus proving that in some instances, geographical features can be a very useful addition to a property price prediction model.

Advanced Machine Learning Algorithms for House Price Prediction: Case Study in Kuala Lumpur. By Shuzlina Abdul Rahman, Sofanita Mutalib, Ismail Ibrahim, Nor Hamizah Zulkifley
https://www.researchgate.net/publication
This project is the most similar to ours we could find in terms of data preprocessing, feature selection, and the models that were implemented. It is based on a dataset of 53883 entries representing individual properties in Kuala Lumpur, and the four models that were implemented were Multiple Linear Regression, Ridge Regression, LightGBM, and XGBoost, which turned out to be the best performing model. The main difference between our project and this one is the missing value imputations. Rahman, Mutalib, Ibrahim, and Zulkifley opted to simply drop all rows with any missing values, which significantly reduced the size of the dataset to 31899 entries, while we followed a multistep implementation process that would attempt to impute missing values based on local group medians (such as zip code), or global medians if not possible. Price was also log transformed due to extreme skewness, showcasing the many similarities between our dataset and this one, even while taking into account the substantial geographical differences (Kuala Lumpur, a city, vs the United States, a large country). Overall, this project shares many commonalities with our regression component of the project, but it doesn't go into the value analysis (whether the house is undervalued, overvalued, or neither).

5 Results

5.1 Compare the Models

Our analysis compared several predictive models for house price estimation. The baseline linear regression model (OLS) produced an RMSE of 0.71, indicating moderate performance but limited capability in capturing non-linear relationships. Regularized models like Ridge and Lasso offered improvements in feature selection and model simplicity, though their RMSE values remained close to 0.71, suggesting that simply penalizing coefficients did not enhance prediction accuracy. In contrast, ensemble methods performed significantly better a Random Forest model achieved an RMSE of approximately 0.64 by better handling feature interactions and non-linearities. Most notably, gradient boosting techniques —XGBoost attained the lowest RMSE at around 0.57, while LightGBM delivered an RMSE near 0.62, coupled with improved computational efficiency on large datasets. Taking that into account we took LightGBM as the final model and tunned it with a final RMSE of approximately 0.4.

5.2 Major Takeaways

- Incorporating geographic and environmental data improved prediction accuracy.
- Our baseline linear regression models (OLS, Ridge, Lasso) struggled to capture the complexities of real estate pricing, compared to the ensemble models (Random Forest and gradient boosting methods) that significantly improved predictive accuracy.
- The LightGBM model was best-performing model, achieving an RMSE of 0.4 after hyperparameter tuning. Its computational efficiency and ability to handle non-linearity made it the optimal choice. The only issue with this being our best model is that LightGBM is black-box, so we don't really know what's going on inside the model, which makes it harder to interpret.
- Unlike previous studies that dropped missing values, our imputation strategy (using local medians when possible) allowed us to retain more data while maintaining model robustness.

References

Datasets:

- https://www.kaggle.com/datasets/ahmedshahriarsakib/usa-real-estate-dataset/data
- https://www.census.gov/geographies/reference-files/time-series/geo/gazetteer-files.html
- https://www.census.gov/programs-surveys/geography/technical-documentation/records-layout/gaz-record-layouts.html
- https://www.fourfront.us/data/datasets/us-population-density/
- https://ucr.fbi.gov/crime-in-the-u.s/2015/crime-in-the-u.s.-2015/tables/table-8/table_8_offenses_known_to_law_enforcement_by_state_by_city_2015.xls/view
- https://ucr.fbi.gov/crime-in-the-u.s/2015/crime-in-the-u.s.-2015/tables/table-8/table-8-data-declaration_final
- https://nces.ed.gov/ipeds/datacenter/DataFiles.aspx?year=20
8b20-4dc7-9338-4b14f9d94e9c&rtid=7
- https://taxfoundation.org/data/all/state/property-taxes-by-state-county/

Citations:

- https://researchmaniacs.com/ZipCodes/9/Where-is-Zip-Code-96999.html
- https://oag.ca.gov/sites/all/files/agweb/pdfs/cjsc/stats/compu
- https://www.newscentermaine.com/article/news/nation-world/most-expensive-home-in-the-us-on-the-market/507-ab381921-8d2f-4236-86c0-6f785f73dbd9
- https://www.housebeautiful.com/lifestyle/g46520154/biggest-houses-across-the-world/
- https://www.silive.com/news/2022/01/americas-most-expensive-home-21-bedrooms-49-bathrooms-twice-as-big-as-the-white-house-295m.html
- https://www.familyhandyman.com/list/the-biggest-home-in-each-state-that-will-stun-you/?srsltid=AfmBOoqAs9h6YdZAM2KcBt9QwmYL4bxOLH
PKcqTQ0OEFf-oE7
- https://www.researchgate.net/publication/...
- https://thesai.org/Downloads/Volume12No12/Paper_91-Advanced_Machine_Learning_Algorithms.pdf

Appendix

1.

$$d = 2r \arcsin\left(\sqrt{\sin^2(\frac{\phi_2 - \phi_1}{2}) + \cos(\phi_1)\cos(\phi_2)\sin^2(\frac{\lambda_2 - \lambda_1}{2})}\right)$$