

# Telco Customer Analytics Project – Step-by-Step Analysis Report

---

## 1. Objective

The goal of this project was to **analyze Telco customer data** to:

1. Understand **customer churn patterns**
  2. Identify **revenue trends**
  3. Pinpoint **cross-selling opportunities**
  4. Build an **interactive dashboard** for stakeholders with KPIs, charts, and actionable insights
- 

## 2. Data Loading and Exploration

### Steps Taken:

- Loaded the cleaned dataset using pandas:

```
import pandas as pd
```

```
df = pd.read_csv("cleaned_data.csv")
```

- Explored dataset structure:

```
df.head()
```

```
df.info()
```

```
df.describe()
```

- Checked for missing values and data types to ensure quality.

### Key Observations:

- Column `churn` had inconsistent capitalization (yes/no) → normalized later.
  - Columns `tenure` and `monthlycharges` are numeric, suitable for revenue analysis.
  - Contract types: month-to-month, one-year, two-year.
- 

## 3. Data Cleaning

### **Steps Taken:**

- Standardized churn column to "Yes" or "No":

```
df['churn'] = df['churn'].astype(str).str.strip().str.capitalize()
```

- Ensured no missing or invalid values in key columns used for KPIs and charts.

### **Reasoning:**

Normalization avoids errors in visualizations and crosstab operations.

---

## **4. Key Performance Indicators (KPIs)**

### **KPIs Defined:**

1. **Total Revenue:** Sum of monthly charges × tenure
2. **Churn Rate:** Percentage of customers with churn = "Yes"
3. **Total Customers:** Number of records in the filtered dataset

### **Calculations:**

```
total_revenue = (df["monthlycharges"] * df["tenure"]).sum()
```

```
churn_rate = df["churn"].value_counts(normalize=True).get("Yes", 0) * 100
```

```
total_customers = len(df)
```

### **Insights from KPIs:**

- Total revenue grows with high-tenure customers.
  - Churn is higher among month-to-month contracts.
  - Identified key segments for retention and upselling.
- 

## **5. Visual Analysis**

### **5.1 Churn Distribution**

- Counted customers by churn status and plotted a bar chart:

```
churn_counts = df["churn"].value_counts().reset_index()
```

**Insight:** Month-to-month customers contribute most to churn.

---

## 5.2 Churn by Contract Type

- Created crosstab between contract and churn:

```
contract_churn = pd.crosstab(df["contract"], df["churn"]).reset_index()
```

- Melted to long format for Plotly plotting.

### Insight:

- Long-term contracts have significantly lower churn.
  - Month-to-month contracts are high-risk.
- 

## 5.3 Average Monthly Charges by Tenure

- Grouped by tenure to find average charges:

```
tenure_avg = df.groupby("tenure")["monthlycharges"].mean().reset_index()
```

**Insight:** Revenue grows with tenure; longer-tenure customers are more valuable.

---

## 5.4 Monthly Charges Distribution

- Histogram of monthlycharges:

```
px.histogram(df, x="monthlycharges", nbins=20)
```

**Insight:** Most customers pay mid-range charges; a few high-paying customers are prime for upselling.

---

## 5.5 Cross-Selling Opportunities

- Scatter plot of tenure vs monthlycharges:

```
px.scatter(df, x="tenure", y="monthlycharges", hover_data=["contract","churn"])
```

**Insight:** Customers with **high tenure and high monthly charges** are ideal for cross-selling.

---

## 6. Interactive Dashboard

**Technology:** Dash + Plotly (Dash 3.x)

**Features Implemented:**

1. **Dynamic KPIs** – Total Revenue, Churn Rate, Total Customers
2. **Charts** – Churn distribution, churn by contract, revenue by tenure, histogram, scatter
3. **Interactive Filters:**
  - o Contract type dropdown
  - o Tenure range slider
4. **Insights Panel** – Key business insights
5. **Modern styling** – Colors, shadows, responsive grid layout

**Key Code Patterns:**

```
@app.callback(  
    Output(...),  
    Input("contract-filter", "value"),  
    Input("tenure-filter", "value"))  
)  
  
def update_dashboard(contract_value, tenure_range):  
    df = df.copy()  
  
    if contract_value != "All":  
        df = df[df["contract"] == contract_value]  
  
    df = df[(df["tenure"] >= tenure_range[0]) & (df["tenure"] <= tenure_range[1])]  
  
    ...  
  
    return kpis, figures
```

**Outcome:**

Dashboard is fully interactive, allowing stakeholders to explore different segments in real-time.

---

## 7. Business Recommendations

- 1. Reduce Churn Among Month-to-Month Customers**
    - Incentivize contract upgrades, loyalty programs, and retention campaigns.
  - 2. Target High-Value Customers for Cross-Selling**
    - Identify high-tenure, high-spending customers for premium service offers.
  - 3. Revenue Optimization**
    - Use insights from monthly charges trends to design bundled services and promotions.
  - 4. Segmented Marketing**
    - Use contract type, tenure, and monthly charges for targeted campaigns.
  - 5. Monitoring and Reporting**
    - Keep the dashboard as a live tool for executives to monitor KPIs and churn trends.
- 

## **8. Technical Notes**

- **Python Libraries Used:**
  - pandas – Data manipulation
  - dash – Interactive dashboard
  - plotly – Charts and visualizations
- **Environment Setup:**
  - requirements.txt:

pandas>=2.0.0

dash>=3.0.0

plotly>=6.0.0

- **Dashboard Execution:**

python Dashboard\_Creation.py

- Open browser: <http://127.0.0.1:8050/>
-

## 9. Conclusion

The analysis revealed that **contract type and tenure are the main drivers of churn and revenue.**

- Month-to-month contracts are **high-risk for churn**
- Long-tenure and high-paying customers represent **high-value segments** for cross-selling and retention programs

The dashboard provides **actionable insights** in real-time, enabling data-driven business strategies that:

1. Reduce churn
2. Maximize revenue
3. Optimize marketing and retention campaigns