Kaggle Competition: Predicting House Prices

Soorya Paturi (sopaturi@gmail.com)

***Abstract*— A house has many characteristics that describe it from the type of material that is used on the exterior of the house to the number of rooms in the house. Typically, only a few key characteristics are used to describe a house instead of all of the descriptors when describing a house. A common descriptor of a house is home price because it takes into account the value of many other characteristics. The goal of this project is to predict home prices in Ames, Iowa. Currently, there is not a published deterministic relationship between home price and the characteristics of a home. However, the objective is to find a relationship between house characteristics and home price so that home price can be predicted.**

I. INTRODUCTION

Predicting home price is important to the buyer, seller, as well as an assessor of the home. Once home price can be predicted, recommendations can be made to the home seller on what modifications to the home will lead to the greatest increase in price. An assessor that is determining the value of the home can use the results of this prediction model to verify that their valuation is accurate. Finally, a home buyer would find it useful to predict home price so that they can verify they are buying the home for a fair value compared to the other homes in Ames, Iowa. The best performing model can be used to predict home prices in any suburb given similar input features.

Home price prediction is not deterministic because a house can have many different characteristics. In addition, a seller and a buyer can agree on a sale price that is not characteristic of similar homes. However, the objective is to find a relationship between home price and characteristics of the house, so that the value of home price can be predicted.

The client interested in the results of this report is any home buyer, seller, or assessor. The client should care about home price prediction for these houses because unknown house prices given house characteristics can be identified. With a home price predictor, a home buyer or seller can verify that the price listed by the assessor is accurate and change the house so that the price increases.

The outline of this report and the final report is summarized here with Section II introducing the data set. In Section III, the data wrangling of the data set is summarized by explaining how the data set was obtained, cleaned, and wrangled. In Section IV, exploratory data analysis of the data is summarized by a field by field analysis of home attributes that affect home price. In Section V, the prediction task is described. In addition, the last section, Section VI, conclusions and next steps are discussed.

II. DATA SET

The dataset was downloaded from https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data [1]. This dataset is part of a Kaggle competition. Each row in this dataset represents a house in Ames, Iowa that has 79 features describing the house. The goal is to predict the sale price feature of the test set. The models will be evaluated using the RMSE between the log of the predicted house sale price and the actual house sale price. The log scale is used so that more expensive homes do not have a larger weight on the error compared to the less expensive homes.

The dataset was not sampled because the computation time did not need to be decreased for the prediction task. The size of the training set and test set were 1460 rows and 1459 rows respectively. In addition, the prediction task will be completed on a cluster in the cloud that uses two cores. The Koalas library was used when interacting with big data because it implements the Pandas DataFrame API on top of Apache Spark. Pandas is typically used for single node Dataframe implementation in Python while Spark is used for big data processing. With Koalas, Pandas syntax can be used while

processing dataframes with Apache Spark. More description of the prediction task will be handled later in the report.

The prediction task is to use house attributes and individual reviews to predict house price. In Figure 1, is a distribution of the home sale prices in Ames, Iowa. The mean value of the home sale price distribution is 180921.19. Sale price is not normally distributed as the value of the skewness for the sale price distribution is 1.88 which is above 0. This value for skewness means the sale price is skewed to the right. A distribution that is skewed to the right means the right tail is larger than the left tail. There are quite a few homes that worth more than 336,625$, which is 2 standard deviations more than the mean. A kurtosis value of 6.5 is larger than 3 which means the distribution is heavy tailed meaning there are more outliers compared to the number of outliers in a normal distribution.

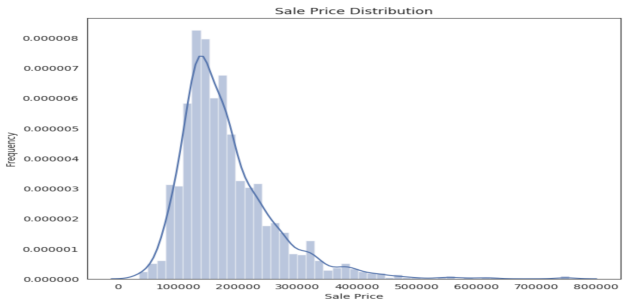

Fig. 1. Histogram of House Sale Price

A selected number of fields, six to be exact, that may cause confusion, are described below so that interpretation of the data set above is easier full description of all fields is shown in the link below.

-BldgType: Type of dwelling

-YearBuilt: Original construction date

-SalePrice: the property's sale price in dollars. This is the target variable that you're trying to predict.

-FullBath: Full bathrooms above grade

-Bedroom: Number of bedrooms above basement level

-2ndFlrSF: Second floor square feet

-1stFlrSF: First Floor square feet

Description of all the fields in the data set can be found at https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data [2].

III. Data Wrangling

Because home sale price is a numerical feature, it does not need to be converted. The id feature was dropped because it does not describe the house and is unique for every row in the dataset.

The columns with missing values include:['PoolQC', 'MiscFeature', 'Alley', 'Fence', 'FireplaceQu', 'LotFrontage', 'GarageYrBlt', 'GarageFinish', 'GarageQual', 'GarageCond', 'GarageType', 'BsmtCond', 'BsmtExposure', 'BsmtQual', 'BsmtFinType2', 'BsmtFinType1', 'MasVnrType', 'MasVnrArea', 'MSZoning', 'Utilities', 'BsmtFullBath', 'BsmtHalfBath', 'Functional', 'Exterior1st', 'Exterior2nd', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Electrical', 'KitchenQual', 'GarageCars', 'GarageArea', 'SaleType']. None of the columns with missing values were dropped nor were any of the rows with missing values dropped. In Figure 2, a bar plot of the percent of values that are missing in features with missing values was created to show a few features contain most of the missing values in the dataset.

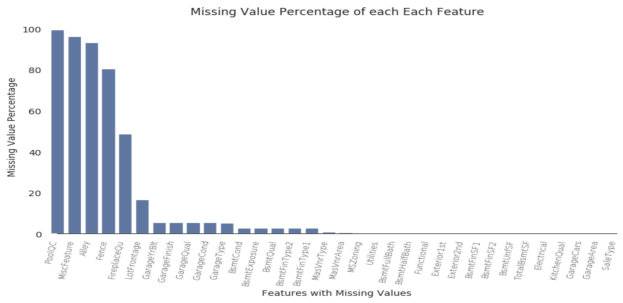

Fig. 2. Bar plot of features with missing values and their percent of values that are missing

If the feature was a numerical feature, the missing value was replaced with the mean value of the feature. If the feature was a categorical feature, the missing value was replaced with the string 'None'. There were several features that were represented as integers in the dataset but should have been represented as strings in the dataset. For example, the MSSubClass feature is a code

that represents the type of dwelling. Each code is not a measure of a certain characteristic as a numerical feature should be, rather it is a label and should be converted to a string format and treated as a categorical variable. MSSubClass along with MoSold, and YrSold were converted to categorical variables.

Categorical variables were hot one encoded so that a machine learning model can take in these features. A machine learning model can only take in numerical features as inputs. Each unique value in a column gets a new column in which if the row contains the unique value, a value of 1 is used else a value of 0 is used. Hot one encoding increased the number of features from 87 to 349. Several numerical features could have have been converted to strings and subsequently hot one encoded but the curse of dimensionality prevented these features from getting converted. For example, the features of the year the house was bought or the year the house was remodeled are very high cardinality variables. So if they were hot one encoded, the number of features would have been greater than six hundred. Adding so many features for such a few number of original features is not helpful for a machine learning model that will have trouble accurately fitting features when there are too many.

A. Feature Engineering

Several new features were created such as the number of years since a remodel, the total home quality, the total square footage, the total number of baths, and the total porch square footage. The year since remodel feature was created by subtracting the year of remodel from the the year the house was sold. The total_home_quality feature was created from adding the overall quality and overall condition feature. The total square feet feature was created from adding the total basement square feet, the first floor square feet, and the second floor square feet. The total bath feature was created from summing the number of full baths and 0.5*(number of half baths). In addition the binary features has_pool, has2ndfloor, hasbsmt, and hasfireplace were created. If the value in any of these features were greater than zero, a value of one was used to denote that the characteristic in the house exists. Feature engineering creates more features for the machine learning models to account for and could possibly create features that contribute to home price prediction.

IV. Exploratory Data Analysis

The distribution of values in a home's attributes and how it affects home sale price is explored in this section. In the exploratory data analysis section, scatter plots were constructed for all of the numerical features and their relationship with the variable being predicted. Bar plots were constructed for categorical features and their relationship with the variable being predicted. In the following subsections five important features that are typically used to describe a home are graphed in relation to home price.

A. Total Square Feet

Figure 3 is a graph of house sale price vs total square feet. The house sale price is trending upwards as total square feet increases. The slope of the polynomial fit is 75.63 which means an additional square foot costs on average 75.63 dollars assuming all of the other features are constant. Since the house price vs. total square feet is a nonuniform distribution, total square feet could be an important predictor in the machine learning phase.

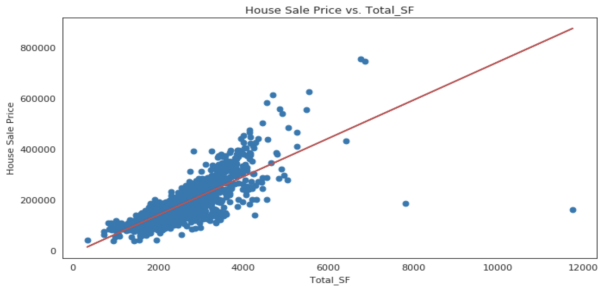

Fig. 3. Scatter plot of the total square feet vs. home price sale

B. Total Number of Bathrooms

Figure 4 is a graph of house sale price vs total bathrooms. The house sale price is trending upwards as total bathrooms increases. The slope of the polynomial fit is 63899.15 which means an additional bathroom costs on average 63899.15 dollars. Since the house price vs. total bathrooms is a nonuniform distribution, total bathrooms could

be an important predictor in the machine learning phase.

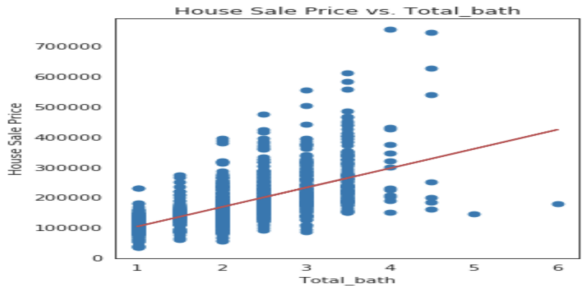

Fig. 4. Scatter plot of total number of bathrooms vs. the home sale price

C. Total Number of Bedrooms

Figure 5 includes a graph of house sale price vs number of bedrooms. The house sale price is trending upwards as number of bedrooms increases. The slope of the polynomial fit is 16381.02. Since the house price vs. number of bedrooms is a nonuniform distribution, year built could be an important predictor in the machine learning phase.

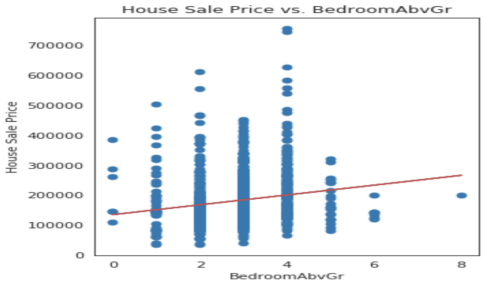

Fig. 5. Scatter plot of the number of bedrooms vs. the home sale price

D. Building Type

Figure 6 is a graph of house sale price vs bulding type. Since the house price vs. building type is a nonuniform distribution, building type could be an important predictor in the machine learning phase.

E. Year Built

Figure 7 includes a graph of house sale price vs year built. The house sale price is trending upwards as year built increases. The slope of the polynomial fit is 1375.37. Since the house price vs. year built

Fig. 6. Bar plot of the the number of building type vs the house sale price

is a nonuniform distribution, year built could be an important predictor in the machine learning phase.

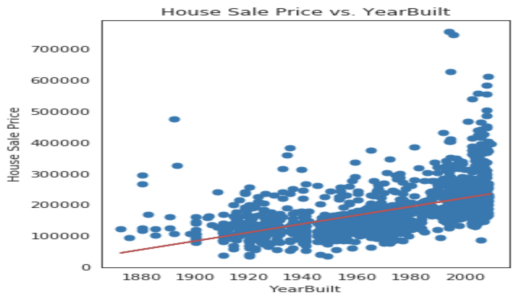

Fig. 7. Scatter plot of year built vs. the home sale price

V. Prediction Task

The dataset for this section was obtained from the exploratory data analysis section. The goal of the prediction task is to predict home price from the original 79 features. In addition, features of a home that can be changed to increase home sale price need to be identified. All of the fields will be inputted into the regression models so that no important fields are mistakenly left out. A few additional wrangling steps were performed before the machine models were implemented and compared.

The numeric features in the training and test sets were collected so that they could be transformed to normal distributions. A normal distribution helps a machine learning model evaluate samples because there will be less outliers for each feature that the model will not have to overfit for. If the value of skew was above 0.5, the boxcox1p transformation

was applied. The transformation involves taking the input value to the exponent of the optimal lambda, subtracting one, and then dividing by lambda. This transformation occurs when lambda is not 0 and when lambda is equal to 0, the log of the value of the feature are found. The house sale price was transformed by a log transform because the Kaggle competition is scored based on RMSE between log of predicted. Finally the dataset was split into a train and test dataset.

The linear regression models were run using pyspark dataframes while the ensembles methods were run using pandas dataframes. Pyspark dataframes are not allowed as inputs for many of the ensemble methods used. One hot encoding was applied to the pyspark dataframe in addition to the encoding already completed for the pandas dataframes. In pyspark, a StringIndexer is used to assign a numerical value to each unique value of a feature. Then the one-hot encoder converts each sample into a binary vector containing information about which values the sample has. The binary vector is appended to the end of a each row. A binary vector contains the index value from the string indexer.

All of the features after the wrangling step were used as inputs to the machine learning models in order to prevent important features from being left out. Regularization was relied upon to do feature selection instead. The machine learning models used to predict home sale price include lasso regression, ridge regression, elastic net regression, random forest , gradient boosted regression, lightgbm, xgboost, stacked ensemble, and a weighted average blend. The goal of the prediction task, the final step in the report, is to identify a regressor that achieves the best RMSE of predicted home prices.

A. Regression Metrics

The following metrics were used to compare the performance of the regression models against each other.

Root Mean Square Error: The square root of the average of the squared differences between predicted and actual values.

$$RootMeanSquareError = \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$$

where
$\hat{y}_i$ is the predicted value,
y_i is the actual value,
N is the number of samples

B. Hyperparameter Selection

The hyperopt package was used to find the optimal hyperparameters in the linear regression models. The optimal hyperparameters in a space are found by implementing Bayesian optimization. By finding an optimal set of hyperparameters, before the model is fitted, only one set of hyperparameters need to be tested, the optimal ones. Grid search is a much more computationally expensive tuning method as it requires the model to check every hyperparameter combination.

Bayesian optimization can be explained in four steps. First, the process is initialized by sampling the hyperparameter space and getting these observations. Next a Gaussian process is fitted to the observations from step 1. The mean from the Gaussian process is used as the function most likely to model the black box function in step 2. The maximal location of the acquisition function is used to figure out where to sample next in the hyperparameter space in step 3. New observations are added with newly sampled hyperparameter points in step 4. This process, (steps 2-4), is repeated until a maximum number of iterations is met. By iterating through the method explicated above, Bayesian optimization effectively searches the hyperparameter space while homing in on the global optima.

The space of regularization hyperparameters for linear regression were inputted into hyperopt so that the optimal set could be found. The optimal set of hyperparameters for lasso regression, ridge regression, and elastic net regression were found. The space searched for the L1 penalty was a choice between the values 0.001 and 0.008 where the values are incremented by 0.001. The space searched for the L2 penalty was a choice between the values of 14.5 and 15.4 where the values are incremented

by 0.1. The optimal hyperparameter set for the elastic net model cancelled out the ridge regression term which means lasso regression performed better than ridge regression. Now that the optimal hyperparameters from Bayesian optimization were found, we can build small hyperparameter grids with only one set of hyperparameters that can be searched quickly, which will lower training times.

C. Models

This subsection describes all of the regression models used to predict home price.

A.) Linear Regression: Linear regression predicts the response variable, in this case home sale price, as the dot product of p distinct predictors and their corresponding coefficients. An intercept term is added to the dot product so the response variable has a non zero value when all of the predictor values are zero. In this case home sale price should have an intercept term because home price must be greater than 0. The multiple linear regression model takes the form $Y = 0 + 1X1 + 2X2 + \cdots + pXp + ...$ j can be interpreted as the increase in the response variable with one unit increase of Xj assuming all other slope coefficients are set to zero.

Three different estimators were tested: ridge regression, lasso regression, and elastic net regression. These three types of regression include different regularization types. The L1 penalty (lasso) is a regularization parameter multiplied by the sum of the absolute value of the regression coefficients while the L2 penalty (ridge) involves summing the square of the regression coefficients and multiplying the result with a regularization parameter. The purpose of regularization is to prevent overfitting by adding penalty terms that cause the loss function to have less predictor terms.

The elastic net parameter is multiplied by the L1 penalty and one minus the elastic net parameter is multiplied by the L2 penalty. In order to implement ridge regression, the elastic net parameter term must be set to zero which cancels out the L1 penalty and leaves the L2 penalty. To implement lasso regression, the elastic parameter must be 1 to cancel out the L2 term leaving only the L1 term. To implement elastic net, the elastic net parameter as well as the L1 term must be greater than zero and less than one.

B.) Random Forest: The random forest classifier is an ensemble method that uses many decision trees that have had bootstrap aggregating or bagging done on the samples. Bagging means that samples are chosen at random with replacement to reduce variance in any single tree and the results are averaged. The prediction in an individual tree is the average value of the response variable for the samples of a leaf node. The prediction of the bagged regression decision tree is the average of the predictions for each individual decision tree.

Random forests in addition take a subset of the predictors when constructing a tree so that each tree is more different from one another. The combination of the trees made from bagged samples and subsetted predictors is the random forest result.

C.) Gradient Boosting Classifier: A gradient boosted classifier involves combining a large number of trees with bagging and boosting. Each subsequent tree that is constructed is done so based on the least pure splits created by the previous tree. The trees are added back to the previous tree so that the tree can learn slowly and improve RMSE. An advantage to gradient boosted trees is that compared to a decision tree that may overfit to one set of the data, gradient boosted trees learn by focusing on the worst split in a tree and improve over time. The disadvantages of gradient boosted trees is that they may overfit compared to random forests but not in this case.

D.) LightGBM and XGBoost: The lightgbm is a type of gradient boosting regressor in that new trees are learning from the worst splits in the previous tree. One key difference is that lightgbm regressor grow leaf wise as opposed to level wise in gradient boosting regressors. Leaf wise growth constructs a new tree from a new leaf and keeps creating new trees from the leaf nodes of the previously created trees. Level wise growth must have all the leaves on a level of a tree split into new nodes between the resulting nodes can be split.

Both lighgbm and xgboost use histogram based bins to split the data of an attribute into bins. The decision tree only needs to check the number of bins for each attribute as opposed to all of the possible splits.

E.) Stacked Ensemble Model

Stacking enables a method of ensembling mul-

tiple regression models.

The training data is split into K-folds and a base model is fitted on the K-1 parts. A prediction is made for the Kth part and repeated for each part of the train set. The base model is then fitted on the whole train dataset. The previous steps are repeated for all of the base models. The predictions of the base models from the k-fold cross validation of the train set are used as features in the meta-regressor model. This second level model is used to make final predictions on the test set which has added features from the prediction made from the fitted base model predictions on the test set.

F.) Weighted Average Model

A weighted average model simply receives the best performing models as inputs and takes a fraction of the prediction for each model and sums them. The fraction can be thought of a weight and can be adjusted to maximize RMSE. A weighted average model can capture the predictions of multiple models and account for the shortcomings in each individual model by combining the results.

VI. Evaluation and Results

The original dataset downloaded from Kaggle was already split into a train and test set. The train set contained 50% of the total data while the test set contained the other 50% of the data. Cross validation was used with ten folds when training the ML models so that the fit was representative of the whole dataset and not just the data from the train-test split. The error metric used during cross validation was RMSE.

A.) Linear Regression Model

The ridge regression performed the worst out of the regression models because it achieved a test RMSE of 0.5076 compared to 0.1276 for elastic net regression and 0.1256 for lasso regression. The optimal regularization hyperparameter found for ridge regression was 14.5. The optimal regularization hyperparameter for lasso regression was found to be 0.008. The optimal regularization hyperparameter and elastic net was found to be 0.007 and 1 for elastic net regularization. The lasso regression model identified the features that most influence home sale price so that it will go up if the values of these features are increased. The top ten features of the fitted lasso model were `Total_SF`, `GrLivArea`, `Total_Home_Quality`, `'NeighborhoodclassVec_Crawfor'`, `'YrBltAndRemod'`, `'NeighborhoodclassVec_StoneBr'`, `'KitchenQualclassVec_Ex'`, `'BsmtQualclassVec_Ex'`, `NeighborhoodclassVec_NoRidge`, and `Exterior1stclassVec_BrkFace`.

B.) Random Forest Model

The random forest performed the worst out of the ensemble tree methods because it achieved a test RMSE of 0.1734 which is greater than all of the other ensemble methods. Standard hyperparameters were chosen for the random forest. The square root of the number of features was used on each decision tree in the random forest. In addition, 3000 estimators were used in the random forest and each had a max depth of 5.

C.) Gradient Boosted Tree

The gradient boosted tree achieved a test set RMSE of 0.1287, the fifth best test RMSE out of the individual models. The number of estimators used was 3000, the learning rate was 0.05, the max depth for each tree was 4, the number of features used for each tree was the square root of all of the features, the minimum samples in a leaf was 15, and the minimum number of samples needed to split was 10.

D.) LightGBM and XGBoost

The xgboost model achieved a test RMSE of 0.1273, the third best test RMSE out of the individual models. The learning rate for the model was 0.01, the number of estimators used was 3460, the max depth for each tree was 3, and 70% of the features were used to construct each tree.

The lightGBM model performed the best out of all of the individual tree ensemble methods with a test RMSE of 0.1229. The number of leaves in the lightgbm was 4, the learning rate was 0.01, the number of estimators was 5000, 20% of the features were used to construct each tree, and 75% of the data was used for each estimator constructed. The ten features with the highest feature importance were `LotArea`, `GrLivArea`, `Total_SF`, `1stFlrSF`, `GarageArea`, `TotalBsmtSF`, `Total_porch_sf`, `LotFrontage`, `YrBltAndRemod`, `YearBuilt`.

E.) Stacked Ensemble Model

The stacked ensemble model achieved a test RMSE of 0.1221. The base models in the the stacked ensemble model were the lasso regression and lightgbm models because they were the two best performing models run so far and this combination resulted in the best test RMSE compared to other combinations. The secondary model or meta-regressor used was the lightgbm because it was the best performing individual model.

F.) Weighted Average Model

The weighted average model performed the best out of all of the models because it achieved the best test RMSE of 0.1194. The weighted average model used the best performing models which included the elastic net model, lasso regression model, gradient boosted regression, xgboost, lightgbm, and a stacked ensemble. The weights chosen were uniform which means each model's predictions was multiplied by 1/6. The purpose of using weights allows all of the models to be expressed and some of the models may compensate for the error in other models.

TABLE I

MACHINE LEARNING CLASSIFICATION METRICS

Model	Training Validation RMSE	Test RMSE
Lasso Regression	0.1251	0.1256
Ridge Regression	0.2808	0.5067
Elastic Net Regression	0.1284	0.1276
Random Forest	0.1682	0.1734
Gradient Boosted Regressor	0.1208	0.1287
XGBoost	0.1207	0.1273
LightGBM	0.1166	0.1229
Stacked Ensemble Model	0.1195	0.1221
Weighted Average Model	0.0740	0.1194

VII. CONCLUSION AND FUTURE WORK

The best performing machine learning regressor was the weighted average model because it achieved a test RMSE of 0.1194, the lowest test RMSE of all of the models. This RMSE score garnered a top 13% of all submissions to Kaggle for this competition. The most important features for the best performing ensemble tree (lightgbm) in the stacked model were `LotArea`, `GrLivArea`, `Total_SF`, `1stFlrSF`, `GarageArea`, `TotalBsmtSF`, `Total_porch_sf`, `LotFrontage`, `YrBltAndRemod`, `YearBuilt`.

Based on the top features of the lightgbm model, it is apparent that house size is important as the top eight most important features were lot area in square footage, above ground living in square footage, first floor square footage, garage area in square feet, total basement square feet, total porch square feet, and linear feet of street connected to the property. It is safe to say that the larger the property or house will lead to a higher house sale price. In addition, the later the house was built or remodeled, the higher the house sale price will be according the the lightgbm model.

Feature importance does not tell us how the feature can be can be changed to increase home sale price. A home seller may want to change their house so that its price is increased. Based on the largest positive coefficients in the lasso regression model can be used to determine which features of a house can be changed to increase house sale price the most. The more recent a remodel has been done to a house will increase the home sale price. In addition, if a remodel has been done to the kitchen or in the basement it will increase the house price because kitchens and basements with an excellent quality as a feature greatly contributed to home sale price.

A weighted average model can be used by realtors, assessors, or home assessors to verify the home sale price is a fair price compared to similar houses before they are involved in a house sale.

In the future, in order to improve the RMSE for the ensemble tree based methods, a larger hyperparameter space can be searched. A standard set of hyperparameters was inputted to decrease training time for the ensemble tree methods.

REFERENCES

[1] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview/description

[2] https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

[3] https://hyperopt.github.io/hyperopt/

[4] https://textblob.readthedocs.io/en/dev/

[5] https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/