TEXAS A&M
UNIVERSITY®

STAT 654

Final Project -Predicting Housing Prices

Submitted By:

Aiman Manzoor

Table of Contents

Introduction

Overview

The housing market in the United States is a significant sector of the economy, and its size can be measured in various ways. According to the National Association of Realtors, the total value of existing homes sold in the United States in 2021 was approximately $1.7 trillion. Moreover, the US housing market includes various other sectors such as new construction and real estate services. In 2021, the new residential construction market was valued at around $710 billion, according to the US Census Bureau. Furthermore, The industrial chain of the housing market is long and complex, and many different sectors and industries are involved in the process. Each stage of the chain has its own unique set of businesses, professionals, and services, all of which contributes to the overall functioning of the housing market.

Overall, prospective and current homeowners, developers, investors, appraisers, tax assessors, and other players in the real estate industry, such as mortgage lenders and insurance, care about an accurate projection of the property price. Traditional house price forecasting relies on cost and sale price comparisons without an established benchmark or certification procedure. Consequently, having a house price prediction model available closes a critical information gap and boosts the effectiveness of the real estate market.

Goal

Linear regression is the foundation of many statistical analysis, which we covered extensively in this course. To be familiar with all aspects of the linear regression model by this project, we should understand its basic assumptions, such as linearity, independence, homoscedasticity, and normality. We should also know how to perform multiple linear regression, and polynomial regression in Python. Understanding how to evaluate the goodness of fit of a linear regression model using metrics such as R-squared, adjusted R-squared, and residual plots is crucial. Additionally, we should learn how to make predictions using a linear regression model and assess their accuracy.

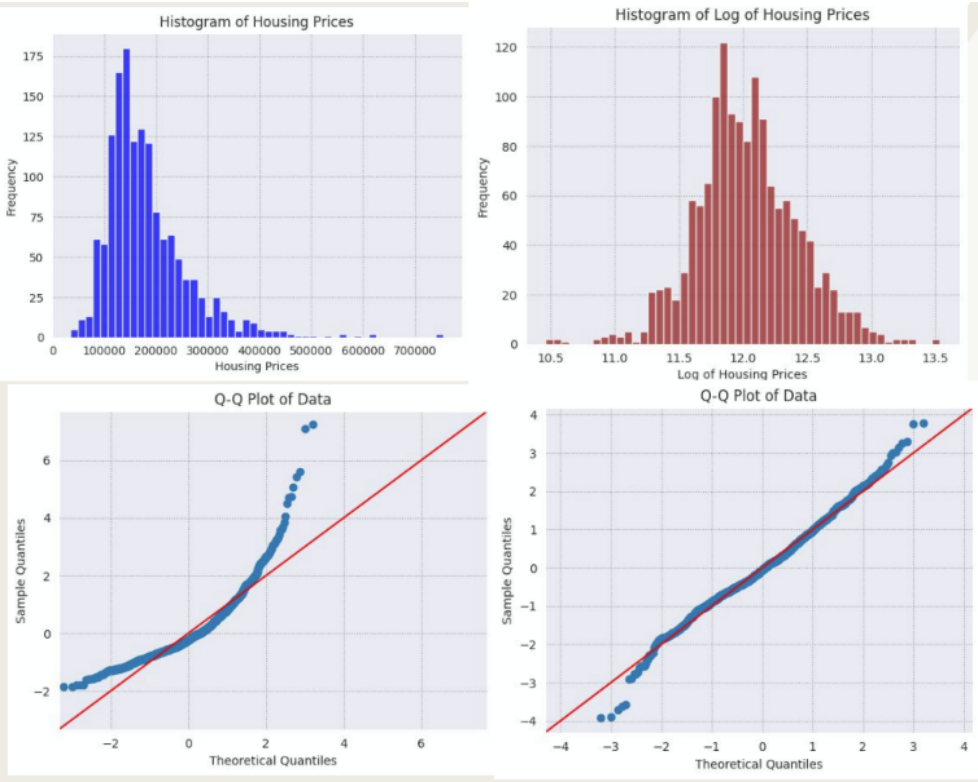
In addition to linear regression, it is important to learn to use some of the other popular machine learning techniques because we have a good and comprehensive dataset allowing us to

do so. This includes understanding the difference between supervised and unsupervised learning, and knowing when to use each type of learning. We use popular machine learning algorithms such as decision trees, random forests, principal components analysis (PCA), and neural networks. Knowing how to preprocess data for machine learning and using popular machine learning libraries such as Scikit-learn is also crucial. Understanding how to evaluate the performance of a machine learning model with continuous response using metrics such as R-squared or mean squared error (MSE). Lastly, knowing how to perform hyperparameter tuning to optimize the performance of a machine learning model is needed.

Finally, we should know how to compare the performance of different models by doing this project. This includes understanding how to choose an appropriate evaluation metric based on the nature of the problem and the data. We should learn how to use cross-validation to assess the generalization performance of different models and use statistical tests such as t-tests and AIC/BIC to compare the performance of different models. Additionally, knowing how to perform feature selection to identify the most important predictors for a given model is important. Lastly, we should also understand the trade-offs between model complexity and model performance and how to choose the appropriate level of model complexity for a given problem.

Explanation of DataSet

The project has opted to utilize a dataset from Kaggle, which pertains to an ongoing competition and can be accessed through the following link: House Prices - Advanced Regression Techniques | Kaggle. This dataset consists of a total of 79 variables, each of which delineates various attributes of a given house. The dependent variable in our study shall be the sale price of these houses, which possesses a continuous nature and is, therefore, amenable to supervised learning approaches.



In order to explore the distribution of the response variable, we constructed a histogram and a QQ plot shown above. Our findings indicate that the distribution of the variable is left skewed. To address this skewness, we applied a logarithmic (log) transformation to the original variable and generated new histogram and QQ plot figures. The results of this transformation demonstrate a notable improvement in normality, as the data now exhibits a nearly perfect normal distribution.

Pre-Processing

In order to pre-process our data we started by dropping the variables which contained null values. In our dataset there were in total nineteen variables which we dropped as they contained null values. Then we segregated the categorical variables from the numerical variables. To encode our dummy variables we used the Pandas library. We used the function pd.dummies to encode our categorical variables. Then we normalized the data using the z-score method to achieve the standard deviation of 1 and mean of zero for every variable. We splitted our data in 70:30 for training and testing our dataset for the machine learning algorithms.

Feature Selection

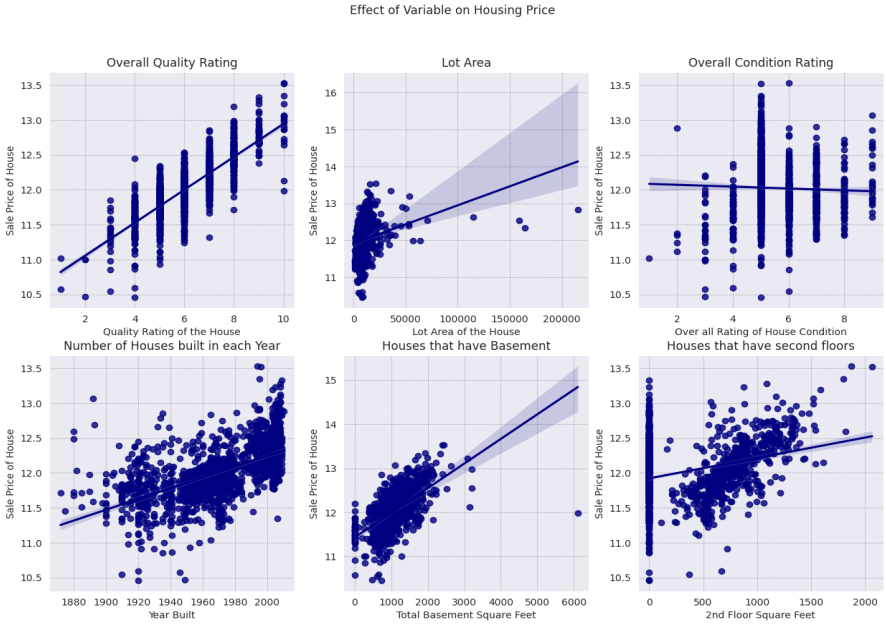
Since our dataset contained a large number of variables, we had to use a technique to trim them down. We decided to implement the Ordinary Least Squares (OLS) method to reduce our number of variables and to choose the top **twenty five** variables based on their p-values which needed to be less than **0.001**. We ran models with the supervisor which was not in the log form and also with the one which was in log form. We decided to go with the log-linear model after checking all the OLS assumptions . We selected six numerical variables and the rest were categorical variables which were encoded using the Pandas library.We had 60 columns and 1460 rows. Following is the list of variables which we selected:

LotArea	**Neighborhood_Mitchel**	**Condition2_RRAn**
OverallQual	**Neighborhood_NAmes**	**Condition2_RRNn**
OverallCond	**Neighborhood_NPkVill**	**RoofMatl_ClyTile**
YearBuilt	**Neighborhood_NWAmes**	**RoofMatl_CompShg**
TotalBsmtSF	**Neighborhood_NoRidge**	**RoofMatl_Membran**
2ndFlrSF	**Neighborhood_NridgHt**	**RoofMatl_Metal**
GrLivArea	**Neighborhood_OldTown**	**RoofMatl_Roll**
LandSlope_Gtl	**Neighborhood_SWISU**	**RoofMatl_Tar&Grv**
LandSlope_Mod	**Neighborhood_Sawyer**	**RoofMatl_WdShake**
LandSlope_Sev	**Neighborhood_SawyerW**	**RoofMatl_WdShngl**
Neighborhood_Blmngtn	**Neighborhood_Somerst**	**KitchenQual_Ex**
Neighborhood_Blueste	**Neighborhood_StoneBr**	**KitchenQual_Fa**
Neighborhood_BrDale	**Neighborhood_Timber**	**KitchenQual_Gd**
Neighborhood_BrkSide	**Neighborhood_Veenker**	**KitchenQual_TA**
Neighborhood_ClearCr	**Condition2_Artery**	**GarageQual_Ex**
Neighborhood_CollgCr	**Condition2_Feedr**	**GarageQual_Fa**
Neighborhood_Crawfor	**Condition2_Norm**	**GarageQual_Gd**
Neighborhood_Edwards	**Condition2_PosA**	**GarageQual_Po**
Neighborhood_Gilbert	**Condition2_PosN**	**GarageQual_TA**
Neighborhood_IDOTRR	**Condition2_RRAe**	
Neighborhood_MeadowV		

Ordinary Least Square Assumptions

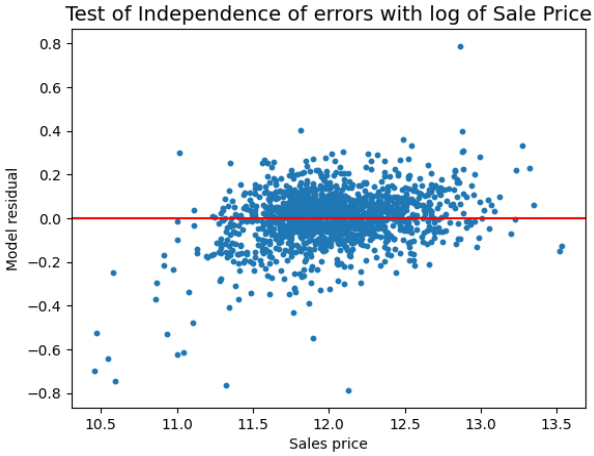
There were five assumptions against which we tested for the Log-Linear Model:

1) **Linearity:** The relationship between the dependent variable and the independent variables is linear, meaning that changes in the independent variables are associated with constant changes in the dependent variable.



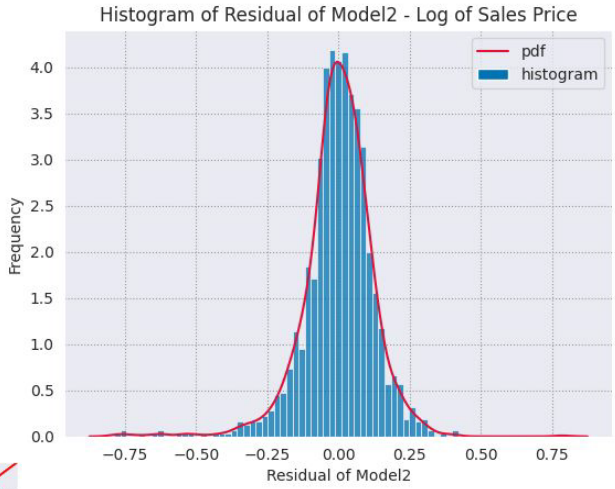
We can see from the graph that the linearity assumption is met for each of the variables. Each of the variables has a linear relationship with the dependent variable of log of sales price.

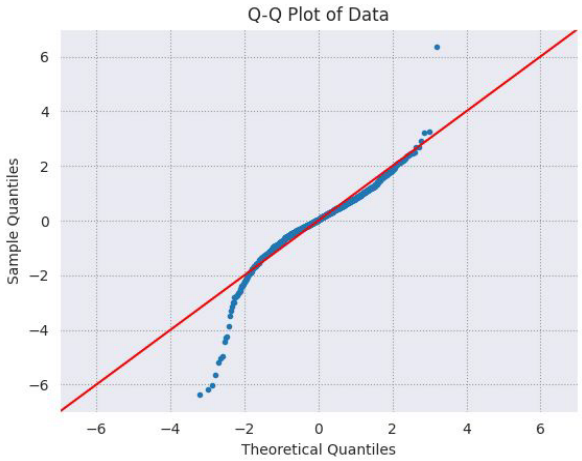
2) **Independence of errors:** The errors or residuals (the differences between the observed values and the predicted values) are independent, meaning that the errors for one observation are not influenced by the errors of other observations.



We can see from the graph that the errors are not influenced by any of the observations, which means that the log-linear model meets this assumption.

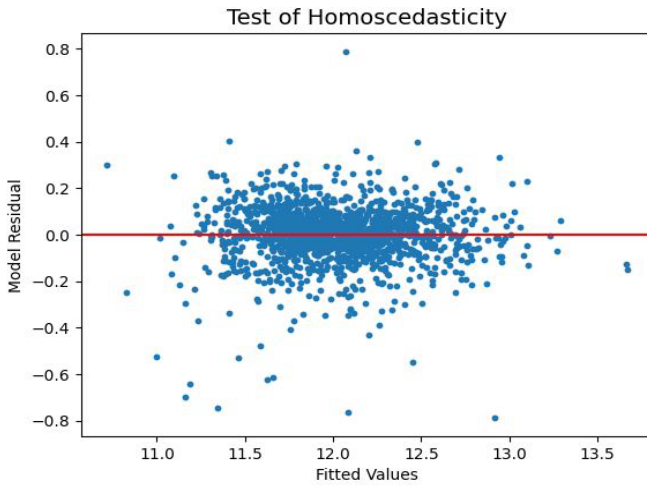
3) **Normality of errors:** The errors are normally distributed, meaning that the distribution of the errors follows a bell-shaped normal distribution.





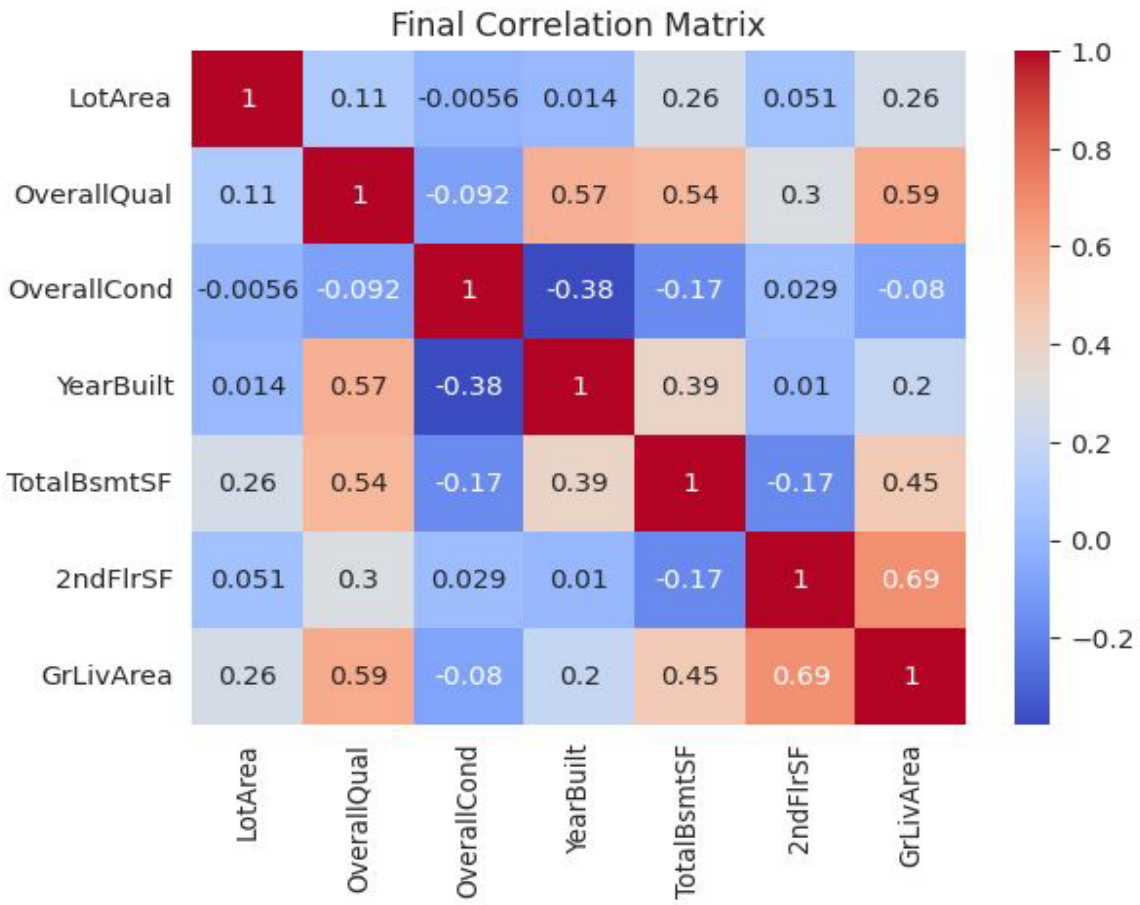
The errors are normally distributed for the log-linear model. The probability distribution function also has a normal shape which is also corroborated by the Quantile to Quantile plot which is fitted on the red line. This means that the assumption regarding the normality of errors is met by the log-linear model.

4) **Homoscedasticity:** The errors have constant variance across all levels of the independent variables, meaning that the variability of the errors is the same for all values of the independent variables.



We can see from the graph that the errors have a constant variance. They are not diverging from line at zero rather they are clustered together and are not increasing. Therefore, we claim that the log-linear model fits this assumption as well.

5) **Multicollinearity:** There is no perfect or near-perfect linear relationship among the independent variables, as this can cause issues in estimating the regression coefficients and interpreting their effects.

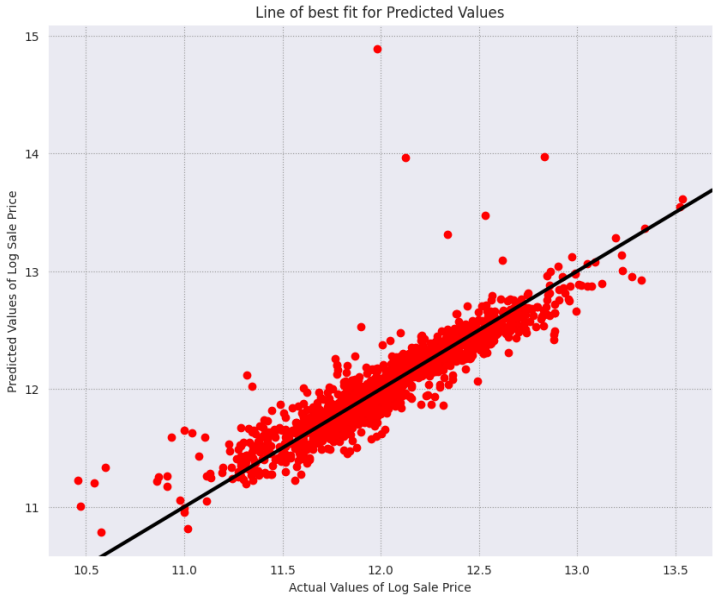


The colors in deep red show a perfect collinearity and the colors that are light red show strong correlation (1-0.8). The colors in light orange and skin highlight a moderate to slightly weak correlation (0.6-0.4). Colors that are in light blue show a very weak to no correlation (0.2-0.0). Colors that are in dark blue show a negative correlation. From the correlation matrix we can see that most of our values have a weak correlation since the matrix is mostly light blue. There are some negative correlations as well, for example the correlation coefficient between “YearBuilt” and “OverallCond” is -0.38. Which means that as the year increases by 1 unit the condition of the house deteriorates, which makes sense. There is no strong correlation between the variables which means that our model also meets the assumption of no perfect collinearity.

Results from Ordinary Least Squares

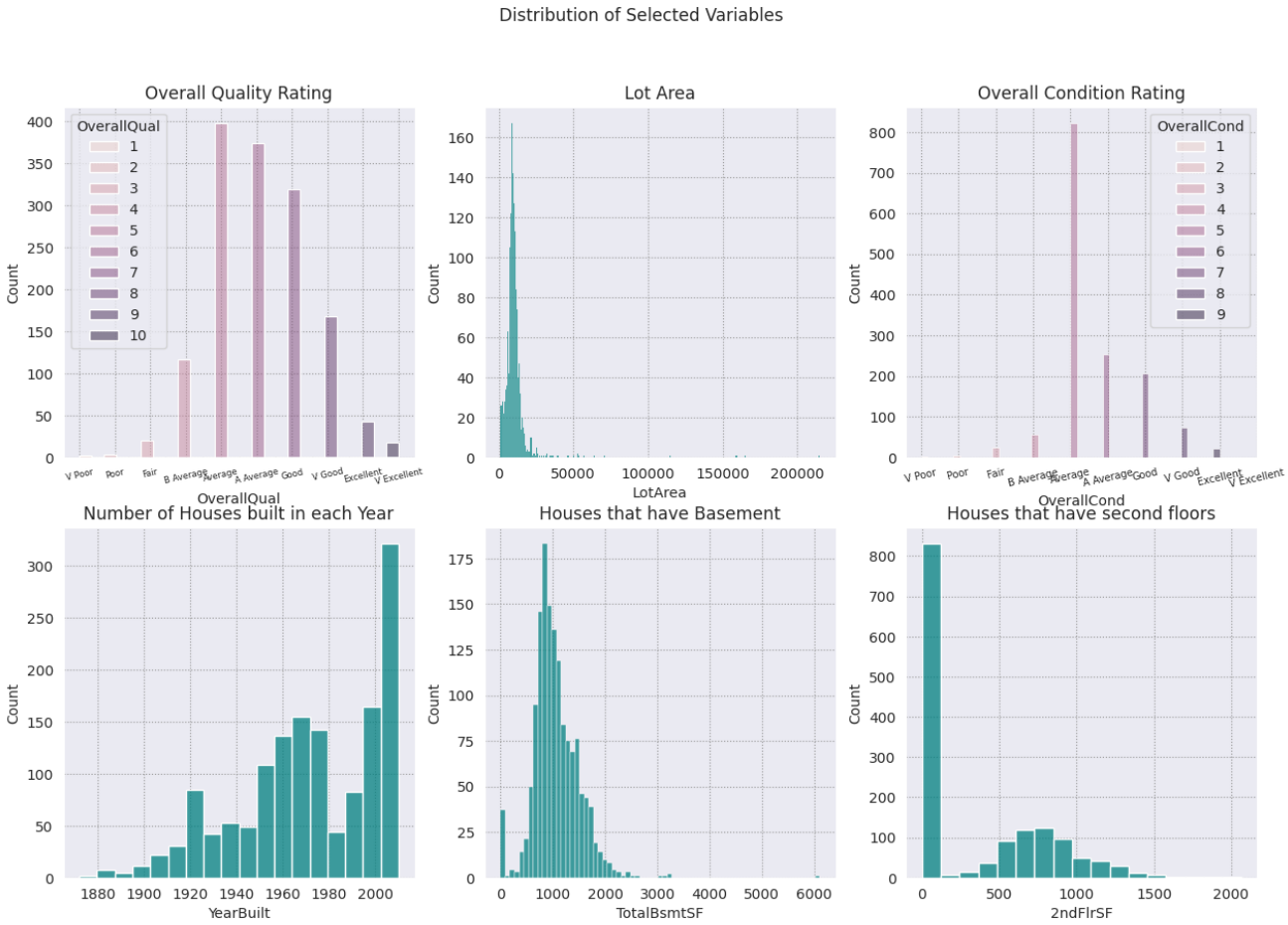
We computed the R-squared and the MSE for our model. The R-Squared for our model turned out to be 0.897, which means that 89.7% of the variation in the log of the housing sales prices is shown by the features selected in our model. This means that our model performed quite well, as most of the features that show variation in the log of the housing sales prices is a part of our model. The mean squared error of our model was 0.034, which is also quite low which means that our model performed quite well. The graph on the right shows the curve for the actual and the fitted value. Most of the fitted values overlap the actual values which means that our model performed quite well.

Another graph shows the line of best fit for the predicted values of the OLS regression. This graph also corroborates the finding that the predicted values are mostly on the line of best fit and are quite close to actual values except a few which means that our model performed well in predicting the log of housing sales price.



Exploratory Data Analysis

We conducted Exploratory Data Analysis (EDA) for all our continuous variables which is shown on the graph on right. The histogram for the 'Overall Quality Rating' and 'Overall Condition Rating' shows that the average (code=5) was the most frequent value for the variables. The 'Lot Area' has a left skewed graph and the value for it ranges mostly from 0 square feet to 3000 square feet. This makes sense because as you add more Lot Area the sales price of the house also increases. The histogram for the 'YearBuilt' shows that over the years the number of houses built has increased. It is interesting to note that there was a surge in the number of houses built in the 1920's and then in the time period between the 1960's-1980's. There is again a surge in the number of houses built in 2000's. The histogram for 'TotalBsmtSF' is a left skewed histogram. Most of the houses have a basement floor of 1000-2000's square feet. The histogram for '2ndFlrSF' shows that most houses do not have a second floor, and those who do have second floors that range from 500 to 1500 square feet.

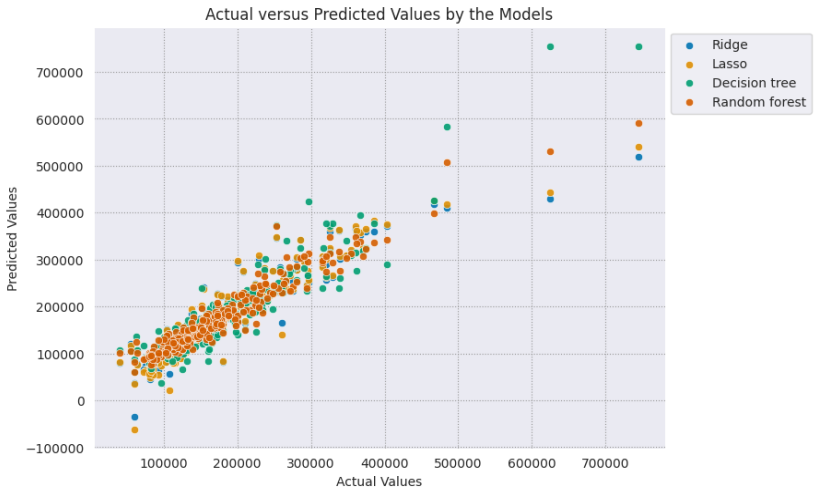


Machine Learning Models

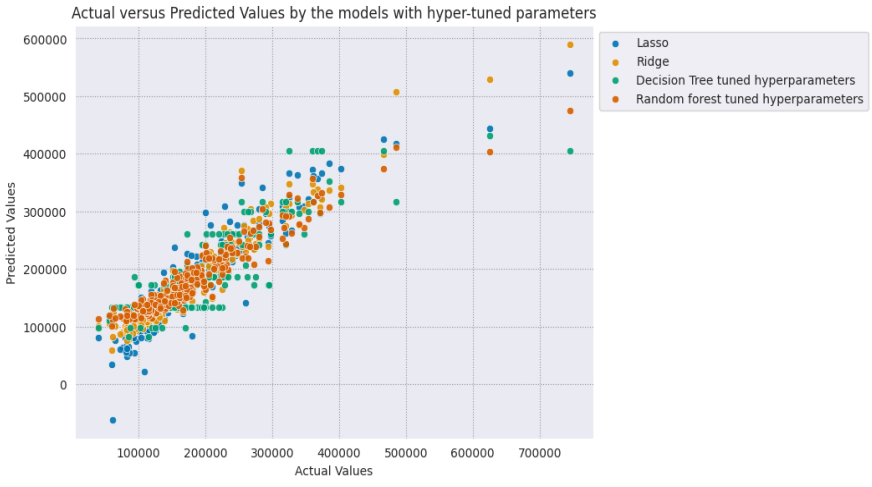
The machine learning models which we choose to analyze the relationship between the housing sales price and the list of selected variables are as follows:

1) Ridge Regression
2) Lasso Regression
3) Decision Tree Regression
4) Random Forest Regression

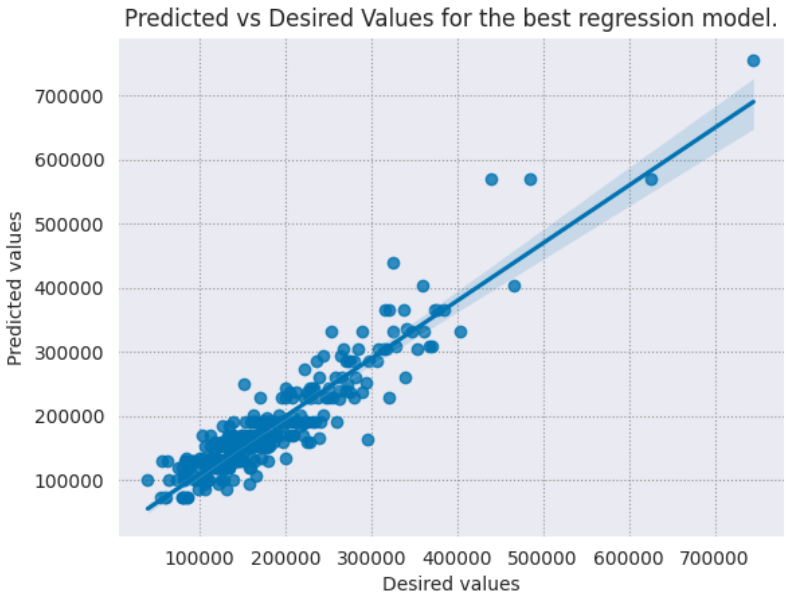
The data was split in 70:30 for training and test, respectively. We decided to go with the ridge and lasso machine learning model since they are used when there are features that are highly correlated with each other. However, we dealt with this issue in our OLS model. The decision tree and random forest were also used to compare the performance of the model. We performed our analysis with hyper-tuned and without hyper-tuned parameters.



The graph on the side shows the performance of the models without hyper tuned parameters. It is also important to note that we are taking the continuous sales price as the supervisor so that the models do not show us biased and skewed results by ignoring the data values that are at the higher end when taking logs of the supervisor. The results show the actual and predicted values for each of the models and we can see that random forest performs the best amongst all the models. The actual and predicted value are quite overlapping,



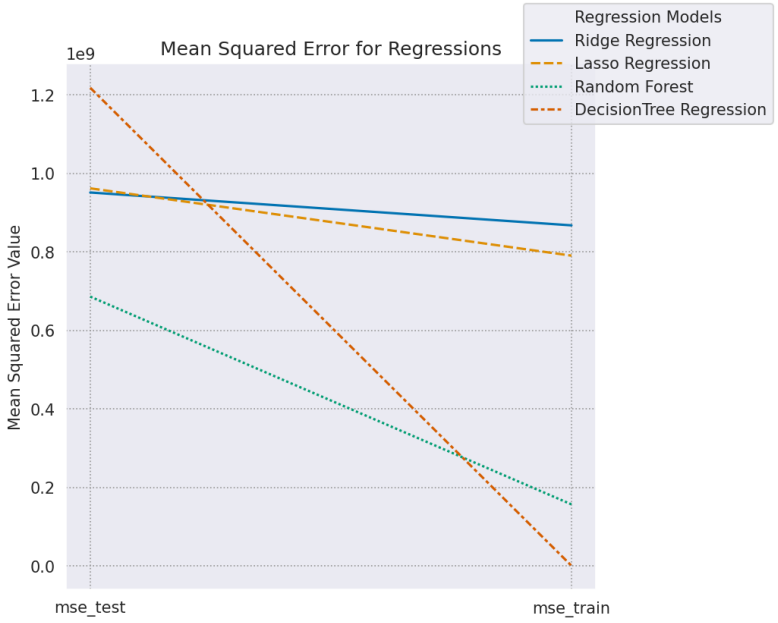
The next graph shows the results for the hyper-tuned in which we did a grid search for the ridge and lasso regression to find the best alpha value to improve the accuracy of our model. For the decision tree we did a grid search for parameters of maximum depth and maximum features and in a random forest we also added the feature of the number of estimators. For hyper-tuned parameters the random forest model performed the best amongst all the other models. We can see that the values are now more fitted/clustered together which shows that for each of the models the actual and the predicted values are more close together.



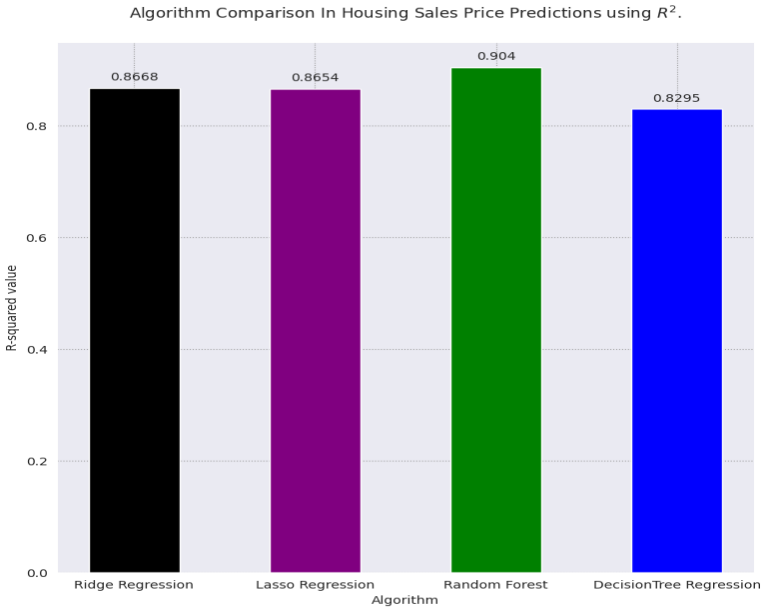
We also tried to improve the accuracy for the decision tree model in which we did a search for the parameters that give us the least possible MSE. However, it is important to bear in mind that since our supervisor is of continuous nature, the decision tree model tends to overfit the values for such datasets due to its inherent nature of high variance and low bias. The graph on the right shows the actual versus predicted values for the best decision tree regression.

Results of the Models

To analyze the performance of the models we executed the R-squared and the Mean squared errors for the models with hyper-tuned parameters. We can see that the mean squared error for random forest has decreased the most. We can also see that the decision tree has overfitted the model because of which we see a steeper decrease in the mean squared error from test to train.



The comparison of r-squared showed that the random forest model performed the best amongst all other models with the value of 0.904. Which means that 90.4% of variation in the housing sales price is shown by the features in the model. Ridge and lasso regressions almost perform equally best in terms of the r-squared with the values of 0.867 and 0.866 respectively. Therefore, we conclude that random forest gave the highest accuracy of the model in terms of mean squared error and r-squared.



PCA, Neural Networks

In addition to the multivariate linear model, ridge/lasso regression, decision tree and random forests, we also considered using PCA and Neural Networks to predict the housing prices. While the previous models relatively well predicted the housing prices at transaction, they were built on selected continuous and categorical variables to fulfill the linear regression model assumptions. This kaggle competition included 79 variables, covering a wide range of housing aspects. Limiting the model to only 7 continuous variables and 7 categorical variables would be a big waste of data. With this consideration, we introduced PCA and neural networks, which allows us to fully use all of the provided data columns.

To test the performance of PCA models, we first randomly split the labeled data into a training dataset and a validation dataset, with training data consisting of 70% of the labeled data, and validation dataset consisting of 30% of the labeled dataset. This split ratio is comparable to the other machine learning methods introduced in the previous section.

PCA reduces the dimensionality of the thirty nine continuous variables by transforming the matrix formed with the original continuous variables into an eigenvectors matrix. This eigenvectors matrix also consists of thirty nine variables, but the variances of each variable are descending, and represents a fraction of each of the original variables. With this transformation,

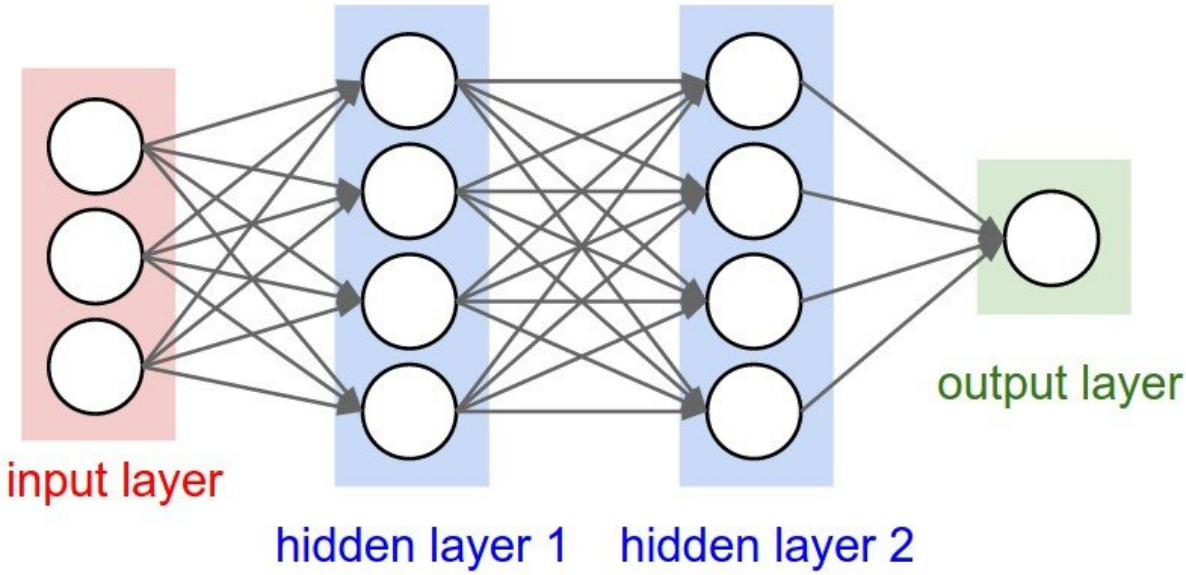
the new variables on the eigenvectors matrix can be understood as latent variables that are unexplainable, but directly affecting the housing values. While performing the PCA, we also acknowledged that the matrix transformations used to produce the eigen matrix are not based on the labeled dataset alone. Instead, we combined the 1460 labeled dataset and the 1460 unlabeled dataset to produce the PCA matrix. This effort may reduce the performance of PCA approaches on the labeled dataset, as the transformation may not perfectly reflects the distribution of the training dataset, but on the other hand, this effort also minimizes the possibility of overfitting and are beneficial in producing the targeted values for the unlabeled dataset.

One main goal of using PCA is to choose the optimal numbers of PCA components. Conventionally, 1 to 3 PCA components are sufficient in explaining most of the variances in the dataset. This could be observed with a diagram showing the explained variances by individual variables. However, as our major objective is to produce a model that most accurately predicts the sale prices of a house, we would choose the number of PCA components according to the model performances with different numbers of PCA components. The diagram below indicates the model performances by different numbers of PCA components, measured by R2 score. Note that the R2 scores are calculated by creating a multivariate linear regression model using selected numbers of PCA components and the 6 categorical variables that we chose earlier. To be consistent with the other model evaluation approaches, we chose both the sale prices and log of sale prices as our dependent variables.

Figure 1. PCA Model Performances by Number of PCA Components, Measured in Model R2 Scores

As the diagram above shows, the PCA models performances are optimal at 10 PCA components, if dependent variables are the log of sale prices, and 9 if the dependent variable is sale prices. As for the model performance, the PCA achieved an R2 of 0.898 for the Log Sale Price model and an R2 of 0.879 for the Sale Price model.

In addition to the PCA model, we also tested the neural networks. While the neural network contains various models, we specifically tested a simple three layer neural network, with only dense layers and dropout layers. The dense layer is comparable to the linear model, except that it allows any pairs of independent variables to multiply by each other, accounting for the possible interactions between different variables. Another benefit of neural networks is its ability to consider all continuous variables and categorical variables provided in the dataset. Unlike the linear regression model, the neural network is completely unexplainable, therefore does not have any prerequisite to fit its model. When using the neural network, the typical approach is to use as much data as available.



An image to illustrate the concept of neural network. Note that we also selected 2 hidden layers, but the input dimensions (78) and hidden layer dimensions (128) are different from the diagram above

Source: A 3-layer neural network with three inputs, two hidden layers... | Download Scientific Diagram (researchgate.net)

Even though our neural network is a three layer, simple neural network, there are still various hyperparameters to choose from. We tested the L1 and L2 regularization, which are similar to ridge and lasso regression. Unfortunately, both L1 and L2 regularization don't yield good results. We therefore decided to only use the dropout layer, with a dropout rate of 0.3 to

avoid model overfitting. As for the two dense layers, we have 128 neurons for each layer. We chose other numbers of neurons on each layer, and also changed layer number, but the 128 and 128 neurals seem to yield the best performance model. After 2000 epochs of training, the model yields a R2 score of 0.895 on the Log Sale Price model and a R2 score of 0.875 on the Sale Price model.

Note that for the neural network model, we split the labeled dataset into three subsets - the training dataset, the validation dataset and the test dataset, with a split rate of 0.8-0.1-0.1. Unlike the other models, whose R2 scores are tested on the validation dataset, the neural network tested the model performance on the test dataset, yielding an even more accurate measurement for its performance on the real unlabeled dataset. Note that although the neural network already yields a comparable or even better performance than other linear models and machine learning models, there is still room for improvement. Not only the types of neural networks, but also the hyper parameters are adjustable and a possible better performance neural network is still yet to be discovered for this dataset.

Although both the PCA and neural network worked relatively well in predicting housing prices, we also lost the ability to interpret the coefficients for each variable. We therefore will not be able to identify the positive or negative correlation between any independent variables and the dependent variable, as well as the relative significance of each independent variable in predicting the housing prices.

Accuracy of results/Model Comparison

As our main objective of the project is to accurately predict the sale price of a house, we compare the performance of various models and pick the one with highest R2 and lowest errors. As we used both the sale price and log of sale price as the dependent variables in most of our models, we compare the model performances on each of the dependent variables separately.

As a baseline for the comparison, we fitted all variables in a single regression model before the variable selection, which yield a R2 score of 0.934 and MSE (mean squared error) of 5.1 * 10^8 for sale price model, and a R2 score of 0.946 and MSE of 0.0106 for log of sale prices model. This performance is the best among all other models, however, this model has low validity due to the violation of linear regression assumptions. The table below shows the comparison of model performances.

	Metric	Linear Model (All Variables)	Linear Model (6 continuous and 6 categorical)	Ridge Regression	Lasso Regression	Random Forest	Decision Tree	PCA	**Neural Network**
Y = Sale Price	R2	0.934	0.872	0.867	0.865	**0.904**	0.829	0.879	0.892
	MSE	5.1 * 10^8	8.4 * 10^8	9.5 x 10^8	9.6 x 10^8	6.9 x 10^8	1.2 x 10^9	7.6 x 10^8	**6.3 x 10^8**
Y = Log Sale Price	R2	0.946	**0.897**	0.893	0.773	0.818	0.765	**0.897**	0.887
	MSE	0.0106	0.0171	0.0179	0.038	0.031	0.039	**0.0164**	0.020
Interpretability		Yes	Yes	Yes (but less)	Yes (but Less)	No	No	No	No
Robustness		Low	Middle	High	High	High	High	Middle to Low	High

Summary of model performances measured by R2 and MSE

As described above, the number of variables reduced to 6 continuous and 6 categorical after variables selections. Surprisingly, the linear regression model with reduced number of variables on log sale price outperformed other models and achieved R2 of 0.897. This performance is higher than all machine learning models, including random forest, decision tree and Neural Network.

Among all models set to predict the log of sale prices, the PCA also achieved an R2 score of 0.897, equivalent to the linear regression model. However, the PCA model achieved an even lower Mean Squared Error of 0.0164, lowest among all models.

Along the models in predicting the sale prices, Random Forest achieved the highest R2 of 0.904. However, the neural network achieved lower MSE of 6.3 * 10^8. The decision tree performed extremely well on the training dataset, but the performance significantly reduced on the testing dataset, suggesting model overfitting.

While comparing the performance of the models, it is also essential to acknowledge that the performance of the linear model was tested on the training dataset, not the testing dataset. The testing dataset was only created for the machine learning models. The only exception is the neural network, whose dataset was further split into a training, validation and testing dataset.

Conclusion

Our report focused on producing a model that most accurately predicts the housing sale prices. We used the dataset from a Kaggle ongoing competition, with 79 independent variables and 1 dependent variable. We built various models, ranging from linear regression to ridge, lasso regression, decision tree, random forest, PCA and neural network.As we discussed in the model performance comparison section, the models that most accurately predicted the housing sales price are random forest and neural networks. The model that most accurately predicted the log of housing prices is the PCA. However, we still rely on the outcomes of linear regression models to interpret the relationship between the independent variables and dependent variables. Most of the models achieved a similar level of accuracy, with identical R-squared scores in the range of 0.85 to 0.9. As no model outperformed the other models significantly, we are inclined to believe that the most robust models, random forest and neural network, are the most optimal models for this problem.

In terms of the relationship between independent variables and dependent variables, the linear regression model suggests that housing conditions are almost equally important as the housing qualities. The dataset includes the attributes of housing quality and housing conditions, both of which ranged from 0 to 10 in a Likert scale. With a coefficient of 0.055, a one level increase in housing condition will increase the sale prices by around 5.5%; similarly, a one level increase in housing quality will increase the housing price by 6.3% (beta = 0.063). Newer homes are preferred by the market, but its impact on the housing sales prices are weak. With a beta of 0.0033, the houses that are one year newer only slightly increased the sale price by around 0.33%. The most significant categorical variables in predicting the housing prices are roof types, with the metal (beta = 0.84) and membrane (beta = 0.77) being the most preferred roofing materials and the clay or tile (beta = -2.1) being the most unfavorable material.