ANAND INSTITUTE OF HIGHER TECHNOLOGY OLD MAHABALIPURAM ROAD, KALASALINGAM NAGAR, KAZHIPATTUR – 603103

CUSTOMER CHURN PREDICTION WITH DATA ANALYTICS USING COGNOS

PHASE – 4

NAME : MOUNIKA V

REG No. : 310121104065

BRANCH : COMPUTER SCIENCE & ENGINEERING

YEAR/SEM : III / V

CUSTOMER CHURN PREDICTION

INTRODUCTION

- Churn prediction is predicting which customers are at high risk of leaving your company or canceling a subscription to a service, based on their behavior with your product.
- To predict churn effectively, you'll want to synthesize and utilize key indicators defined by your team to signal when a customer has a probability of churning so that your company can take action.
- At a high level, predicting customer churn requires a detailed grasp of your clientele. Both qualitative and quantitative customer data are usually needed to start building an effective churn prediction model. To ensure that predictions aren't being made by arbitrary human guesses, these models are often built by a data scientist using machine learning.
- The ability to predict that a customer is at high risk of churning while there is still time to do something about it represents a huge additional potential revenue source for every online business.

PHASE – 4 : DEVELOPMENT PART 2

- To continue building the analysis by creating visualizations using IBM Cognos and developing a predictive model.
- Create interactive dashboards and reports in IBM Cognos to visualize churn patterns, retention rates, and key factors influencing churn.
- Use machine learning algorithms to build a predictive model that identifies potential churners based on historical data and relevant features.

PROBLEM DEFINITION

Customer churn is when customers cease their relationship with a company or business, typically by discontinuing their use of its products or services. It is a significant concern for businesses across various industries, including telecommunications, software, e-commerce, and subscription-based services. Customer churn is a crucial metric impacting a company's profitability and growth. Losing existing customers leads to revenue decline and incurs costs associated with acquiring new customers to replace the lost ones. High customer churn can also damage a company's reputation and hinder its long-term sustainability.

Required steps:

Step1: univariate Analysis

Univariate analysis is the simplest form of analyzing data. *Uni* means "one", so the data has only one variable . Univariate data requires to analyze each variable separately. Data is gathered for the purpose of answering a question, or more specifically, a research question.

Step2: Bivariate Analysis

Bivariate analysis is stated to be an analysis of any concurrent relation between two variables or attributes. This study explores the relationship of two variables as well as the depth of this relationship to figure out if there are any discrepancies between two variables and any **causes of this** difference. Some of the examples are percentage table, scatter plot, etc.

Step3: Time series analysis

Time Series models are based on data where we observe predictors and churn simultaneously as they occur period to period. So, each customer would be observed for several periods. Each period potential predictors would be collected, as well as whether or not the customer churned.

Step4: Dashboard creation

A customer churn dashboard is a visual representation of key metrics and data related to customer churn within a business. Customer churn refers to the rate at which customers stop using a product or service and switch to a competitor or discontinue their usage altogether.

Step5: heatmap

A Churn Heatmap is a visual summary of every account journey of your churned accounts - from customer acquisition to customer exit - scoring the company's performance at each step with a color code.

Step6: Cohort analysis

The examination of different groups of users based on how they interact with your product. Cohort analysis is just one way to use behavioral data from your product or site, but it is often the most useful in reducing churn.

Step7: Geospatial

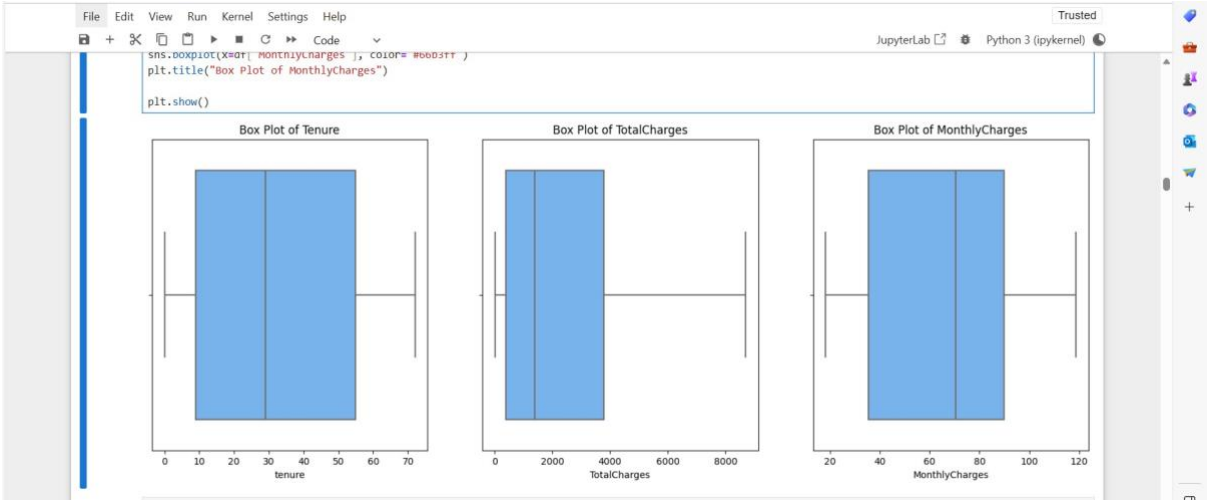
Geospatial data analysis involves collecting, combining, and visualizing various types of geospatial data.It is used to model and represent how people, objects, and phenomena interact within space, as well as to make predictions based on trends in the relationships between places.

Step8:

Churn analysis helps you proactively identify customers who are likely to churn. Creating alerts to notify you about real-time changes is a great way to stay on top of your churn metrics. You can create such custom alerts in Chargebee's RevenueStory.

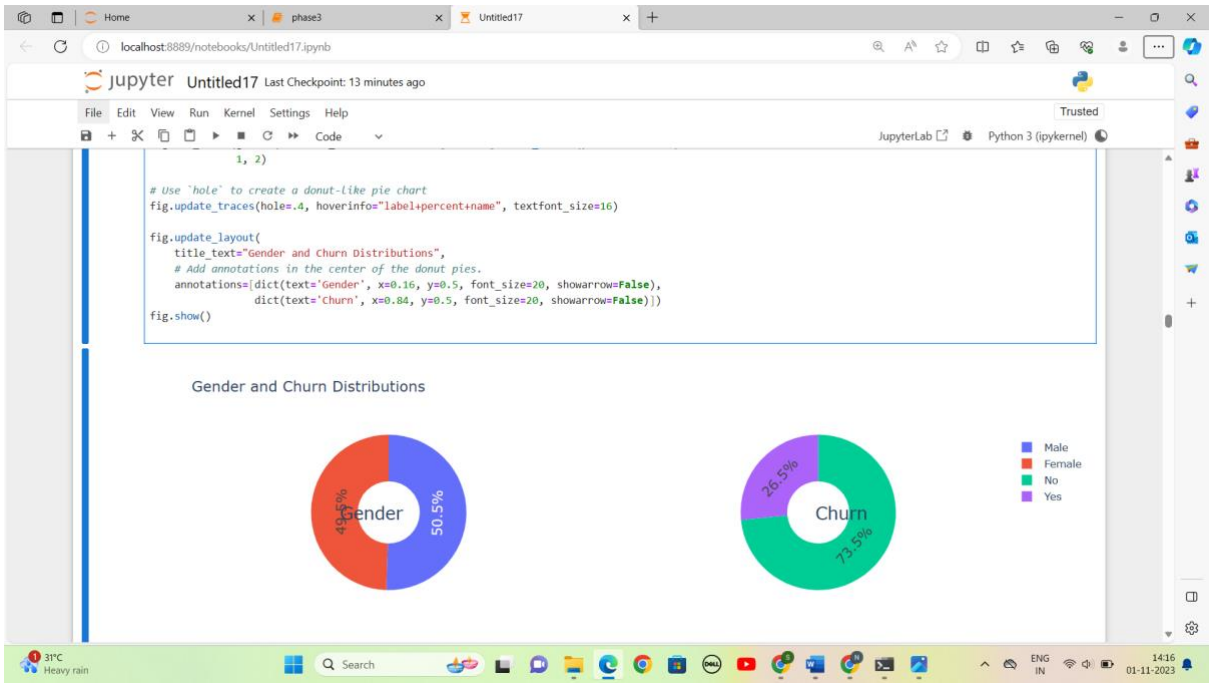
Use machine learning algorithms to build a predictive model that identifies potential churners based on historical data and relevant features.

```
# Create separate box plots for 'tenure', 'TotalCharges', and 'MonthlyCharges'
plt.figure(figsize=(18, 6))

# Box Plot of 'tenure'
plt.subplot(131)  # 1 row, 3 columns, plot 1
sns.boxplot(x=df['tenure'], color='#66b3ff')
plt.title("Box Plot of Tenure")

# Box Plot of 'TotalCharges'
plt.subplot(132)  # 1 row, 3 columns, plot 2
sns.boxplot(x=df['TotalCharges'], color='#66b3ff')
plt.title("Box Plot of TotalCharges")

# Box Plot of 'MonthlyCharges'
plt.subplot(133)  # 1 row, 3 columns, plot 3
sns.boxplot(x=df['MonthlyCharges'], color='#66b3ff')
plt.title("Box Plot of MonthlyCharges")

plt.show()
```




```
g_labels = ['Male', 'Female']

c_labels = ['No', 'Yes']


# Create subplots: use 'domain' type for Pie subplot

fig = make_subplots(rows=1, cols=2, specs=[[{'type':'domain'},
{'type':'domain'}]])

fig.add_trace(go.Pie(labels=g_labels,
values=df['gender'].value_counts(), name="Gender"),

              1, 1)

fig.add_trace(go.Pie(labels=c_labels,
values=df['Churn'].value_counts(), name="Churn"),

              1, 2)


# Use `hole` to create a donut-like pie chart

fig.update_traces(hole=.4, hoverinfo="label+percent+name",
textfont_size=16)
```

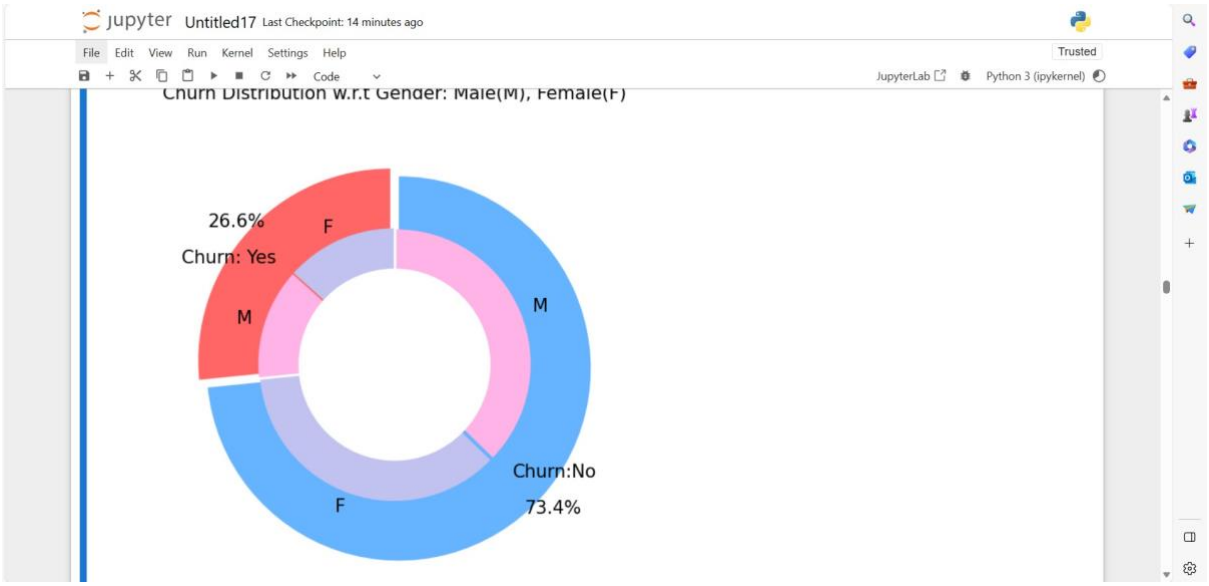
```
fig.update_layout(
    title_text="Gender and Churn Distributions",
    # Add annotations in the center of the donut pies.
    annotations=[dict(text='Gender', x=0.16, y=0.5, font_size=20, showarrow=False),
                 dict(text='Churn', x=0.84, y=0.5, font_size=20, showarrow=False)])
fig.show()
```



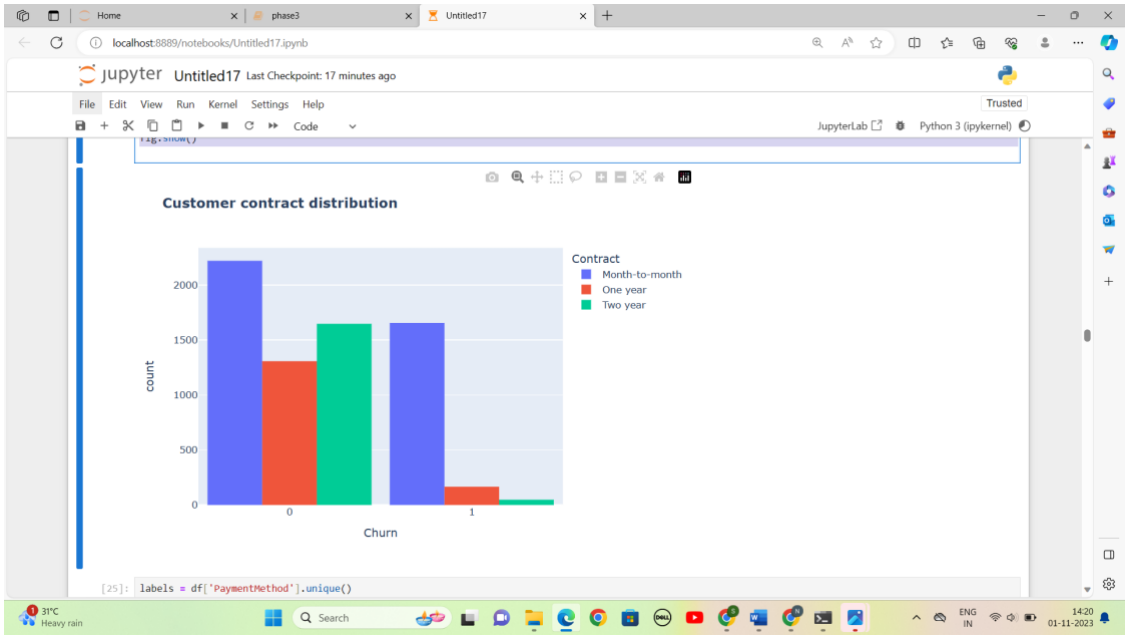

```
plt.figure(figsize=(6, 6))
labels =["Churn: Yes","Churn:No"]
values = [1869,5163]
labels_gender = ["F","M","F","M"]
sizes_gender = [939,930 , 2544,2619]
```

```
colors = ['#ff6666', '#66b3ff']

colors_gender = ['#c2c2f0','#ffb3e6', '#c2c2f0','#ffb3e6']

explode = (0.3,0.3)

explode_gender = (0.1,0.1,0.1,0.1)

textprops = {"fontsize":15}

#Plot

plt.pie(values,
labels=labels,autopct='%1.1f%%',pctdistance=1.08,
labeldistance=0.8,colors=colors, startangle=90,frame=True,
explode=explode,radius=10, textprops =textprops, counterclock
= True, )

plt.pie(sizes_gender,labels=labels_gender,colors=colors_gend
er,startangle=90, explode=explode_gender,radius=7, textprops
=textprops, counterclock = True, )

#Draw circle

centre_circle = plt.Circle((0,0),5,color='black',
fc='white',linewidth=0)

fig = plt.gcf()

fig.gca().add_artist(centre_circle)


plt.title('Churn Distribution w.r.t Gender: Male(M), Female(F)',
fontsize=15, y=1.1)


# show plot


plt.axis('equal')

plt.tight_layout()
```

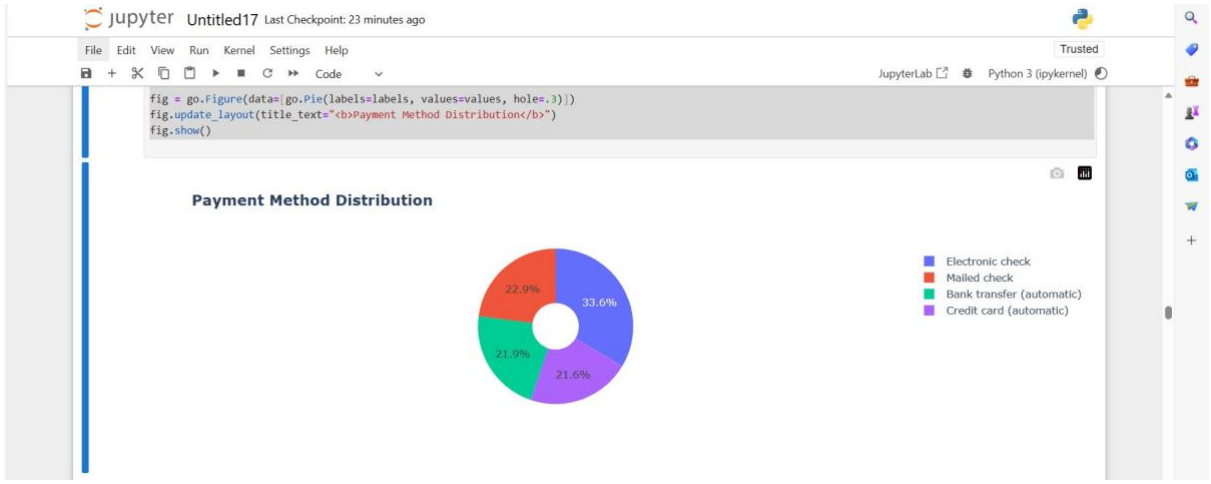
```
plt.show()
```



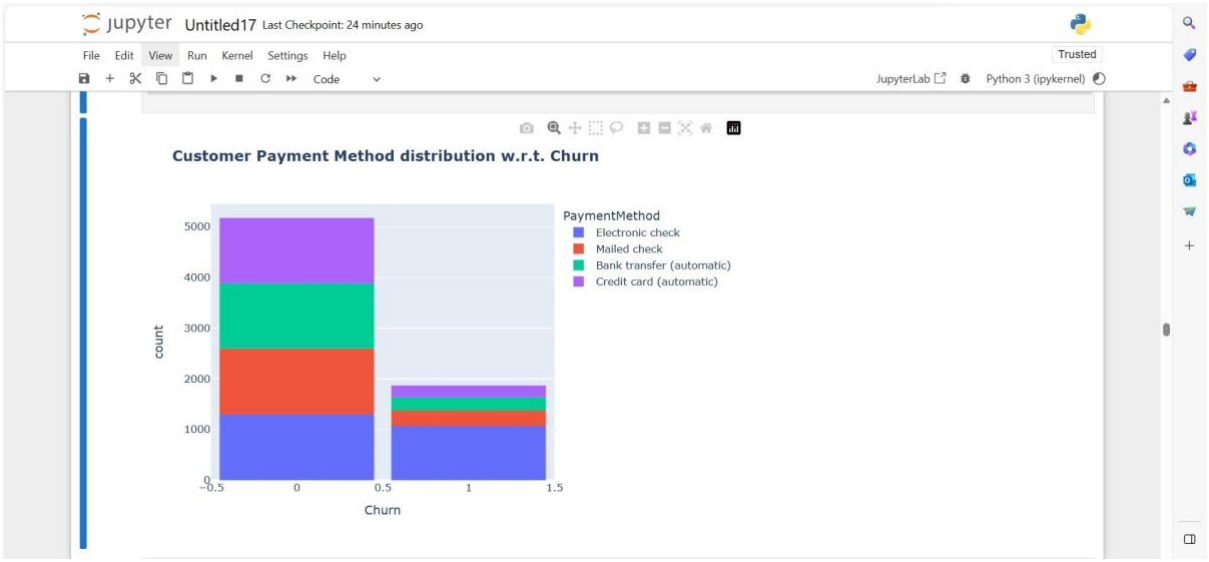

```
fig = px.histogram(df, x="Churn", color="Contract", barmode="group", title="<b>Customer contract distribution<b>")

fig.update_layout(width=700, height=500, bargap=0.1)

fig.show()


labels = df['PaymentMethod'].unique()

values = df['PaymentMethod'].value_counts()
```




```
labels = df['PaymentMethod'].unique()

values = df['PaymentMethod'].value_counts()


fig = go.Figure(data=[go.Pie(labels=labels, values=values, hole=.3)])

fig.update_layout(title_text="<b>Payment Method Distribution</b>")

fig.show()
```



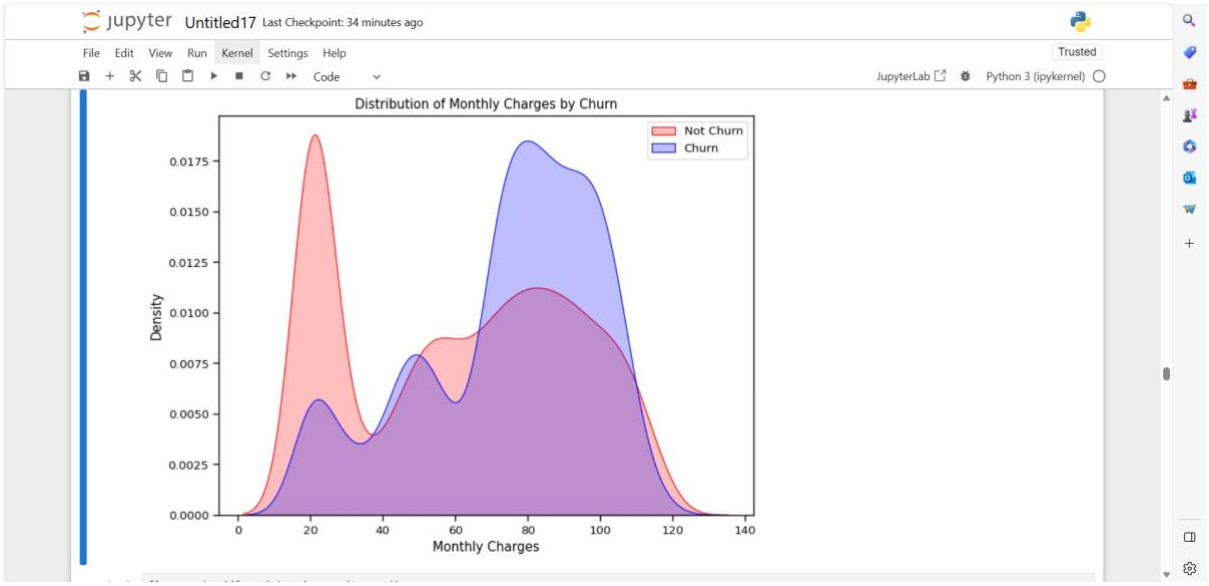
```
fig = px.histogram(df, x="Churn", color="PaymentMethod",
title="<b>Customer Payment Method distribution w.r.t.
Churn</b>")
fig.update_layout(width=700, height=500, bargap=0.1)
fig.show()
```




```
fig = go.Figure()
fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [965, 992, 219, 240],
  name = 'DSL',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
```

```
  y = [889, 910, 664, 633],
  name = 'Fiber optic',
))

fig.add_trace(go.Bar(
  x = [['Churn:No', 'Churn:No', 'Churn:Yes', 'Churn:Yes'],
       ["Female", "Male", "Female", "Male"]],
  y = [690, 717, 56, 57],
  name = 'No Internet',
))

fig.update_layout(title_text="<b>Churn Distribution w.r.t. Internet Service and Gender</b>")

fig.show()
```

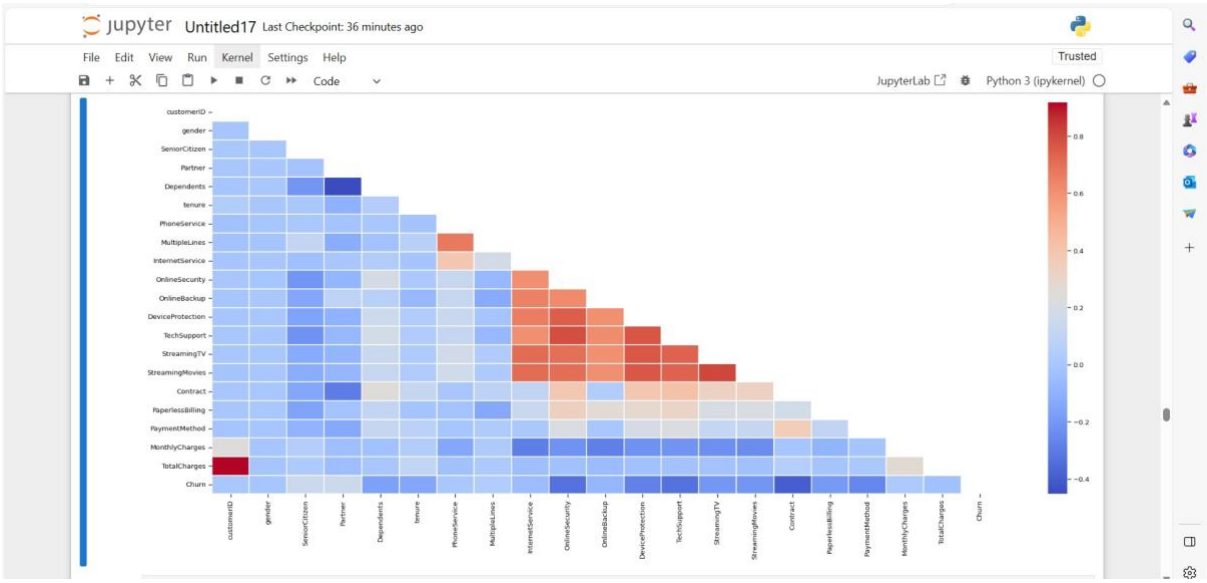
```
sns.set_context("paper", font_scale=1.1)
plt.figure(figsize=(8, 6))

# Line plot for customers who do not churn (Churn = 0)
sns.kdeplot(df.MonthlyCharges[df['Churn'] == 0], color='red',
label='Not Churn', shade=True)

# Line plot for customers who churn (Churn = 1)
sns.kdeplot(df.MonthlyCharges[df['Churn'] == 1], color='blue',
label='Churn', shade=True)

plt.xlabel('Monthly Charges')
plt.ylabel('Density')
plt.title('Distribution of Monthly Charges by Churn')
plt.legend()

plt.show()
```



```
fig = px.box(df, x='Churn', y = 'tenure')

# Update yaxis properties
fig.update_yaxes(title_text='Tenure (Months)', row=1, col=1)
# Update xaxis properties
fig.update_xaxes(title_text='Churn', row=1, col=1)

# Update size and title
fig.update_layout(autosize=True, width=750, height=600,
    title_font=dict(size=25, family='Courier'),
    title='<b>Tenure vs Churn</b>',
)

fig.show()
```


```
# Correlation between all variables
```

```
plt.figure(figsize=(25, 10))

corr = df.apply(lambda x: pd.factorize(x)[0]).corr()

mask = np.triu(np.ones_like(corr, dtype=bool))

ax = sns.heatmap(corr, mask=mask, xticklabels=corr.columns,
yticklabels=corr.columns, annot=True, linewidths=.2,
cmap='coolwarm')
```

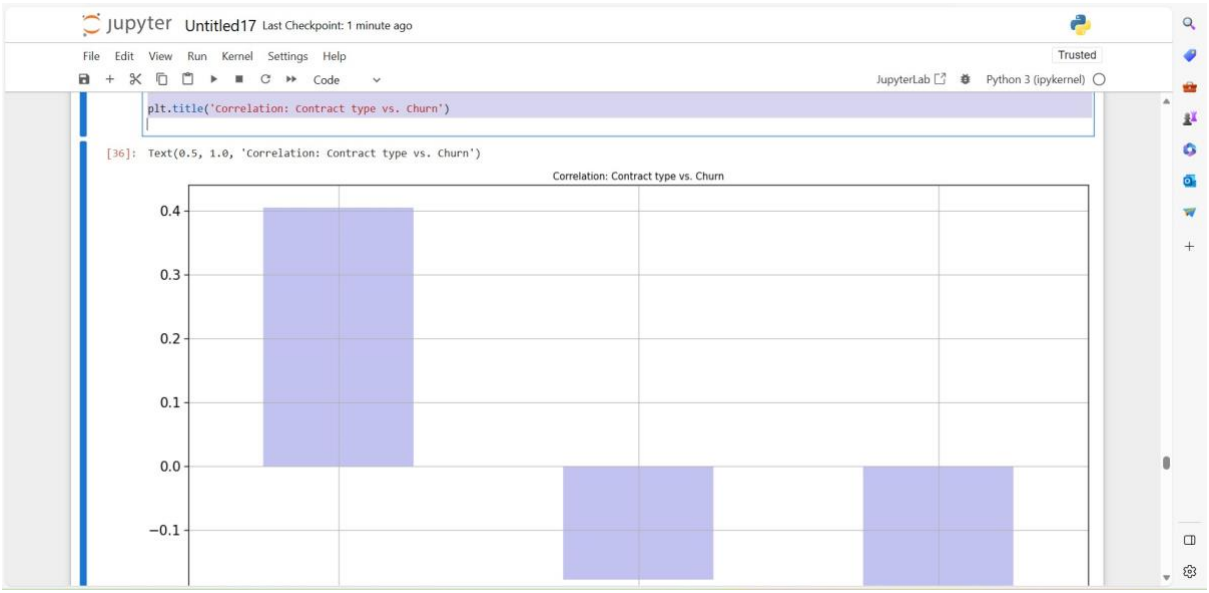



```
# Correlation between churn and selected boolean and numeric
variables

plt.figure

ds_corr = df[['SeniorCitizen', 'Partner', 'Dependents',
        'tenure', 'PhoneService', 'PaperlessBilling',
         'MonthlyCharges', 'TotalCharges']]
```

```
correlations = ds_corr.corrwith(df.Churn)
correlations = correlations[correlations!=1]
correlations.plot.bar(
        figsize = (18, 10),
        fontsize = 15,
        color = '#c2c2f0',
        rot = 45, grid = True)

plt.title('Correlation with Churn Rate',
horizontalalignment="center", fontstyle = "normal", fontsize =
"22", fontfamily = "sans")
```

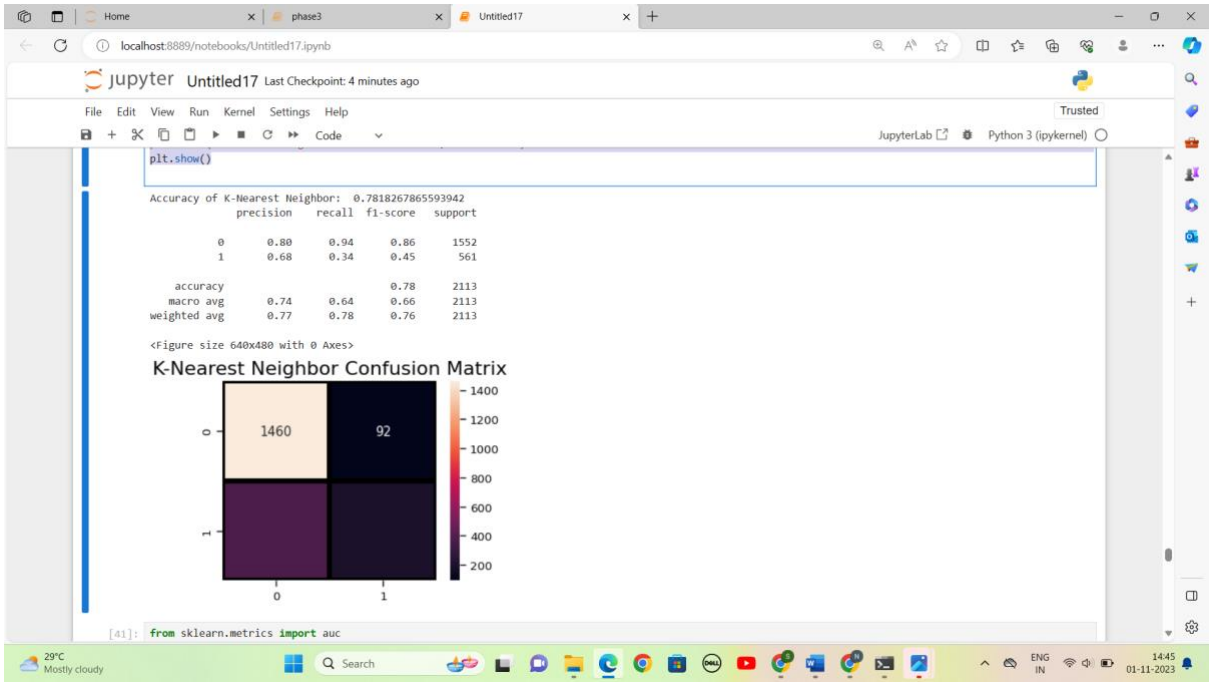

```
# Correlation: Contract type vs. Churn
plt.figure

ds_contract_type_corr = dataset[['Contract_Month-to-month',
'Contract_One year', 'Contract_Two year']]
```

```
correlations = ds_contract_type_corr.corrwith(dataset.Churn)
correlations = correlations[correlations!=1]
correlations.plot.bar(
        figsize = (18, 10),
        fontsize = 15,
        color = '#c2c2f0',
        rot = 45, grid = True)

plt.title('Correlation: Contract type vs. Churn')
```

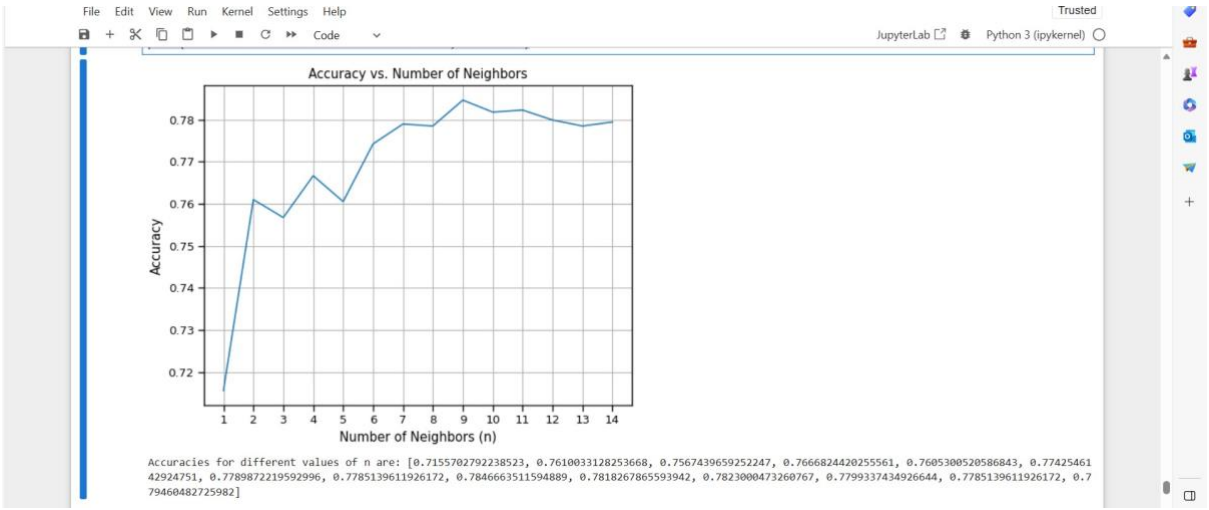



```
knn_model = KNeighborsClassifier(n_neighbors = 10)

knn_model.fit(X_train,y_train)

# Evaluate model
accuracy_knn = knn_model.score(X_test,y_test)
```

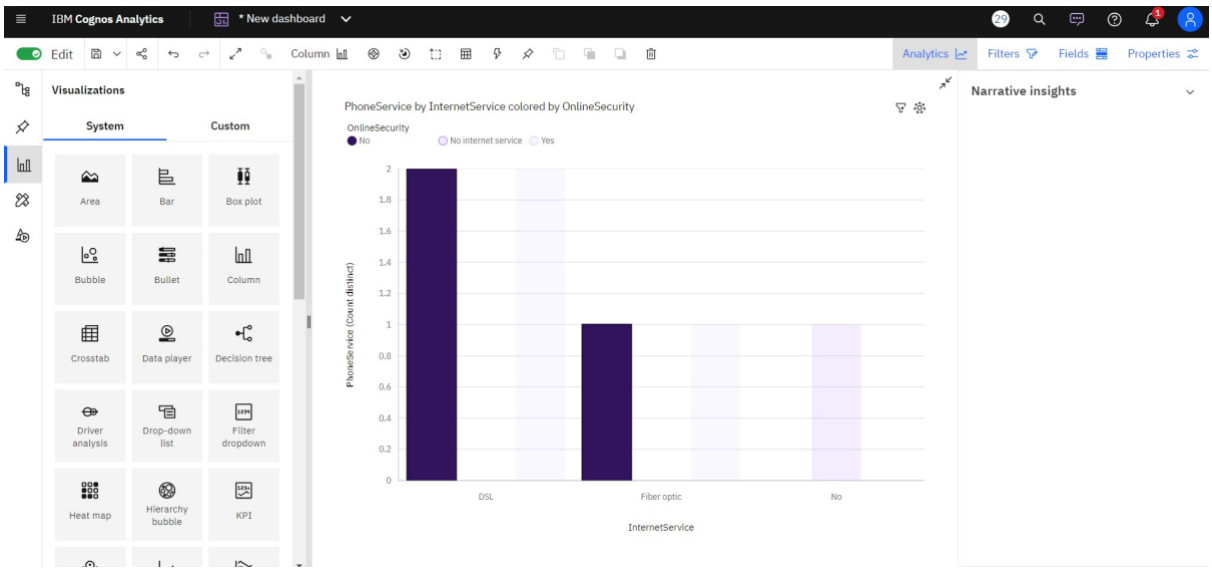
```
print("Accuracy of K-Nearest Neighbor: ", accuracy_knn)

# Classification report
knn_prediction = knn_model.predict(X_test)
print(classification_report(y_test, knn_prediction))

plt.figure(14)
plt.figure(figsize=(4,3))
sns.heatmap(confusion_matrix(y_test, knn_prediction),
            annot=True, fmt = "d", linecolor="k", linewidths=3)

plt.title("K-Nearest Neighbor Confusion Matrix", fontsize=16)
plt.show()
```



```
a_index = list(range(1, 15))
accuracies = []  # Initialize an empty list to store accuracy values

for i in a_index:
    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train, y_train)
    prediction = model.predict(X_test)
    accuracy = metrics.accuracy_score(y_test, prediction)
    accuracies.append(accuracy)

# Plot the accuracy values
import matplotlib.pyplot as plt

plt.plot(a_index, accuracies)
plt.xticks(a_index)
plt.xlabel('Number of Neighbors (n)')
plt.ylabel('Accuracy')
plt.title('Accuracy vs. Number of Neighbors')
plt.grid(True)
plt.show()

print('Accuracies for different values of n are:', accuracies)
```

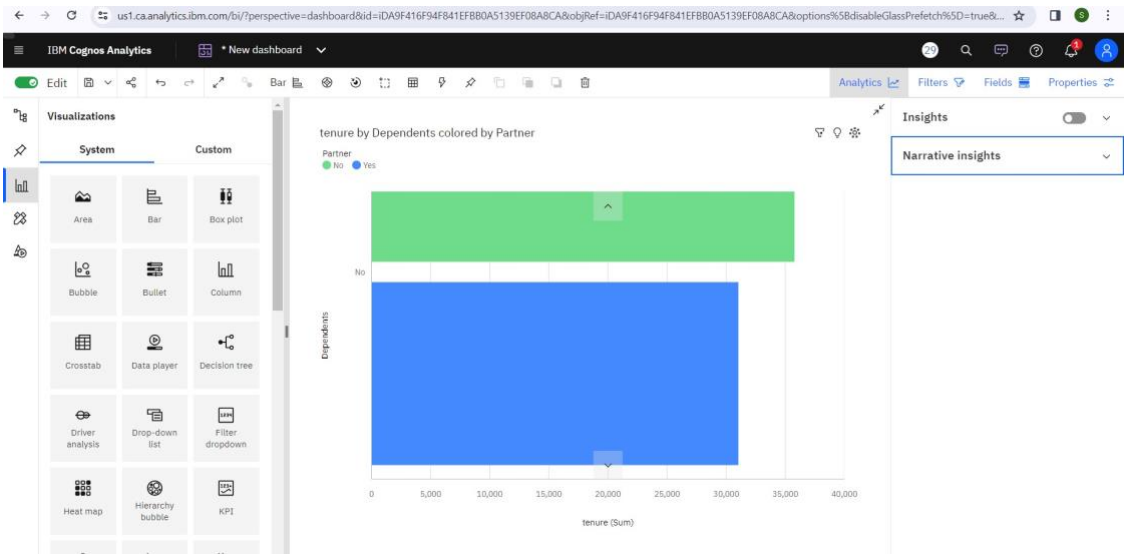


Create interactive dashboards and reports in IBM Cognos to visualize churn patterns, retention rates, and key factors influencing churn.



Fiber optic (**44** %) and **DSL** (**34.4** %) are the most frequently occurring categories of **InternetService** with a combined count of **5517** items with **PhoneService** values (**78.3** % of the total).

InternetService DSL has the highest **PhoneService** due to **OnlineSecurity No**.

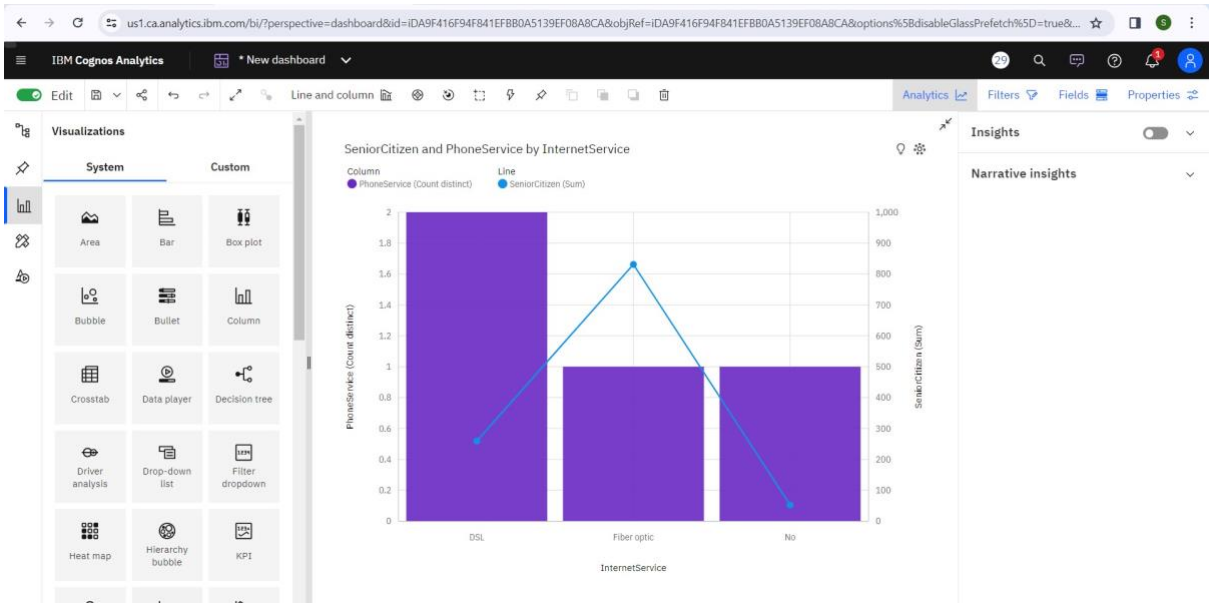


Dependents No has the highest total **tenure** due to **Partner No**.

Partner Yes has the highest total **tenure** due to **MultipleLines Yes**.

Partner Yes has the highest **tenure** at **almost 143 thousand**, out of which **Dependents Yes** contributed the most at **over 72 thousand**.

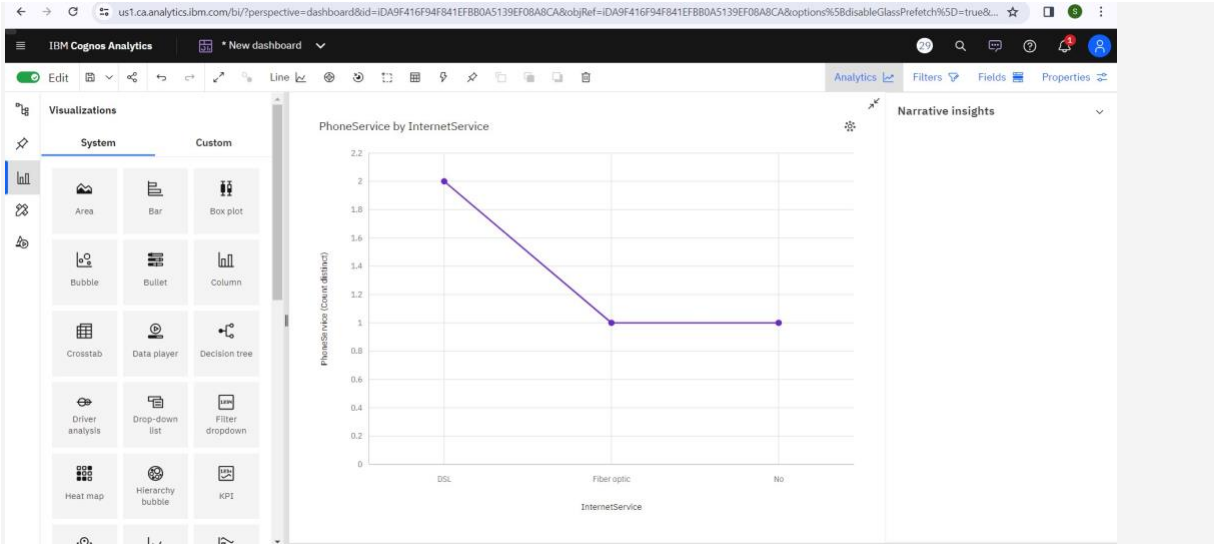
For **tenure**, the most significant value of **Dependents** is **No**, whose respective **tenure** values add up to **over 147 thousand**, or **64.5** % of the total.



InternetService Fiber optic has the highest **Total TotalCharges** but is ranked #**2** in **Count distinct PhoneService**.

Fiber optic (**44** %) and **DSL** (**34.4** %) are the most frequently occurring categories of **InternetService** with a combined count of **5517** items with **PhoneService** values (**78.3** % of the total).

SeniorCitizen is unusually high when **InternetService** is **Fiber optic**.

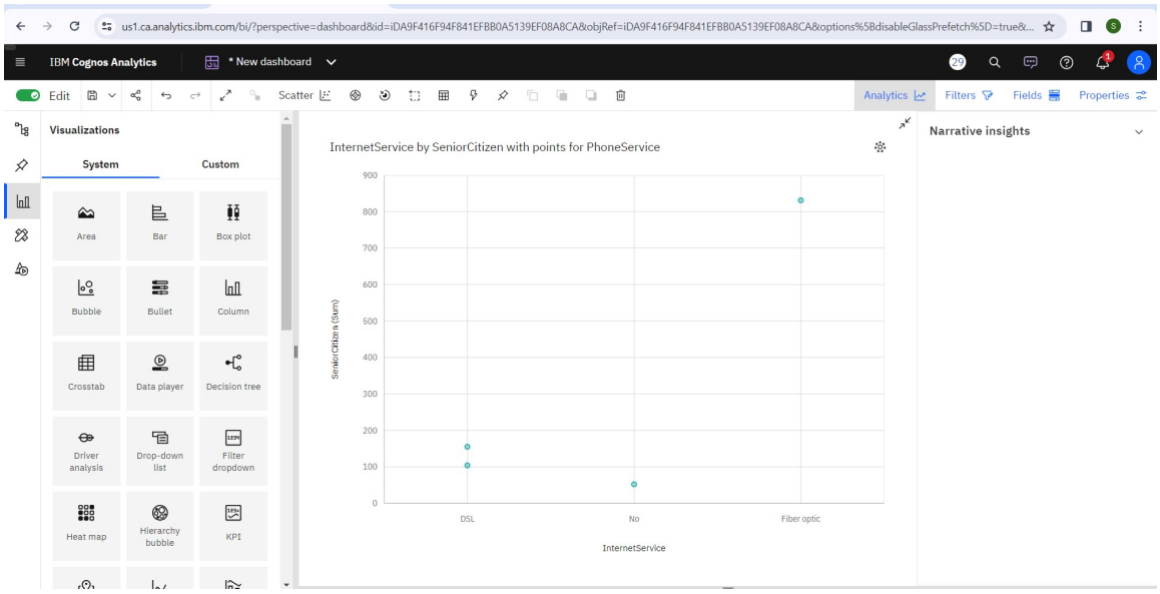


InternetService Fiber optic has the highest **Total TotalCharges** but is ranked #**2** in **Count distinct PhoneService**.

Fiber optic (**44** %) and **DSL** (**34.4** %) are the most frequently occurring categories of **InternetService** with a combined count of **5517** items with **PhoneService** values (**78.3** % of the total).

The total number of results for **PhoneService**, across all **InternetService**, is **over seven thousand**.

InternetService DSL has the highest **Count distinct PhoneService** but is ranked #**2** in **Total TotalCharges**.

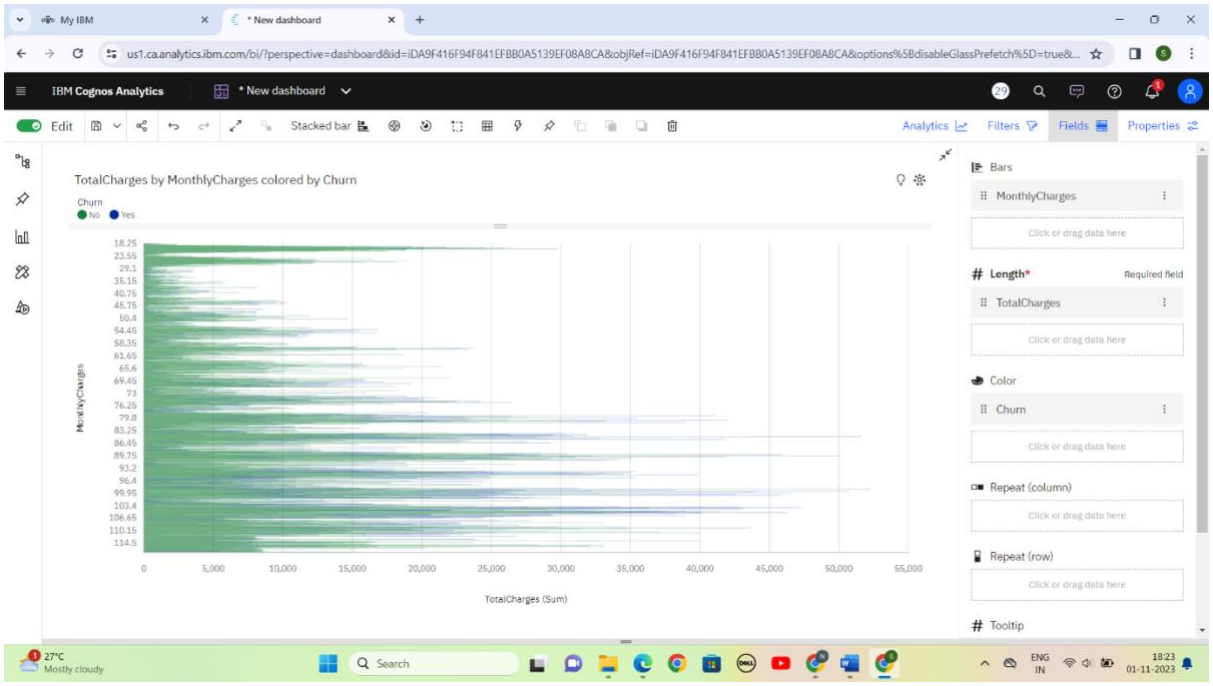


PhoneService Yes has the highest total **SeniorCitizen** due to **InternetService Fiber optic**.

Across all values of **InternetService**, the sum of **SeniorCitizen** is **over a thousand**.

SeniorCitizen ranges from **52**, when **InternetService** is **No**, to **831**, when **InternetService** is **Fiber optic**.

For **SeniorCitizen**, the most significant value of **PaymentMethod** is **Electronic check**, whose respective **SeniorCitizen** values add up to **594**, or **52** % of the total.



TotalCharges is unusually high when the combination of **MonthlyCharges** and **Churn** is **99 and No**.

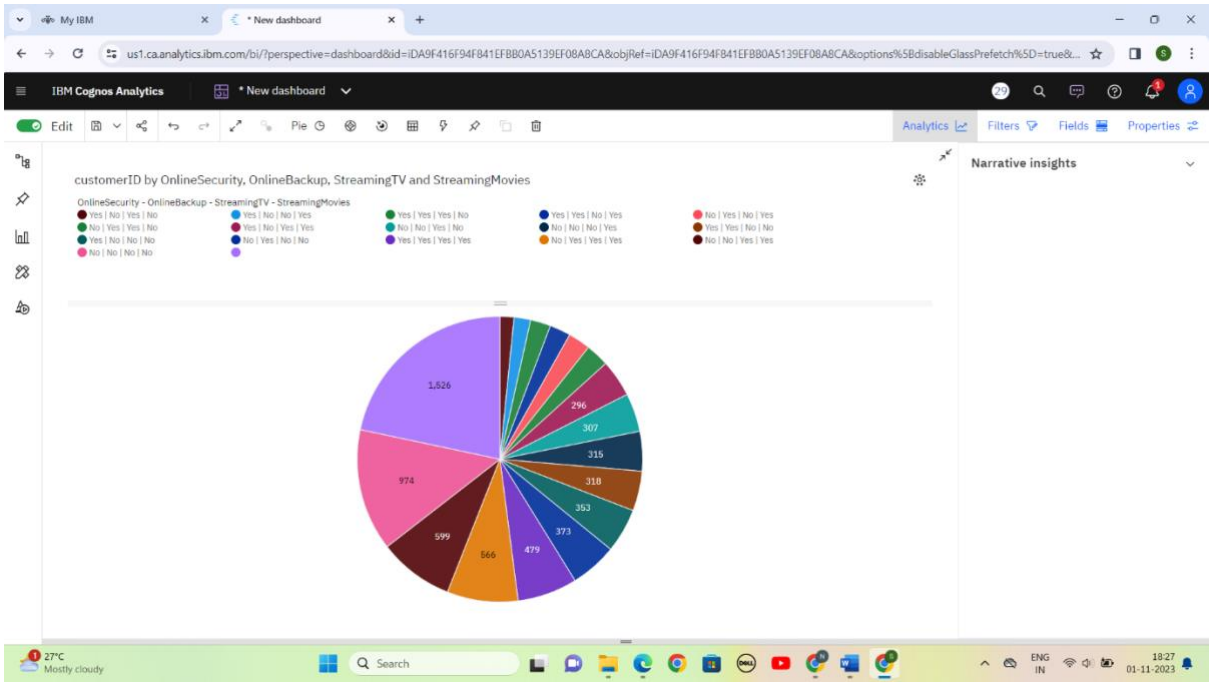
MultipleLines Yes has the highest **TotalCharges** at **over 10 million**, out of which **Churn No** contributed the most at **almost 8.4 million**.

TotalCharges is unusually high when the combinations of **MonthlyCharges** and **SeniorCitizen** are **84.8 and 0, 100.3 and 0** and **100.55 and 0**.

Yes MultipleLines accounted for **70**% of **99.0 TotalCharges** compared to **64**% for **84.8**.

SeniorCitizen 0 has the highest **TotalCharges** at **almost 13 million**, out of which **MonthlyCharges 84.8** contributed the most at **nearly 42 thousand**.

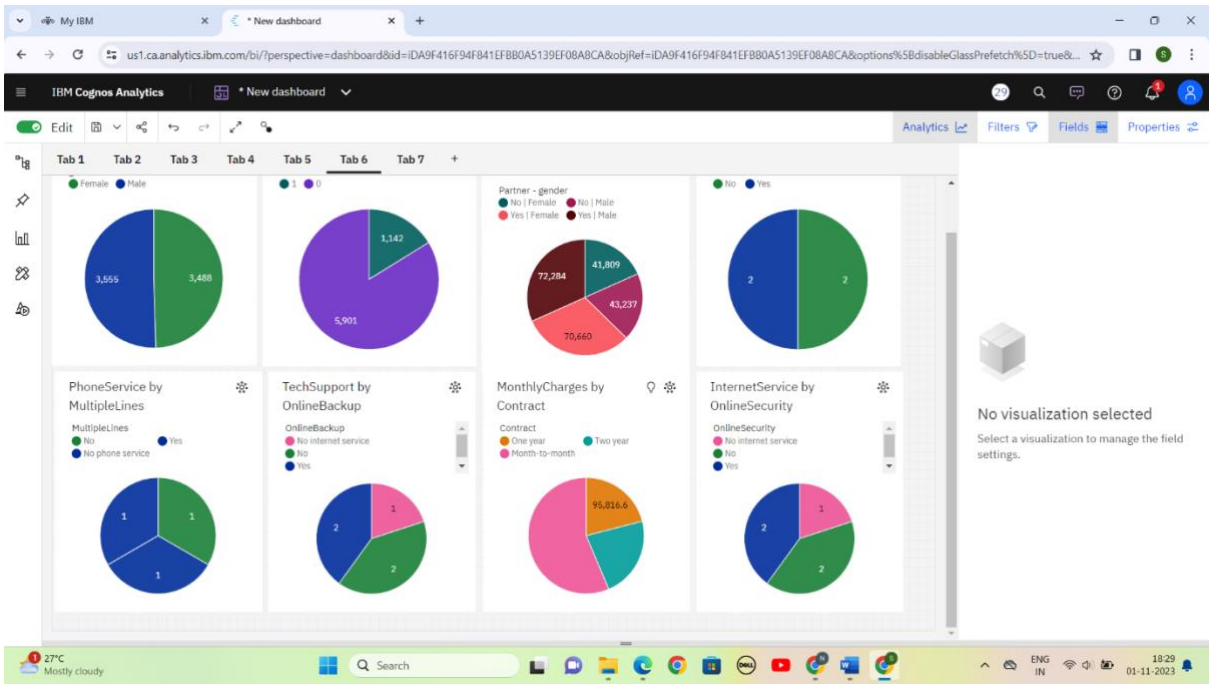
For **TotalCharges**, the most significant values of **MonthlyCharges** are **99**, **84.8**, **99.5**, and **89.85**, whose respective **TotalCharges** values add up to **almost 205 thousand**, or **1.3** % of the total.



No internet service|No internet service|No internet service|No internet service is the most frequently occurring category of **OnlineSecurity - OnlineBackup - StreamingTV - StreamingMovies** with a count of **1526** items with **customerID** values (**21.7** % of the total).

The total number of results for **customerID**, across all **OnlineSecurity - OnlineBackup - StreamingTV - StreamingMovies**, is **over seven thousand**.

MultipleLines No has the highest **customerID** at **almost 3500**, out of which **OnlineBackup No** contributed the most at **almost 1500**.

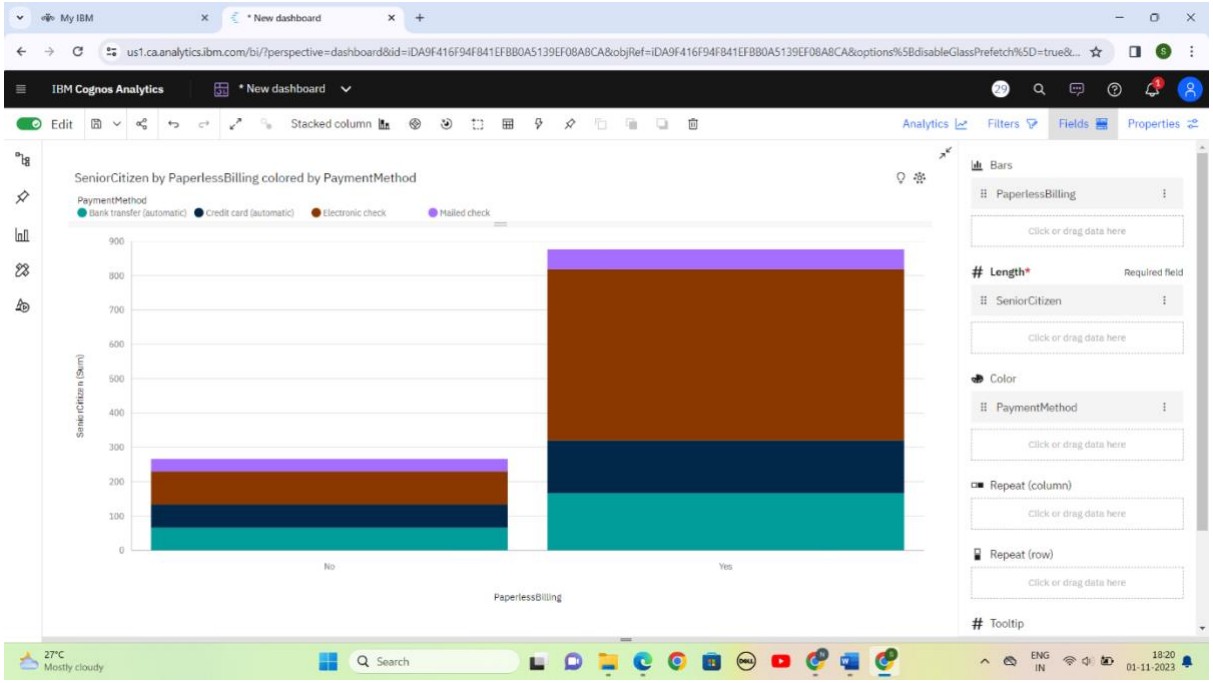


gender Male has the highest **tenure** at **nearly 116 thousand**, out of which **Partner Yes** contributed the most at **over 72 thousand**.

For **tenure**, the most significant values of **Partner - gender** are **Yes|Male** and **Yes|Female**, whose respective **tenure** values add up to **almost 143 thousand**, or **62.7** % of the total.

MultipleLines Yes has the highest **tenure** at **nearly 125 thousand**, out of which **gender Male** contributed the most at **over 62 thousand**.

InternetService Fiber optic has the highest values of both **SeniorCitizen** and **TotalCharges**.

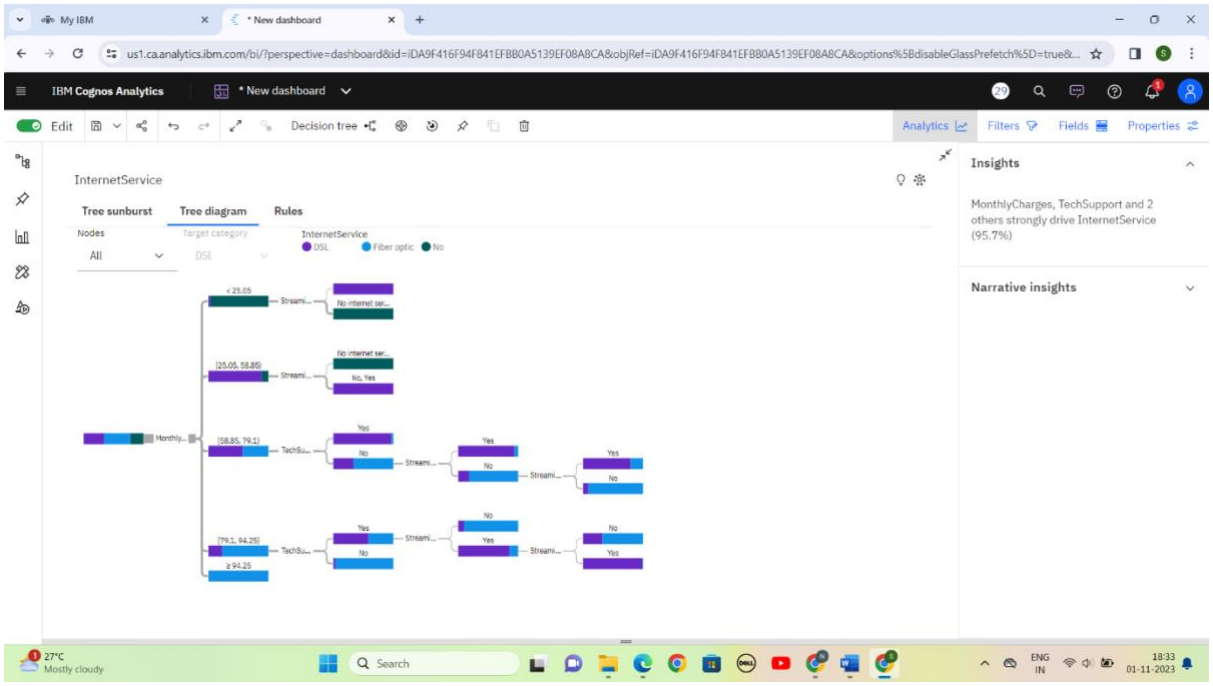


PaperlessBilling Yes has the highest total SeniorCitizen due to PaymentMethod Electronic check.

PaperlessBilling Yes has the highest values of both SeniorCitizen and TotalCharges.

SeniorCitizen is unusually high when the combination of PaperlessBilling and PaymentMethod is Yes and Electronic check.

For SeniorCitizen, the most significant value of PaperlessBilling is Yes, whose respective SeniorCitizen values add up to 876, or 76.7 % of the total.



InternetService No has the lowest total **TotalCharges** at **over 1.0 million**, followed by **DSL** at **over 5.1 million**.
Add insight to favorites

InternetService Fiber optic has the highest total **TotalCharges** at **over 9.9 million**, followed by **DSL** at **over 5.1 million**.