**SARIMAX model**

Certainly! The code you provided performs sales forecasting using the SARIMA (Seasonal Autoregressive Integrated Moving Average) model. Here's a simplified explanation of the code:

1. Import the necessary libraries: pandas, numpy, SARIMAX from statsmodels, and matplotlib.pyplot.

2. Generate dummy data for product sales until June 2023. The data includes product names, categories, subcategories, dates, and the amount to be paid.

3. Ensure that each product has sales recorded for each month until June 2023 by reindexing the data based on the product names and a monthly date range.

4. Aggregate the data to get the total sales per month by grouping it based on the date and summing the units purchased.

5. Set the date column as the index for further analysis.

6. Initialize the SARIMA model with the desired order and seasonal order. In this case, the order is (1, 1, 1), and the seasonal order is (1, 1, 0, 12).

7. Fit the SARIMA model to the historical sales data.

8. Generate a sales forecast for the remaining months of 2023 and 2024 using the SARIMA model. The forecast steps are set to 18, indicating 18 months into the future.

9. Retrieve the forecasted sales and confidence intervals from the forecast results.

10. Print the forecasted sales.

11. Plot the historical sales data and the forecasted sales on a line graph using matplotlib. The x-axis represents the dates, and the y-axis represents the units purchased. The graph shows the historical sales as a line and the forecasted sales as another line.

**CROSTON Model**

Certainly! Here's a simplified explanation of the code:

1. First, we generate dummy data for sales of different products until June 2023. The data includes product names, categories, subcategories, dates, and the number of units purchased.

2. Next, we aggregate the data to get the total sales per month. This gives us a time series of monthly sales.

3. The Croston's method is a forecasting technique that is specifically designed for intermittent demand data, where there are periods of zero demand. It takes into account the sporadic nature of such demand and provides a forecast for future periods.

4. We define a function called `croston\_forecast` that implements Croston's method. It takes the historical sales data, alpha (smoothing parameter for level), beta (smoothing parameter for trend), and the number of forecast periods as input.

5. Inside the `croston\_forecast` function, we calculate the level (a) and trend (b) components using the Croston's method formulas. We iterate over the historical demand data to update the level and trend values.

6. Once we have the level and trend components, we use them to generate the forecasted sales. We create an array to store the forecasted sales values.

7. Finally, we plot the historical sales data and the forecasted sales using the generated Croston's forecast. The x-axis represents the dates, and the y-axis represents the number of units purchased. We also print the forecasted sales for the remaining months of 2023 and the whole year of 2024.