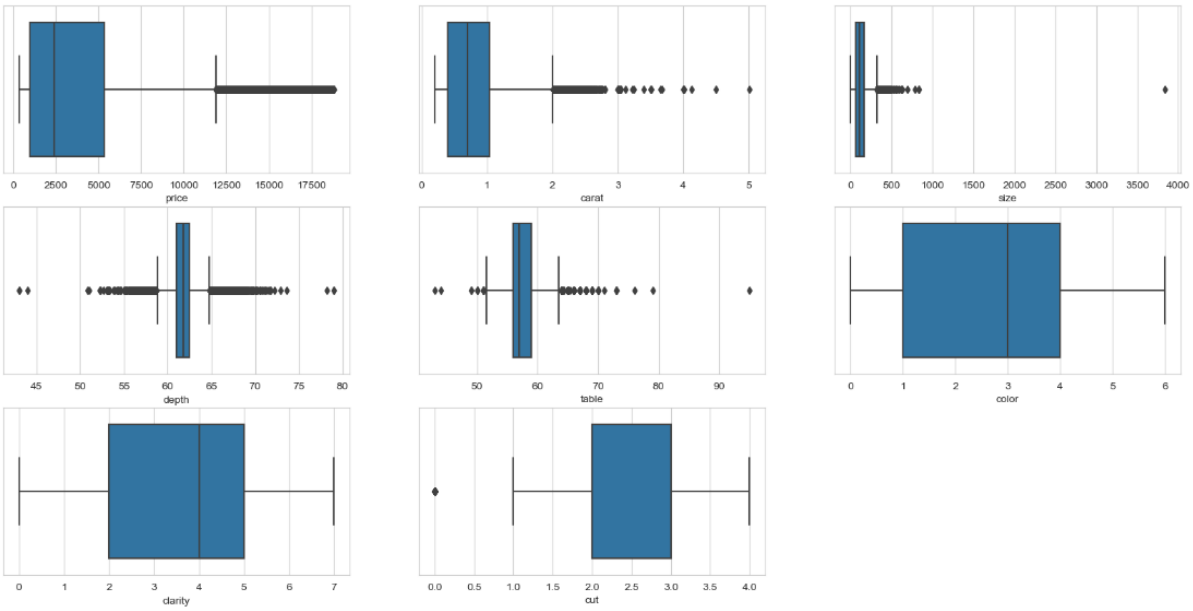
Investigation of the Diamond Dataset

https://github.com/yasking68/Maha-ML-Project/tree/main/Project%20MAHA

The dataset under scrutiny encompasses diamond properties across ten distinct columns, covering key attributes such as weight, cut quality, color grade, clarity rating, dimensions, and price. Despite the expected hierarchy of diamond color, the dataset's initial examination suggests a discrepancy in color level ordering, necessitating clarification. This preliminary understanding lays the foundation for a comprehensive analysis of the dataset's intricacies and insights into the factors influencing diamond characteristics and pricing.

I used LabelEncoder from scikit-learn to convert categorical features ('cut', 'color', and 'clarity') in the 'diamonds' dataset into numerical representations. This step prepares the data for machine learning algorithms. After encoding, I replaced the original categorical values with their corresponding numerical labels in the DataFrame. Finally, I verified the changes by displaying the first few rows of the DataFrame.

I split the 'diamonds' dataset into training and testing sets to prepare for model training and evaluation. Firstly, I copied the dataset into the 'df' DataFrame. Then, I designated the 'cut' column as the predictor variable ('X') and the remaining columns as features, while isolating the 'cut' column as the target variable ('y'). Using train_test_split from scikit-learn, I divided the data into training and testing sets, allocating 20% of the data for testing and setting a random state of 42 for reproducibility. Finally, I confirmed the dimensions of the training sets by printing their shapes. This split facilitates training machine learning models on the training data and assessing their performance on unseen test data.



These box plots offer a comprehensive view of various attributes of diamonds, shedding light on their distribution patterns and identifying outliers across different variables.

Diamond prices exhibit a right-skewed distribution, with the majority falling below $10,000, yet notable outliers surpassing this threshold. Similarly, the carat weight distribution skews to the right, with most diamonds weighing less than 2 carats, though outliers exceed this weight.

In terms of size, a few extreme outliers are observed, with most sizes below 500 but some notable exceptions reaching up to 4000. The depth percentage displays a relatively normal distribution, with outliers at both ends, and the majority of diamonds falling within the 55% to 65% range. Likewise, the table percentage distribution is somewhat normal, concentrated between 50% and 70%, with some outliers.

Diamond color grading shows a more balanced distribution across grades 0 to 6, indicating varying color levels. Clarity grading follows a relatively normal distribution with outliers ranging from 0 to 7. Lastly, cut quality exhibits a similar normal distribution pattern, with grades ranging from 0 to 4 and occasional outliers.

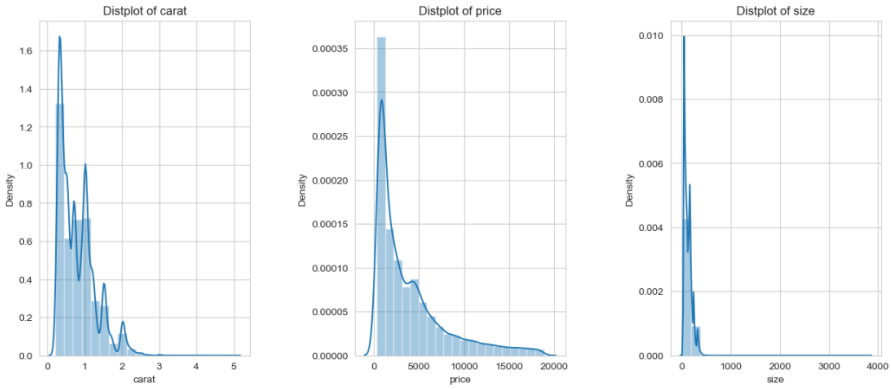
These box plots serve as a valuable tool for visually assessing the distribution and outliers within each variable, facilitating exploratory analysis of diamond data.

Check of correlation:

	price	carat	size	depth	table	color	clarity	cut
price	1.000000	0.921591	0.902385	-0.010647	0.127134	0.172511	-0.071535	0.039860
carat	0.921591	1.000000	0.976308	0.028224	0.181618	0.291437	-0.214290	0.017124
size	0.902385	0.976308	1.000000	0.009157	0.167400	0.284267	-0.206632	0.021440
depth	-0.010647	0.028224	0.009157	1.000000	-0.295779	0.047279	-0.053080	-0.194249
table	0.127134	0.181618	0.167400	-0.295779	1.000000	0.026465	-0.088223	0.150327
color	0.172511	0.291437	0.284267	0.047279	0.026465	1.000000	-0.027795	0.000304
clarity	-0.071535	-0.214290	-0.206632	-0.053080	-0.088223	-0.027795	1.000000	0.028235
cut	0.039860	0.017124	0.021440	-0.194249	0.150327	0.000304	0.028235	1.000000

The analysis unveils key relationships between diamond price and other attributes:

Price is strongly positively correlated with carat weight and diamond size. Conversely, it is negatively correlated with clarity. This suggests that heavier and larger diamonds tend to command higher prices, while those with better clarity, indicating fewer imperfections, tend to be less expensive.



The findings indicate that all three density curves are right-skewed, implying that the mean value is greater than the median value for each distribution.

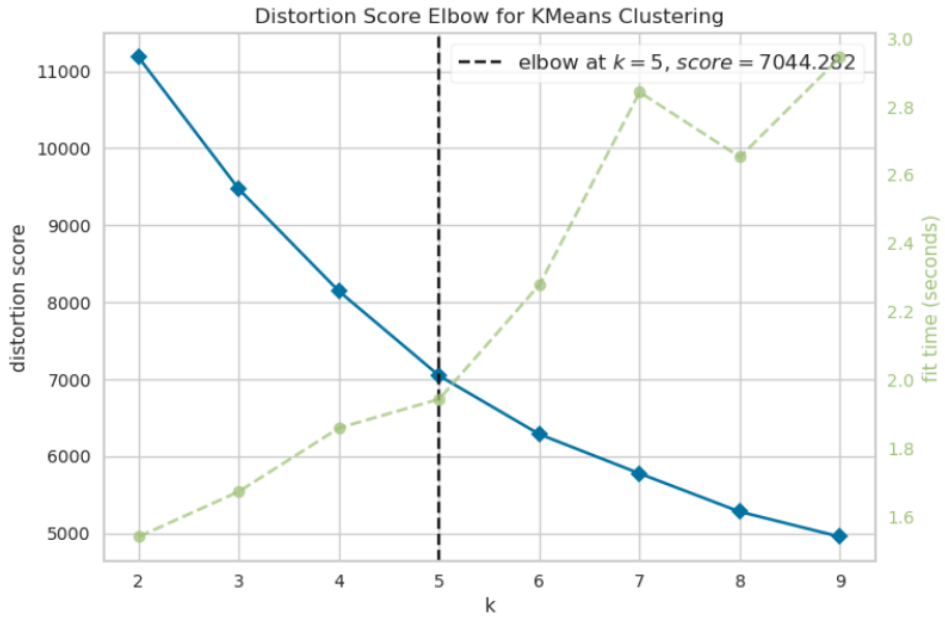
Many outliers were identified in the dataset, making it impractical to remove them as doing so would drastically reduce the amount of available data. Therefore, the decision was made to normalize the data to mitigate the issues caused by outliers.

PCA was utilized to reduce the dimensionality of the data, facilitating accurate cluster visualization. With the implementation of K-means clustering, the aim was to group similar data points into clusters, where "K" signifies the number of clusters. The algorithm entails specifying the cluster count, initializing centroids, iteratively assigning points to their nearest centroids, and updating centroids until stabilization.

The advantages of K-means include its simplicity, flexibility, and efficiency, making it easy to interpret and adjust compared to more complex methods like Neural Networks. It efficiently segments datasets and allows instances to change clusters dynamically as centroids are recomputed, ensuring interpretability in clustering outcomes.

However, K-means has limitations. It lacks an optimal method for determining the number of clusters, requiring predetermined counts for effective clustering. Results can be inconsistent across runs due to the random initialization of centroids. Additionally, it tends to produce clusters of uniform sizes regardless of input data variability and is sensitive to data ordering and scale changes, which can significantly influence results.

For hyperparameter tuning, the Elbow method serves as a common approach to determine the appropriate number of clusters.



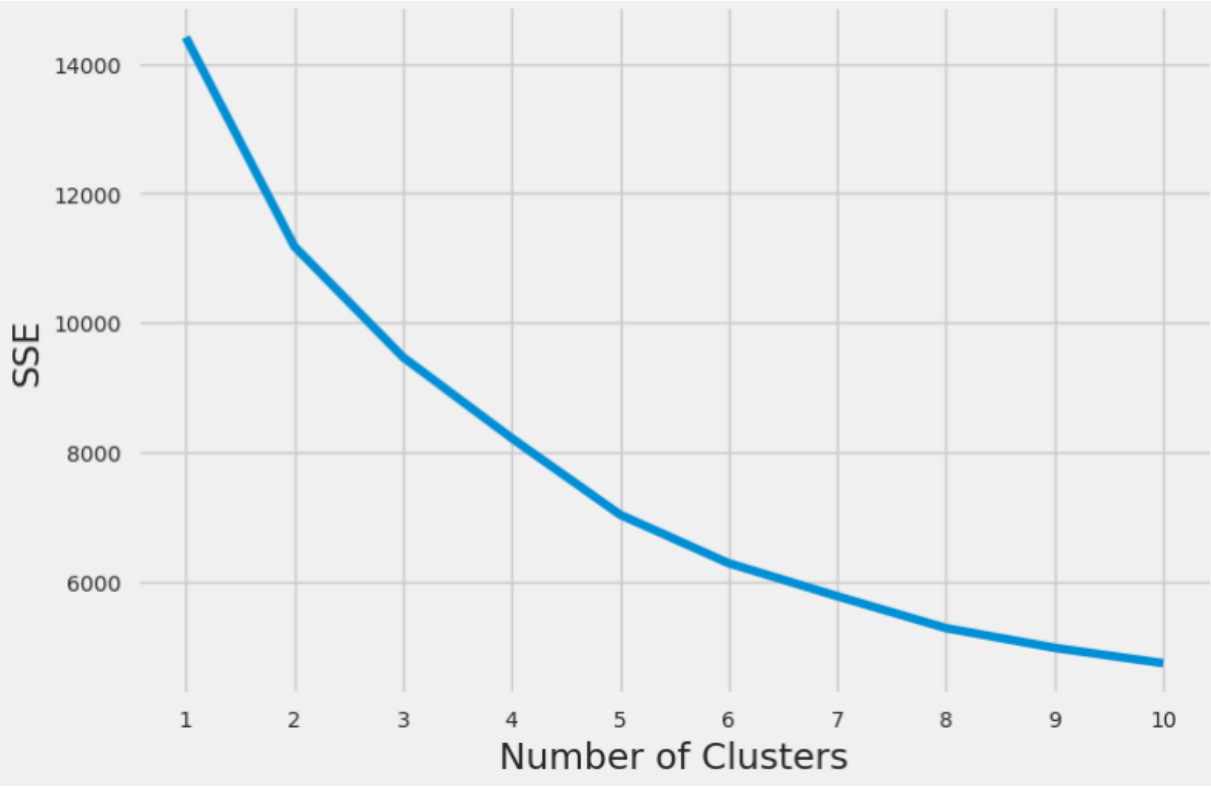
From the graph, I select : k : Clusters = 5

Upon examining the elbow method graph for KMeans clustering, several key observations emerge. The blue line represents the distortion score, reflecting the sum of squared distances from each point

to its cluster center. Initially, the score decreases with increasing clusters, but it plateaus after a certain point, indicating diminishing returns. The green dashed line indicates fit time, which generally rises with more clusters, impacting computational efficiency.

The "elbow" occurs at k = 5, where the distortion score is 7044.282, marked by a vertical dashed line. Beyond this point, additional clusters yield minimal distortion improvements but increase computational costs significantly.

Based on this analysis, k = 5 emerges as the optimal cluster count, balancing distortion reduction with computational efficiency. Therefore, five clusters are the most effective and efficient choice for this clustering analysis.




```
Estimated number of clusters: 5
Estimated number of noise points: 0
```