Customer Churn Analysis Report

Project title:

Predicting Customer Churn in Telecom Industry using Power BI or Tableau and SQL.

Problem Statement:

Predicting Customer Churn in Telecom Industry

Business Use Cases:

Customer Retention: Identify at-risk customers and proactively implement retention strategies.

Marketing Campaigns: Tailor marketing efforts towards customers who are more likely to churn.

Service Improvement: Analyze churn patterns to improve service offerings and customer support.

Revenue Optimization: Reduce churn rates to maintain a steady revenue stream.

Customer Segmentation: Segment customers based on churn probability to offer personalized experiences.

Approach:

1. Data Collection: Use SQL to query the telecom database and extract relevant data.

- Step 1: Schema creation
  Created a schema named "customer_churn".

- Step 2: Table creation
  Created tables within the customer_churn schema.

The following query I wrote for the table creation of customer churn:

```
CREATE TABLE Customer_Churn (
    customer_id VARCHAR(20) PRIMARY KEY,
    gender ENUM('Male', 'Female'),
    age INT,
    married ENUM('Yes', 'No'),
    number_of_dependents INT,
    city VARCHAR(100),
    zip_code VARCHAR(10),
    latitude DECIMAL(10, 6),
    longitude DECIMAL(10, 6),
    number_of_referrals INT,
    payment_method VARCHAR(50),
    monthly_charge DECIMAL(10, 2),
    total_charges DECIMAL(10, 2),
    total_refunds DECIMAL(10, 2),
    total_extra_data_charges INT,
    total_long_distance_charges DECIMAL(10, 2),
    total_revenue DECIMAL(10, 2),
    customer_status VARCHAR(20),
    churn_category VARCHAR(50),
    churn_reason VARCHAR(100));
```

The following are the queries I wrote for customer churn to analyse the dataset:

1. Identify the Number of Churned Customers by Service Type.

```
SELECT
    `Internet Service`,
    `Phone Service`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM
    customer_churn
WHERE
    `Customer Status` = 'Churned'
GROUP BY
    `Internet Service`, `Phone Service`;
```

This query will show the number of customers who churned based on the type of service they subscribed to (e.g., phone, internet).

2. Analyze Churn by Tenure:

```
SELECT   CASE

        WHEN `Tenure in Months` < 12 THEN '0-12 Months'
        WHEN `Tenure in Months` BETWEEN 12 AND 24 THEN '12-24
Months'
        ELSE '24+ Months'
    END AS `Tenure Range`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM
    customer_churn
WHERE
    `Customer Status` = 'Churned'
GROUP BY
    `Tenure Range`;
```

This query analyzes how customer tenure (in months) impacts churn rates, grouping customers by tenure ranges.

3. Find the Average Monthly Charges of Churned vs. Retained Customers:

```
SELECT
```

```
    `Customer Status`,
    AVG(`Monthly Charge`) AS `Avg Monthly Charge`
FROM
    customer_churn
GROUP BY
    `Customer Status`;
```

This query compares the average monthly charges for customers who churned versus those who stayed.

4. Analyze Churn by Contract Type:

```
SELECT
    `Contract`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM   customer_churn
WHERE `Customer Status` = 'Churned'
GROUP BY
    `Contract`;
```

This query shows how churn is distributed across different contract types.

5. Analyze Churn by Payment Method:

```
SELECT
    `Payment Method`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM
    customer_churn
WHERE
    `Customer Status` = 'Churned'
```

```sql
GROUP BY
    `Payment Method`
ORDER BY
    `Churned Customers` DESC;
```

This query helps identify if certain payment methods are more associated with churn.

6. Identify High-Risk Customers Based on Monthly Charges and Low Tenure:

```sql
SELECT
    `Customer ID`,
    `Tenure in Months`,
    `Monthly Charge`,
    `Customer Status`
FROM
    customer_churn
WHERE  `Customer Status` = 'Stayed'
    AND `Monthly Charge` > 80
    AND `Tenure in Months` < 12;
```

This query selects customers who are still active but are considered high-risk due to high monthly charges and low tenure.

7. Find Churned Customers by Reason and Service Type:

```sql
SELECT
    `Internet Service`,
    `Phone Service`,
    `Churn Reason`,
    COUNT(`Customer ID`) AS `Churned Customers`
```

```sql
FROM
    customer_churn
WHERE
    `Customer Status` = 'Churned'
GROUP BY
    `Internet Service`, `Phone Service`, `Churn Reason`;
```

This query will give insight into the main reasons for churn, categorized by the type of service.

8. Find Churn Based on Demographics (Gender, Age):

```sql
SELECT
    Gender,
    CASE
        WHEN Age < 30 THEN 'Under 30'
        WHEN Age BETWEEN 30 AND 50 THEN '30-50'
        ELSE 'Over 50'
    END AS Age_Group,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM   customer_churn
WHERE  `Customer Status` = 'Churned'
GROUP BY  Gender, Age_Group;
```

This query shows churn rates broken down by gender and age group, allowing for a demographic analysis.

9. Determine Churn by Data Usage:

```sql
SELECT
    CASE
```

```
        WHEN `Avg Monthly GB Download` < 10 THEN 'Low Usage (<10 GB)'
        WHEN `Avg Monthly GB Download` BETWEEN 10 AND 50 THEN 'Moderate Usage (10-50 GB)'
        ELSE 'High Usage (>50 GB)'
    END AS `Data Usage Group`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM
    customer_churn
WHERE  `Customer Status` = 'Churned'
GROUP BY
    `Data Usage Group`;
```

This query helps you identify whether heavy data users are more likely to churn.

10. Analyze Churn by Streaming Service Usage:

```
SELECT
    CASE
        WHEN `Streaming TV` = 'Yes' OR `Streaming Movies` = 'Yes' THEN 'Using Streaming Services'
        ELSE 'Not Using Streaming Services'
    END AS `Streaming Usage`,
    COUNT(`Customer ID`) AS `Churned Customers`
FROM
    customer_churn
WHERE  `Customer Status` = 'Churned'
GROUP BY  `Streaming Usage`;
```

This query shows churn based on customers who use streaming services (e.g., TV, movies, music).

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

```
SHOW CREATE PROCEDURE identify_high_value_customers_at_risk;
DROP PROCEDURE IF EXISTS identify_high_value_customers_at_risk;
DELIMITER //

CREATE PROCEDURE identify_high_value_customers_at_risk(
    IN p_revenue_threshold DECIMAL(10, 2),
    IN p_charge_threshold DECIMAL(10, 2)
)
BEGIN
    SELECT
        `Customer ID` AS customer_id,
        Gender AS gender,
        Age AS age,
        City AS city,
        `Total Revenue` AS total_revenue,
        `Monthly Charge` AS monthly_charge,
        `Churn Category` AS churn_category,
        `Churn Reason` AS churn_reason
    FROM
        customer_churn
    WHERE
        (`Total Revenue` > p_revenue_threshold OR `Monthly Charge` >
p_charge_threshold)
```

```
    AND (`Customer Status` = 'Churned' OR `Churn Category` IS NOT NULL);
END //

DELIMITER ;

CALL identify_high_value_customers_at_risk(10000.00, 80.00);
DESCRIBE customer_churn;
```

2: Data preprocessing :

- Handled the missing values.
- Handled outliers.
- Normalized numerical features using Min-Max scaling.
- Encoded categorical variables.
- Cleaned customer status and churn category.
- Final overview of the cleaned data.

Data proprocessing is done using python for writing the code I used Google colab.

This is overall code for the process:

```
import pandas as pd

import numpy as np

from sklearn.preprocessing import MinMaxScaler, LabelEncoder


# Load the dataset

df = pd.read_csv('/mnt/data/customer_churn (1).csv')
```

Step 1: Handling Missing Values

```
# Fill missing values for numerical columns with the median
df['Avg Monthly Long Distance Charges'].fillna(df['Avg Monthly Long Distance Charges'].median(), inplace=True)
df['Avg Monthly GB Download'].fillna(df['Avg Monthly GB Download'].median(), inplace=True)

# Fill missing values for categorical columns with the most frequent value (mode)
categorical_columns = ['Multiple Lines', 'Internet Type', 'Online Security', 'Online Backup',
                       'Device Protection Plan', 'Premium Tech Support', 'Streaming TV',
                       'Streaming Movies', 'Streaming Music', 'Unlimited Data']

for column in categorical_columns:
    df[column].fillna(df[column].mode()[0], inplace=True)
```

Step 2: Handling Outliers (using IQR method)

```
def cap_outliers(col):
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df[col] = np.where(df[col] < lower_bound, lower_bound, df[col])
    df[col] = np.where(df[col] > upper_bound, upper_bound, df[col])
```

```
numerical_columns_with_outliers = ['Monthly Charge', 'Total Charges', 'Total Revenue']

for col in numerical_columns_with_outliers:
    cap_outliers(col)


Step 3: Normalizing Numerical Features using Min-Max scaling

scaler = MinMaxScaler()

numerical_columns = ['Monthly Charge', 'Total Charges', 'Total Revenue', 'Avg Monthly Long Distance Charges', 'Avg Monthly GB Download']

df[numerical_columns] = scaler.fit_transform(df[numerical_columns])


# Step 4: Encoding Categorical Variables

# Label encoding for binary columns

binary_columns = ['Gender', 'Married', 'Phone Service', 'Paperless Billing']

le = LabelEncoder()

for col in binary_columns:
    df[col] = le.fit_transform(df[col])


# One-hot encoding for non-binary categorical columns

df = pd.get_dummies(df, columns=['Contract', 'Payment Method', 'Internet Type', 'Offer'], drop_first=True)


# Step 5: Cleaning Customer Status and Churn Category

# Filling missing values in 'Churn Category' with 'No Churn'

df['Churn Category'].fillna('No Churn', inplace=True)

df['Churn Reason'].fillna('No Reason', inplace=True)
```

```
# Drop irrelevant columns
df_cleaned = df.drop(['Customer ID', 'Customer Status', 'Churn Reason'], axis=1)

# Step 6: Final Overview of the Cleaned Data
print("Cleaned Data Overview:")
print(df_cleaned.info())
print(df_cleaned.head())

# Save cleaned data to a new CSV file (optional)
df_cleaned.to_csv('/mnt/data/customer_churn_cleaned.csv', index=False)
```

3. Exploratory Data Analysis:

Loaded the dataset into the powerbi and cleaned some of the columns in power query editor.

In power query editor added the new column named "Tenure category" for better understanding of pattern in customer churn.

Then made visualizations using the charts.

Charts for analysis:

1. Stacked bar chart:
   Compared churn customers with category of customers and Customer ID.

2. Line chart:
   Compared customer category with total average revenue and average monthly charges this will be useful to identify the churn rate based on monthly charges.

3. Waterfall chart:

Compared tenure category with tenure in months and average total long distance charges.

4. Scatter plot:
   Compared customer status with sum of monthly charges and sum of total revenue.

5. Donut chart:
   Compared churn category with count of CustomerID.

6. Funnel chart:
   For the funnel chart calculated dax function with total customers and compared churn category with total customers.

Visualization:

Created dashboard with the charts created for the understanding of the analysis.

Retention strategies:

- Offering loyalty rewards to customers with short tenure.
- Providing discounts for customers with high monthly charges.
- Improving services (e.g., internet speed, support) to target customers who may churn due to dissatisfaction.
- Effective actions with competitive analysis is what highly required.
- By utilizing customer status we can still retain the customers who are about to churn.

Strategic recommendations:

- Target High Churn Cities: Identifying cities with higher churn rates. These locations may require additional marketing campaigns or improved service.

- Enhance Product Features: "Competitor had better devices" is a recurring reason for churn, consider updating device options or bundling better products to retain customers.
- Referral Programs: Customers with a low “Number of Referrals” might need incentive programs to bring more users to the service, enhancing customer satisfaction and retention.