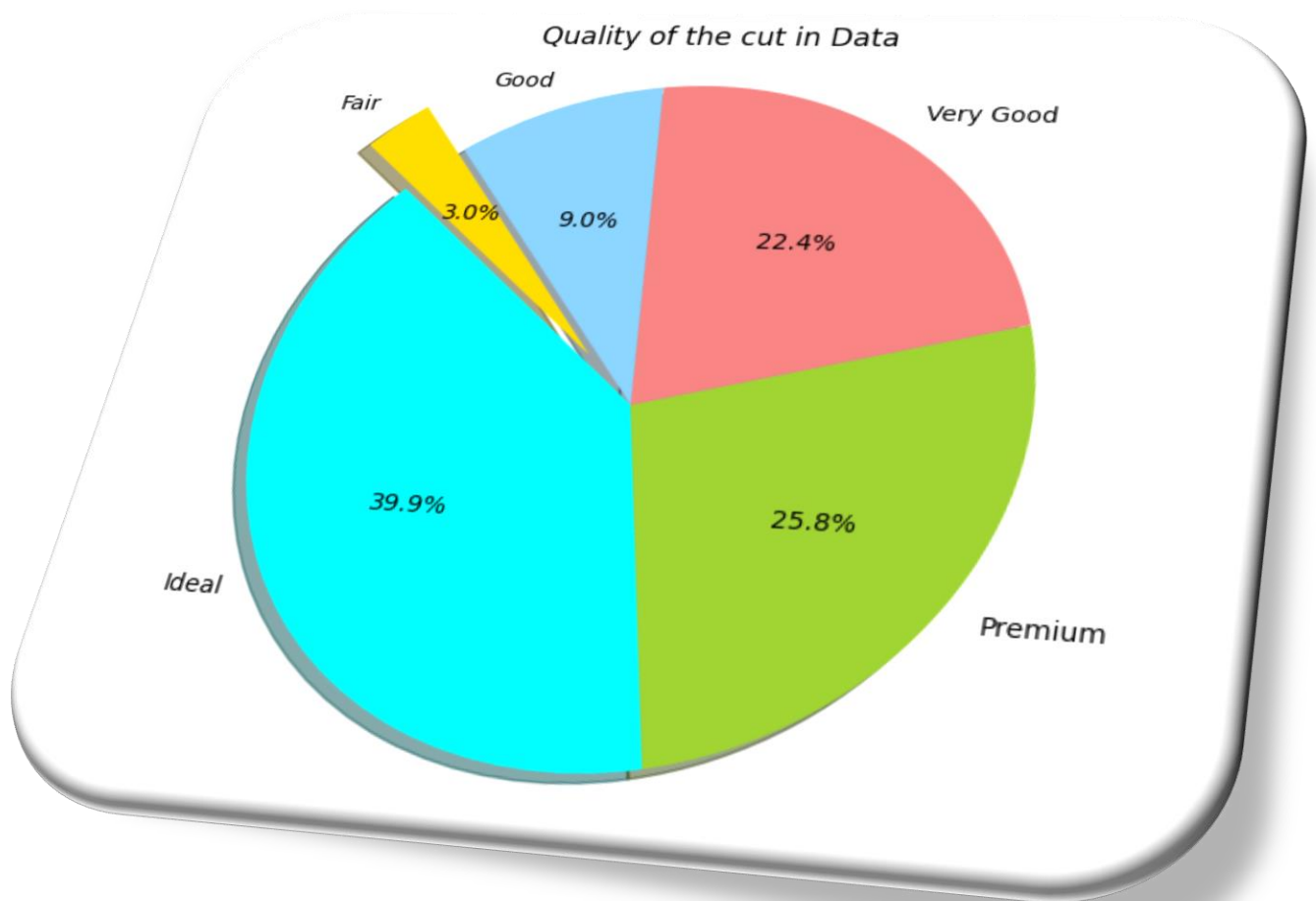
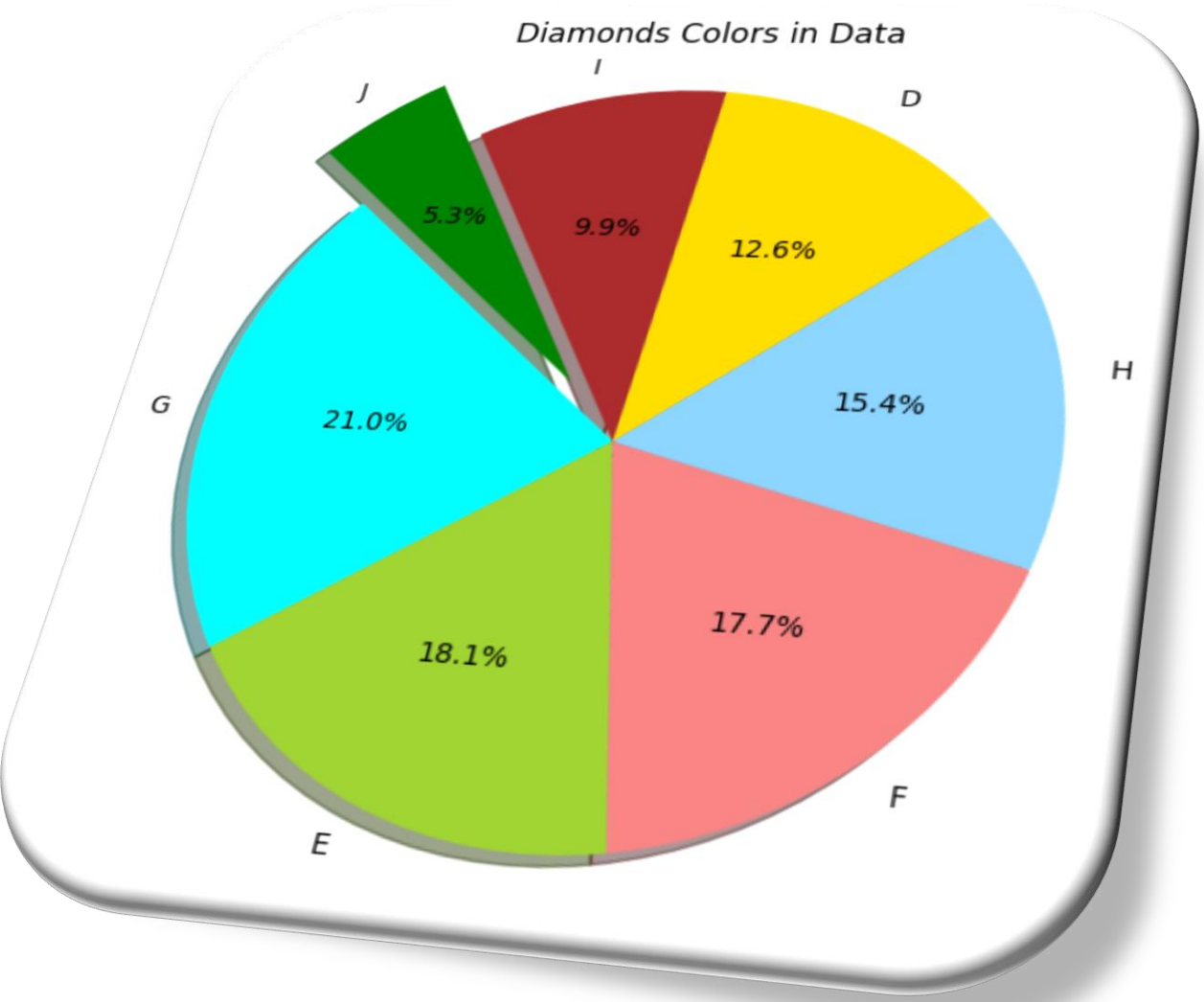
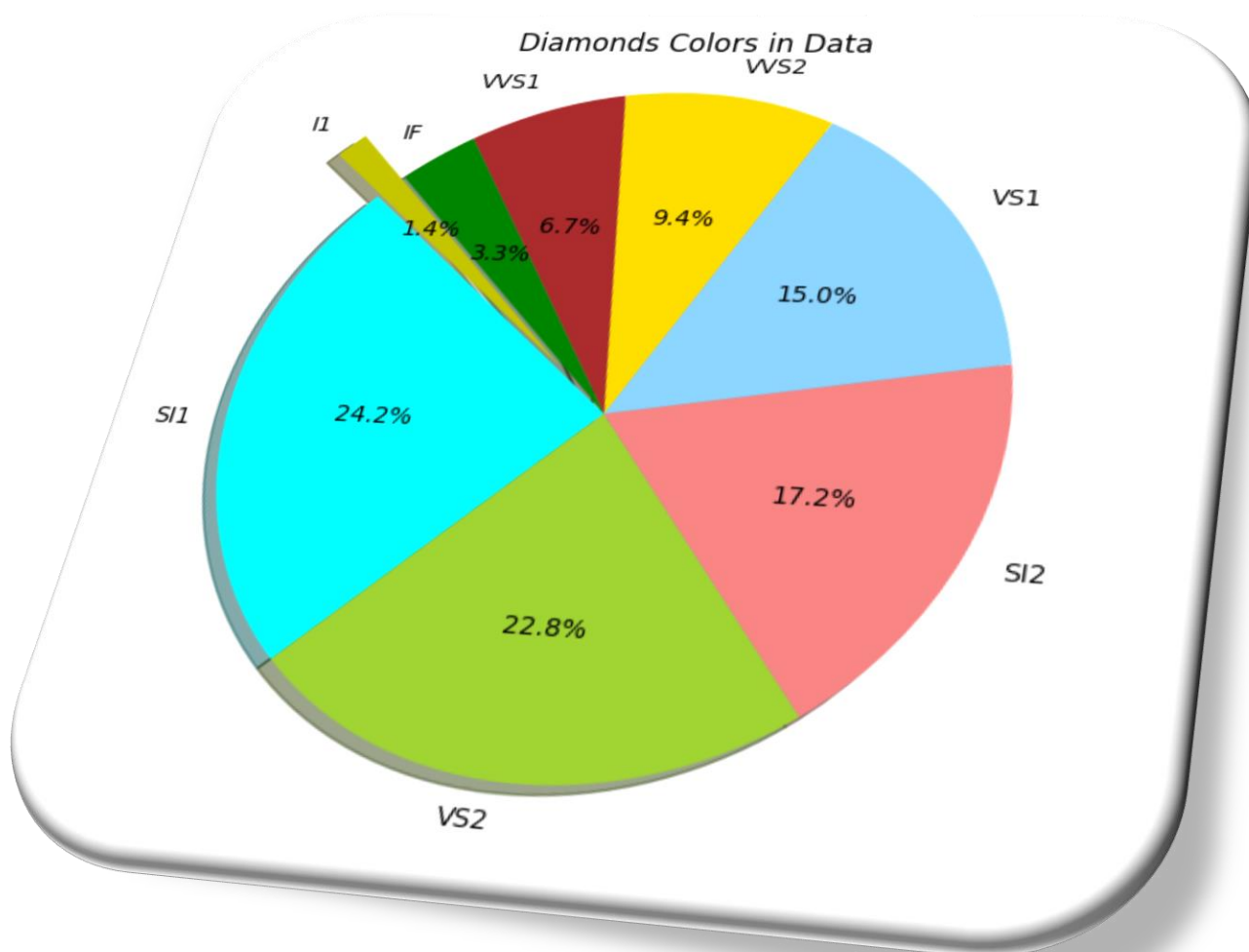
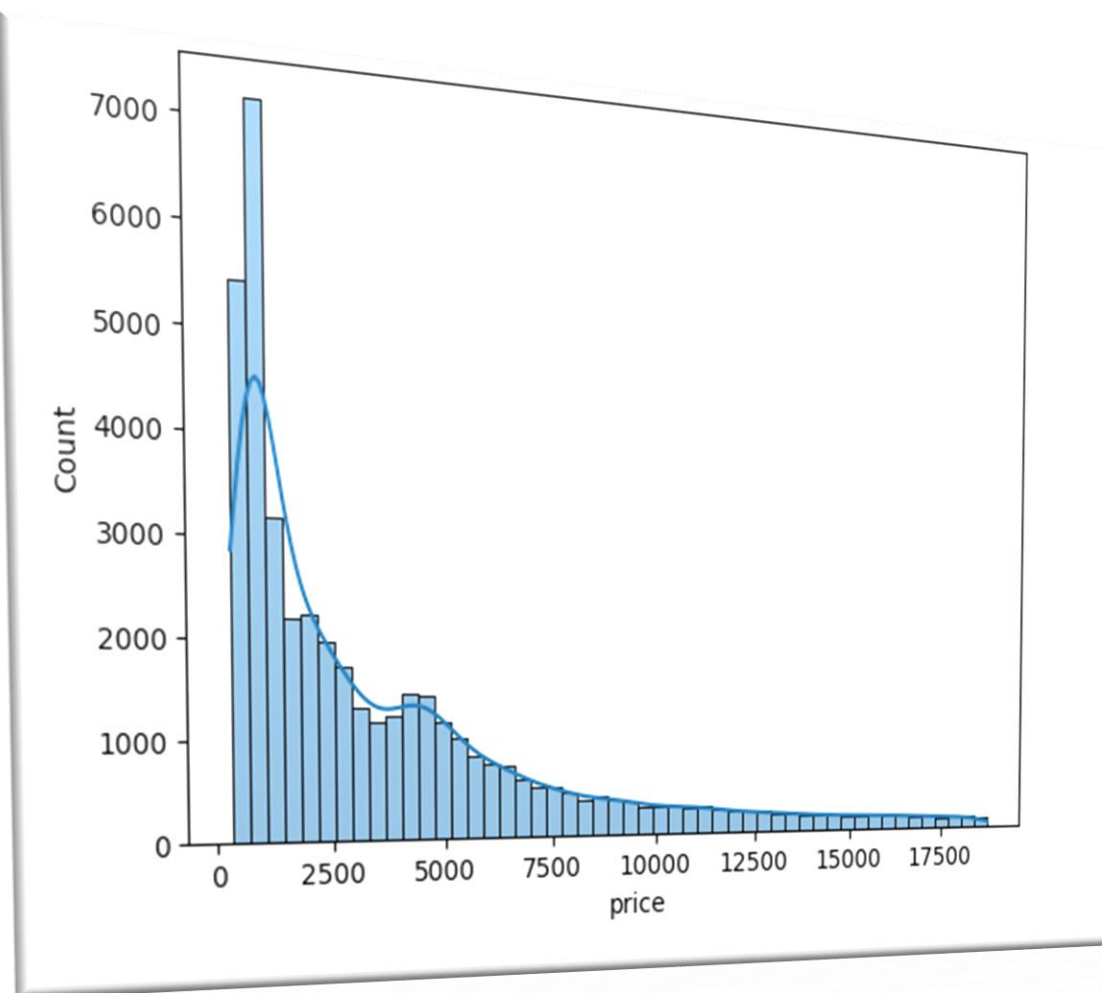


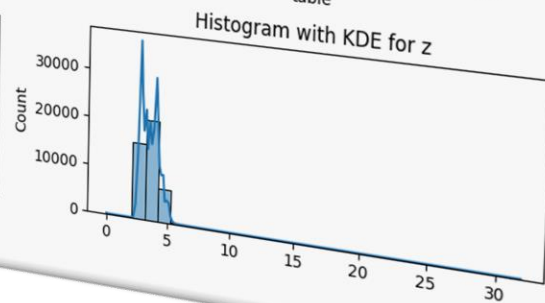
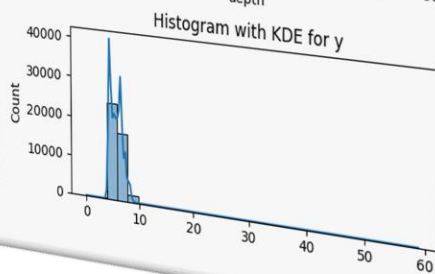
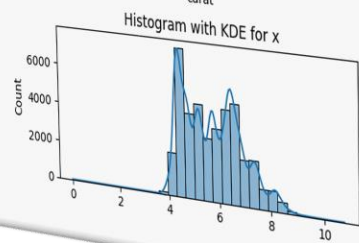
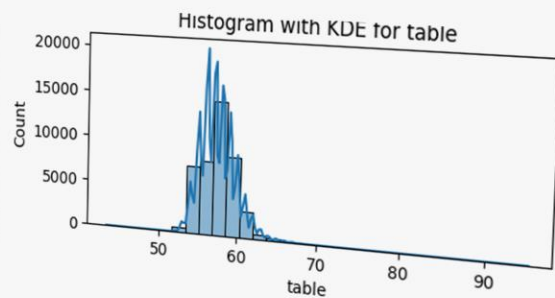
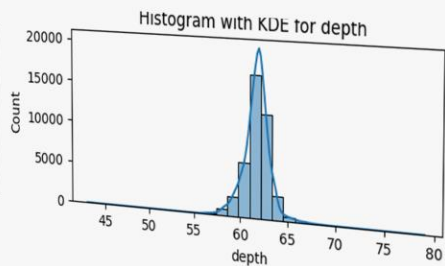
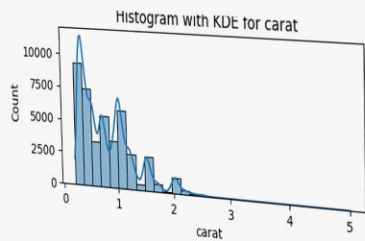
## Looking at the big picture of the data

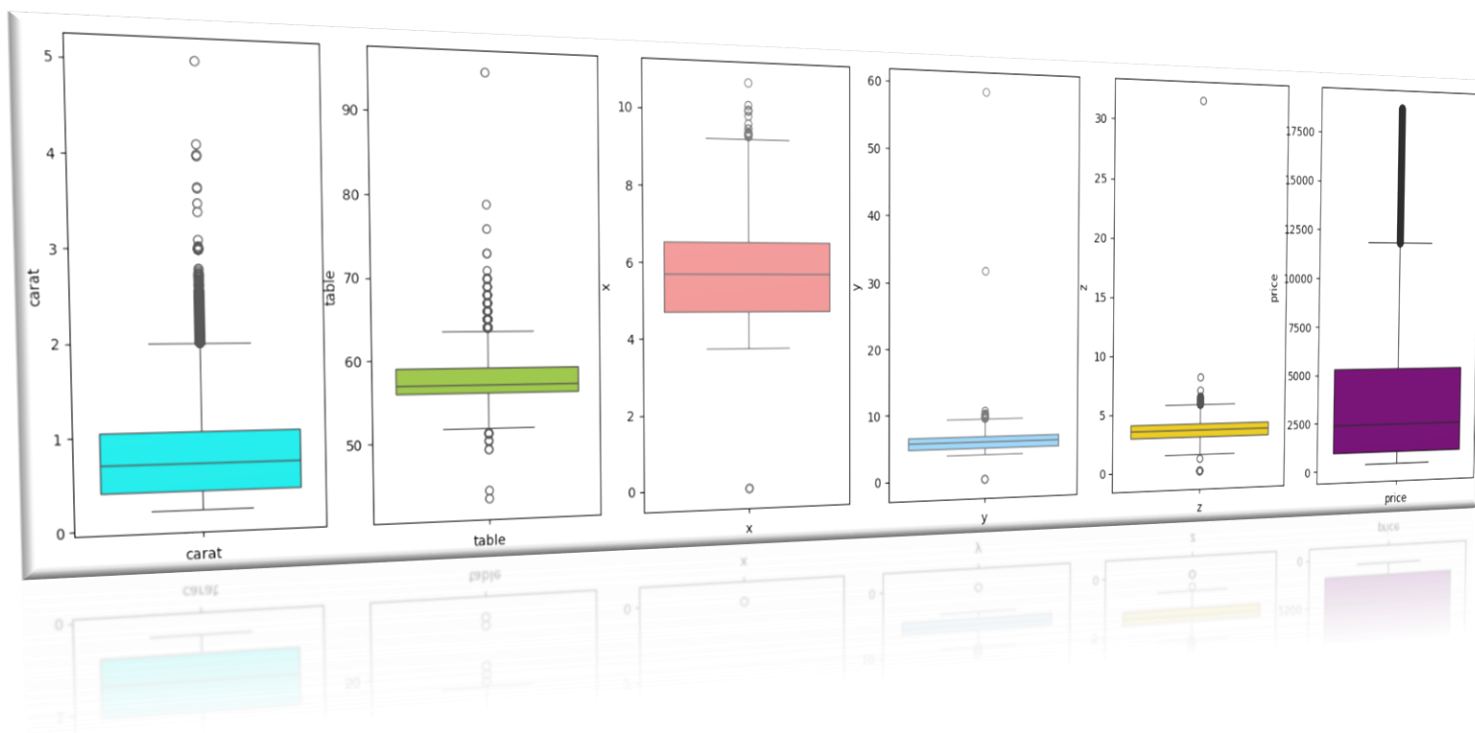














## From this big picture, we have noticed that:

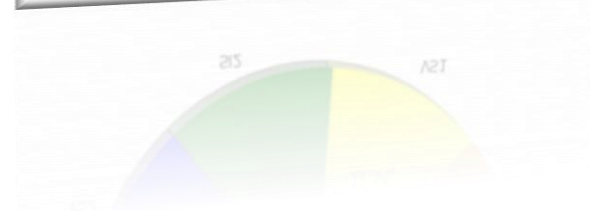
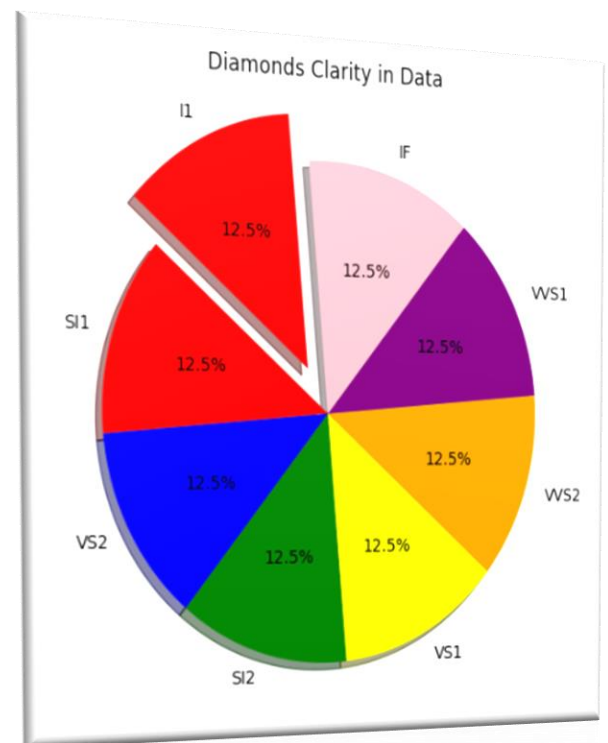
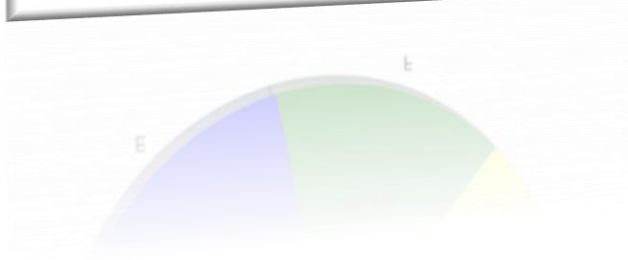
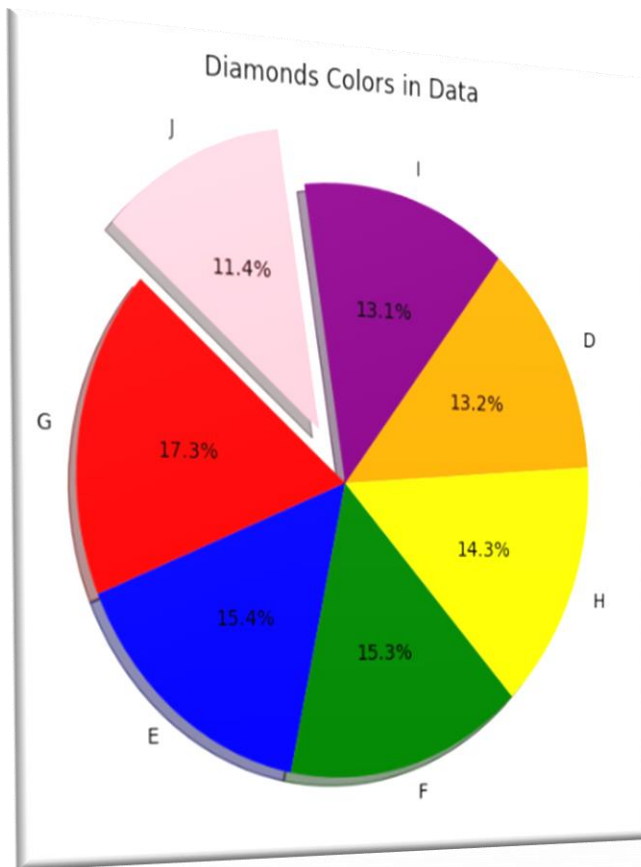
- **Categorical data are not balanced.**
  - We determine specific percentages and apply oversampling to features that exist in a low percentage, and undersampling to features that exist in a high percentage, to ensure that they all exist in the same percentage.
- **The target value, price, is highly left-skewed.**
  - We apply the logarithmic scale to ensure that the data is normalized and to reduce the range of prices.
- **Some features are highly skewed and have different ranges.**
  - We apply the logarithmic scale, as we did with the price feature, and also normalize the data.
- **The features contain high values of outliers.**
  - We can remove the outliers because they greatly influence the model's ability to predict values within a certain range. Instead, we use the logarithmic scale to small range and reduce outliers.
- **X, Y, Z, and carat features are highly correlated with the price.**
  - We create new features from these existing features and do not remove them, even though they are highly correlated. Removing them reduces the model's accuracy.
- **Data doesn't have many features correlated to the price.**
  - We created three new features.



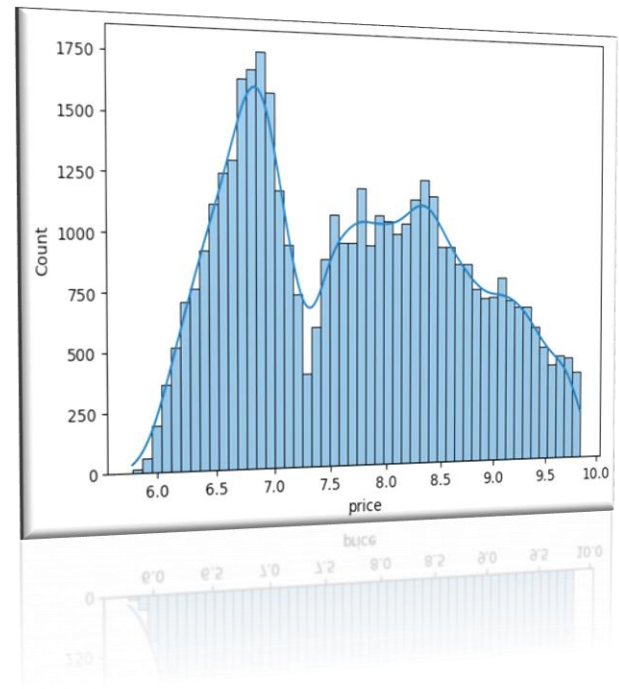
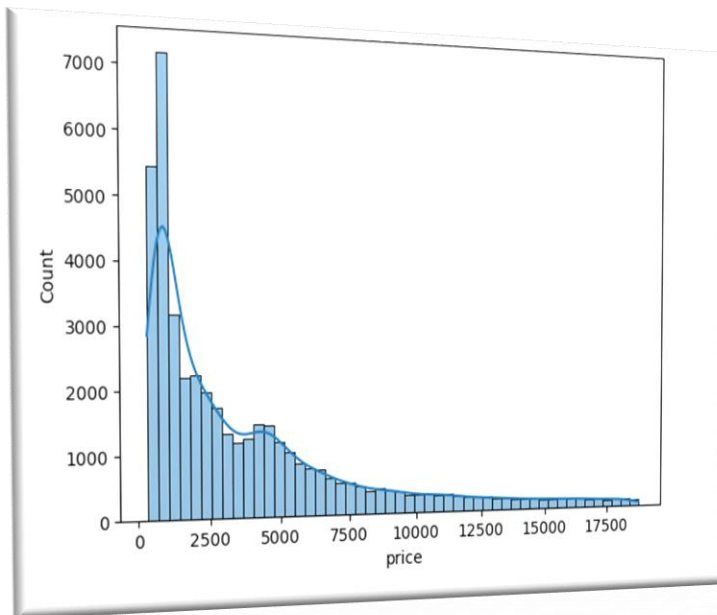
First thing we create new features correlated to the price and remove cut and depth which are not correlated to the price:



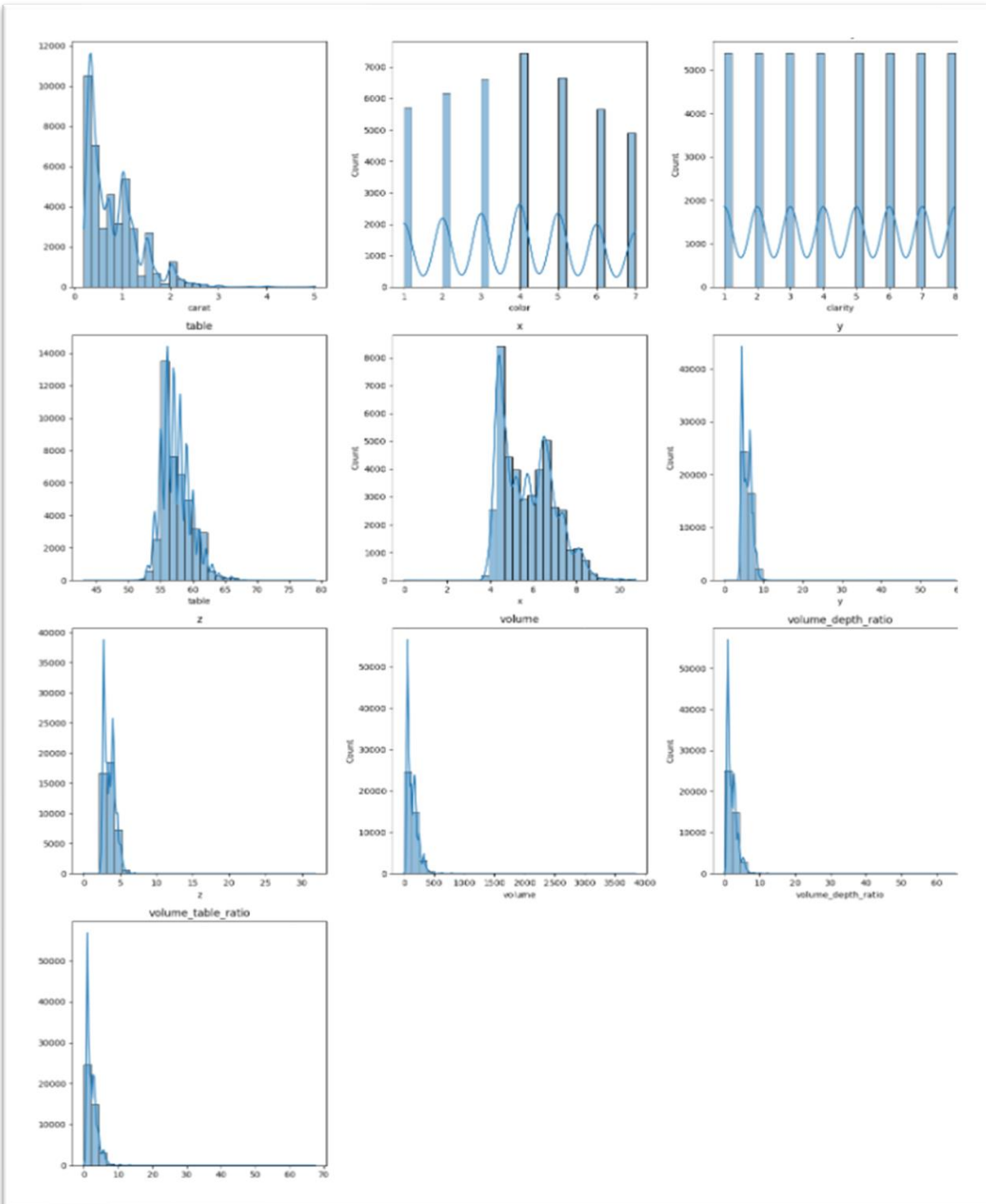
We have applying oversampling and undersampling to the categorical feature:

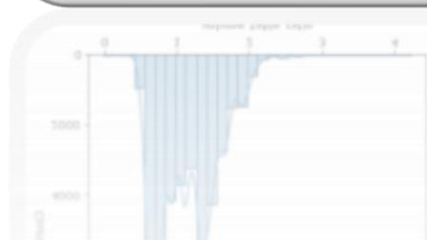
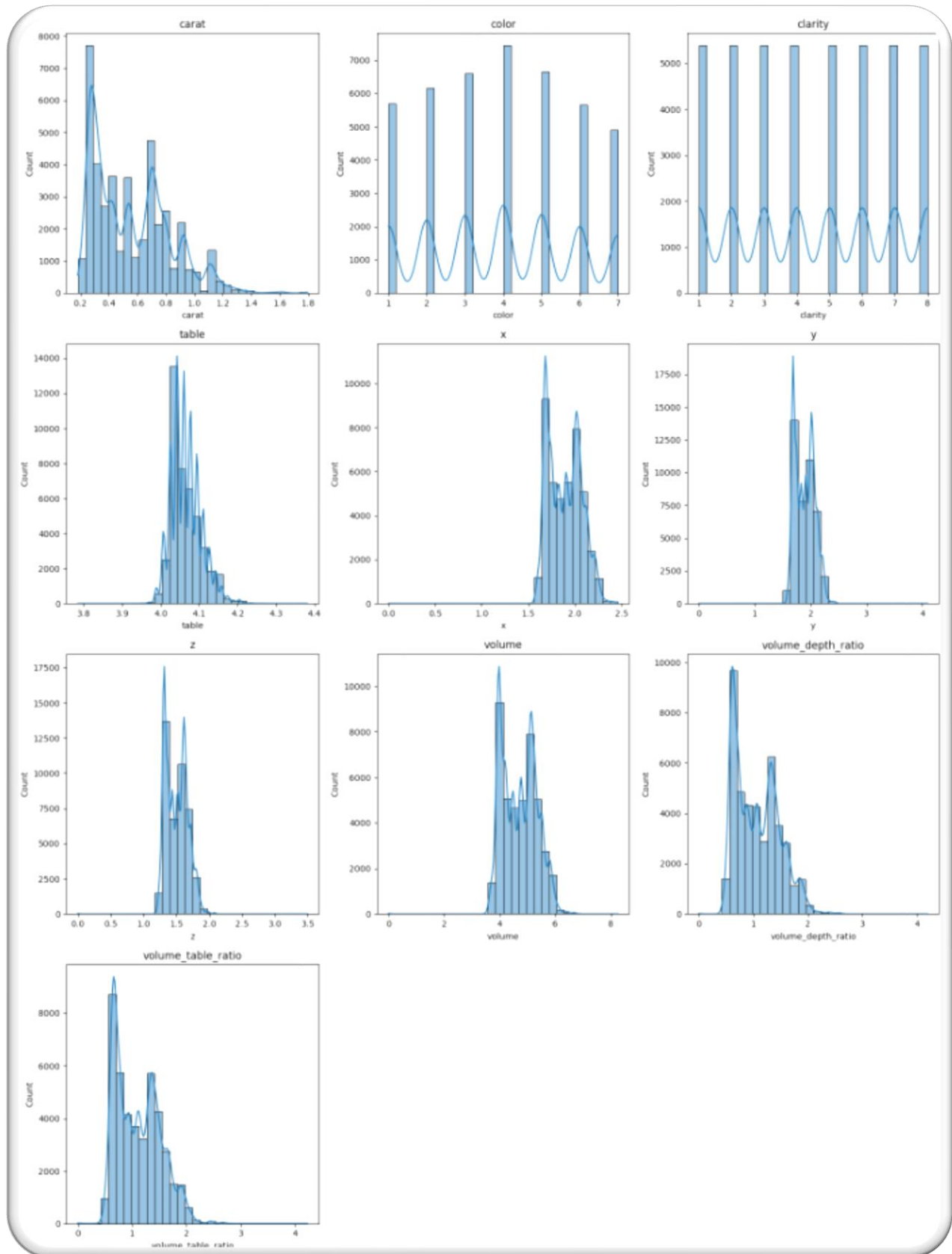


We have applied a logarithmic scale to the prices to address the issue of highly left-skewed data and to reduce the range and solve the problem of outliers:



We applied a logarithmic scale to address the left-skewed distribution of the carat feature and to narrow the range of all features, thereby addressing the issue of outliers. Subsequently, we performed normalization on the features to ensure a smaller scale and a mean of zero:





	carat	table	x	y	z	volume	volume_depth_ratio	volume_table_ratio	price
count	4.312800e+04	4.312800e+04	4.312800e+04	4.312800e+04	4.312800e+04	4.312800e+04	4.312800e+04	4.312800e+04	43128.000000
mean	-1.370737e-16	-1.948028e-15	1.133494e-15	-9.542440e-16	-5.588390e-16	6.010156e-16	4.323094e-16	1.212575e-16	7.745299
std	1.000012e+00	1.000012e+00	1.000012e+00	1.000012e+00	1.000012e+00	1.000012e+00	1.000012e+00	1.000012e+00	1.001382
min	-1.411102e+00	-7.315294e+00	-1.070661e+01	-1.081159e+01	-9.078761e+00	-7.593446e+00	-2.648100e+00	-2.722815e+00	5.789960
25%	-9.189991e-01	-6.142355e-01	-9.352538e-01	-9.351735e-01	-9.329006e-01	-9.314457e-01	-9.301276e-01	-9.382759e-01	6.852243
50%	-1.173909e-01	-1.640217e-01	2.321928e-02	3.103315e-02	3.656647e-02	5.378109e-02	-6.271268e-02	-6.648548e-02	7.723562
75%	6.497424e-01	7.135756e-01	7.835498e-01	7.827099e-01	7.423372e-01	7.716198e-01	7.407497e-01	7.351745e-01	8.530307
max	4.572998e+00	8.160699e+00	3.220093e+00	1.255471e+01	1.201249e+01	5.712088e+00	7.777952e+00	7.625078e+00	9.841399

- It is important to have features with zero mean before training the model.

- After the data is prepared for training, we trained it using AutoGluon, utilizing RMSE as the evaluation metric. The training process took approximately one hour to select the best model and optimal hyperparameters.
- When training the model on 80% of the data and testing it on 20% of it we get:

- **Accuracy:**

0.9812931672741864

- **RMSE:**

538.1464950308223

- The model final score on Kaggle is:

Score: 562.42105

Public score: 571.52336