

Sales Data Analysis and Visualization Using Python, SQL, Excel, Power BI, and Streamlit

Shruti Raval

Department of Computer Science and Engineering
Parul University, Gujarat, India
ravalshruti5@gmail.com

Abstract—In the era of data-driven decision-making, organizations rely heavily on analytical systems to extract meaningful insights from large volumes of business data. Traditional reporting methods, which are often static and manual, fail to provide timely insights and lack the flexibility required for dynamic analysis. This paper presents a comprehensive Sales Data Analytics System that integrates Microsoft Excel, Python, SQL, Power BI, and Streamlit into a unified analytical framework. The system is designed to transform raw transactional data into structured, meaningful, and interactive insights through data preprocessing, storage, analysis, and visualization. Key performance indicators such as total revenue, net revenue, order count, average order value, category-wise contribution, and salesperson performance are generated and analyzed. The final output is deployed through a Streamlit-based web application, enabling real-time interaction and accessibility. The proposed system improves data accuracy, enhances visualization, and supports efficient business decision-making. The framework is scalable, modular, and suitable for both academic and practical applications in business analytics.

Index Terms—Sales Analytics, Data Visualization, Business Intelligence, Python, SQL, Excel, Power BI, Streamlit, Dashboard Systems

I. INTRODUCTION

In today's digital economy, data plays a crucial role in shaping business strategies and operational decisions. Organizations generate vast amounts of transactional data from daily activities such as product sales, customer interactions, and financial operations. However, the raw form of this data is often unstructured, inconsistent, and difficult to interpret without proper processing and analysis.

Traditional reporting methods, which rely heavily on spreadsheets and manual calculations, are limited in their ability to provide real-time insights and interactive exploration. These methods are not only time-consuming but also prone to errors and inconsistencies. Moreover, static reports fail to capture trends, patterns, and relationships within the data, which are essential for effective decision-making.

With advancements in data analytics technologies, modern tools such as Python, SQL, Power BI, and Streamlit provide powerful capabilities for data processing, storage, visualization, and deployment. By integrating these technologies, it is possible to build an end-to-end analytics pipeline that automates data handling and enhances insight generation.

This paper proposes a Sales Data Analytics System that combines multiple tools into a unified framework. The system enables users to upload raw sales data, process it efficiently,

store it in a structured format, and visualize it through interactive dashboards. The integration of these components improves analytical efficiency, enhances user experience, and supports better business decisions.

II. PROBLEM STATEMENT

Despite the availability of large volumes of sales data, many organizations still rely on traditional and manual methods for data analysis. These approaches present several challenges that limit the effectiveness of business intelligence systems.

Firstly, raw sales data often contains inconsistencies such as missing values, duplicate entries, and incorrect formatting. Without proper preprocessing, these issues can lead to inaccurate analysis and misleading conclusions. Secondly, the lack of structured data storage makes it difficult to perform efficient querying, aggregation, and filtering operations. This results in slower analysis and reduced scalability.

Additionally, traditional reporting systems generate static outputs that do not support user interaction. Business users are unable to dynamically explore data, apply filters, or drill down into specific insights. This limits their ability to understand trends, identify performance gaps, and make informed decisions.

Furthermore, the absence of a centralized analytics platform leads to fragmented workflows, where different stages of data processing, analysis, and visualization are handled separately. This increases complexity and reduces overall efficiency.

Therefore, there is a need for an integrated and automated sales analytics system that can handle data preprocessing, structured storage, KPI generation, interactive visualization, and web-based deployment in a seamless manner.

III. PROPOSED SYSTEM

The proposed Sales Data Analytics System is designed as a comprehensive and modular framework that integrates multiple technologies to provide an end-to-end solution for business data analysis. The system follows a layered architecture, where each layer performs a specific function in the data processing pipeline.

A. System Architecture

The architecture of the proposed system consists of the following layers:

- **Data Source Layer:** This layer is responsible for collecting raw sales data from various sources such as Excel files, CSV files, and manual business inputs. The data includes attributes such as product details, sales amount, order quantity, customer information, and transaction dates.
- **Data Processing Layer:** In this layer, Python is used to clean and preprocess the data. Tasks include handling missing values, removing duplicates, correcting inconsistencies, and performing data transformation. Libraries such as Pandas and NumPy are used to ensure that the data is structured and analysis-ready.
- **Database Layer:** The cleaned data is stored in an SQL database, where it is organized into structured tables such as sales, products, and customers. SQL enables efficient querying, filtering, and aggregation, which are essential for analytical operations.
- **Analysis Layer:** This layer computes key performance indicators (KPIs) such as total revenue, net revenue, number of orders, and average order value. It also performs category-wise analysis, salesperson performance evaluation, and trend analysis.
- **Visualization Layer:** Power BI is used to create interactive dashboards that display the analyzed data in the form of charts, graphs, and KPI cards. Users can apply filters and slicers to explore the data dynamically.
- **Deployment Layer:** The final system is deployed using Streamlit, which provides a web-based interface for users to upload data, view dashboards, and interact with analytical insights in real time.



Fig. 1. Detailed architecture of the Sales Data Analytics System.

B. System Workflow

The workflow of the proposed system is described as follows:

- 1) Raw sales data is collected from Excel or CSV files.
- 2) The data is cleaned and preprocessed using Python.
- 3) Processed data is stored in an SQL database.
- 4) SQL queries and Python scripts are used to generate KPIs and insights.
- 5) Power BI dashboards visualize the analytical results.
- 6) The system is deployed using Streamlit for user interaction.

The proposed system ensures a smooth flow of data across all layers, enabling efficient processing, accurate analysis, and effective visualization. It provides a scalable and user-friendly solution for sales data analytics.

C. System Architecture

The proposed system follows a layered architecture including data collection, preprocessing, storage, visualization, and deployment.

The system integrates multiple tools to ensure smooth data flow and efficient analysis.

IV. METHODOLOGY

The methodology of the proposed Sales Data Analytics System follows a structured pipeline that includes data collection, preprocessing, storage, analysis, and visualization. Each stage is designed to ensure data accuracy, consistency, and effective insight generation.

A. Data Collection

The initial stage of the system involves collecting raw sales data from multiple sources. The primary data sources include Microsoft Excel spreadsheets and CSV files, which are commonly used for storing transactional business data. The dataset typically contains attributes such as product name, category, order quantity, sales amount, customer information, salesperson details, and transaction dates.

The system is designed to support flexible data input formats, allowing users to upload files directly through the Streamlit interface. This feature enhances usability and enables non-technical users to easily provide input data without requiring backend access. The collected data serves as the foundation for subsequent preprocessing and analysis stages.

B. Data Cleaning and Preprocessing

Data preprocessing is a critical step in the analytics pipeline, as raw datasets often contain inconsistencies that can negatively impact analysis. In this system, Python is used for data cleaning and transformation, leveraging libraries such as Pandas and NumPy.

The preprocessing tasks include:

- Handling missing values by either removing incomplete records or filling them using appropriate techniques
- Identifying and removing duplicate entries to maintain data integrity
- Standardizing data formats such as dates, numerical values, and categorical labels
- Correcting inconsistent or incorrect data entries
- Converting data types to ensure compatibility with analytical operations

In addition to cleaning, data transformation techniques are applied to prepare the dataset for analysis. These include aggregation, feature creation, and normalization of categorical variables. The result of this stage is a clean, structured, and analysis-ready dataset.

C. Database Storage

After preprocessing, the cleaned data is stored in a structured SQL database. The use of SQL enables efficient management of large datasets and supports advanced querying capabilities. The database schema is designed to organize data into logical tables such as sales, products, and customers.

Structured storage provides several advantages:

- Efficient data retrieval using SQL queries
- Support for aggregation operations such as SUM, COUNT, and AVERAGE
- Improved scalability for handling larger datasets
- Better data organization and consistency

The database layer acts as a central repository, ensuring that processed data is readily available for analysis and visualization.

D. Data Analysis and KPI Computation

Once the data is stored in the database, analytical operations are performed to extract meaningful insights. Key performance indicators (KPIs) are computed using SQL queries and Python logic. These KPIs include:

- Total Revenue
- Net Revenue
- Total Number of Orders
- Average Order Value
- Category-wise Revenue Distribution
- Salesperson Performance Analysis
- Monthly Revenue Trends
- Top-Selling Products

These metrics provide a comprehensive view of business performance and help identify trends, patterns, and areas for improvement.

E. Visualization

The visualization stage is implemented using Microsoft Power BI, which enables the creation of interactive and user-friendly dashboards. The dashboards include various visual elements such as:

- KPI cards for quick performance overview
- Bar charts for category-wise and salesperson analysis
- Line charts for trend analysis over time
- Donut charts for proportional representation of categories
- Tables for detailed product-level insights

Power BI also provides features such as filters and slicers, allowing users to dynamically explore the data based on parameters such as date, category, and salesperson. This interactivity enhances data interpretation and supports better decision-making.

F. Deployment

To make the system accessible to end users, the final analytics solution is deployed using Streamlit. The Streamlit application serves as a web-based interface that integrates data upload functionality with dashboard visualization.

Users can upload datasets, trigger data processing, and view analytical results directly through the browser. This deployment approach eliminates the need for complex installation and ensures ease of use. Additionally, it provides a scalable platform for extending the system with advanced features such as real-time analytics and machine learning models in the future.

V. SYSTEM DESIGN

A. Upload Interface



Fig. 2. Streamlit-based upload interface for sales data.

The upload interface is designed using Streamlit to provide a simple and user-friendly mechanism for data ingestion. It acts as the entry point of the system, allowing users to upload sales data files in formats such as CSV and Excel. The interface is implemented with a clean layout and sidebar navigation, ensuring ease of use even for non-technical users.

The system validates the uploaded file and ensures that it meets the required format before processing. Once the file is uploaded, the backend pipeline is triggered automatically, initiating data preprocessing, storage, and analysis. This eliminates the need for manual intervention and significantly improves efficiency.

The upload module enhances usability and accessibility by enabling real-time interaction with the analytics system. It ensures that users can quickly input data and obtain insights without requiring knowledge of programming or database operations.

VI. RESULTS AND DISCUSSION

A. Upload Interface Results

The implementation of the upload interface demonstrates the system's capability to handle user input dynamically. The interface successfully accepts multiple file formats and initiates automated data processing. The responsiveness and simplicity of the interface improve the overall user experience.

The results show that the upload module reduces the complexity of data entry and ensures that users can seamlessly transition from raw data input to analytical output. This feature is particularly beneficial in real-world scenarios where business users frequently update datasets.

B. Dashboard Analysis



Fig. 3. Streamlit dashboard showing KPIs and insights.



Fig. 4. Power BI dashboard visualization of sales performance.

The dashboard represents the core analytical output of the system, providing a comprehensive view of sales performance through interactive visualizations. The dashboard integrates

key performance indicators (KPIs) and graphical representations to support data-driven decision-making.

The Streamlit dashboard presents summarized insights, while the Power BI dashboard offers detailed visual analytics. The integration of both platforms ensures both accessibility and analytical depth.

The dashboard includes the following key components:

- **KPI Cards:** Display total revenue, net revenue, number of orders, and average order value, providing a quick overview of business performance.
- **Category-wise Analysis:** Visual representation of revenue contribution by different product categories, helping identify high-performing segments.
- **Salesperson Performance:** Bar charts showing the contribution of individual salespersons, enabling performance comparison.
- **Monthly Trends:** Line charts representing revenue trends over time, useful for identifying patterns and seasonality.
- **Top Products:** Tabular representation of best-selling products based on revenue and quantity.

The interactive filters and slicers allow users to explore the data dynamically by selecting parameters such as date, category, and salesperson. This enhances analytical flexibility and improves user engagement.

C. Key Findings

The analysis of the dashboard reveals several important business insights:

- **Category Dominance:** Ethnic wear contributes the highest share of total revenue, indicating strong customer demand in this segment.
- **Trend Analysis:** Revenue trends vary across months, suggesting the presence of seasonal patterns or fluctuations in demand.
- **Salesperson Variation:** Performance differs significantly among salespersons, highlighting the need for targeted performance evaluation and incentive strategies.
- **Product Contribution:** A small number of products contribute significantly to overall revenue, indicating a concentration of sales among top-performing items.

These findings demonstrate the effectiveness of the system in identifying patterns, trends, and performance gaps.

D. Discussion

The results indicate that the proposed system successfully transforms raw sales data into meaningful insights through an integrated analytics pipeline. The combination of preprocessing, structured storage, visualization, and deployment enables efficient and accurate analysis.

The Streamlit interface enhances accessibility, while Power BI provides advanced visualization capabilities. Together, they create a powerful analytical environment that supports both technical and non-technical users.

The system reduces dependency on manual reporting and improves decision-making by providing real-time, interactive insights. It also demonstrates scalability, as the architecture can

be extended to handle larger datasets and additional analytical features.

VII. ADVANTAGES AND LIMITATIONS

A. Advantages

The proposed Sales Data Analytics System provides several significant advantages by integrating multiple technologies into a unified analytical framework.

- **Interactive and Dynamic Dashboard:** The system offers highly interactive dashboards developed using Power BI and Streamlit. Users can apply filters, slicers, and drill-down operations to explore data dynamically. This interactivity enhances data understanding and allows users to analyze specific segments such as categories, salespersons, and time periods without requiring manual calculations.
- **Improved Decision-Making:** The system provides real-time access to key performance indicators (KPIs) such as total revenue, net revenue, order count, and average order value. These metrics enable business users and decision-makers to quickly evaluate performance, identify trends, and make informed strategic decisions.
- **Integrated End-to-End Pipeline:** One of the major strengths of the system is the integration of data collection, preprocessing, storage, analysis, visualization, and deployment into a single pipeline. This reduces the complexity of managing separate tools and ensures a seamless flow of data across different stages.
- **Enhanced Data Quality:** The use of Python-based preprocessing techniques ensures that the dataset is cleaned and validated before analysis. Handling missing values, removing duplicates, and standardizing data formats improves the accuracy and reliability of the results.
- **User-Friendly Interface:** The Streamlit-based web interface simplifies user interaction by providing an intuitive platform for uploading data and viewing results. Even users with limited technical knowledge can operate the system efficiently.
- **Scalability and Flexibility:** The modular architecture allows the system to be easily extended with additional features such as machine learning models, predictive analytics, and real-time data integration. This makes the system adaptable to future requirements.
- **Time Efficiency:** Automation of data processing and reporting significantly reduces the time required for analysis compared to manual methods. This improves productivity and allows faster insight generation.

B. Limitations

Despite the advantages, the proposed system has certain limitations that need to be considered for further improvement:

- **Limited Real-Time Data Processing:** The current implementation primarily relies on static datasets uploaded by users. It does not support real-time data streaming or continuous data updates, which are essential for modern enterprise-level applications.

- **Performance Constraints with Large Datasets:** As the size of the dataset increases, the performance of data processing and dashboard rendering may degrade. This can lead to slower response times, especially when handling large-scale business data.
- **Basic Security Implementation:** The system includes only fundamental security measures. Advanced features such as user authentication, role-based access control, data encryption, and secure cloud deployment are not fully implemented in the current version.
- **Dependency on Data Quality:** The accuracy of analytical results depends heavily on the quality of the input data. If the uploaded dataset contains incorrect or incomplete information, it may affect the reliability of insights despite preprocessing efforts.
- **Limited Automation in Data Integration:** The system requires manual data upload, which may not be suitable for organizations that require automated data pipelines connected to live databases or APIs.
- **Lack of Advanced Predictive Analytics:** The current system focuses primarily on descriptive analytics. Predictive modeling and forecasting capabilities are not yet integrated, limiting the ability to anticipate future trends.

VIII. CONCLUSION

This paper presented a comprehensive Sales Data Analytics System that integrates Microsoft Excel, Python, SQL, Power BI, and Streamlit into a unified and efficient analytical framework. The system is designed to transform raw sales data into structured and meaningful insights through a well-defined pipeline involving data collection, preprocessing, database storage, KPI computation, visualization, and deployment.

The implementation of an interactive upload interface and dynamic dashboards significantly enhances usability and accessibility. Users can easily input data, explore key performance indicators, and analyze business performance through intuitive visualizations. The integration of multiple technologies ensures efficient data handling and provides a scalable solution for both academic and real-world applications.

The results demonstrate that the system effectively improves data interpretation, reduces manual effort, and supports data-driven decision-making. It enables businesses to identify trends, evaluate product and salesperson performance, and monitor overall financial health.

Although the system has certain limitations, such as lack of real-time processing and advanced predictive capabilities, it provides a strong foundation for further development. Future enhancements can include machine learning integration, real-time data streaming, cloud deployment, and advanced security mechanisms.

Overall, the proposed system represents a practical and scalable approach to sales data analytics, highlighting the importance of integrating modern data tools to achieve efficient and insightful business intelligence solutions.

REFERENCES

- [1] Microsoft, "Power BI Documentation," Microsoft Learn, 2024. [Online]. Available: <https://learn.microsoft.com/en-us/power-bi/>. [Accessed: Apr. 22, 2026].
- [2] Oracle Corporation, "MySQL Documentation," Oracle, 2024. [Online]. Available: <https://dev.mysql.com/doc/>. [Accessed: Apr. 22, 2026].
- [3] W. McKinney, *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2017.
- [4] Microsoft, "Microsoft Excel Documentation," Microsoft Support, 2024. [Online]. Available: <https://support.microsoft.com/excel>. [Accessed: Apr. 22, 2026].
- [5] Streamlit Inc., "Streamlit Documentation," 2024. [Online]. Available: <https://docs.streamlit.io/>. [Accessed: Apr. 22, 2026].
- [6] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*, New York, NY, USA: Springer, 2013.
- [7] S. Few, *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*, 2nd ed. Burlingame, CA, USA: Analytics Press, 2013.