**Explanation of the Code**

**1. Data Preprocessing:**

- Loading Data: The dataset is loaded using 'pd.read\_csv("Air Conditioners.csv")'.

- Data Cleaning: The price columns (discount\_price, actual\_price) and no\_of\_ratings column are cleaned by removing non-numeric characters and converted into numeric data types.

- Handling Missing Data: The 'no\_of\_ratings', 'discount\_price', 'actual\_price' and 'ratings' column had missing values, so these were filled with 0.

**2. Univariate Analysis:**

- Product Ratings Distribution: The 'ratings' column is plotted as a histogram to analyze the frequency of different ratings among products.

- Discount Prices Distribution: The 'discount\_price' column is also plotted as a histogram to explore the spread of prices.

**3. Bivariate Analysis:**

- Relationship Between Ratings and Number of Ratings: A scatter plot shows how the number of ratings correlates with the product's average rating.

- Comparison of Discount Price and Actual Price: A scatter plot compares the actual price and the discount price, showing how much discount is generally applied.

- Correlation Between Discount Price and Ratings: A scatter plot visualizes the relationship between product ratings and discount prices.

**4. Random Forest Classifier:**

- Feature Selection: Three features ('no\_of\_ratings', 'discount\_price', 'actual\_price') are selected as predictors(input), and the 'ratings' column is binarized to use as the (output)target (classification: rating >= 4 is considered 1, otherwise 0).

- Train-Test Split: The data is split into training and test sets.

- Data Scaling: The features are standardized using 'StandardScaler' to ensure all features have equal importance during model training. both input and outputs taken here.

- Random Forest for Feature Selection: A 'RandomForestClassifier' is trained on the dataset.

- Model Training: A Random Forest classifier is then trained on the selected features.

- Model Evaluation: The model's accuracy, confusion matrix, and classification report are printed to assess its performance.

**5. Saving the Model:**

- The trained model, scaler, and feature selector are saved as '.sav' files using the 'pickle' library. This allows the model to be loaded later for prediction without retraining.