ISE – 529: Predictive Analytics

Project Report

by

Khushi Gandhi
(6401790989)
kjgandhi@usc.edu

Table of Contents

[all titles are hyperlinked.]

Table of Figures

[all titles are hyperlinked.]

1. Introduction

The chosen dataset for my project is **diamonds.csv**. I chose the diamonds dataset for this project because it contains a variety of rich data that is ideal for delving into different parts of data analysis. It has around 54,000 entries for diamonds, with a range of characteristics such as **carat, cut, color, clarity, depth, table, price, and dimensions (x, y, z).** These characteristics provide a wide and fascinating range of variables to investigate, assisting us in comprehending the causes influencing diamond prices. I want to use this dataset to find important patterns and connections that can guide price decisions and help us comprehend the diamond market better.

This project's main goal is to conduct a thorough study of the diamonds dataset to get important knowledge about the variables influencing diamond pricing. By comprehending these elements, we can add to the body of information about the diamond market and offer practical suggestions for diamond pricing techniques.

For my project on regression analysis, I have implemented and compared the performance of seven different models: **Simple Linear Regression, Multiple Linear Regression, Decision Tree Regressor, Random Forest Regressor, L1 (Lasso) Regression, L2 (Ridge) Regression, and MLP (Multi-Layer Perceptron) Regressor.** Each model's performance was evaluated using two key metrics: **R^2 (Coefficient of Determination) and MSE (Mean Squared Error).** This comprehensive analysis allowed me to identify the most effective model for predicting outcomes based on the given dataset.

2. Literature Review

Regression analysis is a fundamental statistical tool used to explore relationships between variables, often aiming to uncover causal effects, such as the impact of price increases on demand. This literature survey examines the methodologies and applications of regression analysis across various disciplines, highlighting its role in quantifying relationships and informing decision-making processes. [1]

Data preprocessing is a critical step in preparing data for data mining (DM) tasks, ensuring that raw data is transformed into a clean, integrated, and normalized format suitable for analysis. It encompasses several techniques: data cleaning involves correcting errors and handling missing or noisy data; data transformation consolidates and enhances data for efficient mining; data integration merges data from multiple sources while maintaining consistency; data normalization standardizes data attributes for fair analysis; and missing data imputation fills in missing values intelligently. These processes collectively aim to optimize data quality and structure, facilitating accurate and meaningful outcomes from DM algorithms. Without proper preprocessing, DM algorithms may not operate correctly or produce reliable results, highlighting the crucial role of preprocessing in the data analysis pipeline. [2]

Marill [3] states that Linear regression is a mathematical technique for describing relationships between variables using straight-line functions, making inferences from data and predicting future outcomes. Simple linear regression models the relationship between a single outcome (dependent) variable and a single predictor (independent) variable using a straight-line function. Tranner, Murphy, Elliot and Pampaka [4] states that Multiple linear regression extends simple linear regression by incorporating multiple explanatory variables, assuming a linear relationship between the response variable and a combination of these predictors.

Random forest is a versatile supervised machine learning algorithm that leverages ensemble learning by combining multiple decision trees built on various subsets of a dataset, thereby enhancing predictive accuracy through majority voting, and is applicable to both regression and classification problems. [5]

Lasso Regression imposes a penalty equal to the absolute value of the coefficients' magnitudes. Consequently, some coefficients may become zero, leading the model to disregard the corresponding features. Ridge Regression adds a penalty equal to the square of the coefficients' magnitudes. As a result, all coefficients are reduced by the same factor, but unlike the L1 method, none are eliminated.

Artificial neural networks (ANNs), inspired by biological neural networks, consist of interconnected nodes arranged in layers, each with weighted connections, and are designed to model the relationship between input data and expected output. Training ANNs involves determining optimal weights for these connections, a process that is complex and time-consuming due to the large number of weights involved. [6]

3. Data Description

This dataset contains 53940 entries and 10 features about diamonds, which include their physical attributes and its price.

FEATURES:
carat: Carat weight of the diamond
cut: Quality of the cut (e.g., Ideal, Premium, Good)
color: Diamond color, with D being the best and J the worst
clarity: Diamond clarity (e.g., SI2, SI1, VS1, VS2)
depth: Total depth percentage (depth divided by the average diameter)
table: Width of the top of the diamond relative to the widest point
x: Length of the diamond in mm
y: Width of the diamond in mm
z: Depth of the diamond in mm

TARGET VARIABLE:
price: price of the diamond.

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75
...	...	...	...	...	...	...	...	...	...	...
53935	0.72	Ideal	D	SI1	60.8	57.0	2757	5.75	5.76	3.50
53936	0.72	Good	D	SI1	63.1	55.0	2757	5.69	5.75	3.61
53937	0.70	Very Good	D	SI1	62.8	60.0	2757	5.66	5.68	3.56
53938	0.86	Premium	H	SI2	61.0	58.0	2757	6.15	6.12	3.74
53939	0.75	Ideal	D	SI2	62.2	55.0	2757	5.83	5.87	3.64

53940 rows × 10 columns

Figure 1: diamonds.csv

4. Data Preprocessing

It is the most important data mining tasks which include the preparation and transformation of data. It aims to reduce the data in size, find relations between data, normalize data, remove outliers, and extract the important features. [2]

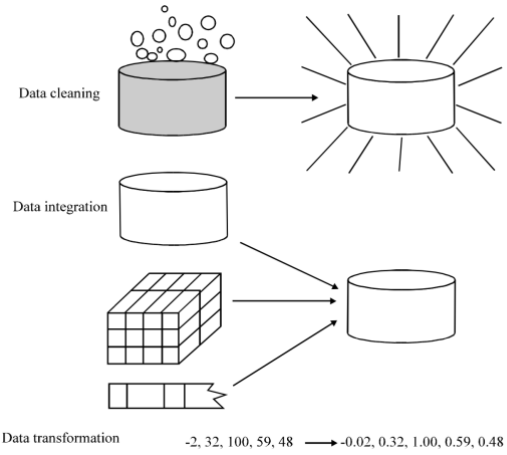


Figure 2: Data Processing (Knowledge Discovery Process)

4.1 Checking for missing values

The main purpose is to identify any missing values in the dataset. It was found that there were no missing values found in any columns.

4.2 Label Encoding

The main purpose is to convert categorical variables into numeric form for modeling. The columns affected by the label encoding are 'cut', 'color', 'clarity'.

4.3 Handling Missing Values

The main purpose is to ensure there are no missing values that could affect model performance.

4.4 Dropping Duplicates

The main purpose is to remove duplicate rows that might skew the analysis.

4.5 Removing Outliers

The main purpose is to remove extreme values that could negatively impact the model. Here, Removal of outliers is done by using the Interquartile Range (IQR) method for numeric columns (excluding 'cut', 'color', 'clarity').

5. Exploratory Data Analysis (EDA)

5.1 Numerical Summary

The numerical summary highlights the data cleaning process effectively removes outliers and reduced variability in the dataset, leading to more consistent and reliable summary statistics. This cleaned dataset is likely to yield more accurate and meaningful insights during further analysis.

5.1.1 Dataset before cleaning

	carat	cut	color	clarity	depth	table	price	x	y	z
count	53940.000000	53940	53940	53940	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000	53940.000000
unique	NaN	5	7	8	NaN	NaN	NaN	NaN	NaN	NaN
top	NaN	Ideal	G	SI1	NaN	NaN	NaN	NaN	NaN	NaN
freq	NaN	21551	11292	13065	NaN	NaN	NaN	NaN	NaN	NaN
mean	0.797940	NaN	NaN	NaN	61.749405	57.457184	3932.799722	5.731157	5.734526	3.538734
std	0.474011	NaN	NaN	NaN	1.432621	2.234491	3989.439738	1.121761	1.142135	0.705699
min	0.200000	NaN	NaN	NaN	43.000000	43.000000	326.000000	0.000000	0.000000	0.000000
25%	0.400000	NaN	NaN	NaN	61.000000	56.000000	950.000000	4.710000	4.720000	2.910000
50%	0.700000	NaN	NaN	NaN	61.800000	57.000000	2401.000000	5.700000	5.710000	3.530000
75%	1.040000	NaN	NaN	NaN	62.500000	59.000000	5324.250000	6.540000	6.540000	4.040000
max	5.010000	NaN	NaN	NaN	79.000000	95.000000	18823.000000	10.740000	58.900000	31.800000

Figure 3: Numerical Summary before cleaning

5.1.2 Dataset after cleaning

	carat	cut	color	clarity	depth	table	price	x	y	z
count	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000	46425.000000
mean	0.698218	2.624211	2.535401	3.886807	61.803673	57.239968	3001.342768	5.522105	5.527182	3.414310
std	0.361951	0.943223	1.685153	1.729550	1.077656	2.013965	2599.798006	0.964303	0.958496	0.597123
min	0.200000	0.000000	0.000000	0.000000	59.000000	52.000000	326.000000	3.730000	3.680000	1.410000
25%	0.370000	2.000000	1.000000	2.000000	61.200000	56.000000	881.000000	4.630000	4.630000	2.850000
50%	0.590000	2.000000	3.000000	4.000000	61.900000	57.000000	2007.000000	5.400000	5.410000	3.330000
75%	1.010000	3.000000	4.000000	5.000000	62.500000	59.000000	4524.000000	6.390000	6.390000	3.960000
max	2.000000	4.000000	6.000000	7.000000	64.600000	63.500000	11047.000000	8.270000	8.270000	5.080000

Figure 4: Numerical Summary after cleaning

5.2 Categorical Summary

The categorical summary highlights the effectiveness of the data cleaning process. There are fewer entries, less variability, and a more uniform representation of categorical variables in the cleaned dataset. These changes likely contribute to more reliable and accurate analyses. The central tendencies remain consistent, indicating that the core characteristics of the data have been preserved despite the cleaning process.

5.2.1 Dataset before cleaning

	carat	cut	color	clarity	depth	table	price	x	y	z
count	53940.0	53940	53940	53940	53940.0	53940.0	53940	53940.00	53940.00	53940.0
unique	273.0	5	7	8	184.0	127.0	11602	554.00	552.00	375.0
top	0.3	Ideal	G	SI1	62.0	56.0	605	4.37	4.34	2.7
freq	2604.0	21551	11292	13065	2239.0	9881.0	132	448.00	437.00	767.0

Figure 5: Categorical Summary before cleaning

5.2.2 Dataset after cleaning

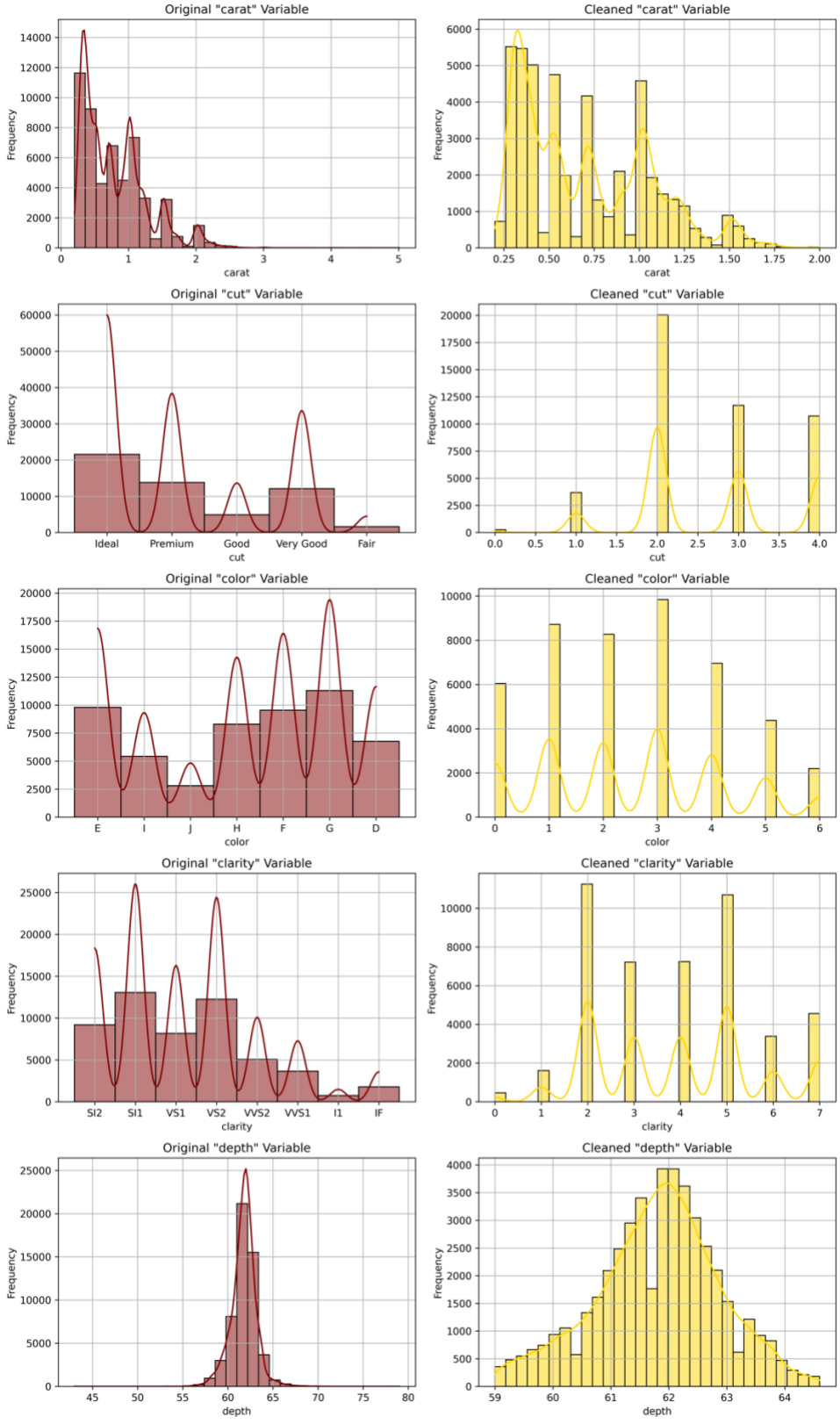
	carat	cut	color	clarity	depth	table	price	x	y	z
count	46425.0	46425	46425	46425	46425.0	46425.0	46425	46425.00	46425.00	46425.0
unique	175.0	5	7	8	57.0	104.0	8501	435.00	432.00	285.0
top	0.3	2	3	2	62.0	56.0	605	4.37	4.34	2.7
freq	2547.0	20037	9844	11245	2058.0	9084.0	131	439.00	435.00	757.0

Figure 6: Categorical Summary after cleaning

5.3 Graphical Summary

- **Histograms**: These histograms help to understand the frequency and spread of these variables before and after data cleaning. The histograms highlight the distribution of key diamond features before and after data cleaning. The cleaned data has more refined, balanced, and standardized distributions, indicating a thorough cleaning process that increased data quality. This analysis emphasizes the significance of data cleaning in producing accurate and dependable results.
- **Boxplots**: These boxplots help to understand how these factors influence diamond pricing. The box plots for both the original and cleaned data show that cut, color, and clarity have a considerable impact on diamond prices. The cleaned data offers a more refined and consistent view, with fewer outliers and clearer trends. This analysis emphasizes the significance of data cleaning in increasing the reliability and understanding of the results.
- **Correlation Matrix**: High positive values indicate strong positive correlations, while high negative values indicate strong negative correlations. Values close to zero suggest weak or no correlation. The correlation matrices demonstrate the key elements impacting diamond pricing. Both original and cleaned data indicate that carat size and dimensions (x, y, z) are the key price factors. The cleaned data provides a more detailed view, including categorical variables that, while less significant, provide a more nuanced knowledge of diamond quality factors such as cut, color, and clarity.
- **Scatterplots**: Pair plots provide a visual overview of the relationships between variables, highlighting potential linear or non-linear relationships. The scatter plots indicate that carat, x, y, and z have a strong linear relationship with price, while other variables show weaker relationships. The cleaned data's pair plots have reduced noise and clearer trends, aiding in better model interpretation and accuracy.
- **Lag Plots**: Lag plots for numerical variables in the cleaned dataset show that there is no significant autocorrelation, indicating that the observations are independent of each other. This lack of autocorrelation supports the use of standard regression models without needing to account for time-series dependencies.
- **QQ Plots**: The QQ plots for numerical variables in the cleaned dataset indicate that the data follows a normal distribution reasonably well, with some deviations at the tails. This normality assumption supports the application of linear regression models, which rely on the normality of residuals for valid inference. Deviations from the diagonal line indicate departures from normality.

5.3.1 Histograms



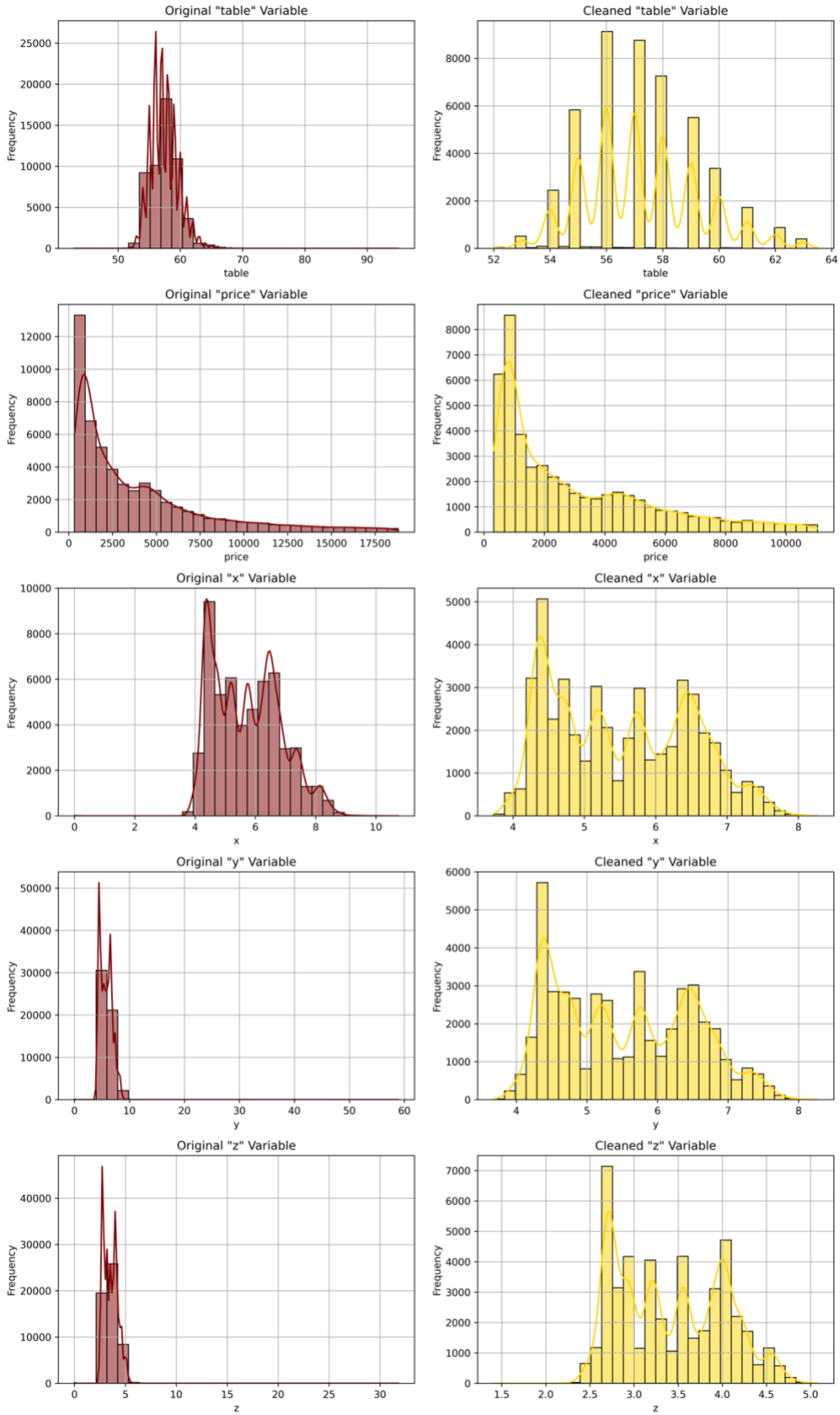


Figure 7: Histogram for each feature (before and after cleaning)

5.3.2 Boxplots

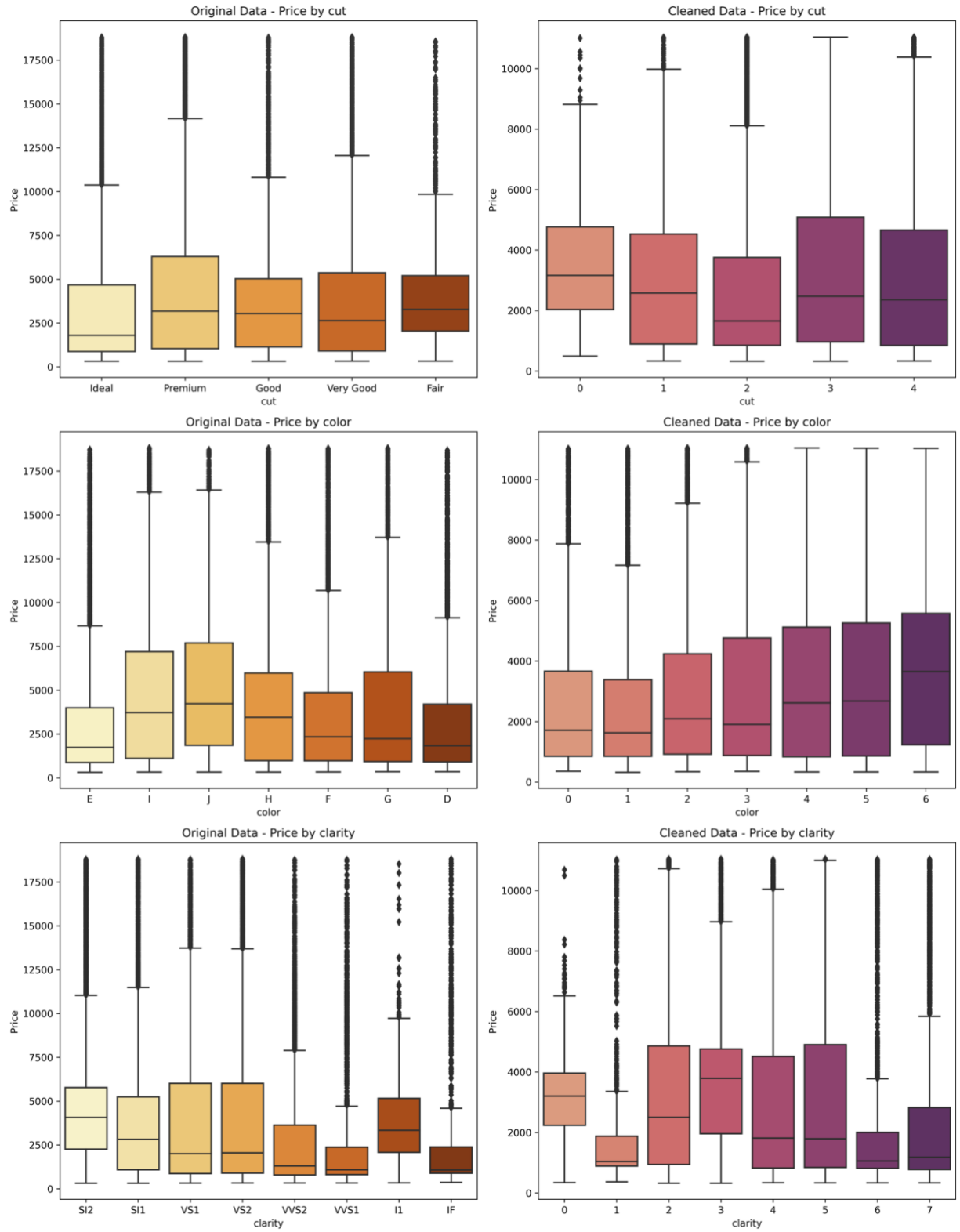
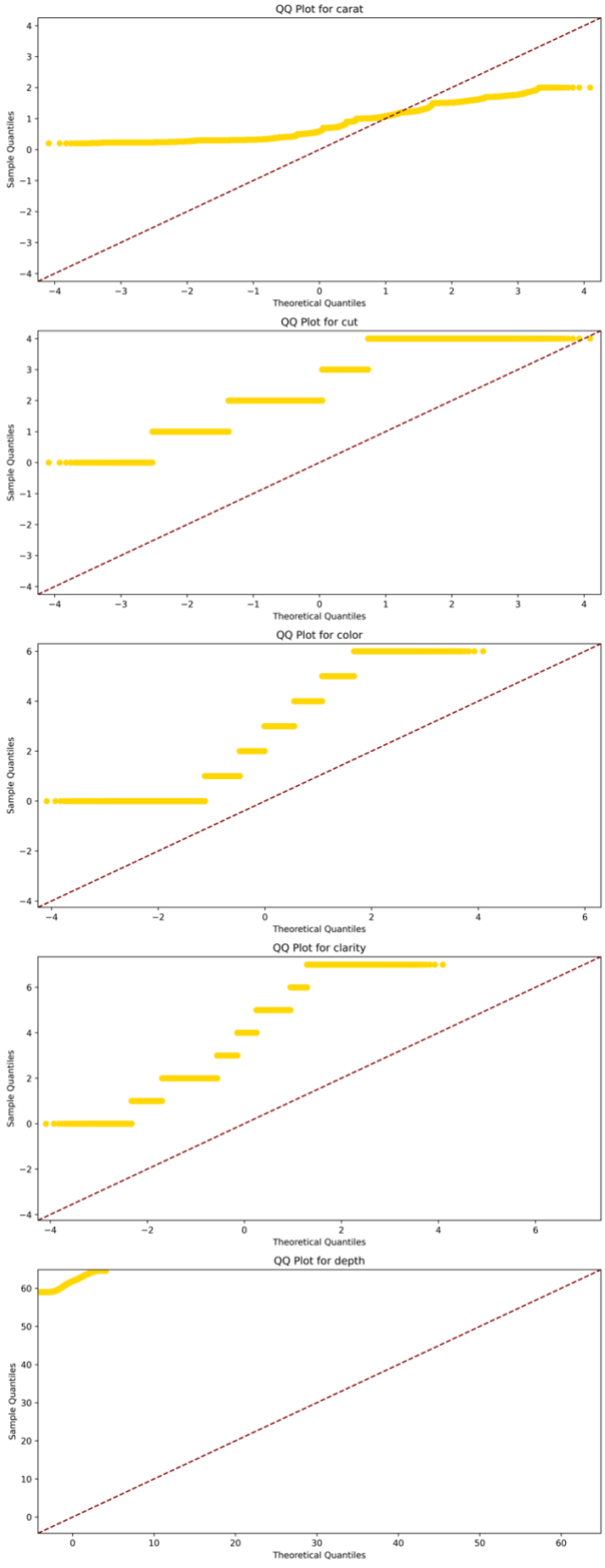


Figure 8: Boxplots for categorical features (before and after cleaning)

5.3.3 QQ Plots


QQ Plot for carat
Sample Quantiles
Theoretical Quantiles
QQ Plot for cut
Sample Quantiles
Theoretical Quantiles
QQ Plot for color
Sample Quantiles
Theoretical Quantiles
QQ Plot for clarity
Sample Quantiles
Theoretical Quantiles
QQ Plot for depth
Sample Quantiles
Theoretical Quantiles

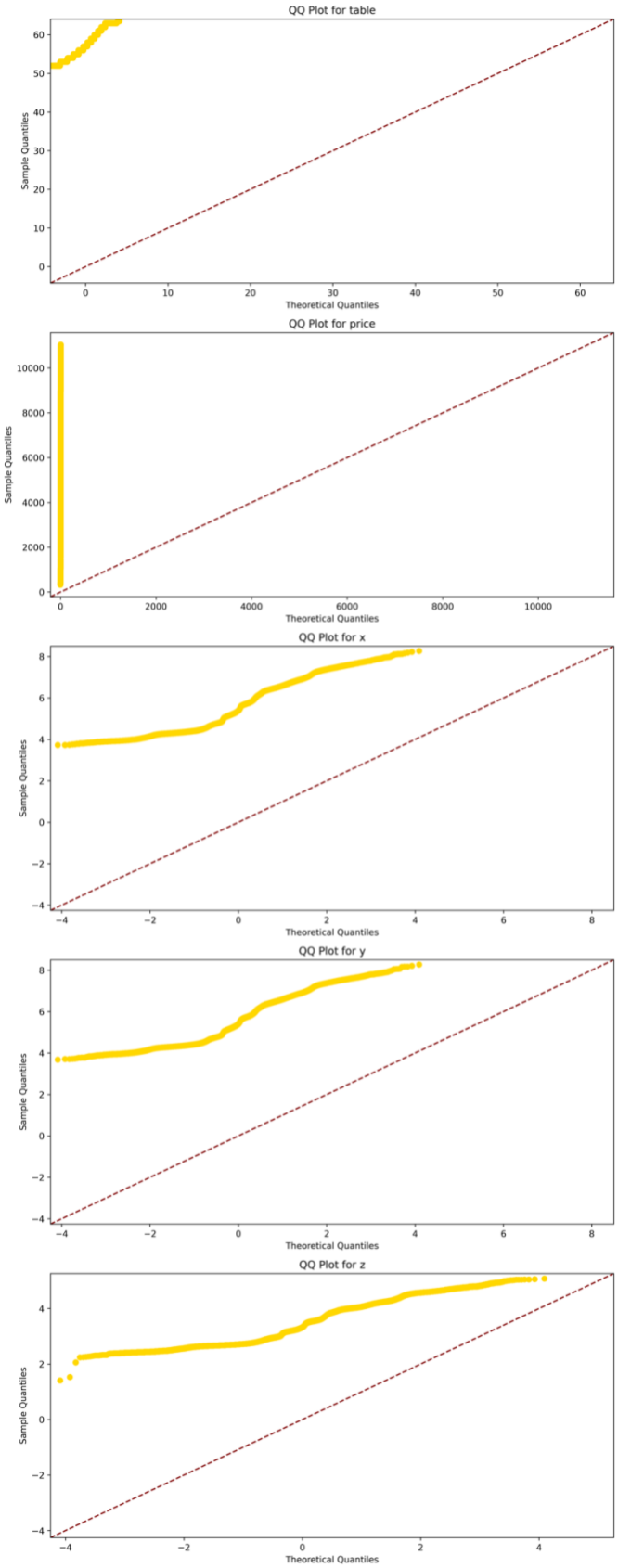


Figure 9: QQ Plots for each feature

5.3.4 Lag Plots

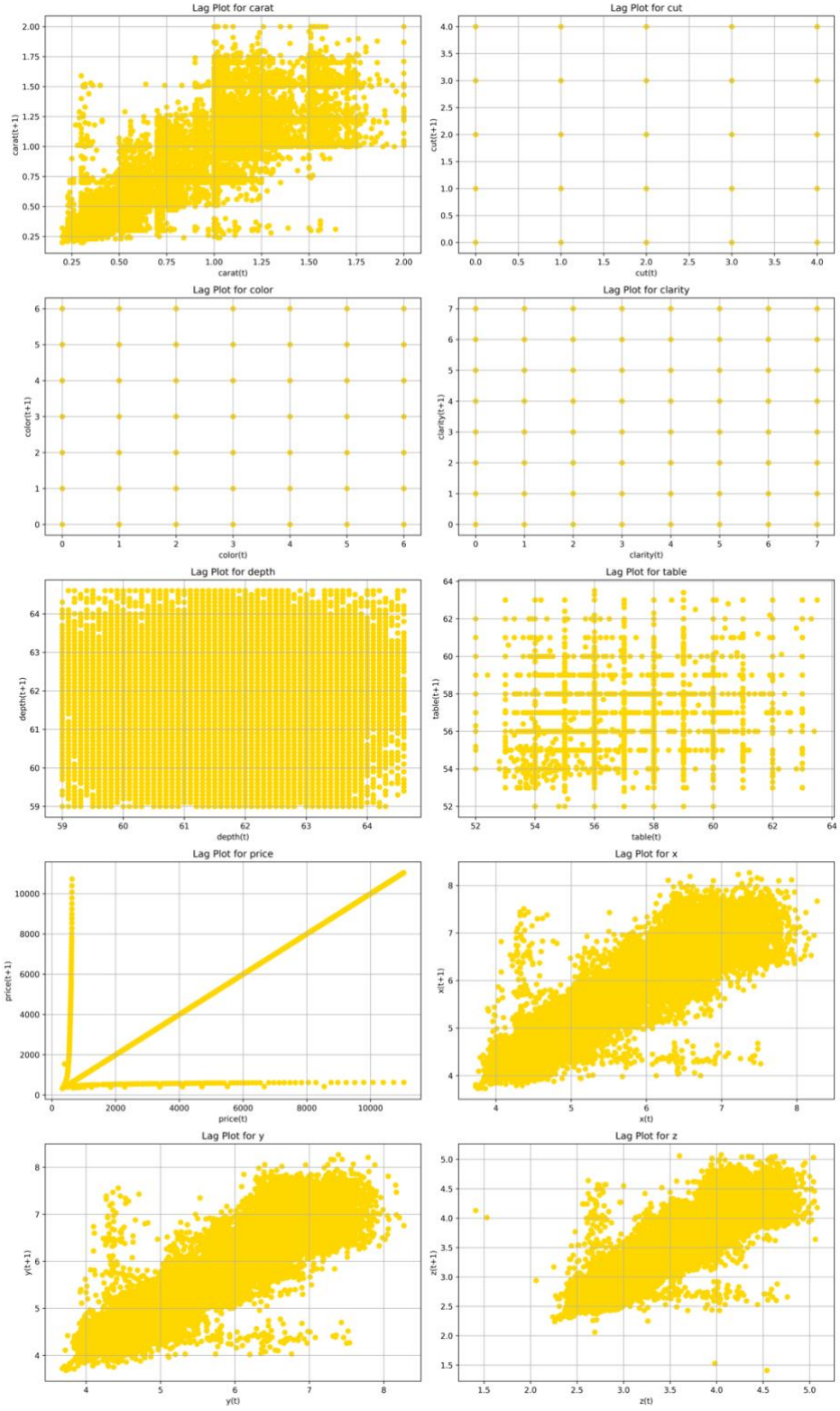


Figure 10: Lag Plots for each feature

5.3.5 Scatter Plots

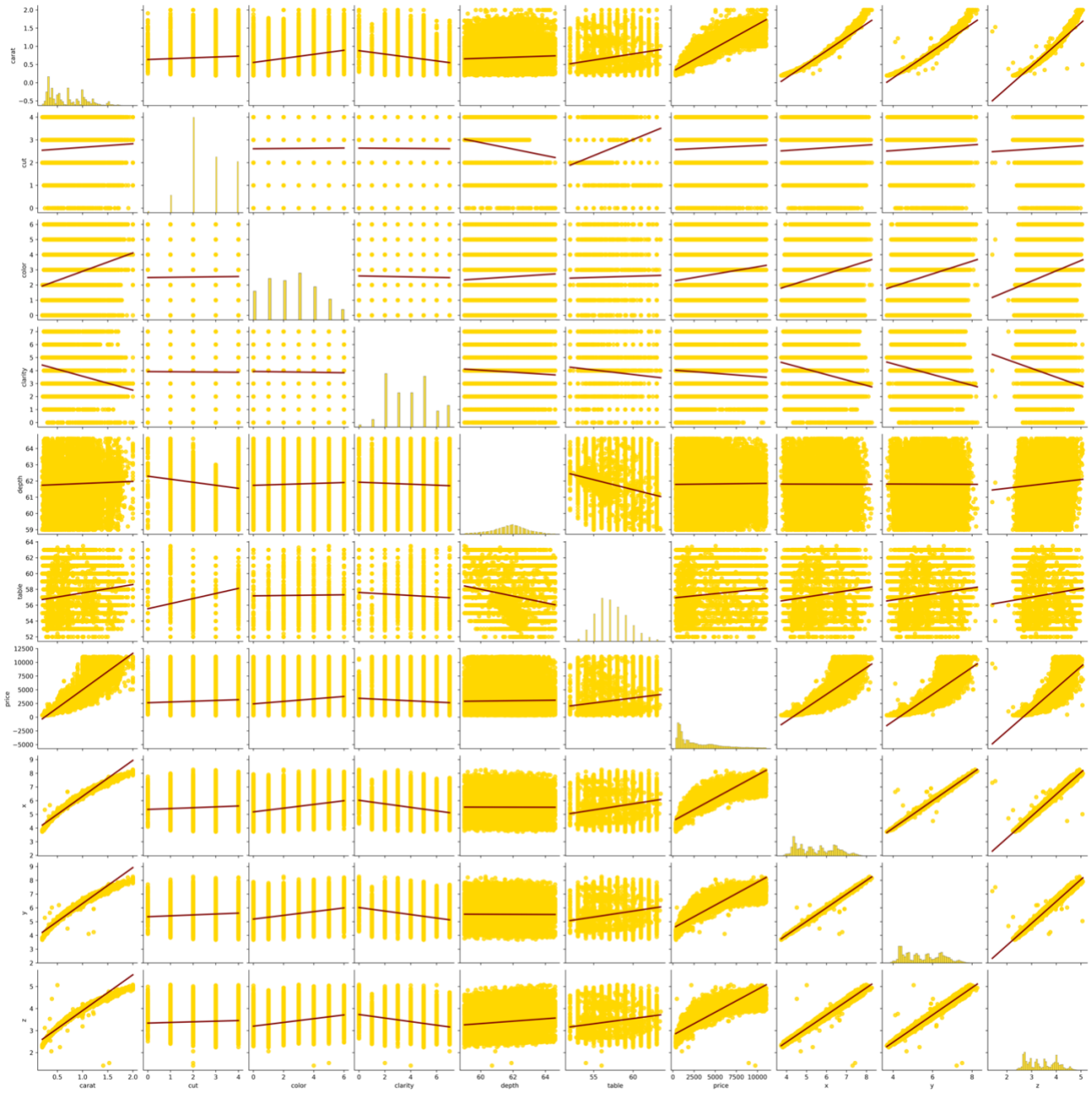


Figure 11: Pair Plot

5.3.6 Correlation Matrix

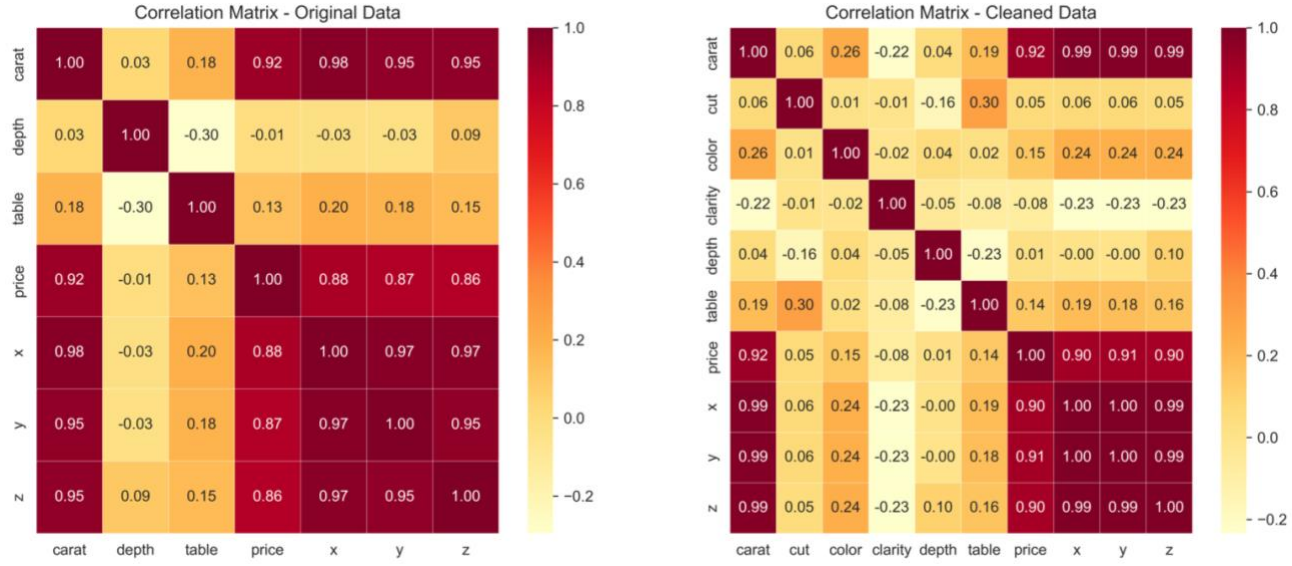


Figure 12: Correlation Matrix (before and after cleaning)

6. Regression Analysis

6.1 Feature Selection Methods

[3]

6.1.1 Correlation

It is used as a measure for quantifying linear dependence between two continuous variables X and Y. Its value varies from -1 to +1. For **different thresholds**, features were recorded.

	threshold	features
0	0.1	[carat, color, table, x, y, z]
1	0.2	[carat, x, y, z]
2	0.3	[carat, x, y, z]

Figure 13: Correlation Feature Selection for different thresholds

6.1.2 ANOVA

ANOVA stands for Analysis of variance. It is like LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not. For this project, ANOVA features for different K values were recorded. On basis of the R^2 scores, **K = 8** has the best performance (Linear Regression was implemented).

	K	R-squared	Selected features
0	5	0.874028	[carat, clarity, x, y, z]
1	6	0.887211	[carat, color, clarity, x, y, z]
2	7	0.888665	[carat, color, clarity, table, x, y, z]
3	8	0.888674	[carat, cut, color, clarity, table, x, y, z]
4	9	0.888652	[carat, cut, color, clarity, depth, table, x, ...

Figure 14: ANOVA Features for different K values

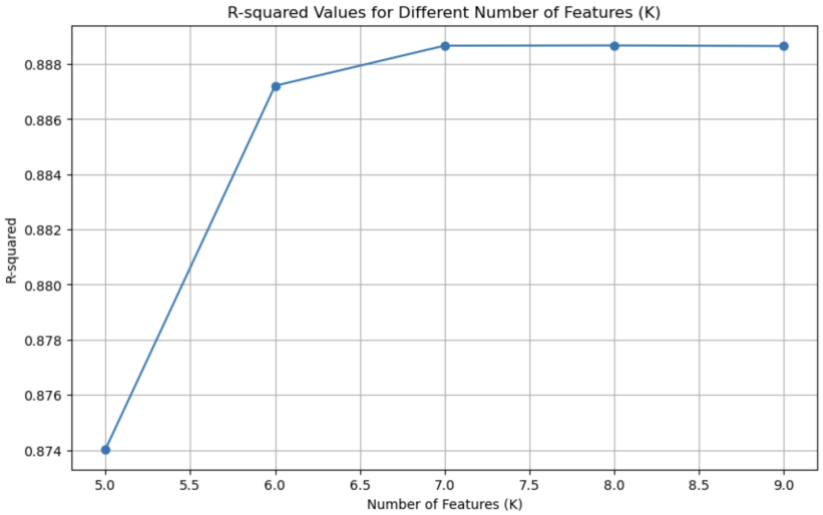


Figure 15: Plot for ANOVA Features with different K values

6.1.3 Forward Selection

Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model. The selected features were **'y', 'carat', 'color', 'clarity', 'z', 'table', 'x', 'depth'.**

6.1.4 Backward Selection

In backward elimination, we start with all the features and removes the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features. The selected features were **'carat', 'color', 'clarity', 'depth', 'table', 'x', 'y', 'z'.**

Method	Train R^2	Test R^2	Difference	Overfit
Correlation	0.8616	0.8565	0.0051	No
ANOVA	0.8780	0.8740	0.0040	No
Forward Selection	0.8919	0.8886	0.0032	No
Backward Selection	0.8919	0.8886	0.0032	No

6.2 Linear Regression

6.2.1 Simple Linear Regression

In Simple Linear regression (SLR), there is **only one independent variable** used to predict the dependent variable. It is called "simple" because it involves a **single** predictor variable, as opposed to Multiple Linear Regression (MLR), which involves multiple predictor variables.

Train R^2	Test R^2	Difference	Overfit
0.892	0.889	0.003	No

6.2.2 Multiple Linear (Polynomial) Regression

A form of regression analysis in which the relationship between the independent variable x and the dependent variable y is modeled as an N degree polynomial. It captures the **non-linear relationship** between the features and the target variables. It performs well for complex datasets where Linear Regression would fail. For this project, Multiple Linear Regression for **all features, ANOVA features only and single features** was implemented. The results are documented as graphs and table as shown below along with R^2 scores.

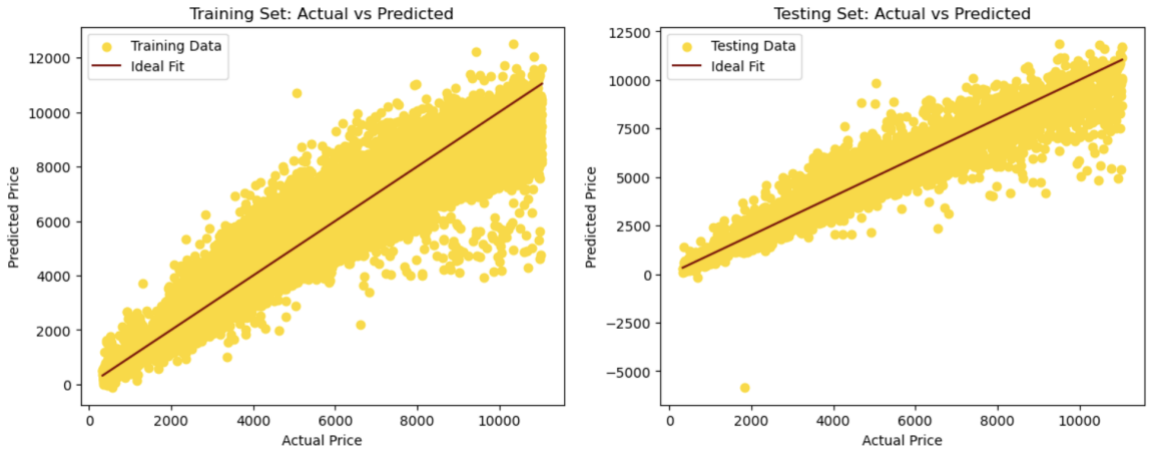


Figure 16: Polynomial Regression - degree: 3 (All Features)

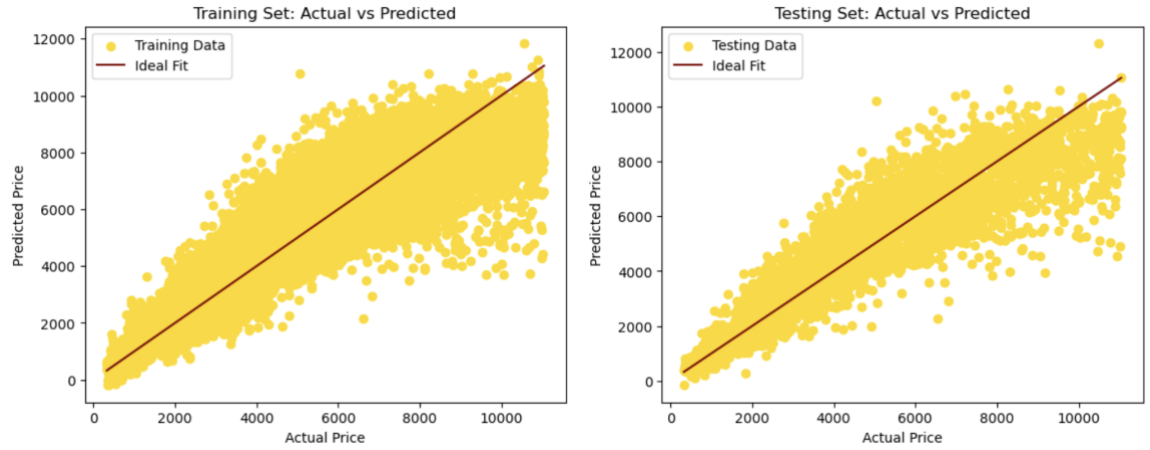


Figure 17: Polynomial Regression - degree: 3 (ANOVA Features)

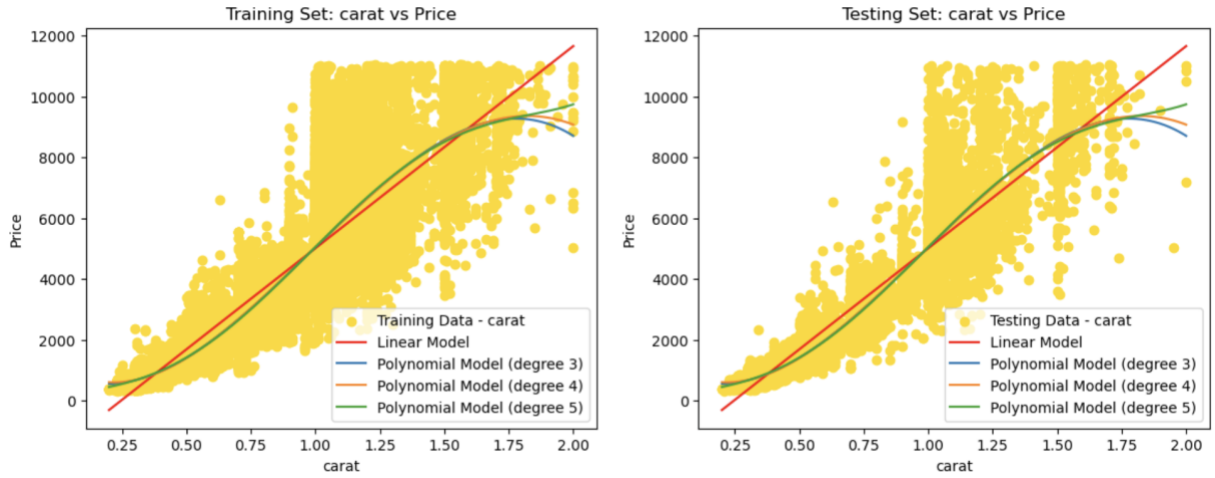


Figure 18: Polynomial Regression (carat)

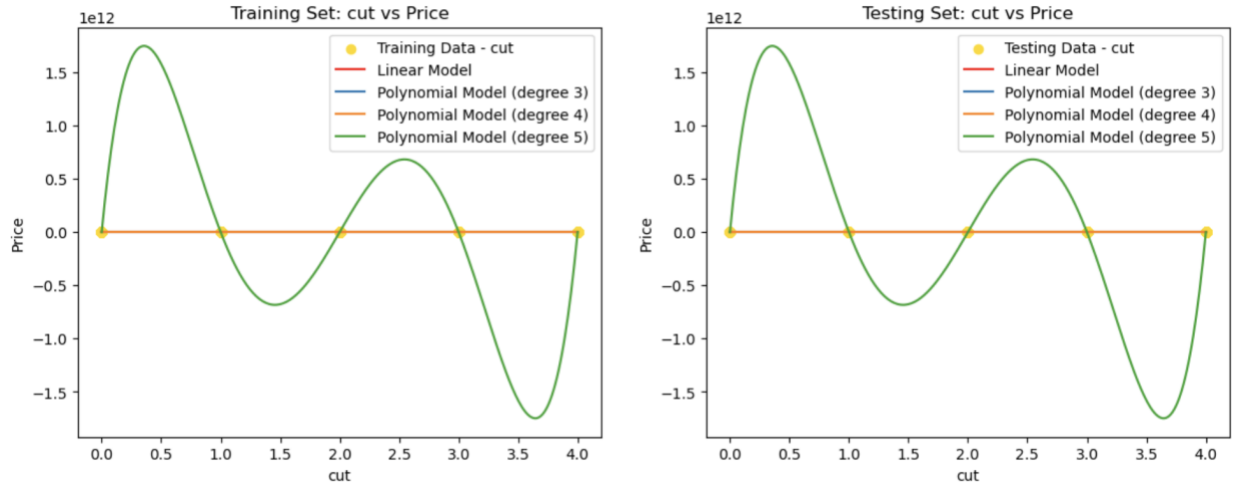


Figure 19: Polynomial Regression (cut)

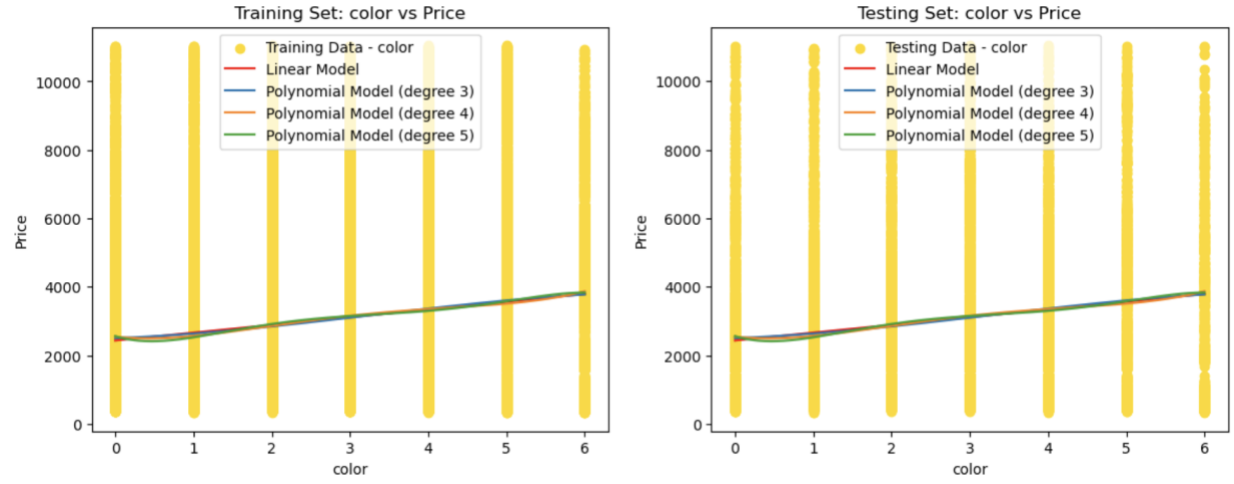


Figure 20: Polynomial Regression (color)

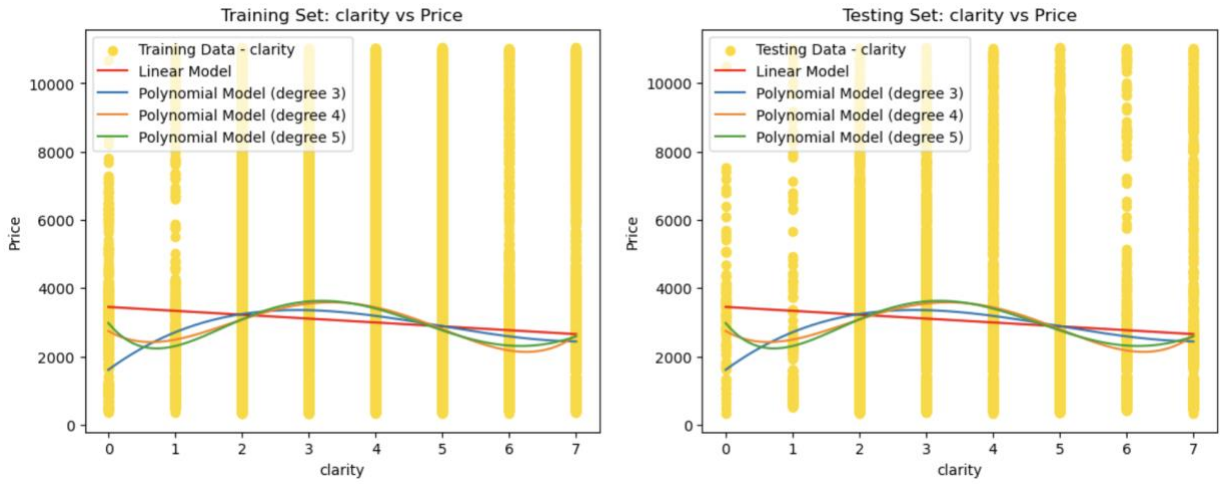


Figure 21: Polynomial Regression (clarity)

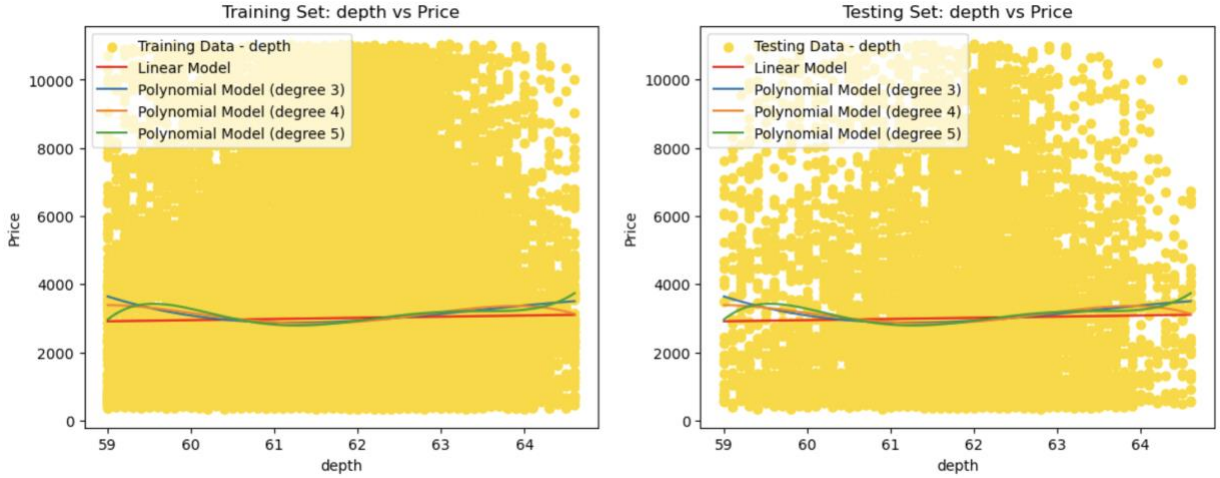


Figure 22: Polynomial Regression (depth)

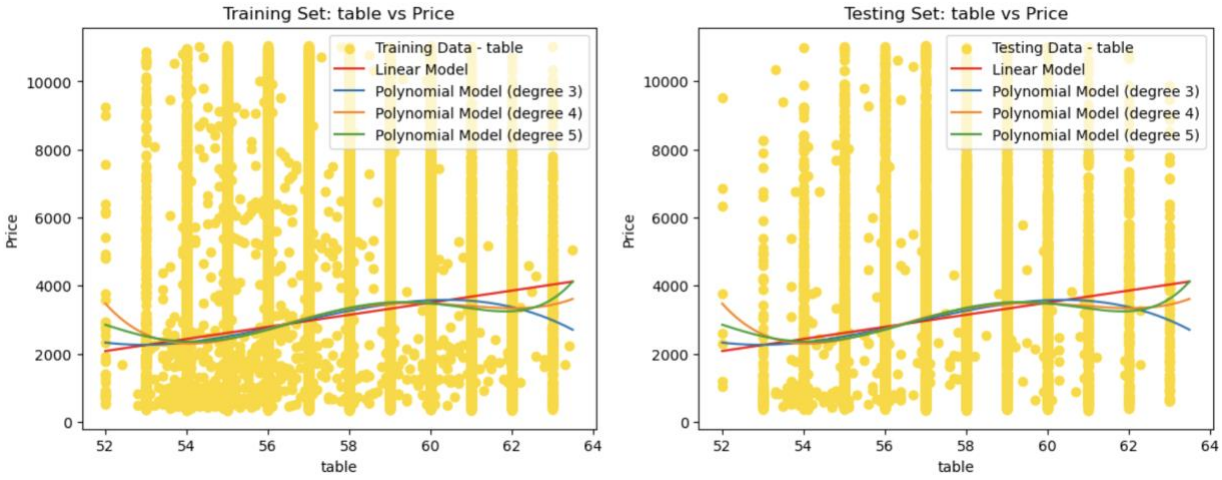


Figure 23: Polynomial Regression (table)

Figure 24: Polynomial Regression (x)

Figure 25: Polynomial Regression (y)

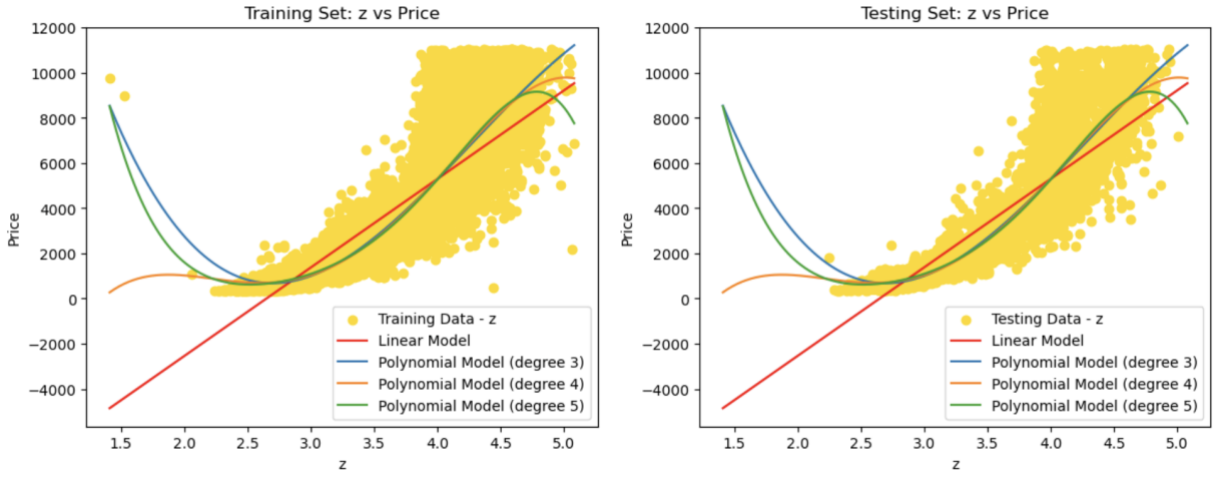


Figure 26: Polynomial Regression (z)

Method	Train R^2	Test R^2	Difference	Degree
All Features	0.944	0.938	0.005	3
ANOVA Features	0.91	0.907	0.003	3
carat	0.867	0.861	0.006	5
cut	0.001	0.01	0.001	4
color	0.022	0.022	0.0	5
clarity	0.024	0.022	0.002	5
depth	0.005	0.004	0.001	5
table	0.024	0.025	-0.002	5
x	0.866	0.859	0.007	5
y	0.871	0.865	0.007	5
z	0.862	0.856	0.005	5

6.3 Decision Tree Regressor

Regression trees are decision trees in which the target variables can take continuous values instead of class labels in leaves. Regression trees use modified split selection criteria and stopping criteria. By using a regression tree, you can explain the decisions, identify possible events that might occur, and see potential outcomes. The analysis helps you determine what the best decision would be. Implementation is done by dividing the data into subsets. Regression tree models use nodes, branches, and leaves. [4]

By performing GridSearchCV for the hyperparameters: **'max_depth': 8, 'min_samples_leaf': 14, 'min_samples_split': 10** were found. This gives us a test R^2 score of 0.9423.

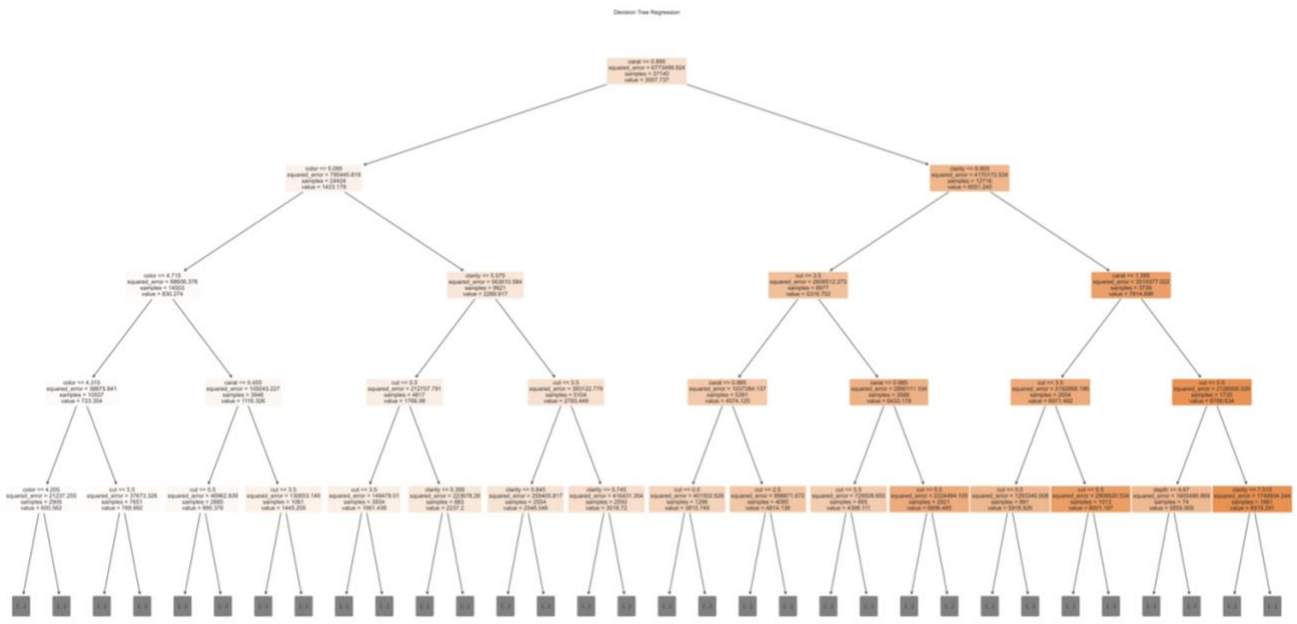


Figure 27: Decision Tree Regressor (shown: 4 levels)

Train R^2	Test R^2	Difference	Overfit
0.9466	0.9423	0.0043	No

6.4 Random Forest Regressor

The Random Forest algorithm combines ensemble learning methods with the decision tree framework to create multiple randomly drawn decision trees from the data, averaging the results to output a new result that often leads to strong predictions/classifications. Here, by performing GridSearchCV for the hyperparameters, **'max_depth': 10, 'min_samples_leaf': 10, 'min_samples_split': 10, 'n_estimators': 100** were found. This gives us a test R^2 score of 0.9446.

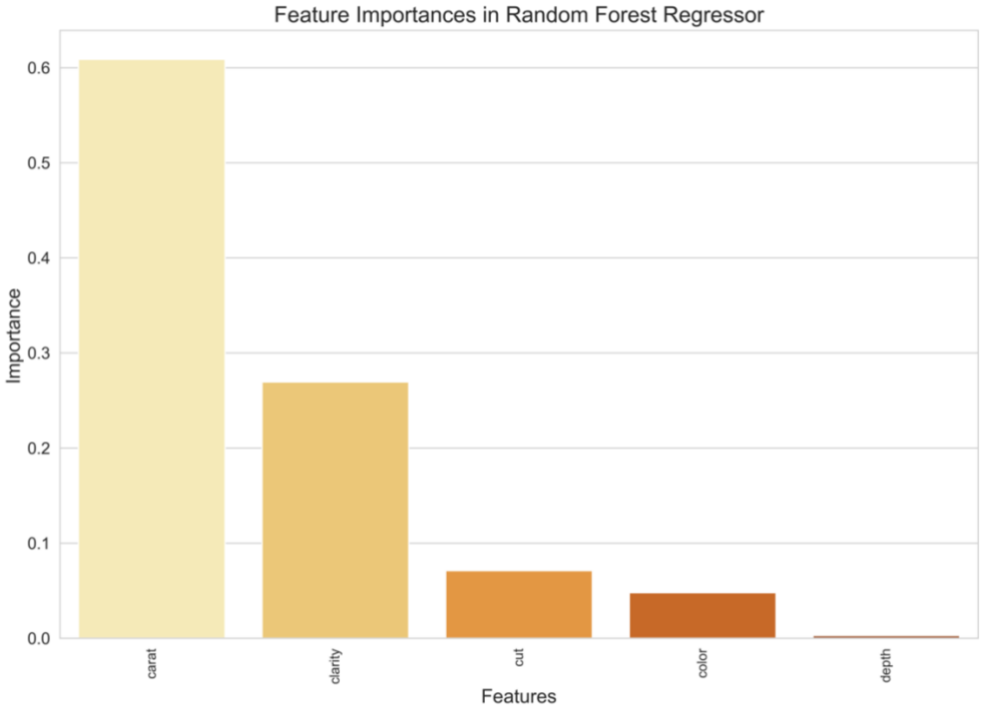


Figure 28: Feature Importance for Random Forest Regressor

Train R^2	Test R^2	Difference	Overfit
0.9521	0.9446	0.0075	No

6.5 Regularization Methods

[5]

The primary goal of regularization is to reduce the model's complexity, enhancing its ability to generalize to new, unseen data, and thereby improving its performance on datasets beyond the training set. Here, L-1 (Lasso) and L-2 (Ridge) are implemented.

6.5.1 L-1 Regularization (Lasso)

This method imposes a penalty equal to the absolute value of the coefficients' magnitudes. Consequently, some coefficients may become zero, leading the model to disregard the corresponding features. This approach is particularly useful for feature selection. The effect of Lasso co-efficients can be explained as:

- As alpha (λ) increases, some coefficients shrink to exactly zero, effectively performing feature selection. This means Lasso Regression can eliminate some features from the model.
- Lasso eliminates less important features as alpha increases, making the model simpler and more interpretable. For instance, coefficients for features like "depth" and "table" become zero at higher alpha values, indicating they are not as influential as other features.

- By setting some coefficients to zero, Lasso Regression simplifies the model, which can improve interpretability and potentially enhance generalization by focusing only on the most significant features.

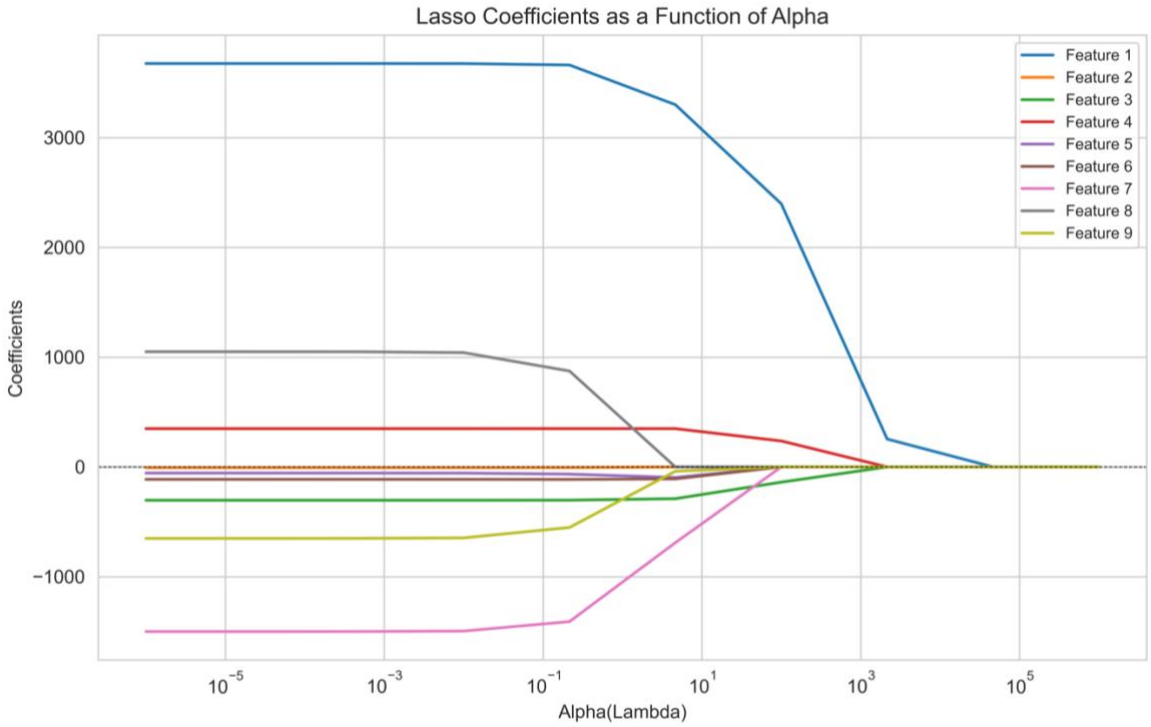


Figure 29: Effect of α for Lasso Regression

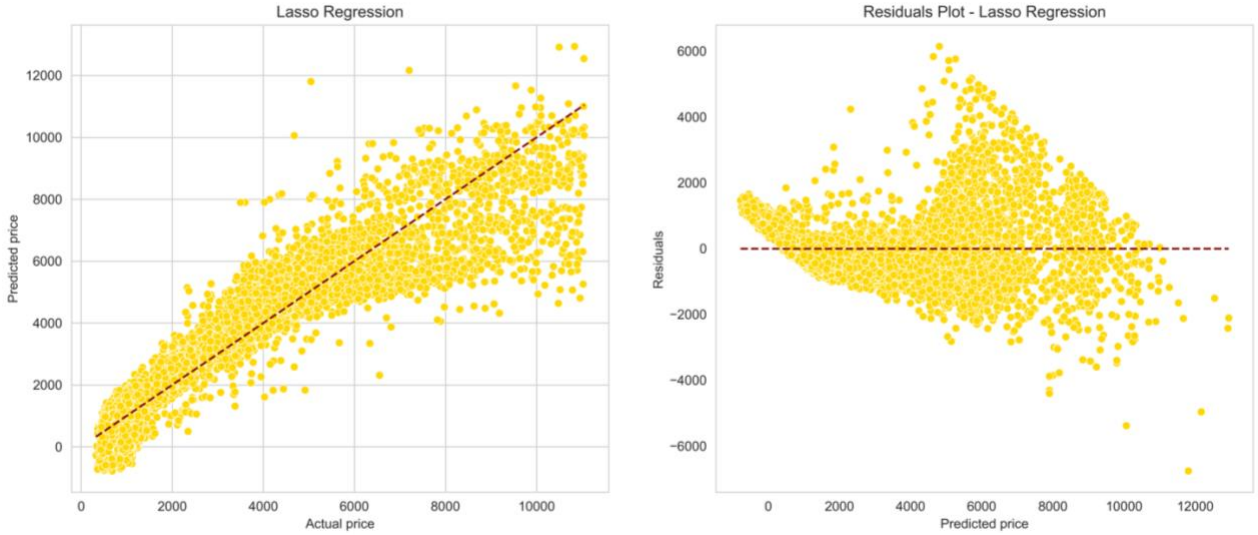


Figure 30: Actual vs Predicted Plot and Residual Plot for Lasso Regression

Train R^2	Test R^2	Difference	Overfit
0.8918	0.8886	0.0032	No

6.5.2 L-2 Regularization (Ridge)

This method adds a penalty equal to the square of the coefficients' magnitudes. As a result, all coefficients are reduced by the same factor, but unlike the L1 method, none are eliminated. The effect of Ridge co-efficients can be explained as:

- As alpha (λ) increases, the coefficients of all features shrink towards zero but do not actually become zero. This indicates that Ridge Regression is effective in reducing the magnitude of the coefficients without eliminating any features entirely.
- Different features shrink at different rates, which reflects their varying importance and sensitivity to regularization. For example, the coefficient for "carat" (blue line) starts very high and decreases significantly with increasing alpha, indicating its strong initial influence on the model which diminishes with regularization.
- By shrinking coefficients, Ridge Regression helps to reduce overfitting, especially for those features that have high multicollinearity. It retains all features in the model, thereby maintaining the overall complexity but with reduced individual feature influence.

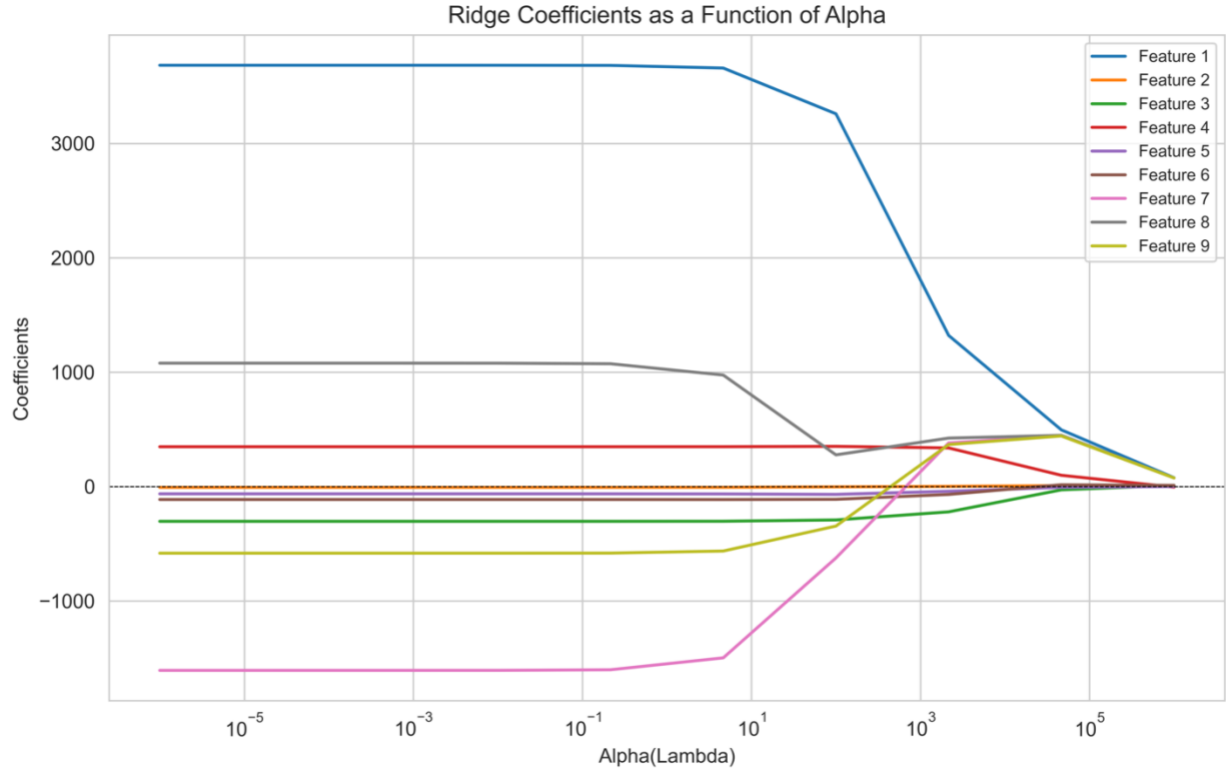


Figure 31: Effect of α for Ridge Regression

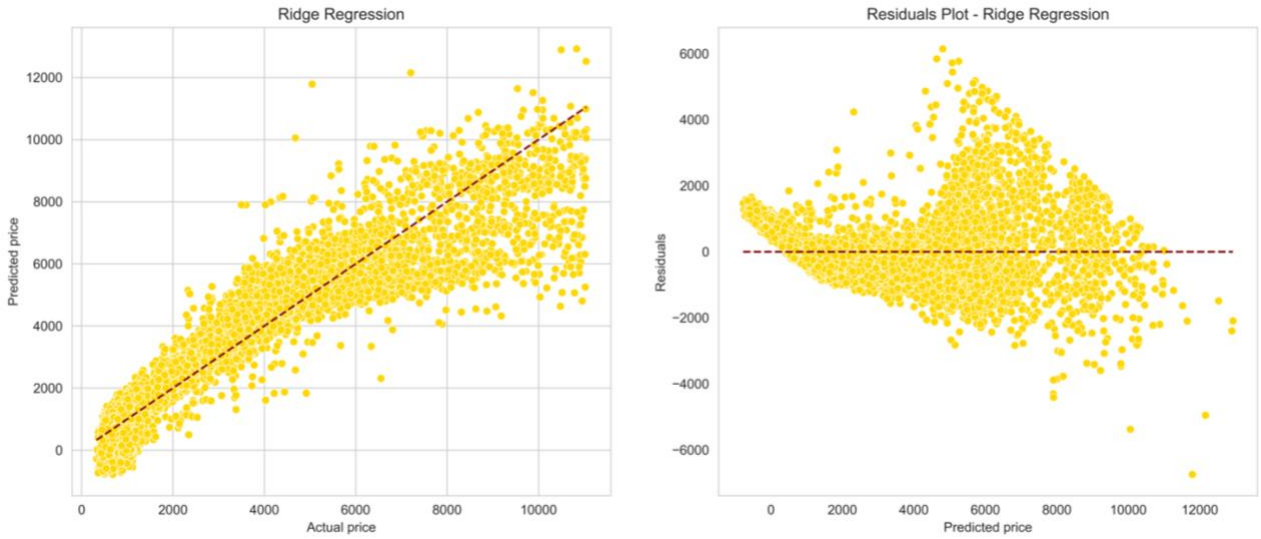


Figure 32: Actual vs Predicted Plot and Residual Plot for Ridge Regression

Train R^2	Test R^2	Difference	Overfit
0.8918	0.8885	0.0033	No

Comparing their co-efficients, it can be observed:

- There are slight differences in the magnitudes of the coefficients between Ridge and Lasso, but they generally align in direction and relative importance.
- Both methods indicate that "carat" is the most important feature.
- Both methods show consistent signs (positive or negative) for each feature, indicating agreement on the direction of their impacts.
- For features like "depth" and "table," both methods indicate minimal impact, suggesting they are less critical in predicting the target variable.

	Features	Ridge Coefficients	Lasso Coefficients
0	carat	3659.353036	3674.377035
1	cut	-6.255310	-6.424950
2	color	-302.737992	-303.221013
3	clarity	348.913501	348.856254
4	depth	-64.103406	-56.239871
5	table	-113.004854	-112.961167
6	x	-1497.044200	-1495.162328
7	y	975.296190	1041.441015
8	z	-563.200822	-646.302290

Figure 33: Ridge and Lasso Co-efficients

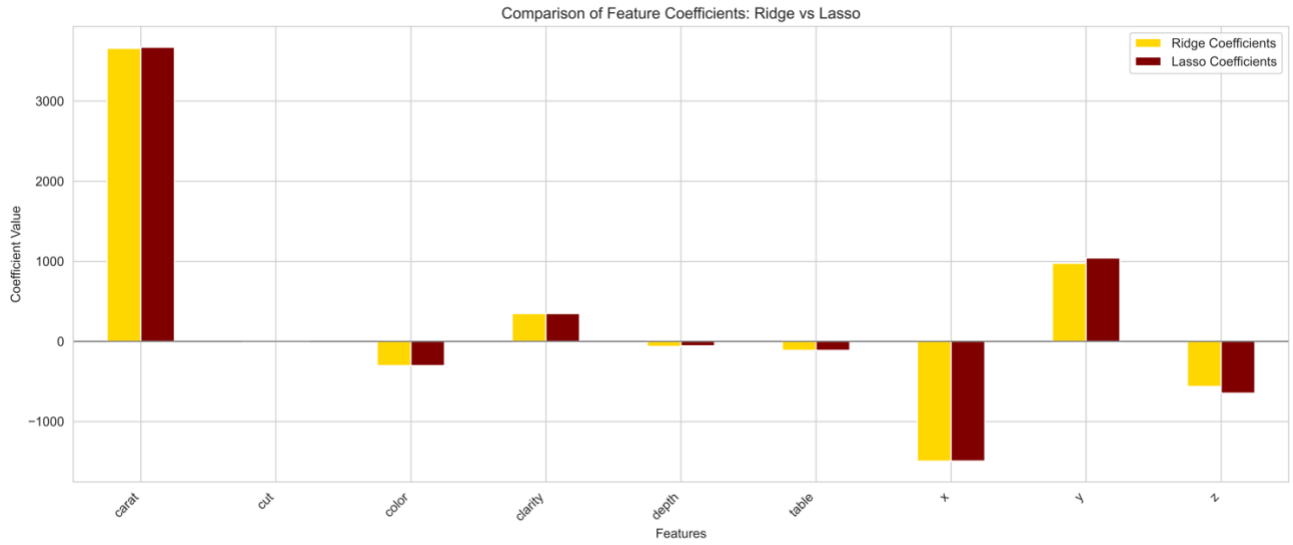


Figure 34: Comparison of co-efficients for Ridge and Lasso Regression

6.6 Neural Networks

[6]

Neural networks are computational models inspired by the human brain's structure, consisting of interconnected nodes (neurons) that process and transmit information. These models are used in machine learning to recognize patterns, classify data, and make predictions by learning from large datasets.

6.6.1 Multi-Layer Perceptron Regressor

The Multi-Layer Perceptron (MLP) model is a type of neural network that consists of **multiple layers of neurons, including an input layer, one or more hidden layers, and an output layer.** MLPs use backpropagation for training, allowing them to learn complex functions and perform tasks such as classification and regression. Upon using RandomizedSearchCV, the below given hyperparameters were found. This model achieved the **highest performance** among all the models trained on this dataset.

Best parameters:

	solver	max_iter	learning_rate_init	learning_rate	hidden_layer_sizes	alpha	activation
0	lbfgs	1000	0.0001	adaptive	50	0.0001	relu
1	lbfgs	1000	0.0001	adaptive	100	0.0001	relu
2	lbfgs	1000	0.0001	adaptive	100	0.0001	relu

Figure 35: Best Parameters for MLP

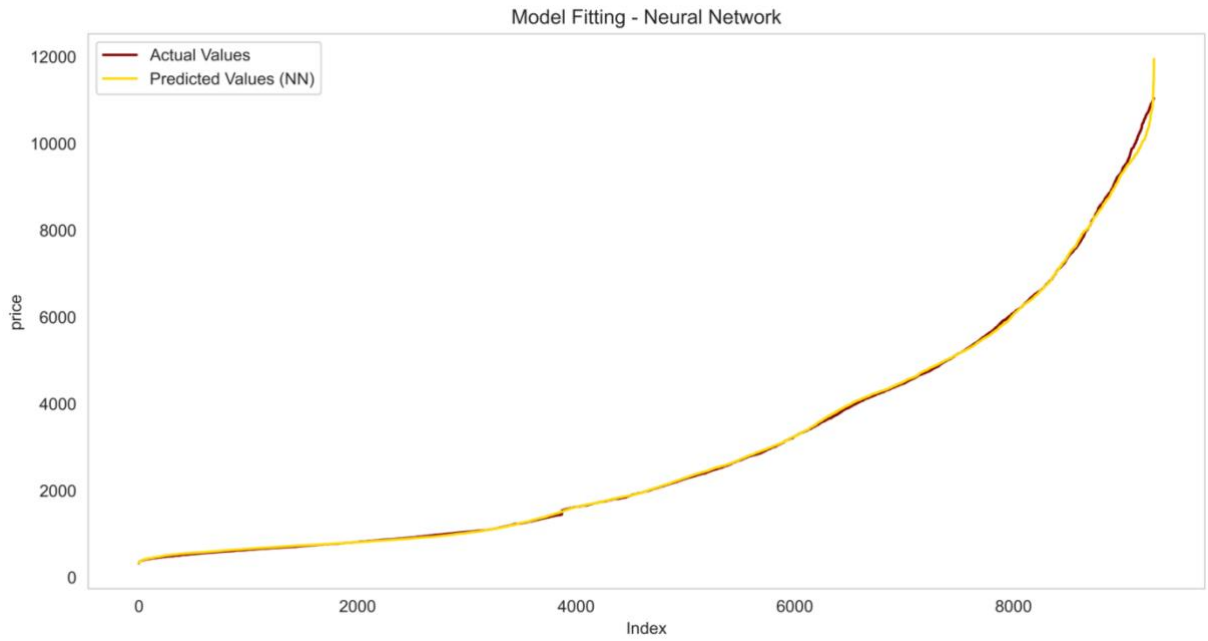


Figure 36: Model Fitting Plot for MLP

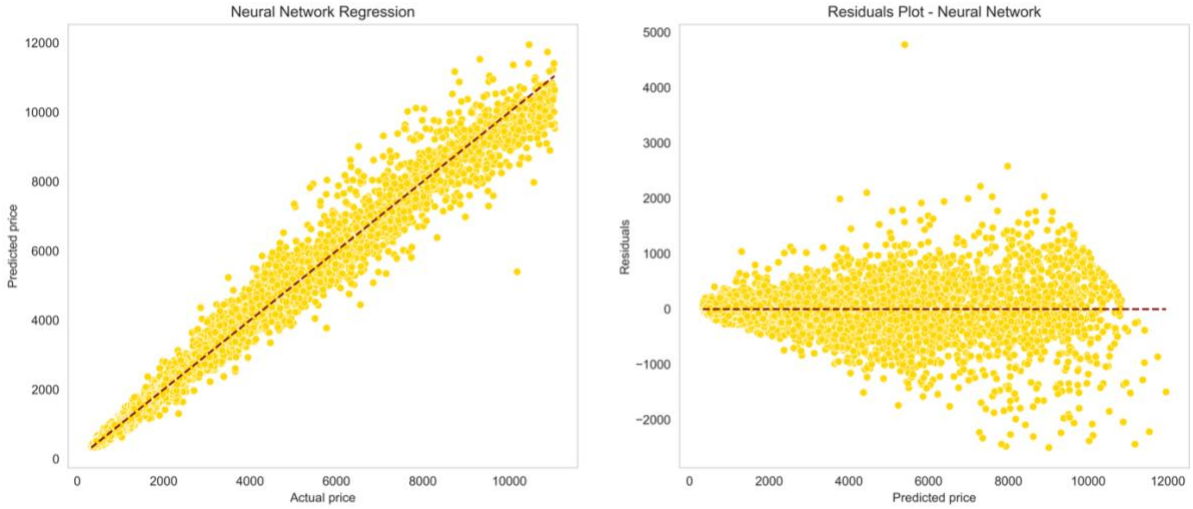


Figure 37: Actual vs Predicted Plot and Residual Plot for MLP

Train R^2	Test R^2	Difference	Overfit
0.984	0.983	0.001	No

7. Results

Method	Train R^2	Test R^2	Difference
Multi-Layer Perceptron Regressor	0.984	0.982	0.002
Random Forest Regressor	0.9521	0.9446	0.0075
Decision Tree Regressor	0.9466	0.9423	0.0043
Multiple Linear (Polynomial) Regression	0.944	0.93	0.014
Simple Linear Regression	0.892	0.889	0.003
Lasso Regression (L1)	0.8918	0.8886	0.0032
Ridge Regression (L2)	0.8918	0.8885	0.0033

8. Conclusion

All models exhibit robust performance, as indicated by high R^2 evaluation scores, with negligible differences between train and test R^2 values. This strong performance, achieved without overfitting, is largely due to comprehensive data preprocessing techniques that ensure the models' predictions are both accurate and generalizable to new, unseen data. Among the models, the MLP regressor performed the best, demonstrating superior predictive accuracy.

9. Bibliography

[1] A. O. Sykes, "An Introduction to Regression Analysis," *Chicago Unbound ,* 1993.

[2] J. L. F. H. Salvador García, Data Preprocessing in Data Mining, Springer International Publishing Switzerland , 2015.

[3] K. A. Marill, "Advanced Statistics: Linear Regression, Part I: Simple Linear Regression," *ACAD EMERG MED,* vol. 11, 2004.

[4] M. E. P. Tranner, Multiple Linear Regression, Manchester, 2020.

[5] P. Sarang, Thinking Data Science, Mumbai, 2023.

[6] "The Best Guide to Regularization in Machine Learning," 11 05 2024. [Online].

[7] S. A. A. a. W. S. Bhaya, "Review of Data Preprocessing Techniques in Data Mining," *Medwell Journals,* 2017.

[8] S. Kaushik, "Introduction to Feature Selection methods with an example," 30 05 2024. [Online].

[9] "Regression trees," 03 11 2022. [Online].

[10] A. Pinkus, "Approximation theory of the MLP model in neural networks," *Cambridge University Press,* 2008.

[11] S. A. A. a. W. S. Bhaya, "Review of Data Preprocessing Techniques in Data Mining," *Medwell Journals,* 2017.