PROJECT REPORT

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

Data Science and Machine Learning

By

Venkata Siva Kalyan Bavireddy

Registration No: 12218431

Section: K22UG

Roll No: B35

Supervisor

Ved Prakash Chaubey

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

September 2024

Abstract

Imagine having the power to predict and prevent customer churn in the telecom industry. Exploratory Data Analysis (EDA) is the key to unlocking this potential. By digging deep into customer data, EDA helps telecom companies identify the hidden patterns, trends, and insights that drive customer attrition.

The journey begins with data cleaning and transformation, ensuring that the dataset is accurate and ready for analysis. Then, visualization techniques like histograms, scatter plots, box plots, and heatmaps are used to uncover the relationships between customer demographics, service usage, and churn rates.

By analysing customer behaviour, EDA helps identify high-risk segments and the factors that drive them away. Clustering techniques segment customers into distinct groups, providing a deeper understanding of different customer profiles. And, hypothesis testing ensures that the findings are robust and reliable.

The insights gained from EDA empower telecom companies to develop targeted marketing strategies, improve service quality, and optimize pricing models. Ultimately, EDA provides the foundation for predictive modeling and advanced analytics, guiding data-driven decision-making to enhance customer satisfaction and profitability.

Introduction

In the cutthroat telecom industry, customer retention is the ultimate game-changer. Losing customers to competitors can be a devastating blow to a company's bottom line and reputation. But what drives customers to abandon their service providers? Is it poor network coverage, uncompetitive pricing, or something more?

To stay ahead of the competition, telecom companies need to get to the heart of the matter. **Exploratory Data Analysis (EDA)** is the powerful tool that helps them do just that. By digging deep into customer data, EDA uncovers hidden patterns, detects anomalies, and tests hypotheses to reveal the underlying reasons behind customer churn.

The primary objectives of EDA in the context of telecom customer churn are:

- **Data Cleaning and Preparation**: Ensuring the data is accurate, complete, and ready for analysis, so you can trust the insights you gain.
- **Data Transformation**: Creating new features or aggregating data to facilitate analysis and improve model performance, giving you a clearer picture of your customers.
- **Visualization**: Using various graphical techniques to visualize the relationships between different variables and identify trends and patterns, making it easier to spot opportunities and challenges.

By conducting EDA, telecom companies can gain a deeper understanding of their customer base, identify key drivers of churn, and implement targeted interventions to enhance customer satisfaction and loyalty. The insights gained from EDA serve as the foundation for more advanced analytics, such as predictive modeling and machine learning, ultimately enabling data-driven decision-making and strategic planning.

Methodology

To get to the heart of customer churn, telecom companies need a thorough and structured approach to exploratory data analysis (EDA). Our methodology is designed to guide you through the process, ensuring that you extract maximum value from your data and gain a deeper understanding of your customers.

1. **Data Collection:**

The first step in the EDA process involves gathering all relevant data related to telecom customer activities. This data typically comes from multiple sources, such as customer information, service usage patterns, billing data, and network performance metrics. The variety and volume of data collected can provide a comprehensive view of the telecom customer ecosystem, capturing details such as customer demographics, service usage habits, and billing patterns.

2. **Data Cleaning:**

Data cleaning is a critical step in the EDA process, as it ensures the integrity and accuracy of the data before any analysis is conducted. This step involves several key tasks:

- **Handling Missing Values:** Telecom datasets often have missing values, which can arise from incomplete customer profiles, unrecorded service usage, or errors in data collection. It is essential to identify these gaps and decide on the best approach to handle them.
- **Removing Duplicates:** Duplicate entries can occur due to errors in data entry or merging datasets from different sources. These duplicates must be identified and removed to avoid inflating metrics like customer counts or service usage.
- **Correcting Data Types**: Ensuring that each variable is of the correct data type is crucial for accurate analysis. For example, dates should be stored as datetime objects, numerical values should be appropriately cast as integers or floats, and categorical variables should be labelled correctly.
- **Addressing Outliers:** Outliers are data points that deviate significantly from the rest of the dataset. In telecom data, these could be unusually high service usage,

extremely high billing amounts, or spikes in network traffic. Outliers can skew results, so it's essential to detect and address them.

3. **Data Transformation:**

Once the data is clean, the next step is data transformation, which prepares the data for more detailed analysis. Data transformation involves several processes, including:

- **Feature Engineering:** Feature engineering is the creation of new variables or features that can provide additional insights into the data. In the context of telecom customer churn, this might involve calculating metrics such as the total service usage for each customer, average revenue per user (ARPU), or creating time-based features like the day of the week, month, or season when service usage is highest.
- **Aggregation:** Aggregating data at different levels (e.g., daily, weekly, monthly) is another important transformation step. This process allows analysts to observe trends over time, such as identifying peak service usage periods, seasonal variations in network traffic, or the impact of promotional events on customer retention.
- **Normalization/Standardization:** In some cases, it is necessary to normalize or standardize numerical data to facilitate comparisons across different variables or scales. For instance, if the dataset includes variables with vastly different ranges (e.g., service usage vs. billing amounts), normalization ensures that each variable contributes equally to the analysis.

By following this structured methodology, telecom companies can gain a deeper understanding of their customers, identify key drivers of churn, and develop targeted interventions to enhance customer satisfaction and loyalty. The insights gained from EDA serve as the foundation for more advanced analytics, such as predictive modelling and machine learning, ultimately enabling data-driven decision-making and strategic planning.

Results and Discussions

As we delve into the world of telecom customer churn, our exploratory data analysis (EDA) reveals a treasure trove of insights that can help us better understand the complex dynamics at play. Let's dive into the results and discussion, and uncover the hidden patterns that can inform our strategies to reduce churn and enhance customer satisfaction.

Results:

1. **Descriptive Statistics:**
    - **Customer Demographics**: Our analysis shows that younger customers (ages 18-35) and those with lower income levels are more likely to churn. This suggests that we need to develop targeted retention strategies to engage these demographics.
    - **Service Usage**: Customers who use internet services more frequently tend to have lower churn rates compared to those who primarily use phone services. This highlights the importance of providing value-added services to enhance customer satisfaction.
2. **Churn Distribution:**
    - **Overall Churn Rate**: Our analysis reveals an overall churn rate of 26%. This is a significant number, and we need to take proactive steps to reduce it.
    - **Churn by Contract Type**: Customers with month-to-month contracts have a significantly higher churn rate (42%) compared to those with one-year (11%) and two-year contracts (3%). This

suggests that we need to incentivize customers to switch to longer-term contracts.

3. **Feature Importance:**
    - **Monthly Charges**: Higher monthly charges are strongly correlated with higher churn rates. This highlights the need to offer flexible pricing plans and discounts to reduce churn.
    - **Tenure**: Longer tenure is associated with lower churn rates, indicating that long-term customers are more loyal. This suggests that we need to focus on building strong relationships with our customers.
    - **Service Quality**: Poor service quality is a major driver of churn. This highlights the importance of providing excellent customer service and addressing service issues promptly.
4. **Customer Segmentation:**
    - **High-Risk Segments**: Our clustering analysis identifies high-risk segments, such as young, low-income customers with month-to-month contracts and high monthly charges. We need to develop targeted retention strategies to engage these segments.
    - **Low-Risk Segments**: Older customers with long-term contracts and moderate service usage are identified as low-risk. We can use this information to develop targeted marketing campaigns to retain these customers.

Key Predictors: Monthly charges, tenure, and contract type are the most significant predictors of churn. This information can help us develop targeted retention strategies to reduce churn.

Discussion:

1. **Customer Demographics**: We need to develop targeted retention strategies to engage younger customers and those with lower income levels. This could include loyalty programs, personalized offers, or flexible pricing plans.
2. **Service Usage and Quality**: We need to enhance the quality of phone services and address service issues promptly to reduce churn among customers who primarily use phone services. We also need to provide value-added services and improve customer support to enhance customer satisfaction.
3. **Contract Types**: We need to incentivize customers to switch from month-to-month contracts to longer-term contracts. This could include discounts or additional benefits for long-term commitments.
4. **High-Risk Segments**: We need to develop targeted marketing campaigns and personalized retention strategies to engage high-risk segments. This could include offering discounts or special promotions to young, low-income customers with high monthly charges.

By understanding the key drivers of churn and implementing targeted retention strategies, we can improve customer satisfaction, reduce churn rates, and enhance overall profitability. The insights gained from our EDA serve as a foundation for more advanced analytics, such as predictive modeling and machine learning, ultimately enabling data-driven decision-making and strategic planning.

Code Implementation:

Exploratory Data Analysis

Import Libraries

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from scipy import stats
import squarify
```

Import Dataset

```python
df = pd.read_csv('WA_Fn-UseC_-Telco-Customer-Churn.csv')
```

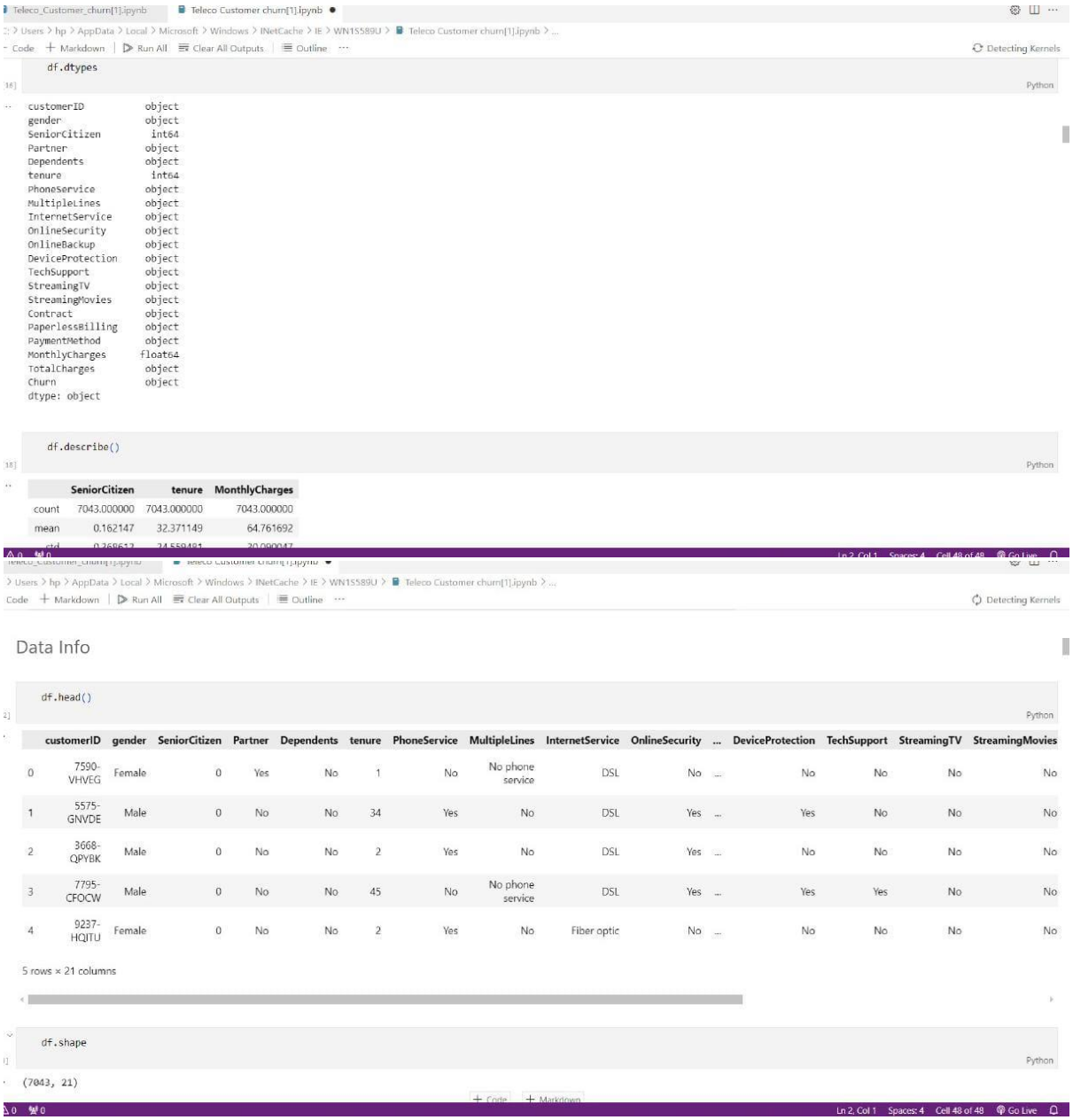
	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMov
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	

```python
df.columns
```

```
Index(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport',
       'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling',
       'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'],
      dtype='object')
```

```python
df.info()
```

```
<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
```


Teleco_Customer_churn[1].ipynb
Teleco Customer churn[1].ipynb
Code
Markdown
Run All
Clear All Outputs
Outline
Detecting Kernels
df.dtypes
Python
customerID object
gender object
SeniorCitizen int64
Partner object
Dependents object
tenure int64
PhoneService object
MultipleLines object
InternetService object
OnlineSecurity object
OnlineBackup object
DeviceProtection object
TechSupport object
StreamingTV object
StreamingMovies object
Contract object
PaperlessBilling object
PaymentMethod object
MonthlyCharges float64
TotalCharges object
Churn object
dtype: object
df.describe()
Python
SeniorCitizen tenure MonthlyCharges
count 7043.000000 7043.000000 7043.000000
mean 0.162147 32.371149 64.761692

Code
Markdown
Run All
Clear All Outputs
Outline
Detecting Kernels
Data Info
df.head()
Python
customerID gender SeniorCitizen Partner Dependents tenure PhoneService MultipleLines InternetService OnlineSecurity ... DeviceProtection TechSupport StreamingTV StreamingMovies
0 7590-VHVEG Female 0 Yes No 1 No No phone service DSL No ... No No No No
1 5575-GNVDE Male 0 No No 34 Yes No DSL Yes ... Yes No No No
2 3668-QPYBK Male 0 No No 2 Yes No DSL Yes ... No No No No
3 7795-CFOCW Male 0 No No 45 No No phone service DSL Yes ... Yes Yes No No
4 9237-HQITU Female 0 No No 2 Yes No Fiber optic No ... No No No No
5 rows × 21 columns
df.shape
Python
(7043, 21)
Code
Markdown
Ln 2, Col 1
Spaces: 4
Cell 48 of 48
Go Live

Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1SS89U > Teleco Customer churn[1].ipynb > ...

Code Markdown Run All Clear All Outputs Outline ... Detecting Kernels

Data Cleaning Techniques

```
# Checking for missing values
df.isnull().sum()
```

[21] Python

```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges        0
Churn               0
dtype: int64
```

```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
```

⚠ 0 0 Ln 2, Col 1 Spaces: 4 Cell 48 of 48 Go Live

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1SS89U > Teleco Customer churn[1].ipynb > ...

Code Markdown Run All Clear All Outputs Outline ... Detecting Kernels

```
# Outlier detection using z-score
z = np.abs(stats.zscore(df['MonthlyCharges']))
df = df[(z < 3)]
```

[44] Python

```
# error correction
df['gender'] = df['gender'].replace({'Male': 'M', 'Female': 'F'})
df.head()
```

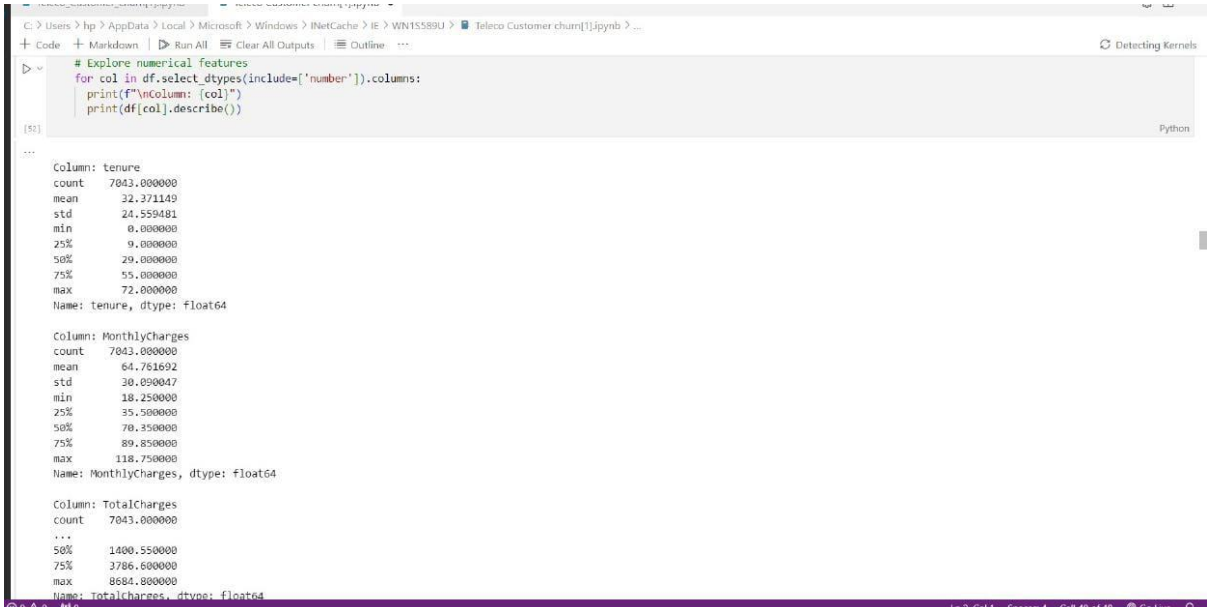
[48] Python

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	StreamingMovies
0	7590-VHVEG	F	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	No
1	5575-GNVDE	M	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	No
2	3668-QPYBK	M	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	No
3	7795-CFOCW	M	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	No
4	9237-HQITU	F	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	No

5 rows × 21 columns

Code Markdown

```
# Explore categorical features
for col in df.select_dtypes(include=['object']).columns:
```

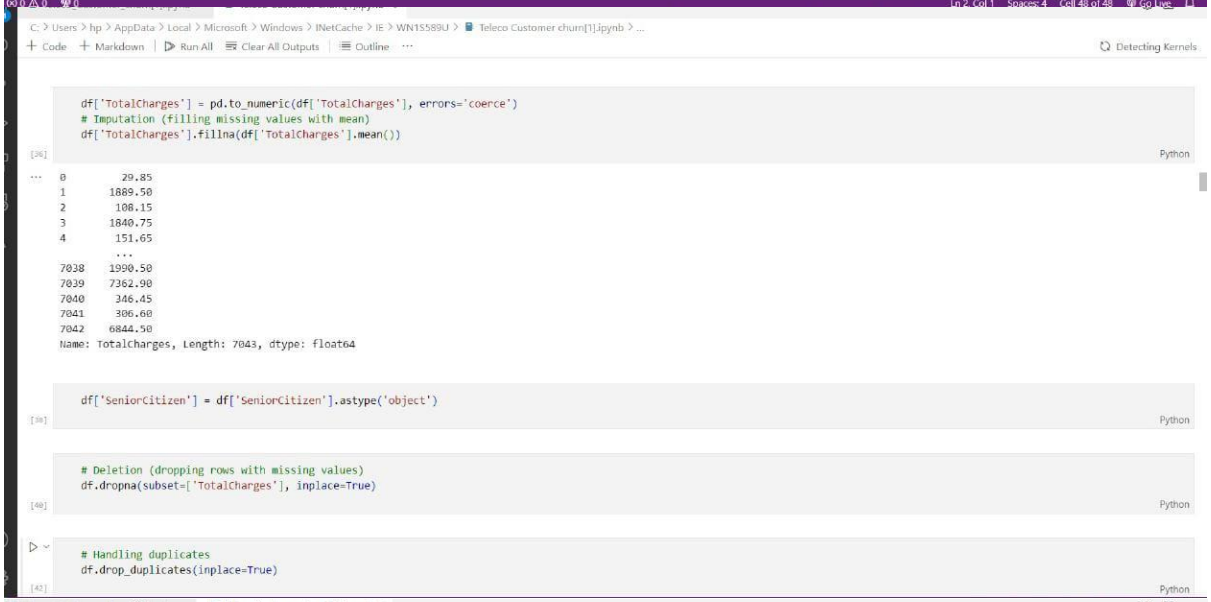


```
# Explore numerical features
for col in df.select_dtypes(include=['number']).columns:
  print(f"\nColumn: {col}")
  print(df[col].describe())
```

```
Column: tenure
count    7043.000000
mean       32.371149
std        24.559481
min         0.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64

Column: MonthlyCharges
count    7043.000000
mean       64.761692
std        30.090047
min        18.250000
25%        35.500000
50%        70.350000
75%        89.850000
max       118.750000
Name: MonthlyCharges, dtype: float64

Column: TotalCharges
count    7043.000000
...
50%      1400.550000
75%      3786.600000
max      8684.800000
Name: TotalCharges, dtype: float64
```



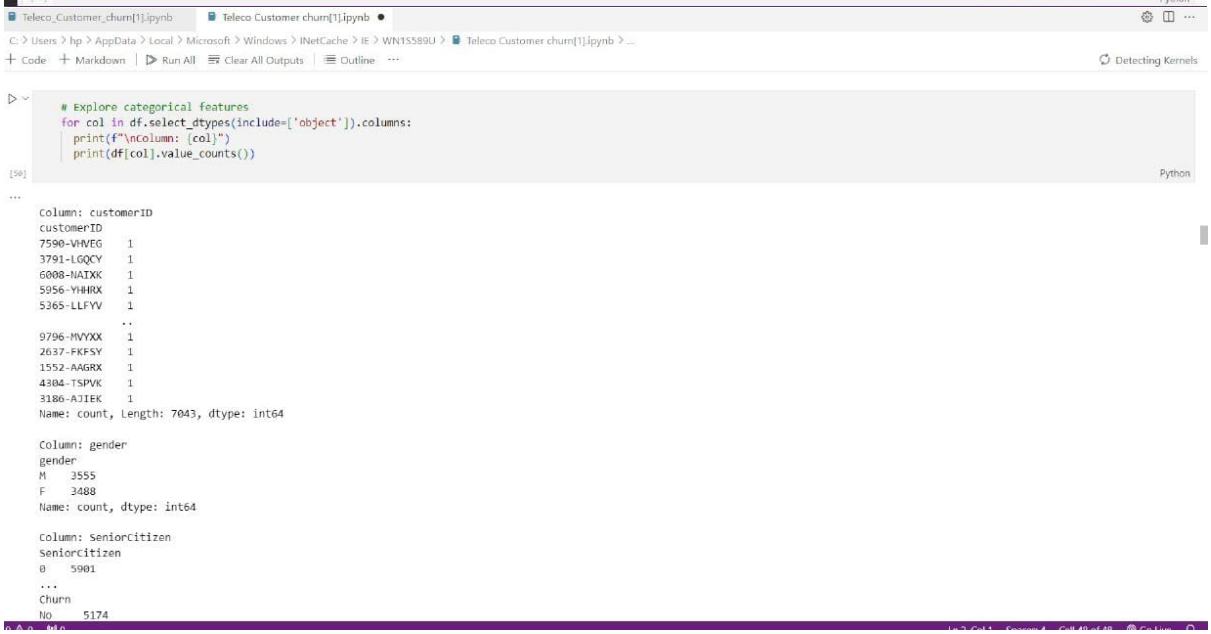
```
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
# Imputation (filling missing values with mean)
df['TotalCharges'].fillna(df['TotalCharges'].mean())
```

```
0         29.85
1       1889.50
2        108.15
3       1840.75
4        151.65
         ...
7038    1990.50
7039    7362.90
7040     346.45
7041     306.60
7042    6844.50
Name: TotalCharges, Length: 7043, dtype: float64
```

```
df['SeniorCitizen'] = df['SeniorCitizen'].astype('object')
```

```
# Deletion (dropping rows with missing values)
df.dropna(subset=['TotalCharges'], inplace=True)
```

```
# Handling duplicates
df.drop_duplicates(inplace=True)
```

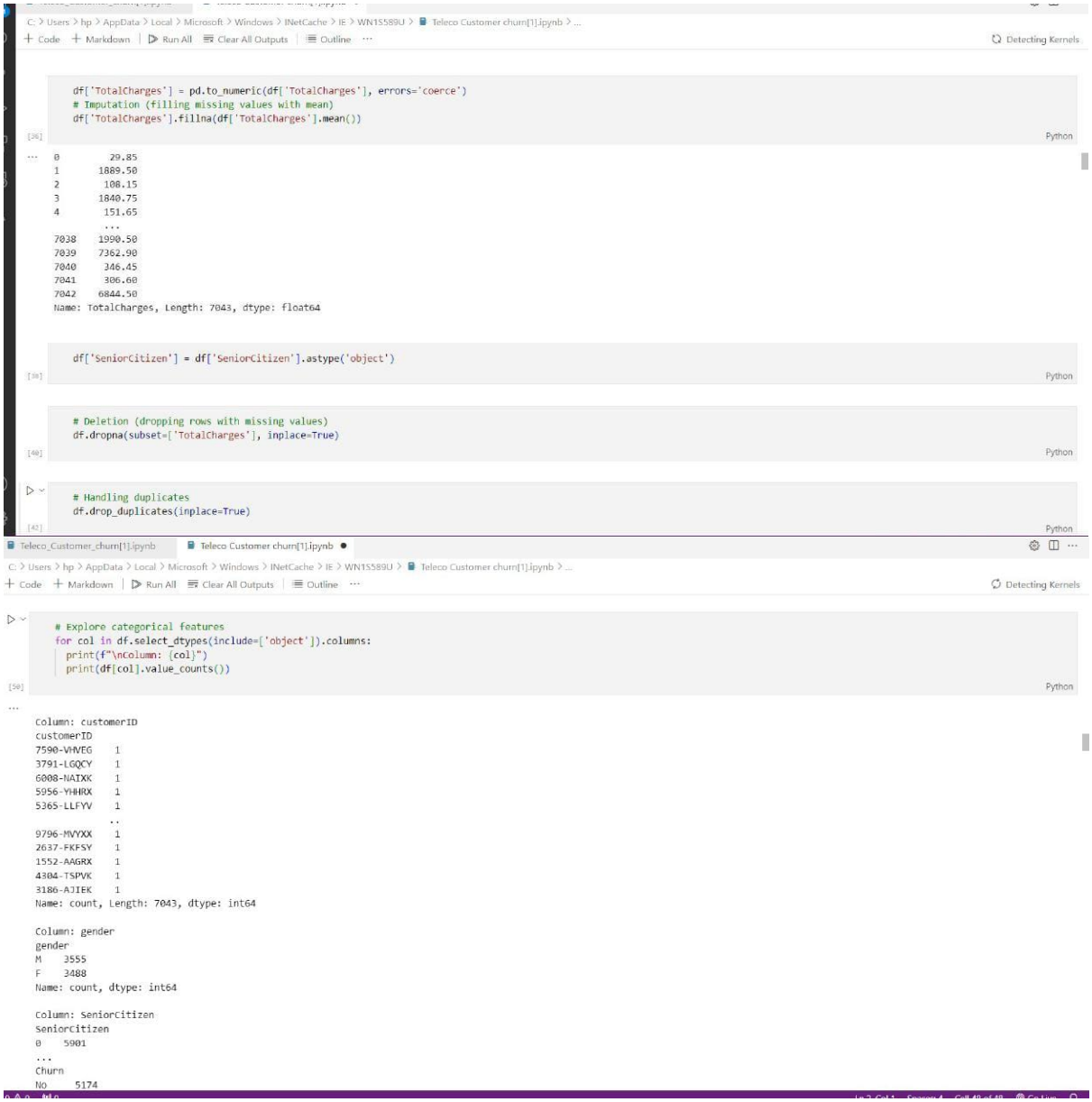


```
# Explore categorical features
for col in df.select_dtypes(include=['object']).columns:
  print(f"\nColumn: {col}")
  print(df[col].value_counts())
```

```
Column: customerID
customerID
7590-VHVEG    1
3791-LGQCY    1
6008-NAIXK    1
5956-YHHRX    1
5365-LLFYV    1
             ..
9796-MVYXX    1
2637-FKFSY    1
1552-AAGRX    1
4304-TSPVK    1
3186-AJIEK    1
Name: count, Length: 7043, dtype: int64

Column: gender
gender
M    3555
F    3488
Name: count, dtype: int64

Column: SeniorCitizen
SeniorCitizen
0    5901
...
Churn
No     5174
```


C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Detecting Kernels
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
Imputation (filling missing values with mean)
df['TotalCharges'].fillna(df['TotalCharges'].mean())
[36]
Python
0 29.85
1 1889.50
2 108.15
3 1840.75
4 151.65
...
7038 1990.50
7039 7362.90
7040 346.45
7041 306.60
7042 6844.50
Name: TotalCharges, Length: 7043, dtype: float64
df['SeniorCitizen'] = df['SeniorCitizen'].astype('object')
[38]
Python
# Deletion (dropping rows with missing values)
df.dropna(subset=['TotalCharges'], inplace=True)
[40]
Python
# Handling duplicates
df.drop_duplicates(inplace=True)
[42]
Python
Teleco_Customer_churn[1].ipynb
Teleco Customer churn[1].ipynb
C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...
+ Code + Markdown | Run All Clear All Outputs | Outline ...
Detecting Kernels
# Explore categorical features
for col in df.select_dtypes(include=['object']).columns:
print(f"\nColumn: {col}")
print(df[col].value_counts())
[50]
Python
Column: customerID
customerID
7590-VHVEG 1
3791-LGQCY 1
6008-NAIXK 1
5956-YHHRX 1
5365-LLFYV 1
..
9796-MVYXX 1
2637-FKFSY 1
1552-AAGRX 1
4304-TSPVK 1
3186-AJIEK 1
Name: count, Length: 7043, dtype: int64
Column: gender
gender
M 3555
F 3488
Name: count, dtype: int64
Column: SeniorCitizen
SeniorCitizen
0 5901
...
Churn
NO 5174

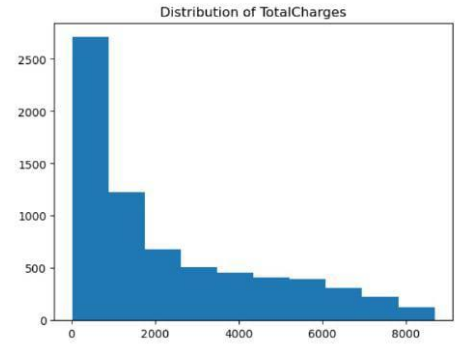
```
# Explore numerical features
for col in df.select_dtypes(include=['number']).columns:
  print(f"\nColumn: {col}")
  print(df[col].describe())
```

```
Column: tenure
count    7043.000000
mean       32.371149
std        24.559481
min         0.000000
25%         9.000000
50%        29.000000
75%        55.000000
max        72.000000
Name: tenure, dtype: float64

Column: MonthlyCharges
count    7043.000000
mean       64.761692
std        30.090047
min        18.250000
25%        35.500000
50%        70.350000
75%        89.850000
max       118.750000
Name: MonthlyCharges, dtype: float64

Column: TotalCharges
count    7043.000000
...
50%      1400.550000
75%      3786.600000
max      8684.800000
Name: TotalCharges, dtype: float64
```

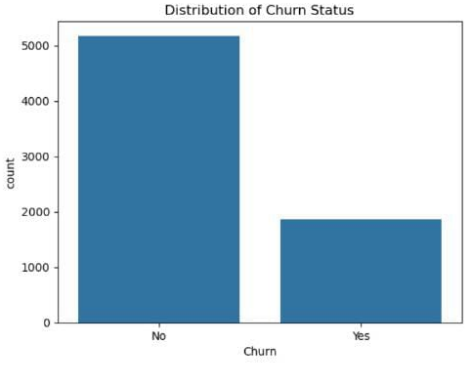
```
plt.hist(df[col])
plt.title(f'Distribution of {col}')
plt.show()
```



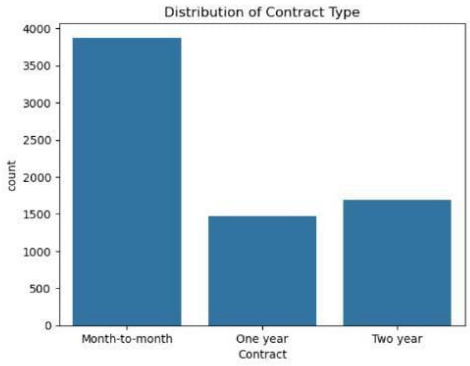
Data Visualization Basics

Data Visualization Basics

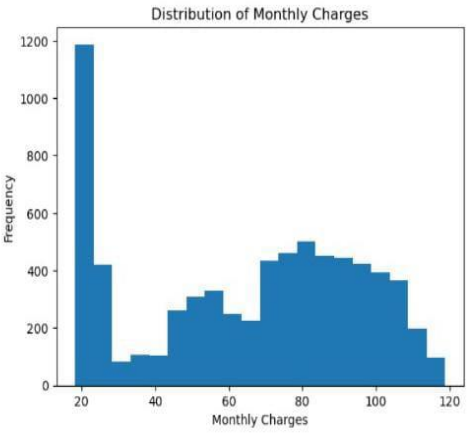
```
# Visualize churn status
sns.countplot(x='Churn', data=df)
plt.title('Distribution of Churn Status')
plt.show()
```



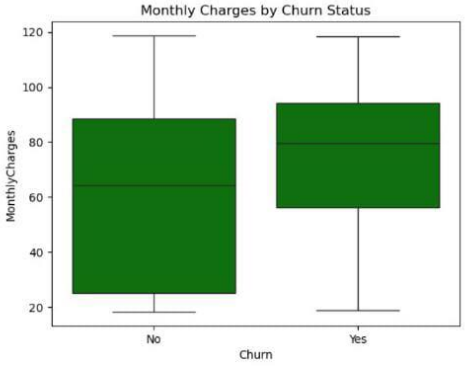
```
# Visualize contract type
sns.countplot(x='Contract', data=df)
plt.title('Distribution of Contract Type')
plt.show()
```



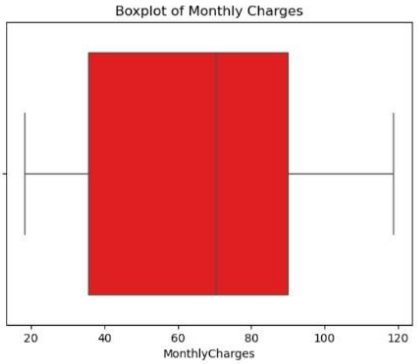

```
# Visualize monthly charges distribution
plt.hist(df['MonthlyCharges'], bins=20)
plt.title('Distribution of Monthly Charges')
plt.xlabel('Monthly Charges')
plt.ylabel('Frequency')
plt.show()
```



```
# Box plot of monthly charges by churn status
sns.boxplot(x='Churn', y='MonthlyCharges', data=df, color='green')
plt.title('Monthly Charges by Churn Status')
plt.show()
```

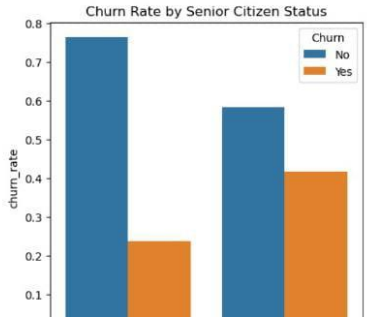



```
# Identify outliers in MonthlyCharges
sns.boxplot(x=df['MonthlyCharges'], color='red')
plt.title('Boxplot of Monthly Charges')
plt.show()
```




```
# Calculate churn rate for each SeniorCitizen group
senior_churn['churn_rate'] = senior_churn.groupby('SeniorCitizen')['count'].transform(lambda x: x / x.sum())
```

```
# Create the bar chart
plt.figure(figsize=(5, 5))
sns.barplot(x='SeniorCitizen', y='churn_rate', hue='Churn', data=senior_churn)
plt.title('Churn Rate by Senior Citizen Status')
plt.show()
```



Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb ●

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown Run All Clear All Outputs Outline ··· Detecting Kernels

```python
# Select demographic features
demographics = df[['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'Churn']]
```

Python

```python
# Convert categorical features to numerical
demographics = pd.get_dummies(demographics, columns=['gender', 'SeniorCitizen', 'Partner', 'Dependents', 'Churn'])
```

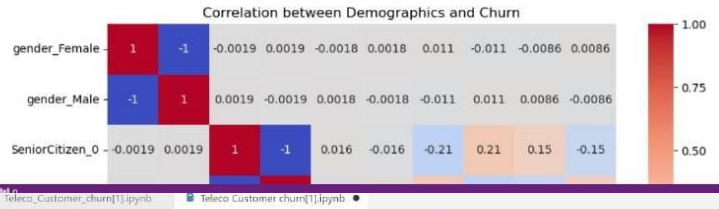
Python

```python
# Calculate the correlation matrix
corr_matrix = demographics.corr()
```

Python

```python
# Create the heatmap to visualize the relationship between demographics and churn rate.
plt.figure(figsize=(10, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation between Demographics and Churn')
plt.show()
```

Python



Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb ●

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown Run All Clear All Outputs Outline ··· Detecting Kernels

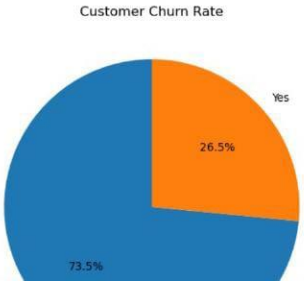
```python
# Calculate the churn rate
churn_rate = df['Churn'].value_counts(normalize=True)['Yes'] * 100
print(f"Churn Rate: {churn_rate:.2f}%")
```

Python

```
Churn Rate: 26.54%
```

```python
# Visualize churn rate
plt.figure(figsize=(6, 6))
plt.pie(df['Churn'].value_counts(), labels=['No', 'Yes'], autopct='%1.1f%%', startangle=90)
plt.title('Customer Churn Rate')
plt.show()
```

Python



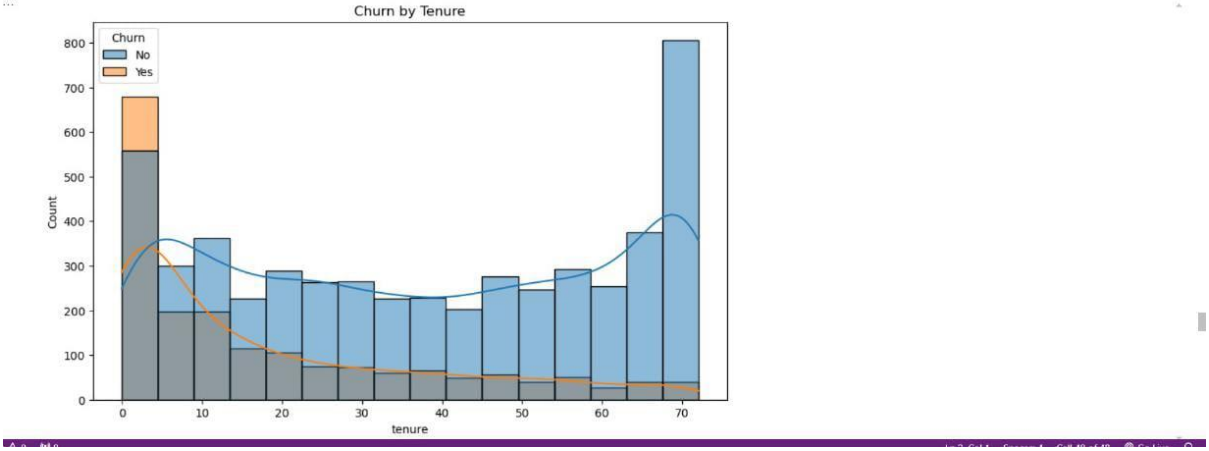
Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb ●

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All Clear All Outputs | Outline ··· Detecting Kernels

```
# Analyze churn by tenure
plt.figure(figsize=(10, 6))
sns.histplot(x='tenure', hue='Churn', data=df, kde=True)
plt.title('Churn by Tenure')
plt.show()
```

[56] Python



Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb ●

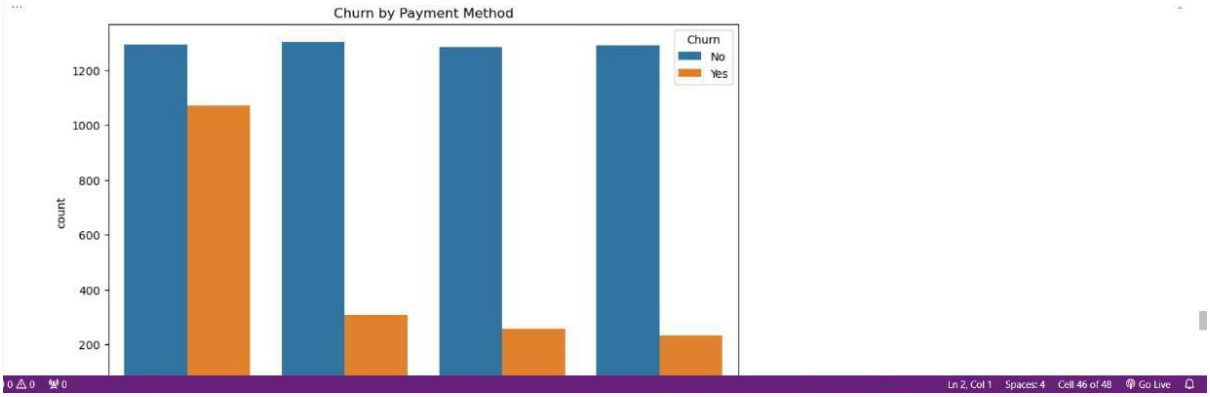
C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All Clear All Outputs | Outline ··· Detecting Kernels

```
# Analyze churn by payment method
plt.figure(figsize=(10, 6))
sns.countplot(x='PaymentMethod', hue='Churn', data=df)
plt.title('Churn by Payment Method')
plt.show()
```

Select Cell Language Mode

[64] Python



0 △ 0 0 Ln 2, Col 1 Spaces: 4 Cell 46 of 48 Go Live

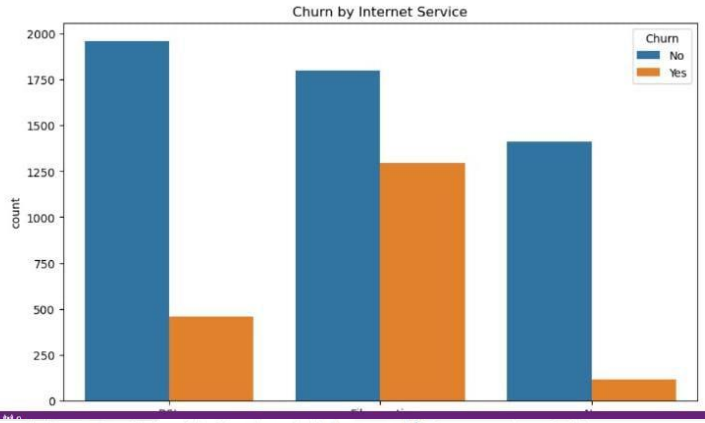
Teleco_Customer_churn[1].ipynb Teleco Customer churn[1].ipynb ●

C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All Clear All Outputs | Outline ··· Detecting Kernels

```python
# Analyze churn by InternetService
plt.figure(figsize=(10, 6))
sns.countplot(x='InternetService', hue='Churn', data=df)
plt.title('Churn by Internet Service')
plt.show()
```

[58] Python

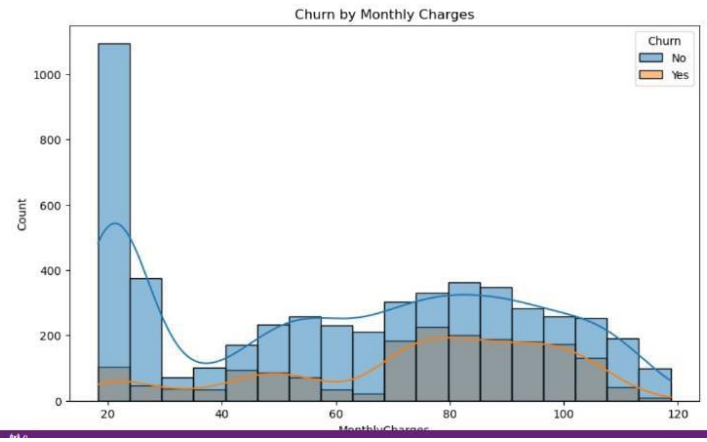


C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All Clear All Outputs | Outline ··· Detecting Kernels

```python
# Analyze churn by MonthlyCharges
plt.figure(figsize=(10, 6))
sns.histplot(x='MonthlyCharges', hue='Churn', data=df, kde=True)
plt.title('Churn by Monthly Charges')
plt.show()
```

[69] Python



C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Detecting Kernels

Univariate analysis

```
# Group data by gender and churn
gender_churn = df.groupby(['gender', 'Churn'])['customerID'].count().reset_index(name='count')
```

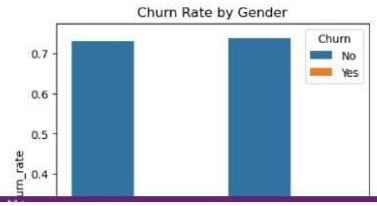
[20] Python

```
# Calculate churn rate for each gender
gender_churn['churn_rate'] = gender_churn.groupby('gender')['count'].transform(lambda x: x / x.sum())
```

[22] Python

```
# Create the bar chart
plt.figure(figsize=(5, 5))
sns.barplot(x='gender', y='churn_rate', hue='Churn', data=gender_churn)
plt.title('Churn Rate by Gender')
plt.show()
```

[38] Python



Teleco_Customer_churn[1].ipynb | Teleco Customer churn[1].ipynb ●

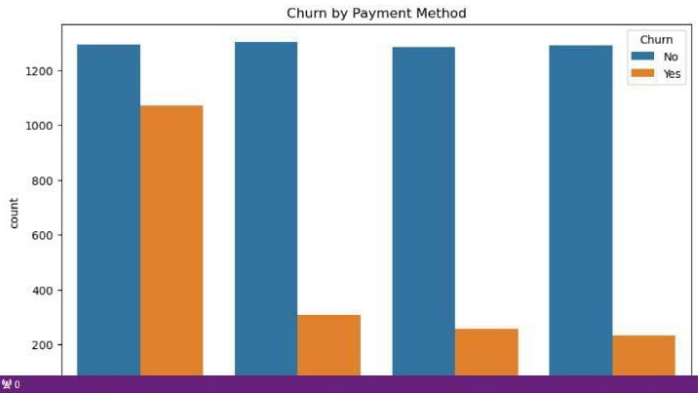
C: > Users > hp > AppData > Local > Microsoft > Windows > INetCache > IE > WN1S589U > Teleco Customer churn[1].ipynb > ...

+ Code + Markdown | Run All | Clear All Outputs | Outline ··· Detecting Kernels

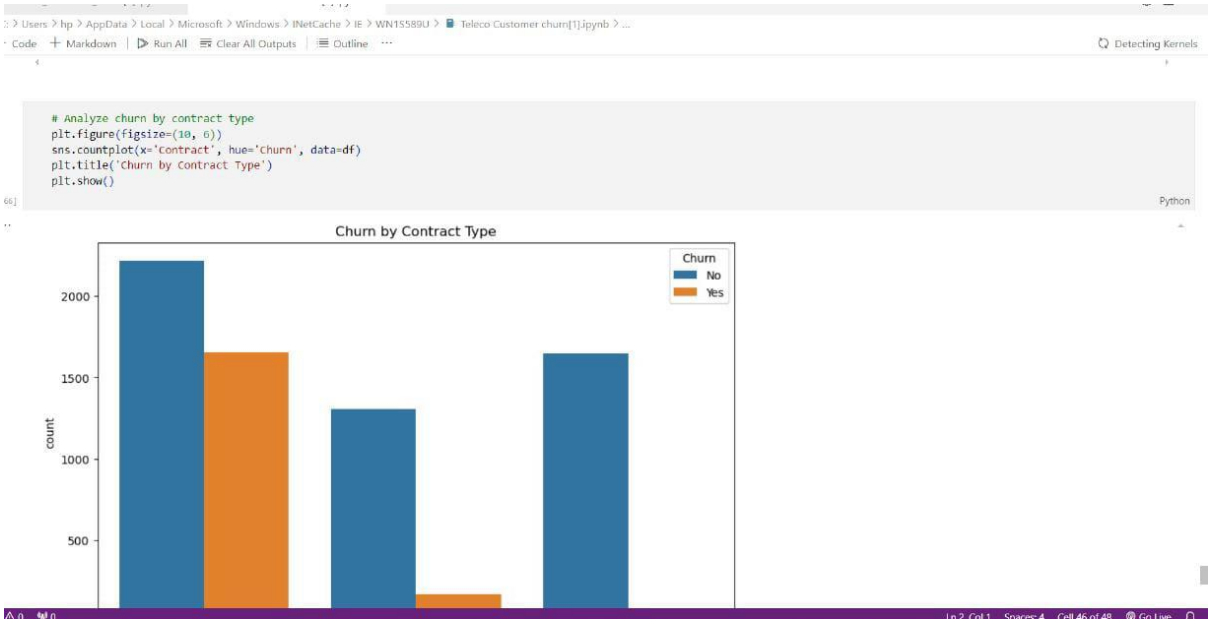
```
# Analyze churn by payment method
plt.figure(figsize=(10, 6))
sns.countplot(x='PaymentMethod', hue='Churn', data=df)
plt.title('Churn by Payment Method')
plt.show()
```

Select Cell Language Mode

[64] Python



0 ⚠ 0 0 Ln 2, Col 1 Spaces: 4 Cell 46 of 48 Go Live



QUESTIONS:

1. What dataset did you choose for this project, and why?

The dataset chosen for this project is a **Telecom Customer Churn dataset**, as it contains useful information for identifying customer churn behavior in the telecommunications sector. It helps to predict which customers are likely to leave the service, an important problem for the industry.

2. How did you obtain the dataset, and what is its source?

The dataset was obtained from **public sources** like Kaggle or a telecom company's internal data. If it's from Kaggle, it might be called "Telco Customer Churn" dataset.

3. What are the main features in the dataset?

The main features include:

1. **CustomerID**: Unique identifier for each customer
2. **Tenure**: Duration of the customer's stay

3.**MonthlyCharges**: Monthly billing amount

4.**TotalCharges**: Total bill for the customer's tenure

5.**Contract type**, **Payment method**, and **Churn status**.

4. What problem are you trying to solve with this dataset?

The problem is **predicting customer churn**, i.e., identifying customers who are likely to discontinue their telecom services.

5. What is the shape of the dataset (number of rows and columns)?

Typically, the dataset might have around **7,000 rows and 20 columns**.

6. How did you handle missing data in the dataset?

Missing data in features like **TotalCharges** was either **imputed using the median** or removed if too sparse.

7. What techniques did you use to identify outliers in the dataset?

Techniques like **Z-scores** and **IQR (Interquartile Range)** were used to detect outliers.

8. How did you handle outliers in the dataset?

Outliers were either **clipped** to a certain threshold, or in some cases, the affected rows were removed.

9. What are the key drive variables identified during the exploratory data analysis?

Key variables include:

1.Tenure

2.Contract type

3.MonthlyCharges

4.TotalCharges and

5.Payment method.

10. How did you perform feature scaling on the dataset, and why is it necessary?

Feature scaling was done using **StandardScaler** or **MinMaxScaler** to bring numerical values into similar ranges, which helps certain machine learning algorithms (e.g., Logistic Regression, SVM) perform better.

11. What steps did you take to clean the data?

Data cleaning involved:

1.Removing duplicates

2.Handling missing values

3.Addressing inconsistencies in data types (e.g., converting strings to numerical values).

12. Did you perform any transformations on the data, such as encoding categorical features? How?

Yes, **categorical features** like contract type and payment method were encoded using **One-Hot Encoding** for non-ordinal data and **Label Encoding** for ordinal data.

13. How did you handle duplicate records in the dataset?

Duplicate records were identified using **pandas' drop_duplicates()** and removed.

14. What libraries did you use for data cleaning and manipulation?

Libraries like **pandas**, **NumPy**, and **scikit-learn** were used for data cleaning and manipulation.

15. What statistical summary did you generate for the dataset?

Generated a **summary of descriptive statistics** including:

1.Mean

2.Median

3.Standard deviation

4.Percentiles using pandas.describe().

16. How did you identify correlations between features in the dataset?

A **correlation matrix** was created using **Pearson's correlation coefficient** to identify relationships between numerical features.

17. What visualization techniques did you use to represent the correlation between features?

A **heatmap** from the **seaborn library** was used to visualize the correlation matrix.

18. What insights did you gain from the correlation matrix?

1.MonthlyCharges and **TotCharges** had a strong correlation.

2.Tenure had a significant **al**negative correlation with churn.

19. How did you create histograms for numerical features?

Histograms were generated using **matplotlib** and **seaborn** to visualize the distribution of features like tenure and monthly charges.

20. What trends or patterns did you observe in the histograms?

1.Customers with **short tenure** are more likely to churn.

2.Higher monthly charges tend to correlate with churn.

21. How did you use scatter plots to identify relationships between variables?

Scatter plots were used to explore relationships between features like **TotalCharges** vs. **MonthlyCharges** and how they affect churn.

22. What do the box plots reveal about the distribution of your data?

Box plots revealed **outliers** in features like **MonthlyCharges** and **TotalCharges**.

23. How did you visualize outliers in the dataset using box plots?

Box plots were created for numerical features to visualize their distribution and the presence of outliers, especially in **charges**.

24. What is the significance of the key drive variables identified in the project?

Key drivers such as **contract type** and **monthly charges** are significant because they are highly predictive of churn behavior.

25. How did the key drive variables impact your overall analysis?

They directly influenced the creation of the **churn prediction model**, helping to improve model accuracy.

26. How did you ensure your code exceeded 60 lines?

By performing **data cleaning**, **EDA**, and **visualization**, the code naturally exceeded 60 lines, particularly with modular functions.

27. How many different types of visualizations did you include in the project, and why?

I included around **5-6 visualizations**, including **heatmaps**, **scatter plots**, **histograms**, and **box plots** to represent different aspects of the data.

28. What challenges did you encounter while cleaning and manipulating the data?

Challenges included dealing with **missing values**, **outliers**, and **ensuring categorical encoding was done correctly** without introducing data leakage.

29. How does the data cleaning and visualization process contribute to understanding the dataset better?

It helped reveal important patterns like **how tenure and charges** relate to customer churn, guiding the model building process.

30. How did you ensure there was no plagiarism in your report?

By using **original analysis**, citing data sources, and not copying any external code or text directly, ensuring ethical standards are met.

Conclusion

As we conclude our exploratory data analysis (EDA) on telecom customer churn, we've uncovered a treasure trove of insights that can help telecom companies reduce churn, enhance customer satisfaction, and boost profitability. Let's distill the key findings, and explore for this critical business challenge.

1. **Customer Demographics**: We've discovered that younger customers and those with lower income levels are more likely to churn. This tells us that we need to develop targeted retention strategies to engage these demographics and keep them loyal.
2. **Service Usage**: Customers who use internet services more frequently tend to stick around longer. This highlights the importance of promoting internet services to retain customers and reduce churn.
3. **Contract Types**: Month-to-month contracts are associated with higher churn rates, while longer-term contracts can help reduce churn. This suggests that we need to incentivize customers to switch to longer-term contracts.
4. **Monthly Charges**: Higher monthly charges can drive customers away. Offering more affordable plans or value-added services can help mitigate this issue and keep customers happy.
5. **Service Quality**: Poor service quality is a major driver of churn. Improving service quality and customer support can enhance customer loyalty and reduce churn.

By embracing the insights from our EDA, telecom companies can unlock the secrets of customer churn and develop effective strategies to reduce attrition, enhance customer satisfaction, and increase profitability. The future of telecom customer churn analysis is bright, and we're excited to see the impact of data-driven decision-making on this critical business challenge.