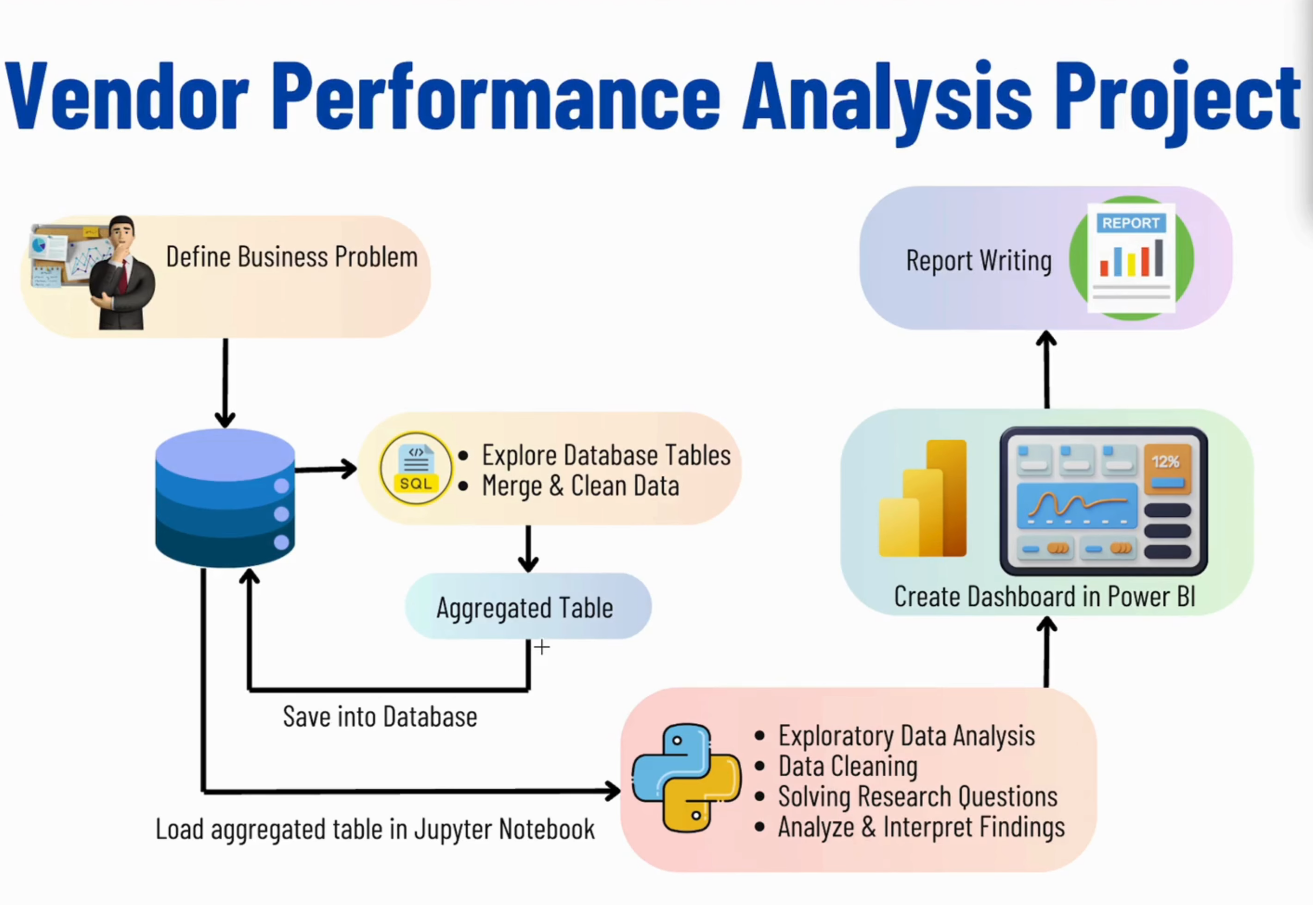
***Vendor Performance and Prediction Analysis Project***

Here we take up a business problem and try to analyse it and give insights along with some solutions.



The dataset is stored in the sql database where we’ll try go understand and explore the dataset using sql queries. We’ll thus create a dataset which can be used for further analysis.

We also merge tables and create an aggregated table that is then stored back into the database so that it can be accessed later.

Then we perform EDA with python in the aggregated table and the cleaned data that was stored in the database after exploring it with sql.

We further perform some data cleaning and solve certain business research questions.

Finally we analyze the results and interpret them.

We then create an Interactive Dashboard on the final data in PoweBI to show the results and findings from the dataset.

Lastly we write a report to document the entire project (can also be done in the github readme) which explains the problem statement and what we have done about it.

**BUSINESS PROBLEM:**

Effective inventory and sales management are critical for optimizing profitability in the retail and wholesale industry. Companies need to ensure that they are not incurring losses due to inefficient pricing, poor inventory turnover or vendor dependency. The goal of this analysis is to :

* Identify underperforming brands that require promotional or pricing adjustments.
* Determine top vendors contributing to sales and gross profit.
* Analyze the impact of bulk purchasing on unit cost.
* Access inventory turnover to reduce holding costs and improve efficiency.
* Investigate the profitability variance between high performing and low performing vendors.

**STEP 1: Database setup and ingestion**

We start by setting up the database. We created a folder named **‘data’** and added all the csv files to the folder. This folder is then placed inside the project folder on desktop.

We begin to setup the dataset inside the database. For that –

We create a python notebook named **‘db\_ingestion.ipynb’** which contains the python code for reading and ingesting the CSVs’ data into the postgresql database. (The python code also makes use of **logs** and 2 **functions**)

The 2 functions used are- **ingest\_db** – will ingest the dataframe into database table and **load\_raw\_data** - load the CSV as dataframe and ingest into db.

**Problem encountered here** – the datasets that we are using are very large, one of them(sales) even has about 1 crore rows. Getting such large files to ingest into the database was taking forever.

**Solution –** With the help of chatgpt we ingested chunks of data having 100000 rows each instead of ingesting the entire data directly but only for the large csv files – sales, purchases and inventory.

**STEP 2: EDA in SQL**

After establishing connection with out postgresql database, we wrote certain sql queries to view data from each of the tables and understand it. These were the observations-

* The purchases table contains actual purchase data, including the date of purchase, products(brands) purchased by the vendors, the amount paid(in dollars), and the quanityt purchased.
* The PurchasePrice column in derived from the purchase\_prices table which provide product wise actual purchase prices. The combination of vendor and the brand Is unique in this table.
* The vendor\_invoicde table aggregated the data from the purchases table, summarizing quantity and dollar amounts, along with an additional column for freight. This table meaintains uniqueness in terms of vendor and PO number.
* The sales table captures actual sales transactions detailing the brands purchased by the vendors, the quantity sold, the selling price and the revenue earned.

As the data we need to analyse is distributed in different tables we need to create a summary table containing:

* Purchase transactions made by vendors
* Sales transaction data
* Freight cost for each vendor
* Actual product prices from vendors

We create three summary tables first, one of them even using joins between two tables –

Purchase\_summary, freight\_summar, sales\_summary

We then create final table named – vendor\_saleas\_summary out of these three summary tables using JOINS and CTEs.

* The query involves heavy joins and aggregation on large datasets like sales and purchase
* Storing the pre aggregated results avoid repeated espsensive computations
* Helps in anlayzing sales, purchase and pricing for different vendors and brands
* Future benefits of storing this data is for faster dashboarding and reporting

We then look for any discrepencies and clean the data. We add 4 new features /columns into the table and finally add this final table of vendor\_sales\_summary into the database.

(\*\* we can also create a script out of this entire sql code with appropriate functions, including logging. This script is useful when such tasks are to be scheduled or automated.\*\*)

**STEP 3: EDA in Python**

We analyzed the data in sql to on the basis of the values in it and the columns to finally reach an aggregated summary table having only and all the required columns.

We identified key variables, understand their relationships and also determine which one should be included in the final analysis.

Here we need to further analyze the data with visuals, check for outliers, how they are affecting the data,etc.

Will analyze the resultant table to gain insights into the distribution of each column. This will help us understand data patterns, identify anomalies, and ensure data quality before proceeding with further analysis.

We perform the **Summary Statistics**--

We start by plotting the histograms for each column and then plot the boxplots for the same for better identification of the outliers. These are our **observations**:

* Negative and zero values
  + Gross profit : minimum value is -52,002.78 indicating losses. Some products or transactions maybe selling at a loss due to hight costs or maybe selling at discounted prices lower than the purchase prices.
  + Profit margin: has a minimum of -infinity which suggests cases when revenue is zero or even lower than costs.
  + Total sales quantity and sales dollars: minimum values are 0, meaning some products were purchases but never sold. These could be slow moving or obsolete stock.
* Outliers indicated by high standard deviation
  + Purchase and Actual prices: the max values for these columns are significantly higher than the mean indicating potential premium products.
  + Freight cost: Huge variation, from 0.09 to 257,032.07, suggests logistics inefficiency or bulk shipments
  + Stock Turnover: ranges from 0 to 274.5 implying some products sell really fast while other remain in stock indefinitely. Value more than q2 suggests sold quantiy fir that product is higher than purchase quantity.

We **remove inconsistencies** and then code certain plots like the **boxplots** again for the numerical columns. The **histplot** for categorical columns like VendorName and Descritpion(Product name).

Then we plot a **correlation heatmap** to find the correlation between different featured and these are our observations:

* PurchasePrice has weak correlation with TotalSalesDollar and GrossProfit suggesting that the price variations do not significantly impact sales revenue or profit.
* Strong correlation between total purchase quantity and total sales quantity confirming efficient inventory turnover.
* Negative correlation between profit margin and total sale price which suggests that as sales price increases margins decrease, possibly due to competitive pricing pressures.
* StockTurnover has weak negative correlations with both GrossProfit and ProfitMargin, indicating faster turnover doesn’t necessarily result in higher profits.

**First business problem**: Identify underperforming brands that require promotional or pricing adjustments.

For finding solutions/insights for this problem, we try to first create threshold values for low sales and high profit margin with the help of quantile on brand\_performance df.

And then we use those threshold value to filter the df to find the target brands.

And then we finally plot the above on a scatter plot. (filtering out the totalsalesdollars <10000 in brand\_perfomance)

**Second business problem:** which vendors and brands demonstrate highest sales performance

Here we create variables called top\_vendors and top\_brands which filters the top 10 highest performing ones.

We also create a function called format\_dollars for changing the type/format of values for top vendors and brands.

Then we finally plot them both on barchats.

Then we moved on to find out ‘which vendors contribute the most to total purchase dollars’. Here we added a new feature called **PurchaseContribution%** to see the vendor purchase contribution percentages.

We created a variable called **vendor\_performance** and then eventually found out the top 10 vendors and also added a new future giving us the **Cumulative Contribution%.**

Then finally both of these contributions are plotted on the on a **Pareto chart** which combines bar chart and a line chart.

And finally to show ‘how much of total procurement is dependent on the top vendors’, we plot a pie chart but also create a white circle in between to give it a donut look.

Here we show total contributions, purchase contributions (in %) and the remaining contributions i.e contributions by vendors other than top 10, which is 100 – purchase contribution %.

**Third Business Problem:** does purchasing in bulk reduce the unit price and what is the optimal volume for cost savings.

Here we first introduce a new feature called Unit Purchase Prize which is TotalPurchaseDollars/ TotalPurchaseQuantity.

Then we also divide the df as the ‘OrderSize’ which divide the data into Small, Medium and Large and plot it on a boxplot.

Here are the observations –

* Vendors buying in bulk get the lowest unit price($10.78 per unit) meaning higher margins if they can manage inventory efficiently.
* The price difference between Small and Large order is substantial(~72% reduction in unit cost)
* This suggest that bulk pricing strategies successfully encourage vendors to purchase in large volumes, leading to higher overall sales despite lower per unit revenue.

Fourth Business Problem: which vendors have low inventory turnover, indicating excess stock and slow moving products

Here we used groupby and sorting functions to simply find out list of top 10 vendors whose stock turnover is less than 1 .

Fifth Business Problem: how much capital is locked in unsold inventory per vendor, and which vendors contribute the most to it

Here we added a new feature to find out Unsold Inevtory Value which is total purchase quantity – total sales quantity divided by the total purchase price.

The we find out the aggregate capital sold per vendor and then use it to find out the list of top 10 vendors with highest locked capital.

We also work on an additional problem - what are the 95% confidence intervals for profit margins of top-performing and low-performing vendors.

Here we set quantile at 0.25 and 0.75 of TotalSalesDollar which is basically the top and low threshold and use it to find top\_vendors and low\_vendors.

Then we define a confidence\_interval function and use it to find the top and low mean, upper and lower for top and low vendors.

Finally, we plot both of these using histplots.

Here are the observations –

* The confidence interval for low\_performing vendors (40.48% to 42.62%) is significantly higher than that of top\_performing vendors (30.74% to 31.61%)
* This suggests that vendors with lower sales tend to maintain higher profit margins, potentially due to premium pricing or lower operational costs.
* For High\_performing vendors: if they aim to improve profitability, they could explore selective price adjustments, cost optimization, or bundle strategies.
* For Low\_performing vendors: despite higher margins, their low sales volume might indicate a need for better marketing, competitive pricing or improved distribution strategies.