Predicting California Housing Prices Using Ensemble Models

Xin Feng  
 University of Colorado Boulder  
 [xife6244@colorado.com](mailto:email@email.com)

ABSTRACT

This project delves into the domain of machine learning, with the primary objective of predicting housing prices in California. We begin by exploring the dataset, gaining insights into its background, existing research outcomes, and relevant information. We then perform data preprocessing, categorizing and transforming the data to simplify the subsequent analysis process. Next, we employ various Ensemble models to analyze the data, resulting in improved data accuracy compared to existing benchmarks. Finally, we compare the characteristics of these models and define the direction for future improvements.

Throughout the project, we've observed that while traditional machine learning methods can potentially outperform AutoML in terms of accuracy, they come at the cost of extensive time spent on preprocessing. This efficiency trade-off highlights the potential for AutoML to replace traditional machine learning methods in the future as it continues to evolve and simplify the data analysis process.

**1** Project Introduction

In this project, we will utilize Ensemble Models to predict California Housing Prices. Firstly, we will analyze the project's background. Then we will introduce existing research outcomes, and finally, we will outline our overall timeline and schedule.

**1.1** Problem Statement

This project originates from a Kaggle competition in 2021 hosted by D2L, an online platform for deep learning courses. The project's theme revolves around utilizing machine learning techniques to predict housing prices in California. The evaluation metric is the **Root Mean Square Error (RMSE)** between the logarithm of predictions and the logarithm of true values.

Through this project, participants have the opportunity to enhance their analytical skills while establishing connections between housing prices and various property-related information. Furthermore, this endeavor lays the theoretical groundwork for future services and solutions related to the real estate market.

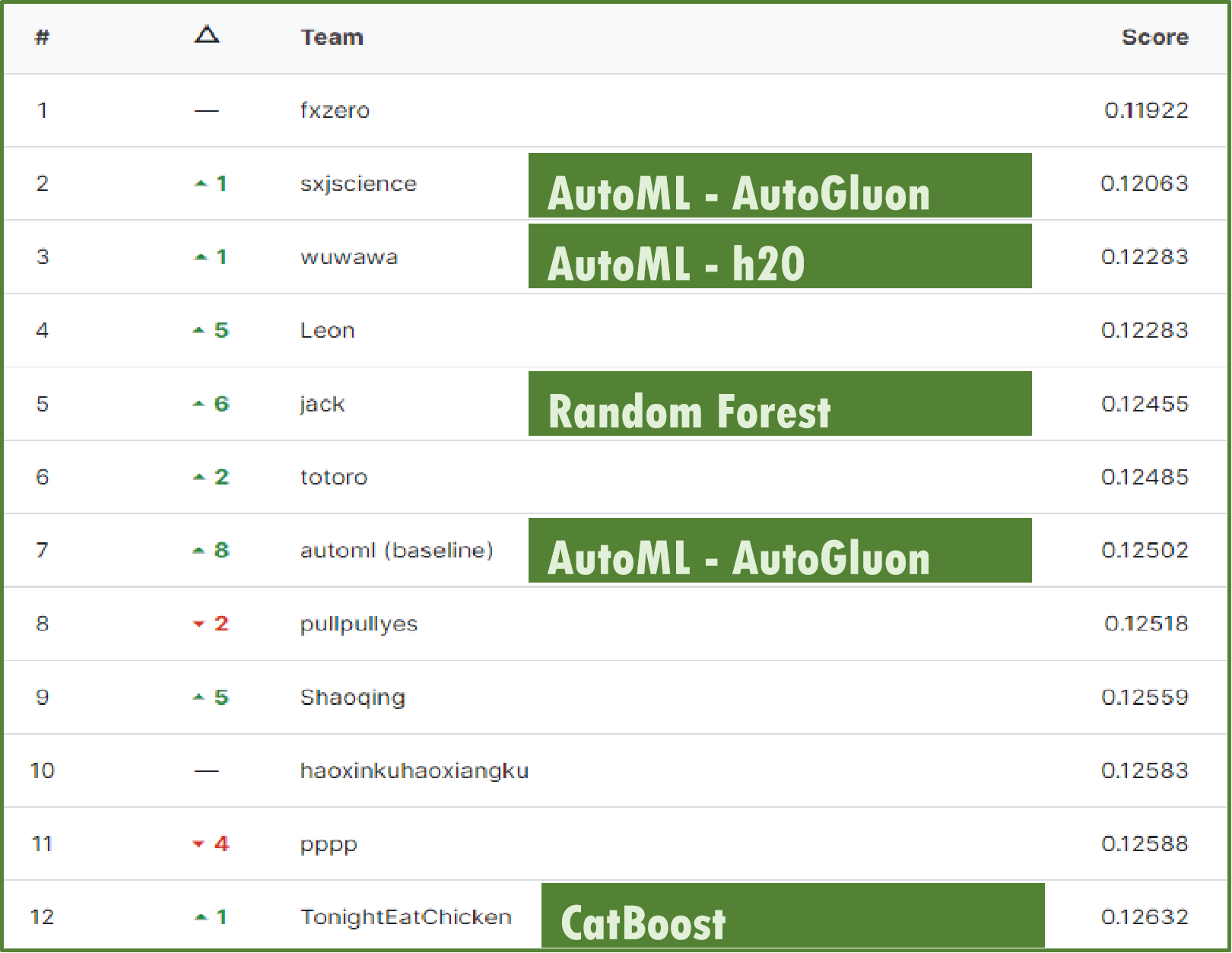
The data for this project is sourced from real-world housing transaction records. It was obtained by web scraping from the Zillow.com website, consisting of 47,439 housing sold in year 2020 as the training dataset and 31,626 housing sold in year 2021 as the test dataset.

The overall dataset comprises 40 features related to the properties. Due to the scraping method, website configurations, user interactions, and other factors, many of these features contain missing values, outliers, and noise, posing certain challenges in predicting housing prices.

The objective of this project is to train data using ensemble models to find the optimal solution and achieve a high ranking in the competition.

**1.2** Related Works

In this competition, a total of 173 teams participated. Among them, 38 teams achieved results within a 15% RMSE (Root Mean Square Error), 29 teams achieved results within a 14% RMSE, 17 teams achieved results within a 13% RMSE, and only 1 team achieved results within a 12% RMSE.



**Figure 1: Competition Leaderboard on Kaggle.com**

The competition host utilized the **AutoGluon automated machine learning model**, achieving approximately a 12.5% RMSE with just ten lines of code. This performance level serves as the target for every participant to challenge. Leading participants in the rankings predominantly employed ensemble learning and automated machine learning techniques.

**1.3** Project Timeline

In the first week, we will establish the project structure, select appropriate models, define project objectives, and draft the proposal.

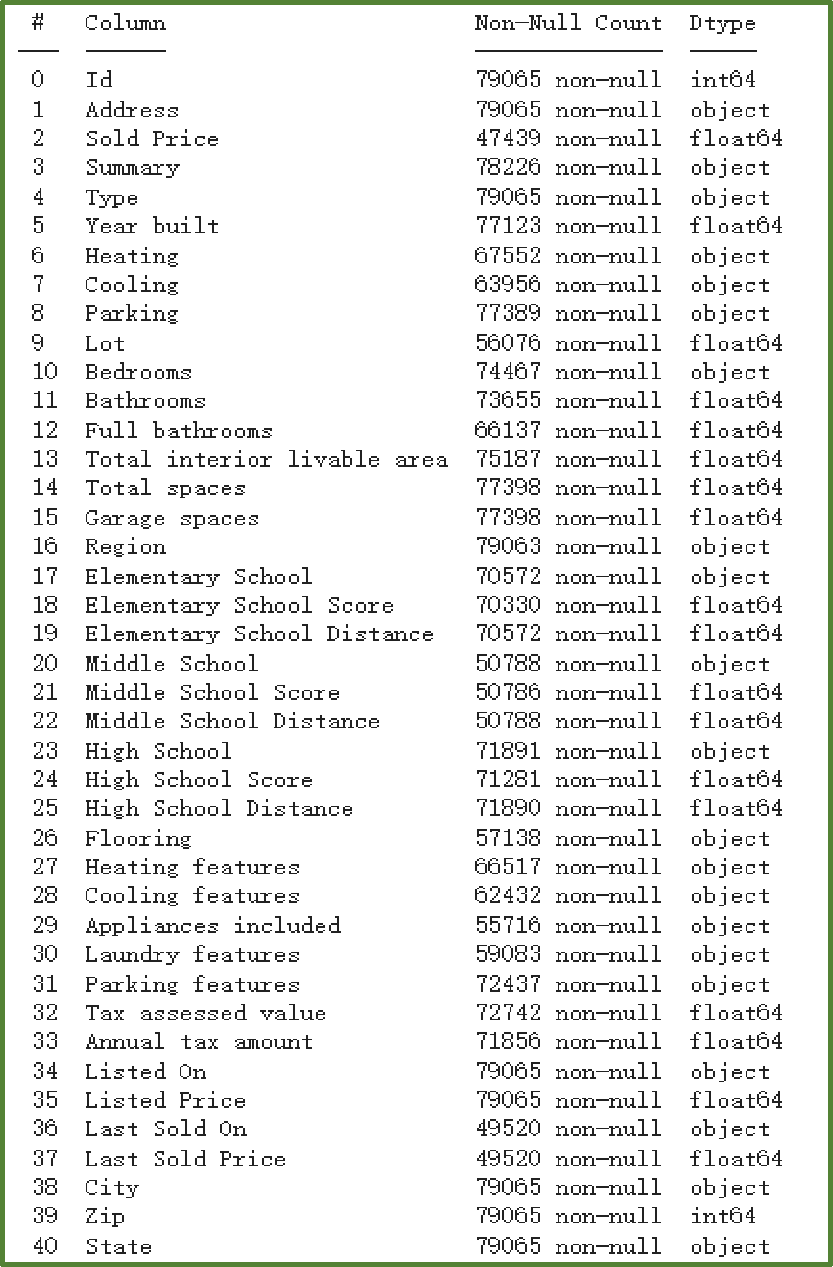
In the second week, we will monitor the project's progress, assess the models used, write the project report, and complete the checkpoint.

In the final week, we will summarize the project's accomplishments, analyze areas for improvement, and finalize the report writing and presentation.

2 Data Preprocessing

The dataset comprises information on a total of 79,065 houses. Among these, 47,439 houses are included in the training set, which includes the Sold Price, while 31,626 houses are designated as the test set, for which we need to predict the Sold Price.

The dataset comprises 40 features, encompassing numeric, categorical, temporal, and geographical attributes. These features collectively provide comprehensive information about the houses in the dataset.

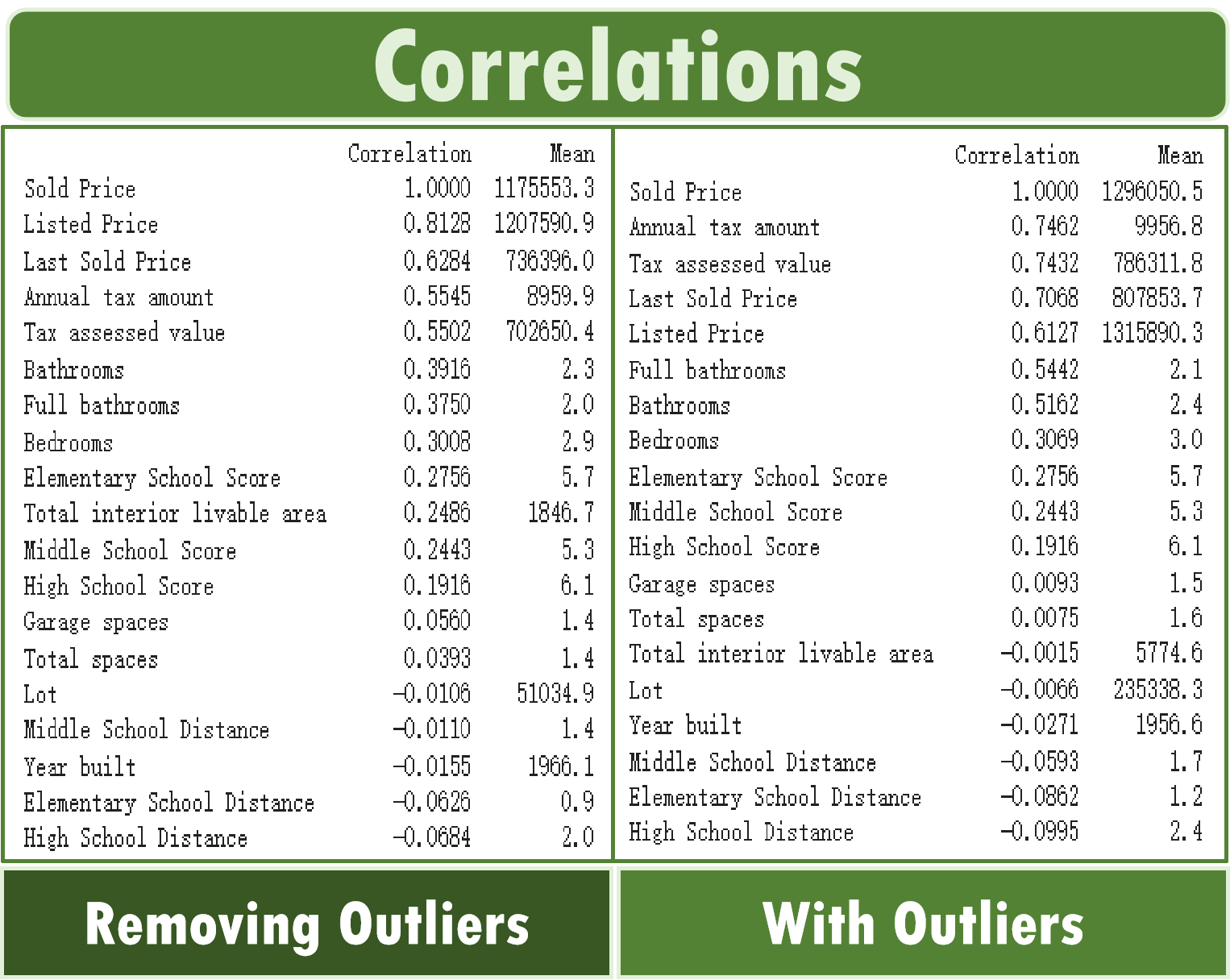


**Figure 2: Features from the Train and Test Sets**

**2.1** Numerical Features

The numeric features include information related to house attributes such as Listed Price, Last Sold Price, Tax, and structural characteristics like the number of bedrooms and bathrooms, as well as interior and total space, among others.

Regarding the numeric features, we initially calculate their correlation with Sold Price and compare the results between including Extreme Values and excluding Extreme Values.



**Figure 3: Correlation between Features and Sold Price**

Our findings lead to the following conclusions:

Price-related features exhibit strong correlations with Sold Price, with Listed Price's correlation coefficient exceeding 0.8, making it a key variable for predicting Sold Price.

Several variables show correlations exceeding 20%, indicating their supportive role in prediction.

Extreme values have a significant impact on correlation coefficients, suggesting the presence of data errors that require careful handling. Balancing variance and bias is crucial in addressing this issue.

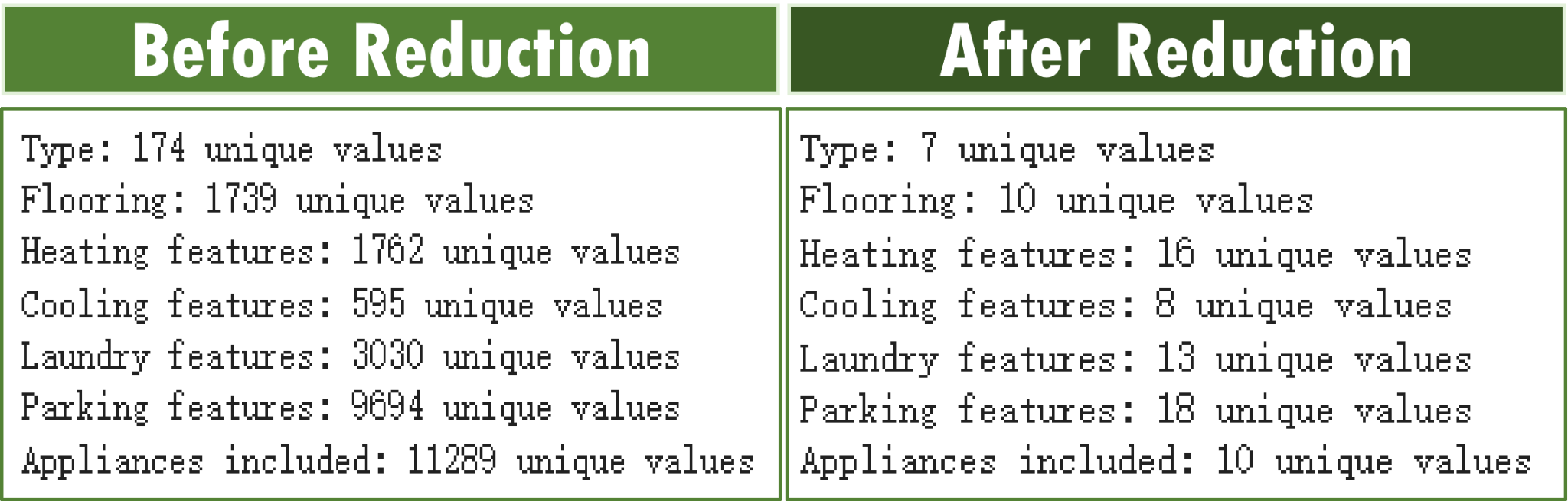
Attention must be given to the handling of missing values in the dataset, as different approaches can have a significant impact on the results.

**2.2** Category Features

The dataset comprises various categorical features related to house characteristics, such as house type, heating, cooling, parking, flooring, laundry, and more, all of which contribute to the overall design and structure of the houses.

However, seven of the major categorical variables present a unique challenge due to an exceptionally high number of distinct values. This challenge stems from a combination of factors, including gaps in text data processing during web scraping, the impact of website structure on text ordering, user input errors like typos or sequencing issues, among others. Consequently, several categories encompass hundreds or even thousands of distinct values.

In response to this challenge, we have adopted an approach focused on consolidating similar observations. Drawing upon our domain knowledge, we consider both the training and testing datasets when merging categories, while closely monitoring the price variance of the combined results. Our objective is to merge as many similar variables as possible while minimizing information loss.



**Figure 4: The Unique Value by Each Categorical Feature**

Through the consolidation process, the number of variables within each category has been reduced to fewer than 20, facilitating subsequent modeling and analysis.

**2.3** Data Transformation

The final steps in data preprocessing involve handling different types of data:

For temporal data, we aggregated it by year level.

For geographical data, we retained the first 3 digits of the Zip, representing geographic information for over 1000 cities with approximately 50 variables.

Categorical variables were uniformly processed using one-hot encoding.

For some numeric variables with relatively large values, we use log transformation to adjust the value range and achieve a distribution closer to normal.

Extreme values were replaced as missing values.

For missing values, distinct handling methods were employed based on the specific requirements of different models.

3 Model Evaluation

In this section, we will embark on modeling and analysis using the data we have already preprocessed.

Firstly, we will employ the simplest prediction approach by using Listed Price as our predictive data, allowing us to understand the worst-case scenario for our model.

Next, we will turn to one of the most widely applicable models in various domains, Random Forest, for prediction, and fine-tune its hyperparameters to achieve optimal results.

Subsequently, we will explore the use of the popular CatBoost algorithm to assess whether further improvements in performance can be achieved.

Lastly, we will endeavor to combine the insights from both models to evaluate the potential for enhanced predictive capabilities.

**3.1** Guessing by Listed Price

From the earlier correlation analysis, we gleaned that when excluding outliers, Listed Price exhibits a correlation exceeding 80% with Sold Price in the training data. Consequently, we have reason to believe that utilizing Listed Price directly for prediction may yield reasonable results. To further optimize this approach, we replaced all minimum values with 150,000, reducing the impact of extreme low values on the model. After conducting the analysis, we obtained the following results:

Train RMSE：0.11511

Validation RMSE：0.12323

**Test RMSE：0.14606**

**Test Ranking：#38 / #173**

This is indeed a remarkable finding that merely using a single feature can result in predictions surpassing those of about 80% of the teams. This underscores the fact that if features and models are not employed judiciously, the outcomes may be unexpectedly subpar.

Furthermore, it's worth noting that the Test RMSE is lower than both the Train and Validation RMSE. This discrepancy can likely be attributed to the influence of the timing of listings. Since the sales time in the test data occurs later than in the training data, the time gap between Sold and Listed is, on average, longer in the test set than in the training set.

**3.2** Voting by Random Forest

After establishing a basic baseline, we proceeded to employ Random Forest for prediction. In handling missing values, we opted to fill them with zeros. Random Forest, a meta-estimator, utilizes multiple decision tree classifiers on various subsets of the dataset, leveraging averaging to enhance predictive accuracy and mitigate the risk of overfitting.

Through a series of hyperparameter tuning iterations, we arrived at the optimal hyperparameters:

n\_estimators: 600

min\_samples\_leaf: 9

max\_depth: 18

These settings serve a dual purpose - reducing the influence of outliers to prevent overfitting while ensuring precise predictions without underfitting. Ultimately, we obtained the model's results:

Train RMSE： 0.08056

Validation RMSE： 0.11420

**Test RMSE： 0.12608**

**Test Ranking： #12 / #173**

Overall, the model has enhanced accuracy by approximately 2% compared to mere guesswork and has ascended to a prominent position in the rankings by effectively capturing data features. However, one drawback of this model is the noticeable decrease in computational speed as the number of models and their depth increases, with limited improvement in predictive performance. Additionally, Random Forest necessitates the handling of missing values, and the choice of handling methods can significantly impact the results. To mitigate potential adverse effects, we opted for a straightforward approach by directly imputing missing values with zeros.

**3.3** Boosting by CatBoost

CatBoost is an algorithm for gradient boosting on decision trees. Compared to traditional random forests, CatBoost not only boasts faster computation speed but also allows assigning different weights to individual trees, thereby further enhancing model accuracy through error-driven training. By adjusting hyperparameters, increasing the number of iterations, and reducing the learning rate, we obtained the optimal model parameters as follows:

Iterations: 1700

Learning Rate: 0.023

Depth: 10

These settings took into consideration the presence of significant noise in the data. Consequently, the number of training iterations was increased, while the learning rate was reduced to enhance convergence and stability. As a result, the model achieved the following outcomes:

Train RMSE： 0.06969

Validation RMSE： 0.10514

**Test RMSE： 0.12458**

**Test Ranking： #6 / #173**

**3.4 Stacking by Combining Models**

We observed that both Random Forest and CatBoost achieved favorable rankings, despite their entirely different analytical approaches. Given the distinct strengths of each model, we hypothesized that their areas of excellence might differ. To capitalize on these strengths, we combined the results of both models through 40% Random Forest and 60% CatBoost, resulting in an even more superior performance on the test data:

**Test RMSE：0.12272**

**Test Ranking：#3 / #173**

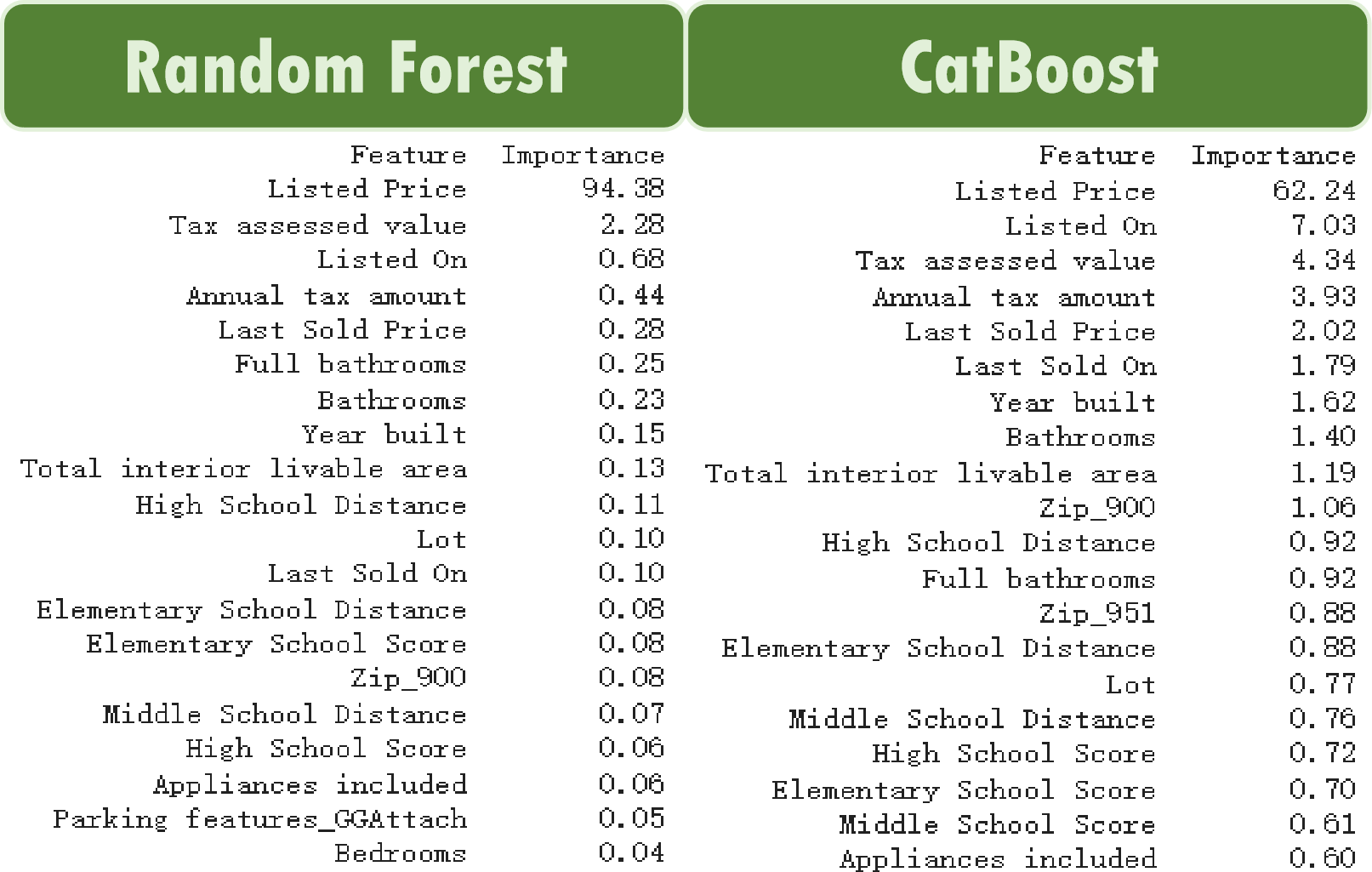
In addition, in the Test dataset, there is a minor data leakage in the column containing house descriptions due to the fact that the scraped data timestamp is later than the sale timestamp. By conducting text analysis, we were able to extract a portion of the house sale prices from this column. By incorporating these updated prices, the model score improved to 0.1144, resulting in a 0.5% improvement over the #1.

4 Discussion

In this section, we will compare the two models we used and summarize the ways in which we can further enhance their performance.

**4.1** Model Comparison

By comparing the feature weights of the two models, we can clearly discern the differences between them.



**Figure 5: Feature Importance from two Models**

Random Forest is a voting method where each tree has equal weight. As a result, important features become prominent due to aggregation. The weight of "Listed Price" exceeds 94%, which, on one hand, confirms that using only this single variable can yield excellent predictive performance. On the other hand, it also reflects the limitations of the Random Forest model. Further optimization is challenging as "Listed Price" itself contains a significant amount of noise.

However, CatBoost is a Boosting method where trees have varying weights, leading to relatively increased importance for other features. While "Listed Price" still holds a dominant position with 62%, it is significantly lower than its weight in Random Forest. Therefore, through techniques such as feature engineering, outlier handling, and other data preprocessing methods, we anticipate that CatBoost's performance can be further improved.

**4.2** Future Enhancement

We can further enhance the performance of CatBoost through four dimensions. Firstly, optimizing the selection of the validation set. In this analysis, we randomly selected the validation set. However, CatBoost has higher data distribution requirements than other models, making it essential to match the distribution of the test set as closely as possible. This will help us find the most suitable hyperparameters.

Secondly, considering that CatBoost excels at utilizing multiple variables to correct the bias in Listed Price, we can employ feature engineering to create more meaningful new features, such as price per room and price per square foot. These additional features could provide the model with more information to enhance its performance.

Thirdly, in terms of identifying outliers, we can detect data errors within both the train and test datasets by examining internal data correlations. For instance, identifying instances where the list price or space significantly exceeds the normal range. Addressing these data errors can further enhance result accuracy.

Lastly, it's about handling missing values. Although one major advantage of CatBoost is its ability to handle missing values gracefully, further improving the model's performance can be achieved by filling in the missing values through appropriate methods. For instance, we can employ ensemble models to predict each missing value based on non-missing variables.

5 Conclusion

In this project, we leveraged traditional machine learning methods, coupled with meticulous data preprocessing and model tuning, to achieve a successful prediction of California housing sales data, securing a commendable position on the competition leaderboard.

During this process, it became evident that Data Understanding played a pivotal role. We discovered that even a simple model utilizing just one feature could outperform 80% of the competing teams. This underscores the significance of grasping the nuances of the dataset. If you don't understand the data, the model's results may be less accurate than random guessing.

Furthermore, Data Preprocessing consumed a substantial portion of our time, surpassing 80% of the project's duration. This substantial investment reflects a fundamental truth: regardless of the model employed, data quality decisively determines the outcome. To enhance model effectiveness, we ultimately circled back to the essential task of elevating data quality.

Lastly, while we achieved commendable results using Ensemble models, it's worth noting that compared to the efficiency of a concise AutoML solution, our traditional approach may appear relatively less efficient. In the not-so-distant future, AutoML might indeed become the predominant tool of choice in data analysis, simplifying the process even further.

REFERENCES

1. Kaggle Competition - Predict California sales prices: https://www.kaggle.com/competitions/california-house-prices
2. AutoGluon: AutoML for Image, Text, Time Series, and Tabular Data: https://auto.gluon.ai/stable/index.html
3. RandomForestRegressor: https://scikit-learn.org/stable/modules/generated/ sklearn.ensemble.RandomForestRegressor.html
4. CatBoost: https://catboost.ai/

[5] Combine predictors using stacking: https://scikit-learn.org/stable/ auto\_examples/ensemble/plot\_stack\_predictors.htmlConference Name:ACM Woodstock conference

Conference Short Name:WOODSTOCK’18

Conference Location:El Paso, Texas USA

ISBN:978-1-4503-0000-0/18/06

Year:2018

Date:June

Copyright Year:2018

Copyright Statement:rightsretained

DOI:10.1145/1234567890

RRH: F. Surname et al.

Price:$15.00