Ministry of Communications and Information Technology

Digital Egypt Pioneers

Project Title:

Sales Forecasting and Demand Prediction

Instructor

Eslam Elreedy

Team Members

- Nourelden Hany Abdel-Moaty
- Goda Saber Goda
- Abdulrahman Ahmed Mahmoud
- Ahmed Tamer Mohammed
- Hussein Elsayed Hussein

Presentation Outline

Topics to Discuss

1. Project Overview
2. Simple Goal-Flow Diagram
3. Dataset Summary
4. Feature Overview
5. Data Exploration
6. Key Visualizations from the Dataset
7. Interactive Visualization – Power BI
8. Data Preparation
9. Feature Engineering
10. Feature Evaluation & Selection
11. Model Building
12. Model Performance Comparison
13. Pipeline and Hyperparameter Tuning
14. MLflow Integration
15. Model Deployment
16. Business Impact

Project Overview

Objective

This project aims to build a data-driven solution that can accurately forecast daily product demand and sales in retail stores. By using machine learning and deep learning models, we enable retail managers to make informed decisions about inventory control, promotions, and pricing strategies.

Key Goals

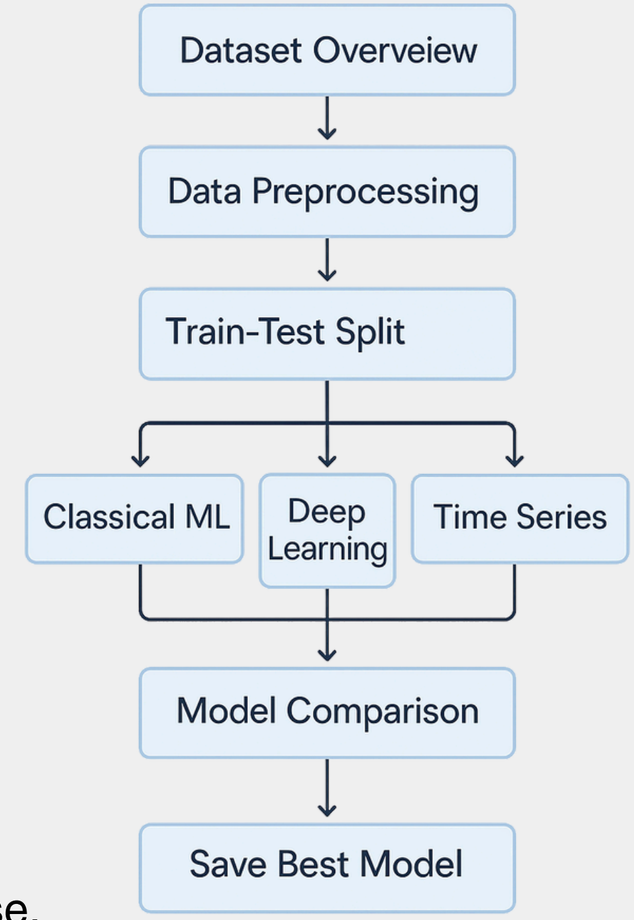
- Minimize overstock and understock situations.

- Improve profit margins through optimized pricing.

- Anticipate seasonal trends and external factor impacts.

Simple Goal-Flow Diagram

1. **Dataset Overview:** Understand the data, structure, and target variable.
2. **Data Preprocessing:** Clean, encode, normalize, and prepare features.
3. **Feature Engineering:** Select important features and reduce dimensionality.
4. **Train-Test Split:** Divide data for training and evaluation.
5. **Modeling Paths:**
   - **Classical ML:** *XGBoost, Random Forest, Gradient Boosting, SVM, Ridge*
   - **Deep Learning:** *MLP, CNN, LSTM, BiLSTM, LSTM+CNN, BiLSTM+CNN*
   - **Time Series:** *ARIMA, SARIMAX, Prophet*
6. **Model Comparison:** Evaluate models with error metrics and select the best.
7. **Save Best Model:** Export the top-performing model for deployment or future use.



Dataset Summary

Dataset Description

- The dataset includes daily transaction-level records from various retail stores. It captures key aspects like product information, inventory metrics, and external influencers.

Key features

Inventory, Sales, Demand, Pricing, Promotions, Weather

Example: Jan 1, 2022
Store S001 sold 127 units of Product P0001 (Groceries)
forecasted demand: 135.47

Dataset Size

- Rows: 73,100 records
- Columns: 15

```
df.dtypes
✓ 0.0s

Date                  object
Store ID              object
Product ID            object
Category              object
Region                object
Inventory Level        int64
Units Sold             int64
Units Ordered          int64
Demand Forecast      float64
Price                float64
Discount               int64
Weather Condition     object
Holiday/Promotion      int64
Competitor Pricing   float64
Seasonality           object
dtype: object
```

List of Dataset Features and Types

Feature Overview

Understanding data distribution, value ranges, and categorical frequency.

Descriptive Statistics Summary (All Columns)

	Date	Store ID	Product ID	Category	Region	Inventory Level	Units Sold	Units Ordered	Demand Forecast	Price	Discount	Weather Condition	Holiday/Promotion	Competitor Pricing	Seasonality
count	73100	73100	73100	73100	73100	73100.000000	73100.000000	73100.000000	73100.000000	73100.000000	73100.000000	73100	73100.000000	73100.000000	73100
unique	731	5	20	5	4	NaN	NaN	NaN	NaN	NaN	NaN	4	NaN	NaN	4
top	2022-01-01	S001	P0001	Furniture	East	NaN	NaN	NaN	NaN	NaN	NaN	Sunny	NaN	NaN	Spring
freq	100	14620	3655	14699	18349	NaN	NaN	NaN	NaN	NaN	NaN	18290	NaN	NaN	18317
mean	NaN	NaN	NaN	NaN	NaN	274.469877	136.464870	110.004473	141.494720	55.135108	10.009508	NaN	0.497305	55.146077	NaN
std	NaN	NaN	NaN	NaN	NaN	129.949514	108.919406	52.277448	109.254076	26.021945	7.083746	NaN	0.499996	26.191408	NaN
min	NaN	NaN	NaN	NaN	NaN	50.000000	0.000000	20.000000	-9.990000	10.000000	0.000000	NaN	0.000000	5.030000	NaN
25%	NaN	NaN	NaN	NaN	NaN	162.000000	49.000000	65.000000	53.670000	32.650000	5.000000	NaN	0.000000	32.680000	NaN
50%	NaN	NaN	NaN	NaN	NaN	273.000000	107.000000	110.000000	113.015000	55.050000	10.000000	NaN	0.000000	55.010000	NaN
75%	NaN	NaN	NaN	NaN	NaN	387.000000	203.000000	155.000000	208.052500	77.860000	15.000000	NaN	1.000000	77.820000	NaN
max	NaN	NaN	NaN	NaN	NaN	500.000000	499.000000	200.000000	518.550000	100.000000	20.000000	NaN	1.000000	104.940000	NaN

Data Exploration

Initial Analysis

- All dates are valid and within expected range
- No negative values in key numerical fields (e.g., *Price*, *Units Sold*)
- Detected anomalies where Units Ordered exceeded Inventory Level

Key Observations

- Demand varies significantly by ***season*** and ***region***
- ***Discounts*** and ***competitor pricing*** have a strong influence on unit sales
- Certain categories (e.g., *Groceries*, *Toys*) consistently show higher demand

Key Visual Insights from the Dataset: See next slides →

Key Visualizations from the Dataset

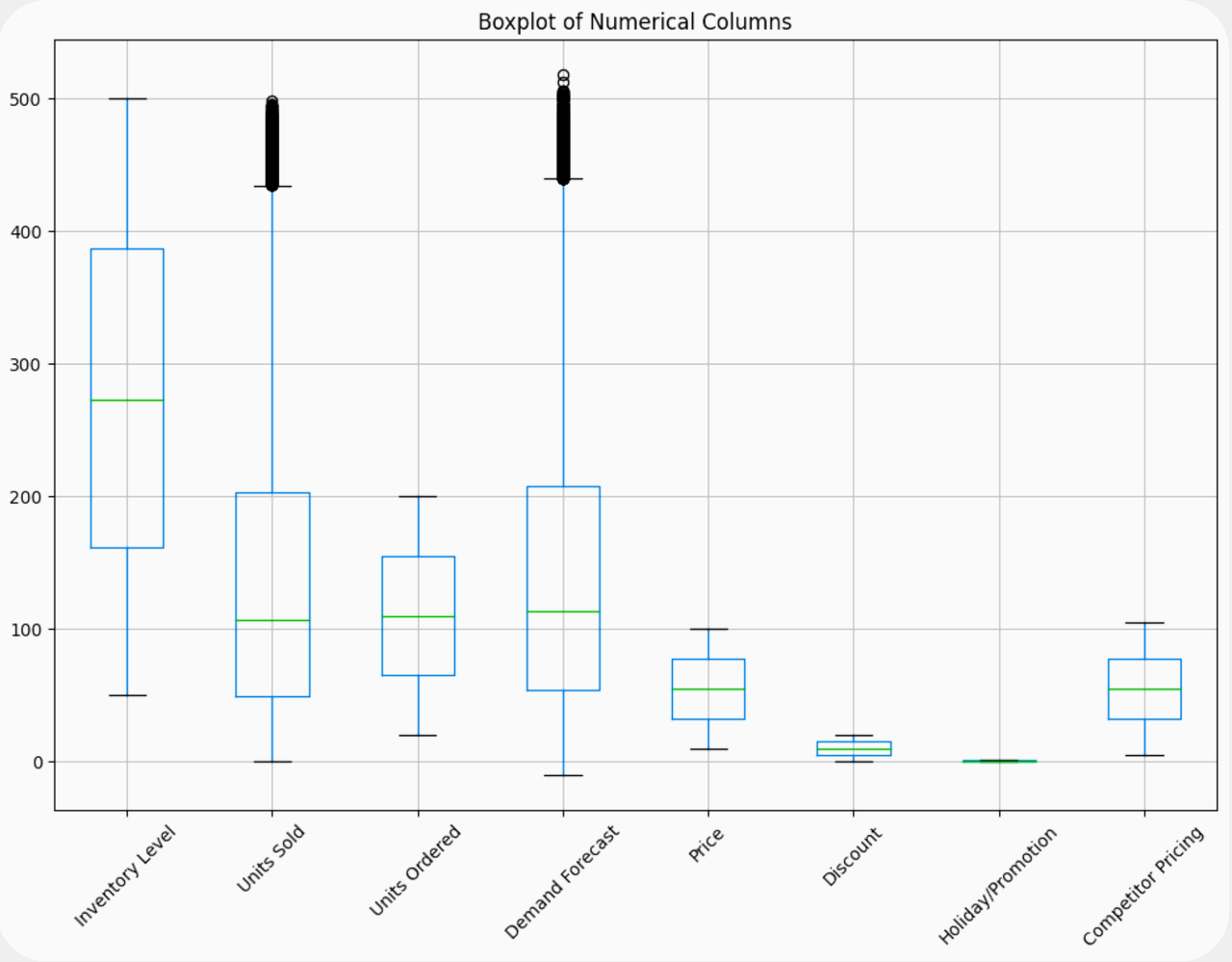
Distribution & Outliers in Numerical Features

Key Insights from the Dataset

Several numerical variables exhibit high variance and extreme outliers, particularly in demand-related features.

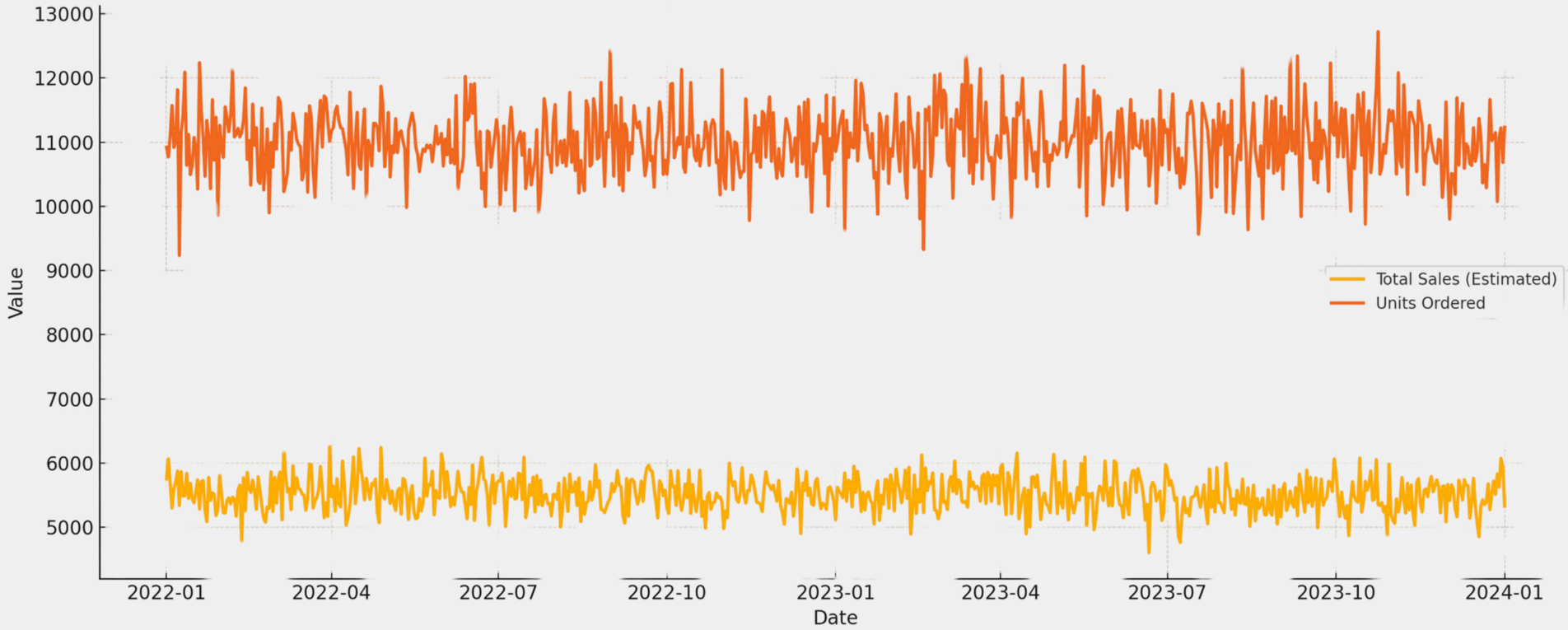
Observations

- ***Units Sold***, ***Inventory Level***, and ***Demand Forecast*** have wide spreads and many outliers.
- ***Price***, ***Discount***, and ***Competitor Pricing*** are more stable with lower dispersion.
- ***Holiday/Promotion*** is binary, showing minimal variation.



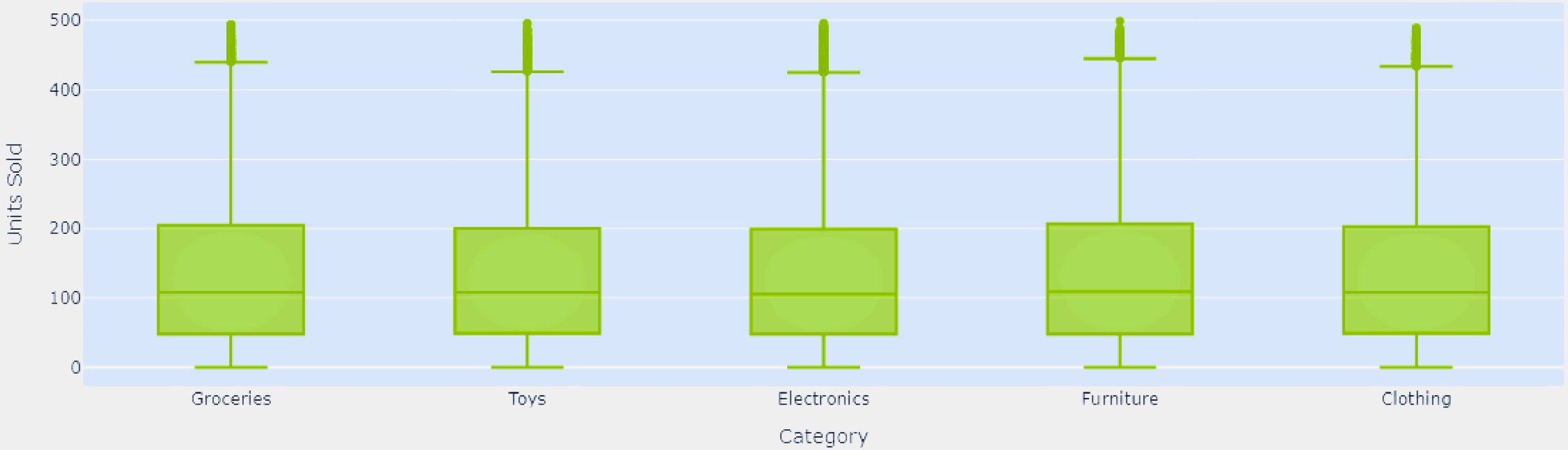
Sales vs Units Ordered Over Time

- Compares the daily number of units sold versus units ordered from Jan 2022 to Jan 2024.
- This visual helps assess supply-demand alignment and detect overordering or stockout risks over time.



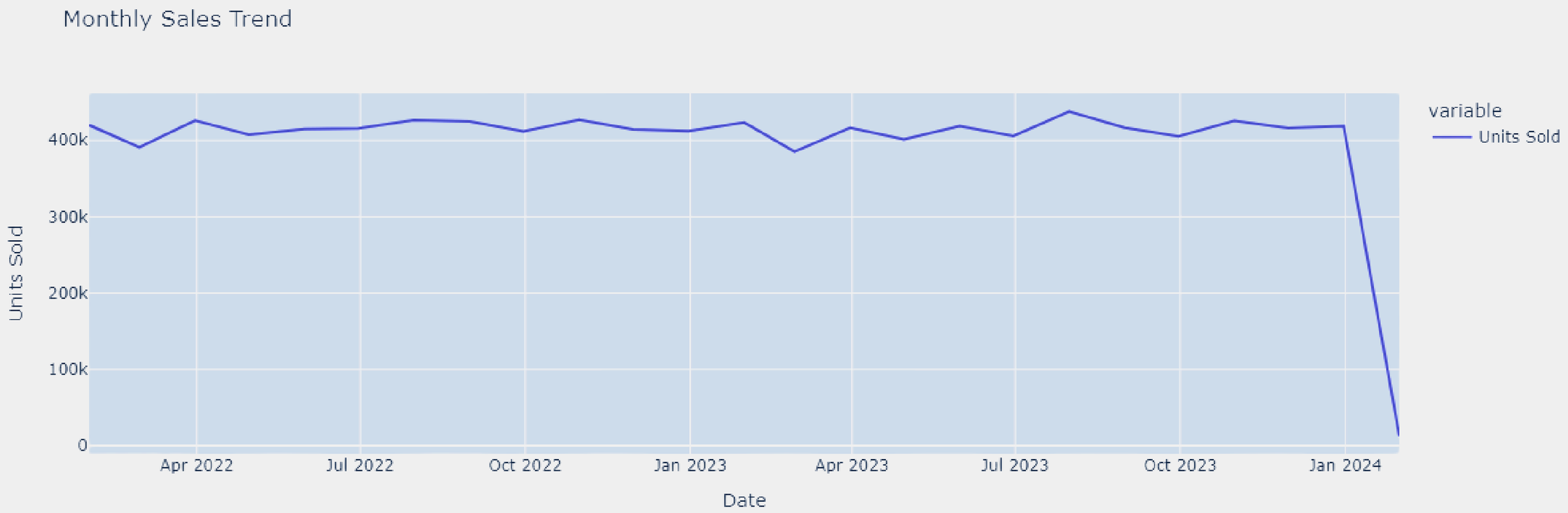
Units Sold by Category

- Displays the spread, median, and outliers of units sold for each product category.



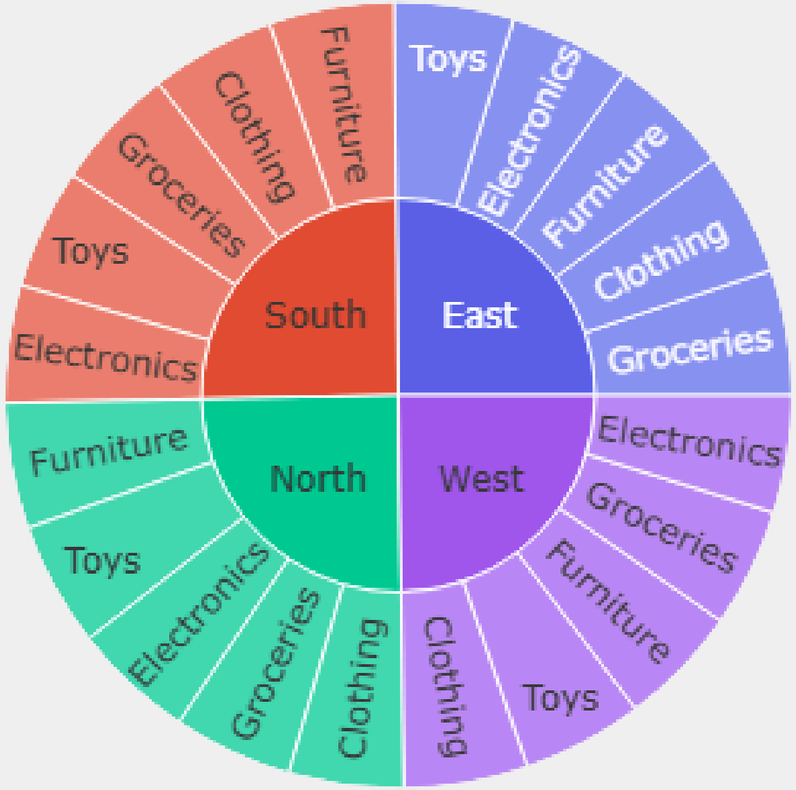
Monthly Sales Trend

- Tracks monthly sales volume over two years.
- Helps identify overall trend direction and monthly fluctuations in demand.



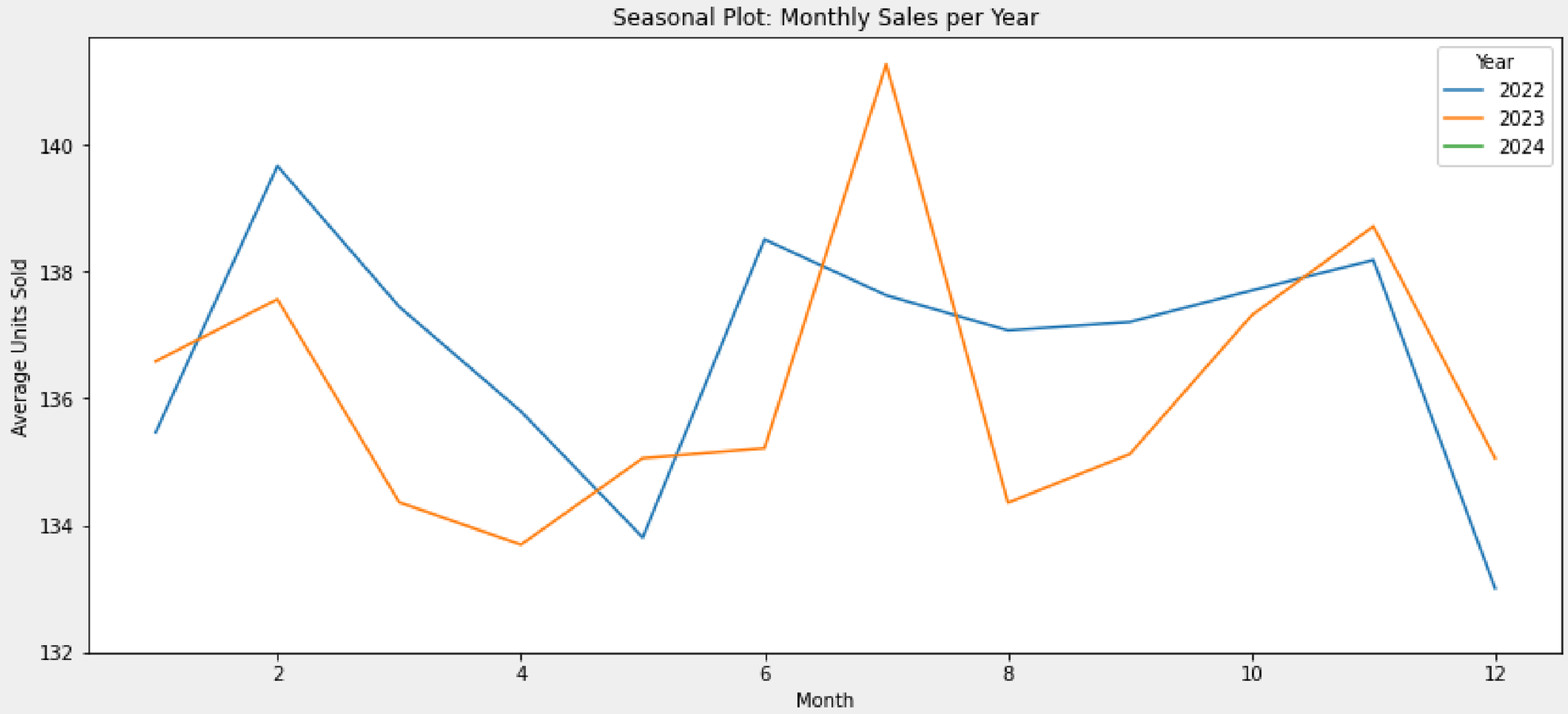
Sales by Region and Category

- A hierarchical view of sales distribution across regions and their internal product categories.
- Useful for exploring region-specific category performance in a single compact chart.



Sales by Region and Category

Sales follow a seasonal pattern, with noticeable peaks in summer and dips in December. Trend is consistent across years and useful for planning.



Interactive Visualization – Power BI

Power BI

Data Preparation

Date Cleaning & Indexing

Data Preparation

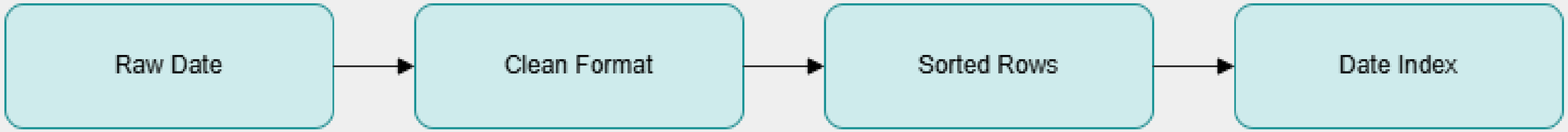
Categorical Encoding

Date Cleaning & Indexing

Prepared the dataset for time-series analysis by converting and indexing the date column, ensuring accurate ordering and enabling temporal feature extraction.

```
df["Date"] = pd.to_datetime(df["Date"])
df = df.sort_values(by="Date")
df.set_index("Date", inplace=True)
```

✓ 0.0s



Data Preparation

Outlier Detection & Feature Scaling

- Identified and quantified outliers using IQR method
- Removed extreme outliers from Units Sold and Demand Forecast
- Applied MinMaxScaler to normalize numerical features between 0 and 1

Normalized Sample Table

Date	Store ID	Product ID	Category	Region	Inventory Level	Units Sold	Units Ordered	Demand Forecast	Price	Discount	Weather Condition	Holiday/Promotion	Competitor Pricing	Seasonality
2022-01-01	S001	P0001	Groceries	North	0.402222	0.292627	0.194444	0.323539	0.261111	1.00	Rainy	0.0	0.246822	Autumn
2022-01-01	S004	P0013	Furniture	East	0.313333	0.129032	0.250000	0.143375	0.575667	0.00	Sunny	0.0	0.589430	Autumn
2022-01-01	S004	P0012	Electronics	North	0.664444	0.020737	0.805556	0.024333	0.047222	0.25	Rainy	1.0	0.135422	Spring
2022-01-01	S004	P0011	Electronics	West	0.344444	0.105991	0.038889	0.125981	0.498222	0.00	Sunny	1.0	0.527775	Spring
2022-01-01	S004	P0010	Groceries	East	0.882222	0.239631	0.422222	0.278076	0.260889	0.75	Cloudy	0.0	0.321489	Summer

Categorical Encoding

Label & One-Hot Encoding

- Used **LabelEncoder** for Store ID and Product ID
- Applied **OneHotEncoder** for categorical features (Region, Seasonality, etc.)
- Ensured encoded data is model-ready

Encoded Numeric & Label Features

Date	Store ID	Product ID	Inventory Level	Units Sold	Units Ordered	Demand Forecast	Price	Discount	Holiday/Promotion	Competitor Pricing	...
2022-01-01	0	0	0.402222	0.292627	0.194444	0.323539	0.261111	1.00	0.0	0.246822	...
2022-01-01	3	12	0.313333	0.129032	0.250000	0.143375	0.575667	0.00	0.0	0.589430	...
2022-01-01	3	11	0.664444	0.020737	0.805556	0.024333	0.047222	0.25	1.0	0.135422	...
2022-01-01	3	10	0.344444	0.105991	0.038889	0.125981	0.498222	0.00	1.0	0.527775	...
2022-01-01	3	9	0.882222	0.239631	0.422222	0.278076	0.260889	0.75	0.0	0.321489	...

One-Hot Encoded Categorical Features

Category_Toys	Region_North	Region_South	Region_West	Weather Condition_Rainy	Weather Condition_Snowy	Weather Condition_Sunny	Seasonality_Spring	Seasonality_Summer	Seasonality_Winter
False	True	False	False	True	False	False	False	False	False
False	False	False	False	False	False	True	False	False	False
False	True	False	False	True	False	False	True	False	False
False	False	False	True	False	False	True	True	False	False
False	False	False	False	False	False	False	False	True	False

“LabelEncoder is ideal for ordinal or ID-type data, while OneHotEncoder is used for nominal categorical features.”

Feature Engineering

Feature Creation

Feature Interaction

Feature Creation

- Converted boolean columns into integer format to ensure compatibility across ML models
- Engineered new value-added features from existing columns:
  - ***Price_Discount*** = Price × Discount
  - ***Inventory_Demand*** = Inventory Level × Demand Forecast
  - ***UnitsSold_Price*** = Units Sold × Price

Feature Interaction

- Created interaction feature: Weather_Seasonality
  - Captures relationships between weather conditions and seasonal trends
- **Weather_Seasonality**= (Weather Condition_Rainy x Seasonality_Spring +
  Weather Condition_Sunny x Seasonality_Summer +
  Weather Condition_Snowy x Seasonality_Winter)
- Helps reflect environmental effects on customer demand behavior

These features enhance model logic and adaptiveness to seasonal patterns.

Feature Evaluation & Selection

ADF

Correlation

ANOVA

RFE

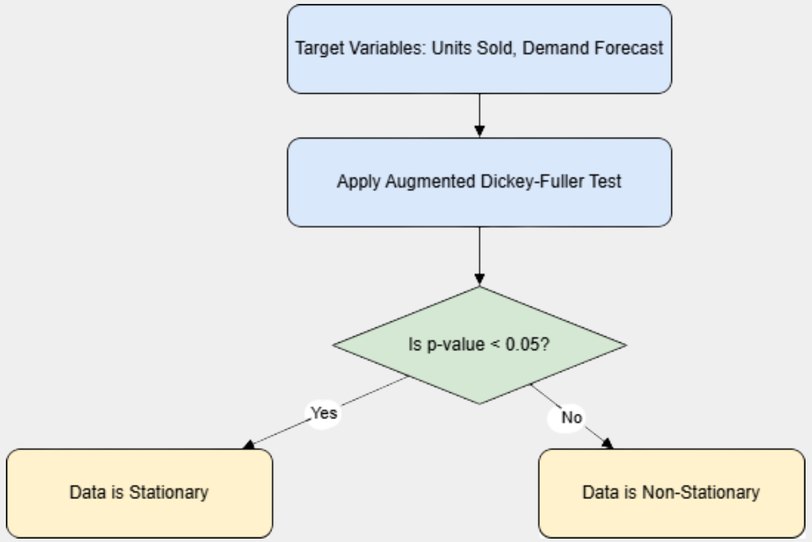
Augmented Dickey-Fuller Test

Test whether the time series target variables are stationary — a critical assumption for many forecasting models.

Variables Tested:

- ***Units Sold***
- ***Demand Forecast***

Flowchart



Result

```
 Augmented Dickey-Fuller Test for: Units Sold
ADF Statistic: -269.04806632897885
p-value: 0.0
Critical Values:
   1%: -3.430440458719505
   5%: -2.861579981167954
   10%: -2.5667912806060857
 Units Sold data is stationary (Null hypothesis rejected, no unit root).

 Augmented Dickey-Fuller Test for: Demand Forecast
ADF Statistic: -268.99713029689605
p-value: 0.0
Critical Values:
   1%: -3.430440458719505
   5%: -2.861579981167954
   10%: -2.5667912806060857
 Demand Forecast data is stationary (Null hypothesis rejected, no unit root).
```

Correlation Matrix

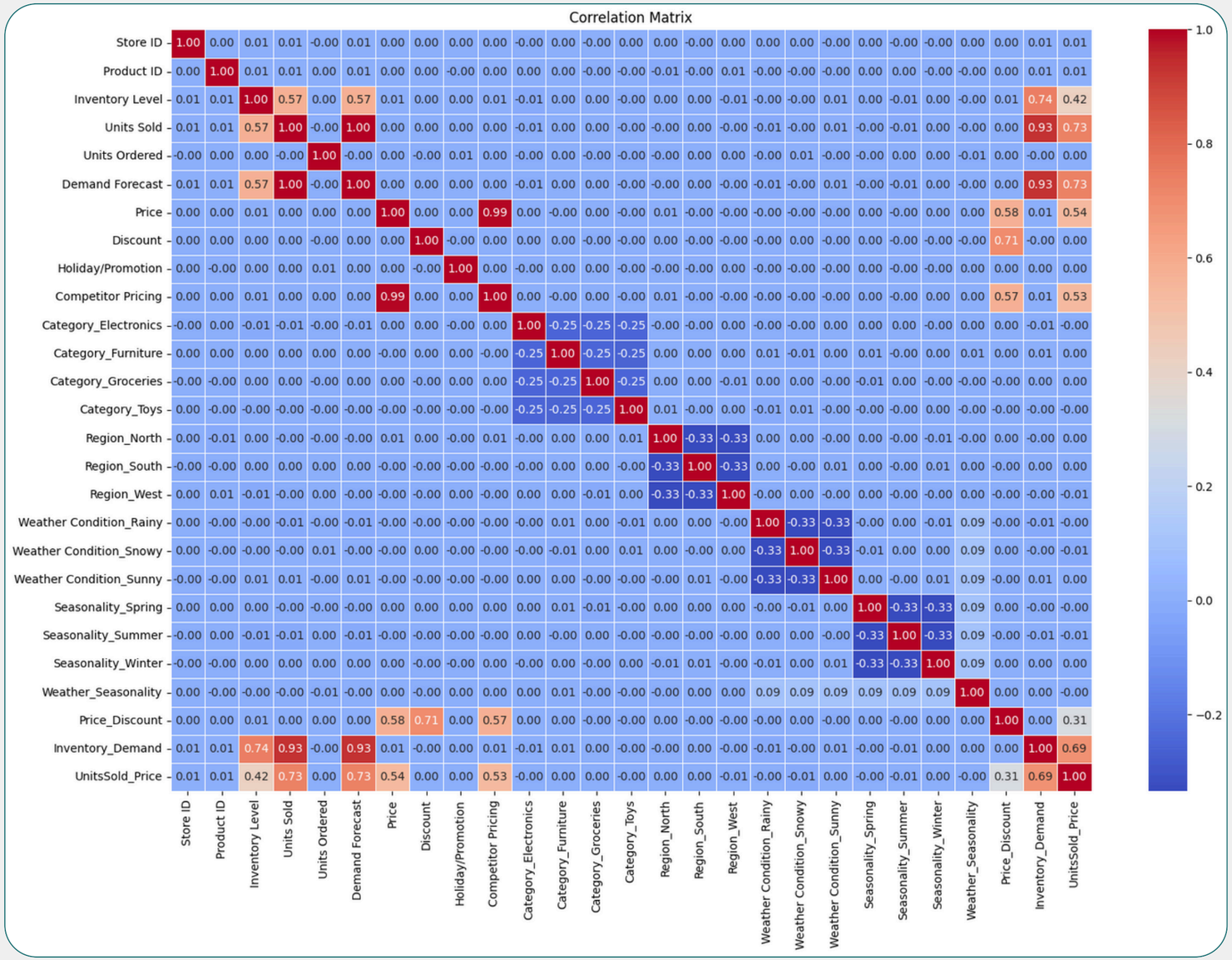
Understand the strength and direction of relationships between numerical features and identify redundant variables.

Key Insights

- Strong positive correlation between:
  - ***Units Ordered*** and ***Units Sold***
  - ***Inventory Level*** and ***Demand Forecast***
  - ***UnitsSold_Price*** and ***Demand Forecast***
- ***Price*** and ***Competitor Pricing***

were nearly identical (high multicollinearity)

- **Correlated features were removed.**
- **This enhanced model clarity.**



ANOVA (F-Test)

Used to statistically evaluate which features significantly impact the target variables.

Results for Units Sold

- Demand Forecast —-- F = 24,119
- Inventory_Demand — F = 1,500
- UnitsSold_Price —---- F = 197
- Inventory Level — ---- F = 84

→ These features show extremely high predictive value with p-values = 0.0000

Results for Demand Forecast

- Units Sold ———— F = 350
- Inventory_Demand — F = 26
- UnitsSold_Price —— F = 6.4

Inventory Level, Price_Discount — Also statistically significant ($p < 0.001$)

✵ **Features with high F-statistics and low p-values are strong predictors and should be retained for model training.**

Recursive Feature Elimination (RFE)

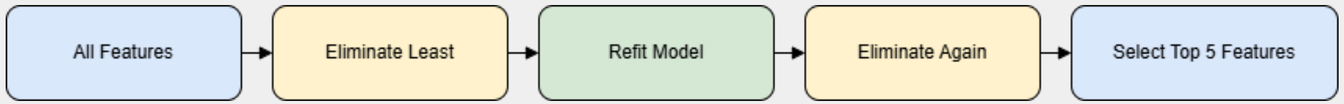
An iterative model-based approach that selects the most important predictors.

Process

- **Estimator**: Linear Regression
- **Strategy**: Recursively remove the least important features
- **Goal**: Select top 5 predictive features for each target

Top Features Selected for Demand Forecast

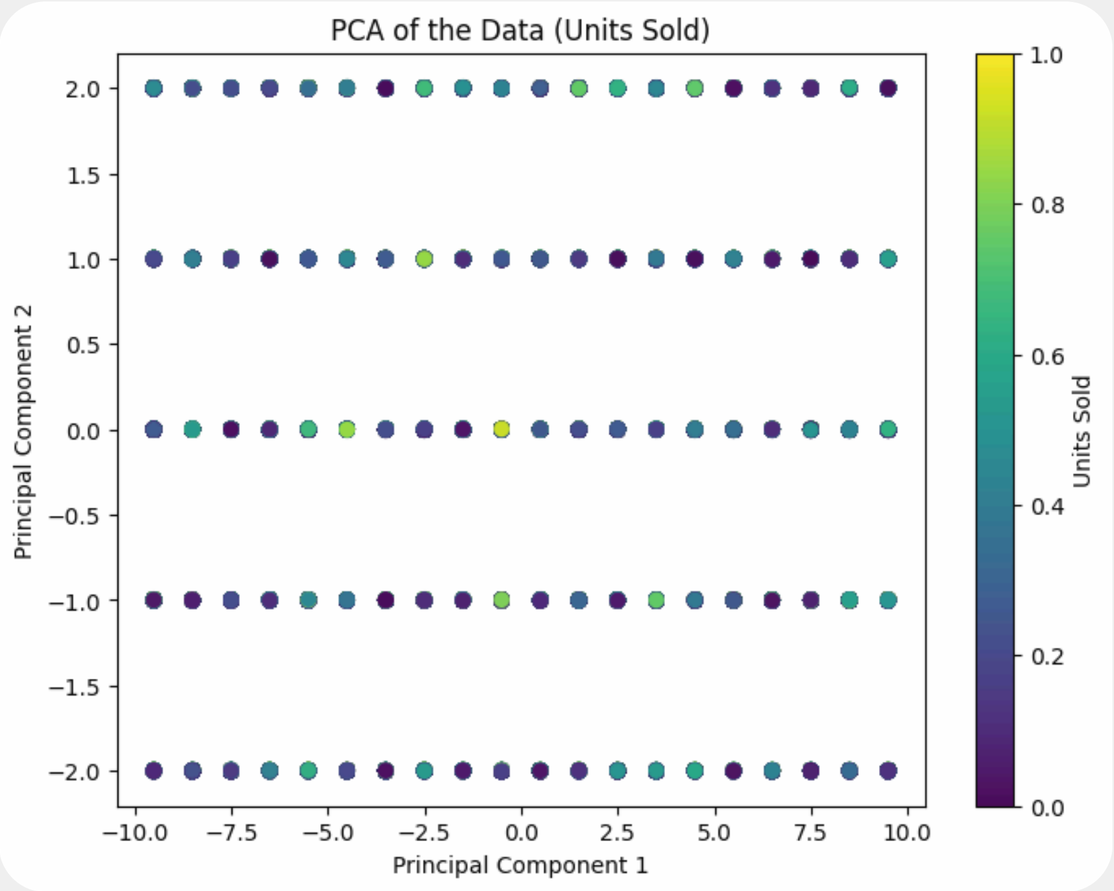
- ***Inventory Level***
- ***Price***
- ***Competitor Pricing***
- ***Inventory_Demand***
- ***UnitsSold_Price***

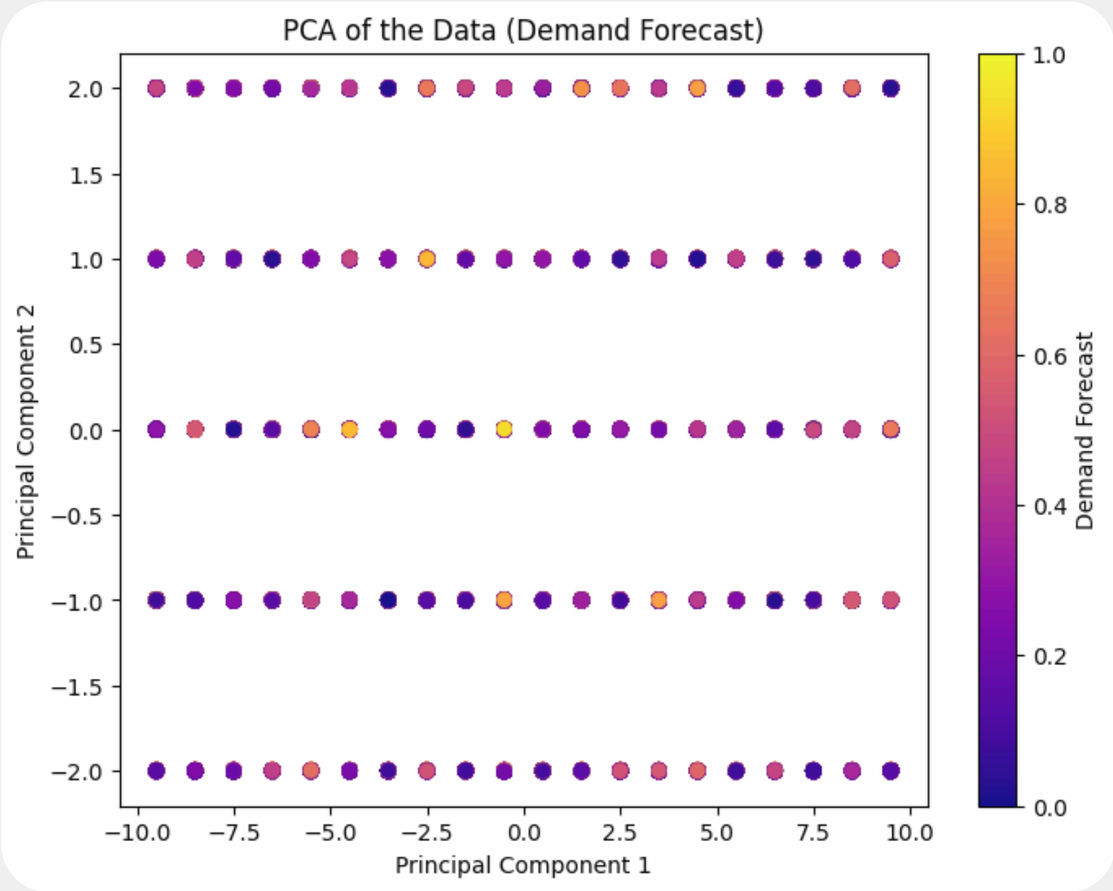


RFE Process Flowchart

✵ **RFE confirmed many of the features identified by ANOVA — boosting confidence in their relevance.**

Dimensionality Reduction – PCA





Dimensionality Reduction – PCA

Reduce the number of input features while retaining most of the data's variability. PCA helps simplify the dataset, reduce noise, and improve model efficiency.

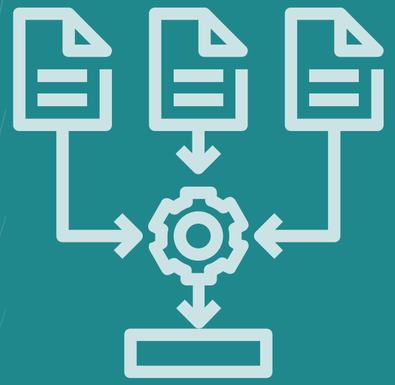
Approach

- Removed target columns (***Units Sold, Demand Forecast***) from the feature set
- Selected only numerical columns
- Applied Principal Component Analysis (PCA) to reduce to 2 components

Explained Variance

- The scatter plots show the distribution of data points in reduced 2D space
- Coloring by ***Units Sold*** and ***Demand Forecast*** reveals potential patterns or clusters

Principal Component	Explained Variance	Variance (%)
Component 1	0.8623	86.23%
Component 2	0.0519	5.19%



Model Building

Traditional ML Models

Deep Learning Models

Time Series Models

Modeling Approach

We experimented with both classical machine learning and deep learning models to determine the best-performing algorithms for sales and demand forecasting.

Strategy

- Applied train/test split (80/20) to validate performance.
- Evaluated using MAE, MAPE, and RMSE.

Goals

- Capture both linear and non-linear patterns in the data.
- Address temporal dependencies with deep learning models.
- Compare performance across model families to ensure robustness.

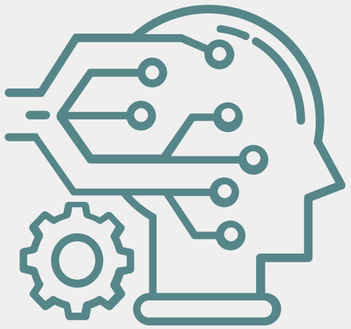
Algorithms Explored

- Classical ML: XGBoost, Random Forest, Gradient Boosting, SVM, Ridge
- Deep Learning: MLP, LSTM, CNN, BiLSTM, LSTM+CNN, BiLSTM+CNN
- Time Series: ARIMA, SARIMAX, Prophet

Traditional ML Models

We applied a variety of classical machine learning algorithms to benchmark performance on sales and demand prediction tasks.

Models Used



XGBoost Regressor

- Gradient-boosted trees for robust and accurate predictions.

Random Forest Regressor

- Ensemble of decision trees to reduce variance.

Gradient Boosting Regressor

- Sequential tree boosting to optimize performance.

Ridge Regression

- Linear model with L2 regularization to handle multicollinearity.

Support Vector Machines (SVM)

- Regression with optimal margin using kernel tricks.

Traditional Models – Evaluation Metrics

	Model	MAE	MSE	RMSE	R^2
0	XGBRegressor - Units Sold	0.010032	0.000182	0.013499	0.996828
1	XGBRegressor - Demand Forecast	0.005243	0.000048	0.006956	0.999100
2	RandomForestRegressor - Units Sold	0.011052	0.000252	0.015873	0.995615
3	RandomForestRegressor - Demand Forecast	0.005330	0.000086	0.009290	0.998394
4	GradientBoostingRegressor - Units Sold	0.017715	0.000521	0.022816	0.990939
5	GradientBoostingRegressor - Demand Forecast	0.014542	0.000356	0.018869	0.993376
6	SVR - Units Sold	0.035783	0.001837	0.042859	0.968028
7	SVR - Demand Forecast	0.041026	0.002261	0.047546	0.957944
8	Ridge Regression - Units Sold	0.051181	0.004280	0.065423	0.925503
9	Ridge Regression - Demand Forecast	0.049260	0.004007	0.063302	0.925455

Deep Learning Models

Architectures Explore

- **LSTM (Long Short-Term Memory)**
- **BiLSTM (Bidirectional LSTM)**
- **CNN (1D Convolutional Neural Network)**
- **Hybrid: LSTM+CNN, BiLSTM+CNN**

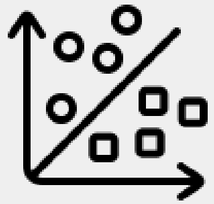
Why Deep Learning?

- Captures sequential dependencies in time-series data
- Effective for modeling seasonal and holiday trends

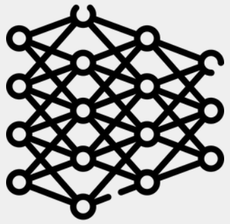
✵ **LSTM+CNN provided the best trade-off between training time and accuracy.**

Model Performance Comparison

Traditional vs Deep Learning



vs



Deep Learning Models

	Model	MAE	MSE	RMSE	R^2
0	XGBRegressor - Units Sold	0.010032	0.000182	0.013499	0.996828
1	XGBRegressor - Demand Forecast	0.005243	0.000048	0.006956	0.999100
2	RandomForestRegressor - Units Sold	0.011052	0.000252	0.015873	0.995615
3	RandomForestRegressor - Demand Forecast	0.005330	0.000086	0.009290	0.998394
4	GradientBoostingRegressor - Units Sold	0.017715	0.000521	0.022816	0.990939
5	GradientBoostingRegressor - Demand Forecast	0.014542	0.000356	0.018869	0.993376
6	SVR - Units Sold	0.035783	0.001837	0.042859	0.968028
7	SVR - Demand Forecast	0.041026	0.002261	0.047546	0.957944
8	Ridge Regression - Units Sold	0.051181	0.004280	0.065423	0.925503
9	Ridge Regression - Demand Forecast	0.049260	0.004007	0.063302	0.925455
10	LSTM - Sales Prediction	0.009036	0.000143	0.011938	0.997520
11	LSTM - Demand Forecast Prediction	0.008172	0.000114	0.010667	0.997883
12	MLP - Sales Prediction	0.002784	0.000021	0.004555	0.999639
13	MLP - Demand Forecast Prediction	0.003455	0.000023	0.004833	0.999565
14	CNN - Sales Prediction	0.010742	0.000193	0.013908	0.996633
15	CNN - Demand Forecast Prediction	0.013455	0.000273	0.016513	0.994927
16	BiLSTM - Sales Prediction	0.007977	0.000111	0.010548	0.998064
17	BiLSTM - Demand Forecast Prediction	0.008313	0.000118	0.010850	0.997810
18	CNN+LSTM - Sales Prediction	0.004165	0.000042	0.006514	0.999262
19	CNN+LSTM - Demand Forecast Prediction	0.004169	0.000038	0.006147	0.999297
20	BiLSTM+CNN - Sales Prediction	0.007394	0.000085	0.009223	0.998520
21	BiLSTM+CNN - Demand Forecast Prediction	0.003974	0.000032	0.005624	0.999412

Time Series Models

Models Used

- **ARIMA**
- **SARIMAX**
- **Prophet**

Performance Highlights

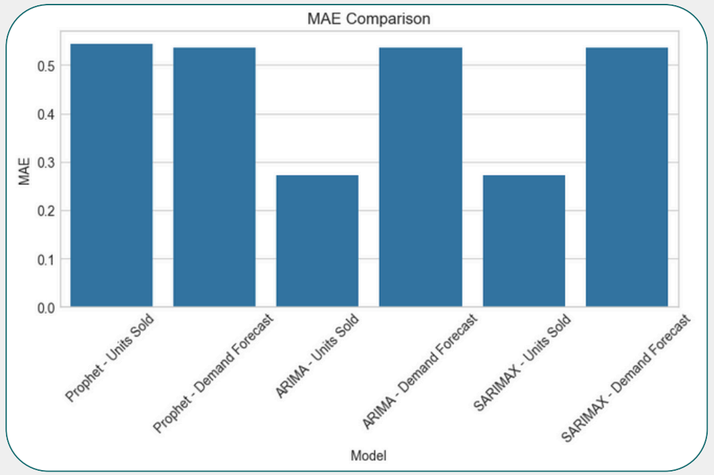
- **SARIMAX** handled seasonality better than **ARIMA**.
- Prophet offered trend + seasonality decomposition and interpretability.

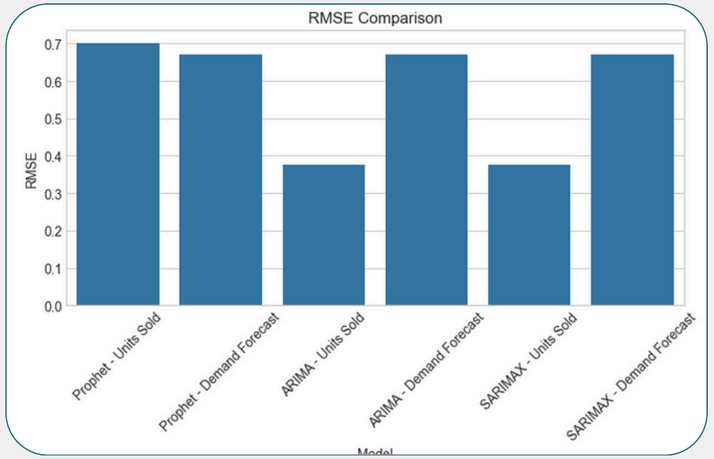
✵ **Time series models captured temporal trends but underperformed compared to deep learning models in forecasting accuracy and complexity.**

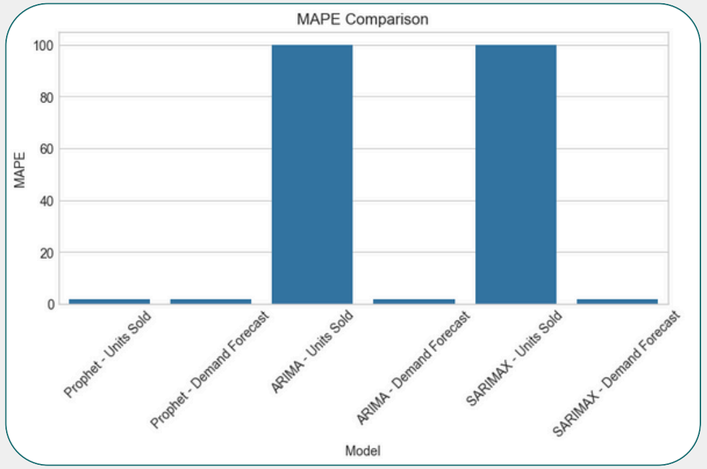
Time Series Models – Forecast Accuracy

	Model	MAE	RMSE	MAPE
0	Prophet - Units Sold	0.543112	0.700367	1.785396
1	Prophet - Demand Forecast	0.534832	0.669993	1.645397
2	ARIMA - Units Sold	0.271121	0.376648	100.000000
3	ARIMA - Demand Forecast	0.534832	0.669993	1.645397
4	SARIMAX - Units Sold	0.271121	0.376648	100.000000
5	SARIMAX - Demand Forecast	0.534832	0.669993	1.645397

Time Series Models – Forecast Accuracy







Across all time series models, **ARIMA exhibited high MAPE**, signaling erratic relative errors. In contrast, **Prophet** and **SARIMAX** provided stable and lower absolute errors, making them more reliable for short-term forecasting in this context.

Pipeline and Hyper Parameter

Efficient, Consistent, and Reproducible ML Workflows

Raw Data → Preprocessing → Model Training → Evaluation

Consistency: Automates steps like scaling & encoding.

Efficiency: Reduces complexity & human error.

Reproducibility: Modular experiments across datasets.

Key Pipeline Components

Preprocessing:

- **Standardize with StandardScaler**
- **Encode categories with OneHotEncoder**

Model Integration:

- **Example: RandomForestRegressor()**
- **Smooth data flow through the pipeline**

Unified Interface:

- **Use .fit(), .predict() & GridSearchCV seamlessly**

[Scaler] → [OneHotEncoder] → [Model (e.g., RF)]

Optimizing with Hyperparameter Tuning

What Are Hyperparameters?

- Configurations not learned from data
- Examples: max_depth, n_estimators

Parameter	Values Tested
n_estimators	100, 200
max_depth	None, 10
min_samples_split	2, 5

Why Tune Them?

- Prevent overfitting
- Maximize performance on unseen data

Our Tuning Approach & Results

Grid Search:

Exhaustive search over param combinations

Cross-Validation:

Used 3-fold CV for robust evaluation

Outcome:

Improved metrics: ↓ MSE, ↑ R^2

Final model generalized well

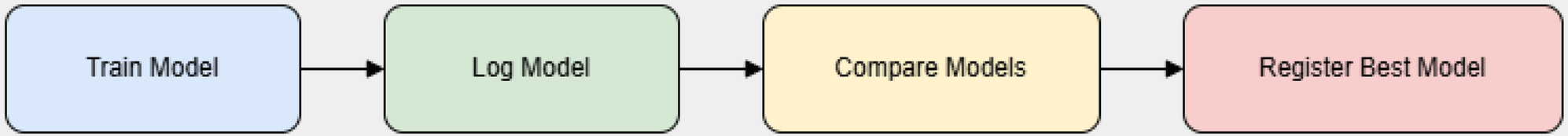
[Train Data] → [Pipeline] → [GridSearchCV] → [Best Estimator]

MLflow Integration

Used MLflow to track experiments, log models, and manage versions during training and tuning.

Key Features

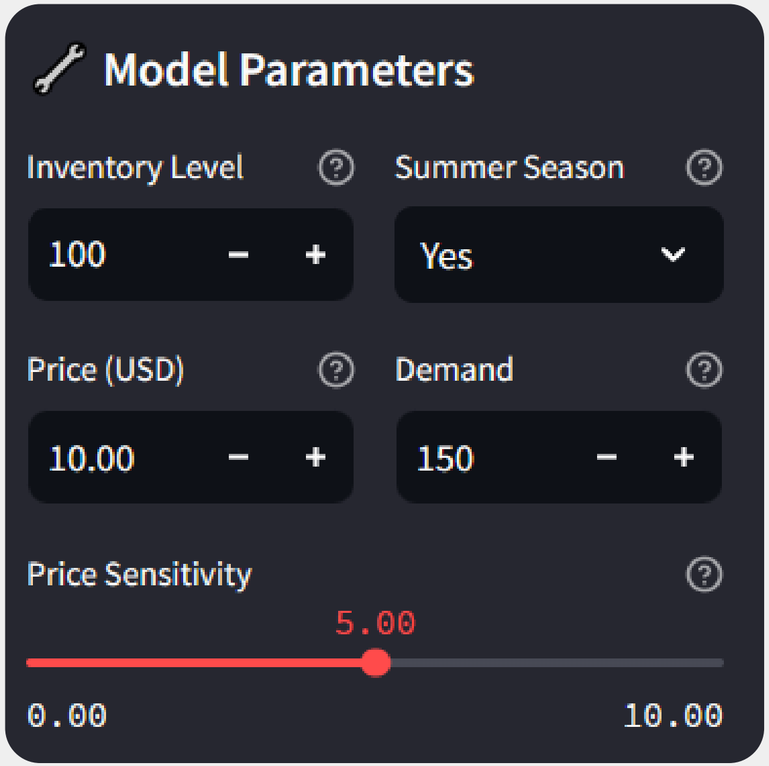
- **Run Tracking:** Logged parameters (e.g., model type, hyperparameters) and performance metrics (MAE, RMSE, R^2).
- **Model Registry:** Saved best-performing models for both Units Sold and Demand Forecast.
- **Reproducibility:** Enabled reproducible training workflows and easier collaboration.



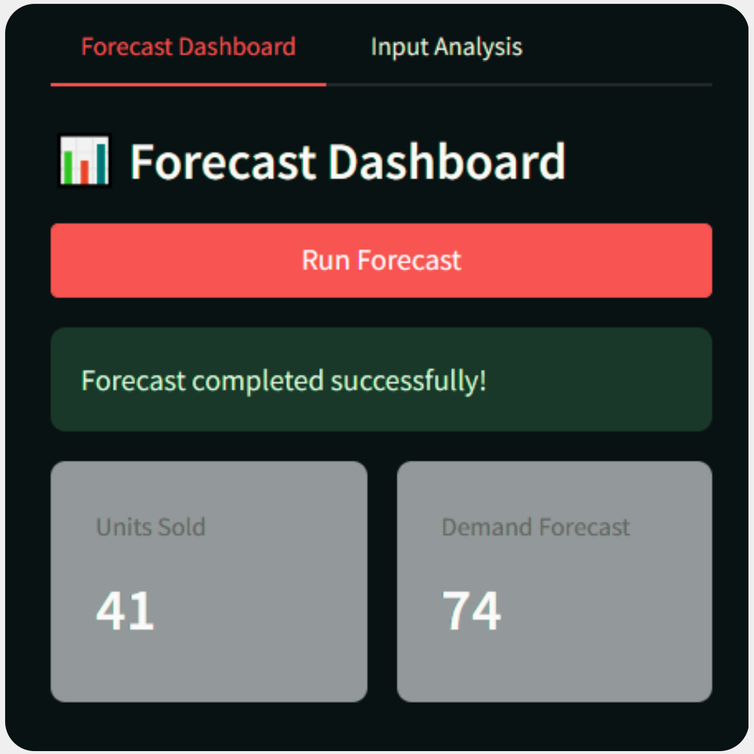
Model Deployment – Streamlit App

A web-based application is built using Streamlit to serve real-time predictions.

Input features



Output



Business Impact

Translate forecasts into meaningful outcomes for retail operations.

Predicted Outcomes

- **Better Inventory Planning: Reduce stockouts and overstocks by forecasting accurate demand.**
- **Sales Optimization: Align pricing and promotional strategies with predicted sales volume.**
- **Cost Savings: Minimize holding costs and improve turnover by stocking based on real demand.**

Key Value Adds

- **Demand predictions drive smarter procurement.**
- **Sales forecasts inform staffing and logistics.**
- **Seasonality & promotion awareness leads to more agile decision-making.**

Strategic Execution Plan

1. Actionable Insights

- **Align forecasts with sales and inventory goals.**
- **Embed predictions into planning processes.**

2. Team Enablement

- **Equip teams with forecast-driven tools.**
- **Use clear dashboards for quick decisions.**

3. Impact Tracking

- **Monitor KPIs: stock levels, lost sales, accuracy.**
- **Review and adjust regularly.**

"Empowering leaders to act faster, cut costs, and respond proactively with data-driven insights."

Predicted Outcomes

Improved Inventory Planning

- Prevent stockouts and overstock situations
- Predict item-level demand in real time

Smarter Sales Optimization

- Align pricing and promotions with demand forecasts
- Enable adaptive discounting strategies

Operational Cost Reduction

- Lower holding and storage expenses
- Boost inventory turnover efficiency

Thank You