**Predicting Customer Churn in Telecom Industry using Power BI and SQL**

Step 1:

**Load Data**: Import the CSV file into MySql Workbench.

Check the data is loaded correctly:

SELECT \* FROM customer\_churn LIMIT 10;

Check the total number of rows:

SELECT COUNT(\*) AS total\_rows FROM customer\_churn;

Check for duplicate rows:

SELECT Customer\_ID, COUNT(\*) FROM customer\_churn GROUP BY Customer\_ID HAVING COUNT(\*) > 1;

Step 2:

Handle Missing Values for Specific Columns

SHOW COLUMNS FROM customer\_churn;

SELECT

'Customer Status',

COUNT(\*) AS Total\_Rows,

SUM(CASE WHEN 'Customer Status' IS NULL THEN 1 ELSE 0 END) AS Missing\_Values

FROM customer\_churn

GROUP BY 'Customer Status';

Step 3:

Clean and Transform the Data

Convert values like "Yes/No" into 1/0 for better analysis:

UPDATE customer\_churn

SET Unlimited\_Data = CASE

WHEN Unlimited\_Data = 'Yes' THEN 1

WHEN Unlimited\_Data = 'No' THEN 0

END;

**Create a Churn\_Flag column to represent churned customers:**

ALTER TABLE customer\_churn ADD Churn\_Flag INT; UPDATE customer\_churn SET Churn\_Flag = CASE WHEN Customer\_Status = 'Churned' THEN 1 ELSE 0 END;

Create a Tenure\_Group column to group customers by their tenure:

ALTER TABLE customer\_churn ADD Tenure\_Group VARCHAR(20);

UPDATE customer\_churn

SET Tenure\_Group = CASE

WHEN Tenure\_in\_Months <= 12 THEN '0-12 Months'

WHEN Tenure\_in\_Months BETWEEN 13 AND 24 THEN '13-24 Months'

WHEN Tenure\_in\_Months > 24 THEN '24+ Months'

END;

Step 4:

Perform Exploratory Data Analysis (EDA): Analyse Churn Rate

**Calculate Overall Churn Rate:**

SELECT

COUNT(\*) AS Total\_Customers,

SUM(CASE WHEN Churn\_Flag = 1 THEN 1 ELSE 0 END) AS Churned\_Customers,

ROUND(SUM(CASE WHEN Churn\_Flag = 1 THEN 1 ELSE 0 END) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn;

**Churn Rate by Tenure Group:**

SELECT

Tenure\_Group,

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY Tenure\_Group;

**Churn Rate by Internet Type:**

SELECT

Internet\_Type,

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY Internet\_Type;

**Churn Rate by Monthly Charges:**

SELECT

CASE

WHEN Monthly\_Charge <= 50 THEN 'Low (<= $50)'

WHEN Monthly\_Charge BETWEEN 51 AND 100 THEN 'Medium ($51-$100)'

ELSE 'High (>$100)'

END AS Charge\_Group,

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY Charge\_Group;

**Analyse Churn by Gender:**

SELECT

Gender,

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY Gender;

**Analyse Churn by Age Group**

SELECT

CASE

WHEN Age < 25 THEN 'Under 25'

WHEN Age BETWEEN 25 AND 35 THEN '25-35'

WHEN Age BETWEEN 36 AND 50 THEN '36-50'

ELSE 'Over 50'

END AS Age\_Group,

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY Age\_Group;

**Churn Rate by Contract Type**

SELECT

'Contract Type',

COUNT(\*) AS Total\_Customers,

SUM(Churn\_Flag) AS Churned\_Customers,

ROUND(SUM(Churn\_Flag) / COUNT(\*) \* 100, 2) AS Churn\_Rate

FROM customer\_churn

GROUP BY 'Contract Type';

**Creating a Dashboard in Power BI**

Load Data

Prepare and Transform Data:

Open Power Query Editor, Explore the Data, Identify Missing Values. And created some new column using Dax

Churn Indicator:

ChurnIndicator = IF('customer\_churn'[Churn\_Flag] = 1, "Churned", "Active")

Churn Rate (%):

ChurnRate = DIVIDE(

COUNTROWS(FILTER('customer\_churn', 'customer\_churn'[Churn\_Flag] = 1)),

COUNTROWS('customer\_churn')

) \* 100

Created dashboard :

**Questions to find the solution of churn analysis:**

**1. Identify the Total Number of Customers and the Churn Rate:**

Create a Card Visual for the total customers using a distinct count of CustomerID.

Create another Card Visual for churn rate using the DAX formula:

DAX

Copy code

ChurnRate = DIVIDE(

COUNTROWS(FILTER(customer\_churn, customer\_churn[Churn\_Flag] = 1)),

COUNTROWS(customer\_churn)

) \* 100

**2. Find the Average Age of Churned Customers:**

Create a **Measure** for the average age of churned customers:

DAX

Copy code

AverageAgeChurned = AVERAGEX(

FILTER(customer\_churn, customer\_churn[Churn\_Flag] = 1),

customer\_churn[Age]

)

**3. Discover the Most Common Contract Types Among Churned Customers**:

Use a **Pie Chart** or **Bar Chart** with Contract\_Type and the count of churned customers.

**4. Analyze the Distribution of Monthly Charges Among Churned Customers**

Use a **Histogram** with MonthlyCharges for churned customers.

**5.Create a Query to Identify Contract Types Most Prone to Churn**

SELECT

Contract\_Type,

COUNT(CASE WHEN Churn\_Flag = 1 THEN 1 END) AS Churned\_Customers,

COUNT(\*) AS Total\_Customers,

ROUND(COUNT(CASE WHEN Churn\_Flag = 1 THEN 1 END) / COUNT(\*) \* 100, 2) AS Churn\_Rate\_Percentage

FROM customer\_churn

GROUP BY Contract\_Type

ORDER BY Churn\_Rate\_Percentage DESC;

**6. Identify Customers with High Total Charges Who Have Churned:**

Create a **Table** visualization to display CustomerID and TotalCharges for churned customers, sorted by total charges.

**7. Calculate the Total Charges Distribution for Churned and Non-Churned Customers**

SELECT

Churn\_Flag,

MIN(`Total Charges`) AS Min\_TotalCharges,

MAX(`Total Charges`) AS Max\_TotalCharges,

AVG(`Total Charges`) AS Avg\_TotalCharges,

SUM(`Total Charges`) AS Total\_Sum

FROM customer\_churn

GROUP BY Churn\_Flag;

**8. Calculate the Average Monthly Charges for Different Contract Types Among Churned Customers**

SELECT

Contract,

AVG(`Monthly Charge`) AS Avg\_MonthlyCharges

FROM customer\_churn

WHERE Churn\_Flag = 1

GROUP BY Contract;

**9.Identify Customers with Both Online Security and Online Backup Services and Have Not Churned**

SELECT

`Customer ID`

FROM customer\_churn

WHERE `Online Security` = 'Yes' AND `Online Backup` = 'Yes' AND Churn\_Flag = 0;

**10. Determine the Most Common Combinations of Services Among Churned Customers**

SELECT

`Online Security`, `Online Backup`, `Streaming TV`, `Streaming Movies`,

COUNT(\*) AS Count

FROM customer\_churn

WHERE Churn\_Flag = 1

GROUP BY `Online Security`,`Online Backup`, `Streaming TV`, `Streaming Movies`

ORDER BY Count DESC;

**11. Identify the Average Total Charges for Customers Grouped by Gender and Marital Status:**

Create a **Matrix** visual. Place **Gender** and **MaritalStatus** in Rows and **Average of TotalCharges** in Values.

**12. Calculate the Average Monthly Charges for Different Age Groups Among Churned Customers**

Create a **calculated column** for Age Groups:

AgeGroup =

IF([Age] < 25, "Under 25",

IF([Age] <= 40, "25-40",

IF([Age] <= 60, "41-60", "60+")))

Use a **Bar Chart** to visualize the average of MonthlyCharges by AgeGroup.

**13. Determine the Average Age and Total Charges for Customers with Multiple Lines and Online Backup**

SELECT

AVG(Age) AS AvgAge,

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

WHERE `Multiple Lines` = 'Yes' AND `Online Backup` = 'Yes';

**14. Identify the Contract Types with the Highest Churn Rate Among Senior Citizens (Age 65 and Over)**

SELECT

Contract,

COUNT(\*) AS Total,

SUM(CASE WHEN Churn\_Flag = 1 THEN 1 ELSE 0 END) AS Churned,

SUM(CASE WHEN Churn\_Flag = 1 THEN 1 ELSE 0 END) / COUNT(\*) \* 100 AS ChurnRate

FROM customer\_churn

WHERE Age >= 65

GROUP BY Contract

ORDER BY ChurnRate DESC;

**15. Calculate the Average Monthly Charges for Customers Who Have Multiple Lines and Streaming TV**

SELECT

AVG(`Monthly Charge`) AS AvgMonthlyCharges

FROM customer\_churn

WHERE `Multiple Lines` = 'Yes' AND `Streaming TV` = 'Yes';

**16. Identify the Customers Who Have Churned and Used the Most Online Services**

SELECT

`Customer ID`,

(CASE WHEN `Online Security` = 'Yes' THEN 1 ELSE 0 END +

CASE WHEN `Online Backup` = 'Yes' THEN 1 ELSE 0 END +

CASE WHEN `Device Protection Plan` = 'Yes' THEN 1 ELSE 0 END +

CASE WHEN `Premium Tech Support` = 'Yes' THEN 1 ELSE 0 END) AS OnlineServicesCount

FROM customer\_churn

WHERE Churn\_Flag = 1

ORDER BY OnlineServicesCount DESC

LIMIT 10;

**17. Calculate the Average Age and Total Charges for Customers with Different Combinations of Streaming Services**

SELECT

`Streaming TV`,

`Streaming Movies`,

AVG(Age) AS AvgAge,

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

GROUP BY `Streaming TV`, `Streaming Movies`;

**18.Identify the Gender Distribution Among Customers Who Have Churned and Are on Yearly Contracts**

SELECT

Gender,

COUNT(\*) AS Count

FROM customer\_churn

WHERE Churn\_Flag = 1 AND Contract = 'One Year' & 'Two year'

GROUP BY Gender;

**19. Average Monthly and Total Charges for Churned Customers Grouped by Contract Type and Internet Service Type**

SELECT

Contract,

`Internet Service`,

AVG(`Monthly Charge`) AS AvgMonthlyCharges,

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

WHERE Churn\_Flag = 1

GROUP BY Contract, `Internet Service`;

**20. Identify Churned Customers Not Using Online Services and Their Average Total Charges**

SELECT

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

WHERE Churn\_Flag = 1 AND `Premium Tech Support` = 'No';

**21. Average Monthly and Total Charges for Churned Customers Grouped by Number of Dependents**

SELECT

`Number of Dependents`,

AVG(`Monthly Charge`) AS AvgMonthlyCharges,

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

WHERE Churn\_Flag = 1

GROUP BY `Number of Dependents`;

**22. Contract Duration in Months for Churned Customers with Monthly Contracts**

SELECT

`Customer ID`,

`Tenure in Months` AS ContractDurationMonths

FROM customer\_churn

WHERE Churn\_Flag = 1 AND Contract = 'Month-to-Month';

**23. Determine the Average Age and Total Charges for Customers Who Have Churned, Grouped by Internet Service and Phone Service**

SELECT

`Internet Service`,

`Phone Service`,

AVG(Age) AS AvgAge,

AVG(`Total Charges`) AS AvgTotalCharges

FROM customer\_churn

WHERE Churn\_Flag = 1

GROUP BY `Internet Service`, `Phone Service`;

**24. Create a View to Find the Customers with the Highest Monthly Charges in Each Contract Type**

CREATE VIEW HighestMonthlyCharges AS

SELECT

Contract,

`Customer ID`,

`Monthly Charge`

FROM customer\_churn c1

WHERE `Monthly Charge` = (

SELECT MAX(`Monthly Charge`)

FROM customer\_churn c2

WHERE c1.Contract = c2.Contract

);

SELECT \* FROM HighestMonthlyCharges ;

**25. Create a View to Identify Customers Who Have Churned and the Average Monthly Charges Compared to the Overall Average**

CREATE VIEW ChurnedVsOverallAvg AS

SELECT

`Customer ID`,

`Monthly Charge`,

(SELECT AVG(`Monthly Charge`) FROM customer\_churn) AS OverallAvgMonthlyCharges

FROM customer\_churn

WHERE Churn\_Flag = 1;

SELECT \* FROM ChurnedVsOverallAvg;

**26. Create a View to Find the Customers Who Have Churned and Their Cumulative Total Charges Over Time**

CREATE VIEW CumulativeTotalCharges AS

SELECT

`Customer ID`,

`Tenure in Months`,

SUM(`Total Charges`) OVER (PARTITION BY `Customer ID` ORDER BY `Tenure in Months`) AS CumulativeTotalCharges

FROM customer\_churn

WHERE Churn\_Flag = 1;

SELECT \* FROM CumulativeTotalCharges;

**27. Stored Procedure to Calculate Churn Rate**

DELIMITER $$

CREATE PROCEDURE CalculateChurnRate()

BEGIN

SELECT

(SUM(CASE WHEN Churn\_Flag = 1 THEN 1 ELSE 0 END) / COUNT(\*)) \* 100 AS ChurnRate

FROM customer\_churn;

END $$

DELIMITER ;

CALL CalculateChurnRate();

**28. Stored Procedure to Identify High-Value Customers at Risk of Churning**

DELIMITER $$

CREATE PROCEDURE IdentifyHighValueChurnRisk()

BEGIN

SELECT

`Customer ID`,

`Total Charges`,

`Monthly Charge`,

Contract

FROM customer\_churn

WHERE Churn\_Flag = 1 AND `Total Charges` > (

SELECT AVG(`Total Charges`) FROM customer\_churn

)

ORDER BY `Total Charges` DESC;

END $$

DELIMITER ;

CALL IdentifyHighValueChurnRisk();

**Thank You**