Telco Customer Churn

Exploratory Data Analysis (EDA)

Ringga Prasetya Al Muthasyr

Data Science Portfolio

Content

1. Dataset Information & Objectives
2. Preliminary Look & Data Cleansing
3. Data Understanding
4. EDA Questions
5. Conclusions & Recommendations

Dataset Information

&

Objectives

Dataset Information

The dataset contains **customer information** from telco company about services that each customer has signed up for, customer account information, customer demographic, and **customers who left within the last month**.
(source : https://www.kaggle.com/datasets/blastchar/telco-customer-churn)

Data Dictionary

Churn	Demographic	Account Information	Services
Customers who left within the last month	Gender	Tenure	Phone
	SeniorCitizen	Contract	Multiple lines
	Partners	Payment Method	Internet
	Dependents	Paperless Billing	Online security
		Monthly Charges	Online backup
		Total Charges	Device protection
			Tech support
			Streaming TV
			Streaming movies

Objectives

This project will help answer:

1. Who is the customer with the most money spent?
2. Who is the churn customer with the most money spent?
3. Customer churn relationship with other variables?

Preliminary Look

&

Data Cleansing

Check Missing Values & Duplicated Value

```
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
 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)
```

```
[13] df.duplicated().sum()
```

```
0
```

Observation :

- There are 7043 rows and 21 columns in this dataset
- There are no missing values in each column
- There are no duplicated value in each column
- The data types are good, except for the "TotalCharges" column which is an object. Need to change to numeric data type (Float)

Data Preprocessing

Change "TotalCharges" to numeric data type (Float)

```
[9]  #exclude rows with total charges columncontain white space
     df = df.loc[~df['TotalCharges'].str.contains(' ')]
```

```
[10] #totalcharges to float
     df['TotalCharges'] = df['TotalCharges'].astype(float)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7032 entries, 0 to 7042
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)
```

We can see the column "TotalCharges" has changed to numeric data type (float)

Data Understanding

Numerical Statistical Summary

Observation:

- Overall, the minimum and maximum values make sense for each column
- "SeniorCitizen" column is boolean/binary column since the value is 0 or 1, no need to conclude its simmetricity.
- Mean ~ 50% (Median) in "tenure", "MonthlyCharges", and "TotalCharges" column, indicating somewhat a skew distribution

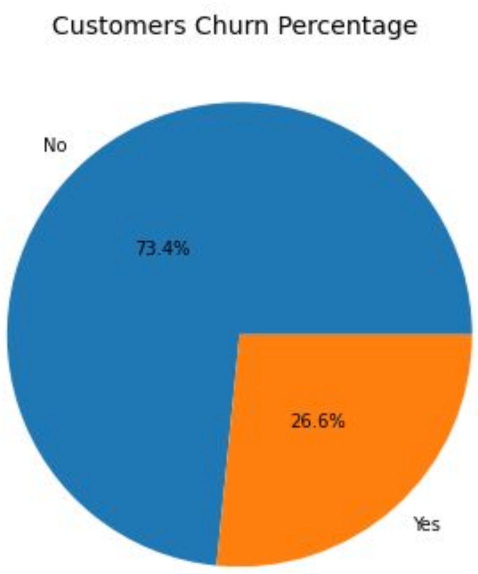
```
# group column names based on type

cats = ['customerID','gender','Partner','Dependents','PhoneService',
        'MultipleLines','InternetService','OnlineSecurity','OnlineBackup',
        'DeviceProtection','TechSupport','StreamingTV','StreamingMovies',
        'Contract','PaperlessBilling','PaymentMethod','Churn']

nums = ['SeniorCitizen','tenure','MonthlyCharges','TotalCharges']
```

```
# numerical statistical sumary
df[nums].describe()
```

	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

Categorical Statistical Summary

Top Value for each category	
Top Gender : Male (3549)	Device protection : No (3094)
Partners : No (3639)	Tech support : No (3472)
Dependents : No (4933)	Streaming TV : No (2809)
Phone Service : Yes (6352)	Streaming movies : No (2781)
Multiple lines : No (3385)	Contract : Month-to-month (3875)
Internet Service : Fiber optic (3096)	Paperless Billing : Yes (4168)
Online security : No (3497)	Payment Method : Electronic check (2365)
Online backup : No (3087)	Churn : No (5163)



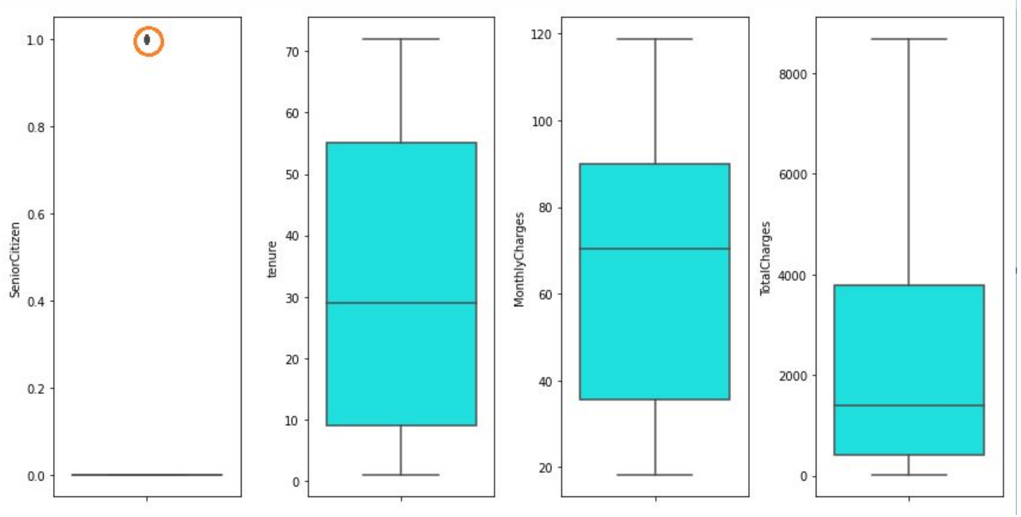
26.6% of customers has churn

Boxplot to detect outliers (1/2)

An *outlier* is an observation that lies an abnormal distance from other values in a random sample from a population.They can negatively affect the statistical analysis and the training process of a machine learning algorithm resulting in lower accuracy.

```
[59] # adjust the figure size for better readability
     plt.figure(figsize=(12,6))

     # plotting
     features = nums
     for i in range(0, len(features)):
         plt.subplot(1, len(features), i+1)
         sns.boxplot(y=df[features[i]], color='cyan')
         plt.tight_layout()
```

Boxplot to detect outliers (2/2)



Observation:

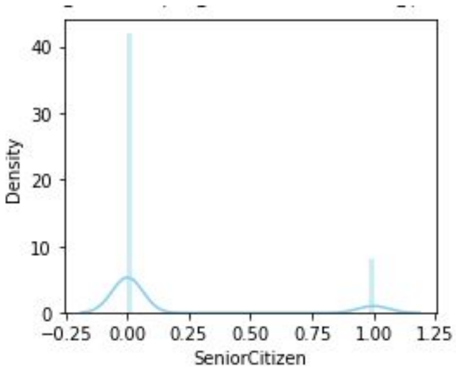
1. There is outlier in the "SeniorCitizen" column (value = 1). But because this column is boolean, it doesn't need to be considered
2. Column with continuous value does not have data outliers

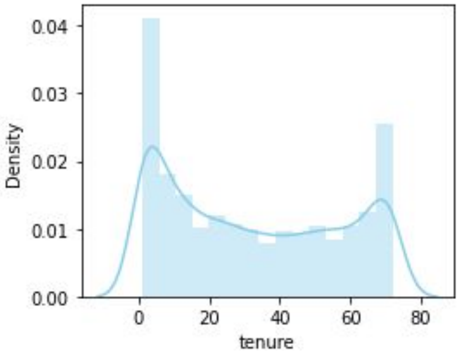
KDE plot for knowing the distribution form (1/2)

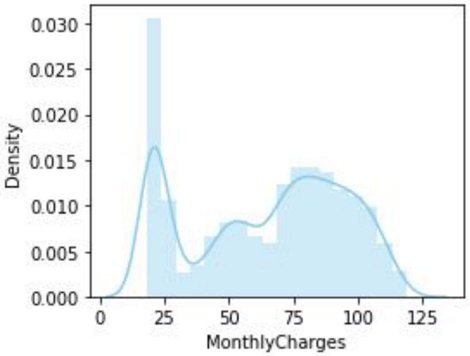
A kernel density estimate (KDE) plot is a method for visualizing the distribution of observations in a dataset, analagous to a histogram. KDE represents the data using a continuous probability density curve in one or more dimensions.

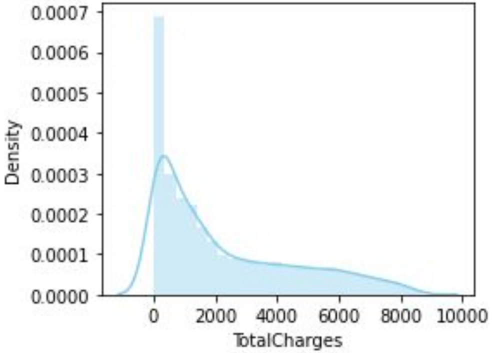
```
[61] plt.figure(figsize=(12,6))

     features = nums
     for i in range(0, len(features)):
         plt.subplot(2, len(features)//2 + 1, i+1)
         #plt.subplot(1, len(features), i+1)
         sns.distplot(x=df[features[i]], color='skyblue')
         plt.xlabel(features[i])
         plt.tight_layout()
```

KDE plot for knowing the distribution form (2/2)





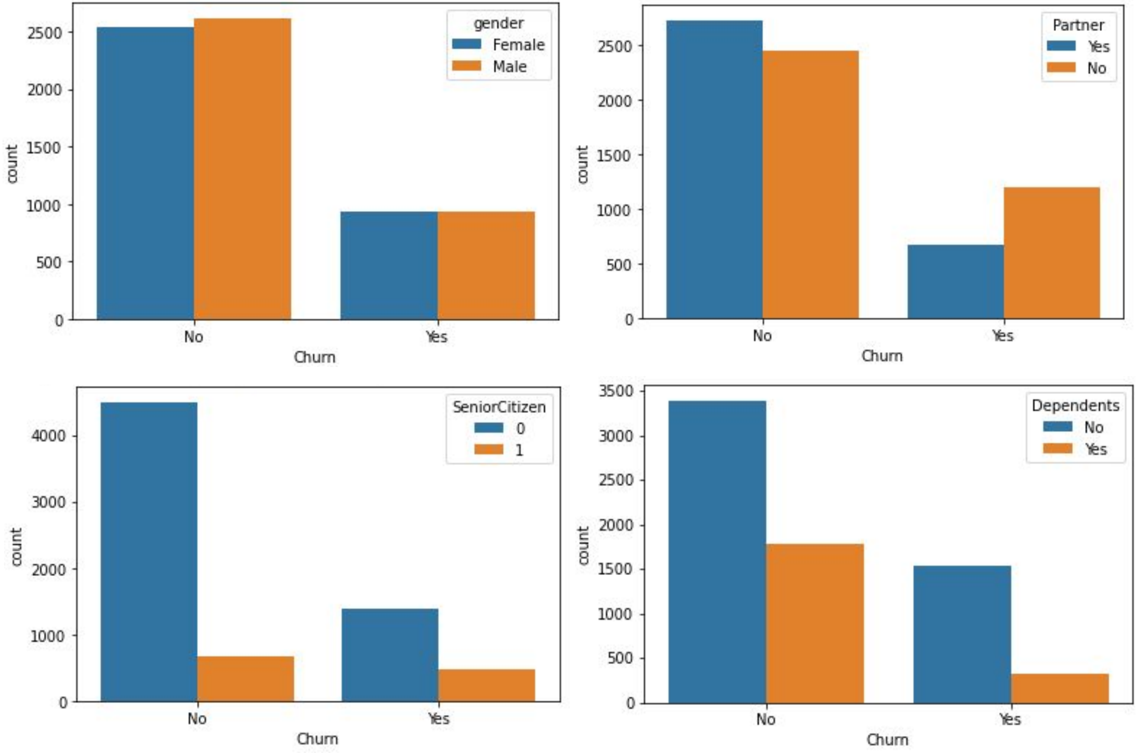




Observation:

- Continuous numeric features: “tenure”, “MonthlyCharges”, and “TotalCharges” are slightly skewed (need to change to approximate normal distribution if want to proceed to machine learning)
- “SeniorCitizen = 0” is more frequent in the data set.

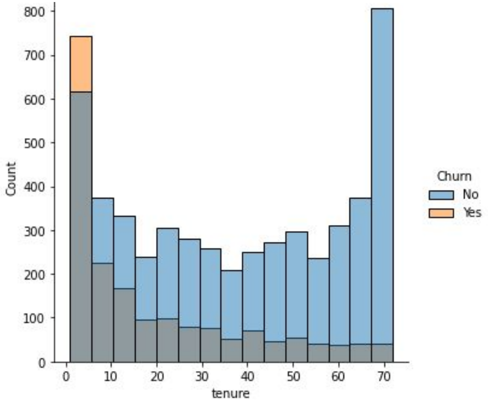
Churn x Demographic

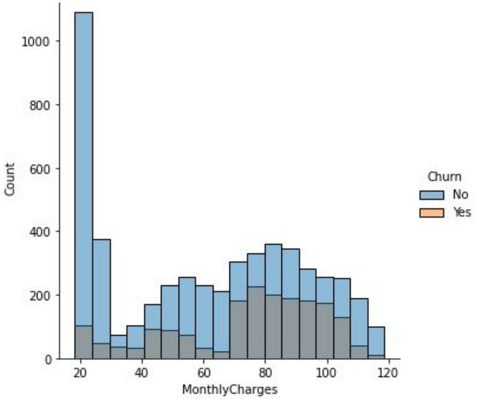


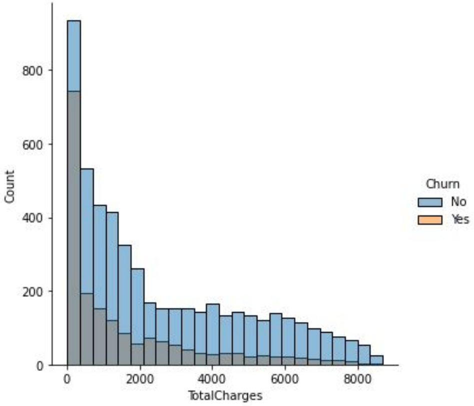
Observations :

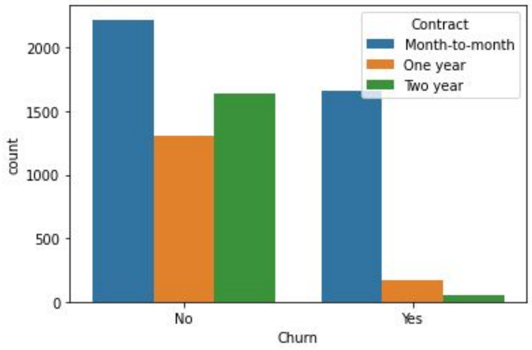
- Both Male & Female have the same tendency to leave, so neither one is dominant
- Customers who left are dominated by young people
- Even though it is dominated by young people, the percentage of customers left from the elderly is very significant
- Customers who do not have a partner left very significant
- Customers without dependents left very significant

Churn x Account Information





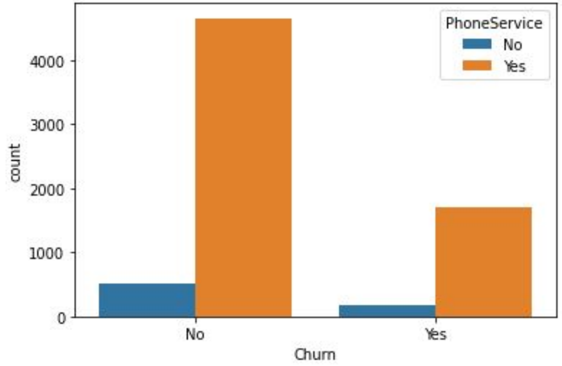


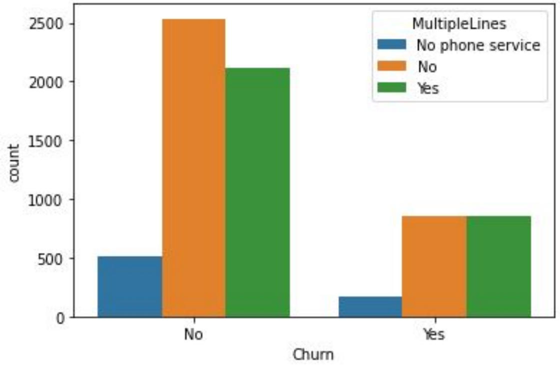


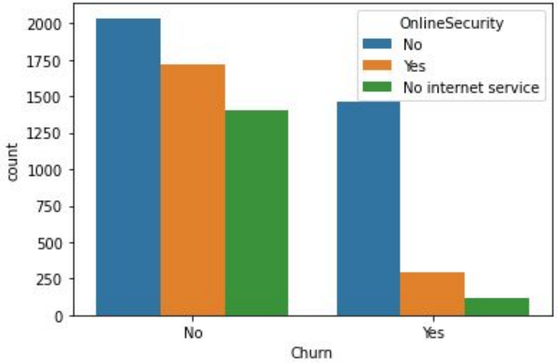
Observations :

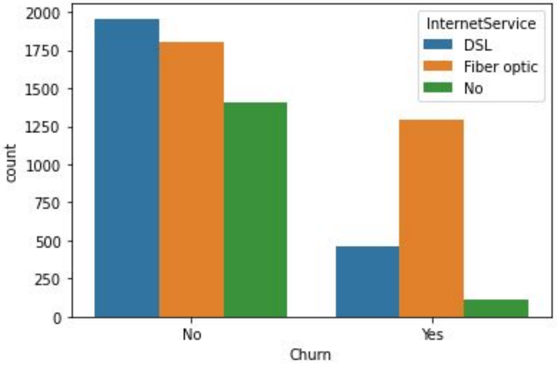
- Customers with longer tenure tend to stay, which also affects Total Charges and Monthly Charges
- Customers with Month-to-month contracts leave significantly

Churn x Services (1/2)





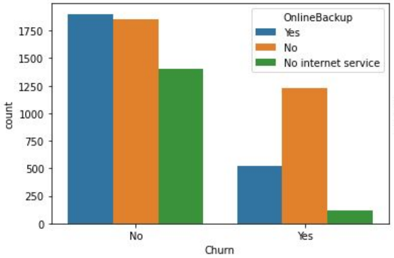


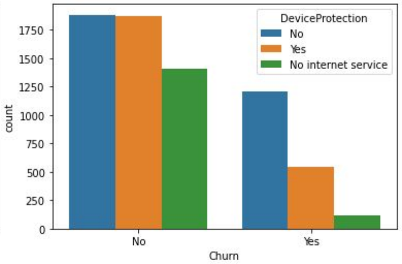


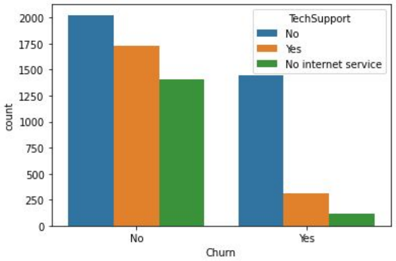
Observations

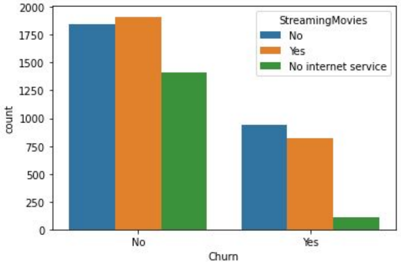
- Customers who use Telephone Service have left significantly
- Customers who use the Multiple Lines service and do not use the Multiple Lines service have left significantly
- Customers using fiber optic services have left significantly
- Customers who do not have Online Security services have left significantly

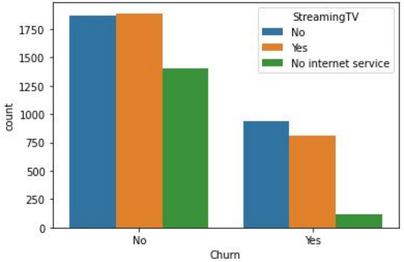
Churn x Services (2/2)











Observations :

- Customers who do not use the Online Backup service have left significantly
- Customers who do not use Device Protection services have left significantly
- Customers who do not use Tech Support services have left significantly
- Customers who have left is dominated by those who do not have StreamingTV services.
- However, it can also be seen that customers using Streaming TV services have left very significant
- Customers who left are dominated by customers who do not use streaming movies services
- However, it can also be seen that customers using Streaming Movies service have left significantly

Correlation Heatmap

```
# correlation heatmap
correlation = df.corr()
sns.heatmap(correlation, annot=True, fmt='.2f')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f3daf7d92d0>
```

Observation :

TotalCharges and tenure are highly correlated each other

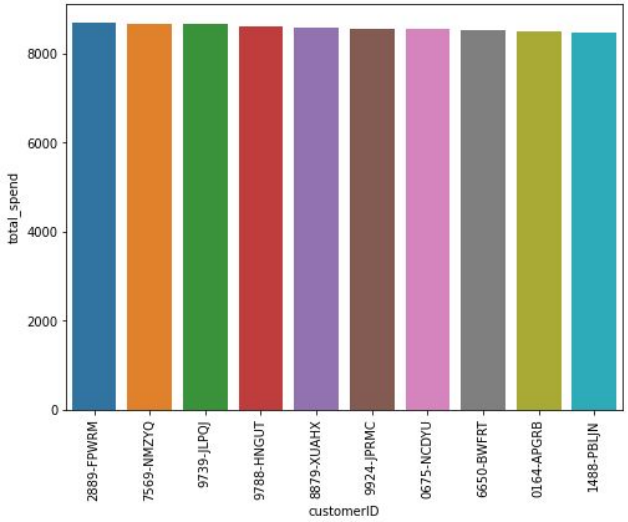
EDA Questions

Who are Top 10 Customers with the most money spend?

```
[ ] # group by aggregation
    top10_cust = (df
                  .groupby('customerID')
                  .agg(total_spend=('TotalCharges','sum'))
                  .reset_index()
                  .sort_values('total_spend',ascending=False)
                  .head(10)
                  )

    top10_cust
```

```
# visualize it
plt.figure(figsize=(8,6))
top10_cust['customerID'] = top10_cust['customerID'].astype(str)
sns.barplot(data=top10_cust, x='customerID', y='total_spend')
plt.xticks(rotation=90)
```

	customerID	total_spend
2000	2889-FPWRM	8684.80
5350	7569-NMZYQ	8672.45
6844	9739-JLPQJ	8670.10
6881	9788-HNGUT	8594.40
6264	8879-XUAHX	8564.75
6982	9924-JPRMC	8547.15
462	0675-NCDYU	8543.25
4710	6650-BWFRT	8529.50
95	0164-APGRB	8496.70
1028	1488-PBLJN	8477.70



Observation :

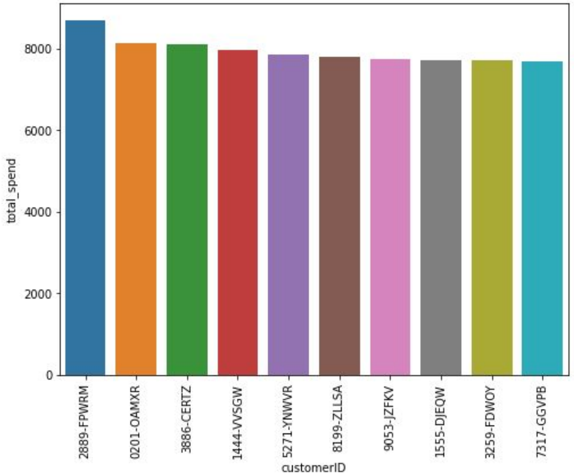
The top 10 customers with the most money spent don't have much difference in spending money (almost the same)

Who are Top 10 Customers Churn with the most money spend?

```python
# group by aggregation
top10_churn = (data_churn
               .groupby('customerID')
               .agg(total_spend=('TotalCharges','sum'))
               .reset_index()
               .sort_values('total_spend',ascending=False)
               .head(10)
               )

top10_churn
```

```python
# visualize it
plt.figure(figsize=(8,6))
top10_churn['customerID'] = top10_churn['customerID'].astype(str)
sns.barplot(data=top10_churn, x='customerID', y='total_spend')
plt.xticks(rotation=90)
```

	customerID	total_spend
535	2889-FPWRM	8684.80
28	0201-OAMXR	8127.60
736	3886-CERTZ	8109.80
262	1444-VVSGW	7968.85
1022	5271-YNWVR	7856.00
1566	8199-ZLLSA	7804.15
1707	9053-JZFKV	7752.30
283	1555-DJEQW	7723.90
617	3259-FDWOY	7723.70
1401	7317-GGVPB	7690.90



Observation :

It can be seen that Top 1 customer with the most money spent (2889-FPWRM) has churn

Conclusions

&

Recommendations

Conclusions

Demographic

- Quantitatively, customers churn are dominated by young generations. However, the percentage of elderly customers who left is very significant
- Customers who do not have a partner has left very significant
- Customers without dependents has left significantly

Account Information

- Customers with longer tenure tend to stay, which also affects Total Charges and Monthly Charges
- Customers with Month-to-month contracts leave significantly

Services

- The majority of customers who use the service has left significantly

Recommendations

Demographic

- Evaluate whether the product service is in line with the current trend to avoid losing customers from the younger generation.
- Develop convenient telecommunication services for elderly customers

Account Information

- Provide special benefits for customers who has a long tenure and campaign to all other customers so that they are interested so as to minimize customers churn
- Provide special services and prices for 1 year and 2 year contracts

Services

- Evaluate and improve all existing services so the customers are satisfied and prevent customers from leaving

Thanks!

ringgaislam@gmail.com

https://github.com/RinggaMuthasyr

https://www.linkedin.com/in/ringga-prasetya/