VENDOR PERFORMANCE DATA ANALYTICS

Retail Operations & Vendor Performance Analytics

SOMNATH MONDAL

Business Problem

In the retail and wholesale industry, managing inventory, pricing, and vendor relationships plays a crucial role in maintaining profitability. Businesses often struggle with slow-moving stock, inconsistent vendor performance, and pricing strategies that reduce margins. Without proper data analysis, companies may overstock low-performing products, depend too heavily on certain vendors, or miss opportunities to optimize purchasing decisions.

This project focuses on analysing sales, inventory, and vendor performance data to identify areas where profitability can be improved and operational efficiency can be strengthened.

The key objectives of this analysis are to:

- Identify underperforming brands that may require pricing revisions, promotional strategies, or inventory adjustments.
- Determine top-performing vendors contributing significantly to total sales and gross profit.
- Evaluate how bulk purchasing affects unit cost and overall margin improvement.
- Compare profitability between high-performing and low-performing vendors to understand margin variability and supplier dependency risk.

The ultimate goal is to provide data-driven insights that help improve pricing strategy, vendor selection, purchasing efficiency, and overall business profitability.

Operations

Task 1: Creating a database to store using python specifically pandas, os and sqlalchemy library

```
[1]: import pandas as pd
     import os
     from sqlalchemy import create_engine

[2]: engine = create_engine('sqlite:///inventory.db')

[3]: data_path = '../Data/data'
     for file in os.listdir(data_path):
         print(file)

.ipynb_checkpoints
begin_inventory.csv
end_inventory.csv
purchases.csv
purchase_prices.csv
sales.csv
vendor_invoice.csv
```

```
[6]: for file in os.listdir(data_path):
         if '.csv' in file:
             file_path = os.path.join(data_path, file)
             df = pd.read_csv(file_path)
             print(df.shape)
             ingest_db(df, file[:-4],engine)

(206529, 9)
(224489, 9)
(2372474, 16)
(12261, 9)
(12825363, 14)
(5543, 10)

[4]: def ingest_db(df, table_name, engine):
         df.to_sql(table_name, con = engine, if_exists = 'replace', index = 'False')     # append for increamental load
```

Name	Modified	File Size
DB_Creation_Basic.ipynb	12 minutes ago	2.5 KB
inventory.db	49 seconds ago	2.3 GB

In a professional setting, data pipelines are implemented as automated Python scripts rather than notebooks. These scripts include structured logging to monitor execution and record operational details, along with proper exception handling to ensure secure, stable, and fault-tolerant data loading.

This is the industry level script for automation : Link

Log File Sample:

jupyter ingestion_db.log Last Checkpoint: 20 minutes ago

File Edit View Settings Help

```
17-02-2026 01:33:25 | INFO | root | begin_inventory.csv loaded successfully with shape (206529, 9)
17-02-2026 01:33:30 | INFO | root | begin_inventory.csv ingested into database successfully
17-02-2026 01:33:30 | INFO | root | end_inventory.csv loaded successfully with shape (224489, 9)
17-02-2026 01:33:34 | INFO | root | end_inventory.csv ingested into database successfully
17-02-2026 01:33:42 | INFO | root | purchases.csv loaded successfully with shape (2372474, 16)
17-02-2026 01:34:56 | INFO | root | purchases.csv ingested into database successfully
17-02-2026 01:34:56 | INFO | root | purchase_prices.csv loaded successfully with shape (12261, 9)
17-02-2026 01:34:56 | INFO | root | purchase_prices.csv ingested into database successfully
17-02-2026 01:35:30 | INFO | root | sales.csv loaded successfully with shape (12825363, 14)
17-02-2026 01:52:37 | INFO | root | sales.csv ingested into database successfully
17-02-2026 01:52:39 | INFO | root | vendor_invoice.csv loaded successfully with shape (5543, 10)
17-02-2026 01:52:40 | INFO | root | vendor_invoice.csv ingested into database successfully
17-02-2026 01:52:40 | INFO | root | Ingestion Completed Successfully
17-02-2026 01:52:40 | INFO | root | Total Time Taken: 19.25 minutes
```

Task 2: Perform EDA or Exploratory Data Analysis on the database using python notebook with the help of libraries like pandas and sqlite. This process is to determine if there is need of creating aggregated tables from the present tables in database, which can help to understand the top performing vendors and if some product needs to go through pricing optimization.

```
[1]: import pandas as pd
     import sqlite3
```

```
[2]: conn = sqlite3.connect('../Database Creation/inventory.db')
```

```
[3]: tables =  pd.read_sql_query("SELECT name FROM sqlite_master WHERE type = 'table'",conn)
```

```
[4]: tables
```

[4]:

	name
0	begin_inventory
1	end_inventory
2	purchases
3	purchase_prices
4	sales
5	vendor_invoice

```
[9]: for table in tables['name']:
         print('-'*150)
         print(f'Table Name : {table}, ','Count of records: ',pd.read_sql(f"select count(*) as count from {table}",conn)['count'].values[0])
         display(pd.read_sql(f"select * from {table} limit 2",conn))
```

```
----------------------------------------------------------------------------------------------------------------------------------------------------------------
Table Name : begin_inventory,  Count of records:  206529
```

	False	InventoryId	Store	City	Brand	Description	Size	onHand	Price	startDate
0	0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	8	12.99	2024-01-01
1	1	1_HARDERSFIELD_60	1	HARDERSFIELD	60	Canadian Club 1858 VAP	750mL	7	10.99	2024-01-01

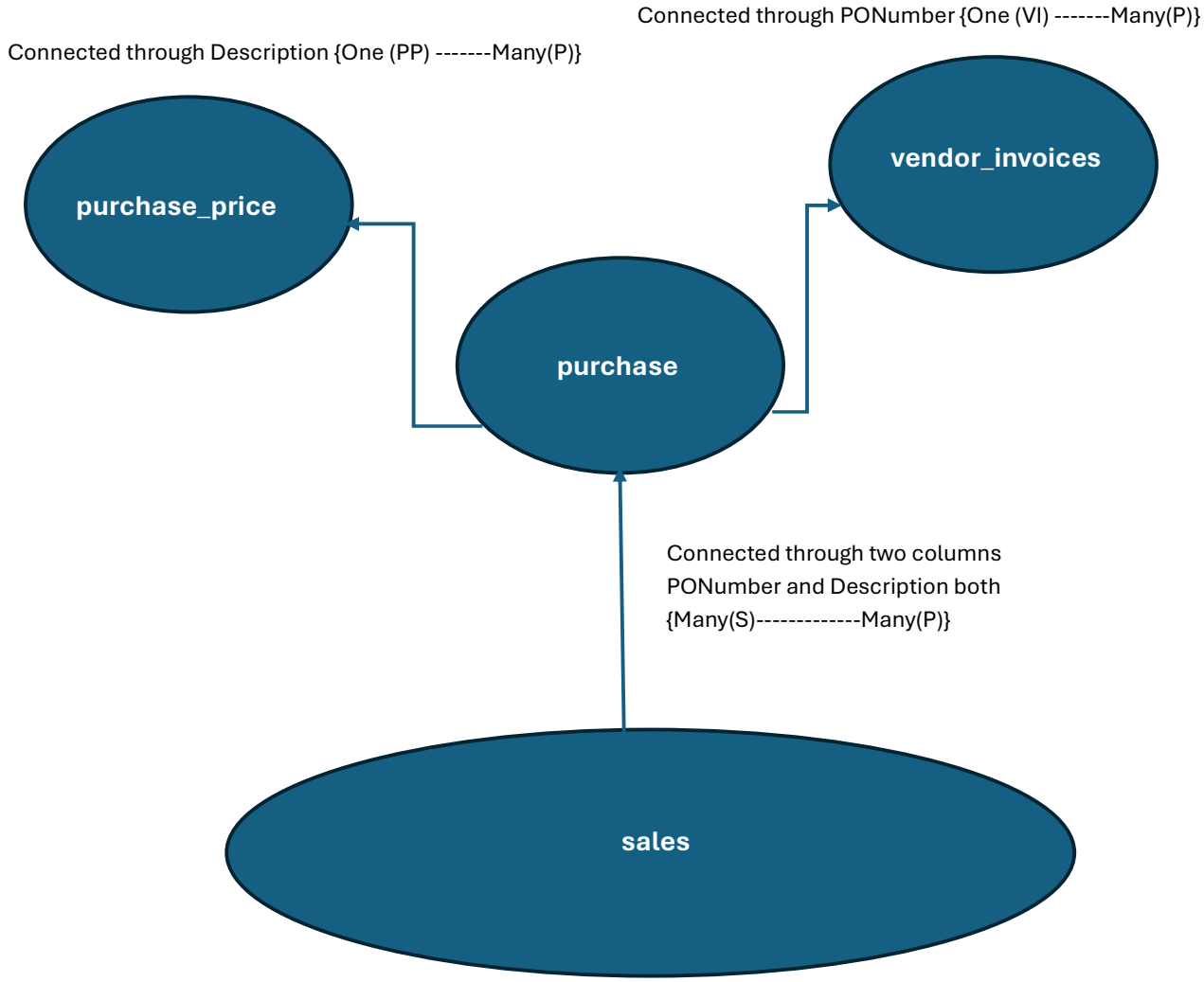
```
----------------------------------------------------------------------------------------------------------------------------------------------------------------
Table Name : end_inventory,  Count of records:  224489
```

	False	InventoryId	Store	City	Brand	Description	Size	onHand	Price	endDate
0	0	1_HARDERSFIELD_58	1	HARDERSFIELD	58	Gekkeikan Black & Gold Sake	750mL	11	12.99	2024-12-31
1	1	1_HARDERSFIELD_62	1	HARDERSFIELD	62	Herradura Silver Tequila	750mL	7	36.99	2024-12-31

```
----------------------------------------------------------------------------------------------------------------------------------------------------------------
Table Name : purchases,  Count of records:  2372474
```

	False	InventoryId	Store	Brand	Description	Size	VendorNumber	VendorName	PONumber	PODate	ReceivingDate	InvoiceDate	PayDate	Purchas
0	0	69_MOUNTMEND_8412	69	8412	Tequila Ocho Plata Fresno	750mL	105	ALTAMAR BRANDS LLC	8124	2023-12-21	2024-01-02	2024-01-04	2024-02-16	
1	1	30_CULCHETH_5255	30	5255	TGI Fridays Ultimte Mudslide	1.75L	4466	AMERICAN VINTAGE BEVERAGE	8137	2023-12-22	2024-01-01	2024-01-07	2024-02-21	

After analysing each table structure, it is evident that two off the table related to inventory is not related to vendor performance analysis. So, the main focus is to other 4 tables which are interconnected, in this manner:



Observation About the flow of Retail and Tables

The flow of any retail company goes like this, we buy product from vendors, then we determine the true cost of those products, by summing up purchase prices and freight. Fright is some extra cost that has been included which are indirectly related to the sale. Then we sell that to the customer.

Now, these are the table structure:

Purchases – What We Buy from Vendors, this table contains

- Which vendor we bought from
- Which brand/product we purchased
- Purchase date
- Quantity purchased
- Amount paid

Purchase Prices – Product Cost Details, this table contains

- Actual cost of each product
- Purchase price agreed with the vendor
- Unique combination of vendor + brand

Vendor Invoice – Total Cost Including Freight, this table contains

- Summarizes purchase transactions by vendor and PO
- Includes total quantity and total amount
- Adds freight cost

Sales – What We Sell to Customers, this table contains

- Brand sold
- Quantity sold
- Selling price
- Revenue earned

Now for vendor and brand analysis, a consolidated table needs to be created which has quantity purchased, purchase amount, actual product cost, fright cost, quantity sold, revenue, profitability matrices.

Steps to create the above-mentioned consolidated table

CREATING TABLE FREIGHT SUMMARY

[28]:
```
freight_summary = pd.read_sql(""" SELECT VendorNumber, SUM(Freight) AS Total_Freight
FROM vendor_invoice
GROUP BY VendorNumber""",conn)
```

[29]:
```
freight_summary
```

[29]:

	VendorNumber	Total_Freight
0	2	27.08
1	54	0.48
2	60	367.52
3	105	62.39
4	200	6.19
...	...	...
121	98450	856.02
122	99166	130.09
123	172662	178.34
124	173357	202.50
125	201359	0.09

126 rows × 2 columns

CREATE VENDOR PURCHASE SUMMARY TABLE

[53]:
```
vendor_purchase_summary = pd.read_sql("""
SELECT
    p.VendorNumber,
    p.VendorName,
    p.Brand,
    pp.Volume,
    p.Description,
    SUM(p.Quantity) AS Quantity,
    AVG(p.PurchasePrice) AS PurchasePrice,
    SUM(p.Dollars) AS TotalPurchasePrice
FROM purchases p
JOIN purchase_prices pp
    ON p.Brand = pp.Brand
GROUP BY
    p.VendorNumber,
    p.VendorName,
    p.Brand,
    pp.Volume,
    p.Description
""", conn)
```

[54]:
```
vendor_purchase_summary
```

[54]:

	VendorNumber	VendorName	Brand	Volume	Description	Quantity	PurchasePrice	TotalPurchasePrice
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	750	Ch Lilian 09 Ladouys St Este	8	23.86	190.88
1	2	IRA GOLDMAN AND WILLIAMS, LLP	90609	162.5	Flavor Essence Variety 5 Pak	320	17.00	5440.00
2	54	AAPER ALCOHOL & CHEMICAL CO	990	3750	Ethyl Alcohol 200 Proof	1	105.07	105.07
3	60	ADAMBA IMPORTS INTL INC	771	750	Bak's Krupnik Honey Liqueur	39	11.44	446.16
4	60	ADAMBA IMPORTS INTL INC	3401	1750	Vesica Vodka	6	11.10	66.60
...	...	...	...	...	...	...	...	...
10688	173357	TAMWORTH DISTILLING	2804	750	Camp Robber Whiskey	210	32.14	6749.40
10689	173357	TAMWORTH DISTILLING	3666	375	Art in the Age Chicory Root	520	18.79	9770.80
10690	173357	TAMWORTH DISTILLING	3848	750	Chicory Root Vodka	28	23.30	652.40
10691	173357	TAMWORTH DISTILLING	3909	750	White Mountain Vodka	1232	19.37	23863.84
10692	201359	FLAVOR ESSENCE INC	90609	162.5	Flavor Essence Variety 5 Pak	1	17.00	17.00

10693 rows × 8 columns

We use an average (preferably weighted average) of PurchasePrice because PurchasePrice represents a unit cost. Since purchases occur at different quantities and possibly different dates/prices, summing unit prices would produce incorrect values. Aggregating with an average preserves a meaningful representative unit price when summarizing multiple transactions.

[58]:
```
sales_summary = pd.read_sql(""" SELECT VendorNo,
Brand,
Description,
Volume,
SUM(SalesQuantity) AS TotalQuantity,
AVG(SalesPrice) AS SalesPrice,
SUM(SalesDollars) AS TotalRevenue
FROM sales
GROUP BY VendorNo, Brand, Description, Volume""",conn)
```

[59]:
```
sales_summary
```

[59]:

	VendorNo	Brand	Description	Volume	TotalQuantity	SalesPrice	TotalRevenue
0	2	90085	Ch Lilian 09 Ladouys St Este	750.0	18	36.990000	665.82
1	2	90609	Flavor Essence Variety 5 Pak	162.5	24	24.990000	599.76
2	60	771	Bak's Krupnik Honey Liqueur	750.0	47	14.990000	704.53
3	60	3979	Vesica Potato Vodka	1750.0	3931	17.020216	66871.69
4	105	2529	Right Gin	750.0	12	29.990000	359.88
...	...	...	...	...	...	...	...
11267	173357	2804	Camp Robber Whiskey	750.0	140	44.990000	6298.60
11268	173357	3666	Art in the Age Chicory Root	375.0	360	24.990000	8996.40
11269	173357	3848	Chicory Root Vodka	750.0	6	30.990000	185.94

CREATE SALES SUMMARY TABLE

[62]:
```
sales_summary = pd.read_sql(""" SELECT VendorNo,
Brand,
Description,
Volume,
SUM(SalesQuantity) AS TotalQuantity,
AVG(SalesPrice) AS SalesPrice,
SUM(SalesDollars) AS TotalRevenue
FROM sales
GROUP BY VendorNo, Brand, Description, Volume""",conn)
```

[63]:
```
sales_summary
```

[63]:

	VendorNo	Brand	Description	Volume	TotalQuantity	SalesPrice	TotalRevenue
0	2	90085	Ch Lilian 09 Ladouys St Este	750.0	18	36.990000	665.82
1	2	90609	Flavor Essence Variety 5 Pak	162.5	24	24.990000	599.76
2	60	771	Bak's Krupnik Honey Liqueur	750.0	47	14.990000	704.53
3	60	3979	Vesica Potato Vodka	1750.0	3931	17.020216	66871.69
4	105	2529	Right Gin	750.0	12	29.990000	359.88
...	...	...	...	...	...	...	...
11267	173357	2804	Camp Robber Whiskey	750.0	140	44.990000	6298.60
11268	173357	3666	Art in the Age Chicory Root	375.0	360	24.990000	8996.40

For the next step to achieve the final table for visualization I merged two summary datasets of sales and vendor_purchase. After that added doing some cleaning and transformation added freight cost in the main picture to get accurate profit margin.

[67]:
```
vendor_purchase_summary['Volume'] = pd.to_numeric(
    vendor_purchase_summary['Volume'], errors='coerce'
)

sales_summary['Volume'] = pd.to_numeric(
    sales_summary['Volume'], errors='coerce'
)
```

[78]:
```
vendor_sales_comparison = pd.merge(
    vendor_purchase_summary,
    sales_summary,
    left_on=['VendorNumber','Brand','Description','Volume'],
    right_on=['VendorNo','Brand','Description','Volume'],
    how='left'
)
```

[79]:
```
vendor_sales_comparison
```

[79]:

	VendorNumber	VendorName	Brand	Volume	Description	Quantity	PurchasePrice	TotalPurchasePrice	VendorNo	TotalQuantity	SalesPrice	TotalRevenue
		IRA GOLDMAN			Ch Lilian 09							

[89]:
```
vendor_sales_comparison = vendor_sales_comparison.merge(
    freight_summary,
    on='VendorNumber',
    how='left'
)
```

[90]:
```
vendor_sales_comparison['Total_Freight'] = (
    vendor_sales_comparison['Total_Freight'].fillna(0)
)
```

[91]:
```
vendor_sales_comparison['NetRevenue'] = (
    vendor_sales_comparison['TotalRevenue']
    - vendor_sales_comparison['TotalPurchasePrice']
    - vendor_sales_comparison['Total_Freight']
)
vendor_sales_comparison['Freight_Percent'] = (
    vendor_sales_comparison['Total_Freight']
    / vendor_sales_comparison['TotalPurchasePrice']
)
```

[92]:
```
vendor_sales_comparison
```

[92]:

Brand	Volume	Description	PurchaseQuantity	PurchasePrice	TotalPurchasePrice	SalesQuantity	SalesPrice	TotalRevenue	Total_Freight	NetRevenue	Freight_Percen

Also, some of the profit margin deciding table was giving na, means no sell is there for that particular product.

```
[94]: cols = ['SalesQuantity','SalesPrice','TotalRevenue']

      vendor_sales_comparison[cols] = (
          vendor_sales_comparison[cols].fillna(0)
      )
```

```
[95]: vendor_sales_comparison
```

[95]:

	VendorNumber	VendorName	Brand	Volume	Description	PurchaseQuantity	PurchasePrice	TotalPurchasePrice	SalesQuantity	SalesPrice	TotalRevenue	Tota
0	2	IRA GOLDMAN AND WILLIAMS, LLP	90085	750.0	Ch Lilian 09 Ladouys St Este	8	23.86	190.88	18.0	36.99	665.82	
1	2	IRA GOLDMAN AND WILLIAMS, LLP	90609	162.5	Flavor Essence Variety 5 Pak	320	17.00	5440.00	24.0	24.99	599.76	
		AAPER ALCOHOL &			Ethyl							

Now to load the dataframe into a sql table I made a table with same columns and added the data to the table from dataframe.

```
          dtype='object')
```

```
[97]: cursor = conn.cursor()
```

```
[98]: cursor.execute("""CREATE TABLE vendor_performance (
          VendorNumber INT,
          VendorName VARCHAR(255),
          Brand INT,
          Volume FLOAT,
          Description VARCHAR(255),

          PurchaseQuantity FLOAT,
          PurchasePrice DECIMAL(12,2),
          TotalPurchasePrice DECIMAL(14,2),

          SalesQuantity FLOAT,
          SalesPrice DECIMAL(12,2),
          TotalRevenue DECIMAL(14,2),

          Total_Freight DECIMAL(14,2),
          NetRevenue DECIMAL(14,2),
          Freight_Percent DECIMAL(10,4)
      )"""
                  )
```

```
[98]: <sqlite3.Cursor at 0x26f48dcb140>
```

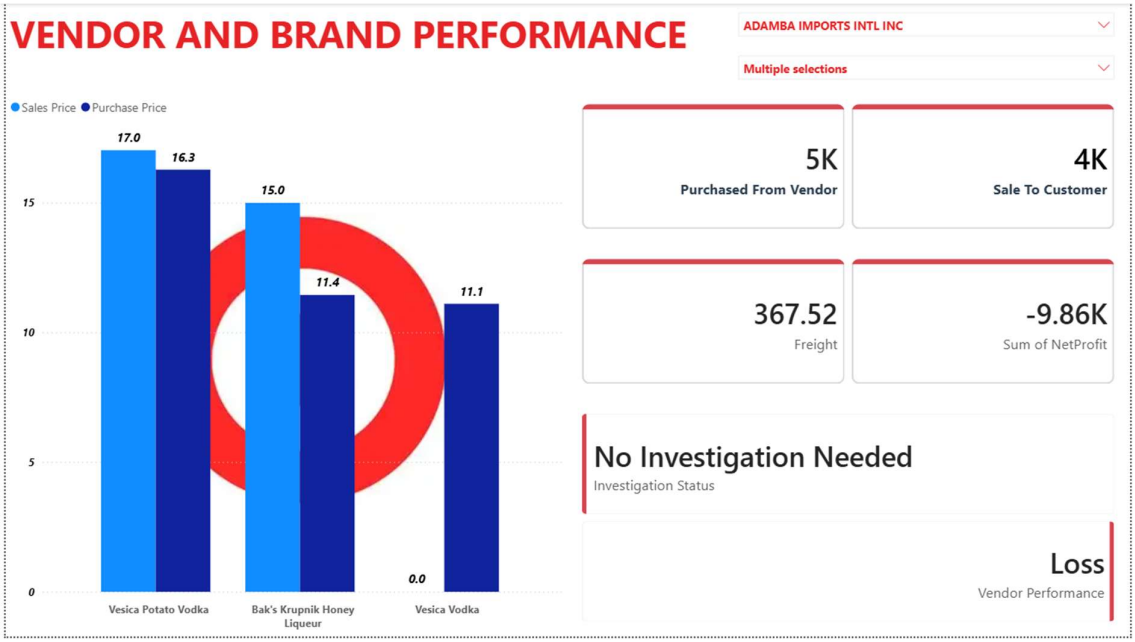
```
[99]: vendor_sales_comparison.to_sql('vendor_sales_comparison',conn,if_exists='replace',index=False)
```

```
[99]: 10693
```

```
[100]: pd.read_sql_query(" SELECT * FROM vendor_sales_comparison",conn)
```

Task 3: Power BI Report

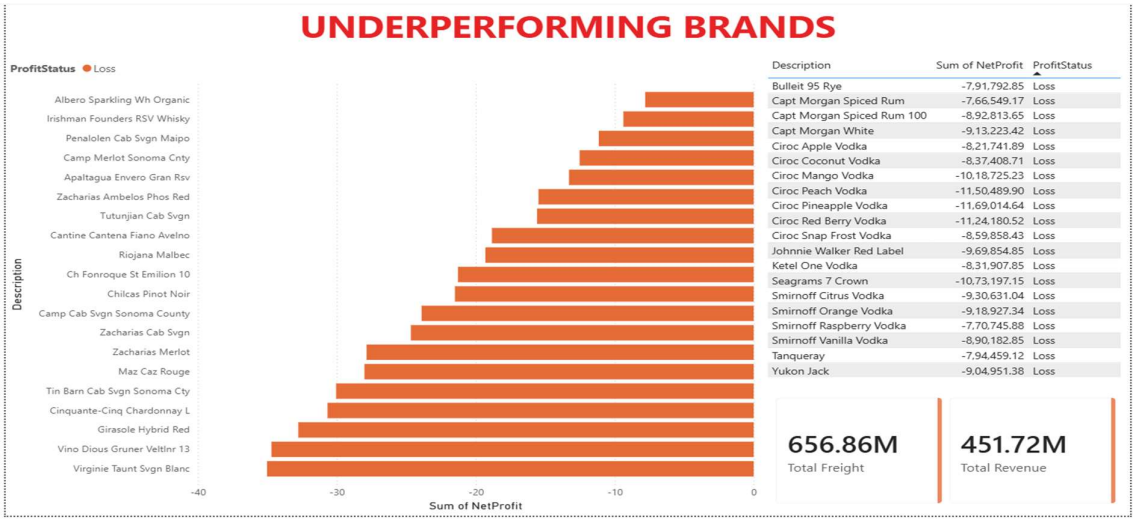
This is the **first page of the Power BI report**, designed as a high-level **Vendor and Brand Performance** overview that gives users an immediate summary of financial and operational results. A vendor is dynamically selected through a slicer, allowing the visuals and KPIs to update based on the chosen supplier. The page combines a comparative bar chart showing sales price versus purchase price by product with key performance indicators such as purchased quantity, sales to customers, freight cost, and overall net profit. Investigation status and vendor performance labels provide quick insight into whether action is required, making this landing page a concise executive snapshot for evaluating vendor profitability and pricing performance at a glance.



Now it is time to answer business questions mentioned in the beginning,

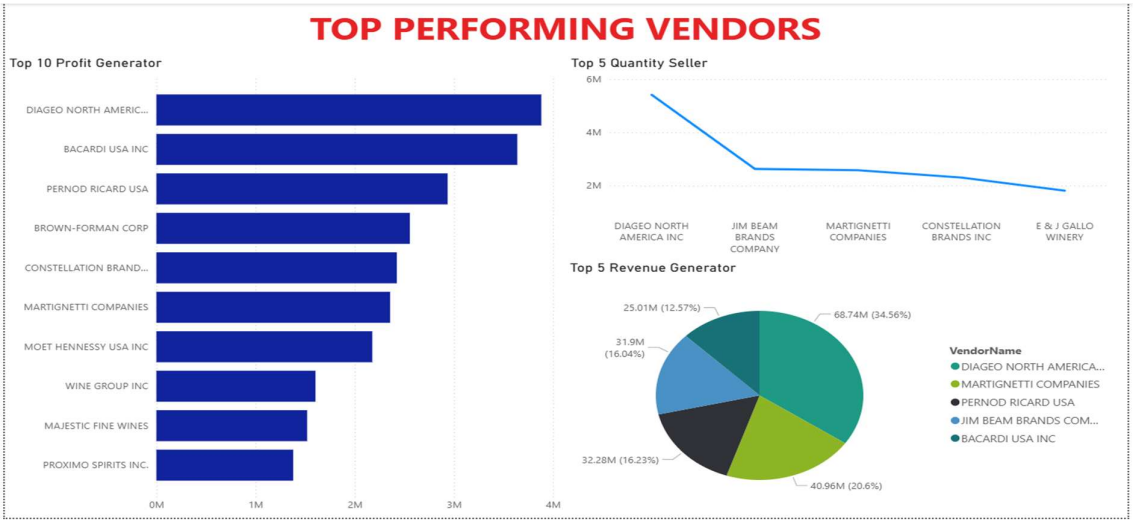
Identify underperforming brands that may require pricing revisions, promotional strategies, or inventory adjustments.

The Underperforming Brands page highlights products generating negative net profit, helping identify brands that may need pricing adjustments, targeted promotions, or inventory optimization. A horizontal bar chart ranks brands by total net loss, clearly showing the most financially underperforming items, while a detailed table on the right lists each product's description, net profit value, and loss status for deeper analysis. Supporting KPIs such as total freight and total revenue provide additional context to evaluate cost pressures versus sales performance. This page enables stakeholders to quickly pinpoint loss-making brands and prioritize strategic actions to improve margins and overall profitability.



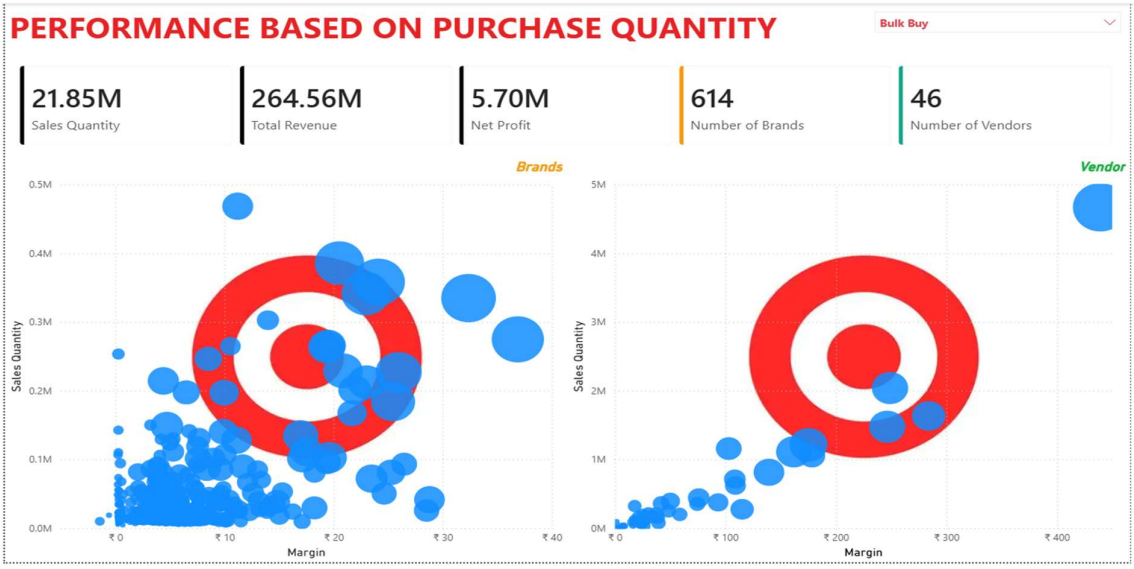
Determine top-performing vendors contributing significantly to total sales and gross profit.

The **Top Performing Vendors** dashboard identifies vendors that contribute the most to overall sales, quantity movement, and gross profit performance. A ranked bar chart highlights the **top 10 profit-generating vendors**, showing that companies such as Diageo North America Inc, Bacardi USA Inc, and Pernod Ricard USA lead in profitability, while supporting visuals provide additional performance insights through the **Top 5 Quantity Seller** trend and a **Top 5 Revenue Generator** distribution. Together, these visuals allow stakeholders to quickly recognize high-value vendor partnerships, understand revenue contribution patterns, and focus strategic efforts on vendors that drive the strongest financial impact on the business.



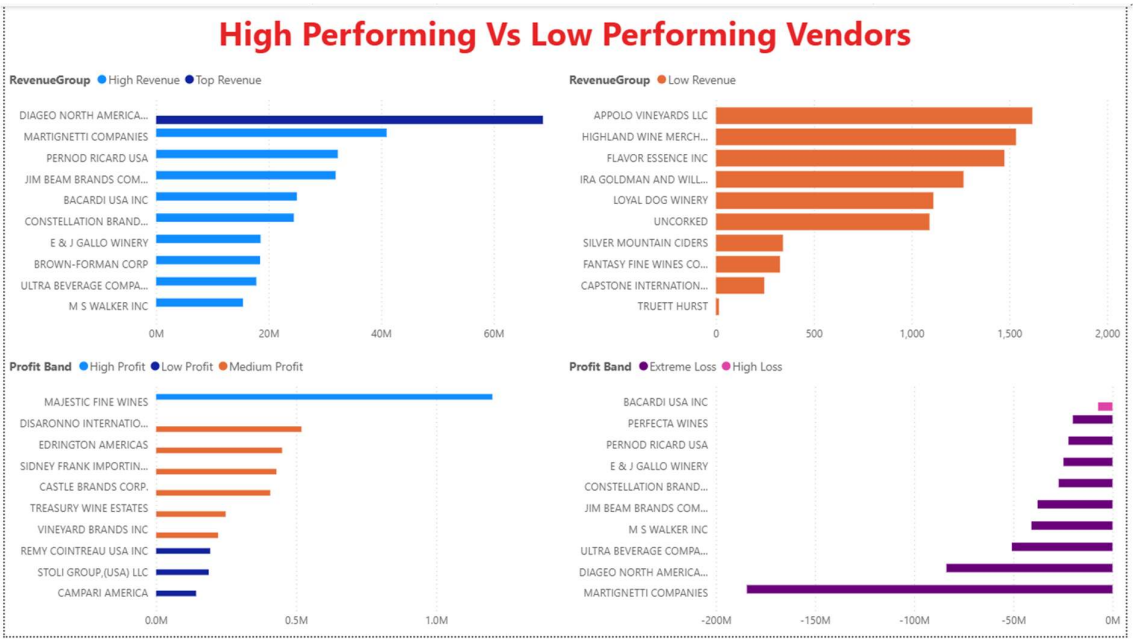
Evaluate how bulk purchasing affects unit cost and overall margin improvement.

The **Performance Based on Purchase Quantity** page evaluates how bulk purchasing influences unit cost efficiency and margin improvement by allowing users to select different purchase quantity levels through a slicer. Key KPIs at the top summarize overall sales quantity, total revenue, net profit, and the number of brands and vendors involved, providing a high-level performance snapshot. The brand-level and vendor-level scatter plots visualize the relationship between **margin and sales quantity**, showing that higher purchase quantities generally align with improved margins and stronger sales performance, indicating potential cost advantages from bulk buying. This page enables decision-makers to analyze whether increasing order volumes leads to better profitability and to identify brands or vendors that benefit most from bulk purchasing strategies.



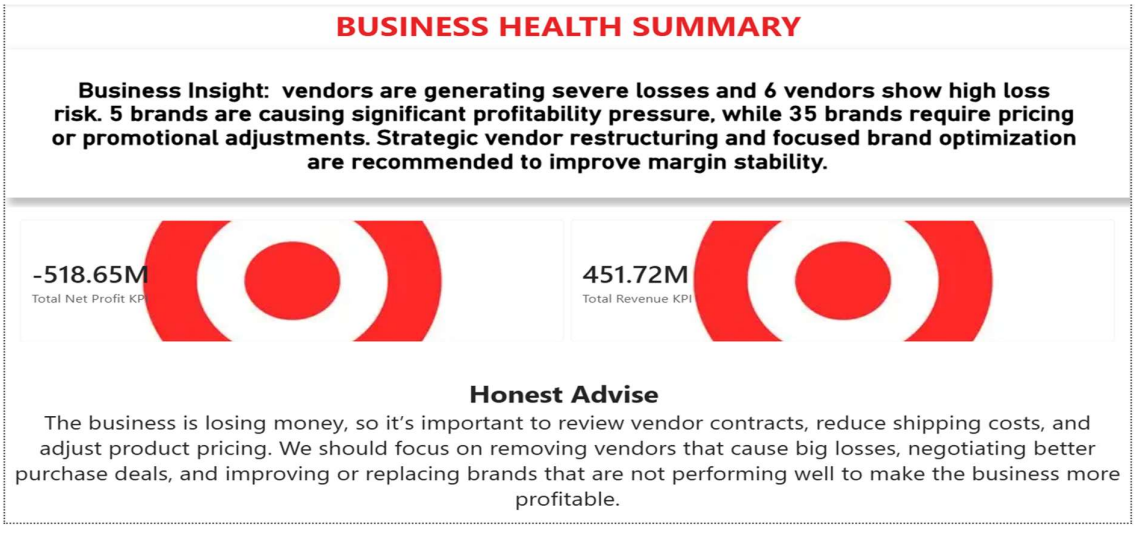
Compare profitability between high-performing and low-performing vendors to understand margin variability and supplier dependency risk.

The High Performing vs Low Performing Vendors dashboard compares vendor profitability to highlight margin variability and potential supplier dependency risks. High-revenue and top-revenue vendors are visualized on the left, showing which suppliers contribute the most to overall sales performance, while a separate chart identifies low-revenue vendors that may require strategic review. Additional profit band analysis categorizes vendors into high, medium, and low profit segments, helping stakeholders understand profitability distribution, and the extreme/high loss view emphasizes vendors generating significant negative margins. Together, these visuals provide a balanced comparison of strong versus weak vendor performance, enabling decision-makers to assess reliance on key suppliers, identify margin risks, and prioritize optimization strategies.



Lastly, I created a Business Health Summary:

The **Business Health Summary** page provides an executive-level overview of the company's overall financial condition by consolidating key insights from the report into clear, actionable conclusions. It highlights critical performance concerns, indicating that several vendors are generating significant losses and multiple brands are creating profitability pressure, resulting in a negative total net profit despite strong revenue performance. The dashboard combines high-level KPIs with strategic business insights and advisory recommendations, emphasizing the need for vendor restructuring, pricing adjustments, cost optimization, and focused brand management to stabilize margins and improve long-term profitability. This page serves as a final decision-support summary, helping stakeholders quickly understand risks, performance gaps, and recommended actions for improving business health.



Vendor Performance Data Analytics — Project Summary

The **Vendor Performance Data Analytics** project focuses on improving profitability and operational efficiency in the retail and wholesale sector by analysing vendor performance, pricing strategy, purchasing behaviour, and sales outcomes. The core business challenge addressed is the difficulty companies face in managing slow-moving inventory, inconsistent vendor performance, and margin-reducing pricing decisions without structured data insights.

Objective

The main goal of the project is to deliver data-driven decision support for retail operations by identifying underperforming brands, recognizing high-value vendors, analysing the impact of bulk purchasing on margins, and evaluating profitability variability across suppliers. These insights aim to support smarter pricing, vendor selection, and purchasing strategies.

Data Engineering & Preparation

The project begins with building a structured data pipeline using Python, pandas, and SQLAlchemy to automate database creation and ensure reliable data loading with logging and error handling (shown in the workflow on page 3). Exploratory Data Analysis (EDA) is then performed to understand relationships across key retail tables — purchases, purchase prices, vendor invoices, and sales — while excluding irrelevant inventory tables.

A consolidated analytical dataset is created by merging these tables, calculating weighted average purchase prices, integrating freight costs, and generating profitability metrics such as revenue, cost, and net margin. This structured dataset forms the foundation for advanced business insights and visualization.

Analytics & Visualization

Using Power BI, the project delivers an interactive dashboard suite:

- **Vendor & Brand Performance Overview:** Executive snapshot showing KPIs like purchased quantity, freight cost, and net profit with dynamic vendor filtering (page 10).
- **Underperforming Brands Analysis:** Identifies products generating losses to guide pricing adjustments and inventory optimization.
- **Top Performing Vendors:** Highlights vendors driving the highest revenue and gross profit, helping prioritize strategic partnerships (page 11).

- **Bulk Purchasing Impact:** Scatter plot analysis demonstrates how higher purchase quantities often correlate with improved margins and sales performance (page 12).
- **High vs Low Performing Vendors:** Compares supplier profitability and identifies dependency risks and margin variability.

Key Insights & Business Value

The analysis reveals that despite strong revenue, several vendors and brands contribute to negative net profit due to freight costs, pricing inefficiencies, or low sales performance. Bulk purchasing can reduce unit cost and improve margins when applied strategically. The final **Business Health Summary** dashboard consolidates findings into actionable recommendations such as vendor restructuring, pricing optimization, and cost control initiatives (page 13).

Conclusion

Overall, this project demonstrates an end-to-end retail analytics workflow — from automated data engineering and EDA to advanced BI visualization — delivering actionable insights that enable businesses to improve vendor management, optimize pricing strategies, reduce margin risks, and enhance long-term profitability.