International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

SuperStore Sales Insights Analysis Dashboard System

Mareedu Kusuma Sri
PG MCA Scholar, Department of Computer Science
Sir C R Reddy College, Eluru, Andhra Pradesh, India-534005
kusumasrimareedu3@gmail.com

Mehaboob Karishma
Assistant Professor, MCA Department
Sir C R Reddy College Autonomous PG Courses, Eluru, Andhra Pradesh, India-534005
mehaboob.karishma@gmail.com

Musunuru Ratnakar
HOD, Department of Computer Science
Sir C R Reddy College, Eluru, Andhra Pradesh, India-534005

Abstract*— Retail transaction records contain valuable evidence about revenue, customer behaviour, product demand, shipping efficiency and regional performance, yet raw CSV exports are difficult to interpret without a reproducible analytical workflow. This paper presents the SuperStore Sales Insights Analysis Dashboard System, an end-to-end Business Intelligence solution integrating Python, MySQL and Microsoft Power BI. Pandas and NumPy clean 9,994 order-line records spanning 2015–2018 by correcting date types, removing incomplete postal-code records, eliminating a duplicate transaction and validating order-to-shipment consistency. The cleaned data are stored in MySQL, where fourteen analytical queries use aggregation, common table expressions and window functions to answer business questions. Power BI then provides coordinated Sales, Customer and Product dashboards driven by reusable DAX measures and interactive slicers. The system reports approximately $2 million in revenue across 4,916 orders, 793 customers and 1,860 products. Technology contributes 36.67% of sales, followed by Furniture at 32.11% and Office Supplies at 31.22%. The completed workflow demonstrates traceable, decision-ready retail analytics for non-technical stakeholders.

Keywords*— Power BI, Python, SQL, Data Analytics, Business Intelligence, Sales Analysis, Dashboard, Data Visualization.

I. INTRODUCTION

Retail organizations continuously generate order, customer, product, location, shipping and monetary records. These records can reveal sales trends, high-value customers, regional demand and product concentration, but raw transaction files do not communicate such findings directly. When reporting is performed through manually maintained spreadsheets, analysts must repeatedly correct data types, reconstruct formulas and create separate pivot tables for every new question. This increases turnaround time, makes calculations difficult to audit and limits the ability of managers to explore several dimensions simultaneously.

The proposed work replaces this fragmented process with a three-stage analytical pipeline. Python provides deterministic data preparation; MySQL offers a structured repository and reusable business queries; and Power BI converts the validated data into interactive dashboards. The stages are connected through a stable schema so that every Key Performance Indicator (KPI) can be traced from its visual representation to a DAX expression, an SQL aggregation and a cleaned source column. This traceability is important because visually attractive dashboards are not useful when their underlying calculations cannot be verified.

The Superstore dataset represents United States retail transactions recorded between 2015 and 2018. It includes customer segments, geographic attributes, product categories, order and shipment dates, identifiers and sales values. The project analyses three complementary perspectives. The Sales dashboard summarizes revenue, order volume, city rankings, category contribution and temporal change. The Customer dashboard studies customer count, repeat purchasing, segment composition and regional customer value. The Product dashboard highlights category performance, product trends and the highest-selling items.

The principal contribution is not a new predictive algorithm; it is a complete and auditable Business Intelligence implementation that joins data-quality control, advanced SQL analysis and interactive visual exploration. Fourteen SQL queries answer questions involving rankings, cumulative values, shipment duration, regional dominance and rising-sales streaks. Twelve validation cases verify data integrity, formula consistency and slicer behaviour. The system therefore demonstrates a realistic workflow for data analysts and BI developers while remaining accessible to decision makers who do not write Python, SQL or DAX.

II. RELATED WORK

Retail analytics draws on data cleaning, relational database design, dimensional modelling and information visualization. Pandas and NumPy support vectorized type conversion, null handling, duplicate detection and tabular transformation [1]–[3]. Relational database principles emphasize explicit schemas, consistent data types and reusable queries [4], [5], while dimensional modelling organizes business facts and descriptive attributes for analytical workloads [6], [7]. These foundations reduce ambiguity between a transaction stored in a source file and the measure displayed to a business user.

Dashboard design research recommends a compact set of clearly labelled KPIs, consistent encodings and details-on-demand rather than overcrowded pages [8]–[11]. Power BI applies these principles through a semantic data model, DAX measures, slicers and cross-filtering [12]–[14]. MySQL supplies aggregations, common table expressions and window functions for ranked or sequential analysis [15]. When these technologies are used together, the dashboard becomes the final interface of a governed data pipeline rather than an isolated visualization file.

Many public Superstore demonstrations emphasize only one stage: a Power BI report built directly from a CSV file, a notebook containing exploratory charts or a small set of SQL queries. Data cleaning is often undocumented, customer-level measures may be absent and calculations shown on different pages may not share a common definition. Such implementations are useful for learning a tool but offer

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

limited evidence that the final figures are reproducible. The present system addresses this gap by documenting the complete path from raw data to cleaned CSV, MySQL table, analytical query, DAX measure and dashboard visual.

The project also adopts a deliberate denormalized staging table. Although normalized schemas reduce update anomalies in transactional systems, a read-oriented analytical table can simplify aggregation and Power BI import when the dataset is moderate and refreshed as a controlled batch. The design retains conceptual customer, order, product, location and sales entities while using one physical table for efficient reporting. This choice is consistent with the separation between operational database design and analytical presentation described in data-warehouse literature [6], [19].

III. MATERIALS AND METHODS

The methodology consists of dataset acquisition, structural inspection, cleaning, database loading, SQL analysis, DAX measure construction, dashboard design and validation. The same cleaned dataset supports every analytical page, preventing independent calculations from drifting apart. Fig. 1 shows the project architecture and the flow of information from the source CSV to the three business dashboards.

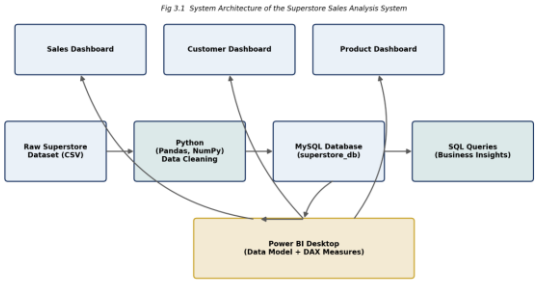


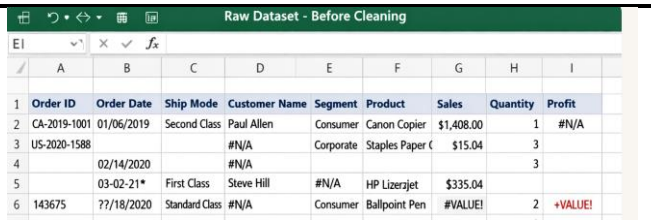
Fig. 1. System architecture of the Superstore sales analysis system.

The architecture separates data preparation, storage, analysis and presentation. Python receives the raw file and exports a type-corrected, de-duplicated dataset. MySQL preserves the cleaned schema and executes the business queries. Power BI imports the validated table, applies shared DAX measures and presents Sales, Customer and Product views. Because each layer has a defined input and output, data-quality errors can be located before they propagate to management reports.

A) Dataset Collection

The Superstore Sales Dataset is a public tabular retail dataset containing 9,994 transaction lines and 18 business attributes. Each row represents an order line rather than a complete customer order; consequently, the Order_ID can occur in several records when one order contains multiple products. The data span 2015–2018 and represent 4,916 distinct orders, 793 customers, 1,860 products and three principal categories. Important fields include Order_Date, Ship_Date, Ship_Mode, Customer_ID, Customer_Name, Segment, City, State, Postal_Code, Region, Product_ID, Category, Sub_Category, Product_Name and Sales.

The raw file is loaded into a Pandas DataFrame and inspected using head(), shape(), info() and column-level summaries. This inspection establishes the row granularity, identifies text-based date fields, reveals missing Postal_Code values and confirms the presence of duplicate records. The dataset image in Fig. 2 provides a readable sample of the original transaction structure used by the project.



	A	B	C	D	E	F	G	H	I
1	Order ID	Order Date	Ship Mode	Customer Name	Segment	Product	Sales	Quantity	Profit
2	CA-2019-1001	01/06/2019	Second Class	Paul Allen	Consumer	Canon Copier	$1,408.00	1	#N/A
3	US-2020-1588			#N/A	Corporate	Staples Paper (	$15.04	3	
4		02/14/2020		#N/A				3	
5		03-02-21*	First Class	Steve Hill	#N/A	HP Lizerjet	$335.04		
6	143675	??/18/2020	Standard Class	#N/A	Consumer	Ballpoint Pen	#VALUE!	2	+VALUE!

Fig. 2. Raw Superstore dataset

Table I. Dataset summary

Item	Value
Records	9,994
Columns	18
Period	2015–2018
Orders	4,916
Customers	793
Products	1,860
Categories	3

The dataset is appropriate for descriptive BI because it combines time, customer, product and location dimensions with a numerical sales measure. It does not include streaming updates or a separate profitability target, so the present analysis focuses on historical revenue, demand and customer behaviour. This boundary is retained throughout the paper to avoid treating descriptive dashboard outputs as forecasts.

B) Data Pre-processing

Data cleaning is performed as a fixed sequence so that repeated execution produces the same output. Order_Date and Ship_Date are converted from text to datetime values, enabling chronological sorting and shipment-duration calculations. Rows with missing Postal_Code values are removed because a fabricated postal code would distort city and state aggregation. Postal_Code is then cast to integer. Fully duplicated transaction lines are detected across the business columns and one duplicate is deleted to prevent double counting.

Logical validation confirms that Ship_Date is never earlier than Order_Date. After cleaning, the DataFrame index is reset and the result is exported as superstore_sales_cleaned.csv. The process protects later computations from three common defects: duplicate revenue, invalid time intervals and incomplete geographic grouping. The cleaned row count is compared with the imported MySQL row count as an additional integrity check.

C) SQL Analysis and Feature Construction

The cleaned CSV is loaded into the MySQL table superstore. Fourteen queries answer specific business questions, including same-day shipping percentage, the highest-value customers, top items by average daily sales, customer ranking by average order value, city-level order extremes, dominant sub-categories, shipping-mode preferences, average order-to-shipment time and the longest sequence of rising daily sales. Simple questions use GROUP BY and aggregate functions, while ranking and sequential questions use common table expressions, DENSE_RANK(), ROW_NUMBER(), RANK() and LAG().

The queries serve two purposes. First, they produce actionable standalone insights for analysts. Second, they provide independent values against which Power BI measures can be checked. The following equations define the main dashboard indicators, where n is the number of order lines, c is a category and distinct counts are evaluated in the active filter context.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

$$Total\ Sales = \Sigma_{i=1}^{n} Sales_i \quad (1)$$

$$Average\ Order\ Value = Total\ Sales / Total\ Orders \quad (2)$$

$$CLV = Total\ Sales / Total\ Customers \quad (3)$$

$$RCR = Repeat\ Customers / Total\ Customers \quad (4)$$

$$Share\_c = (Sales\_c / Total\ Sales) \times 100 \quad (5)$$

Equation (1) aggregates line-level revenue. Equation (2) normalizes sales by distinct orders, while (3) expresses sales relative to the distinct customer population. Equation (4) measures the proportion of customers associated with more than one order, and (5) converts each category contribution into a percentage of total sales. These definitions are implemented once and reused across visuals to prevent inconsistent KPI logic.

D) Power BI Dashboard Development

The cleaned table is imported into Power BI Desktop and enriched with DAX measures. Total Sales uses SUM(Sales); Total Orders, Total Customers and Total Products use DISTINCTCOUNT on their identifiers; Average Order Value divides Total Sales by Total Orders; and category or regional values are evaluated through the current filter context. Reusable measures are preferred to visual-level calculations because they provide one governed definition for all report pages.

Table II. Representative measures

Measure	DAX definition
Total Sales	SUM(Sales)
Total Orders	DISTINCTCOUNT (Order_ID)
AOV	Total Sales / Total Orders
Customers	DISTINCTCOUNT (Customer_ID)
CLV	Total Sales / Customers
Products	DISTINCTCOUNT (Product_ID)

Year, Month, Region and City slicers apply to the relevant visuals and enable interactive cross-filtering. The Sales page emphasizes revenue, order volume, city ranking, category share and time trend. The Customer page displays customer count, customer value, repeat purchasing and segment structure. The Product page compares categories, product trends and the strongest individual products. A shared visual language, consistent slicer positions and page-navigation buttons reduce cognitive effort when users move between analytical perspectives.

E) Dashboard Design and Interactivity

Each page begins with a small set of headline KPI cards followed by supporting charts. Bar charts are used for ranked city and product comparisons, donut charts communicate category or segment composition, and line or stacked-column charts show temporal change. This arrangement follows the overview-first principle: the user sees the current business state immediately and can then inspect the dimensions that explain it. Selecting a year, month, region or city updates all connected visuals without editing SQL or DAX.

Interactivity is treated as an analytical function rather than decoration. A manager can isolate a region, identify its leading cities, compare category composition and then move to the customer or product page while preserving the interpretation of shared measures. The coordinated pages therefore replace several disconnected static reports with one exploratory environment.

F) Validation Procedure

Validation covers the complete pipeline. Python outputs are checked for missing values, duplicate records, corrected data types and valid order-to-shipment chronology. MySQL row counts are matched with the cleaned CSV, and representative query outputs are manually recalculated. Power BI KPI values are compared with SQL aggregates, category percentages are checked to sum to the whole and slicers are tested for consistent cross-filtering. Twelve functional and data-integrity cases pass, including CSV import, null removal, duplicate removal, date validation, KPI accuracy, page navigation and repeat-customer calculation.

The validation strategy does not treat a dashboard screenshot as proof of correctness. Instead, the numerical path is tested at each boundary between Python, MySQL and Power BI. This layered verification improves confidence that changes introduced during cleaning, import or visualization have not altered the intended business meaning.

IV. EXPERIMENTAL RESULTS

A) Sales Dashboard Analysis

The Sales Dashboard reports approximately $2 million in sales, 4,916 distinct orders and an average order value of $230. New York City is the leading revenue location at about $252K, followed by Los Angeles at $173K and Seattle at $116K. Technology contributes 36.67% of sales, Furniture 32.11% and Office Supplies 31.22%. The close category shares indicate a diversified product portfolio, whereas the city ranking reveals stronger geographic concentration.

Fig. 3. Sales Dashboard—final Power BI output.

Table III. Sales dashboard findings

Sales KPI	Result
Total Sales	≈ $2.0M
Total Orders	4,916
Average Order Value	$230
Top City	New York City $252K
Top Category	Technology 36.67%
Peak Month	≈ $118K

The time-series visual shows an overall increase from 2015 to 2018 with periodic fluctuations and a peak monthly value near $118K. Such variation can support inventory planning and campaign timing, but it should not be interpreted as a forecast because the report is descriptive. The city chart suggests that marketing and stock allocation may reasonably prioritize major metropolitan areas while investigating lower-performing locations through the same slicers.

The dashboard combines executive summary and detailed exploration. KPI cards answer immediate questions, while city, category and time visuals reveal the composition behind the totals. Cross-filtering makes it possible to isolate a selected year, region or city without modifying the model, and the resulting values remain consistent because all visuals use the same underlying measures.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

B) Customer Dashboard Analysis

The Customer Dashboard identifies 793 customers, a reported customer lifetime value of $335K and a repeat-customer rate of 0.99. Consumers form 51.58% of the customer population, Corporate customers 29.76% and Home Office customers 18.66%. Regional customer values are comparatively close, ranging from approximately $0.32M to $0.35M. The dashboard therefore distinguishes broad customer retention from local segment differences.

Fig. 4. Customer Dashboard—final Power BI output.

Table IV. Customer dashboard findings

Customer KPI	Result
Total Customers	793
Customer Lifetime Value	$335K
Repeat Customer Rate	99%
Consumer Segment	51.58%
Corporate Segment	29.76%
Home Office Segment	18.66%

A 99% repeat rate indicates that most customers place more than one order within the four-year historical period. The measure is valuable for retention analysis, although it should be interpreted in the context of the long observation window: a customer making two purchases across four years is classified as repeat. Segment charts add important detail. Consumers dominate most cities, while localized exceptions, such as a comparatively high Home Office share, can be examined through city filters.

Regional customer values are closely grouped. Consequently, differences in total regional sales appear to be influenced more by customer volume and city concentration than by extreme changes in value per customer. Management can combine acquisition efforts in lower-volume locations with retention programs across all segments. Because the dashboard does not include marketing cost or margin data, these actions should be prioritized together with external profitability information in operational use.

Customer outputs are cross-checked against DISTINCTCOUNT(Customer_ID), the configured value measures and the distinct-order condition used for repeat customers. Segment percentages are verified against the complete customer population under the same filter context. This consistency confirms that the page presents a connected analytical view rather than unrelated visual calculations.

C) Product Dashboard Analysis

The Product Dashboard evaluates 1,860 distinct products. Technology generates approximately $826K, Furniture $723K and Office Supplies $703K. The Canon imageCLASS 2200 Advanced Copier is the strongest individual product at approximately $61.6K, more than twice the revenue of the next-ranked item. The result indicates that balanced category-level revenue can coexist with substantial concentration at the product level.

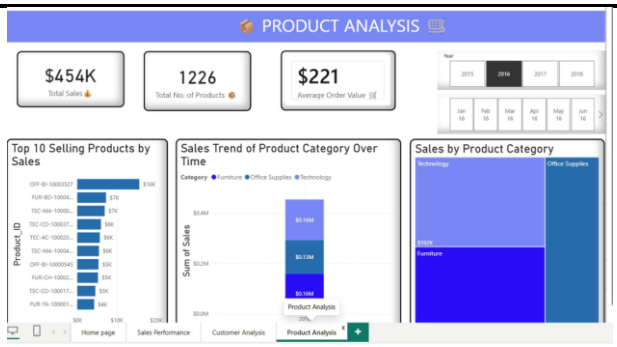


Fig. 5. Product Dashboard—final Power BI output.

Table V. Product dashboard findings

Product KPI	Result
Total Products	1,860
Technology Sales	$826K
Furniture Sales	$723K
Office Supplies	$703K
Top Product	Canon imageCLASS $61.6K

Office Supplies has the smallest total category share, yet several binder products appear among the top-selling items. This pattern shows why category and product rankings must be interpreted together: a broad category can be slightly smaller overall while containing several consistently demanded products. The leading copier is a clear revenue outlier and may require dedicated inventory, supplier and risk monitoring because stock disruption would affect a disproportionate amount of sales.

The product trend visual supports comparison of category movement over time, while the ranking chart identifies individual contributors. Through Year, Month, Region and City slicers, analysts can determine whether a leading product is consistently strong or dependent on a limited period or location. These interactions turn a static top-ten list into a diagnostic business tool.

D) Overall Discussion

Across the three pages, the principal totals remain internally consistent. Category shares reconcile with total sales, customer and product counts match distinct SQL counts, and filtered subsets retain the same measure definitions. The integrated design therefore avoids duplicated calculations and enables users to move from an executive overview to customer or product detail without changing data sources.

The analysis reveals three complementary business patterns. First, revenue is diversified across categories but concentrated in a small number of cities. Second, customer retention is high over the observed period, with Consumers forming the largest segment. Third, category balance masks strong product-level outliers. These findings support targeted inventory allocation, regional marketing, customer-retention planning and closer monitoring of high-impact products.

The results are subject to limitations. The dataset is static and historical, the report is descriptive rather than predictive, and the available sales field does not by itself represent profit. Refresh is manual, the Power BI file is desktop-based and no row-level security is implemented. These constraints define the system as a reproducible analytical prototype rather than a live enterprise reporting service.

E) Business Implications

The integrated dashboard supports several levels of decision making. At the executive level, Total Sales, Total Orders and Average Order Value provide an immediate view

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

of commercial scale. At the regional level, city rankings and slicers reveal where revenue is concentrated and where market development may be required. At the operational level, category and product rankings assist inventory prioritization, while shipment-duration queries can highlight process performance. Customer measures support retention planning by distinguishing overall customer volume from repeat purchasing and segment composition.

The value of these insights depends on their coordinated interpretation. A high-sales city may also contain a narrow customer or product mix, increasing exposure to local demand changes. A category with a balanced overall share may depend on one exceptional product, creating supply risk. Likewise, a high repeat rate may reflect the length of the observation period rather than frequent recent engagement. The dashboard enables these relationships to be explored together instead of relying on isolated headline numbers.

For organizational use, the pipeline can serve as a governed reporting pattern. Python scripts form the repeatable data-quality layer, MySQL queries provide reviewable analytical logic and DAX measures define the user-facing semantics. Assigning ownership to each layer would allow changes in a KPI to be traced, reviewed and tested before publication. This governance principle is as important as the visual design because it preserves confidence when new records or business rules are introduced.

F) Limitations and Validity

The study uses one public Superstore dataset and therefore demonstrates technical capability rather than the performance of a specific real retailer. The transaction period ends in 2018, economic conditions are not represented and the source does not provide a complete operational context. Results such as city leadership, category share and customer repetition are valid for the supplied records but should not be generalized to other markets without independent data. The analysis also focuses on sales value; a product can generate high revenue while producing low margin after discount, logistics and service costs.

Internal validity is strengthened by deterministic cleaning, row-count reconciliation and comparison between SQL and Power BI values. Nevertheless, manual data refresh creates the possibility of version mismatch if a cleaned CSV, database table and report file are not updated together. A production system should record source-file versions, refresh timestamps, row counts and validation status. Automated checks should stop publication when required columns change, dates fail validation or dashboard totals do not reconcile with the database.

The present dashboard is designed for interactive desktop analysis and does not evaluate concurrent usage, access control or network latency. It also does not test whether business users interpret every visual correctly. Future evaluation can include usability studies, task-completion time, error rates and stakeholder feedback. Such assessment would complement technical testing and determine whether the arrangement of cards, charts and slicers supports reliable decisions in practice.

V. CONCLUSION

This paper presented the SuperStore Sales Insights Analysis Dashboard System, a complete Python–MySQL–Power BI workflow for transforming raw retail transactions into validated business information. Python standardizes dates, removes incomplete and duplicate records and produces an analysis-ready dataset. MySQL stores the cleaned data and supports fourteen reusable queries using aggregation, common table expressions and window functions. Power BI exposes the results through coordinated Sales, Customer and Product dashboards with shared DAX measures and interactive slicers.

The system converts 9,994 transaction lines into interpretable indicators, including approximately $2 million in revenue, 4,916 orders, 793 customers and 1,860 products. Technology contributes 36.67% of revenue, New York City is the leading city, the repeat-customer rate is 99% over the historical period and the top copier is a major product-level outlier. Twelve validation cases confirm the integrity of cleaning, loading, calculations and interaction. The project therefore meets its objective of replacing manual spreadsheet reporting with an auditable and accessible decision-support dashboard.

Future development will automate the ETL refresh, publish the report through Power BI Service, add row-level security and introduce forecasting. RFM-based customer segmentation, profitability measures, external marketing or economic data and natural-language Q&A can extend the analytical depth. A production implementation should also define scheduled quality checks, controlled measure ownership and monitoring for changes in source structure so that the traceability established by the prototype is preserved as the system evolves.

A scheduled workflow can execute the Python cleaning script, load validated records into MySQL and trigger a Power BI refresh without manual file handling. Incremental loading and audit columns would distinguish new, changed and rejected records. Publishing through Power BI Service would enable controlled sharing, refresh monitoring and mobile access, while row-level security could restrict regional managers to authorized data. These changes would convert the current desktop report into a maintainable multi-user service.

Predictive extensions should be introduced only after data governance is stable. Time-series models can estimate future sales and seasonal demand, while RFM features can group customers by recency, frequency and monetary value. Forecast accuracy, confidence intervals and drift monitoring would need to be reported separately from descriptive KPIs. Combining external marketing expenditure, economic indicators or inventory availability with the existing dataset could further explain why sales change rather than merely showing when the change occurred.

Overall, the project demonstrates that carefully connected tools can provide more value than an isolated dashboard. The Python, MySQL and Power BI stages form a traceable analytical chain in which preparation, computation and presentation remain distinct but mutually verifiable. This structure provides a practical foundation for future retail intelligence systems that require transparent measures, interactive exploration and reproducible data quality.

A) Deployment Roadmap

A practical deployment can be organized as a controlled sequence of ingestion, validation, publication and monitoring. The ingestion task receives a dated source file or database extract and stores it in a protected landing area. The Python cleaning module then applies the same schema checks used in the prototype, records rejected rows and writes a

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

validated output. Only records that satisfy mandatory field, date and duplicate rules are loaded into MySQL. This prevents the dashboard from refreshing against partially processed or structurally incompatible data.

Power BI Service can then replace manual desktop distribution. Scheduled refresh, workspace permissions and deployment pipelines would support controlled movement from development to testing and production. Row-level security can restrict regional managers to their authorized locations, while executives retain a complete view. Refresh history and usage metrics would help administrators identify failures or rarely used visuals. These operational controls extend the existing dashboard without changing the analytical definitions established in the prototype.

B) Reproducibility and Data Governance

Reproducibility requires more than saving the final PBIX file. The source dataset, Python notebook, cleaned CSV, database schema, SQL queries and DAX definitions should be versioned as one release. A data dictionary should define each column, its type, valid range and business meaning. A measure catalogue should document the formula, filter behaviour, owner and dashboard pages that use each KPI. This documentation enables reviewers to determine whether a change is a correction, a new requirement or an unintended calculation difference.

Automated validation should generate an audit summary for every refresh. Useful fields include source filename, extraction date, raw and cleaned row counts, missing-value counts, duplicates removed, minimum and maximum dates, distinct orders, customers and products, and total sales before and after loading. The report can also compare database totals with selected DAX outputs. Storing these values creates a history of data quality and makes it possible to identify when a metric changed because of new business activity rather than a processing defect.

C) Future Analytical Extensions

The next analytical stage is to move from descriptive reporting toward diagnostic and predictive insight. Profit, discount, return, inventory and fulfilment-cost fields would allow the system to distinguish high revenue from high profitability. Decomposition analysis could identify whether a change in sales is associated with order volume, average value, category mix, customer segment or geography. What-if parameters could let managers test the effect of discount or growth assumptions without modifying the underlying dataset.

Time-series forecasting can estimate expected monthly sales and seasonal demand after the historical series is checked for missing periods, outliers and structural changes. Forecasts should be presented with confidence intervals and evaluated on holdout periods using measures such as mean absolute error. They should remain visually distinct from actual sales so that users do not confuse predicted values with observed performance. Alert rules can notify managers when actual results fall materially outside the expected range.

D) Replication and Maintenance Guidance

A reproducible implementation should use a fixed project structure. Source files are stored separately from generated cleaned files; Python notebooks or scripts are placed in a processing folder; SQL schema and query files are versioned in a database folder; and the Power BI report is stored with a measure catalogue and refresh notes. The working database should not be treated as the only copy of analytical logic. Keeping the scripts, queries and report definition together allows another analyst to reproduce the result on a new machine or restore the system after an environment failure.

Execution should follow a documented order. The analyst first confirms the expected column names and source period, then runs the cleaning procedure and reviews the audit counts. The cleaned file is loaded into a staging table, where row totals, distinct identifiers and sales aggregates are checked. Analytical queries are executed only after these controls pass. Power BI is then refreshed, and the displayed totals are compared with the approved database values. This sequence prevents a visually successful refresh from hiding a data-quality or import error.

Environment details should be recorded because library, database and connector versions can influence behaviour. The Python version, Pandas and NumPy versions, MySQL server version, character set, date format and Power BI Desktop build should be included in a release note. Connection paths and credentials must be externalized rather than embedded in shared notebooks. In a production setting, secrets should be stored through approved credential mechanisms, and the report should access a read-only reporting account instead of a database administrator account.

The measure catalogue should provide a plain-language definition alongside every DAX expression. For example, Total Orders is a distinct count of Order_ID rather than a count of order lines; Average Order Value divides total sales by that distinct order count; and customer measures are evaluated in the active filter context. Recording these choices avoids common interpretation errors when a new developer adds visuals. The same catalogue can identify which SQL query was used to validate each measure and the expected result for the baseline dataset.

Maintenance should include both technical and business review. Technical review confirms that refresh, query execution, filters, navigation and rendering remain correct. Business review confirms that measure definitions, thresholds and labels still match stakeholder requirements. When a definition changes, the revision should be dated and described, and historical comparisons should indicate whether earlier values were recalculated. This disciplined handover process enables the dashboard to evolve without losing the transparency and consistency demonstrated by the project.

REFERENCES

[1] W. McKinney, “Data structures for statistical computing in Python,” in Proc. 9th Python in Science Conf. (SciPy), Austin, TX, USA, 2010, pp. 56–61.
[2] W. McKinney, Python for Data Analysis: Data Wrangling with pandas, NumPy, and Jupyter, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.
[3] C. R. Harris et al., “Array programming with NumPy,” Nature, vol. 585, pp. 357–362, 2020, doi: 10.1038/s41586-020-2649-2.
[4] R. Elmasri and S. B. Navathe, Fundamentals of Database Systems, 7th ed. Boston, MA, USA: Pearson, 2017.
[5] C. J. Date, An Introduction to Database Systems, 8th ed. Boston, MA, USA: Addison-Wesley, 2004.
[6] R. Kimball and M. Ross, The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling, 3rd ed. Indianapolis, IN, USA: Wiley, 2013.
[7] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. San Francisco, CA, USA: Morgan Kaufmann, 2011.
[8] S. Few, Information Dashboard Design: Displaying Data for At-a-Glance Monitoring, 2nd ed. Burlingame, CA, USA: Analytics Press, 2013.

International Journal of
DATA SCIENCE AND IOT MANAGEMENT SYSTEM
Peer Reviewed, Referred & Indexed Journal
ISSN: 3068-272X www.ijdim.com Original Research Paper

[9] E. R. Tufte, The Visual Display of Quantitative Information, 2nd ed. Cheshire, CT, USA: Graphics Press, 2001.
[10] A. Cairo, The Truthful Art: Data, Charts, and Maps for Communication. Berkeley, CA, USA: New Riders, 2016.
[11] T. Munzner, Visualization Analysis and Design. Boca Raton, FL, USA: CRC Press, 2014.
[12] A. Ferrari and M. Russo, The Definitive Guide to DAX: Business Intelligence for Microsoft Power BI, SQL Server Analysis Services, and Excel, 2nd ed. Redmond, WA, USA: Microsoft Press, 2019.
[13] Microsoft, "Power BI documentation," Microsoft Learn, accessed Jul. 2026.
[14] Microsoft, "DAX function reference," Microsoft Learn, accessed Jul. 2026.
[15] Oracle Corporation, "MySQL 8.0 Reference Manual," accessed Jul. 2026.
[16] pandas Development Team, "pandas documentation," accessed Jul. 2026.
[17] NumPy Developers, "NumPy documentation," accessed Jul. 2026.
[18] P. J. Sadalage and M. Fowler, NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. Boston, MA, USA: Addison-Wesley, 2012.
[19] M. Golfarelli and S. Rizzi, Data Warehouse Design: Modern Principles and Methodologies. New York, NY, USA: McGraw-Hill, 2009.
[20] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in Proc. IEEE Symp. Visual Languages, Boulder, CO, USA, 1996, pp. 336–343.
[21] J. Heer, M. Bostock, and V. Ogievetsky, "A tour through the visualization zoo," Commun. ACM, vol. 53, no. 6, pp. 59–67, 2010.
[22] C. Ware, Information Visualization: Perception for Design, 4th ed. Cambridge, MA, USA: Morgan Kaufmann, 2020.
[23] R. W. Palmatier and S. Sridhar, Marketing Strategy: Based on First Principles and Data Analytics. London, U.K.: Palgrave Macmillan, 2017.
[24] D. Loshin, Business Intelligence: The Savvy Manager's Guide, 2nd ed. Waltham, MA, USA: Morgan Kaufmann, 2012.
[25] E. Turban, R. Sharda, and D. Delen, Decision Support and Business Intelligence Systems, 10th ed. Boston, MA, USA: Pearson, 2014.
[26] J. W. Tukey, Exploratory Data Analysis. Reading, MA, USA: Addison-Wesley, 1977.
[27] ISO/IEC 25012:2008, Software Engineering—Software Product Quality Requirements and Evaluation (SQuaRE)—Data Quality Model, ISO, Geneva, Switzerland, 2008.
[28] M. Stonebraker and U. Çetintemel, "One size fits all: An idea whose time has come and gone," in Proc. 21st Int. Conf. Data Engineering (ICDE), Tokyo, Japan, 2005, pp. 2–11.
[29] A. C. Telea, Data Visualization: Principles and Practice, 2nd ed. Boca Raton, FL, USA: CRC Press, 2014.
[30] Superstore Dataset Contributors, "Superstore sales dataset," public CSV dataset, accessed Jul. 2026.