Andrew Seal

Capstone 2: Ames, Iowa – Housing Data

1. Introduction:

The US Census Department reports that 28.9% of US household wealth is in the form of home equity, making housing the second most significant store of wealth behind retirement accounts[1].

In fact, for most people in the United States, the purchase of a home is one of the largest financial transactions they will ever make [2]. Homes are not traded on a liquid market and unlike some other assets the market price of real estate is often difficult to determine. This can leave buyers without a reliable mechanism to determine if they are paying an appropriate sum for a given house.

The goal of this project is to create a model that will use the distinct features of homes to predict the Sales Price of a given house.

2. Dataset

For this project, I utilized a publicly available dataset of recent home sales in the town of Ames, Iowa. The dataset consisted of 2,930 distinct sales within the town of Ames between 2006 and 2010. In total there were 82 columns in our data set, each representing a feature of the home. Features include quantifiable characteristics of the home such as:

- Total Square Footage
- Lot Size
- Number of Bedrooms
- Number of Fireplaces…etc.

Additionally, the features include categorical data such as:

- Zoning
- Neighborhood
- Condition of the home
- Whether the garage is attached or detached …etc

The original data was provided in a .CSV file and was imported into a pandas data frame called housing_data.

[1] US Census Bureau: https://www.census.gov/content/dam/Census/library/publications/2020/demo/p70br-170.pdf

[2] National Association of Realtors: https://www.nar.realtor/research-and-statistics/research-reports/highlights-from-the-profile-of-home-buyers-and-sellers

3. Data Cleaning and Data Wrangling

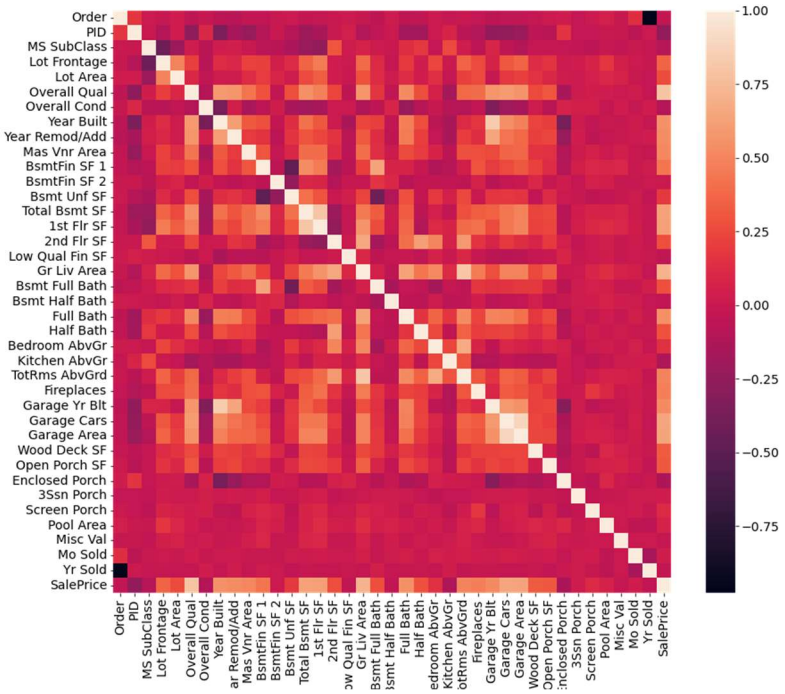
Each home in our dataset was assigned a 'Parcel ID' which is a unique identifier that I used to ensure no home sales showed up in our dataset more than one time.

With no duplicated data in our dataset, I graphed all of the numerical data into histograms in order to visually inspect the data for outliers or other anomalies. Some of our data showed outliers, for example the lot frontage on a few parcels was around 300 feet. While wider than most, a lot width of 300 feet seemed reasonable for a smaller town in Iowa and unlikely to be an error.

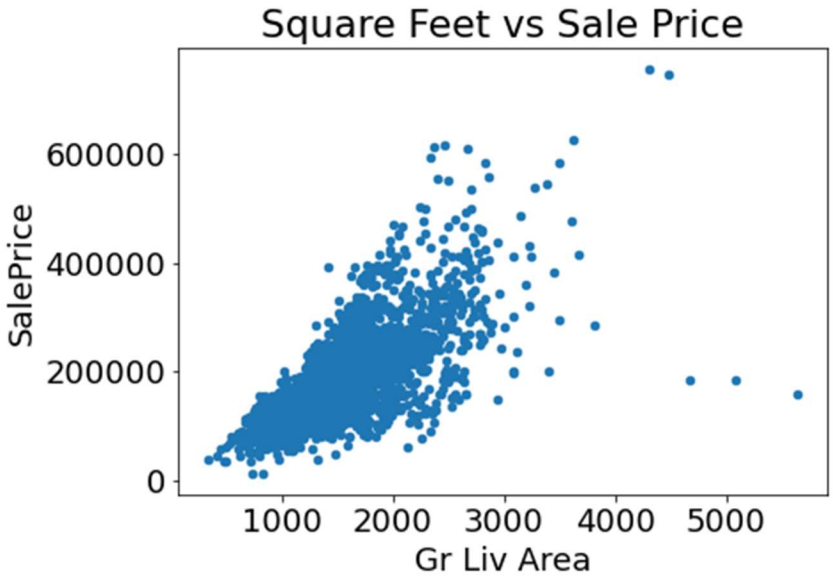
During this step I also looked into missing data and into data that was mostly homogenous. The dataset included four variables where more than 80% of the data was missing. These were examined individually and one variable "Misc Feature" was deleted as it was determined that the value of each miscellaneous feature identified in this column was reported in a separate column.

Similarly, some variables were dropped because the data was exceedingly homogenous. For example, a column named 'Utilities' was dropped since all but three of the 2,930 sales had the same value indicating they were connected to public utilities.

4. Exploratory Data Analysis

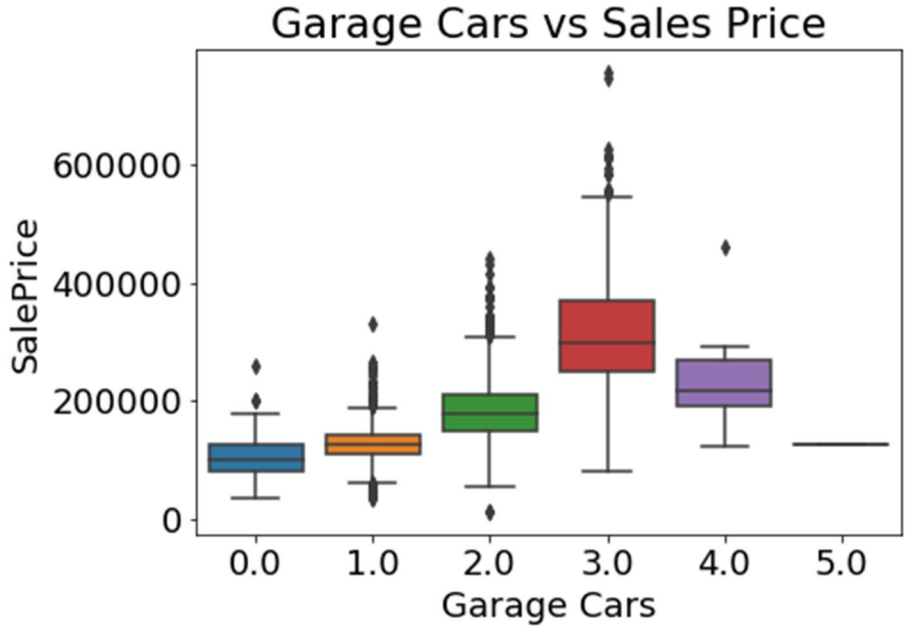


In the exploratory data analysis step, the relationship between variables is assessed. Of particular importance is the relationship between Sales Price and other variables. In the above heatmap we can see a positive correlation between Sales Price and a few variables including Overall Qual, Gr Liv Area, Full Bath, Garage Cars and others.

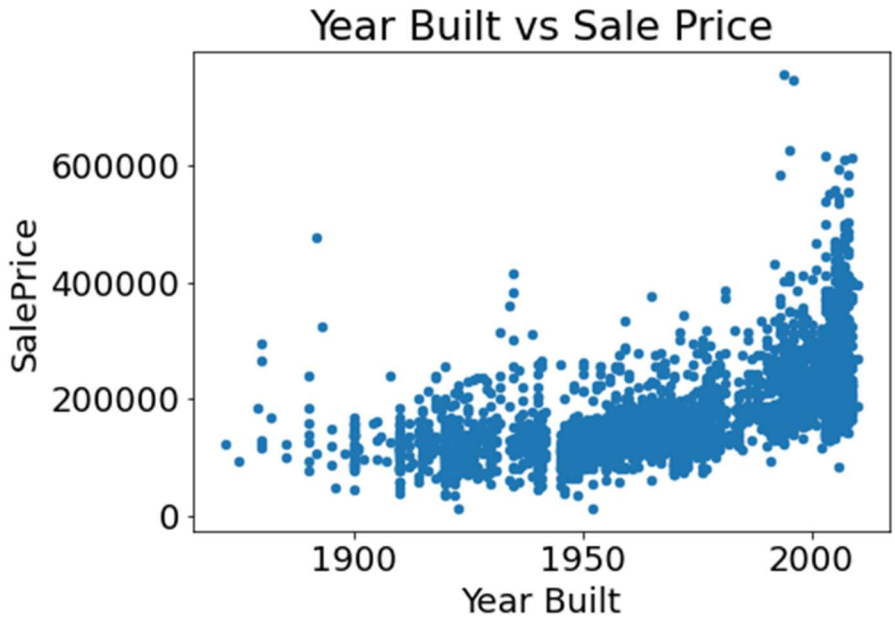


A strong positive relationship between Gross Livable Area (square feet) and sales price is apparent in the scatter plot above.

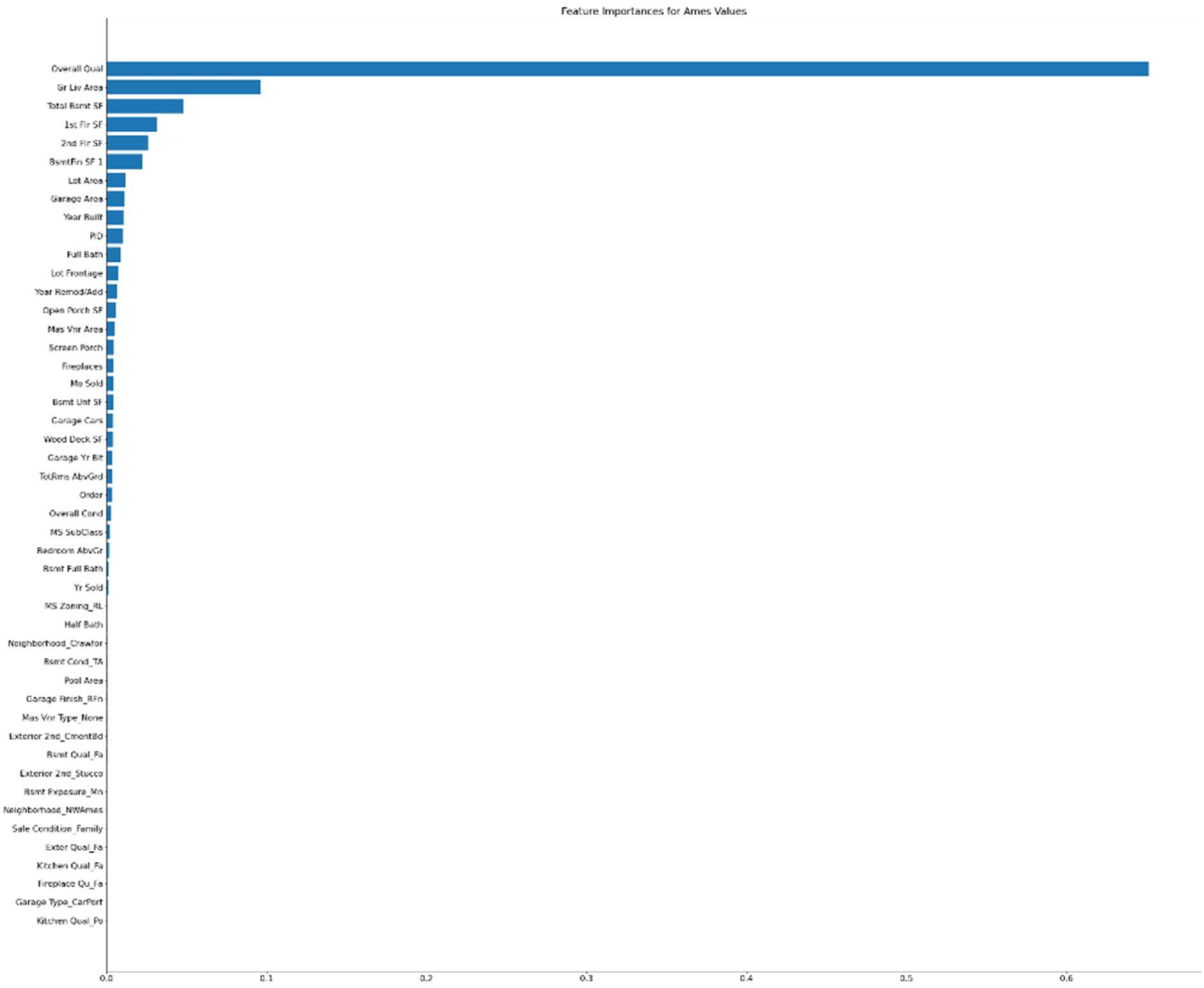
Likewise, a logical relationship is found between sales price and the number of cars the garage will hold where homes without garages sell for a discount while those with three-car garage command a higher sales price.



One additional variable of note is the year built. As shown in the chart below, recently constructed homes appear to command a high sales price.



Finally, during this step I ran a Random Forest Feature Importance analysis.



The results of the feature analysis, shown above, indicate that by far the most important variable in the analysis was the 'Overall Quality' Variable. Unfortunately, there was limited information available regarding how this quality was assessed or measured and by whom. Unlike square footage and lot size had concerns about how whether this variable would be consistent among various datasets including using this model on new and unseen information. Ultimately, I determined that it would be best to remove this variable altogether.

5. Pre-Processing and Training

Imputing Values

During this stage missing values were imputed. For the total square footage, the lot area or the lot frontage, any missing values were imputed with the mean. For some of our categorical features, missing data was imputed with the mode. At this stage, I was careful not to impute data for garage, pools, air conditioning or other features that may not be present in every property.

Creating Dummy Variables

As mentioned previously, many of our variables were categorical so at this stage I utilized the get_dummies function to create dummy variables for these categorical features. In order to avoid unnecessary collinearity, I drop one dummy variable from each categorical feature.

After imputing data and creating dummy variables, I drop my missing data and the resulting data frame now has 2,747 rows (unique sales) and 244 variables (columns).

Addressing multicollinearity

The first step is addressing simple collinearity among my variables. I examined all variables with a correlation coefficient of 0.95 or greater with any other variable and then simply eliminated one of the variables. In this step I eliminated four variables.

Next, I examined a measure of multicollinearity called the Variance Inflation Factor (VIF). In this step I used some value judgements to eliminate variables with an emphasis on retaining variables who were show to be important in the Random Forest Feature Importance exercise. The first variables to be eliminated were dummy variables who said basically the same as another variable ie condition good and fair convey similar information. At this point I eliminated many of the zoning variables as I suspected they may be correlated with other location variables such as 'neighborhood' and because they were not shown to be important in our Feature Importance analysis. This process was repeated using my best judgement until the remaining variables all had a VIF score below 10.

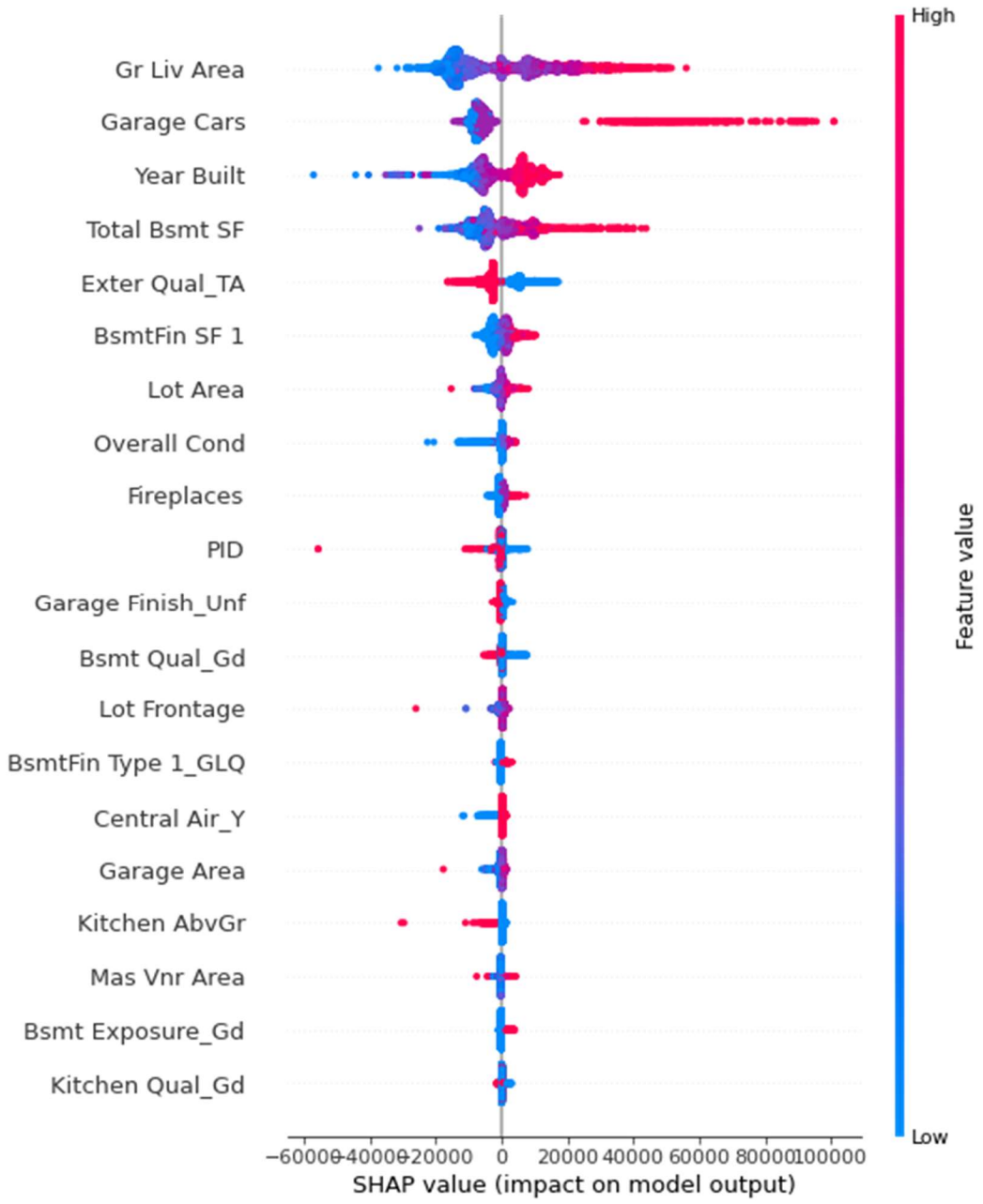
6. Two more measures of Feature Importance

Linear Regression Feature Coefficients

At this stage a significant amount of multicollinearity had been removed from our dataset so another feature importance analysis was conducted, this time using OLS regression. The first step was to standardize the data. Using the Robust Scaler feature, all of the independent variables were Standardized. The standardized data was fit into an ordinary least squares regression and the coefficients were extracted from the regression. Given the data was standardized, the coefficients should represent the relative impact of the features on the Sales Price. What was interesting is that the features identified as most important on the OLS model did not match what we had previously identified as important in the Random Forest Feature Importance. It should be noted that many of the variables identified as important in the OLS model were relatively rare features like a swimming pool. It may not make sense to build a model around these fairly rare features so a second Feature Importance analysis was conducted.

SHAP Feature Importance

The second Feature Importance Analysis that I utilized is called the Shapley Additive Explanations (SHAP) which is a measure of the average marginal contribution of a feature across all the possible combination of features. This analysis, presented below was more consistent with both logic and the Random Forest analysis previously conducted.



7. Train Test Split

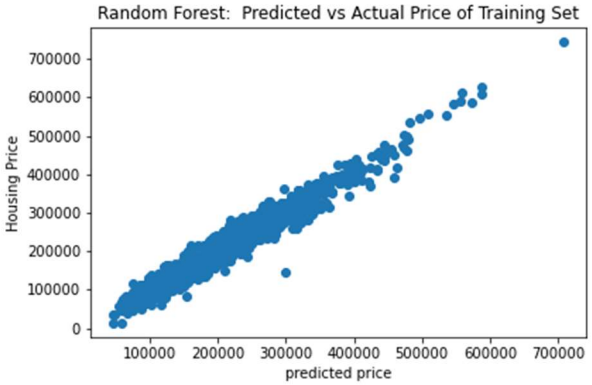
In advance of modelling, I split the data into a training set consisting of 75% of the recent sales and a training set consisting of the remaining 25% of the data. Each of the models was fitted on the training set and then tested on the test data in order to evaluate the performance of the model.

8. Modelling

Random Forest

The first model employed was Random Forest. For this algorithm I elected to optimize both the n_estimators and the max_depth hyperparameters. Instead of a grid search I elected to use a random search to optimize the two hyperparameters since grid search was computationally very expensive. It was determined the optimal hyperparameters were n_estimator of 1000 and max_depth of 40.

This model showed a strong relationship between actual and predicted values and robust R-squared and RMSE.



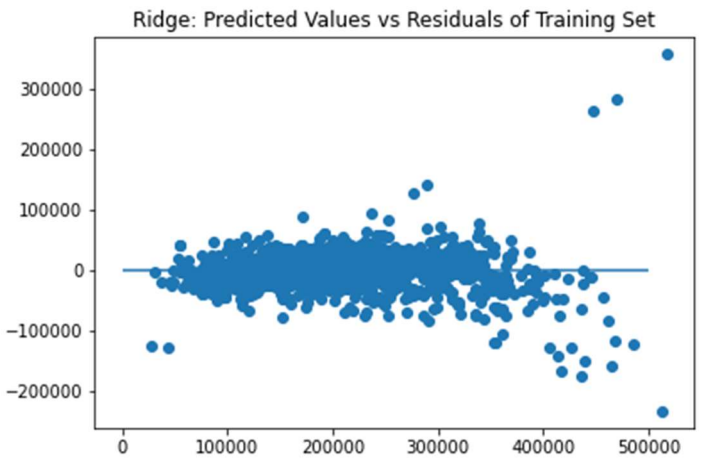
Random Forest Evaluation Metrics	Metric
r2 Score Train	0.983148
r2 Score Test	0.86261
Mean Absolute Error (MAE) Train	6187.3
Mean Absolute Error(MAE) Test	17898.6
Root Mean Square Error (RMSE) Train	10285.5
Root Mean Square Error (RMSE) Test	29735.1

Ridge Regression

Ridge Regression is useful when multicollinearity still exists between the variables. Given the large number of features in our model I felt it was prudent to run a ridge regression on the data set.

The hyperparameter of alpha was optimized using Grid Search cross validation and run using the optimized alpha.

For the ridge regression I went ahead and plotted the residuals to visually ensure linearity, and a constant variance. As shown below, nothing unusual was detected in the plot of the residuals.



Ridge Regression Evaluation Metrics	Metric
r2 Score Train	0.879826
r2 Score Test	0.882977
Mean Absolute Error (MAE) Train	17309.2
Mean Absolute Error(MAE) Test	18166.6
Root Mean Square Error (RMSE) Train	27466.3
Root Mean Square Error (RMSE) Test	27442.7

Again, this model produced a satisfactory R-squared and we take note of the RMSE.

Linear Regression

The third model evaluated for this project was a linear regression model. There were no hyperparameters to tune so instead I evaluated the model on a limited feature set consisting solely of the top twenty features identified in our SHAP feature analysis. Unfortunately, the linear regression model with a smaller feature set did not perform as well as when all variables were included. Some basic evaluation metrics of the linear regression model are presented below.

Linear Regression Evaluation Metrics	Metric
r2 Score Train	0.892771
r2 Score Test	0.883481
Mean Absolute Error (MAE) Train	16841.5
Mean Absolute Error(MAE) Test	18177.6
Root Mean Square Error (RMSE) Train	25944.7
Root Mean Square Error (RMSE) Test	27383.6

Gradient Boosting

The final algorithm was a gradient boosting model. Utilizing a grid search cross validation method, I optimized the learning rate hyperparameter. This model performed slightly better than our other model when measuring both the R-squared and the RMSE. Some evaluation metrics of the Gradient Boosting model are shown below.

Gradient Boosting Evaluation Metrics	Metric
r2 Score Train	0.955194
r2 Score Test	0.892698
Mean Absolute Error (MAE) Train	12262.2
Mean Absolute Error(MAE) Test	17322.4
Root Mean Square Error (RMSE) Train	16771.1
Root Mean Square Error (RMSE) Test	26278.1

Selecting a Model

For this project I used four different regression algorithms to model the Ames, Iowa housing data with the objective of finding the best model to predict a dependent variable (Sales Price) based on the features of the property.

For each model I used a grid (or random) search cross-validation to optimize the hyperparameters. For the normal Linear Regression Model, no hyperparameter optimization was available so I reduced the number of features to just the top twenty variables as identified by our SHAP feature importance.

The table below outlines which hyperparameters were optimized for each model as well as the optimal hyperparameters as indicated by the cross-validation.

Model	r2 Score (Test-Data)	Hyperparameter To Optimize	Optimal Hyperparameter
Random Forest	0.86261	n_estimators	n_estimators = 1000
Random Forest	0.86261	max_depth	max_depth = 40
Ridge Regression	0.882977	Alpha	Alpha = 6
Linear Regression	0.883481	Number of Variables	Include All Variables
Gradient Boosting	0.892716	Learning Rate	Learning Rate = 0.1

A key metric that was examined is the R-squared, which is the extent to which the variance in the independent variables influences the variance in the dependent variable. This is often referred to as the 'Goodness of Fit' and is an indication of how well the data fits the regression model. A higher R-squared will indicate a model that creates a better fit of the data.

Of our models, the Gradient Boosting yielded the best R-squared on the testing data (though all models performed fairly well.)

The other metric I focused on was the Root Mean Square Error (RMSE) which is a measure of how far our predictions fall from our actual Sales Price. The RMSE is in the same units as our dependent variable

(US Dollars) and is therefore easy to conceptualize its magnitude. A lower RMSE is obviously preferred. By this measure the Gradient Boosting Model performed better than the other models.

Evaluation Metric	Random Forest	Ridge Regression	Linear Regression	Gradient Boosting
r2 Score Train	0.983148	0.879826	0.892771	0.955194
r2 Score Test	0.86261	0.882977	0.883481	0.892716
Mean Absolute Error (MAE) Train	6187.3	17309.2	16841.5	12262.2
Mean Absolute Error(MAE) Test	17898.6	18166.6	18177.6	17340.8
Root Mean Square Error (RMSE) Train	10285.5	27466.3	25944.7	16771.1
Root Mean Square Error (RMSE) Test	29735.1	27442.7	27383.6	26276

While the Random Forest performs best on the training set, the objective of the model is to determine the price of a real estate offering where the sales price is unknown. Given the superior performance of the Gradient Boosting model both in terms of its R-Squared and RMSE on the test data, I would select the Gradient Boosting Model and expect it to most accurately predict the Sales Price.

Selected Model	Gradient Boosting
Model	Gradient Boosting
Hyperparameter to Optimize	Learning Rate
Optimal Hyperparameter	Learning Rate = 0.1
r2 Score Train	0.95519
r2 Score Test	0.89272
Root Mean Square Error (RMSE) Train	16771.0
Root Mean Square Error (RMSE) Test	26276.0

9. Running a Test Case

To illustrate the functionality of the model constructed I randomly selected one property from our test data and used the gradient boosting model to predict the sales price of the home.

On this particular home, the model predicted a house price of $133,746. When I looked up this actual sales price of this home, it in fact sold for $124,500. While a single sale is not a good indicator of the performance of the model, it is nonetheless fun to put the model to use in a manner that simulates its real-world application!