Data Analysis
DataCo Smart Supply Chain

November 21, 2025

Project Timeline

1. Choose dataset from the Kaggle platform

Python connection PostgreSql

2. Creating a dedicated project environment and setting up the PostgreSQL database.

3. Integrating data using Python, then performing detailed analysis and generating insightful data visualizations.

4. Developing a final report outlining key findings, and providing actionable recommendations based on the data analysis.

Our Technology Stack

Data Processing

Python + Pandas

Efficient data manipulation, cleaning, andanalysis, forming our data processing backbone.

Database

PostgreSQL
with the **psycopg2**

Enabling communication between our Python applications and the database.

Visualization

Matplotlib & Seaborn

For extensive customization and high-level interface to create insightful and aesthetically pleasing statistical graphics.

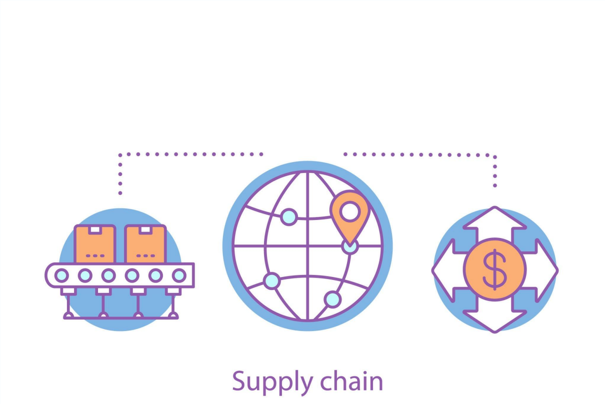
Collaboration

GitHub

Allowing our team to work together efficiently, manage code changes, and maintain project history.

Supply Chain Data Analysis

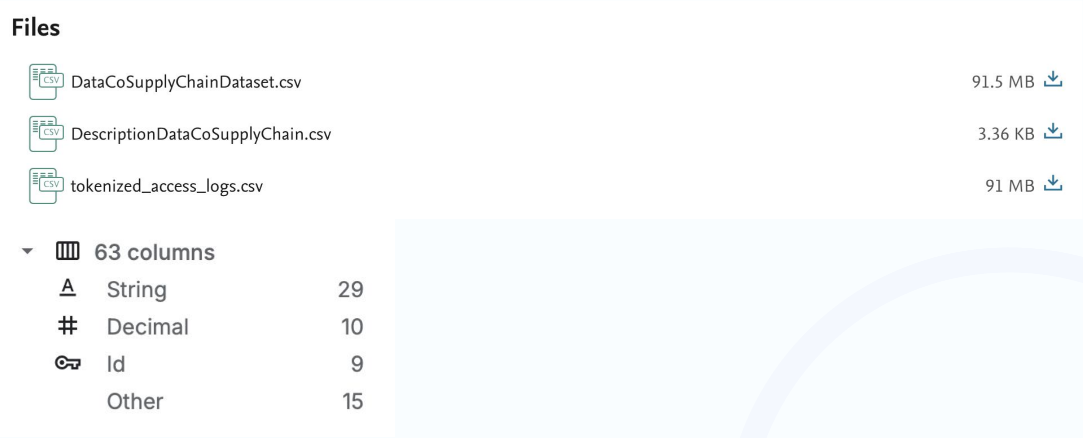
“DataCo Smart Supply Chain for Big Data Analysis” from Kaggle



URL: https://www.kaggle.com/datasets/shashwatwork/dataco-smart-supply-chain-for-big-data-analysis

GOAL

Focuses on analyzing supply chain data, and identify key trends and patterns, risks, and opportunities for business improvement.



Files

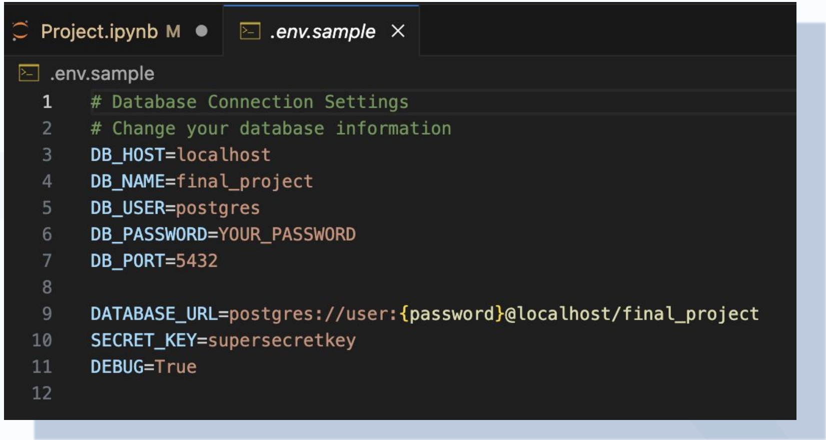
DataCoSupplyChainDataset.csv	91.5 MB
DescriptionDataCoSupplyChain.csv	3.36 KB
tokenized_access_logs.csv	91 MB

63 columns

String	29
Decimal	10
Id	9
Other	15

Python Integration

Create Environment & Database Setup

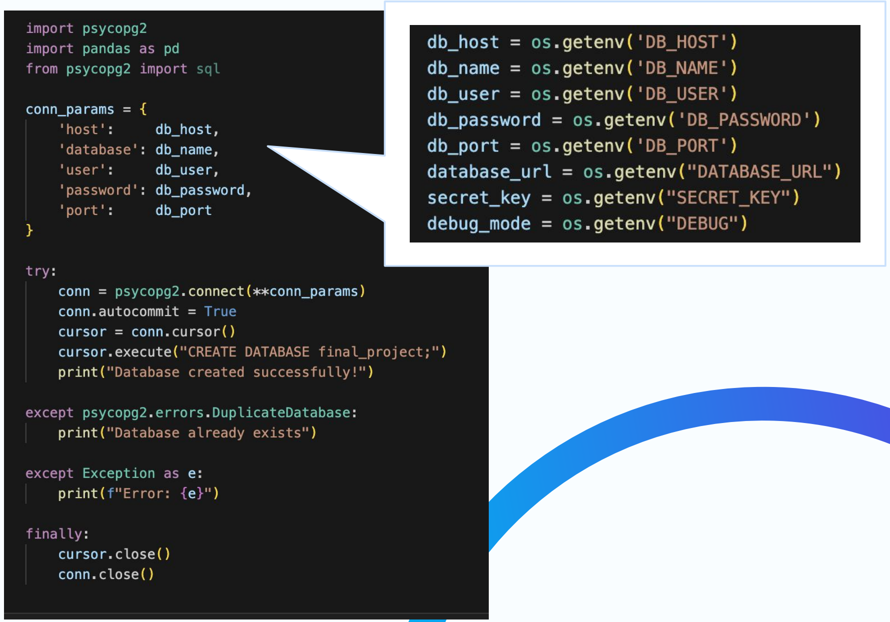


```
# Database Connection Settings
# Change your database information
DB_HOST=localhost
DB_NAME=final_project
DB_USER=postgres
DB_PASSWORD=YOUR_PASSWORD
DB_PORT=5432

DATABASE_URL=postgres://user:{password}@localhost/final_project
SECRET_KEY=supersecretkey
DEBUG=True

```

Each team member can use their own `.env` file

→ Keeps sensitive information secure and prevents hardcoding passwords in code

→ Allows easy switching between different database environments



```python
import psycopg2
import pandas as pd
from psycopg2 import sql

conn_params = {
    'host':     db_host,
    'database': db_name,
    'user':     db_user,
    'password': db_password,
    'port':     db_port
}

try:
    conn = psycopg2.connect(**conn_params)
    conn.autocommit = True
    cursor = conn.cursor()
    cursor.execute("CREATE DATABASE final_project;")
    print("Database created successfully!")

except psycopg2.errors.DuplicateDatabase:
    print("Database already exists")

except Exception as e:
    print(f"Error: {e}")

finally:
    cursor.close()
    conn.close()
```

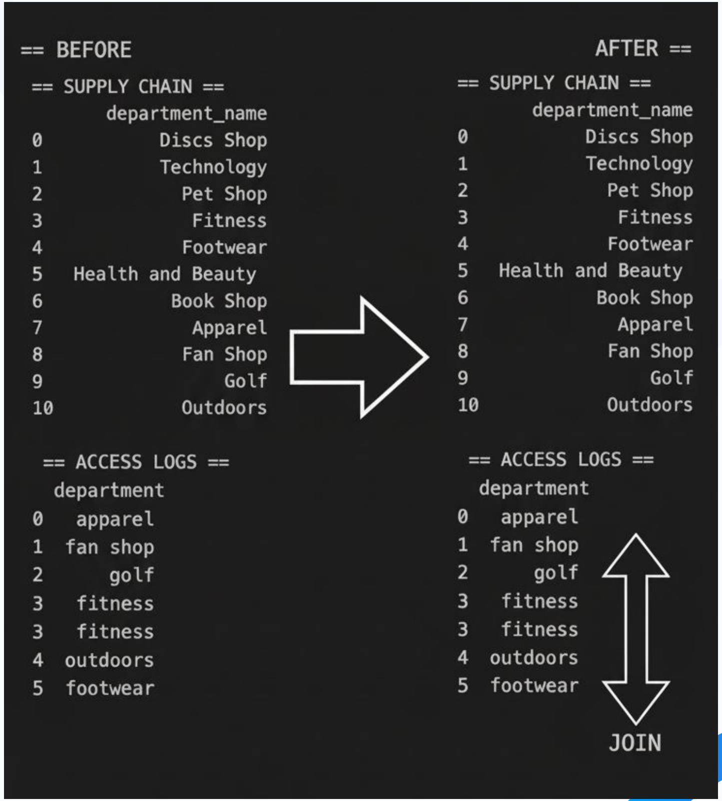
```python
db_host = os.getenv('DB_HOST')
db_name = os.getenv('DB_NAME')
db_user = os.getenv('DB_USER')
db_password = os.getenv('DB_PASSWORD')
db_port = os.getenv('DB_PORT')
database_url = os.getenv("DATABASE_URL")
secret_key = os.getenv("SECRET_KEY")
debug_mode = os.getenv("DEBUG")
```

Python Integration

Combining Data with JOINs

```
# Convert to lowercase all
for col in
supply_chain_df.select_dtypes(include=['object']).columns:
    supply_chain_df[col] = supply_chain_df[col].str.lower()
```

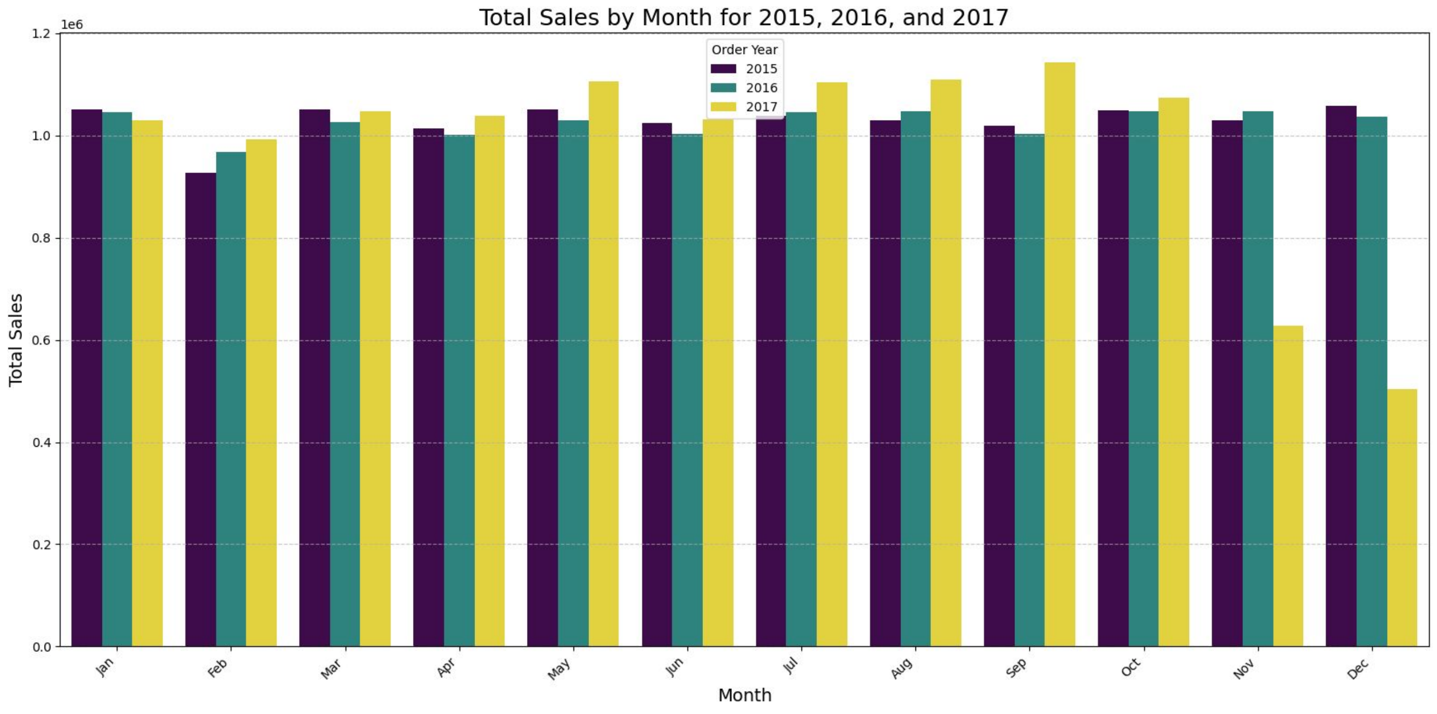
```
 # JOIN
join_sql = """
  SELECT *
  FROM  supply_chain_df s
  LEFT JOIN access_log_df a
  ON s.department_name = a.department;
  """
df_joined = pd.read_sql_query(join_sql, conn)
df_joined
```



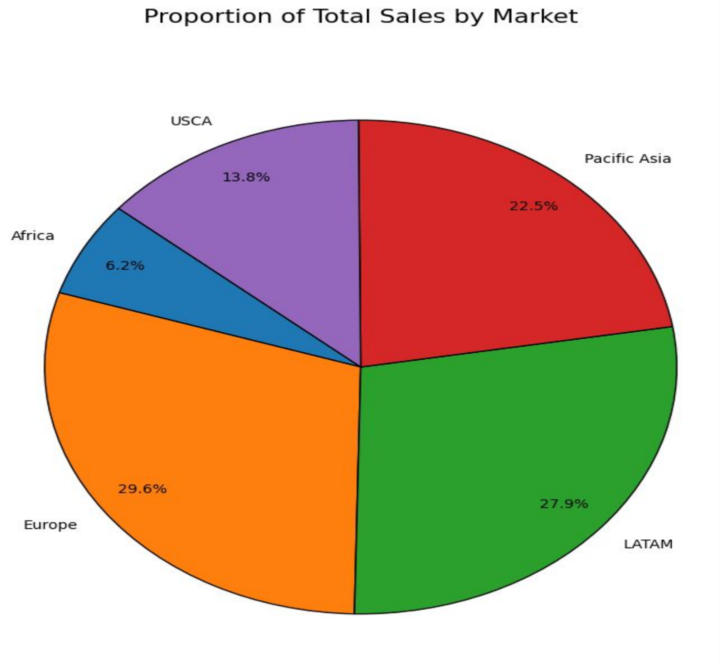
```
== BEFORE                                            AFTER ==
 == SUPPLY CHAIN ==                        == SUPPLY CHAIN ==
           department_name                           department_name
 0              Discs Shop                 0              Discs Shop
 1              Technology                 1              Technology
 2                Pet Shop                 2                Pet Shop
 3                 Fitness                 3                 Fitness
 4                Footwear                 4                Footwear
 5      Health and Beauty                  5      Health and Beauty
 6               Book Shop                 6               Book Shop
 7                 Apparel                 7                 Apparel
 8                Fan Shop                 8                Fan Shop
 9                    Golf                 9                    Golf
 10               Outdoors                 10               Outdoors

  == ACCESS LOGS ==                         == ACCESS LOGS ==
   department                                department
 0   apparel                               0   apparel
 1  fan shop                               1  fan shop
 2      golf                               2      golf
 3   fitness                               3   fitness
 3   fitness                               3   fitness
 4  outdoors                               4  outdoors
 5  footwear                               5  footwear

                                                     JOIN
```

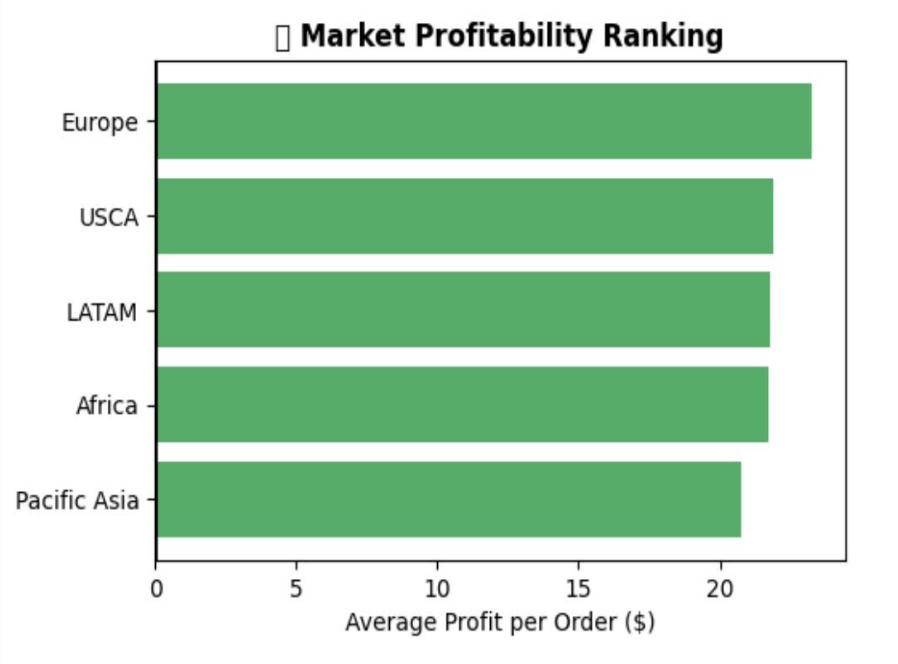
Data Visualization

Date Analysis

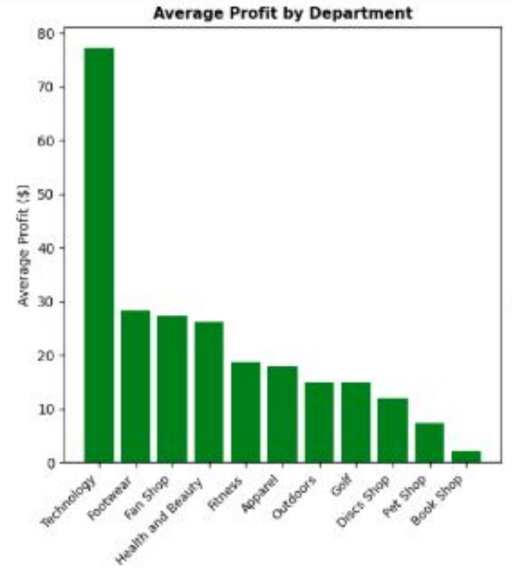


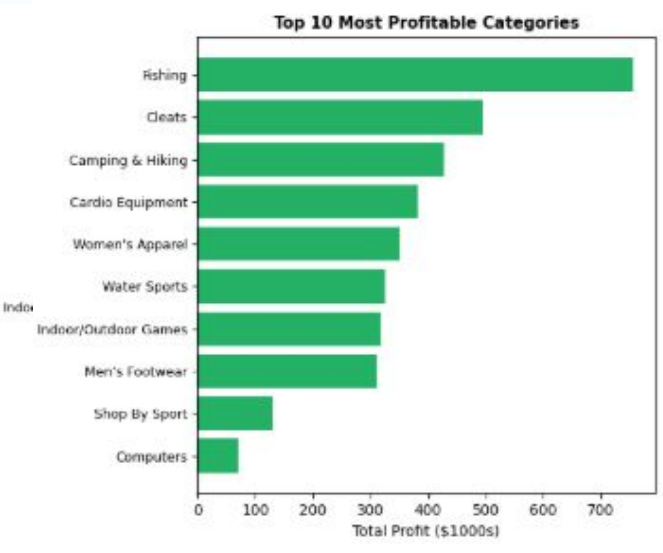
Market Analysis

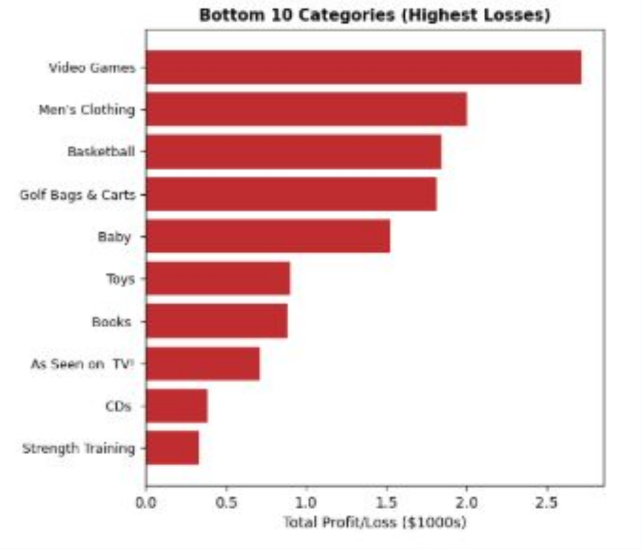




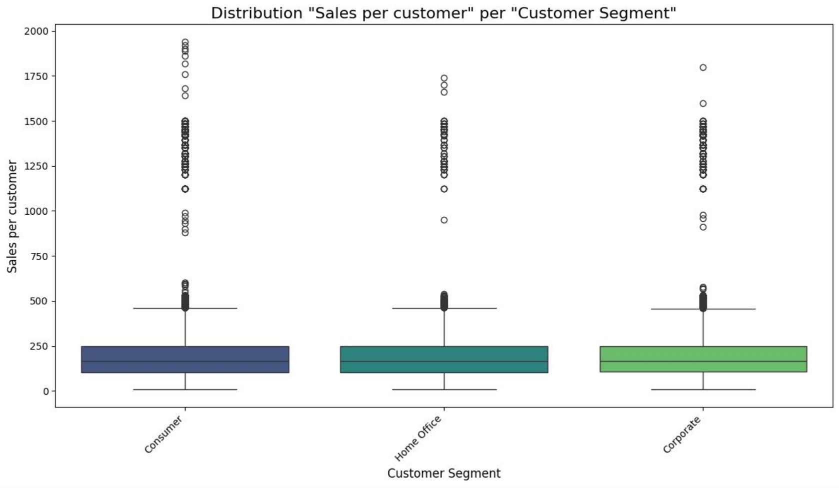
Product & Category Performance Analysis

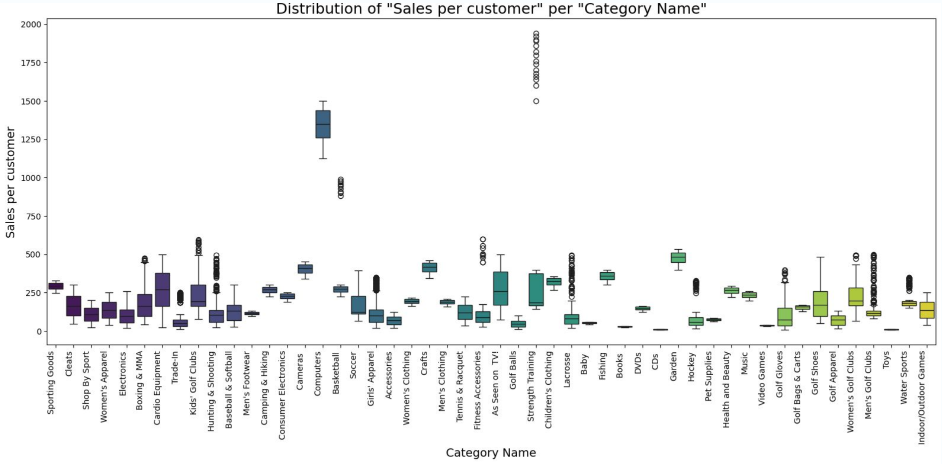




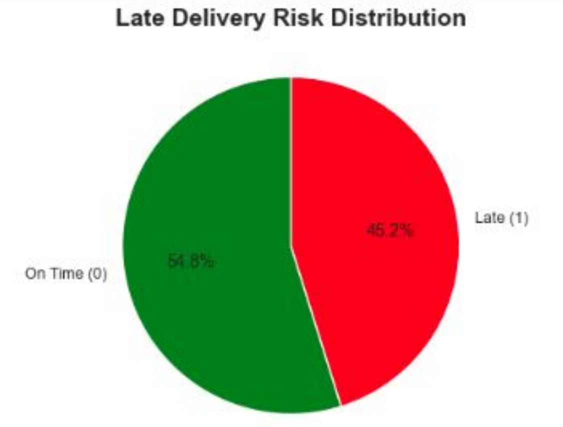


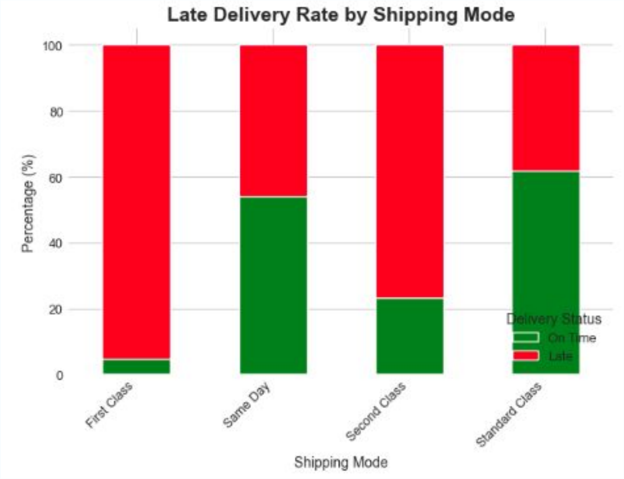
Sales per Customer Analysis

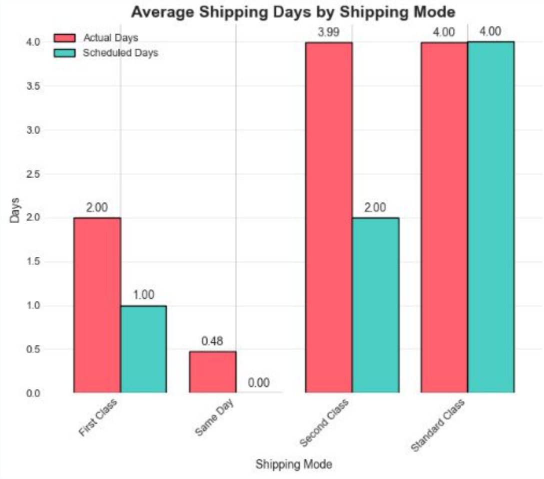




Delivery Analysis


Late Delivery Risk Distribution


Late Delivery Rate by Shipping Mode


Average Shipping Days by Shipping Mode

Business Recommendation

Actionable Recommendations

Operational Excellence

Customer Experience

Category Strategy

Actionable Recommendations

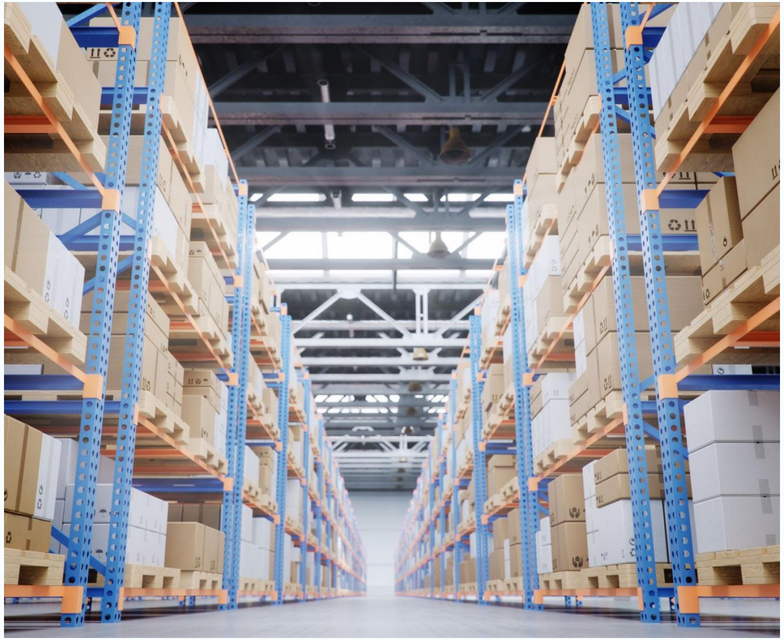
Operational Excellence

Customer Experience

Category Strategy

Actionable Recommendations

Operational Excellence



45.2% late delivery happened — our highest exposure area. This directly impacts customer trust and drives support costs.

Immediate actions:

- Implement predictive late-delivery alert system
- Optimize fulfillment center staffing during market-specific peak hours
- Review carrier partnerships and SLA compliance

Actionable Recommendations

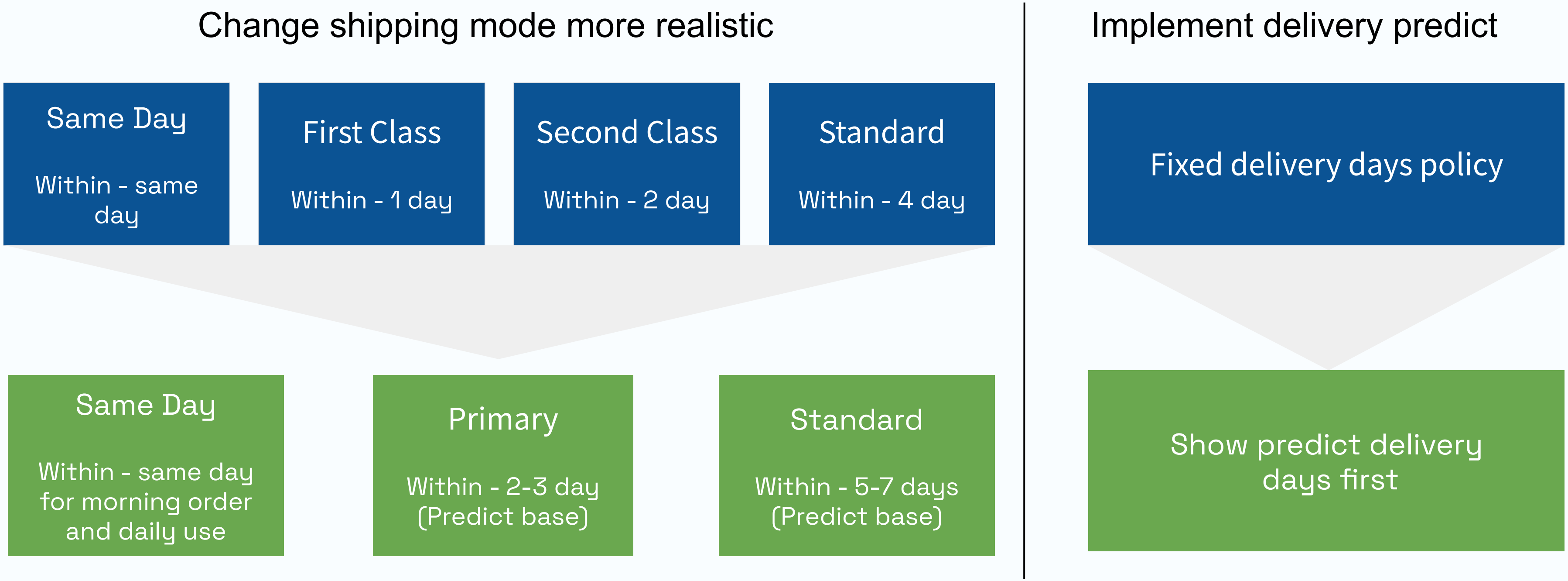
Operational Excellence

Customer Experience

Category Strategy

Actionable Recommendations

Customer Experience - Delivery



Change shipping mode more realistic

Same Day

Within - same day

First Class

Within - 1 day

Second Class

Within - 2 day

Standard

Within - 4 day

Same Day

Within - same day for morning order and daily use

Primary

Within - 2-3 day (Predict base)

Standard

Within - 5-7 days (Predict base)

Implement delivery predict

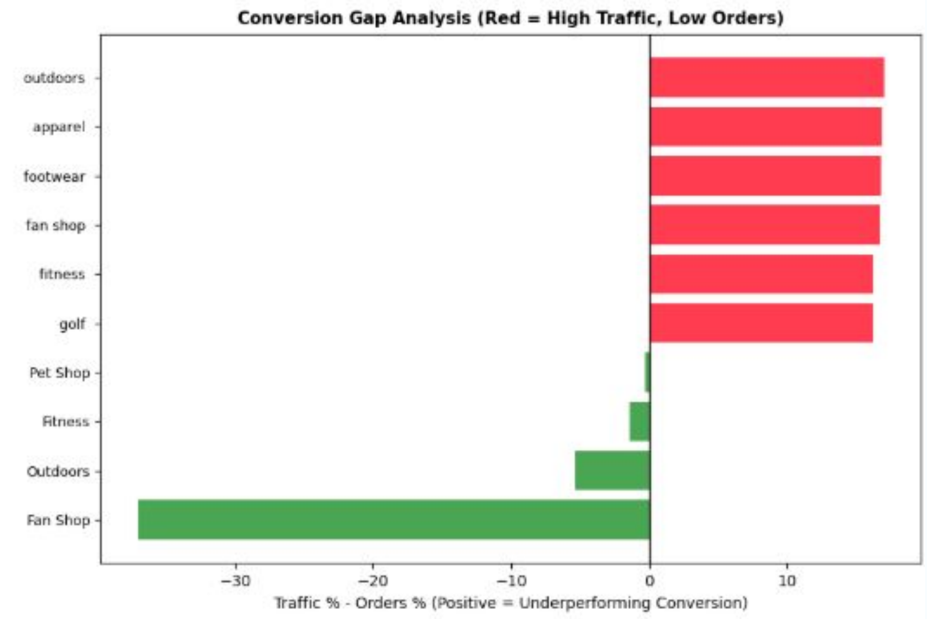
Fixed delivery days policy

Show predict delivery days first

Actionable Recommendations

Customer Experience - Web page

Some category lost a lot of opportunities on web traffic.
Should make more attractive web page or publish time sale coupon.



Actionable Recommendations

Operational Excellence

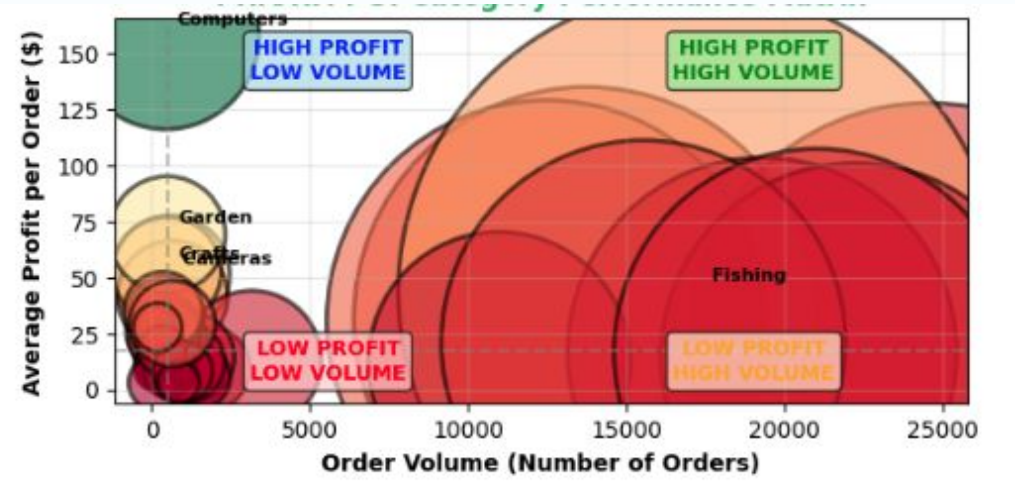
Customer Experience

Category Strategy

Actionable Recommendations

Category Strategy

- Reduce low performance category products
- Expand high profit low volume zone



Key Learnings

- How to share environment
- How we can make visualizations for big data
- How hard to handle big data includes a lot of columns
- Difficult to collaborate with others about data science rather than developing applications

Thank You & Questions