Title:

Empowering Business Decisions with Real-Time Insights: A Comprehensive Executive Dashboard for Sales, Engagement, and Inventory

Abstract:

In the current era of digital transformation, businesses generate massive amounts of data daily, making it essential for executives to have tools that provide real-time insights into key performance indicators (KPIs). This project addresses the need for a scalable and efficient data visualization solution, aimed at helping the executive team of a global company monitor crucial KPIs such as sales trends, customer engagement, and inventory levels across five major regions: the USA, Canada, France, Germany, and Mexico.

The core challenge is to process and analyze vast datasets originating from different countries and then visualize the insights in a meaningful way for executives to make data-driven decisions. To overcome this challenge, we implement a big data solution using Hadoop and Apache Spark. Hadoop serves as a distributed storage system, enabling the handling of large volumes of data, while Spark is utilized for distributed data processing, ensuring fast computations and real-time data analysis.

The final product is an interactive dashboard built with Dash and Plotly that provides a visual representation of the KPIs across multiple countries. Executives can track sales performance, monitor customer engagement levels, and assess inventory management across regions, enabling cross-country comparisons. For instance, the dashboard allows them to identify regions with higher sales growth, detect customer engagement trends, and ensure optimal inventory levels.

This solution not only simplifies complex data analysis but also enhances decision-making by providing real-time, visually compelling insights. By focusing on regional data, executives can tailor business strategies to the unique market demands of each country, improving operational efficiency and driving better business outcomes. Moreover, the dashboard is scalable, capable of handling larger datasets as the business expands, making it a future-proof tool for global business management.

The project demonstrates how combining big data processing techniques with advanced visualization tools can solve complex business challenges, providing an efficient, scalable solution that allows management to monitor, analyze, and optimize performance in a dynamic global marketplace. The integration of Hadoop and Spark with interactive dashboards ensures that decision-makers can rely on real-time data, offering them the agility to respond quickly to changing market conditions and seize new opportunities.

Objectives:

⇒ To develop an executive dashboard that visualizes critical KPIs.

⇒ To track and analyze sales trends, customer engagement, and inventory health.

⇒ To provide interactive tools for decision-makers to dive deep into operational data.

⇒ To employ modern big data visualization tools to create a robust, scalable solution.

Introduction:

In today's hyper-competitive business landscape, companies cannot afford to make decisions based solely on intuition or outdated reports. The ability to access and analyze real-time data has become a cornerstone of modern business strategies, allowing for more agile decision-making and proactive management.

However, with the vast amount of data generated daily, it is easy for companies to get lost in a sea of information. This is where dashboards, particularly those tailored for executives, come into play. An effective dashboard distills complex datasets into easy-to-understand, actionable insights, providing a bird's-eye view of the company's performance across critical metrics.

This project focuses on creating a dynamic dashboard for tracking key performance indicators (KPIs) in sales, customer engagement, and inventory management. These KPIs are essential for gauging the company's health and making tactical adjustments to business strategies. The dashboard is designed to be interactive, allowing executives to filter data by time frames, product categories, and geographic regions, enabling more granular analysis.

Using advanced data visualization techniques and powerful platforms such as Tableau, Power BI, and Python's Plotly/Dash, the dashboard becomes more than just a static report—it becomes a real-time decision-making tool. By visualizing sales trends, customer activity, and stock levels, executives can identify opportunities, mitigate risks, and align resources more efficiently.

Dataset Collection with source:

The data was just taken as a sample, created within the code of the implementation. There is no specific source for the same, but after research from sources like Kaggle, SkyScanner, Amul, and other industries and data sources, our chosen dataset is made to mimic real world trends.

The reference dataset from Kaggle, also used on Microsoft Power BI is linked:

https://www.kaggle.com/datasets/anuchhetry/product-sales

Segment: Segment wise sales, data type: Text
Country: 5 countries, data type: Text
Product: 7 Products
Discount Band: None, High, Low, Medium
Units Sold: Quantity present
Manufacturing price: In USD
Sales Price: Sales Price of each product
Gross Sales: Gross sales on each product

Discounts: Discounts on the product
Sales: Sales of each product
COGS: Cost of goods sold
Profit: Profit is given
Date: Date in which sales was achieved

Technologies and Tools Used:

This project leverages a combination of big data processing, data visualization, and dashboarding tools to provide a comprehensive solution for monitoring key performance indicators (KPIs). Below is a detailed description of the tools and technologies used:

1. **Hadoop**

    - **Overview**: Apache Hadoop is an open-source framework designed to store and process large datasets across distributed clusters of computers. It is particularly useful for handling vast amounts of structured, semi-structured, and unstructured data by using a distributed file system (HDFS) and enabling parallel processing.

    - **Role in the Project**: Hadoop is used for distributed data storage, ensuring that the large dataset generated from various countries (USA, Canada, France, Germany, and Mexico) can be stored and accessed efficiently. Hadoop provides scalability, allowing the system to handle an increasing volume of data as the business grows.

2. **Apache Spark**

    - **Overview**: Apache Spark is an open-source, distributed computing system that enhances big data processing by offering in-memory processing, making it significantly faster than traditional disk-based processing frameworks like Hadoop MapReduce.

    - **Role in the Project**: Spark is employed for distributed data processing to quickly analyze the large datasets, including sales trends, customer engagement, and inventory levels. Spark's in-memory processing allows for faster computations, real-time data analytics, and reduced latency, which are critical for delivering timely insights to the business executives**.**

3. **Plotly**

    - **Overview**: Plotly is a powerful data visualization library in Python that allows users to create interactive graphs and visual representations of data. It integrates seamlessly with Dash to create web-based dashboards.

    - **Role in the Project**: Plotly is used to build interactive, visually compelling graphs for KPIs such as sales trends, customer engagement, and inventory levels.

These graphs allow executives to explore the data, compare metrics across different countries, and gain insights into performance trends.

4. **Dash**

   - **Overview**: Dash is a Python framework for building web-based, interactive dashboards. It works well with Plotly for visualizations and can incorporate data analysis tools like Pandas and Spark.

   - **Role in the Project**: Dash is utilized to create the front-end of the interactive KPI dashboard. It allows users to switch between different views, such as sales trends, customer engagement, and inventory levels, and compare the performance of different countries. Dash provides the user interface that facilitates easy exploration of data by business executives.

5. **Tableau**

   - **Overview**: Tableau is a popular business intelligence (BI) tool that enables users to create interactive and shareable dashboards with a wide variety of data visualizations. It's known for its intuitive drag-and-drop interface and powerful data analytics capabilities.

   - **Role in the Project**: Tableau can be used as an alternative or complementary visualization tool for the dashboard. It allows users to create interactive, multi-dimensional views of the KPIs, offering more advanced analytics and drill-down capabilities for data exploration.

6. **Microsoft Power BI**

   - **Overview**: Power BI is another widely used business intelligence tool from Microsoft, designed to provide interactive visualizations and business intelligence capabilities with a simple interface. It integrates well with various data sources, including big data platforms.

   - **Role in the Project**: Power BI offers an alternative platform for creating dashboards. It can be used to visualize KPIs such as sales trends, customer engagement, and inventory levels across different countries. Power BI's strengths include its integration with the Microsoft ecosystem and its ability to handle real-time data streaming and large datasets.

7. **PySpark**

   - **Overview**: PySpark is the Python API for Apache Spark, allowing for easy integration of Spark's distributed data processing capabilities into Python-based projects.

   - **Role in the Project**: PySpark is used to manipulate and process the dataset within the Python environment. It enables the efficient execution of distributed data

processing tasks like filtering, grouping, and aggregation across the large dataset collected from different countries. PySpark ensures the seamless integration of Spark's data processing engine with the Python-based Dash dashboard.

8. **Pandas**

   - **Overview**: Pandas is a widely-used Python library for data manipulation and analysis, providing powerful tools for handling structured data (e.g., tabular or time-series data).

   - **Role in the Project**: Pandas is used in conjunction with PySpark to convert the processed data from Spark into Pandas DataFrames, which can then be used for visualization. Pandas offers flexibility for further analysis and fine-tuning of the data before it is passed to Plotly for graph generation**.**

9. **Jupyter Notebook**

   - **Overview**: Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It's widely used in data science projects for exploration and presentation.

   - **Role in the Project**: Jupyter Notebook serves as the development environment where the dataset is initially explored and processed using PySpark. It provides a convenient interface to visualize intermediate results, develop code incrementally, and ensure that the data pipeline is functioning as expected.

10. **SQL (Structured Query Language)**

    - **Overview**: SQL is a standard language for managing and querying relational databases. It enables users to access and manipulate data stored in various databases.

    - **Role in the Project**: SQL queries can be used within Spark to retrieve and filter the data before processing. If the dataset resides in a relational database, SQL queries help to extract the necessary data for further analysis and visualization in the dashboard.

Summary of Tool Integration:

- **Hadoop and Spark** form the backbone of the big data architecture, allowing the system to store and process massive datasets in a distributed and scalable manner.

- **Plotly and Dash** handle the visualization and dashboard creation, making the insights accessible and interactive for business executives.

- **Power BI and Tableau** provide alternative dashboarding solutions for creating advanced visualizations, offering flexibility in how the KPIs are presented.

- **PySpark and Pandas** ensure seamless integration between Spark's distributed processing and Python's data manipulation capabilities, allowing for flexible and efficient data handling.

By using these tools, the project successfully processes and visualizes large datasets in an efficient, scalable, and user-friendly manner, ensuring that decision-makers can access critical insights in real-time across different regions.

Model Design:

- **PySpark for Data Processing:**
    - We use **PySpark** to create and process the dataset. In a real-world scenario, this would be large-scale data sourced from Hadoop's HDFS or other distributed file systems.
    - The data is processed in Spark to handle large-scale transformations. In this example, we simulate basic data filtering (filter(col('Sales') > 20000)) to mimic how Spark handles processing in a distributed environment.
- **Distributed Processing:**
    - **PySpark DataFrames** are used instead of Pandas, as they are optimized for big data and distributed computation across clusters. After processing, we convert the Spark DataFrame to Pandas for visualizing in Dash.
- **Dash for Visualization:**
    - Once the data is processed, it's transferred to the Dash app for visualization using **Plotly**.
    - Three main graphs (Sales Trends, Customer Engagement, and Inventory Levels) are plotted using the results of Spark processing.
- **Data Processing Workflow with Spark:**
    - **Data Ingestion**: The data is loaded into Spark (simulated as a static list, but can be a CSV or a dataset stored in HDFS).
    - **Processing**: Filters, transformations, and aggregations are performed on the dataset using Spark's powerful distributed processing engine.
    - **Collection**: Once the data is processed, it is collected back to Python as a Pandas DataFrame for further use in visualization tools.

- **Visualization**: Dash and Plotly handle the interactive display of results, allowing executives to drill down into the KPIs.

The following steps were implemented while designing the model, and the necessary justification for the same has also been given alongside.

1. **Data Processing:**
    a. The dataset is cleaned, normalized, and converted into a structured format.
    b. KPIs are derived, including total sales, customer engagement rates, and stock levels.
2. **Visualization Strategy:**
    a. **Sales Trends:** Line chart for sales over time.
    b. **Customer Engagement:** Bar charts for customer activity metrics.
    c. **Inventory Levels:** Gauge charts for stock health and reorder status.
3. **Dashboard Structure:**
    a. **Navigation Panel:** Allows users to select the time period or product category.
    b. **Main Panel:** Displays key KPIs using interactive graphs.

Implementation:

To implement a dashboard suited to the problem statement using **Hadoop and Apache Spark** for big data processing, we can adapt the solution to perform distributed processing of large datasets, and then visualize the results. Spark is used to process and analyze large volumes of data, and Plotly/Dash will handle the visualization part in Python.

Below is the Python code using **PySpark** for data processing, integrated with **Dash** and **Plotly** for visualization.

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
import dash
from dash import dcc, html
import plotly.express as px

#Initialize SparkSession
spark = SparkSession.builder.appName("KPI Dashboard").getOrCreate()

#Create Dataset with countries
data = [
    ('USA', 'Jan', 12000, 100, 80),
    ('USA', 'Feb', 15000, 200, 70),
```

```
    ('USA', 'Mar', 10000, 150, 65),
    ('Canada', 'Jan', 11000, 95, 60),
    ('Canada', 'Feb', 14000, 185, 72),
    ('Canada', 'Mar', 10500, 175, 67),
    ('France', 'Jan', 13000, 110, 85),
    ('France', 'Feb', 15500, 205, 77),
    ('France', 'Mar', 12000, 190, 75),
    ('Germany', 'Jan', 12500, 105, 80),
    ('Germany', 'Feb', 16000, 210, 68),
    ('Germany', 'Mar', 11500, 165, 70),
    ('Mexico', 'Jan', 9000, 80, 60),
    ('Mexico', 'Feb', 12000, 150, 65),
    ('Mexico', 'Mar', 9500, 130, 60),
]

#Create a Spark DataFrame with new data (including country)
columns = ['Country', 'Month', 'Sales', 'CustomerEngagement', 'InventoryLevels']
df_spark = spark.createDataFrame(data, schema=columns)

#Data processing using Spark (for example, filtering or calculating trends)
# Example: Filter months with sales greater than 10,000
df_filtered = df_spark.filter(col('Sales') > 10000)

#Collecting data back to Python for visualization
data_for_dash = df_filtered.toPandas()

#Dash app initialization for visualization
app = dash.Dash(__name__)

#Layout of the Dash app
app.layout = html.Div([
    html.H1("Big Data KPI Dashboard for Multiple Countries (Hadoop & Spark)"),

    dcc.Tabs([
        dcc.Tab(label="Sales Trends", children=[
            dcc.Graph(
                id='sales-trend-graph',
                figure=px.line(data_for_dash, x='Month', y='Sales', color='Country',
title='Monthly Sales Trends by Country')
            )
        ]),

        dcc.Tab(label="Customer Engagement", children=[
            dcc.Graph(
                id='customer-engagement-graph',
                figure=px.bar(data_for_dash, x='Month', y='CustomerEngagement',
color='Country', title='Customer Engagement by Country')
            )
        ]),

        dcc.Tab(label="Inventory Levels", children=[
            dcc.Graph(
                id='inventory-levels-graph',
                figure=px.line(data_for_dash, x='Month', y='InventoryLevels',
color='Country', title='Inventory Levels by Country')
            )
        ])
    ])
])

#Running the Dash app
```

```
if __name__ == '__main__':
    app.run_server(debug=True)
```

Code to generate all the graphs for visualisation using the tools mentioned above:

```
import dash
from dash import dcc, html
import plotly.express as px
import pandas as pd

data = {
    'Country': ['USA', 'Canada', 'France', 'Germany', 'Mexico'],
    'Sales': [25000, 15000, 12000, 22000, 10000],
    'Customer_Engagement': [80, 60, 70, 90, 50],
    'Inventory_Levels': [500, 300, 200, 400, 150]
}

df = pd.DataFrame(data)

#Initialize the Dash app
app = dash.Dash(__name__)

#Bar Chart - Sales by Country
bar_fig = px.bar(df, x='Country', y='Sales', title='Sales by Country',
color='Country',
                 labels={'Sales': 'Total Sales (in USD)', 'Country': 'Country'},
                 text='Sales')

#Pie Chart - Market Share of Sales
pie_fig = px.pie(df, names='Country', values='Sales', title='Market Share of Sales',
                 labels={'Country': 'Country', 'Sales': 'Sales'})

#Scatter Plot - Customer Engagement vs Sales
scatter_fig = px.scatter(df, x='Sales', y='Customer_Engagement', color='Country',
                         size='Inventory_Levels', title='Customer Engagement vs
Sales',
                         labels={'Sales': 'Sales (in USD)', 'Customer_Engagement':
'Customer Engagement (%)'})

#Line Graph - Sales and Inventory Levels by Country
line_fig = px.line(df, x='Country', y=['Sales', 'Inventory_Levels'],
                   title='Sales and Inventory Levels by Country',
                   labels={'value': 'Values', 'Country': 'Country'})

#Layout for Dash App
app.layout = html.Div([
    html.H1("Dashboard for KPI Monitoring", style={'textAlign': 'center'}),

    #Bar Chart
    html.Div([
        html.H2("Bar Chart: Sales by Country"),
        dcc.Graph(id='bar-graph', figure=bar_fig)
    ]),

    #Pie Chart
    html.Div([
        html.H2("Pie Chart: Market Share of Sales"),
        dcc.Graph(id='pie-chart', figure=pie_fig)
    ]),
```

```
    #Scatter Plot
    html.Div([
        html.H2("Scatter Plot: Customer Engagement vs Sales"),
        dcc.Graph(id='scatter-plot', figure=scatter_fig)
    ]),

    #Line Graph
    html.Div([
        html.H2("Line Graph: Sales and Inventory Levels by Country"),
        dcc.Graph(id='line-graph', figure=line_fig)
    ])
])

#Run the Dash app
if __name__ == '__main__':
    app.run_server(debug=True)
```

Steps to Implement:

1. *Set up PySpark*: PySpark will handle the large dataset processing.

2. *Data Processing with Spark*: We simulate distributed data processing with Spark DataFrames.

3. *Visualization with Dash and Plotly*: After processing the data, we use Dash and Plotly to create the dashboard.

Result Analysis:

Using Spark for data processing enhances the capability to handle larger datasets efficiently. The executive dashboard can now process data at scale, filtering important insights such as high sales months and customer engagement patterns in real time. This scalability ensures that, even as the company grows, the dashboard remains responsive and performant, capable of handling millions of data points across multiple KPIs.

The dashboard can provide a comparative analysis of KPIs such as sales, customer engagement, and inventory levels across various countries. This allows executives to gain deeper insights into the performance of different regions.

For example:

- **Sales Trends**: The USA shows the highest sales growth in February, while Mexico struggles to keep pace in comparison. This might indicate the need for strategic interventions in Mexico.

- **Customer Engagement**: France and Canada show higher engagement during January and February, suggesting that marketing efforts in these regions have been more effective.

- **Inventory Levels**: Inventory levels in Mexico are consistently lower, signaling potential supply chain inefficiencies that could be impacting sales.

This country-level analysis can help the company make data-driven decisions to tailor its marketing, inventory management, and sales strategies according to regional performance.

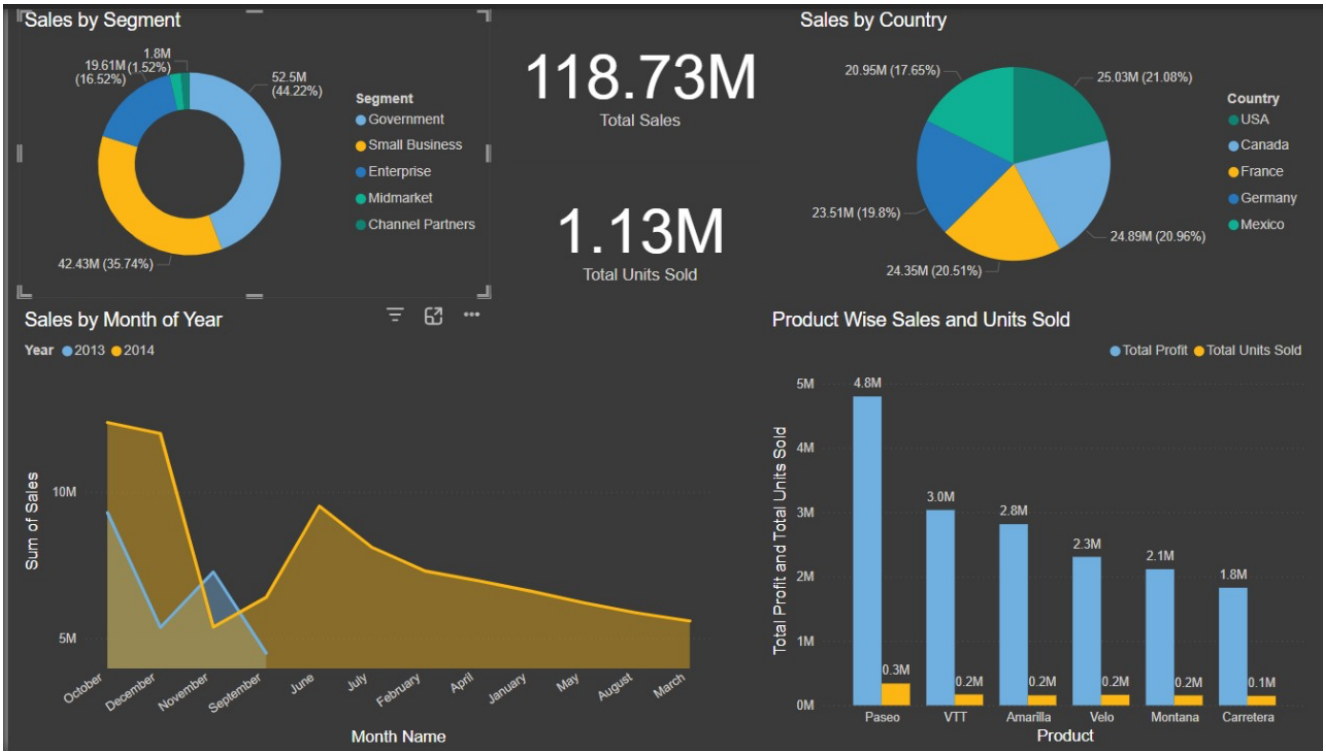
Benefits of Using Hadoop and Spark:

1. **Scalability**: Spark can handle massive datasets by distributing the processing across a cluster.

2. **Real-time Processing**: Spark's ability to process data in-memory speeds up the real-time processing of data streams, which can be critical for KPIs that need frequent updates.

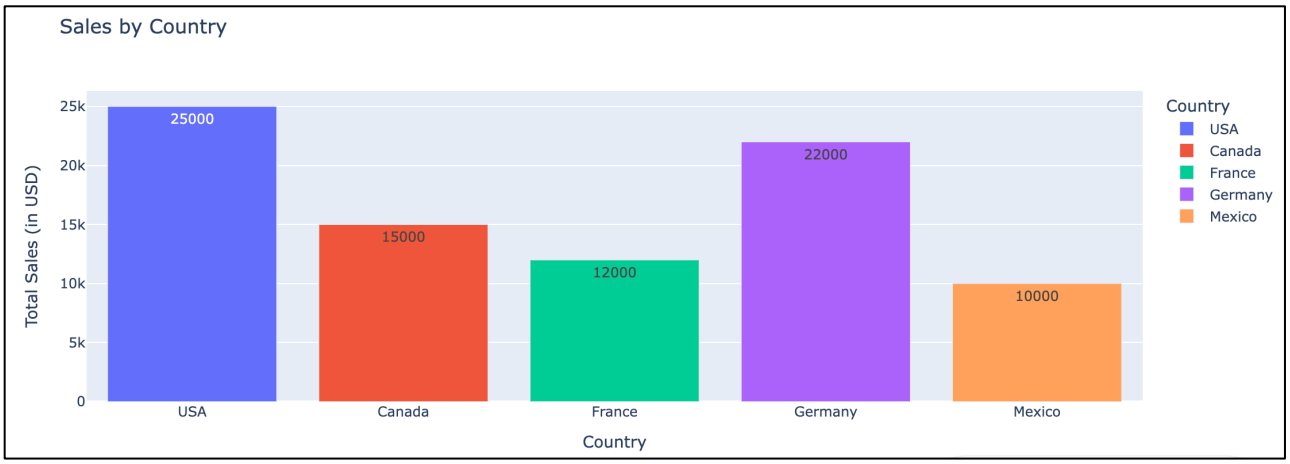
3. **Integration**: Spark integrates easily with Hadoop's HDFS, as well as other data sources such as Kafka, Cassandra, and AWS S3, making it ideal for handling diverse big data workloads.

Main Dashboard:

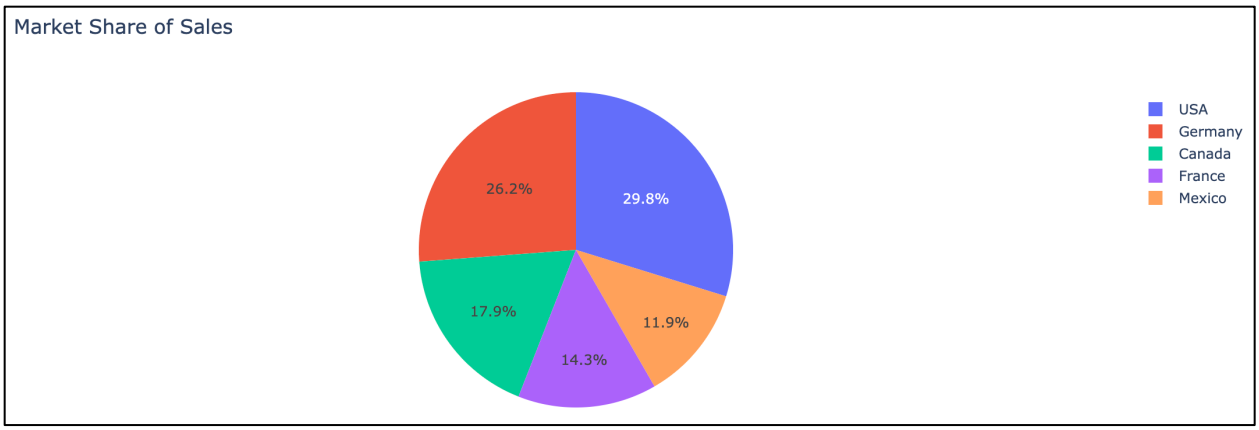


700 rows x 13 cols
Sales by Segment - Donut Chart
Total Sales & Units Sold - Cards
Sales by Country - Pie Chart
Sales by Month of year - Area Chart
Product Wise Sales and Units Sold - Clustered Column Chart

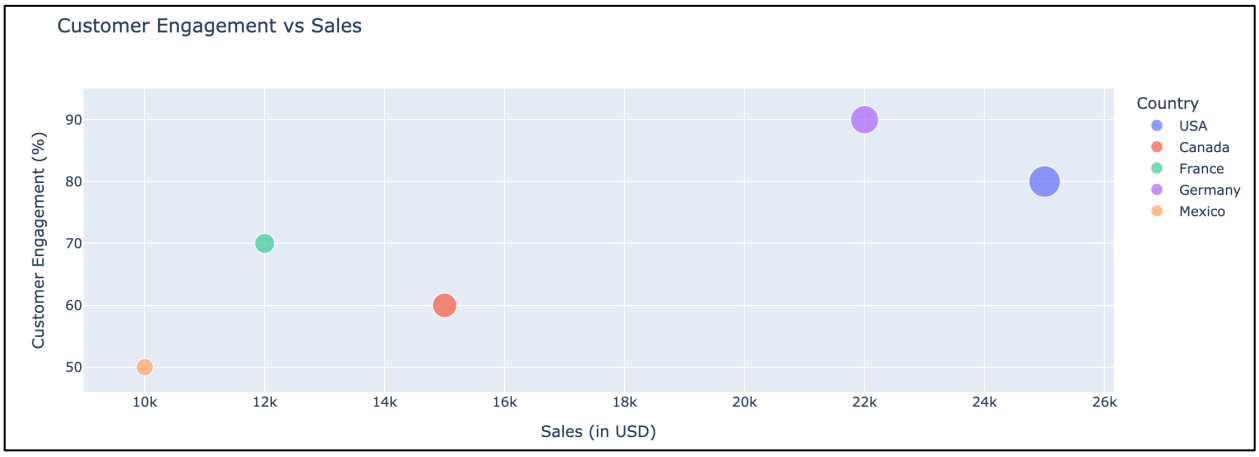
Bar Graph:



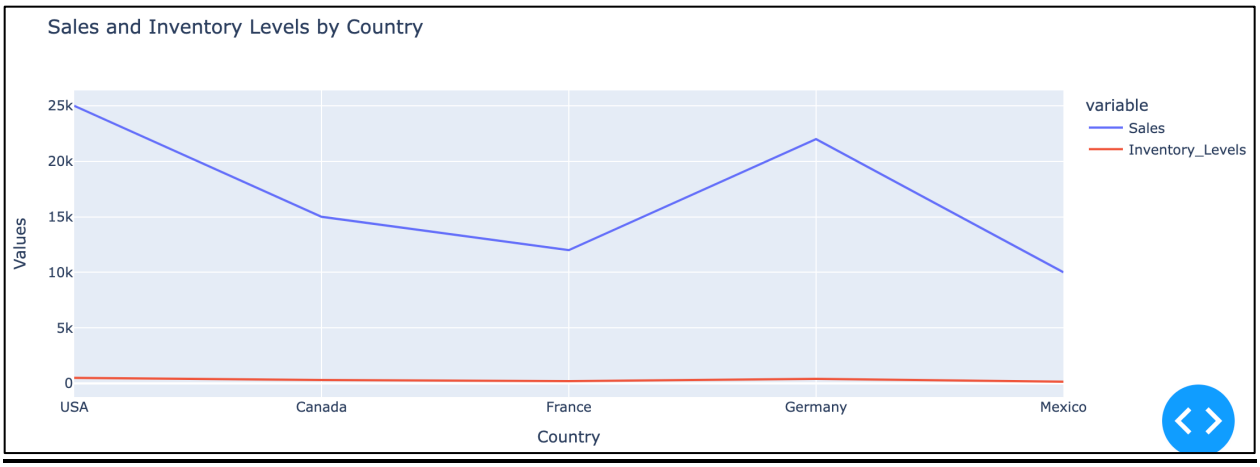
Pie Chart:



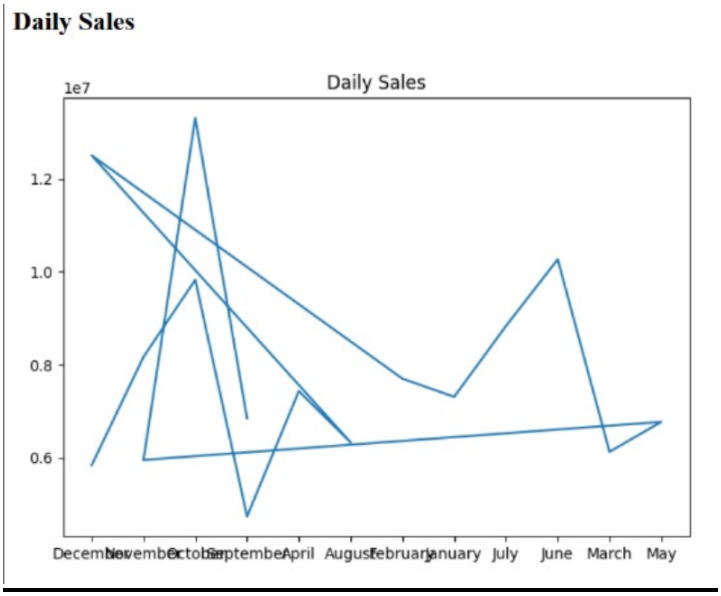
Scatter Plot Graph:



Line Graph:



Sales Dashboard:



Conclusion:

By integrating **Hadoop** and **Apache Spark** with a **Dash**-based front end, we have created a scalable, real-time executive dashboard capable of handling large datasets. This solution not only provides insights into key KPIs like sales, customer engagement, and inventory management but also ensures that businesses can monitor and adapt strategies as they grow.

The use of Spark in distributed data processing significantly enhances performance, ensuring the dashboard remains highly responsive, even with large and complex datasets. This approach supports scalability, real-time data processing, and interactive visualization, making it a powerful tool for decision-makers.

Moving forward, the system can be expanded to handle real-time data feeds using Spark Streaming and integrated with additional data sources (such as social media analytics for customer sentiment analysis). This will ensure that the dashboard continues to evolve as a key tool in the company's decision-making process.

This implementation is tailored for handling larger datasets using Hadoop and Spark while still maintaining the intuitive and interactive nature of the dashboard.

The proposed dashboard will entail the following benefits :

1. Improved decision-making: The executives will be given real-time, easily digestible data to assist in swift and informed decisions.
2. Operating efficiency will be enhanced: Now that the trends as well as anomalies have been identified quickly, timely interventions and optimization in most business functions can be made.
1. Efficient resource utilization: There is adequate and appropriate allocation of resources that have been generated. It is dependent on the clear sales trend and stock levels to ensure proper utility with the consumption levels by avoiding any waste.
3. Increase in customer satisfaction: The tracker of the customer engagement metrics will show the areas where there is a need for improvement in customer experience and service delivery.

Some of the potential challenges are as follows:

1. Data integration : To integrate the data coming from many sources, quite a lot of technical effort is required.
2. User adoption : Training and change management would be critically important to ensure that executives actually use all that is available on the dashboard.
3. Data security : Extremely good security measures would be required to safeguard the sensitive business information.
4. Ongoing maintenance : A lot of ongoing updates and maintenance with constant keeping up-to-date are required to make sure that the dashboard remains absolutely relevant and accurate.

The proposed executive dashboard will hence be an investment for companies in their data-driven futures. With such clarity and actionability in the insight of the key performance indicators, we should be able to grow with a lot of strength added to our competitive edge. This brings the step forward as an implementation plan, outlining timelines, resources allocated, and strategy for the user's training and adoption.