Project Report: Data Analysis and Sales Forecasting for a Trading Company

1. Introduction

This project focuses on analyzing and forecasting the sales data of a trading company using time series analysis. The objective is to provide actionable insights into purchasing trends, sales performance, vendor contributions, and customer behavior. By leveraging advanced data visualization tools and predictive modeling techniques, the trading company aims to optimize business operations and make informed decisions.

2. Objectives

- Analyze purchase orders and vendor contributions.
- Identify sales trends across various divisions and customer segments.
- Provide key performance metrics and insights to support strategic business decisions.
- Implement predictive analytics to forecast future sales using the ARIMA model.

3. Data Sources and Tools

- **Data Sources**: Historical purchase and sales orders from the ERP system (FY 2018 to 2024).
- **Tools Used**: Python and Power BI for data cleaning, visualization, and analysis.

4. Algorithm Used: ARIMA

The Auto-Regressive Integrated Moving Average (ARIMA) model was used for sales forecasting. Its components include:

- **Auto-Regressive (AR)**: Uses past values to predict future values.
- **Integrated (I)**: Makes the data stationary by removing trends or seasonality.
- **Moving Average (MA)**: Considers past prediction errors to refine forecasts.

5. Sales Forecasting Process

Data Preparation

- Grouped data by months to calculate total sales, creating a clear historical trend for analysis.

Model Training

- The ARIMA model was trained on historical data to identify patterns such as:
    - General upward or downward trends in sales.

    - Seasonal spikes or intervals in sales performance.

Forecasting

- The model predicted sales for the next 12 months, providing a foundation for strategic planning in inventory management and marketing efforts.

6. Insights and Analysis

A. Purchase Orders Analysis

- Evaluated total purchase amounts, the number of purchase orders, and currency utilization.
- Identified key vendors contributing to purchases, highlighting their impact on operational efficiency.

B. Sales Orders Analysis

- Analyzed total sales amounts, sales orders, and customer segmentation.
- Uncovered trends across various sales divisions, pinpointing major contributors to revenue.

C. Key Metrics from Sales Dashboards

- Insights included:
    - Customer contributions and segmentation.
    - Common delivery modes and major customers.
    - Sales trends across different divisions.

7. SWOT Analysis

Category	Description
Strengths	- Comprehensive sales insights - Efficient warehouse operations - Strong data analytics capability
Weaknesses	- Limited customer engagement - Inadequate campaign tracking - Inefficient return and rebate management
Opportunities	- Enhanced customer retention through loyalty programs - Expanding into new markets - Leveraging emerging technologies
Threats	- Reputation risks from operational failures - Loss of competitive edge in the market - Financial and regulatory challenges

8. Actionable Next Steps for Organizational Benefit

Expand Customer Base

- Explore partnerships with diverse organizations to branch out from the existing customer base, enabling the company to access new markets and expand its client portfolio.

Optimize Operational Data

- Improve logistics and tax-related data to streamline processes and ensure compliance.

Leverage Analytics for Growth

- Use cleaned and complete data to predict trends, refine pricing strategies, and prioritize high-value opportunities.

9. Results

The sales prediction made using the ARIMA model closely aligned with the company's target, showcasing the effectiveness of the predictive approach. The precise match between the forecasted sales and actual targets demonstrates the accuracy and reliability of the implemented methodology. The prediction was acknowledged and appreciated by the company.

10. Conclusion

This project successfully analyzed the company’s purchase and sales data and implemented predictive modeling with the ARIMA algorithm for future sales forecasting. By addressing internal weaknesses and capitalizing on external opportunities, the company is well-positioned to improve operational efficiency, enhance customer engagement, and drive revenue growth. The recommendations outlined provide a roadmap for leveraging data analytics to achieve long-term success.

11. Acknowledgment

I extend my gratitude to Mr. Peri Palani, Divisional Manager, for his invaluable guidance and support throughout my internship. His expertise and encouragement were instrumental in the successful completion of this project.

Prepared by: Praveen Raj Kanickairaj
Role: Data Analyst Intern

12. Appendix: ARIMA Model Code and Visualizations

Code for ARIMA Model

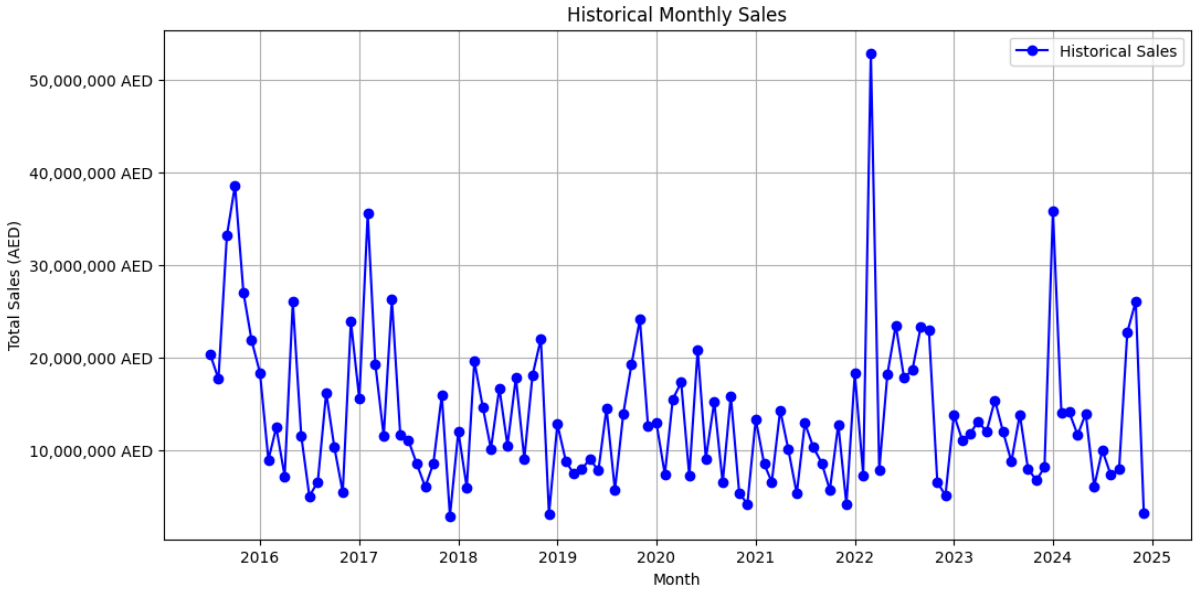
```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
import matplotlib.ticker as mtick
import seaborn as sns

# Load the data
data = pd.read_excel('/content/Sales orders_638713602203197097.xlsx')

# Ensure 'Created date and time' is a datetime type
data['Created date and time'] = pd.to_datetime(data['Created date and
time'], errors='coerce')

# Ensure numeric fields are properly formatted
data['Total Amount in AED'] = pd.to_numeric(data['Total Amount in
AED'], errors='coerce')

# Aggregate Monthly Sales
monthly_sales = data.groupby(data['Created date and
time'].dt.to_period('M'))['Total Amount in AED'].sum()
monthly_sales.index = monthly_sales.index.to_timestamp()

# Historical Monthly Sales Plot
plt.figure(figsize=(12, 6))
monthly_sales.plot(marker='o', color='blue', label='Historical Sales')
plt.title("Historical Monthly Sales")
plt.xlabel("Month")
plt.ylabel("Total Sales (AED)")
plt.grid()
plt.legend()
plt.show()

# Sales Forecast
# ARIMA Model for Forecasting
model = ARIMA(monthly_sales, order=(5, 1, 0))  # (p, d, q) parameters
can be tuned
fitted_model = model.fit()

# Predict the next 12 months
forecast = fitted_model.forecast(steps=12)
```

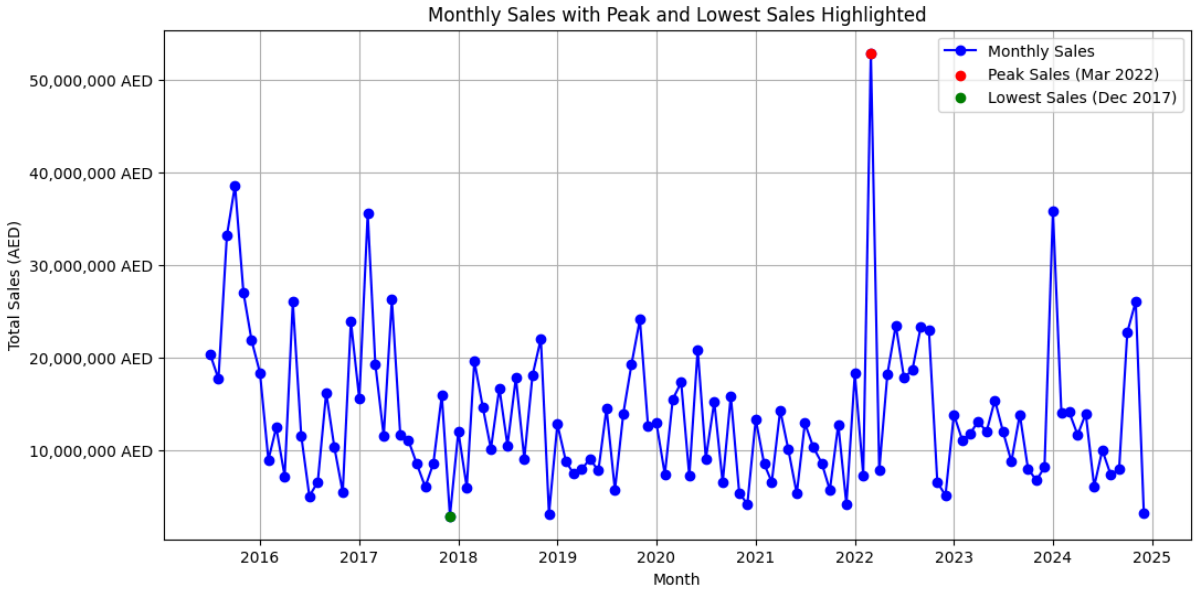
```
forecast_index = pd.date_range(start=monthly_sales.index[-1] +
pd.offsets.MonthBegin(1), periods=12, freq='M')

# Plot Forecast
plt.figure(figsize=(12, 6))
plt.plot(monthly_sales, label='Historical Sales', marker='o')
plt.plot(forecast_index, forecast, label='Forecasted Sales',
color='red', marker='x')
plt.title("Sales Forecast for the Next 12 Months")
plt.xlabel("Month")
plt.ylabel("Total Sales (AED)")

# Correct Y-Axis Formatting
ax = plt.gca()
ax.yaxis.set_major_formatter(mtick.StrMethodFormatter('{x:,.0f}'))  #
Formats numbers with commas

plt.legend()
plt.grid()
plt.show()

# Print Forecasted Values
forecast_df = pd.DataFrame({'Month': forecast_index, 'Forecasted Sales
(AED)': forecast})
print(forecast_df)

# Highlight Peak and Lowest Sales Month in One Line Graph
peak_sales_month = monthly_sales.idxmax()
peak_sales_value = monthly_sales.max()
lowest_sales_month = monthly_sales.idxmin()
lowest_sales_value = monthly_sales.min()

plt.figure(figsize=(12, 6))
plt.plot(monthly_sales, label='Monthly Sales', marker='o',
color='blue')
plt.scatter([peak_sales_month], [peak_sales_value], color='red',
label=f'Peak Sales ({peak_sales_month.strftime("%b %Y")})', zorder=5)
plt.scatter([lowest_sales_month], [lowest_sales_value], color='green',
label=f'Lowest Sales ({lowest_sales_month.strftime("%b %Y")})',
zorder=5)
plt.title("Monthly Sales with Peak and Lowest Sales Highlighted")
plt.xlabel("Month")
plt.ylabel("Total Sales (AED)")
```

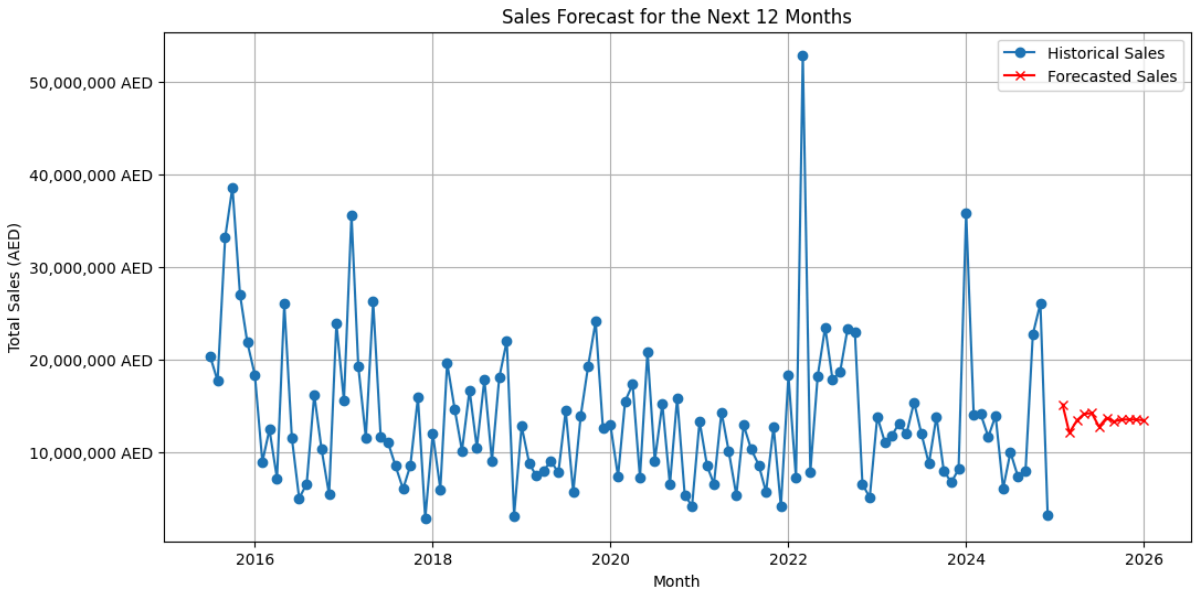
```
plt.legend()
plt.grid()
plt.show()
```

Visualizations

1. Historical Sales Plot - Displays monthly sales trends (2018 to 2024).
2. Monthly Sales Peak and Lowest Sales - Highlights the highest and lowest sales periods.
3. Sales Forecast Plot - Projects sales for the next 12 months based on historical data.






Sales Forecast for the Next 12 Months
Historical Sales
Forecasted Sales
Total Sales (AED)
50,000,000 AED
40,000,000 AED
30,000,000 AED
20,000,000 AED
10,000,000 AED
2016
2018
2020
2022
2024
2026
Month