Exploratory Data Analysis (EDA) Report

Project Title: Customer Churn Analysis for Telecommunication Company
Project Sponsor: Doris Lee, CEO of Advanced Consultant Services
Project Manager: Maniram Luitel

Project Start Date: July 1, 2024
Expected Project End Date: September 15, 2024

Prepared by: Bhavesh Chaudhary
Date: August 01, 2024
Role: Data Engineer

Introduction

Purpose of the Report

The only objective of this Exploratory Data Analysis (EDA) report is to fully comprehend the data related to customer churn for Advanced Consultant Service. This paper will strive to bring to light underlying patterns, relationships, and anomalies within the data and consequently inform the following phases of the project on customer churn analysis. We go further to perform an EDA, with the intention of drawing effective insights to guide the feature engineering, model selection, and evaluation processes that will help predict and mitigate customer churn.

Project Overview

Customer churn is the worst nightmare of companies because it refers to the condition where customers stop using products or services. Understanding and predicting customer churn remain some of the most important issues in business, ensuring that customer bases are retained and businesses remain sustainable. The aim of this project is to analyze the customer data received from Advanced Consultant Service and build a predictive model that can accurately determine which customers are likely to churn. Using this model, the company will have specific target retention strategies for their customers so as to alleviate high levels of churning and concurrently increase satisfaction among their customers.

The dataset that I use in this analysis contains various attributes with regard to customer demographics, account information, and service usage.

Major Variables in the Dataset:

- Gender: Gender of the customer.
- SeniorCitizen: Whether the customer is a senior citizen.
- Dependents: Whether the customer has dependents.
- Tenure: Number of months the customer has been with the company.
- PhoneService: If the customer has a telephone line or not.
- MultipleLines: Whether the customer has multiple lines.
- InternetService: Which type of internet service the customer has (DSL, Fiber optic, or None).
- Contract: What kind of contract the customer has (Month-to-month, One year, Two year).
- MonthlyCharges: Amount the customer is charged monthly.
- Churn: Whether the customer has churned (Yes or No).

The processed dataset includes additional engineered features:

- Gender_Female, Gender_Male: Encoded gender variables.
- Dependents_No, Dependents_Yes: Encoded dependents variables.
- PhoneService_No, PhoneService_Yes: Encoded phone service variables.
- MultipleLines_No, MultipleLines_Yes: Encoded multiple lines variables.
- InternetService_DSL, InternetService_Fiber optic: Encoded internet service variables.
- Contract_Month-to-month, Contract_One year, Contract_Two year: Encoded contract variables.
- Churn_No, Churn_Yes: Encoded churn variables.
- Charges_Per_Tenure: Output of feature engineering.
- TotalCharges: Amount that was charged in total to the customer.

EDA involves a few crucial steps:

1. **Data Cleaning and Preprocessing:** Activities for maintaining data quality, which involve treatments of missing values and inconsistencies.
2. **Descriptive Statistics:** Summarizing the main features of the dataset through numerical and graphical methods.
3. **Target Variable Analysis:** Analyzing the distribution of the target variable (churn) to understand the rate of churn and its characteristics.
4. **Univariate Analysis:** Check the individual distribution of the various variables to identify some patterns and also outliers.
5. **Bivariate and Multivariate Analysis:** Examination of the relationship between a pair of variables or groups of variables to identify an association or interaction.
6. **Correlation Analysis:** Compute and visualize the correlation matrix between numerical variables in order to find important relationships.
7. **Outlier Detection:** Detection of observations that are abnormally small or large in relation to the rest, which could be either extreme values from the model or human errors.
8. **Visualization:** The process of creating various plots and charts for visually investigating data and focusing attention on crucial information.

We end this EDA report with a high understanding of the structure, quality, and key features embedded within the data. These insights would be the basis of our future predictive model regarding customer churn, which could enable Advanced Consultant Service to take proactive measures for the retention of their valuable customers.

Data Overview

Data Description

The data under consideration has been provided by Advanced Consultant Service and has detailed information on customers and service usage, along with the customer churn status. It is the goal to perform an analysis of the data with the intention of finding out the patterns in the data and factors which can be contributory for churn so that a predictive model can be developed.

Data Source

The data for the Advanced Consultant Service comes from customer records. The data records contain demographics, account information, and service usage information.

Number of Records and Features

There are 7,043 records in all, with each record corresponding to a different customer. There is a total of 20 features in this dataset including input features and the target variable (Churn). These features contain various types of details that concern the customer, ranging from demographic information to the account and services usage.

Data Dictionary

The table below explains each of the features in the dataset:

Feature	Description
gender	Gender of the customer (Male, Female).
SeniorCitizen	Tells if a customer is a senior citizen (1, 0).
Dependents	Whether a customer has dependents (Yes, No).
Tenure	Number of months the customer has stayed with the company.

PhoneService	Is the customer using a phone service? (Yes, No)
MultipleLines	If the customer has multiple lines, (Yes, No)
InternetService	The type of internet service the customer has DSL or Fiber optic, (Yes, No).
Contract	The type of contract the customer has taken (Month-to-month, One year, Two year).
MonthlyCharges	The amount charged to the customer on a monthly basis.
TotalCharges	The total amount charged to the customer.
Churn	Indicates whether the customer churned (Yes, No).
gender_Female	Encoded gender variable (1 if Female, 0 otherwise).
gender_Male	Encoded gender variable (1 if Male, 0 otherwise).
Dependents_No	Encoded dependents variable (1 if No, 0 otherwise).
Dependents_Yes	Encoded dependents variable (1 if Yes, 0 otherwise).
PhoneService_No	Encoded phone service variable (1 if No, 0 otherwise).
PhoneService_Yes	Encoded phone service variable (1 if Yes, 0 otherwise).
MultipleLines_No	Encoded multiple lines variable (1 if No, 0 otherwise).
MultipleLines_Yes	Encoded multiple lines variable (1 if Yes, 0 otherwise).
InternetService_DSL	Encoded internet service variable (1 if DSL, 0 otherwise).
InternetService_Fiber optic	Encoded internet service variable (1 if Fiber optic, 0 otherwise).
Contract_Month-to-month	Encoded contract variable (1 if Month-to-month, 0 otherwise).
Contract_One year	Encoded contract variable (1 if One year, 0 otherwise).
Contract_Two year	Encoded contract variable (1 if Two years, 0 otherwise).
Churn_No	Encoded churn variable (1 if No, 0 otherwise).
Churn_Yes	Encoded churn variable (1 if Yes, 0 otherwise).
Charges_Per_Tenure	Feature engineered variable representing charges per tenure.

Data Preprocessing

Data Cleaning Process

This is the most critical activity that is required to prepare the dataset in terms of accuracy, completeness, and readiness for analysis. The steps followed in cleaning the data were as follows:

1. **Handling Missing Values**:

- **Dropping Rows with Missing Values:**
  Missing values from rows of the dataset were dropped to keep the correctness of the dataset.

```
df = df.dropna()
```

- **Imputing Missing Values:**
  For the numeric columns, missing values were replaced using mean imputation to maintain the data maximum and having no bias.

```
from sklearn.impute import SimpleImputer
imputer = SimpleImputer(strategy='mean')
df_numeric = df.select_dtypes(include=['float64', 'int64'])
df_numeric_imputed = pd.DataFrame(imputer.fit_transform(df_numeric), columns=df_numeric.columns)
df[df_numeric.columns] = df_numeric_imputed
```

2. **Encoding Categorical Variables**

- **One-Hot Encoding:**

  For categorical variables, it is transformed. This is a process that changes one column for every category through this transformation, in order for such variables to be usable in machine learning models.

```
from sklearn.preprocessing import OneHotEncoder
encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
categorical_columns = df.select_dtypes(include=['object']).columns
encoded_data = pd.DataFrame(encoder.fit_transform(df[categorical_columns]),
columns=encoder.get_feature_names_out(categorical_columns))
df = df.drop(columns=categorical_columns)
df = pd.concat([df, encoded_data], axis=1)
```

These steps ensured that the dataset was free from missing values and that categorical data was appropriately transformed for analysis.

Feature Engineering

It permits new feature creation or transformation of the already existing features that have been done to empower the dataset in a predictive manner.

Created New Features

1. Charges Per Tenure:

- **Description:** The Average monthly charges per month of tenure.
- **Significance:** It would help gain knowledge about the expenses of customers with respect to tenure with the company. Higher charges per tenure could be an indication of more valuable customers or potentially higher dissatisfaction leading to churn.
-
```
df['Charges_Per_Tenure'] = df['MonthlyCharges'] / (df['tenure'] + 1)
```

2. Total Charges:

- **Description:** This attribute re-calculates the total charges to maintain consistency and correctness.
- **Significance:** This makes sure that the total amount charged to the customer is represented accurately, and this forms a crucial point for financial analysis and understanding customer value.
- ```
  df['TotalCharges'] = df['MonthlyCharges'] * df['tenure']
  ```

3. Encode Contract Type:

- **Description:** Encoded the contract type into numerical values.
- **Reason:** This would make it possible to analyze and use in predictive models.
- ```
  contract_mapping = {'Month-to-month': 0, 'One year': 1, 'Two year': 2}
  df['Contract_Type'] = df['Contract'].map(contract_mapping)
  ```

4. Payment Method Encoding:

- **Description:** Encoded the payment method into numerical values.
- **Importance:** When we encode the payment methods, we can then find if there is any relationship between the payment method and churn.
- ```
  payment_mapping = {'Electronic check': 0, 'Mailed check': 1, 'Bank transfer (automatic)': 2, 'Credit card (automatic)': 3}
  df['Payment_Method'] = df['PaymentMethod'].map(payment_mapping)
  ```

These features are created to be able to extract better information from the features of the dataset, which would help build better predictive models with a relevant and informative set of data points. Feature engineering is a process that not only sharpens data into a more effective dataset but also provides a greater understanding of the factors responsible for customer churn.

Exploratory Data Analysis (EDA)

EDA is a preliminary step to help understand the dataset, identify patterns, and extract insights that will guide further analysis. Let's run several analyses and visualizations to explore the data well.

Summary Statistics

Provides a brief gist of the main features of the dataset—measures of central tendency and variability.

```
# Display summary statistics for numerical features
print(df.describe())

# Display summary statistics for categorical features
print(df.describe(include=['object']))
```

- **Summary Statistics for Numerical Features**:

```
       SeniorCitizen       tenure  MonthlyCharges
count    7043.000000  7043.000000     7043.000000
mean        0.162147    32.371149       64.761692
std         0.368612    24.559481       30.090047
min         0.000000     0.000000       18.250000
25%         0.000000     9.000000       35.500000
50%         0.000000    29.000000       70.350000
75%         0.000000    55.000000       89.850000
```
```

```
max         1.000000    72.000000      118.750000
```

- **Summary Statistics for Categorical Features**:

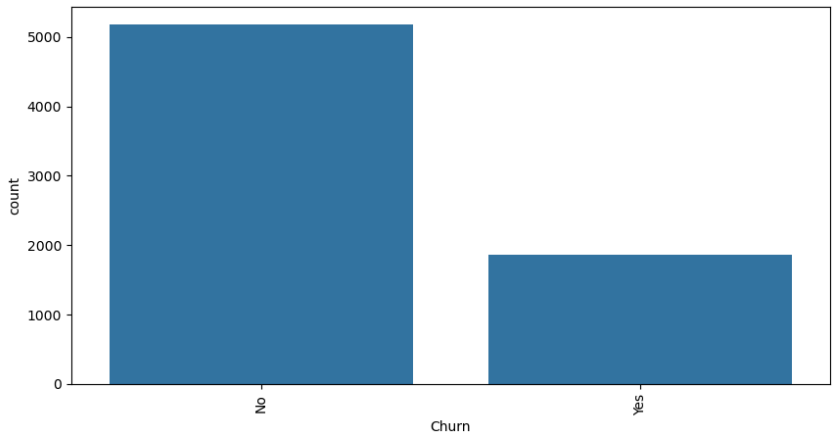
```
       gender Dependents PhoneService MultipleLines InternetService        Contract Churn
count    7043       7043         7043          7043            7043            7043  7043
unique      2          2            2             2               2               3     2
top      Male         No          Yes            No             DSL  Month-to-month    No
freq     3555       4933         6361          4072            3947            3875  5174

```

Target Variable Analysis

Analyzing the target variable, `Churn`, helps understand its distribution and the proportion of customers who churned versus those who did not.

```
# Plot the distribution of the target variable 'Churn'
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Churn')
plt.title('Distribution of Churn')
plt.savefig('churn_distribution.png')
plt.show()
```

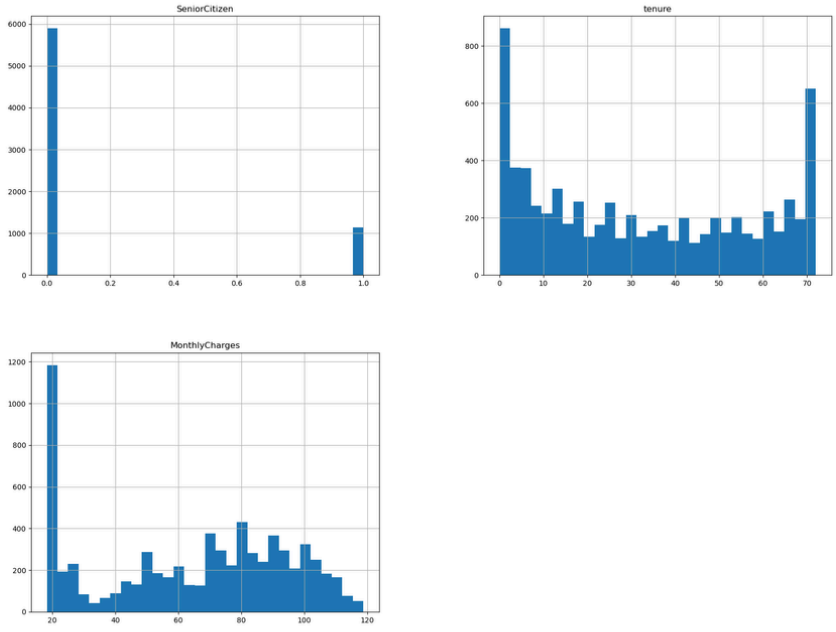


Distribution of Churn

Numerical Feature Distributions

Examining the distributions of numerical features helps identify patterns, outliers, and the spread of data.

```
# Plot histograms for numerical features
df.hist(bins=30, figsize=(20, 15))
plt.savefig('numerical_distributions.png')
plt.show()
```

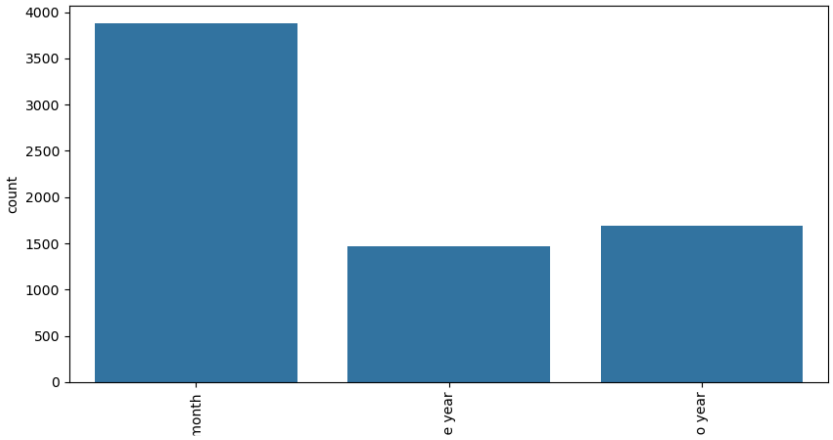


Distribution of Numerical Features

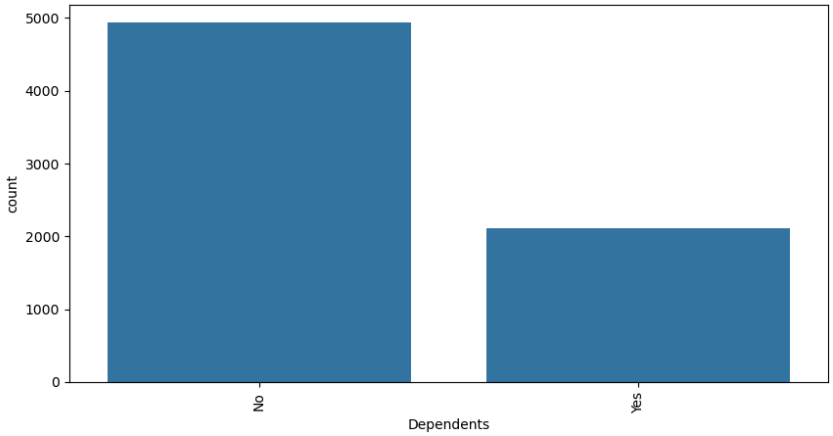
Categorical Feature Distributions

Analyzing categorical feature distributions provides insights into the frequency and distribution of different categories within each feature.

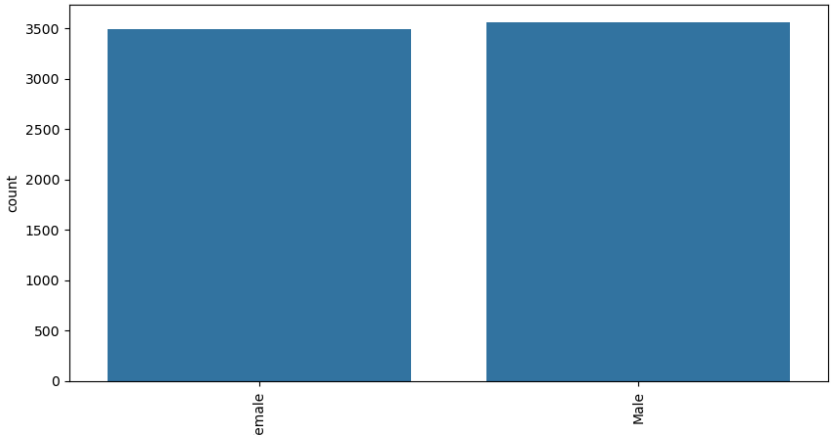
```
# Plot bar plots for categorical features
categorical_columns = ['PhoneService', 'MultipleLines', 'InternetService', 'Contract', 'Dependents', 'gender']
for column in categorical_columns:
    plt.figure(figsize=(10, 5))
    sns.countplot(data=df, x=column)
    plt.xticks(rotation=90)
    plt.savefig(f'{column}_distribution.png')
    plt.show()
```



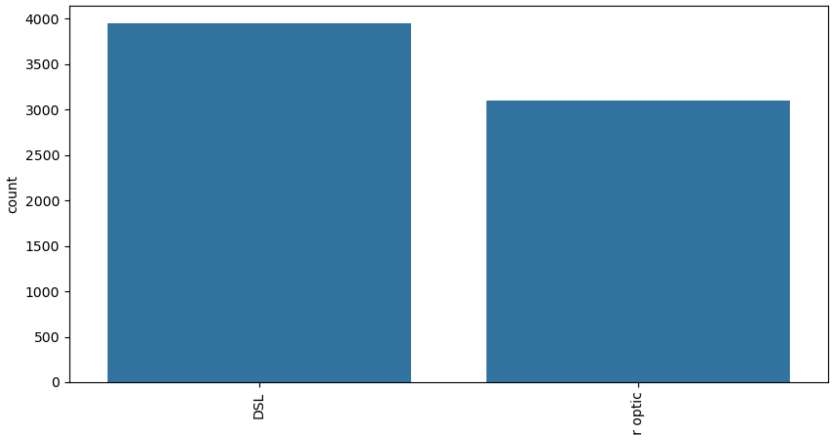
Contract distribution



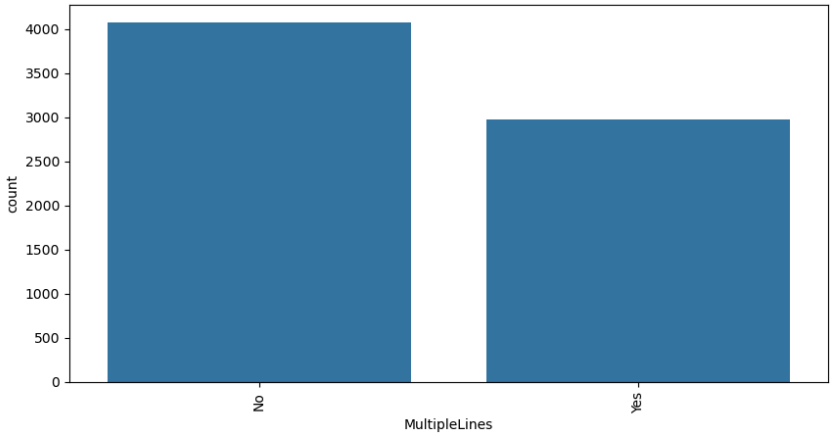
Dependents distribution



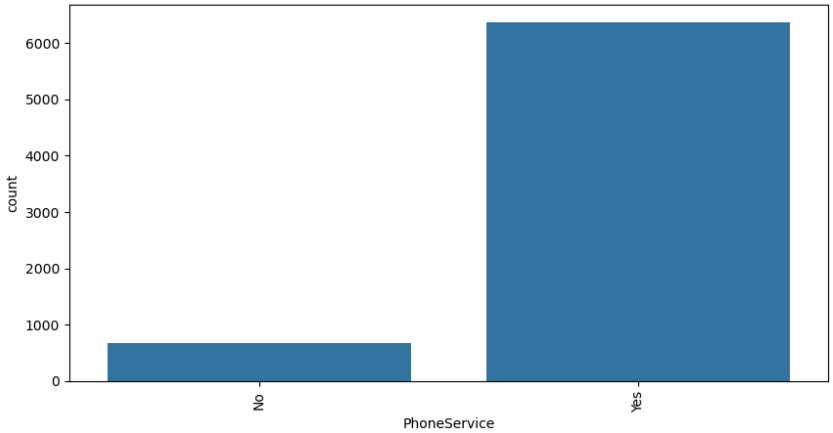
gender distribution



InternetService distribution



MultipleLines distribution

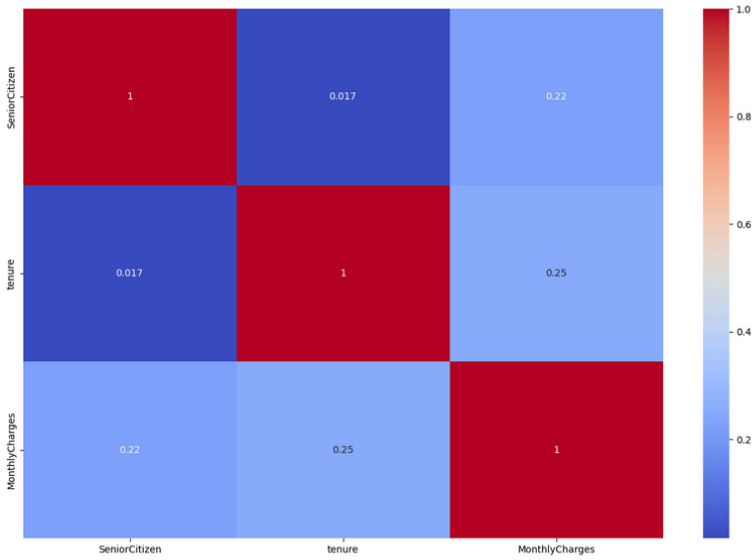


PhoneService distribution

Correlation Analysis

Correlation analysis identifies relationships between numerical features and can reveal significant associations.

```
# Plot the correlation matrix
corr_matrix = df.corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.savefig('correlation_matrix.png')
plt.show()
```

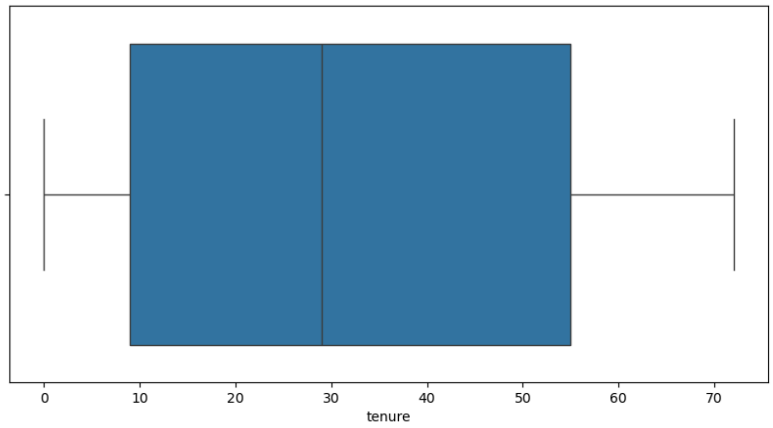


Correlation Matrix

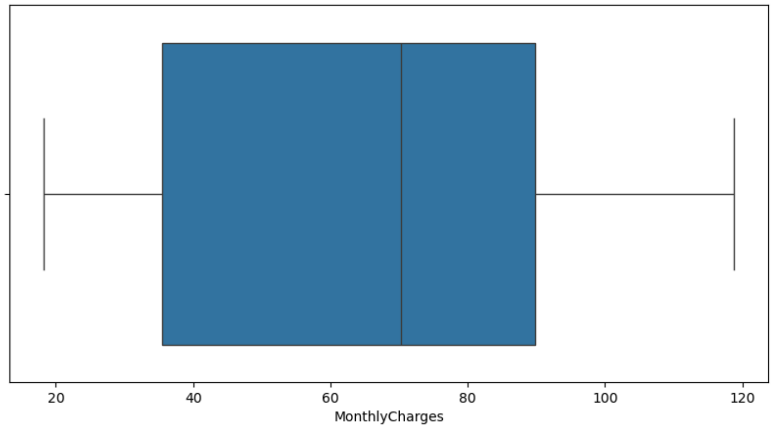
Outlier Detection

Detecting outliers in numerical features is crucial for understanding data variability and identifying potential anomalies.

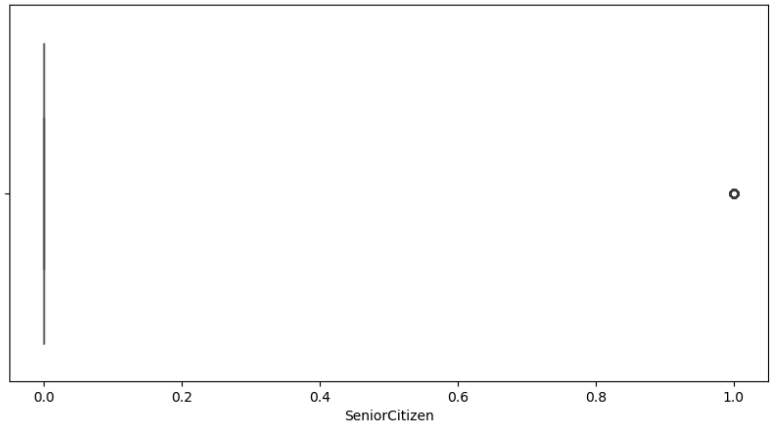
```
# Plot box plots for numerical features to identify outliers
numeric_columns = ['SeniorCitizen', 'tenure', 'MonthlyCharges']
for column in numeric_columns:
    plt.figure(figsize=(10, 5))
    sns.boxplot(data=df, x=column)
    plt.savefig(f'{column}_outliers.png')
    plt.show()
```



tenure outliers



MonthlyCharges outliers

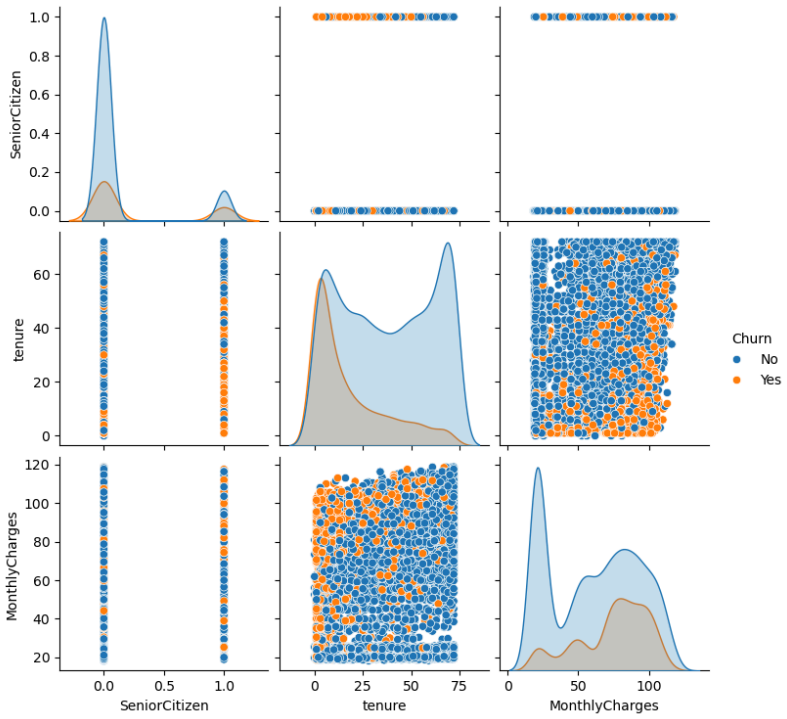


SeniorCitizen outliers

Pairwise Relationships

Visualizing the pairwise relationships helps to understand the interactions between different features and their combined effect on the target variable.

```
# Pairwise plots with numeric variables as hue
sns.pairplot(df, hue='Churn', vars=numeric_columns)
plt.savefig('pair_plots.png')
plt.show()
```



Pairwise Relationships among Numeric Variables

EDA Summary

We have received key insights about the structure of the data, their distribution, and relationships. All of this is important information towards the Feature Selection and Modeling stage of the project.

Data visualization and analysis allow us to identify important patterns and potential predictors related to customer churn. The results we obtain from the EDA help us formulate further steps in developing a robust predictive model to mitigate customer churn at Advanced Consultant Service.

Conclusion

Summary of Findings

The exploratory data analysis (EDA) on the customer churn dataset was performed to identify:

1. **Target Variable Distribution** Analysis of the target variable, Churn, showed that there exists a vast number of customers who are non-churners. There is some class imbalance in this data, so during modeling, one needs to be careful that the model does not become biased due to imbalanced classes.

2. **Numerical Features**

- **SeniorCitizen:** The majority of customers are not senior citizens.
- **Tenure:** Tenure is widely distributed as most customers have rather short or medium tenures. The vast differences in monthly charges suggest this is a combination of customer plans and usage levels.

3. **Categorical Features**

- **PhoneService:** Most customers have phone service.
- **MultipleLines:** A large number of customers do not have multiple lines.
- **InternetService:** Customers are evenly split between DSL and Fiber optic, with a small proportion having no internet service.
- **Contract:** Month-to-month contracts are the most common, followed by one-year and two-year contracts.
- **Dependents:** Many customers do not have dependents.
- **Gender:** The dataset has an almost equal number of male and female customers.

4. **Correlation Analysis** There are notable correlations between several numerical features. For example, TotalCharges is highly correlated with MonthlyCharges and tenure, as expected.
5. **Outlier Detection** Outliers were identified in features such as MonthlyCharges and tenure. These outliers need to be carefully considered during model training to ensure they do not adversely affect the model's performance.
6. **Pairwise Relationships** The pairwise plots did a very good job at revealing the underlying relationships between features and how only in tandem these affected churn. For example, customers with higher tenure and monthly charges may show different churning behavior than others.

Implications for Further Analysis

The above EDA insights can become the stepping-off point for additional next steps in customer churn prediction:

1. **Feature Selection & Engineering:** Looking at both our analytic findings and domain knowledge, it is evident that we should consider features like tenure, MonthlyCharges, Contract_Type (e.g., two-year versus month-to-month), and InternetService as the most important predictors of churn. Further feature engineering, such as adding interaction terms or polynomial features, would likely help the model perform even better.
2. **Class Imbalance:** Since the target variable Churn is imbalanced, we may need to deal with this by using methods such as oversampling the minority class, undersampling the majority class, or employing weight-based parameter settings in modeling algorithms.
3. **Modeling:** Here, we will choose the machine learning algorithms to make predictions with and train accordingly. Deciding on the algorithms should take into consideration what is known about both the data and the problem—for example, whether interpretability or non-linear relationships need to be considered.
4. **Treatment of Outliers:** Outliers detected during your EDA should be treated adequately. If necessary, we can use robust scaling, transformation techniques, or even exclusions (when justified) to prevent these variables from hurting the predictive power of your model.
5. **Validation & Testing:** A validation and test strategy should be put in place to understand how well the model works. Cross-validation, in conjunction with a hold-out test set to evaluate the model on unseen data, will guard against overfitting to training instances.

Following these suggestions will enable us to develop a solid predictive model that can allow Advanced Consultant Service to identify at-risk customers and deploy retention strategies effectively to decrease churn rates. These are the key takeaways from this EDA, which will guide the later stages of the project and help maintain a data-driven approach for customer retention.

References

The following resources were used during the Exploratory Data Analysis (EDA) process for the Customer Churn Analysis project:

1. **Pandas Documentation**: Used for data manipulation and analysis, including functions such as `read_csv`, `describe`, and DataFrame operations.
   - McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference. Retrieved from pandas documentation — pandas 2.2.2 documentation
2. **Matplotlib Documentation**: Utilized for creating static, interactive, and animated visualizations in Python.
   - Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering, 9(3), 90-95. Retrieved from Using Matplotlib — Matplotlib 3.9.2 documentation
3. **Seaborn Documentation**: Used for making statistical graphics, including count plots, histograms, box plots, pair plots, and heatmaps.
   - Waskom, M. L. (2021). *Seaborn: Statistical Data Visualization*. Journal of Open Source Software, 6(60), 3021. Retrieved from seaborn: statistical data visualization — seaborn 0.13.2 documentation
4. **Scikit-learn Documentation**: Employed for data preprocessing tasks such as imputation and encoding, as well as feature scaling.
   - Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from scikit-learn: machine learning in Python — scikit-learn 1.5.1 documentation
5. **Python Official Documentation**: Referenced for general Python programming practices and functionalities.
   - Python Software Foundation. (2021). *Python Language Reference, version 3.9*. Retrieved from 3.12.6 Documentation
6. **Project Scripts and Utilities**: Custom scripts and utilities developed for the project, including:
   - Chaudhary, B. (2024). *Custom Data Preprocessing and EDA Scripts*. Advanced Consultant Service Internal Documentation.

These references and resources were integral to the successful completion of the EDA process, providing the necessary tools and frameworks to analyze and visualize the customer churn dataset effectively.

Appendix

Additional Code Snippets

This section provides additional code snippets used during the data preprocessing and exploratory data analysis phases. These code snippets illustrate key operations and functions implemented to clean, preprocess, and analyze the data.

A. Data Loading and Initial Inspection

The following code snippet demonstrates how the dataset was loaded and an initial inspection was performed to understand the data structure and types.

```
import pandas as pd

# Load the dataset
file_path = 'data/raw/Dataset (ATS)-1.csv'
df = pd.read_csv(file_path)

# Display basic information about the dataset
print(df.info())

# Display the first few rows of the dataset
print(df.head())
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 10 columns):
```

```
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   gender           7043 non-null   object
 1   SeniorCitizen    7043 non-null   int64
 2   Dependents       7043 non-null   object
 3   tenure           7043 non-null   int64
 4   PhoneService     7043 non-null   object
 5   MultipleLines    7043 non-null   object
 6   InternetService  7043 non-null   object
 7   Contract         7043 non-null   object
 8   MonthlyCharges   7043 non-null   float64
 9   Churn            7043 non-null   object
dtypes: float64(1), int64(2), object(7)
memory usage: 550.4+ KB
```

B. Data Cleaning Process

The following code snippets illustrate how missing values were handled and categorical variables were encoded.

1. **Handling Missing Values**:

```python
from sklearn.impute import SimpleImputer

# Handle missing values by imputing mean values for numeric columns
imputer = SimpleImputer(strategy='mean')
df_numeric = df.select_dtypes(include=['float64', 'int64'])
df_numeric_imputed = pd.DataFrame(imputer.fit_transform(df_numeric), columns=df_numeric.columns)
df[df_numeric.columns] = df_numeric_imputed
```

2. **Encoding Categorical Variables**:

```python
from sklearn.preprocessing import OneHotEncoder

# Encode categorical variables using OneHotEncoder
encoder = OneHotEncoder(sparse=False, handle_unknown='ignore')
categorical_columns = df.select_dtypes(include=['object']).columns
encoded_data = pd.DataFrame(encoder.fit_transform(df[categorical_columns]),
columns=encoder.get_feature_names_out(categorical_columns))
df = df.drop(columns=categorical_columns)
df = pd.concat([df, encoded_data], axis=1)
```

C. Feature Engineering

The following code snippets show how new features were created to enhance the dataset's predictive power.

1. **Charges Per Tenure**:

```python
# Create a new feature 'Charges_Per_Tenure'
df['Charges_Per_Tenure'] = df['MonthlyCharges'] / (df['tenure'] + 1)
```

2. **Total Charges**:

```python
# Recalculate the 'TotalCharges' feature
df['TotalCharges'] = df['MonthlyCharges'] * df['tenure']
```

3. **Contract Type Encoding**:

```python
# Encode the 'Contract' feature
contract_mapping = {'Month-to-month': 0, 'One year': 1, 'Two year': 2}
df['Contract_Type'] = df['Contract'].map(contract_mapping)
```

4. **Payment Method Encoding**:

```
# Encode the 'PaymentMethod' feature
payment_mapping = {'Electronic check': 0, 'Mailed check': 1, 'Bank transfer (automatic)': 2, 'Credit card (automatic)': 3}
df['Payment_Method'] = df['PaymentMethod'].map(payment_mapping)
```

D. Exploratory Data Analysis (EDA)

The following code snippets illustrate key visualizations used in the EDA process.

1. **Distribution of Churn**:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Plot the distribution of the target variable 'Churn'
plt.figure(figsize=(10, 5))
sns.countplot(data=df, x='Churn')
plt.title('Distribution of Churn')
plt.savefig('churn_distribution.png')
plt.show()
```

2. **Numerical Feature Distributions**:

```
# Plot histograms for numerical features
df.hist(bins=30, figsize=(20, 15))
plt.savefig('numerical_distributions.png')
plt.show()
```

3. **Categorical Feature Distributions**:

```
# Plot bar plots for categorical features
categorical_columns = ['PhoneService', 'MultipleLines', 'InternetService', 'Contract', 'Dependents', 'gender']
for column in categorical_columns:
    plt.figure(figsize=(10, 5))
    sns.countplot(data=df, x=column)
    plt.xticks(rotation=90)
    plt.savefig(f'{column}_distribution.png')
    plt.show()
```

4. **Correlation Matrix**:

```
# Plot the correlation matrix
corr_matrix = df.corr()
plt.figure(figsize=(15, 10))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.savefig('correlation_matrix.png')
plt.show()
```

5. **Outlier Detection**:

```
# Plot box plots for numerical features to identify outliers
numeric_columns = ['SeniorCitizen', 'tenure', 'MonthlyCharges']
for column in numeric_columns:
    plt.figure(figsize=(10, 5))
    sns.boxplot(data=df, x=column)
```

```
    plt.savefig(f'{column}_outliers.png')
    plt.show()
```

6. **Pairwise Relationships**:

```
# Plot pair plots for numerical features with the target variable 'Churn' as hue
sns.pairplot(df, hue='Churn', vars=numeric_columns)
plt.savefig('pair_plots.png')
plt.show()
```

These code snippets illustrate the critical steps and operations performed during the data preprocessing and exploratory data analysis phases. By including these snippets in the appendix, we provide a detailed reference for the methods and techniques used in this project.