Telco Customer Churn – EDA and Business Understanding

This notebook performs exploratory data analysis (EDA) for a telecom customer churn dataset. The goal is to understand the business problem, explore the data, and identify key patterns that influence whether a customer leaves the company (churns) or stays.

Business Problem

In the telecom industry, customers can switch providers easily, and losing existing customers (customer churn) directly reduces recurring revenue. Acquiring a new customer is typically more expensive than retaining an existing one, so telecom operators focus heavily on predicting and reducing churn. [web:47][web:49]

The objective of this project is to build a data-driven churn prediction system that:

- Identifies customers who are likely to churn in the near future.
- Highlights key drivers of churn so the business can design targeted retention campaigns.
- Provides interpretable insights that can guide pricing, service quality, and contract strategies.

1. Import Libtaries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Load raw data

```
df = pd.read_csv("Telco-Customer-Churn-dataset.csv")
df.head()
df.info()
df.describe(include="all")
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   customerID        7043 non-null   object
 1   gender            7043 non-null   object
 2   SeniorCitizen     7043 non-null   int64
 3   Partner           7043 non-null   object
 4   Dependents        7043 non-null   object
 5   tenure            7043 non-null   int64
 6   PhoneService      7043 non-null   object
 7   MultipleLines     7043 non-null   object
 8   InternetService   7043 non-null   object
 9   OnlineSecurity    7043 non-null   object
 10  OnlineBackup      7043 non-null   object
 11  DeviceProtection  7043 non-null   object
 12  TechSupport       7043 non-null   object
 13  StreamingTV       7043 non-null   object
 14  StreamingMovies   7043 non-null   object
 15  Contract          7043 non-null   object
 16  PaperlessBilling  7043 non-null   object
 17  PaymentMethod     7043 non-null   object
 18  MonthlyCharges    7043 non-null   float64
 19  TotalCharges      7043 non-null   object
 20  Churn             7043 non-null   object
dtypes: float64(1), int64(2), object(18)
memory usage: 1.1+ MB
```

2. Dataset Description

The dataset contains customer-level information from a fictional telecom company that provides phone and internet services. Each row represents one customer and includes: [web:35][web:57]

- **Customer demographics** – e.g., gender, senior citizen flag, partner, dependents.
- **Account information** – e.g., tenure (months with the company), contract type, payment method, paperless billing.
- **Services subscribed** – e.g., phone service, multiple lines, internet service type, online security, tech support, streaming services.
- **Billing information** – e.g., monthly charges, total charges.
- **Target variable:** `Churn` – indicates whether the customer left the company in the last month (`Yes`) or not (`No`). [web:35][web:57]

In this notebook, the focus is on understanding how these features relate to churn, not on building models yet.

3. Basic cleaning (TotalCharges)

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...
count	7043	7043	7043.000000	7043	7043	7043.000000	7043	7043	7043	7043	
unique	[illegible]	[illegible]	NaN	2	2	NaN	2	3	3	3	
top	3186-AJIEK	Male	NaN	No	No	NaN	Yes	No	Fiber optic	No	
freq	[illegible]	[illegible]	NaN	3641	4933	NaN	6361	3390	3096	3498	
mean	[illegible]	[illegible]	[illegible]	NaN	NaN	32.371149	NaN	NaN	NaN	NaN	
std	NaN	NaN	[illegible]	NaN	NaN	[illegible]	NaN	NaN	NaN	NaN	
min	NaN	NaN	0.000000	NaN	NaN	0.000000	NaN	NaN	NaN	NaN	

11 rows × 21 columns

Data Cleaning (TotalCharges Fix)

During inspection, the `TotalCharges` column appears as an object (string) type due to spaces in some rows. This prevents correct numeric analysis. To address this, the notebook:

- Replaces blank spaces in `TotalCharges` with `NaN`.
- Converts `TotalCharges` to a numeric type.
- Drops rows where `TotalCharges` is missing after conversion.

This ensures that billing information can be used correctly in later analysis and modeling.

```
df["TotalCharges"] = df["TotalCharges"].replace(" ", np.nan)
df["TotalCharges"] = pd.to_numeric(df["TotalCharges"])
df = df.dropna(subset=["TotalCharges"])
df = df.reset_index(drop=True)

df.info()
df.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7032 entries, 0 to 7031
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7032 non-null   object 
 1   gender            7032 non-null   object 
 2   SeniorCitizen     7032 non-null   int64  
 3   Partner           7032 non-null   object 
 4   Dependents        7032 non-null   object 
 5   tenure            7032 non-null   int64  
 6   PhoneService      7032 non-null   object 
 7   MultipleLines     7032 non-null   object 
 8   InternetService   7032 non-null   object 
 9   OnlineSecurity    7032 non-null   object 
 10  OnlineBackup      7032 non-null   object 
 11  DeviceProtection  7032 non-null   object 
 12  TechSupport       7032 non-null   object 
 13  StreamingTV       7032 non-null   object 
 14  StreamingMovies   7032 non-null   object 
 15  Contract          7032 non-null   object 
 16  PaperlessBilling  7032 non-null   object 
 17  PaymentMethod     7032 non-null   object 
 18  MonthlyCharges    7032 non-null   float64
 19  TotalCharges      7032 non-null   float64
 20  Churn             7032 non-null   object 
dtypes: float64(2), int64(2), object(17)
memory usage: 1.1+ MB
```

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges
count	7032.000000	7032.000000	7032.000000	7032.000000
mean	0.162400	32.421786	64.798208	2283.300441
std	0.368844	24.545260	30.085974	2266.771362
min	0.000000	1.000000	18.250000	18.800000
25%	0.000000	9.000000	35.587500	401.450000
50%	0.000000	29.000000	70.350000	1397.475000
75%	0.000000	55.000000	89.862500	3794.737500
max	1.000000	72.000000	118.750000	8684.800000

4. Target distribution

Target Variable: Churn Distribution

Understanding the distribution of the target variable (`Churn`) is essential: [web:25][web:52]

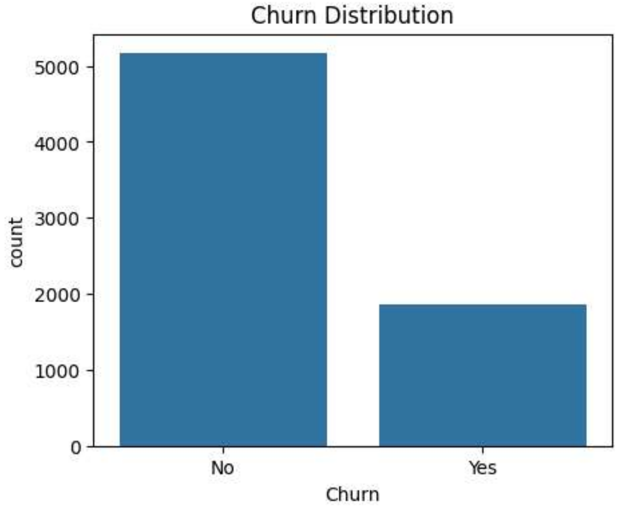
- A **countplot** shows how many customers churned vs. did not churn.
- A **percentage view** (pie chart) highlights class imbalance.

In many telecom datasets, churners are a minority compared to non-churners. This imbalance affects model choice and evaluation, because accuracy alone can be misleading if most customers do not churn.

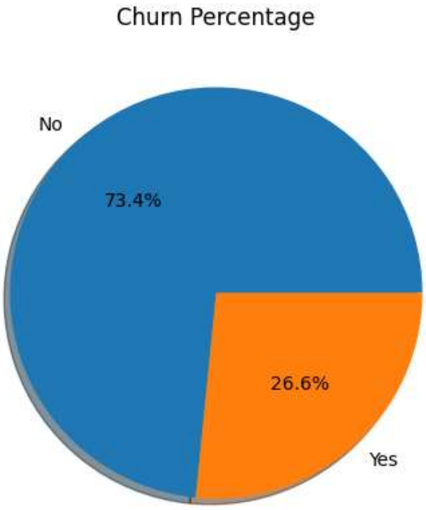
```
plt.figure(figsize=(5, 4))
sns.countplot(x="Churn", data=df)
plt.title("Churn Distribution")
plt.show()

print(df["Churn"].value_counts(normalize=True) * 100)

plt.figure(figsize=(5, 5))
df["Churn"].value_counts().plot.pie(
    autopct="%1.1f%%", labels=["No", "Yes"], shadow=True
)
plt.title("Churn Percentage")
plt.ylabel("")
plt.show()
```




```
Churn
No     73.421502
Yes    26.578498
Name: proportion, dtype: float64
```



5. Numerical features – distributions

Numerical Features – Univariate Analysis

Key numeric variables in this dataset include:

- `tenure` – number of months the customer has stayed with the company.
- `MonthlyCharges` – amount charged to the customer each month.
- `TotalCharges` – total amount billed over the customer's lifetime.

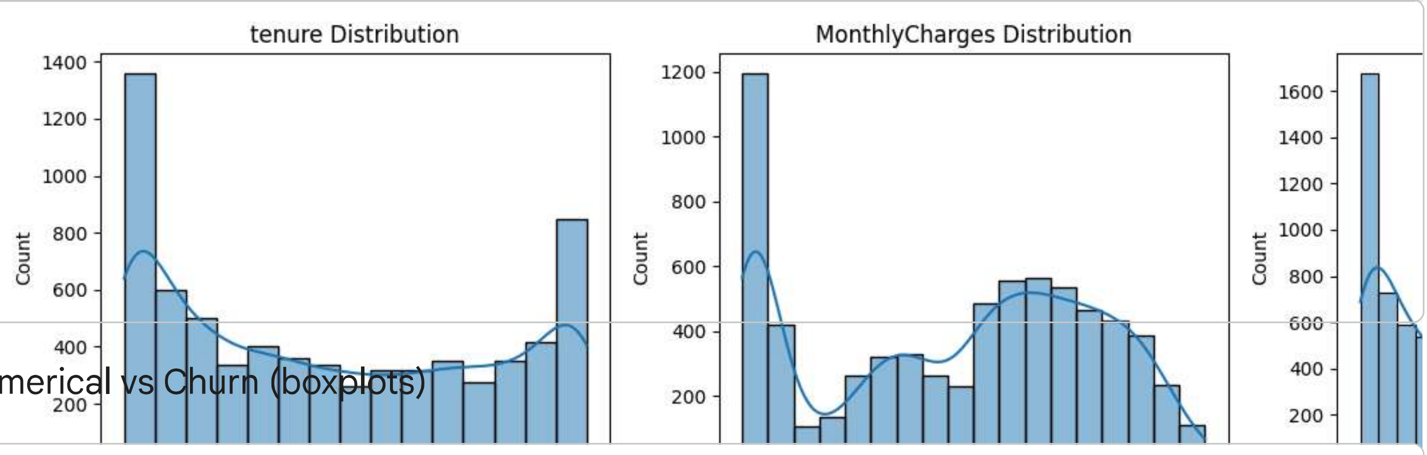
Using histograms and KDE plots, the notebook explores:

- Overall distribution and spread of each variable.
- Presence of skewness or outliers.

These insights help later when choosing transformations, scaling, and model types.

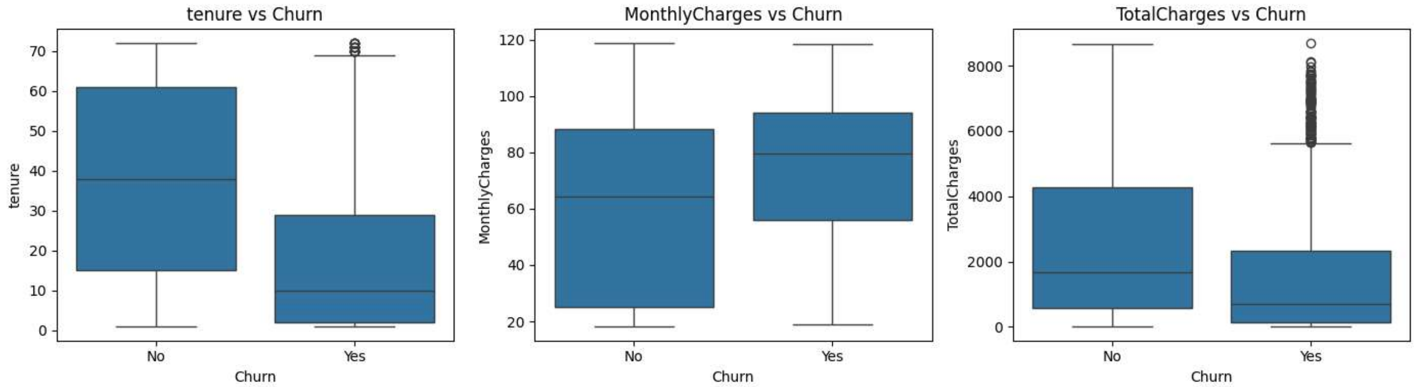
```
num_cols = ["tenure", "MonthlyCharges", "TotalCharges"]

plt.figure(figsize=(14, 4))
for i, col in enumerate(num_cols, 1):
    plt.subplot(1, 3, i)
    sns.histplot(df[col], kde=True)
    plt.title(f"{col} Distribution")
plt.tight_layout()
plt.show()
```



Numerical vs Churn (boxplots)

```
plt.figure(figsize=(14, 4))
for i, col in enumerate(num_cols, 1):
    plt.subplot(1, 3, i)
    sns.boxplot(x="Churn", y=col, data=df)
    plt.title(f"{col} vs Churn")
plt.tight_layout()
plt.show()
```



Numerical Features vs Churn

To see how numeric features relate to churn, the notebook uses boxplots and density plots split by churn label.

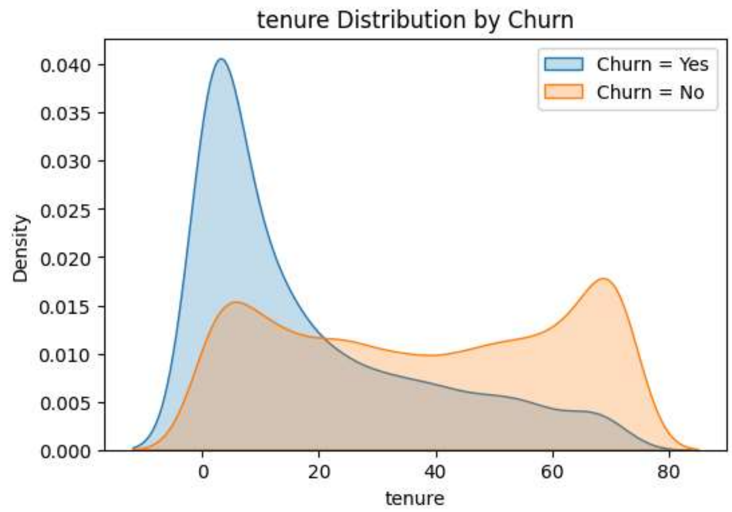
Typical patterns in telecom churn datasets include:

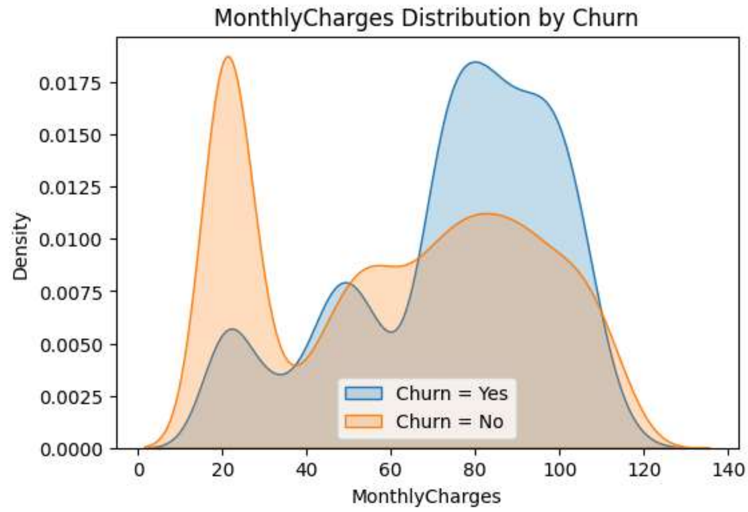
- **Tenure**: customers with shorter tenure tend to churn more, while long-tenure customers are more loyal.
- **MonthlyCharges**: higher monthly charges can be associated with higher churn, as customers are more price-sensitive.
- **TotalCharges**: churners often have lower total charges because they leave earlier in their lifecycle. [web:47][web:49]

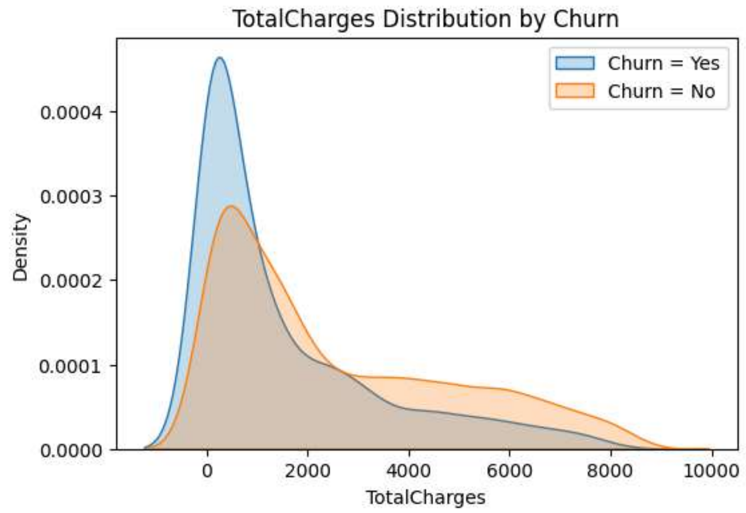
These relationships provide early evidence for which variables are strong churn drivers.

KDE per churn class

```
for col in num_cols:
    plt.figure(figsize=(6, 4))
    sns.kdeplot(
        df[df["Churn"] == "Yes"][col],
        label="Churn = Yes",
        fill=True,
        common_norm=False,
    )
    sns.kdeplot(
        df[df["Churn"] == "No"][col],
        label="Churn = No",
        fill=True,
        common_norm=False,
    )
    plt.title(f"{col} Distribution by Churn")
    plt.legend()
    plt.show()
```







6. Categorical vs Churn

Categorical Features vs Churn

This section examines how different categorical variables behave for churners vs non-churners using grouped bar charts. Important groups include: [web:47][web:52]

- **Contract type** (Month-to-month, One-year, Two-year).
- **Payment method** (Electronic check vs other methods).
- **Internet service type** (DSL, Fiber optic, No internet).
- **Security and support services** (OnlineSecurity, TechSupport, etc.).
- **Billing type** (PaperlessBilling).

Common patterns observed in telecom churn studies:

- Month-to-month contracts have significantly higher churn than long-term contracts.
- Customers paying via electronic check churn more than those using automatic or bank transfers.
- Fiber optic customers often show higher churn due to higher pricing or performance expectations.

- Customers without online security or tech support tend to churn more frequently.

```
cat_cols = [
    "Contract",
    "PaymentMethod",
    "InternetService",
    "OnlineSecurity",
    "TechSupport",
    "PaperlessBilling",
]

plt.figure(figsize=(18, 12))
for i, col in enumerate(cat_cols, 1):
    plt.subplot(3, 2, i)
    sns.countplot(x=col, hue="Churn", data=df)
    plt.title(f"{col} vs Churn")
    plt.xticks(rotation=30, ha="right")
plt.tight_layout()
plt.show()
```