Progress Report Home Price Prediction in Ames, Iowa
Team 84

Reena Sahai (RSahai6); Jonathan Gerszberg (JGerszberg3); Vignesh Thavamani Thenmozhi (VThenmozhi3); Siddharth Gudiduri (SGudiduri3)

Overview of the Project:

Background – The United States, as of 2020, has a housing shortage of over 3.8 million dwellings and trending upwards.[1] This critical shortage can have a long-term detrimental impact on displaced persons and the greater community. To fix this issue, a variety of industries such as building contractors, realtors, sellers, and public policy administrators all must come together to not only provide homes but also homes that buyers desire and can reasonably maintain.

Real estate is unique when compared to other industries. Houses are durable and last multiple decades. It is a necessity, and at the same time, it offers stable growth; hence, it is one of the best long-term investment options. Every house is unique in terms of style, design, features, location, and construction type, making the pricing highly complex. The real estate business can be very lucrative if you understand the key factors that drive people to buy or sell homes. It has been challenging to predict the price of a home because there are a lot of factors contributing to its value.[2] In addition to the physical and geographical aspects of a home, there is a temporal component to home pricing too. Since the pandemic, US home prices have soared compared to income growth, due to multiple factors including inflation, interest rates hikes, supply chain issues, labor constraints, and the economy's overall health. All these factors play an important part in consumer confidence, which impacts the willingness of someone to invest in real estate. Going hand in hand with when to buy/sell a home, is determining what a consumer wants in their home. What features of a home are a necessity versus a nice to have and a do not want to have? Essentially, when considering buying/selling real estate, it is critical to understand what aspects of a home a buyer is willing to pay for and what a buyer sees as a liability. This suggests that the ability to create a model that can predict home prices based on various factors about a home could be very beneficial.

Hypothesis – We expect to build a model that has a minimum RMSE < 6000 and $\boldsymbol{R^2}$ > 80% when predicting against the test dataset. Iterative methods i.e., using feature engineering, performing dimension reduction for model building, and testing various models will aid in finding a few good models. We hope to find a reduced subset of attributes that can predict the sales price.

Domain focus for this project has been on real estate and the results of our analysis will benefit realtors or home buyers in looking for their perfect house or making a profit(realtors). This can also help contractors decide what to include in new developments. Some of the potential benefits include

1. Maximizing profit with minimal investment for real estate promoters
2. Easy risk assessment for the lenders
3. Buyers can be assured that they are paying a fair price
4. A promoter can decide what feature to offer in a particular neighborhood

General Approach – The purpose of this project is to create a model that elucidates how and what physical and geographical aspects of one's home impact the sales price. Traditionally, it is understood that having more beds and baths increases the price of homes. Similarly, having a home in a low crime area and better schools tends to increase home prices.[3, 4]. A study in 2016 by Ozgurs examined a set of twelve factors to predict home prices in Indiana. This study ended up showing the Homeowner's Association Fee (HOA) is the best linear predictor of sales price, demonstrated by checking the linearity assumptions. They show in this paper that even with twelve factors, price predictions are quite difficult, especially with many outliers in the data. The novelty of this project is that this project examines a significantly larger set of features than what is currently published. This study also attempts to illustrate how disparate features, physical and environmental features, could create a model that is more robust and better at understanding home purchasing behaviors.

Planned Approach:

The approach that we took for this project is summarized in Figure 1

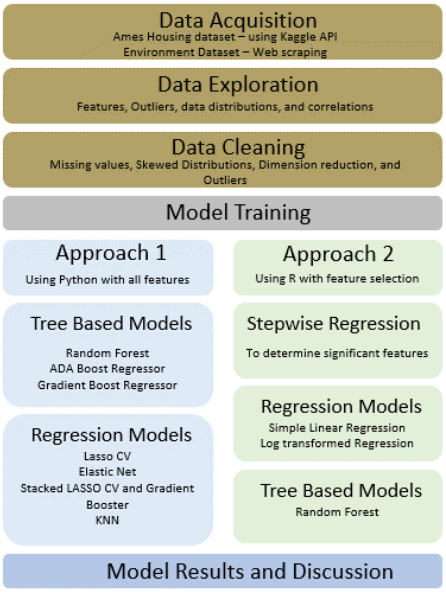


Figure 1: Project Approach

Overview of Data:

We downloaded the Ames housing dataset describing the sale of individual residential property in Ames, Iowa compiled by Dean De cook from Kaggle, using Kaggle APIs. The dataset was already divided equally into train and test dataset with 81 variables and 1460 records for each category. Out of those, 50 variables were of categorical type and 31 were numeric type.

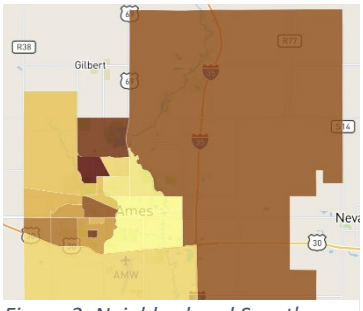


Figure 2: Neighborhood Scout's Neighborhood Break Down for Ames, IA

The Ames housing dataset only includes physical aspects of homes, hence we scraped environment data with 6 additional features from *Neighborhood Scout* using web scraping pertaining to specific

neighborhoods. The factors that were scraped from *Neighborhood Scout* are environmental in nature like the level of crime, quality of the school district, and a variety of other attributes for neighborhoods in Ames, IA. The missing values were filled with 11, indicating the neighborhood did not make the top 10 for that feature.

The links for the dataset are provided in Appendix A.

The next step was to merge both these datasets to create a master dataset for our exploratory analysis and model training. Unfortunately, the breakdown of neighborhoods and their names were different, so we created a reference table to join them manually. Surprisingly, only one neighborhood from the Ames Housing dataset did not cleanly map over. To resolve this, the Ames Housing dataset neighborhood was assigned to the neighborhood, that seemed to contain most of the residential housing. Figure 2 shows the breakdown of neighborhoods by color, there are 18 in total. The Ames housing dataset included 25 neighborhoods.

After combining the Ames housing dataset with the environmental variable, we got the master dataset with 87 variables, and 1460 records in the training dataset.

Data Exploration:

Since we wanted to build an optimal model that can be used for sellers, buyers, realtors etc., we wanted to explore the dataset with all features to get an understanding of their relative importance to the dependent variable, sale price, as well as how they relate to each other. Hence, it is important to do the comprehensive exploratory data analysis to understand the linear relationships among the variables and to get important insights from the data. Any means of reducing the number of variables required for modeling would be quite helpful, considering that there are over 80 variables to consider.

In the Exploratory analysis, we wanted to get the below mentioned information regarding dataset:

1. Number and types of features
2. Missing Values
3. Distribution of numerical and categorical
4. Outliers
5. Relationship between predictors and target features

For the initial exploratory data analysis, we decided to use an open-source module in pandas called "Pandas Profiling" which does a great job for a first level Exploratory Data Analysis. The output results in an interactive HTML report with the statistics like unique, missing values, descriptive and quantile statistics like mean, median, Q1, Q3, most frequent values, histogram and correlations based on the datatypes.

We looked at the frequency distribution of all the variable and the distribution for some of the numerical columns like Lot Frontage, Lot Area, LowQualFinSF etc. showed the presence of outliers and could be standardized or scaled accordingly. The distribution charts are shown below in appendix E.

When we looked at the target variable (sale price) distribution plot as shown in the figure below, it was clear that the distribution is right skewed with minimum sale price as $34,900, maximum sale price as

$755,000, mean as $180,921 and median value as $163,000. To reduce the effect of outliers, heteroscedasticity and to get more normal distribution, we decided to log transform the target variable and observed much more normal distribution as shown below in the plots 3a and 3b.

Figure 3a: Distribution of Sale Price

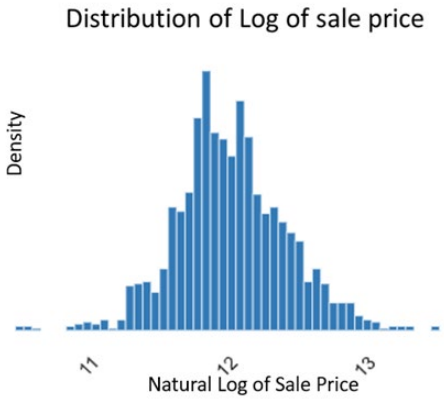


Figure 3b: Distribution of log(Sale Price)

.

We tried to analyze the distribution of the discrete categorical variables and their relationship with the target variable (sale price) using bar charts. We were able to see which category for a particular feature is contributing to the higher value. We observed that for most of the variables, there exists a high positive linear relationship between the independent and dependent variables. We also figured out that for some of the variables with high missing values like Lot Frontage, Alley, Sale price was also high. Hence, we figured that there exists a relationship between these predictor and target variables and so these features should not just be thrown out.

To identify the relationship between sale price and numerical features we examined the correlation between them and created a heatmap of the correlations shown below.

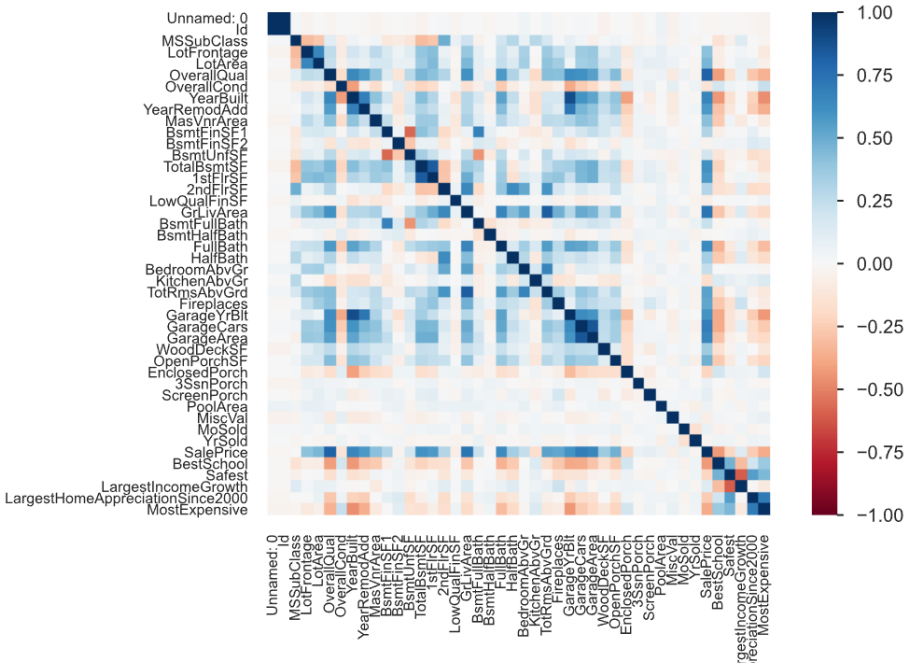


Figure 4 Correlations between numerical variables

This chart has some satisfying results such as 'OverallQuality' and 'GLivArea' are strongly positively corre lated to 'SalesPrice', which is intuitive since a larger house or a better-quality house would be more expe nsive but it was a bit surprising that '1stFlrSF' was strongly correlated with 'SalesPrice' while '2ndFlrSF' w as only weakly correlated with 'SalesPrice'.

We also noticed a negative correlation between sale price with YearSold but positive correlation with Ye arBuilt and YearRemodAdd as shown in the plot below. Hence, we thought that instead of focusing on in dividual year feature, a new variable derived from the above-mentioned variables (YearSold – YearBuilt) could be beneficial representing the age of the house.

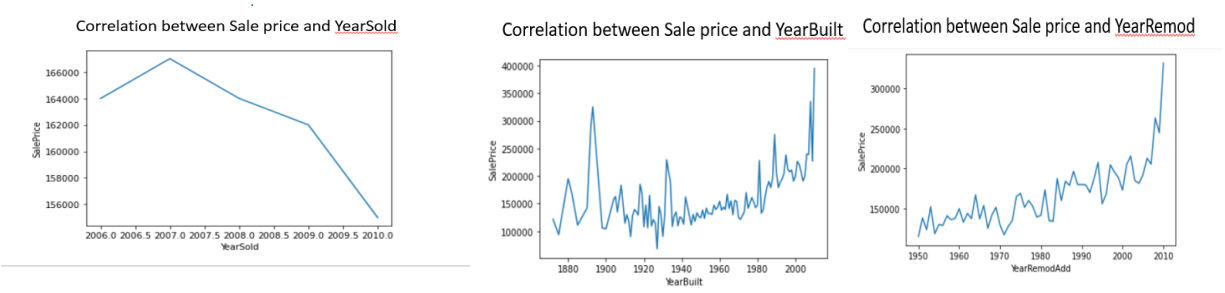


Figure 5: Relationship between Year variables and Sale Price

We also observed the following columns containing missing values:

LotFrontage, Alley, MasVnrType, MasVnrArea, BsmtQual, BsmtCond, BsmtExposure, BsmtFinType1, Bsm tFinType2, FireplaceQu, GarageType, GarageYrBlt, GarageFinish, GarageQual, GarageCond, PoolQC, Fenc e, MiscFeatures.

Since we observed that some of the features with the missing values are also highly correlated with the sale price, we decided they have to be clensed as opposed to removed. For the other discrete numerical variables, we found some of the features like overall quality have a good exponential relationship with sales price while the relation to other variables is not so defined.

For the continuous variables, we looked at the histogram, mean, median and extreme values to see if there is an outlier and we were able to find outliers for some of the variables like Lot Frontage, Lot Area, !stFlrSF, LowQualFinSF etc.

We also tried to analyze the distribution of the discrete categorical variables and their relationship with the target variable (sale price) using bar charts and were able to see which category for a particular feature is contributing to the higher value.

Data Cleaning and Feature Selection:

We have used a two-prong approach to doing data cleaning, analysis and feature selection, one in R and the other in python.

For analysis in Python, we first tried to find out the NaN values in categorical features to handle them, we assigned a new category as “Unavailable”. To handle missing values for numerical variables, since we observed lot of outliers for some of the continuous variables, we replaced the missing values with median. After that we did min-max scaling to standardize the values.

For analysis in R, factors have made data cleaning very trivial. Many of the variables, in the housing dataset, use a “NA”, implying “Not Applicable”. An example of this could be not having an “Alley” next to your home. R can easily turn the character “NA” strings into a “NA” factor. Three other variables that are numeric in nature, ‘Lot Frontage’, ‘GarageYrBlt’, and ‘MasVnrArea’ include a ‘NA’ when they are not part of the home. It was decided, as a first attempt, to treat ‘NA’ as zero. This assumption is intuitive for “Lot Frontage” and ‘MasVnrArea’ (Masonry veneer area in square feet) but not so much for GarageYrBlt, which is categorical in nature. It should be noted that there are other variables that do capture a garage not being built by showing the number of cars spots as 0 or a value of “NA” for GarageFinished.

Overview of Modeling:

The team pursued two avenues for model training – A python-based approach with all the features and a R based approach with feature selection.

Approach 1:

This approach focused on all the non-correlated features to train both tree-based models – Random Forest, ADA Boost Regression, and Gradient boost, and Regression models – Linear Regression, Lasso, Elastic net, Stacked Lasso, and KNN. The focus was to use both a training and test set with a 80/20 split in

order compare the resulting R2 and mean square error (MSE) of the various models against the test set. The performance of these models is discussed in the Table 2 and Table 3 below.

Approach 2:

This approach focused on the key features that explains the 'SalesPrice'. After scaling the data, the team trained a Step wise regression model with all features and determined that only 42 out of the 85 features were enough to explain the 'SalesPrice'. Next, the team trained a random forest model using the 42 features. Based on the variable importance measures, only 30 features were enough to explain the 'SalesPrice'. Table 1 gives the list of key features determined by the Random Forest Model.

Table 1: Key features in the order of importance – From Random Forest model

Feature	Importance	Feature	Importance	Feature4	Importance
GrLivArea	1	KitchenQual	11	Foundation	21
Neighborhood	2	GarageFinish	12	BedroomAbvGr	22
OverallQual	3	FullBath	13	GarageCond	23
TotalBsmtSF	4	HalfBath	14	BsmtCond	24
GarageCars	5	OverallCond	15	KitchenAbvGr	25
BsmtFinSF1	6	BsmtQual	16	BsmtExposure	26
LotArea	7	MasVnrArea	17	GarageQual	27
BsmtFinType1	8	HeatingQC	18	WoodDeckSF	28
ExterQual	9	Exterior1st	19	MasVnrType	29
MSSubClass	10	BldgType	20	Functional	30

Environmental Data Set Results – Neighborhood Scout, the source of the environmental data, only shows the top ten ranking for each of the categories. The relative ranking of the other remaining eight neighborhoods for each category was not given, resulting in 8 out of 18 or 44% of the neighborhoods having missing values. These missing values were assigned a value of 11, indicating that the neighborhood was not in the top ten for that category. When this environmental data was finally joined onto the Ames housing dataset, each Ames neighborhood could not be uniquely defined by its set of environmental factors. The cause of this is due to a portion of Ames neighborhoods being subsets of that from *Neighborhood Scout*. Because of this lack of differentiation in environmental factors for each neighborhood, none of the environmental factors were seen in any of the final models as shown below; however, the neighborhood was important as shown below.

Table 2: Using a 80/20 training/testing split. The table shows the R2 value of the model and the mean square, using the trained model, on the test data.

Model	R^2	Mean Square Error
RandomForestRegressor	85.67%	653.289
AdaBoostRegressor	80.44%	696.289
GradientBoostRegressor	87.55%	653.289
RandomForestRegressor W/PCA	85.10%	693.289
AdaBoostRegressor W/PCA	81.53%	653.289
GradientBoostRegressor W/PCA	84.95%	693.289
RandomForestRegressor W/KPCA	78.21%	653.289
AdaBoostRegressor W/KPCA	72.06%	697.289
GradientBoostRegressor W/KPCA	78.54%	695.234

Model Results and Discussion – In general, tree models have superior performance when compared to that of regression models; moreover, they do not require initial scaling, which makes them an ideal first candidate for getting a rough upper bound of potential performance. From Table 2, most trained models have an R^2 score between mid 70s and mid 80s. Notice that the models that did not implement PCA or KPCA had the highest R^2 score when compared to the equivalent that used one of these transformations. One main drawback of using tree-based models, particularly random forests, is their inability to tease out coefficients and be able to easily allow for drawing conclusions about the relationship between coefficients. Linear models and their variants, on the other hand, tend to have lower performance but are much easier to evaluate in terms of the coefficients and goodness of fit.

Table 3: Using a 80/20 training/testing split. The table shows the R2 value of the model and the mean square error, using the trained model, on the test data. The last model MSE is calculated using the scaled coefficients while other model uses unscaled coefficients.

Model	R^2	Mean Square Error
BSpines reduction with Linear Regression	78.43%	593.289
BSpines reduction with Group Lasso	76.60%	596.880
Lasso Regression with scaled data	76.88%	596.882
Ridge Regression with scaled data	74.01%	602.840
Linear Regression with scaled and W/PCA	72.05%	602.334
Linear Regression with various reduced parameters.	88.78%	468.661
Linear Regression W/Stepwise Regression & RF Importance For Feature Selection	89.71%	2.115 (Scaled MSE)

A variety of different linear models were implemented. This is shown in Table 3. As expected, the linear regression models have a lower R^2 score than that of the tree-based models. Interestingly, while the advanced regression techniques, in general, created models with R^2 scores in mid to high 70s, it was possible to create even better trained models, using simple linear regression with careful feature selection. This is highlighted in the last two models of Table 3. It is noteworthy that while the R^2 increases as you add more variables, one can also boost the "fit" by removing extraneous predictors. The last two linear models have a surprisingly high R^2 score when compared to that of the tree-based models. As the final linear regression model is simple yet seems to explain a high degree of the variability in the data, a further investigation of this model is warranted.

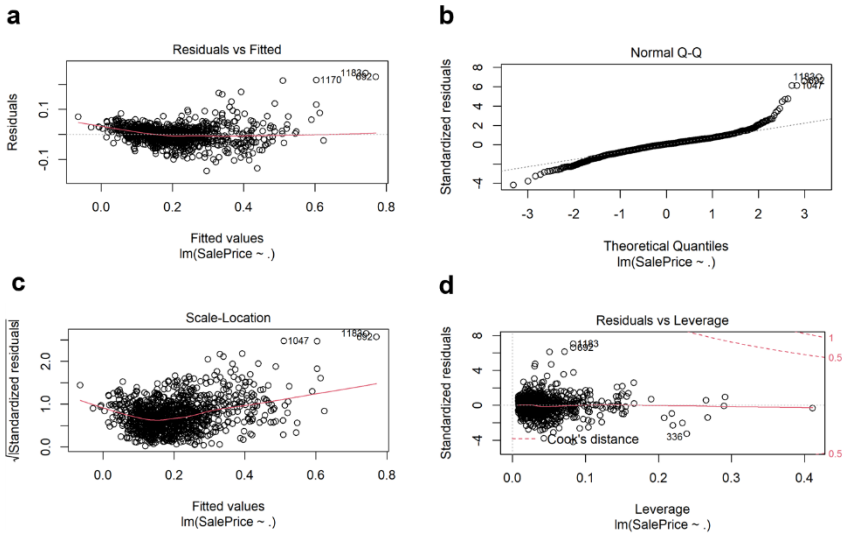


Figure 6: All plots are concerned with the final regression model in Table 3. (a) A plot of Residuals vs Fitted values. (b) Normal Q-Q plot. (c) A plot of the square root of Standardized Residuals vs Fitted values. (d) A plot of Residuals vs Leverage.

Figure 6, includes a set of plots that highlight some key metrics of the final linear regression model in Table 3. The plot of Residuals vs Fitted values indicates that the choice of using a linear model appears to be appropriate, given the nearly horizontal nature of the red line. Figure 6 plot b indicates that there is some heteroscedasticity in the model. There were additional attempts at using the log of the dependent variable, sales price, to help remove heteroscedasticity but this did not seem to provide much improvement; therefore, the linear form of the dependent variable was chosen due to simplicity. We can also see from plot c that some

outliers remain in the fit. Many of the "extreme" outliers, sqrt of standardized residuals > 4, were removed but others that proved not to be leverage points were left since it was difficult given 12 independent variables to find points that should be removed. We can also see from Figure 6, plot d, the absence of leverage points, which were previously removed. The set of plots in Figure 6 indicate that this model is potentially a "good candidate". In order to further expand on this it was critical to determine if there was multicollinearity in the model. The dependent variable "GrLivArea" was modeled against all other independent variables to get Generalized Variance Inflation Factors, used when categorical variables are included in the model, shown in Appendix C. From the fit in Appendix C, there is no multicollinearity in the model due the $GVIF^{1/(2* Df)}$ being less than 2 for each predictor. The absence of multicollinearity suggests one can trust the relative magnitude of each of the coefficients. From Appendix B, by evaluating the order of magnitude of the fit coefficients the most important features in descending order are GrLIvArea, TotalBsmtSf, BsmtFinSF1, LotArea, and OverallQual. This result is particularly satisfying as the correlation plot earlier suggests that "Sale Price" is highly correlated to "GrLivArea" and "OverallQual."

What Model is Best – A variety of different approaches were used to try to ascertain the "best set of features" for a model. The linear model used above, as well as, the lasso cross validation and extra tree classifier, shown in Appendix D, all find that Neighborhood, BsmtFinSF1, and TotalBsmtSF should be incorporated into the model. The remaining set of features for each of the respective models differ. While all these models have relatively high performance when looking at their R^2 score, it suggests that one's choice of model should possibly be dependent on what set of features can be reliably collected, as well as, interesting to study. In short, it appears that while model selection should not only be based on the ability to fit well to training data and test well against test data but also include an interesting set of predictor variables.

Feature Engineering for Creating Better Models – The experimentation above utilized some feature engineering but we were curious if performance could be improved by performing a more rigorous set of feature engineering, as shown in Appendix E. Appendix F indicates that for the set of models tested, feature engineering did enhance the model fit as shown by the R^2 score.

Conclusion And Further Directions – The set of studies here demonstrated the ability to create models that could relatively easily track residential homes' sales price as a function of the physical aspects of the home. This achievement can now be leveraged by homeowners, among a variety of other interested parties, to better understand the relationship between a home's value and its amenities, resulting in increasing profitability and service to the community. Unfortunately, the environmental dataset was unable to add additional insights into this modeling effort. Regardless, this study demonstrated the ability to identify a significantly paired down set of features that could help predict home prices. While a variety of models, with some identical features, were shown to have respectable predictive power, further investigations should be performed on figuring out the optimal set of features. It should be noted that these sets of models were focused on one particular city in Iowa but it is quite possible the home preferences could be different in other geographical areas. The set of procedures incorporated in this investigation would be ideal to investigate this further. One last direction that could be greatly beneficial is the use of scaping homes from online realtor sites to get a clearer picture of the set of relevant environmental factors for that home.

Works Cited:

1 https://www.freddiemac.com/perspectives/sam-khater/20210415-single-family-shortage, accessed 07/04/2022 2022.
2 Ozgur, C., Hughes, Z., Rogers, G., and Parveen, S.: 'Multiple Linear Regression Applications in Real Estate Pricing', Business Faculty Publications, 2016, 61.
3 Ceccato, V., and Wilhelmsson, M.: 'Do crime hot spots affect housing prices?', Nordic Journal of Criminology, 2019, 21, (1), pp. 84-102.
4 Davidoff, I.A.N., and Leigh, A.: 'How Much do Public Schools Really Cost? Estimating the Relationship between House Prices and School Quality', Economic Record, 2008, 84, (265), pp. 193-206.

Appendix A:

1. Dataset 1 link
   https://www.kaggle.com/c/house-prices-advanced-regression-techniques
2. Dataset 2 link
   https://www.neighborhoodscout.com/ia/ames

Appendix B:

Below are the independent variables that all have a greater than 99.9% confidence for a linear model of SalesPrice to all other selected features.

```
Call:
lm(formula = SalePrice ~ ., data = selectedVariableDf)

Residuals:
      Min         1Q     Median         3Q        Max
-0.145752 -0.018705  0.001035  0.017652  0.244159

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         -0.022747   0.022071  -1.031  0.30295
GrLivArea            0.483219   0.021499  22.476  < 2e-16 ***
NeighborhoodBlueste -0.012590   0.038151  -0.330  0.74147
NeighborhoodBrDale  -0.023288   0.015330  -1.519  0.12903
NeighborhoodBrkSide -0.005477   0.013094  -0.418  0.67582
NeighborhoodClearCr -0.002290   0.013949  -0.164  0.86961
NeighborhoodCollgCr  0.008164   0.011324   0.721  0.47112
NeighborhoodCrawfor  0.011823   0.012887   0.917  0.35911
NeighborhoodEdwards -0.008336   0.012247  -0.681  0.49622
NeighborhoodGilbert  0.007996   0.011875   0.673  0.50089
NeighborhoodIDOTRR  -0.024855   0.013378  -1.858  0.06346 .
NeighborhoodMeadowV -0.011573   0.016181  -0.715  0.47464
NeighborhoodMitchel -0.011800   0.012806  -0.921  0.35706
NeighborhoodNAmes   -0.008152   0.011764  -0.693  0.48847
NeighborhoodNoRidge  0.061671   0.013142   4.692 3.06e-06 ***
NeighborhoodNPkVill -0.019774   0.017820  -1.110  0.26741
NeighborhoodNridgHt  0.028417   0.012074   2.353  0.01878 *
NeighborhoodNWAmes  -0.015436   0.012366  -1.248  0.21221
NeighborhoodOldTown -0.037550   0.012142  -3.093  0.00204 **
NeighborhoodSawyer  -0.007203   0.012579  -0.573  0.56703
NeighborhoodSawyerW -0.003659   0.012239  -0.299  0.76502
NeighborhoodSomerst  0.022122   0.011603   1.907  0.05686 .
NeighborhoodStoneBr  0.060857   0.013518   4.502 7.49e-06 ***
```

```
NeighborhoodStoneBr  0.060857   0.013518   4.502 7.49e-06 ***
NeighborhoodSWISU   -0.041480   0.014882  -2.787  0.00541 **
NeighborhoodTimber   0.003750   0.012814   0.293  0.76986
NeighborhoodVeenker  0.022281   0.017612   1.265  0.20612
OverallQual          0.137176   0.014122   9.714  < 2e-16 ***
TotalBsmtSF          0.216452   0.022436   9.648  < 2e-16 ***
GarageCars           0.055397   0.008295   6.678 3.91e-11 ***
BsmtFinSF1           0.169736   0.017777   9.548  < 2e-16 ***
LotArea              0.167011   0.025472   6.557 8.62e-11 ***
ExterQualFa         -0.044394   0.015133  -2.934  0.00342 **
ExterQualGd         -0.049507   0.007651  -6.471 1.49e-10 ***
ExterQualTA         -0.054097   0.008586  -6.301 4.36e-10 ***
KitchenQualFa       -0.048012   0.010068  -4.769 2.12e-06 ***
KitchenQualGd       -0.039419   0.005659  -6.966 5.75e-12 ***
KitchenQualTA       -0.046953   0.006379  -7.360 3.71e-13 ***
OverallCond          0.046905   0.008128   5.771 1.04e-08 ***
BedroomAbvGr        -0.078514   0.015302  -5.131 3.43e-07 ***
FunctionalMaj2       0.034318   0.022846   1.502  0.13337
FunctionalMin1       0.042110   0.015219   2.767  0.00576 **
FunctionalMin2       0.039972   0.015194   2.631  0.00864 **
FunctionalMod        0.036374   0.016949   2.146  0.03209 *
FunctionalSev       -0.073245   0.038924  -1.882  0.06015 .
FunctionalTyp        0.056941   0.013378   4.256 2.26e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.03636 on 1048 degrees of freedom
Multiple R-squared:  0.8971,    Adjusted R-squared:  0.8928
F-statistic: 207.7 on 44 and 1048 DF,  p-value: < 2.2e-16
```

Appendix C:

Below is the Generalized Variance Inflation Factor for a fit of the independent variable “GrLivArea” to all other independent variables. Notice that the values of $GVIF^{1/(2* Df)}$ are all below 2, indicating no multicollinearity. Please note “Df” in the figure below is the degrees of freedom.

```
                 GVIF Df GVIF^(1/(2*Df))
Neighborhood 10.762565 24        1.050747
OverallQual   3.461456  1        1.860499
TotalBsmtSF   1.972555  1        1.404477
GarageCars    1.951318  1        1.396896
BsmtFinSF1    1.517046  1        1.231684
LotArea       1.254821  1        1.120188
ExterQual     5.945441  3        1.345955
KitchenQual   4.508254  3        1.285291
OverallCond   1.374985  1        1.172597
BedroomAbvGr  1.251572  1        1.118737
Functional    1.334901  6        1.024363
```

Appendix D:

Set of important variables found from a lasso cross validation and an extra tree classifier.

```
from sklearn.linear_model import LassoCV

reg = LassoCV(cv=5, random_state=42, fit_intercept=False).fit(X_train,y_train)
X_train.columns[reg.coef_>= 1e-9]
```

```
Index(['Neighborhood', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF',
       '1stFlrSF', '2ndFlrSF', 'ScreenPorch'],
      dtype='object')
```

```
from sklearn.ensemble import ExtraTreesClassifier
clf = ExtraTreesClassifier(n_estimators=50)
clf = clf.fit(X_train, y_train)
min_val = np.min(clf.feature_importances_[clf.feature_importances_ < 5])
max_val = np.max(clf.feature_importances_)
X_train.columns[np.argwhere(clf.feature_importances_ > 0).reshape(-1)]
```

```
Index(['MSSubClass', 'MSZoning', 'Street', 'LotShape', 'LandContour',
       'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1',
       'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl',
       'Exterior1st', 'Exterior2nd', 'Foundation', 'Heating', 'CentralAir',
       'Functional', 'PavedDrive', 'SaleType', 'SaleCondition', 'BsmtFullBath',
       'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr',
       'TotRmsAbvGrd', 'Fireplaces', 'GarageCars', 'BsmtFinSF1', 'BsmtFinSF2',
       'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'OverallQual', 'OverallCond', 'ExterQual', 'ExterCond', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'HeatingQC', 'KitchenQual', 'FireplaceQu',
       'GarageQual', 'GarageCond', 'PoolQC', 'Fence', 'LotArea', 'GrLivArea',
       'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea'],
      dtype='object')
```

Appendix E:

The use of feature engineering shown below for better categorizing the independent variables.

```
#Nominal A variable that has no numerical importance, for example color or city.
categorical_columns_labels = ["MSSubClass", "MSZoning", "Street", "LotShape", "LandContour", "Utilities",
                              "LotConfig",  "LandSlope", "Neighborhood", "Condition1", "Condition2", "BldgType",
                             "HouseStyle", "RoofStyle", "RoofMatl", "Exterior1st", "Exterior2nd",
                            "Foundation", "Heating","CentralAir","Functional","PavedDrive","SaleType","SaleCondition"]

#Ordinal A variable that has some order associated with it like our place example above
categorical_columns_ranking = ["OverallQual", "OverallCond", "ExterQual", "ExterCond", "BsmtQual", "BsmtCond",
                              "BsmtExposure","HeatingQC","KitchenQual","FireplaceQu","GarageQual","GarageCond",
                              "PoolQC","Fence"]

#Nominal A variable that has no numerical importance, for example color or city.
categorical_columns_ordinal = ["BsmtFullBath", "BsmtHalfBath", "FullBath",
                              "HalfBath", "BedroomAbvGr","KitchenAbvGr","TotRmsAbvGrd", "Fireplaces",
                              "GarageCars","BsmtFinSF1","BsmtFinSF2", "BsmtUnfSF", "TotalBsmtSF",
                             "1stFlrSF","2ndFlrSF", "LowQualFinSF","BsmtFullBath"]

categorical_columns_year = ["YearBuilt", "YearRemodAdd", "MoSold","YrSold"]

continious_columns = ["LotArea","GrLivArea", "GarageArea","WoodDeckSF","OpenPorchSF",
                      "EnclosedPorch","3SsnPorch", "ScreenPorch", "PoolArea"]

continous_currency = ["MiscVal"]
```

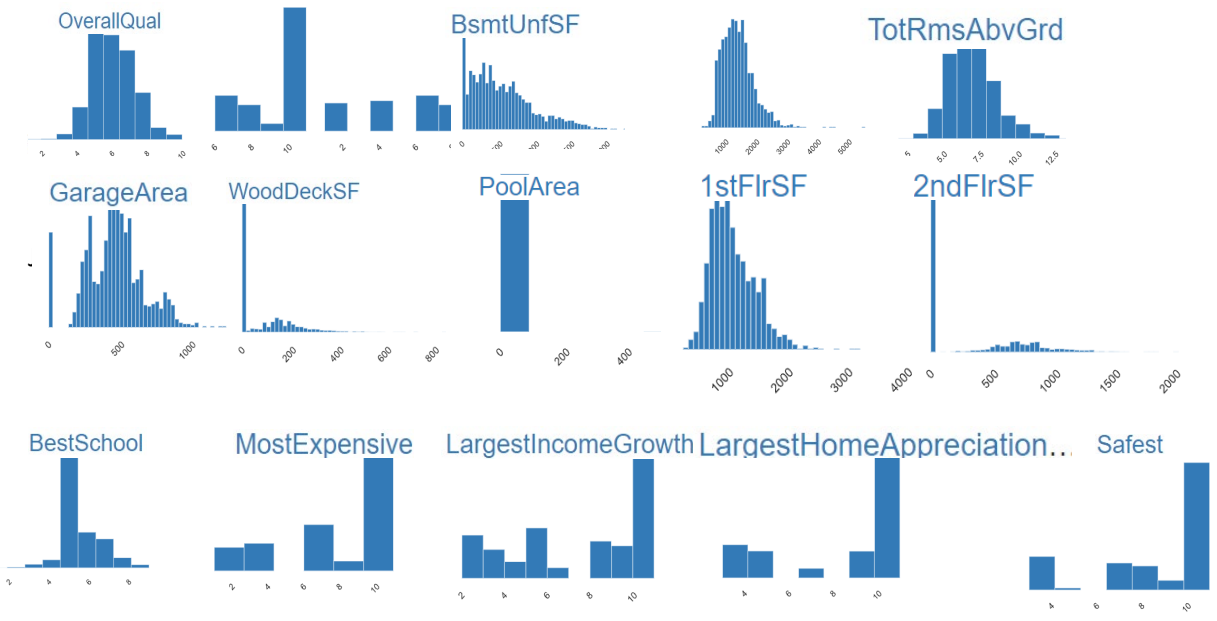
Appendix F:

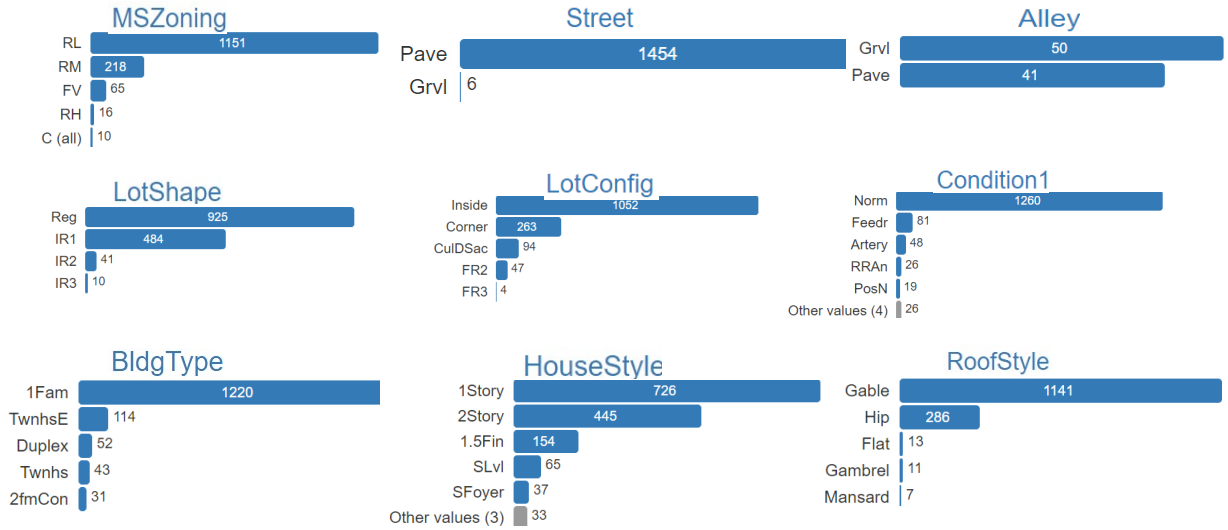
After performing feature engineering, from Appendix D, a table of R^2 score and mean square error was created on a dataset that was split 80/20 training/test.

Model	R^2	Mean Square Error
RandomForestRegressor W/Feature Engineering & Feature Selection	89.92%	653.289
AdaBoostRegressor W/Feature Engineering & Feature Selection	88.82%	696.289
GradientBoostRegressor W/Feature Engineering & Feature Selection	91.44%	653.289
RandomForestRegressor W/PCA	85.10%	693.289
AdaBoostRegressor W/PCA	81.53%	653.289
GradientBoostRegressor W/PCA	84.95%	693.289
RandomForestRegressor W/KPCA	78.21%	653.289
AdaBoostRegressor W/KPCA	72.06%	697.289
GradientBoostRegressor W/KPCA	78.54%	695.234

Appendix G:

Frequency distribution plots for numeric and categorical variables





Appendix H:

MGT6203 - Team 84 Project Plan

Proposal	**Team Members**	**Start**	**Complete**	**Status**
Explore Data Sets	ALL	6/6/2022	6/10/2022	Complete
Choose Data Set	ALL	6/13/2022	6/17/2022	Complete
Data download APIs	Jonathan, Vignesh	6/15/2022	6/20/2022	Complete
Data Exploration	Siddharth, Reena	6/15/2022	6/20/2022	Complete
Project Timeline creation	Vignesh, Siddharth	6/20/2022	6/22/2022	Complete
Proposal Document	ALL	6/20/2022	6/22/2022	Complete
Development/Progress Report	**Team Members**	**Start**	**Complete**	**Status**
Generate Dataset and Samples	Jonathan, Reena	6/23/2022	7/8/2022	Complete
Initial Software Setup	Siddharth, Vignesh	6/23/2022	7/8/2022	Complete
Write Progress Report	ALL	7/4/2022	7/6/2022	Complete
Progress Video	ALL	7/4/2022	7/6/2022	Complete
Hyperparameter tuning	Jonathan, Reena	7/11/2022	7/15/2022	Complete
Visualization	Siddharth, Vignesh	7/11/2022	7/15/2022	Complete
Final Report	**Team Members**	**Start**	**Complete**	**Status**
Final Video Presentation	ALL	7/18/2022	7/20/2022	Complete
Code and Data package	Siddharth, Reena	7/18/2022	7/22/2022	Complete
README-User Guide	Jonathan, Vignesh	7/18/2022	7/24/2022	Complete
Final Report Slides	Jonathan, Vignesh	7/20/2022	7/24/2022	Complete
Final Report	Siddharth, Reena	7/20/2022	7/24/2022	Complete