



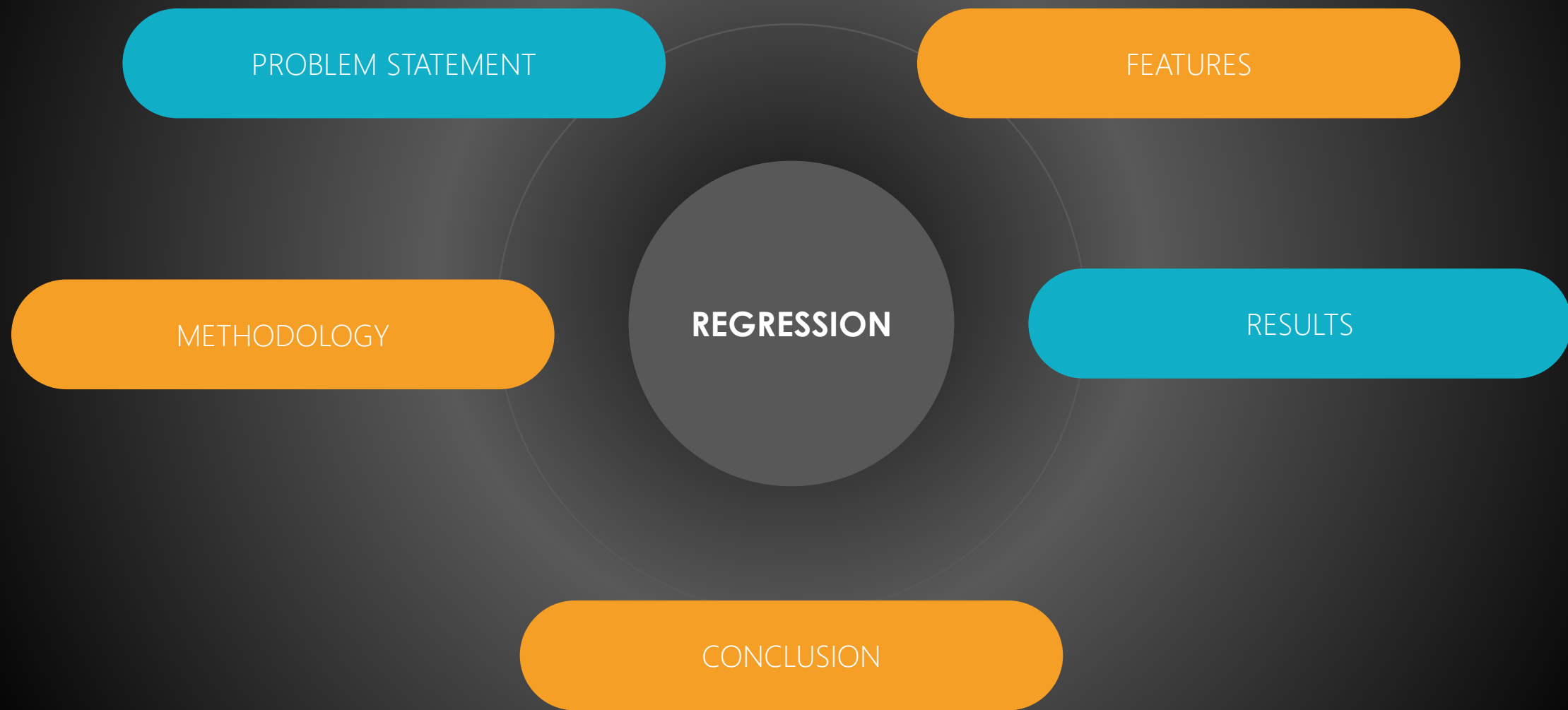
DIAMOND PRICE PREDICTION

ANJALI YADAV
DHANUSH GOWDA
KARTHIK M K

ATHARVA AGRAWAL
HARSHAD BANDI
YESHPAL SINGH

18-NOVEMBER-2022

INDEX



PROBLEM STATEMENT

We have datapoints for 53,940 across 10 features of diamonds. In this supervised model we have been provided with the target column which is the Price of the diamond. The goal is to predict the price of Diamond using different Regression Algorithms.

FEATURES

CARAT

- Unit of weight for Diamond

CUT

- The cut type of the Diamond, it determines the shine

COLOR

- Hue of a Diamond based on the GIA's color scale

CLARITY

- Visual appearance of Diamond in qualitative metrics

DEPTH

- The value of how deep or shallow the Diamond is

TABLE

- The flat surface on very top of the stone

LENGTH

- Length of the Diamond

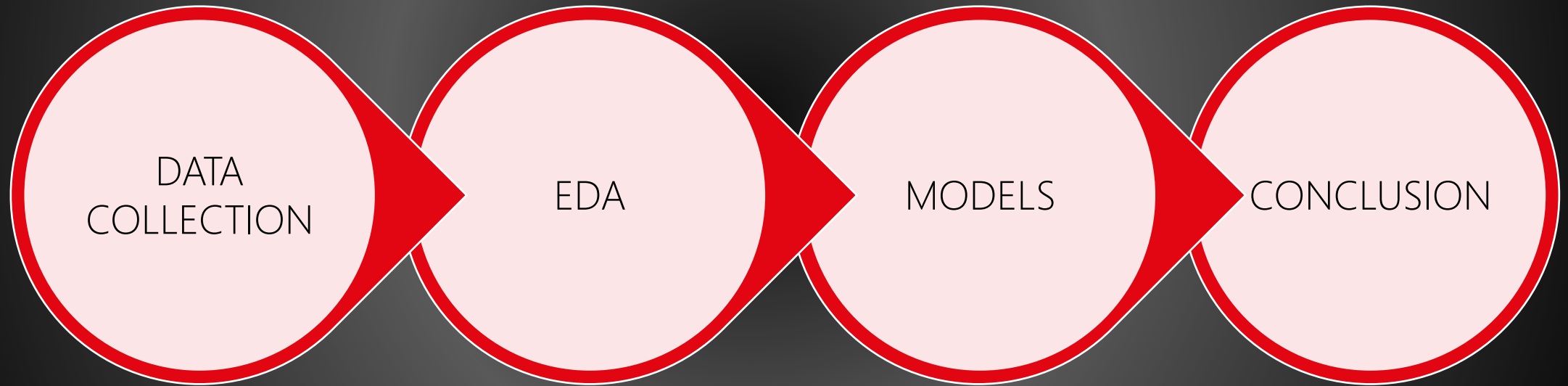
WIDTH

- Width of Diamond

HEIGHT

- Height of Diamond

METHODOLOGY



DATA COLLECTION

The Diamond price prediction data chosen for this project is taken from Kaggle. We used pandas `read_csv()` function to load the data to the notebook. This dataset comes with both categorical and numerical data which has been cleaned and processed to build the model.

Source : [Data Source](#)

EXPLORATORY DATA ANALYSIS

1. Missing value treatment : We used `isnull()` function on our dataset to find the missing values.
2. Outlier treatment : We checked the outliers in the data through boxplot.
3. Linearity : We checked the data linearity through Pairplot.
4. Normality : We used Displot to find the normality of target variable.
5. Multi-collinearity : We used heatmap to display correlation between columns
6. Count : We used `value_counts()` to get the count of attributes
7. Numerical data conversion : We used LabelEncoder to convert the categorical columns to numerical data.

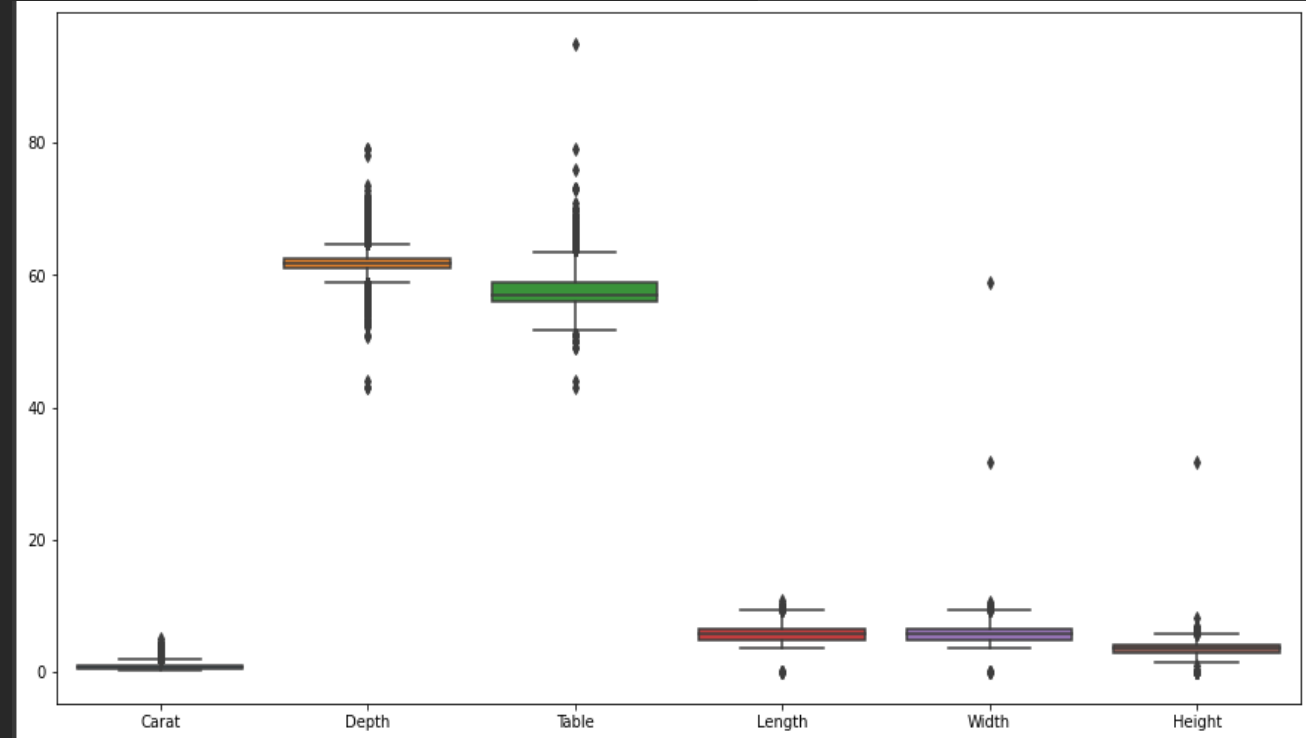
MODEL BUILDING

- We divided the data as train and test set using `train_test_split()` function.
- We imported the required libraries for the models used
- Models built were:
 - Linear Regression
 - Decision Tree
 - Random Forest
 - ADABoostRegressor
 - SVM Regressor
- Models were evaluated using `r2_score`

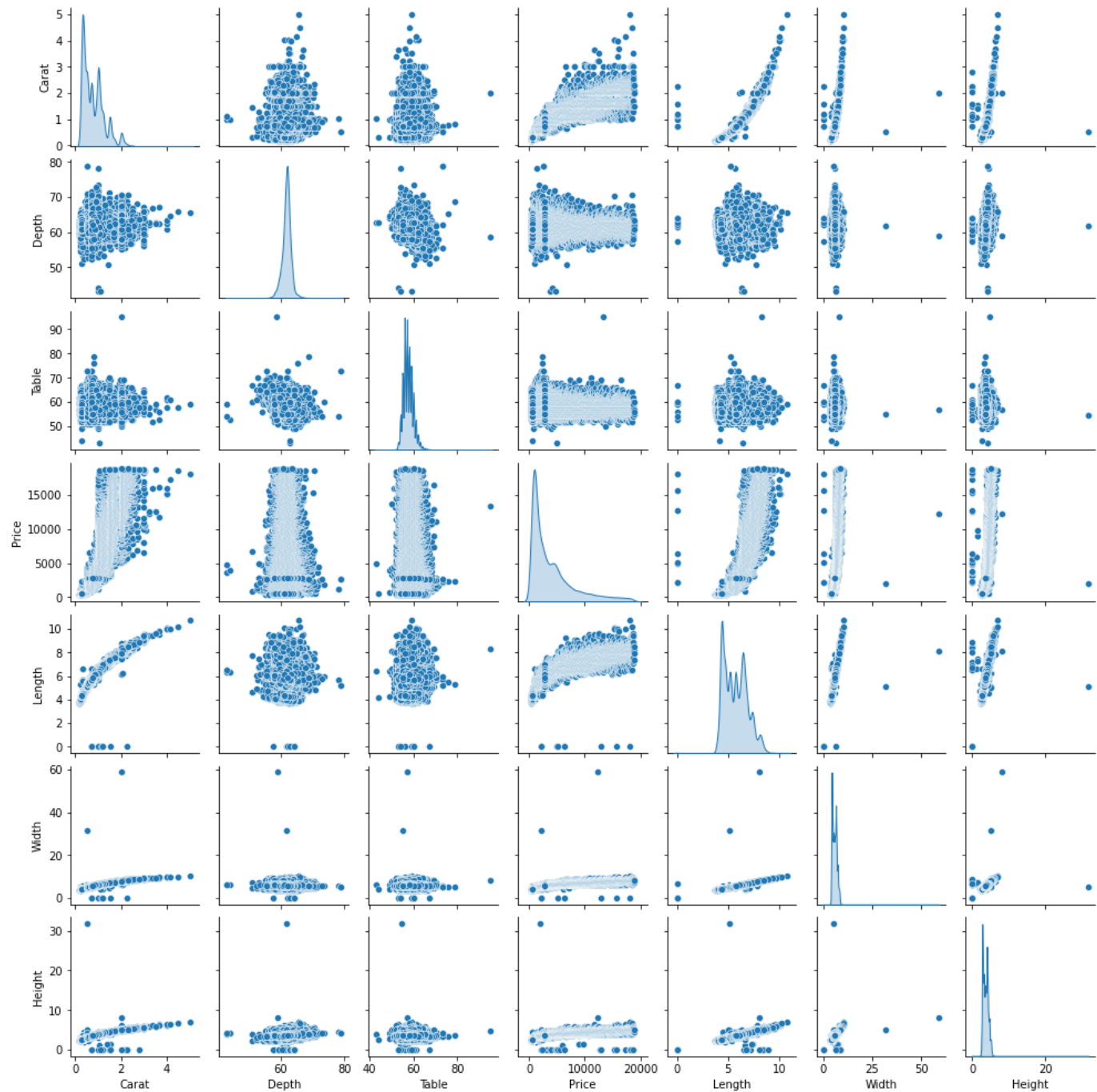
RESULTS

EDA

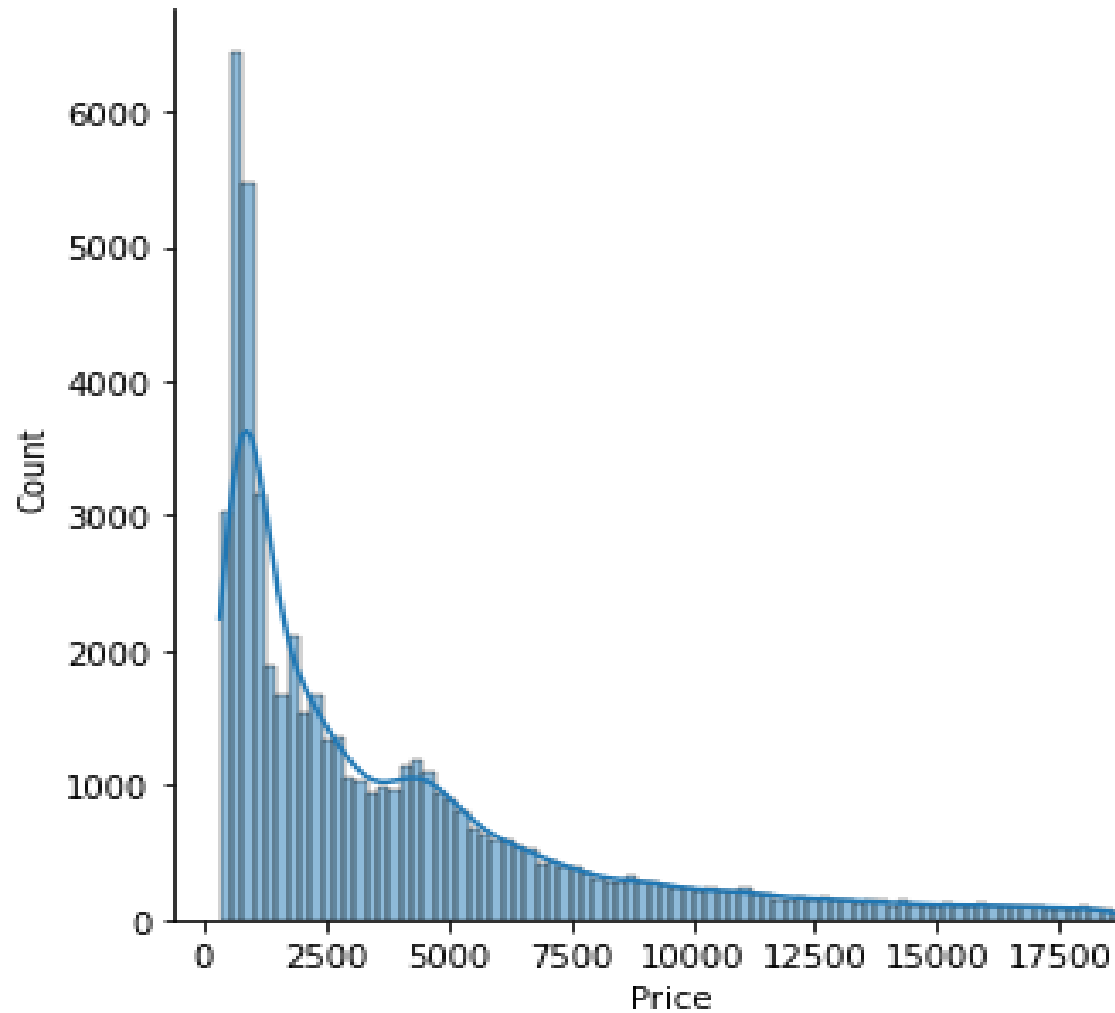
1. The Dataset did not contain any missing values.
2. Even though the boxplot shows many outliers we have taken the data as it is for the model building, as the diamond features may vary with different types.
3. We have converted categorical columns into numerical using LabelEncoder



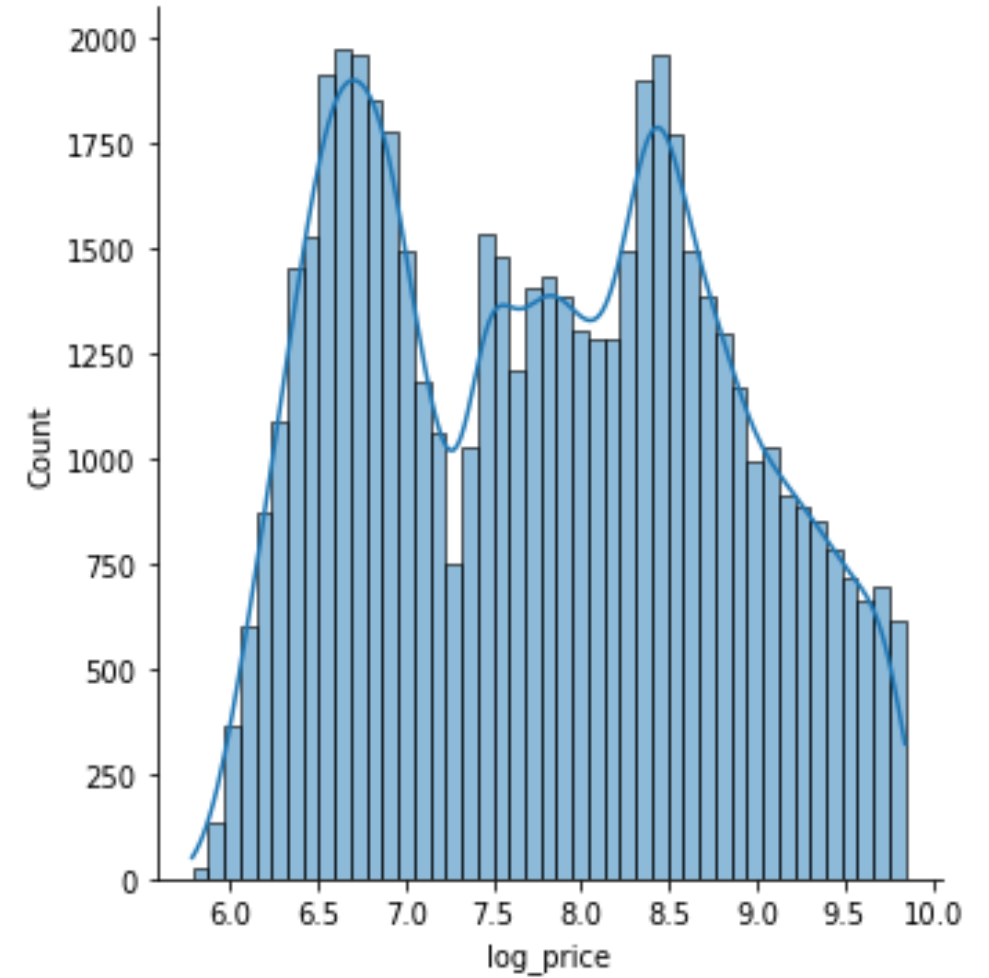
4. From the below pair plot we can say that the data is linear



Before logging



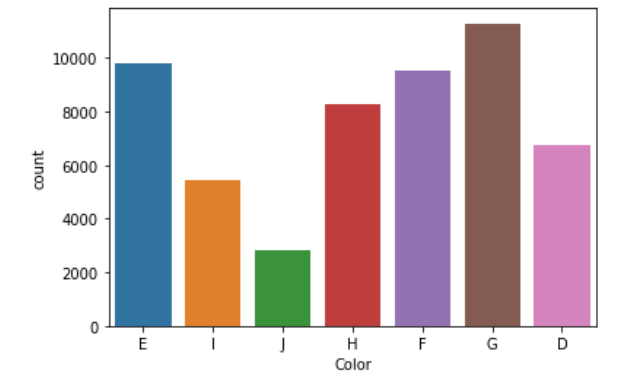
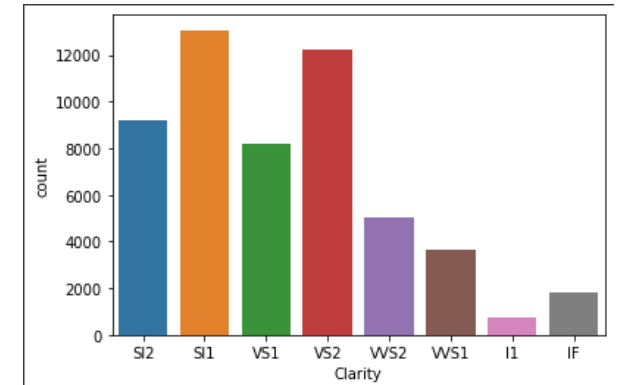
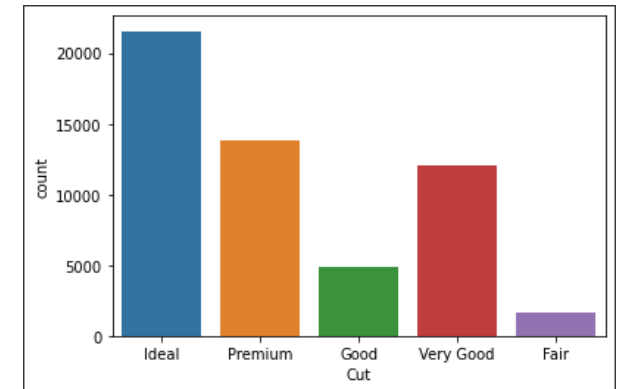
After logging



5 . We transferred target column price from skewed data to normal data

6. Counts of Categorical columns

1. In the cut feature of Diamond Ideal cut has highest value count with 21551
2. In the color feature of Diamond G type has highest count with 11292
3. The clarity feature of diamond has SI1 type with highest count of 13065

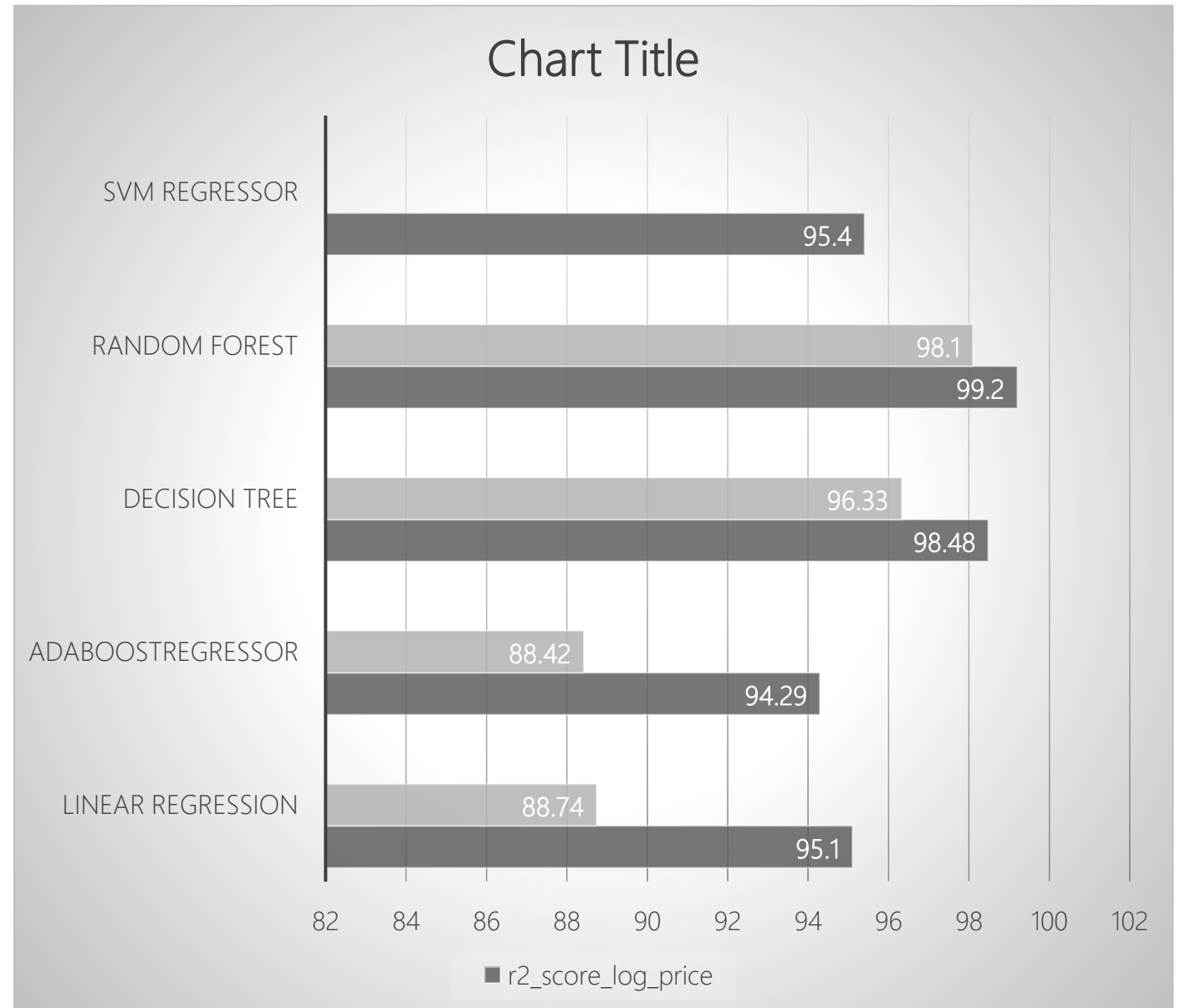


7. There are multiple columns which are collinear to each other



Models

- We have built 5 regression models i.e., Linear Regression, ADA Boost Regressor, Decision Tree, Random Forest and Support Vector Machine('Poly') Regressor.



CONCLUSION

- 53,940 data points from 10 columns have been processed to build the regression models.
- Ideal cut of diamonds are used or sold more in the market
- G colored diamonds are more frequent
- Diamond with SI1 clarity have appeared more in the dataset
- With 98.1% r^2 _score we can conclude that Random Forest Regressor gives the best model with or without outliers.

An aerial photograph of a multi-lane highway bridge spanning a body of green water. The bridge has several lanes in each direction, with white lane markings. Several vehicles, including cars and trucks, are visible traveling across the bridge. The water has a textured, rippled surface. The text "THANK YOU" is overlaid in the center of the image in a large, white, sans-serif font. The text is contained within a white rectangular border that is slightly offset from the edges of the image.

THANK YOU