Sales Forecasting and Trend Analysis

A data-driven approach with a retail store dataset

Background

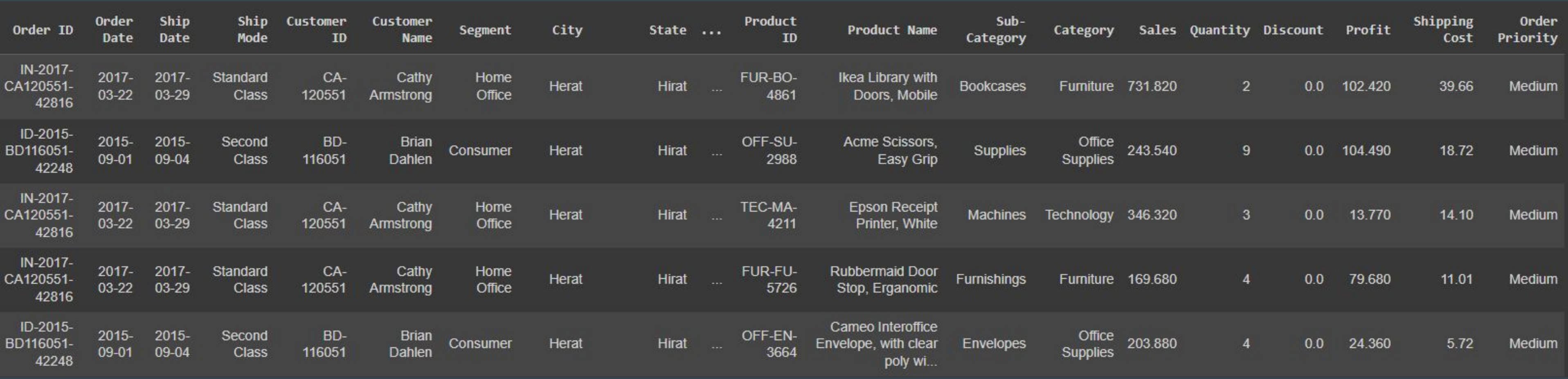
- Today's business landscape is more competitive than ever - having all the information and insight you can is critical

- Companies must adapt quickly to shifts in market trends, consumer behavior, and economic conditions

- Sales forecasting and data analysis enable businesses to quickly adjust strategies in response to market changes

Dataset

- 51,290 rows, 23 columns containing orders
- Captures data on sales - order date, customer demographics, product info, revenue earned, quantity purchased, discounts given, shipping costs, and order priority.



Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	City	State	...	Product ID	Product Name	Sub-Category	Category	Sales	Quantity	Discount	Profit	Shipping Cost	Order Priority
IN-2017-CA120551-42816	2017-03-22	2017-03-29	Standard Class	CA-120551	Cathy Armstrong	Home Office	Herat	Hirat	...	FUR-BO-4861	Ikea Library with Doors, Mobile	Bookcases	Furniture	731.820	2	0.0	102.420	39.66	Medium
ID-2015-BD116051-42248	2015-09-01	2015-09-04	Second Class	BD-116051	Brian Dahlen	Consumer	Herat	Hirat	...	OFF-SU-2988	Acme Scissors, Easy Grip	Supplies	Office Supplies	243.540	9	0.0	104.490	18.72	Medium
IN-2017-CA120551-42816	2017-03-22	2017-03-29	Standard Class	CA-120551	Cathy Armstrong	Home Office	Herat	Hirat	...	TEC-MA-4211	Epson Receipt Printer, White	Machines	Technology	346.320	3	0.0	13.770	14.10	Medium
IN-2017-CA120551-42816	2017-03-22	2017-03-29	Standard Class	CA-120551	Cathy Armstrong	Home Office	Herat	Hirat	...	FUR-FU-5726	Rubbermaid Door Stop, Erganomic	Furnishings	Furniture	169.680	4	0.0	79.680	11.01	Medium
ID-2015-BD116051-42248	2015-09-01	2015-09-04	Second Class	BD-116051	Brian Dahlen	Consumer	Herat	Hirat	...	OFF-EN-3664	Cameo Interoffice Envelope, with clear poly wi...	Envelopes	Office Supplies	203.880	4	0.0	24.360	5.72	Medium

Data Preparation

- Load the Excel file into a Pandas dataframe

- Inspect the data - basic structure and types

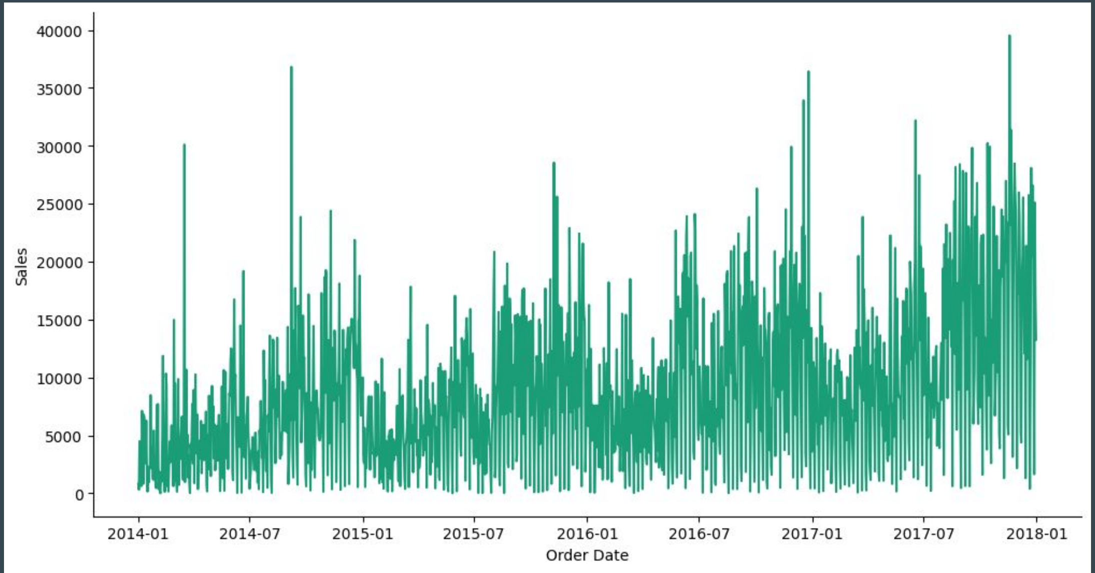
- Feature engineering
  - Resampling based on the start of the month
  - Aggregate sales per day and month

- Use of Pandas, NumPy, MatPlotLib, StatsModels API, Seaborn

Data Analysis

Variety of analysis performed on the dataset including:

- Seasonality
- Promotions strategy
- Sales growth over time
- Most profitable categories and markets



Seasonal Autoregressive Integrated Moving Average (SARIMA)

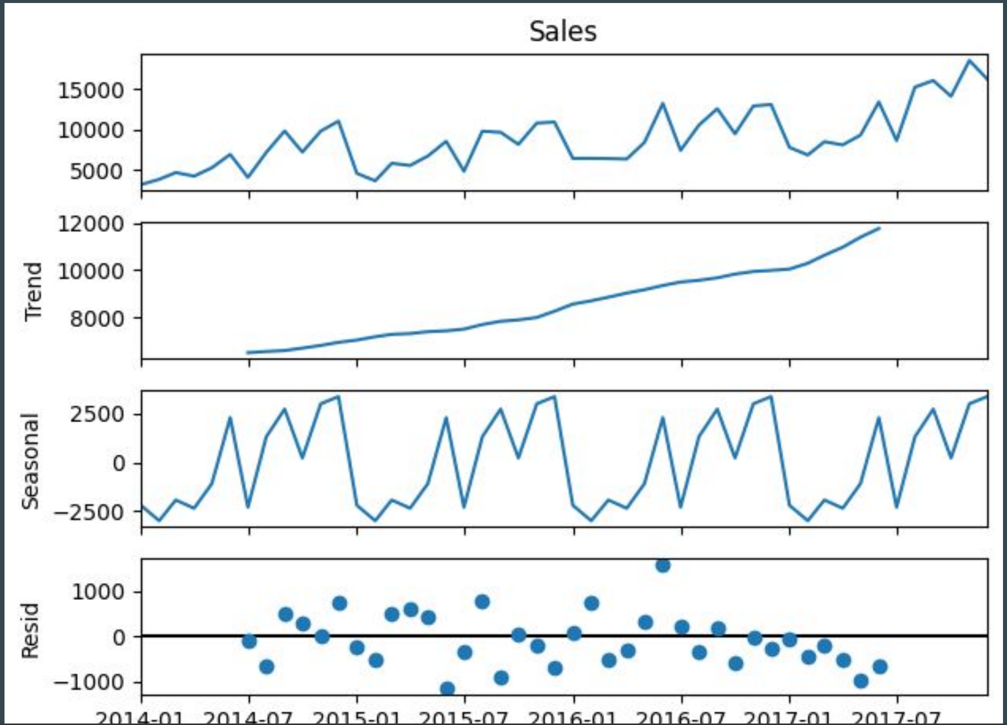
- Extension of the ARIMA model that can be used for forecasting time series data with seasonal patterns
  - AR (Autoregressive): Uses past values.
  - I (Integrated): Makes the series stationary by differencing.
  - MA (Moving Average): Uses past forecast errors.

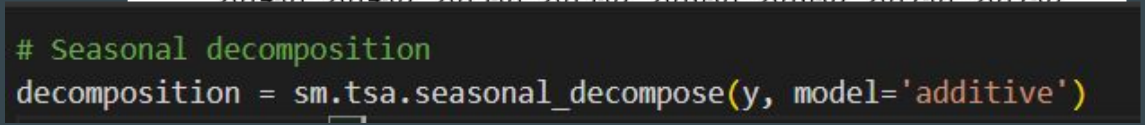
- **Notation:** SARIMA(p, d, q)(P, D, Q)m
  - p, d, q: Non-seasonal parameters.
  - P, D, Q: Seasonal parameters.
  - m: Number of periods in a season (ex. 12 for monthly data).

SARIMA - Seasonal Decomposition

- Python library ‘statsmodels’
  - Assumes the time series is composed of a linear combination of trend + seasonal + residual components

- **Trend** - long-term direction
- **Seasonal** - repeating short-term cycles in the data
- **Residual** - randomness, noise, or outliers in the data

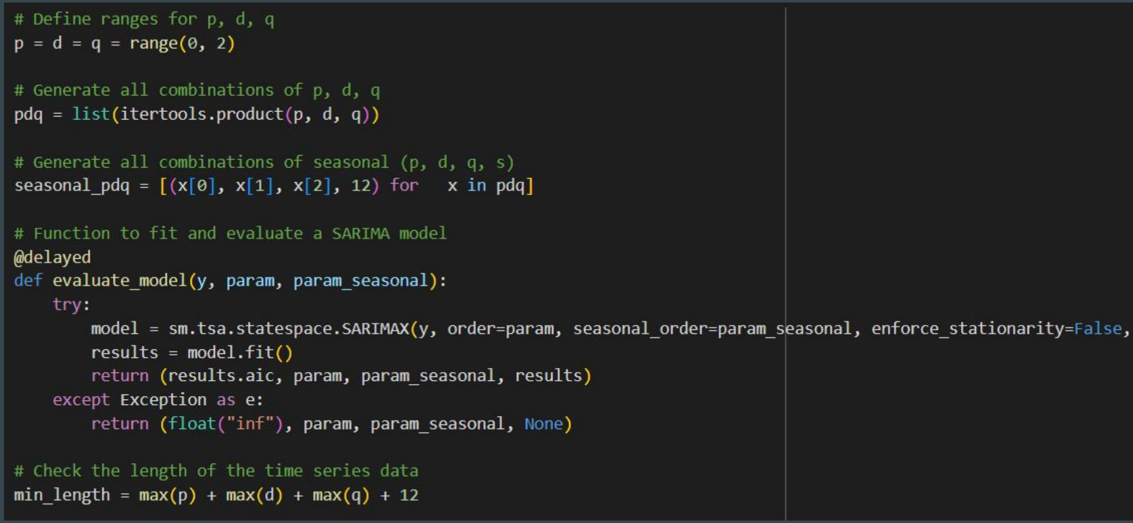




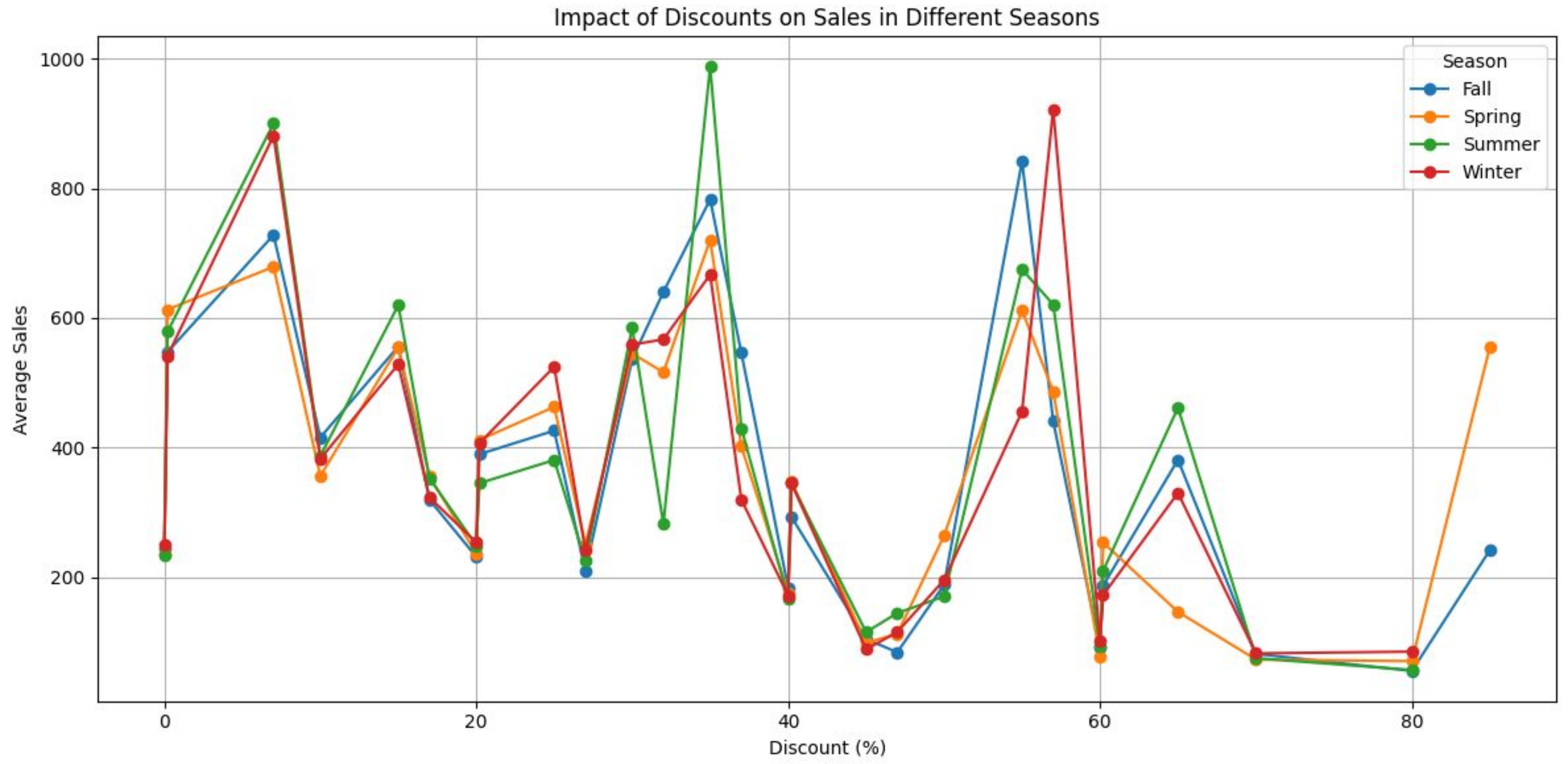
```
# Seasonal decomposition
decomposition = sm.tsa.seasonal_decompose(y, model='additive')
```

SARIMA - Parameter Selection

- Generate all possible combinations of p, d, q - and their seasonal combinations
- Fit and evaluate the SARIMA model using AIC (Akaike Information Criterion)
  - metric used to evaluate the quality of a statistical model, balancing model fit and complexity
- Find the best parameters with the lowest AIC

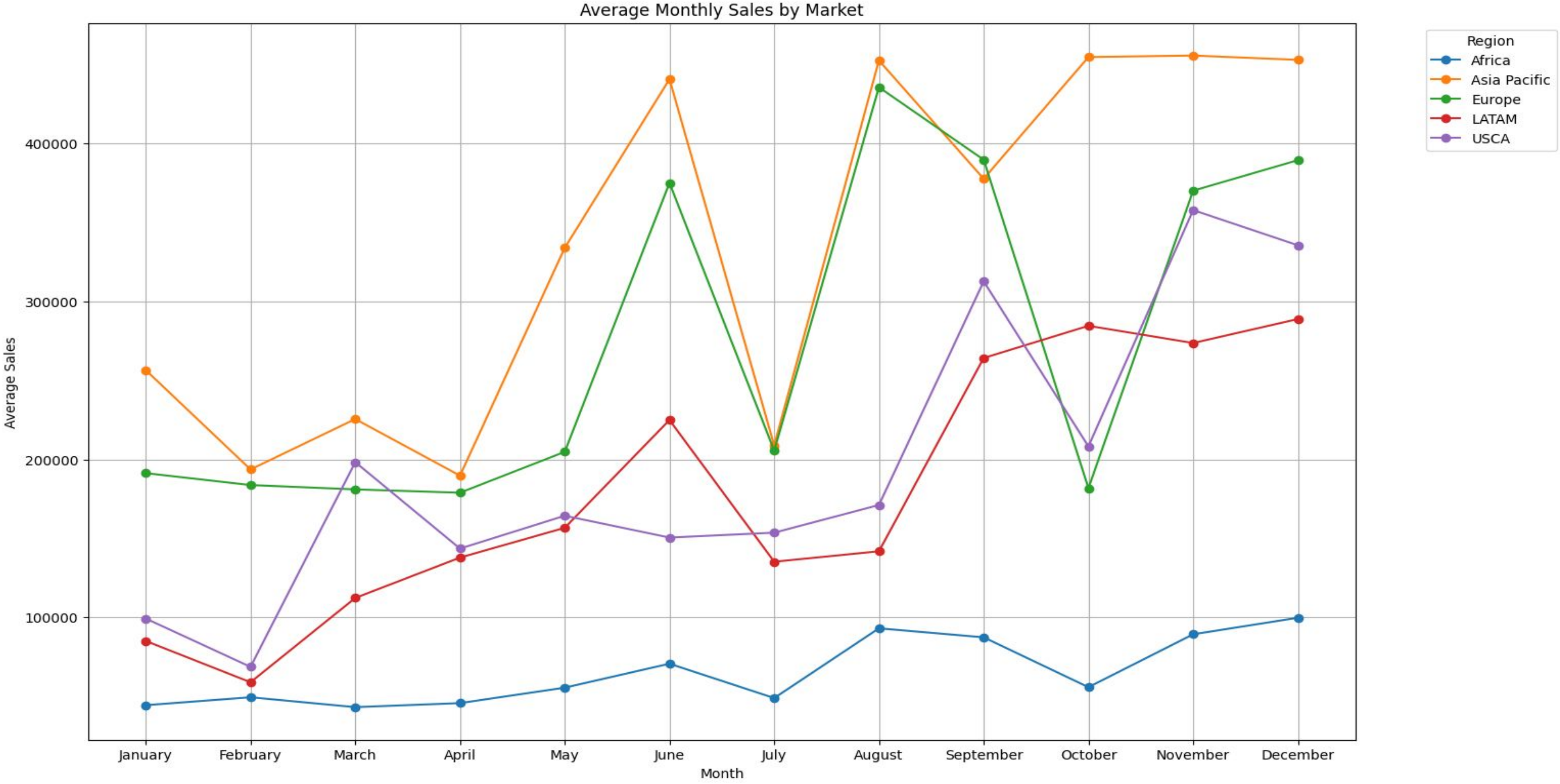


```
# Define ranges for p, d, q
p = d = q = range(0, 2)

# Generate all combinations of p, d, q
pdq = list(itertools.product(p, d, q))

# Generate all combinations of seasonal (p, d, q, s)
seasonal_pdq = [(x[0], x[1], x[2], 12) for   x in pdq]

# Function to fit and evaluate a SARIMA model
@delayed
def evaluate_model(y, param, param_seasonal):
    try:
        model = sm.tsa.statespace.SARIMAX(y, order=param, seasonal_order=param_seasonal, enforce_stationarity=False,
        results = model.fit()
        return (results.aic, param, param_seasonal, results)
    except Exception as e:
        return (float("inf"), param, param_seasonal, None)

# Check the length of the time series data
min_length = max(p) + max(d) + max(q) + 12
```

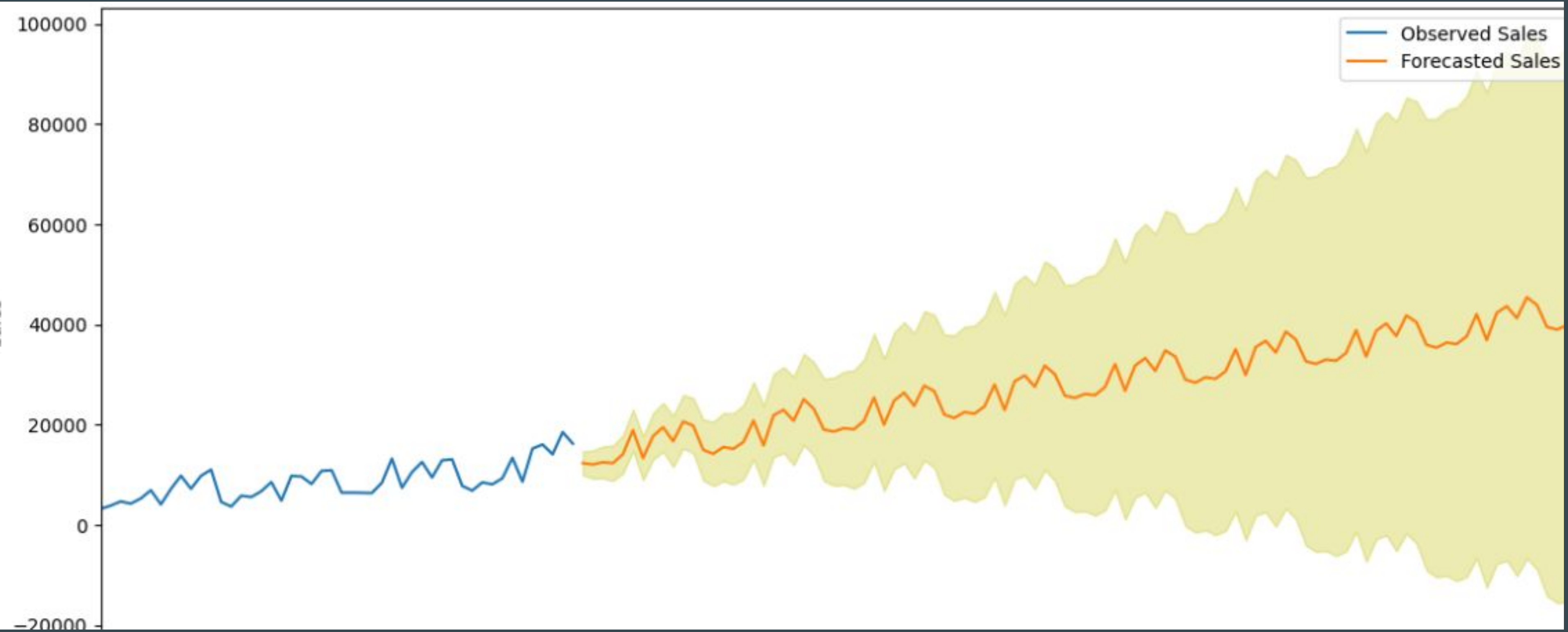


Optimal discounting for sales increase vs. maintaining profitability likely around 30%

Seasonal pattern variance across markets



Upward and continuously seasonal trend over time



Challenges include an increasingly wide confidence interval as time goes on

Conclusions

- SARIMA is a good choice for time-series forecasting, in the short term

- Limitation of the past predicting the future

- In this example dataset, seasonality and promotional strategies are crucial to consider

- Lots of opportunity to build from the basics and add more to evaluate more specific examples based on the needs of a business

Challenges

- Learning and working with Python and machine learning models as a beginner
- Limited dataset size, especially for granular analysis
- Could possibly integrate different machine learning models for more advanced analysis, or complete more feature engineering to drill down on different areas

Future Research/Expansions

- External factors such as economic indicators, market trends, weather patterns, or social media sentiments could be integrated to enhance analysis.

- A closer analysis of customer segments could be performed to identify high-value customers and tailor marketing and promotion strategies to them.

- Dynamic pricing models could be developed to adjust prices in real-time based on demand forecasts, competitor’s pricing, or inventory levels.

- Natural language processing (NLP) could be used to extract insights from customer reviews or feedback for analysis.

Questions?