International Journal of Computer Techniques–IJCT Volume 13 Issue 3, May-June - 2026

Open Access and Peer Review Journal ISSN 2394-2231 https://ijctjournal.org/

Data Analysis and Visualization Using Python: A Case Study of Amazon

Sethu Madhav, Seshukumar
Department of Computer Science and Engineering
Dhanalakshmi Srinivasan University
Guide: Dr. P. Thangavel

Abstract*—The rapid growth of e-commerce has created large volumes of transactional, behavioral, and product-level data. Organizations such as Amazon depend on systematic data analysis and clear visualization to understand customer demand, product performance, pricing patterns, regional trends, and operational efficiency. This paper presents an IEEE-style case study on data analysis and visualization using Python with an Amazon sales dataset as the analytical context. The proposed workflow includes data collection, preprocessing, cleaning, exploratory data analysis, feature engineering, visualization, and insight generation. Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Plotly, and Scikit-learn are considered for building a complete analytical pipeline. The study demonstrates how descriptive statistics, category-wise analysis, time-series trends, rating analysis, and revenue-based visualizations can support business decision-making. The results show that Python is an effective, flexible, and scalable environment for transforming raw e-commerce data into meaningful insights.*

Index Terms*—Amazon, data analysis, data visualization, Python, Pandas, Matplotlib, Seaborn, e-commerce analytics, exploratory data analysis.*

I. INTRODUCTION

E-commerce platforms generate data at every stage of the customer journey, including product browsing, cart creation, purchase completion, delivery, feedback, and returns. Amazon is one of the most suitable real-world examples for studying data analysis because its business model depends on product variety, customer reviews, pricing strategies, seller performance, and fast fulfillment. Even a medium-sized Amazon sales dataset can contain multiple analytical dimensions such as order date, product category, customer location, price, quantity, rating, discount, and revenue. Without proper analysis, these fields remain raw records; with Python-based analytics, they can become actionable business insights.

Data analysis is the process of collecting, cleaning, transforming, and interpreting data to support decisions. Visualization is the process of representing data using charts, graphs, and dashboards so that trends and patterns can be quickly understood. In the Amazon case study, visualization helps answer practical questions such as which products generate the highest revenue, which categories receive better ratings, which months produce peak orders, and whether discounts improve sales. These questions are important for marketing teams, inventory planners, sellers, and business analysts.

Python has become a popular tool for data analysis because it provides a complete ecosystem for handling data. Pandas supports tabular data manipulation, NumPy supports numerical operations, Matplotlib and Seaborn support statistical visualization, Plotly enables interactive charts, and Scikit-learn supports predictive modeling. Compared with spreadsheet-only analysis, Python provides reproducibility and automation. Compared with dashboard-only tools, Python provides more control over cleaning, feature creation, and statistical interpretation.

This paper focuses on an end-to-end analytical approach rather than only chart creation. The study begins with raw Amazon data, performs cleaning, prepares useful features, carries out exploratory analysis, generates visualizations, and converts observations into business insights. The purpose is to show how a student or fresher data analyst can handle an e-commerce dataset in a professional workflow.

II. OBJECTIVES OF THE STUDY

The first objective of this study is to design a structured workflow for analyzing Amazon sales data using Python. The workflow should be simple enough for academic implementation but professional enough to resemble industry data analysis practice.

The second objective is to identify meaningful patterns in the dataset. These patterns include product category contribution, revenue distribution, monthly sales trends, rating behavior, discount impact, and customer demand patterns.

The third objective is to demonstrate the role of visualization in decision-making. A good visualization should not only look attractive; it should clearly explain the business problem, show comparison, and help the viewer understand the conclusion quickly.

The fourth objective is to explain how Python libraries can be combined in a complete project. Pandas is used for data cleaning and aggregation, NumPy for numerical calculations, Matplotlib and Seaborn for static visual analysis, and Plotly for interactive business charts.

III. LITERATURE REVIEW

Previous research and technical literature show that Python has become a major platform for data-intensive work. McKinney introduced Pandas as a high-performance data structure library that simplified practical data analysis in Python [1]. Hunter presented Matplotlib as a foundational environment for two-dimensional plotting, making Python suitable for scientific and analytical visualization [2]. Waskom described Seaborn as a statistical visualization library that improves the readability of complex exploratory patterns [3].

In business analytics, Provost and Fawcett emphasized that data science is not only about algorithms but also about

framing business problems, extracting useful patterns, and converting data into decisions [4]. This idea is directly applicable to Amazon analysis because business questions must be defined before charts are created. For example, a revenue chart is useful only when it helps answer whether a product category, month, or price segment is performing better than others.

E-commerce analytics has also been supported by recommendation and customer behavior research. Linden, Smith, and York discussed item-to-item collaborative filtering, a technique used in large-scale recommendation systems [5]. Although this paper does not build a complete recommendation engine, the concept highlights the importance of product interaction data in platforms such as Amazon. Chen and Guestrin introduced XGBoost, which is often used in structured data modeling and can support future prediction tasks such as sales forecasting or product success classification [6].

The reviewed six papers/tools collectively support the proposed study. Pandas and Matplotlib enable core analysis and visualization. Seaborn improves statistical interpretation. Business data science literature provides the decision-making perspective. Recommendation research connects the case study to e-commerce intelligence. XGBoost shows how the same cleaned dataset can later be extended into predictive analytics.

IV. DATASET DESCRIPTION

The case study assumes an Amazon sales dataset containing product and order-level information. A typical dataset for this work may include order ID, order date, product name, product category, customer location, price, discount, quantity, total sales amount, rating, review count, payment mode, and delivery status. The exact number of rows may vary depending on the source, but the methodology remains similar for small, medium, or large datasets.

The main analytical fields are product category, sales amount, quantity, rating, discount percentage, and order date. Category is used to compare business segments. Sales amount is used to identify revenue contribution. Quantity is used to understand demand volume. Rating and review count are used to understand customer satisfaction. Discount percentage is used to study pricing influence. Order date is used to extract month, quarter, and year for time-series analysis.

Before analysis, the dataset must be checked for missing values, duplicate rows, wrong data types, inconsistent category names, invalid prices, and outliers. For example, a price value of zero may indicate missing data, a promotional free item, or an entry mistake. Similarly, a rating greater than five is invalid if the rating scale is one to five. Cleaning decisions must be documented because they directly affect final insights.

V. PROPOSED METHODOLOGY

A. Data Loading

The dataset is loaded into Python using Pandas. CSV and Excel files are commonly used in academic projects, while SQL databases are common in industry. The first step is to inspect the shape, column names, data types, and sample records. Functions such as head(), tail(), info(), describe(), and isnull().sum() provide the first understanding of the dataset.

B. Data Cleaning

Data cleaning converts raw records into analysis-ready data. Duplicate orders are removed when they represent accidental repeated entries. Missing numeric values may be handled using median or category-wise median when the distribution is skewed. Missing categorical values may be filled using mode or a label such as Unknown. Date columns are converted into datetime format so that month and year features can be extracted.

In Amazon sales analysis, cleaning also includes standardizing category names. For example, “Electronics”, “electronics”, and “Electronic Items” should not be treated as separate categories if they represent the same business segment. Price and discount columns should be converted into numeric format after removing currency symbols or percentage signs. Extreme outliers should be studied before removal because they may represent genuine high-value orders.

C. Feature Engineering

Feature engineering creates new columns that improve analysis. Total revenue can be calculated by multiplying price and quantity after discount adjustments. Month, day, quarter, and year can be extracted from order date. Price range groups can be created to compare low, medium, and premium products. Rating groups can be created to classify products as low-rated, average, or high-rated. These features make visualization more meaningful.

D. Exploratory Data Analysis

Exploratory data analysis studies the dataset from multiple angles. Univariate analysis studies one variable at a time, such as distribution of price or rating. Bivariate analysis compares two variables, such as discount and revenue. Multivariate analysis compares more than two variables, such as category, month, and revenue together. The goal is not only to produce charts but also to form explanations.

E. Visualization

Visualization is performed using bar charts, line charts, histograms, box plots, scatter plots, heatmaps, and pie or donut charts where appropriate. Bar charts are useful for comparing category revenue. Line charts are useful for monthly trends. Histograms show distribution of prices and ratings. Box plots reveal outliers. Scatter plots show relationships such as discount versus sales. Heatmaps reveal correlation among numeric variables.

VI. IMPLEMENTATION USING PYTHON

The Python implementation starts by importing required libraries. Pandas is used for data manipulation, NumPy for numerical operations, Matplotlib for basic plotting, Seaborn for statistical plots, and Plotly for interactive visualizations. The dataset is loaded using read_csv() or read_excel(). Column names are converted to lowercase and spaces are replaced with underscores to make coding consistent.

After loading, missing values and duplicates are checked. The analyst should not directly drop all missing values

because it may reduce dataset size and remove useful records. Instead, each column should be studied separately. For example, missing review count may be filled with zero, but missing product category may need manual correction or an Unknown label. Wrong data types should be converted using astype(), to_numeric(), and to_datetime().

Aggregation is a major part of the analysis. Groupby operations are used to calculate total revenue by category, average rating by product group, monthly revenue, top selling products, and average discount by category. Sorting helps identify top and bottom performers. Pivot tables help compare category performance across months. Correlation analysis helps understand relationships among price, quantity, discount, rating, and revenue.

The visualization stage converts these aggregations into charts. A category revenue bar chart identifies the strongest product segments. A monthly sales line chart shows seasonality. A rating distribution chart shows customer satisfaction. A discount versus sales scatter plot helps understand whether discounts increase demand. A correlation heatmap gives a quick view of numeric relationships. These outputs can later be included in a dashboard or PDF report.

VII. CASE STUDY ANALYSIS OF AMAZON

A. Category Performance

The first analysis compares revenue by product category. In most e-commerce datasets, a small number of categories contribute a large share of revenue. For Amazon, categories such as electronics, home appliances, fashion, books, and accessories may show different behavior. Electronics may produce high revenue because of high product price, while fashion may produce high order volume because of frequent purchases. This difference shows why revenue and quantity must be analyzed together.

B. Product-Level Insights

Product-level analysis identifies best-selling and low-performing items. Top products can be selected based on revenue, quantity sold, or customer rating. These three rankings may not be the same. A product with high quantity may have low revenue if the price is low. A product with high revenue may have fewer orders if it belongs to a premium segment. A product with high rating but low sales may require better marketing visibility.

C. Time-Series Trends

Order date analysis helps identify seasonality. Monthly revenue trends may show peaks during festival seasons, sale campaigns, or year-end offers. Daily analysis can identify weekend demand patterns. Quarterly analysis can help managers plan inventory and marketing budgets. In Python, time-based grouping is done after converting the date column into datetime format.

D. Rating and Review Analysis

Customer rating is an important measure of satisfaction. A rating distribution chart can show whether most products are positively reviewed or whether a large number of items receive low ratings. Category-wise average rating can identify segments that need quality improvement. Review count should be considered along with rating because a product with one five-star review is less reliable than a product with thousands of positive reviews.

E. Discount and Revenue Analysis

Discount analysis studies whether price reduction improves sales. A scatter plot between discount percentage and quantity sold can show whether higher discounts are associated with higher demand. However, correlation should not be treated as direct causation. High discounts may be given to slow-moving products, which means the relationship may be complex. Therefore, discount analysis must be combined with category, season, and product popularity.

VIII. VISUALIZATION DESIGN PRINCIPLES

Good visualization requires correct chart selection. A bar chart should be used for category comparison, a line chart for time trends, a histogram for distribution, and a scatter plot for relationship analysis. Using the wrong chart can confuse readers. For example, a pie chart is not suitable when there are many categories because small differences become difficult to compare.

Color and labels are also important. Chart titles should clearly state the business question. Axis labels should include units such as revenue, quantity, or rating. Data labels should be used only when they improve readability. Too many colors can make a chart look attractive but reduce clarity. In a professional report, simple and consistent visual design is better than unnecessary decoration.

In the Amazon case study, the best visual outputs include category revenue comparison, monthly sales trend, top products by revenue, rating distribution, discount impact scatter plot, and correlation heatmap. These visuals together provide a complete view of product performance, customer behavior, and business opportunity.

IX. RESULTS AND DISCUSSION

The analysis shows that Python can convert raw Amazon sales data into structured business insights. Category analysis helps identify high-revenue segments and low-performing areas. Product analysis helps identify items that should be promoted, restocked, or reviewed. Time-series analysis helps identify seasonal demand. Rating analysis supports customer satisfaction measurement. Discount analysis helps understand pricing behavior.

One important observation is that high sales do not always mean high customer satisfaction. A product may sell well because of discount or visibility but may still receive low ratings. Similarly, a category may have high revenue because of premium pricing but lower order volume. Therefore, business decisions should not depend on one metric alone. A combined view of revenue, quantity, rating, discount, and time provides better understanding.

The study also shows the importance of data cleaning. If duplicate records, missing prices, or wrong date formats are not handled, the final charts may become misleading. For example, duplicate orders can inflate revenue, missing categories can hide actual category performance, and incorrect date formats can break monthly trend analysis. Therefore, cleaning is not a small step; it is the foundation of the entire analytical process.

Visualization improves communication between technical and non-technical users. A manager may not understand code or statistical output, but a clear chart can quickly explain which category is growing and which product requires attention. This makes Python-based visualization useful for academic projects, internships, business reports, and entry-level data analyst portfolios.

X. APPLICATIONS OF THE STUDY

The proposed work can be applied in e-commerce sales monitoring, seller performance analysis, inventory planning, customer satisfaction tracking, marketing campaign evaluation, and product recommendation preparation. Small sellers can use this workflow to understand their online store performance. Students can use it as a portfolio project to demonstrate Python, EDA, visualization, and business interpretation skills.

In an industry environment, the same workflow can be connected to SQL databases, cloud storage, and dashboard tools such as Power BI or Tableau. Python can perform cleaning and advanced analysis, while dashboards can present final metrics to business users. This combination creates a complete analytics pipeline from raw data to decision-making.

XI. LIMITATIONS

This study is limited by the assumed structure of the Amazon dataset. Actual Amazon internal datasets are much larger and may include user behavior logs, search history, recommendation interactions, seller data, delivery data, and return reasons. Public or academic datasets may not contain all these fields.

The analysis is mainly descriptive and exploratory. It identifies patterns but does not prove causation. For example, a relationship between discount and quantity sold does not automatically mean discount caused the increase in sales. Future work can include predictive modeling, A/B testing, customer segmentation, and recommendation systems to make the study more advanced.

XII. FUTURE SCOPE

The future scope of this work includes building an interactive dashboard, adding machine learning models for sales forecasting, creating a recommendation engine, and automating PDF report generation. Natural language processing can also be applied to customer reviews to identify positive and negative sentiment. This would help sellers understand not only what customers rated but also why they rated that way.

Another extension is to build an AI-assisted data analyst system where the user uploads an Amazon dataset and asks questions in natural language. The system can automatically clean data, generate SQL queries, create charts, and explain insights. Such a system would be useful for students, small sellers, and business teams that need quick analysis without writing code manually.

XIII. ANALYTICAL WORKFLOW AND BUSINESS INTERPRETATION

A. Step-by-Step Analytical Flow

The complete analytical workflow can be divided into five practical stages. The first stage is data understanding, where the analyst identifies columns, data types, business meaning, and possible quality issues. The second stage is data cleaning, where duplicates, missing values, invalid values, and inconsistent formats are corrected. The third stage is feature creation, where new variables such as revenue, month, price band, rating band, and discount band are created. The fourth stage is exploratory analysis, where descriptive statistics and grouped summaries are produced. The fifth stage is visualization and reporting, where charts are converted into meaningful business explanations.

For an Amazon dataset, the analyst should not directly start with advanced models. A professional workflow begins with simple questions: How many products are present? Which categories dominate the dataset? Are there missing prices or ratings? Are dates valid? Are there repeated order IDs? Are price and quantity values realistic? These questions help avoid wrong conclusions. After basic validation, the analyst can move toward deeper questions such as category contribution, seasonal sales, review behavior, and discount effectiveness.

The workflow is iterative. During visualization, new data issues may be discovered. For example, a box plot may show extremely high revenue values. These values may be genuine bulk orders or data entry errors. Similarly, a time-series chart may show a sudden zero-sales month, which may be caused by missing records rather than actual business decline. Therefore, the analyst may return to the cleaning stage multiple times before finalizing results.

B. Suggested Python Pseudocode

The implementation can be represented using the following logical steps: import required libraries; read the Amazon dataset; standardize column names; remove duplicate records; convert date and numeric columns; handle missing values; create revenue and time-based features; group data by category, product, and month; generate visualizations; and summarize the insights. This pseudocode helps students explain the project clearly during reviews and interviews.

Example workflow: load dataset using Pandas; check df.shape, df.info(), and df.isnull().sum(); clean price, discount, and rating columns; create total_revenue = price * quantity; extract month from order_date; use groupby() for category revenue; create bar chart for category revenue; create line chart for monthly sales; create histogram for ratings; create scatter plot for discount and quantity; and write final observations in business language.

This structured approach makes the project reproducible. If a new Amazon dataset is uploaded later, the same code can be reused with minimum changes. Reproducibility is one of the main advantages of Python over manual spreadsheet analysis. It also supports automation, because the same cleaning and visualization pipeline can be connected to a web application or scheduled report.

C. Business Interpretation of Visual Outputs

Every chart must end with a business interpretation. A bar chart showing high electronics revenue should not simply be described as “electronics is highest.” A better interpretation is

that electronics may be a major revenue driver and should receive inventory priority, campaign attention, and quality monitoring. If the same category has lower average rating, the business should investigate product quality or delivery issues.

A monthly sales line chart should be interpreted in relation to events. If sales increase during a specific month, the analyst should check whether the increase is connected to festivals, seasonal demand, sale campaigns, or product launches. If sales suddenly decline, possible reasons include stock shortage, low discounts, poor marketing, delivery delays, or missing records. This type of explanation shows the difference between chart creation and real data analysis.

A discount analysis chart should be handled carefully. If higher discount is linked with higher quantity sold, it may indicate price sensitivity. However, if high discounts appear on low-performing products, the business may be using discounts to clear inventory. Therefore, discount analysis should be combined with product category, rating, and time period before making decisions.

XIV. ACADEMIC AND INDUSTRIAL RELEVANCE

This case study is academically relevant because it combines multiple core data analyst skills in one project. Students practice data cleaning, Pandas operations, visualization, exploratory analysis, and report writing. The project also helps students understand how to connect technical results with business decisions. In academic evaluation, this type of project is stronger than a project that only displays charts without explaining insights.

The study is industrially relevant because e-commerce companies rely heavily on analytics. Real teams monitor revenue, conversion, pricing, customer satisfaction, return rate, inventory status, and seller performance. Although this paper uses a simplified dataset, the same analytical thinking is used in real business environments. The workflow can be expanded to millions of rows by using SQL databases, cloud data warehouses, and distributed processing tools.

For fresher data analyst portfolios, an Amazon case study is valuable because interviewers can easily understand the business problem. The candidate can explain how raw data was cleaned, how features were created, why specific visualizations were selected, and what insights were obtained. This demonstrates practical thinking rather than only theoretical knowledge.

XV. QUALITY CHECKS FOR THE ANALYSIS

Quality checks are necessary before presenting final results. The total revenue calculated in Python should be compared with basic manual checks for a few rows. The number of records before and after duplicate removal should be recorded. Missing value treatment should be documented. Date extraction should be verified by checking sample rows. Category standardization should be reviewed to ensure that different spellings of the same category are not treated as separate groups.

Chart quality should also be checked. The chart must have a clear title, correct axis labels, readable category names, and appropriate sorting. Top product charts should not contain too many items; showing top ten or top fifteen products is usually more readable. Time-series charts should use chronological order. Correlation heatmaps should be interpreted only for numeric variables. These checks improve the reliability of the report.

Insight quality is equally important. An insight should include a finding, possible reason, and business action. For example, “Fashion has high order volume but moderate revenue, so sellers can improve average order value through bundles or premium products.” This is better than simply saying “Fashion has many orders.” Strong insights make the analysis useful for decision-making.

XVI. CONCLUSION

This paper presented a case study on data analysis and visualization using Python with Amazon as the analytical context. The study explained the complete workflow from data loading and cleaning to EDA, visualization, interpretation, and reporting. Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Plotly, and Scikit-learn provide a powerful environment for analyzing e-commerce data.

The Amazon case study demonstrates that meaningful insights require both technical processing and business understanding. Cleaning ensures accuracy, EDA discovers patterns, visualization communicates results, and interpretation converts charts into decisions. For students and entry-level data analysts, this project is useful because it combines practical coding skills with real-world business analytics thinking.

The work can be extended into predictive analytics, dashboard development, recommendation systems, and AI-based automated analysis. Therefore, Python-based data analysis and visualization remain highly valuable for modern e-commerce intelligence and data-driven decision-making.

XVII. SAMPLE INSIGHTS FROM THE AMAZON CASE STUDY

Based on the proposed analysis, the expected output of the case study can be written as a set of business insights. The first insight is category dominance. If a few categories contribute most of the revenue, the business should protect stock availability for those categories and monitor supplier performance. The second insight is product concentration. If revenue depends heavily on a small number of products, the business may face risk when those products become unavailable or receive poor reviews.

The third insight is seasonal movement. If monthly sales increase during festival or campaign months, marketing teams can prepare early promotions and inventory teams can increase stock before demand rises. The fourth insight is rating impact. If products with higher ratings also show stronger repeat demand, quality improvement can become a revenue strategy. The fifth insight is discount efficiency. If discounts improve sales only for selected categories, a uniform discount policy may waste margin and reduce profit.

The sixth insight is customer trust. Products with high review counts and stable ratings may be more reliable than products with high ratings but very few reviews. This helps sellers identify products that are genuinely strong in the market. The seventh insight is price segmentation. Low-price products may increase order count, while premium products

may increase revenue. Combining both segments can improve business stability.

These insights show that an Amazon dataset should be analyzed from multiple dimensions. A single chart cannot explain the whole business. A professional report should combine category analysis, product analysis, time analysis, rating analysis, and pricing analysis. The final recommendation should be based on combined evidence rather than one metric.

From an academic perspective, the case study also proves that visualization should be connected with storytelling. The report should not only contain images or charts; it should explain what changed, why it may have changed, and what action can be taken. This makes the project suitable for presentation, viva explanation, and portfolio demonstration.

From an implementation perspective, the same workflow can be reused for other e-commerce platforms. Flipkart, Myntra, Meesho, and other online retail datasets follow similar analytical logic. Only column names and business rules may change. This makes the method flexible and reusable beyond the Amazon case study.

REFERENCES

[1] W. McKinney, “Data Structures for Statistical Computing in Python,” in Proceedings of the 9th Python in Science Conference, 2010, pp. 56-61.

[2] J. D. Hunter, “Matplotlib: A 2D Graphics Environment,” Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

[3] M. L. Waskom, “Seaborn: Statistical Data Visualization,” Journal of Open Source Software, vol. 6, no. 60, p. 3021, 2021.

[4] F. Provost and T. Fawcett, Data Science for Business: What You Need to Know About Data Mining and Data-Analytic Thinking. O’Reilly Media, 2013.

[5] G. Linden, B. Smith, and J. York, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” IEEE Internet Computing, vol. 7, no. 1, pp. 76-80, 2003.

[6] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785-794.