TELCO

Customer data analysis and churn prediction

PRESENTED BY

GROUP 13

Yash Tyagi
Abhinav Sharma

Table of Contents

TELCO

What is Our Information Product?

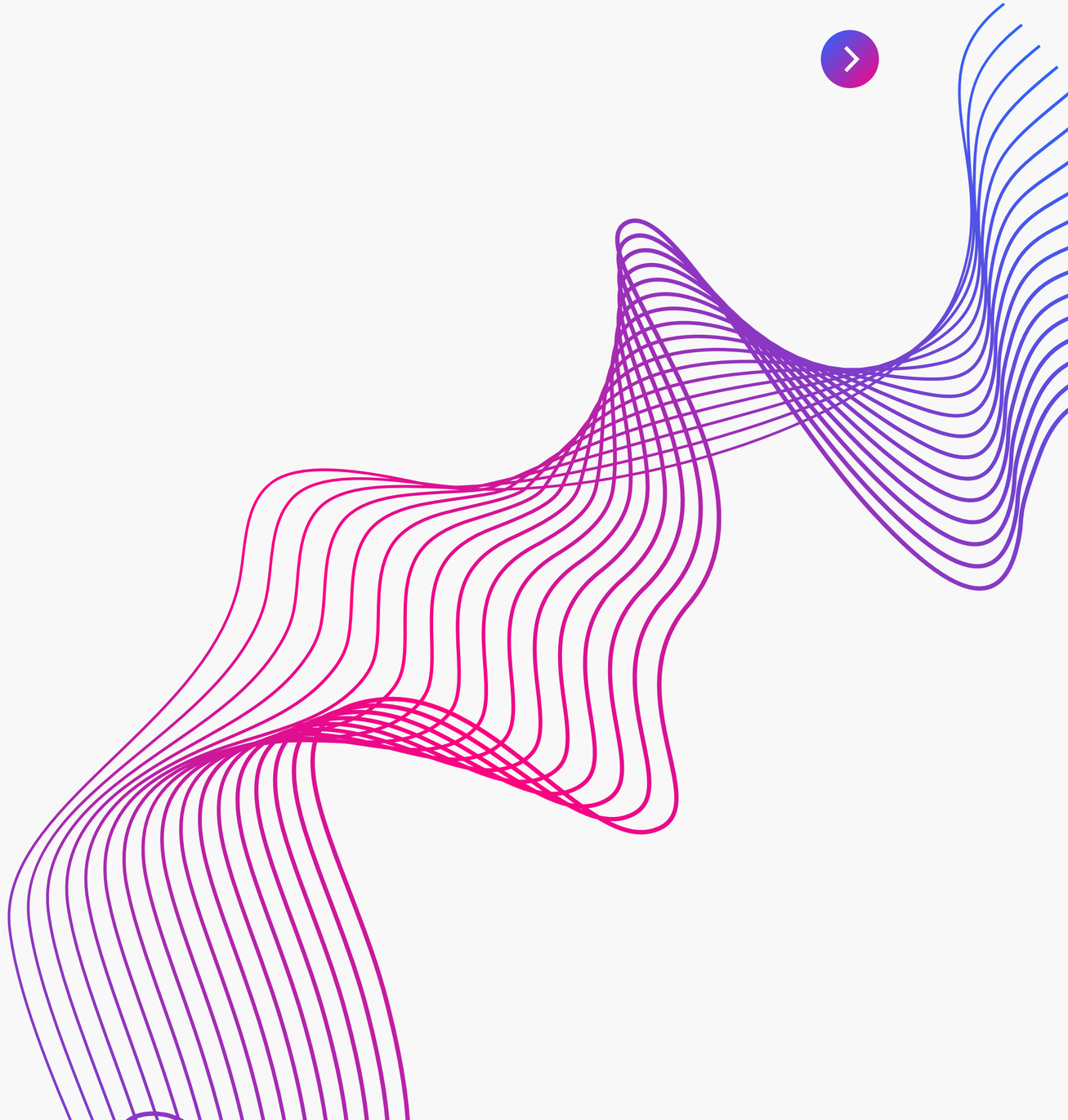
- Our IP is a Tool to predict Customer churn for the company Telco Systems
- Analysis of Predicted Customer Churn data
- Customer churn is when customers stop doing business with a company. Predicting churn helps prevent customer loss and increase revenue

TELCO

Case Study

- Today, we will examine a case involving the well-known telecommunications company TELCO.

- We have gathered our datasets for TELCO from open sources such as KEGG and will use them to study and comprehend the use of these datasets and overall effect of Churn



1) Threat of new entrants: The telecommunications industry is highly regulated and requires significant investments in infrastructure, which makes it difficult for new entrants to compete with established players like Telco. However, with the advancement in technology, new players can enter the market with low-cost solutions, making the industry even more competitive.

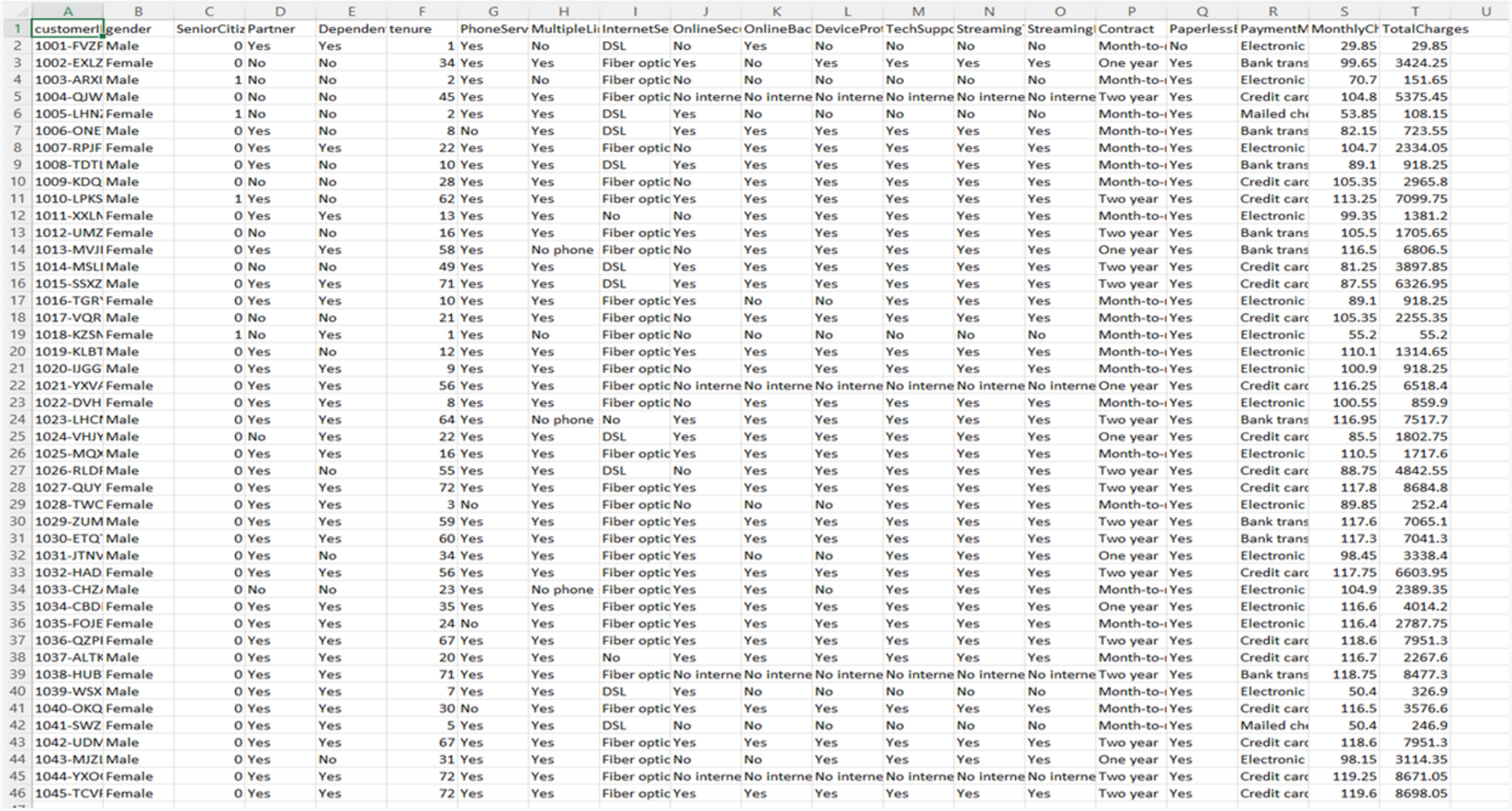
- **Competitors: AT&T, Verizon,T-Mobile, Sprint.**

2) Bargaining power of buyers: Buyers in the telecommunications industry have significant bargaining power due to the high competition and the low switching costs.

3) Intensity of competitive rivalry: The telecommunications industry is highly competitive, with several established players and new entrants vying for market share. To stay ahead in the market we need to use data analysis to retain the customers.

TELCO

DATASET



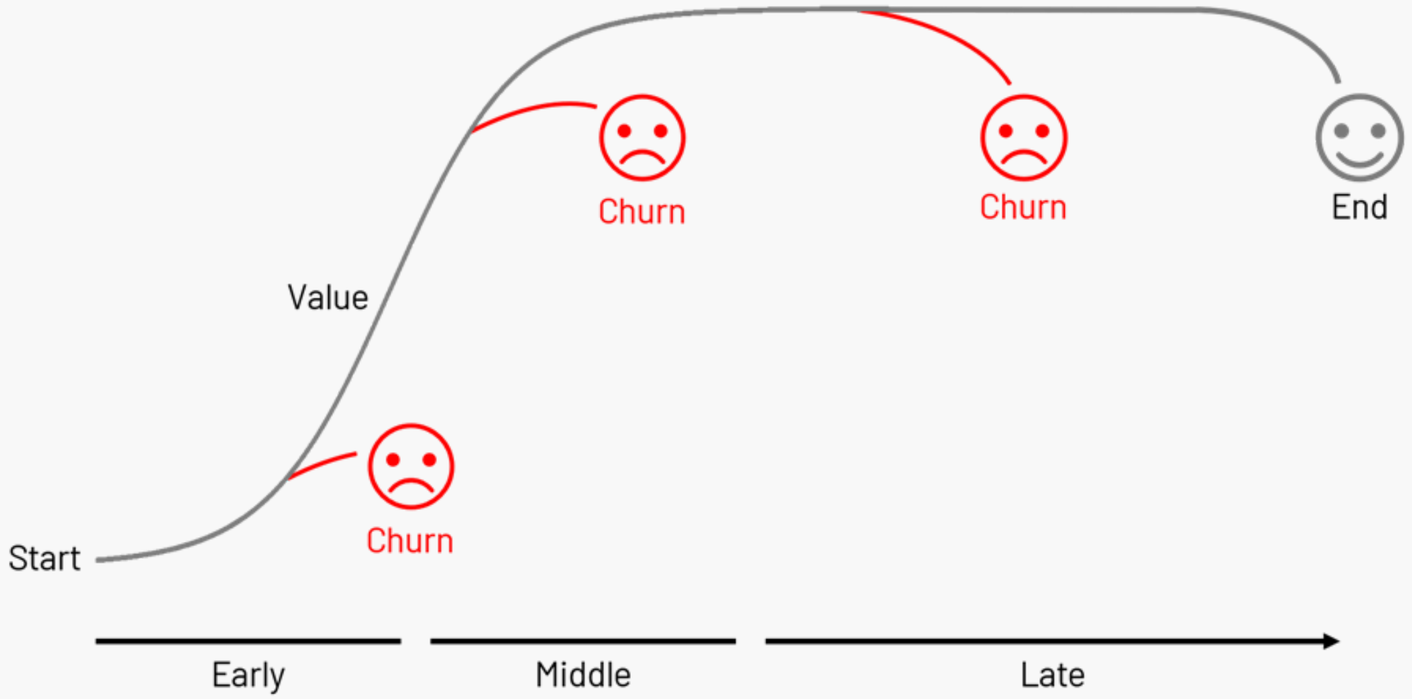
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	customerI	gender	SeniorCitiz	Partner	Dependen	tenure	PhoneServ	MultipleLi	InternetSe	OnlineSec	OnlineBac	DevicePro	TechSuppo	Streaming	Streaming	Contract	Paperless	PaymentM	MonthlyCh	TotalCharges
2	1001-FVZF	Male	0	Yes	Yes	1	Yes	No	DSL	No	Yes	No	No	No	No	Month-to-	No	Electronic	29.85	29.85
3	1002-EXLZ	Female	0	No	No	34	Yes	Yes	Fiber optic	Yes	No	Yes	Yes	Yes	Yes	One year	Yes	Bank trans	99.65	3424.25
4	1003-ARXI	Male	1	No	No	2	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	Electronic	70.7	151.65
5	1004-QJW	Male	0	No	No	45	Yes	Yes	Fiber optic	No interne	No interne	No interne	No interne	No interne	No interne	Two year	Yes	Credit carc	104.8	5375.45
6	1005-LHN	Female	1	No	No	2	Yes	Yes	DSL	Yes	No	No	No	No	No	Month-to-	Yes	Mailed che	53.85	108.15
7	1006-ONE	Male	0	Yes	No	8	No	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Bank trans	82.15	723.55
8	1007-RPJF	Female	0	Yes	Yes	22	Yes	Yes	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	104.7	2334.05
9	1008-TDTI	Male	0	Yes	No	10	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Bank trans	89.1	918.25
10	1009-KDQ	Male	0	No	No	28	Yes	Yes	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Credit carc	105.35	2965.8
11	1010-LPKS	Male	1	Yes	No	62	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	113.25	7099.75
12	1011-XXLN	Female	0	Yes	Yes	13	Yes	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	99.35	1381.2
13	1012-UMZ	Female	0	No	No	16	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Bank trans	105.5	1705.65
14	1013-MVJI	Female	0	Yes	Yes	58	Yes	No phone	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	One year	Yes	Bank trans	116.5	6806.5
15	1014-MSLI	Male	0	No	No	49	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	81.25	3897.85
16	1015-SSXZ	Male	0	Yes	Yes	71	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	87.55	6326.95
17	1016-TGR	Female	0	Yes	Yes	10	Yes	Yes	Fiber optic	Yes	No	No	Yes	Yes	Yes	Month-to-	Yes	Electronic	89.1	918.25
18	1017-VQR	Male	0	No	No	21	Yes	Yes	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Credit carc	105.35	2255.35
19	1018-KZSN	Female	1	No	Yes	1	Yes	No	Fiber optic	No	No	No	No	No	No	Month-to-	Yes	Electronic	55.2	55.2
20	1019-KLBT	Male	0	Yes	No	12	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	110.1	1314.65
21	1020-IJGG	Male	0	Yes	Yes	9	Yes	Yes	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	100.9	918.25
22	1021-YXV	Female	0	Yes	Yes	56	Yes	Yes	Fiber optic	No interne	No interne	No interne	No interne	No interne	No interne	One year	Yes	Credit carc	116.25	6518.4
23	1022-DVH	Female	0	Yes	Yes	8	Yes	Yes	Fiber optic	No	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	100.55	859.9
24	1023-LHC	Male	0	Yes	Yes	64	Yes	No phone	No	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Bank trans	116.95	7517.7
25	1024-VHJY	Male	0	No	Yes	22	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	One year	Yes	Credit carc	85.5	1802.75
26	1025-MQX	Male	0	Yes	Yes	16	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	110.5	1717.6
27	1026-RLDI	Male	0	Yes	No	55	Yes	Yes	DSL	No	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	88.75	4842.55
28	1027-QUY	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	117.8	8684.8
29	1028-TWC	Female	0	Yes	Yes	3	No	Yes	Fiber optic	No	No	No	Yes	Yes	Yes	Month-to-	Yes	Electronic	89.85	252.4
30	1029-ZUM	Male	0	Yes	Yes	59	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Bank trans	117.6	7065.1
31	1030-ETQ	Male	0	Yes	Yes	60	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Bank trans	117.3	7041.3
32	1031-JTN	Male	0	Yes	No	34	Yes	Yes	Fiber optic	Yes	No	No	Yes	Yes	Yes	One year	Yes	Electronic	98.45	3338.4
33	1032-HAD	Female	0	Yes	Yes	56	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	117.75	6603.95
34	1033-CHZ	Male	0	No	No	23	Yes	No phone	Fiber optic	Yes	Yes	No	Yes	Yes	Yes	Month-to-	Yes	Electronic	104.9	2389.35
35	1034-CBD	Female	0	Yes	Yes	35	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	One year	Yes	Electronic	116.6	4014.2
36	1035-FOJE	Female	0	Yes	Yes	24	No	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Electronic	116.4	2787.75
37	1036-QZPI	Female	0	Yes	Yes	67	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	118.6	7951.3
38	1037-ALTK	Male	0	Yes	Yes	20	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Credit carc	116.7	2267.6
39	1038-HUB	Female	0	Yes	Yes	71	Yes	Yes	Fiber optic	No interne	No interne	No interne	No interne	No interne	No interne	Two year	Yes	Bank trans	118.75	8477.3
40	1039-WSX	Male	0	Yes	Yes	7	Yes	Yes	DSL	Yes	No	No	No	No	No	Month-to-	Yes	Electronic	50.4	326.9
41	1040-OKQ	Female	0	Yes	Yes	30	No	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Month-to-	Yes	Credit carc	116.5	3576.6
42	1041-SWZ	Female	0	Yes	Yes	5	Yes	Yes	DSL	No	No	No	No	No	No	Month-to-	Yes	Mailed che	50.4	246.9
43	1042-UDM	Male	0	Yes	Yes	67	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	118.6	7951.3
44	1043-MJZI	Male	0	Yes	No	31	Yes	Yes	Fiber optic	No	No	Yes	Yes	Yes	Yes	One year	Yes	Electronic	98.15	3114.35
45	1044-YXO	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	No interne	No interne	No interne	No interne	No interne	No interne	Two year	Yes	Credit carc	119.25	8671.05
46	1045-TCV	Female	0	Yes	Yes	72	Yes	Yes	Fiber optic	Yes	Yes	Yes	Yes	Yes	Yes	Two year	Yes	Credit carc	119.6	8698.05

1. Gender
2. Age
3. Senior Citizen
4. Phone Service
5. Multiple Lines
6. Payment Mode
7. Contract Time Period

TELCO

Business Problem

Customer churn Prevention is the ultimate goal that enables customer retention and growth in revenue



TELCO

APPROACH TO SOLVE USING ML MODELS

- PATTERN IDENTIFICATION: THERE IS A PATTERN IN CUSTOMER BEHAVIOR THAT HAS TO BE IDENTIFIED

- PREDICTIONS: WE CAN PREDICT THE CHURN RATE FOR A SET OF CUSTOMERS WITH GIVEN CONDITIONS AND IDENTIFY THE LAPSE

TELCO

Implementation

- Churn can be prevented by giving undeniable offers based on the predictions made for each group of customers

- Targeted marketing can be done based on observed customer behavior

Data Collection and Cleaning

- Steps we have followed to wrangle the data

- Data Cleaning
- Data Transformation
- Imputation
- Data scaling and normalization

- After performing the above steps we have taken sample data our for analysis and predictions

Data Cleaning :Removing irrelevant columns for data analysis from the data set

```
In [79]: #making the copy of original data set before dropping the customer id column
         df1=df.copy()

In [80]: # Drop the customerID column as it is not relevant for analysis
         df = df.drop('customerID', axis=1)

In [81]: print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 20 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   gender            7043 non-null   object
 1   SeniorCitizen     7043 non-null   int64
 2   Partner           7043 non-null   object
 3   Dependents        7043 non-null   object
 4   tenure            7043 non-null   int64
 5   PhoneService      7043 non-null   object
 6   MultipleLines     7043 non-null   object
 7   InternetService   7043 non-null   object
 8   OnlineSecurity    7043 non-null   object
 9   OnlineBackup      7043 non-null   object
 10  DeviceProtection  7043 non-null   object
 11  TechSupport       7043 non-null   object
 12  StreamingTV       7043 non-null   object
 13  StreamingMovies   7043 non-null   object
 14  Contract          7043 non-null   object
 15  PaperlessBilling  7043 non-null   object
```

Data transformation using lambda and one-hot encoding:

In [82]:
```
#converting churn values into 0 and 1
df['Churn'] = df['Churn'].apply(lambda x: 1 if x=='Yes' else 0)
df.head()
```

Out[82]:

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	Tech
0	Female	0	Yes	No	1	No	No phone service	DSL	No	Yes	No	
1	Male	0	No	No	34	Yes	No	DSL	Yes	No	Yes	
2	Male	0	No	No	2	Yes	No	DSL	Yes	Yes	No	
3	Male	0	No	No	45	No	No phone service	DSL	Yes	No	Yes	
4	Female	0	No	No	2	Yes	No	Fiber optic	No	No	No	

In [83]:
```
print(df['Churn'].value_counts())
```

```
0    5174
1    1869
Name: Churn, dtype: int64
```

In [84]:
```
# Convert categorical variables into numerical formats using one-hot encoding
df = pd.get_dummies(df, columns=['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'InternetService',
df.head()
```

Out[84]:

	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	gender_Female	gender_Male	Partner_No	Partner_Yes	Dependents_No	...	StreamingMovi
0	0	1	29.85	29.85	0	1	0	0	1	1	...	
1	0	34	56.95	1889.5	0	0	1	1	0	1	...	
2	0	2	53.85	108.15	1	0	1	1	0	1	...	
3	0	45	42.30	1840.75	0	0	1	1	0	1	...	
4	0	2	70.70	151.65	1	1	0	1	0	1	...	

5 rows × 46 columns

In [85]:
```
#Converting total charge attribute into float data type
df['TotalCharges'] = pd.to_numeric(df['TotalCharges'], errors='coerce')
df['TotalCharges'] = df['TotalCharges'].astype(float)
```

Imputation: By imputing all the missing values with the mean value for each column.

In [86]:
```
#checking the total number of missing values in the data set
print(df.isna().sum())
```

```
SeniorCitizen                          0
tenure                                 0
MonthlyCharges                         0
TotalCharges                          11
Churn                                  0
gender_Female                          0
gender_Male                            0
Partner_No                             0
Partner_Yes                            0
Dependents_No                          0
Dependents_Yes                         0
PhoneService_No                        0
PhoneService_Yes                       0
MultipleLines_No                       0
MultipleLines_No phone service         0
MultipleLines_Yes                      0
InternetService_DSL                    0
InternetService_Fiber optic            0
InternetService_No                     0
OnlineSecurity_No                      0
```

In [87]:
```
#replacing all the missing values of each column with mean of that column
df = df.fillna(df.mean())
print(df.isna().sum())
```

```
SeniorCitizen                          0
tenure                                 0
MonthlyCharges                         0
TotalCharges                           0
Churn                                  0
gender_Female                          0
gender_Male                            0
Partner_No                             0
Partner_Yes                            0
Dependents_No                          0
Dependents_Yes                         0
PhoneService_No                        0
PhoneService_Yes                       0
MultipleLines_No                       0
MultipleLines_No phone service         0
MultipleLines_Yes                      0
InternetService_DSL                    0
InternetService_Fiber optic            0
InternetService_No                     0
OnlineSecurity_No                      0
OnlineSecurity_No internet service     0
OnlineSecurity_Yes                     0
OnlineBackup_No                        0
OnlineBackup_No internet service       0
OnlineBackup_Yes                       0
DeviceProtection_No                    0
```

Scaling And normalization:
Normalization using MinMaxScaler

In [102]:
```
# This method fits the scaler to the data and scales the values to be between 0 and 1
#Scale and normalize the features to ensure that they have similar ranges and distributions.
""" The scaled values are then used to replace the original values in the dataframe,
transforming the original numerical features into scaled numerical features"""
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
num_features = ['tenure', 'MonthlyCharges', 'TotalCharges']
df[num_features] = scaler.fit_transform(df[num_features])
df.shape[1]
```

Out[102]: 46

Model Selection : Choose a machine learning algorithm that is suitable for the problem using accuracy score

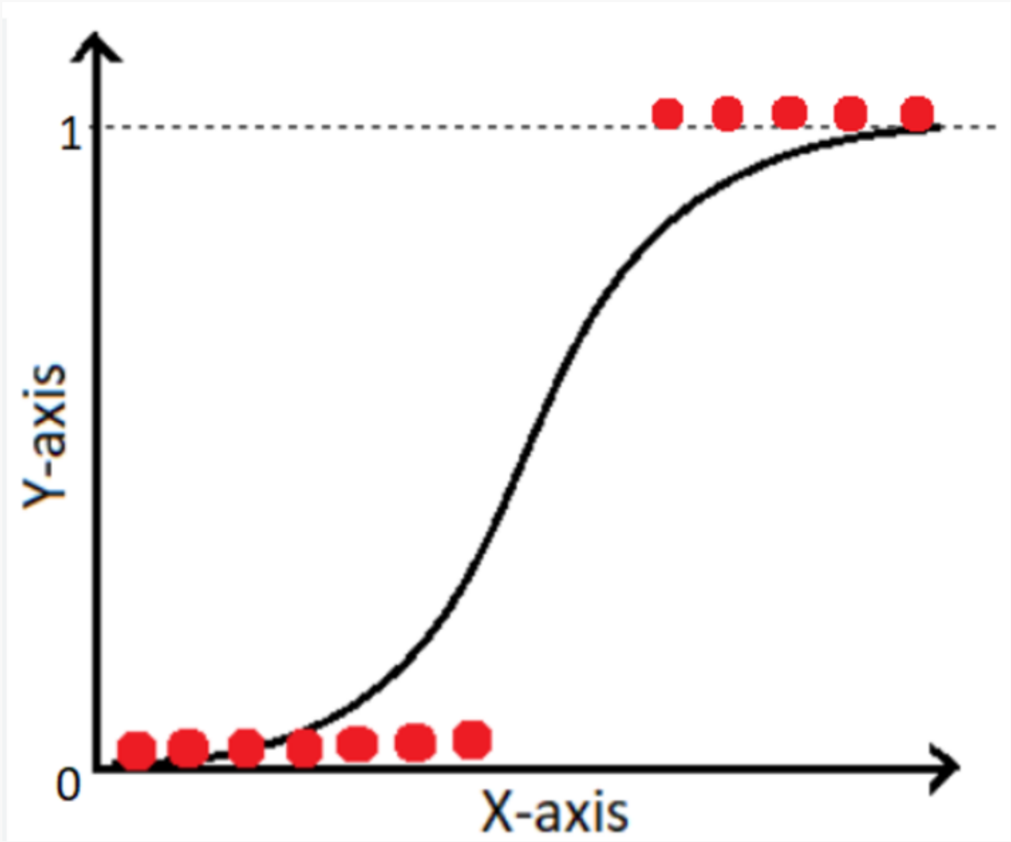
```
#Select an appropriate machine learning algorithm such as logistic regression, decision trees, random forests and XGB.
"""To select which is the best model for predicting churn, we will evaluate the model's performance usingmetrics
such as the accuracy of each model after training it using 70% of the data for training and 30% for testing and
finding an accuracy score at the end for each of the models."""

First, we need to split the dataset into training and testing sets. We will use 70% of the data for training and 30% for
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
import xgboost as xgb

logreg = LogisticRegression()
tree = DecisionTreeClassifier(random_state=42)
rf = RandomForestClassifier(random_state=42)
xgb = xgb.XGBClassifier(random_state=42)
```

```
logreg.fit(X_train, y_train)
tree.fit(X_train, y_train)
rf.fit(X_train, y_train)
xgb.fit(X_train, y_train)
```

```
#Finding accuracy score using metrics below for each of the mo
from sklearn.metrics import accuracy_score

models = {'Logistic Regression': logreg,
          'Decision Tree': tree,
          'Random Forest': rf,
          'XGBoost': xgb}

for name, model in models.items():
    y_pred = model.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    print(f'{name} accuracy: {acc}')
```

```
Logistic Regression accuracy: 0.8204400283889283
Decision Tree accuracy: 0.7154009936124911
Random Forest accuracy: 0.7927608232789212
XGBoost accuracy: 0.7927608232789212
```

Logistic Regression



TELCO

Logistic Regression

- In this case, we're using logistic regression to model the relationship between customer characteristics and churn behavior
- Our dataset includes features such as customer demographics, usage patterns, and account information
- We can use logistic regression to identify the features that are most strongly associated with churn, and to understand how these features interact with each other

Logistic Regression

- One benefit of logistic regression is that it provides interpretable coefficients that can help us understand the direction and magnitude of the relationships between features and outcomes

- However, one challenge with logistic regression is that it assumes a linear relationship between the features and the log-odds of the outcome. If this assumption is not met, we may need to consider more complex modeling techniques

- Once we've **built** a logistic regression model, we can use it to make **predictions** about whether a given **customer is likely to churn.** This can help us take proactive steps to prevent churn and improve customer retention

Model Development: Using logistic regression and tuning Hyperparmeters

```
#Based on the accuracy score, we can conclude that logistic regression is the best models f
#In above analysis we found out that logistic regression is the best model for the customer
# the two steps given below using logistic regression:
# Step1:Now  we will use the above trained logistic model on the training dataset and evalu
# Step2: Tune the hyperparameters of the algorithm to optimize its performance.
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Make predictions on the test set
y_pred = logreg.predict(X_test)

# Evaluate the performance of the model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
```

```
Accuracy: 0.8204400283889283
Precision: 0.6863354037267081
Recall: 0.5924932975871313
F1 Score: 0.6359712230215828
```

```
#We can use GridSearchCV to tune the hyperparameters of the logistic regression algorithm:
from sklearn.model_selection import GridSearchCV

# Define the hyperparameters to tune
hyperparameters = {"penalty": ["l1", "l2"],
                   "C": [0.01, 0.1, 1, 10, 100]}



# Create a GridSearchCV object
grid_search = GridSearchCV(logreg, hyperparameters, cv=5, scoring='accuracy')

# Fit the GridSearchCV object to the data
grid_search.fit(X_train, y_train)

# Print the best hyperparameters
print("Best Hyperparameters:", grid_search.best_params_)

# Make predictions on the test set using the best model
best_model = grid_search.best_estimator_
y_pred = best_model.predict(X_test)

# Evaluate the performance of the best model
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
filename = 'logistic_regression_model.joblib'
pickle.dump(lr_model, open(filename, 'wb'))
```

Predicting Churn: Using the best model we have created, we will predict churn for the new data set

In [118]:
```
#NOw we test the best model we have created on the new data set to predict the customer churn
New_data=pd.read_csv("C:/Users/AbhiStyles/Downloads/churn_Sample.csv")
New_data_copy=New_data.copy()

#performing data preprocessing steps
New_data = New_data.drop('customerID', axis=1)
New_data['TotalCharges'] = pd.to_numeric(New_data['TotalCharges'], errors='coerce')
New_data['TotalCharges'] = New_data['TotalCharges'].astype(float)
New_data = pd.get_dummies(New_data, columns=['gender', 'Partner', 'Dependents', 'PhoneService', 'MultipleLines', 'Intern



New_data = New_data.fillna(New_data.mean())
num_features = ['tenure', 'MonthlyCharges', 'TotalCharges']
New_data[num_features] = scaler.fit_transform(New_data[num_features])
```

In [119]:
```
#predicting the churn on the new data set using the best model we have created
churn_pred=best_model.predict(New_data)
churn_pred
```

Out[119]:
```
array([1, 0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 1,
       0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0,
       0], dtype=int64)
```

K-means Clustering

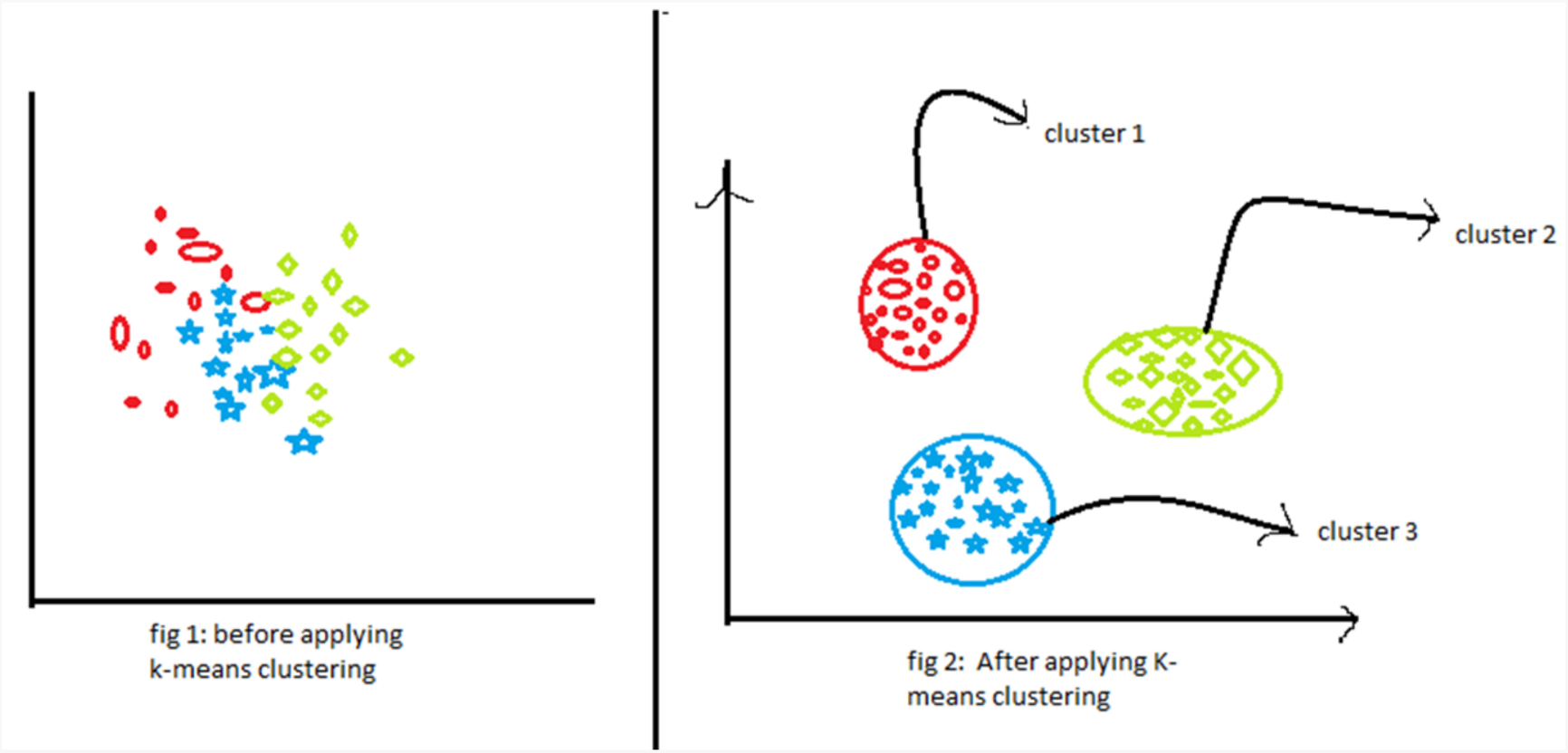


fig 1: before applying k-means clustering

fig 2: After applying K-means clustering

TELCO

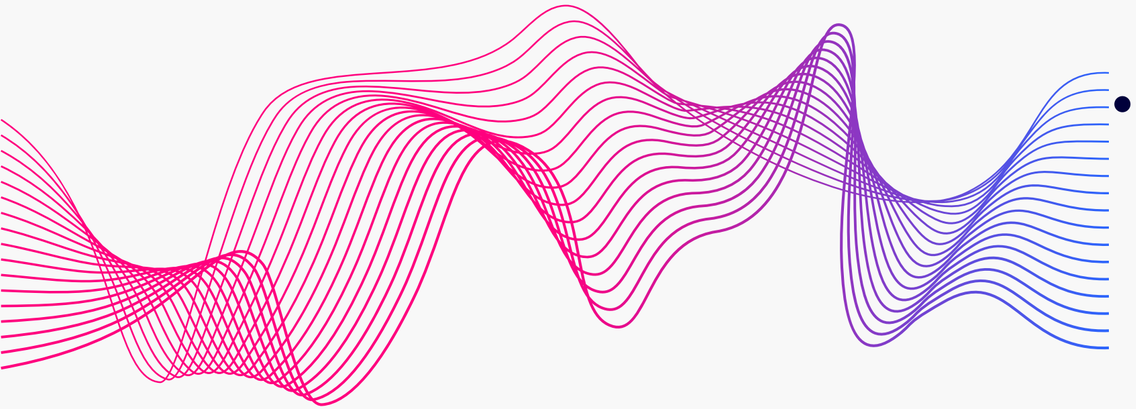
K-means Clustering

- K-means clustering is a popular unsupervised learning technique used to group similar data points into clusters.
- In this case, we're using k-means clustering to group customers based on their churn behavior
- Our dataset includes features such as customer demographics, usage patterns, and account information

TELCO

K-means Clustering

- We can use k-means clustering to identify groups of customers who are more likely to churn, and to understand the characteristics that define each group
- One challenge with k-means clustering is determining the optimal number of clusters to use. We can use techniques such as the elbow method or silhouette score to help with this decision
- Once we've identified the clusters, we can use the insights gained to take action to reduce churn. For example, we might create targeted marketing campaigns or offer special promotions to customers in certain clusters



Applying K-Means Clustering on the churn data set

In [152]:
```
#Run K-Means clustering to identify clusters of customers
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

New_data_clusters=New_data.copy()
New_data_clusters['Churn']=New_data_copy['Churn']
kmeans = KMeans(n_clusters=3, random_state=0)
kmeans.fit(New_data_clusters)

#Adding the cluster labels to the original dataset
New_data_copy['Cluster'] = kmeans.labels_
```

In [153]:
```
New_data_copy.head()
```

Out[153]:

ervice	OnlineSecurity	...	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges	Churn	cluster	Cluster
DSL	No	...	No	No	Month-to-month	No	Electronic check	29.85	29.85	True	1	1
r optic	Yes	...	Yes	Yes	One year	Yes	Bank transfer (automatic)	99.65	3424.25	False	0	0
r optic	No	...	No	No	Month-to-month	Yes	Electronic check	70.70	151.65	True	1	1
r optic	No internet service	...	No internet service	No internet service	Two year	Yes	Credit card (automatic)	104.80	5375.45	False	2	2
DSL	Yes	...	No	No	Month-to-month	Yes	Mailed check	53.85	108.15	True	1	1

Output

```
#Visualizing the clustering using a scatter plot for tenure and total charges
plt.scatter(New_data_copy['TotalCharges'], New_data_copy['tenure'], c=New_data_copy['Cluster'])
plt.xlabel('TotalCharges')
plt.ylabel('tenure')
plt.show()
```

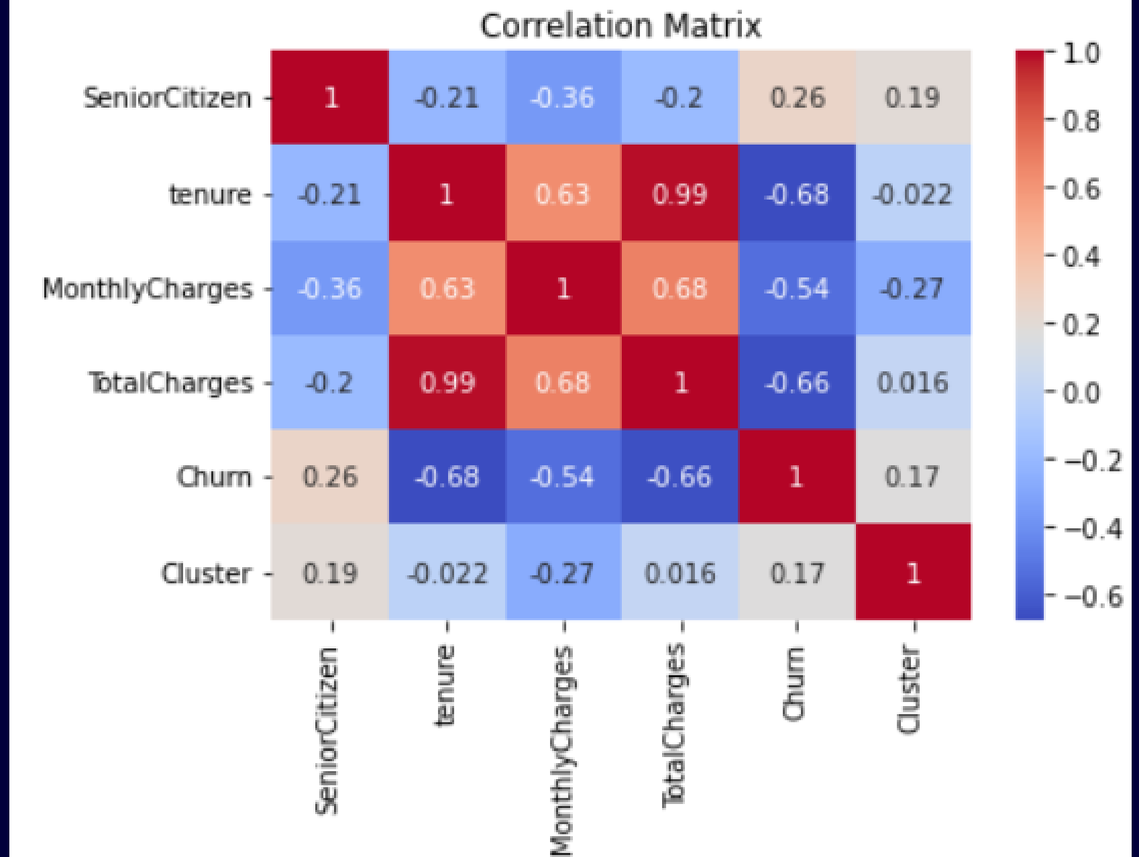
```
#identifying the characteristics of each cluster by calculating the mean values of the variables for each cluster.
New_data_copy.groupby('Cluster').mean()
```

]:

Cluster	SeniorCitizen	tenure	MonthlyCharges	TotalCharges	Churn	cluster
0	0.030303	37.212121	105.99697	4069.593939	0.242424	0.0
1	0.375000	3.875000	61.16875	261.162500	1.000000	1.0
2	0.000000	61.000000	114.76250	7260.550000	0.000000	2.0

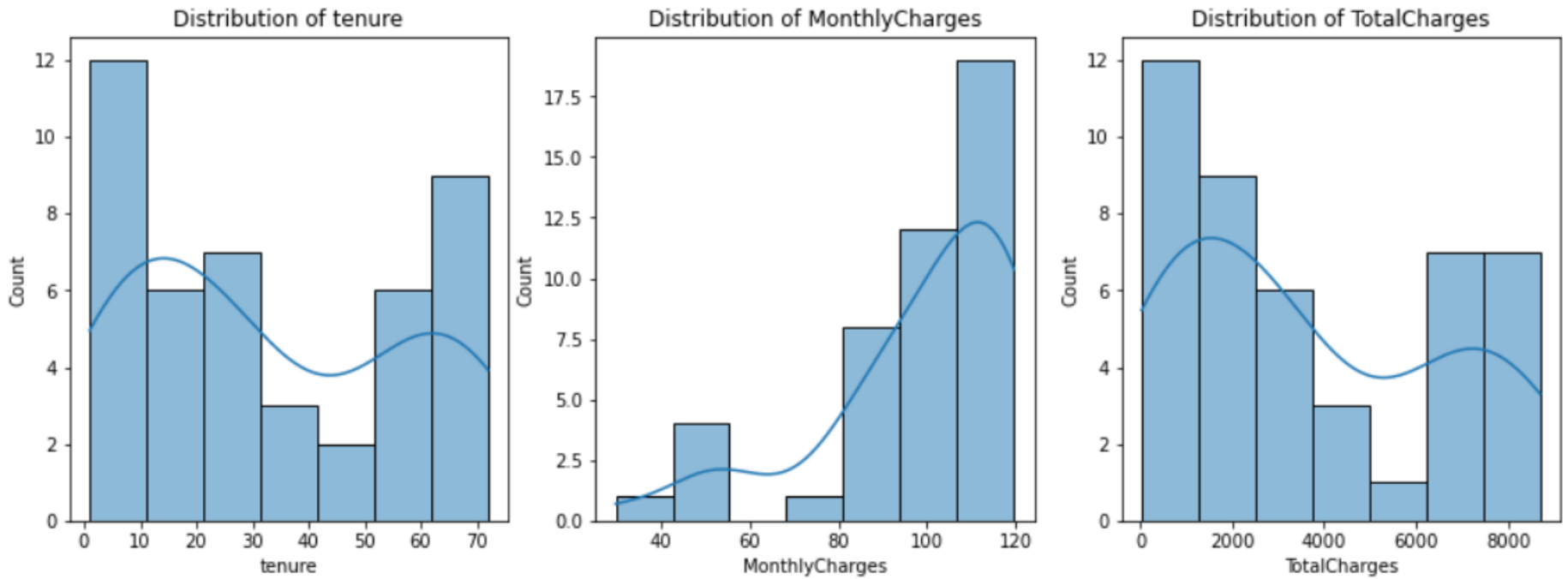
Heat Map:

```
corr = New_data_copy.corr()
sns.heatmap(corr, cmap='coolwarm', annot=True)
plt.title('Correlation Matrix')
plt.show()
```

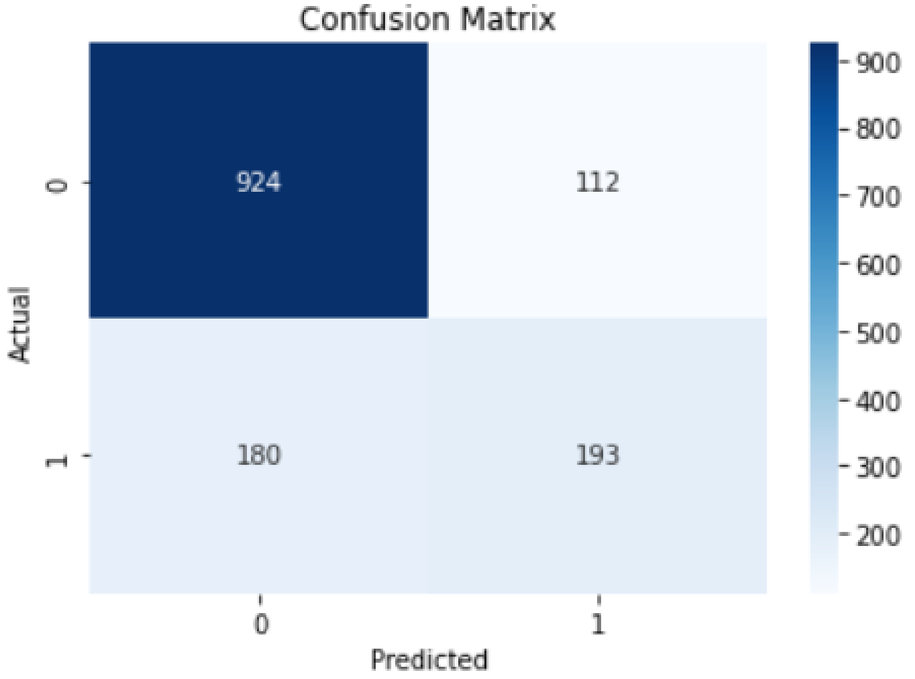


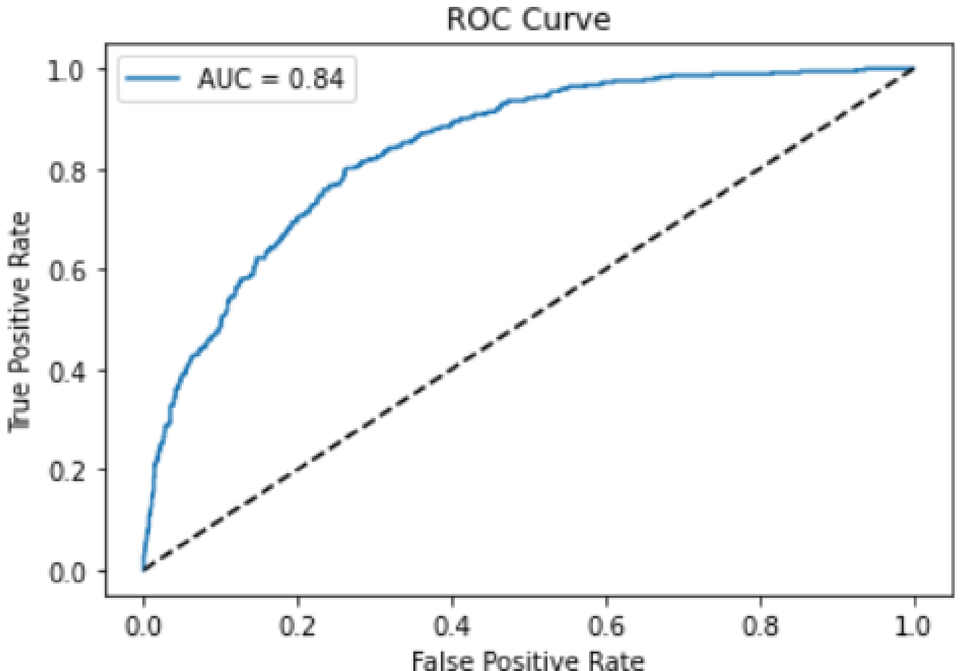
Distribution Curve:

```
fig, ax = plt.subplots(1,3,figsize=(15,5))

sns.histplot(x='tenure', data=New_data_copy, kde=True, ax=ax[0])
ax[0].set_title('Distribution of tenure')

sns.histplot(x='MonthlyCharges', data=New_data_copy, kde=True, ax=ax[1])
ax[1].set_title('Distribution of MonthlyCharges')

sns.histplot(x='TotalCharges', data=New_data_copy, kde=True, ax=ax[2])
ax[2].set_title('Distribution of TotalCharges')

plt.show()
```



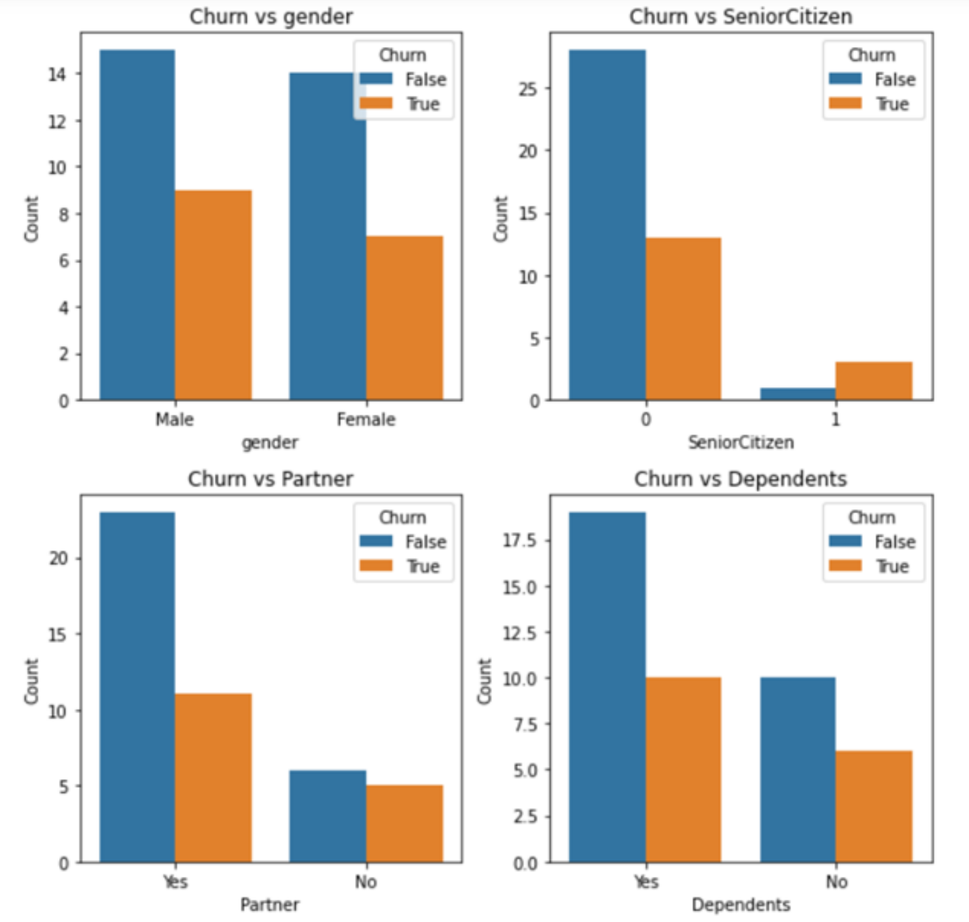
Confusion Matrix and Roc Curve :





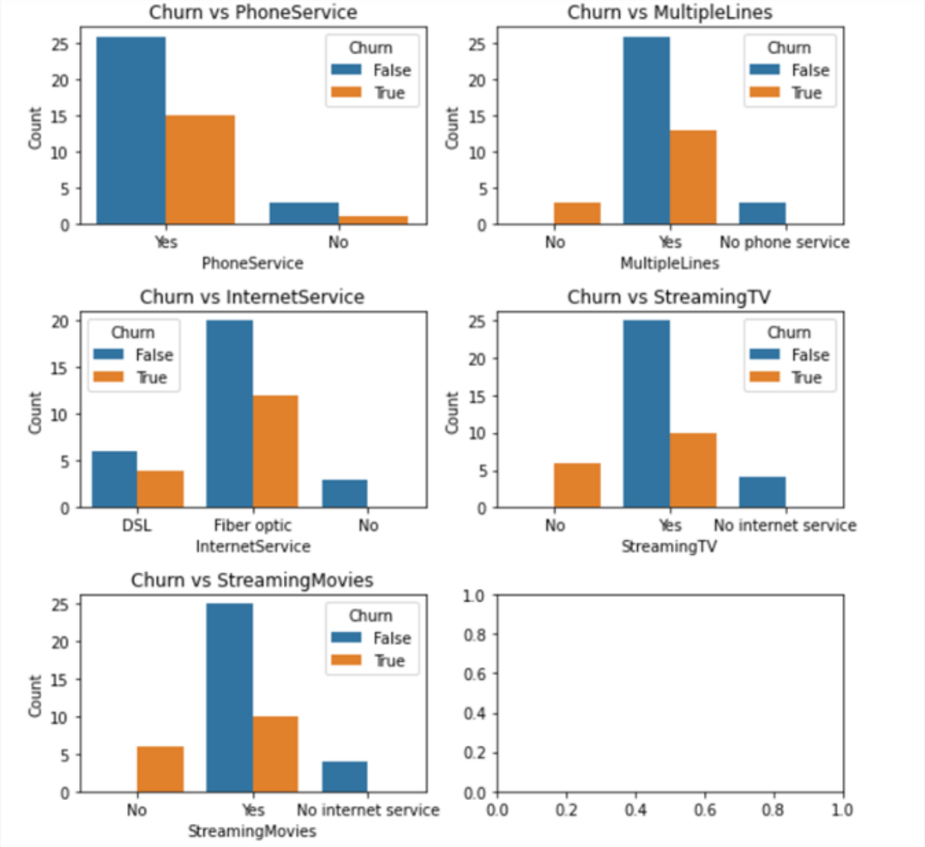
TELCO

Graphical Analysis



TELCO

Graphical Analysis



Insights

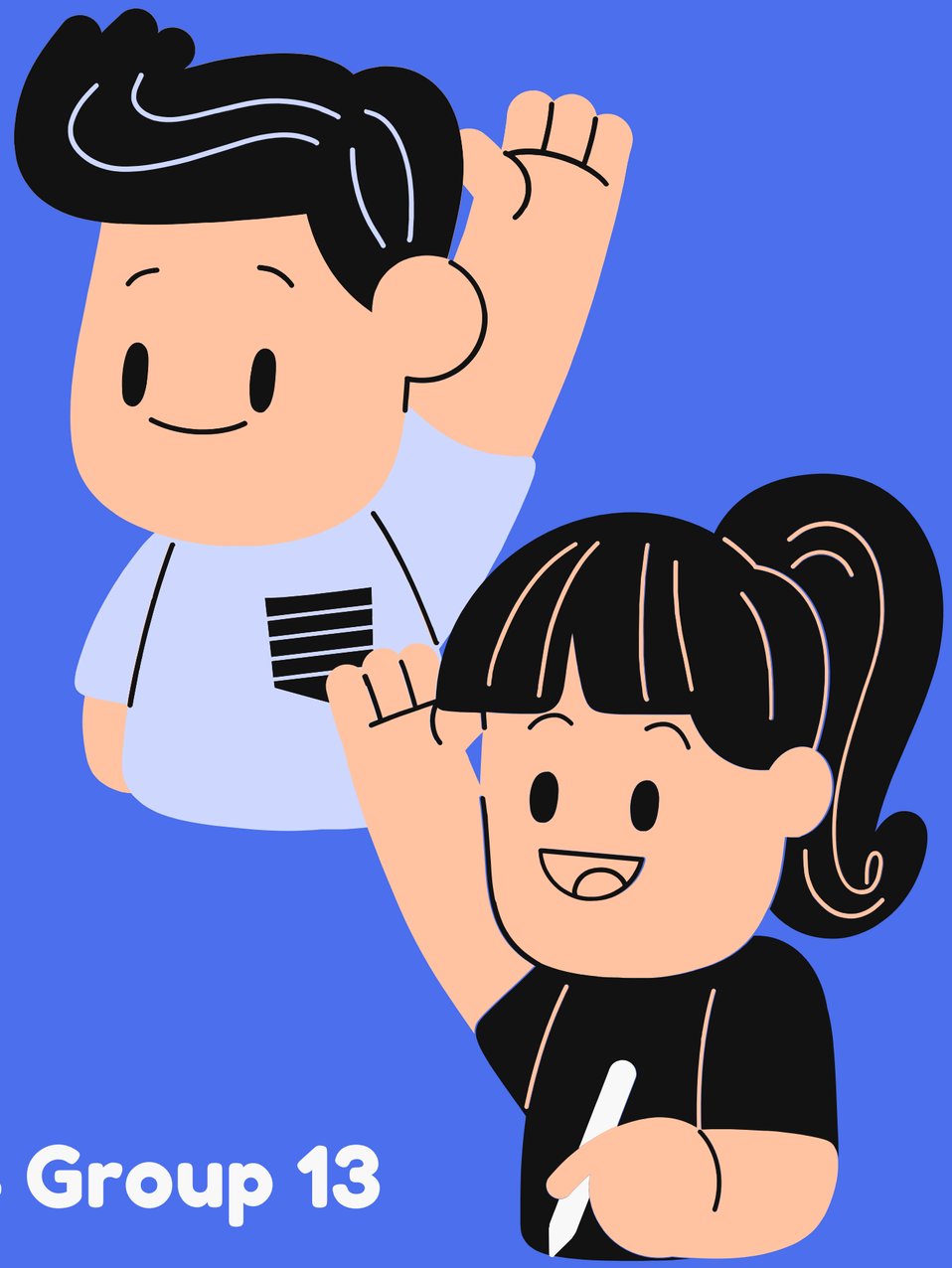
- Confusion matrix shows that the model correctly predicted 1127 true positives (customers who churned) and 391 true negatives (customers who did not churn).
- Negative correlation: between Tenure and Churn, which indicates that customers who have been with the telco service for longer are less likely to churn.
- The histogram suggests that there is a large number of customers who have subscribed to the service for less than 10 months, and the number of customers decreases as the length of tenure increases.
- Customers who have fiber optic internet service are more likely to churn than customers with DSL internet service.
- Customers who have month-to-month contracts are more likely to churn than customers with one or two-year contracts.
- Customers who pay higher monthly charges are more likely to churn than customers who pay lower monthly charges.

TELCO

Takeways :

- Identification of customer behavior patterns and preferences that lead to churn across the three different customer segments
- Segmentation of customers into three different groups based on their behavior and demographic data, allows companies to tailor retention strategies to each group.
- Prediction of whether each customer in the different segments is likely to churn or not using logistic regression, enables companies to take proactive measures to prevent churn.
- Understanding of which factors have the most significant impact on churn within each customer segment and how they relate to each other.
- Development of a data-driven approach to customer retention and experience improvement, leading to increased customer satisfaction and loyalty across the three customer segments.

Thank You !!



By - Group 13