TELCO CUSTOMER CHURN ANALYSIS

End-to-End Data Analytics Portfolio Project

Tools: [**Excel], [MySQL], [Python (Pandas, Matplotlib, Seaborn)], [Power BI]**

Churn Rate	Customers Analysed	Revenue Lost to Churn	Monthly Revenue at Risk
26.6%	***7,043***	***$2,862,926***	***$139,130 / month***

1. THE BUSINESS PROBLEM

Telco, a fictional mid-sized telecommunications company, is losing customers. Despite spending millions acquiring new ones, a significant portion leave within the first year quietly, with no warning. The company approached me to dig into the data and answer three critical questions. Their marketing team was spending heavily to bring new people in, but a significant portion of those customers were quietly cancelling within months of signing up.

The CEO framed the challenge in three questions:

- **Why are our customers leaving?**
- **Who is most likely to churn next?**
- **What can we do to keep them?**

The financial motivation was clear: **it costs 5x more to acquire a new customer than to retain one,** and a 5% reduction in churn would save the business an estimated **$2 million annually**. My job was to turn 7,043 rows of customer data into answers the business could act on.

2. THE DATASET

Source: IBM Telco Customer Churn Dataset via Kaggle

Size: 7,043 customers, 21 columns [7,032 after cleaning]

The dataset covers:

- Customer demographics (gender, age, dependents),

- Account details (tenure, contract type, payment method),

- Services subscribed to (phone, internet, security, backup, streaming),

- Billing amounts, and a churn column indicating whether each customer left.

3. MY APPROACH

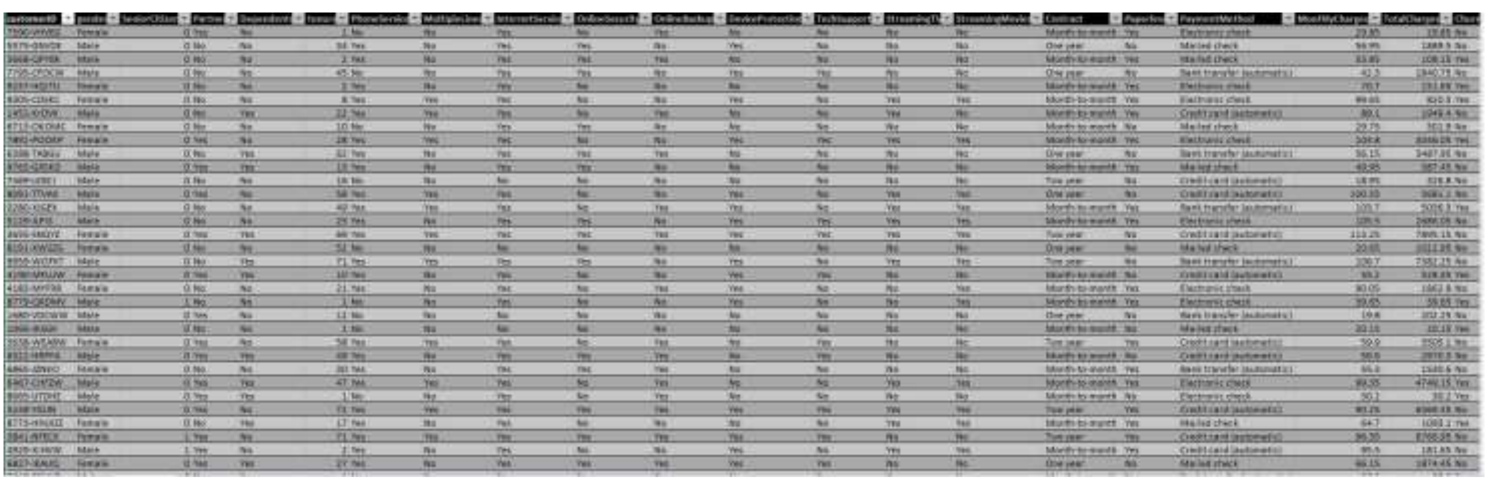
I used a progressive cleaning pipeline   each tool built on the work of the one before it. I never modified the original file; every stage produced a cleaner version that fed into the next.

Raw CSV - Excel (initial clean) - MySQL (deep clean + analysis) - Python (EDA + visualisations) - Power BI (dashboard)

Step 1 - Excel: First Look and Initial Cleaning

Before writing a single query or line of code, I opened the raw CSV in Excel to understand what I was working with. This is where I caught some obvious problems.

- Removed duplicate rows
- Standardised redundant values (seven service columns all contained 'No internet service' as a value, which was just a verbose way of saying 'No.' I replaced all of them to keep the data consistent)
- Dropped unnecessary columns that added noise without adding insight
- Renamed columns to be clearer and more readable
- Saved the cleaned version as the source file for SQL



Step 2 - MySQL: Deep Cleaning and Business Analysis

With the cleaned Excel file imported into MySQL Workbench, I went much deeper. SQL is where I turned raw rows into actual business answers.

Cleaning

- Trimmed whitespace from all text columns to prevent grouping errors in queries
- Used UPDATE statements to catch any remaining inconsistent values
- Fixed a BOM encoding issue that corrupted the customerID column name on import (renamed ï»¿customerID back to customerID.) I'd use Google Sheets or Notepad++ to convert the file before importing next time to avoid this
- Split the tenure column into four customer groups: New (0–5 months), Early (6–11), One-Year (12–23), Loyal (24+)
- TO TURN ON/OFF PREVENTATIVE UNINTENDED DATA UPDATE - When running UPDATE queries to prevent accidental omission of a WHERE clause, which could affect all rows in a table.

```sql
SELECT * FROM telco_datasets;

-- DATA CLEANING
SELECT TRIM(gender), TRIM(MultipleLines), TRIM(InternetService), TRIM(OnlineSecurity), TRIM(OnlineBackup), TRIM(DeviceProtection),
       TRIM(TechSupport), TRIM(StreamingTV), TRIM(StreamingMovies), TRIM(Contract), TRIM(PaymentMethod)
FROM telco_datasets;

SET SQL_SAFE_UPDATES = 0; -- TO TURN OFF PREVENTATIVE UNINTED DATA UPDATE

-- UPDATED COLUMN VALUES
UPDATE telco_datasets
SET
    phoneservice= 'No',
    multiplelines = 'No',
    OnlineSecurity = 'No',
    OnlineBackup = 'No',
    DeviceProtection = 'No',
    TechSupport = 'No',
    StreamingTV = 'No',
    StreamingMovies = 'No'
WHERE OnlineSecurity = 'No internet service';
```

```sql

UPDATE telco_datasets
SET
    multiplelines= 'No'
where multiplelines = 'No phone service';

-- RENAMED COLUMN CAUSED BY BOM
ALTER TABLE telco_datasets -- renamed BOM caused by utf-8 csv inport, would use notepad+++ OR GOOGLE SHEET to change it to normal csv to avoid  BOM
RENAME COLUMN `ï»¿customerID` TO `customerID`;

SELECT * FROM telco_datasets;
SET SQL_SAFE_UPDATES = 1; -- TO TURN ON PREVENTATIVE UNINTENDED DATA UPDATE
```

Business Queries

- Overall churn rate
- Churn rate by contract type
- Churn rate by tenure group
- Revenue analysis - five separate angles: total vs lost, by churn status, by contract type, by payment method, and Monthly Recurring Revenue (MRR) at risk
- Churn rate by service combinations

```sql
-- BUSINESS QUERIES
/*OVERALL CHURN RATE*/
SELECT COUNT(*) AS TOTAL_CUSTOMER,
SUM(CASE WHEN CHURN = 'YES' THEN 1 ELSE 0 END) AS CHURNED_CUSTOMERS,
ROUND((SUM(CASE WHEN CHURN = 'YES' THEN 1 ELSE 0 END) *100.0 / COUNT(*)),1)AS CHURNED_PERCENTAGE
FROM telco_datasets;


-- CHURN BY CONTRACT TYPE
SELECT
    COUNT(*) AS total_customers,

    SUM(CASE WHEN Contract = 'Month-to-month' AND Churn = 'Yes' THEN 1 ELSE 0 END) AS monthly_churned,
    SUM(CASE WHEN Contract = 'One year' AND Churn = 'Yes' THEN 1 ELSE 0 END) AS oneyear_churned,
    SUM(CASE WHEN Contract = 'Two year' AND Churn = 'Yes' THEN 1 ELSE 0 END) AS twoyear_churned,

    ROUND((SUM(CASE WHEN Contract = 'Month-to-month' AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN Contract = 'Month-to-month' THEN 1 ELSE 0 END)), 2) AS monthly_churn_rate,

    ROUND((SUM(CASE WHEN Contract = 'One year' AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN Contract = 'One year' THEN 1 ELSE 0 END)), 2) AS oneyear_churn_rate,


    ROUND((SUM(CASE WHEN Contract = 'Two year' AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN Contract = 'Two year' THEN 1 ELSE 0 END)), 2) AS twoyear_churn_rate

FROM telco_datasets;

-- CHURN BY TENURE GROUPS
-- 0-5, 6-11, 12-23, 24-above[new,early,1year,loyal]
SELECT
    COUNT(*) AS total_customers,

    SUM(CASE WHEN tenure < 6 AND Churn = 'Yes' THEN 1 ELSE 0 END) AS Below_6months_churned,

    SUM(CASE WHEN tenure >= 6 AND tenure < 12 AND Churn = 'Yes' THEN 1 ELSE 0 END) AS Between_6_12months_churned,

    SUM(CASE WHEN tenure >= 12 AND tenure < 24 AND Churn = 'Yes' THEN 1 ELSE 0 END) AS Between_12_24months_churned,

    SUM(CASE WHEN tenure >= 24 AND Churn = 'Yes' THEN 1 ELSE 0 END) AS Above_24months_churned,

    -- Churn rates per group
    ROUND((SUM(CASE WHEN tenure < 6 AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN tenure < 6 THEN 1 ELSE 0 END)), 2) AS Below_6months_churn_rate,
```

```sql
    ROUND((SUM(CASE WHEN tenure >= 6 AND tenure < 12 AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN tenure >= 6 AND tenure < 12 THEN 1 ELSE 0 END)), 2) AS Between_6_12months_churn_rate,

    ROUND((SUM(CASE WHEN tenure >= 12 AND tenure < 24 AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN tenure >= 12 AND tenure < 24 THEN 1 ELSE 0 END)), 2) AS Between_12_24months_churn_rate,

    ROUND((SUM(CASE WHEN tenure >= 24 AND Churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 /
           SUM(CASE WHEN tenure >= 24 THEN 1 ELSE 0 END)), 2) AS Above_24months_churn_rate

FROM telco_datasets;

-- REVENUE ANALYSIS [5 stages]
-- total revenue vs lost revenue
SELECT
round(sum(totalcharges),2) as total_revenue,
round(sum(case when churn = 'yes' then TotalCharges else  0 end),2) as churned_revenue
from telco_datasets;
```

```sql
-- revenue by churn status
select churn,
count(*) as customers,
round(sum(monthlycharges),2) as monthly_revenue,
round(sum(totalcharges),2) as total_revenue
from telco_datasets
group by churn;

-- revenue by contract typec [cant use total_revenue because of monthly subscribers]
SELECT
    Contract,
    ROUND(SUM(MonthlyCharges), 2) AS monthly_revenue,
    ROUND(SUM(CASE WHEN Churn = 'Yes' THEN MonthlyCharges END), 2) AS monthly_revenue_at_risk
FROM telco_datasets
GROUP BY Contract;
```

```sql

-- REVEUE BY PAYMENT METHOD
select
paymentmethod,
round(sum(totalcharges),2) as Total_revenue,
round(sum(case when churn = 'yes' then totalcharges else 0 end),2) as revenue_at_risk,
count(*) as total_customers,
sum(case when churn = 'yes' then 1 else 0 end) as churned_customers,
round((sum(case when churn ='yes' then 1 else 0 end) * 100.0 / count(*)),2) as churn_rate_percent
from telco_datasets
group by PaymentMethod
order by churn_rate_percent desc;


SELECT * FROM telco_datasets;

-- monthly recuring revenue -
select
count(*) as total_customer_charged_monthly,
round(sum(MonthlyCharges),2) as total_monthly_revenue,
round((sum(case when churn = 'yes' then monthlycharges else 0 end )),2) as month_month_revenue_at_risk,
round((sum(case when  churn = 'no' then monthlycharges else 0 end )),2) as month_month_revenue_not_at_risk
from telco_datasets;
```

```sql
-- chuned * 100/ total customer [FOR PERCENTAGES]

/* CHURN RATE BY SPECIFIC SERVICE COMBINATIONS */
SELECT
     internetservice,
     phoneservice,
    multiplelines,
    OnlineBackup, OnlineSecurity, DeviceProtection, TechSupport, StreamingMovies, StreamingMovies,
    COUNT(customerID) AS total_customers,
    ROUND((SUM(CASE WHEN churn = 'Yes' THEN 1 ELSE 0 END) * 100.0 / COUNT(customerID)), 2) AS Churned_percentage
FROM
    telco_datasets
GROUP BY
    phoneservice, multiplelines, internetservice, OnlineBackup, OnlineSecurity, DeviceProtection, TechSupport, StreamingMovies, StreamingMovies
ORDER BY
    Churned_percentage DESC;

select* from telco_datasets;
```

STEP 3 - Python: exploratory Analysis and Visualisations

After validating the key numbers in SQL, I moved into Python for deeper statistical work and visualisations that would make the patterns visible.

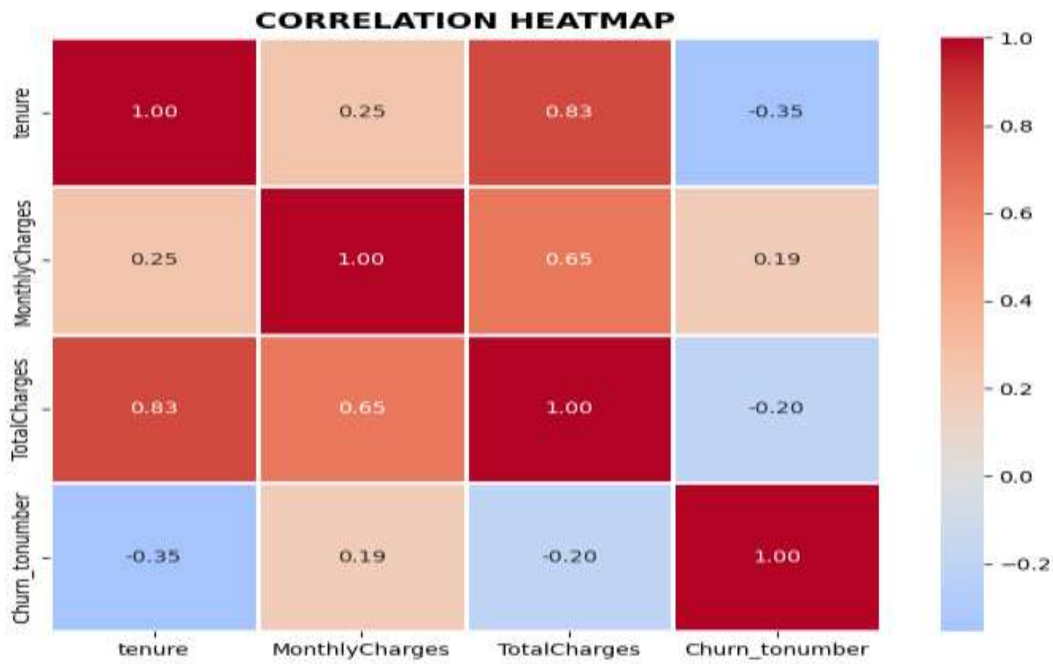
What I built

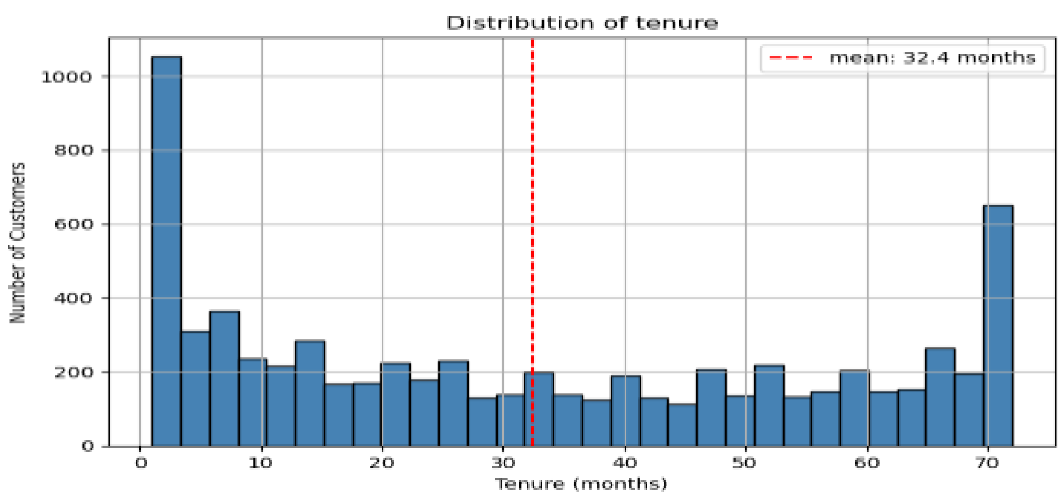
- Customer Lifetime Value (CLV) column: MonthlyCharges × tenure - the total value each customer has delivered
- Total_Service_Used column: converted all service columns from Yes/No to 1/0 and summed them to see how many services each customer uses out of 9 available
- Risk Score model: a custom 0–100 scoring system that classifies every customer as Low, Medium, or High Risk based on five churn indicators
- Correlation matrix to statistically validate the patterns found in SQL
- Distribution analysis on monthly charges and tenure

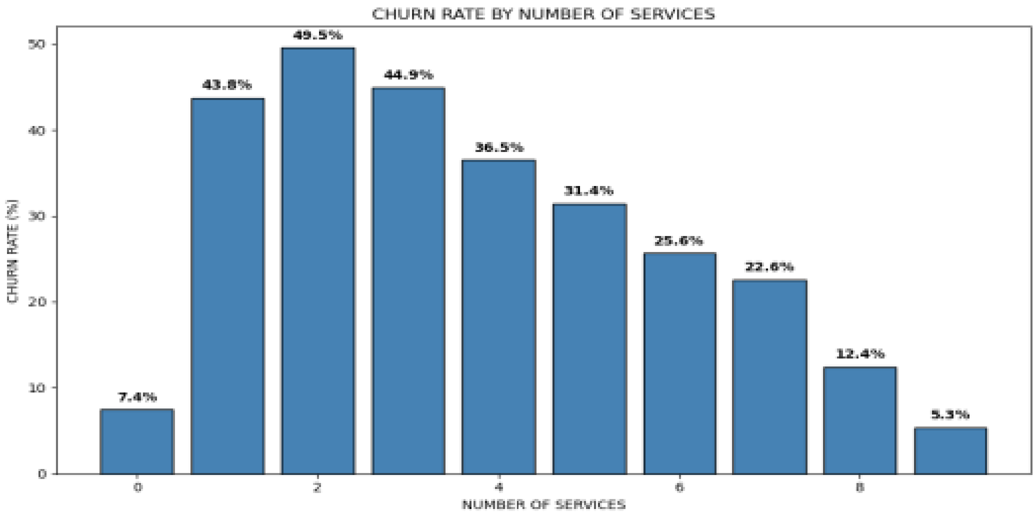
Visualisations

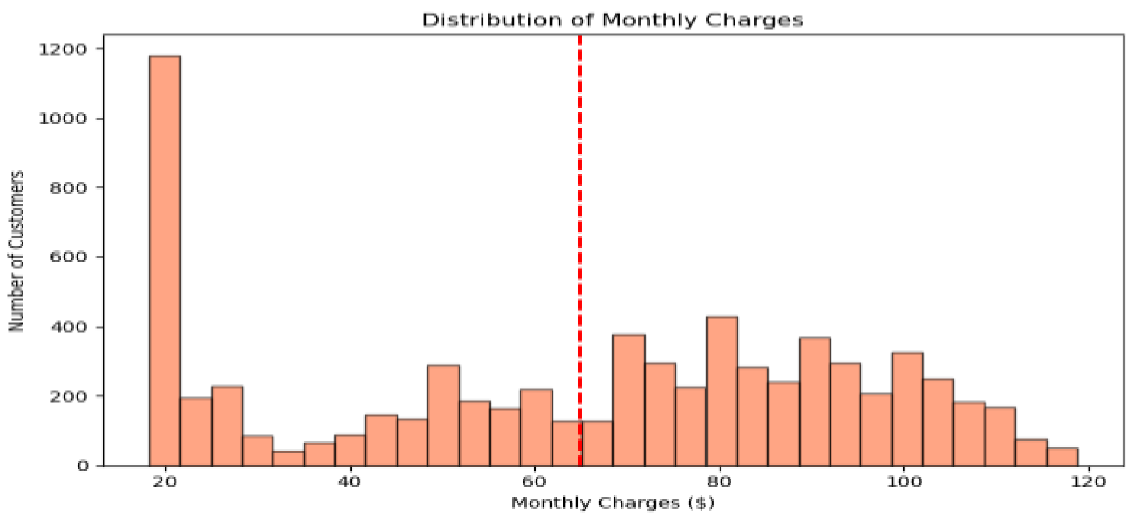
- Correlation heatmap
- tenure - line chart showing the sharp drop as customers stay longer
- Churned vs retained customers - overlapping histogram showing how new customers dominate the churned group
- Distribution of monthly charges

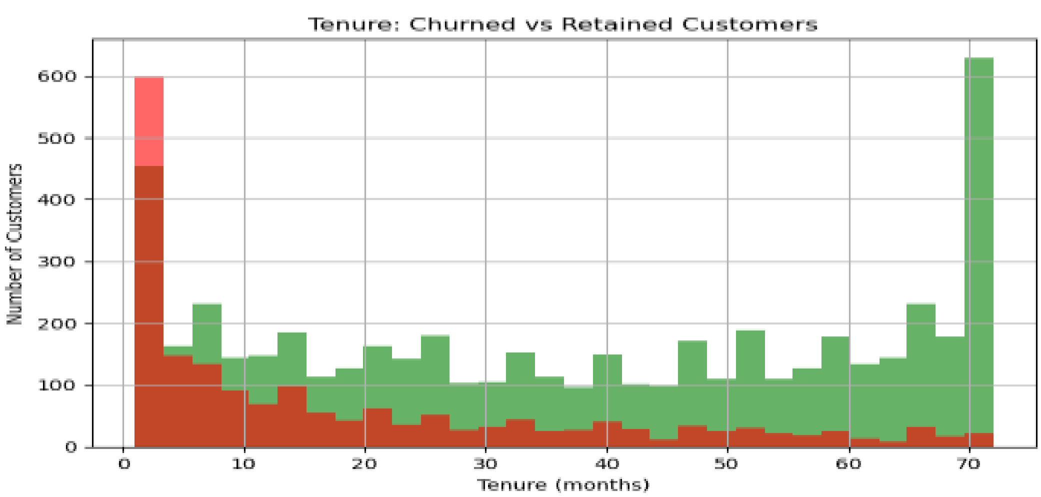
- Churn rate by number of services used - bar chart
- Churn rate by contract type - bar chart

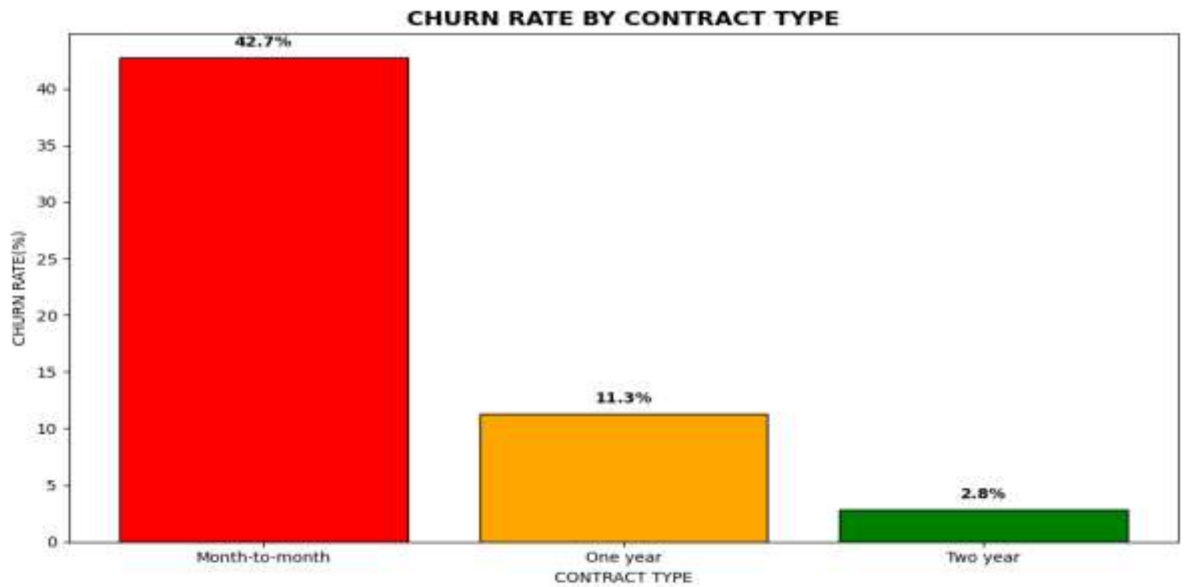


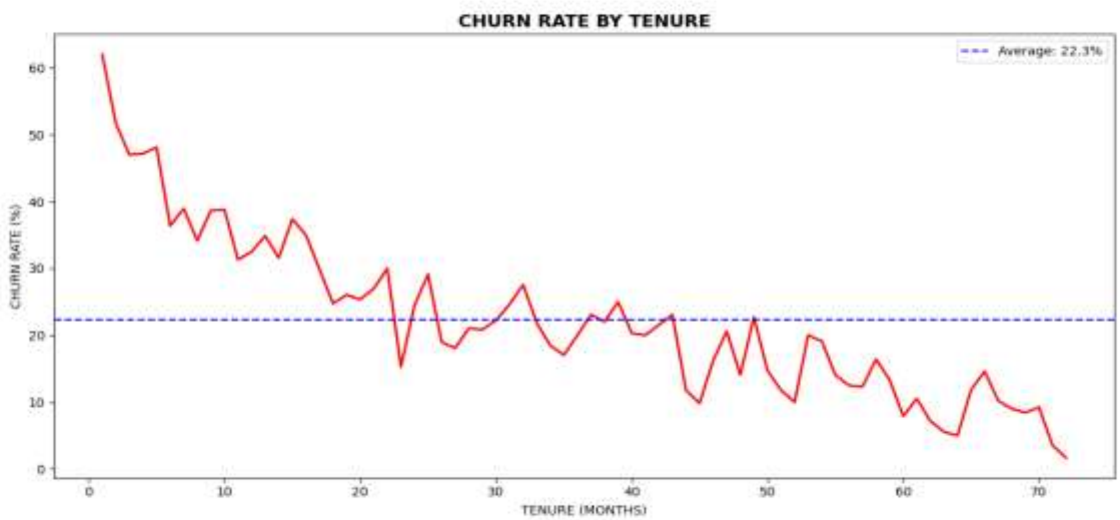



CHURN RATE BY NUMBER OF SERVICES
CHURN RATE (%)
NUMBER OF SERVICES
7.4%
43.8%
49.5%
44.9%
36.5%
31.4%
25.6%
22.6%
12.4%
5.3%


Distribution of Monthly Charges
Number of Customers
Monthly Charges ($)


Tenure: Churned vs Retained Customers
Number of Customers
Tenure (months)



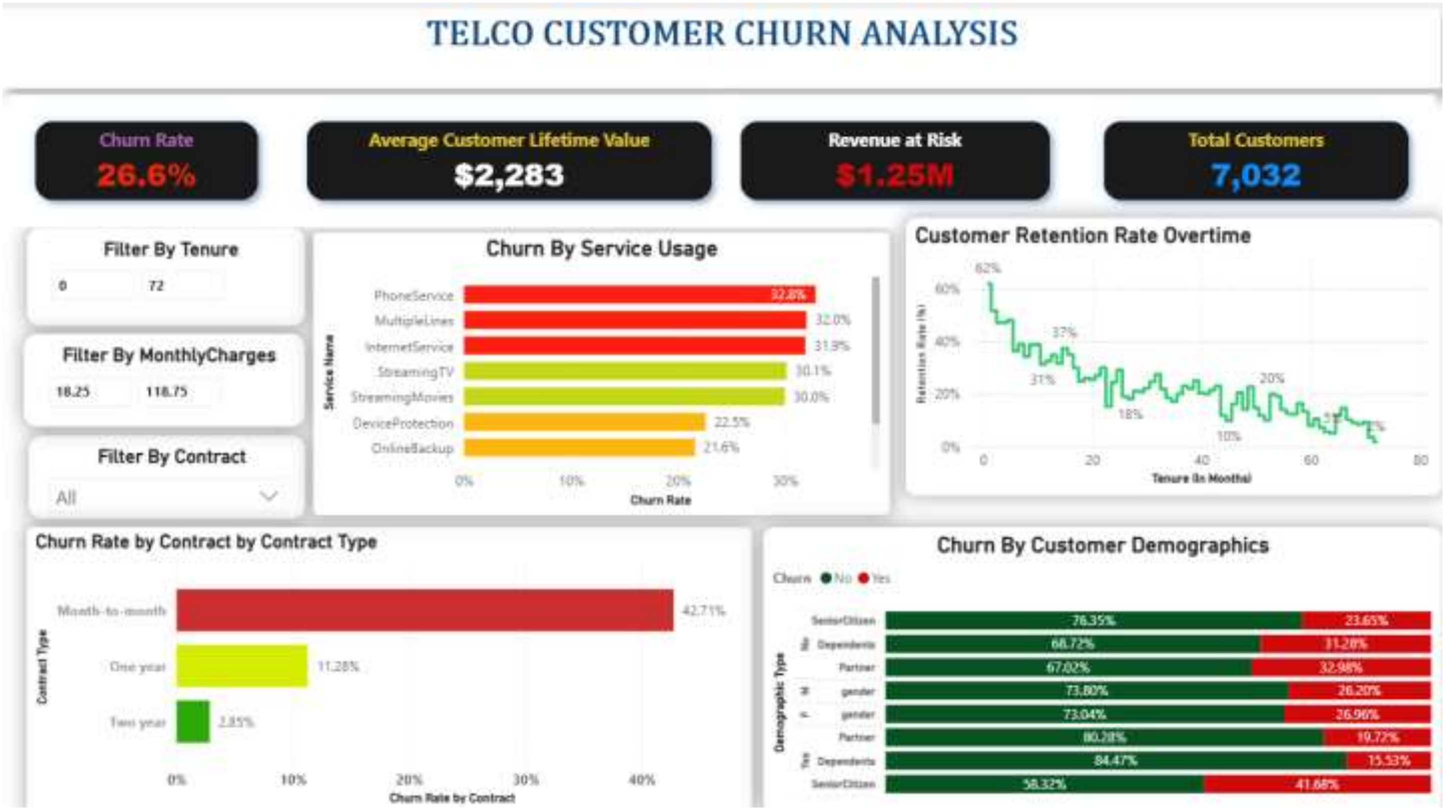


STEP 4 - POWER BI: INTERACTIVE DASHBOARD

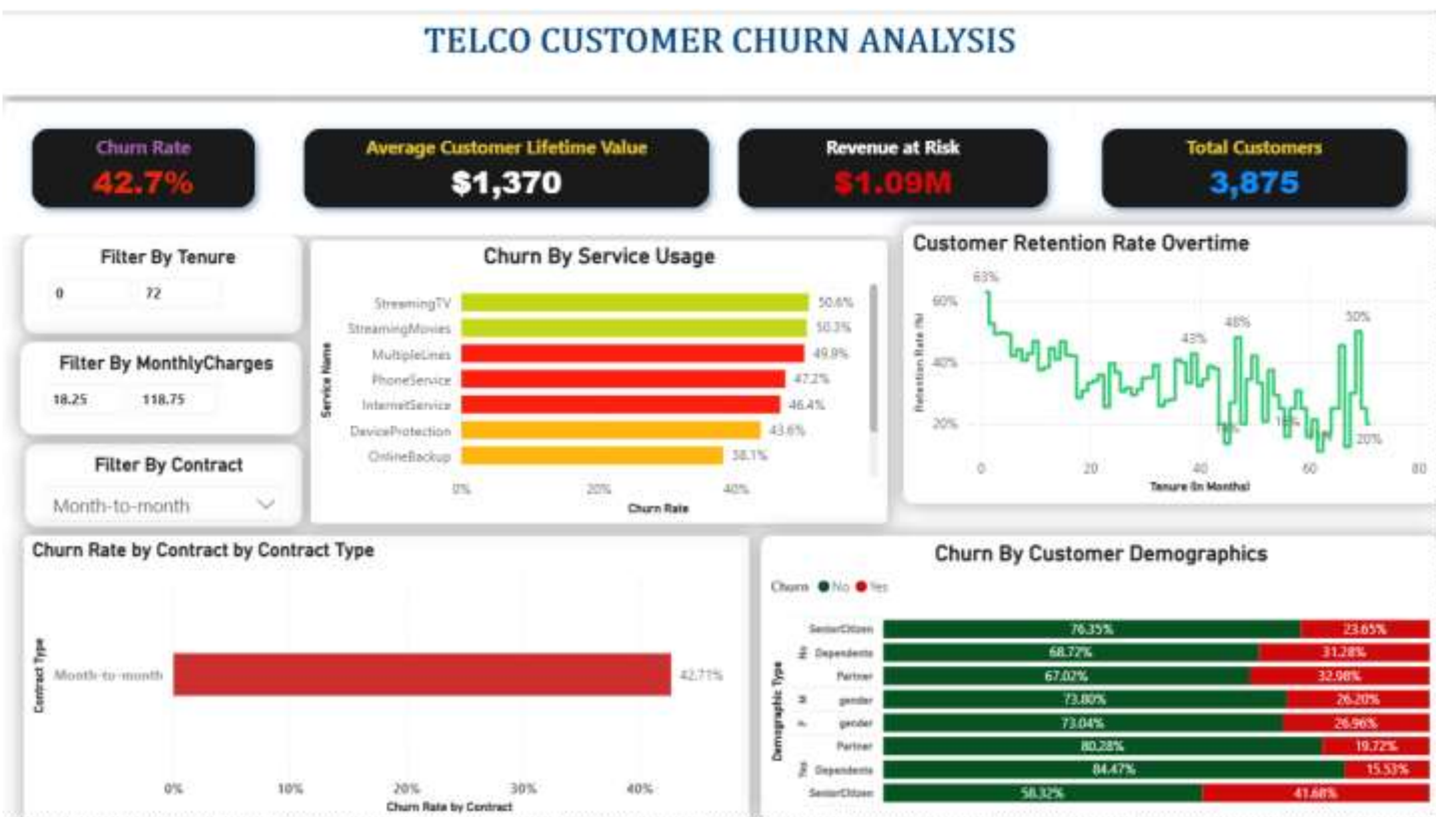
The final step was turning everything into a dashboard a non-technical manager could use. no SQL, no code, just answers on a screen.

- KPI cards: overall churn rate, total customers, revenue at risk, average CLV
- Churn by contract type bar chart
- Retention curve showing how churn rate falls as tenure increases
- Service engagement breakdown
- Slicers for contract type, tenure range, and monthly charge range

I used a reference table to unpivot the service columns rather than modifying the original dataset, this keeps the source data intact and is considered best practice in Power BI modelling.



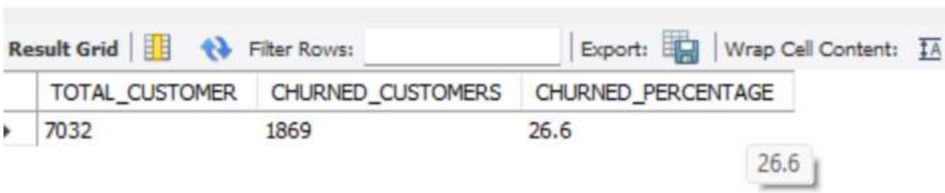
POWER BI - DASHBOARD FILTERED TO MONTH-TO-MONTH CUSTOMERS



4. KEY FINDINGS

Finding 1 - The churn rate is worse than the industry average

The company's churn rate is 26.6% , meaning roughly 1 in 4 customers is leaving. The telecom industry benchmark sits at 15–20%. That gap is not a minor inefficiency. Of 7,043 customers, 1,869 have already churned, taking $2.86 million in lifetime revenue with them. The business is losing customers at a rate it cannot sustain without addressing the root causes.



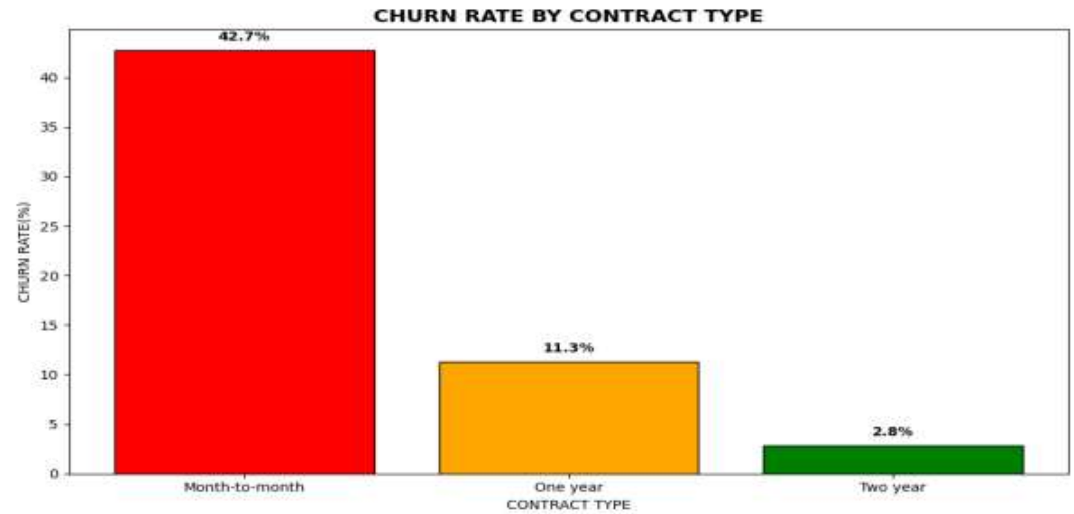
TOTAL_CUSTOMER	CHURNED_CUSTOMERS	CHURNED_PERCENTAGE
7032	1869	26.6

Finding 2 - Contract type is the single biggest churn driver

This was the most striking finding in the entire analysis. The difference between contract types is not gradual but dramatic.

Contract Type	Churn Rate	Interpretation
Month-to-Month	42.71%	Nearly half of these customers will leave
One Year	11.28%	Committed customers - manageable risk
Two Year	2.85%	Almost zero churn - these are loyal customers

A month-to-month customer is 15x more likely to churn than someone on a two-year contract. The contract is not just a billing arrangement . it is the strongest predictor of whether a customer stays or goes. With no long-term commitment, there is no barrier to leaving the moment a competitor makes a better offer.

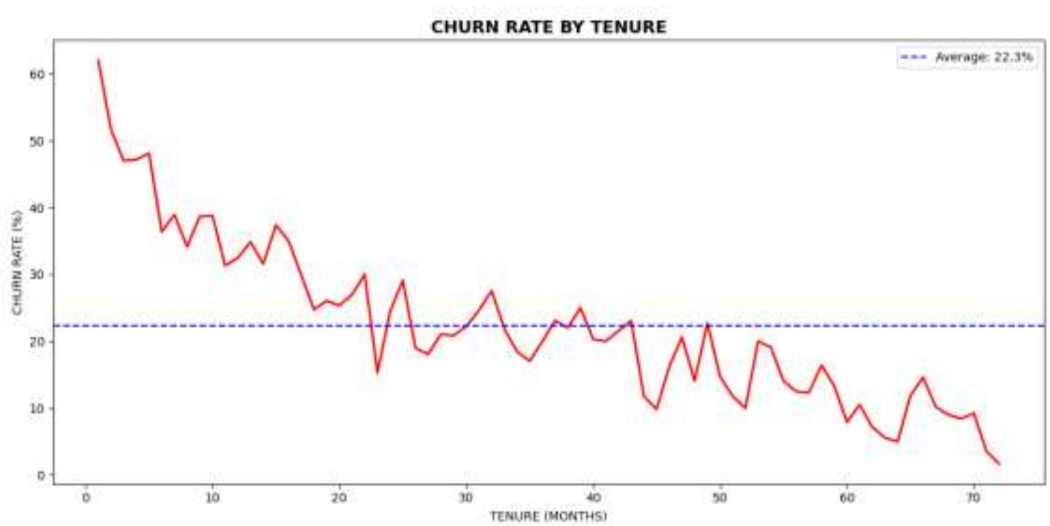


Finding 3 - New customers are leaving before they have settled in

Breaking churn down by tenure group revealed the clearest pattern in the data.

Tenure Group	Period	Churn Rate
New	0–5 months	54.71%
Early	6–11 months	36.53%
One-Year	12–23 months	29.51%
Loyal	24+ months	14.29%

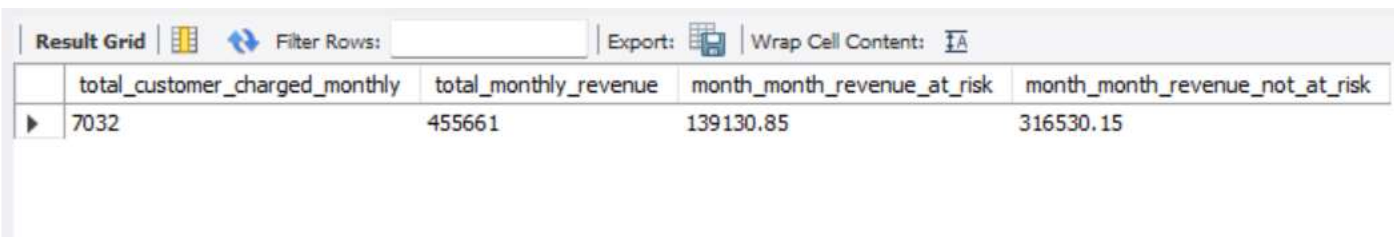
More than half of all new customers leave within the first five months. They never stayed long enough to see the full value of the service. The positive side of this finding is that if the company can successfully engage customers through the first six months, retention improves significantly. The first 90 days are make or break.



Finding 4 - Revenue is bleeding every month

Metric	Amount
Total revenue (all customers)	$16,056,168
Revenue lost to churned customers	$2,862,926
Revenue retained (active customers)	$13,193,241
Monthly revenue currently at risk	$139,130 per month

Churned customers were contributing $139,130 every month before they cancelled. That recurring gap needs to be replaced just to stand still, before any growth target is considered. Annualised, that is over $1.67 million in recurring revenue the business has lost access to.


Result Grid | Filter Rows: | Export: | Wrap Cell Content:

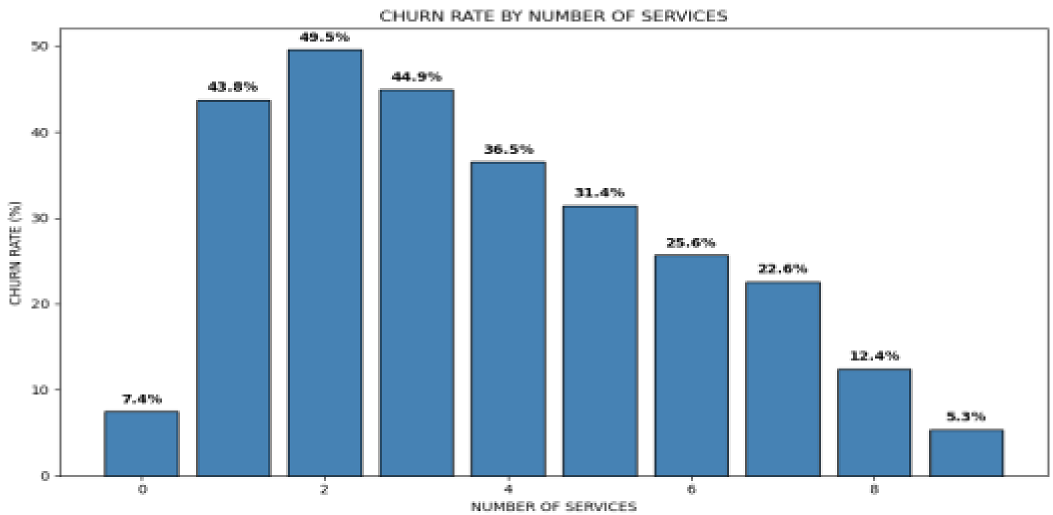
total_customer_charged_monthly	total_monthly_revenue	month_month_revenue_at_risk	month_month_revenue_not_at_risk
7032	455661	139130.85	316530.15

Finding 5 - The more services a customer uses, the less likely they are to leave

I created a calculated column in Python counting how many of the nine available services each customer subscribed to. The pattern was consistent and clear, churn falls steadily as service count rises. Loyalty sign which assures longer tenure, the more services they use the more committed they get.

Services Used	Approximate Churn Rate
1–2 services	~49%
3–4 services	~37–44%
5–6 services	~25–31%
7–9 services	~5–22%

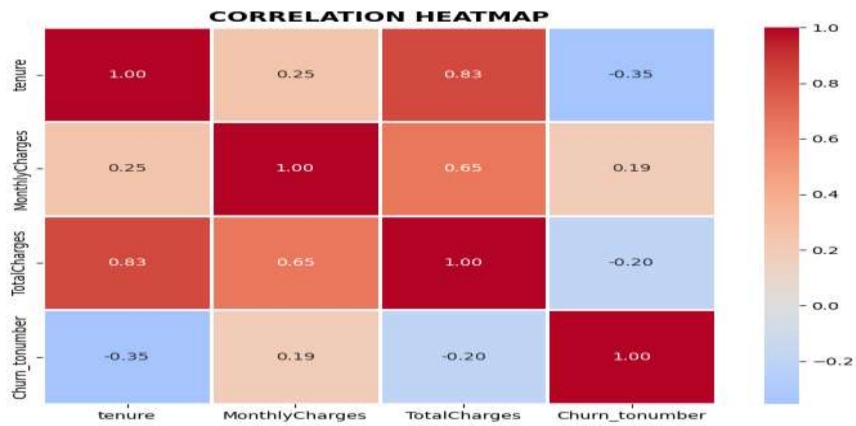
This makes intuitive sense. Every additional service a customer subscribes to increases switching costs. Leaving means losing multiple things at once, not just one. The data makes a clear case that bundling is not just a sales strategy. it is a retention strategy.



Finding 6 - Correlation analysis confirms the patterns statistically

Variable	Correlation with Churn	What It Means
Tenure	-0.35	Strongest predictor, longer tenure means significantly less churn
MonthlyCharges	+0.19	Higher bills slightly increase churn risk
TotalCharges	-0.20	High total spend reflects long tenure, not high risk

Tenure is the strongest single variable correlated with churn. The longer someone has been a customer, the more embedded they are and the less likely they are to leave. Monthly charges show a mild positive correlation. customers on higher plans may feel more price-sensitive, particularly those on flexible month-to-month contracts.



Finding 7 - Risk scoring model flags at-risk customers before they cancel

I built a risk scoring system in Python that assigns each customer a score from 0 to 100 based on five churn indicators, then groups them into Low, Medium, or High Risk. To better understand the likelihood of customer churn.

Risk Factor	Points	Reasoning
Tenure under 12 months	+30	New customers are the highest-risk group
Month-to-month contract	+25	No commitment means no barrier to leaving
Monthly charges over $70	+20	Higher bills increase price sensitivity
Fewer than 3 services used	+15	Low engagement means low switching cost
No tech support	+10	Unresolved issues push customers away

Risk Category	Customer Count	Churn Rate
Low Risk (0–39)	2,599	17.63%
Medium Risk (40–69)	4,402	49.33%
High Risk (70–100)	31	61.11%

The High Risk group is small but urgent, 31 customers flagged by the model as the most likely to leave. These are the customers the retention team should prioritize contacting this week.

```
## RISK SCORE ANALYSIS/DISTRIBUTION

dataset['Risk_score'] = 0
dataset.loc[dataset['tenure'] < 12, 'Risk_score'] += 30
dataset.loc[dataset['Contract'] == 'Month-to-Month', 'Risk_score'] += 25
dataset.loc[dataset['MonthlyCharges'] > 70, 'Risk_score'] += 20
dataset.loc[dataset['Total_Service_Used'] <3, 'Risk_score'] +=15
dataset.loc[dataset['TechSupport'] == 0, 'Risk_score'] += 10



dataset['Risk_Category'] = 'Low Risk'
dataset.loc[dataset['Risk_score'] >= 40, 'Risk_Category'] = 'Medium Risk'
dataset.loc[dataset['Risk_score'] >= 70, 'Risk_Category'] = 'High Risk'
```

```
print(dataset[['Risk_score', 'Risk_Category', 'Churn']].head(20))
```

```
   Risk_score Risk_Category Churn
0          55   Medium Risk    No
1          10      Low Risk    No
2          40   Medium Risk   Yes
3           0      Low Risk    No
4          75     High Risk   Yes
5          60   Medium Risk   Yes
6          30      Low Risk    No
```

```
dataset.groupby('Risk_Category')['Churn'].apply(lambda x: (x == 'Yes').sum() / len(x) * 100).round(2)
```

```
Risk_Category
High Risk      61.11
Low Risk       17.63
Medium Risk    49.33
Name: Churn, dtype: float64
```

5. BUSINESS RECOMMENDATIONS

Every recommendation below is tied directly to a finding in the data. These are not generic suggestions but rather they are responses to specific patterns found in 7,032 real customer records.

1. *Fix the onboarding experience*

Over half of new customers leave within five months. They never got the chance to experience the product's full value. I would recommend a structured 90-day onboarding programme. check-ins at days 7, 30, and 90, a personalised welcome guide in week one, and automated alerts when a new customer has not engaged with more than two services after 30 days.

2. *Incentivise contract upgrades*

Month-to-month customers churn at 42.71%. Two-year customers churn at 2.85%. The gap is enormous. Offering a meaningful discount around 15% to month-to-month customers who upgrade to an annual contract would directly address the biggest driver of churn. Even converting 20% of month-to-month customers to annual contracts could save over $500,000 in annual recurring revenue.

3. Drive service bundle adoption

Moving a customer from two services to four reduces their churn probability by roughly 15 percentage points based on the data. I would recommend discounted bundle packages, training customer service agents to mention one additional service on every inbound call, and a 30-day free trial for Online Security and Tech Support specifically the two add-ons most correlated with retention.

4. Move electronic check users onto auto-pay

Electronic check users showed the highest churn rate of any payment method. Offering a small monthly discount around $5 to switch to bank transfer or credit card auto-pay reduces billing friction and makes cancellation a more deliberate decision rather than an automatic one.

5. Use the risk score model to prioritise retention spend

Rather than contacting all customers reactively, the retention team should export the High and Medium Risk segments from the dashboard each week. The 31 High Risk customers should receive a personal call with a loyalty offer. The 4,402 Medium Risk customers should receive an automated personalised email monthly. The goal is to reach people before they cancel, not after.

6. TOOLS AND SKILLS

Tool	How I Used It	Key Techniques
Excel	Initial cleaning and exploration	Duplicate removal, value standardisation, column formatting
MySQL	Data cleaning and business queries	UPDATE, CASE WHEN, GROUP BY, SUM, COUNT, ROUND, tenure segmentation
Python [Pandas]	Feature engineering and statistics	Calculated columns (CLV, Total Services, Risk Score), correlation matrix
Python [Matplotlib / Seaborn]	Visualisations	Line charts, bar charts, histograms, heatmaps
Power BI	Interactive dashboard	DAX measures, slicers, KPI cards, reference table for unpivoting
GitHub	Documentation and version control	README, project folder structure

7. CONCLUSION

This project started with a simple but urgent business question, why are customers leaving, and what can we do about it? Working through four tools across the full analytics pipeline, the data gave clear answers. The churn problem is real, it is measurable, and it is fixable. Contract type, tenure, and service engagement are not just interesting patterns , they are levers the business can pull. Locking more customers into annual contracts, getting new customers through the critical first 90 days, and encouraging bundle adoption are not expensive interventions. But based on the numbers in this analysis, they could realistically recover hundreds of thousands in recurring revenue annually and bring the churn rate down from 26.6% closer to the industry benchmark of 15–20%. The goal of this analysis was never just to describe the problem, it was to give the business something it could act on tomorrow morning. I believe this does that.

This project is part of my data analytics portfolio. Built end-to-end to demonstrate real-world problem solving from raw data to business recommendations.