CHURN ANALYTICS

INTERNSHIP PROJECT

ANKIT KUMAR

CUSTOMER CHURN

- Customer churn, when customers discontinue their subscription or leave a service provider, is a significant concern for businesses as it directly impacts revenue and market competitiveness.

- Churn analytics and customer churn are critical concepts in business analytics and customer relationship management.
- Leveraging a comprehensive dataset from the Telecom industry, we employ advanced machine learning techniques to develop an accurate and robust churn prediction model.

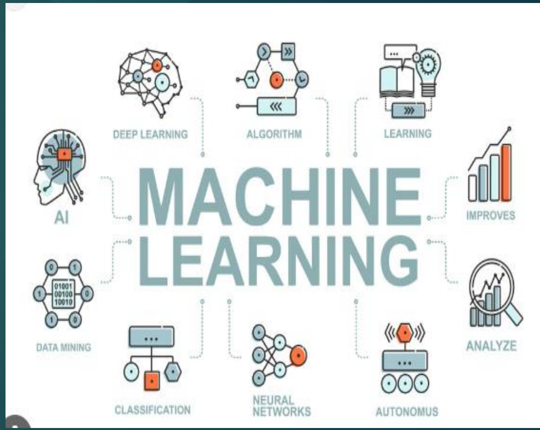
PROBLEM SOLUTION

- The proposed model utilizes various features related to customer behavior, service usage patterns, demographic information, and customer interaction history. By leveraging historical data, our model aims to identify key factors contributing to customer churn and provide insights for a proactive customer retention strategy.
- To construct the model, we adopt a multi-step approach involving data preprocessing, feature engineering, and model selection.
  We apply appropriate preprocessing techniques to handle missing values, outliers, and categorical variables. Feature engineering is employed to extract meaningful information from the dataset and enhance the predictive power of the model. Various machine learning algorithms, such as logistic regression, decision trees, random forests, and gradient boosting, are evaluated and compared to identify the most effective algorithm for churn prediction.

- Also, I have used Power BI as a BI Tool to showcase some insights to the stakeholders.

Tools & Techniques Used

- Jupyter Notebook
- Python- Machine Learning Algorithms, Numpy and Pandas Libraries, sklearn Libraries
- BI Tools- Power BI



MODEL BUILDING

1. Importing all the necessary libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
pd.set_option('display.max_columns',None)
```

2. Loading the dataset.

```
data=pd.read_csv('/content/Telecom Customer Churn Dataset.csv')
```

3. Analyzing the shape of the dataset.

- 7043 rows and 21 columns
- 21 columns- 'customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents', 'tenure', 'PhoneService', 'MultipleLines', 'InternetService', 'OnlineSecurity', 'OnlineBackup', 'DeviceProtection', 'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract', 'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges', 'TotalCharges', 'Churn'

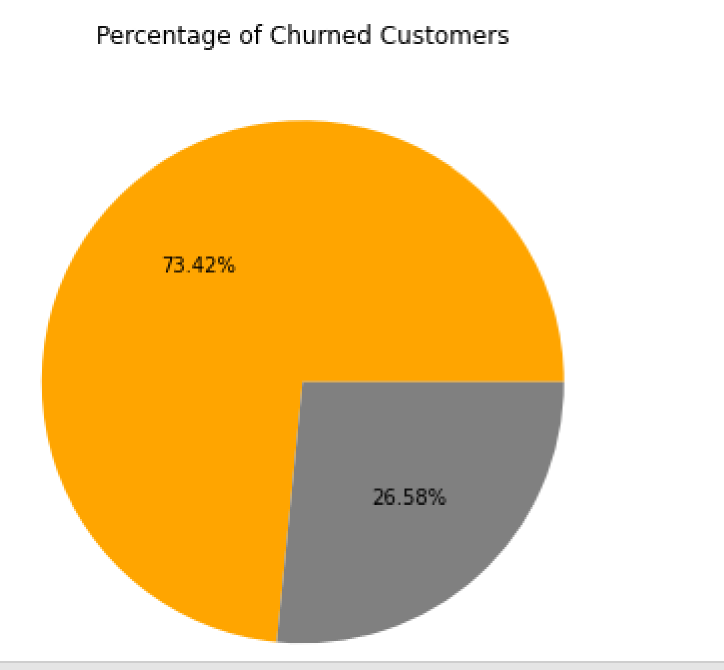
Contd...

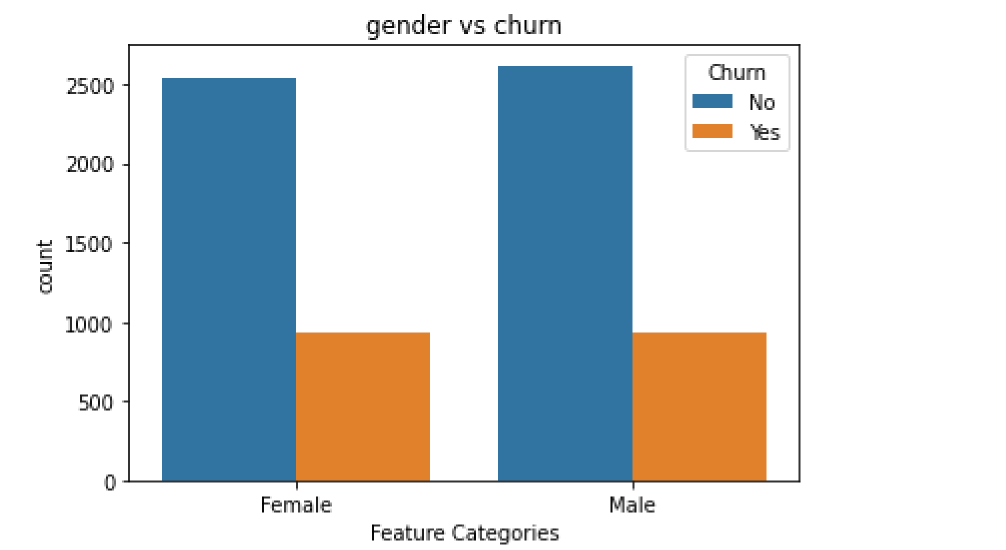
4. Dealing with Null Values

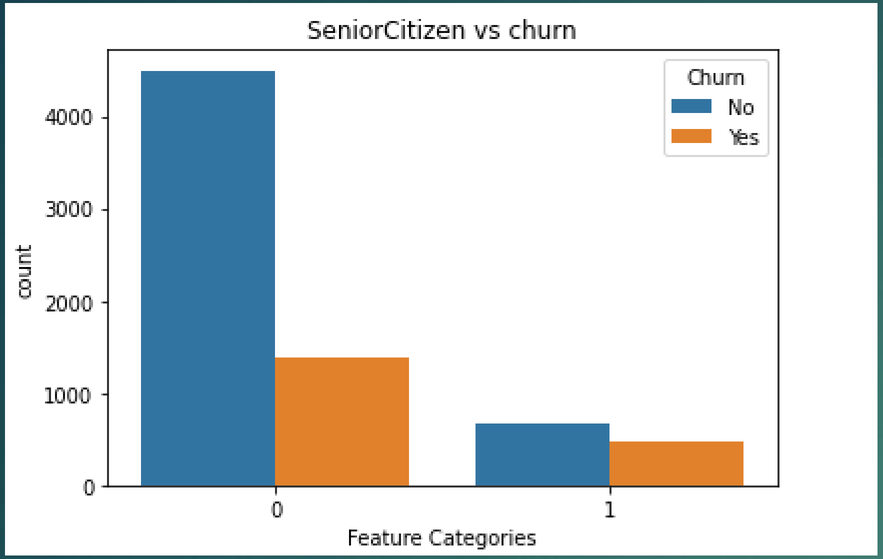
```
data.isnull().sum()
```

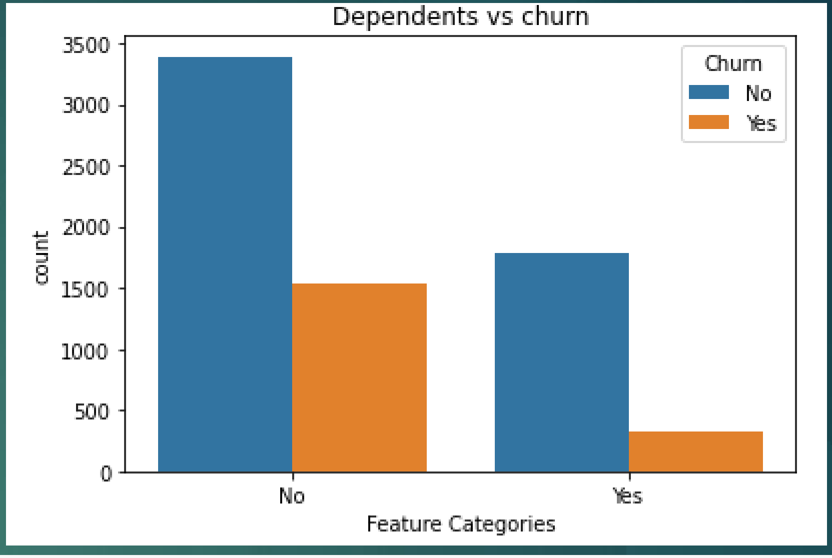
- There was very less null values. So they have been dropped by using dropna().

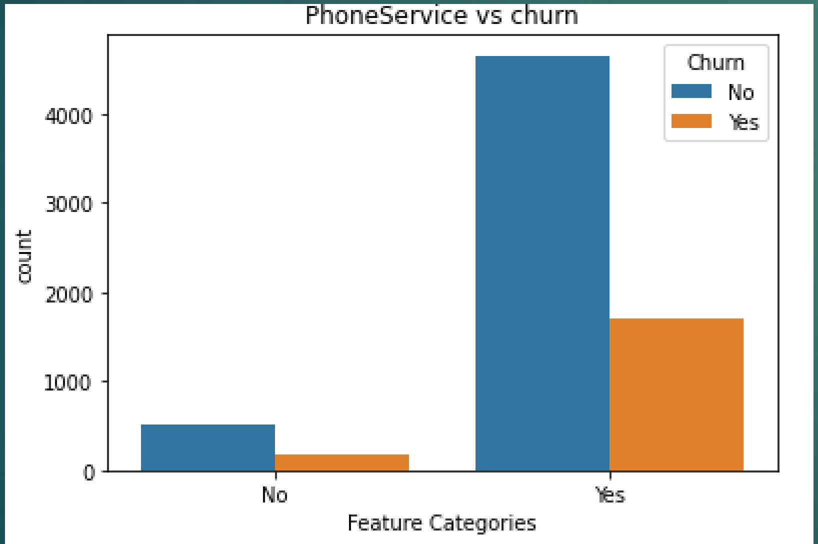
5. Data Visualization





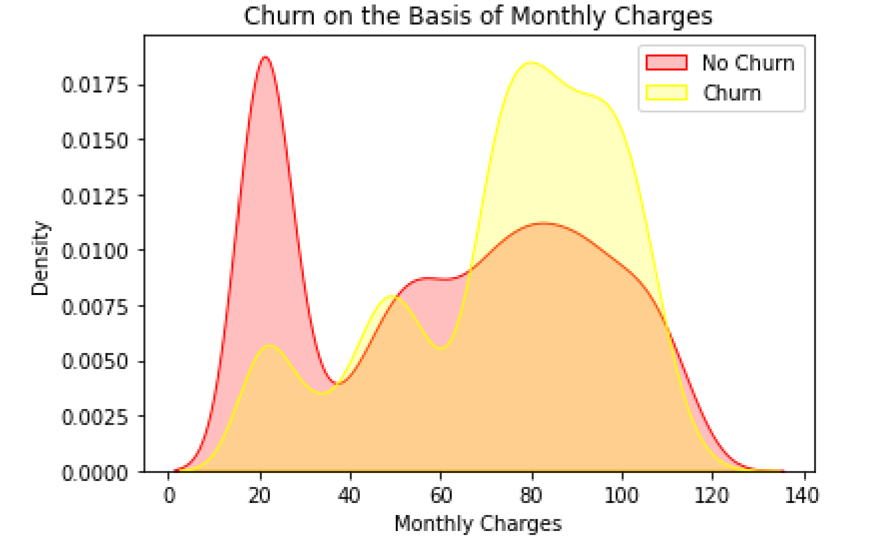






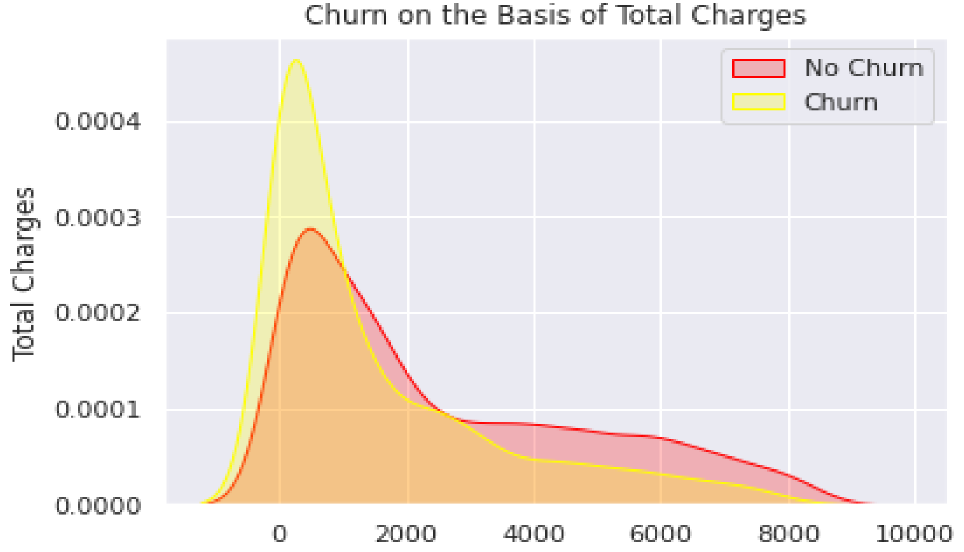

```
#Churn is high when monthly charges are high
```

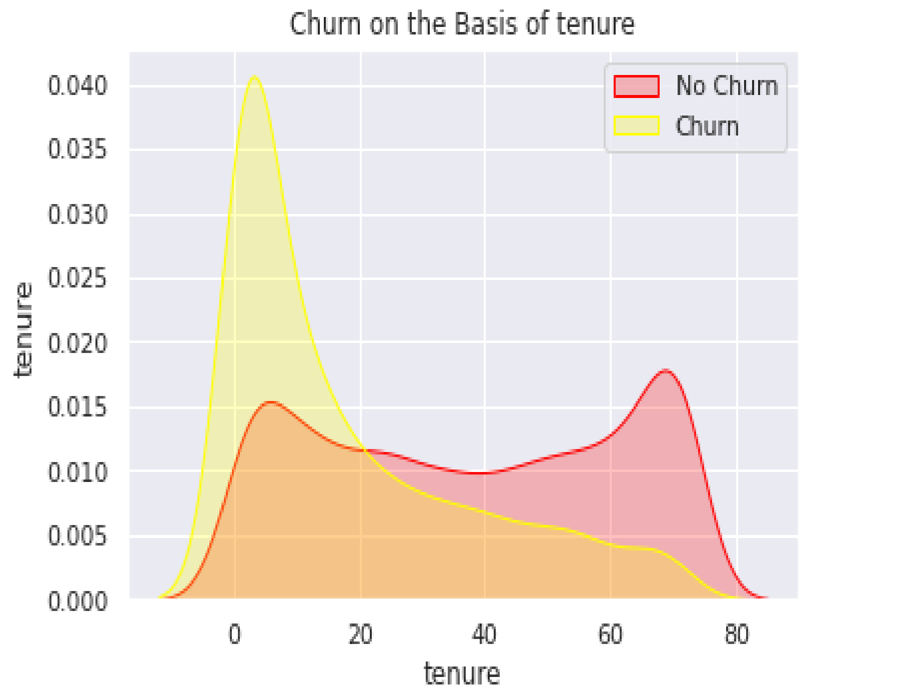
```
Text(0.5, 1.0, 'Churn on the Basis of Monthly Charges')
```



```
#here, churn is high when total charges are minimum
```

```
Text(0.5, 1.0, 'Churn on the Basis of Total Charges')
```





Contd…

6. MODEL BUILDING

```
#TRAIN TEST SPLIT
from sklearn.model_selection import train_test_split
```

```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.25,random_state=42)
```

```
print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(5274, 30)
(1758, 30)
(5274,)
(1758,)
```

```python
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

```python
sc.fit(X_train)
X_train_sc=sc.transform(X_train)
X_test_sc=sc.transform(X_test)
```

```python
X_train_sc
```

```
array([[-0.4377158 , -0.74817539, -0.52638   , ..., -0.53100285,
        -0.71708569, -0.54360352],
       [-0.4377158 , -0.05601627,  0.85826225, ...,  1.88322907,
        -0.71708569, -0.54360352],
       [-0.4377158 ,  0.59542761,  1.41543565, ..., -0.53100285,
        -0.71708569, -0.54360352],
       ...,
       [-0.4377158 , -0.9517516 ,  0.54651046, ..., -0.53100285,
         1.39453348, -0.54360352],
       [-0.4377158 ,  0.71757334, -1.48982568, ..., -0.53100285,
        -0.71708569, -0.54360352],
       [ 2.28458741, -0.50388393,  0.29777233, ..., -0.53100285,
         1.39453348, -0.54360352]])
```

- Before balancing the dataset, when I built different Machine Learning models. Logistic regression was showing the greatest accuracy among all of them.

```
[ ] from sklearn.linear_model import LogisticRegression
```

```
[ ] model1=LogisticRegression()
```

```
[ ] model1.fit(X_train_sc,y_train)

    LogisticRegression()
```

```
[ ] y_pred=model1.predict(X_test_sc)
    y_pred

    array([0, 0, 1, ..., 0, 0, 0])
```

```
from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))

              precision    recall  f1-score   support

           0       0.84      0.89      0.86      1300
           1       0.61      0.52      0.56       458

    accuracy                           0.79      1758
   macro avg       0.73      0.70      0.71      1758
weighted avg       0.78      0.79      0.78      1758
```

- After balancing the dataset and again building machine learning models. Random Forest was showing more accuracy.

WE KNOW THAT OUR DATASET IS IMBALANCE:

-Here, we will use the 'upsampling' method of balancing the dataset. -So, we will now work on balancing and then check the accuracy. -There are 2 famous methods used for 'upsampling'

1.SMOTETomek

2.RandomOverSampler

These both methods are present inside the 'imblearn' library.

Here, we have used the second method---> RandomOverSampler

```
from sklearn.ensemble import RandomForestClassifier
```

```
model6=RandomForestClassifier()
```

```
model6.fit(X1_train_sc,y1_train)
```

```
RandomForestClassifier()
```

```
y_pred6=model6.predict(X1_test_sc)
y_pred6
```

```
array([1, 0, 0, ..., 1, 1, 1])
```

```
print(classification_report(y1_test,y_pred6))
```

```
              precision    recall  f1-score   support

           0       0.95      0.83      0.89      1304
           1       0.85      0.95      0.90      1278

    accuracy                           0.89      2582
   macro avg       0.90      0.89      0.89      2582
weighted avg       0.90      0.89      0.89      2582
```

BUSINESS INTELLIGENCE

- After building machine learning models to predict the churn rate of customers.
- Used Power BI tool to make a dashboard to derive some insights from the dataset.
- Few answers which we got.

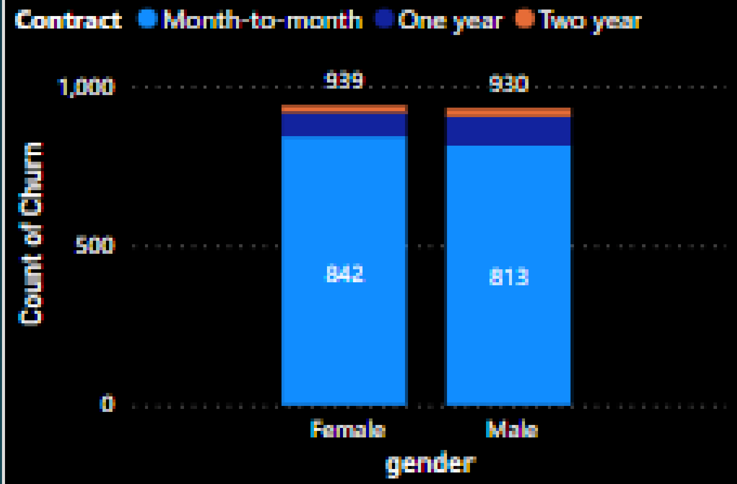
1. When we calculate the churn percentage we got, 26.92% of females have been churned and 26.16% of male customers have been churned.
2. Monthly Customers are more likely to churn in both Male (87%) and Female(90%) in which female monthly customers are more likely to churn as compared to male monthly customers.
3. Male 2 yearly customers are more likely to churn as compared to Female 2 yearly customers
4. the maximum churn is of customers who have 0- 19 months of tenure

CUSTOMER CHURN ANALYTICS DASHBOARD

TOTAL 7043	CHURNED 1869	MALE 3555	FEMALE 3488	CHURNED MALE 930	CHURNED FEMALE 939

CONTRACT

Churn Count Male & Female- Basis of Contract



TOTAL CUSTOMERS

Total Charges

16.06M

Monthly Charges

456.12K

Average Charges

2.28K

CHURNED CUSTOMERS

Total Charges

2.86M

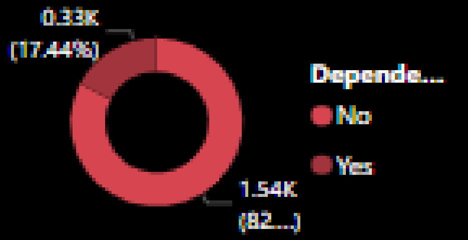
Monthly Charges

139.13K

Average Charges

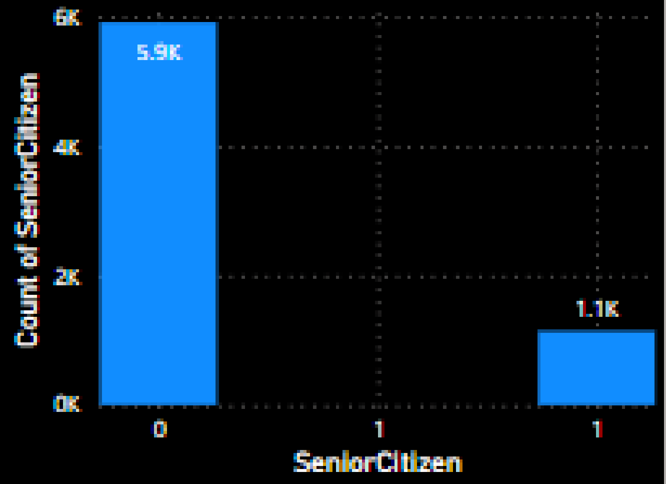
406.49

DEPENDENTS



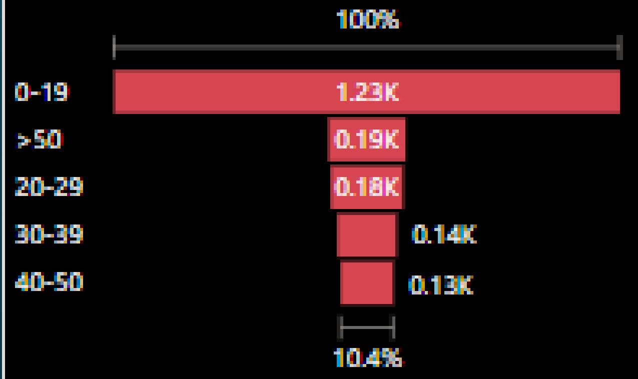
SENIOR CITIZENS

Count of SeniorCitizen by SeniorCitizen

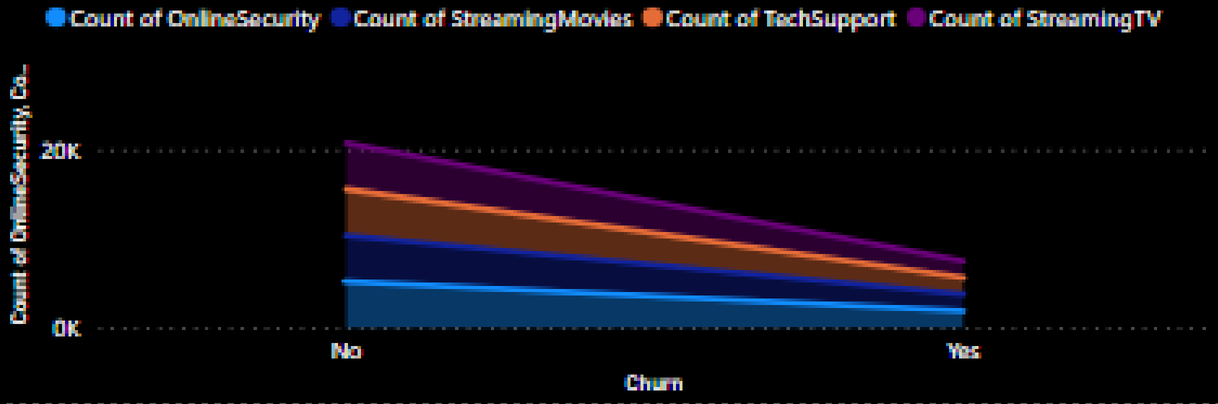


TENURE

Churn Counts on the Basis of Tenure



Effect of Various Facilities Provided By Company to Customers on Churn



Thank You!!!