Data Science Project Report

Sales Data Analysis and Forecasting/Prediction

Mukand Krishna 20K-0409 **12/31/23** **Data Science**

Section 1: Introduction:

This Sales dataset comprises sales information, including product details, quantities, and financial aspects. The goal is to provide a comprehensive analysis, uncovering patterns and trends that can inform strategic business decisions.

Data Overview:

The dataset encompasses **290,514 entries and 12 columns**.

<u>Key columns include:</u>

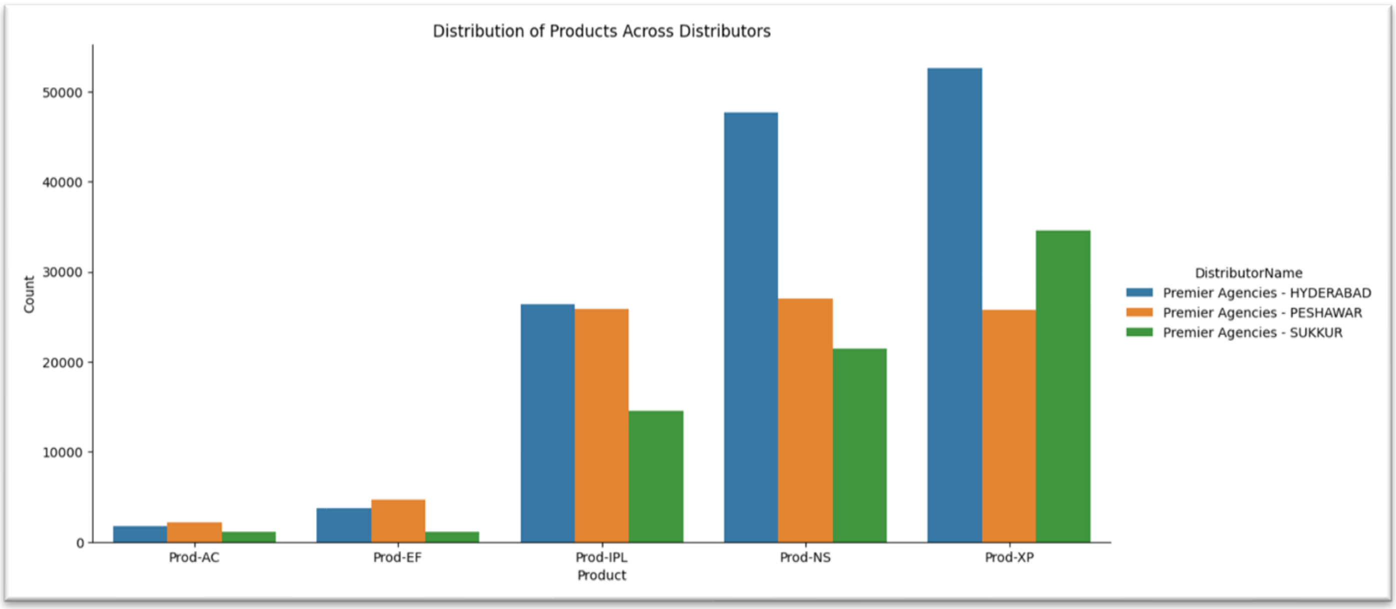
- DistributorCode & DistributorName: Unique identifier and name of the distributor.
- ClientCode & ClientName: Unique identifier and name of the client or store.
- BrickName: A detailed location or category identifier.
- Product & SKU: Product details, with SKU offering a granular identifier.
- InvoiceDate: Date of the transaction.
- Units, Bonus, Discount: Quantities and financial aspects of the transaction.
- ValueNp (Net Profit): Total revenue from sales.

```
df_order.head()
```

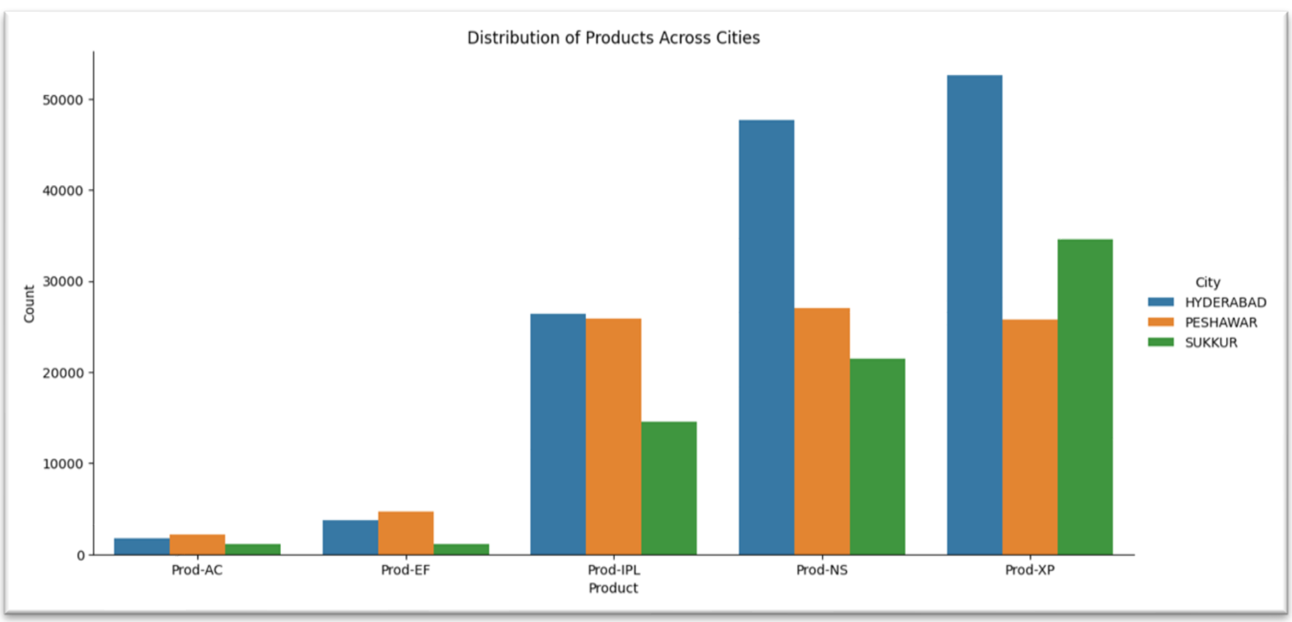
	DistributorCode	DistributorName	ClientCode	ClientName	BrickName	Product	SKU	InvoiceDate	Units	Bonus	Discount	ValueNp
0	2715	Premier Agencies - HYDERABAD	3129643	STAR MEDICAL STORE/HIRABAD/HIRABAD-HIRABAD-AZA...	HIRABAD-HIRABAD-AZAD MEEZAN MASJID HIRABAD	Prod-NS	Prod-NS-Tab	4/1/2017	1	0	0.0	97.75
1	2715	Premier Agencies - HYDERABAD	1301969	BHITAI MEDICAL STORE/BIHAR COLONY HOSRI/HOSRI-...	HOSRI-HOSRI-BIHAR COLONY HOSRI	Prod-NS	Prod-NS-Tab	4/1/2017	3	0	0.0	293.25
2	2715	Premier Agencies - HYDERABAD	1301971	SARFARAZ MEDICAL STORE/HOSRI PUL PAR/HOSRI-HOS...	HOSRI-HOSRI-HOSRI PULL PAR	Prod-NS	Prod-NS-Tab	4/1/2017	2	0	0.0	195.50
3	2715	Premier Agencies - HYDERABAD	1466465	MEHRAN MEDICAL STORE/HOSRI PUL PAR/HOSRI-HOSRI...	HOSRI-HOSRI-HOSRI PULL PAR	Prod-NS	Prod-NS-Tab	4/1/2017	2	0	0.0	195.50
4	2715	Premier Agencies - HYDERABAD	1301976	RANA MUKESH MEDICAL STORE/HOSRI PUL PAR/HOSRI-...	HOSRI-HOSRI-HOSRI PULL PAR	Prod-NS	Prod-NS-Tab	4/1/2017	1	0	0.0	97.75

Section 2: Exploratory Data Analysis

1. Product Wise Analysis:



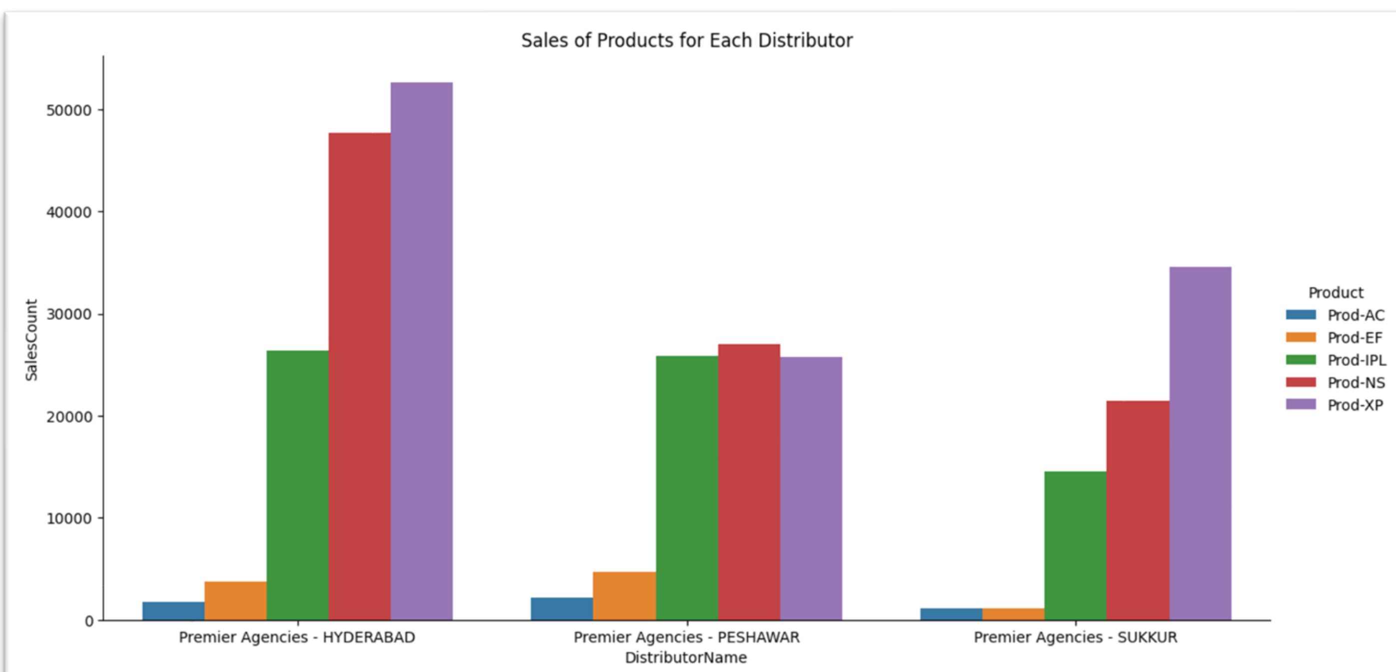
- Products Distribution in Cities:



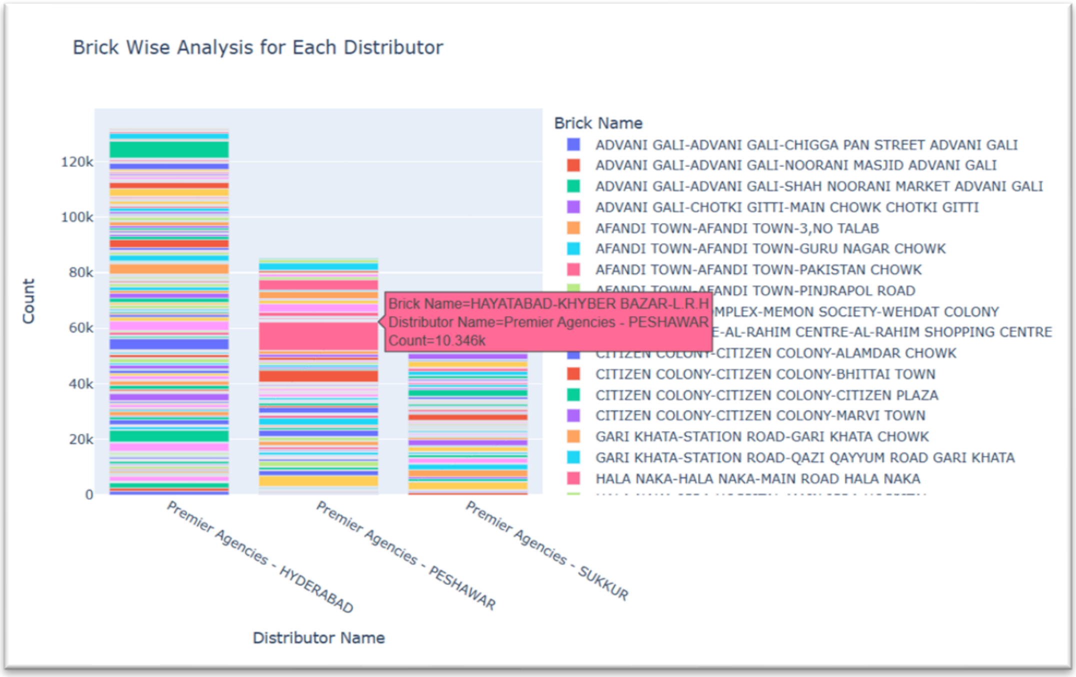
- Products Distribution in Bricks:

2. Distributor Wise Analysis:

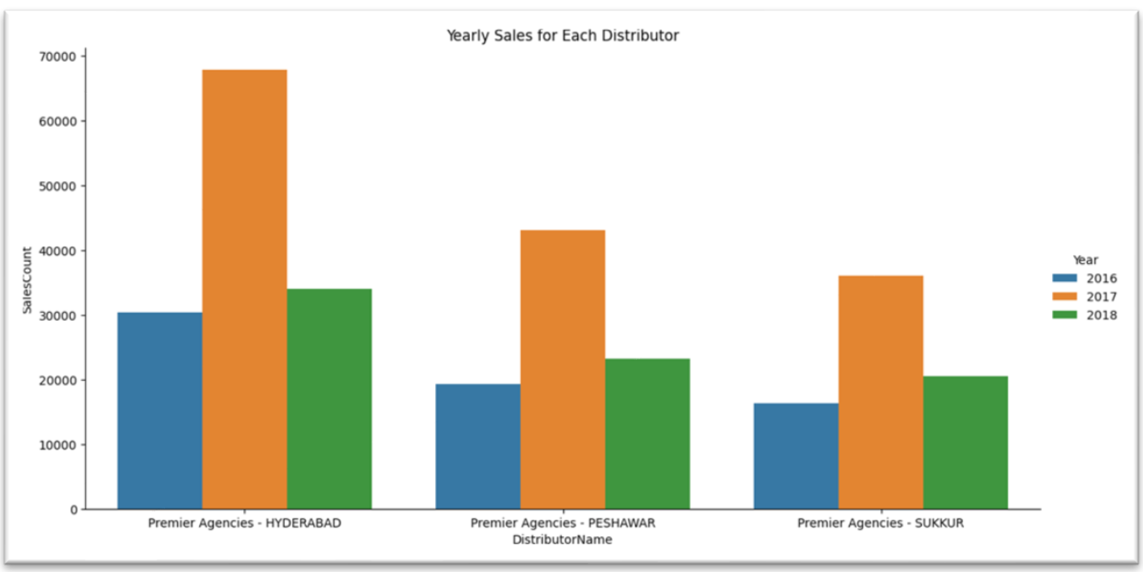
- Product Sales for Each Distributor:



- Brick Wise Analysis for Each Distributor:



- Yearly Sales for Each Distributor:

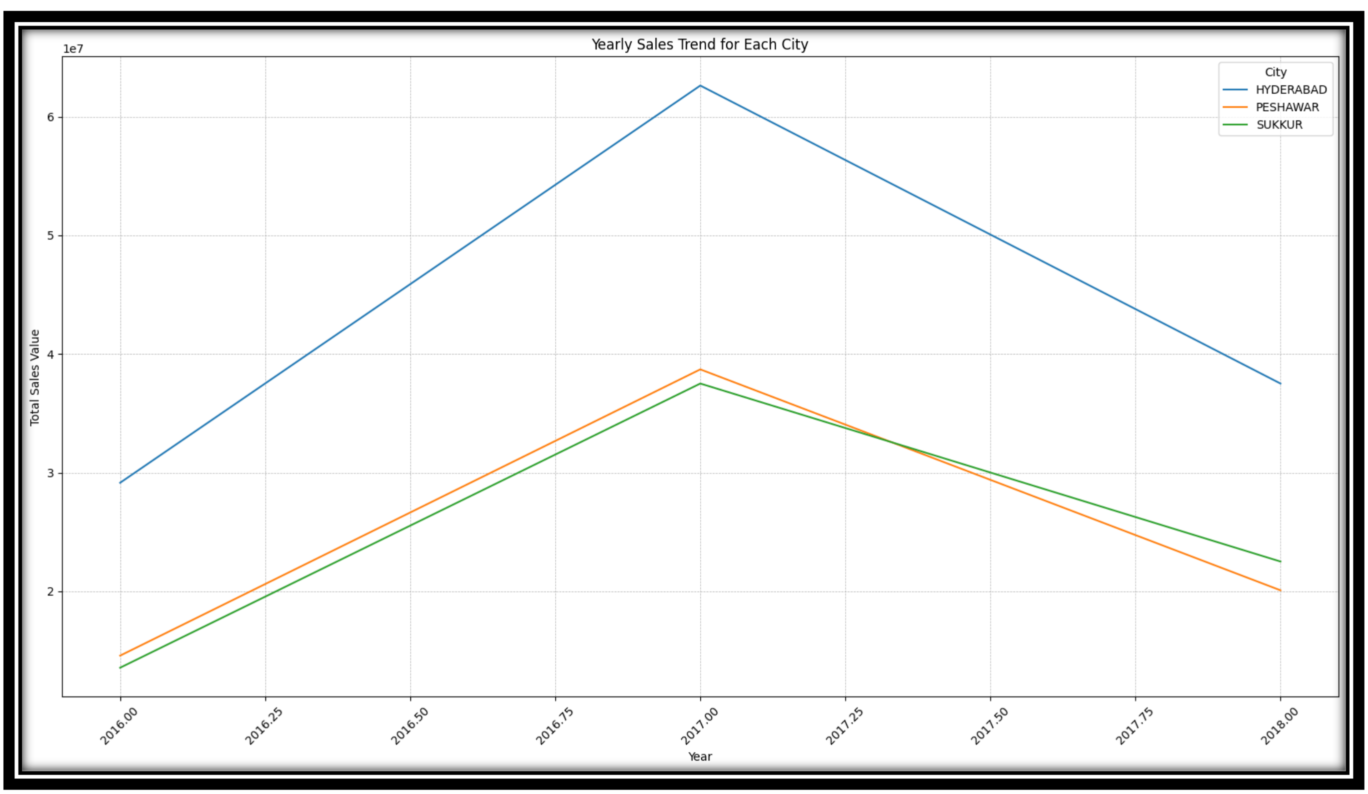


<u>'Premier Agencies HYDERABAD' had a substantial increase in sales from 2016 to 2017.</u>

3. Brick Analysis:

- Top 10 Bricks by Sales:

4. Yearly Sales Trends for Each City:



Section 3: Data Preprocessing:

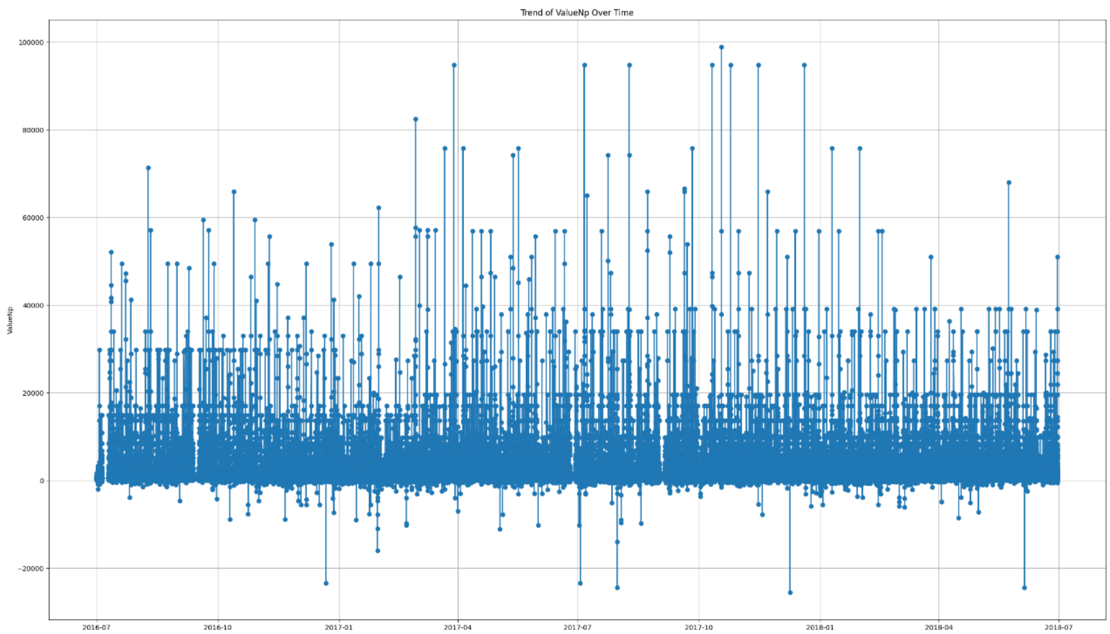
1. Handling DateTime:

Began by converting the 'InvoiceDate' column to a datetime format to facilitate temporal analysis.

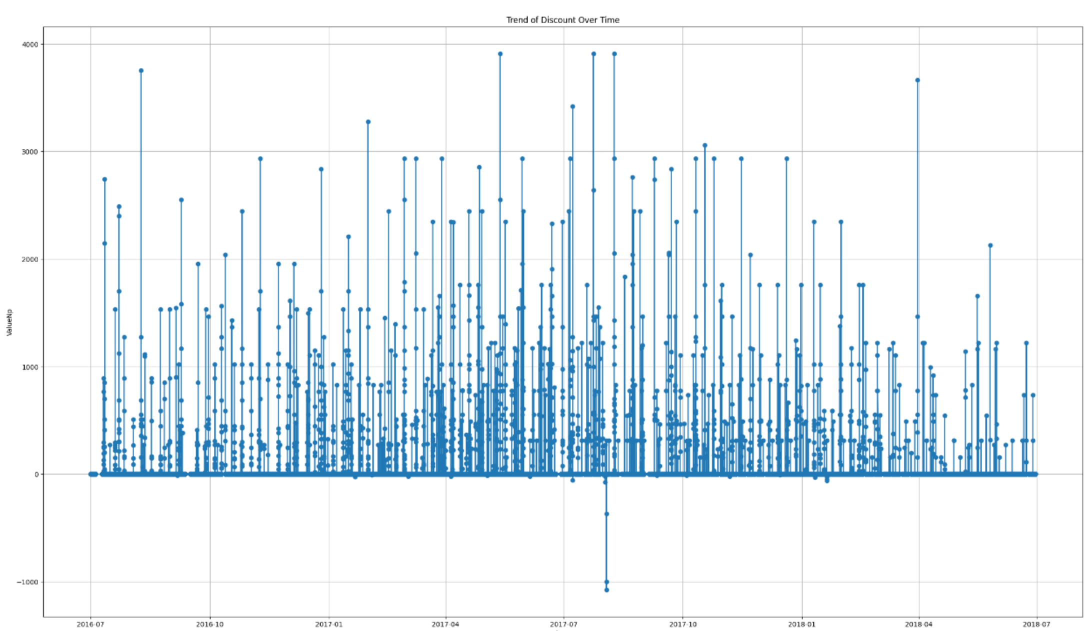
2. Time Series Analysis:

Plotted the trends of 'ValueNp,' 'Discount,' and 'Units' over time to identify any patterns or anomalies.

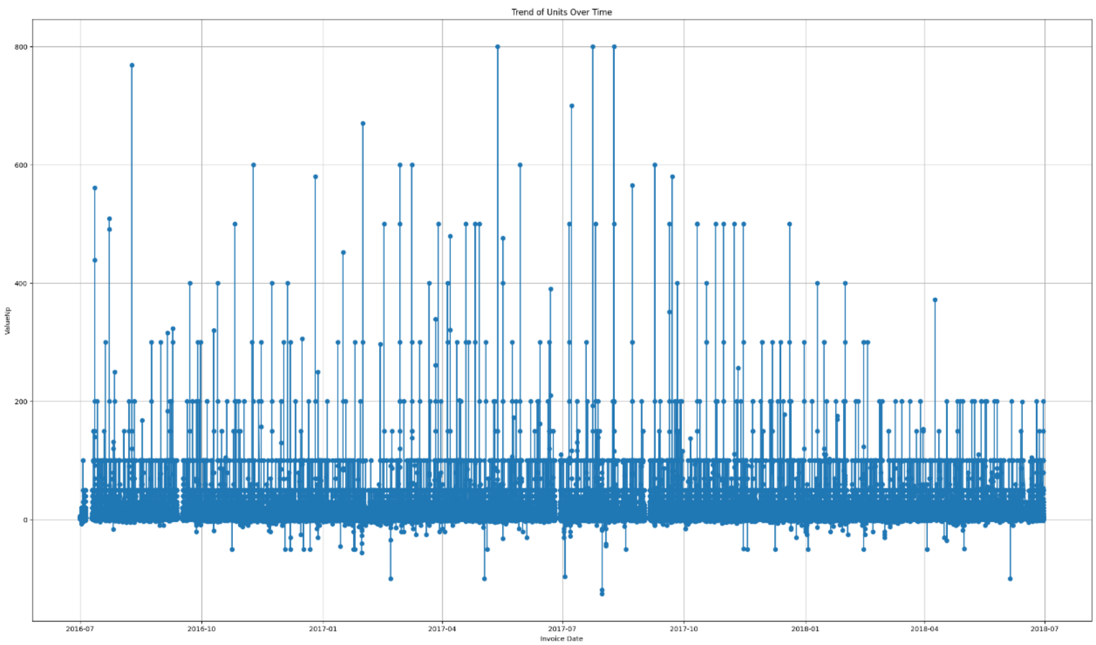
- Trend of ValueNp Over Time reveals overall sales trends.



- Trend of Discount Over Time showcases variations in discount patterns.

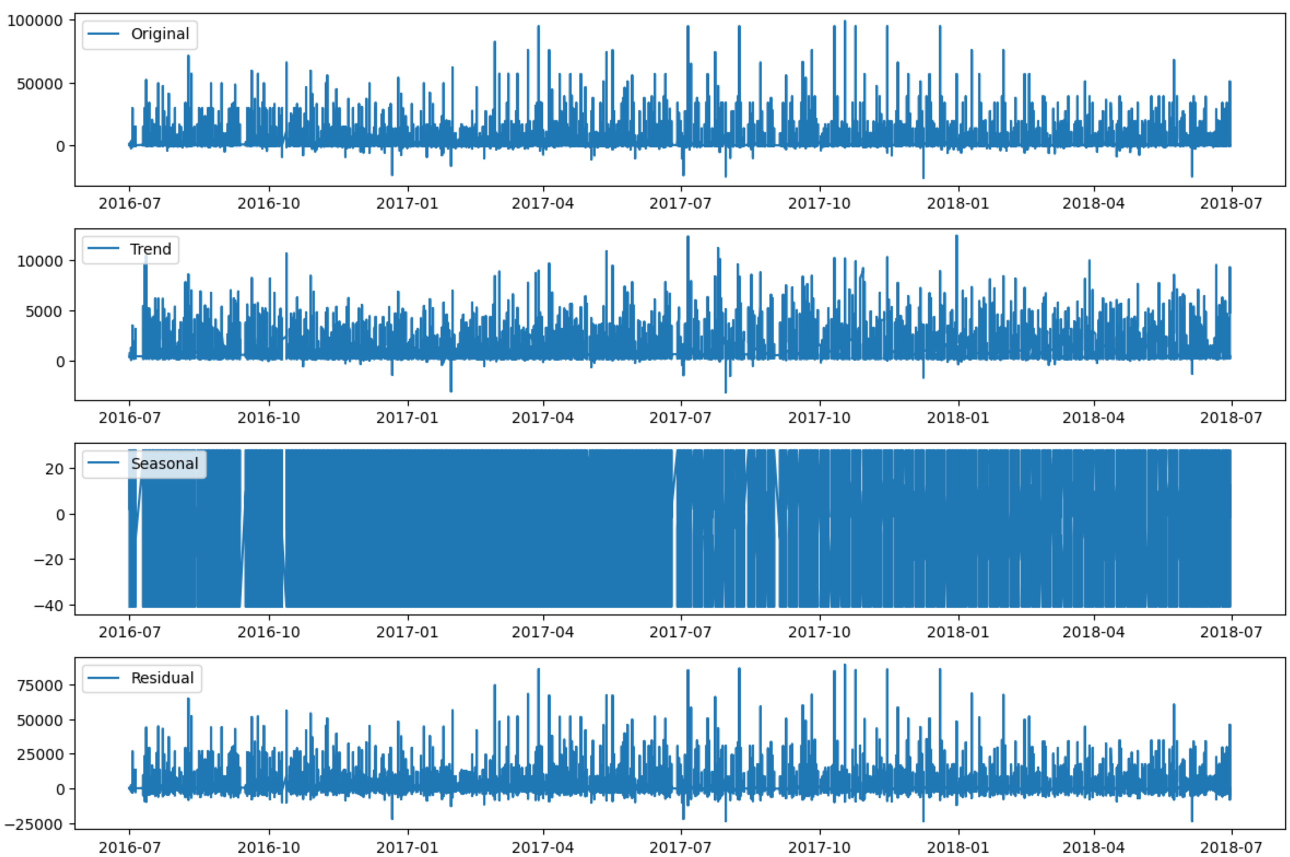


- Trend of Units Over Time illustrates fluctuations in product units sold.



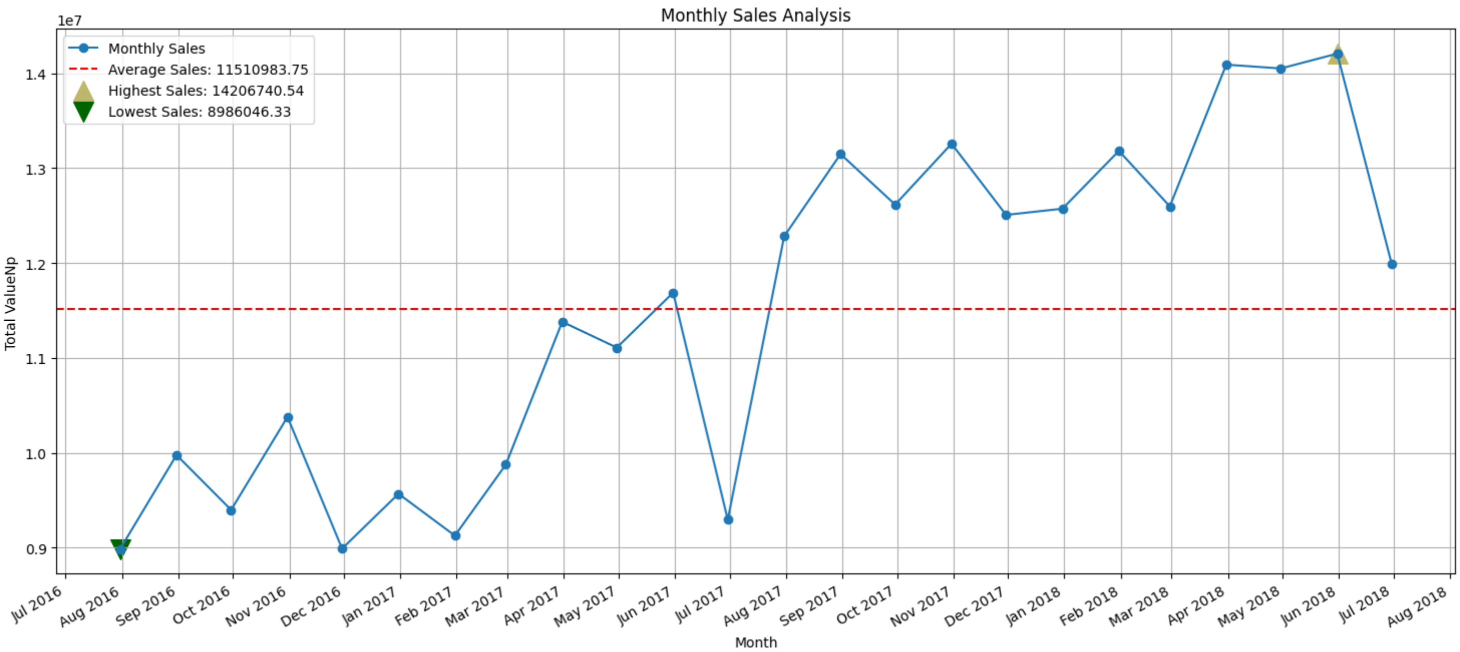
3. **Seasonal Decomposition**:

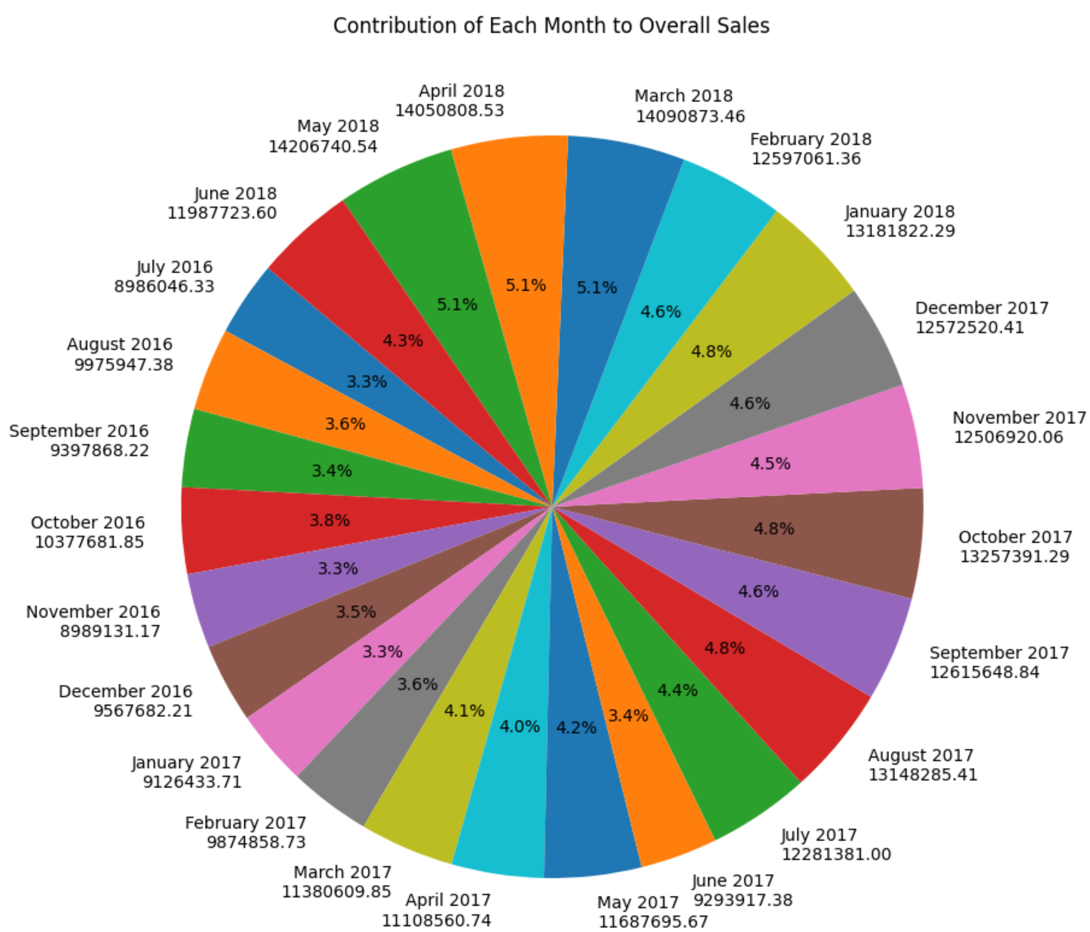
We performed seasonal decomposition using the additive model to identify underlying trends, seasonality, and residuals. The decomposition helps discern patterns in the data.



4. Monthly Sales Analysis:

- Monthly sales were analyzed to uncover patterns and trends over the two-year period.

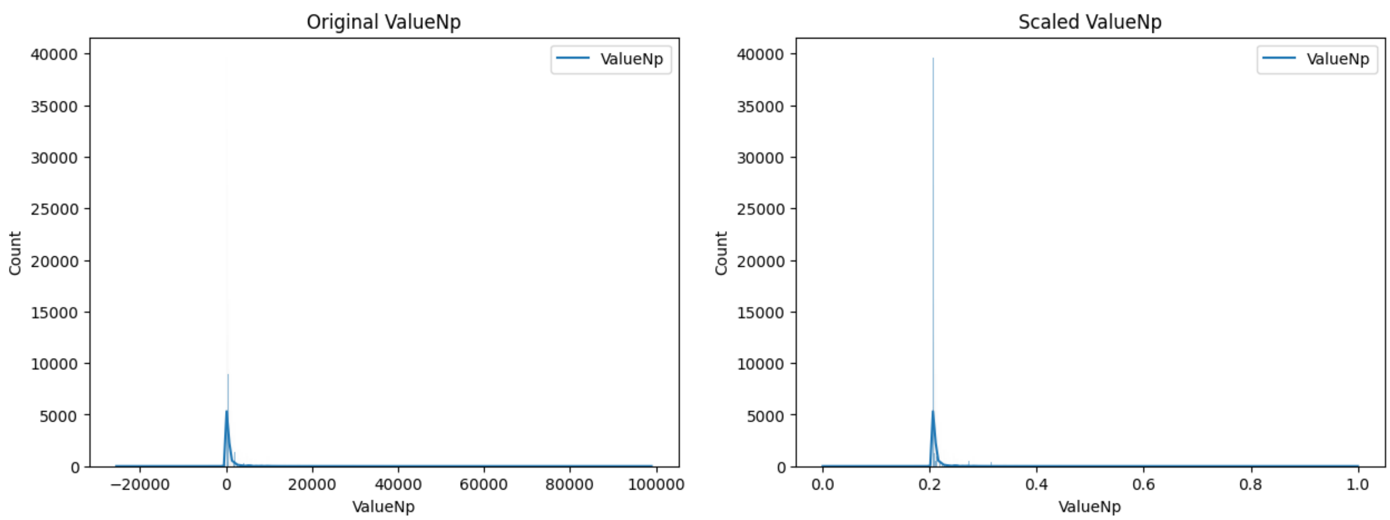


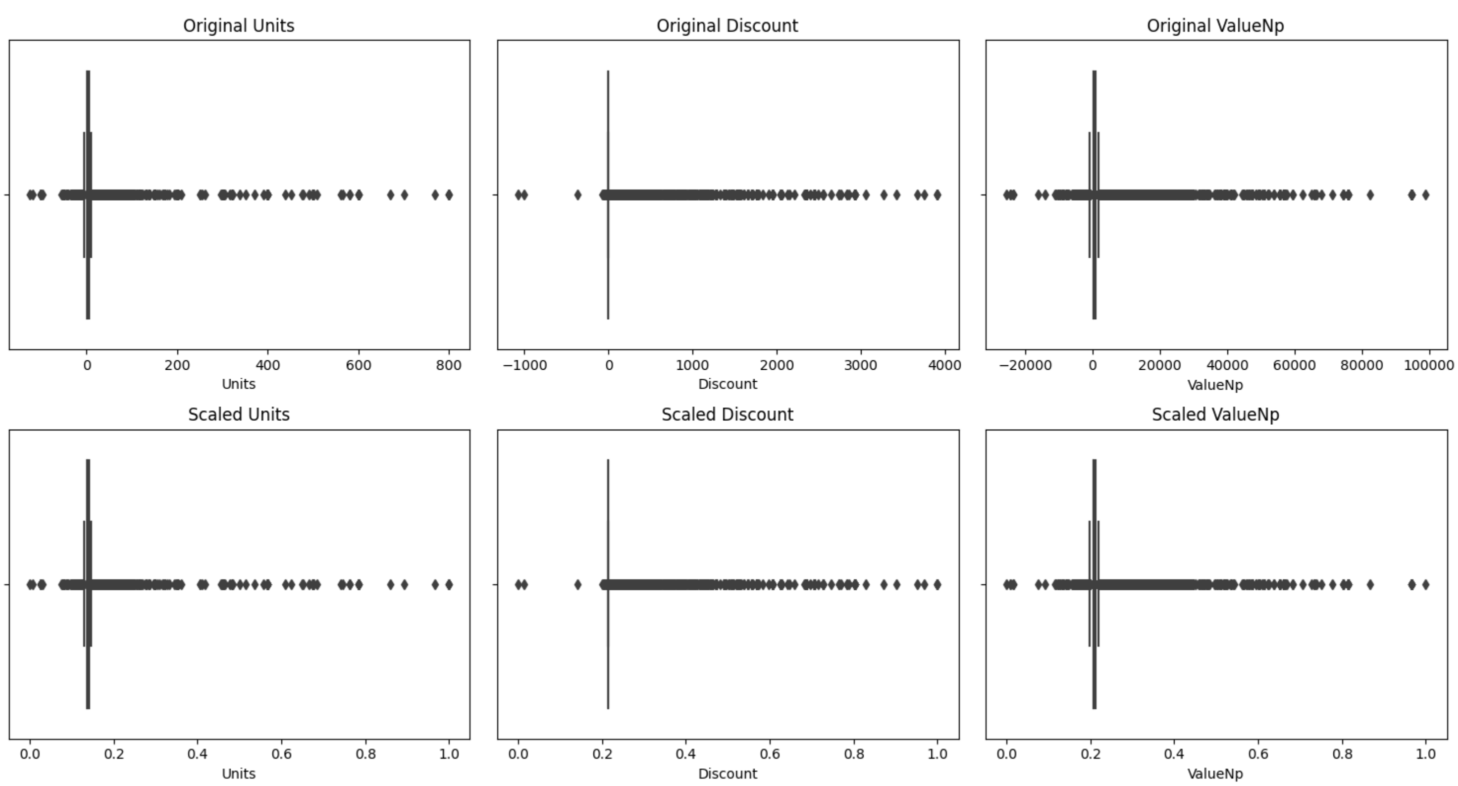


Section 4: Feature Engineering for Prediction Purpose:

1. Min-Max Scaling:

Applied Min-Max scaling to three columns **('Units,' 'Discount,' 'ValueNp')** to bring them to a standard scale. Visualizations of the original and scaled distributions are presented.





2. Encoding Categorical Features:

Categorical features such as 'DistributorName,' 'Product,' 'SKU,' 'ClientName,' and 'BrickName' were encoded for model compatibility.

```
[ ]
dist_name = {
    "Premier Agencies - HYDERABAD": 1,
    "Premier Agencies - Peshawar": 2,
    "Premier Agencies - Sukkur": 3,
}

df_order["DistributorName"] = df_order["DistributorName"].replace(dist_name)
```

```
[ ]
dist_code = {
    2715: 1,
    3996: 2,
    2718: 3
}

df_order["DistributorCode"] = df_order["DistributorCode"].replace(dist_code)
```

```
sku_map = {
    "Prod-NS-Tab": 1.1,
    "Prod-XP-05": 2.1,
    "Prod-XP-10": 2.2,
    "Prod-XP-20": 2.3,
    "Prod-IPL-500 10's": 3.1,
    "Prod-IPL-500 14's": 3.2,
    "Prod-IPL-850 10's": 3.3,
    "Prod-IPL-1000 10's": 3.4,
    "Prod-IPL-1000 14's": 3.5,
    "Prod-EF-30": 4.1,
    "Prod-EF-60": 4.2,
    "Prod-EF-Cap": 4.3,
    "Prod-EF-DS": 4.3,
    "Prod-AC-Tab": 5.1
}

df_order["SKU_encoded"] = df_order["SKU"].replace(sku_map)
```

+ Code + Text

```

le = LabelEncoder()
df_order["ClientName_encoded"] = le.fit_transform(df_order["ClientName"])

```

3. Drop Unnecessary Columns and Final Dataset:

Columns like 'InvoiceDate,' 'ValueNp,' 'Discount,' and 'Units' were dropped to streamline the dataset for modeling purposes.

dropping categorical columns

```
[ ]

drop = ['DistributorName','ClientCode', 'ClientName', 'BrickName', 'Product', 'SKU', 'Bonus']
df_order = df_order.drop(columns=drop, inplace=False)

```

```

drop = ['InvoiceDate', 'ValueNp', 'Discount', 'Units']
df_order = df_order.drop(columns=drop, inplace=False)

df_order.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 290514 entries, 0 to 290513
Data columns (total 5 columns):
 #   Column              Non-Null Count   Dtype
---  ------              --------------   -----
 0   DistributorCode     290514 non-null  int64
 1   Product_encoded     290514 non-null  int64
 2   SKU_encoded         290514 non-null  float64
 3   ClientName_encoded  290514 non-null  int64
 4   BrickName_encoded   290514 non-null  int64
dtypes: float64(1), int64(4)
memory usage: 11.1 MB
```

The dataset is now ready for **model training and evaluation**.

Section 5: Predictive Modeling:

The predictive modeling phase aimed to forecast future sales using a robust ensemble model. Utilized a **Voting Regressor combining RandomForest and GradientBoosting models**.

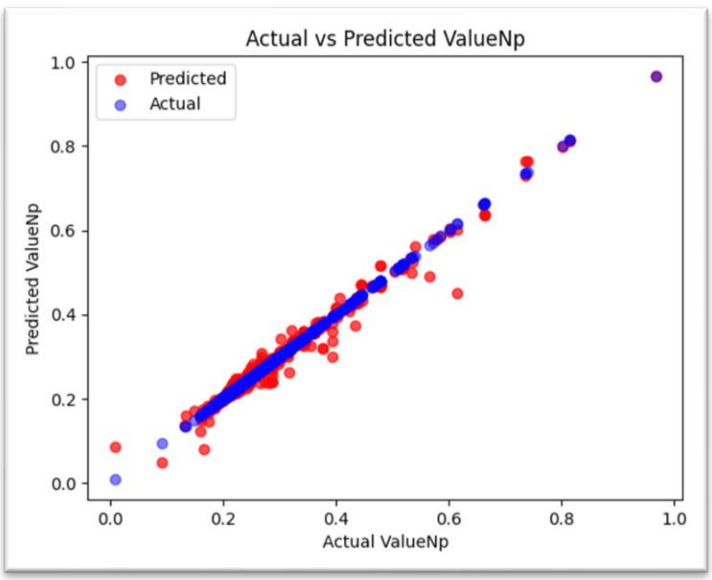
The model was trained on *80% dataset*, *tested on 20% dataset*, and evaluated using metrics such as *Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared.* The evaluation results indicated high accuracy and reliability.

Model Evaluation Results:

```
print(f"R-squared: {r_squared:.4f}")

Mean Squared Error (MSE): 0.0000
Mean Absolute Error (MAE): 0.0003
R-squared: 0.9928
```

The predictive model's accuracy was further visualized through a scatter plot comparing actual vs. predicted values.



Predicted and Actual Scaled ValueNp values (20%):

	Actual	Predicted
0	0.221311	0.220446
1	0.209734	0.209806
2	0.207650	0.207605
3	0.206489	0.206466
4	0.208128	0.208001
...	...	...
58098	0.206489	0.206479
58099	0.205704	0.205704
58100	0.211516	0.211547
58101	0.207213	0.207274
58102	0.209631	0.209679

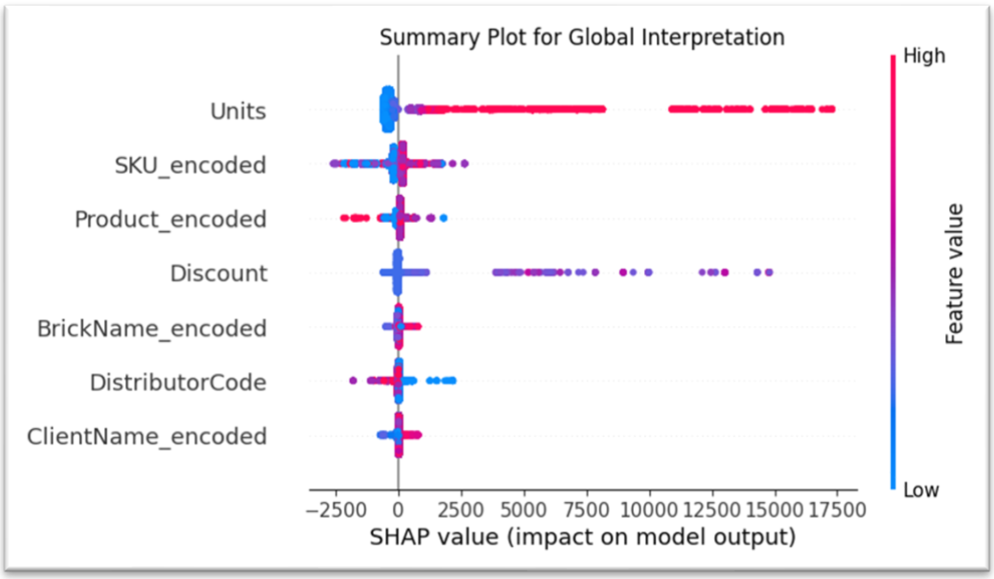
[58103 rows x 2 columns]

Section 6: SHAP Interpretation:

Employed SHAP values to interpret the model's decisions and understand the importance of features in predicting sales.

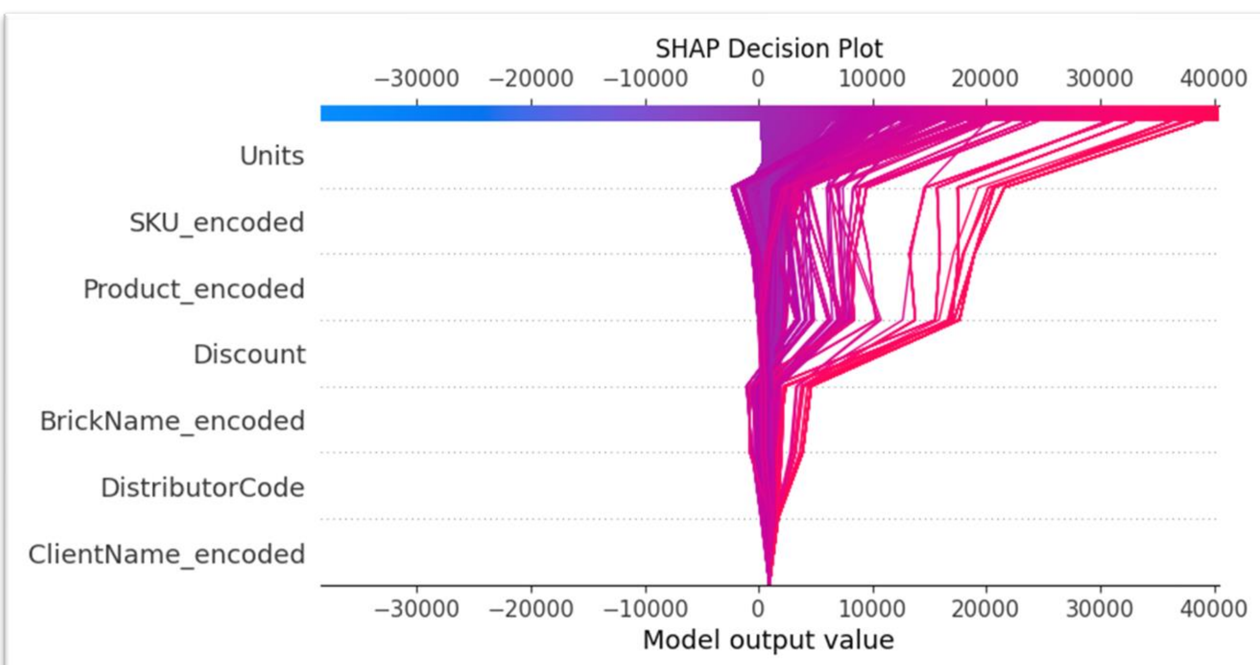
SHAP summary plots provided a global interpretation, while dependence plots showcased the impact of individual features on predictions. They helped in identifying key factors influencing sales predictions and insights into the model's decision-making process.

- **SHAP Summary Plot:**

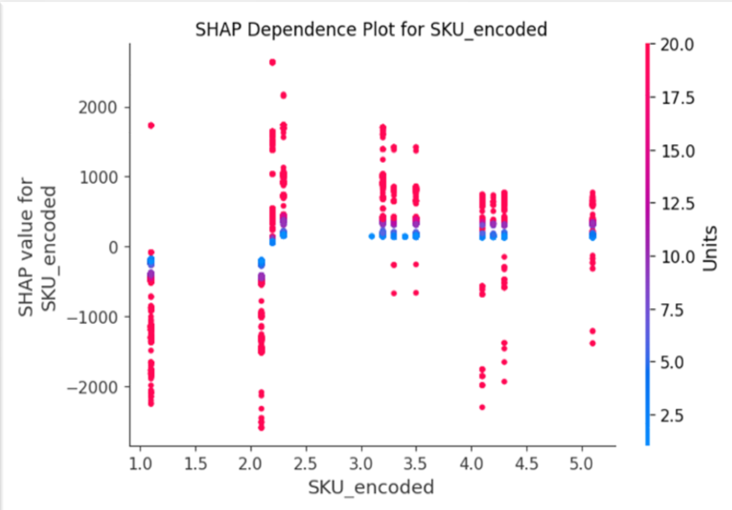


Plot describes that **Units** has major impact followed by Product Types i.e., SKU

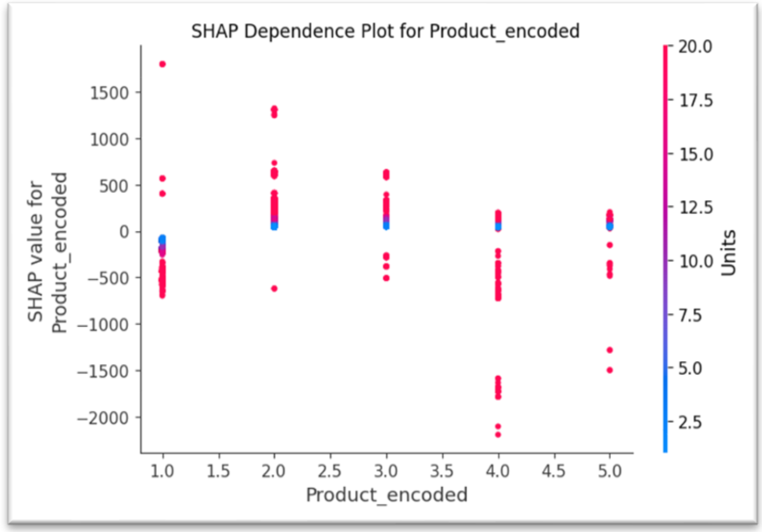
- **SHAP Decision Plot:**



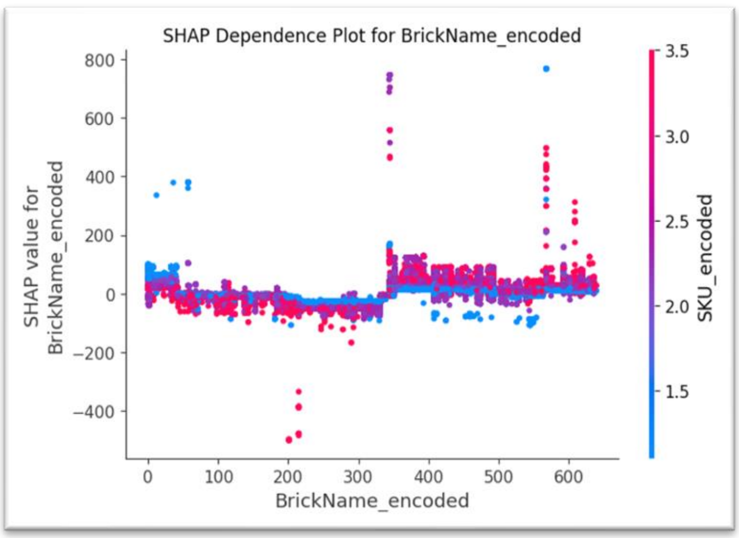
SKU (Stock Keeping Unit)



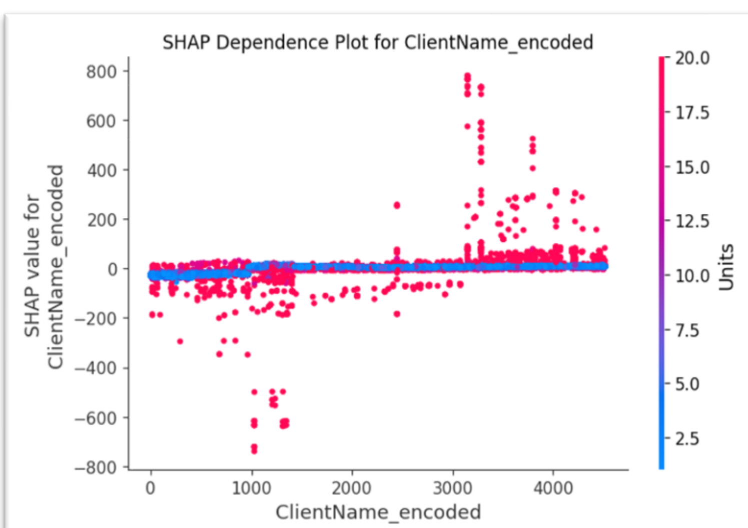
Product



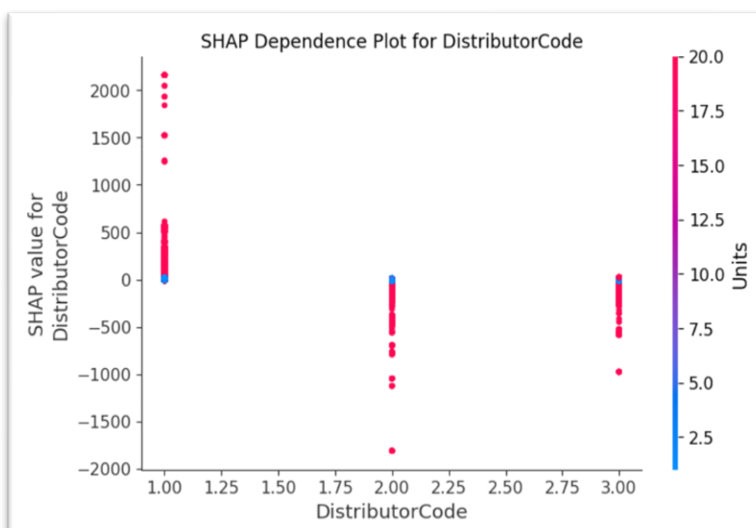
Brick Name



Client Name(Encoded)



Distributor Code



Section 7: Conclusion:

- What We Explored:
  We dug into a big set of sales data to understand how things work. It's like looking at sales trends, figuring out when sales are high or low, and finding patterns over time.

- What We Found:
  We noticed interesting things in the data, like when sales go up or down. We also checked out what happens each month and saw which months bring in the most sales.

- What We Predicted:
  We used cool math stuff to predict future sales. Our predictions were quite accurate, and we measured this accuracy using numbers like MSE, MAE, and R-squared.

- How We Figured It Out:
  We made a smart model that learned from the data. It's like having a super-smart friend who can predict what might happen next based on what happened before.

- Understanding the Predictions:
  We didn't stop at predictions; we also wanted to know why the model made those predictions. We uncovered the secrets behind the predictions by looking at important factors like the distributor, product, and client names.

- Why It Matters:
  All this isn't just about numbers; it's about helping businesses make smart decisions. By understanding sales patterns and predicting the future, we're handing over a powerful tool for making wise choices.

- The Big Picture:
  In wrapping up, our journey was like solving a puzzle. We looked at data, predicted the future, and understood why. It's not just about numbers; it's about empowering businesses to make informed decisions.

End of Project — Unraveling Sales Secrets for Smart Decisions!