Linear Model

For the linear model, the first step I did was data cleaning. I found out for the 'PoolQC'(Pool quality), 'Fence'(Fence quality), 'MiscFeature'(Miscellaneous feature not covered in other categories), 'FireplaceQu'(Fireplace quality), 'Alley'(Type of alley access) features, they all have above 80% missing values (NA), so I simply dropped them. The next step is dealing with other missing values. If the type of the feature is int or float, I fill it in with the mean value (for 'LotFrontage': Linear feet of street connected to the property, I filled in the mean value). If the type of the feature is char, I found out the char that most frequently occurred, and fill it into the missing values (fill in 'SBrkr' for the 'Electrical' feature). Also, use 'LabelEncoder' to fit and transform some strings to int as dummy variables. I transformed the skewed numeric features by taking log (feature + 1), and also use StanderaScaler to standardize the feature - this will make the features more normal. I then split the train set and test set as 7:3.

I used the lasso regularized linear regression model from the scikit learn module. The reason why I am using Lasso is that Lasso offers models with high prediction accuracy. The accuracy increases since the method include shrinkage of coefficients, which reduces variance and minimizes bias. It performs best when the number of observations is low, and the number of features is high which is exactly the trait of our data. The main tuning parameter for the Lasso model is alpha - a regularization parameter that measures how flexible our model is. The higher the regularization the less prone our model will be to overfit. However, it will also lose flexibility and might not capture all of the signals in the data. After much testing, I used [1, 0.1, 0.001, 0.0005] as alpha to fit our model.

Figure 1 showed the most important coefficients from a linear model. In figure 1 it is apparent that the most five important features are: 'OverallQual' (Overall material and finish quality), '1stFlrSF'(First Floor square feet), '2ndFlrSF'(Second-floor square feet), 'GarageCars'(Size of garage in square feet), 'TotRmsAbvGrd'(Total rooms above grade). So, it is obvious that for house pricing, people care the overall quality of the house and the square feet of the house. Those factors will affect the house pricing the most, and this makes sense. Because the house pricing was usually according to the house conditions, and sales as the square feet per dollar. It means that the linear model was used correctly. For the feature interactions, 'OverallQual', '1stFlrSF', '2ndFlrSF', 'GarageCars', and 'TotRmsAbvGrd', have a positive correlation between them, so they will affect the house pricing in the same direction. 'OverallQual' has the most negative correlation with 'BsmtQual'(Height of the basement), which means the height of the basement will have a big effect on the house pricing if the overall quality of the house is a pool, but if the overall quality of the house is great, the height of the basement will become irrelevant towards the house pricing.

Limitations: It represents only linear connections. It is necessary to explicitly model non-linearities. It is necessary to clearly model interactions. frequently a poor performance. For instance, it is unable to see the precise value of the "OverallQual" coefficient.

Non-linear model

For the Non-linear model, to find the best model, you will need to do a model selection. I fitted three different types of non-linear models, which were the Decision Tree Regressor, Random Forest Regressor, and KNeighbors Regressor. Grid search had been used to deal with the hyperparameters. The grid search provided by GridSearchCV exhaustively generates candidates from a grid of parameter values specified with the param_grid parameter. Grid search is used to find the optimal hyperparameters of a model which results in the most 'accurate' predictions. Also, cross-validation had been used for calculating the score. Both negative roots mean squared error and F1 Score had been used to compare those non-linear models. In the end, the Random Forest Regressor scored highest for both models, so I used Random Forest Regressor to fit and predict the non-linear model.

Individual conditional expectation (ICE) plot

An ICE plot's line depicts how the feature value impacts the example's forecast. The marginal impact of a variable on the response is graphically represented by the individual Conditional Expectation (ICE) plot. Similar to partial dependency plots (PDP), ICE plots display the influence of a characteristic for a single occurrence rather than the average effect that PDP displays. An ICE plot's lines are generated by altering the value of the relevant feature while maintaining the other features' values constant. The viewer may view several conditional relationship variations that were estimated by the black box due to the ICE algorithm.

Since the 'OverallQual' is the most important value for the house pricing from the linear coefficients plot, I used the 'OverallQual' as the parameter for the ICE plot. For most house, there is an increase in predicted house pricing probability with increasing overall quality of the house. In Figure 2,3 it is apparent that for house pricing above 20000, the 'OverallQual' affect each house pricing significantly. For most houses, there is an increase in predicted pricing with increasing 'OverallQual'. Figure 4 shows that 'BsmtQual' has a negative effect on house pricing, which means 'OverallQual' and 'BsmtQual' has a negative correlation for feature interaction.

Limitation: Each plot only has room for one feature. Assumes that the trait of interest and the other features are independent of one another. Can occasionally need minor adjusting in conjunction with PDP to prevent becoming packed. It is quite difficult to notice the association between the "SalePrice" and the "BsmtQual" in Figure 4 because the lines appear to be straight.

Feature importance

The term "feature importance" relates to methods for scoring each input feature for a certain model; the scores merely indicate the "importance" of each feature. A higher score indicates that the particular characteristic will have more of an impact on the model being used to forecast a particular variable.

For evaluating the feature importance, I simply create a table and display the correlation between the 'SalePrice' with the features and displayed the top 10 most important ones. In Figure 5 it is apparent that the 'OverallQual' is the most important feature with a score of roughly 0.817. 'GrLivArea', 'GarageCars', 'GarageArea' are also important towards the 'SalePrice'.

Limitation: The model did not reveal the feature interactions, such as the relationship between "OverallQual" and "BsmtQual."

Shapley values

A game theoretic technique called SHAP (SHapley Additive exPlanations) can be used to explain any machine learning model's output. With the help of the traditional Shapley values from game theory and their related extensions, it links optimal credit allocation with local explanations (see papers for details and citations).

First, install and import the SHAP library and put in Random Forest Regressor as the parameter for the explainer and pass in x_test for shap_values. As you can see from Figure 6, it showed the top 20 most important features when predicting 'SalePrince'. 'OverallQual' is the most important feature for predicting 'SalePrice', 'GrLivArea', '1stFlrSF', and 'TotalBsmtSF' is also important. I then randomly tested many different 'SalePrice' for the SHAP force plot. Figure 7 is the 91st force plot for the 'SalePrice' it showed clearly that the 'OverallQual' is the most significant feature to increase the 'SalePrice', and '1stFlrSF' is the second most significant feature to increase the 'SalePrince'. 'BsmtFinSF1' is the most significant feature to lower the 'SalePrice' which has a negative feature interaction with the 'OverallQual'. Then I plot the first 1000 'SalePrice' into one force plot and found out that the latter 'SalePrice' are higher and there are more factors to increase the 'SalePrice'. The shap_values and x_test had been passed in to make the SHAP summary plot, for Figure 9, the SHAP summary plot told us that when the 'SalePrice' value becomes higher, the 'OverallQual' becomes a more important feature to influence the 'SalePrice'. I also made a SHAP-dependent plot to see the relation between 'OverallQual' and 'BsmtQual'. Figure 10 is the SHAP-dependent plot which told us that if the 'OverallQual' of the house is low, then 'BsmtQual' is important, but when 'OverallQual' of the house goes higher, 'BsmtQual' becomes irrelevant, which shows the negative correlation between 'OverallQual' and 'BsmtQual'.

Limitation: High computational complexity is needed. Coalitions with N! features must be tested. unable to forecast how altering inputs will affect predictions. Each feature's impact on home prices for various force plots is noticeably varied, and occasionally even entirely different. To calculate the Shapley values, access to the data is required. It is possible for unrealistic data examples to be included if feature values are dependent.

Conclusion

For all methods used, the most important feature is always 'OverallQual' which is pretty consistent. And 'OverallQual' and 'BsmtQual' is always having a negative correlation. But the most five important features are a bit different and contradict each other. For the linear model, the most five important features are: 'OverallQual', '1stFlrSF', '2ndFlrSF', 'GarageCars', and 'TotRmsAbvGrd'. As for the non-linear model's feature importance plot, the most five important features are: 'OverallQual', 'GrlivArea', 'GarageCars', 'GarageArea', and '1stFlrSF'. For the SHAP importance plot, the most five important features are OverallQual', 'GrlivArea', '1stFlrSF', 'TotalBsmtSF', 'And firePlaces'. 'GrlivArea' means above grade (ground) living area square feet which is similar thing as the 1st-floor area and 2nd-floor area. For feature importance, the Total room above grade is important for the prediction, for the SHAP model, the basement is, and the Number of fireplaces is important.

The reason that they contradict each other is the randomness of the SHAP model and could be because of the algorithm and training difference between linear and non-linear models. Feature important method is more reliable because people do not really care about the number of fireplaces when they are purchasing houses, I do not think this would be an important feature when predicting house pricing.

Appendix

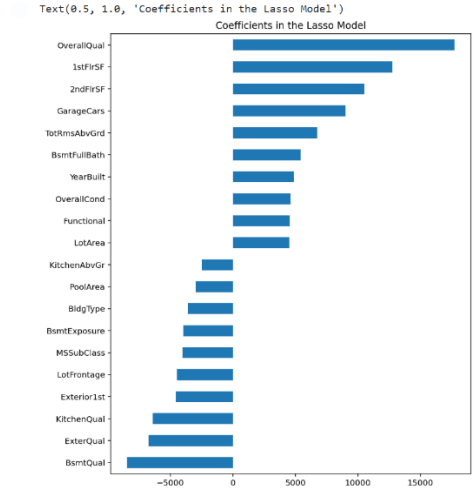
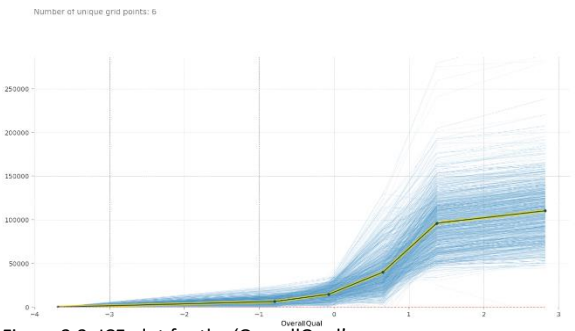


Figure 1: Coefficient plot for Lasso



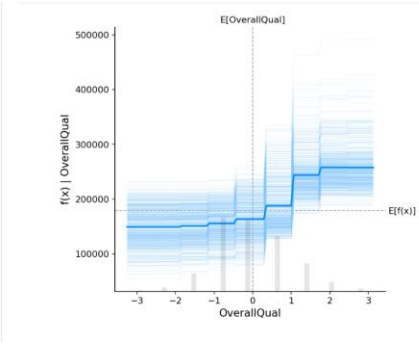


Figure 2,3: ICE plot for the 'OverallQual'

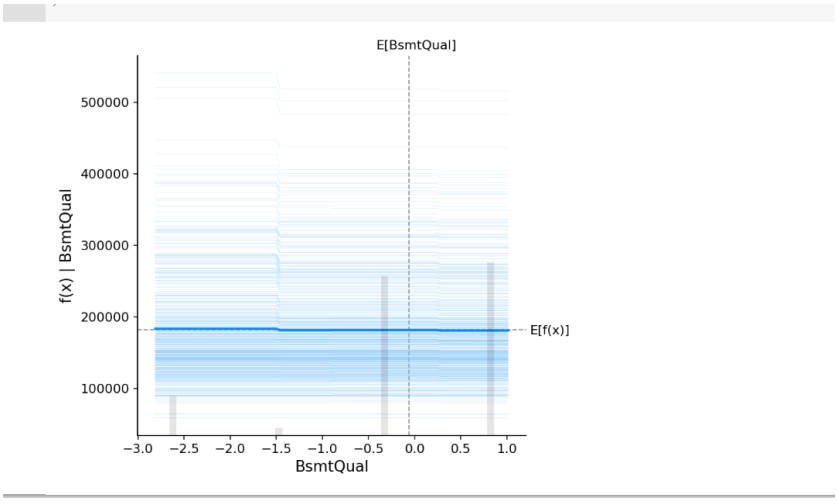


Figure 4: ICE plot for the 'BsmtQual'

Sl. No.	Feature	Score
1	OverallQual	0.8171846144867677
2	GrLivArea	0.7302537651626396
3	GarageCars	0.6806248726581888
4	GarageArea	0.650887681143594
5	1stFlrSF	0.6089550420477832
6	FullBath	0.5947706649972533
7	YearBuilt	0.586570192789716
8	YearRemodAdd	0.5656077814623218
9	TotRmsAbvGrd	0.5344224002094399
10	GarageYrBlt	0.4957939015604326

Weight	Feature
0.5303 ± 0.1089	OverallQual
0.1045 ± 0.0993	GrLivArea
0.0576 ± 0.1032	2ndFlrSF
0.0295 ± 0.0482	BsmtFinSF1
0.0250 ± 0.0522	1stFlrSF
0.0249 ± 0.0331	TotalBsmtSF
0.0230 ± 0.0857	FullBath
0.0169 ± 0.0346	LotArea
0.0159 ± 0.0374	GarageArea
0.0147 ± 0.0185	YearBuilt
0.0146 ± 0.0476	GarageCars
0.0099 ± 0.0388	ExterQual
0.0092 ± 0.0302	BsmtQual
0.0091 ± 0.0340	TotRmsAbvGrd
0.0075 ± 0.0205	GarageYrBlt
0.0073 ± 0.0163	Neighborhood
0.0069 ± 0.0149	LotFrontage
0.0063 ± 0.0145	YearRemodAdd
0.0062 ± 0.0120	OpenPorchSF
0.0057 ± 0.0239	MasVnrArea

... 54 more ...

Figure 5: Feature importance plots (eli5 plot and normal one) for non-linear model

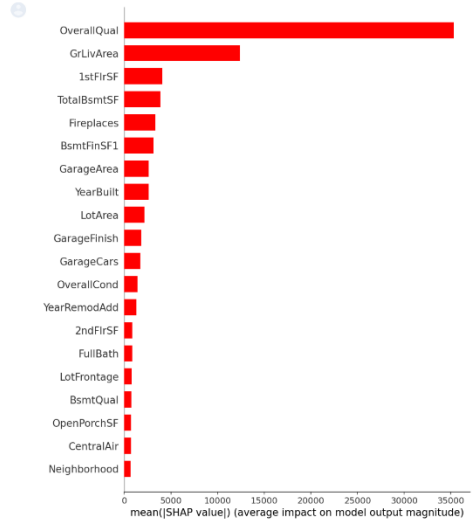


Figure 6: SHAP feature importance plot

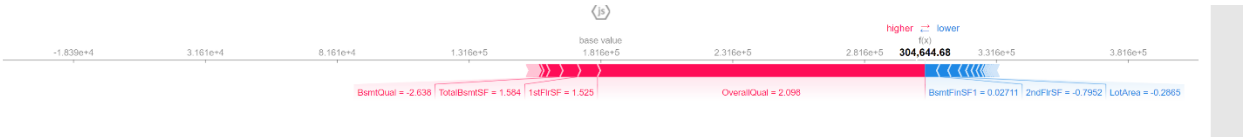


Figure 7: SHAP force plot for the 91st 'SalePrice'

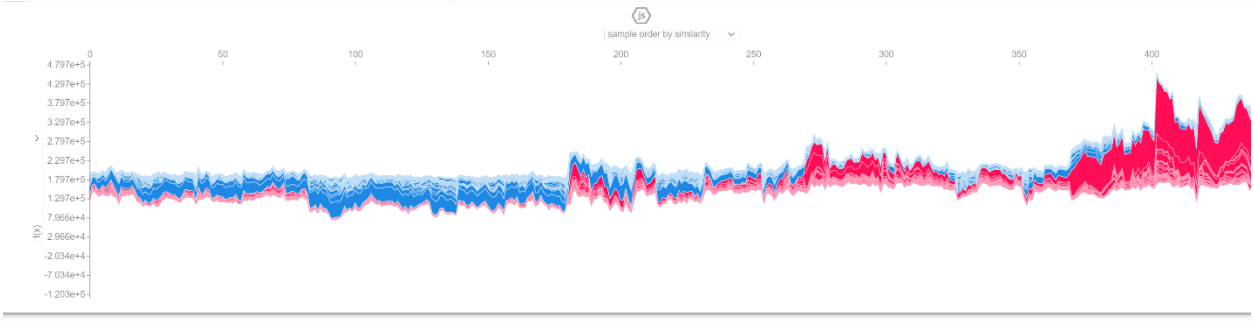


Figure 8: A segment of the 0-1000 'SalePrice' Plot

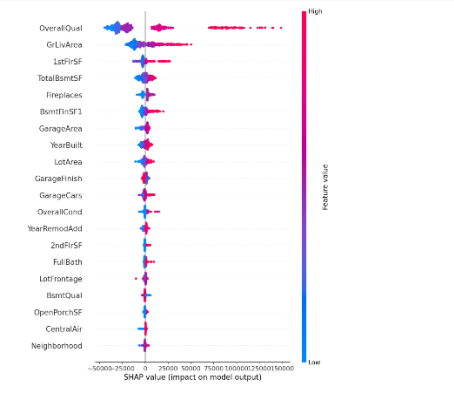


Figure 9: SHAP summary plot

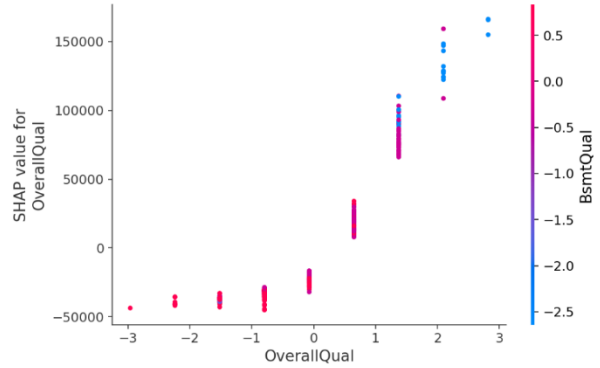


Figure 10: SHAP dependence plot.