CUSTOMER CHURN PREDICTION

TEAM MEMBERS :	REG NO:
1. SENTHIL VEL S(LEADER)	731221104034
2. VIGNESH M	731221104039
3. GUNASEKARAN S	731221104014
4. GUNASEKARAN P	731221104013
5. KAMALESH K	731221104017

Overview

Here in this part of the project we have collected the customer data according to the dataset provided . After that we have preprocessed those data by clearing all unwanted data and making it easier to analyze for the next step of the project.

Goals

1. Dataset has been loaded and perprocessed for the next step.
2. We have used some different visualization to the dataset for predicting customer churn.

LOADING DATASET

- You load data by reading from or writing to a file. You can read and write to files using Python's built-in open() method.
- The . writer() and . DictReader() methods from Python's CSV library make it even easier to work with CSV files in your Python code.

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.

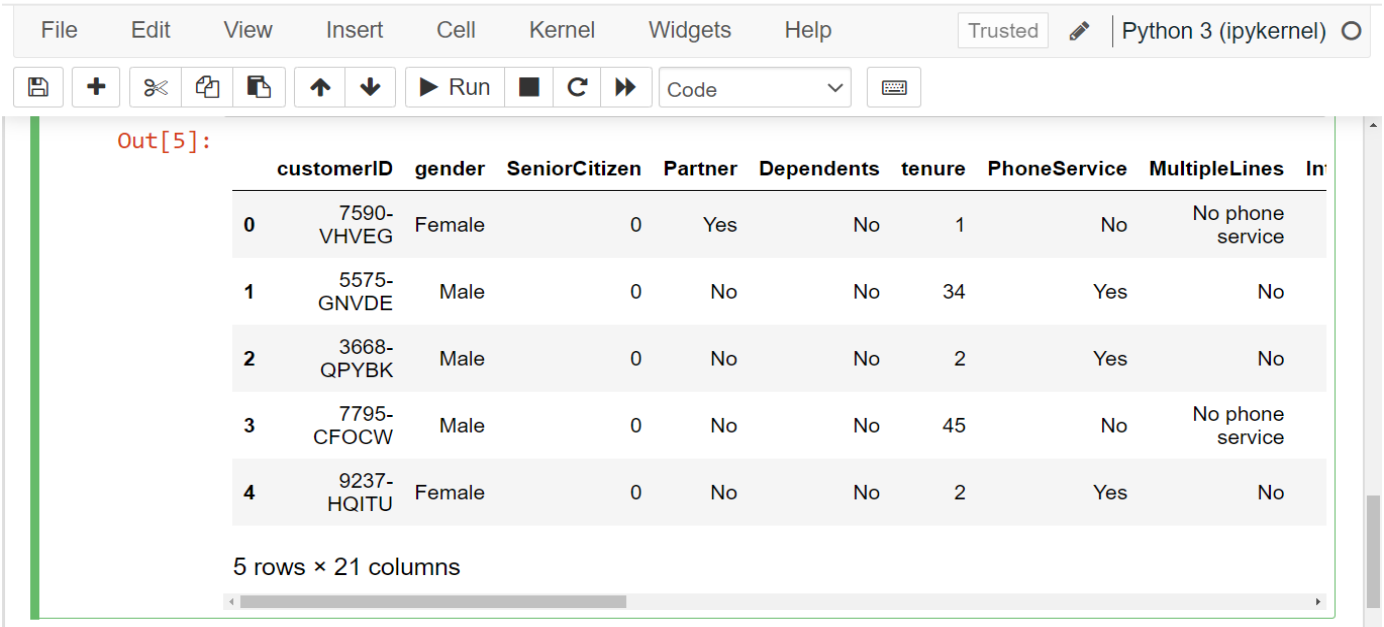
Import Necessary Libraries:

```
import pandas as pd
import numpy as np
```

Load Data:

Load your data into a Pandas DataFrame.

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



	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

Handling Missing Data:

Deal with missing values in your dataset.

Remove rows with missing values:

df.dropna(inplace=True)

Fill missing values with a specific value (e.g., mean or median):

df['column_name'].fillna(df['column_name'].mean(), inplace=True)

Handling Duplicates:

Remove duplicate rows from the dataset.

df.drop_duplicates(inplace=True)

Data Type Conversion:

Ensure that data types are correct for each column.

- Change data types:

```
df['column_name'] = df['column_name'].astype('new_data_type')
```

- Outliers Detection and Handling:

Identify and handle outliers.

Visualize and detect outliers:

```
import seaborn as sns
sns.boxplot(x=df['column_name'])
```

- Handle outliers (e.g., by removing or transforming them):

```
q1 = df['column_name'].quantile(0.25)
q3 = df['column_name'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
df = df[(df['column_name'] >= lower_bound) & (df['column_name'] <=
upper_bound)]
```

Feature Engineering:

Create new features or transform existing ones if needed.

- Feature scaling:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
df['scaled_column'] = scaler.fit_transform(df[['column_to_scale']])
```

- One-hot encoding for categorical variables:

```
df = pd.get_dummies(df, columns=['categorical_column'])
```

Normalization:

Normalize numerical data if necessary.

- Min-max normalization:

  from sklearn.preprocessing import MinMaxScaler

  minmax_scaler = MinMaxScaler()

  df['normalized_column']=minmax_scaler.fit_transform(df[['column_to_normal']])

Save the Cleaned Data:

Save the cleaned dataset to a new file.

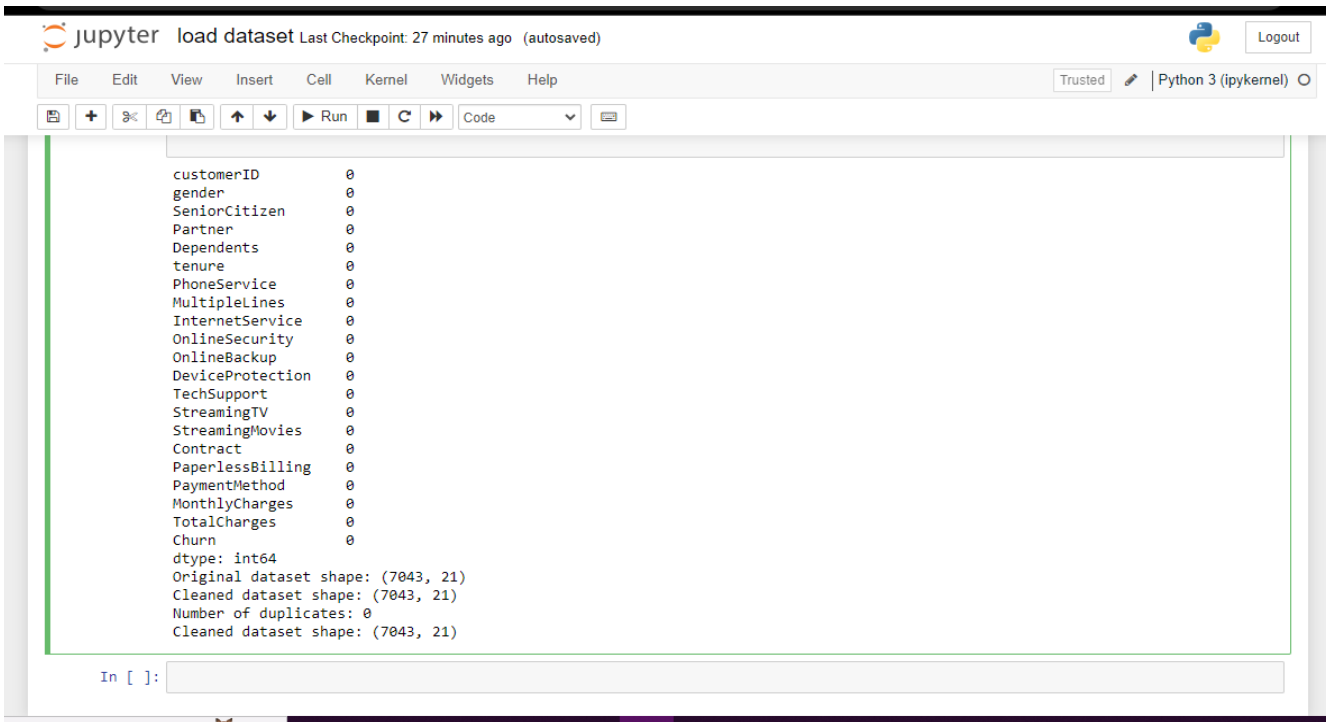
df.to_csv('telco.csv', index=False)

PROGRAM FOR PREPROCESSING THE DATASET:

```
import pandas as pd

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

df.head()

missing_values = df.isnull().sum()

print(missing_values)

df_cleaned = df.dropna()

print(f"Original dataset shape: {df.shape}")

print(f"Cleaned dataset shape: {df_cleaned.shape}")

duplicates = df.duplicated().sum()

df_cleaned = df_cleaned.drop_duplicates()

print(f"Number of duplicates: {duplicates}")
```

```
print(f"Cleaned dataset shape: {df_cleaned.shape}")

df_cleaned.to_csv('telco_customer_churn_cleaned.csv', index=False)
```

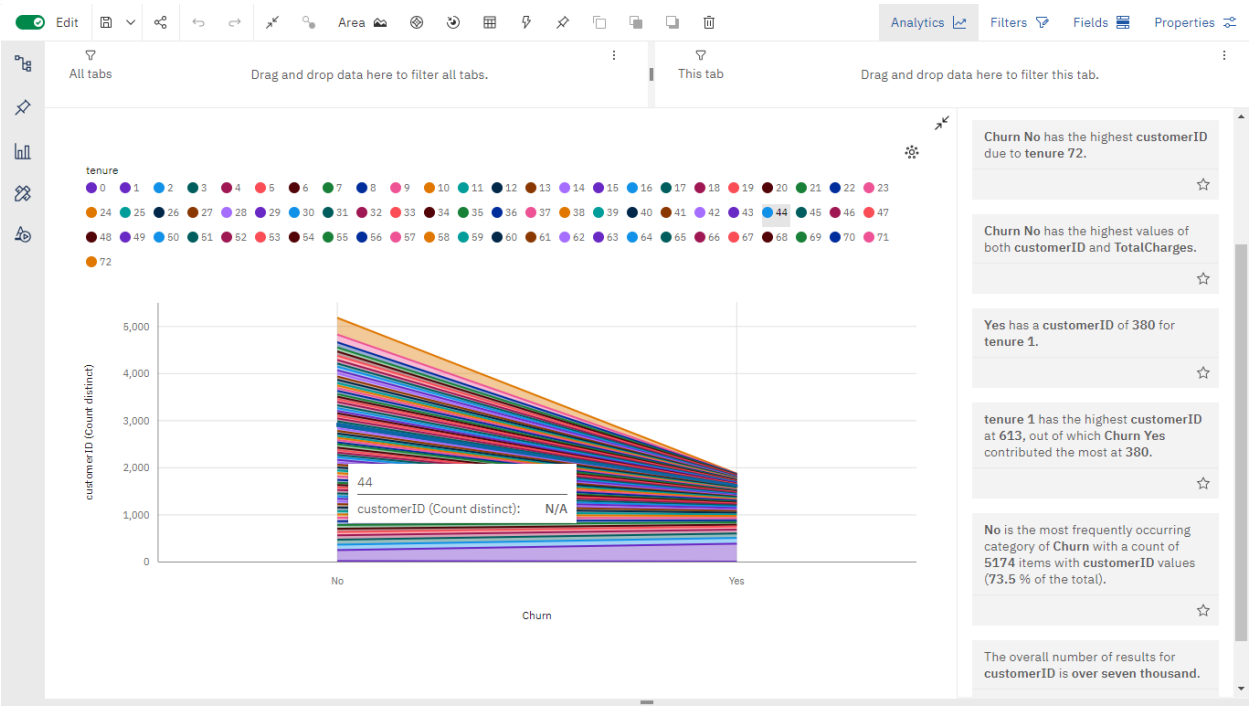
OUTPUT



VISUALIZATION:

- Data visualization is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations.
- These visual displays of information communicate complex data relationships and data-driven insights in a way that is easy to understand.

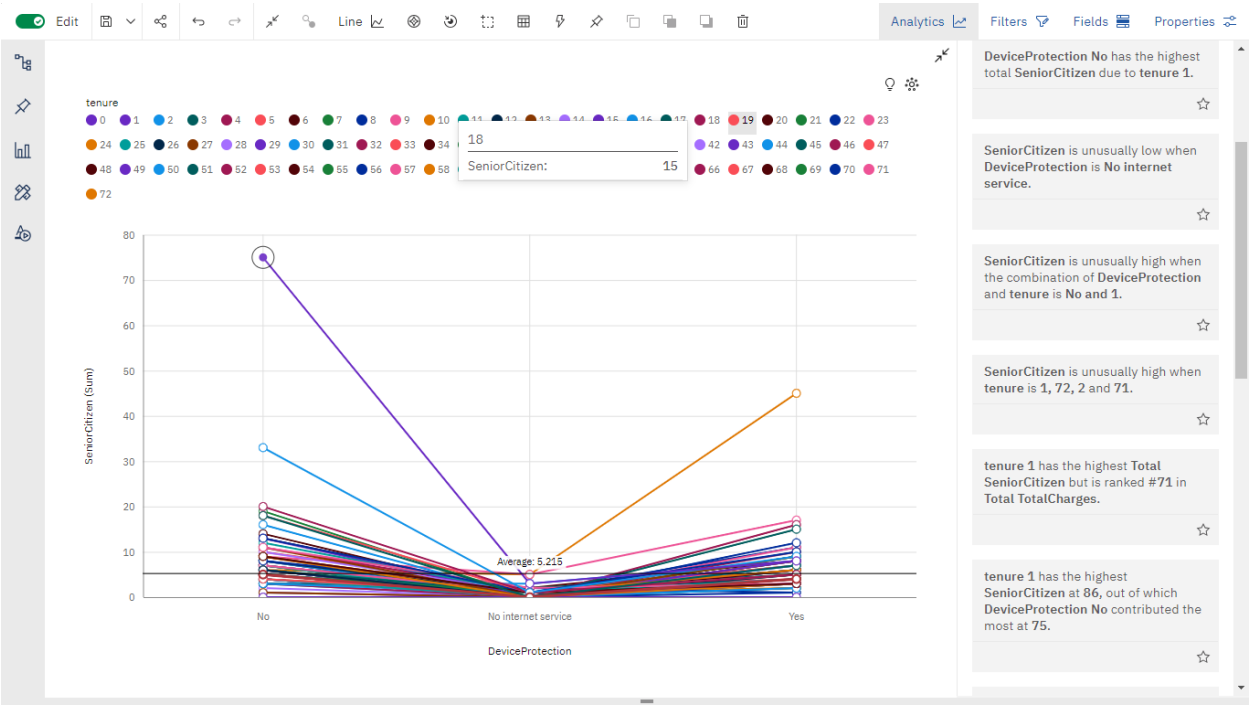
VISUALIZATION:



ANALYTICS INSIGHTS:-

- Churn No has the highest customerID due to tenure 72.
- Churn No has the highest values of both customerID and TotalCharges.
- No is the most frequently occurring category of Churn with a count of 5174 items with customerID values (73.5 % of the total)
- The overall number of results for customerID is over seven thousand.

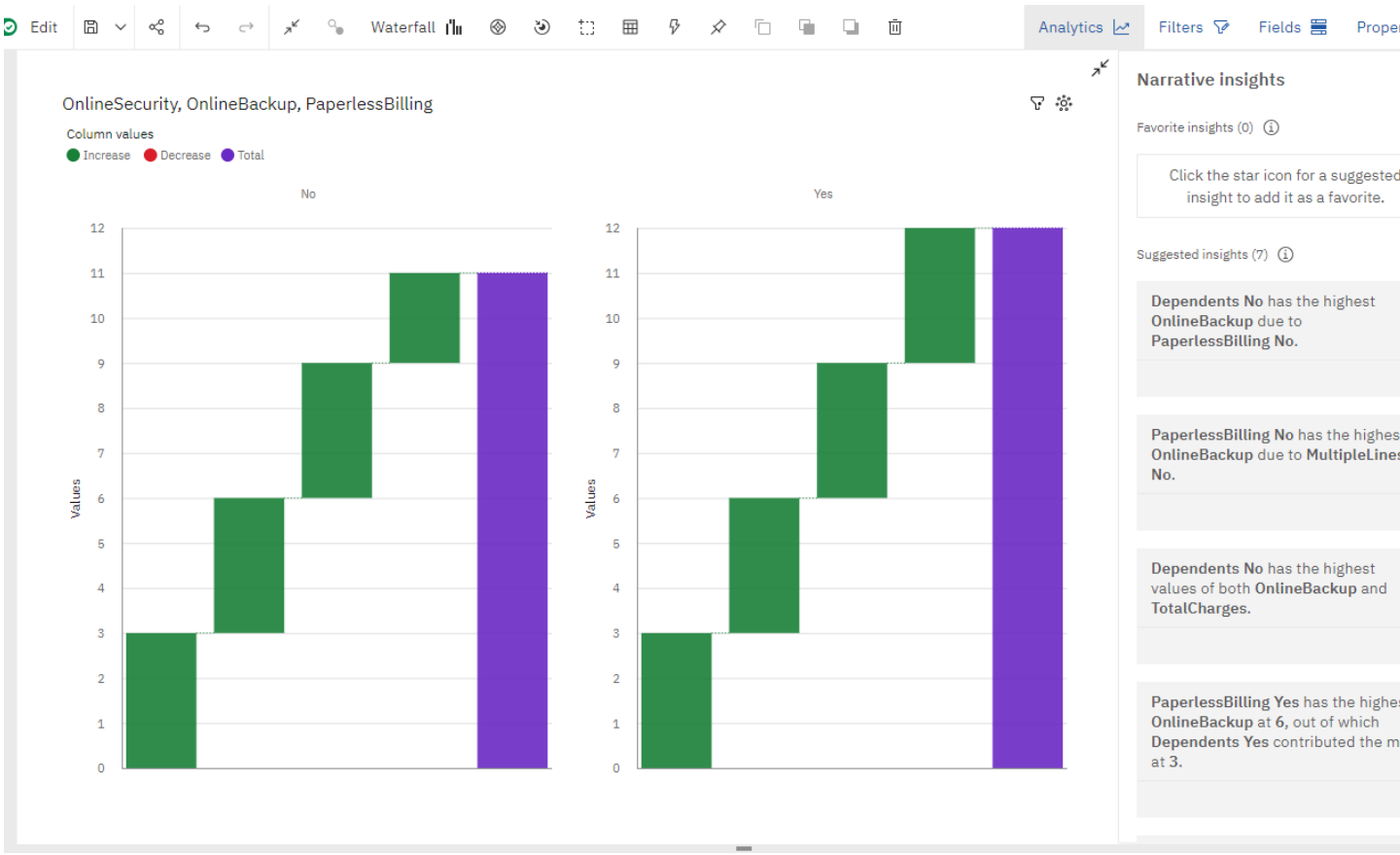
VISUALIZATION ON DEVICE PROTECTION :



ANALYTICS INSIGHTS:-

- Churn No has the highest total TotalCharges at over thirteen million.
- Churn Yes has the lowest total TotalCharges at almost 2.9 million.
- PaymentMethod Electronic check has the highest Contract due toMultipleLines No.
- The total number of results for Contract, across all PaymentMethod, is over seven thousand.

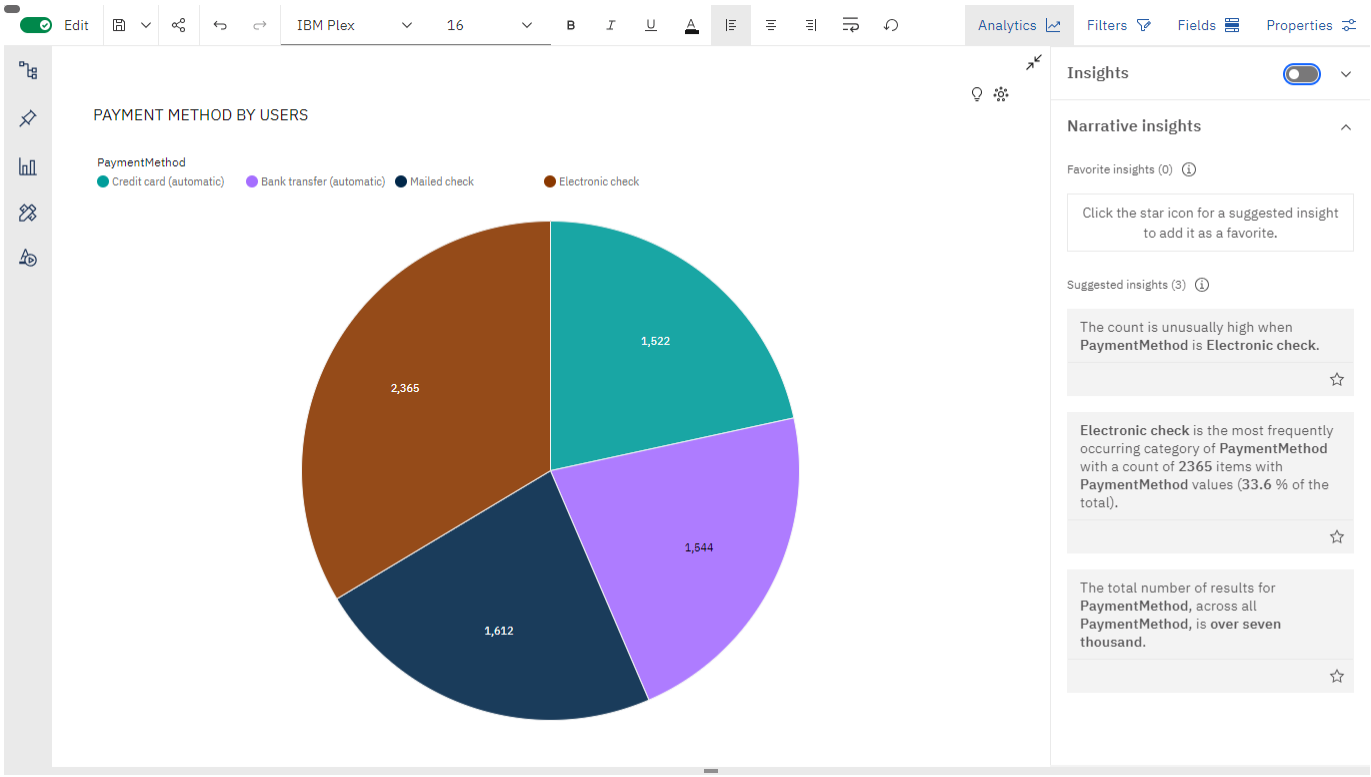
VISUALIZATION ON ONLINE SERVICES:



ANALYTICS INSIGHTS:-

- PaperlessBilling No has the highest OnlineBackup due to MultipleLines No.
- Dependents No has the highest values of both OnlineBackup and TotalCharges.
- The total number of results for OnlineBackup, across all PaperlessBilling, is over seven thousand.
- The total number of results for OnlineSecurity, across all PaperlessBilling, is over seven thousand.

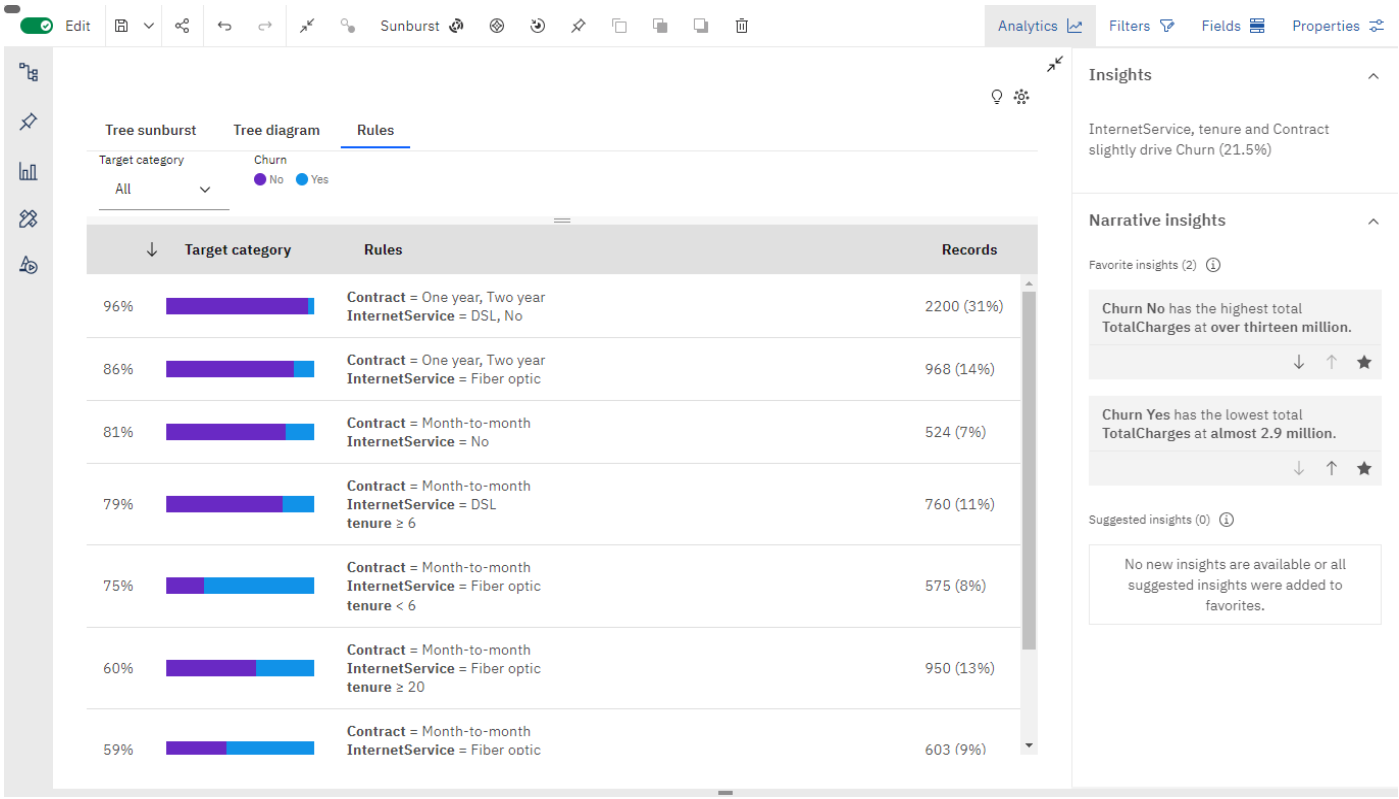
VISUALIZATION ON PAYMENT METHOD:



ANALYTICS INSIGHTS :

- The count is unusually high when PaymentMethod is Electronic check.
- Electronic check is the most frequently occurring category of PaymentMethod with a count of 2365 items with PaymentMethod values (33.6 % of the total).
- The total number of results for PaymentMethod, across all PaymentMethod, is over seven thousand.

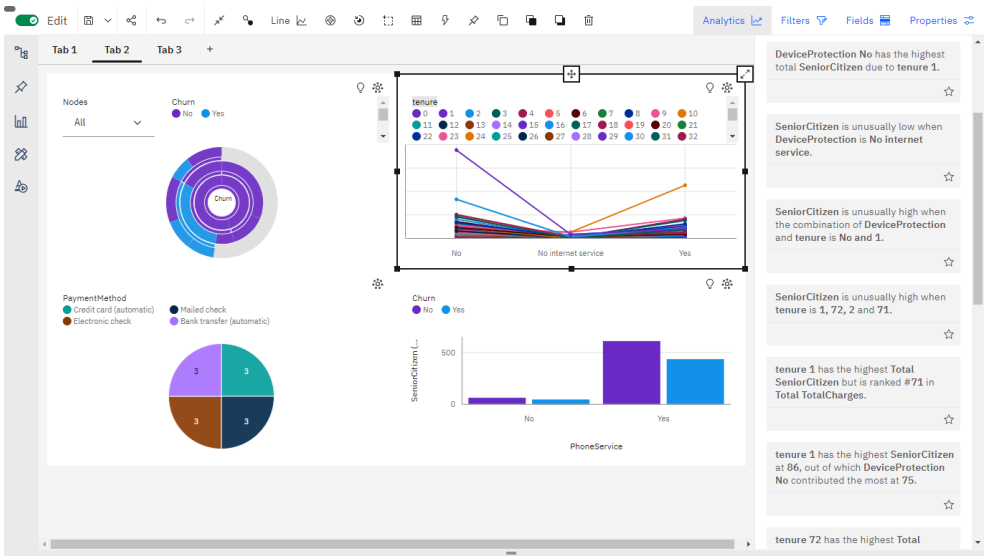
VISUALIZATION ON CHURN RULES:

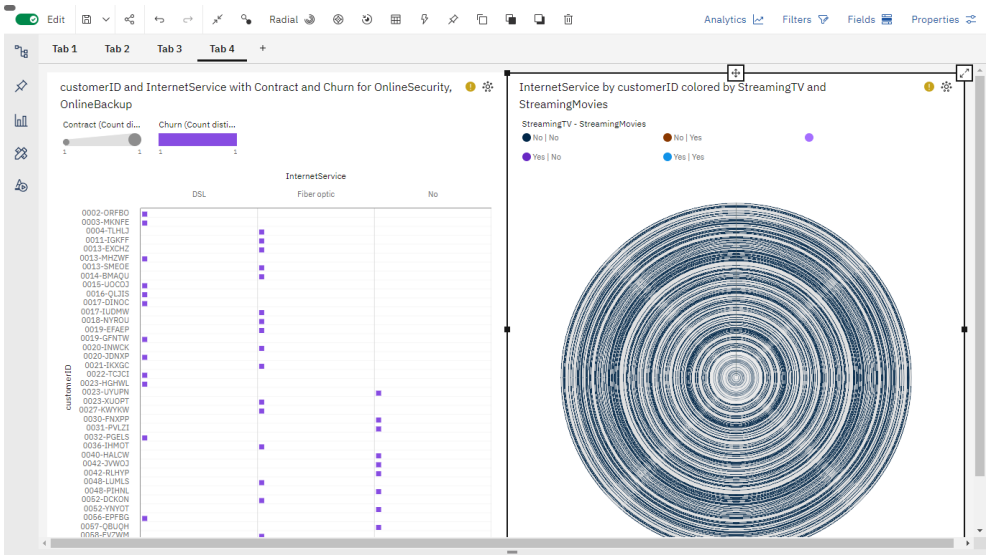


ANALYTICS INSIGHTS :

- Churn No has the highest total TotalCharges at over thirteen million.
- Churn Yes has the lowest total TotalCharges at almost 2.9 million.
- PhoneService Yes has the highest total SeniorCitizen due to Churn No.
- The summed values of SeniorCitizen range from 44 to 606.

SAMPLE VISUALIZATION :





THANKING YOU !