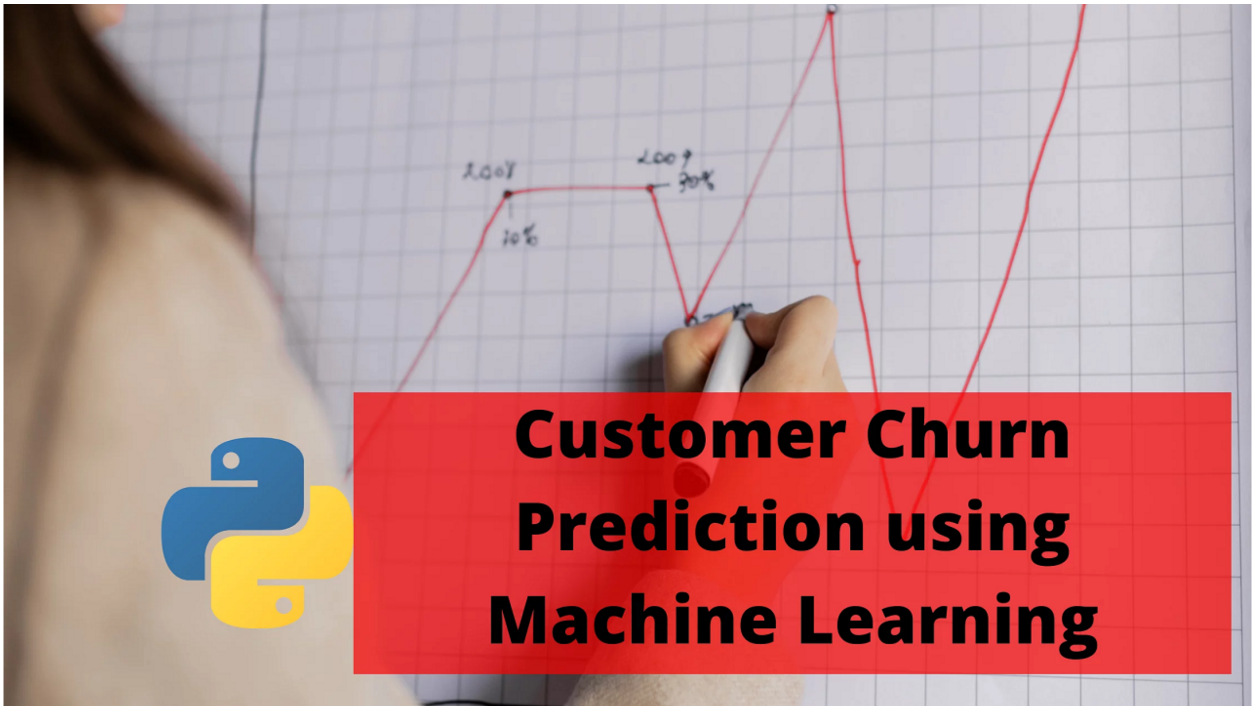
DATA ANALYTICS WITH COGNOS

CUSTOMER CHURN PREDICTION



TEAM MEMBER'S :

1. **SENTHIL VEL S (LEADER) 731221104034**
2. **VIGNESH M 731221104039**
3. **GUNASEKARAN S 731221104014**
4. **GUNASEKARAN P 731221104013**
5. **KAMALESH K 731221104017**

INTRODUCTION :

- Churn prediction means detecting which customers are likely to leave a service or to cancel a subscription to a service.
- It is a critical prediction for many businesses because acquiring new clients often costs more than retaining existing ones.
- Once you can identify those customers that are at risk of canceling, you should know exactly what marketing action to take for each individual customer to maximize the chances that the customer will remain.

Important of churn:

- Customer churn is a common problem across businesses in many sectors. If you want to grow as a company, **you have to invest in acquiring new clients**. Every time a client leaves, it represents a significant investment lost.
- Both time and effort need to be channeled into replacing them. Being able to predict when a client is likely to leave, and offer them incentives to stay, can offer huge savings to a business.

ANALYSIS OBJECTIVES:

OBJECTIVE 1: understanding churn pattern:

1. **Poor network quality:** Customers expect efficient, reliable, and far-reaching network coverage, and will stop at nothing to find the service provider that delivers these things. Some 45% of smartphone user churn is a result of problems with network quality.

2. **Complicated billing processes:**Telecom billing processes can be confusing and cumbersome. Customers who feel that they're paying too much and can't decipher why are more likely to leave.

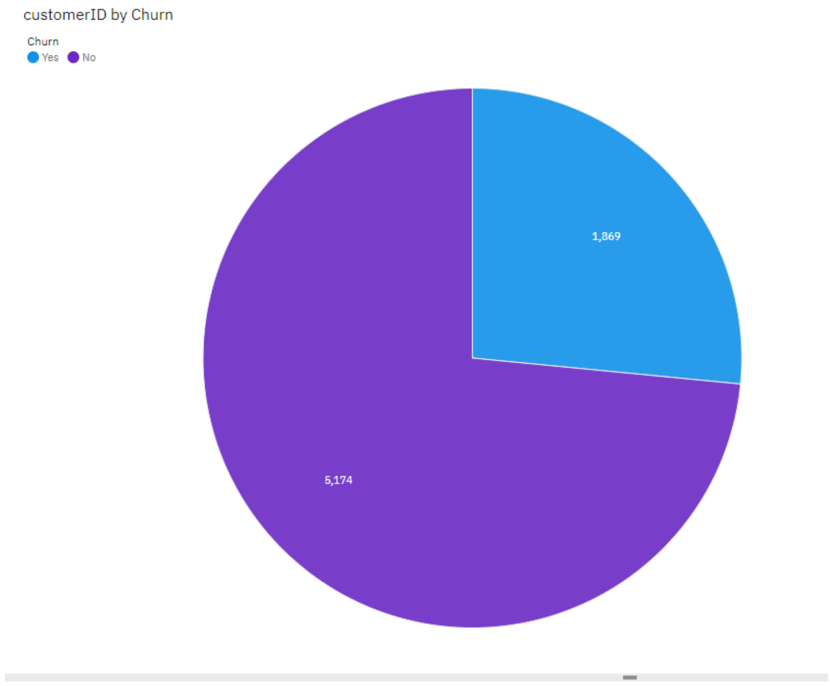
3. **Inefficient customer service:**Telecoms is a customer service-centric industry, which means that this area has to be a priority if companies are going to hang onto their customers. In fact, 39% of customers have admitted to leaving a telecoms company over poor customer service. Of these:

- 27% left because they felt like their time had been wasted
- 51% left because they had to call more than once to have an issue resolved
- 37% left because they felt that agents were rude, untrained or incompetent
- 41% left because they felt that the self-service options were inferior

4. **Offer a better deal.** Not all customers leave in a fit or fury, though. Many simply leave because another provider offers them a better deal. This has made innovation a critical customer retention tool.

5. **Low barriers to switching.** While this isn't necessarily a bad thing from a customer perspective, it can create serious problems for poor suppliers. Customers switch quickly and easily between service providers because they can. The barriers to switching are remarkably low.

VISUALIZATION:



OBJECTIVE 2: RETENTION RATE ANALYSIS

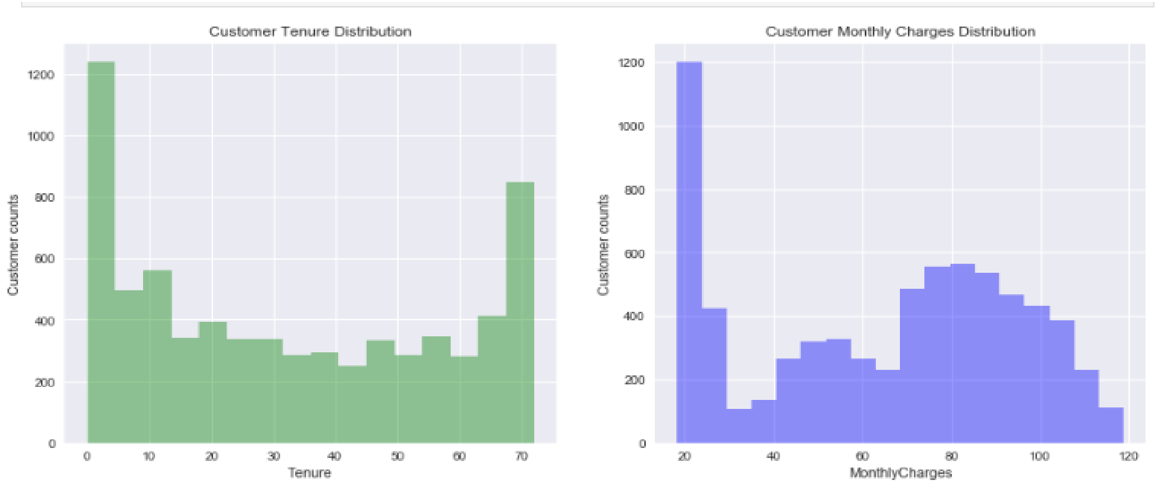
CORRELATION MATRICES:

Distribution of Tenure, Monthly Charges:

program:

```
f, axes = plt.subplots(ncols = 2, figsize = (15,6))

sns.distplot(df_cal.Tenure ,color='g'  ,kde = False  , ax = axes[0]
).set_title("Customer Tenure Distribution")

axes[0].set_ylabel("Customer counts")

sns.distplot(df_cal.MonthlyCharges  , color='b',kde =
False,ax=axes[1]).set_title("Customer Monthly Charges Distribution")

axes[1].set_ylabel("Customer counts")

plt.show()
```

OUTPUT:



Customer tenure:

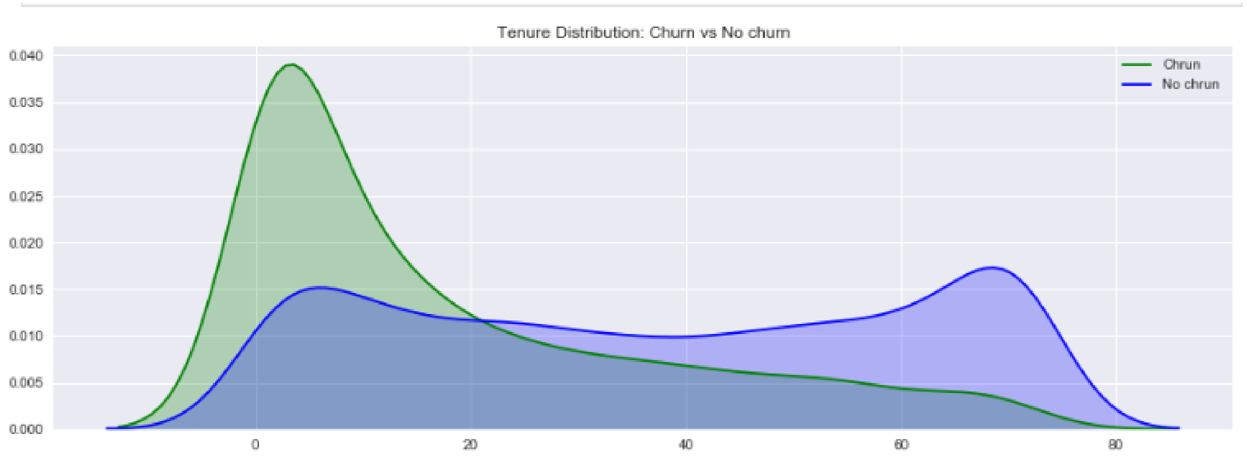
```
fig = plt.figure(figsize=(15,5))
```

ax = sns.kdeplot(df_cal.loc[(df_cal['Churn']==1),'Tenure'] , color = 'g' , shade = **True** , label='Chrun')

ax = sns.kdeplot(df_cal.loc[(df_cal['Churn']==0) ,'Tenure'] , color = 'b' , shade = **True** , label='No chrun')

plt.title("Tenure Distribution: Churn vs No churn")

plt.show()



Customer monthly charges:

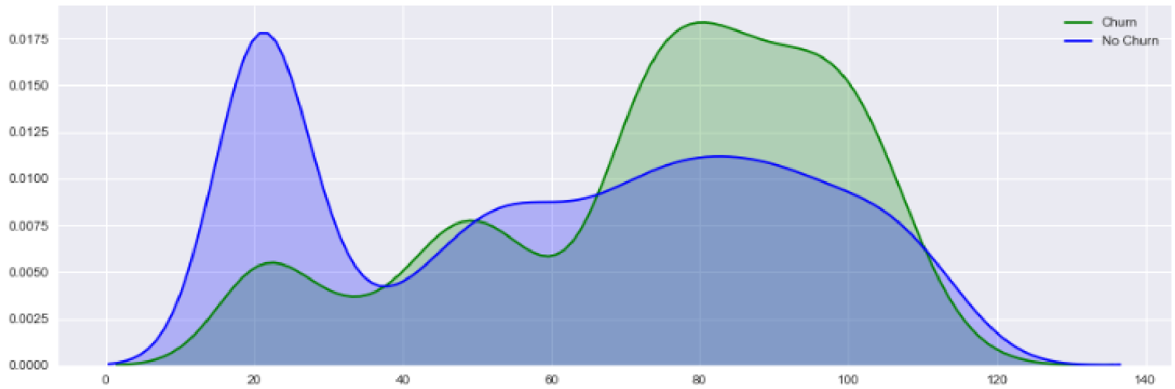
fig = plt.figure(figsize=(15,5))

ax = sns.kdeplot(df_cal.loc[(df_cal['Churn']==1),'MonthlyCharges'] , color = 'g' , shade = **True** , label='Churn')

ax = sns.kdeplot(df_cal.loc[(df_cal['Churn']==0),'MonthlyCharges'] , color = 'b , shade = **True** , label=('No Churn')

plt.show()

output:



OBJECTIVE 3: IDENTIFIER THE KEY CHURN FACTORS

Demographic analysis:

Here, we have gender, age bands (in terms of married, have children and senior citizen) related information.

1. **Less tenure and high monthly charge**

   More likely to be Female

2. **High tenure and High monthly charges**

   More likely to be male and senior citizen

3. **Less tenure and low monthly charges**

   More likely to be male

PROGRAM :

```
group_gp = df_cluster_gp['Gender'].value_counts(normalize=True).to_frame()
#pd.concat([group_gp.index.name , group_gp.values])
group_gp.columns = ['Count']
group_gp = group_gp.reset_index()
group_gp_new = group_gp.copy()
Group_gp_new
```

output:

	Cluster	Gender	Count
0	0	Female	0.531085
1	0	Male	0.468915
2	1	Male	0.519397
3	1	Female	0.480603
4	2	Male	0.535088
5	2	Female	0.464912

PROGRAM:

```
df_cluster_gender_gp =
df_cal[df_cal['Churn']==1].groupby(['Cluster','Gender'],as_index=False)
```

```
tenure_charges_gp =  df_cluster_gender_gp['Tenure','MonthlyCharges'].mean()

tenure_charges_gp
```

	Cluster	Gender	Tenure	MonthlyCharges
0	0	Female	8.418651	83.463889
1	0	Male	9.179775	84.226854
2	1	Female	45.699552	90.538117
3	1	Male	49.261411	90.909959
4	2	Female	7.231132	37.701651
5	2	Male	6.881148	38.901844

DATA COLLECTION :

PREPROCESSING THE DATA SET:

Perform various preprocessing tasks like handling missing values, removing duplicates, removing unnecessary columns, renaming columns, and encoding categorical variables if needed.

Import Necessary Libraries:

```
import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns
```

Loading data :

df_raw=pd.read_csv(r'C:\Users\admin\Downloads\WA_Fn-UseC_-Telco-Customer-Churn.csv')

df_raw.head()

]:

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	In
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	
4	9237-HQITU	Female	0	No	No	2	Yes	No	

5 rows × 21 columns

DATA CLEANING:

Row checking:

df_raw.shape

Output: (7043, 21)

Column checking:

df_raw.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')

Finding null:

```
df_raw.isnull().any()

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

D type: bool

Data cleaning :

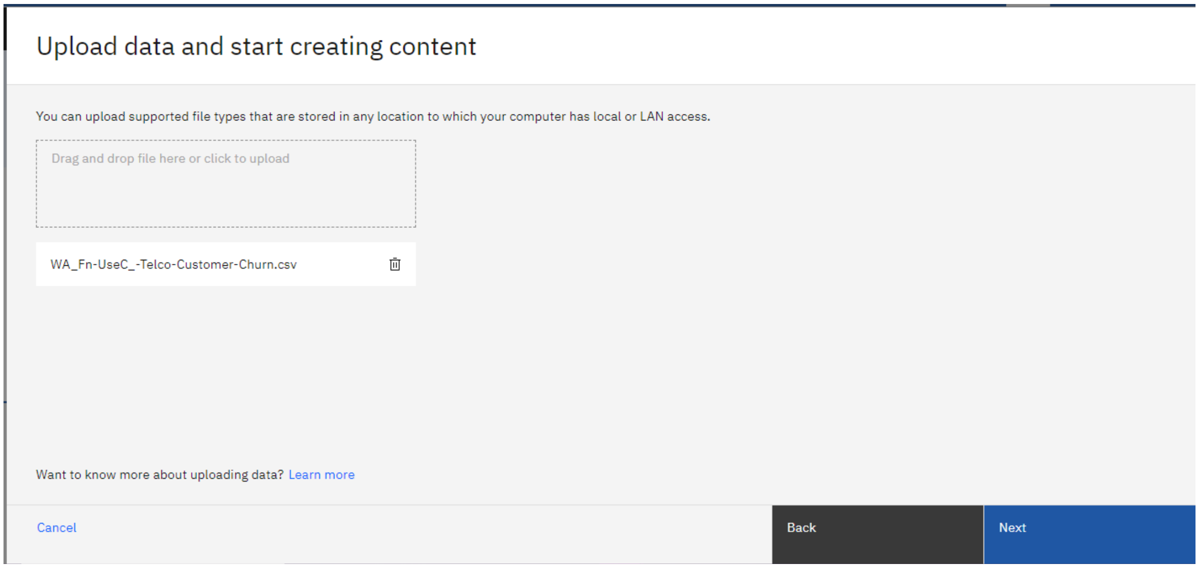
```
df_cal['Partner'] = df_cal.Partner.map({'Yes':1,'No':0})

df_cal['Dependents'] = df_cal.Dependents.map({'Yes':1,'No':0})

df_cal['PhoneService'] = df_cal.PhoneService.map({'Yes':1,'No':0})

df_cal['MultipleLines'] = df_cal.MultipleLines.map({'Yes':1,'No':0,'No phone service':0})

df_cal['InternetService'] = df_cal.InternetServiceType.map({'DSL':1,'Fiber optic':1,'No':0})

df_cal['OnlineSecurity'] = df_cal.OnlineSecurity.map({'Yes':1,'No':0,'No internet service':0})

df_cal['OnlineBackup'] = df_cal.OnlineBackup.map({'Yes':1,'No':0,'No internet service':0})

df_cal['DeviceProtection'] = df_cal.DeviceProtection.map({'Yes':1,'No':0,'No internet service':0})

df_cal['TechSupport'] = df_cal.TechSupport.map({'Yes':1,'No':0,'No internet service':0})

df_cal['StreamingTV'] = df_cal.StreamingTV.map({'Yes':1,'No':0,'No internet service':0})

df_cal['StreamingMovies'] = df_cal.StreamingMovies.map({'Yes':1,'No':0,'No internet service':0})

df_cal['IsContracted'] = df_cal.ContractType.map({'One year':1,'Two year':1,'Month-to-month':0})

df_cal['PaperlessBilling'] = df_cal.PaperlessBilling.map({'Yes':1,'No':0})

df_cal['Churn'] = df_cal.Churn.map({'Yes':1,'No':0})

df_cal.head()
```

	CustomerID	Gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	MultipleLines	InternetServiceType	OnlineSecurity
0	7590-VHVEG	Female	0	1	0	1	0	0	DSL	0
1	5575-GNVDE	Male	0	0	0	34	1	0	DSL	1
2	3668-QPYBK	Male	0	0	0	2	1	0	DSL	1
3	7795-CFOCW	Male	0	0	0	45	0	0	DSL	1
4	9237-HQITU	Female	0	0	0	2	1	0	Fiber optic	0

5 rows × 23 columns

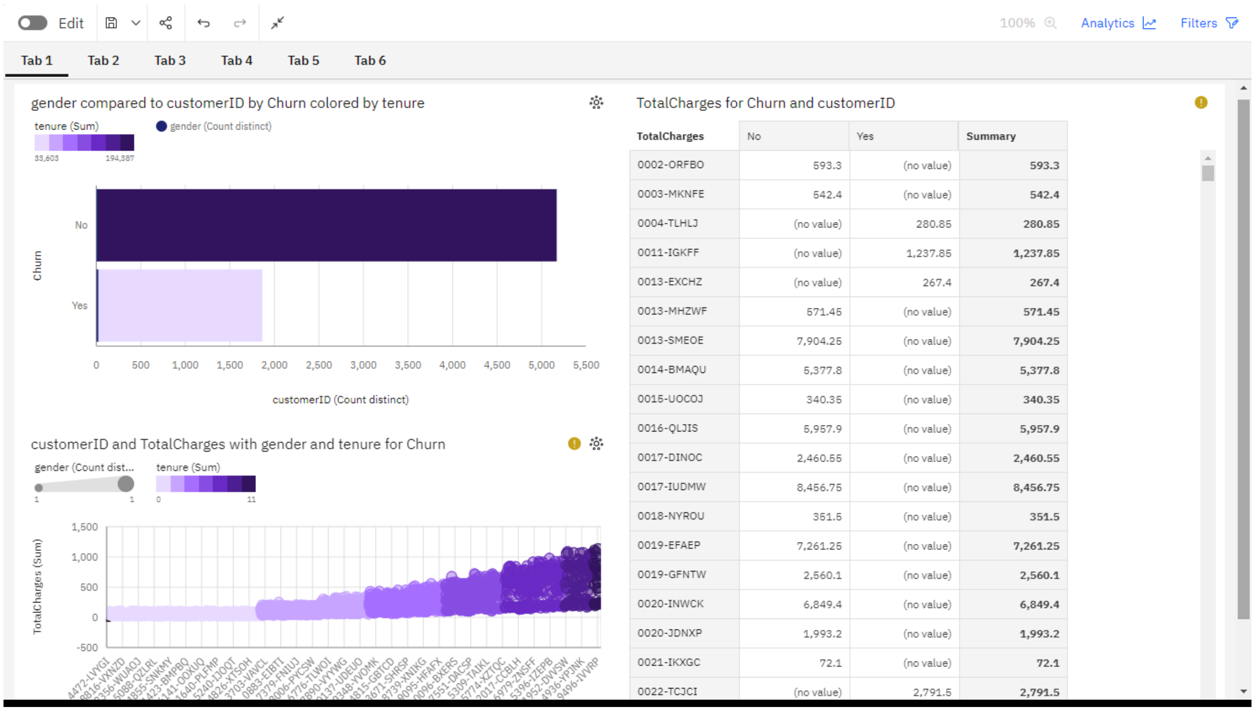
DATA VISUALIZATION USING IBM COGNOS:

DATA CONNECTION:1



DASHBOARD CREATION:2

Sample dashboard creation,



PROCEDURE:

Create a New Dashboard:

- In IBM Cognos, select "Authoring" or "Dashboard" from the menu.

Add Widgets:

- Start adding widgets to your dashboard. Widgets are the components that display data and visualizations.
- Common widget types include charts, tables, and text elements.
- Drag and drop widgets onto the dashboard canvas.

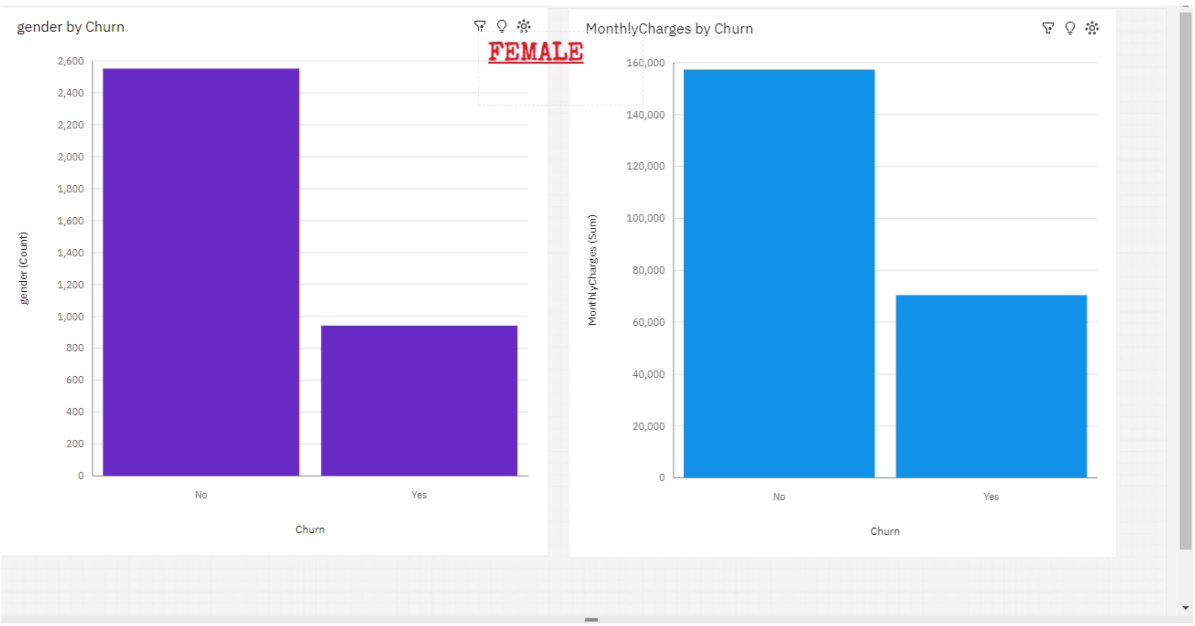
Connect Data:

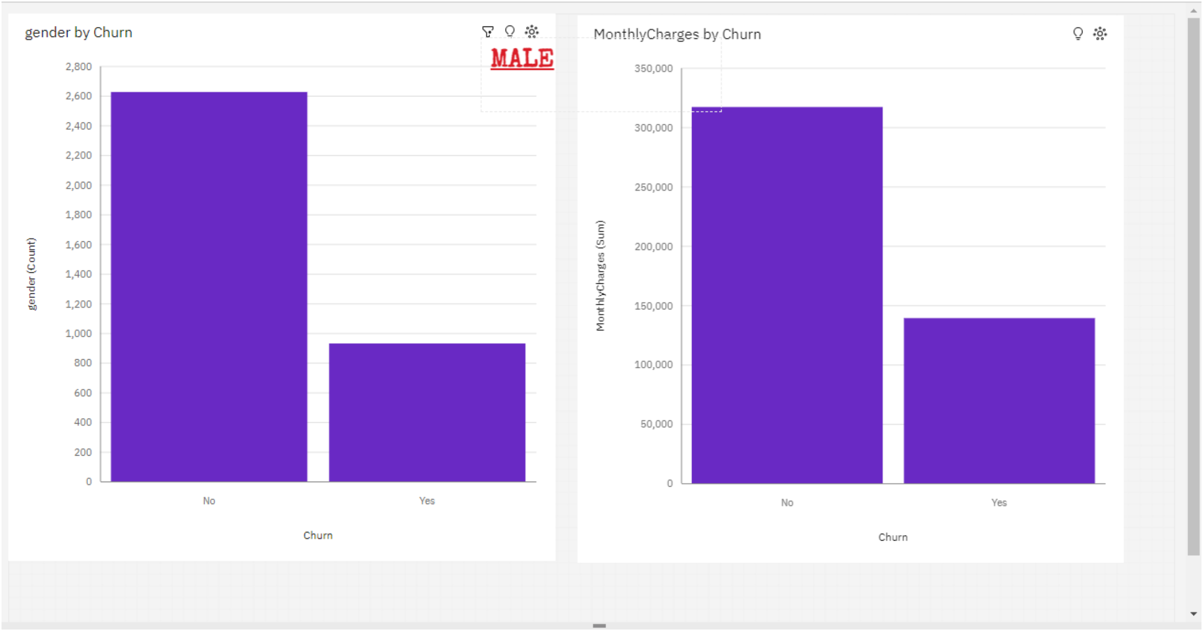
- Configure data sources for your widgets.
- This involves selecting the data you want to use for each widget.
- Define data connections to your dataset.

KEY VISUALIZATION:

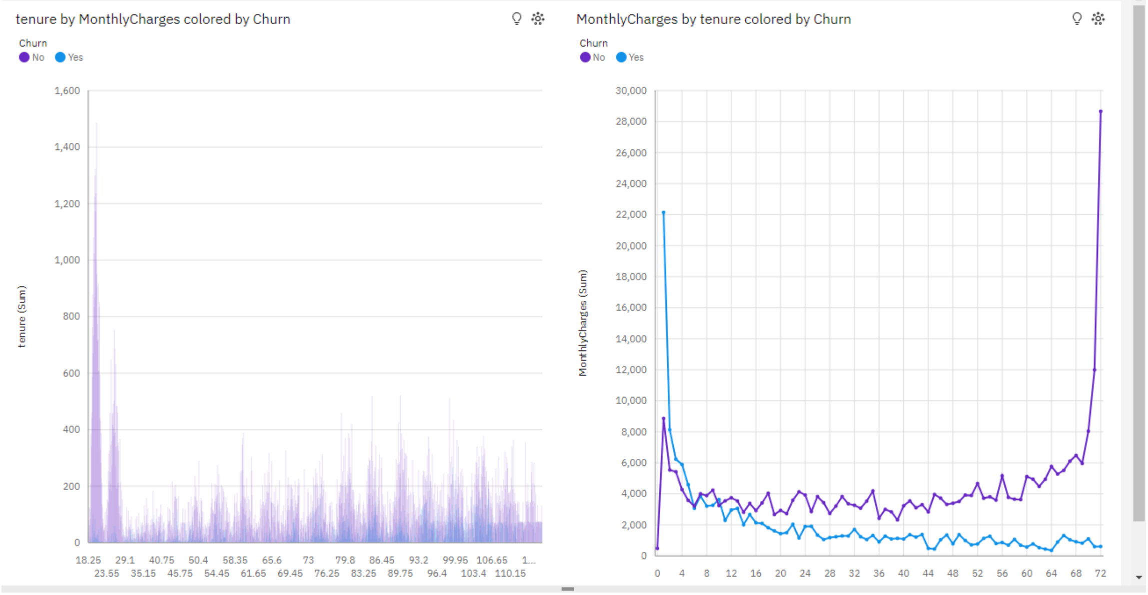
Customer demographics :

VISUALIZATION:





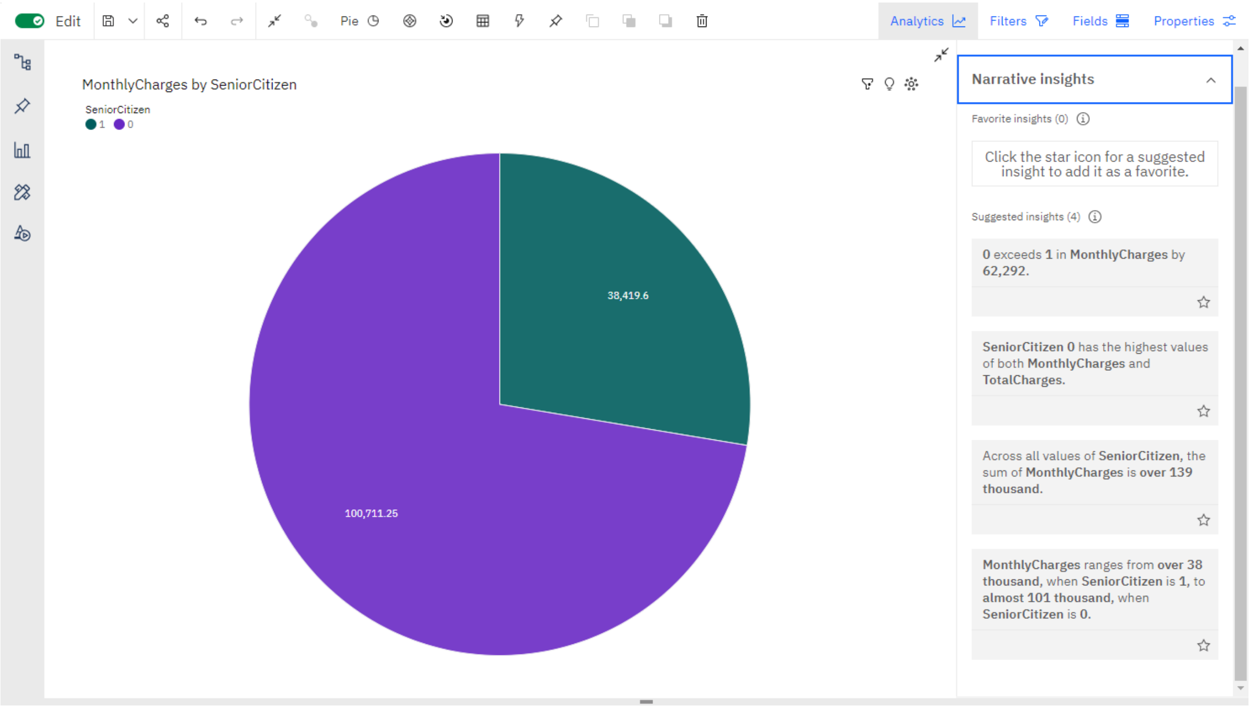
VISUALIZATION :



ANALYTICS INSIGHTS:

- 0 exceeds 1 in MonthlyCharges by 62,292.
- SeniorCitizen 0 has the highest values of both MonthlyCharges and TotalCharges.
- Across all values of SeniorCitizen, the sum of MonthlyCharges is over 139 thousand.
- MonthlyCharges ranges from over 38 thousand, when SeniorCitizen is 1, to almost 101 thousand, when SeniorCitizen is 0

VISUALIZATION:



ANALYTICS:

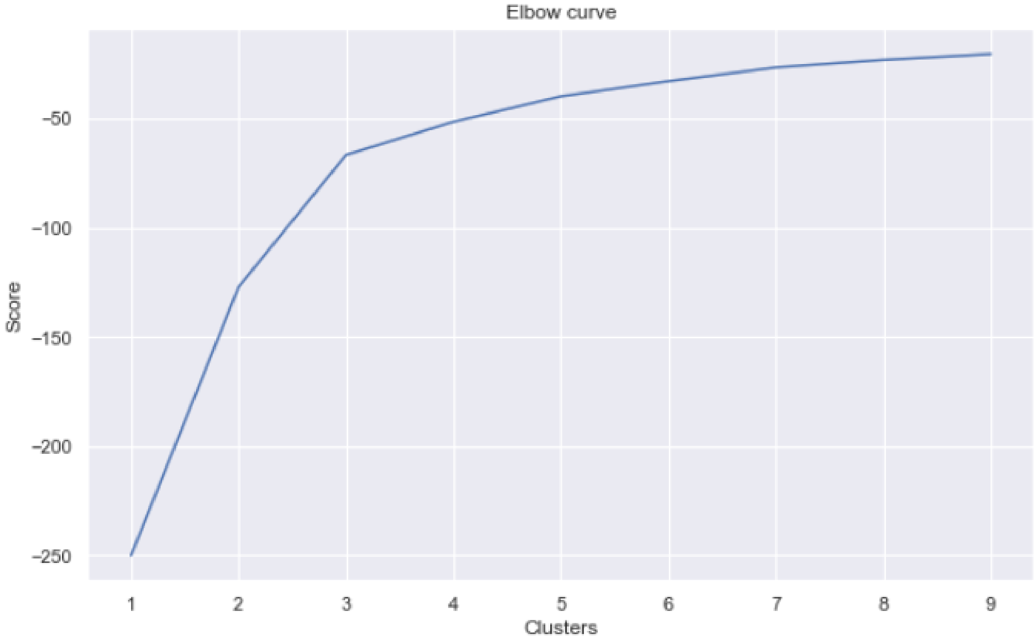
- ❖ Contract Month-to-month has the highest InternetService due to Churn No.
- ❖ Contract Month-to-month has the highest InternetService due to MultipleLines No.
- ❖ Churn No has the highest values of both InternetService and TotalCharges.
- ❖ The overall number of results for InternetService is over seven thousand.
- ❖ Month-to-month is the most frequently occurring category of Contract with a count of 3875 items with InternetService values (55 % of the total).
- ❖ MultipleLines No has the highest InternetService at 9, out of which Contract Month-to-month contributed the most at 3.

PREDICTIVE MODEL:

K means clustering is used as prediction model in unsupervised learning algorithm.

```
from sklearn.cluster import KMeans

df_kmeans_data = df_cal[df_cal.Churn==1][['Tenure_norm','MonthlyCharges_norm']]

k = range(1,10)

k means = [KMeans(n_clusters=i) for i in k]

score = [k means[i].fit(df_kmeans_data).score(df_kmeans_data)  for i in range(len(k
means))]

plt.figure(figsize=(10,6))

plt.plot(k,score)

plt.xlabel("Clusters")

plt.ylabel("Score")

plt.title("Elbow curve")
```

```
Text(0.5,1,'Elbow curve')
```

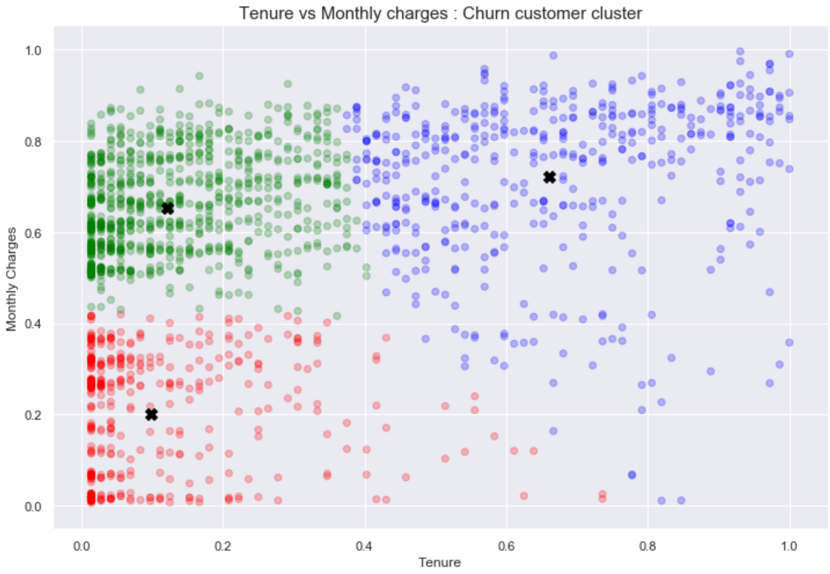


PROGRAM:

```
df_cal['Cluster'] = -1 # by default set Cluster to -1

K means = KMeans(n_clusters=3 # No of cluster in data , random_state = 2 # Selecting same
training data )
```

```
kmeans.fit(df_cal[df_cal.Churn==1][['Tenure_norm','MonthlyCharges_norm']])

kmean_colors = ['green' if c == 0 else 'blue' if c == 1 else 'red' for c in kmeans.labels_]

df_cal.loc[(df_cal.Churn==1),'Cluster'] =
kmeans.fit_predict(df_cal[df_cal.Churn==1][['Tenure_norm','MonthlyCharges_norm']])

fig = plt.figure(figsize=(12,8))

plt.scatter(x='Tenure_norm'

          , y='MonthlyCharges_norm'

          , data=df_cal[df_cal.Churn==1]

          , color=kmean_colors # color of data points

          , alpha=0.25 # transparancy of data points

         )

plt.xlabel("Tenure")

plt.ylabel("Monthly Charges")

plt.scatter(x=kmeans.cluster_centers_[:,0]

          , y=kmeans.cluster_centers_[:,1]

          , color='black'

          , marker='X' # Marker sign for data points

          , s=100 # marker size

         )

plt.title("Tenure vs Monthly charges : Churn customer cluster",fontsize=15)

plt.show()
```

OUTPUT:



REASON FOR CUSTOMER TO CHURN:

There are three types of customer group , who are more likely to churn

1. Less tenure and high monthly charges
2. High tenure and High monthly charges
3. Less tenure and low monthly charges

Program :

```
df_cluster_gp = df_cal[df_cal['Churn']==1].groupby('Cluster')

print(df_cluster_gp['Tenure','MonthlyCharges'].mean())
```

OUTPUT:

```
             Tenure  MonthlyCharges
Cluster
0          8.775553       83.821654
1         47.549569       90.731250
2          7.043860       38.343860
```

CONCLUSION:

By analyzing the given dataset on telecom industries and have used the prediction model of k-means clustering .Thus we have predicted the cause for the customer to churn is because the increase in charges for every service in that company has provided .By rectifying this we solve problem on the customer churn.

THANKING YOU !!!