

PROJECT FASHION INSIGHT

Kendall, Lily, Taleen,
Karen, Kali →



README

Easy-To-Understand Project Overview in
Github Repository

[README.md](#)

BUSINESS PROBLEM

BACKGROUND:

- Alo Yoga is looking for an Analytics Manager to lead their analytics and growth strategies to drive business growth in their retail operations.

OBJECTIVES:

- Improve customer satisfaction, increasing market share
 - measured by average ratings of products

KEY RESULTS:

- Improve customer satisfaction by 10% within the next fiscal year



RAW DATASET

- Inputting missing values with median:
 - Purchase amount
 - Review rating

Fashion Retail Sales

Jupyter Notebook Cleaning Steps

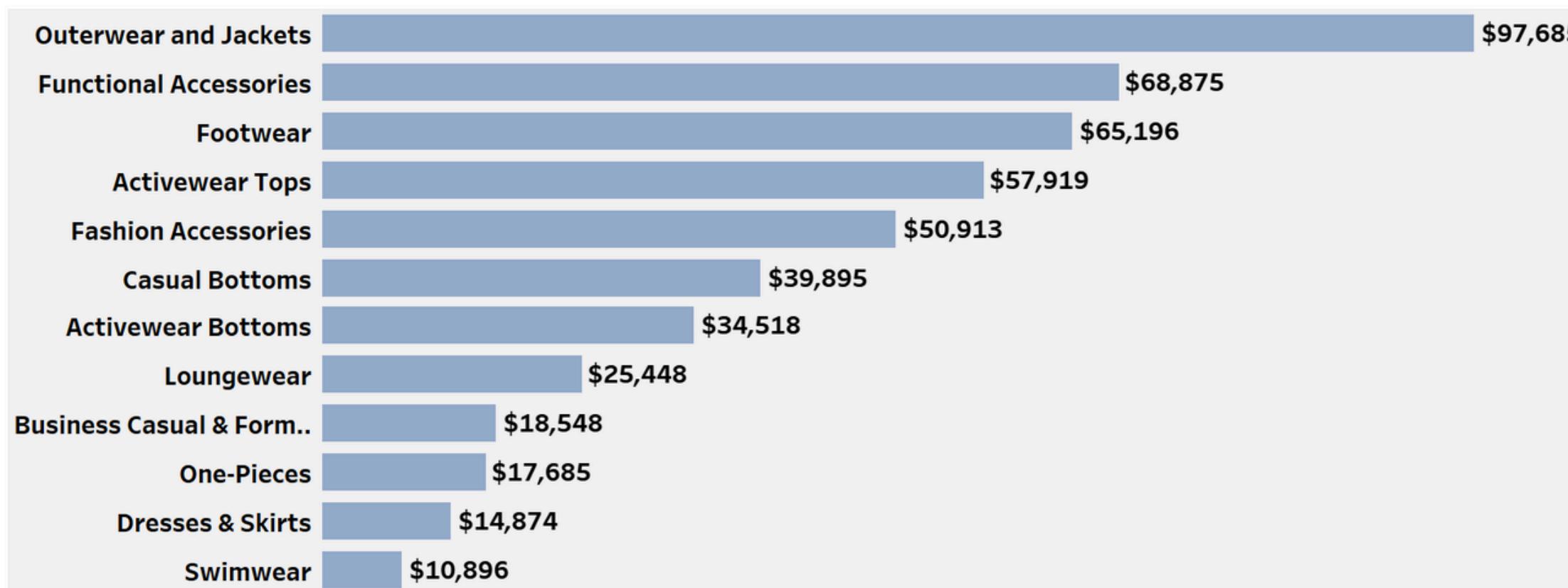
CLEANED DATASET

Fashion_Retail_Sales_Clean							
	Customer Reference ID	Item Purchased	Purchase Amount (USD)	Date Purchase	Review Rating	Payment Method	
0	4018	Handbag	4619.0	2023-02-05	3.2	Credit Card	
1	4115	Tunic	2456.0	2023-07-11	2.0	Credit Card	
2	4019	Tank Top	2102.0	2023-03-23	4.1	Cash	
3	4097	Leggings	3126.0	2023-03-15	3.2	Cash	
4	3997	Wallet	3003.0	2022-11-27	4.7	Cash	
5	4080	Onesie	2914.0	2022-12-11	4.5	Credit Card	
6	4055	Jacket	2571.0	2023-07-08	1.3	Cash	
7	3973	Trousers	2419.0	2022-11-10	4.6	Cash	
8	4044	Jeans	4771.0	2023-05-19	4.1	Cash	
9	4010	Loafers	4233.0	2023-06-11	2.7	Credit Card	
10	4108	Slippers	2356.0	2023-03-19	4.8	Credit Card	

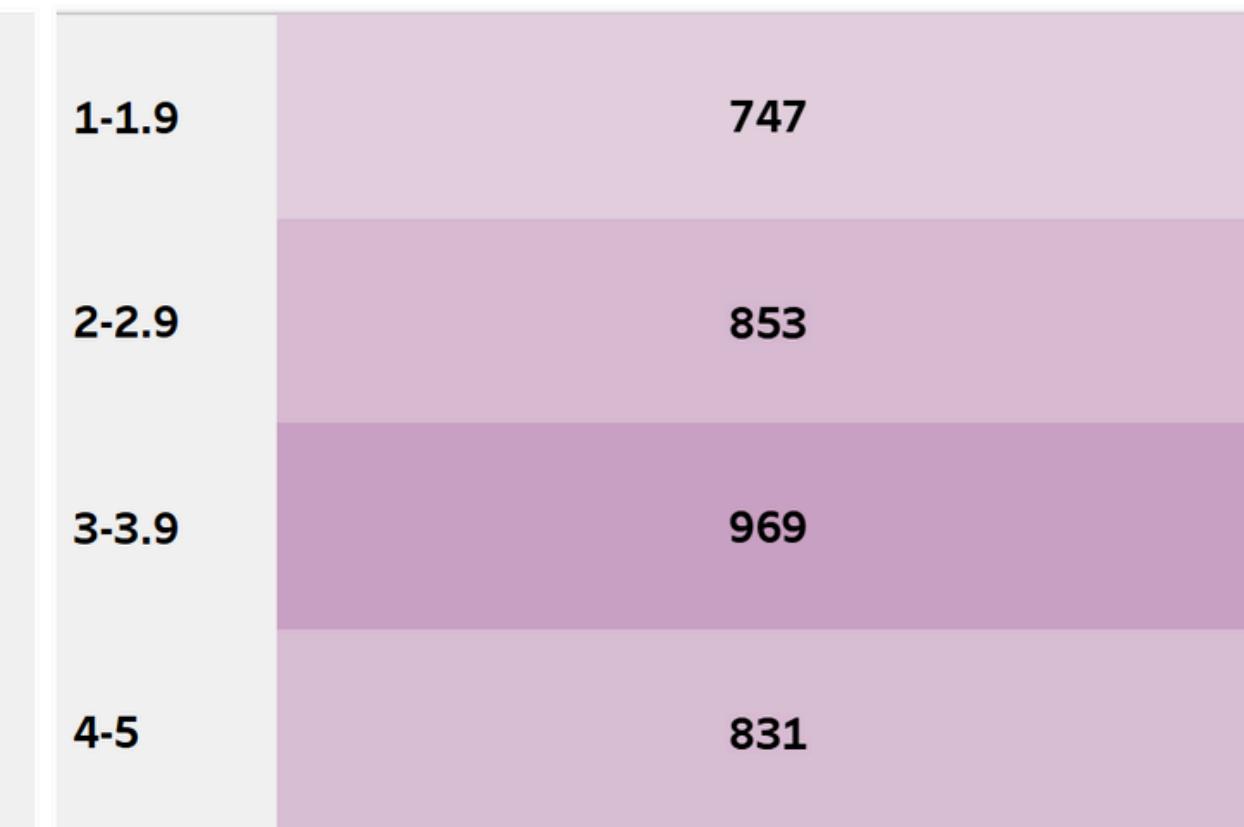
Fashion_Retail_Sales_Cleaned.csv

Fashion Company Rating and Revenue Correlation Dashboard

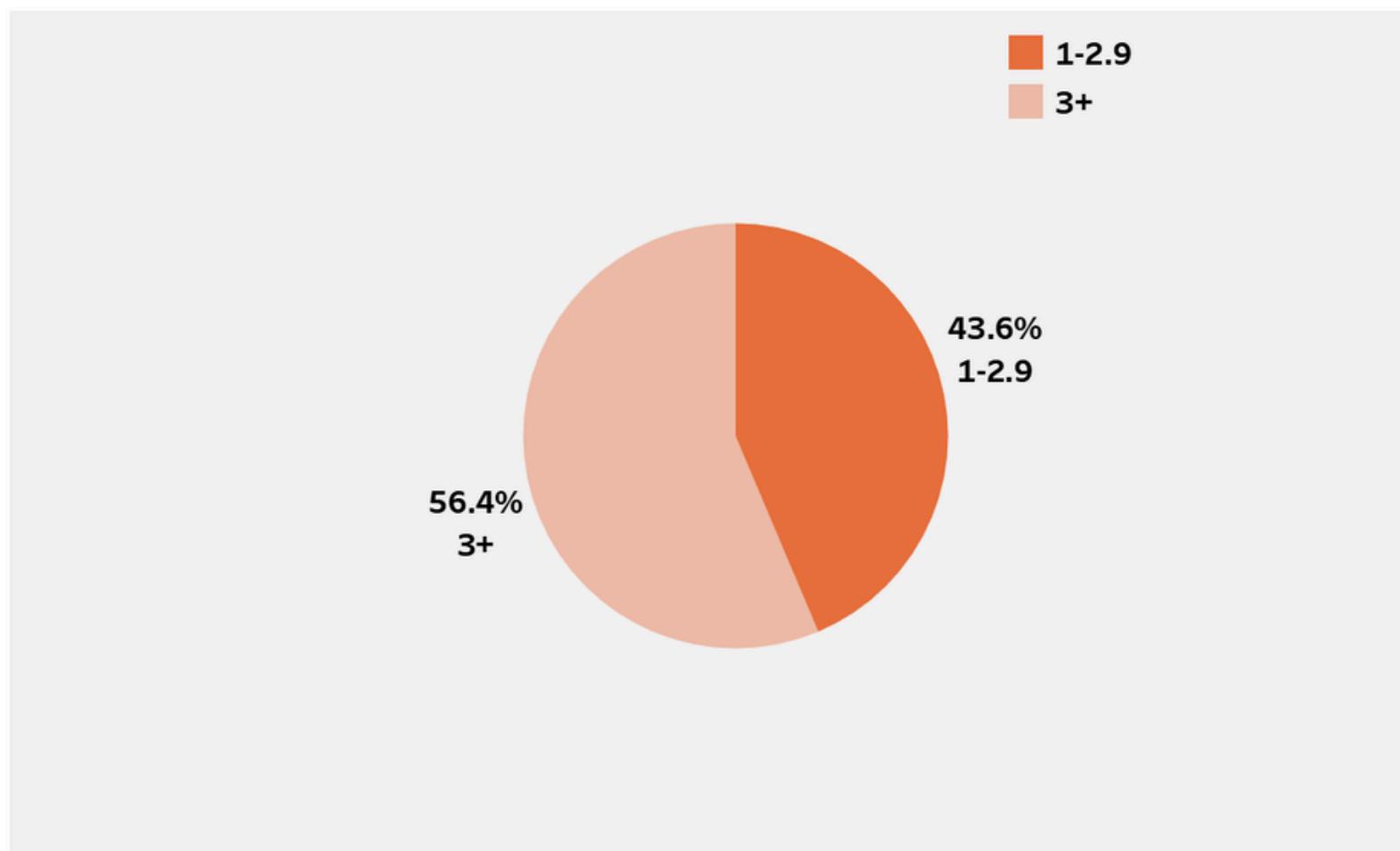
Total Revenue by Item Grouped



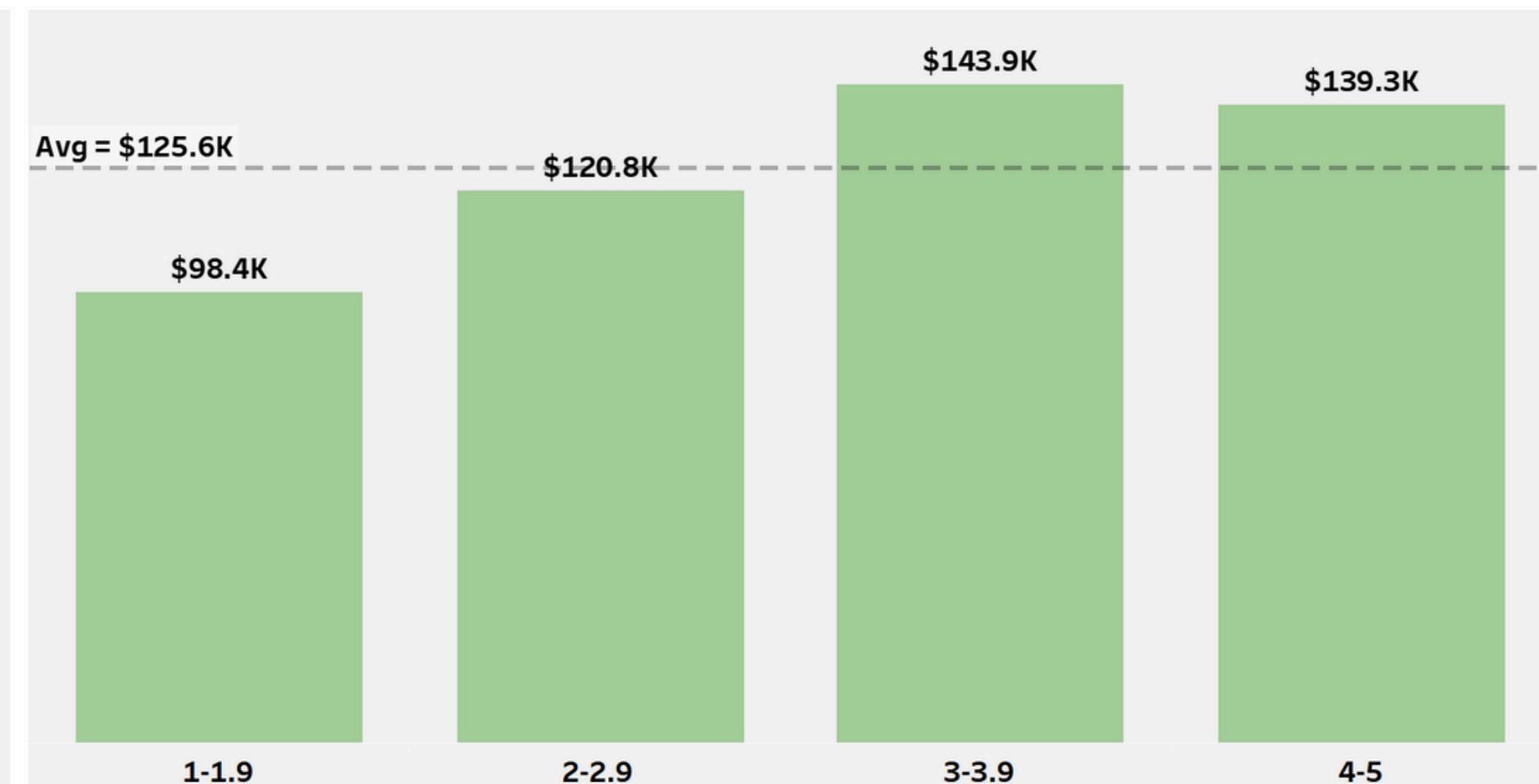
Count of Item Sold by Rating



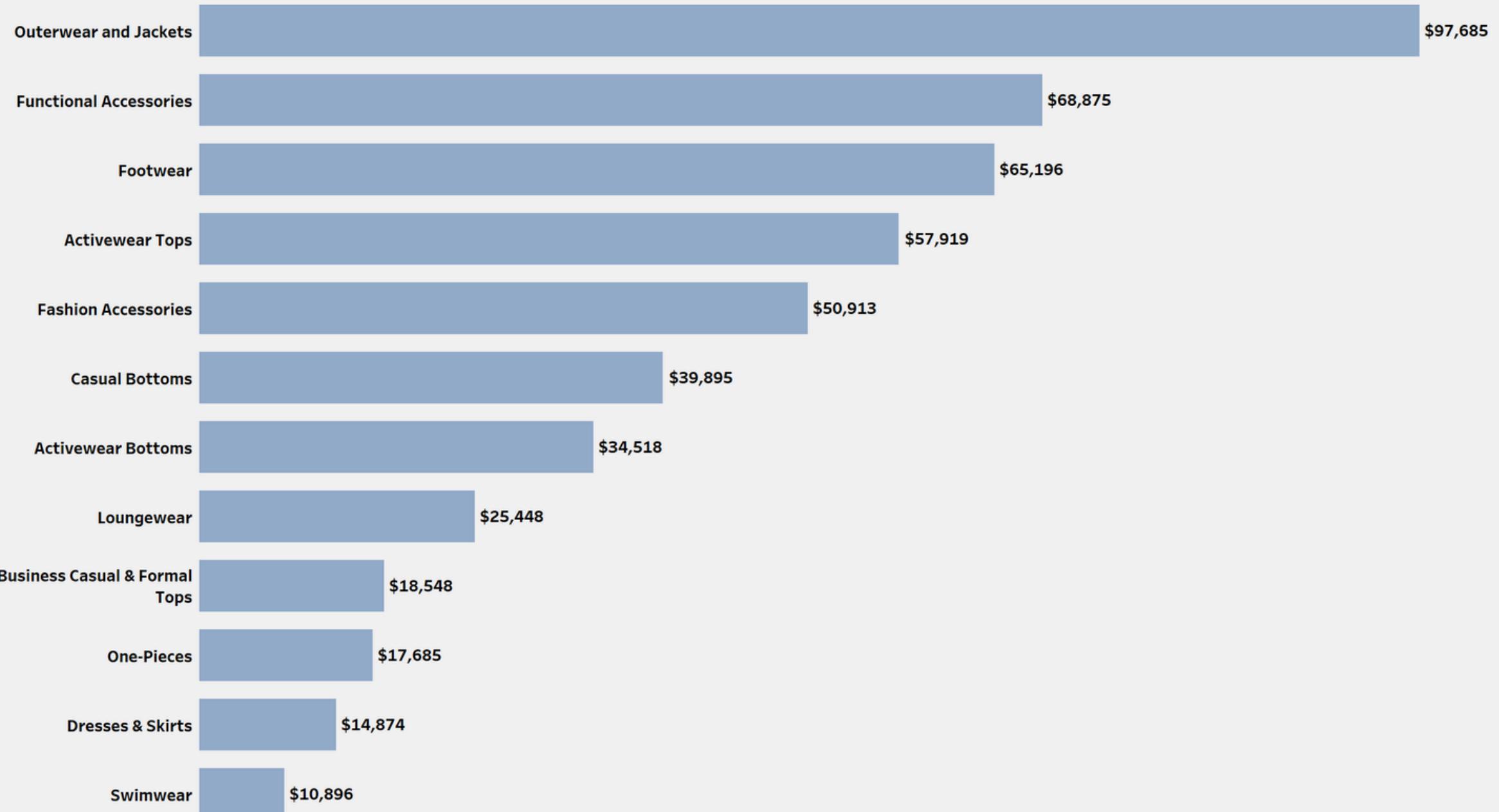
Percentage of Revenue by Rating



Total Revenue by Rating



TOP REVENUE CATEGORIES: OUTERWEAR & ACCESSORIES LEAD



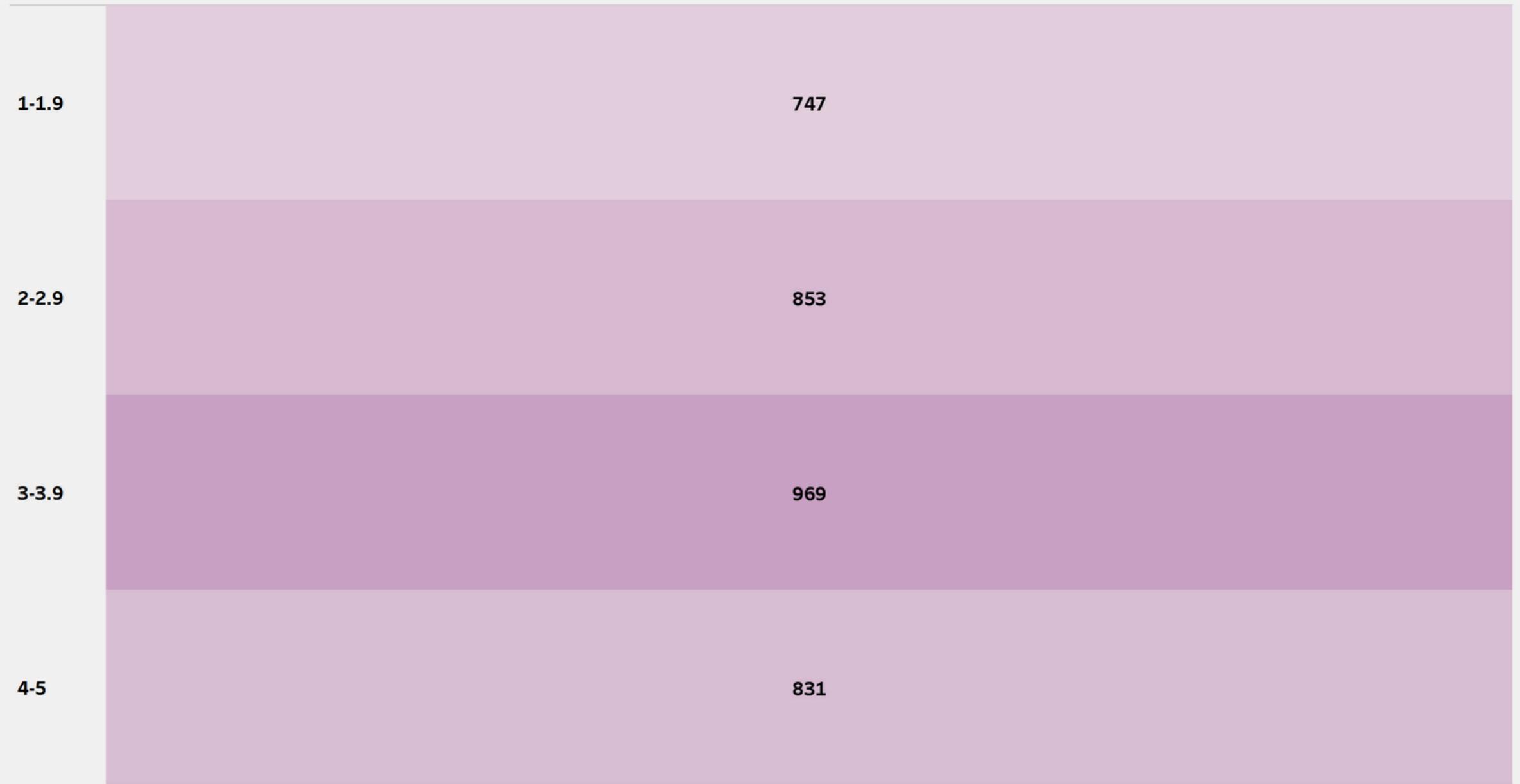
RECOMMENDATION:

- Focus on marketing and stocking from 'Outerwear and Jackets' and 'Accessories' categories

PREDICTION:

- Allocating resources to most profitable item groups will increase overall revenue

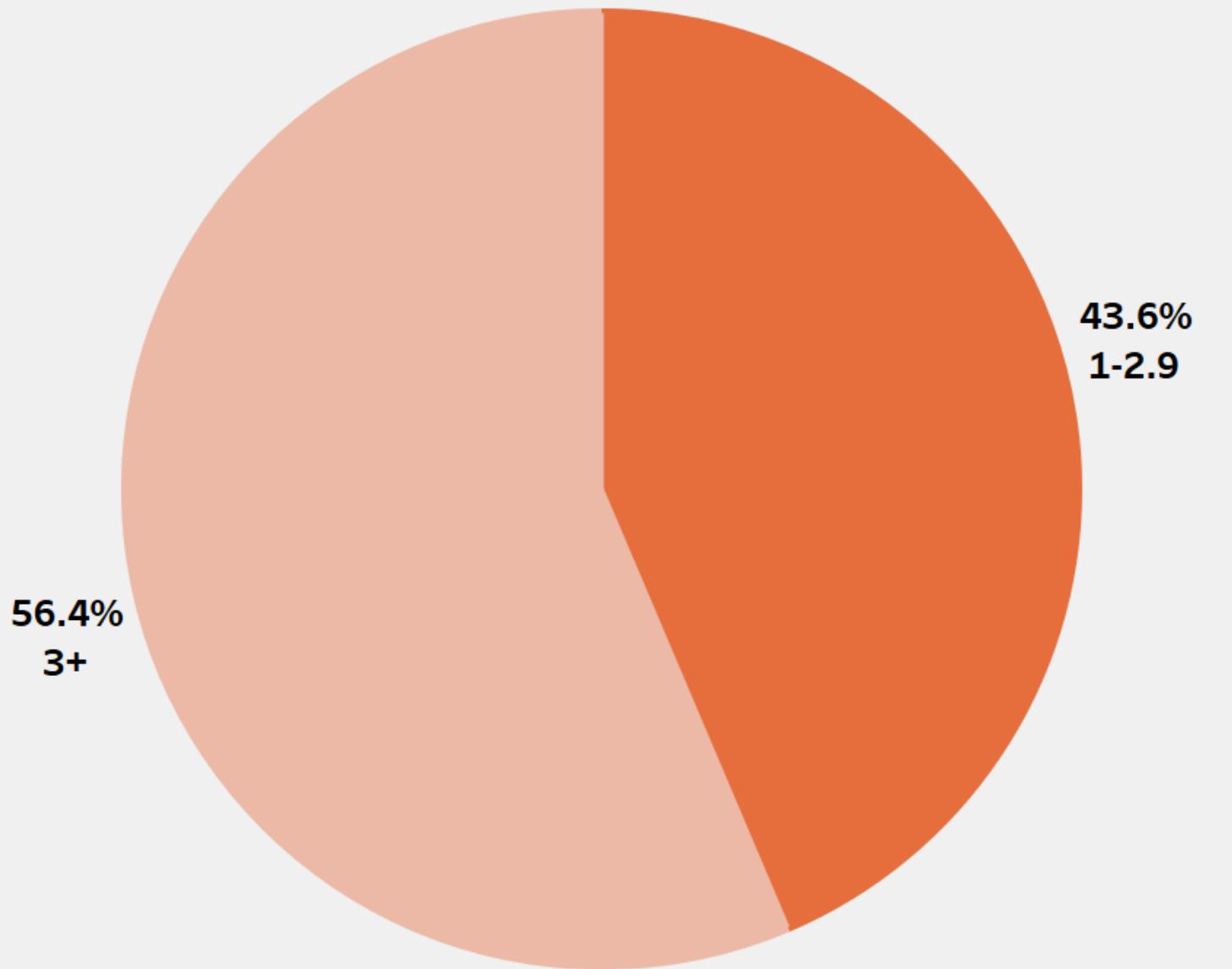
SALES VOLUME VS. SATISFACTION: MID-RANGE RATINGS DOMINATE



- **RECOMMENDATION:**
- Assess why mid-range ratings (3-3.9) are most sold
- For items with highest ratings (4-5), find strategies to increase sales volume to match rating levels

- **PREDICTION:**
- Adjusting focus to improve sales of high-rated items can lead to higher overall satisfaction rating and repeat purchases

REVENUE IMPACT OF RATINGS: HIGHER RATINGS CONTRIBUTE MORE



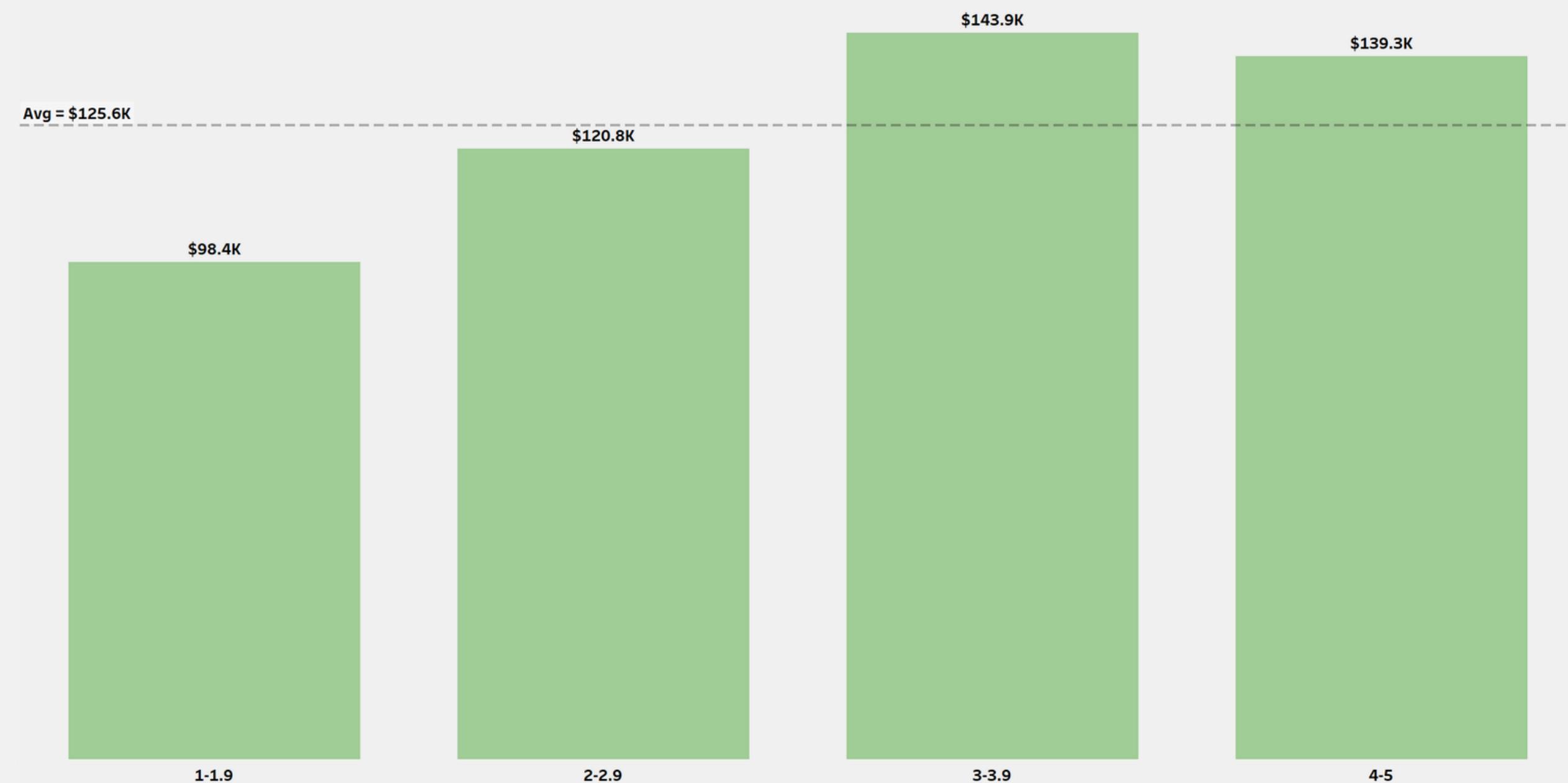
RECOMMENDATION:

- Investigate items in 1 -2.9 rating groups
- Improve quality of these lower rated items to enhance customer satisfaction and increase revenue

PREDICTION:

- Shifting revenue distribution towards higher ratings can increase customer retention and long-term revenue

HIGHER RATINGS CORRELATE TO HIGHER REVENUE



Recommendation:

- Prioritize enhancing customer experience and quality control for products to maintain or push ratings into higher brackets

Prediction:

- Elevating and maintaining high product ratings will likely lead to increased revenue per customer

DATA GOVERNANCE POLICY

Quality

- Data accuracy
- Reliability

Compliance

- Regular policy updates
- GDPR and CCPA

Security

- Safeguarding customer information
- Team training for secure data

Ethics

- Ethical Data use
- Ongoing Review

DATA DICTIONARY

Customer Reference ID:

- Unique identifies

Item Purchased:

- How many items bought

Purchase Amount:

- Higher spending could be an indicator of customer confidence in the brand

Date Purchase:

- Purchase dates can help track customer trends over time.
- Study seasons

Review Rating:

- Ratings are the clearest measure of customer satisfaction.

Payment Method:

- Possible correlate with purchase process ease

Data Dictionary for Fashion Retail Sales Dataset

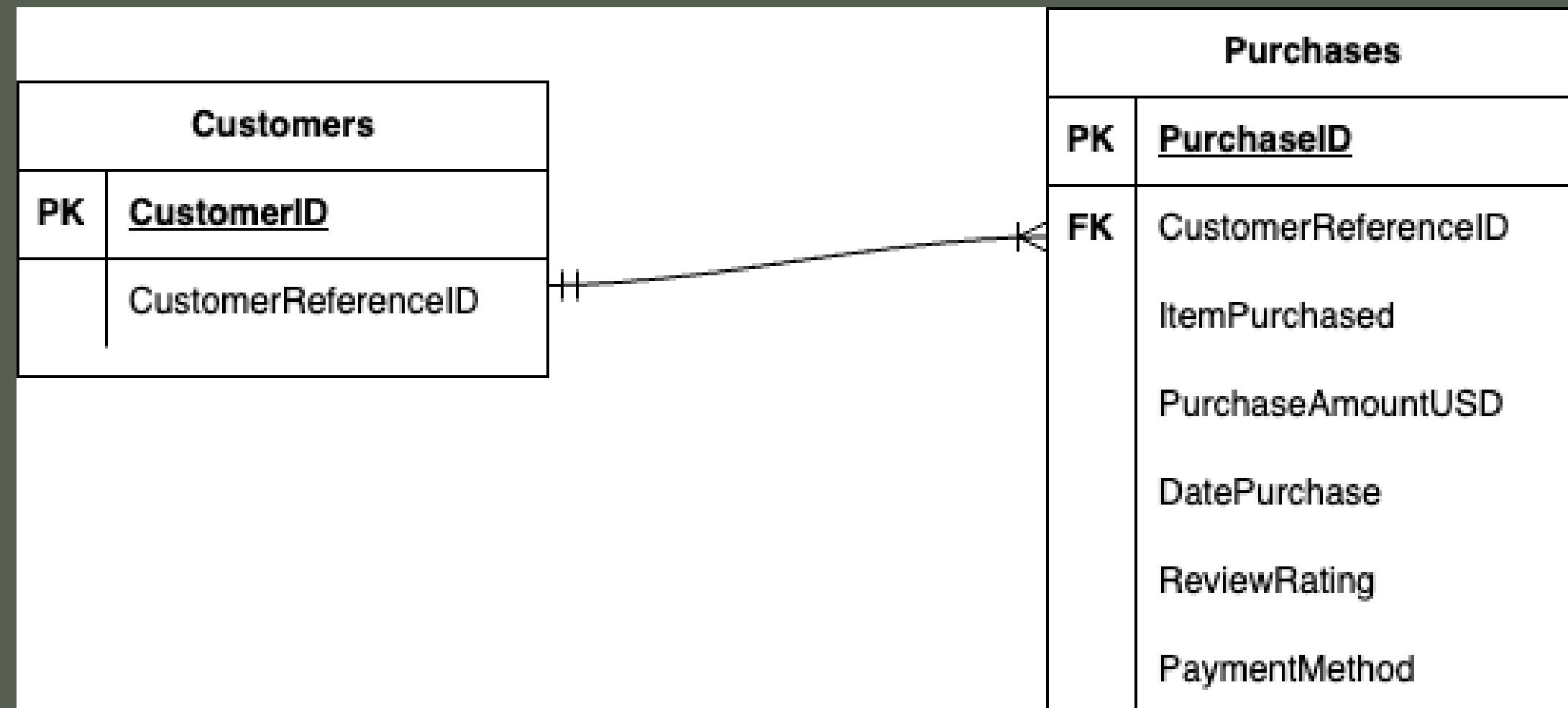
Dataset Name

Fashion Retail Sales

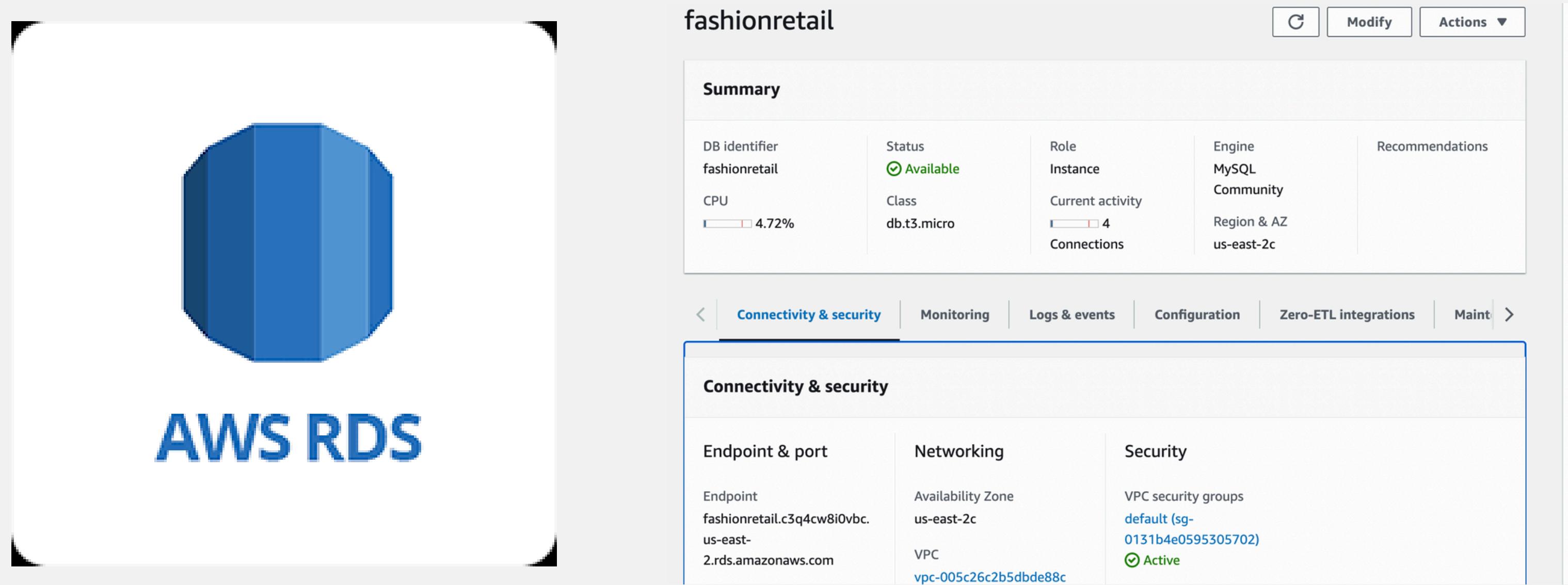
Data Attributes

Attribute Name	Data Type	Description	Metadata	Source	Owner	Last Updated	Notes
Customer Reference ID	Integer	Unique identifiers for customers		Retail transaction system	Fekih Mohammed el Amin	5 months ago	
Item Purchased	String	Clothing items bought by customers	Categories: T-shirts, jeans, accessories, etc.	Retail transaction system	Fekih Mohammed el Amin	5 months ago	
Purchase Amount (USD)	Float	Amount spent by customers for their purchases, in USD	May contain outliers for high-value purchases	Retail transaction system	Fekih Mohammed el Amin	5 months ago	
Date Purchase	Date	Date when each transaction occurred	Format: YYYY-MM-DD	Retail transaction system	Fekih Mohammed el Amin	5 months ago	Useful for analyzing buying trends and seasonality
Review Rating	Float	Customer satisfaction level with the product, from 1 (low) to 5 (high)	Range: 1.0 - 5.0	Customer feedback	Fekih Mohammed el Amin	5 months ago	Reflects product quality and customer experience
Payment Method	String	Method used by customers to make payments	Options: 'Credit Card', 'Cash', etc.	Retail transaction system	Fekih Mohammed el Amin	5 months ago	

ERD



AWS RDS MYSQL INSTANCE FOR DATAWAREHOUSE



The screenshot shows the AWS RDS console interface for managing a MySQL instance named "fashionretail".

Summary Tab:

DB identifier	Status	Role	Engine	Recommendations
fashionretail	Available	Instance	MySQL Community	
CPU	Class	Current activity		
4.72%	db.t3.micro	4		
		Connections		

Connectivity & security Tab:

Endpoint & port	Networking	Security
Endpoint fashionretail.c3q4cw8i0vbc. us-east- 2.rds.amazonaws.com	Availability Zone us-east-2c VPC vpc-005c26c2b5dbde88c	VPC security groups default (sg-0131b4e0595305702) <input checked="" type="checkbox"/> Active

ETL

```
In [ ]: pip install pandas sqlalchemy pymysql
```

```
In [1]: import pandas as pd
from sqlalchemy import create_engine
from sqlalchemy.exc import SQLAlchemyError
```

```
In [7]: from sqlalchemy import create_engine
# Correctly formatted database connection details
DATABASE_TYPE = 'mysql'
DBAPI = 'mysqlconnector' # Assuming you're using mysqlconnector
HOST = 'fashionretail.c3q4cw8i0vbc.us-east-2.rds.amazonaws.com'
USER = 'admin'
PASSWORD = 'bsan-6080-project'
DATABASE = 'fashion_retail_sales'

# Correctly create the DATABASE_URL
DATABASE_URL = f'{DATABASE_TYPE}+{DBAPI}://{USER}:{PASSWORD}@{HOST}/{DATABASE}'
```

```
# Create the SQLAlchemy engine
engine = create_engine(DATABASE_URL, echo=False)
```

```
# Test the connection
```

```
try:
    with engine.connect() as conn:
        print("Successfully connected to the database!")
except Exception as e:
    print(f"An error occurred: {e}")
```

```
Successfully connected to the database!
```

```
create_customers_table_sql = """
CREATE TABLE IF NOT EXISTS Customers (
    CustomerID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerReferenceID INT UNIQUE NOT NULL
);
"""

create_purchases_table_sql = """
CREATE TABLE IF NOT EXISTS Purchases (
    PurchaseID INT AUTO_INCREMENT PRIMARY KEY,
    CustomerReferenceID INT NOT NULL,
    ItemPurchased VARCHAR(255),
    PurchaseAmountUSD DECIMAL(10, 2),
    DatePurchase DATE,
    ReviewRating DECIMAL(3, 1),
    PaymentMethod VARCHAR(50),
    FOREIGN KEY (CustomerReferenceID) REFERENCES Customers(CustomerReferenceID)
);
"""

with engine.begin() as conn:
    conn.execute(create_customers_table_sql)
    conn.execute(create_purchases_table_sql)

csv_file_path = 'Fashion_Retail_Sales_Clean.csv' # Update this path
df = pd.read_csv(csv_file_path)

customers_df = df[['Customer Reference ID']].drop_duplicates().rename(columns={'Customer Reference ID': 'CustomerReferenceID'})
customers_df.to_sql('Customers', con=engine, if_exists='append', index=False, method='multi')
: -1

purchases_df = df.drop(columns=['Unnamed: 0']).rename(columns={
    'Customer Reference ID': 'CustomerReferenceID',
    'Item Purchased': 'ItemPurchased',
    'Purchase Amount (USD)': 'PurchaseAmountUSD',
    'Date Purchase': 'DatePurchase',
    'Review Rating': 'ReviewRating',
    'Payment Method': 'PaymentMethod'
})
purchases_df.to_sql('Purchases', con=engine, if_exists='append', index=False, method='multi')

print("Data insertion complete.")
```

Data insertion complete.

ETL

Result Grid

Filter Rows: Search

Edit:

CustomerID CustomerReference...

43	3957
97	3958
61	3959
128	3960
84	3961
164	3962
122	3963
72	3964
103	3965
104	3966
136	3967
158	3968

Customers 6

Action Output

	Time	Action	Response
✓	7 18:19:45	SELECT * FROM Customers LI...	166 row(s) returned

100% 10:4

Result Grid

Filter Rows: Search

Edit:

Export/Import:

Fetch rows:

Purchases 5

Result Grid

Form Editor

Field Types

Purchases 5

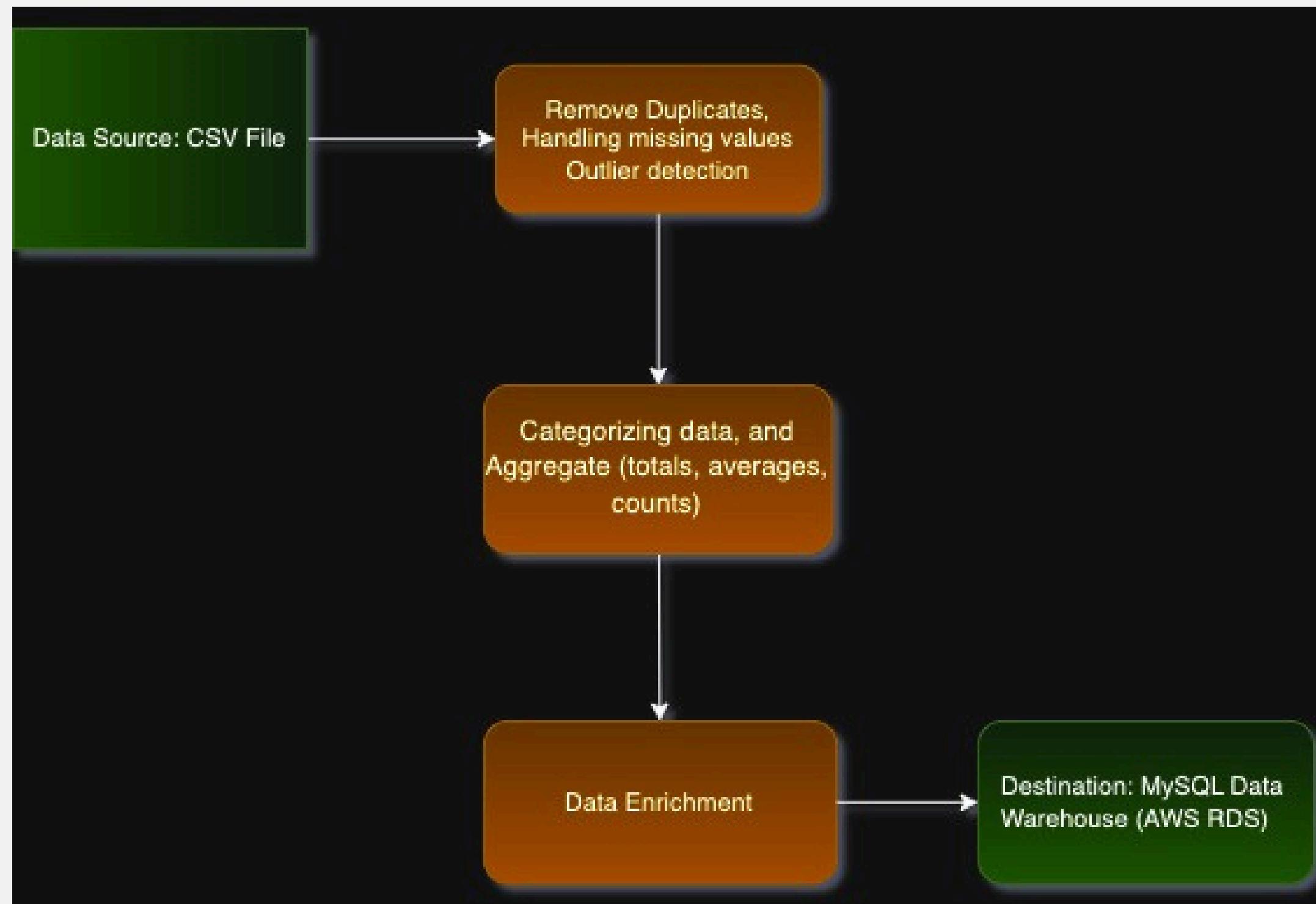
Action Output

PurchaseID	CustomerReference...	ItemPurchased	PurchaseAmountUSD	DatePurchase	ReviewRating	PaymentMethod
1	4018	Handbag	4619.00	2023-02-05	3.2	Credit Card
2	4115	Tunic	2456.00	2023-07-11	2.0	Credit Card
3	4019	Tank Top	2102.00	2023-03-23	4.1	Cash
4	4097	Leggings	3126.00	2023-03-15	3.2	Cash
5	3997	Wallet	3003.00	2022-11-27	4.7	Cash
6	4080	Onesie	2914.00	2022-12-11	4.5	Credit Card
7	4055	Jacket	2571.00	2023-07-08	1.3	Cash
8	3973	Trousers	2419.00	2022-11-10	4.6	Cash
9	4044	Jeans	4771.00	2023-05-19	4.1	Cash
10	4010	Loafers	4233.00	2023-06-11	2.7	Credit Card
11	4108	Slippers	2356.00	2023-03-19	4.8	Credit Card
12	4067	Bowtie	4418.00	2022-11-21	3.4	Cash

Time Action Response Duration / Fetch Time

6 18:17:34 SELECT * FROM Purchases LI... 1000 row(s) returned 0.170 sec / 0.070 sec

ETL DIAGRAM



RECOMMENDATION