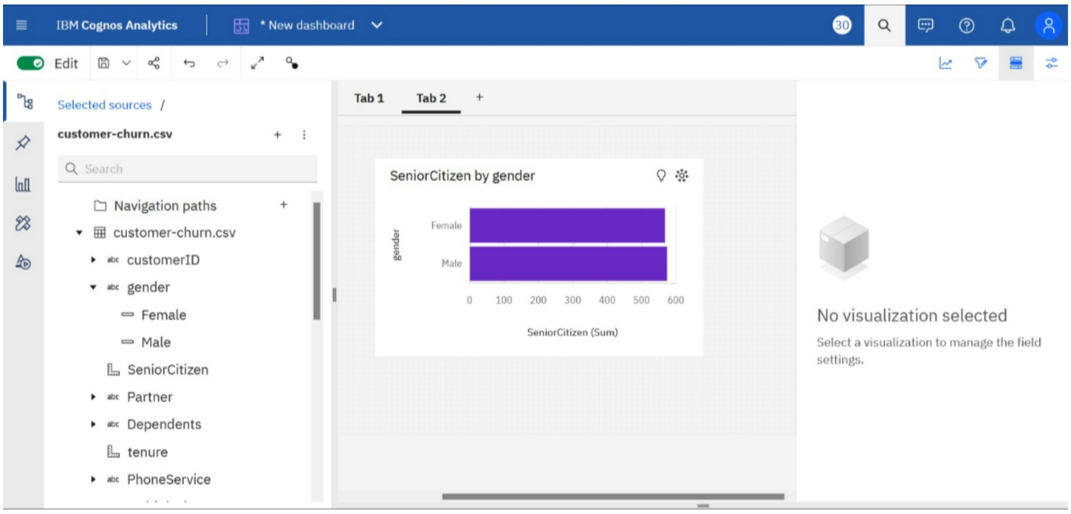
Phase 4

Customer Churn Prediction

Using IBM Cognos for data visualization with a dataset (Customer Churn Prediction) is a straightforward process that can provide valuable insights for businesses. First, you can import your dataset into Cognos, which supports various data sources, such as spreadsheets, databases, and even cloud-based data. Once your data is loaded, you can begin creating compelling visualizations. Cognos offers a wide array of visualization options, from basic bar charts and line graphs to more advanced options like heat maps, geographical maps, and interactive dashboards.

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

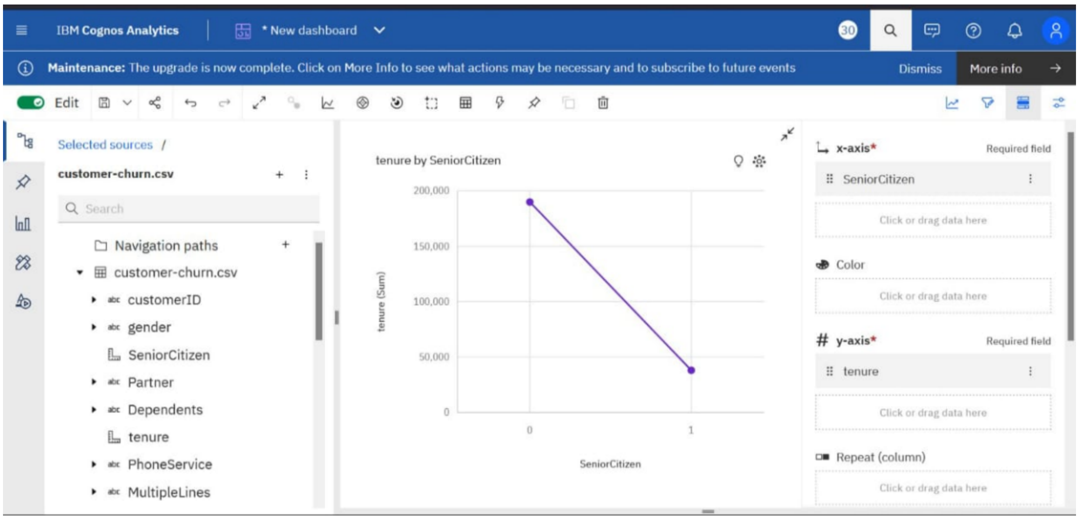
1.Bar Chart: A bar chart or bar graph is a chart or graph that
presents categorical data with rectangular bars
with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally. A vertical bar chart is sometimes called a column chart.



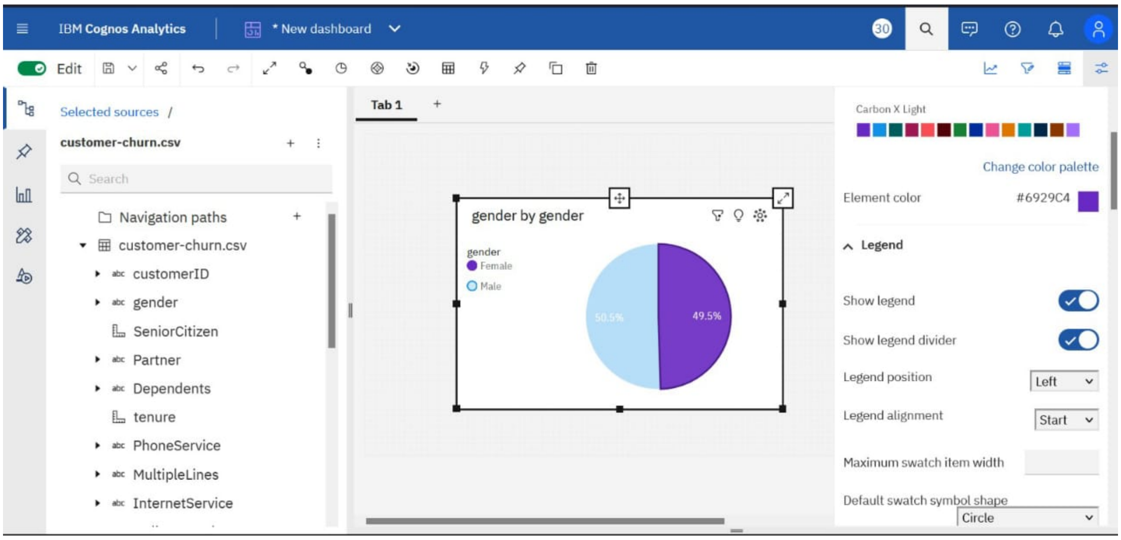
No visualization selected

Select a visualization to manage the field settings.

2. Line Chart: A line chart is a type of chart used to show information that changes over time. Line charts are created by plotting a series of several points and connecting them with a straight line. Line charts are used to track changes over short and long periods.

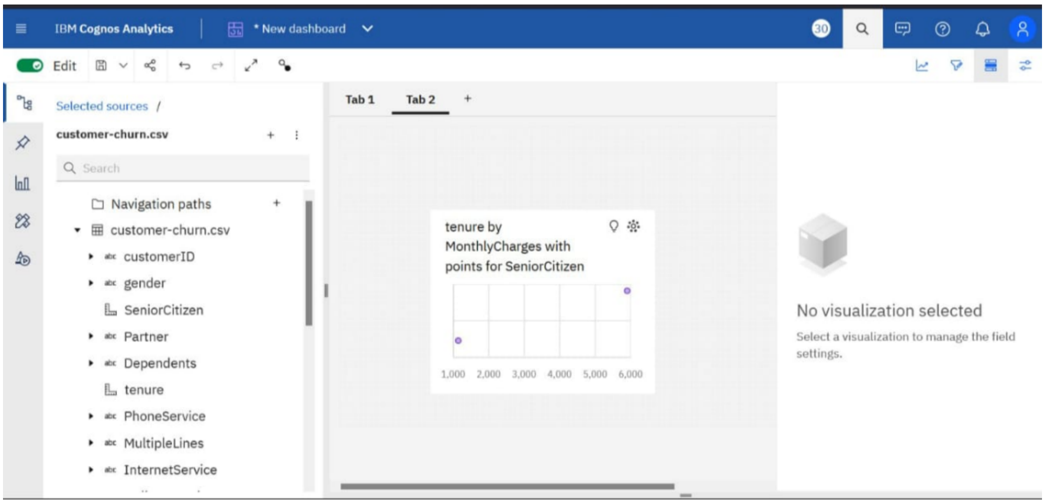


3.Pie Chart: A pie chart is a type of graph that represents the data in the circular graph. The slices of pie show the relative size of the data, and it is a type of pictorial representation of data. A pie chart requires a list of categorical variables and numerical variables. Here, the term “pie” represents the whole, and the “slices” represent the parts of the whole.



4.Box Plot: A box and whisker plot—also called a box plot—displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum.

In a box plot, we draw a box from the first quartile to the third quartile. A vertical line goes through the box at the median. The whiskers go from each quartile to the minimum or maximum.

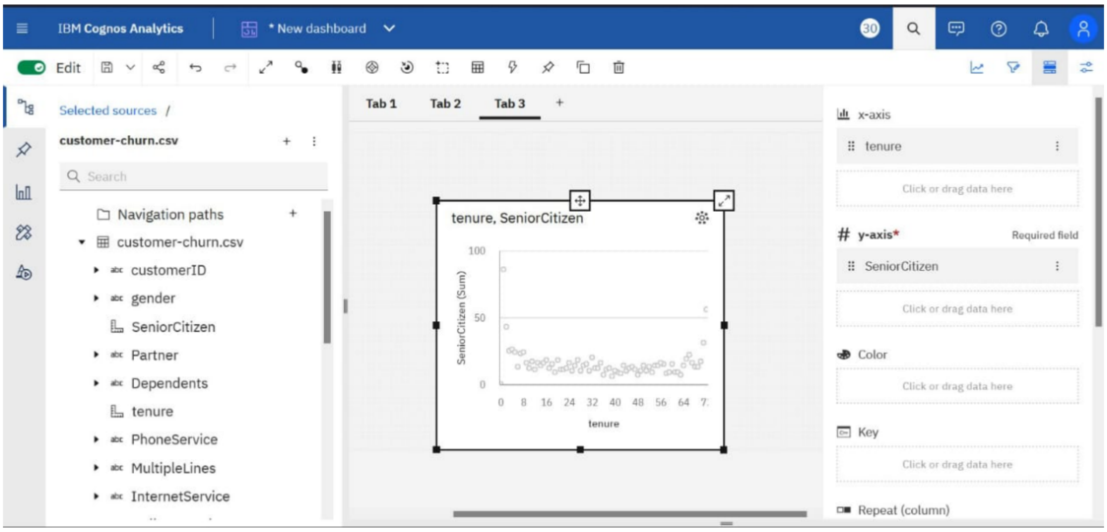


tenure by MonthlyCharges with points for SeniorCitizen

No visualization selected

Select a visualization to manage the field settings.

5. Scatter Plot: A scatter plot (aka scatter chart, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.



tenure, SeniorCitizen

x-axis

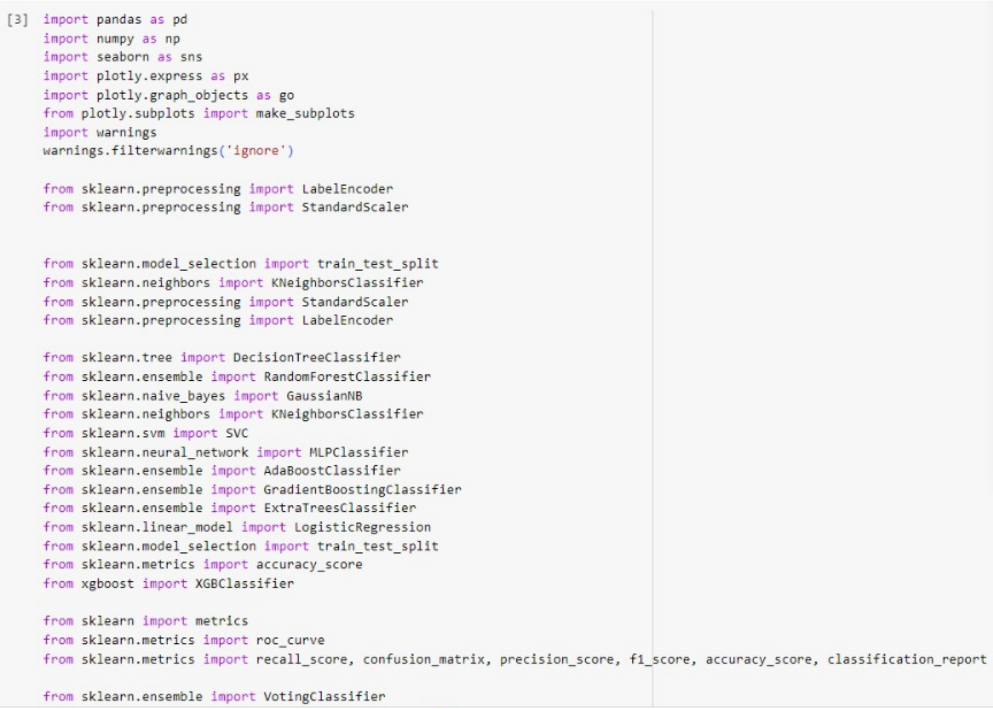
tenure

y-axis* Required field

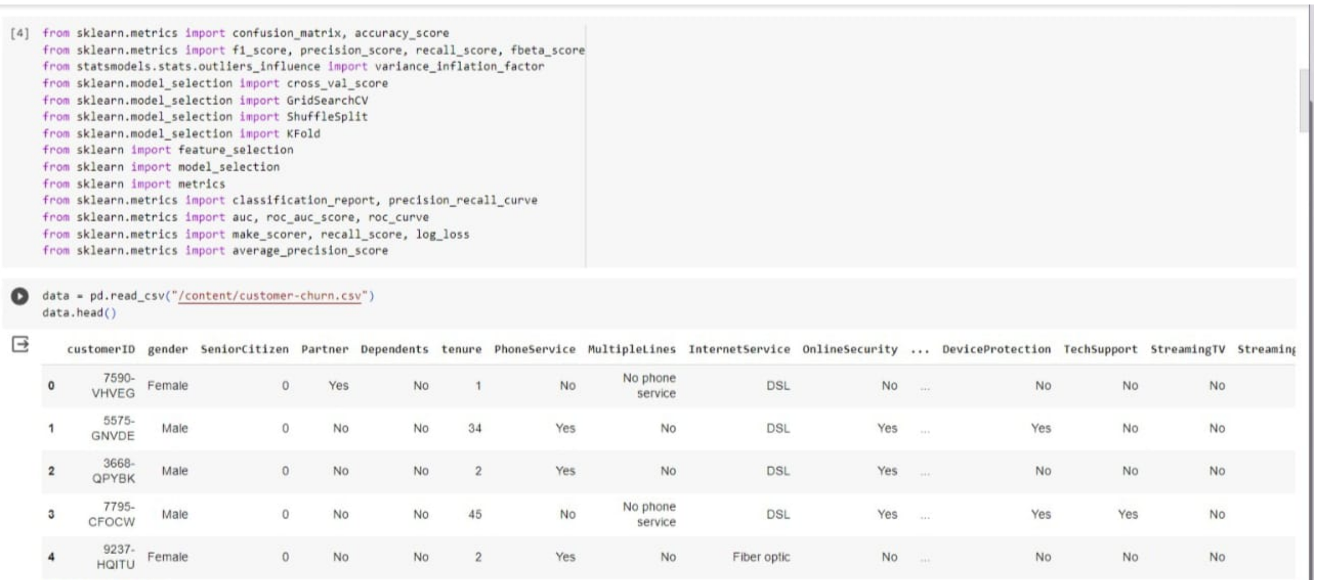
SeniorCitizen

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

1.importing the modules and packages.



```
[3] import pandas as pd
    import numpy as np
    import seaborn as sns
    import plotly.express as px
    import plotly.graph_objects as go
    from plotly.subplots import make_subplots
    import warnings
    warnings.filterwarnings('ignore')

    from sklearn.preprocessing import LabelEncoder
    from sklearn.preprocessing import StandardScaler


    from sklearn.model_selection import train_test_split
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.preprocessing import StandardScaler
    from sklearn.preprocessing import LabelEncoder

    from sklearn.tree import DecisionTreeClassifier
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.naive_bayes import GaussianNB
    from sklearn.neighbors import KNeighborsClassifier
    from sklearn.svm import SVC
    from sklearn.neural_network import MLPClassifier
    from sklearn.ensemble import AdaBoostClassifier
    from sklearn.ensemble import GradientBoostingClassifier
    from sklearn.ensemble import ExtraTreesClassifier
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import accuracy_score
    from xgboost import XGBClassifier

    from sklearn import metrics
    from sklearn.metrics import roc_curve
    from sklearn.metrics import recall_score, confusion_matrix, precision_score, f1_score, accuracy_score, classification_report

    from sklearn.ensemble import VotingClassifier
```

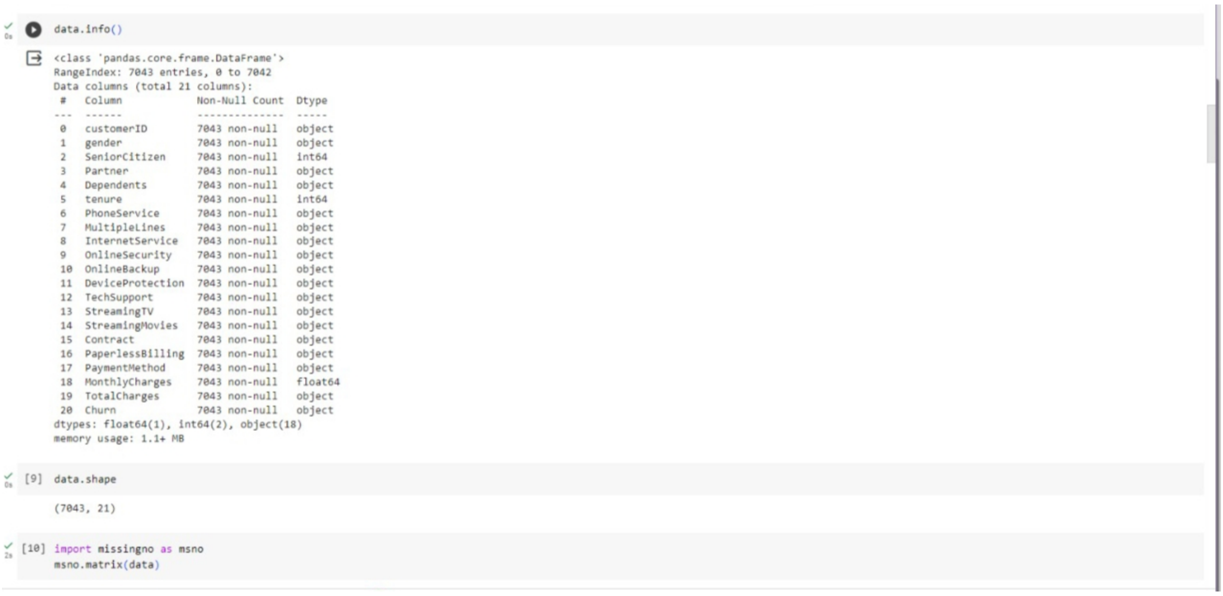


```
[4] from sklearn.metrics import confusion_matrix, accuracy_score
    from sklearn.metrics import f1_score, precision_score, recall_score, fbeta_score
    from statsmodels.stats.outliers_influence import variance_inflation_factor
    from sklearn.model_selection import cross_val_score
    from sklearn.model_selection import GridSearchCV
    from sklearn.model_selection import ShuffleSplit
    from sklearn.model_selection import KFold
    from sklearn import feature_selection
    from sklearn import model_selection
    from sklearn import metrics
    from sklearn.metrics import classification_report, precision_recall_curve
    from sklearn.metrics import auc, roc_auc_score, roc_curve
    from sklearn.metrics import make_scorer, recall_score, log_loss
    from sklearn.metrics import average_precision_score
```

```
data = pd.read_csv("/content/customer-churn.csv")
data.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	...	DeviceProtection	TechSupport	StreamingTV	Streaming
0	7590-VHVEG	Female	0	Yes	No	1	No	No phone service	DSL	No	...	No	No	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	No	DSL	Yes	...	Yes	No	No	
2	3668-QPYBK	Male	0	No	No	2	Yes	No	DSL	Yes	...	No	No	No	
3	7795-CFOCW	Male	0	No	No	45	No	No phone service	DSL	Yes	...	Yes	Yes	No	
4	9237-HQITU	Female	0	No	No	2	Yes	No	Fiber optic	No	...	No	No	No	

2.getting the information of the dataset



```
data.info()
```

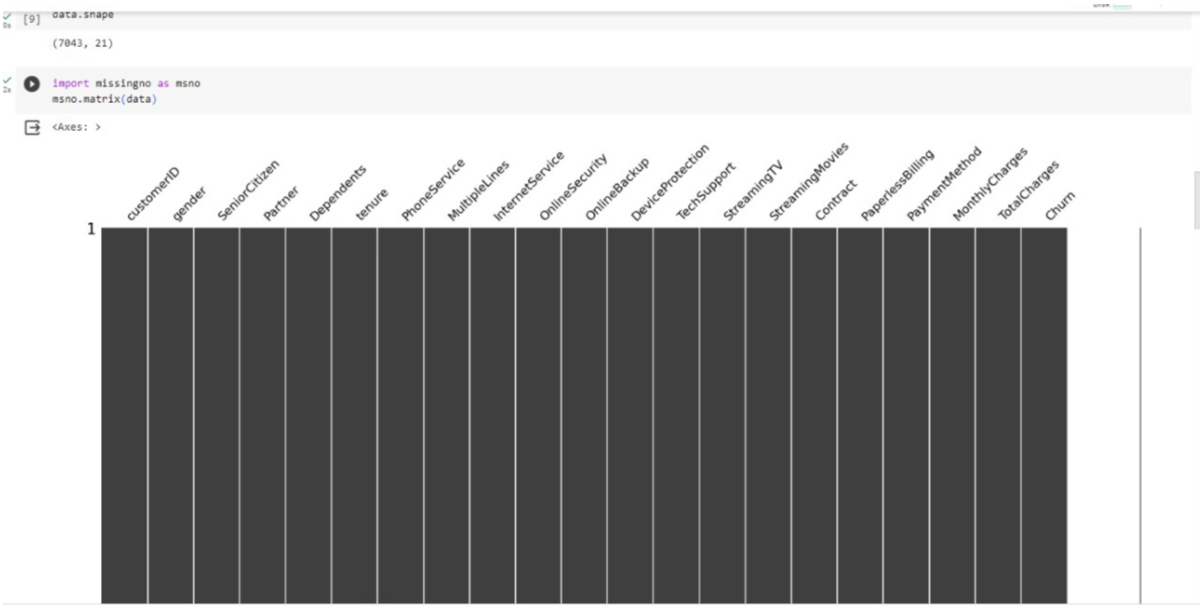
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 21 columns):
 #   Column            Non-Null Count  Dtype  
---  ------            --------------  -----  
 0   customerID        7043 non-null   object 
 1   gender            7043 non-null   object 
 2   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)
memory usage: 1.1+ MB
```

```
[9] data.shape
```

```
(7043, 21)
```

```
[10] import missingno as msno
     msno.matrix(data)
```

3.creating the matrix of the dataset



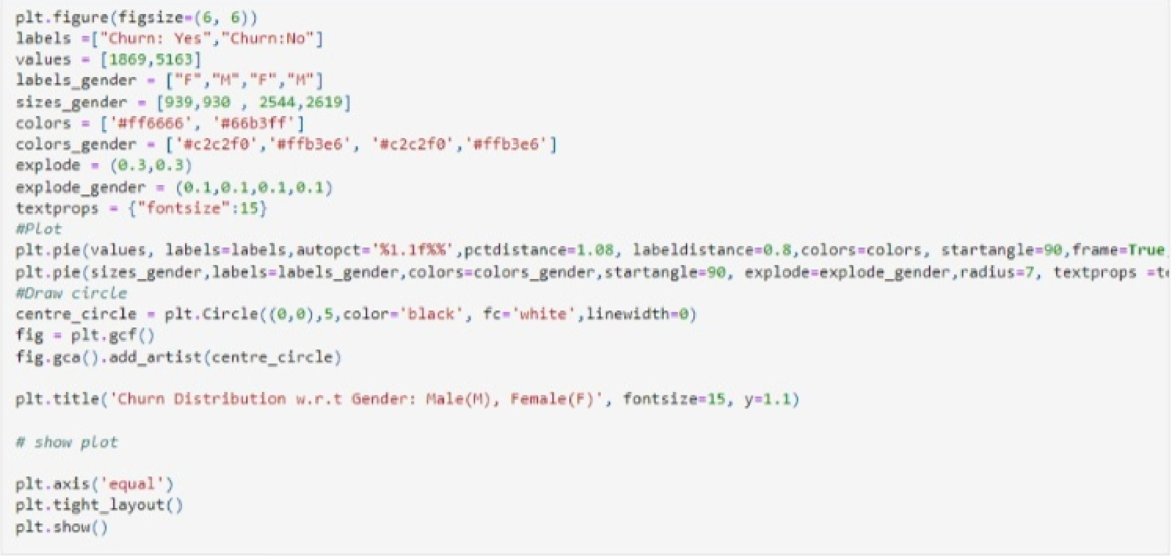
```
[9] data.shape
```

```
(7043, 21)
```

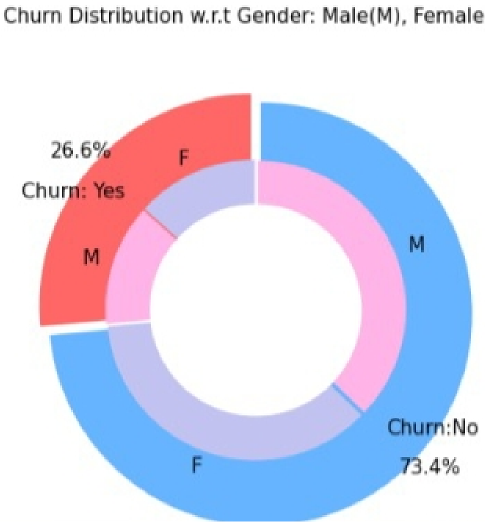
```
import missingno as msno
msno.matrix(data)
```

```
<Axes: >
```

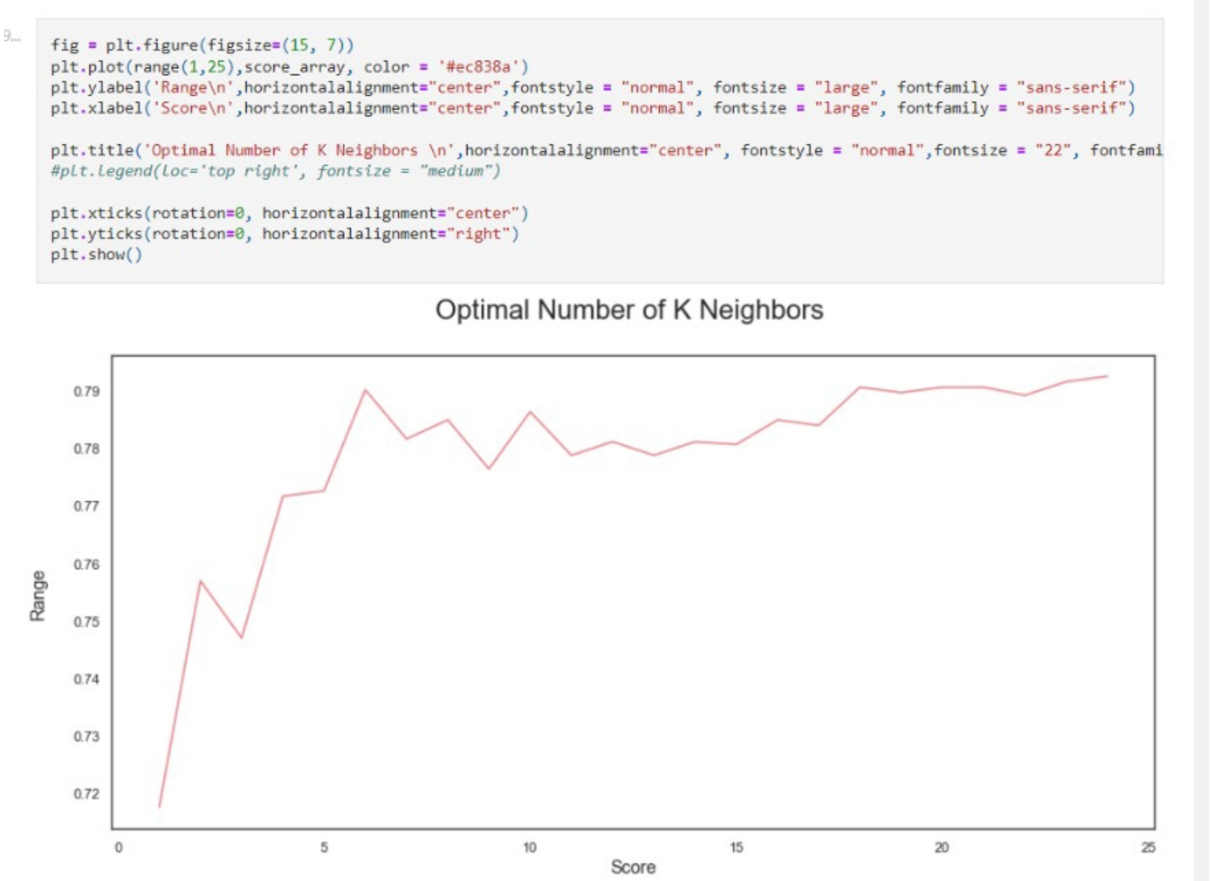
4.Analysing our dataset with the visualization of pie chart model.



```
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
plt.pie(sizes_gender,labels=labels_gender,colors=colors_gender,startangle=90, explode=explode_gender,radius=7, textprops =t
#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()
```

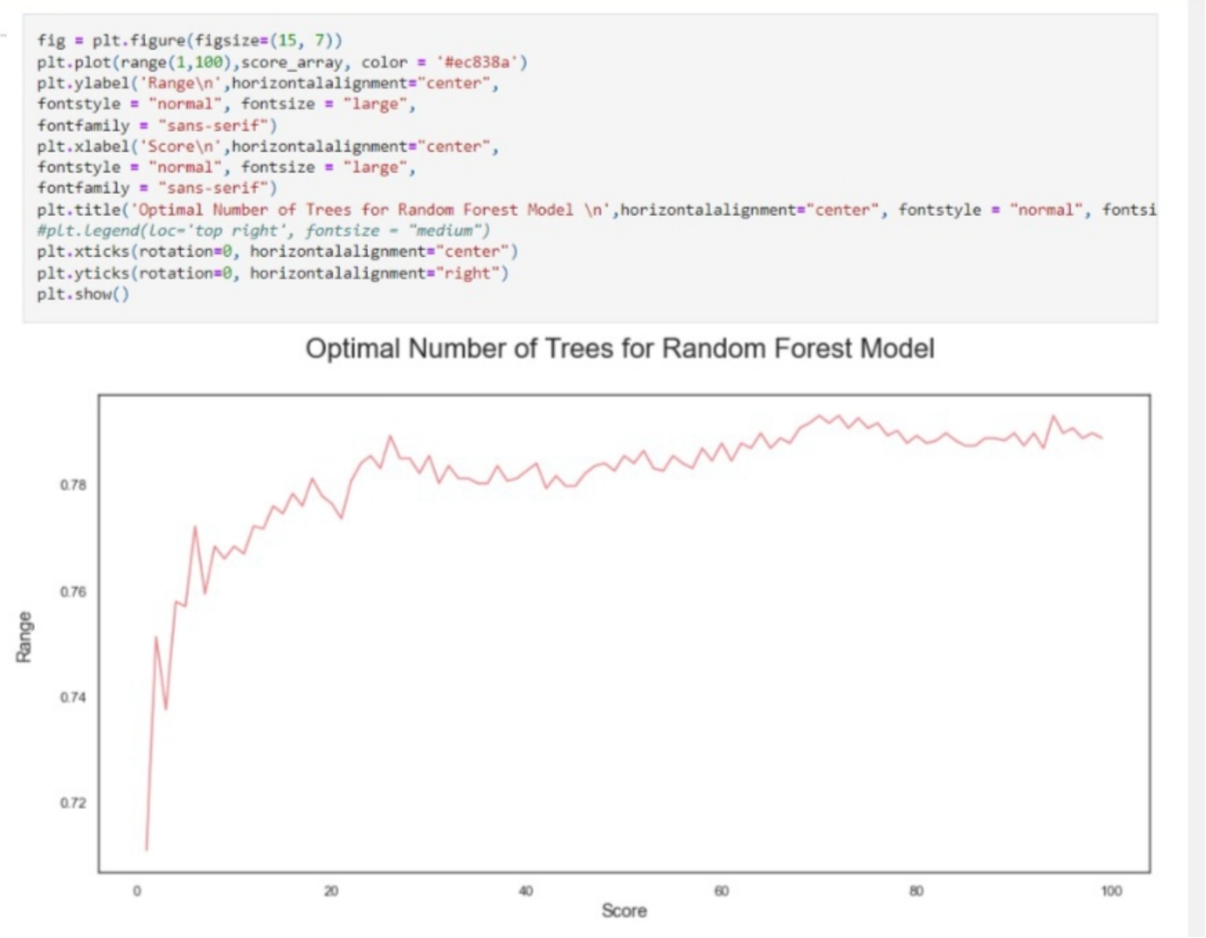


5.The k-Nearest Neighbors (KNN) algorithm is a versatile and simple machine learning model that can be applied to various classification and regression tasks. It operates on the principle of similarity, where it classifies or predicts based on the majority class or average value of its k-nearest data points in the feature space.Using the k-nearest the dataset of the customer churn is predicted.



```
fig = plt.figure(figsize=(15, 7))
plt.plot(range(1,25),score_array, color = '#ec838a')
plt.ylabel('Range\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfamily = "sans-serif")
plt.xlabel('Score\n',horizontalalignment="center",fontstyle = "normal", fontsize = "large", fontfamily = "sans-serif")

plt.title('Optimal Number of K Neighbors \n',horizontalalignment="center", fontstyle = "normal",fontsize = "22", fontfami
#plt.legend(loc='top right', fontsize = "medium")

plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")
plt.show()
```

6.Random Forest is a popular ensemble machine learning algorithm that is widely used for both classification and regression tasks. It's considered one of the most powerful and versatile algorithms in the field of machine learning.



```
fig = plt.figure(figsize=(15, 7))
plt.plot(range(1,100),score_array, color = '#ec838a')
plt.ylabel('Range\n',horizontalalignment="center",
fontstyle = "normal", fontsize = "large",
fontfamily = "sans-serif")
plt.xlabel('Score\n',horizontalalignment="center",
fontstyle = "normal", fontsize = "large",
fontfamily = "sans-serif")
plt.title('Optimal Number of Trees for Random Forest Model \n',horizontalalignment="center", fontstyle = "normal", fontsi
#plt.legend(loc='top right', fontsize = "medium")
plt.xticks(rotation=0, horizontalalignment="center")
plt.yticks(rotation=0, horizontalalignment="right")
plt.show()
```