SUPPLY CHAIN OPTIMIZATION

Pankhuri Jain
23030124029

Marjari Pusadkar
23030124031

Sanchi Mittal
23030124265

Agenda

01	Problem Definition
02	Technology used
03	Flowchart
04	Data Collection
05	Understanding and preprocessing
06	Analytics
07	Model selection
08	ML/Time series
09	Model training
10	Optimization
11	Visualization
12	Conclusion

PROBLEM DEFINITION

Supply Chain Optimisation focuses on improving the efficiency of product flow from suppliers to customers. In this project, we analyse sales and inventory data to improve demand forecasting and optimise stock levels. The goal is to reduce costs and ensure smooth operations. By using data-driven insights, we help the business make smarter supply chain decisions.

Business Challenge

- Inefficient demand forecasting
- Poor inventory planning across locations
- Lack of data-driven decision-making

Project Objective

- Optimise supply chain operations
- Improve demand prediction accuracy
- Ensure timely product availability

Expected Business Impact

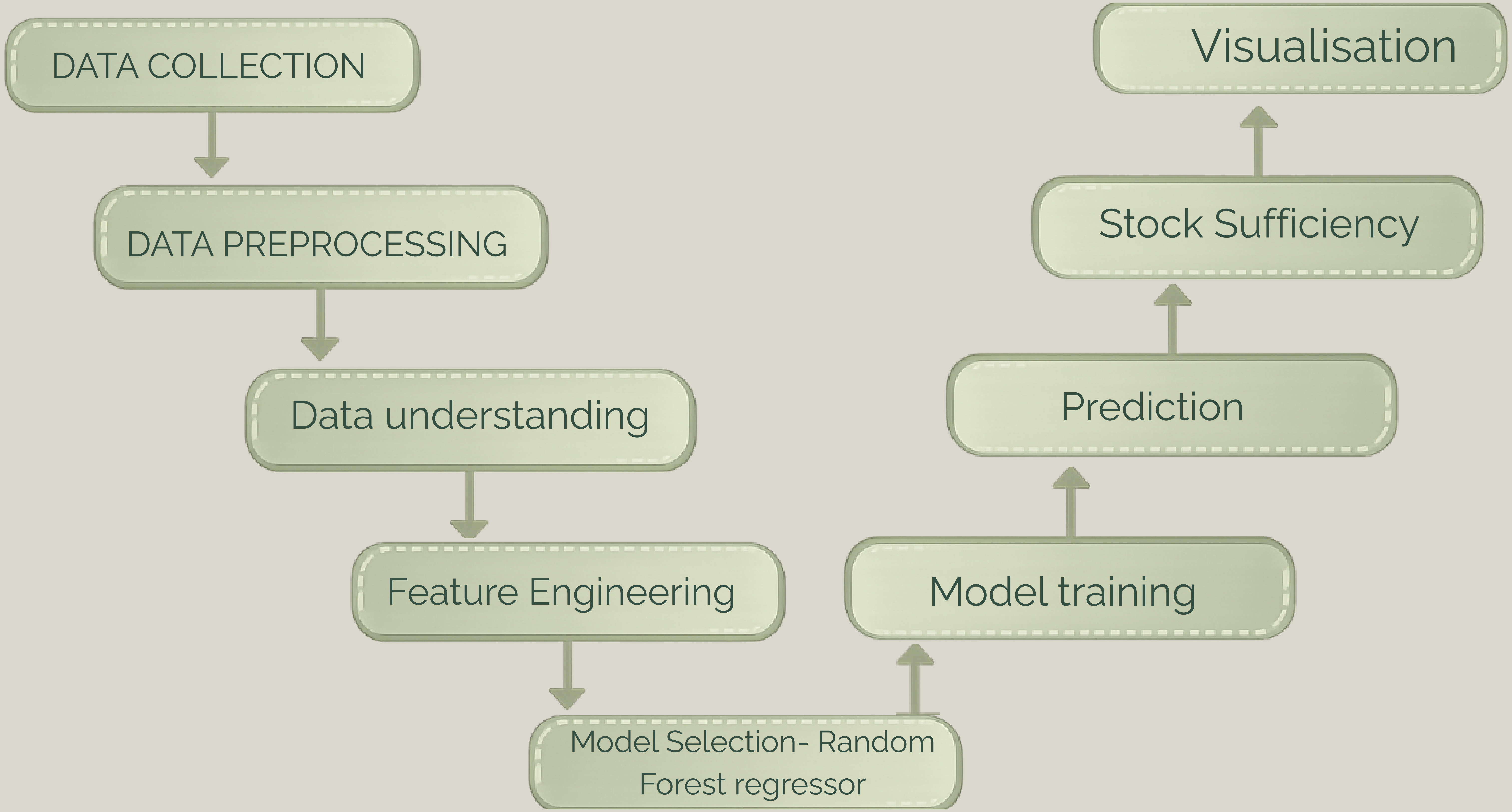
- Lower operational costs
- Faster replenishment cycles
- Improved profitability
- Better customer satisfaction

TECHNOLOGY USED

- MATPLOTLIB
- NUMPY
- PANDAS
- SCIKIT LEARN
- SEABORN

- PYTHON
- JENKINS

FLOWCHART


DATA COLLECTION
DATA PREPROCESSING
Data understanding
Feature Engineering
Model Selection- Random Forest regressor
Model training
Prediction
Stock Sufficiency
Visualisation

DATA COLLECTION

- The dataset was sourced from Kaggle and contains retail inventory and sales data.
- The dataset is deemed suitable for analyzing demand forecasting and supply chain optimization.
- df.head() displays the first 5 records of the dataset.
- It helps in understanding column names, data structure, and sample values.

	Date	Store ID	Product ID	Category	Region	Inventory Level	Units Sold	Units Ordered	Demand Forecast	Price	Discount	Weather Condition	Holiday/Promotion	Competitor Pricing	Seasonality
0	2022-01-01	S001	P0001	Groceries	North	231	127	55	135.47	33.50	20	Rainy	0	29.69	Autumn
1	2022-01-01	S001	P0002	Toys	South	204	150	66	144.04	63.01	20	Sunny	0	66.16	Autumn
2	2022-01-01	S001	P0003	Toys	West	102	65	51	74.02	27.99	10	Sunny	1	31.32	Summer
3	2022-01-01	S001	P0004	Toys	North	469	61	164	62.18	32.72	10	Cloudy	1	34.74	Autumn
4	2022-01-01	S001	P0005	Electronics	East	166	14	135	9.26	73.64	0	Sunny	0	68.95	Summer

UNDERSTANDING & PRE-PROCESSING

	Inventory Level	Units Sold	Units Ordered	Demand Forecast	\
count	73100.000000	73100.000000	73100.000000	73100.000000	
mean	274.469877	136.464870	110.004473	141.494720	
std	129.949514	108.919406	52.277448	109.254076	
min	50.000000	0.000000	20.000000	-9.990000	
25%	162.000000	49.000000	65.000000	53.670000	
50%	273.000000	107.000000	110.000000	113.015000	
75%	387.000000	203.000000	155.000000	208.052500	
max	500.000000	499.000000	200.000000	518.550000	

	Price	Discount	Holiday/Promotion	Competitor Pricing
count	73100.000000	73100.000000	73100.000000	73100.000000
mean	55.135108	10.009508	0.497305	55.146077
std	26.021945	7.083746	0.499996	26.191408
min	10.000000	0.000000	0.000000	5.030000
25%	32.650000	5.000000	0.000000	32.680000
50%	55.050000	10.000000	0.000000	55.010000
75%	77.860000	15.000000	1.000000	77.820000
max	100.000000	20.000000	1.000000	104.940000

- After running df.describe(), we observe that the dataset contains 73,100 records with no missing values in the numerical columns.
- The average inventory is 274 units, while the average sales are 136 units, indicating that stock is generally higher than sales.
- The average demand forecast (141) is close to actual sales.

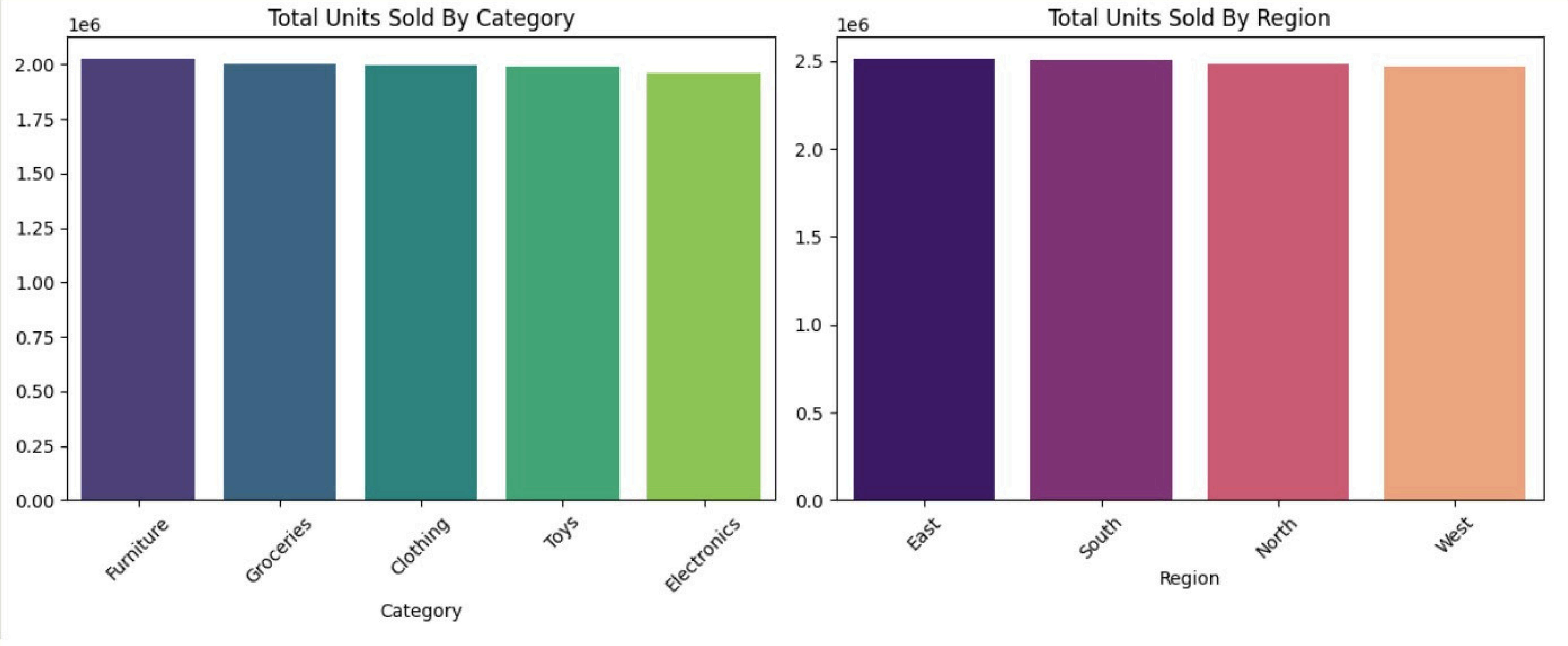
UNDERSTANDING & PRE-PROCESSING

```
Date                   0
Store ID               0
Product ID             0
Category               0
Region                 0
Inventory Level        0
Units Sold             0
Units Ordered          0
Demand Forecast        0
Price                  0
Discount               0
Weather Condition      0
Holiday/Promotion      0
Competitor Pricing     0
Seasonality            0
dtype: int64
```

- After applying df.isnull().sum(), we found that all columns show 0 missing values.
- This means the dataset is complete and clean, with no data gaps.
- Therefore, no missing value handling or imputation is required during preprocessing.

```
df['date'] = pd.to_datetime(df['Date'])
df['month'] = df['date'].dt.month
df['day_of_week'] = df['date'].dt.dayofweek
df['is_weekend'] = df['day_of_week'].isin([5,6]).astype(int)
df['sales']=df['Units Sold'] * df['Price']
```

ANALYTICS



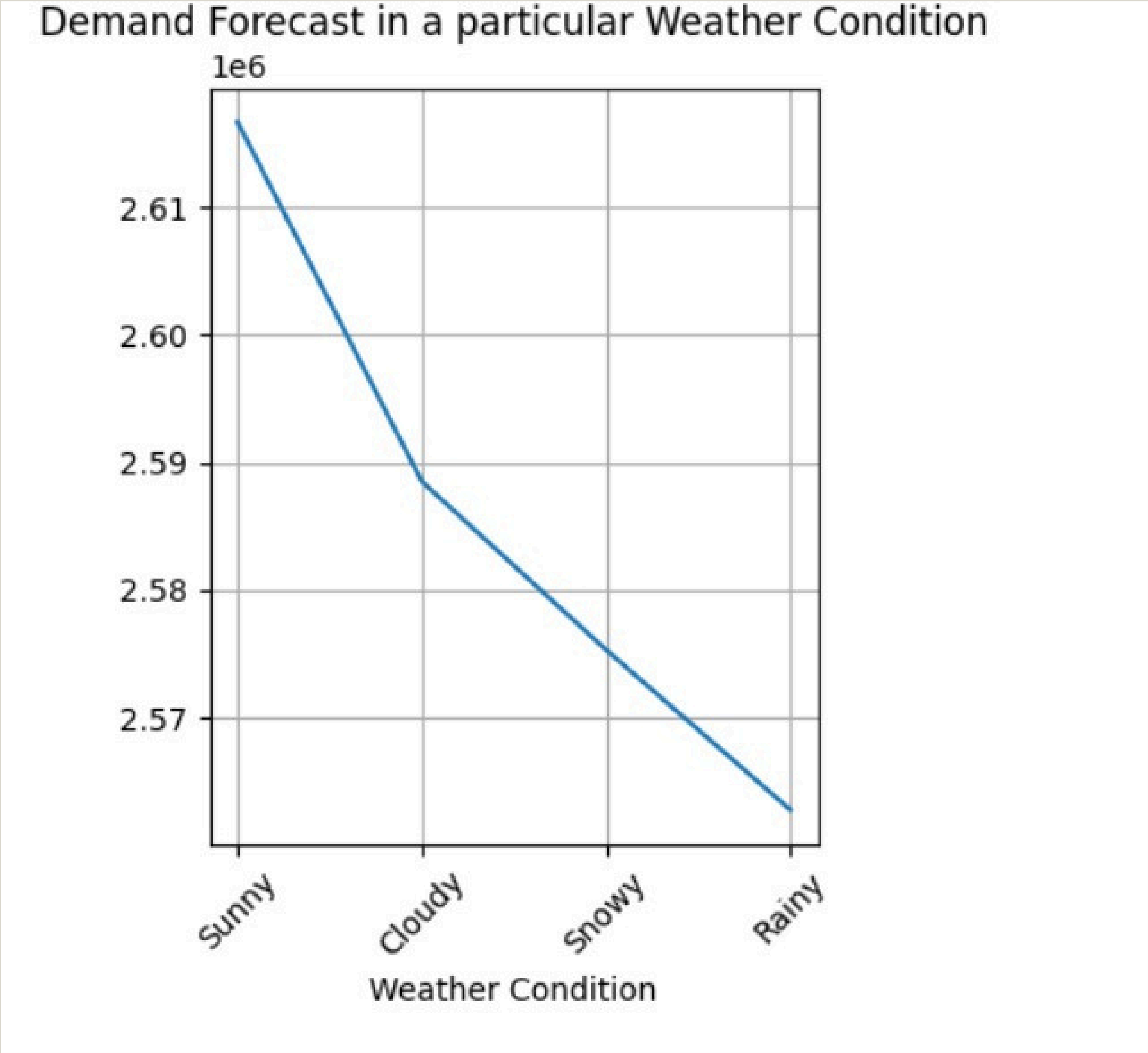
From the analysis, we observe that product demand is evenly spread across categories and regions. There are no significant demand spikes or shortages in any specific area. This indicates a stable supply chain structure. Minor optimizations in high-performing categories and regions can further improve efficiency.

ANALYTICS

Key Observations

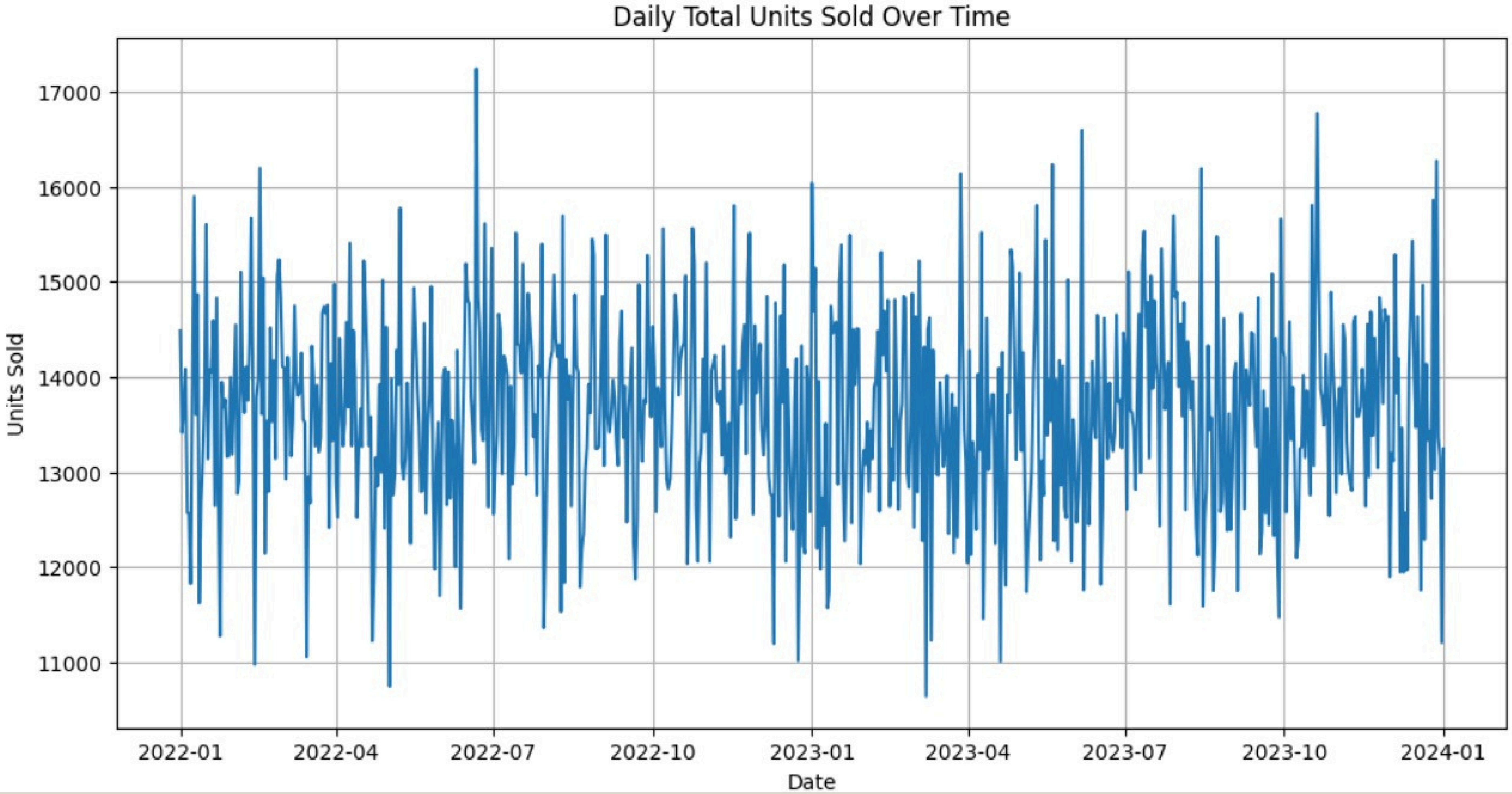
- Taking into consideration of external forces
- Demand is highest during Sunny weather conditions.
- Demand further drops during Snowy conditions.
- Rainy weather shows the lowest demand among all conditions.

This indicates a negative trend in demand as weather conditions become less favorable.



MACHINE LEARNING / TIME SERIES

```
Features selected: ['Inventory Level', 'Units Ordered', 'Demand Forecast', 'Price', 'Discount', 'Holiday/Promotion', 'Competitor Pricing', 'Month', 'Day
OfWeek', 'Year', 'Category_Electronics', 'Category_Furniture', 'Category_Groceries', 'Category_Toys', 'Region_North', 'Region_South', 'Region_West', 'We
ather Condition_Rainy', 'Weather Condition_Snowy', 'Weather Condition_Sunny', 'Seasonality_Spring', 'Seasonality_Summer', 'Seasonality_Winter']
```



MODEL SELECTION

Chosen Algorithm: Random Forest Regressor
For this project, we selected the Random Forest Regressor algorithm to build our prediction model. Random Forest Regressor is an Machine learning algorithm used to predict continuous numeric values and builds many decision trees and takes the average of their predictions to produce the final output.

Relevance to Supply Chain Optimization

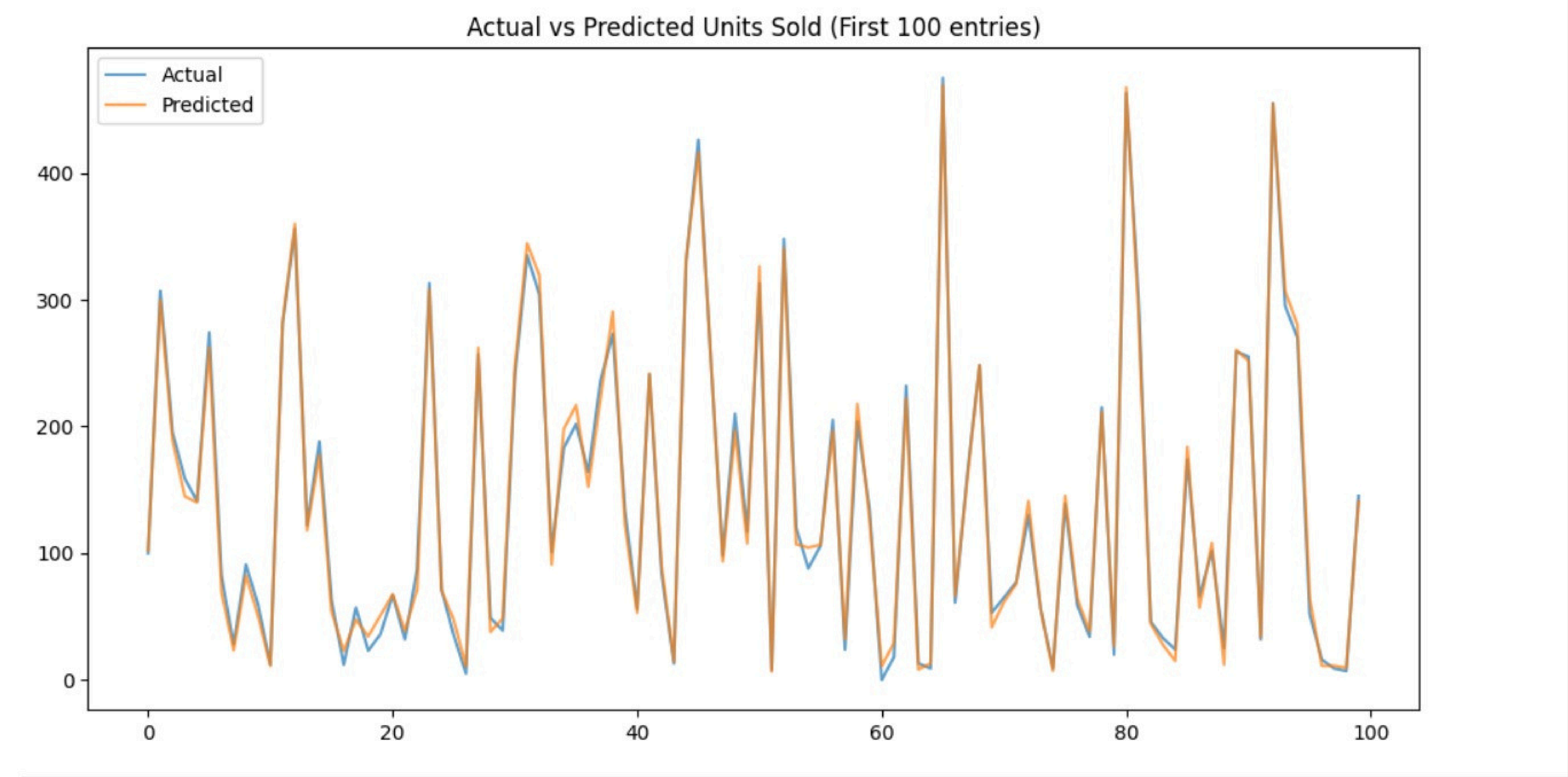
Random Forest Regressor helps in accurately predicts continuous variables such as demand, lead time and cost by handling complex and non linear relationships in supply chain data.This helps in improving inventory planning, cost control and overall operational efficiency.

Role in Supply Chain Optimization

By using Random Forest Regressor, we can predict product demand more precisely, which helps reduce stockouts, avoid overstocking, and improve overall supply chain efficiency.

MODEL TRAINING

RMSE SCORE:8.6111
R2 SCORE:0.9937

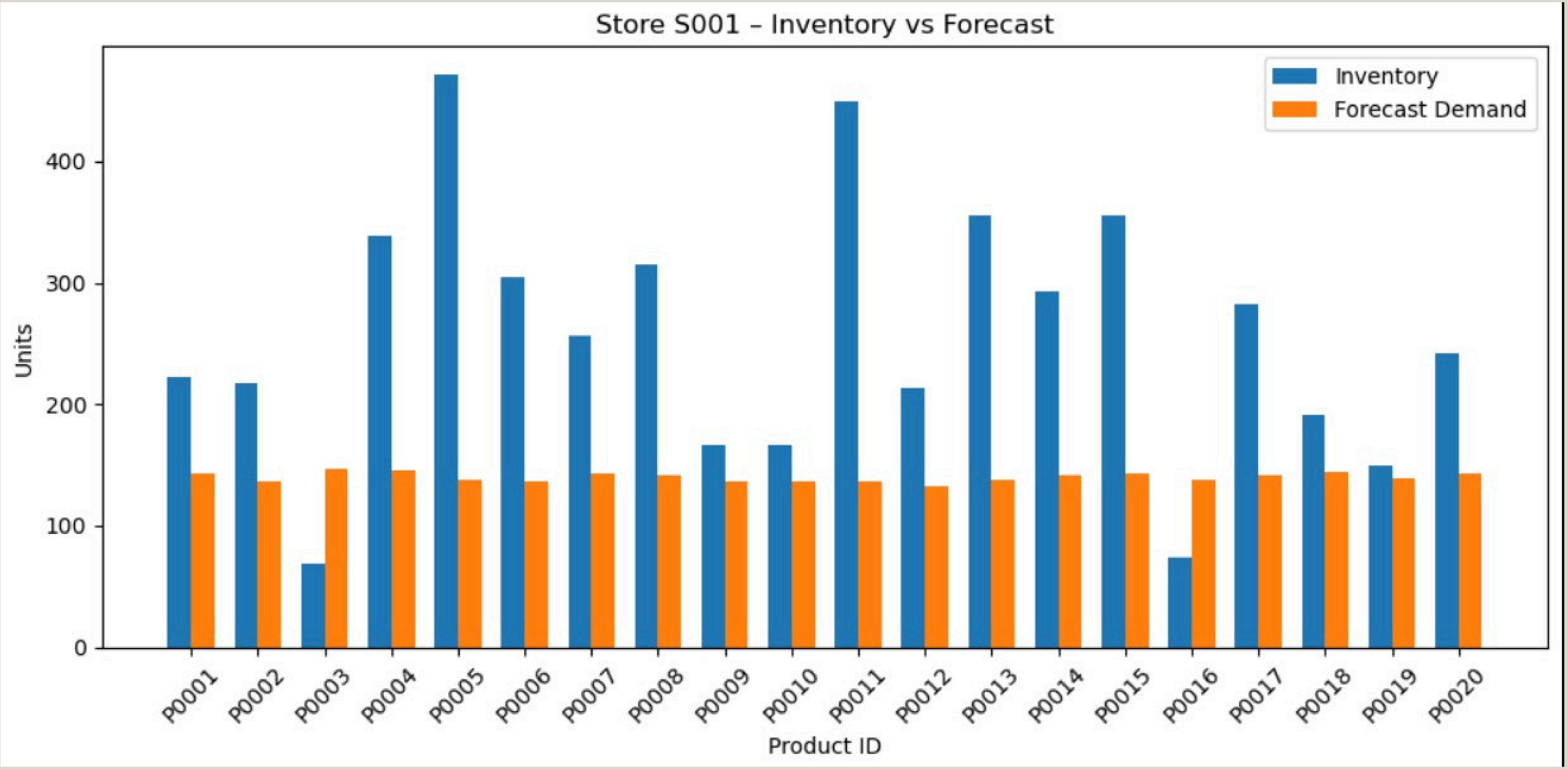


OPTIMIZATION

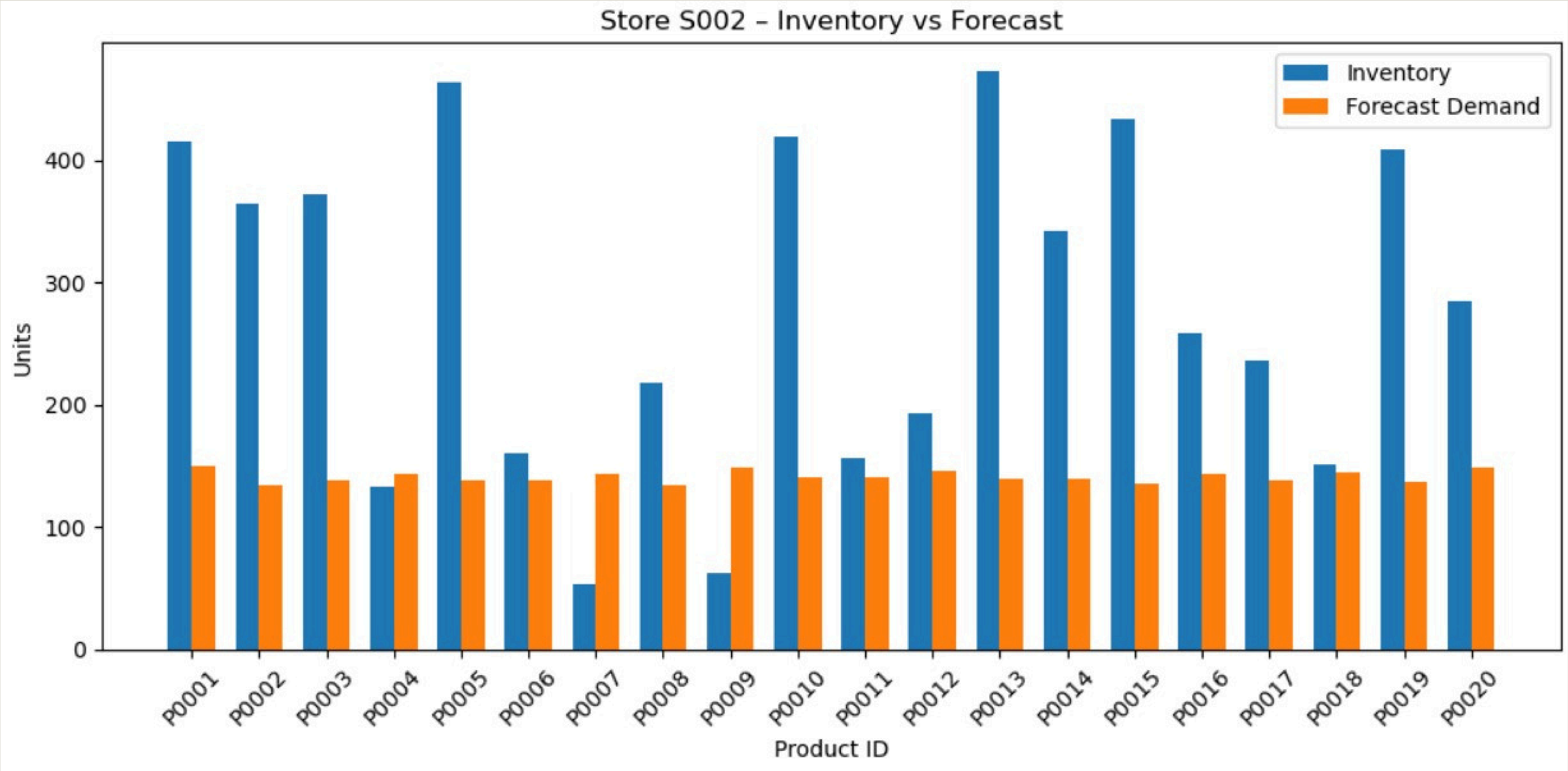
- Inventory Level represents the current quantity of products available in the store.
- Demand Forecast shows the aggregated mean demand forecast.
- Stock_Sufficient is a logical indicator (True/False) that shows whether the current inventory can meet forecasted demand.
- Shortage_Qty calculates the additional quantity required when inventory is lower than predicted demand.

	Store ID	Product ID	Inventory Level	Demand Forecast	stock_sufficient	shortage_qty
0	S001	P0001	223	143.063187	True	0.000000
1	S001	P0002	217	136.417538	True	0.000000
2	S001	P0003	69	146.561204	False	77.561204
3	S001	P0004	338	146.057196	True	0.000000
4	S001	P0005	471	138.232791	True	0.000000

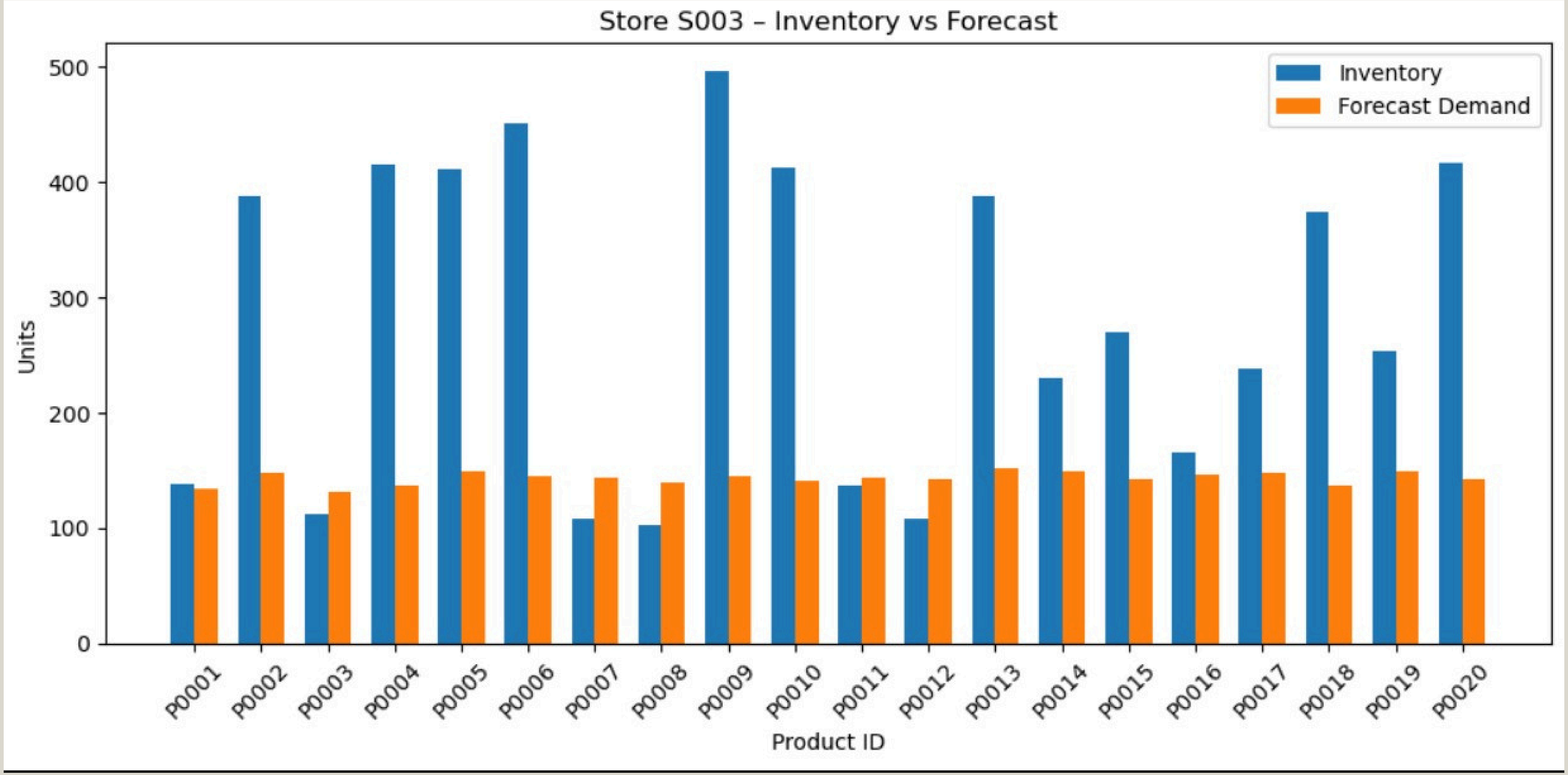
OPTIMIZATION & VISUALIZATION



OPTIMIZATION & VISUALIZATION



OPTIMIZATION & VISUALIZATION



CONCLUSION

This project focused on optimising supply chain operations by integrating demand forecasting with inventory analysis. Using a real-world retail dataset, we conducted data understanding, preprocessing, feature engineering, and exploratory data analysis to identify demand patterns across categories, regions, and external factors such as weather conditions.

A Random Forest Regressor model was implemented to accurately forecast product demand. The model demonstrated excellent predictive performance with a high R^2 score (~0.99) and low RMSE (~8.61), indicating strong reliability and minimal prediction error.

ACKNOWLEDGEMENT

We sincerely acknowledge Mr. Vikas Choudhary Sir for giving us the opportunity to work on this project and for his valuable guidance throughout the process.
His support and encouragement helped us successfully complete this project and enhance our learning experience.

REFERENCES

- Dataset: https://www.kaggle.com/datasets/programmer3/retail-and-supply-chain-operations-data
- Github repository: https://github.com/Pankhuri1709/Open-Sources-Tools-Project

THE END

THANK YOU FOR LISTENING

Pankhuri Jain
23030124029

Marjari Pusadkar
23030124031

Sanchi Mittal
23030124265