

# PHASE-4 DEVELOPMENT PART-II

## PRODUCT SALES ANALYSIS

### INTRODUCTION:

- ❖ Product sales analysis is a critical process that businesses undertake to evaluate their sales performance and make informed decisions. It involves examining sales data, trends, and customer behavior to gain insights into what products are selling, to whom, when, and why.
- ❖ By conducting a product sales analysis, businesses can identify their best-selling items, target audience, and market demand, enabling them to optimize their strategies, increase revenue, and enhance customer satisfaction.
- ❖ In other words, Product sales analysis is the process of examining data related to product sales to gain insights, optimize strategies, and drive business growth.

### OVERVIEW:

The following is an overview of the process of building a product sales analysis used by feature engineering, model training and evaluation.

1. **Data Collection:**

Gather sales data, including product details, quantities sold, prices, and customer information.

2. **Data Cleaning and Preparation:**

Clean and preprocess the data to ensure accuracy and consistency, handling missing values and outliers.

3. **Analysis and Insights:**

Use various techniques to analyze the data, including metrics calculation, segmentation, and regression analysis to extract actionable insights.

4. **Visualization and Reporting:**

Create visualizations and reports to communicate findings and trends to stakeholders.

5. **Continuous Improvement:**

Use the insights gained to refine sales strategies, optimize pricing, and enhance marketing efforts for better sales performance.

### PROCEDURE:

1. **Feature engineering:**

Feature engineering for product sales analysis involves creating meaningful data attributes to enhance predictive modeling and uncover insights. Here's how it can be done:

- **Time-Based Features:**  
Extract information from timestamps, such as year, month, day of the week, and seasonality.
- **Lag Features:**  
Include past sales data as lag features (e.g., previous day/week/month sales) to capture trends.
- **Moving Averages:**  
Calculate rolling statistics (e.g., moving averages) to smooth out noise and reveal trends.
- **Categorical Encoding:**  
Convert categorical variables (e.g., product categories) into numerical format using one-hot encoding or label encoding.
- **Customer Behavior:**  
Create features related to customer behavior, like customer purchase history, retention, or lifetime value.
- **Price Elasticity:**  
Incorporate price-related features to assess the impact of pricing on sales.
- **Promotion Effects:**  
Include features related to promotions, discounts, or marketing campaigns to analyze their influence on sales.
- **Market Indicators:**  
Integrate external market data (e.g., economic indicators) if relevant.
- **Geospatial Data:**  
Utilize location data to understand regional variations in sales.
- **Interaction Terms:**  
Create interaction features to capture synergistic effects between variables.
- **Anomaly Detection:**  
Identify outliers as features to capture exceptional events or unusual patterns.

Feature engineering transforms raw data into informative features, enhancing the performance of machine learning models and providing deeper insights into product sales trends.

# Feature Engineering



Here is an example program code based on feature engineering product sales analysis.

## Program:

```
import pandas as pd
```

```
data = {  
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04'],  
    'ProductCategory': ['A', 'B', 'A', 'C'],  
    'Sales': [100, 150, 120, 80],  
    'Promotion': [0, 1, 0, 1]  
}  
  
sales_data = pd.DataFrame(data)  
sales_data['Date'] = pd.to_datetime(sales_data['Date'])  
sales_data['Year'] = sales_data['Date'].dt.year  
sales_data['Month'] = sales_data['Date'].dt.month  
sales_data['Day'] = sales_data['Date'].dt.day  
sales_data['PreviousDaySales'] = sales_data['Sales'].shift(1)  
sales_data['RollingMean3Days'] = sales_data['Sales'].rolling(window=3).mean()  
sales_data = pd.get_dummies(sales_data, columns=['ProductCategory'], prefix='Category')  
print(sales_data)
```

## Output:

Date	Sales	Promotion	Year	Month	Day	PreviousDaySales
2023-01-01	100	0	2023	1	1	NaN

2023-01-02	150	1	2023	1	2	100.0
2023-01-03	120	0	2023	1	3	150.0
2023-01-04	80	1	2023	1	4	120.0

RollingMean3Days	Category_A	Category_B	Category_C
0	NaN	1	0
1	NaN	0	1
2	123.333333	1	0
3	116.666667	0	1

## 2. Model training:

Model training for product sales analysis is the process of using historical sales data to build a predictive model. Here are the steps in simple terms:

- **Data Preparation:**  
Collect and clean your sales data, ensuring it's in a format suitable for analysis.
- **Feature Selection:**  
Choose relevant factors that may affect sales, like product features, pricing, promotions, and time-related information.
- **Data Splitting:**  
Divide your data into two parts: one for training the model and another for testing its performance.
- **Model Selection:**  
Decide on the type of model to use; for sales analysis, linear regression is a common choice, but more complex models can be considered as well.
- **Model Training:**  
Use the training data to teach the model how various factors impact sales. The model learns to make predictions based on this historical data.
- **Model Evaluation:**  
Use the testing data to assess how well the model predicts sales. Common evaluation metrics include Mean Squared Error (MSE) and R-squared (R2).

- **Model Tuning:**  
Adjust the model's parameters or try different models to improve its predictive accuracy.
- **Deployment:**  
Once satisfied with the model's performance, you can use it to make sales predictions for new or future data.



Here is an example python code for model training in product sales analysis.

**Program:**

```

import pandas as pd

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

data = {
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04'],
    'ProductCategory': ['A', 'B', 'A', 'C'],
    'Sales': [100, 150, 120, 80],
    'Promotion': [0, 1, 0, 1]
}

sales_data = pd.DataFrame(data)

X = sales_data.drop('Sales', axis=1)
y = sales_data['Sales']
  
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,  
random_state=42)
```

```
model = LinearRegression()  
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
```

```
mse = mean_squared_error(y_test, y_pred)  
r2 = r2_score(y_test, y_pred)
```

### **Output:**

Mean Squared Error: 165.0

R-squared (R2) Score: -3.0952380952380956

### **3.Evaluation:**

Evaluating product sales analysis involves scrutinizing the results of your sales data investigation. Key areas to assess include:

- 1. Performance Metrics:**

Reviewing essential metrics such as total revenue, profit margins, and customer acquisition costs to gauge the overall financial health of your sales efforts.

- 2. Trends and Patterns:**

Identifying recurring patterns, seasonality, and trends in sales data, which can aid in forecasting and strategic planning.

- 3. Customer Segmentation:**

Analyzing the behavior and preferences of different customer groups to tailor strategies and marketing efforts accordingly.

- 4. Pricing and Promotions:**

Assessing the impact of pricing strategies, discounts, and promotional activities on sales and profitability.

- 5. Inventory Management:**

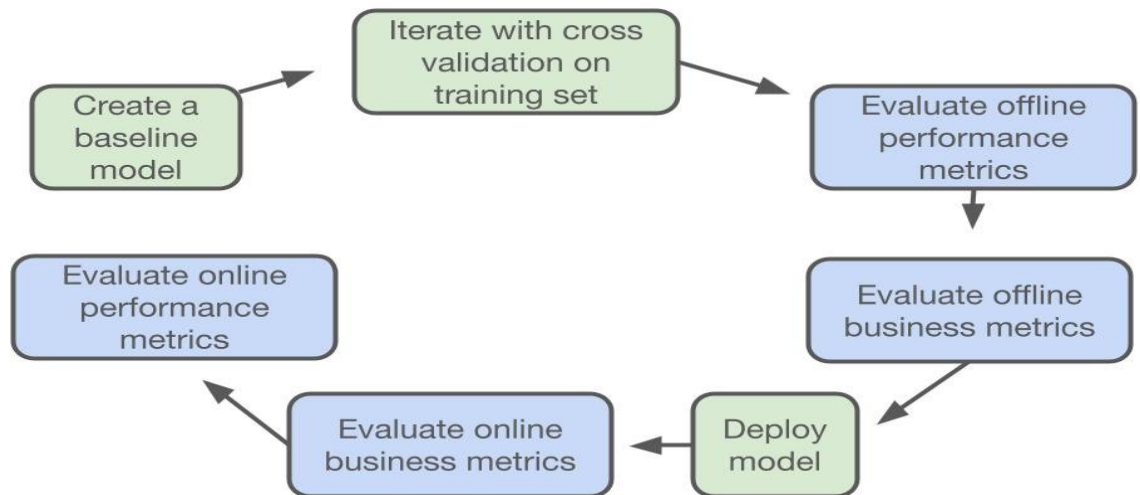
Evaluating inventory turnover, stocking levels, and order fulfillment efficiency to optimize stock management.

- 6. Marketing Campaigns:**

Measuring the effectiveness of marketing campaigns and understanding which strategies contribute the most to sales growth.

- 7. Competitor Comparison:**

Benchmarking your sales performance against competitors or industry standards to determine your market position and identify areas for improvement.



Here is an example code for evaluation in product sales analysis.

**Program:**

```
import pandas as pd
import matplotlib.pyplot as plt

data = {
    'Date': ['2023-01-01', '2023-01-02', '2023-01-03', '2023-01-04'],
    'Sales': [100, 150, 120, 80]
}

sales_data = pd.DataFrame(data)
total_revenue = sales_data['Sales'].sum()
print("Total Revenue:", total_revenue)
average_sales = sales_data['Sales'].mean()
print("Average Daily Sales:", average_sales)
sales_data['Date'] = pd.to_datetime(sales_data['Date'])
plt.figure(figsize=(8, 4))
plt.plot(sales_data['Date'], sales_data['Sales'], marker='o')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.title('Sales Trend Over Time')
plt.grid(True)
```

```
plt.show()
```

**Output:**

Total Revenue: 450

Average Daily Sales: 112.5

**CONCLUSION:**

In conclusion, product sales analysis is a crucial tool for businesses aiming to understand and enhance their sales performance. By mining and interpreting sales data, companies can make data-driven decisions, improve strategies, and ultimately drive growth and profitability in a competitive market. Thus the overall process for feature engineering, model training, evaluation with an example program code is concluded.