**PHASE 3 PROJECT SUBMISSION**

**PROJECT 6 - CUSTOMER CHURN PREDICTION**

**TEAM MEMBERS:**

1. Rithees S M - 2021504032
2. Akshay G S – 2021504502
3. Dhanesh C N E – 2021504510
4. Divyavarshini K – 2021504513

**Problem Definition:**

The project involves using IBM Cognos to predict customer churn and identify factors influencing customer retention. The goal is to help businesses reduce customer attrition by understanding the patterns and reasons behind customers leaving. This project includes defining analysis objectives, collecting customer data, designing relevant visualizations in IBM Cognos, and building a predictive model.

**Phase Objective:**

In this phase 3, it is asked to clean the given dataset and make it more meaningful, preprocess the dataset and perform various analysis and visualization using IBM Cognos.

**Dataset Link:**

[**https://www.kaggle.com/datasets/blastchar/telco-customer-churn**](https://www.kaggle.com/datasets/blastchar/telco-customer-churn)

**Source Code:**

*# Import necessary modules*

import numpy as np

import pandas as pd

from sklearn.preprocessing import OneHotEncoder

from sklearn.preprocessing import MinMaxScaler

*# Load your dataset (please replace the path with your own)*

df = pd.read\_csv("F:/MIT/NM assn/Data Analytics Using Cognos/Dataset/archive/WA\_Fn-UseC\_-Telco-Customer-Churn.csv")

*# Initial data exploration*

print("Initial dataset shape:", df.shape)

df.info()

*# Data cleaning: Remove duplicates and null values*

df = df.drop\_duplicates()

df = df.dropna()

*# Verify removal of duplicates and null values*

print("Count of missing values after cleaning:")

print(df.isnull().sum())

print("Count of duplicate values after cleaning:", df.duplicated().sum())

*# Remove unnecessary columns (e.g., 'customerID' not related to churn)*

df = df.drop('customerID', *axis*=1)

print("Dataset shape after dropping 'customerID':", df.shape)

*# Replacing certain categorical values with binary values*

*# InternetService: DSL = 1, FiberOptic = 1, No = 0*

df['FiberOptics'] = df.loc[:, 'InternetService']

internet\_service\_data = {'DSL': 1, 'Fiber optic': 1, 'No': 0}

df.replace({'InternetService': internet\_service\_data}, *inplace*=True)

*# Further categorizing InternetService: DSL = 0, Fiber Optic = 1, No = 0*

fiber\_optic\_data = {'DSL': 0, 'Fiber optic': 1, 'No': 0}

df.replace({'FiberOptics': fiber\_optic\_data}, *inplace*=True)

*# Handling Internet-related data columns*

internet\_related\_data = {'No': 0, 'Yes': 1, 'No internet service': 0}

internet\_related\_cols = ['OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies']

df[internet\_related\_cols] = df[internet\_related\_cols].replace({k: v for k, v in internet\_related\_data.items()})

*# Convert other columns to binary values*

binary\_columns = ['Partner', 'Dependents', 'PhoneService', 'PaperlessBilling', 'Churn']

df[binary\_columns] = df[binary\_columns].replace({'Yes': 1, 'No': 0})

*# Convert 'gender' to binary values: Male = 1, Female = 0*

df['Gender\_BIN'] = df['gender'].apply(lambda *x*: 1 if *x* == 'Male' else 0)

*# Replacing binary values for 'MultipleLines'*

phone\_service = {'No phone service': 0, 'No': 0, 'Yes': 1}

df.replace({'MultipleLines': phone\_service}, *inplace*=True)

*# One-hot encoding for 'Contract' and 'PaymentMethod'*

ohe = OneHotEncoder()

encoded\_data = pd.DataFrame(ohe.fit\_transform(df[['Contract', 'PaymentMethod']]).toarray())

encoded\_data.columns = ohe.get\_feature\_names\_out()

df = df.join(encoded\_data)

df.drop(['Contract', 'PaymentMethod'], *axis*=1, *inplace*=True)

*# Data preprocessing: scaling the non-binary columns*

*# Remove rows with empty 'TotalCharges' values and convert to float*

empty = df[df['TotalCharges'] == " "].index

df.drop(empty, *inplace*=True)

df['TotalCharges'] = df['TotalCharges'].astype(float)

*# Scaling columns: 'tenure', 'MonthlyCharges', and 'TotalCharges'*

scalable\_columns = ['tenure', 'MonthlyCharges', 'TotalCharges']

mm\_scaler = MinMaxScaler()

df\_scaling = pd.DataFrame(mm\_scaler.fit\_transform(df[scalable\_columns]))

df\_scaling.columns = mm\_scaler.get\_feature\_names\_out()

*# Replace original columns with scaled versions*

df.drop(scalable\_columns, *axis*=1, *inplace*=True)

df = df.join(df\_scaling)

*# Ensure 'Churn' is the last column*

df['Churn'] = df.pop('Churn')

*# Final data exploration*

print("Final dataset info:")

print(df.info())

*# Save the cleaned and processed dataset to a new CSV file (please replace the path)*

df.to\_csv('F:/MIT/NM assn/Data Analytics Using Cognos/ output.csv', *index*=False)

**Github Link:**

[**https://github.com/dhanesh15/NM-Data-Analytics-Using-Cognos/tree/main/Phase3-Code-OP**](https://github.com/dhanesh15/NM-Data-Analytics-Using-Cognos/tree/main/Phase3-Code-OP)

(Contains the source code and output CSV file)

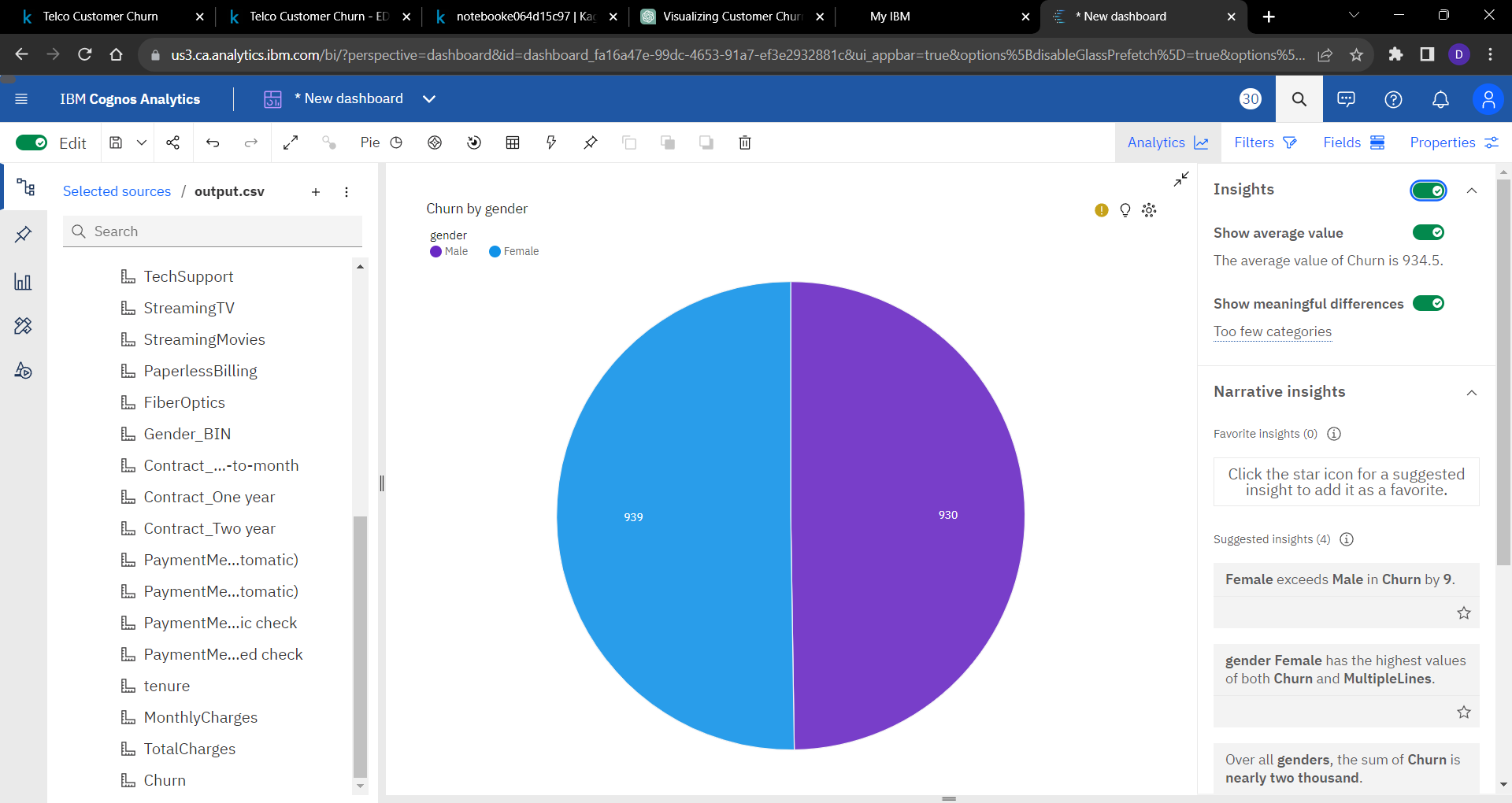
The above given code was successfully executed and resultant csv file was obtained.

This was then uploaded to IBM Cognos for Visualization and the following graphs were plotted for visualizing the output csv file.

The following are the data and their insights obtained from IBM Cognos:

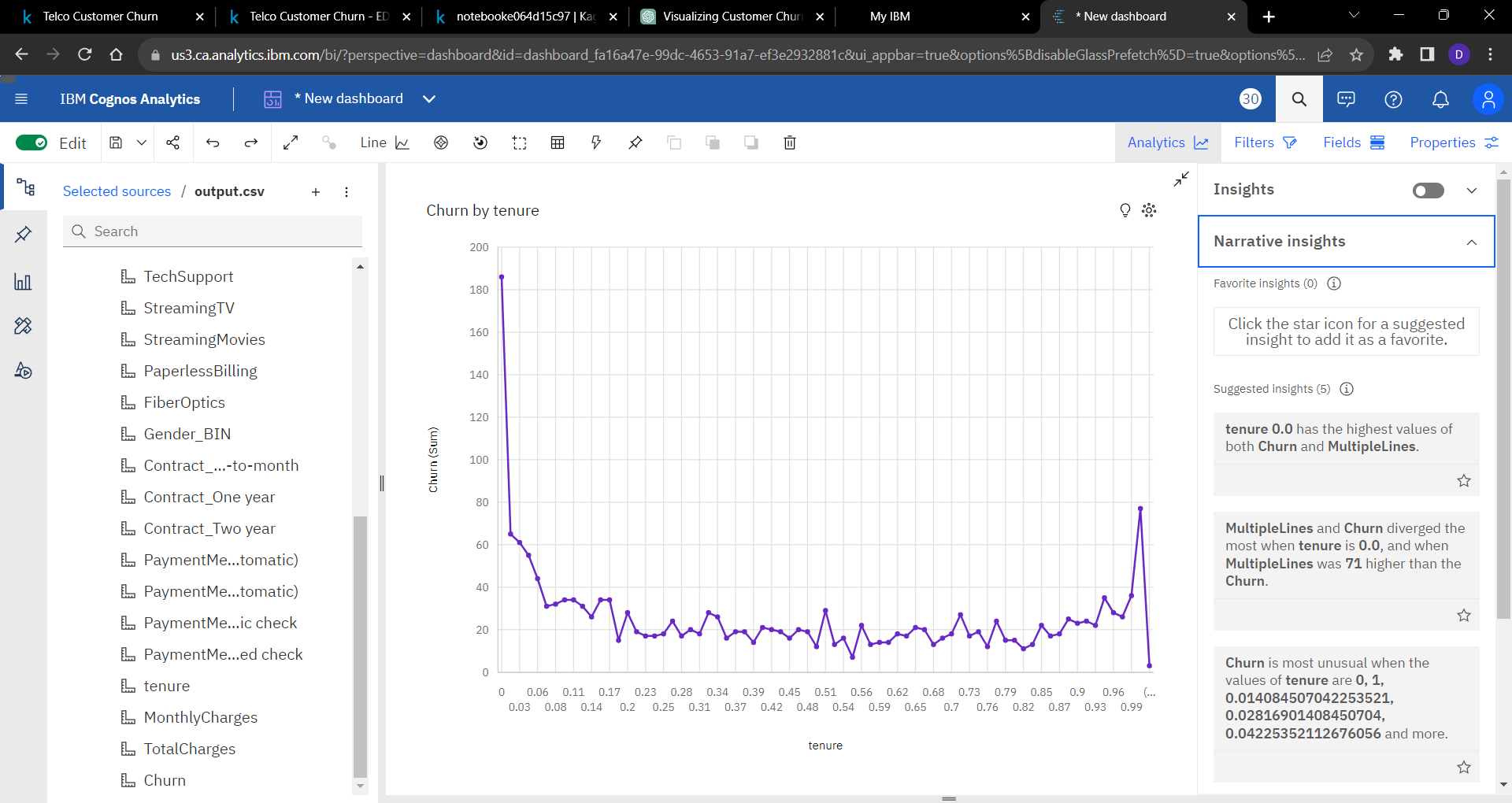
\*All values are normalized to 1

A pie chart was plotted for number of churners based on gender :



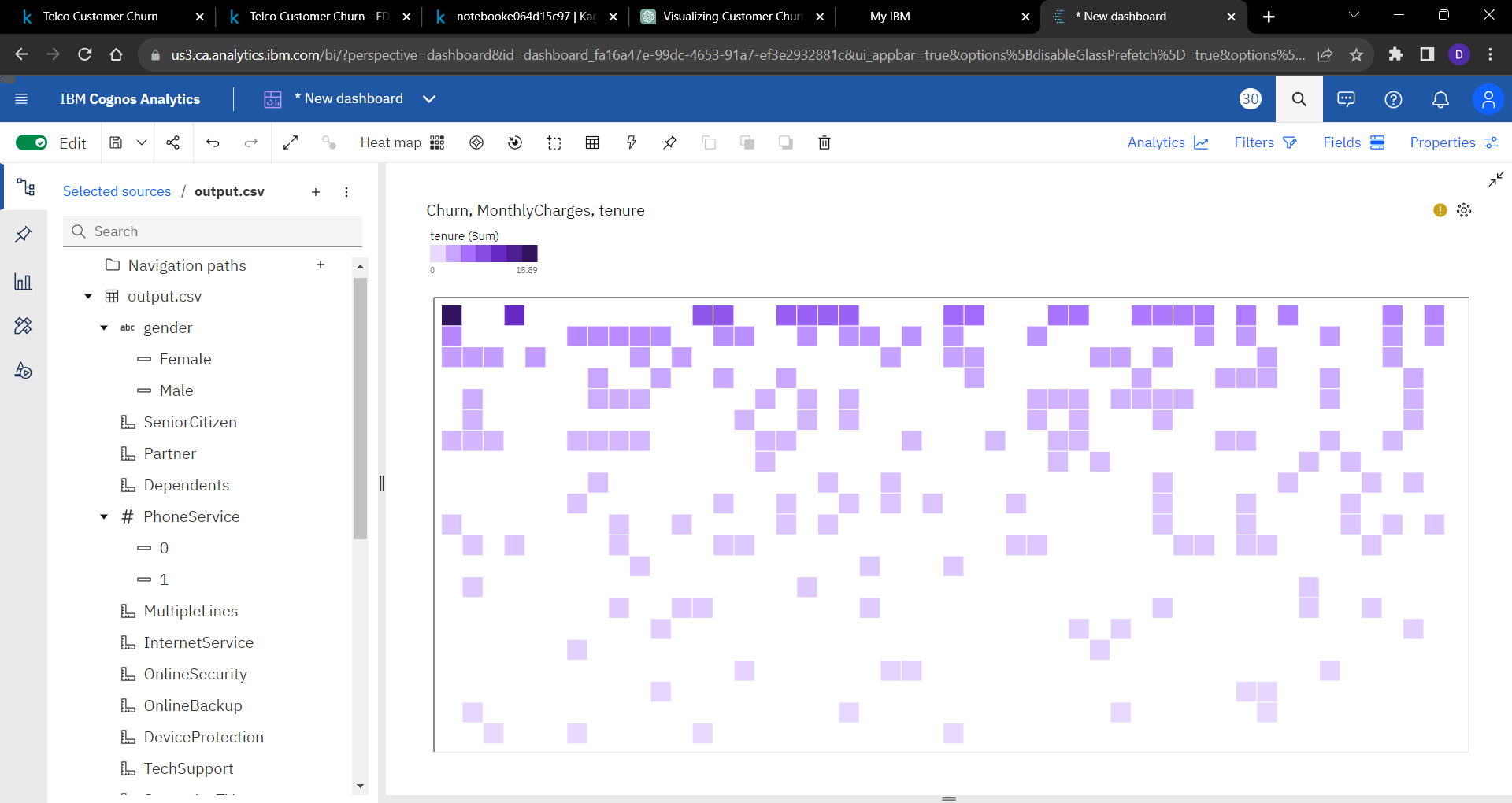
* It is observed that number of male churners are 930 while for women its 939.
* Over all genders, the sum of Churn is nearly two thousand.

A line graph was plotted between tenure and churn:



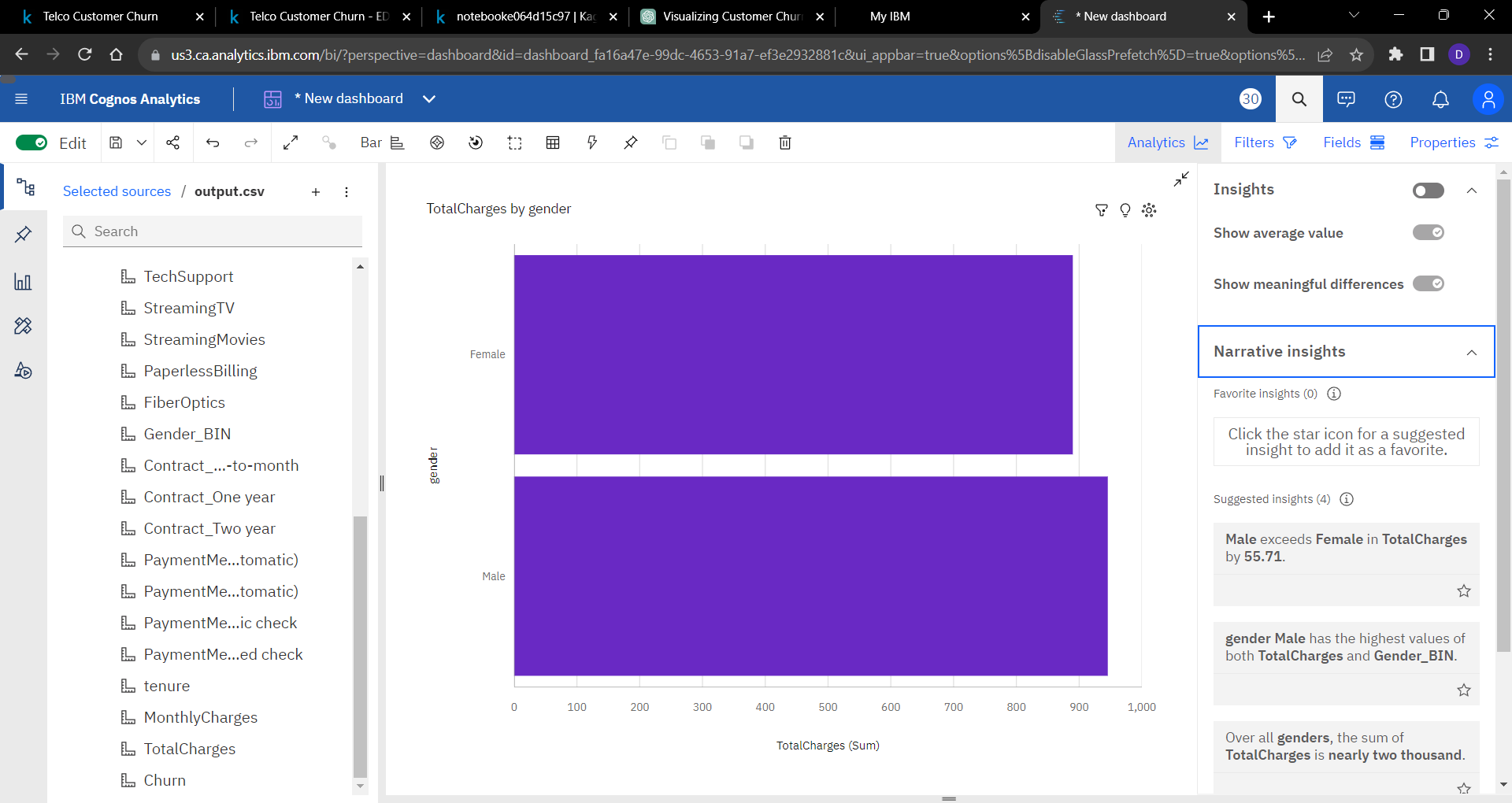
* Churn is most unusual when the values of tenure are

0, 1, 0.014084507042253521, 0.02816901408450704, 0.04225352112676056 and more.

An heatmap was plotted for Churn, Monthly Charges and Tenure to show how these parameters are related as follows

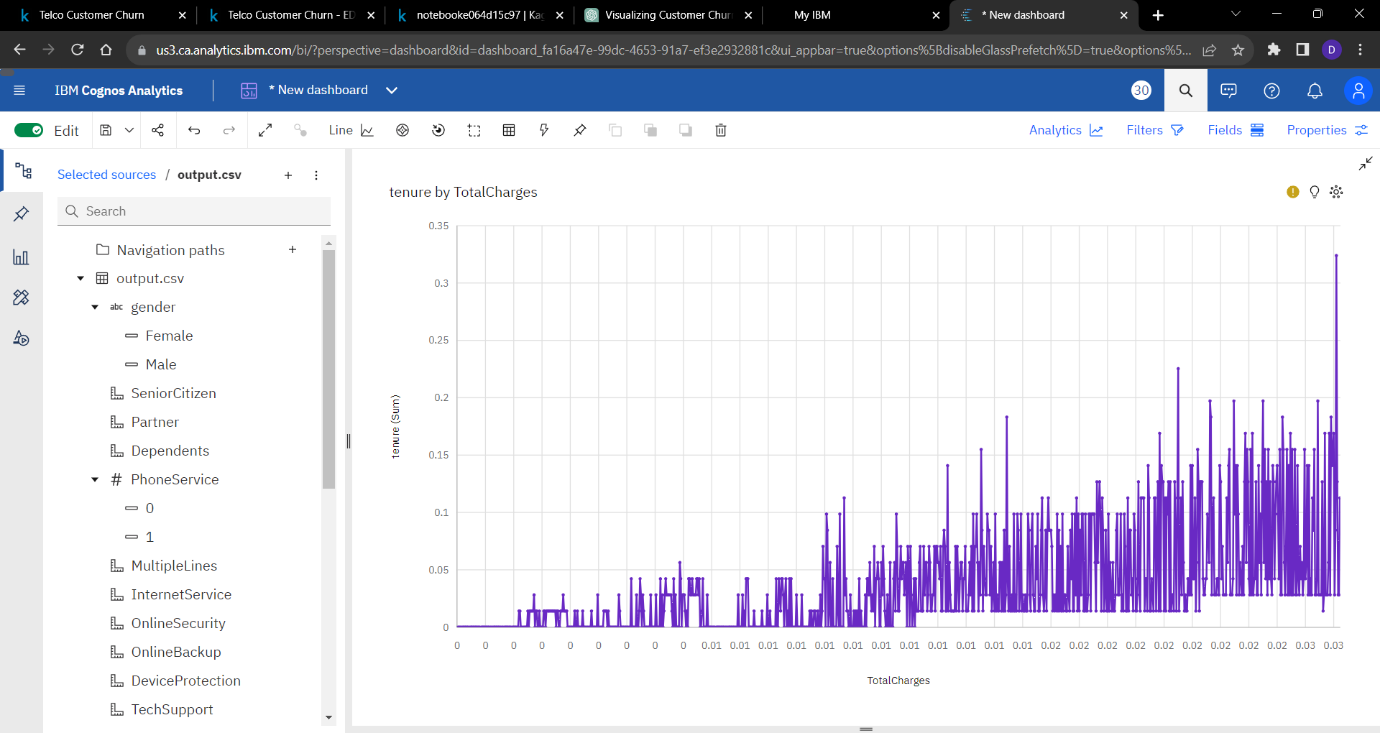
* Churn 0 has the highest total tenure due to MonthlyCharges 0.015920398009950265.
* MonthlyCharges 0.8034825870646766 has the highest Total TotalCharges but is ranked #34 in Total tenure.
* MonthlyCharges 0.01791044776119402 has the highest tenure at 20.18, out of which Churn 0 contributed the most at 14.46.

A bar graph is plotted based on gender ( Male / Female ) vs Total Charges inorder to visualized the variation of total charges paid by men and women for this application.



* TotalCharges ranges from 889.9, when gender is Female, to 945.6, when gender is Male.
* Male exceeds Female in TotalCharges by 55.71.

Another bar plot for Total Chargers vs Tenure was plotted is shown below:



**Conclusion:**

Thus, the given dataset was cleaned and were visualized using IBM Cognos successfully.