Explanation

Import Required Libraries:
yfinance: A library to fetch historical stock price data from Yahoo Finance.
pandas: A powerful data manipulation library.
numpy: A library for numerical operations.
statsmodels.tsa.api.ExponentialSmoothing: This is from the statsmodels library and provides functions to perform Exponential Smoothing.

List of Company Tickers:
A list named companies contains the ticker symbols of companies (e.g., "MSFT" for Microsoft, "AAPL" for Apple, etc.) for which you want to perform the forecasting.

Initialize Results Dictionary:
A dictionary named results is initialized to store the forecasting results.

Loop Through Companies:
The code iterates through each company ticker in the companies list.

Fetch Historical Data:
Using the yfinance.Ticker object, historical stock price data for the last 5 years with a weekly interval is fetched.

Prepare Data:
The "Close" prices are extracted from the historical data and converted into a pandas DataFrame.
The "Date" column is set as the index.
Missing values are filled using forward filling.
The data is resampled to a weekly frequency and aggregated using the mean.

Split Data:
The data is split into training and testing sets. 80% of the data is used for training, and the rest is used for testing.

Model Configurations and Forecasting:
The script defines several configurations to try for Exponential Smoothing models, including "additive," "multiplicative," and None (no trend/seasonality).
For each configuration, the script iterates through different model names, and for each model, it fits the Exponential Smoothing model to the training data.
If the configuration is None, an Exponential Smoothing model is created without any trend or seasonality.
If the configuration is "add," a model with additive trend and seasonality is created.

If the configuration is "mul," a model with multiplicative trend and seasonality (using Box-Cox transformation) is created.
The script then tries to fit the model with the training data and forecast the values for the test data.

Calculate Forecasting Metrics:
Metrics like Mean Percentage Error (MPE), Weighted Mean Percentage Error (WMPE), Mean Absolute Percentage Error (MAPE), and Weighted Mean Absolute Percentage Error (WMAPE) are calculated for each model.

Store Results:
The calculated metrics are stored in the results dictionary with keys (company, config, model_name).

Printing Results:
The calculated metrics are printed for each model configuration and model name.

Creating DataFrame and Saving Results:
After iterating through all companies, configurations, and models, the script creates a pandas DataFrame from the results dictionary.
The DataFrame is then saved to an Excel file named 'exponential_smoothing_results.xlsx'.