Department of Computer & Informatics

Store Sales and Profit Analysis and Improvement

Usama Habib[1] Ghaith Jamjoum[2]

Department of Computer & Informatics, University of Sharjah, Sharjah, UAE

ARTICLE INFO

Keywords:
Data Frame (DF)

ABSTRACT

This project conducts a comprehensive sales and profit analysis utilizing a retail dataset sourced from Kaggle.com, known as the Superstore dataset, comprising approximately 10,000 rows of sales data. Leveraging Python libraries such as Pandas, Plotly, and mlxtend, the code extracts meaningful insights from diverse features. The primary focus is on meticulous data handling and processing using the Pandas library and show a proper analysis of sale as well as profit. Over six key parameters related to sales and profit are derived, providing a detailed understanding of the dataset. Furthermore, the implementation of the Apriori algorithm enhances the project by uncovering user buying patterns and behavior within the retail transactions.

1. Introduction

In the dynamic landscape of retail, understanding store sales and profit is paramount for businesses striving to thrive in a competitive market. Traditional methods of analysis are often time-consuming and may lack the precision needed to make informed decisions. Python, a versatile and powerful programming language, has emerged as a game-changer in the realm of data analysis and offers a robust platform for conducting in-depth store sales and profit analysis. This introductory exploration delves into the significance of leveraging Python for comprehensive business insights and provides a glimpse into the potential it holds for enhancing decision-making processes. Understanding the **retail landscape** requires a nuanced approach that goes beyond basic revenue tracking. Store sales and profit analysis involves extracting valuable insights from vast datasets, encompassing factors such as customer behavior, product performance, and market trends. Python, with its rich ecosystem of libraries and tools, empowers businesses to harness the full potential of their data. From data collection and cleaning to

[1] Usama Habib is a student in University of Sharjah of Master in Artificial Intelligence program (2023). Email address: U23102906@sharjah.ac.ae, usamah651@gmail.com

[2] Ghaith Jamjoum is a student in University of Sharjah of Master in Artificial Intelligence program (2023). Email address: U23102953@sharjah.ac.ae

sophisticated statistical modeling, Python provides a seamless workflow for professionals engaged in retail analytics.

One of the key advantages of using Python for store sales and profit analysis is its **versatility in handling diverse data sources**. Whether the data comes from point-of-sale systems, online platforms, or customer relationship management (CRM) tools, Python's compatibility with various data formats allows for seamless integration and analysis. This adaptability enables businesses to consolidate information from different channels, providing a holistic view of their operations.

Python's extensive library support, including popular ones like **Pandas, NumPy, and Matplotlib,** facilitates efficient data manipulation, analysis, and visualization. These libraries streamline tasks such as data cleaning, aggregation, and graphical representation, making it easier for analysts and decision-makers to derive meaningful insights. The combination of these tools enables the creation of interactive dashboards and reports, allowing stakeholders to monitor key performance indicators and make data-driven decisions in real-time.

Moreover, Python's capabilities extend beyond descriptive analytics. Advanced statistical models and machine learning algorithms can be seamlessly integrated into the analysis pipeline, providing predictive insights into future sales trends and potential areas for profit optimization. By leveraging machine learning, businesses can uncover hidden patterns and correlations within their data, enabling proactive decision-making and strategic planning.

1.1 Pandas

Pandas is a powerful and popular open-source data manipulation and analysis library for Python. It provides high-performance, easy-to-use data structures, and functions needed to work with structured data seamlessly. The name "pandas" is derived from "Panel Data," a term used in econometrics to describe multidimensional structured data sets.

The **central data structure** in Pandas is the Data-Frame, a two-dimensional table with labeled axes (rows and columns). Data-Frames are similar to spreadsheets or SQL tables, making them suitable for handling and analyzing structured data. Series are the building blocks of Data-Frames, representing either a single column or a single row in a Data-Frame. Pandas supports reading data from various file formats, including CSV, Excel, SQL databases, JSON, and more. It provides functions for handling missing data, removing duplicates, and reshaping data. It supports both label-based and position-based indexing, allowing users to access and modify data with ease.

Pandas includes **functionality** for working with time series data, making it a valuable tool for tasks like financial modeling, stock market analysis, and more. Merge and join operations facilitates the combination of data from multiple sources for comprehensive analysis. It includes built-in statistical functions for descriptive statistics and summary metrics. Although Pandas itself does not handle visualization, it integrates seamlessly with matplotlib/plotly, a popular Python plotting library, to enable data visualization within the Pandas environment. It is often used in conjunction with other libraries in the PyData ecosystem, such as NumPy for numerical operations and scikit-learn for machine learning tasks.

1.2 NumPy

NumPy, short for Numerical Python, is a fundamental package for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with mathematical functions to operate on these arrays. NumPy serves as the foundation for many other scientific computing libraries in the Python ecosystem.

The **core feature** of NumPy is the ndarray (n-dimensional array), which is a highly efficient, flexible, and homogeneous array data structure. NumPy allows for vectored operations, meaning that operations can be performed on entire arrays, eliminating the need for explicit looping. It provides a wide range of mathematical functions that

operate element-wise on arrays. Common mathematical operations, such as addition, subtraction, multiplication, and exponentiation, can be performed with ease using NumPy. The underlying implementation of NumPy in C and Fortran ensures high performance and memory efficiency.

1.3 Plotly

Plotly is a Python graphing library that enables interactive and high-quality visualizations in web-based applications and notebooks. It allows users to create a wide range of static and dynamic plots, including line charts, scatter plots, bar charts, heat maps, 3D plots, and more. It is known for its versatility, ease of use, and interactivity, making it a popular choice for data visualization in various domains.

Plotly **provides interactive visualizations** that can be explored and manipulated by users. It supports a wide range of plot types, from basic charts like line plots and scatter plots to more complex visualizations, including 3D plots and geographic maps. While Plotly is primarily associated with Python, it also supports other programming languages, including R, Julia, and JavaScript. This enables users to create visualizations in their preferred language while still leveraging Plotly's capabilities.

1.4 Apriroi Algorithm

In Python, Apriori is a popular algorithm for association rule mining. Association rule mining is a technique used in data mining to discover interesting relationships or associations among a set of items in large datasets. The Apriori algorithm specifically focuses on finding frequent item sets and generating association rules based on their occurrences.

Apriori algorithm works by **identifying all frequent item sets in the dataset**. An item set is considered frequent if it occurs in the dataset with a frequency greater than or equal to a predefined threshold (support). The Apriori algorithm relies on the Apriori property, which states that if an item set is frequent, then all of its subsets must also be frequent. This property helps prune the search space and reduces the number of candidate item sets that need to be examined. The algorithm iteratively generates candidate item sets of increasing size based on the frequent item sets discovered in the previous iteration. Once all frequent item sets are discovered, association rules are generated based on these item sets.

In Python, you can implement the Apriori algorithm using the `mlxtend` library, which provides a convenient implementation for association rule mining.

1.5 Cloud Computing

Cloud computing is a technology that involves delivering computing services over the internet. Instead of owning and maintaining physical servers or computing infrastructure, businesses can access a wide range of computing resources, including storage, processing power, databases, networking, and more, on a pay-as-you-go basis. Cloud computing providers offer these services through a network of remote servers, providing scalability, flexibility, and cost-efficiency.

Cloud computing can bring several benefits to **Python-based retail store data projects**, Retail data can vary significantly in volume, especially during peak seasons or special promotions. Cloud computing allows for easy scalability, enabling retailers to scale their computing resources up or down based on demand. This ensures optimal performance without the need for large upfront investments in infrastructure.

This cost-effective approach eliminates the need for significant upfront capital expenditure on hardware and allows for more predictable budgeting. Cloud providers offer machine learning services that can be seamlessly integrated into Python projects. Providers invest heavily in security measures, including encryption, access controls, and compliance certifications. This ensures that sensitive retail data is stored and processed securely, meeting regulatory requirements and industry standards.

The report is structured as follows: section 2 provides a review of related work, section 3 explains

the methodology in AI based security solutions, Section 4 talks about conclusion and future work.

2. Related Work

Several techniques and tools related to Python for sales and profit analysis have gained popularity. Keep in mind that the field of data analysis and Python libraries is dynamic, and new developments may have occurred since then. Here are some of the latest or widely used techniques and tools for sales and profit analysis mentioned in different papers:

Researchers often explore advanced time series models for accurate sales forecasting. Techniques like Long Short-Term Memory (LSTM) networks, Prophet by Facebook, or hybrid models combining traditional statistical methods with machine learning algorithms are commonly discussed [1]. Studies may focus on advanced techniques for calculating and optimizing Customer Lifetime Value. Machine learning models, predictive analytics, and customer segmentation methodologies may be explored [2]. Research papers may delve into optimization methods, including linear programming, for pricing strategies to maximize profits. Constraints related to production, demand, and market conditions are often considered [3]. Machine learning-based customer segmentation approaches could be explored to better understand customer behavior and preferences. Clustering algorithms, such as K-Means or DBSCAN, might be discussed [4].

Error in profit estimation in recommender systems that are based on association mining. It investigates the impact of information loss on profit estimation by comparing two cases: one with complete information and the other with incomplete information. The study provides insights into the relationship between expected profits and support threshold levels, highlighting the importance of accurate profit estimation in recommender systems. The research uses real-world data and presents a numerical example to illustrate the concepts and formulas developed [5].

Apriori algorithm based on Map Reduce model, with the goal of addressing the performance bottleneck that occurs when the data set is slightly larger. Prior to filtering the frequent item sets that meet the requirements based on the minimum support threshold, all local frequent item sets on each sub node in the cluster are first calculated. These local frequent item sets are then combined into the global candidate item sets. The enhanced algorithm has the advantage of being more efficient because it only needs to calculate the frequent item set in parallel and scan the transaction database twice [6].

Companies use data mining (DM) to extract knowledge from available data so they can make informed decisions. To maximize revenue, a business should only invest in the line of products that its customers regularly purchase and at the right price [7]. The study comes to the conclusion that there is a substantial correlation between the firm's marketing costs and profitability as well as a significant correlation between turnover and marketing expenses. In order to evaluate the variables that were included in this study and analyze the data, regression analysis was utilized [8].

3. Methodology

The given Python code performs sales and profit analysis on a retail dataset using the Pandas, Plotly, and mlxtend libraries. The dataset is obtained from Kaggle.com[3] as superstore dataset. The data involves around 10,000 rows of sales data mainly consisting of order ID, order date, shipping and receiving date, customer ID, customer name, customer segment and more. The code is then extended with some improvements as well. Table 1 gives the detail of all columns in dataset.

[3] https://www.kaggle.com/datasets/vivek468/superstore-dataset-final/data

Table 1 Columns in Dataset

Columns	Description
Row ID	Unique ID for each row.
Order ID	Unique Order ID for each Customer.
Order Date	Order Date of the product.
Ship Date	Shipping Date of the Product.
Ship Mode	Shipping Mode specified by the Customer.
Customer ID	Unique ID to identify each Customer.
Customer Name	Name of the Customer.
Segment	from where the Customer belongs.
Country	Country of residence of the Customer.
City	City of residence of the Customer.
State	State of residence of the Customer.
Postal Code	Postal Code of every Customer.
Region	Region where the Customer belongs.
Product ID	ID of the Product.
Category	Category of the product.
Sub-Category	Sub-Category of the product ordered.
Product Name	Name of the Product.
Sales	Sales of the Product.
Quantity	Quantity of the Product.
Discount	Discount provided.
Profit	Profit/Loss incurred.

After installing required libraries, following are the steps executed. Steps are also visually described in Figure 3.

Step1- Firstly, we use Pandas to read a CSV file named "Sample-Superstore.csv" into a Data Frame (DF) called data. The specified encoding is 'latin-1', ensuring proper interpretation of characters. This DF can now be used to analyze and manipulate the tabular data from the CSV file using Pandas functionalities.

Step2- 'Order Date' column in the Pandas DF 'data' is converted to a date time format using the pd.to_datetime() function. This conversion enables time-related operations and analysis on the 'Order Date' data within the DF.

Step3- Adding a new column 'Order Month' to the Pandas DF 'data', extracting the month component from the 'Order Date' column using the **dt.month**. Similarly, obtaining year and days of week.

Step4- A New DF **sales by month** is created by grouping the 'data' DF by the 'Order Month' column and calculating the sum of sales for each month. The resulting DF is then reset to have a default integer index.

Step5- A line plot using Plotly Express, visualizing the total sales per month DF. The x-axis represents the order months, and the y-axis shows the corresponding sales values, providing a graphical representation of monthly sales analysis.

Step6- After that using step 4 and 5, we extract and plot results in Figure 1 2 & 4, sales by category and sub category, profit by month, category and subcategory. Figure 1 shows sales by month, Figure 2 show sales by category & Figure 4 show sales by sub-category.

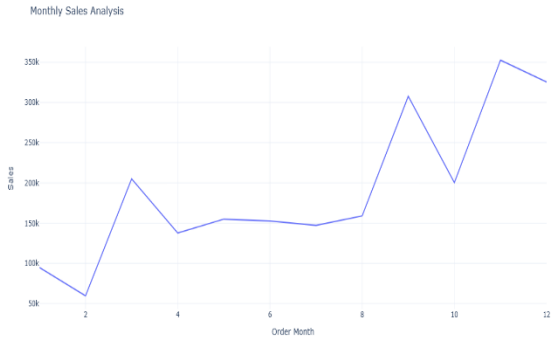

Figure 1 Sales by Month

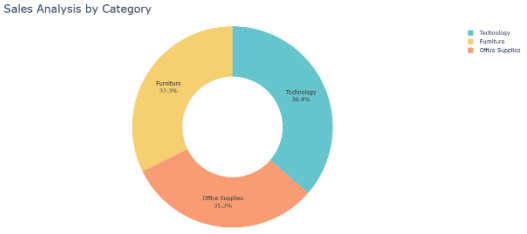

Figure 2 Sales by Category

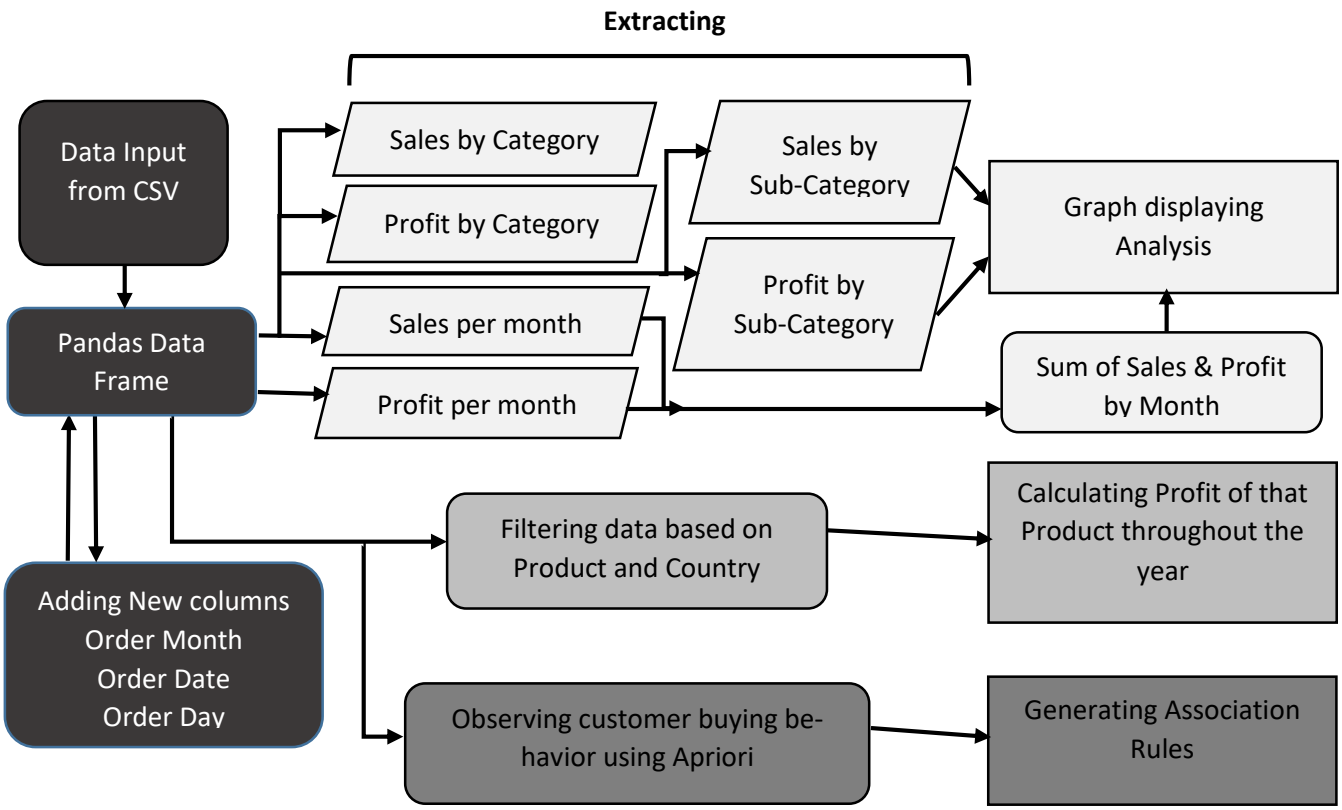

Figure 3 Methodology

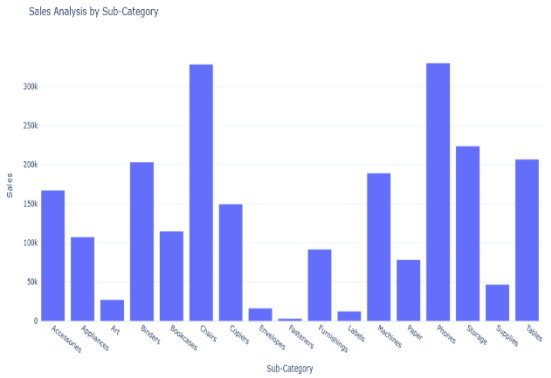

Figure 4 Sales by Sub-Category

Figure 5 Sales & profit By Segment

Step7- This code groups the 'data' DF by the 'Segment' column and aggregates the sum of 'Sales' and 'Profit' for each segment. The resulting DF for 'sales_profit_by_segment' is then reset, creating a new DF that shows the total sales and profit for different segments in the original dataset. Displayed in Figure 5.

Step8- Following improvements are carried out.

- User input (commented out in the code) is used to filter data based on a specific product and country. The resulting data is then grouped by month, and sales, profit, and their ratio are displayed.

- Apriori algorithm (from the mlxtend library) is applied for association rule mining. It's used to find patterns in customer buying behavior. The

code filters data for a specific segment (Home Office) and creates a pivot table of customer and product interactions (quantity bought). Apriori is then applied to find frequent item sets, and association rules are generated & saved in CSV file. A glance of CSV file is in Table 2.

The improvements showcase practical applications of data analysis, including personalized sales analysis for a specific product and country and association rule mining for understanding customer purchasing patterns.

Table 2 Association Rules

Antecedent's	Consequent's
Message Book, Phone, Wire bound Standard Line Memo	KI Adjustable-Height Table, Westinghouse Clip-On Gooseneck Lamps
Bevis 36 x 72 Conference Tables	Harmony Air Purifier
Atlantic Metals Mobile 3-Shelf Bookcases, Custom Colors, Global Stack Chair without Arms, Black	4009 Highlighters by Sanford
Atlantic Metals Mobile 3-Shelf Bookcases, Custom Colors, 4009 Highlighters by Sanford	Global Stack Chair without Arms, Black

4. Conclusion & Future Work

In this project, detailed sales and profit analysis is carried out using dataset with more than 10,000 transactions. Pandas library in python had the most important role in data handling and processing. Based on different features in dataset more than six parameters are extracted related to sales and profit. Apriori algorithm is also implemented to get to know user buying behavior. This project can be extended to a recommender system where recommendations can be generated based on user profile. Further, based on sales and profit data automatically more products can be suggested to company to increase sales.

References

[1] J. Brownlee, "Time Series Prediction with LSTM Recurrent Neural Networks in Python with Keras," Machine Learning Mastery, 2022.

[2] Z. Pollak, "Predicting Customer Lifetime Values - ecommerce use case," 2021.

[3] R. Phillips, "Pricing and Revenue Optimization," 2017.

[4] T. Jain, A. Sinha and V. Tanwar, "Review on Customer Segmentation Methods Using Machine Learning," in *International Conference on IoT*, 2023.

[5] G. Ertek, X. Chi, G. Yee, O. B. Yong and B.-G. Choi, "Profit Estimation Error Analysis in Recommender," in *IEEE International Conference on Big Data*, Santa Clara, 2015.

[6] H.-B. WANG and Y.-J. GAO, "Research on parallelization of Apriori algorithm in association rule," *Procedia Computer Science,* vol. 183, pp. 641-647, 2021.

[7] O. J. Adelakun and A. Daramola, "Analysis of the Effect of Advertising on Sales and Profitability of Company," *International Journal of Management and Network Economics,* vol. 2, no. 3, pp. 81-90, 2019.

[8] A. Bejju, "Sales Analysis of E-Commerce Websites using DataMining Technique," *nternational Journal of Computer Applications,* vol. 133, no. 5, 2016.