01/01/2023

# FINANCIAL ANALYSIS

PRESENTED BY : DISHA SINDHI

# TOPIC'S

- Introduction
- Power BI
- DAX functions
- Goal
- Problem Statement

# INTRODUCTION

Financial analysis involves evaluating a company's financial statements to assess its performance, stability, and profitability. It is an essential process for stakeholders such as investors, management, and creditors to make informed decisions.

## Purpose of Financial Analysis

- Performance Assessment: To gauge the financial health and operational efficiency of a business.
- Decision Making: To support investment, lending, and management decisions.
- Trend Analysis: To identify patterns and predict future financial conditions.

# INTRODUCTION

## Power BI

Power BI is a powerful business analytics tool developed by Microsoft, designed to transform raw data into meaningful insights through interactive visualizations and reports. Creating reports in Power BI involves combining data sources, modeling data, and using dynamic visualizations to make data accessible and actionable.

## DAX Functions

DAX (Data Analysis Expressions) is a powerful formula language used in Microsoft Power BI, Excel Power Pivot, and SQL Server Analysis Services. It is designed to create calculated columns, measures, and tables, enabling users to perform advanced data analysis and create dynamic reports.

# GOAL IS TO CALCULATE FINANCIAL METRICS

- Running totals
- Moving averages
- Growth rates,
- KPIs
- Credit utilization
- Delinquency risk

# Problem Statement

- **Running Total** of Credit Card Transactions
- Calculate the **4-week moving average** of the creditLimit for each client.
- Calculate the **mom% growth and wow% groth** on transaction amount.
- Calculate **Customer Acquisition Cost (CAC)** as a Ratio of Transaction Amount.
- Calculate the **yearly average** of **avg_utilization_ratio** for all clients.
- Calculate the **percentage of Interest_Earned** compared to **Total_Revolving_Bal** for each client.
- Calculate **Top 5 Clients** by Total Transaction Amount.
- Identify clients whose **Avg_Utilization_Ratio exceeds 80%**.
- **Customer Churn Indicator**: Create a KPI that flags clients who have not made any transactions (Total_Trans_Amt = 0) in the **last 6 months.**

- **Delinquency Rate**: Calculate the percentage of clients with Delinquent_Acc > 0.
- **Credit Risk Score:** Create a score for each client based on their Avg_Utilization_Ratio, Delinquent_Acc, and Total_Revolving_Bal.
- **Income vs Credit Limit Correlation:** Show the correlation between Income and Credit_Limit for all clients.
- **Average Customer Satisfaction Score by Credit Card Category**: Calculate the average Cust_Satisfaction_Score by Card_Category.
- **Loan Approval vs Credit Limit:** Analyze how Credit_Limit affects Personal_loan approval by calculating the average credit limit for clients with and without loans.
- **High Risk Clients Flag:** Create a flag for clients whose Total_Revolving_Bal exceeds 90% of their Credit_Limit and who have a high Avg_Utilization_Ratio.

# Running Total of Credit Card Transactions

```
1  Table = {BLANK()}
```

Table = {BLANK()}: Creates a single-row table containing a blank value, likely as a placeholder.

Total Transaction Amount: Calculates the sum of the Total_Trans_Amt column in the Credit Card table.

```
1  Total Transaction Amount = sum('Credit Card'[Total_Trans_Amt])
```

Running Total: Computes a cumulative total of Total Transaction Amount based on the Week_Start_Date, showing the progression of transactions over time.

```
Running Total = CALCULATE([Total Transaction Amount],
FILTER(ALL(credit_card),credit_card[Week_Start_Date]
<= MAX(credit_card[Week_Start_Date])))
```

# Calculate the 4_week moving average of the creditLimit for each client.

```
1 Calendar = CALENDAR(DATE(2023,1,1),MAX(credit_card[Week_Start_Date]))
```

The Calendar table provides a backbone for date-based analysis and ensures consistency in time calculations.

The Week num column is essential for grouping and aggregating data on a weekly basis.

```
1 Week num = WEEKNUM('Calendar'[Date])
```

```
Moving_average_4_week =

var weeks4 = DATESINPERIOD('Calendar'[Date],MAX('Calendar'[Date]), -28,day)

var total_amount = CALCULATE([Total Transaction Amount],weeks4)

var num_of_weeks = CALCULATE(DISTINCTCOUNT('Calendar'[Week num]),weeks4)

RETURN DIVIDE( total_amount,num_of_weeks,0)
```

The Moving_average_4_week measure is useful for trend analysis, as it reduces noise and highlights underlying patterns in transaction data over a rolling 4-week period.

# Calculate the mom% growth and wow% groth on transaction amount.

```
MOM%growth =

var prev_month = CALCULATE([Total
Transaction Amount],DATEADD('Calendar'
[Date],-1,MONTH))

RETURN DIVIDE([Total Transaction Amount]
-prev_month,prev_month,0)
```

Month-over-month growth provides a medium-term performance measure. It is commonly used for strategic analysis, such as tracking improvements due to campaigns or seasonal trends.

Useful for identifying broader patterns or evaluating strategies on a larger scale.

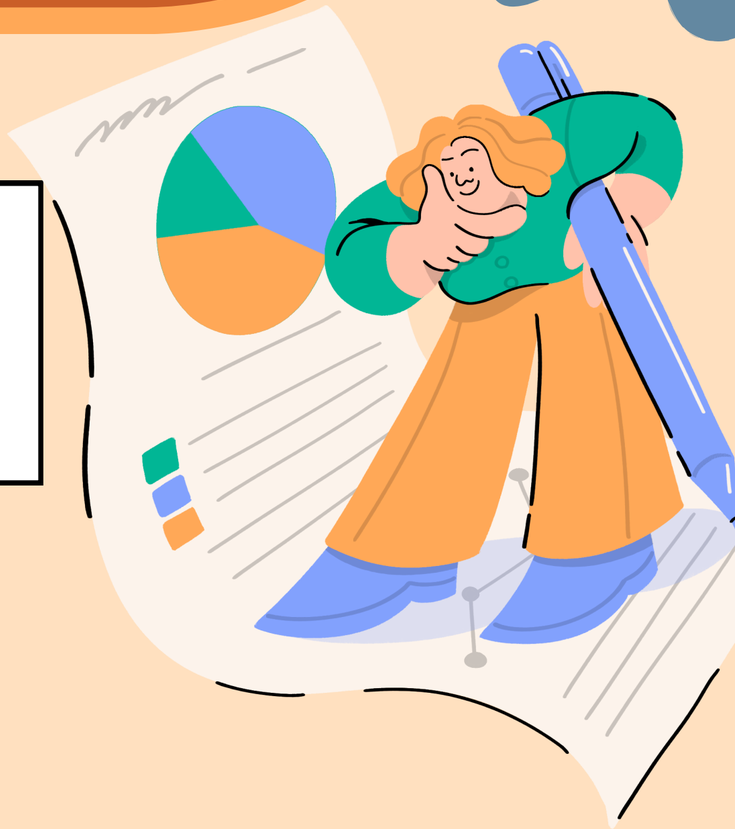
```
WOW%growth =

var prev_week = CALCULATE([Total Transaction
Amount],DATEADD('Calendar'[Date],-7,DAY))

RETURN DIVIDE([Total Transaction Amount]
-prev_week,prev_week,0)
```

Week-over-week growth is critical in businesses that track short-term performance, like retail or online sales. It helps identify sudden spikes or declines.

Useful for understanding the impact of weekly events or short-term decisions.

# Calculate Customer Acquisition Cost (CAC) as a Ratio of Transaction Amount.

```
Ratio_cac_transaction_amount =
 DIVIDE(SUM(credit_card[Customer_Acq_Cost]),[Total Transaction Amount],0)
```



The formula calculates the ratio of Customer Acquisition Cost (CAC) to the total transaction amount. This metric is critical for assessing the efficiency of customer acquisition strategies, allowing businesses to optimize costs and improve profitability.

# Calculate the yearly average of avg_utilization_ratio for all clients.

```
avg_utilization_ratio =
AVERAGE(credit_card[Avg_Utilization_Ratio])
```

This formula is used to compute the average credit utilization ratio for all entries in the credit_card dataset. The credit utilization ratio is a key financial metric that indicates how much credit is being used compared to the available limit.

# Calculate the percentage of Interest_Earned compared to Total_Revolving_Bal for each client.

```
Interest_earned_by_revol_balance =
DIVIDE(SUM('credit_card'[Interest_Earned]),
SUM('credit_card'[Total_Revolving_Bal]))
```

The formula calculates the proportion of interest earned derived from the total revolving bal. It's a critical financial metric for credit card companies to assess income efficiency relative to customer credit behavior.

# Calculate Top 5 Clients by Total Transaction Amount.

```
Tpo_5_Clints = TOPN(5,SUMMARIZE('credit_card','credit_card'[Client_Num],"total amount",
[Total Transaction Amount]),[total amount],DESC)
```

This formula identifies the top 5 clients with the highest total transaction amounts in descending order. The results are useful for understanding high-value customers and analyzing trends in customer transactions.

Without SUMMARIZE, the TOPN function would work on the raw credit_card table, which may have multiple rows per client, making it impossible to compute aggregated totals for ranking.

# Identify clients whose Avg_Utilization_Ratio exceeds 80%.

```
Check_exceeds_80 = IF([Avg_Utilization_Ratio]>0.80,TRUE,FALSE)
```

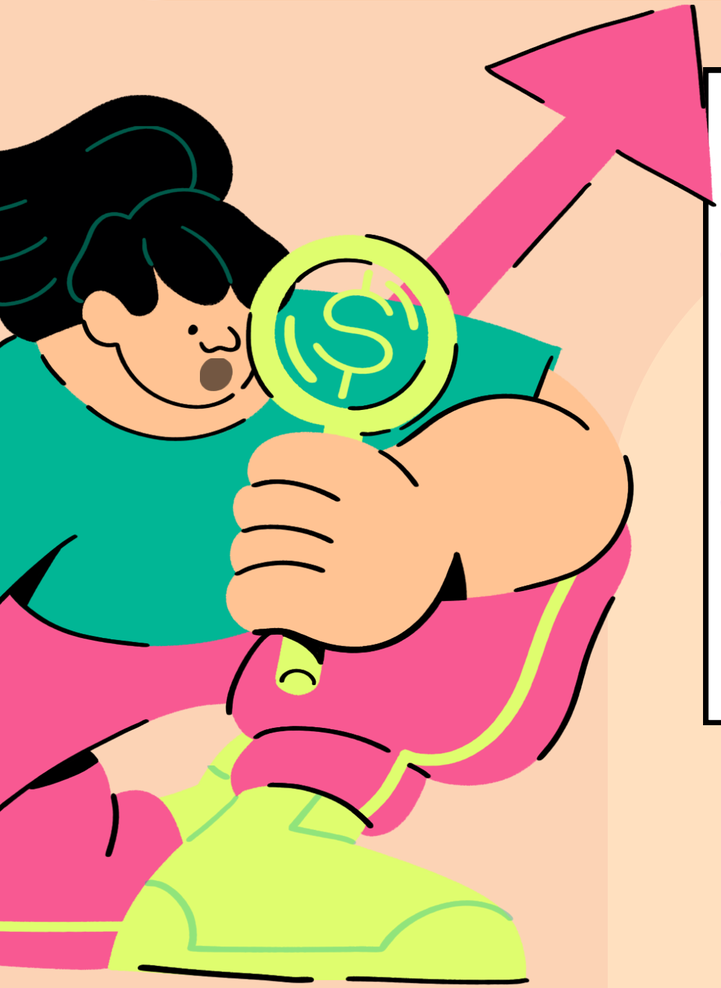
This formula creates a binary indicator (TRUE or FALSE) to classify data based on whether the Avg_Utilization_Ratio exceeds 80%.
Highlighting or filtering data for further analysis.
Creating visualizations that distinguish records based on the condition.
Supporting decision-making by identifying high utilization cases.

# Customer Churn Indicator: Create a KPI that flags clients who have not made any transactions (Total_Trans_Amt = 0) in the last 6 months.

```
Churn =

VAR balance =  CALCULATE([Total Transaction Amount],DATESINPERIOD('Calendar'[Date],MAX
('Calendar'[Date]),-6,MONTH))

RETURN if(ISBLANK(balance),"Churned","Not Churned")
```

The formula uses a 6-month transaction window to classify customers as "Churned" or "Not Churned." It applies DAX functions like CALCULATE, DATESINPERIOD, and ISBLANK to dynamically analyze data and return insightful results for retention analysis or business decision-making.

# Delinquency Rate: Calculate the percentage of clients with Delinquent_Acc > 0.



```
Delinquency rate =

VAR greater_zero = CALCULATE(COUNTROWS('credit_card'),
'credit_card'[Delinquent_Acc]>0)

VAR Total_rows = COUNTROWS('credit_card')

RETURN DIVIDE(greater_zero,Total_rows,0)
```

The formula calculates the ratio of delinquent accounts to total accounts using DAX functions like CALCULATE, COUNTROWS, and DIVIDE. The use of variables improves readability and performance. This measure is valuable for identifying trends in delinquency and assessing the effectiveness of credit management strategies.

# Credit Risk Score: Create a score for each client based on their Avg_Utilization_Ratio, Delinquent_Acc, and Total_Revolving_Bal.

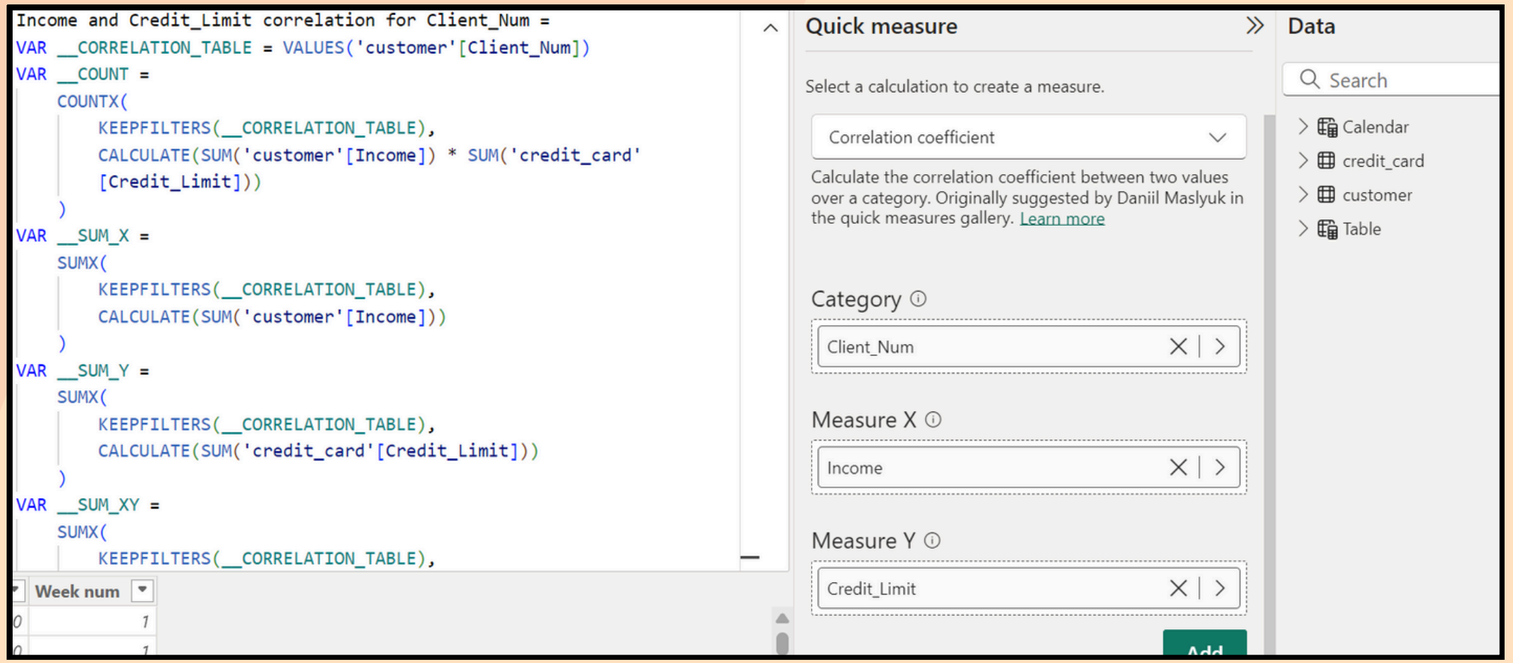
```
Normalized_revlving_balance =
DIVIDE('credit_card'[Total_Revolving_Bal]-MIN('credit_card'
[Total_Revolving_Bal]),MAX('credit_card'[Total_Revolving_Bal])-MIN
('credit_card'[Total_Revolving_Bal]),0)
```

Normalization is used to scale data to a standard range (0 to 1 in this case) for comparison purposes. It ensures that features with larger scales (monetary values) don't dominate smaller-scale features (ratios or percentages) when combining them.

```
Credit_risk_score =
[Avg_Utilization_Ratio]* 0.5 +
'credit_card'[Normalized_revlving_balance]* 0.3 +
'credit_card'[Delinquent_Acc]* 0.2
```

combines multiple relevant factors (credit utilization, revolving balance, delinquency) into a single score. Weighting allows prioritization of the most impactful factors based on domain knowledge or historical data.

# Income vs Credit Limit Correlation: Show the correlation between Income and Credit_Limit for all clients.




How the Quick Measure Helps:
The "Quick Measure" feature in Power BI automatically generates this complex DAX code for you. Instead of writing it manually, you simply:
Select the Category: Client_Num.
Choose Measure X: Income.
Choose Measure Y: Credit_Limit.
This saves time, avoids errors, and ensures accurate results.

Why Use It?
To understand if higher income is associated with higher credit limits, enabling data-driven decisions in business ( credit analysis).
Quick Measure simplifies this by automating the DAX logic.

# Average Customer Satisfaction Score by Credit Card Category: Calculate the average Cust_Satisfaction_Score by Card_Category.

```
avg_Satisfaction_Score = SUMMARIZE('credit_card','credit_card'
[Card_Category],"avg_Satisfaction_Score",AVERAGE('customer'
[Cust_Satisfaction_Score]))
```

Purpose:To summarize customer satisfaction scores by card category.Provides insights into how satisfied customers are based on the type of credit card they use.
.Why Averaging?
The average is a central measure that reflects the overall customer sentiment for each category.
It's useful for comparing satisfaction levels across multiple groups ( card categories).

# Loan Approval vs Credit Limit: Analyze how Credit_Limit affects Personal_loan approval by calculating the average credit limit for clients with and without loans.

```
Loan_approval_yes = CALCULATE(AVERAGE('credit_card'
[Credit_Limit]),customer[Personal_loan] = "yes")
```

Purpose: This calculates the average credit limit for customers who have a personal loan (customer[Personal_loan] = "yes").
Why?: this helps analyze the average credit limit for customers who already have personal loans.

---

```
Loan_approval_no = CALCULATE(AVERAGE('credit_card'
[Credit_Limit]),customer[Personal_loan] = "no")
```

Purpose: This calculates the average credit limit ('credit_card'[Credit_Limit]) for customers who do not have a personal loan (customer[Personal_loan] = "no").
Why?: This is used to analyze the credit limit patterns for customers without personal loans. Understanding these patterns can help identify customer segments with different loan preferences.

# High Risk Clients Flag: Create a flag for clients whose Total_Revolving_Bal exceeds 90% of their Credit_Limit and who have a high Avg_Utilization_Ratio.

```
Flag_Clients =

IF('credit_card'[Normalized_revlving_balance]>0.9 &&
[Avg_Utilization_Ratio]>0.8, "flagged", "Not flagged")
```

Why IF in This Case?

The goal here is to create a calculated column or measure to identify high-risk clients based on their credit usage. The IF function is ideal because:

It evaluates two specific conditions using logical operators (like &&).

Returns either "flagged" or "Not flagged", which makes it easy for financial analysts to interpret the results.

# CONCLUSION

The financial analysis conducted showcases the power of integrating data visualization and advanced analytical tools like Power BI and DAX to derive actionable insights. By addressing critical metrics such as credit utilization, delinquency risks, and customer churn, the study highlights areas crucial for operational efficiency and strategic decision-making.

Key outcomes include:

Identification of high-value customers and their spending patterns.

Enhanced understanding of financial health through metrics like average utilization ratios and delinquency rates.

Improved customer segmentation based on risk profiles and satisfaction scores.

Correlation analysis between income levels and credit limits, offering insights into customer credit behavior.

These insights empower businesses to optimize resource allocation, refine credit strategies, and improve customer retention efforts. Moreover, the ability to generate real-time dynamic visualizations provides a significant edge in monitoring trends and adapting strategies effectively.

# THANK YOU