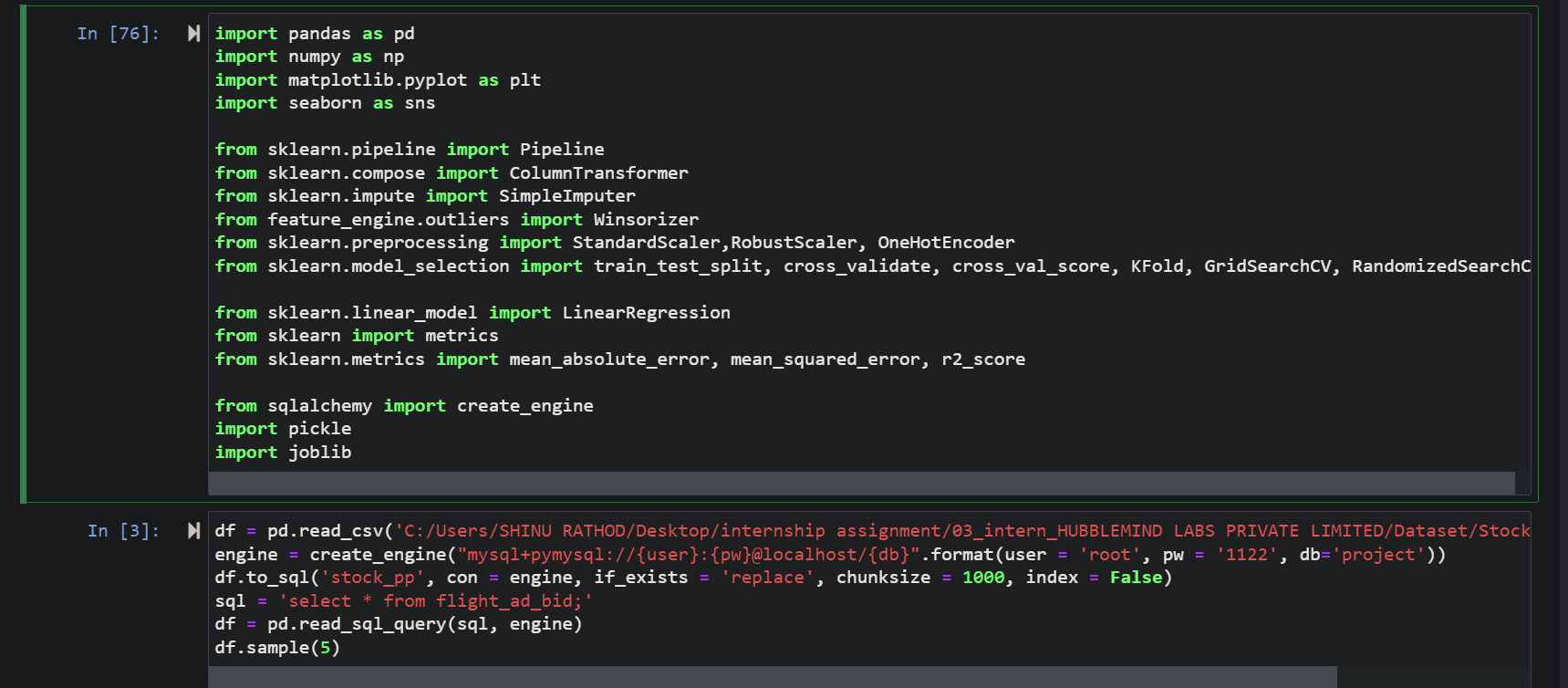
***Task 1: Document the Entire Process***

Task 1: Document the Entire Process

● Create detailed documentation of each step, including data preprocessing, model development, and evaluation.

● Include code snippets, explanations, and visualisations.

**1. Importing all Required lib and Dataset** :

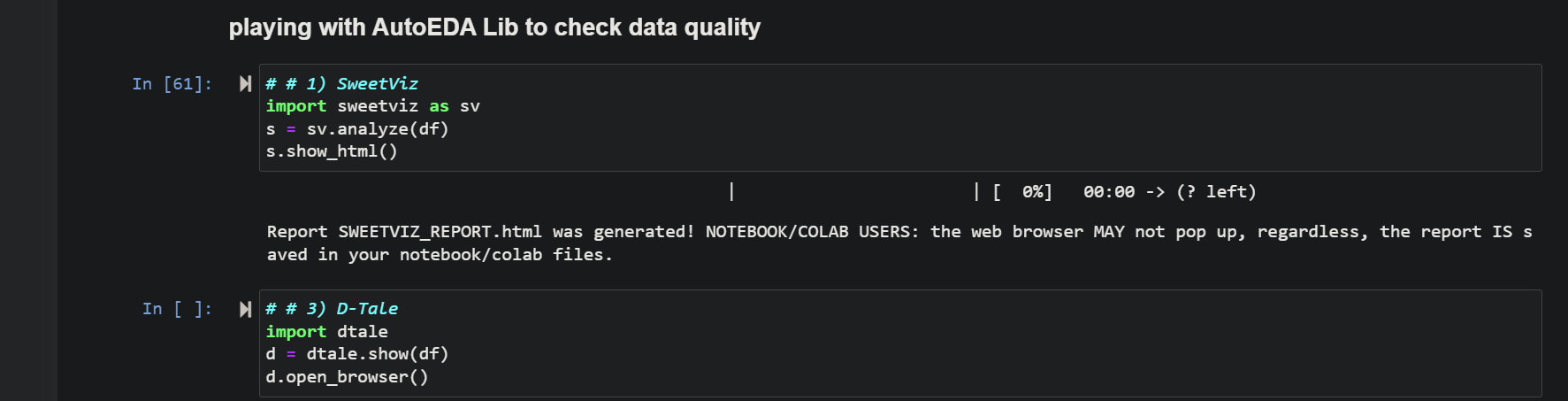


Here importing all required lib and dataset then pushing dataset to mysql database after pushing to mysql database retrieving all data again for further analysis on that data using the sql squery

. because in general we get the dataset from database only we not extract required portion of data from database

. after retrieving data Displaying the first few rows to understand its structure.

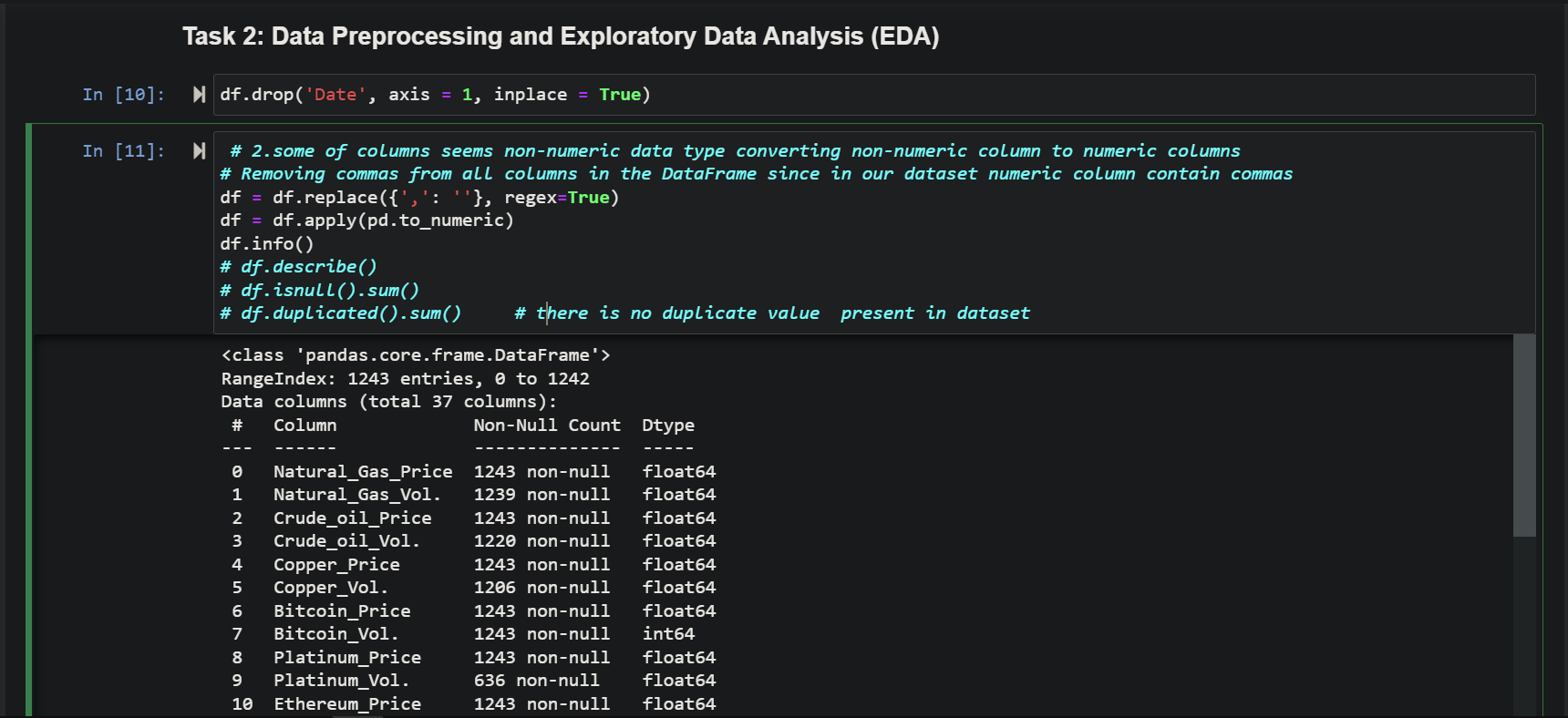
**2. Data Preprocessing** Before diving into model building, I focused on preparing the dataset to ensure it was clean and ready for analysis. This phase, known as data preprocessing and feature engineering, is crucial because the quality of the input data directly affects the performance of the machine learning model.

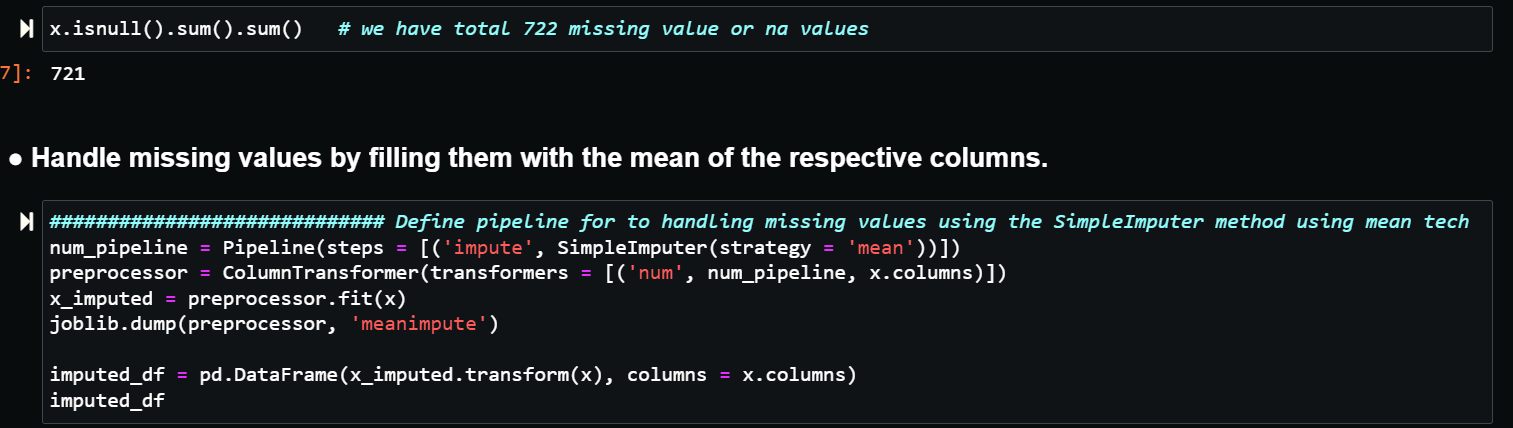
* **Loading the Dataset**: The dataset was loaded into a DataFrame from a CSV file. An initial inspection was done using **df.head(), df.info(),** **and df.describe()** to understand the data types, missing values, and summary statistics.
* **AutoEDA** : Significant time is required in the initial steps to analyze the data. The univariate analysis will reveal a lot of information about the business conditions and the Data. To conserve time in Exploratory Data Analysis (EDA), automation has been introduced with the help of Python libraries.

. here iam using sweetviz and dtale AutoEDA to understand the about actual data in easiest and efficient manner

. first we need to install those package by using pip install command then we can use it

* **Data Cleaning**:

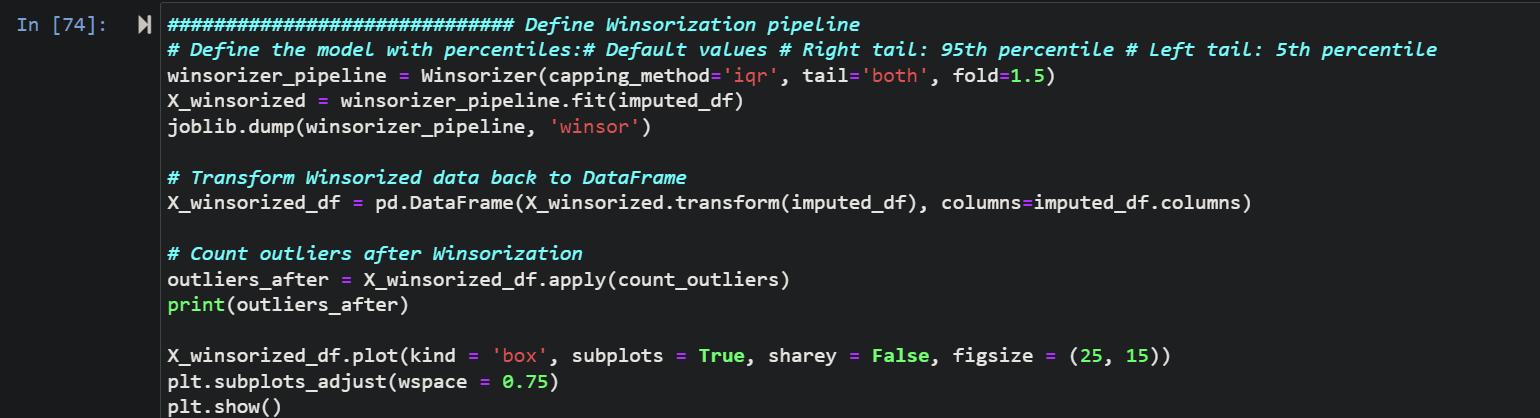
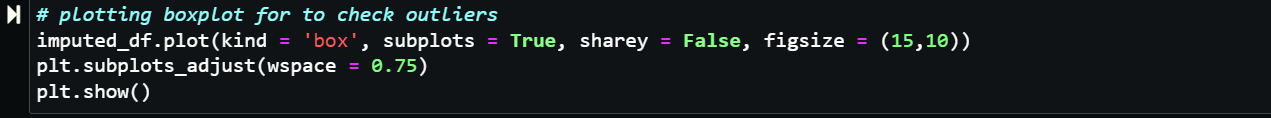
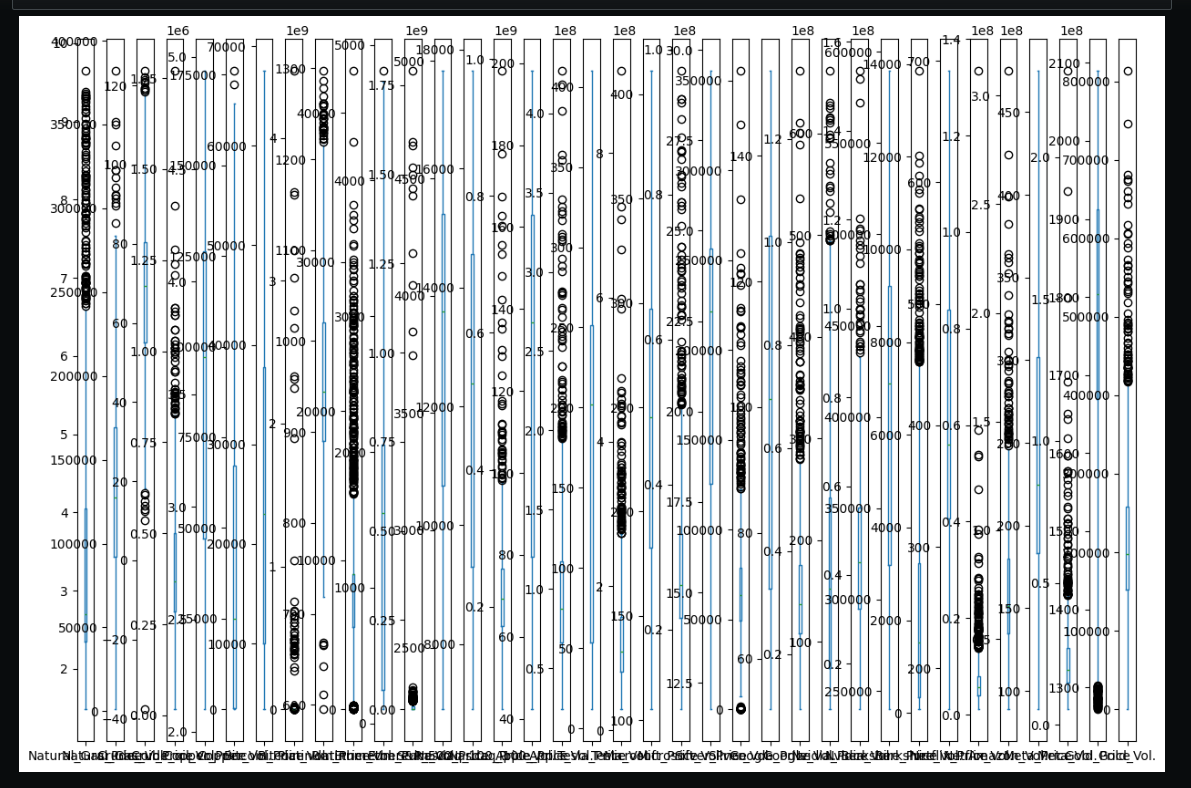


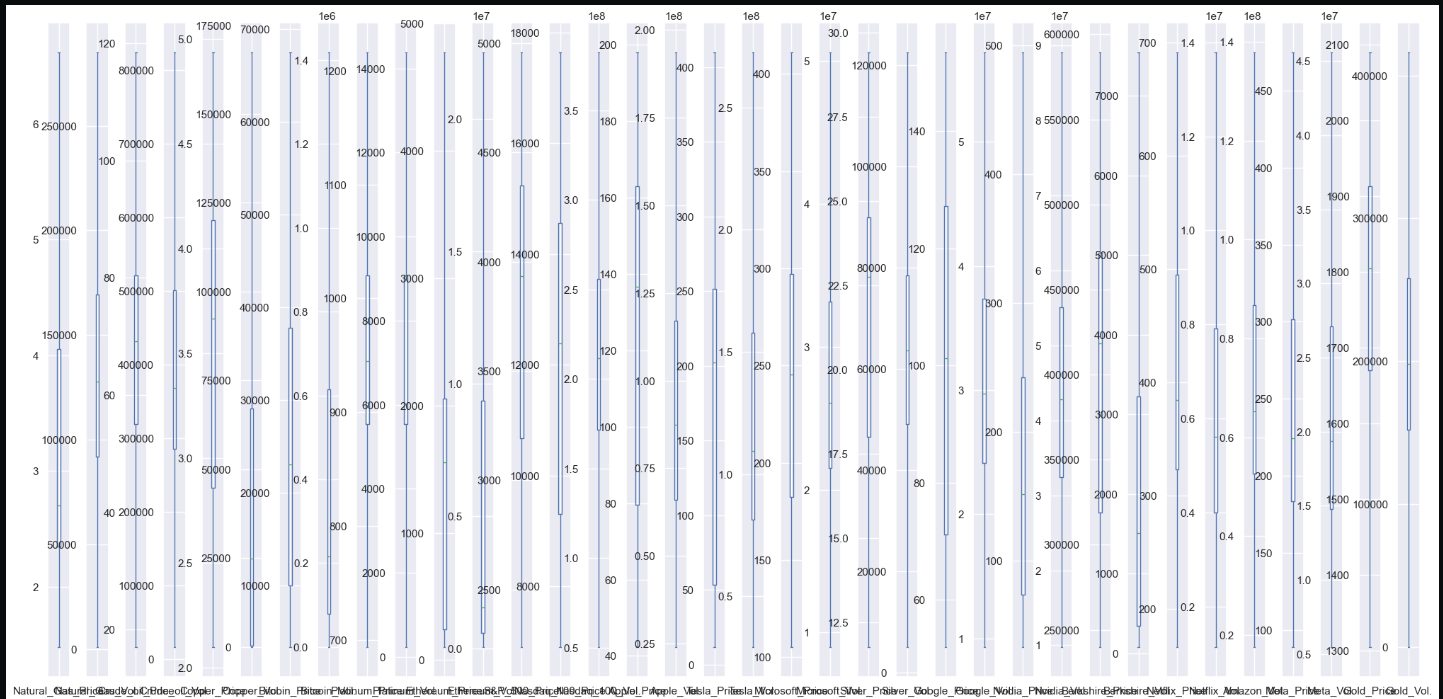
* + **Date Column**: The Date column was dropped as it was not needed for the analysis.
  + **Comma Removal**: Non-numeric columns containing commas were cleaned by replacing commas with empty strings to convert them to numeric.
  + **Numeric Conversion**: The cleaned columns were converted to numeric types using ***pd.to\_numeric().***
  + **Removing Commas and Converting Data Types** : In this dataset, some numeric columns contained commas, which needed to be removed for proper numerical analysis. I replaced these commas with empty strings, effectively cleaning the data. After this, I converted all columns to numeric data types to ensure that the dataset was ready for mathematical operations and statistical analysis
* **Missing Values**: The first step was to address any missing values in the dataset. Missing data can lead to inaccurate predictions if not handled properly. I used a technique called imputation to fill in the missing values. Specifically, I applied the median imputation method, which replaces missing values with the median of the respective column. This method is less sensitive to outliers compared to mean imputation and helps in maintaining the integrity of the data. Missing values were imputed using the mean strategy through a pipeline with ***SimpleImputer****.*

. here handling missing values using mean since I have found total 721 missing values in dataset

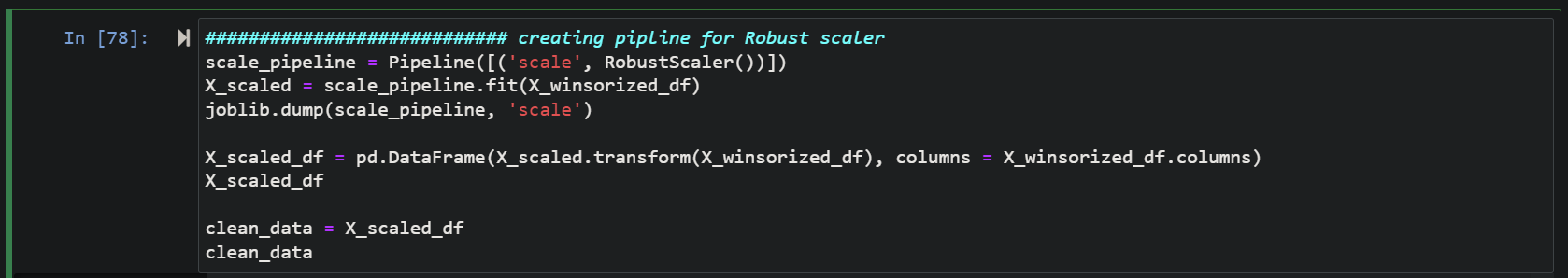
* **Outlier Treatment**:

**Winsorization**: In this dataset, some numeric columns contained commas, which needed to be removed for proper numerical analysis. I replaced these commas with empty strings, effectively cleaning the data. After this, I converted all columns to numeric data types to ensure that the dataset was ready for mathematical operations and statistical analysis.

* + A Winsorizer was applied to limit extreme values to reduce the effect of outliers. Before applying winsorization tech plotted boxplot we can see the there is outliers found outlier after plotting box plot we can see the outliers in entire dataset
  + 

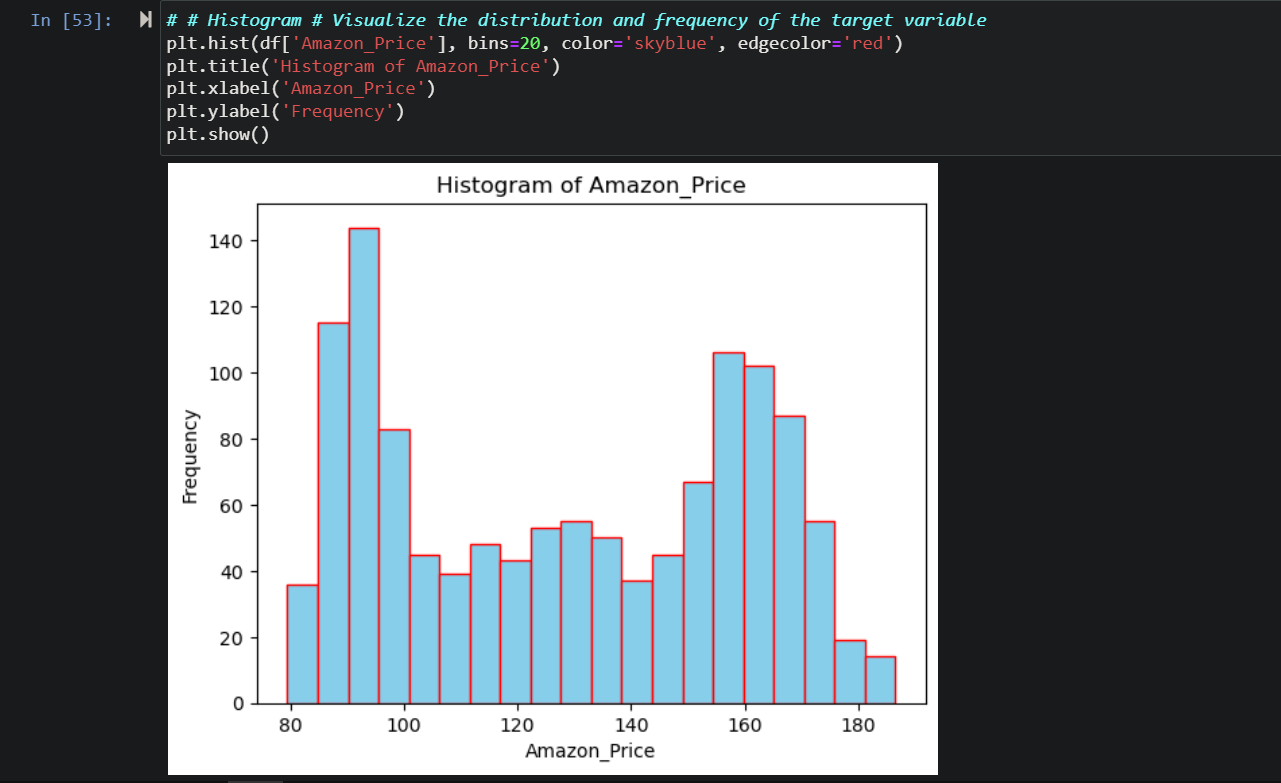
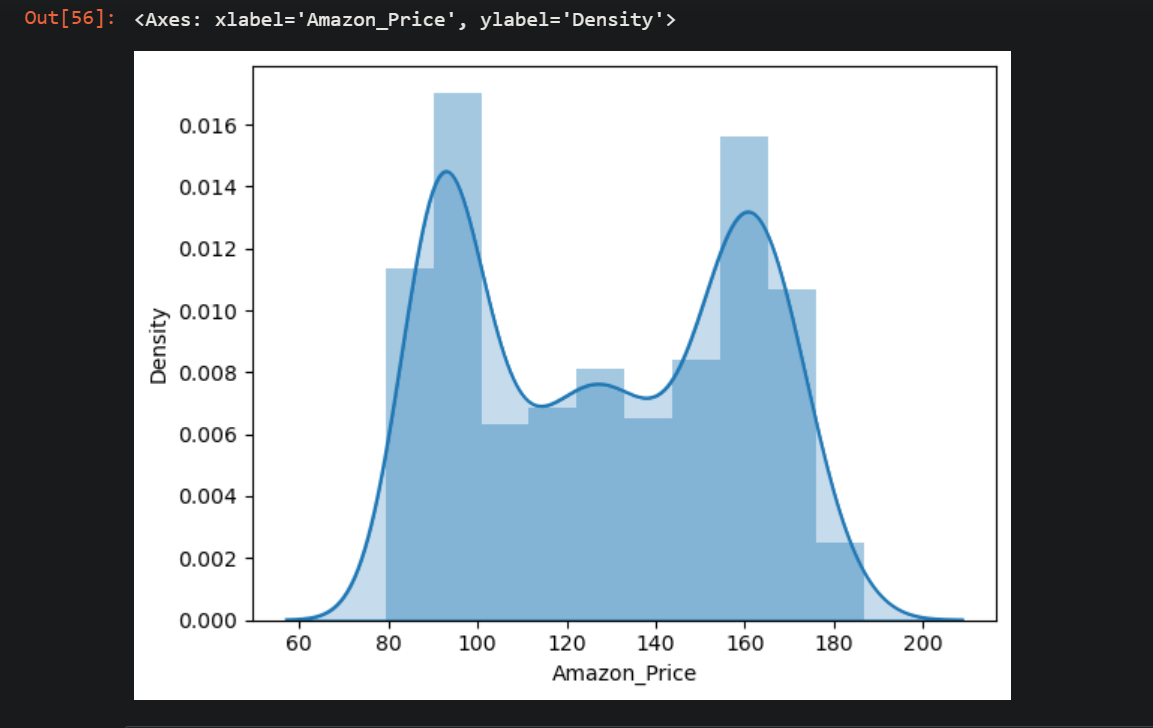
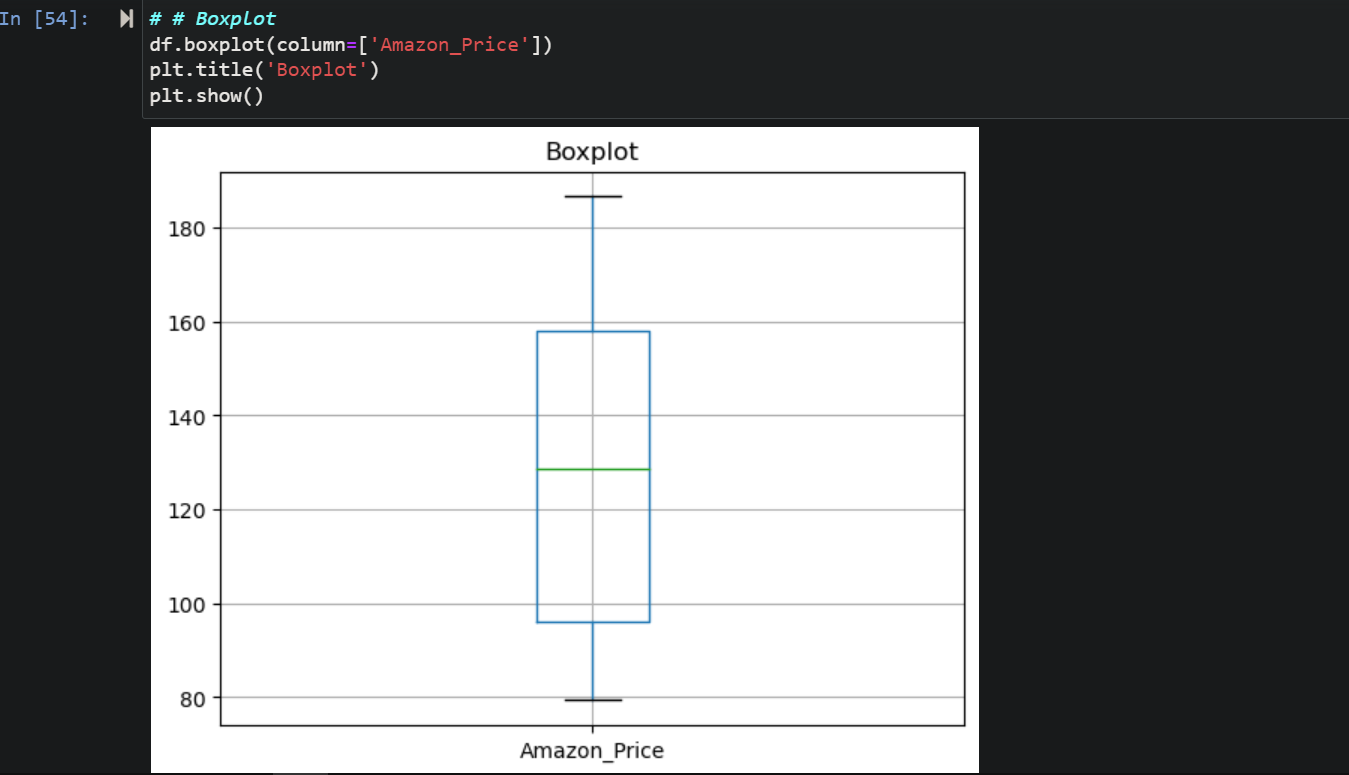
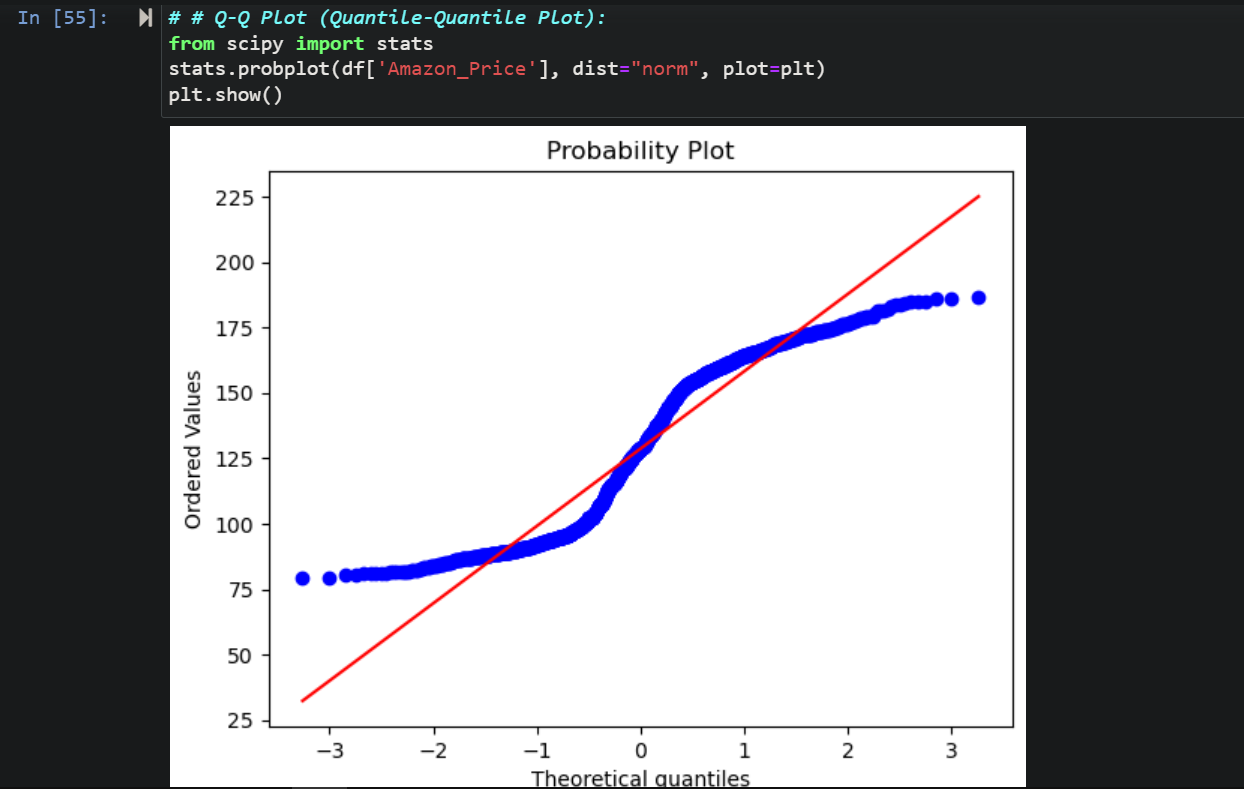
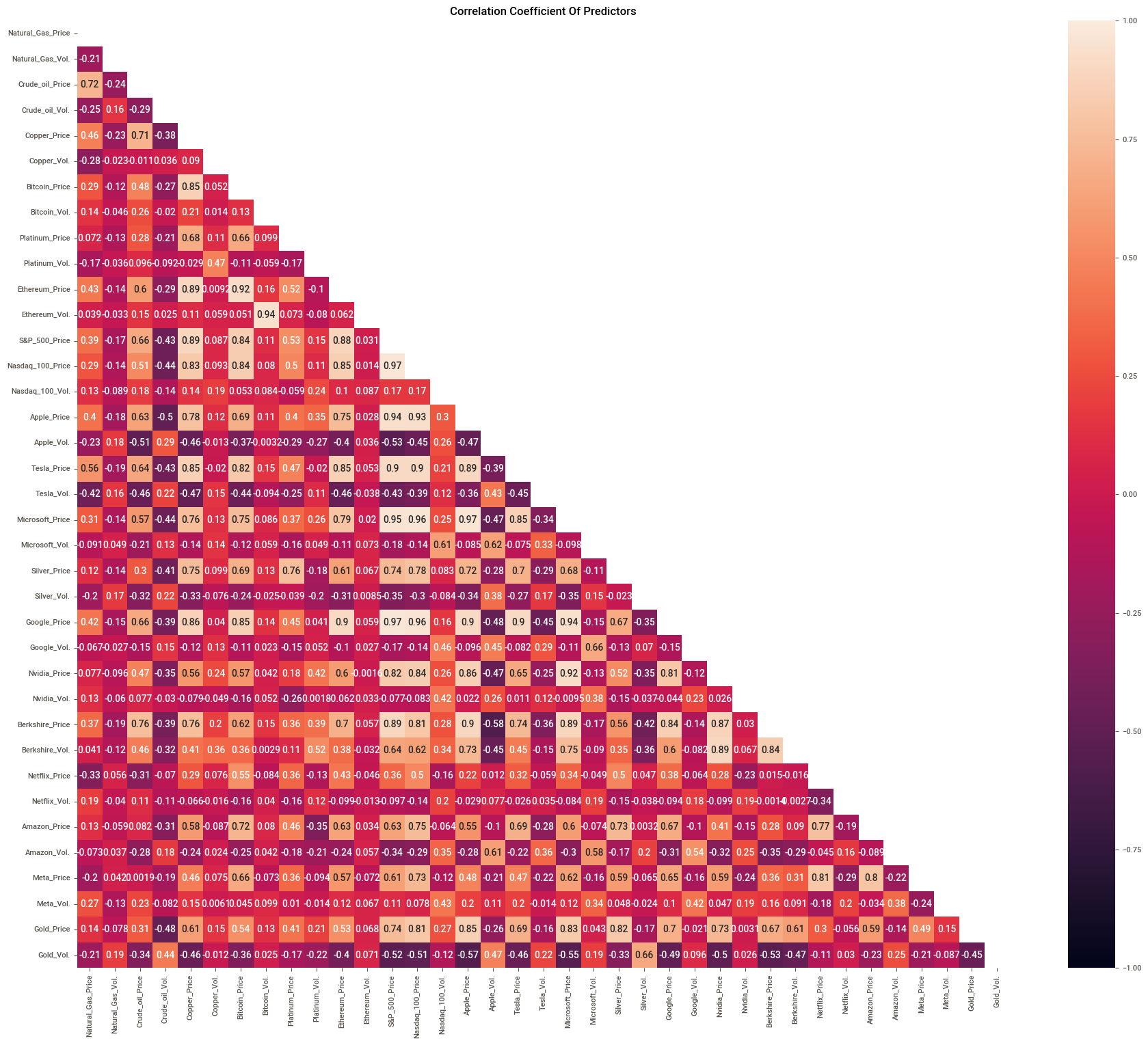
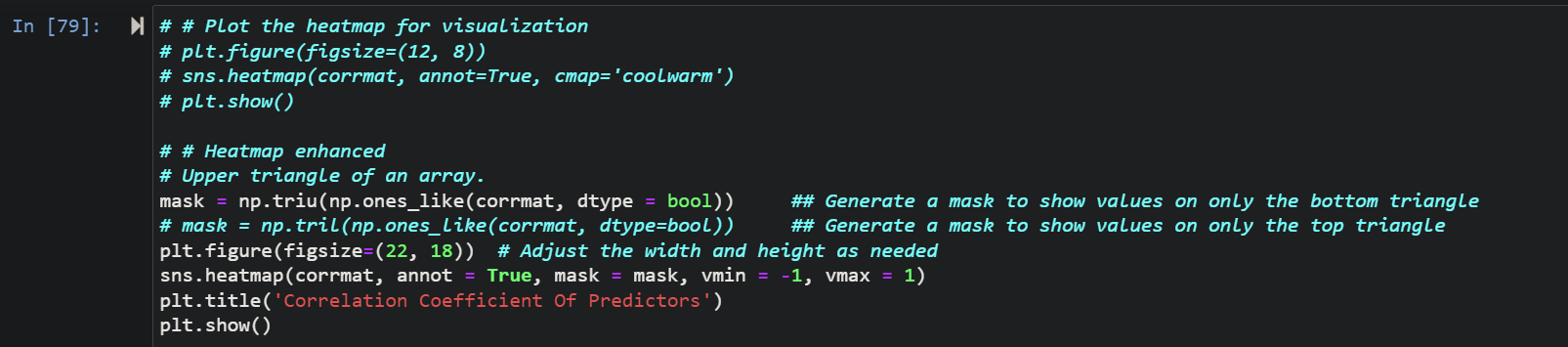
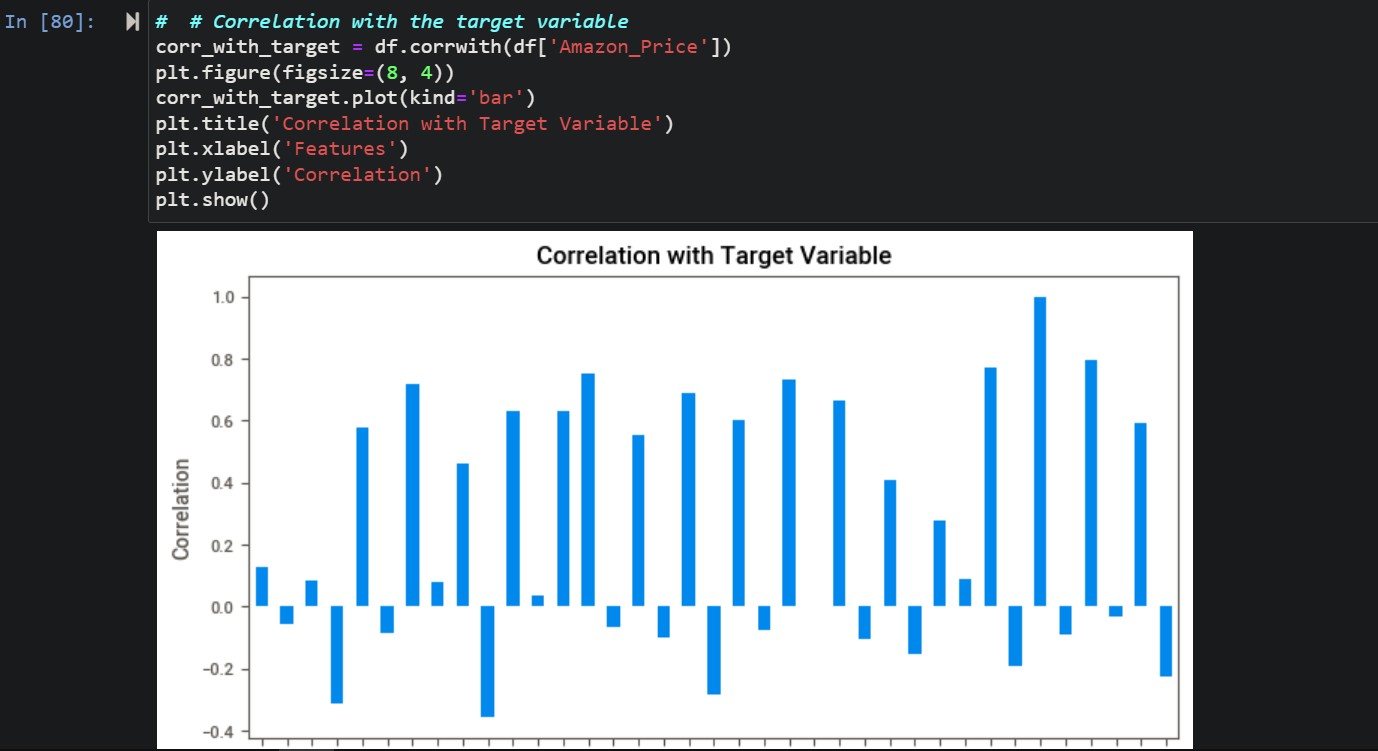
. after applying winsorization with capping method ‘iqr’ with both side folding her in the plot there no extreame value available 

* **Feature Scaling**: After handling outliers, the next step was to scale the features. Feature scaling is essential because it ensures that all features contribute equally to the model. I used the RobustScaler, a scaling technique that is particularly effective in dealing with outliers. This scaler transforms the data by removing the median and scaling it according to the interquartile range, making the dataset more robust to outliers.

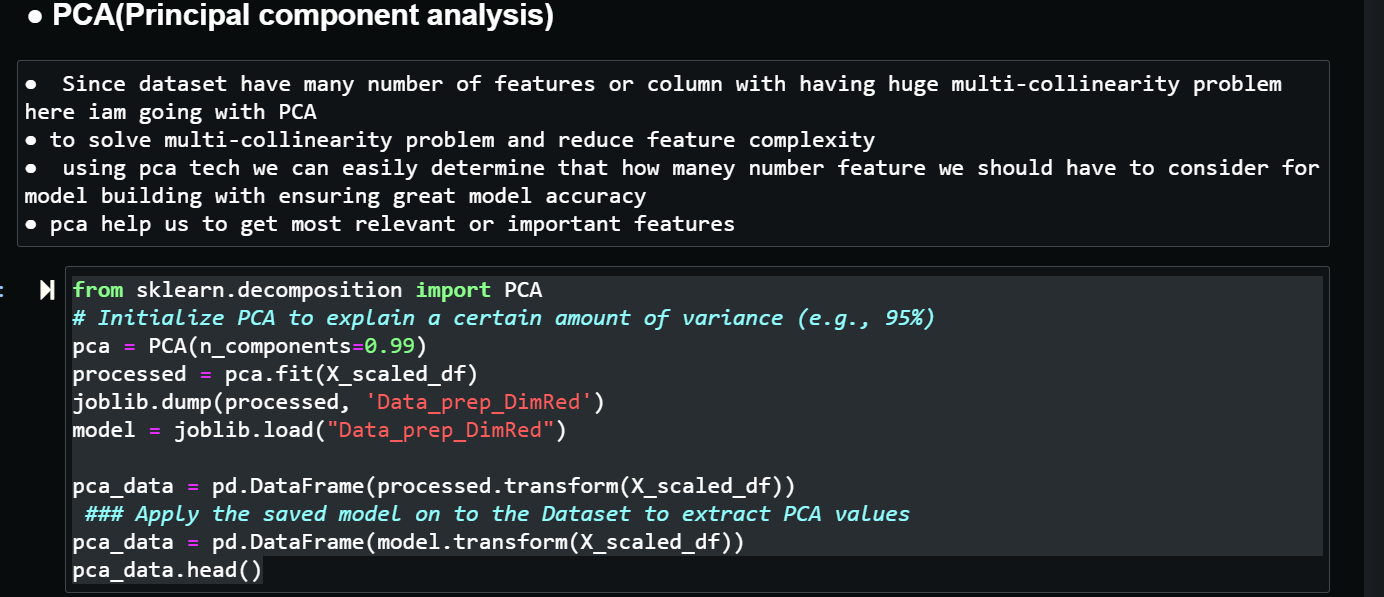
**Robust Scaling**: The features were scaled using ***RobustScaler*** to handle the presence of outliers more effectively.

. here I applied RobustSclaler tech to scale the features value in same formate

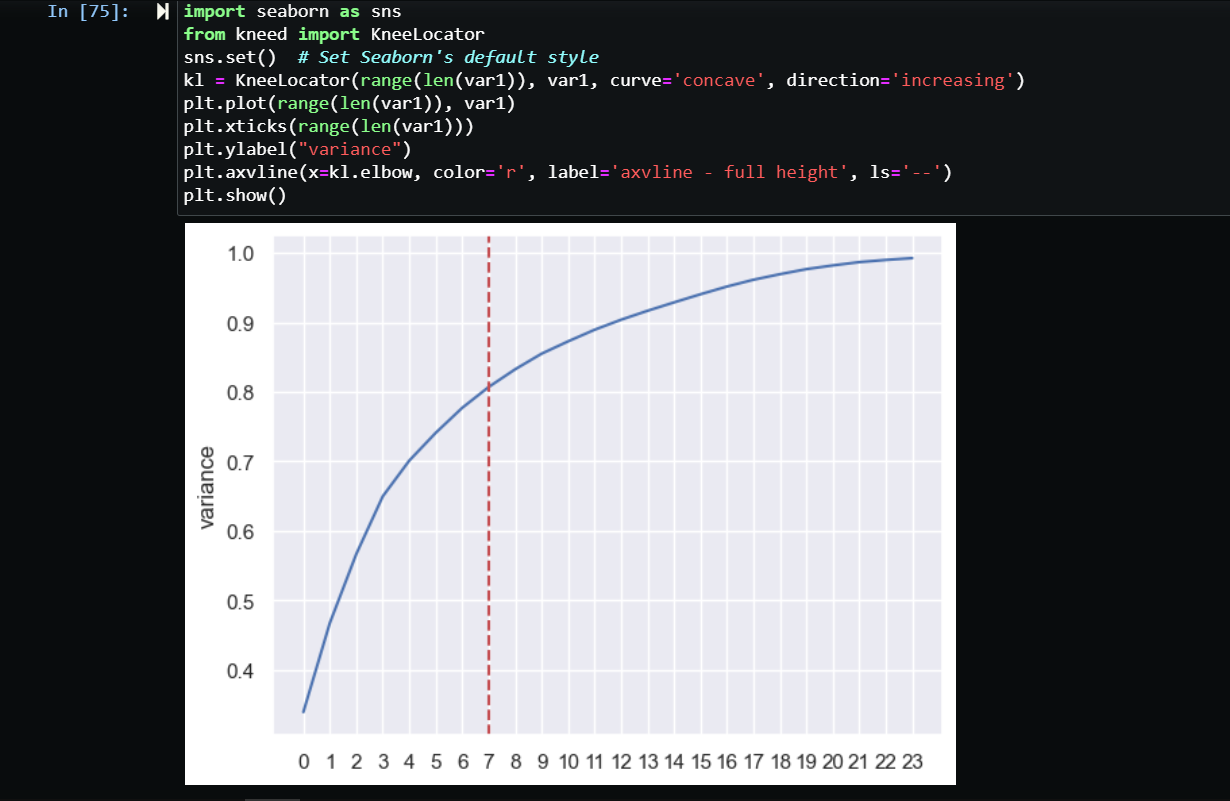
**3. Exploratory Data Analysis (EDA)**

* **Univariate Analysis**:
  + **Histogram**:  The distribution of **Amazon\_Price** was visualized to understand its frequency distribution.
  + **Boxplot**: A boxplot was created to visualize the presence of outliers.
  + **P-P Plot and Q-Q Plot**: These plots were used to assess the normality of the Amazon\_Price distribution.
* **Bivariate Analysis**:
  + **Correlation Matrix**: A heatmap was generated to visualize the correlations between the features.
  + **Correlation with Target Variable**: A bar plot was created to show the correlation of each feature with the target variable Amazon\_Price.
* **Multivariate Analysis**:
  + **Pairplot**: A pairplot was used to visualize relationships between multiple features.

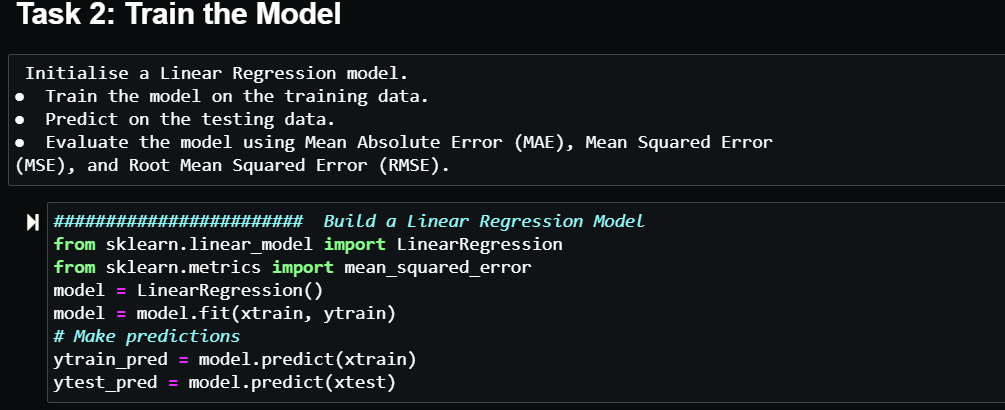
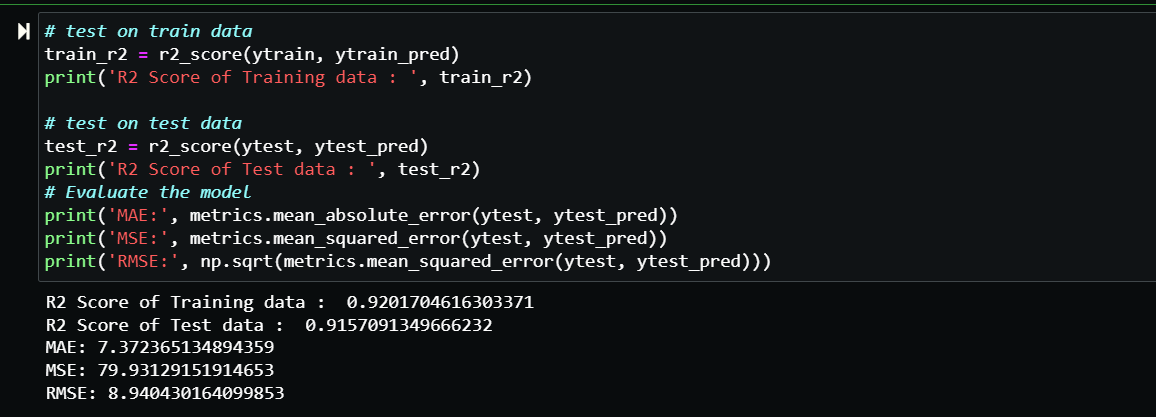
**4. Feature Selection**

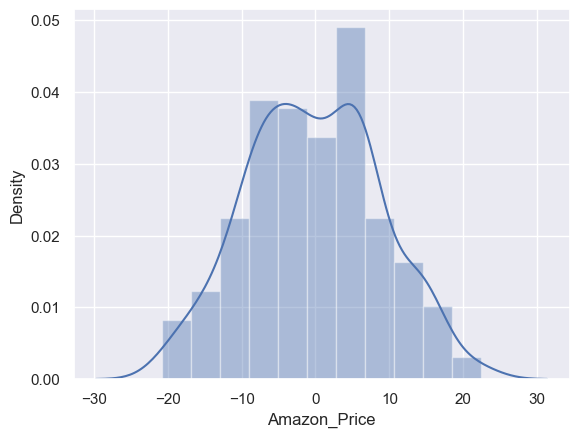
1. **Correlation Analysis** : Performed correlation analysis to identify and drop features with high correlations to reduce multicollinearity. Ensured that p-values of coefficients were below 0.05, as insignificant p-values could indicate multicollinearity.
2. **Variance Inflation Factor (VIF)** : Calculated VIF to identify and drop features with high multicollinearity. Used the rule of thumb: VIF > 10 indicates high multicollinearity.
3. **Regularization** : Applied regularization techniques to prevent multicollinearity, model complexity, and overfitting due to the large number of columns in the dataset
4. **Principal Component Analysis (PCA)** : In the final model, PCA was primarily used to reduce dimensionality, focusing on retaining fewer columns with high information content.Influence Plot: With the data cleaned, scaled, and imputed, I performed Principal Component Analysis (PCA) to reduce the dimensionality of the dataset. High-dimensional data can lead to overfitting, where the model performs well on training data but poorly on unseen data. PCA helps in selecting the most important features that capture the majority of the variance in the data while discarding the less important ones. By retaining 99% of the variance, I was able to reduce the number of features while still keeping most of the important information intact.
5. . Pca tech to reduce the dimension helps to get maximum information using minimal number of pcs(princimal components less number of pcs capture most of information

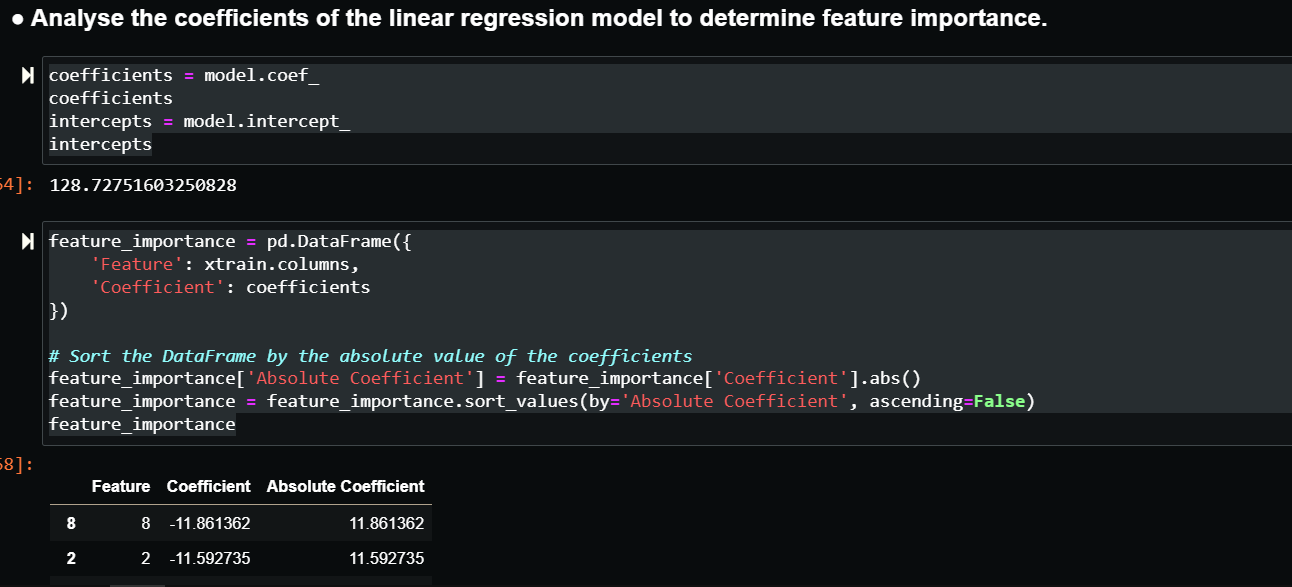
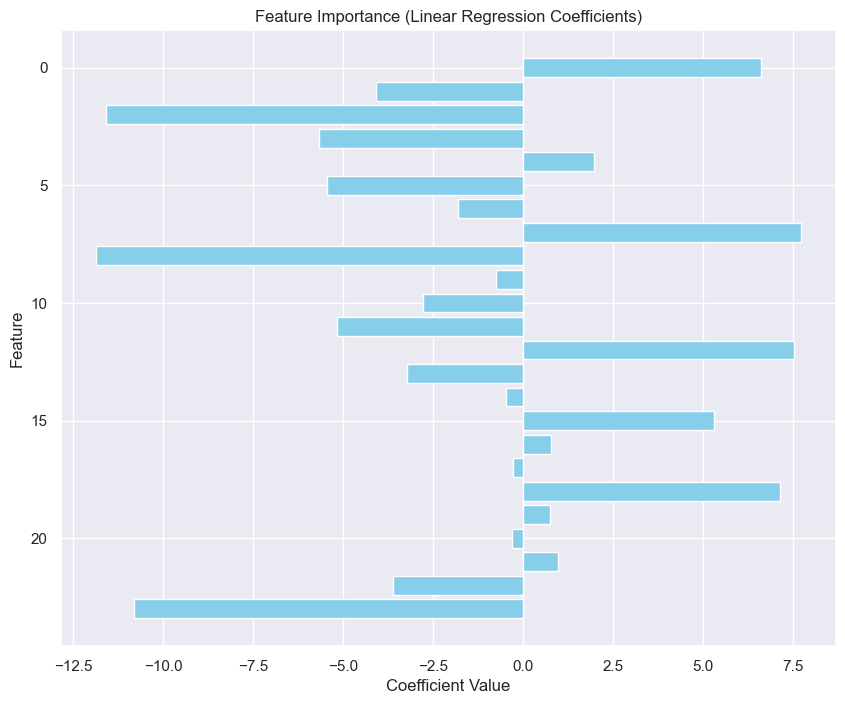
. got great accuracy but again here also we need consider more pcs like around 23 pcs if i consider this many pcs then only pcs are gathering more than 95% infomation or variance

. elbow curve or scree plot tech used for to consider number of pcs

**5. Model Development**

* **Train-Test Split**: The data was split into training and testing sets with an 80-20 ratio.
* **Model Building**:
  + **LinearRegression**: A linear regression model was built using **linearRegression()** to predict **Amazon\_Price.** 
  + **Model Evaluation**: 
    - **R-squared**: The model's R-squared value was calculated for both training and testing datasets.
    - **Root Mean Squared Error (RMSE)**: RMSE was calculated for both training and testing datasets to evaluate model accuracy.
    - **Prediction plot** : plotting the distplot of error **sns.distplot(Y\_test-ytest\_pred)**



Task 3: Feature Importance Analysis importance.   
● Analyse the coefficients of the linear regression model to determine feature ● Visualise the feature importances using a bar plot. 

* The coefficient value indicates how much the target variable is expected to change with a one-unit change in the feature.
* Positive coefficients indicate a direct relationship, while negative coefficients indicate an inverse relationship.
* The absolute value of the coefficient shows the strength of the relationship.

This analysis can give you a good understanding of which features are most influential in predicting the target variable.

**6. Model Testing on New Data**

* The developed model was tested on a new dataset to predict the target variable after applying the same preprocessing steps (imputation, Winsorization, scaling) used in the model development phase.