USA HOUSING DATASET: House Value Prediction

❖ INTRODUCTION

The housing market today is one of the most competitive and profitable sales markets in the world. Because of this, being able to accurately predict and have knowledge of sales prices for houses is very important for everyone involved in the housing market. For housing buyers in particular having an accurate idea of the true price of a property they are looking to buy, or how other factors can play into the price is very useful information for them to have in order to make an informed decision about purchasing property (Mukhlishin [1]). For these reasons this is why we decided to focus our research on creating accurate machine learning models to predict the Sales Price for houses. Machine learning has been an ever growing and very useful tool in many different aspects of society, making overall accurate predictions for use in a variety of different fields, as Ayush Varma, and Colleagues, stated in their paper "Various modern applications of this technique [machine learning] include predicting stock prices, predicting the possibility of an earthquake, predicting company sales and the list has endless possibilities," (Varma [2]). Using these data driven methods for housing can be very useful. As in many other fields, machine learning can be used to make accurate and informed predictions for housing prices based on real data that influences the overall price of a house. This can be used to show people involved in the housing market direct ways that they can improve the value of their property, or a way for new buyers to look at accurate predictions for the pricing of a house. The usefulness of machine learning in the housing market was detailed in a paper by Lan Hu, and Colleagues, about using random forest models to housing sale price where they stated:

House price prediction has attracted considerable research attention largely because of its utility as an economic indicator for stakeholders, homeowners, investors, and other real estate market participants. The availability of a house price predictor that generates accurate predictions can fill a critical information gap in the literature and enhances real estate market efficiency. (Hu [3])

With how useful machine learning can be in the housing market, we hope that our research will lead to accurate predictive results that can be useful for any individual that is based in the housing industry.

With these goals in mind, picking the dataset that we wanted to use was our next step. There were a variety of different factors that we considered when picking our dataset. Some of the main things that we were looking for were a larger dataset so we could a rather large sample size for our modeling, one that had a variety of different variables so we weren't limited in our analysis when it came to picking predictors, and one that would not require a large amount of preprocessing or cleaning in order to streamline the analysis. Having this in mind, we ended up deciding to base our analysis on was a dataset that we got from Kaggle called "USA Housing Dataset" in this dataset, there are 81 different variables that covers a range of information regarding houses in the US, and nearly 3000 different houses that are contained within the dataset. Additionally, this dataset did not seem to contain any egregious formatting errors or aspects of it that would require a lot of cleaning or preprocessing. With this large amount of overall data, variables to consider, and presentation of the data, this dataset seemed perfect for our analysis going forward.

❖ METHODS

Dataset Background

The dataset used for this project was found on kaggle.com:

https://www.kaggle.com/datasets/gpandi007/usa-housing-dataset

This Dataset was compiled by Dean De Cock and is a modernized and expanded version of the often-cited Boston Housing Dataset. It contains information on 80 different features of residential homes in the USA, including the size of the lot, the number of bedrooms and bathrooms, the type of foundation, and many others. The dataset also includes the sale price of each home, which is the target variable for most predictive modeling tasks.

Data exploration

We began our data exploration by creating various views such as histograms, boxplots, and scatterplots to gain insights into the dataset. The histogram shown in **Figure 1** illustrates that the sale price of the houses is positively skewed and most of the sale prices fall within the lower range. This indicates that most homes in the dataset are priced lower.

The vertical box plot shown in **Figure 2**, based on the Sale Price column in the dataset, confirms the right-skewed distribution of the sale prices. The whisker is shorter on the lower end of the box, indicating that there are fewer homes with low sale prices. Conversely, the box plot displays a long tail to the upper side of the median, indicating that there are a small number of homes with very high sale prices. The summary shows property prices, with the minimum value at $35,311 and the maximum value at $538,000. The median property price is $160,000, with Q1 at $129,000 and Q3 at $207,500. There are several outliers in the dataset, with values exceeding the maximum price of $325,000.

The scatterplot constructed between the Sale price and the total living area above ground level in square feet, shown in **Figure 3**, reveals that most of the house living areas range from 1000sq ft to 3000 sq ft, with very few homes above 3000 sq ft. As the area of the house increases, the sale price also increases. This indicates that there is a positive correlation between the size of the living area and the sale price of the house.

The next scatter plot shown in **Figure 4** demonstrates that the Overall Quality rating of the homes in the dataset ranges from 4 to 8. This suggests that most of the homes have moderate to high-quality finishes and materials, based on the expert rating system.

In conclusion, our initial data exploration using various views has provided some valuable insights into the dataset. We have learned that most homes in the dataset are priced lower and have a moderate to high overall quality rating. Additionally, the size of the living area has a positive correlation with the sale price of the house, indicating that larger homes tend to have higher sale prices.

Data Cleaning

Data cleaning is an essential step in the data analysis process that involves identifying and correcting or removing errors and inconsistencies from the dataset. In the dataset, we performed various data cleaning steps to ensure the data's quality and accuracy.

One of the first steps we took was to remove columns where more than 10% of the data was missing. Missing data can significantly impact the analysis and predictions based on the dataset, so it's essential to remove such columns from the dataset. These are the parameters that are removed in above condition LotFrontage, Alley, FireplaceQu, PoolQC, Fence, MiscFeature.

Next, we filled the missing values of other columns with the mean or mode of the column, depending on whether it was numerical or categorical. Filling in the missing data can help to ensure that the dataset is complete and accurate, and it also prevents data loss due to missing values.

We also checked for any duplicate rows in the dataset. Duplicate rows can skew the analysis and lead to inaccurate results, so it's crucial to identify and remove them from the dataset.

Overall, these data-cleaning steps help to ensure that the dataset is accurate, complete, and suitable for analysis. By cleaning the data, we can ensure that any analysis or predictions based on the dataset are reliable and accurate.

Preprocessing

Based on our initial analysis, we observed that the dataset contains some outliers. These outliers can significantly affect the performance of our machine-learning model by influencing the model's parameters. Hence, it is essential to remove these outliers before training our model.

Z-score-based outlier removal is a commonly used technique for identifying and removing outliers in numerical data. It works by calculating the Z-score for each data point in a column, which represents how many standard deviations away from the mean the data point is.

The formula for calculating the Z-score for a data point is $Z = (x - \mu) / \sigma$ Where x is the data point, μ is the mean of the column, and σ is the standard deviation of the column.

Once we have calculated the Z-score for each data point in a column, we can then determine which data points are considered outliers based on a certain threshold value. A common threshold value is between 3 to 4.5, which means that any data points with a Z-score greater or less than the threshold value are considered outliers.

Once the outliers have been identified, they are removed from the dataset. Removing outliers helps to ensure that extreme values in the data do not skew the analysis or predictions based on the dataset.

Z-score-based outlier removal is a powerful technique for identifying and removing outliers in numerical data. It is easy to implement and can help to improve the accuracy and reliability of data analysis and predictions based on the dataset. After multiple Iterations, we selected the threshold value of 4.5 which removed 118 rows which is almost 10% of the rows in the entire train dataset.

Feature Engineering

Feature engineering is the process of transforming raw data into features that can improve the performance of machine learning models. In our analysis, we identified highly correlated features using the correlation matrix. We removed the columns 'Exterior1st', 'GarageArea', 'GrLivArea', 'Exterior2nd', 'TotRmsAbvGrd', 'GarageCars', 'GarageCond', and 'GarageQual' from the dataset because they were highly correlated with other features. The removal of these features reduced the dimensionality of the dataset and prevented multicollinearity issues that can negatively affect the performance of machine learning models.

Price Prediction using Machine Learning Models

The aim is to predict the sale price using several machine learning models, including linear regression, elastic net, ridge regression, lasso, decision tree, random forest, XGBoost, and Gradient Boosting. Each of these models will be trained on the dataset and evaluated to determine which model performs the best in predicting the sale price. By using multiple models, we hope to improve the accuracy of our predictions and identify which model is most suitable for this specific dataset. Ultimately, the goal is to find the most accurate and reliable model that can be used to predict the sale price of a house given its features.

- **RESULTS**

Performance of the models

1. **Linear Regression & Elastic Net**

There are 947 rows and 66 columns in the train data. On the other hand there are 395 rows and 66 columns in validation data. We started by plotting the basic linear regression model first based on the train and validation data. This is the most basic model, that gives us an root mean square error (RMSE) value of 24053 followed by an accuracy of 88.72%.

Similarly, we tried the Elastic net model, which combines properties of lasso and ridge model. The 'glmnet' funct ion is used, that fits a regularized linear model using the Elastic Net penalty. The RMSE value for this model is 23780 which is more compared to linear regression. The accuracy score is 88.9% which is a bit more than linear regression.

2. Lasso & Ridge Regression

After using these, we tried implementing the Lasso and Ridge regression models.Lasso and Ridge regression models are two popular types of linear regression models that use regularization techniques to prevent overfitting and improve the performance of the model. Using Lasso regression, with alpha value = 1 and lamda value as 0.1, we get an RMSE value of 24056 and accuracy of 88.72%. Whereas for ridge regression, keeping alpha = 0 and lambda 0.1 the same, we get the same RMSE and accuracy values. This implies that there is not much difference between the lasso and ridge outputs. Comparing all these 4 models we can say that the regularized models are performing better than the simple linear regression model.

3. Decision Tree & Random Forest

We then plotted the decision tree. “OverQual” was the first node selected based on which the tree was further splitted. This you can see from the tree plotted in fig5. After fitting the model when we perform predictions we observe that the error rate is much higher than the previous plots. The value is around 39350 followed by an accuracy score of 69% only. Decision trees are prone to overfitting if the tree is too deep or if the tree is allowed to grow until all the leaves are pure. Overfitting can lead to a high variance model that performs well on the training data but poorly on new, unseen data. To address overfitting, you can try reducing the depth of the tree, pruning the tree, or using regularization techniques like Lasso or Ridge regression(as above).

Hence we tried plotting the random forest instead. Random Forest reduces overfitting by using bagging, which involves training multiple decision trees on different subsets of the data. By averaging the predictions of multiple trees, Random Forest reduces the variance of the model and produces more robust predictions. Plotting a random forest with 500 trees, we get a reduced RMSE value of 25450 and accuracy of 87.92% which is well compared to the decision tree.

4. Xgboost and & Gradient Boost

Both XGBoost and GBM are powerful regression models and can be used for various regression tasks. From **figures 5 and 6** we can see that the RMSE value for XGboostis 23924 while the accuracy is 88.8%. From **figure 7**, we can see how well the actual and predicted values fit on the plot using the gbm model.

On the other hand, the GBM, the RMSE performed well and gave the lowest score of 22594. The GBM model performed the best out of all the 8 models, with an accuracy of almost 90%.

The GBM model performed better than the XGBoost model, which may be due to several factors. Firstly, the GBM model may have been better suited to handle the complexity of the dataset, as it is designed to handle simpler data distributions without overfitting. Secondly, the hyperparameters for the GBM model may have been better tuned for the specific dataset, leading to better performance. Additionally, the GBM model may have been more effective at identifying and utilizing the most important features in the dataset, and the preprocessing steps used in the analysis may have been more suitable for the GBM model than the XGBoost model.

The final test data SalePrice predictions

For this particular project, we had two datasets given, train and test. We built all the above models on train data by splitting them into train and validation. Now the test dataset contains 1459 rows and 80 columns. The SalePrice attribute that the predictor variable needs to be predicted here based on a particular model. We choose the gradient boosting machine as the model performs better compared to all. We have then performed the similar data cleaning and preprocessing steps as that of the train data on our test data. **Figure 8**, shows the dataframe for the same. We can say that we are 90% confident that the predictions made by our model are correct and so are the sale price values for the unseen test data. We save and write this dataframe in a csv format.

❖ DISCUSSIONS

The analysis of housing market dynamics has long been a topic of interest in economic research. Predictive modeling has become a valuable tool for understanding and forecasting housing prices. We aim to explore the significance of the USA Housing dataset and its implications for future research on house price prediction, by examining various parameters such as LotArea, OverallQual, OverallCond, 1stFirSF, 2ndFirSF and other relevant factors. We aim to develop robust models that accurately predict housing prices. From the RMSE and accuracy values provided, it appears that the Gradient Boost model performed the best, with an RMSE of 22,594.26 and an accuracy of 89.98%.

The predictive models built using the USA Housing dataset can provide valuable insights for market analysis and investment strategies. By accurately forecasting housing prices, investors can make informed decisions on property acquisition, portfolio management, and timing of investments. Future research can delve deeper into analyzing the impact of different housing market indicators, such as interest rates, employment rates, and local economic conditions, to refine predictive models and develop more robust investment strategies. By leveraging predictive models based on the USA Housing dataset, lenders can assess the risk associated with mortgage loans, determine appropriate interest rates, and make informed decisions regarding loan approvals. Future research can explore the integration of credit risk metrics, macroeconomic indicators, and borrower-specific data to enhance the predictive power of models and refine risk assessment processes.

In our project, we have applied several popular machine learning techniques for predicting house prices, including Linear Regression, Elastic Net, Lasso Regression, Ridge Regression, Decision Tree, Random Forest, Gradient Boost, and XgBoost. Each of these methods has its own strengths and weaknesses.

Linear regression models are simple and easy to interpret, but they may not capture complex non-linear relationships between variables. Elastic Net and Lasso Regression are useful for feature selection and can help prevent overfitting, but they may struggle with high-dimensional data. Ridge Regression can help prevent overfitting, but it may not be as effective for feature selection.

Decision Trees and Random Forests are useful for modeling complex relationships between variables, but they may require larger and more diverse datasets to avoid overfitting. Gradient Boost and XgBoost are powerful ensemble techniques that can improve the accuracy of predictions, but they may be computationally expensive and may require careful tuning of hyperparameters.

One of the main challenges in predicting house prices is obtaining high-quality data that is both comprehensive and diverse. Real estate datasets can be costly to acquire, and not all datasets contain all the relevant variables needed for accurate predictions. Incomplete or inaccurate data can result in biased or unreliable predictions, limiting the usefulness of the model. Another challenge is that the data used for training the model may not accurately reflect future market conditions. Economic or demographic changes, zoning regulations, or other factors can cause fluctuations in the real estate market that can affect the accuracy of the model.

Real estate is a complex and dynamic market, and there are often non-linear and complex relationships between variables that can affect house prices. For example, the impact of a neighborhood's crime rate on house prices may not be straightforward, as it may depend on other factors such as the type of crime, the demographic makeup of the neighborhood, or the availability of other amenities. To accurately predict house prices, it is important to identify and account for these complex relationships between variables. This can be challenging, as traditional linear regression models may not be sufficient to capture these non-linear relationships. Techniques such as decision trees or random forests can be useful for modeling non-linear relationships, but they may require larger and more diverse datasets to avoid overfitting.

References

[1] M. F. Mukhlishin, R. Saputra and A. Wibowo, "Predicting house sale price using fuzzy logic, Artificial Neural Network and K-Nearest Neighbor," *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, Semarang, Indonesia, 2017, pp. 171-176, doi: 10.1109/ICICOS.2017.8276357.

[2] A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," *2018 Second International Conference*

on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.

[3] Hu, L., Chun, Y., & Griffith, D. A. (2022). Incorporating spatial autocorrelation into house sale price prediction using random forest model. *Transactions in GIS*, 26, 2123– 2144. https://doi.org/10.1111/tgis.12931

Tables and Figures

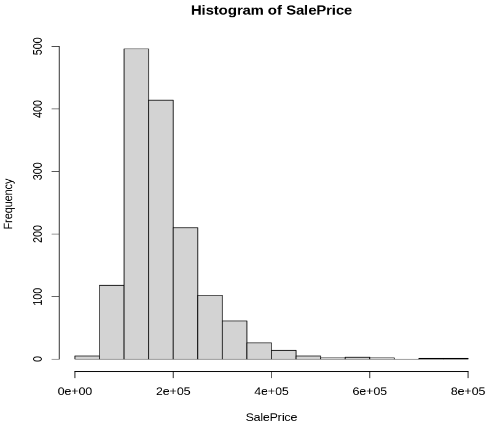


Figure 1 - Histogram on Saleprice

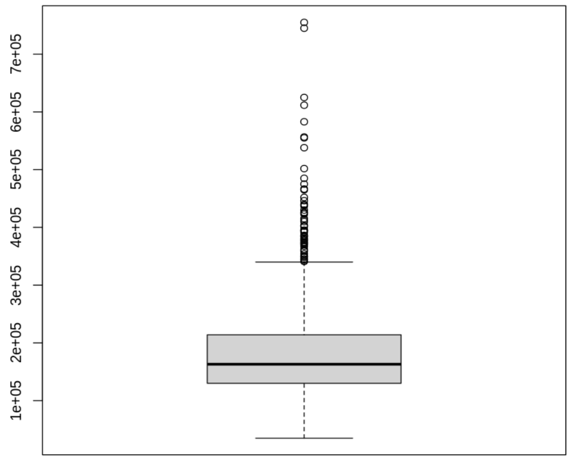


Figure 2 - Boxplot on Saleprice

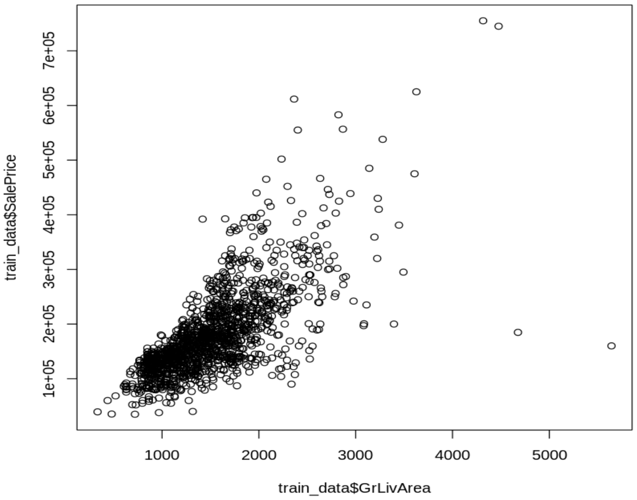


Figure 3 - Scatterplot on SalePrice vs GrtLivArea

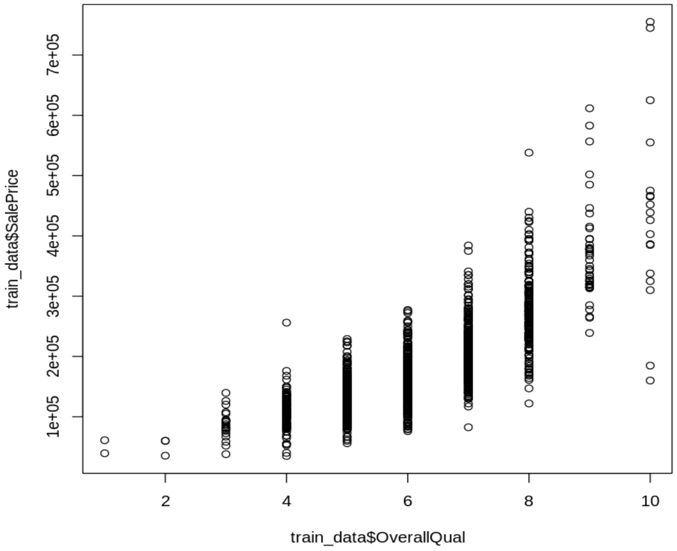
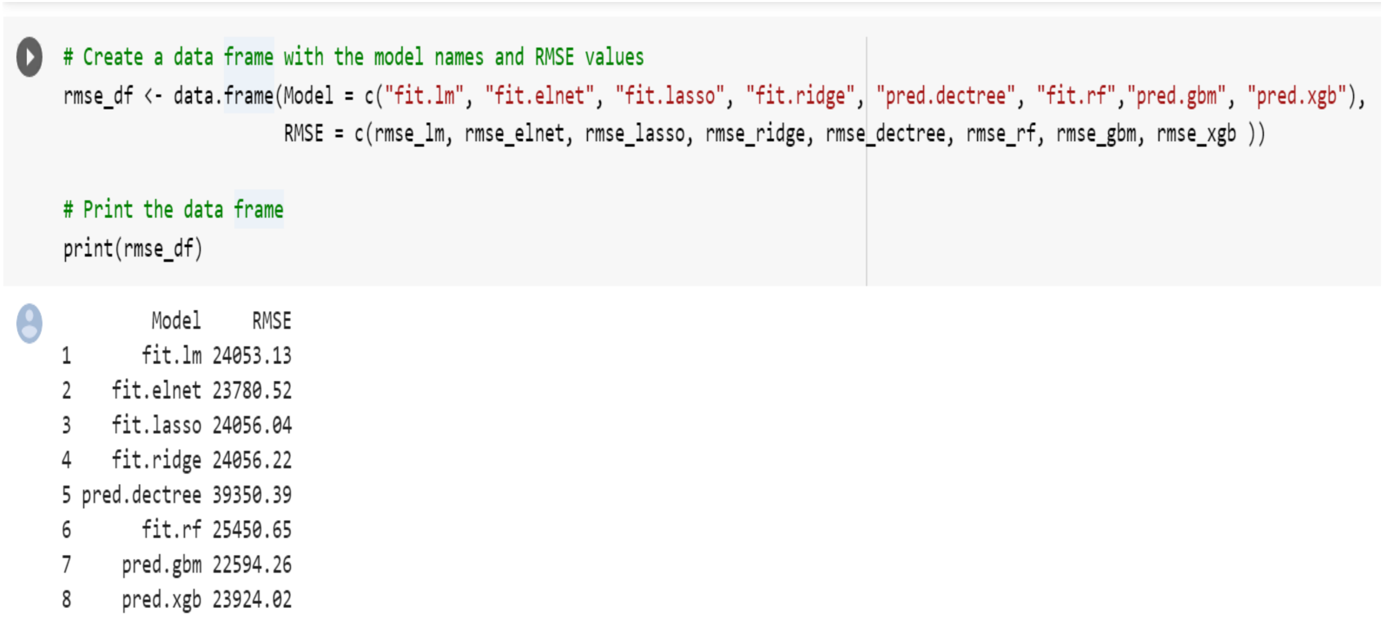


Figure 4 - Scatterplot on SalePrice vs OverallQual

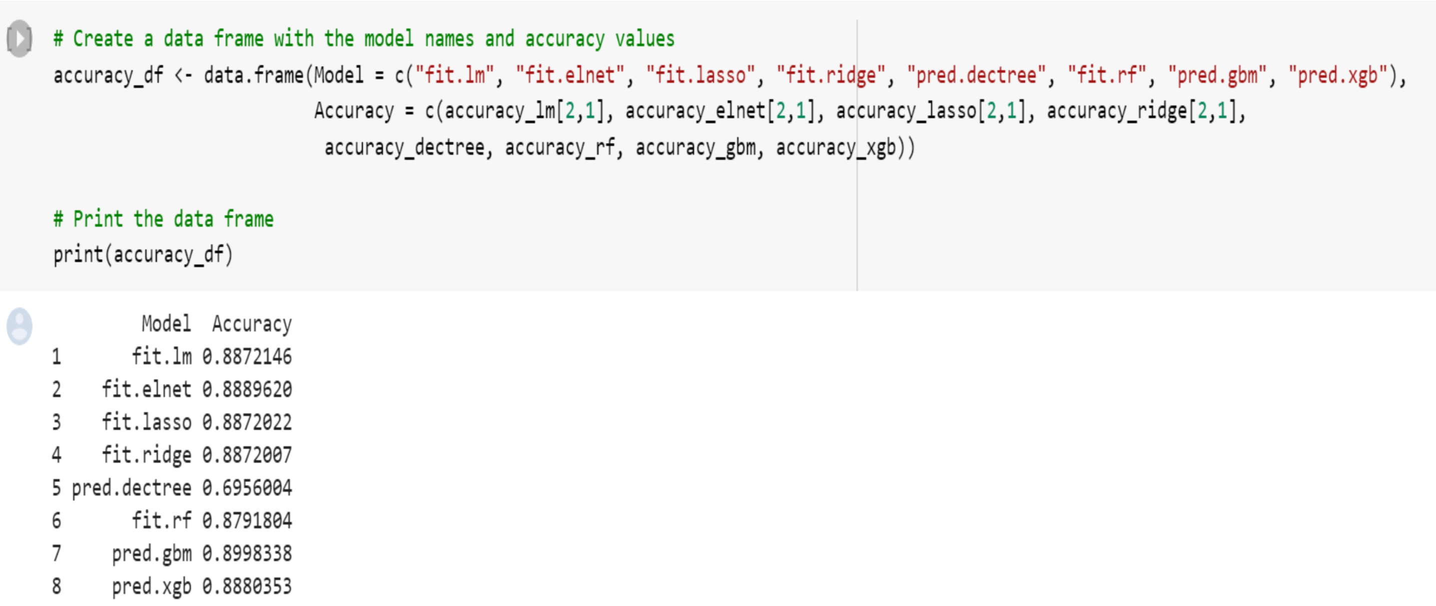


```
# Create a data frame with the model names and RMSE values
rmse_df <- data.frame(Model = c("fit.lm", "fit.elnet", "fit.lasso", "fit.ridge", "pred.dectree", "fit.rf","pred.gbm", "pred.xgb"),
                      RMSE = c(rmse_lm, rmse_elnet, rmse_lasso, rmse_ridge, rmse_dectree, rmse_rf, rmse_gbm, rmse_xgb ))

# Print the data frame
print(rmse_df)
```

```
         Model     RMSE
1       fit.lm 24053.13
2    fit.elnet 23780.52
3    fit.lasso 24056.04
4    fit.ridge 24056.22
5 pred.dectree 39350.39
6       fit.rf 25450.65
7     pred.gbm 22594.26
8     pred.xgb 23924.02
```

Figure 5 - RMSE Comparison



```
# Create a data frame with the model names and accuracy values
accuracy_df <- data.frame(Model = c("fit.lm", "fit.elnet", "fit.lasso", "fit.ridge", "pred.dectree", "fit.rf", "pred.gbm", "pred.xgb"),
                          Accuracy = c(accuracy_lm[2,1], accuracy_elnet[2,1], accuracy_lasso[2,1], accuracy_ridge[2,1],
                           accuracy_dectree, accuracy_rf, accuracy_gbm, accuracy_xgb))

# Print the data frame
print(accuracy_df)
```

```
         Model  Accuracy
1       fit.lm 0.8872146
2    fit.elnet 0.8889620
3    fit.lasso 0.8872022
4    fit.ridge 0.8872007
5 pred.dectree 0.6956004
6       fit.rf 0.8791804
7     pred.gbm 0.8998338
8     pred.xgb 0.8880353
```

Figure 6 - Accuracy Comparison

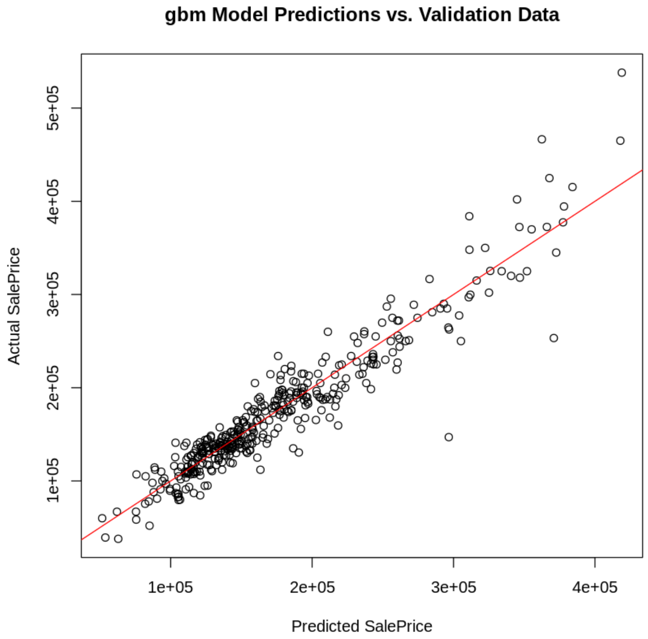
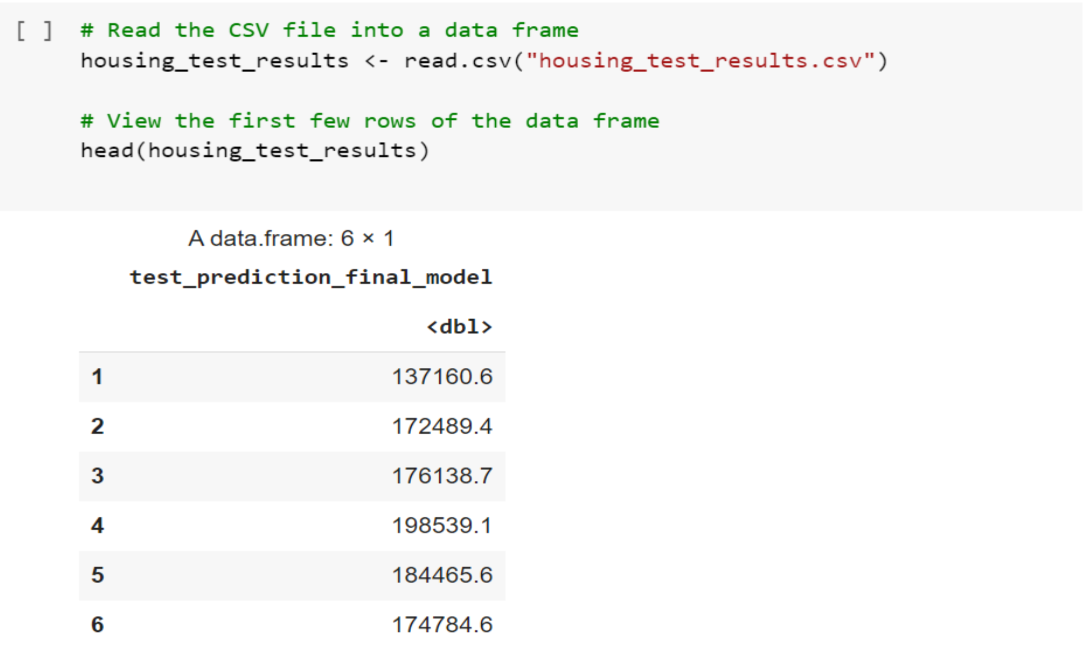


Figure 7 - The Predicted vs Actual “SalePrice plot”



```
[ ]  # Read the CSV file into a data frame
     housing_test_results <- read.csv("housing_test_results.csv")

     # View the first few rows of the data frame
     head(housing_test_results)
```

A data.frame: 6 × 1

	test_prediction_final_model
	<dbl>
1	137160.6
2	172489.4
3	176138.7
4	198539.1
5	184465.6
6	174784.6

Figure 8 - Final test data predictions