Project Goals

Create an entire ETL process in a database and a power BI dashboard to utilize the customer Data and achieve below goals

* Analyze Customer Data at below levels (Demographic, Geographic, Payment and Account info, services
* Study Churner profile and identify areas for implementing marketing campaigns
* Identify the method to predict future churners

Metrics Required

* Total Customer
* Total Churn and Churn Rate
* New Joiners

Project Documentation

SQL

Database is created in SSMS, CREATE DATABASE db\_Churn, and the customer table is loaded into that database. The customer table is named stg\_Churn (staging area). Now we explore the data.

CREATE DATABASE db\_Churn

-- create database

-- DATA EXPLORATION (Checking distint values)

SELECT

\*

FROM stg\_Churn

-- Check data

SELECT

Gender,

Count(Gender) as TotalCount,

Count(Gender) \* 1.0 / (Select Count(\*) from stg\_Churn) as Percentage

FROM stg\_Churn

GROUP BY Gender

/\* Check Male and Female composition in database (Count of gender is multiplied by 1.0 to enable use see float values

else, percentage would have be 0, we can multiple by 100 or 100.0 to give actual percentage instead of fractions (36% instead of 0.36)\*/

SELECT

Contract,

Count(Contract) as TotalCount,

Count(Contract) \* 1.0 / (Select Count(\*) from stg\_Churn) as Percentage

FROM stg\_Churn

GROUP BY Contract

-- Checking contract type composition

SELECT

Customer\_Status,

Count(Customer\_Status) as TotalCount,

Sum(Total\_Revenue) as TotalRev,

Sum(Total\_Revenue) / (Select sum(Total\_Revenue) from stg\_Churn) \* 100 as RevPercentage

FROM stg\_Churn

GROUP BY Customer\_Status

-- Checking customer churn status and impact on revenue

SELECT

State,

Count(State) as TotalCount,

Count(State) \* 100.0 / (Select Count(\*) from stg\_Churn) as Percentage

FROM stg\_Churn

GROUP BY State

ORDER BY Percentage desc

-- Checking state by state contribution, also from largest to smallest

SELECT DISTINCT

Internet\_Type

FROM stg\_Churn

-- Check all the kinds of internet in our database

-- DATA EXPLORATION (Checking NULL values)

SELECT

SUM(CASE WHEN Customer\_ID IS NULL THEN 1 ELSE 0 END) AS Customer\_ID\_Null\_Count,

SUM(CASE WHEN Gender IS NULL THEN 1 ELSE 0 END) AS Gender\_Null\_Count,

SUM(CASE WHEN Age IS NULL THEN 1 ELSE 0 END) AS Age\_Null\_Count,

SUM(CASE WHEN Married IS NULL THEN 1 ELSE 0 END) AS Married\_Null\_Count,

SUM(CASE WHEN State IS NULL THEN 1 ELSE 0 END) AS State\_Null\_Count,

SUM(CASE WHEN Number\_of\_Referrals IS NULL THEN 1 ELSE 0 END) AS Number\_of\_Referrals\_Null\_Count,

SUM(CASE WHEN Tenure\_in\_Months IS NULL THEN 1 ELSE 0 END) AS Tenure\_in\_Months\_Null\_Count,

SUM(CASE WHEN Value\_Deal IS NULL THEN 1 ELSE 0 END) AS Value\_Deal\_Null\_Count,

SUM(CASE WHEN Phone\_Service IS NULL THEN 1 ELSE 0 END) AS Phone\_Service\_Null\_Count,

SUM(CASE WHEN Multiple\_Lines IS NULL THEN 1 ELSE 0 END) AS Multiple\_Lines\_Null\_Count,

SUM(CASE WHEN Internet\_Service IS NULL THEN 1 ELSE 0 END) AS Internet\_Service\_Null\_Count,

SUM(CASE WHEN Internet\_Type IS NULL THEN 1 ELSE 0 END) AS Internet\_Type\_Null\_Count,

SUM(CASE WHEN Online\_Security IS NULL THEN 1 ELSE 0 END) AS Online\_Security\_Null\_Count,

SUM(CASE WHEN Online\_Backup IS NULL THEN 1 ELSE 0 END) AS Online\_Backup\_Null\_Count,

SUM(CASE WHEN Device\_Protection\_Plan IS NULL THEN 1 ELSE 0 END) AS Device\_Protection\_Plan\_Null\_Count,

SUM(CASE WHEN Premium\_Support IS NULL THEN 1 ELSE 0 END) AS Premium\_Support\_Null\_Count,

SUM(CASE WHEN Streaming\_TV IS NULL THEN 1 ELSE 0 END) AS Streaming\_TV\_Null\_Count,

SUM(CASE WHEN Streaming\_Movies IS NULL THEN 1 ELSE 0 END) AS Streaming\_Movies\_Null\_Count,

SUM(CASE WHEN Streaming\_Music IS NULL THEN 1 ELSE 0 END) AS Streaming\_Music\_Null\_Count,

SUM(CASE WHEN Unlimited\_Data IS NULL THEN 1 ELSE 0 END) AS Unlimited\_Data\_Null\_Count,

SUM(CASE WHEN Contract IS NULL THEN 1 ELSE 0 END) AS Contract\_Null\_Count,

SUM(CASE WHEN Paperless\_Billing IS NULL THEN 1 ELSE 0 END) AS Paperless\_Billing\_Null\_Count,

SUM(CASE WHEN Payment\_Method IS NULL THEN 1 ELSE 0 END) AS Payment\_Method\_Null\_Count,

SUM(CASE WHEN Monthly\_Charge IS NULL THEN 1 ELSE 0 END) AS Monthly\_Charge\_Null\_Count,

SUM(CASE WHEN Total\_Charges IS NULL THEN 1 ELSE 0 END) AS Total\_Charges\_Null\_Count,

SUM(CASE WHEN Total\_Refunds IS NULL THEN 1 ELSE 0 END) AS Total\_Refunds\_Null\_Count,

SUM(CASE WHEN Total\_Extra\_Data\_Charges IS NULL THEN 1 ELSE 0 END) AS Total\_Extra\_Data\_Charges\_Null\_Count,

SUM(CASE WHEN Total\_Long\_Distance\_Charges IS NULL THEN 1 ELSE 0 END) AS Total\_Long\_Distance\_Charges\_Null\_Count,

SUM(CASE WHEN Total\_Revenue IS NULL THEN 1 ELSE 0 END) AS Total\_Revenue\_Null\_Count,

SUM(CASE WHEN Customer\_Status IS NULL THEN 1 ELSE 0 END) AS Customer\_Status\_Null\_Count,

SUM(CASE WHEN Churn\_Category IS NULL THEN 1 ELSE 0 END) AS Churn\_Category\_Null\_Count,

SUM(CASE WHEN Churn\_Reason IS NULL THEN 1 ELSE 0 END) AS Churn\_Reason\_Null\_Count

FROM stg\_Churn;

-- This will check each column for nulls, and replace it with 1, then sum it all up

SELECT

Customer\_ID,

Gender,

Age,

Married,

State,

Number\_of\_Referrals,

Tenure\_in\_Months,

ISNULL(Value\_Deal, 'None') AS Value\_Deal,

Phone\_Service,

ISNULL(Multiple\_Lines, 'No') As Multiple\_Lines,

Internet\_Service,

ISNULL(Internet\_Type, 'None') AS Internet\_Type,

ISNULL(Online\_Security, 'No') AS Online\_Security,

ISNULL(Online\_Backup, 'No') AS Online\_Backup,

ISNULL(Device\_Protection\_Plan, 'No') AS Device\_Protection\_Plan,

ISNULL(Premium\_Support, 'No') AS Premium\_Support,

ISNULL(Streaming\_TV, 'No') AS Streaming\_TV,

ISNULL(Streaming\_Movies, 'No') AS Streaming\_Movies,

ISNULL(Streaming\_Music, 'No') AS Streaming\_Music,

ISNULL(Unlimited\_Data, 'No') AS Unlimited\_Data,

Contract,

Paperless\_Billing,

Payment\_Method,

Monthly\_Charge,

Total\_Charges,

Total\_Refunds,

Total\_Extra\_Data\_Charges,

Total\_Long\_Distance\_Charges,

Total\_Revenue,

Customer\_Status,

ISNULL(Churn\_Category, 'Others') AS Churn\_Category,

ISNULL(Churn\_Reason , 'Others') AS Churn\_Reason

INTO [db\_Churn].[dbo].[prod\_Churn]

FROM [db\_Churn].[dbo].[stg\_Churn];

-- All where we found NULL value in the previous query, we would replace with "None", "No","Others"

-- Now inserting the new query into a new table called prod\_Churn (Production Table)

-- CREATING VIEW FOR POWER BI

CREATE VIEW vw\_ChurnData as

SELECT

\*

FROM prod\_Churn

WHERE Customer\_Status In ('Churned', 'Stayed');

-- view one

CREATE VIEW vw\_JoinData as

SELECT

\*

FROM prod\_Churn

WHERE Customer\_Status In ('Joined');

-- view two

Power BI

We get data from SSMS, impit our server name , we also use import (not direct query)

We take the prod\_Churn table first and start transforming with power query,

We would create two extra columns called Churn Status (based on customer\_status column) and Monthly Charge Range (based on Monthly\_charge column), the former is basically hard coding the texts to numbers because we would need some calculations to be done, while the later is to reduce the cardinality a bit as visualizing the monthly charge will be cumbersome.

After loading data, we now create some measures

Total customers = COUNT(prod\_Churn[Customer\_ID])

Total Churn = SUM(prod\_Churn[Churn Status])

New Joiners = CALCULATE(COUNT(prod\_Churn[Customer\_ID]),prod\_Churn[Customer\_Status] = "Joined")

Churn Rate = [Total Churn] / [Total customers]

Now we create our charts

We use the card visuals for Total customers, total churn, New joiners and Join rate

Then we move to demographics and create a donut chart for gender. As we want to create the a visual for the age, we see that we have a lot of ages in the data, so just like the Monthly\_charge earlier we create an age range. So back to power query, we create another table from referencing the prod\_Churn table, we keep only the age column in this new table (which we have named mapping\_AgeGrp), we remove duplicates from the age, then we create a new column for the ranges. After this we link the age column in both tables in the modelling tab.

We create the total customer and churn rate by Age rage with the line and stacked column charge, we create the contract and payment method by churn rate with bar charts.

As we want to create for tenure, we face the same issues as the age column, so we create similar solutions as above.

Our tenure and Age charts have are not sorting as they should, in chronological for tenure and in numerical for Age as the groups are considered as texts, so we create help columns to sort them.

Then we use clustered bar chart for the top 5 state churn rate, total churn by churn category, churn rate by internet type.

Our table includes various services, each represented by its own columns containing 'Yes' or 'No' values. To effectively analyze each service, we need to unpivot these columns and convert them into rows. Now we can use a matrix visual to see all this services and use the churn status column created earlier as the values, now because we unpivoted the rows, the values in our matrix will have a lot of repetitions, so instead of summing the values we use percentage of.

Now lets ad some dropdowns for interactions, lets also add a tooltip page (add a table for the actual churn reasons) for us to be able to see hover on the churn category and see the reason behind each category.

Now its time to beautify our report.

After adjusting the shapes and formats, we want create a nice background with powerpoint. We do this by taking a screenshot of our report while making sure its in 16:9, then we paste that screenshot in powerpoint, we use the report to guide the shapes we want to create. We save powerpoint file as png. We go and replace our cancas background in our power bi report with our newly created customized background.

Now lets analyze our report, we first use the narrative visual to use AI for report summary, these gives us some write up about our report, and even dynamic as we filter our report.

For our own analysis

* Most of the customers leaving are female, with 64% of the total churn. Among the female customers, 496 left for the competition, of which 183 were offered better deals, 180 sought better devices, and 133 desired higher download speeds and more data.
* We are setting a benchmark of 65% for our services. Among the customers who have not subscribed to certain services and are churning, the rates are Device protection plan (69.5%), Online backup (70.7%), Online security (83.3%), and Premium support (82.6%). For the services that customers have subscribed to but still have a high churn rate, the figures are Internet services (93.6%), Phone service (91.8%), and Unlimited data (80.6%). To address these issues, we recommend enhancing device protection plans, boosting online backup and security features, improving premium support offerings, and conducting a thorough review of internet, phone, and unlimited data services to identify and resolve pain points.
* Our report highlights that female customers have a higher churn rate (64.1%) compared to male customers (35.9%), with the highest churn observed in the over 55 age group (33.3%), and the lowest in the 46-55 age group (22.9%); Imo state has the highest churn rate (57.2%), customers using Fiber Optic have the highest churn rate (41.1%), and those without internet service have the lowest (7.8%); the highest churn rate is for customers using Mailed Check (37.8%) and the lowest for those using Credit Card (14.8%); Month-to-Month contracts have the highest churn rate (46.5%) compared to Two-Year contracts (2.7%); customers with a tenure of less than 6 months have a churn rate of 26.4%, while those with 24 months or more have a churn rate of 27.5% and finally, the highest churn category is Competitor (761), with the lowest being Other (174).

Can we predict future churners?

Yes, we can predict future churners using the random forest algorithm, a robust machine learning method that combines multiple decision trees to produce accurate and reliable results. This project involves Python programming in Jupyter Notebook to develop and evaluate the model.

Data Preparation - The prediction data is created from SQL Server views (Churn Data and Join Data) and exported to an Excel file. After importing necessary libraries like numpy, pandas, and scikit-learn, we load the churn data from the workbook and verify proper data loading.

Data Preprocessing - We start by cleaning the data. Columns that are not predictive, such as:

Customer\_ID (an arbitrary identifier that could bias the model)

Churn\_Category and Churn\_Reason (directly related to the target variable)

...are dropped from the dataset.

Next, we encode categorical variables. Machine learning models like random forests work poorly with categorical data, so we convert these columns into numerical values using LabelEncoder from sklearn.preprocessing. The LabelEncoder sorts values alphabetically and assigns numerical labels starting from 0. However, the target column Customer\_Status (Stayed or Churned) is encoded separately to ensure the labels reflect our desired mapping (Stayed = 0 and Churned = 1).

Splitting Data for Training and Testing - We separate the data into features (independent variables) and targets (the dependent variable, Customer\_Status). The dataset is further split into training and testing sets, ensuring we have enough data to train the model and a separate subset to evaluate its performance. This random split ensures robust validation of the model's predictive ability.

Model Training - We use RandomForestClassifier to build the model, setting n\_estimators=100 to create 100 decision trees. The random\_state=42 ensures consistency and reproducibility of results.

Model Evaluation - The model is evaluated using a confusion matrix and a classification report:

Confusion Matrix: [[800, 57], [127, 218]]

800 (True Negative): Correctly predicted customers who stayed.

57 (False Positive): Incorrectly predicted customers churned.

127 (False Negative): Incorrectly predicted customers stayed.

218 (True Positive): Correctly predicted customers churned.

Classification Report: Provides insights into:

Precision (how precise the predictions are for each class).

Recall (ability to identify all relevant instances for each class).

F1-score (balance between precision and recall).

The overall model performs at 84% accuracy, but it performs better for predicting customers who stay (class 0) than those who churn (class 1). This is largely due to an imbalance in the data, where there are more instances of customers staying.

Feature Importance and Fine-Tuning - To improve the model, we analyze feature importance to identify which predictors most influence the target variable. Features with importance values below a threshold (e.g., 0.01) are removed to avoid unnecessary noise in the data. The model is retrained with the refined features, leading to a cleaner and potentially more accurate model.

We use our trained model to make predictions on a new dataset. First, we load the joined data from an Excel file and preserve a copy of the original data before any encoding. This is important because, after prediction, we want to combine the results with the original data to ensure the output retains meaningful columns rather than encoded values (e.g., 1s and 0s).

Next, we save the Customer\_ID column separately, as it is an identifier and will not be used for predictions. Along with Customer\_ID, we remove other columns that were not part of the model's training process and are unnecessary for predictions.

We then encode any categorical variables in the dataset using our previously fine-tuned label encoders to ensure consistency with the model's expectations. Once the data is prepared, we pass it through our model to generate predictions, which are stored in the variable new\_predictions.

To make the predictions interpretable, we create a new column called Customer\_Status\_Predicted in the original dataset and populate it with the predicted values. We filter the data to include only customers predicted as "Churned" (where the predicted value is 1), as this is the focus of the analysis.

Finally, we save the filtered results, including both the original columns and the predictions, to a new file called Predictions.csv.

Lets visualize our prediction, we load the csv file into power bi, create a few aggregation measures