Medical Inventory Optimization and Forecasting

Contents

Project Overview and Scope

Overview

Project Purpose:
Develop a pharmaceutical forecasting model to prevent drug shortages.

3 Major Phases:
1. Data preprocessing
2. Exploratory data analysis (EDA)
3. Forecasting Model (time-series analysis)

Project Scope:
Develop a forecasting model that predicts the demand for medical supplies.

Business Problem

- Drug shortages in stocks
- Increase of bounce rate

Business Understanding

Objective

- Minimize drug shortages.
- Maximize availability of drug, customer satisfaction, and profits.

Constraints

- Maximize medicine availability.
- Minimize return quantity.

CRISP-ML(Q) Methodology

This project involve 6 phases of CRISP-ML(Q) methodology:

Business and Data Understanding:

- Address drug shortages and improve patient care.
- Gather data on drugs sales, stock levels, and quantity.

Data Preparation:

- Obtain historical sales data.
- Clean and preprocess data: Handle missing values, remove duplicates, and format data for analysis.

Model Building and Tuning:

- Develop forecasting models: Create machine learning models to predict future drug demand.
- Tune model parameters: Optimize model performance by adjusting parameters.

Evaluation:

- Assess model performance: Measure how accurately the model predicts medicine demand.
- Evaluate against objectives: Check if the model reduces shortages of drugs.

CRISP-ML(Q) Methodology

Model Deployment:

- Integrate with pharmacy system: Incorporate the forecasting model into the pharmacy's inventory management system.
- Ensure compatibility: Verify that the model interacts smoothly with existing processes.

Monitoring and Maintenance:

- Continuously monitor performance: Regularly check how well the model forecasts and prevents shortages.
- Update as needed: Adjust the model if changes in drugs demand patterns occur.

Technical Stacks

Programming Languages:
- Python

Data Manipulation and Analysis:
- Pandas
- NumPy

Data Visualization:
- matplotlib
- Seaborn

AutoEDA
D-tale

Time Series Forecasting:
- Statsmodels

Optimization:
- SciPy

Machine Learning (for Continuous Improvement):
- scikit-learn

Database (for Data Storage and Retrieval):
- MySQL
- mysql.connector

Notebooks and Development Environment:
- Jupyter Notebook
- Visual Studio Code

Project Architecture

Data Collection
Collect data from pharmacy transactions (12 months) and store data in MySQL database.

Data Preprocessing
Clean, transform, handle missing values, outliers, duplicates, and type casting.

Exploratory Data Analysis (EDA)
Analyze sales patterns, quantity, and trends over time

Feature Engineering:
Convert Dateofbill to datetime format and encode the ordinal column i.e., Dateofbill

Forecasting Model Development:
Python (statsmodels) for time-series forecasting.
Model A: Random Forest Regression
Model B: Linear Regression

Model Evaluation
Choose the best model for time-series forecasting.

Data Collection and Understanding

Data collection in this project involves gathering all the necessary information that will be used for analysis, modeling, and optimization.

Based on the secondary data source, the data types can be categorized as below:

Data	Data Type
Typeofsales	Nominal, Categorical
Patient_ID	Nominal, Categorical
Specialisation	Nominal, Categorical
Dept	Nominal, Categorical
Dateofbill	Ordinal, Categorical
Quantity	Ratio, Numerical
ReturnQuantity	Ratio, Numerical
Final_Cost	Ratio, Numerical
Final_Sales	Ratio, Numerical
RtnMRP	Ratio, Numerical
Formulation	Nominal, Categorical
DrugName	Nominal, Categorical
SubCat	Nominal, Categorical
SubCat1	Nominal, Categorical

Data Information

Typeofsales: Different types of transactions, such as ‘Sale’ or ‘Return’. It provides insight into the nature of the transaction.

Patient_ID: A unique identifier for each patient. This can help track individual patient behaviors and preferences.

Specialisation: Represents the specialization (e.g. Specialisation1) of the healthcare professional associated with the transaction. This could affect the types of drugs prescribed and the sales patterns.

Dept: Indicates the department (e.g. Department1) in the pharmacy or hospital where the transaction occurred. This can provide context about the location and context of the transaction.

Dateofbill: The date of the transaction (YYYY:MM:DD). Time-related information is to develop temporal patterns, trends, and seasonality in drug sales.

Quantity: The quantity of medicine sold in a particular transaction. This is directly related to sales volume.

ReturnQuantity: The quantity of medicine returned. It provides insights into bounce rate.

Data Information

Final_Cost: The cost of the medicine in a transaction. This helps determine the financial aspect of each sale.

Final_Sales: The total sales amount generated from a transaction. It is for assessing revenue.

RtnMRP: The Maximum Retail Price (MRP) of the returned medicine. This is relevant for analyzing returns and pricing strategies.

Formulation: Indicates the formulation (e.g. Form1, Patent) of the medicine. It could impact sales based on the type of formulation.

DrugName: The name of the medicine being sold. This provides specific information about the products being handled.

SubCat: A subcategory classification for the drugs. It can help group drugs with similar characteristics together.

SubCat1: Secondary subcategory classification for the drugs, potentially providing a more detailed categorization.

Data Dictionary

Field Name	Description	Data Type	Data Format	Data Size	Relevance
Typeofsales	Type of sales transaction (e.g., Sale, Return)	Categorical	Varchar	10	Transaction classification
Patient_ID	Unique ID for each patient	Numeric	Int	11	Patient identification
Specialisation	Specialization of medical professional	Categorical	Varchar	30	Medical context
Dept	Department in pharmacy	Categorical	Varchar	30	Transaction context
Dateofbill	Date of the sales transaction	Date	YYYY-MM-DD	Date	Transaction date
Quantity	Quantity of medicine sold	Numeric	Int	2	Sales volume
ReturnQuantity	Quantity of medicine returned	Numeric	Int	2	Returns volume
Final_Cost	Final cost after discounts/adjustments	Numeric	Decimal	10,3	Products costs
Final_Sales	Final sales amount	Numeric	Decimal	10,3	Total transaction sales amount
RtnMRP	Maximum Retail Price of returned medication	Numeric	Decimal	10,3	Price comparison
Formulation	Medication formulation	Categorical	Varchar	10	Product characteristics
DrugName	Name of the drug	Categorical	Varchar	1000	Product identification
SubCat	Subcategory of the drug	Categorical	Varchar	1000	Product categorization
SubCat1	Secondary subcategory of the drug	Categorical	Varchar	1000	Detailed categorization

System Requirements

1. Software Requirements:

- **Operating System:** Windows11
- **Python:** Python 3.10.9, packages (pandas, matplotlib, seaborn, scikit-learn, etc.)
- **Database:** MySQL
- **Data Visualization Tools:** Jupyter Notebook, D-tale

1. Collaboration Tools:

- Google Meet for communication and coordination within our project team.

2. Backup and Recovery:

- Back up project data and code to prevent data loss using cloud storage.

Data Preprocessing: Type Casting

```python
pharma_data["Patient_ID"] = pharma_data["Patient_ID"].astype('str')
```

✓ 0.0s   Python

```python
pharma_data.dtypes
```

✓ 0.0s   Python

```
Typeofsales       object
Patient_ID        object
Specialisation    object
Dept              object
Dateofbill        object
Quantity           int64
ReturnQuantity     int64
Final_Cost        object
Final_Sales       object
RtnMRP            object
Formulation       object
DrugName          object
SubCat            object
SubCat1           object
dtype: object
```

Missing Values Observation : Impute of Mode

```python
pharma_data.isnull().sum()
```

✓ 0.0s Python

```
Typeofsales       0
Patient_ID        0
Specialisation    0
Dept              0
Dateofbill        0
Quantity          0
ReturnQuantity    0
Final_Cost        0
Final_Sales       0
RtnMRP            0
Formulation       0
DrugName          0
SubCat            0
SubCat1           0
dtype: int64
```

Removing Duplicates

```python
duplicate = pharma_data.duplicated()
sum(duplicate)
```

✓ 0.1s Python

```
26
```

```python
# Remove duplicates
pharma_data = pharma_data.drop_duplicates()
duplicate = pharma_data.duplicated()
sum(duplicate)
```

✓ 0.0s Python

```
0
```

One-Hot Encoding

	Dateofbill	Quantity	Apr	Aug	Dec	Feb	Jan	Jul	Jun	Mar	May	Nov	Oct	Sep	log_Quantity	t	t_square
0	Jan	2309	0	0	0	0	1	0	0	0	0	0	0	0	7.744570	1	1
1	Feb	2118	0	0	0	1	0	0	0	0	0	0	0	0	7.658228	2	4
2	Mar	2812	0	0	0	0	0	0	0	1	0	0	0	0	7.941651	3	9
3	Apr	2947	1	0	0	0	0	0	0	0	0	0	0	0	7.988543	4	16
4	May	2645	0	0	0	0	0	0	0	0	1	0	0	0	7.880426	5	25
5	Jun	2124	0	0	0	0	0	0	1	0	0	0	0	0	7.661056	6	36
6	Jul	3006	0	0	0	0	0	1	0	0	0	0	0	0	8.008366	7	49
7	Aug	2982	0	1	0	0	0	0	0	0	0	0	0	0	8.000349	8	64
8	Sep	2460	0	0	0	0	0	0	0	0	0	0	0	1	7.807917	9	81
9	Oct	2567	0	0	0	0	0	0	0	0	0	0	1	0	7.850493	10	100
10	Nov	2631	0	0	0	0	0	0	0	0	0	1	0	0	7.875119	11	121
11	Dec	3102	0	0	1	0	0	0	0	0	0	0	0	0	8.039802	12	144

Data Manipulation

After formatting 'Quantity' values and sorting 'Dateofbill':

	Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	Final_Cost	Final_Sales	RtnMRP	Formulation	DrugName	Sub(
11037	Sale	12018076250	Specialisation5	Department1	2022-01-01	1	42	43	0.000	Form1	GLYCOPYRROLATE	INJECTIO
1863	Sale	12018044636	Specialisation21	Department1	2022-01-01	2	45	87	0.000	Form1	SODIUM CHLORIDE 0.9%	IV FLUII ELECTROLYT T
3751	Sale	12018081111	Specialisation11	Department2	2022-01-01	1	45	47	0.000	Form1	EPHEDRINE 30MG	INJECTIO
7004	Sale	12018071876	Specialisation3	Department1	2022-01-01	1	65	75	0.000	Form1	THYROXINE SODIUM 25MCG TAB	TABLETS CAPSUL
7021	Sale	12018076573	Specialisation4	Department1	2022-01-01	1	49	61	0.000	Form1		
12536	Sale	12018038526	Specialisation5	Department1	2022-01-01	3	60	139	0.000	Form1	DEXTROSE 10%W/V 500ML IVF	IV FLUII ELECTROLYT T
13070	Sale	12018072643	Specialisation11	Department1	2022-01-01	5	87	304	0.000	Form1		
13642	Sale	12018080109	Specialisation21	Department1	2022-01-01	5	68	360	0.000	Form1		

First Moment Business Decision

Measure of Central Tendency

Attribute	Mean	Median	Mode
Quantity	2.233864	1.000000e+00	1
Final_Cost	124.656919	5.400000e+01	42
Final_Sales	233.779594	8.600000e+01	0
RtnMRP	29.154758	0.000000e+00	0

- On average, the quantity of medicines sold per transaction is around **2.23 units.**
- Average cost of the medicines sold per transaction is around **$124.66.**
- Average sales revenue generated per transaction is around **$233.78.**
- Return value of a medicine, based on the manufacturer's retail price, is around **$29.15.**
- Median quantity is 1.0 shows that 50% of transactions involve purchasing just **1 unit** of a drug.
- Mode of the data is Patient_ID 12018071649 with final cost of **42$**.

Second Moment Business Decision

Measure of Dispersion

Attribute	Variance	Standard deviation
Quantity	26.382694	5.136409

- Variance of approximately 26.38 indicates that there is a notable amount of dispersion in the "Quantity" values. This suggests that the quantities of products sold across transactions show significant differences from the mean.
- The calculated standard deviation of around 5.14 further illustrates the spread of the "Quantity" data.

Third Moment Business Decision

Measure of asymmetry in distribution

Attribute	Skewness	Skew Type
Quantity	11.331675	Positively skewed
Final_Cost	34.528927	Positively skewed
Final_Sales	21.038080	Positively skewed
RtnMRP	15.784347	Positively skewed

Quantity

- 11.33 suggests that the distribution of the data is positively skewed.
- Positive skewness indicates that the tail of the distribution is extended to the right, with a concentration of lower values and a few very high values.
- This skewness indicate that there are many transactions with relatively lower quantities of medicines sold, but a few transactions involving significantly higher quantities.

Final Sales

- 21.04 suggests that the distribution of the "Final_Sales" data is positively skewed.
- This implies that there are many transactions with lower sales amounts and fewer transactions with higher sales amounts.
- The skewness could be due to the varying prices and sales quantities of different medicines.

Forth Moment Business Decision

Measure of peakedness - represents the overall spread in the data

Attribute	Kurtosis
Quantity	180.153858
Final_Cost	2025.845677
Final_Sales	948.581412
RtnMRP	403.524941

Quantity

- 179.85 suggests that the distribution of the data has high **positive kurtosis.**
- High positive kurtosis indicates that the distribution has heavy tails, more than what would be expected in a normal distribution.

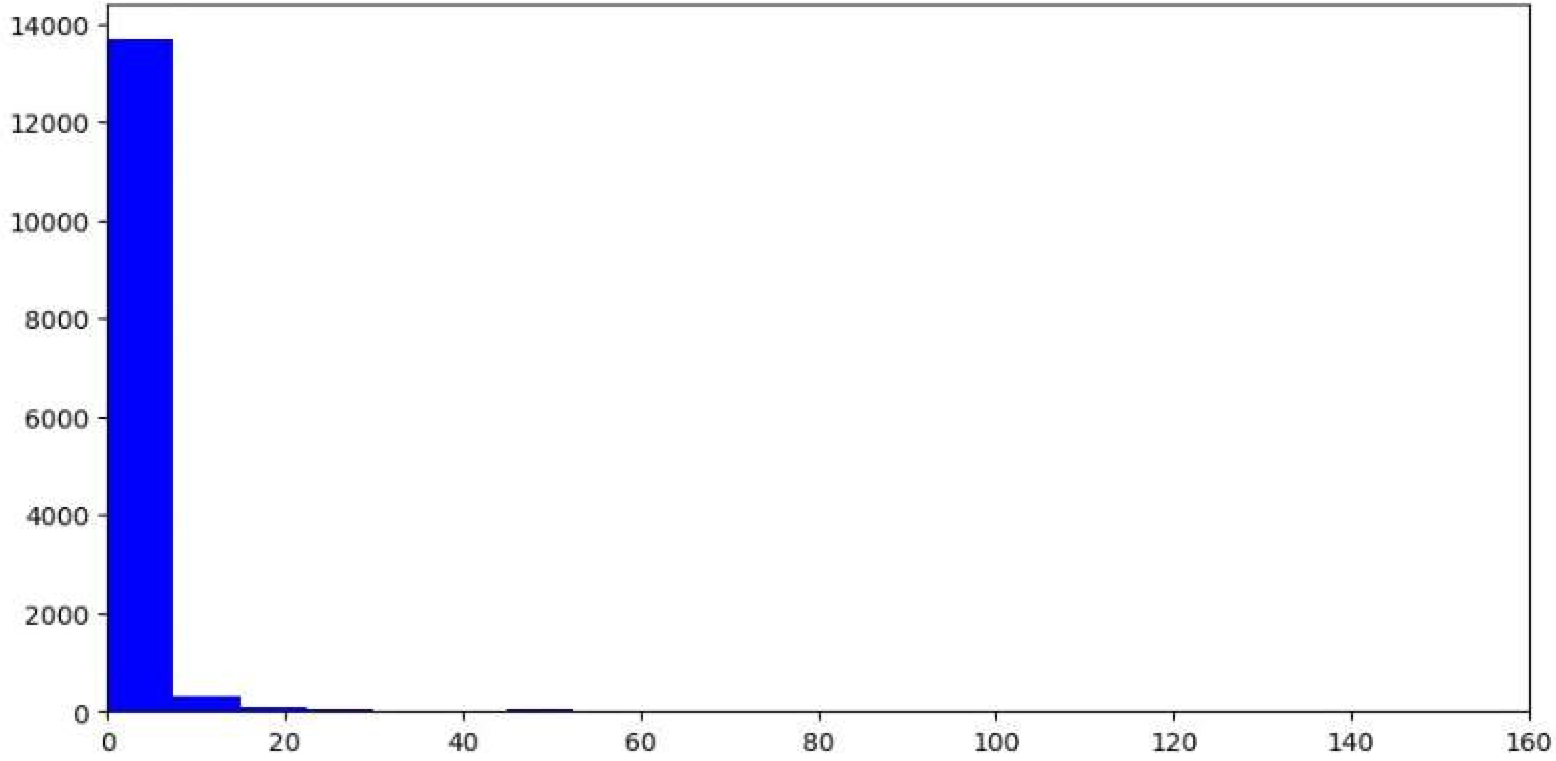
Final Sales

- 949.99 suggests very high **positive kurtosis** in the data.
- The high kurtosis could result from a few transactions with exceptionally high sales amounts

Exploratory Data Analysis [EDA]: Description

Histogram:

Quantity

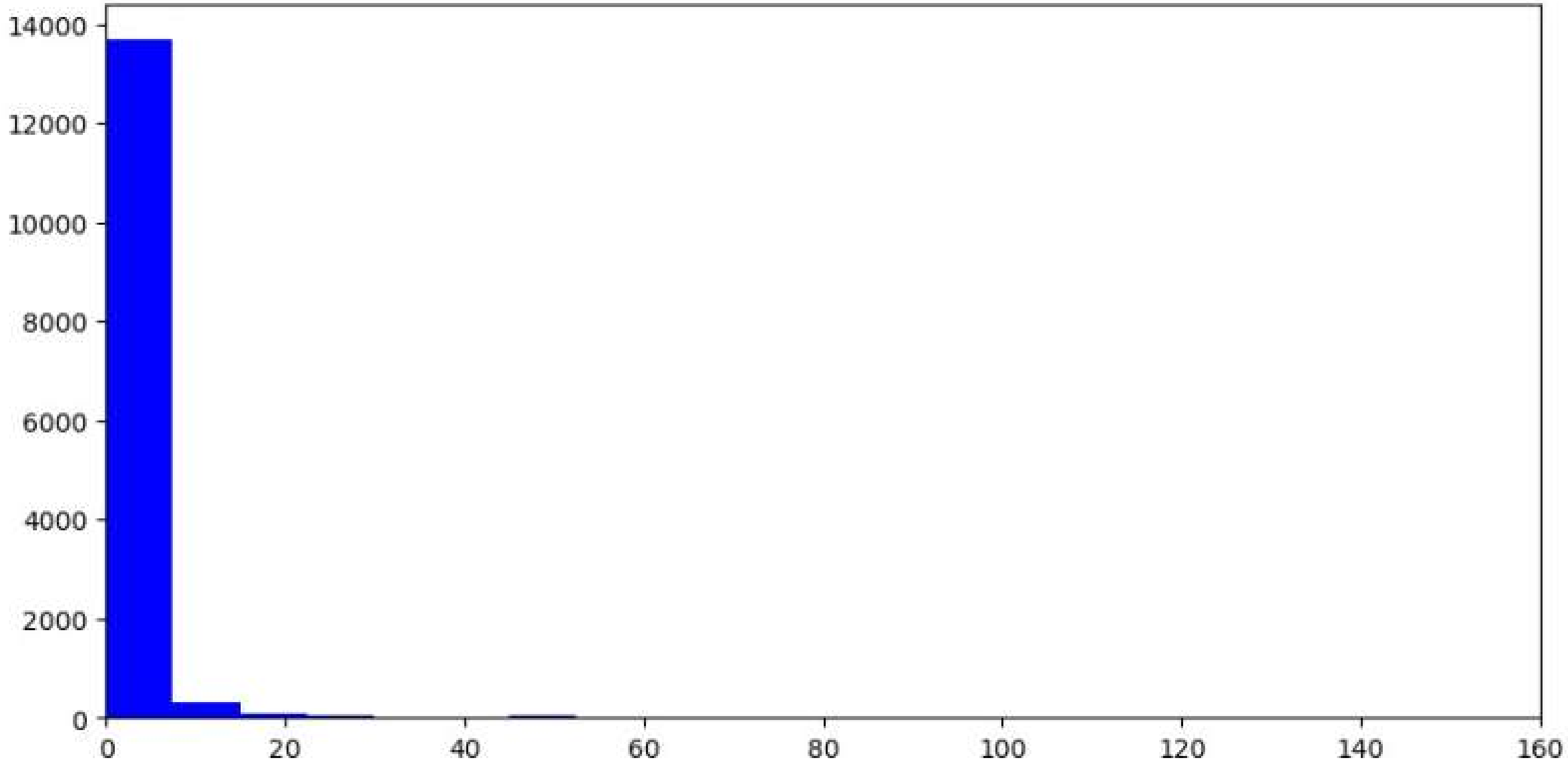


- Max. = 150
- Right-skewed
- Majority of transactions involve relatively lower quantities, but there are occasional instances with higher quantities.

Exploratory Data Analysis [EDA]: Description

Histogram:

Final Cost

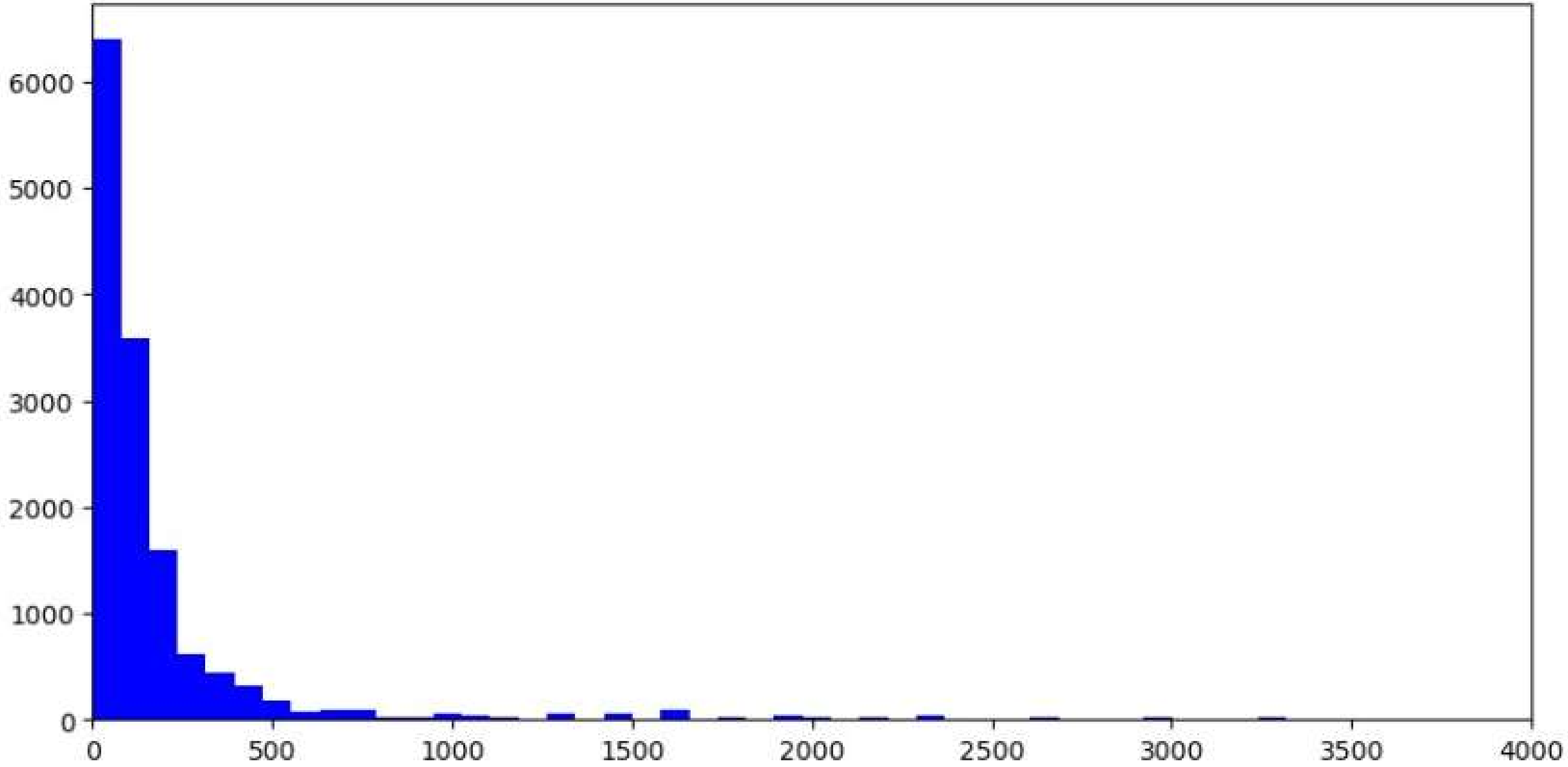


- Max. = 33178
- Right-skewed
- There are relatively few instances with very high final costs, contributing to the elongated tail on the right side.

Exploratory Data Analysis [EDA]: Description

Histogram:

Final Sales

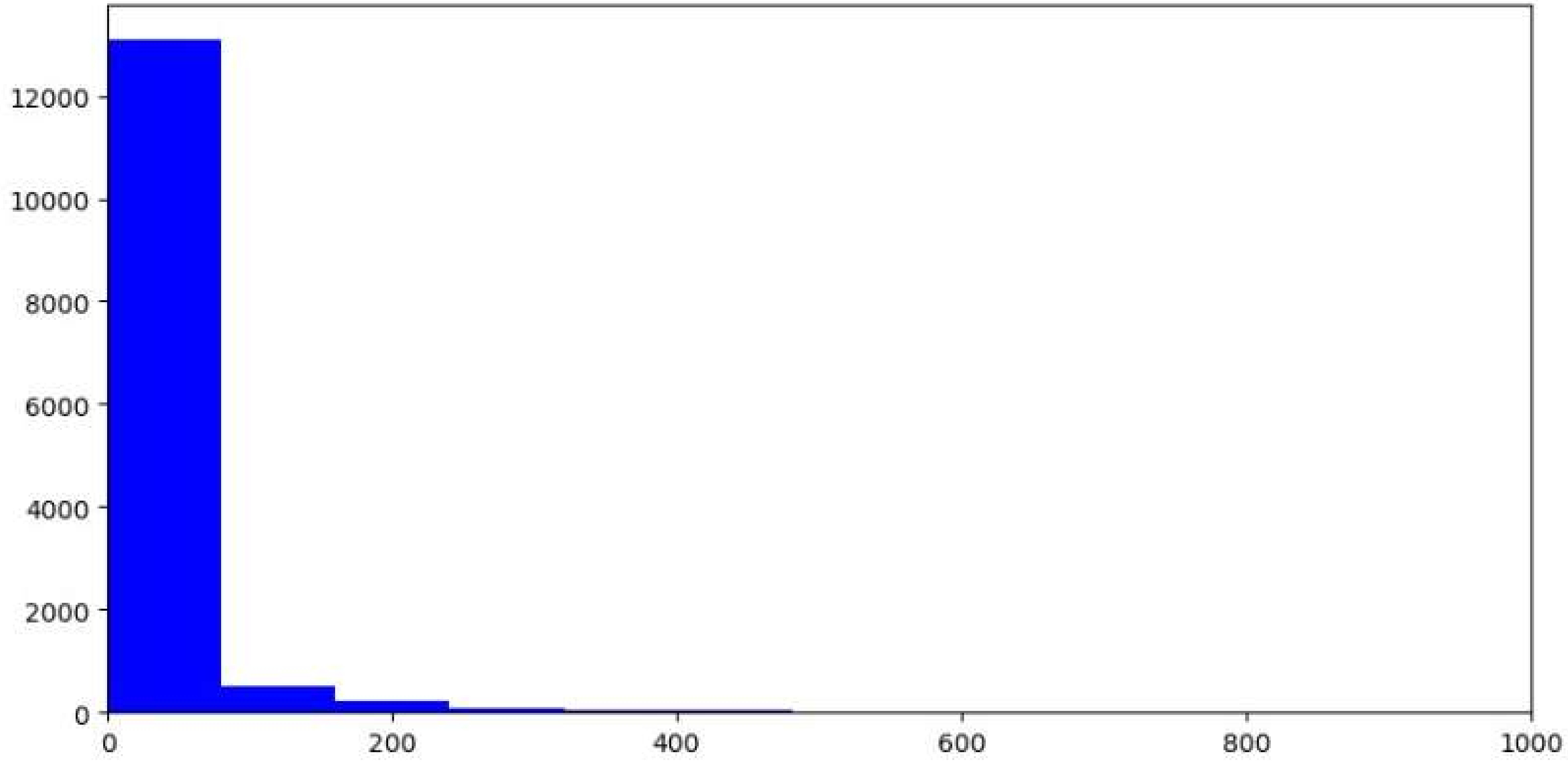


- Max. = 39490
- Right-skewed
- Most of transactions involve lower sales amounts, but there are a few instances with significantly higher sales amounts.

Exploratory Data Analysis [EDA]: Description

Histogram:

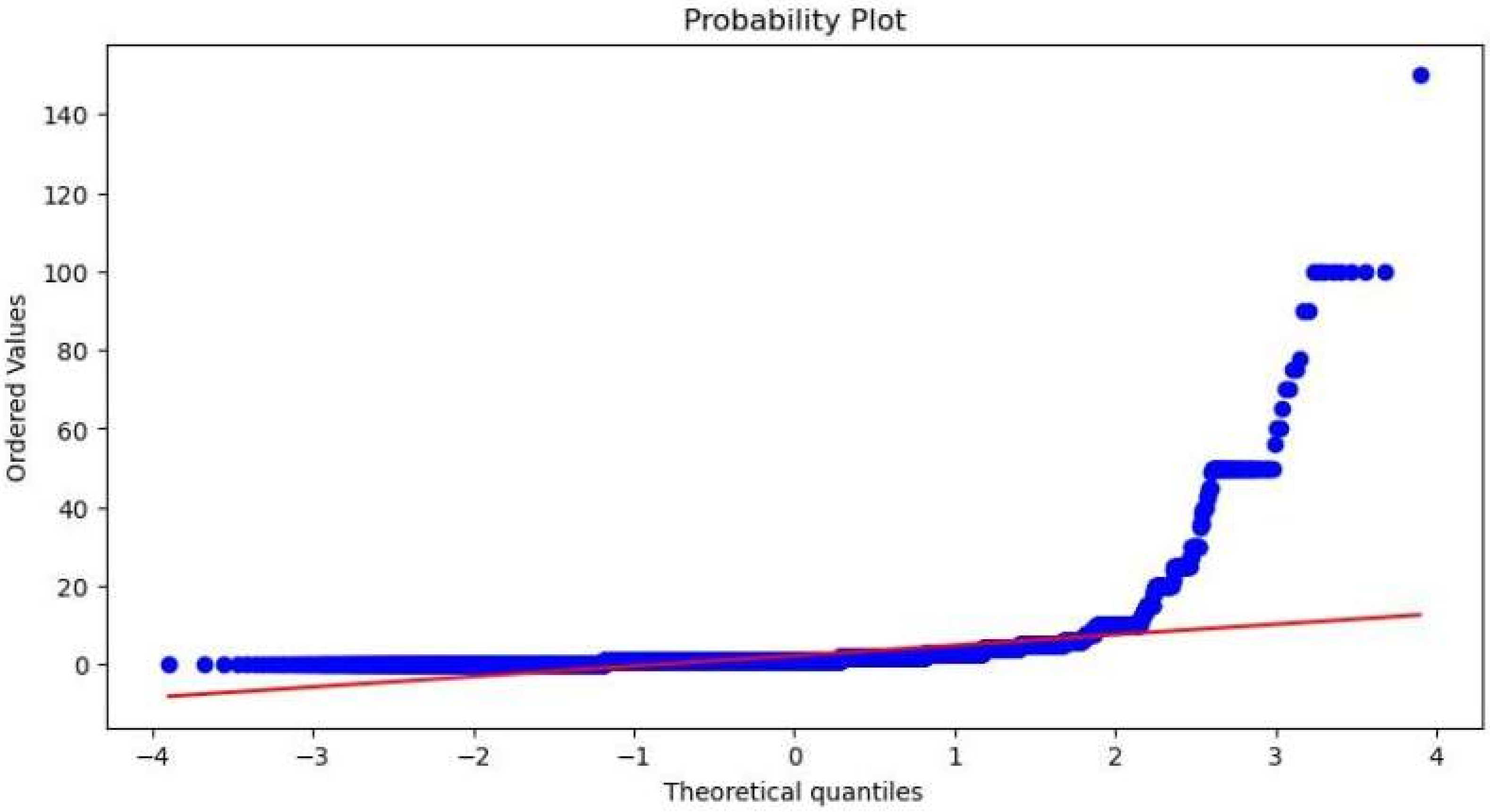
RtnMRP



- Max. = 8014.000
- Right-skewed
- Majority of return values are lower, but there are a few instances with significantly higher return values.

Data Distribution

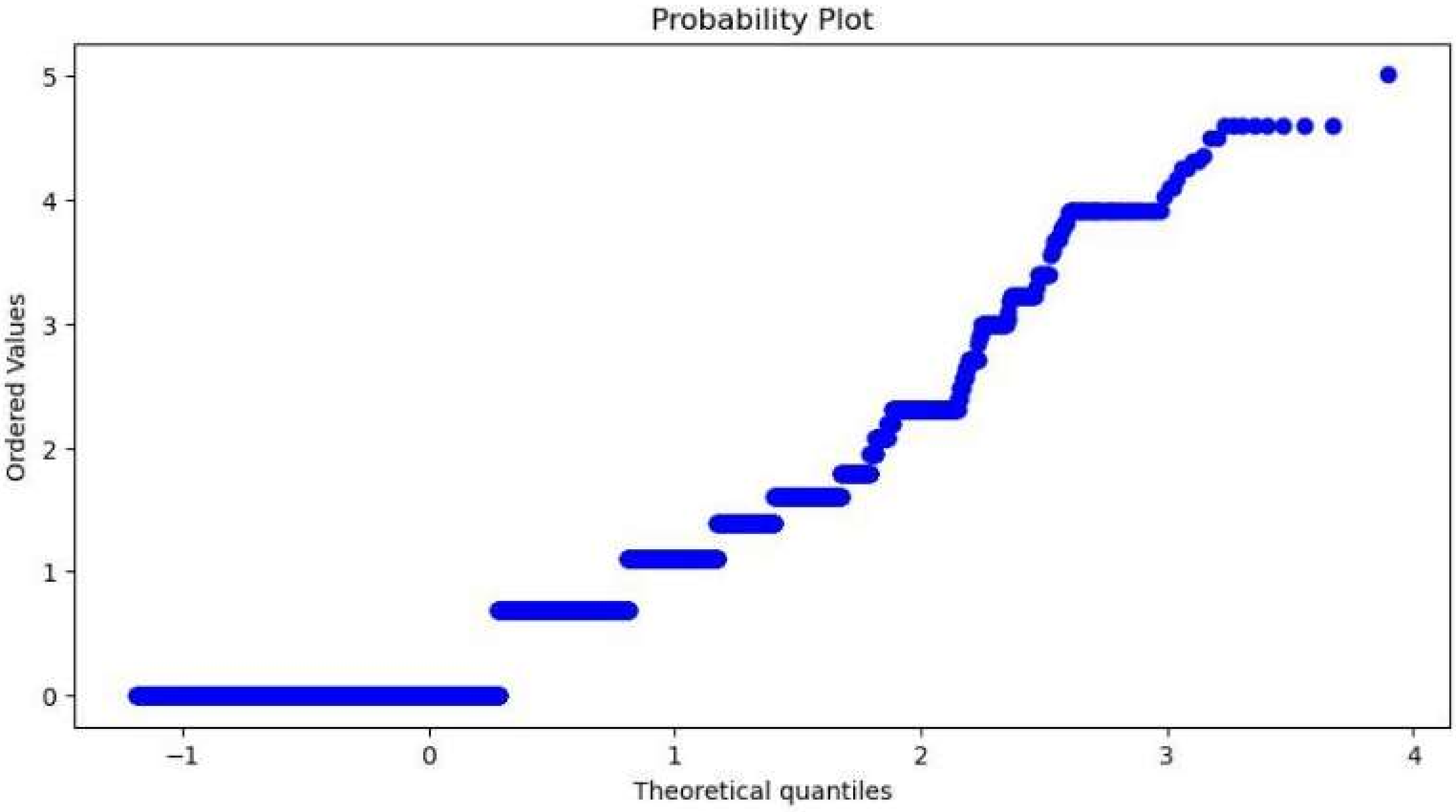
The Quantity data shows it is not normally distributed because the data is not falling within the straight line.



Normal Distribution

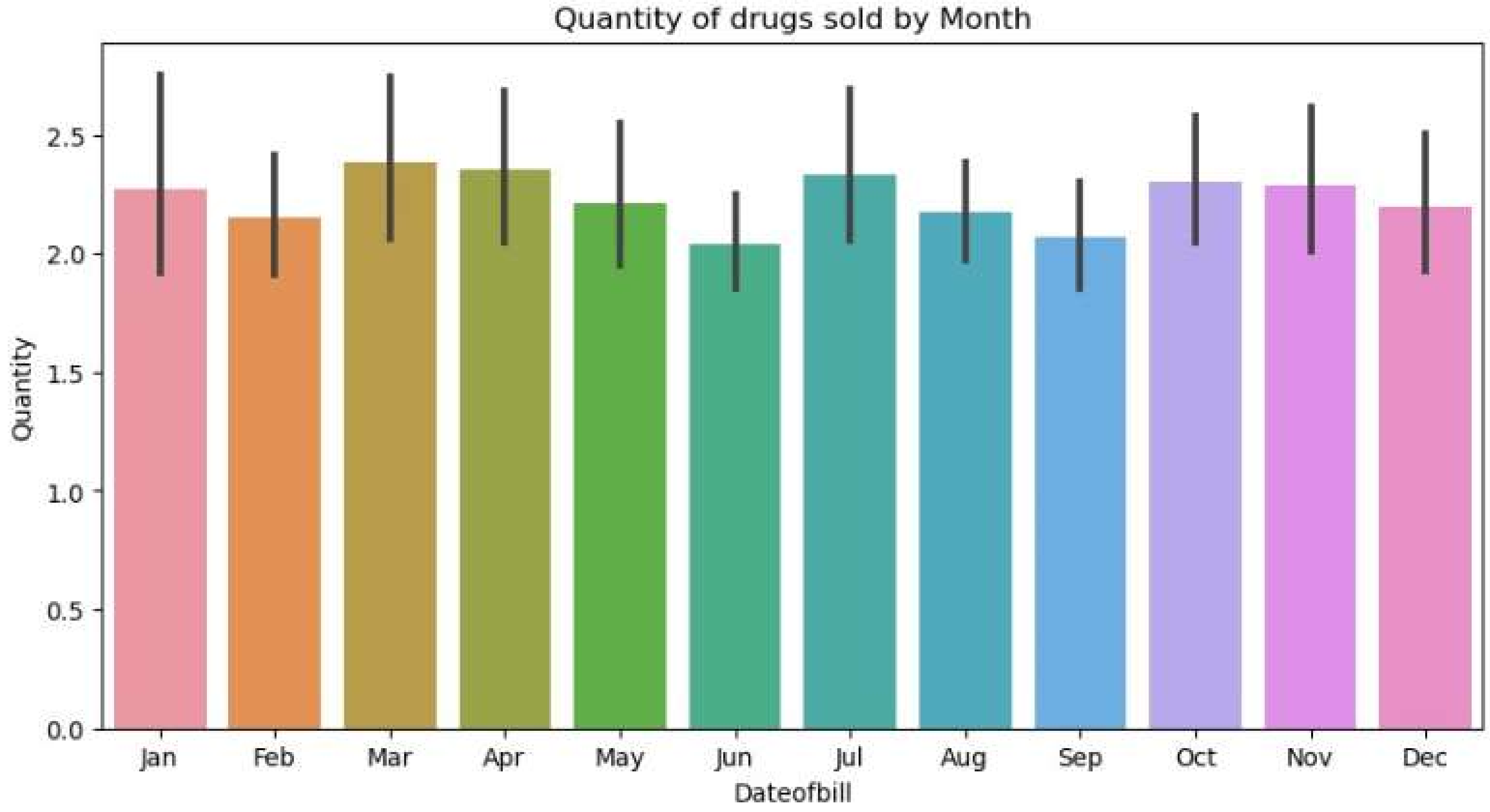
Data Transformation: Log Transformation

After log transformation, the data is normally distributed.



Data Visualization

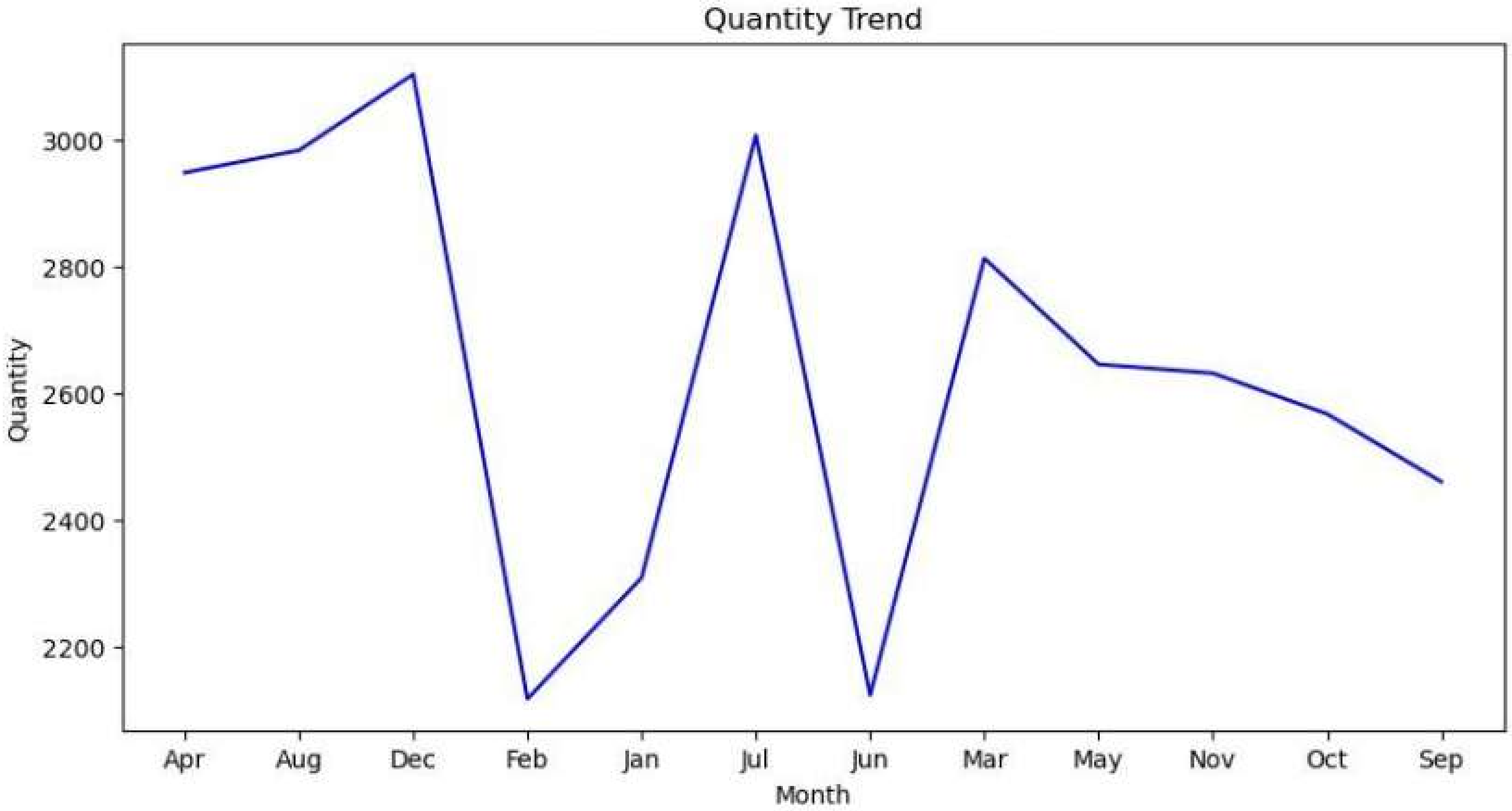
Bar Plot (Quantity of drugs sold by Month)



The month of March, April, July, and September has highest quantity of medicines sold and it is approximately same.

Data Visualization

Line Chart (Trend in Quantity)



December has the highest quantity of medicines sold while February and June is the lowest.

AutoEDA

D-Tale

14192 / 14	index	Typeofsales	Patient_ID	Specialisation	Dept	Dateofbill	Quantity	Final_Cost	Final_Sales	RtnMRP	Formulation
0	11037	Sale	12018076250	Specialisation5	Department1	Jan	1	42	43	0.000	Form1
1	1863	Sale	12018044636	Specialisation21	Department1	Jan	2	45	87	0.000	Form1
2	3751	Sale	12018081111	Specialisation11	Department2	Jan	1	45	47	0.000	Form1
3	7004	Sale	12018071876	Specialisation3	Department1	Jan	1	65	75	0.000	Form1
4	7021	Sale	12018076573	Specialisation4	Department1	Jan	1	49	61	0.000	Form1
5	12536	Sale	12018038526	Specialisation5	Department1	Jan	3	60	139	0.000	Form1
6	13070	Sale	12018072643	Specialisation11	Department1	Jan	5	87	304	0.000	Form1
7	13642	Sale	12018080109	Specialisation21	Department1	Jan	5	68	360	0.000	Form1
8	9712	Sale	12018080590	Specialisation7	Department1	Jan	2	51	100	0.000	Form1
9	9762	Sale	12018081582	Specialisation3	Department1	Jan	3	66	449	0.000	Form1
10	9902	Sale	12018080633	Specialisation14	Department1	Jan	1	78	126	0.000	Form2
11	14030	Sale	12018064578	Specialisation10	Department2	Jan	3	121	308	0.000	Form1
12	8004	Sale	12018071419	Specialisation4	Department2	Jan	1	52	56	0.000	Form2
13	10886	Return	12018081506	Specialisation2	Department1	Jan	0	132	0	336.760	Form1
14	8855	Sale	12018050532	Specialisation4	Department1	Jan	1	41	41	0.000	Form1
15	10876	Sale	12018079227	Specialisation21	Department1	Jan	2	42	83	0.000	Form1
16	10871	Sale	12018072994	Specialisation7	Department1	Jan	1	960	1488	0.000	Form1

Model Building

REGRESSION MODEL

1. Select Models:

- 2 types of model being developed in this project which is the **Random Forest Regression** Model and **Linear Regression** Model.

2. Train and Test Models:

- Split historical data into training and testing sets.
- The training set is used to train the models, while the testing set is used to assess their performance.
- Each selected model is trained on the training set using appropriate parameters.

Model Building

REGRESSION MODEL

3. Generate Forecasts:

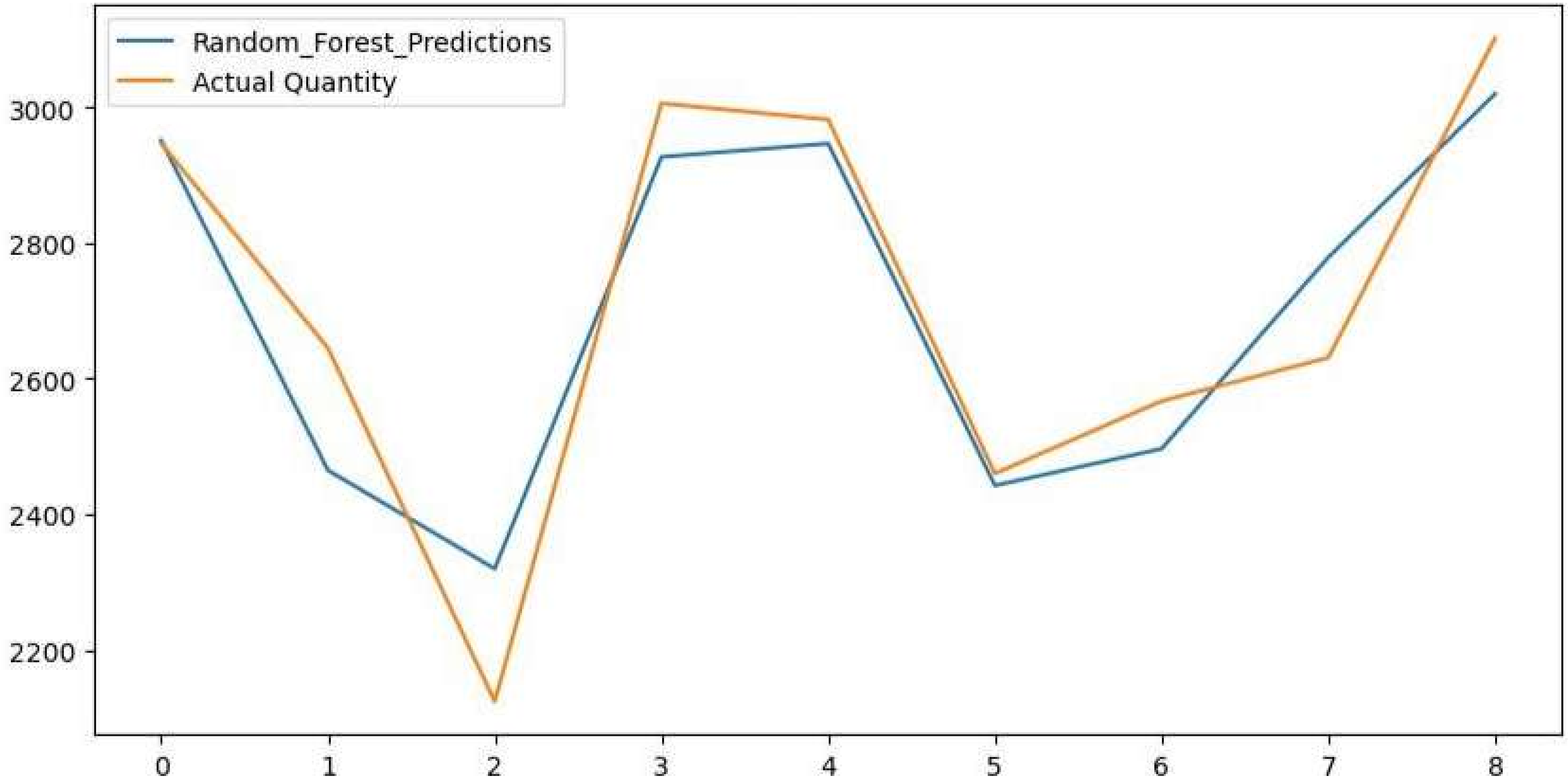
- Trained models is used to generate forecasts for the required period.
- For each forecasted point, the predicted value is compared with the actual value from the testing set.

4. Calculation of Evaluation Metrics:

- Common evaluation metrics are calculated to assess the accuracy of each model.
- Metrics for time series forecasting include:

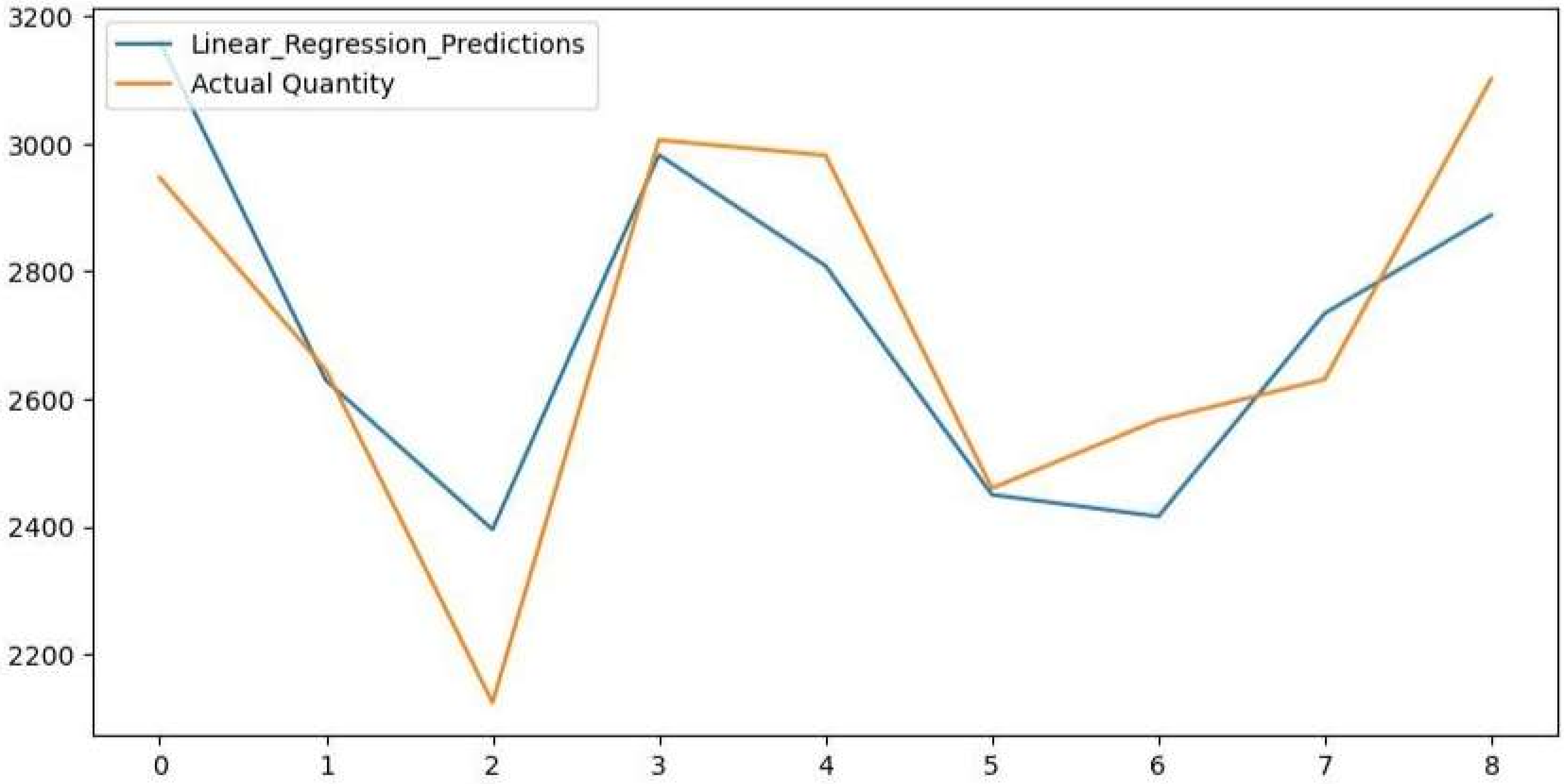
  a) Mean Squared Error (MSE)
  b) Root Mean Squared Error (RMSE)

Model Building – Random Forest Regression



The MSE of **111.61** suggests that, on average, the squared differences between the actual and predicted values of quantity of drugs sold are **relatively low**. This indicates a relatively **good fit** of the Random Forest Model to the data.

Model Building – Linear Regression



The MSE **159.58** suggests that the average squared differences between the actual and predicted values are somewhat **higher** compared to the Random Forest Model. This indicates **a higher level of prediction error** compared to the Random Forest Model.

Model Accuracy Comparison

From the plot of the forecasted values against the actual values, we can visually assess how closely they align.

To compare the forecasted values, Mean Squared Error (MSE) is calculated

Mean Squared Error (MSE):

1. Mean Squared Error for Random Forest Model is 111.60992195041523
2. Mean Squared Error for Linear Regression Model is 159.57892788366377

Best Model – Random Regression Model

1. Considering the lower MSE value of the Random Forest Model **(111.61)** compared to the Linear Regression Model **(159.58),** the **Random Forest Model** appears to be the **better performer** in terms of prediction accuracy.

2. A **lower MSE** indicates that the Random Forest Model's predictions are, on average, closer to the actual values, resulting in better overall model performance.

3. The Random Forest Model is capable of **capturing complex non-linear relationships** in the data, which can be important when dealing with diverse and intricate patterns in pharmaceutical sales and demand.

4. It is **less sensitive to outliers** compared to Linear Regression, potentially resulting in more reliable predictions.

Model Deployment - Strategy

1. Test on New Data:

- After selecting the best model, new data should be tested for its performance on a separate "test" dataset that was not used during model development or validation. This provides a final validation of its accuracy.

2. Continuous Monitoring and Refinement:

- Even after selecting a model, continuously monitor its performance as new data becomes available.
- Update and refine the model as needed to maintain accuracy over time.
- Regularly update the model with new data to ensure it remains relevant and effective.

3. Feedback and Iteration:

- Collect feedback from patients to identify areas for improvement.
- This feedback is used to iterate and refine the deployment strategy and model over time.

4. Documentation:

- Documentation on interpretation of the results, and how to troubleshoot any issues that may arise.

Screen shot of output

Describe Data

```
pharma_data.describe()
```

✓ 0.0s

	Quantity
count	14192.000000
mean	2.233864
std	5.136409
min	0.000000
25%	1.000000
50%	1.000000
75%	2.000000
max	150.000000

Measure of Central Tendancy

```
# Mean
print(pharma_data.mean())
```

✓ 0.4s

```
Patient_ID            inf
Quantity         2.233864
Final_Cost     124.656919
Final_Sales    233.779594
RtnMRP          29.154758
dtype: float64
```

```
# Median
print(pharma_data.median())
```

✓ 0.0s

```
Patient_ID     1.201809e+10
Quantity       1.000000e+00
Final_Cost     5.400000e+01
Final_Sales    8.600000e+01
RtnMRP         0.000000e+00
dtype: float64
```

```
# Mode
pharma_data.mode()
```

✓ 0.0s

Typeofsales	Patient_ID	Specialisation	D
Sale	12018071649	Specialisation4	Departm

Second Moment Business Decision

Measure of Dispersion

```
# Variance
pharma_data.var()
```

✓ 0.0s

```
Quantity    26.382694
dtype: float64
```

Screen shot of output

```
# Standard Deviation
print(pharma_data.std())
```

✓ 0.1s

```
Dateofbill    104 days 12:07:15.303353232
Quantity                         5.136409
dtype: object
```

Third Moment Business Decision

Skewness

```
pharma_data.skew()
```

✓ 0.0s

```
Patient_ID     -1.365399
Quantity       11.331675
Final_Cost     34.528927
Final_Sales    21.038080
RtnMRP         15.784347
dtype: float64
```

```
# Pivot the DataFrame
data_pivoted = pharma_data.pivot_table(index="SubCat", columns="Dateofbill", values="Quantity")

# Show the result
data_pivoted.head()
```

✓ 0.0s

Dateofbill	Apr	Aug	Dec	Feb	Jan	Jul	Jun	Mar	May	Nov	Oct	Sep
SubCat												
	6.492308	4.005988	3.714286	6.037500	4.612245	6.158228	4.285714	6.284553	6.483871	3.481707	4.690909	4.879195
DROPS	0.888889	0.666667	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	0.666667	1.000000	1.000000	0.857143
INHALERS & RESPULES	2.360000	2.977778	2.722222	2.935484	3.666667	3.186047	3.266667	3.250000	4.052632	2.954545	3.088889	3.068182
INJECTIONS	1.940663	2.035889	2.046899	1.920177	1.849138	1.849153	1.769397	1.931947	1.621569	2.207865	1.972710	1.824532
IV FLUIDS, ELECTROLYTES, TPN	1.869231	1.686833	1.600858	1.805430	1.927083	1.839844	2.050228	1.732218	1.713208	1.485207	1.622951	1.505376

Fourth Moment Business Decision

Kurtosis

```
pharma_data.kurt()
```

✓ 0.1s

```
Patient_ID        1.620875
Quantity        179.847805
Final_Cost     2026.390946
Final_Sales     949.990222
RtnMRP          402.820195
dtype: float64
```

Screen shot of output

Data Transformation : Log Transformation

```
# Tranform the data to a normal distribution
stats.probplot(np.log(pharma_data.Quantity),dist="norm",plot=pylab)
```

✓ 0.5s

```
((array([-3.89628607, -3.67580637, -3.55497144, ...,  3.55497144,
          3.67580637,  3.89628607]),
  array([       -inf,        -inf,        -inf, ..., 4.60517019, 4.60517019,
         5.01063529])),
 (nan, nan, nan))
```

One-Hot Encoding

```
df_grouped = pharma_data[['Dateofbill','Quantity']]
```

✓ 0.0s

```
# Group by Quantity and Month
df_grouped = df_grouped.groupby('Dateofbill').sum()

# Show result
df_grouped.head(10)
df_grouped = df_grouped.reset_index()
df_grouped
```

✓ 0.0s

	Dateofbill	Quantity
0	Apr	2947
1	Aug	2982
2	Dec	3102
3	Feb	2118

Screen shot of output

```
from sklearn.ensemble import RandomForestRegressor
model=RandomForestRegressor(n_estimators=100,max_features=3, random_state=1)
```

✓ 0.0s

```
import numpy as np
x1,x2,x3,y=df['Quantity_LastMonth'],df['Quantity_2Monthsback'],df['Quantity_3Monthsback'],df['Quantity']
x1,x2,x3,y=np.array(x1),np.array(x2),np.array(x3),np.array(y)
x1,x2,x3,y=x1.reshape(-1,1),x2.reshape(-1,1),x3.reshape(-1,1),y.reshape(-1,1)
final_x=np.concatenate((x1,x2,x3),axis=1)
print(final_x)
```

✓ 0.0s

```
[[2812. 2118. 2309.]
 [2947. 2812. 2118.]
 [2645. 2947. 2812.]
 [2124. 2645. 2947.]
 [3006. 2124. 2645.]
 [2982. 3006. 2124.]
 [2460. 2982. 3006.]
 [2567. 2460. 2982.]
 [2631. 2567. 2460.]]
```

```
X_train,X_test,y_train,y_test=final_x[:],final_x[-10:],y[:],y[-10:]
```

✓ 0.0s

Screen shot of output

```
rmse_rf=sqrt(mean_squared_error(pred,y_test))
rmse_lr=sqrt(mean_squared_error(lin_pred,y_test))
```

386] ✓ 0.0s

```
print('Mean Squared Error for Random Forest Model is:',rmse_rf)
print('Mean Squared Error for Linear Regression Model is:',rmse_lr)
```

387] ✓ 0.0s

```
Mean Squared Error for Random Forest Model is: 111.60992195041523
Mean Squared Error for Linear Regression Model is: 159.57892788366377
```

Challenges

- Difficulty to predict stock with **return quantity** when model implanted on quantity only.
- Restriction on **patients' privacy** info with different nature of health condition leads to another drugs in present environment and it automatically leads drug shortage.
- Limited knowledge in **new drugs development.**
- Ensuring the forecasting model aligns with **pharmaceutical regulations** and patient privacy laws.
- Data integration with existing **inventory systems is complex** and **time-consuming.**
- **Insufficient historical d**ata for certain drugs or specific patient segments may restrict the accuracy of forecasting models.
- External factors of **sudden market shifts**, or unexpected events (e.g., **pandemics**) could disrupt the accuracy of forecasting models and inventory plans.

Future Scopes

Enhanced Forecasting Model:

- Continuously improve and refine the pharmaceutical forecasting model by incorporating more advanced machine learning techniques, considering additional factors like seasonal trends, public health events, and external influences on medicine demand.

Supply Chain Optimization:

- Collaborate with suppliers and distributors to optimize the entire supply chain. This might involve streamlining delivery routes, and reducing lead times

Machine Learning for Bounce Rate Reduction:

- Utilize machine learning to predict and mitigate factors causing high bounce rates in pharmacies, offering insights into optimizing store layouts and reduce return products.