Evan Krook: Market Pricing Analysts

Predicting the House Sale Prices in Ames, Iowa

Hypothetical Scenario: Market analyst for Realtor.com in Ames, Iowa and was tasked with creating a model to predict a house sale price based off 79 variables.

Introduction (Framed as if manager asking me/Problem Statement):

Task: Create a Model to predict a house sale price in Ames, Iowa based on 79 variables ranging from square footage of the basement to the type of neighborhood (residential, city, downtown, etc.).

The housing market throughout the United States has been on whirlwind since the outbreak of covid-19 with prices skyrocketing. In November of 2023 the median house sale price in Ames was $313,000, which is a 16% increase from November 2022. Looking at future statistics this momentum looks to not be stopping anytime soon. This has created a wordiness among potential home buyers because they are afraid to overspend on a house. I ask you to create a predictive model using the 79 most crucial features that go into the pricing of a house for sale (square footage, number of bathrooms, number of bedrooms, year built, etc.) This model will help customers get a glimpse at the estimated price for a house with a certain criterion they desire. Also, this will benefit Realtor.com because it will attract customers to use our pricing model for their home buying needs. We have provided a dataset with the variables to be utilized within the model and have a preset train/test split with the test set sale price feature unavailable. The model will be tested against the test set actual values and graded with a root mean squared error (RMSE) value to see how far your predictions are from the true values.

Approach: Intro Model and Dataset/Data Preprocessing/Creation and Running Model/Results/Findings:

All my code for data preprocessing, model generation, and strength testing was conducted in Python.

The model I will be using is a gradient boosting regressor. The model takes many simple models and generates a final composite model. The benefit to gradient boosting is that each simple model corrects the errors of the next and captures the key patterns within to input into the final composite model.

To go along with that I will be using a random search CV to get the optimal parameters for the model. This technique goes through a set list for each parameter in the model to get the highest scoring set of parameters for the final model.

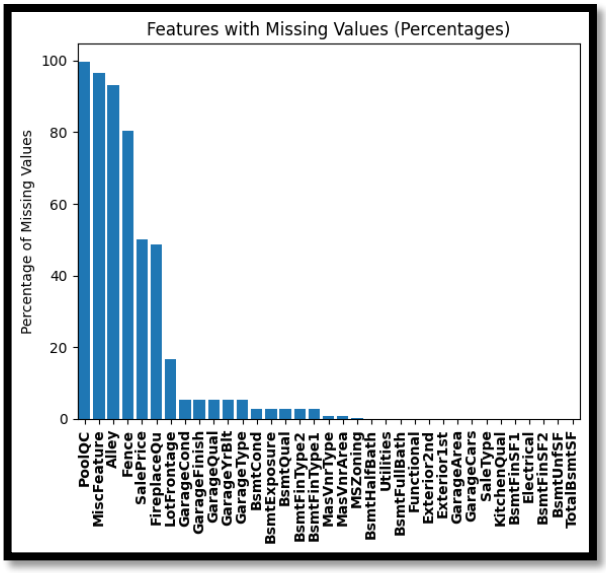
The dataset the I will be using for this model creation is comprised of 2919 different house sales with 79 different variables (This is a link to the variables and their definitions). My manager has preset the train/test split with 1459 house sales in the train, and 1459 in the test set with their sale price as 'NaN' (missing). The test set is what I am going to predict and will be judged of the RMSE score for the difference of my predictions to the actual sale price. This dataset goes beyond the classical variables people see as important as the number of bedrooms and bathrooms. The dataset incorporates variables like year build, basement square footage, and type of foundation that have a heavy effect on the average sale price in Ames, Iowa. The dataset was complied by Dean De Cock and encompasses nearly every aspect of houses within Ames, Iowa.

Data Preprocessing:

Before conducting the Gradient Boosted Model, I did some data manipulation to get my data ready for modeling. The main manipulations I made to the data pertain to missing values, feature engineering (adding column that have importance), categorical features, numerical features, feature correlation with sale price, and dealing with features that have many outlier values.

Missing Values:

In model creation dealing with missing values is crucial, especially gradient boosted model because they can create incorrect splits in the decision trees within the model. Initially I dove into the missing value for every column within the dataset. I did this to see if any columns have a substantial amount of missing value and see if I would benefit from deleting them entirely. After finding the counts of missing values I found four columns with substantial number of missing value es: Alley (Type of alley access), PoolQC (Pool Quality), Fence (Fence quality), and MiscFeature (Misc. feature not covered in other categories). All these variables had over 80% of their values missing. The graph shows the distinct proportion of their values missing compared to other columns.



Feature Engineering:

I conducted some feature engineering to craft some new variables I believed to be beneficial to the gradient boosted model. Along with capturing various aspects of houses in Ames that were not covered in the initial data set.

- **TotalSF →Total Square Footage** (calculation: TotalBsmtSF (basement sq ft) + 1stFlrSF (1st floor sq ft) + 2ndFlrSF (2nd floor sq ft)): This variable was created to get a number for total quantity of square footage for all levels of the house. This variable will help home buyers who are looking for differentiation are not worried about different levels of square footage but an overall square footage number.
- **HouseAge → Age of House** (calculation: YrSold (year house was sold) - YearBuilt (year house was built)): This variable obtains the age of the house on the day it was sold. This variable will benefit homebuyers who are looking for a newer, older, middle aged, etc. home.
- **RemodAge→ Years Since Most Recent Remodel** (calculation: YrSold – YearRemodAdd (Remodel Date): This variable grasps the number of years since a house has had any major remodeling. This can help customers wanting a new model home but do not want to pay the premium price for a recently constructed house.
- **Overall_rate_qual_cond → Combination Score of Overall Quality and Condition** (calculation: OverallQual (Overall Quality Score) + OverallCond (Overall Condition Score): This variable obtains a combination of quality and condition to give customers a second fold to look at when it comes to the general condition and materials used to build a house.
- **Total_bathrooms → Total Bathroom Throughout the House** (calculation: HalfBath*0.5 (half bathrooms) + FullBath (full bathrooms) + BsmtFullBath (full bathroom in basement) + BsmtHalfBath (half bathrooms in basement): This variable obtains a total number of bathrooms to include the basement. This gives another quantity for customers to see so they can have knowledge of all bathrooms throughout the house.
- **Combined_deck_porch_sq → Total Quantity of Sq ft That's Either a Deck or Porch** (calculation: WoodDeckSF (wood deck sq ft) + OpenPorchSF (open porch sq ft) + 3SsnPorch (3 season porch sq ft) + EnclosedPorch (enclosed porch sq ft)): This variable encompasses a total quantity for the square footage of any outdoor porch or deck. Some customers have a deep desire for outdoor seating, and this variable helps them have knowledge of the available space.
- **Quantity_premium_add_ons → Quantity of House Add Ons Deemed Premium** (Calculation adds 1 if any of these variables have values greater than 0: WoodDeckSF, OpenPorchSF, PoolArea (Pool sq ft), 3SsnPorch, EnclosedPorch, Fireplaces (number of fireplaces)): This variable acknowledges if any of these premiums add of feature are present for that house. These features are often in heavy desire by potential home buyers, and this variable can help encompass all the premium add on features into one overarching variable.

Categorical/Numerical Features:

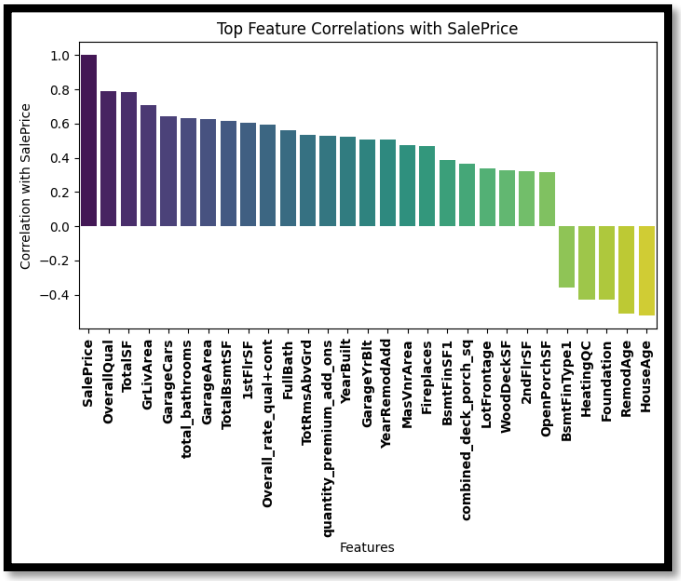
Categorical variables are variables that have a set number of groups. Some examples from the Ames housing dataset: MSZoning (Zoning Classification), Street (Type of Road Access), and RoofStyle (Type of Roof). Numerical Columns variables are variables that are quantified usually by a measurement.

The main task I had to deal with was columns that still have missing values and representing categorical columns as numerical values.

First, I broke the categorical and numerical columns into two different lists. The categorical column list had 39 variables, and the numerical list has 39 variables with the additions of manually input variables. I left ID (house sale identification) and all year variables alone because they had no missing values. For the categorical variables with missing values, I replaced it with the mode of their corresponding column. I altered the missing values in this way to help preserve the distribution of the data. On the other hand, I fill the missing numerical column values with the mean value of the corresponding column. I alter the missing values in this way because it retains the pattern within those columns and the numerical columns did not have any substantial missing value throughout.

Lastly, I had to represent each of the categorical columns with numerical values. I did this through manual imputation. I found the total number of unique values for each categorical column and associated a number starting at 1 to each of the unique values. This sets the categorical columns up perfectly for the model creation because they have an identical representation of their original value just in a numerical sense.

Correlation Analysis:



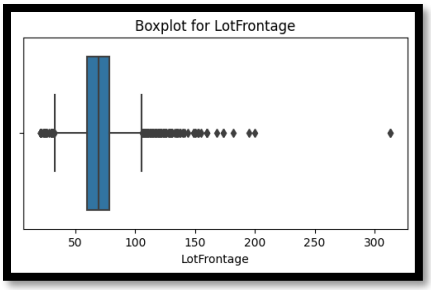
Correlation between the explanatory variables and the target variable (SalePrice) help identify strength and direction of the relationship between an isolated feature and the goal for predicting the sale price of a house. Within a gradient boosted model high correlation is crucial because relationships between variables are weighted heavily with this model. Also, having high correlation between the variables with the target variable can help identify the parameters and features to use within the gradient boosted model.

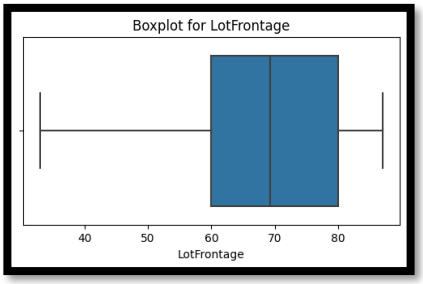
For this model I have generated all the correlations between the explanatory variables and the sale price variable the I am trying to predict. After generating the correlation some variables with high positive correlation: OverallQual, GrLivArea (above grade living are sq ft), GarageCars (Size of Garage in Car Capacity), and Total_bathrooms. Some variables with high negative correlation: HouseAge, RemodAge, Foundation (Type of foundation), and HeatingQC (Heating quality and condition).

To go along with this, I decided to create a cutoff of pos or neg 0.30 for variables to keep as anything under this level is deemed to have low correlation with the target variable. The graph to the right showcases the variables with strong correlation with the sale price of a house.

Outlier Values Alteration:

A crucial step in the data process for me was detecting numerical features that are greatly skewed by outliers. Since I am dealing with a lot of variables and values, generalizing data while training the model will help make the model more equipped for unforeseen data that will arise when customers fill out their desired features of their dream house. On top of that, reducing the effect outlier values have on the model reduces the potential for bias to be incorporated into the model because the goal of my model is to find the key patterns in the data to get true relationship between feature of a house and the sale price.





My procedure to alter outlier values in the data was utilizing the different quartiles within the values of certain features. First, I generated a boxplot for each of the numerical columns, and observed their distribution to generate a list of values that will benefit from me altering their distribution. This list consisted of 21 numerical columns. This distinction was made by looking at the boxplot and seeing if there was most of its values depicted as outliers. This distinction means that the mean of values for the feature is skewed by some customers wanting a very far-fetched desire for that feature or underestimating the true value of that feature. After getting these features, I calculated the first quantile, third quantile, interquartile range, and outlier bounds (1.5 times interquartile range for upper and lower bound). From there I altered any outlier beyond the upper or lower bound to take on that value. This makes the distribution for these variables generalized around the key summary statistics that follow the usual customer desires to have when purchasing a home. To the right I showcased one of the variable alterations.

Gradient Boosted Regression Model Procedure:

After conducting all my desired data preprocessing the dataset the model was ready to be constructed. The model I am using to generate predictions of home sale prices is a Gradient Boosted Regressor because it can handle non-linear relationships, structures the model around the most important features, and handles unforeseen data efficiently.

Gradient Boosting Regressor:

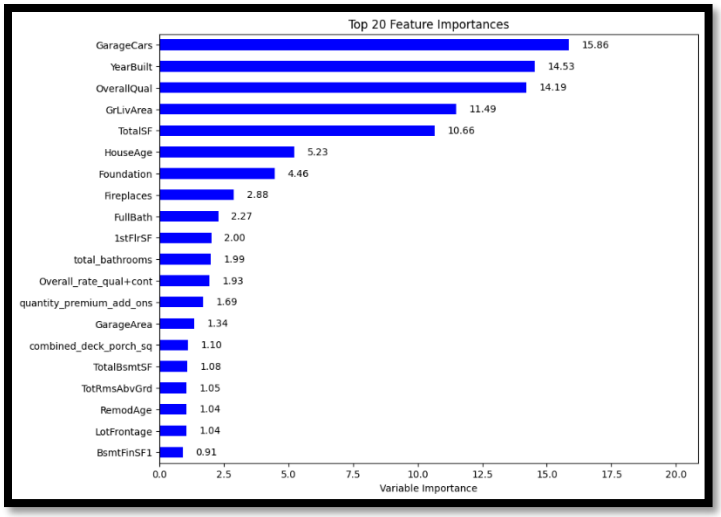
My procedure for building the model starts with splitting the data back into the train and test dataset that you provided for me. After that I made my data frames of the explanatory variables and the predictor variable (sale price). Coming from that, I conducted two alterations to the train dataset: log transformed the predictor variable and standard scaled the explanatory variables. These two alterations guaranteed that I was working with data that was apple to apples with each other and reduced the effect of potential skewness still being present.

After shaping the dataset, I generated my model using a randomized search to generate the best hyperparameters for the boosted model. This random search was able to lockdown on the highest scoring parameters from the lists I preset using the parameters with the highest mean square error, this test is very similar to the test you are going to use with the test set data. After getting the parameters these were plugged into my gradient boosting model and fit that model against the training data set. After running the model, I generated the 20 most important features. Then, I ran a prediction with the model to generate predictive sale prices on the given test data set to predict the sale price for each of those House IDs. After I got the predictions of the house sale prices, I figured them into a data frame with the respective house ID and ran the test you gave me to see my RMSE score.

In this next section I will go over the most important features within my model, the strength of my model within the training set given, and breakdown the success of my model to predict the sale prices of homes within the test data set.

Results/Findings/Strength of Model:

Feature Importance:



The graph to the right showcases the top 20 features within the house price model represented as a percentage. These values showcase the contribution each feature makes towards the predictions of the sale price for homes in Ames. For example, YearBuilt has a percentage of 14.53%. This means that the year a house was built has a predictive power of 14.53% within the overall model prediction of home sale prices. From looking at this graph a key observation I see is yearly, square footage/area, and overall quality statistics generate a lot of predictive power for my model. This notion can help me dictate the key features I want to look at when customers make their dream house portfolio. For example, if a customer demands a 2,000 sq ft house built in the 1990s. I know my model will be heavily influenced by this determination, but at the same time present great accuracy for the customer since my model will use these features as guiding forces to offer a prediction price for them.

R-squared Value (For the Train Data):

R-Squared on Train Data: 0.9552

The r-squared value from the training dataset helps provide strength and performance outlook for my model.

The r-squared value represents proportion of variance explained in target variable by the feature variables. For my model I was able to garner a 0.9552 r-squared value, or I was able to explain 95.52% of the variance within the sale price given the modeling prediction power of each feature. This showcases the strength in my model to determine home sale prices. It can explain nearly all the variance potential to arise for all the features present in the dataset. This gives me confidence in my model to perform well on unseen data, and help customers get an accurate price for their desired home no matter the features list they desire.

RMSE Value (From my Predictions for the Test Data):

RMSE of Test Data: 0.13730

Lastly, the RMSE is a difference calculation between the actual values and predicted values (calculated by taking square root of mean square differences of the two values). A value of 0.13730 showcases that on average my model prediction for a house sale price given the feature within the data set, 0.13730 unites of error are present. Having a narrow margin of error present in my model build strength. This score on the test set that was initially unforeseen to my model, presents a strong outlook that it will accurately predict sale prices for future homes given a set criterion of features.

Conclusion/Final Statement on Model:

Using a data set of over 80 variables and over 1400 house sales, I am pleased to showcase my model as having great predictive power when generating house sale prices for customers. My model can capture key patterns and relationships within unseen housing data. This distinction gives customers confidence to use our house pricing predictions because we can take all their desires incorporated in their dream home and give them an accurate price, they can utilize to achieve their goal. This initial model builds the foundation for us to adapt to constant change within the housing market and create a drive for improvement toward a one hundred percent accurate model to predict sale prices of houses in Ames, Iowa.