FINAL PROJECT REPORT :

COFFEE SHOP GROWTH FEASIBILITY ANALYSIS

business inteliigence and database management mini-project

Presented by

Hadil Mabrouk
Rouaida Hentati
Mohamed Ayadi

Submitted to

Prof . Ameni Azouz

Tunis Business School

January 2025

Table of Contents

introduction

This report explores the feasibility of **opening a drive-through coffee shop** using data-driven insights. The goal is to determine whether this business model aligns with customer needs, operational capabilities, and market trends. By leveraging data-driven insights from sales, customer feedback, and operational data, this study provides actionable recommendations for decision-making.

Key objectives include identifying **high-demand products**, analyzing **customer preferences for take-out services**, and evaluating **financial viability**. Tools such as Python, MySQL, and Power BI are used for ETL, analysis, and visualization.

Methodology

Our methodology integrates powerful tools and techniques for comprehensive analysis:

- **Google Colab**: Facilitates ETL (Extract, Transform, Load) processes using **Python.**
- **MySQL** : Serves as a robust database management system, providing a solid foundation for our analytical endeavors.
- **Power BI:** Enables OLAP (Online Analytical Processing) and visualization for insightful reporting.

Key phases include:

a. ***Data Gathering:*** Collecting diverse data from Excel and CSV formats.
b. ***ETL Process****:* Cleaning and transforming the data for accuracy and consistency.
c. ***Data Modeling****:* Structuring data into a star schema for efficient analysis.
d. ***Data Visualization:*** Creating dashboards to reveal actionable insights

Key Objectives

- **Understand Customer Preferences:** Identify satisfaction levels and peak times for take-out orders.
- **Identify high-demand products for drive-through prioritization :** identify most selling products that can be introduced in the drive-through
- **Optimize Operations:** Analyze operational and financial feasibility for implementing a drive-through
- **Provide Strategic Recommendations:** Deliver actionable insights to guide decisions on drive-through implementation and menu design.

work explanation

Data gathering

Our comprehensive data exploration embarks on the utilization of a coffee shop public database from the kaggle website .
In a meticulous selection process, we curated essential files aligning with our Key Performance Indicators (KPIs). This amalgamation of datasets seamlessly combines both CSV and XSLX formats, presenting a diverse and rich source of information.

The dataset we worked with contain this combination of data files :

- **orders.csv**
- **Items.csv**
- **inventory.csv**
- **recipe.csv**
- **Customer-Feedback-Form.xlsx**

ETL PROCESS

2.2.1 Data Extraction

Data extraction was carried out in Google Colab using Python.
To facilitate data processing, the following Python libraries were utilized:

- **pandas:** For data manipulation and analysis.
- **numpy**: For numerical operations.
- **matplotlib.pyplot** and **seaborn**: For potential data visualization

The datasets were loaded into pandas DataFrames as follows:

- Orders Data: Loaded from orders.xlsx.
- Customer Feedback Data: Loaded from Customer-Feedback-Form.xlsx
- Items Data: Loaded from items.csv.

- Inventory Data: Loaded from inventory.csv
- Recipe Data: Loaded from recipe.csv.

A preview of the first few rows of each dataset was displayed to confirm successful loading and examine the data structure.

```
[ ] import pandas as pd
    import numpy as np
    import matplotlib.pyplot as plt
    import seaborn as sns
    from google.colab import files
```

```
uploaded = files.upload()
orders_data = pd.read_excel('orders.xlsx')
orders_data.head()
```

```
uploaded = files.upload()
feedback_data = pd.read_excel('Customer-Feedback-Form.xlsx')
feedback_data.head(10)
```

```
uploaded = files.upload()
items_data = pd.read_csv('items.csv')
items_data.head(10)
```

Figure 2.1: Screenshots Illustrating the Data Extraction Process

2.2.2 Data Transformation

In the data transformation phase, it became imperative to optimize certain data for more efficient processing ,we used Python to manipulate data and configure it for the data warehouse. We changed the structure of the data so that it could be easily manipulated and managed, mainly using the pandas python library. The operations are all available in the google colab Notebook file

handling missing values

```
print("Missing values before handling:")
print(orders_data.isnull().sum())
# Replace missing values in 'in_or_out' with a placeholder (e.g., 'unknown')
orders_data['in_or_out'] = orders_data['in_or_out'].fillna('unknown')

# Verify there are no missing values
print("Missing values after handling:")
print(orders_data.isnull().sum())
```

Missing values in the in_or_out column were replaced with the placeholder value 'unknown'.


```
missing_data = feedback_data.isnull().sum()
print("Missing values in each column:\n", missing_data)
for col in feedback_data.columns:
    most_common = feedback_data[col].mode()[0]
    feedback_data[col].fillna(most_common, inplace=True)
print("Missing values after handling:\n", feedback_data.isnull().sum())
```

For each column of the customer feedback data, the missing values were filled with the most common value (mode) of the respective column.

Removing duplicate rows

```
duplicate_rows = feedback_data[feedback_data.duplicated()]
print(f"Number of duplicate rows: {duplicate_rows.shape[0]}")
# Drop duplicate rows
feedback_data.drop_duplicates(inplace=True)
```

```
Number of duplicate rows: 3
```

Duplicate rows in the Customer Feedback Data were identified and removed. The number of duplicate rows before removal was logged.

Data standardization

```
# Standardize column names
feedback_data.columns = feedback_data.columns.str.lower().str.replace(' ', '_')
# Standardize data formats
feedback_data['entry_date'] = pd.to_datetime(feedback_data['entry_date'], errors='coerce')
feedback_data.head()
```

Column names in the Customer Feedback Data were standardized by Converting all names
to lowercase and Replacing spaces with underscores . while the entry_date column was standardized, with invalid formats replaced by NaT.

```
inventory_data['created_at'] = pd.to_datetime(inventory_data['created_at'])

# Verify the change
print(inventory_data['created_at'].dtypes)
```

the created_at column was converted to datetime format.

2.2.3 Data Loading

```
# Save dataframes as Excel files
feedback_data.to_csv('/content/cleaned_feedback_data.csv', index=False)
inventory_data.to_csv('/content/cleaned_inventory_data.csv', index=False)
items_data.to_csv('/content/cleaned_items_data.csv', index=False)
orders_data.to_csv('/content/cleaned_orders_data.csv', index=False)
recipe_data.to_csv('/content/cleaned_recipe_data.csv', index=False)



# Import files module for downloading
from google.colab import files

# Download the files
files.download('/content/cleaned_feedback_data.csv')
files.download('/content/cleaned_orders_data.csv')
files.download('/content/cleaned_items_data.csv')
files.download('/content/cleaned_inventory_data.csv')
files.download('/content/cleaned_recipe_data.csv')
```

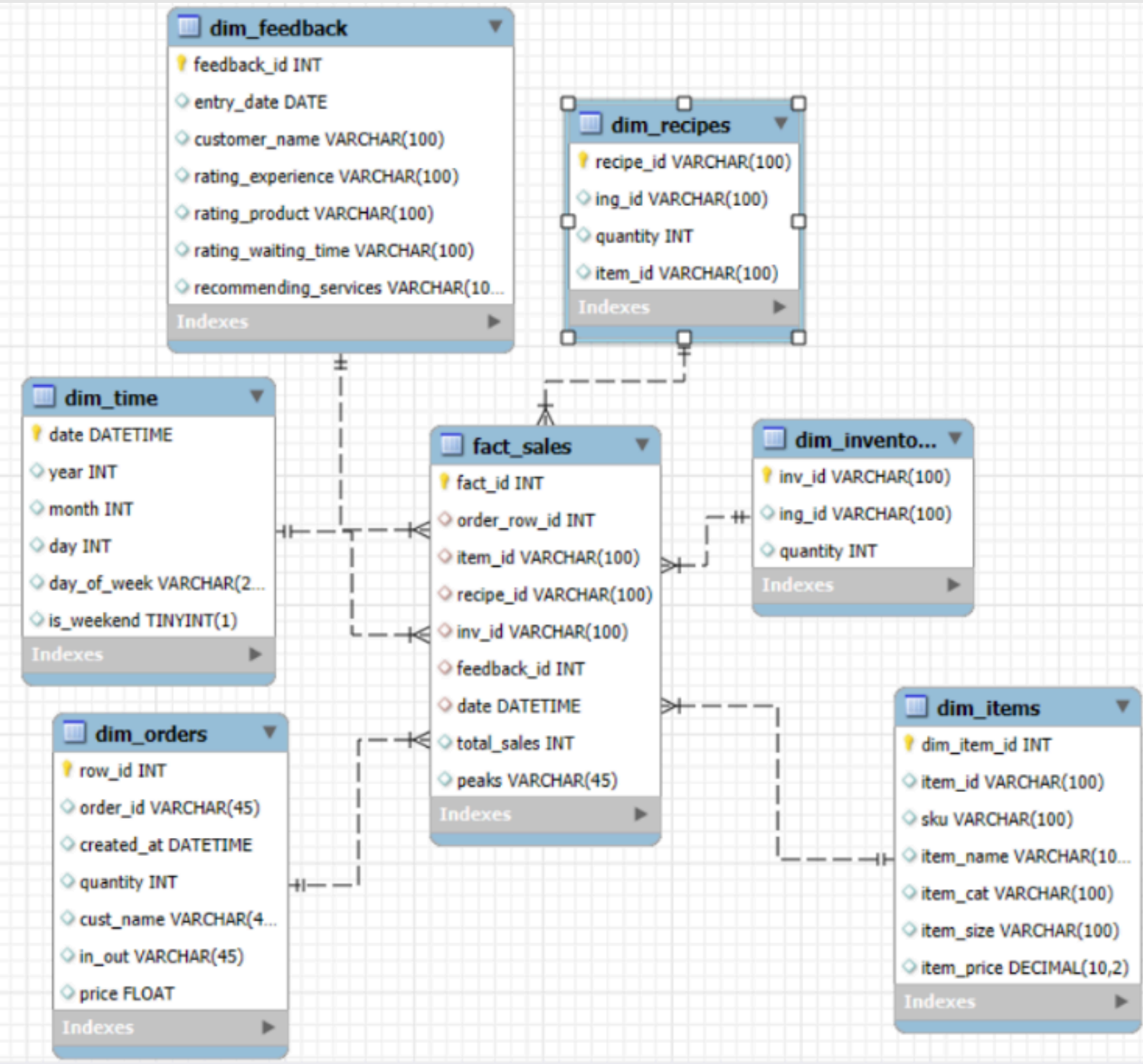
In this script, we save the cleaned data from various pandas DataFrames into CSV files using the to_csv function. These files represent the processed datasets, such as customer feedback, orders, inventory, items, and recipes. Once the files are created in the Google Colab environment, they are made available for download using the files.download function from the google.colab module. This step allows us to retrieve the cleaned data locally for further use, such as loading it into a MySQL database for storage and analysis.

Data modeling

In this section, we delve into the data modeling process, focusing on structuring our dataset to enable efficient analysis and reporting. The primary components of our data model include the Fact Table, Dimensions, Measures, and the chosen schema. This careful design aims to provide a comprehensive foundation for deriving valuable insights from the coffee shop dataset.

work explanation

Star Schema



work explanation

Data modeling

2.3.1.Fact table Review

fact_sales

Purpose: Stores transactional data for orders, including sales and customer interactions.

column	Data type	Description
fact_id	INT	Unique identifier for each sale
order_row_id	VARCHAR	ID for the associated order
item_Id	VARCHAR(100)	identifier for the sold item
feedback_id	int	identifier for the feedback
recipe_id	VARCHAR	identifier for the recipe
date	DATETIME	links to the time dimension
Inv_id	Varchar(100)	IDentifier for the inventory
total_sales	INT	total sales
Peaks	VARCHAR(100)	Time that

work explanation

Data modeling

2.3.2.dimension tables

Recipes_dimension: Stores recipe details for items, including ingredients and quantities.

column	Data type	Description
recipe_id	VARCHAR	Unique identifier for The recipe
ing_Id	VARCHAR	unique identifier for the ingredient
Quantity	INT	Quantity of ingredient needed

orders_dimension: provides detailed information about individual customer orders

column	Data type	Description
order_id	INT	Unique identifier for The Orders
created_at	DATETIME	The date and time when the order was placed
Quantit	int	The number of items included in the order.
cust_name	VARCHAR(45)	Name of the customer Who placed the order
in_out	VARCHAR(45)	Indicates whether the order was dine-in or take out
Price	FLOAT	Total price of the order

work explanation

Data modeling

2.3.2.dimension tables

Inventory_dimension: Tracks the availability of ingredients in inventory.

column	Data type	Description
inv_Id	VARCHAR(100)	Unique identifier for The inventory
ing_id	VARCHAR	unique identifier for the ingredient
Quantity	INT	Quantity available in the inventory

items_dimension: provide descriptive data about items available for orders

column	Data type	Description
item_Id	VARCHAR(100)	Unique identifier for each item
item_name	VARCHAR(100)	Name of the item
item_cat	VARCHAR(100)	category of the item
item_size	VARCHAR(100)	size or packaging details
item_price	DECIMAL(10,2)	price of the item

work explanation

Data modeling

2.3.2.dimension tables

Time_dimension: Enables time-based analysis of sales and operational trends.

column	Data type	Description
Date	Date	Full date
day	INT	Day of the month
month	INT	Month of the year
year	INT	Year of the transaction
weekday	VARCHAR	Day of the week

Feedback_dimension: Captures customer feedback to analyze satisfaction and service quality.

column	Data type	Description
feedback_id	VARCHAR	Unique identifier for The feedback
entry_date	DATE	Date when feedback was submitted
customer_name	VARCHAR(100)	Name of the customer providing feedback
rating_experience	VARCHAR(100)	rating of overall experience
rating_waiting_time	VARCHAR(100)	Rating for the waiting time
recommending_services	VARCHAR(100)	RECOMMENDATION LIKELIHOOD

work explanation

Data modeling

2.3.3.Measures :

Our analysis relies on quantitative metrics, referred to as Measures, to evaluate performance and identify trends. Within the context of our data model, key Measures include:

- **Total Sales:** This measure represents the total revenue generated from all orders, providing insights into overall business performance and sales trends.
- **Peaks:** This measure highlights the peak sales times throughout the day, enabling the identification of high-demand periods crucial for optimizing drive-through operations and resource allocation.

2.3.4 Schema Choice: star Schema :

The schema follows a star schema structure with **fact_sales** as the central fact table and the following dimension tables:

- dim_items
- dim_orders
- dim_time
- dim_inventory
- dim_feedback
- dim_recipes

work explanation

OLAP and Data Visualization Process

We employed Power BI as our OLAP and Data Visualization tool. This powerful tool allowed us to create interactive and insightful reports, transforming raw data into meaningful visualizations. The key steps in this process include:

2.4.1. Data Connectivity

Power BI seamlessly connects to a variety of data sources. We linked our data warehouse, hosted on MySQL server, to Power BI, ensuring real-time access to the most up-to-date information. The direct connectivity facilitated smooth data extraction for reporting.

2.4.2. Data Import and Transformation

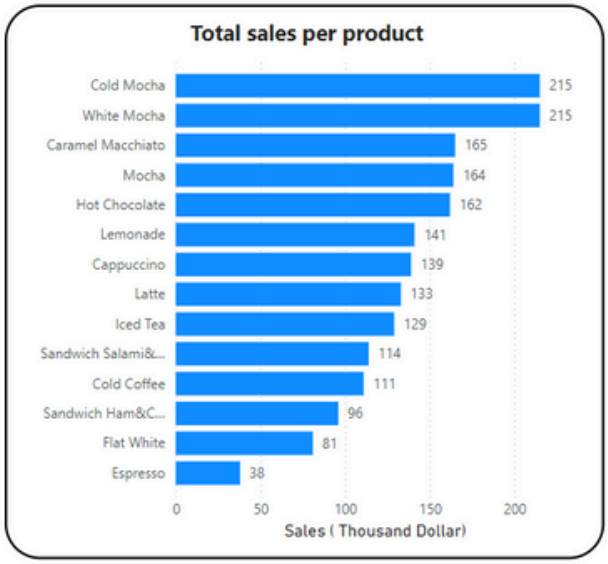
Once connected, we imported data from our MySQL server into Power BI. Leveraging Power BI's transformation capabilities, we refined and shaped the data to meet the specific requirements of our analysis. This step included handling any necessary data cleansing, filtering, or transformation to enhance the quality of our visualizations.

2.4.3. Data Modeling in Power BI

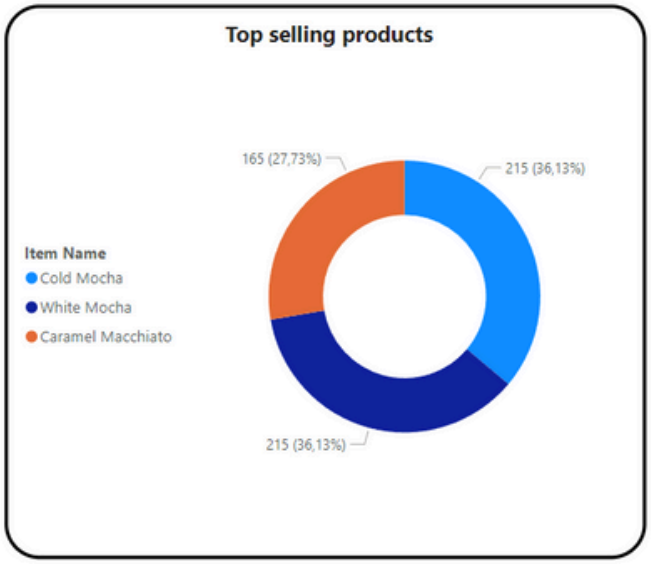
Power BI's robust data modeling features allowed us to define relationships between our Fact and Dimension tables. We mapped out the associations between the fact_sales table and the various Dimension tables , such as items,orders,inventory,time,recipes and feedback .

work explanation

Charts

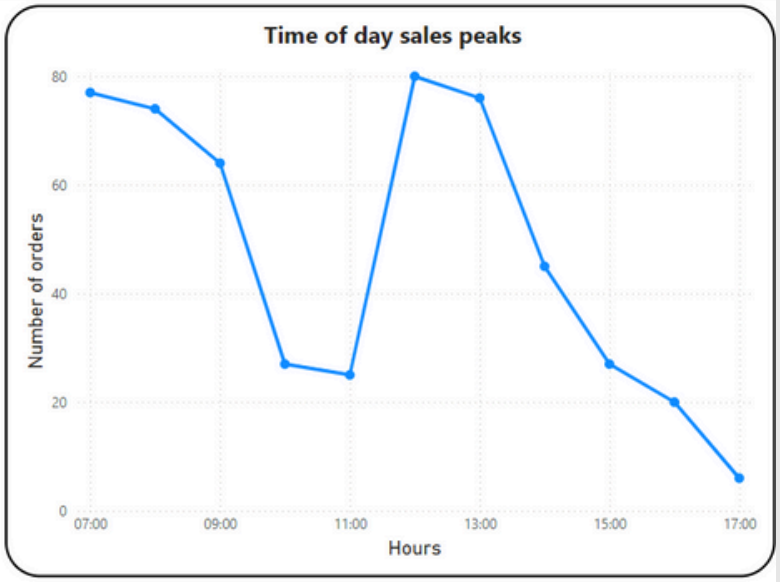


The "Total sales per product" graph shows the sales volumes for various menu items, with the top sellers being Cold Mocha, White Mocha, and Caramel Macchiato, each generating over $215,000 in sales.



The pie chart illustrates that Cold Mocha White Mocha and Caramel Machhiato are the two highest-selling items, comprising over 30% of total sales.

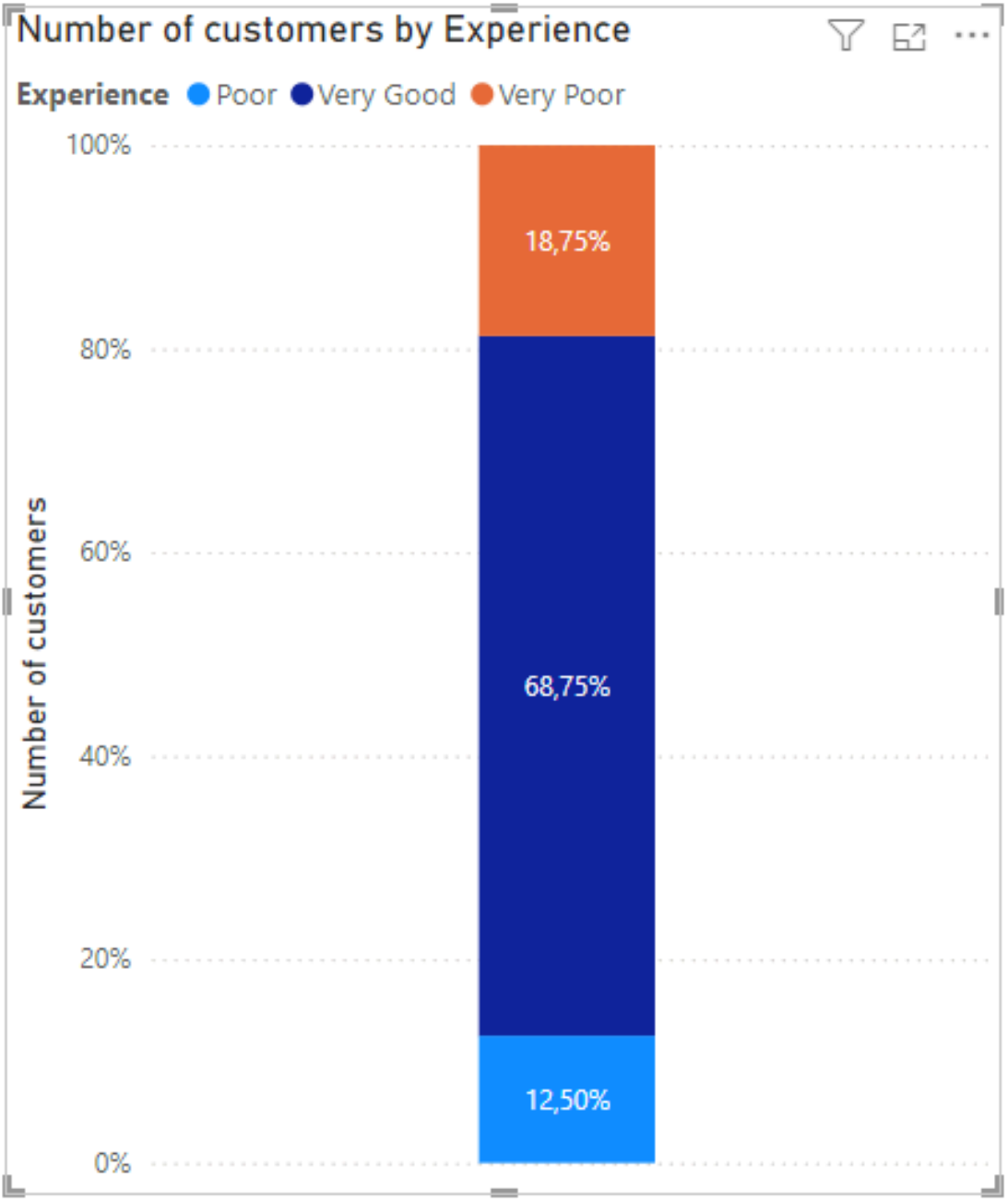
This graph illustrates that Monday is the best day for sales , with 114 orders on average , with Saturday as second with 84 orders, Tuesdays and Fridays also see high order volumes.



The graph indicates an initial decrease from 7AM to 11AM , followed by a steady increase in activity by 11AM until it reaches a peak at approximately 12:30 PM , and sharply declines from 1PM to 7PM.

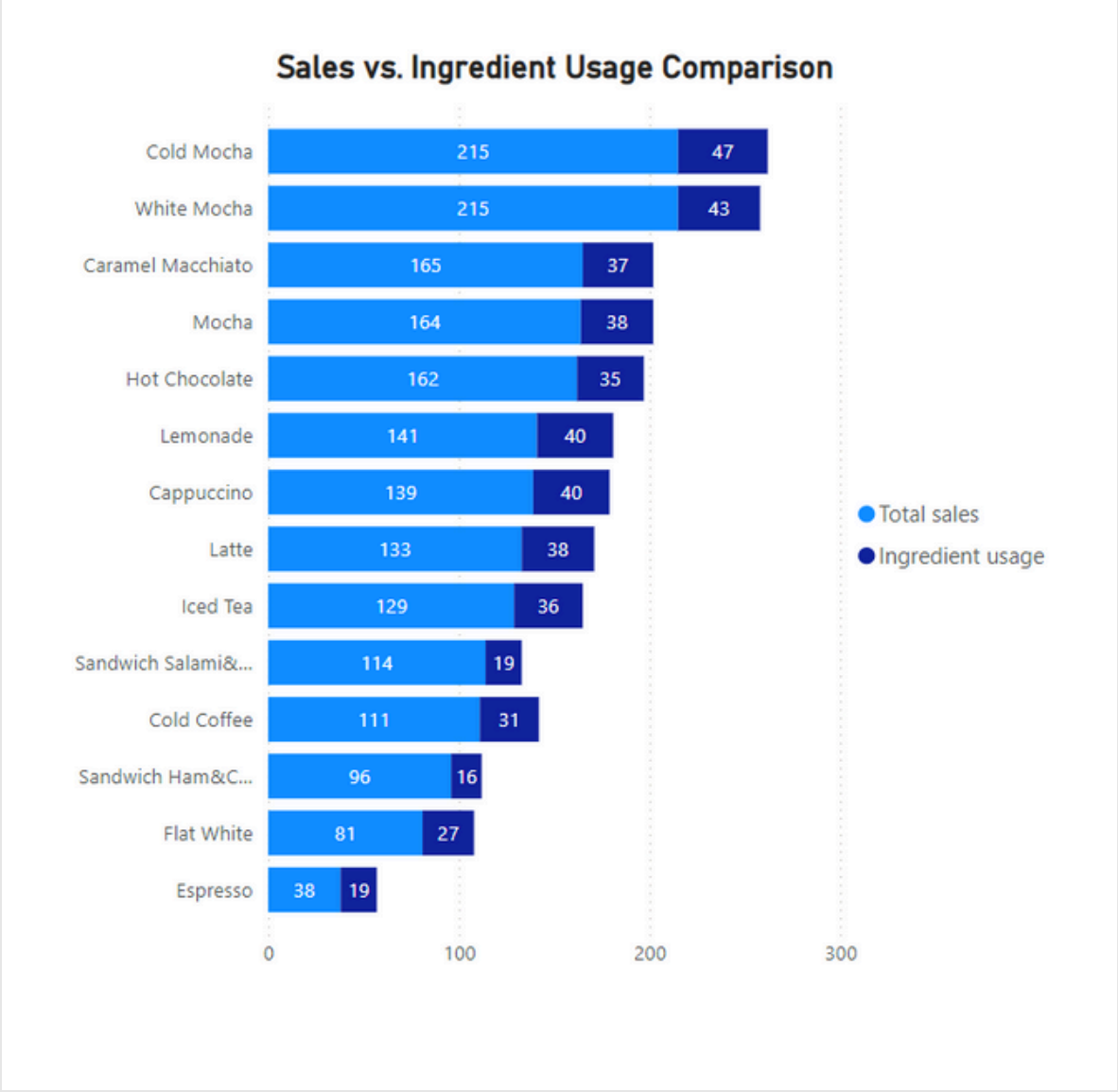
work explanation

Charts and conclusions



work explanation

Charts and conclusions



work explanation

Conclusions

High sales volumes for specific products and consistent customer engagement on peak days suggest a feasible drive-through implementation. The financial feasibility can be further assessed by comparing the average ticket size and potential increase in sales through convenience offerings.

1. Understanding Customer Preferences

A majority of customers are satisfied, indicating strong service quality. However, the 18.75% negative feedback should be addressed to improve overall customer experience in the drive-through.

Peak Times for Take-Out Orders:

- The "Time of day sales peaks" chart indicates:
  - Peak sales occur around 12:30 PM, suggesting lunch hours are the busiest.
  - There is another notable activity around 1 PM, with a decline after 3 PM.
- Conclusion:
- Lunch hours should be a priority for drive-through operations, requiring adequate staffing and preparation to meet high demand.

work explanation

Conclusions

2. Identifying High-Demand Products for Drive-Through

Most Selling Products:

- The top-selling products are:
  a. Cold Mocha (215K)
  b. White Mocha (215K)
  c. Caramel Macchiato (165K)
- Conclusion:
- These items should be introduced and prioritized in the drive-through menu due to their high demand.

3. Optimizing Operations for Drive-Through Implementation

Drive-through operations should be optimized to handle peak demand on Mondays and Saturdays, with potential promotions during slow days.

Thank you !