REPORT FOR THE KAGGLE PROJECT

I. Introduction and Description of the Kaggle Housing Competition:

The Kaggle House Pricing Competition is a playground for Data Science students with basic knowledge of Machine Learning to challenge themselves by predicting the Sale Price of a house based on a multitude of factors. These factors are included as explanatory values in a dataset provided by Kaggle and cover almost all aspects of a house to be taken into consideration when a purchase is made. Using either R or Python, participants of the competition must build a model using the *train* dataset and use it to predict the Sale Price of houses in the *test* dataset as well as identify the key predictors.

II. Description of the Original Dataset:

The original *train* dataset has 1460 rows and the original *test* dataset has 1459 rows, totaling 2919 rows for the combined dataset. Each of them has 81 columns and all data is extracted from a larger dataset (Ames Housing Dataset – compiled by Dean De Cock)(1) that amasses information on residential property sales in Ames, Iowa from 2006 to 2010.

Out of the 81 columns, 79 of which are predictor variables, or all aspects of a house that may constitute the criteria for house-buyers and influence the price negotiations. The other 2 columns are ID number and SalePrice – ID represents the index number of each observation and SalePrice is the response variable. Of the 79 explanatory variables, 43 are of numeric type whilst the remaining 36 are of categorical type (the data type classification of each variable is subject to change during the data cleaning process). Each row in the dataset represents a house on which data have been collected, meaning that there is a total of 2919 house sales in the entire dataset.

III. Discussion on the Cleaning Process:

I started off the cleaning process by creating a table named Gen_Sum, which stands for General Summary, to summarize the data type, the number and the percentage of missing data for each variable in the dataset. I then proceeded to clean the data on a variable-by-variable basis using a variety of steps as follows:

1. Handling missing values:

(1): "House Prices – Advanced Regression Techniques". Kaggle. URL: https://www.kaggle.com/c/house-prices-advanced-regression-techniques

I employed different approaches to handle missing values in different data types (i.e. numerical vs factorial). Numerical variables are identified as quantitative characteristics of a house with numerical inputs and are exclusive of predictors with numeric values representing qualitative levels (such as MSSubClass). Factorial variables are identified as qualitative characteristics of a house with different quality levels, ordered or unordered.

- Numeric Variables: I replaced missing values in each variable using the mean of its available values. I applied this method to the majority of the numeric predictors with the only exception being LotFrontage where the NA Lot Frontage values are replaced by the mean Lot Frontage of houses with the same MSZoning or the same area classification.
- Categorical Variables: I replaced missing values using the level with the highest frequency among all observations. This approach is applied to all factorial variables.

2. Correcting typos or measurement errors:

For variables with names or levels that are inconsistent with the data description, those names and levels are updated so that the available data match what was described.

3. Dealing with outliers:

I identified outliers by plotting the distribution of variables to see points that have abnormal distance from the rest of the data. The distribution of numeric variables is examined by using the boxplot or the density plot, whereas for the categorical variables I used the bar plot.

4. Data type conversion:

All numeric variables with integer data type are converted to numeric data type and all categorical variables with character data type are converted to factor (ordered or unordered) data type. Variables that use numeric values to represent qualitative levels are converted to factor data type.

5. Transformations:

Transformations are applicable to only numeric variables and are determined by the distribution of the data as well as its skewness. Data with high skewness (skewness > 1 or skewness < -1) are transformed with an appropriate transformation (log, $\log_{10}$, sqrt) whilst data with moderate and slight skewness are maintained. Some variables with high skewness are not transformed in order to be combined into a new variable.

6. Creating new variables:

New variables are created as combinations of similar or related existing variables in an attempt to better account for the response variables. For example, the TotalPorchSF variable is the total square feet of porches in a house and is inclusive of all porch types (Open porch, Enclosed porch, 3-season porch, and Screen porch).

7. Removing variables:

Variables with a value or a level making up at least 95% of the data are removed as they will not provide meaningful insights in predicting SalePrice.

IV. Diagnostic Tests and Prescriptions:

1. Linearity - Residual vs Fitted: The plot displays a linear relationship between the residuals and the fitted values with the distribution of the points forming a straight line as the fitted value increases. The red line almost coincides with the dashed line, suggesting strong linearity.
2. Normality - Normal Q-Q plot: The plot shows that the residuals are quite normally distributed with the majority of the points lying on the dashed line.
3. Non-multicollinearity - Scale-Location Plot: This plot suggests some issues of multi-collinearity. Whilst the distribution of the points is somewhat constant, the points appear to slightly narrow closer to the red line as the fitted value increases.
4. Influential points - Residuals vs Leverage Plot: There are some outliers highlighted by the plots in R but none of which are influential points.
5. Prescriptions for multi-collinearity:

I first found the Variance Inflation Factor of all variables and identified those with the highest VIF values. I then created a table of highly correlated numeric-variable pairs and eliminated the variable with a lower correlation with SalePrice in each pair as well as those with VIF values larger than 5.

V. Comparison of models – Best model selection:

In terms of both model accuracy and interpretability, the Lasso regression model produces the best result as it gives the lowest Kaggle score and has strong interpretability. By minimizing the sum of the RSS (Residuals Sum of Squares) and the shrinkage penalty, the Lasso forces some coefficients to 0, meaning that the predictors included in the model is only a subset of the *cleaned* dataset. As a result, it allows for lower degree of multi-collinearity as well as easier identification of strong explanatory values, which is part of the project goals. This increases interpretability in that the model is able to account for the variance in

Sale Price using robust and independent predictors without inflating the model accuracy with multi-collinearity.

Whilst Ridge regression and Regression Tree with Boosting result in a Kaggle score similar to that of the Lasso regression, Lasso offers higher interpretability. Ridge regression includes all input variables from the *cleaned* dataset, and although the shrinkage penalty does penalize the model as the number of predictors increases, it does not actually eliminate any variable from the model. Such mechanism leads to difficulty in interpreting the model as it retains variables with weak correlation (to the response), which is ineffective in singling out key predictors. Similarly, the Lasso model outperforms the Boosting model because it specifies important predictors by the levels whereas Boosting only highlights key predictors in general.

In comparison to the Least Squares regression models, the Lasso model generates a higher model accuracy because it does not seek to fit all variables in the model but performs feature selection. Consequently, it is easier to interpret the model produced by Lasso than that by Least Squares. Whilst Least Squares Regression is not penalized for its assignment of weight to variables (which may cause overfitting), the Lasso does have the shrinkage penalty. As for the Bagging model and the Random Forest model, they result in the lowest Kaggle scores across all models and naturally fail to compete with the Lasso regression model.

VI. Model interpretation:

Based on the output of the Lasso model, the most important predictors are OverallQual, OverallCond, Functional, Neighborhood, Exterior1st, SaleType, and CentralAir. Specifically, each of these variables has at least one level that ranks in top 10 in terms of the absolute values of coefficients, meaning that the response variable is very sensitive to their changes. The levels are:

- OverallQual_2
- OverallCond_2
- Functional_Sev
- NeighborhoodCrawfor
- Exterior1stBrkComm
- Exterior1stStone
- NeighborhoodStoneBr
- SaleTypeNew
- NeighborhoodNoRidge
- CentralAirY

All of the above levels (except for Exterior1st_BrkComm) have a positive coefficient, indicating positive correlation. As a result, the following statements can be made about the relationship between the predictors and the SalePrice:

- The change in the overall rating of a house's material and finish from 1 (Very Poor) to 2 (Poor) results in the highest increase in Sale Price compared to other levels. On average, the Sale Price of a house with Poor finish is higher than that of house with Poor finish by 69.22% (holding all other variables constant).
- The change in the overall rating of a house's condition from 1 (Very Poor) to 2 (Poor) results in the highest increase in Sale Price compared to changing to other levels. On average, the Sale Price of a house with Poor finish is higher than that of house with Poor finish by 33.28% (holding all other variables constant).
- On average, the Sale Price of a house with Severe Damage is higher than that of a house with Salvage Only (only salvage value left) by 17.91% (holding all other variables constant).
- Holding all other variables constant, the Sale Price of a house in Crawford neighborhood on average is 10.29% higher than that of a house in Bloomington Heights.
- Holding all other variables constant, the Sale Price of a house in Stone Brook neighborhood on average is 7.94% higher than that of a house in Bloomington Heights.
- Holding all other variables constant, the Sale Price of a house in Northridge neighborhood on average is 7.21% higher than that of a house in Bloomington Heights.
- Holding all other variables constant, the Sale Price of a house built with Common Brick is lower than that of a house built with Asbestos Shingles by 8.58%.
- Holding all other variables constant, the Sale Price of a house built with Stone is higher than that of a house built with Asbestos Shingles by 8.24%.
- Holding all other variables constant, the Sale Price of a newly constructed house is higher than that of a Conventional Warranty Deed house by 7.69%.
- Holding all other variables constant, the Sale Price of a house with Central Air Conditioning is higher than that of a house with No Central Air Conditioning by 6.92%.

VII. Final Conclusion – Personal Reflection:

1. What I learned:

This project has been an extremely rewarding learning experience because it allows me to review the knowledge and techniques I learned in BAT 3303 regarding data cleaning and Linear Regression. At the same time, it helps me to better understand the Regression Family methods in terms of their underlying principles/logic as well as the advantages and disadvantages of each method. I also had the opportunity to explore more about similarities and differences among the methods, which, in turn, assisted me in the model implementation and interpretation process.

2. Obstacles I have encountered and my solutions:

a. Mismatch between related variables: When there are variables related to each other such as GarageType, GarageFinish, GarageArea, and GarageCars, the change in one variable requires adjustments in other variables to maintain the consistency for each observation. When I replaced the missing values with meaningful data, I also looked at other related columns to ensure that inconsistency such as a house with Detached Garage has GarageArea = 0 did not happen.

b. Missing level in *test* dataset: When I ran the first plainvanilla model after the initial cleaning, R returned an error message because level 150 for MSSubClass was available only in the *train* dataset – there was no observation with level 150 in *test*. As a result, I replaced all level 150 in *train* with level 50 which is closest in nature to level 150 so that the model can run smoothly.

c. Linear Regression model fitting: After getting the plainvanilla model to work, I had to deal with singularities as they were causing the model to return NAs for some coefficients. I constructed correlation plots for both numeric variables and one-hot encoded factor variables in a separate script to identify highly correlated pairs. I then replaced the correlated levels or values with other levels/values to prevent singularities. This was added to the initial cleaning because it was too complicated to replace data after factorizing variables.

3. Overall reflection on the experience:

Overall, this has been a challenging yet exciting experience. Whilst it is not extremely complex as we were provided with all the tools and knowledge to work on it, the different cleaning stages and Machine Learning methods involved created endless possibilities for implementation and troubleshooting. At the same time, this experience enabled me to step outside of my comfort zone and tackle new challenges. I am happy with what I have gained both in terms of knowledge and skills through the project.