Project Title: Monthly Sales Forecasting Using Time Series Analysis

---

Introduction:

This project aims to forecast future monthly sales for a business using time series data. The goal is to provide accurate predictions that can help in decision-making, stock management, and overall business planning.

Step 1: Data Collection and Initial Exploration

In the first step, I searched for a suitable dataset that contains monthly sales figures. Once collected, I performed initial cleaning and formatting of the data using the following tools:

- **Pandas:** for data manipulation and cleaning.

- **Matplotlib & Seaborn:** for visualizing the data trends and patterns.

Tasks:

- Converted the date column to datetime format.

- Aggregated daily data into monthly data using .resample('M').

- Checked for null values and handled them appropriately.

Step 2: Exploratory Data Analysis (EDA)

The EDA phase helped me extract insights from the dataset and formulate analytical questions such as:

- What are the peak months in terms of sales?

- Are there any seasonal trends?

- Are there outliers or unusual patterns?

```
# show it on bar figure
top_selling_category.plot(kind ="bar")
plt.title("Tpo 5 Selling Category In SuperStore")
plt.xlabel('Category Name')
plt.ylabel('Total Selling')
plt.show()
```
✓ 0.1s
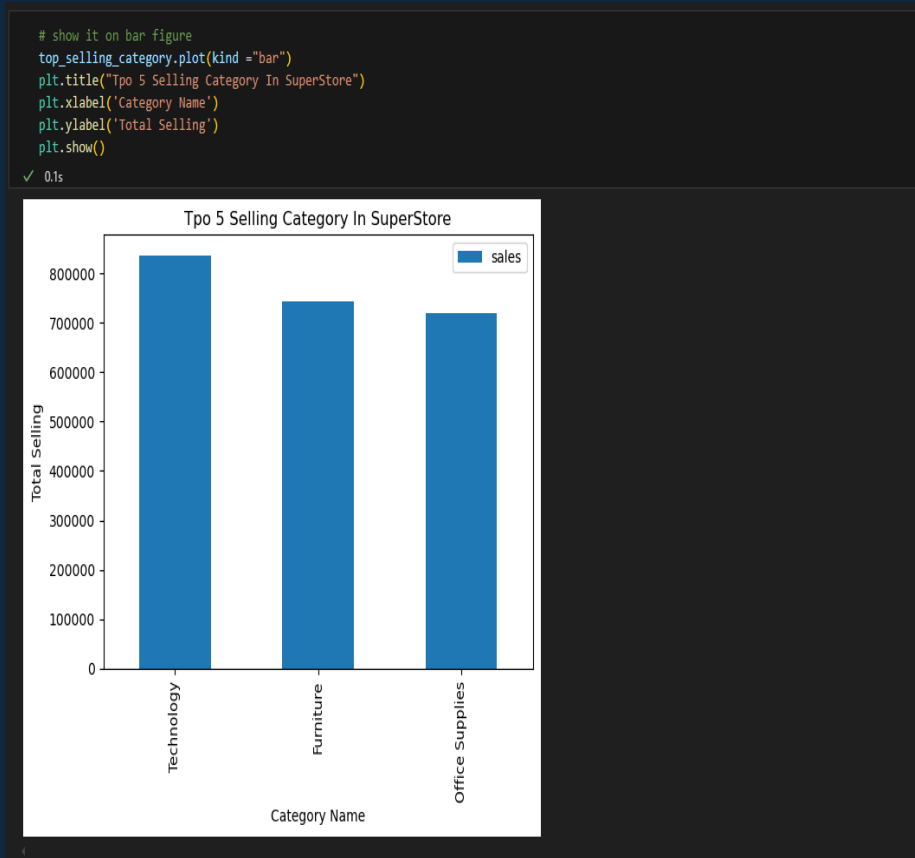
```
# show it on bar figure
top_5_selling_products.plot(kind ="bar")
plt.title("Tpo 5 Selling Products In SuperStore")
plt.xlabel('Product Name')
plt.ylabel('Total Selling')
plt.show()
✓  0.2s
```
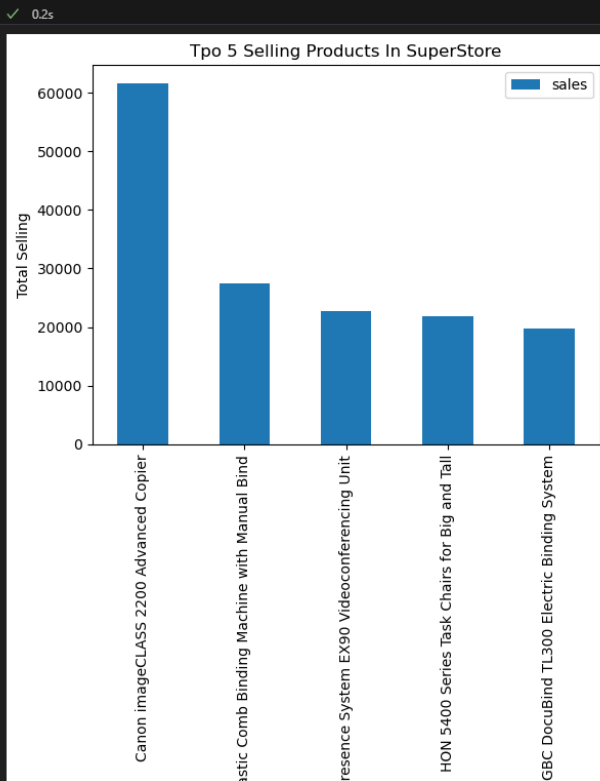

Libraries Used:

- **Matplotlib & Seaborn:** for plotting sales trends and distributions.

- **Statsmodels:** for conducting stationarity tests like ADF (Augmented Dickey-Fuller).

Insights:

- There is a clear upward trend in sales.

- Some seasonality might exist in the data.

Step 3: Deep Analysis

In this phase, I examined the statistical properties of the time series.

Tasks:

- Conducted ADF Test → confirmed the series is non-stationary.

- Differenced the series to make it stationary.

- Used auto_arima() to identify the best parameters.

Libraries Used:

- **pmdarima:** for auto_arima() model selection.

- **Statsmodels:** for SARIMAX modeling and diagnostics.
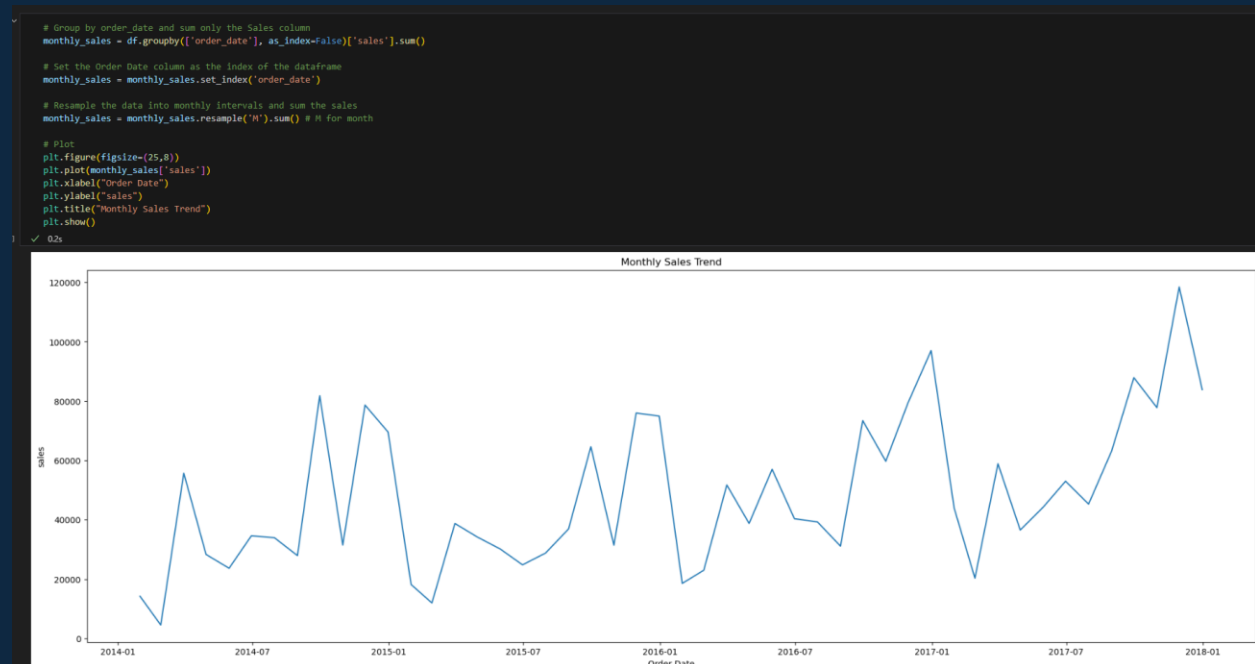
Insights:

- Best model suggested: **ARIMA(0,1,1)**

Step 4: Model Building and Evaluation

Based on the data nature, I concluded that the suitable models should be from the ARIMA family since:

- The data is univariate.

- Exhibits trend and possibly some seasonality.

Models Tried:

- ARIMA(1,1,0)

- ARIMA(0,1,1)

- SARIMA(1,1,1,12)

Evaluation Criteria:

- AIC, BIC, RMSE values

- Residual diagnostics

Best Model:

- **ARIMA(0,1,1)** had the lowest AIC and stable residuals.

Libraries Used:

- **statsmodels.tsa.statespace.SARIMAX**

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                   48
Model:                 ARIMA(0, 1, 1)   Log Likelihood                -540.306
Date:                Thu, 15 May 2025   AIC                           1084.612
Time:                        15:33:18   BIC                           1088.313
Sample:                    01-31-2014   HQIC                          1086.005
                         - 12-31-2017
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.6534      0.123     -5.315      0.000      -0.894      -0.412
sigma2      5.981e+08   2.21e-13    2.7e+21      0.000    5.98e+08    5.98e+08
===================================================================================
Ljung-Box (L1) (Q):                   0.02   Jarque-Bera (JB):                 0.15
Prob(Q):                              0.88   Prob(JB):                         0.93
Heteroskedasticity (H):               1.05   Skew:                            -0.00
Prob(H) (two-sided):                  0.92   Kurtosis:                         2.73
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
[2] Covariance matrix is singular or near-singular, with condition number 5.75e+39. Standard errors may be unstable.
```
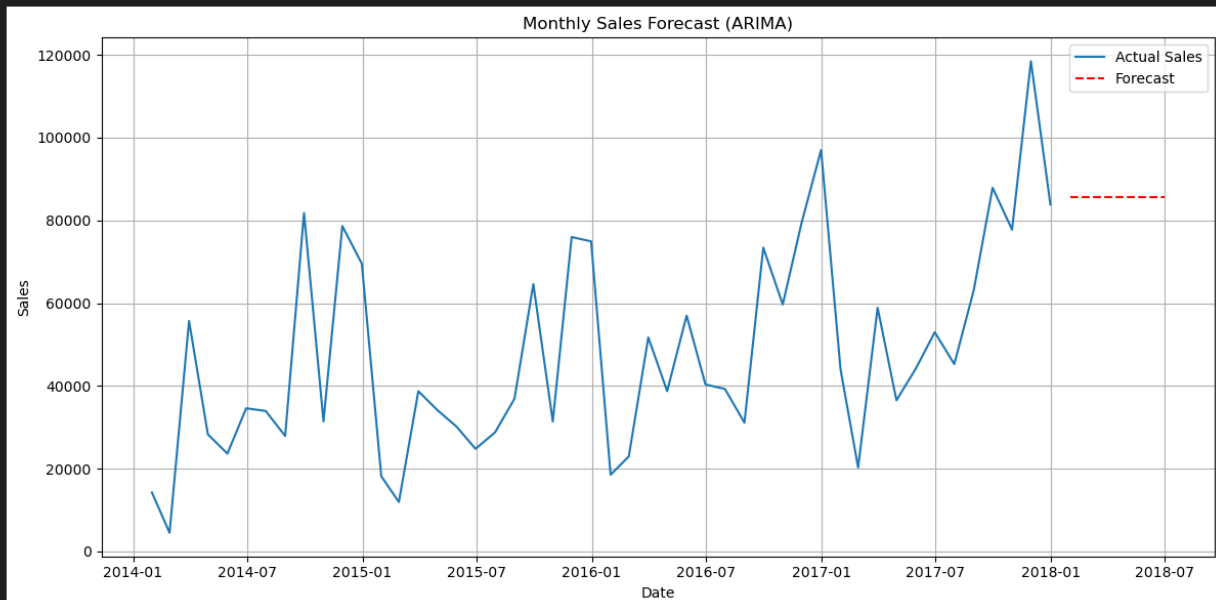

Monthly Sales Forecast (ARIMA)

Step 5: Dashboard Integration

The final model outputs and sales forecasts will be integrated into a dashboard that allows stakeholders to:

- View historical vs. forecasted sales.

- Interact with monthly breakdowns.

- Understand sales patterns easily.

Tools to Be Used (Suggestion):

- **Power BI** or **Tableau** for the dashboard.

- **Plotly Dash** for a Python-based web app.

Conclusion

The analysis and modeling phases provided clear insights into sales trends and enabled accurate forecasting. The chosen model (ARIMA 0,1,1) offers a balance of simplicity and accuracy, and the dashboard will further enhance interpretability and business impact.