

# Supermarket Sales Dashboard

Bipan Neupane, Dishna Chhantyal  
Department of Computer Science and  
Engineering  
Kathmandu University  
Dhulikhel, Nepal,  
[bipanneupane12@gmail.com](mailto:bipanneupane12@gmail.com),  
[dishnachhantyal14@gmail.com](mailto:dishnachhantyal14@gmail.com)

**Abstract**—In this project, we have created an interactive dashboard which offers at-a-glance insights based on key performance indicators on the sales data of a supermarket company in Myanmar. We have used several python libraries like numpy, pandas, plotly and streamlit to build a dashboard. With visual resources that are easy to understand, it allows the total visibility into your business, improved results, increased productivity and increased profits.

**Keywords** – *numpy, pandas, plotly, matplotlib, streamlit*

## I. INTRODUCTION

An interactive dashboard allows you to act on that data instantly to create and assign tasks, or easily generate and filter reports to understand the underlying issues affecting performance and take immediate action. This aids organizational intelligence by highlighting where to assign resources or when to escalate challenging situations [1]. It also create an interface between critical data and employees, democratizing the ability to make operational performance improvements.

Shopping from a Supermarket is common practice nowadays. The variety of products offered, relatively low prices and convenience in shopping results in a huge influx of customers, and as a result a huge revenue through sales. The growth of supermarkets in most populated cities are increasing and market competitions are also high. The understanding of the association between different variables and how they impact each other through an interactive dashboard will create an opportunity for the supermarket company to come up with the solutions which can help them to target and attract their customers in a better manner.

## II. TECHNICAL BACKGROUND

A dashboard for this project is built by using python and its libraries like streamlit, plotly, pandas etc. As for the exploratory data analysis, python libraries like numpy, pandas, matplotlib has been practiced.

### A. Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse [2].

### B. Streamlit

Streamlit is an open source app framework in python language. It helps us create beautiful web-apps for data science and machine learning in a little time. It is compatible with major python libraries such as keras, numpy, pytorch, scikit-learn, pandas, matplotlib, etc. [3].

### C. Plotly

Plotly is an interactive open-source python library. Plotly makes data visualization and understanding simple and easy. Plotly is a high-level, declarative charting library that includes over 30 chart types, including scientific charts, 3D graphs, statistical charts, SVG maps, financial charts, and more [4].

### D. Pandas

Pandas is one of the powerful open source libraries in the Python programming language used for data analysis and data manipulation. If you want to work with any tabular data, such as data from a database or any other forms (Like CSV, JSON, Excel, etc..) then pandas is the best tool [5].

### E. Numpy

Numpy is a general-purpose array-processing package. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more [6].

## III. RELATED WORKS

### A. Google Analytics

Google Analytics is a web analytics service that provides statistics and basic analytical tools for search engine optimization (SEO) and marketing purposes. Google Analytics is used to track website performance and collect visitor insights. It can help organizations determine top sources of user traffic, gauge the success of their marketing activities and campaigns, track goal completions (such as purchases, adding products to carts), discover patterns and trends in user engagement and obtain other visitor information such as demographics [7].

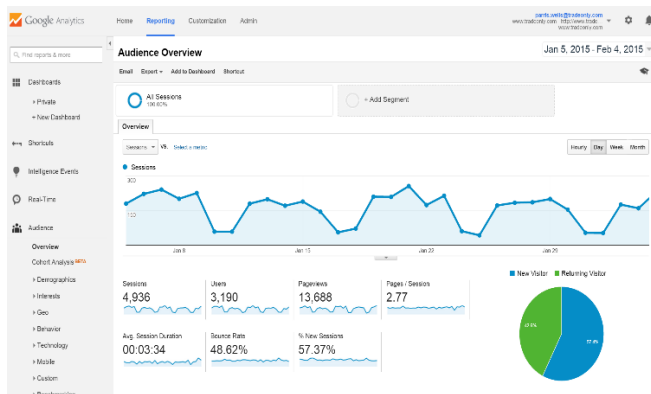


FIGURE 1: GOOGLE ANALYTICS DASHBOARD

#### IV. METHODOLOGY

The steps that we followed to complete this project are:

##### A. Selection of a dataset

There are hundreds of datasets in every topic imaginable — Business, Education, Sports, Entertainment and even COVID. But, for this project, we chose supermarket sales data from kaggle to build an interactive dashboard. The dataset consists of historical sales data of a supermarket company from 3 different branches over 3 months from Jan-March 2019. Following table provides a summary of the variables and the variable description.

S No	Variable/Column Name	Variable/Column Description
1	Invoice id	Computer generated sales slip invoice identification number
2	Branch	Branch of supermarket (3 branches are available identified by A, B and C)
3	City	Location of supermarket
4	Customer type	Type of customers, recorded by Members for customers using member card and Normal for without member card
5	Gender	Gender type of customer (Male/Female)
6	Product line	General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel
7	Unit price	Price of each product in \$
8	Quantity	Number of products purchased by customer
9	Tax	5% tax fee for customer buying
10	Total	Total price including tax
11	Date	Date of purchase (Record available from January 2019 to March 2019)
12	Time	Purchase time (10am to 9pm)
13	Payment	Payment used by customer for purchase (3 methods are available - Cash, Credit card and E-wallet)
14	COGS	Cost of goods sold
15	Gross margin percentage	Gross margin percentage
16	Gross income	Gross income from customers i.e. income of supermarket and spend by customers
17	Rating	Customer stratification rating on their overall shopping experience (On a scale of 1 to 10, 1 being lowest and 10 being highest)

FIGURE 2: VARIABLE DESCRIPTION 1.

##### B. Understanding dataset

First let's look at the data at hand.

```
In [2]: smsales=pd.read_csv('supermarket_sales - Sheet1.csv')
smsales.head()
```

	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Tax 5%	Total	Date	Time	Payment	cogs	gross margin percentage	gross income	Rating
0	759-41-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	26.1415	548.9715	1/5/2019	13:08	E-wallet	522.83	4.761905	26.1415	9.1
1	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	3.8200	80.2200	3/8/2019	10:29	Cash	76.40	4.761905	3.8200	9.6
2	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	16.2155	340.5255	3/9/2019	13:23	Credit card	324.31	4.761905	16.2155	7.4
3	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	23.2880	465.9480	1/27/2019	20:33	E-wallet	465.76	4.761905	23.2880	8.4
4	373-73-7510	A	Yangon	Normal	Male	Sports and travel	86.31	7	30.2085	634.3785	2/8/2019	10:37	E-wallet	604.17	4.761905	30.2085	5.3

FIGURE 3: FIRST FIVE ROWS OF OUR DATA FRAME

Now let's have a look at the detailed account of the information of our data frame.

```
In [3]: smsales.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 17 columns):
#   Column              Non-Null Count  Dtype  
---  --
0   Invoice ID           1000 non-null   object  
1   Branch              1000 non-null   object  
2   City                1000 non-null   object  
3   Customer type       1000 non-null   object  
4   Gender              1000 non-null   object  
5   Product line        1000 non-null   object  
6   Unit price          1000 non-null   float64 
7   Quantity            1000 non-null   int64  
8   Tax 5%              1000 non-null   float64 
9   Total               1000 non-null   float64 
10  Date                1000 non-null   object  
11  Time                1000 non-null   object  
12  Payment             1000 non-null   object  
13  cogs                1000 non-null   float64 
14  gross margin percentage 1000 non-null   float64 
15  gross income         1000 non-null   float64 
16  Rating              1000 non-null   float64 
dtypes: float64(7), int64(1), object(9)
memory usage: 132.9+ KB
```

FIGURE 4: DATA FRAME INFORMATION

```
In [4]: smsales.isnull().sum()

Out[4]: Invoice ID      0
Branch      0
City      0
Customer type  0
Gender      0
Product line  0
Unit price  0
Quantity    0
Tax 5%      0
Total       0
Date        0
Time        0
Payment     0
cogs        0
gross margin percentage  0
gross income  0
Rating      0
dtype: int64
```

FIGURE 5: SUM OF EMPTY VALUES IN ALL COLUMNS

From this we observe that there are:

- 17 columns
- 1000 entries/rows
- no missing value and the data set is clean
- 9 categorical data columns, 8 numerical (7 float and 1 integer) data columns

##### C. Exploratory Data Analysis (EDA)

To have a better grasp of the dataset that we chose, we performed some exploratory data analysis with the help of some python libraries like pandas, numpy, matplotlib, etc. before directly diving into building a dashboard.

###### 1. Net sales in each branch

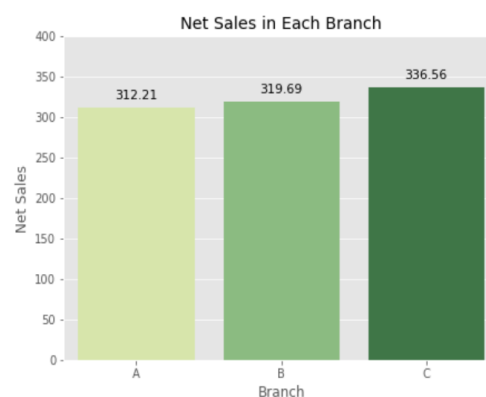


FIGURE 6: NET SALES OF EACH BRANCH

From figure 6, we can say that branch C is the branch with the biggest total or net sales compared to the other two branches.

- The number of product lines sold in each branch

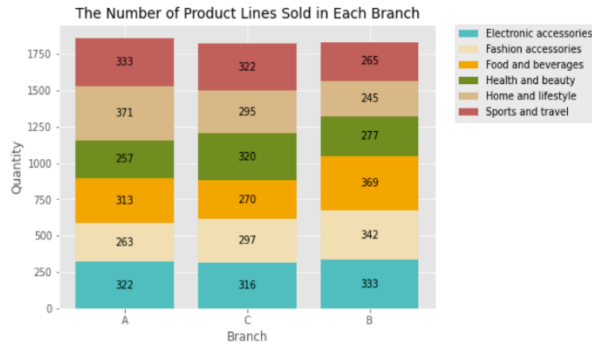


FIGURE 7: PRODUCT LINES SOLD IN EACH BRANCH

From the plot, home and lifestyle line is the most sold in branch A, food and beverages is the most sold in branch B, and sports and travel is the most sold in branch C.

- Sold-hour in each branch

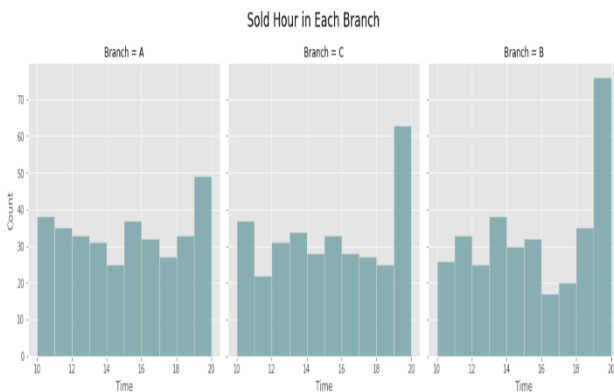


FIGURE 8: SOLD-HOUR IN EACH BRAND

Sales peak in all three branches mostly happened between 7 pm to 8 pm. The value for all branches quite steady in the hours before.

- Rating distribution in each branch

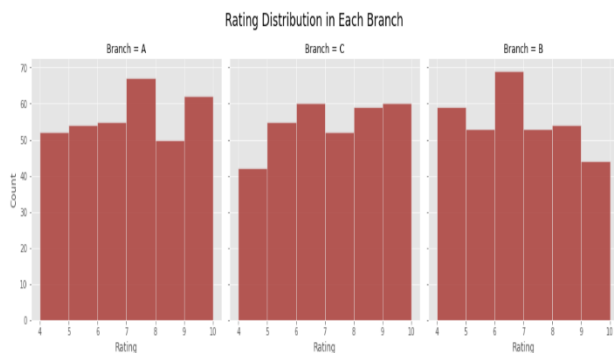


FIGURE 9: RATING DISTRIBUTION IN EACH BRANCH

In figure 9, the rating distribution in each branch is demonstrated. From above figure, we can say that the highest ratings for branch A are between 7 to 8, whereas the highest ratings for Branch B and C are between 6 to 7.

## 5. Numerical data distribution

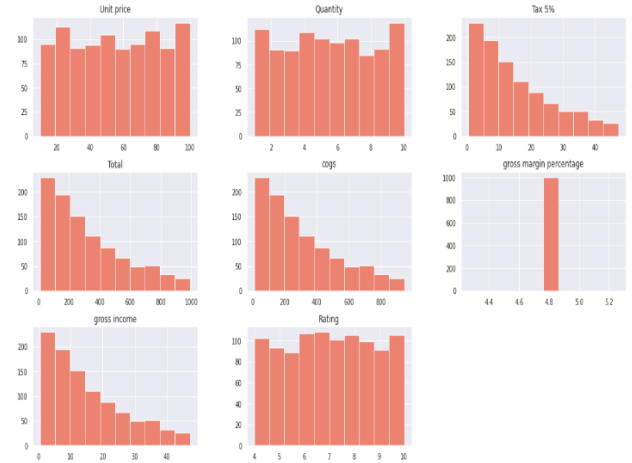


FIGURE 10: NUMERICAL DATA DISTRIBUTION

After performing some exploratory data analysis, we definitely gained more insights about our datasets and started to work on building our dashboard.

## V. IMPLEMENTATION AND RESULTS

While building this we application, we performed data manipulations in pandas, created interactive visualizations using plotly, and of course, created a dashboard using streamlit.

As the first step, we started by setting up some basic configuration of our web application. Next, we specified how the page content should be laid out. Then, we built the side bar section. We used the side bar to gather the filter criteria and apply those filters to our dataset. We planned to filter out dataset based on the different cities, customer types, gender information and different branches. We used streamlit multi select component for all filters.

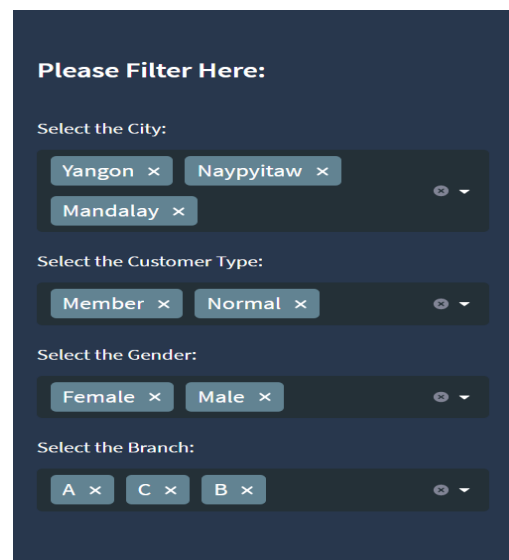


FIGURE 11: SIDE BAR SECTION

Now, for the KPI's, we decided to display the total sales, the average rating and the average sales by transaction. When we change the filters, we also see our KPI's changing.

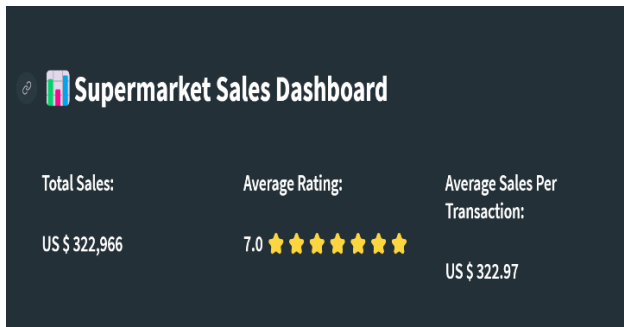


FIGURE 12: KPI'S OF OUR WEB APP

In the next section of our web application, we included two bar charts. The first bar chart plots the sales by hour. Another one plots the sales by product line. Before plotting, some calculations were done in pandas and we used the plotly express library to plot the data. So, in the given manner, supermarket sales dashboard was created by using streamlit.

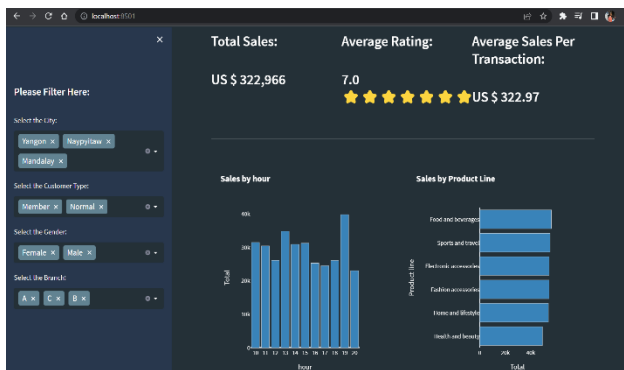


FIGURE 13: FINAL LOOK OF SUPERMARKET SALES DASHBOARD

## VI. CONCLUSION

We have successfully implemented streamlit to create an interactive dashboard of the supermarket data sales. In the dashboard, the user has the different options like city, gender etc. to filter the dataset. The KPI's on top and the charts below are updated accordingly.

## VII. ACKNOWLEDGEMENT

We would like to express our great appreciation to Dr. Rajani Chulyadyo for giving this great opportunity to work on this project. We really learned a lot while working on this project.

## VIII. REFERENCES

- [1] Sean King. (July/ 8/2019). Drive Digital Transformation with Interactive Dashboards. Flowinfinity. <https://www.flowinfinity.com/blog/interactive-dashboards-benefits.aspx>
- [2] Python community. (2001). what is Python? Executive Summary. Python Software Foundation. <https://www.python.org/doc/essays/blurb/>
- [3] Aggarwal Riya. (23/ Oct/2020). Streamlit – Introduction and Setup. GeeksforGeeks. <https://www.geeksforgeeks.org/streamlit-introduction-and-setup>
- [4] Gumber Sumit, Kumar Avtar. (19 /Oct/2022). Python Plotly tutorial. GeeksforGeeks. <https://www.geeksforgeeks.org/python-plotly-tutorial/>
- [5] Devara Gireesha. (18/Nov/2021). Pandas in python. Tutorialspoint. <https://www.tutorialspoint.com/what-is-pandas-in-python>
- [6] NumPy Developers. (2008). NumPy. numpy.org. <https://numpy.org/numpy-tutorials/>
- [7] Wesley Chai. (2010). Google Analytics. techtarget. <https://www.techtarget.com/searchbusinessanalytics/definition/Google-Analytics>