Cooperators: Shahed Hisham and Mohammed Gouda

Diamond competition steps:

-This document aims to focus on the main steps that led to the final score, in addition to all the additional approaches that were tried throughout the project lifetime.

**EDA:**

-Id column was dropped as it gives no essential information to the data.

-after checking for null values, none was found.

-97 duplicated values were found, and they were dropped.

-price column seemed to be the one with most outliers which was tricky to deal with as isn’t efficient to replace neither trim them as prices variation is essential in the market and to the model to release higher prices do exist.

-correlation heatmap was created to understand more how columns are related

-x, y, z columns are highly correlated we can find that carat also have high correlation value with the prices

**Visualizations insights:**

-while exploring skewness it seemed like x, y, z and carat columns are right skewed which meant that the mean is greater than the median so that’s why a lot of outliers appear at these columns

-while depth and table columns seem to be slightly left skewed

-It seems like the ideal cut is the most common and frequently bought

-Sl1 and Vs2 clarities are the most common which aren’t the highest clarities and can’t be detected easily

**Feature Engineering:**

-in order to achieve dimension reduction and create a more informative data set.x, y, z columns are merged into volume column.

-density column is also created from the product of carat and volume to provide more information and show interactions effect on the model.

-exploring the correlation of numerical column with the price column showed that carat is the highest corelated column while depth is lowest corelated which will later help us through the feature reduction decision.

**-full transformation pipeline:**

-a preprocessing pipeline is created with two custom transformer classes.

-Label encoder is used for categorical columns convert categorical data into a numerical format that machine learning algorithms can process, although hot encoder was an option, but label encoder works well for ordinal data.

-Robust scalar was used for numerical columns as it can handle outliers well as it reduces its impact by using median and IQR instead of mean and std in Standard Scalar.

**Model implementing:**

-we have tried many various models to get the lowest RMSE score.

-cross validation was used to approve the score:

>>Model scores will be mentioned:

1-SVM model: RMSE score of 2749.42

2- Decision Tree model: RMSE score of 739.696

3-Random forest Tree model: RMSE score of 556.09

4-votting classifier with (gradient-random forest – ridge): RMSE score of 736

5-neural network model: with RMSE score of 29641496

6-linear discriminant model: RMSE score of 1196.2

**Steps done after the model implementation:**

-it was obvious that the random forest tree seemed to give the most efficient score but not the best, so several tries were done to try reducing the score.

-feature selection through eliminating features with least correlation with price but this didn’t seem to help much as the error score increased.

- after learning about feature importance after hyperparameter tunning a new data frame that only contained important features with highest importance score was created and went through the previous steps again, but this also showed a higher error score.

-at this point we needed to go back a step and investigate the data processing phase so the first thing we tried was changing the scaling methods, but this also didn’t work.

- handling outliers with capping method using IQR wasn’t a success as it seemed to worsen the model prediction and caused overfitting.

-we concluded that any changing in the original data or trimming wasn’t the best option.

**Hyperparameter tunning:**

-at first, we used the normal GridSearchcv which showed score of 557

-BayesSearchcv was used next but gave higher error

-Back to GridSearchcv we tried the extended parameters, but this increased the runtime significantly.

**-**boosting algorithm was used which uses decision tree as weak learner and train them sequentially which help improving the score to reach our least score

**Additional approaches we tried but didn’t document:**

**-**PCA and LDA models were used to decrease data dimensions but didn’t seem to work