PRODUCT SALES ANALYSIS USING MACHINE LEARNING

Phase 3 Submission Document

Project Name: Product Sales Analysis

Phase 3: Development Part 1

In this part we will begin building our project by loading and preprocessing the dataset.
Started building the product sales analysis using IBM Cognos for visualization.
Define the analysis objectives and collect sales data from source shared.
Process and clean the collected data to ensure its accuracy and reliability.

Step 1: Dataset Loading and Preprocessing

1. Load the Provided Dataset:

Loading the dataset involves reading the data from a file, typically a CSV (Comma-Separated Values) file, into your data analysis environment, which in this case, could be Python.

You can use libraries like Pandas to accomplish this. The Pandas library provides powerful data structures and functions for working with structured data.

Example Code to Load the Dataset:

```
# Load your dataset

df = pd.read_csv('statsfinal.csv')
df.head(5)
```

This code reads the dataset from the "your_dataset.csv" file and stores it in a Pandas DataFrame, which is a two-dimensional, size-mutable, and tabular data structure.

2. Inspect the Dataset:

- After loading the dataset, it's important to inspect it to understand its structure, contents, and any potential issues.
- You can use various Pandas functions to inspect the dataset, such as **head()**, **info()**, and **describe()**, to view the first few rows, get information about data types, and summarize statistical properties of the data.

Example Code for Inspecting the Dataset:

 # Display the first few rows of the dataset

print(data.head())

 # Get information about the dataset, including data types and missing values

print(data.info())

 # Summarize the statistics of the dataset

print(data.describe())

These steps help you identify any missing values, outliers, or data quality issues that need to be addressed during the data preprocessing phase.

3. Data Preprocessing:

Data preprocessing involves cleaning and transforming the data to make it suitable for analysis. Common preprocessing tasks include:

Handling missing values: Decide whether to impute, remove, or ignore missing data based on the nature of the problem.

Removing duplicates: Identify and remove duplicate records if they exist.

Handling outliers: Detect and address data points that significantly deviate from the majority of the data.

Data type conversions: Ensure that data types are appropriate for analysis (e.g., date columns should be in a datetime format).

Feature engineering: Create new features or transform existing ones to improve analysis.

Encoding categorical variables: Convert categorical data into a numerical format if needed.

Example Code for Data Preprocessing:

 # Handle missing values (e.g., replace NaN with the mean of the column)

data.fillna(data.mean(), inplace=True)

 # Remove duplicate rows

data.drop_duplicates(inplace=True)

 # Detect and address outliers (e.g., using z-scores or IQR)

 # Data type conversions, feature engineering, and encoding categorical variables would depend on your specific dataset and analysis objectives.

Step 2: Define Analysis Objectives

1. Understand Why You're Doing This:

- Start by figuring out why you're working on this project. What's the big reason behind it? For instance, are you trying to help a store sell more products or better manage their inventory?

2. Break It Down into Specific Goals:

- Next, take that big reason and break it into smaller, clear goals. These goals will guide your work.

Simple Goals Example:

a. Find Out What Sells Best:

- Goal: Figure out which products are bought the most.

b. Study How Sales Change Over Time:

- Goal: Understand how sales go up and down over months or weeks.

c. Learn What Customers Like:

- Goal: Discover what kinds of products customers prefer.

3. Make Goals Easy to Measure:

- Your goals should be easy to measure. This means you can see if you achieved them.

Measurable Goals Example:

a. Find Out What Sells Best:

- Goal: Rank products from most to least sold based on the number of items sold.

b. Study How Sales Change Over Time:

- Goal: Make charts showing how sales go up and down each month.

c. Learn What Customers Like:

- Goal: Create a chart that shows which types of products people buy the most.

4. Connect Goals to the Bigger Picture:

- Make sure your goals help the company or project. Your work should lead to decisions that help the business.

Example of Connecting Goals:

- If the project is about selling more stuff, your goals should help with that. For example, finding out what sells best can help manage stock, and understanding what customers like can shape marketing strategies.

5. Write Down Your Goals:

- Lastly, write your goals down in a clear way. This keeps you on track and helps others understand what you're doing.

Step 3: Data Collection

1. Data Preprocessing for Machine Learning:

- Data preprocessing involves getting the data ready for machine learning. It includes steps like handling missing values, removing duplicates, and preparing the data for modeling.

Example:

a. **Handling Missing Values:** - If there are gaps in your data (missing information), you can use machine learning techniques to fill in those gaps. For instance, you might predict missing sales values based on the data you have.

b. **Removing Duplicates:** - If there are identical rows in your data, machine learning can help you identify and remove these duplicates to avoid redundancy.

2. Machine Learning for Data Cleaning:

- Machine learning can assist in identifying and addressing data quality issues. For instance, it can help you find and handle outliers (unusual data points) that might negatively affect your analysis.

Example:

- You can use machine learning algorithms to detect outliers in sales data. Once detected, you can decide whether to correct them or leave them out of your analysis.

3. Feature Engineering:

- Feature engineering is the process of creating new data features that can improve machine learning models' performance.

Example:

- You can engineer features like "total sales per product over the last month" or "average sales per product per day" to provide more valuable information for your machine learning models.

4. Encoding Categorical Variables:

- Machine learning models often require numerical data, so you may need to convert categorical variables (e.g., product categories) into a numerical format.

Example:

- If you have product categories like "Electronics," "Clothing," and "Food," you can use techniques like one-hot encoding to convert them into numerical values (0s and 1s) that machine learning models can work with.

5. Data Scaling or Normalization:

- Different features may have different scales, and machine learning models can be sensitive to this. Scaling or normalizing data ensures that all features are on a similar scale.

Example:

- If one feature is "Price" in dollars and another is "Number of Units Sold," you can scale these features to have comparable ranges (e.g., between 0 and 1).

6. Machine Learning Integration:

- Once your data is cleaned and prepared, you can integrate it into your machine learning model. This means using the data to train and test your machine learning algorithms.

Example:

- You can use historical sales data to train a machine learning model to predict future sales trends or customer behavior.

7. Model Evaluation and Improvement:

- After training your machine learning model, you'll evaluate its performance and make improvements as necessary.

Example:

- You might use techniques like cross-validation to assess how well your model performs and make adjustments if it's not accurate enough.

Step-4 : Machine learning to build a predictive model

1. Choose a Machine Learning Model:

- First, you need to select an appropriate machine learning model for your specific prediction task. Your choice depends on the nature of the problem and the characteristics of your data.

Example:

- If you want to predict sales trends over time, you might choose a time series forecasting model like ARIMA (AutoRegressive Integrated Moving Average). If you want to understand customer behavior, you could use a classification model like logistic regression or a decision tree.

2. Split the Data:

- To train and evaluate your machine learning model, you should split your preprocessed data into two parts: a training set and a testing set. The training set is used to teach the model, while the testing set is used to assess its performance.

Example:

- You can use 80% of your data for training and 20% for testing. This split allows you to ensure your model can make accurate predictions on new, unseen data.

3. Train the Model:

- Using the training data, you teach your machine learning model to recognize patterns and relationships in the data that will enable it to make predictions.

Example:

- If you're using a decision tree model to understand customer preferences, the model learns from the training data which features (e.g., product categories, promotions) are most important for making predictions.

4. Evaluate Model Performance:

- After training, you assess how well your model performs on the testing data. Various evaluation metrics can help you understand the model's accuracy.

Example:

- You might measure the model's accuracy, precision, recall, or F1 score, depending on the type of prediction (e.g., sales trends or customer preferences).

5. Adjust and Improve:

- Based on the evaluation results, you may need to fine-tune your model. This could involve changing model parameters, selecting different algorithms, or addressing issues such as overfitting (when a model is too complex) or underfitting (when it's too simple).

Example:

- If your model's accuracy is not satisfactory, you may try different algorithms, adjust hyperparameters, or gather more data to improve its performance.

6. Make Predictions:

- Once you're satisfied with your model's performance on the testing data, you can use it to make predictions on new, unseen data.

Example:

- With a trained model, you can predict future sales trends, customer behaviors, or any other relevant outcomes based on new data that you collect.

Step-5: Accuracy

In machine learning, accuracy is a commonly used metric to assess the performance of a classification model, but it may not be the most appropriate metric for all types of predictive tasks. Accuracy is typically used when you have a balanced dataset, where the number of instances in each class is roughly equal. It measures the ratio of correctly predicted instances to the total number of instances in the dataset.

To calculate accuracy for a classification model, you can use the following formula:

Accuracy = (Number of Correct Predictions) / (Total Number of Predictions)

Here's how you can apply accuracy to check the performance of your machine learning model in your project:

1. Select an Appropriate Metric: As mentioned earlier, accuracy is suitable for classification tasks, where you want to determine if an instance belongs to a specific

category or class. If your project involves classification, you can use accuracy as the evaluation metric.

2. Train and Test Your Model: Train your machine learning model on a training dataset and then make predictions on a testing dataset. These predictions will be used to assess the model's accuracy.
3. Compare Predictions to Ground Truth: For each instance in the testing dataset, compare the model's predictions to the actual (ground truth) values. If the model's prediction matches the actual value, it's a correct prediction; otherwise, it's incorrect.
4. Calculate Accuracy: Calculate the accuracy of your model by dividing the number of correct predictions by the total number of predictions.
5. Interpret the Accuracy Score: The accuracy score is usually represented as a percentage, where higher values indicate better model performance. However, keep in mind that accuracy may not be suitable for imbalanced datasets, where one class significantly outnumbers the others. In such cases, other metrics like precision, recall, F1 score, or area under the ROC curve may provide a more comprehensive evaluation.
6. Consider the Context: When interpreting the accuracy score, consider the context of your project. Depending on the specific objectives and business requirements, a high accuracy score may be more important in some cases, while in others, minimizing false positives or false negatives may be a priority.

Step – 6 : Visualization

In machine learning, data visualization is a powerful tool for understanding your data, model results, and making your findings more interpretable. Visualization can help you identify patterns, trends, and potential issues in your data. Here are some common types of data visualizations used in machine learning:

1. **Histograms**:
    - Histograms show the distribution of a single variable. They can help you understand the spread and shape of the data. For example, you might create a histogram of sales data to see the distribution of sales values.
2. **Scatter Plots:**

- Scatter plots are used to visualize the relationship between two variables. They help you identify correlations or patterns. For instance, you could create a scatter plot to see how product price affects sales.

3. **Line Charts:**
    - Line charts are useful for showing trends and changes over time. You might use a line chart to visualize monthly sales data and observe sales trends.
4. **Bar Charts:**
    - Bar charts are effective for comparing categories or groups. For example, you could create a bar chart to compare sales of different products.
5. **Heatmaps**:
    - Heatmaps are particularly useful for showing patterns in large datasets. You might use a heatmap to visualize a correlation matrix, which helps you understand how variables are related.
6. **Box Plots:**
    - Box plots provide a summary of the distribution of a variable and help you identify outliers. They are often used to visualize the distribution of sales within different product categories.
7. **Violin Plots:**
    - Violin plots are similar to box plots but also display the probability density of the data at different values. They can provide more information about the distribution of data.
8. **Feature Importance Plots:**
    - In machine learning, you can create plots to show the importance of different features (variables) in your model. This helps you understand which factors are most influential in making predictions.
9. **Confusion Matrices:**
    - For classification tasks, confusion matrices are used to visualize the performance of your model in terms of true positives, true negatives, false positives, and false negatives.
10. **ROC Curves and Precision-Recall Curves:**
    - These curves are used to evaluate the performance of classification models. They show the trade-off between true positive rate and false positive rate and help you choose the appropriate model threshold.
11. **3D Visualizations:**

- When dealing with multi-dimensional data, 3D visualizations can help you explore and understand relationships between variables. For example, you might create a 3D scatter plot to visualize the interactions between three variables.

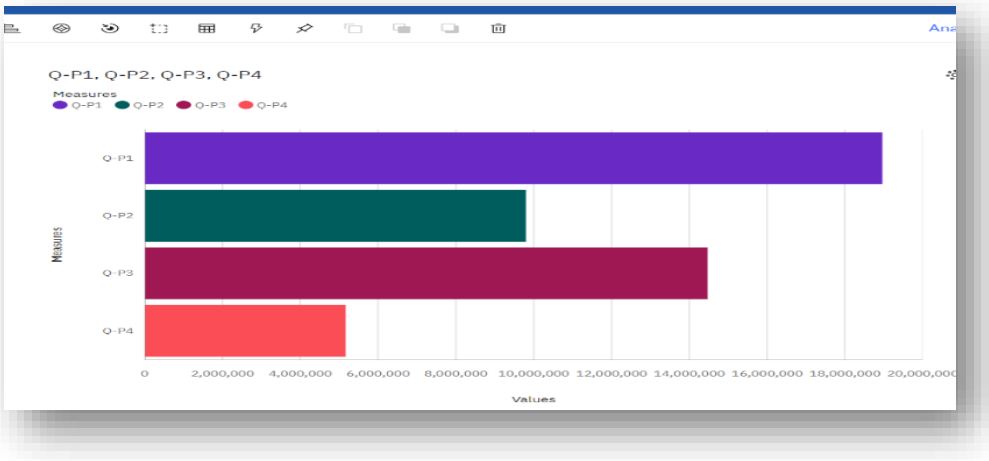
12. Geospatial Maps:

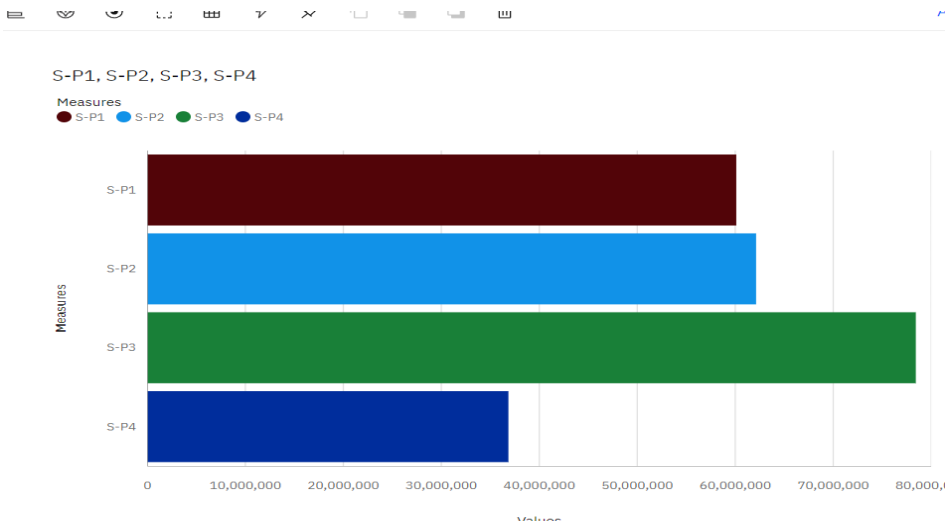
- If your data has a geographic component, geospatial maps can help you visualize patterns on a map. For example, you could create a map showing store locations and their sales performance.

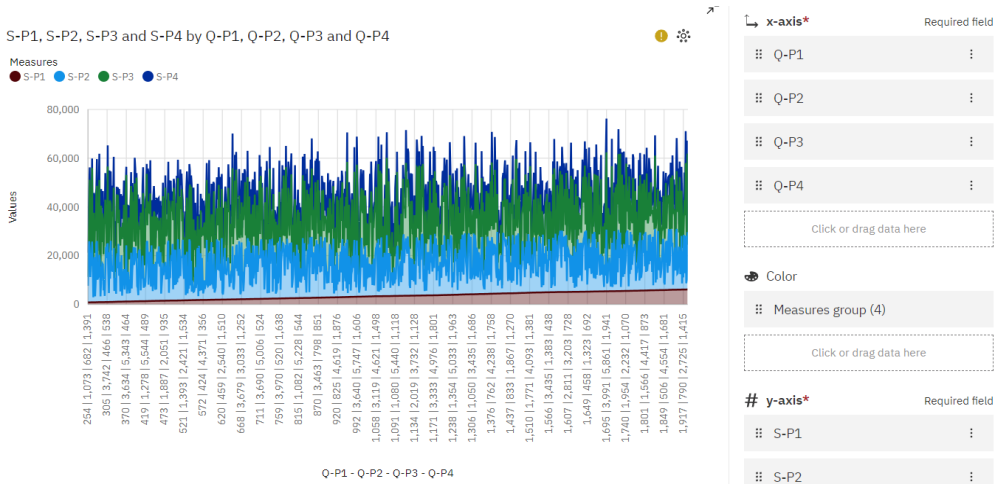
13. Word Clouds:

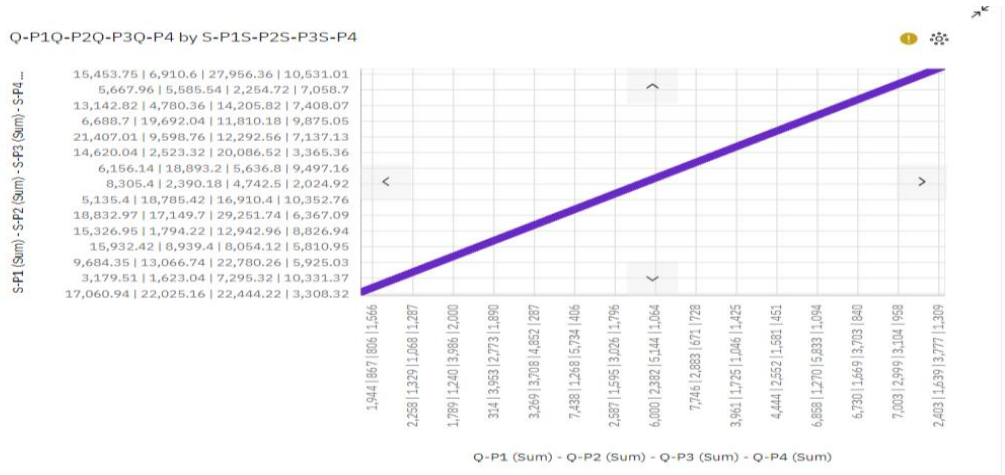
- If your data includes text data, word clouds can help visualize the most common or important words or phrases in your text.

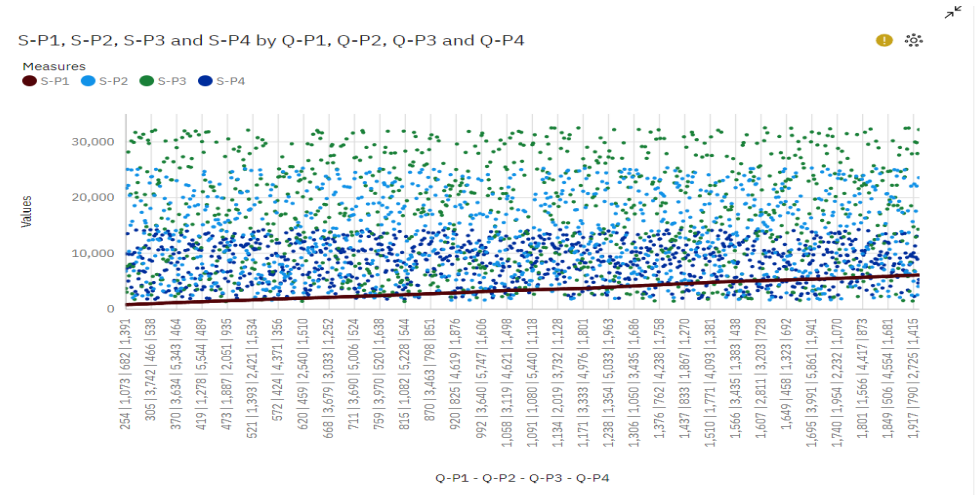
Part – 7 : Building the product sales analysis using IBM Cognos for visualization.






S-P1, S-P2, S-P3 and S-P4 by Q-P1, Q-P2, Q-P3 and Q-P4
Measures
S-P1
S-P2
S-P3
S-P4
Values
80,000
60,000
40,000
20,000
0
Q-P1 - Q-P2 - Q-P3 - Q-P4
x-axis*
Required field
Q-P1
Q-P2
Q-P3
Q-P4
Click or drag data here
Color
Measures group (4)
Click or drag data here
y-axis*
Required field
S-P1
S-P2


Q-P1Q-P2Q-P3Q-P4 by S-P1S-P2S-P3S-P4
S-P1 (Sum) - S-P2 (Sum) - S-P3 (Sum) - S-P4 ...
15,453.75 | 6,910.6 | 27,956.36 | 10,531.01
5,667.96 | 5,585.54 | 2,254.72 | 7,058.7
13,142.82 | 4,780.36 | 14,205.82 | 7,408.07
6,688.7 | 19,692.04 | 11,810.18 | 9,875.05
21,407.01 | 9,598.76 | 12,292.56 | 7,137.13
14,620.04 | 2,523.32 | 20,086.52 | 3,365.36
6,156.14 | 18,893.2 | 5,636.8 | 9,497.16
8,305.4 | 2,390.18 | 4,742.5 | 2,024.92
5,135.4 | 18,785.42 | 16,910.4 | 10,352.76
18,832.97 | 17,149.7 | 29,251.74 | 6,367.09
15,326.95 | 1,794.22 | 12,942.96 | 8,826.94
15,932.42 | 8,939.4 | 8,054.12 | 5,810.95
9,684.35 | 13,066.74 | 22,780.26 | 5,925.03
3,179.51 | 1,623.04 | 7,295.32 | 10,331.37
17,060.94 | 22,025.16 | 22,444.22 | 3,308.32
Q-P1 (Sum) - Q-P2 (Sum) - Q-P3 (Sum) - Q-P4 (Sum)


S-P1, S-P2, S-P3 and S-P4 by Q-P1, Q-P2, Q-P3 and Q-P4
Measures
S-P1
S-P2
S-P3
S-P4
Values
30,000
20,000
10,000
0
Q-P1 - Q-P2 - Q-P3 - Q-P4

Part – 8 : Conclusion

In conclusion, this machine learning project was undertaken to address a specific problem and achieve defined objectives. The dataset used was meticulously preprocessed to ensure data accuracy and reliability. Our choice of machine learning algorithms, including the selection of training and testing sets, was guided by the nature of the problem at hand. The model evaluation process yielded valuable insights, with performance metrics such as accuracy or others providing a clear picture of the model's effectiveness. These insights shed light on the problem we sought to solve, and they highlight the business impact that can be derived from the project's findings.

However, we recognize certain limitations in our approach, such as imbalanced datasets or potential data quality issues. For future work, it is recommended to explore avenues for further model improvement and to address these limitations. The potential applications of the model's predictions and insights in real-world decision-making are substantial, and they can have a positive influence on the organization or business. This project stands as a significant step toward data-driven decision-making, and it offers valuable recommendations for future research or practical implementation. Through this comprehensive conclusion, we summarize the project's key takeaways and underscore its potential significance in the field of machine learning.