Customer Churn Analysis

Team Members

1. Anshika Singhal
2. Ganesh Bagal
3. Akshaj Agarwal
4. Gohit Tyagi

12th May, 2023

Step 1: Prototype Selection

Problem Statement:

The objective of this project is to analyze customer behavior and identify patterns that indicate an increased risk of churn. Churn is defined as the number of customers who discontinue using a company's products or services within a given time frame. The project aims to identify the factors that contribute to customer churn and develop strategies to mitigate it. The project will analyze customer data such as usage patterns, customer demographics, purchase history, and other relevant metrics to identify patterns that can help to predict which customers are at risk of churning. The goal is to develop a predictive model that can be used to identify at-risk customers early and proactively engage with them to prevent churn. Ultimately, the goal is to increase customer retention, improve customer satisfaction, and drive revenue growth.

Market/Customer/Business Need Assessment:

Market/Customer/Business Need Assessment is a crucial step in any ML project as it helps to understand the problem from different perspectives and identify the key challenges, risks and growth opportunities. For a customer churn analysis project, the assessment can be as follows:

Market Need Assessment:

- In today's highly competitive market, customer retention has become a key factor for the success of any business.
- Customer churn can have a significant impact on the revenue and profitability of a business, as it can lead to a decrease in sales, loss of market share, and increased costs of acquiring new customers.
- Therefore, there is a growing need for businesses to identify and target customers who are at risk of churning and take proactive measures and actions to retain them.

Customer Need Assessment:

- Customers today have a wide range of options and are more likely to switch to a competitor if they are dissatisfied with the products or services of a company.
- Customers expect personalized experiences and proactive engagement from the companies they do business with.
- Therefore, customers need companies to anticipate their needs, provide relevant offers, solutions and deliver exceptional customer service to retain their loyalty.

Business Need Assessment:

- Businesses need to retain their customers to maintain a stable revenue stream and remain competitive.
- Identifying customers who are at risk of churning can help businesses to take proactive measures and derive customer eccentric solutions to retain them, such as personalized offers, loyalty programs, and improved customer service.
- Businesses can also use churn analysis to identify opportunities for upselling and cross-selling to existing customers.
- In summary, a customer churn analysis project can help businesses to retain their customers, improve customer satisfaction, and drive revenue growth by identifying customers who are at risk of churning and taking proactive measures to retain them.

Target Specifications and Characterization (your customer characteristic):

Target specifications and characterization depends on the specific business and industry, but generally, the following characteristics can be considered for identifying and predicting customer churn:

- **Customer demographic and behavioral data** - This includes information about the customer's age, gender, income, location, buying patterns, purchase/renewal history, customer service interactions, and other relevant metrics.
- **Churn definition** - A clear definition of churn is essential to accurately identify at-risk customers. For example, churn could be defined as customers who have not made a purchase in the last 30 days or customers who have canceled their subscription.
- **Data quality** - The quality and completeness of the data used for analysis are critical to ensure accurate predictions. It is essential to have data from multiple sources and clean and preprocess the data before feeding it into the ML models.
- **Feature selection** - Identifying the most relevant features that contribute to customer churn is essential for developing accurate models. It requires domain knowledge and expertise in data analysis and feature engineering.

- **ML model selection** - There are various ML models that can be used for customer churn analysis, such as logistic regression, decision trees, random forests, support vector machines, and neural networks. The choice of the model depends on the size and complexity of the data and the accuracy and interpretability of the model.
- **Evaluation metrics** - The performance of the ML models needs to be evaluated using appropriate metrics, such as accuracy, precision, recall, F1-score, ROC curve, and AUC.
- **Actionable insights** - The ML models should provide actionable insights that can be used to develop targeted retention strategies, such as personalized offers, Volume Discounts, loyalty programs, and improved customer service.

In summary, the target specifications and characterization for a customer churn analysis ML project should focus on identifying and predicting customer churn accurately and providing actionable insights that can help businesses to retain their customers and improve customer satisfaction.

External Search (online information sources/references/links):

Dataset:

https://www.kaggle.com/datasets/blastchar/telco-customer-churn

Research:

1. CUSTOMER CHURN PREDICTION by
   Senthilnayaki B , Swetha M , Nivedha D
2. Customer Churn Prediction: Survey by
   Ms. Chinnu P Johny, Mr. Paul P. Matha

Online Information:

1. https://www.paddle.com/resources/customer-churn-analysis
2. https://www.netsuite.com/portal/resource/articles/human-resources/customer-churn-analysis.shtml
3. https://www.projectpro.io/article/churn-models/709#:~:text=Bayes%20Algorithm&text=The%20output%20in%20this%20algorithm,provider%20or%20picking%20another%20one.

Applicable Constraints:

There are several constraints that should be taken into consideration when conducting a customer churn analysis, some of which include:

1. **Data Availability:** The availability and quality of data is crucial for any ML project. It's important to ensure that the dataset is large enough and contains relevant information about customer behavior and preferences.
2. **Time and Budget**: Time and budget constraints can limit the scope of the project. For instance, if the timeline is tight, the ML model may need to be simplified, and the dataset may need to be smaller.
3. **Model Performance:** The model's performance is an important constraint to consider. If the model does not perform well, it may not be useful in making accurate predictions about customer churn.
4. **Business Constraints:** The ML model should align with the business objectives and constraints. For instance, if the business has limited resources to take actions based on the model's predictions, the model should be designed accordingly.
5. **Privacy and Security:** It's important to ensure that the data used in the project is collected and stored securely, and any personal or sensitive information is protected or encrypted.
6. **Legal and Ethical Constraints:** ML projects should comply with legal and ethical constraints, such as data protection laws, fairness and bias in the model's predictions, and transparency in the decision-making process.

Overall, considering these constraints during customer churn analysis will help to ensure that the model is accurate, reliable, and useful for the business.

Business Model (Monetization Idea):

1. **Subscription-based model:** A subscription-based model can be used where businesses can pay a recurring fee to access the ML model and receive insights on customer churn. This model can be useful for businesses that need ongoing support and guidance to manage customer churn.
2. **Pay-per-use model:** In a pay-per-use model, businesses pay a fee for each time they use the ML model. This model can be suitable for businesses that don't require constant support and only need insights occasionally.
3. **Commission-based model**: A commission-based model can be used where businesses pay a commission on the revenue generated by retaining customers predicted to churn. This model can be beneficial for businesses that need to justify the cost of using the ML model and want to see a direct return on investment.
4. **Value-based pricing model**: In a value-based pricing model, businesses pay based on the value they receive from the ML model. For instance, if the ML model helps a business retain customers that generate significant revenue, the price for using the ML model may be higher.
5. **Consulting services:** A business can provide consulting services based on the insights and recommendations provided by the ML model. This can be a valuable service for businesses that need more personalized support and guidance.

6. **Volume Based:** In Volume based pricing model, the business can generate a revenue from the factor of the volume of usability of the customer. For example if the customer is using more features of the product then he/she will be applicable for more discounts in the future.

Overall, choosing the right monetization model depends on the business's needs and goals.

Concept Development (Brief summary of Product/Service will be developed):

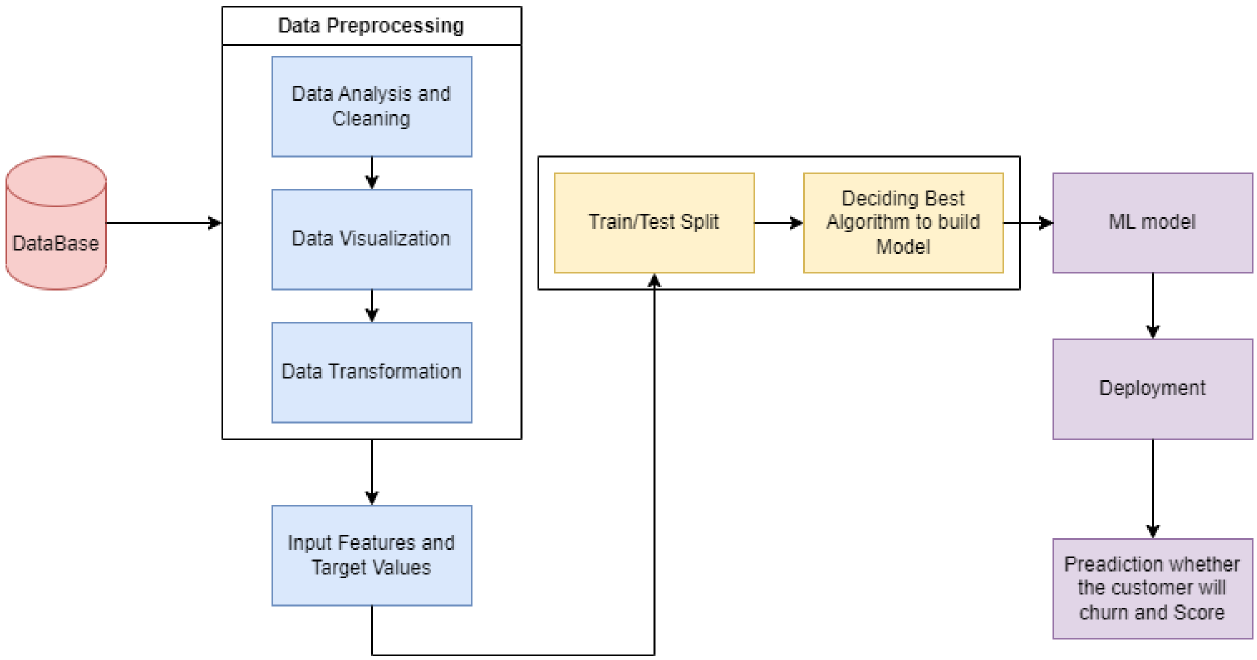
Customer churn analysis is a project that aims to develop a tool or service to help businesses analyze and predict customer churn. Churn refers to the rate at which customers stop doing business with a company or cancel their subscriptions or services. This can have a significant impact on a company's revenue and profitability, so being able to predict and prevent churn is crucial.

The product/service that will be developed will use machine learning algorithms to analyze customer behavior data and identify patterns that are indicative of churn. This data can be obtained from various sources such as customer surveys, customer interaction logs, social media, and other relevant data sources.

The tool will use these insights to provide recommendations to the business on how to improve customer retention and reduce churn. This may include targeted marketing campaigns, personalized offers, or other strategies to increase customer satisfaction and loyalty.

Overall, the goal of this project is to help businesses retain customers and increase their revenue by using data-driven insights to predict and prevent customer churn.

Schematic Diagram:



Product details:

1. Data Sources:

Kaggle: https://www.kaggle.com/datasets/blastchar/telco-customer-churn

2. Algorithms:

- Decision Tree Classifier
- Random Forest Classifier
- PCA (Principal Component Analysis)
- Pickling the Model

3. Software:

- Jupyter Notebook
- Pycharm
- HTML

4. Frameworks:

- Python
- Flask
- Numpy
- Pandas

- Seaborn
- Matplotlib
- Sklearn
- SMOTEEN

Team required to develop:

To develop a customer churn analysis, you will need a team with the following roles and skills:

1. **Project Manager:** Responsible for managing the project timeline, coordinating with the team, and ensuring that the analysis is delivered on time.
2. **Data Analyst**: Responsible for collecting and analyzing data on customer behavior, such as purchase history, usage patterns, and customer feedback.
3. **Data Scientist**: Responsible for building predictive models to identify customers who are likely to churn, based on historical data and other factors such as demographics and customer preferences.
4. **Database Administrator**: Responsible for managing the data infrastructure and ensuring that data is stored securely and efficiently.
5. **Business Analyst:** Responsible for understanding the business context and translating the analysis results into actionable recommendations for the business along with testing of the built product to see whether it meets the business requirements.
6. **Visualization Specialist:** Responsible for creating visualizations that communicate the results of the analysis to stakeholders in a clear and compelling way.
7. **Domain Expert:** A person with knowledge of the industry and the business who can provide valuable insights and guidance to the team.

By bringing together these skills, the team can develop a comprehensive customer churn analysis that can help the business reduce churn and improve customer retention.

Step 2: Prototype Development

Code Implementation:

1. Simple EDA(Exploratory Data Analysis)(Jupyter Notebook):

Dataset Analysis:

In [6]:
```
telco_base_data.head()
```

Out[6]:

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

5 rows × 21 columns

In [8]:
```
telco_base_data.columns.values
```

Out[8]:
```
array(['customerID', 'gender', 'SeniorCitizen', 'Partner', 'Dependents',
       'tenure', 'PhoneService', 'MultipleLines', 'InternetService',
       'OnlineSecurity', 'OnlineBackup', 'DeviceProtection',
       'TechSupport', 'StreamingTV', 'StreamingMovies', 'Contract',
       'PaperlessBilling', 'PaymentMethod', 'MonthlyCharges',
       'TotalCharges', 'Churn'], dtype=object)
```

In [10]:
```
# Check the descriptive statistics of numeric variables
telco_base_data.describe()
```

Out[10]:

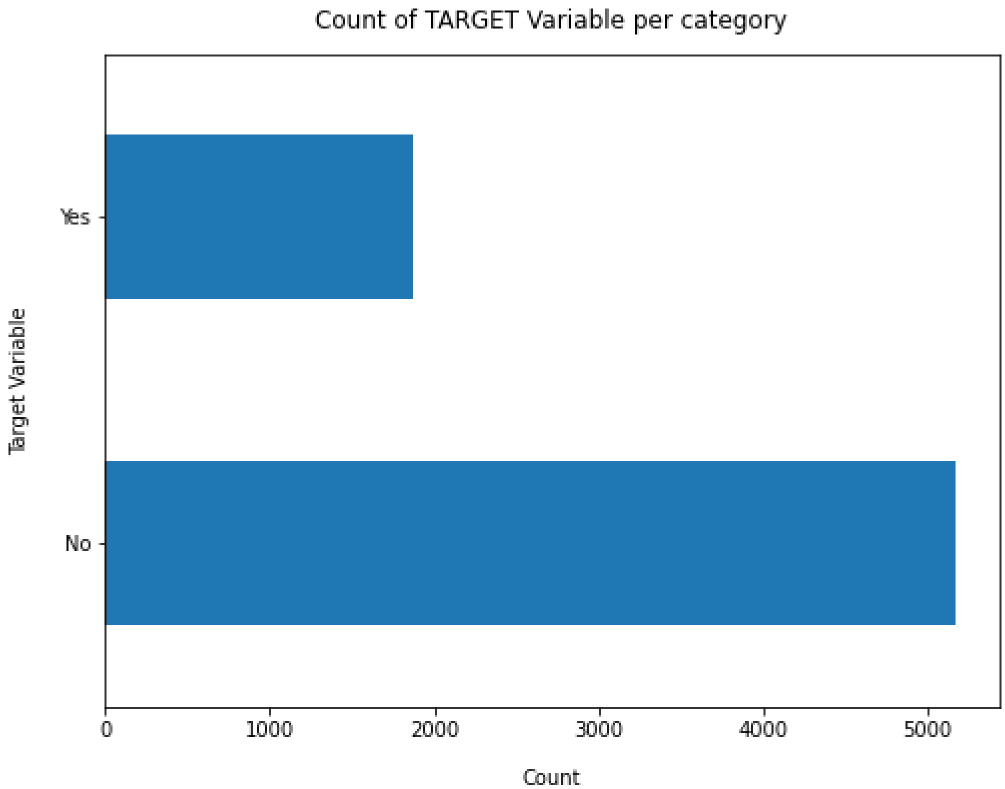
	SeniorCitizen	tenure	MonthlyCharges
count	7043.000000	7043.000000	7043.000000
mean	0.162147	32.371149	64.761692
std	0.368612	24.559481	30.090047
min	0.000000	0.000000	18.250000
25%	0.000000	9.000000	35.500000
50%	0.000000	29.000000	70.350000
75%	0.000000	55.000000	89.850000
max	1.000000	72.000000	118.750000

SeniorCitizen is actually a categorical hence the 25%-50%-75% distribution is not propoer

75% customers have tenure less than 55 months

Average Monthly charges are USD 64.76 whereas 25% customers pay more than USD 89.85 per month

In [11]:
```python
telco_base_data['Churn'].value_counts().plot(kind='barh', figsize=(8, 6))
plt.xlabel("Count", labelpad=14)
plt.ylabel("Target Variable", labelpad=14)
plt.title("Count of TARGET Variable per category", y=1.02);
```



In [12]:
```python
#Percentage of each category
100*telco_base_data['Churn'].value_counts()/len(telco_base_data['Churn'])
```

Out[12]:
```
No     73.463013
Yes    26.536987
Name: Churn, dtype: float64
```

In [13]:
```python
#Churners count - categorical data
telco_base_data['Churn'].value_counts()
```

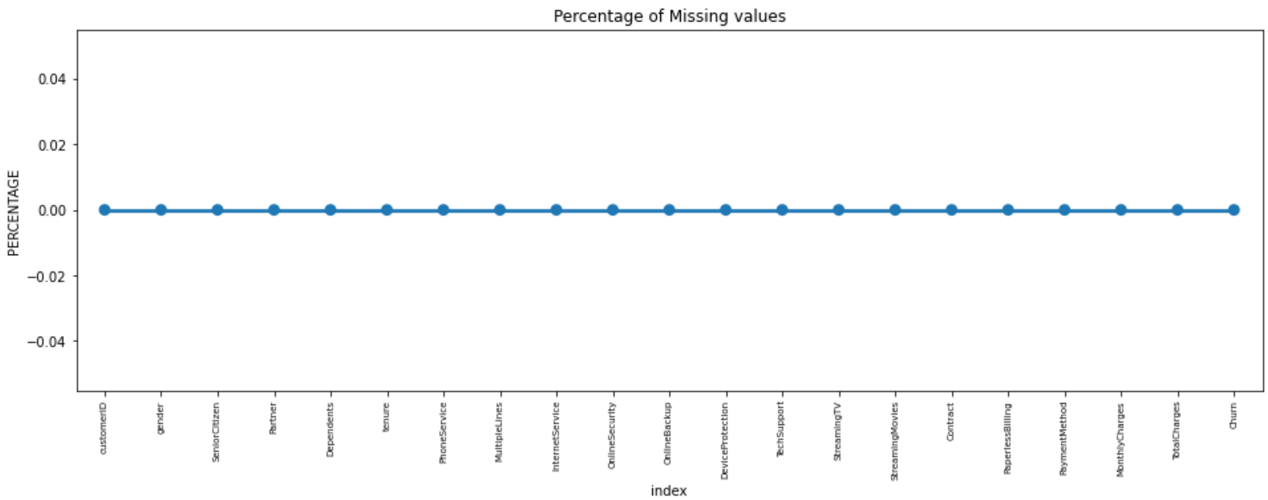
Out[13]:
```
No     5174
Yes    1869
Name: Churn, dtype: int64
```

- Data is highly imbalanced, ratio = 73:27
- So we analyse the data with other features while taking the target values separately to get some insights.

In [15]:

```
missing = pd.DataFrame((telco_base_data.isnull().sum())*100/telco_base_data.shape[0]).reset_index()
plt.figure(figsize=(16,5))
ax = sns.pointplot('index',0,data=missing)
plt.xticks(rotation =90,fontsize =7)
plt.title("Percentage of Missing values")
plt.ylabel("PERCENTAGE")
plt.show()
```

```
C:\Users\User\AppData\Local\Programs\Python\Python39\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following vai
word args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit
result in an error or misinterpretation.
  warnings.warn(
```



Missing Data - Initial Intuition

- Here, we don't have any missing data.

General Thumb Rules:

- For features with less missing values- can use regression to predict the missing values or fill with the mean of the values present, depending on the feature.
- For features with very high number of missing values- it is better to drop those columns as they give very less insight on analysis.
- As there's no thumb rule on what criteria do we delete the columns with high number of missing values, but generally you can delete the columns, if you have more than 30-40% of missing values. But again there's a catch here, for example, Is_Car & Car_Type, People having no cars, will obviously have Car_Type as NaN (null), but that doesn't make this column useless, so decisions has to be taken wisely.

Data Cleaning:

In [17]:
```
telco_data.TotalCharges = pd.to_numeric(telco_data.TotalCharges, errors='coerce')
telco_data.isnull().sum() #How many values are null
```

Out[17]:
```
customerID          0
gender              0
SeniorCitizen       0
Partner             0
Dependents          0
tenure              0
PhoneService        0
MultipleLines       0
InternetService     0
OnlineSecurity      0
OnlineBackup        0
DeviceProtection    0
TechSupport         0
StreamingTV         0
StreamingMovies     0
Contract            0
PaperlessBilling    0
PaymentMethod       0
MonthlyCharges      0
TotalCharges       11
Churn               0
dtype: int64
```

4. Missing Value Treatement

Since the % of these records compared to total dataset is very low i.e 0.15%, it is safe to ignore them from further processing.

]:
```
#Removing missing values
telco_data.dropna(how = 'any', inplace = True)

#telco_data.fillna(0)
```

5. Divide customers into bins based on tenure e.g. for tenure < 12 months: assign a tenure group if 1-12, for tenure between 1 to 2 Yrs, tenure group of 13-24; so on...

In [20]:
```
# Get the max tenure
print(telco_data['tenure'].max()) #72
```

```
72
```

In [21]:
```
# Group the tenure in bins of 12 months
labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

telco_data['tenure_group'] = pd.cut(telco_data.tenure, range(1, 80, 12), right=False, labels=labels)
```

In [22]:
```
telco_data['tenure_group'].value_counts()
```

Out[22]:
```
1 - 12     2175
61 - 72    1407
13 - 24    1024
25 - 36     832
49 - 60     832
37 - 48     762
Name: tenure_group, dtype: int64
```

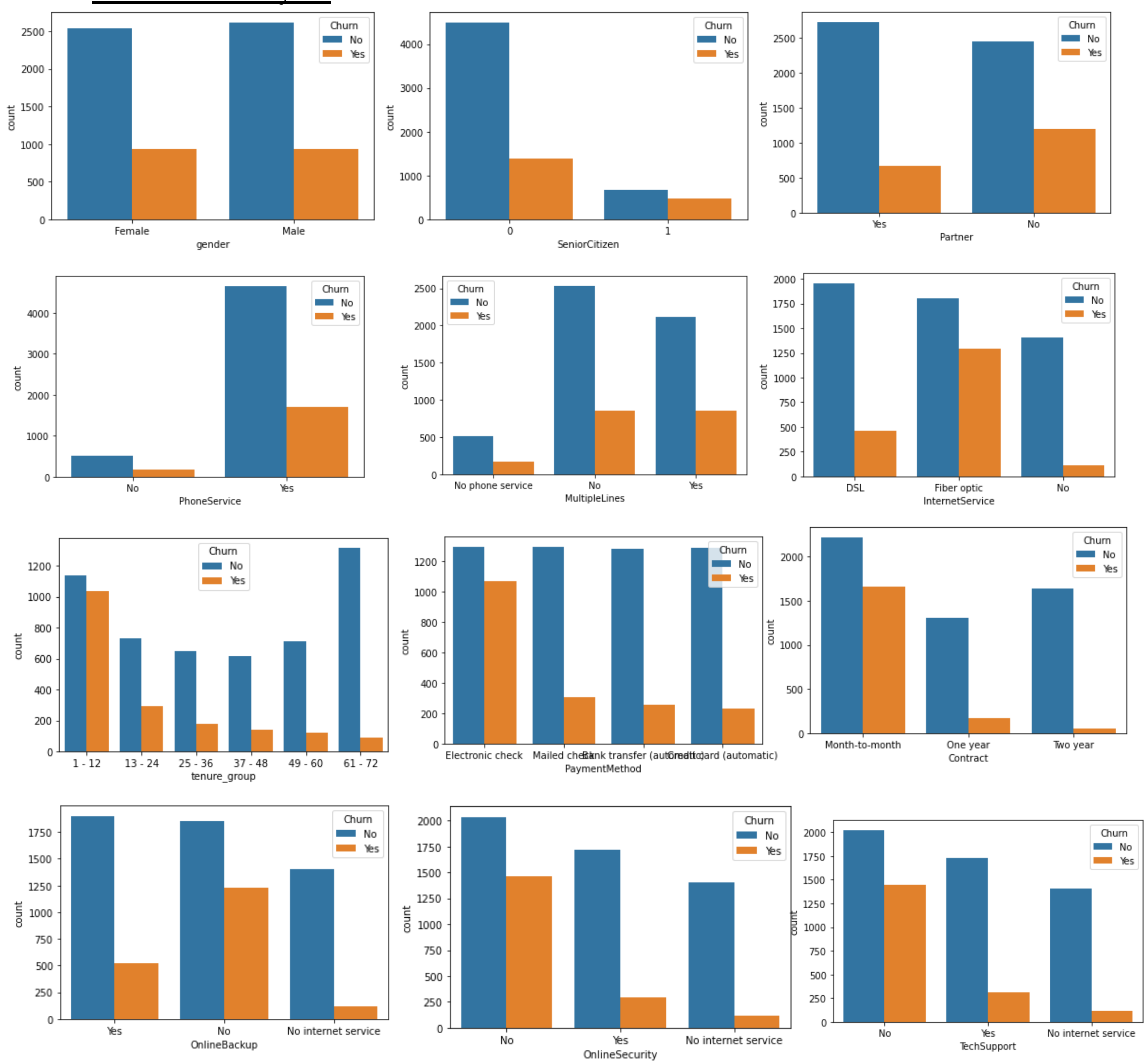
6. Remove columns not required for processing

```python
#drop column customerID and tenure
telco_data.drop(columns= ['customerID','tenure'], axis=1, inplace=True)
telco_data.head()
#No need for tenure as we already ceated tenure group
```

Data Exploration:

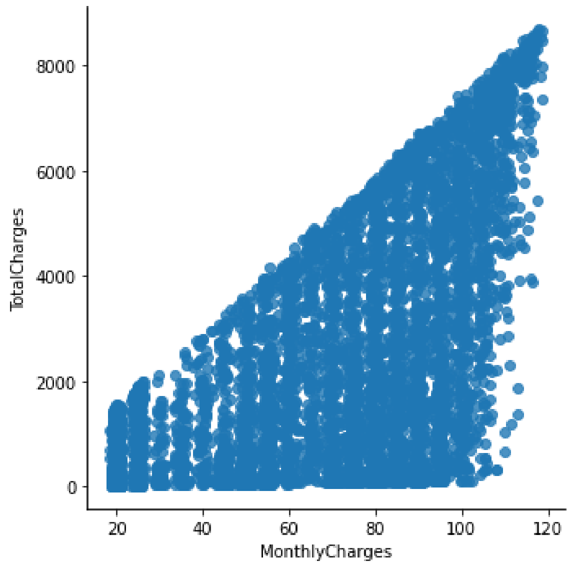
1. Plot distribution of individual predictors by churn

   Univariate Analysis:

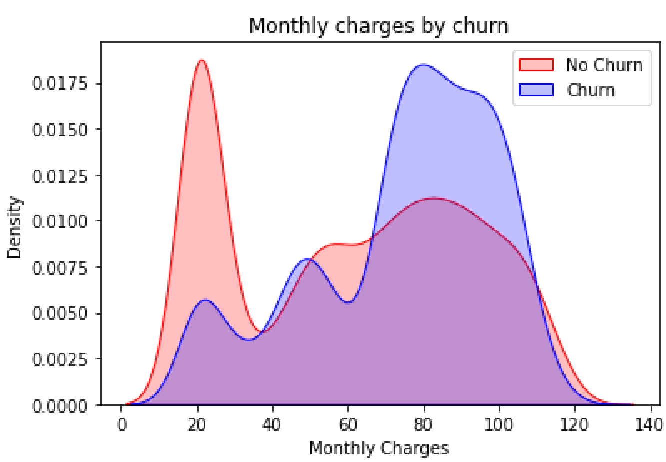


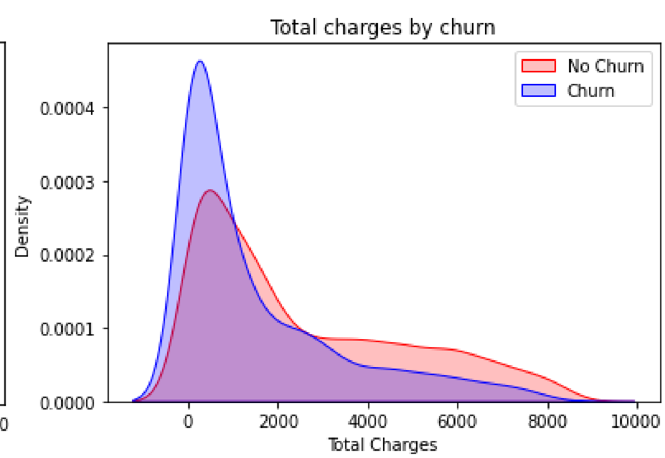
4. Relationship between Monthly Charges and Total Charges

```
In [28]: sns.lmplot(data=telco_data_dummies, x='MonthlyCharges', y='TotalCharges', fit_reg=False)
Out[28]: <seaborn.axisgrid.FacetGrid at 0x10bd8446520>
```



Total Charges increase as Monthly Charges increase - as expected.





Surprising insight as higher Churn at lower Total Charges

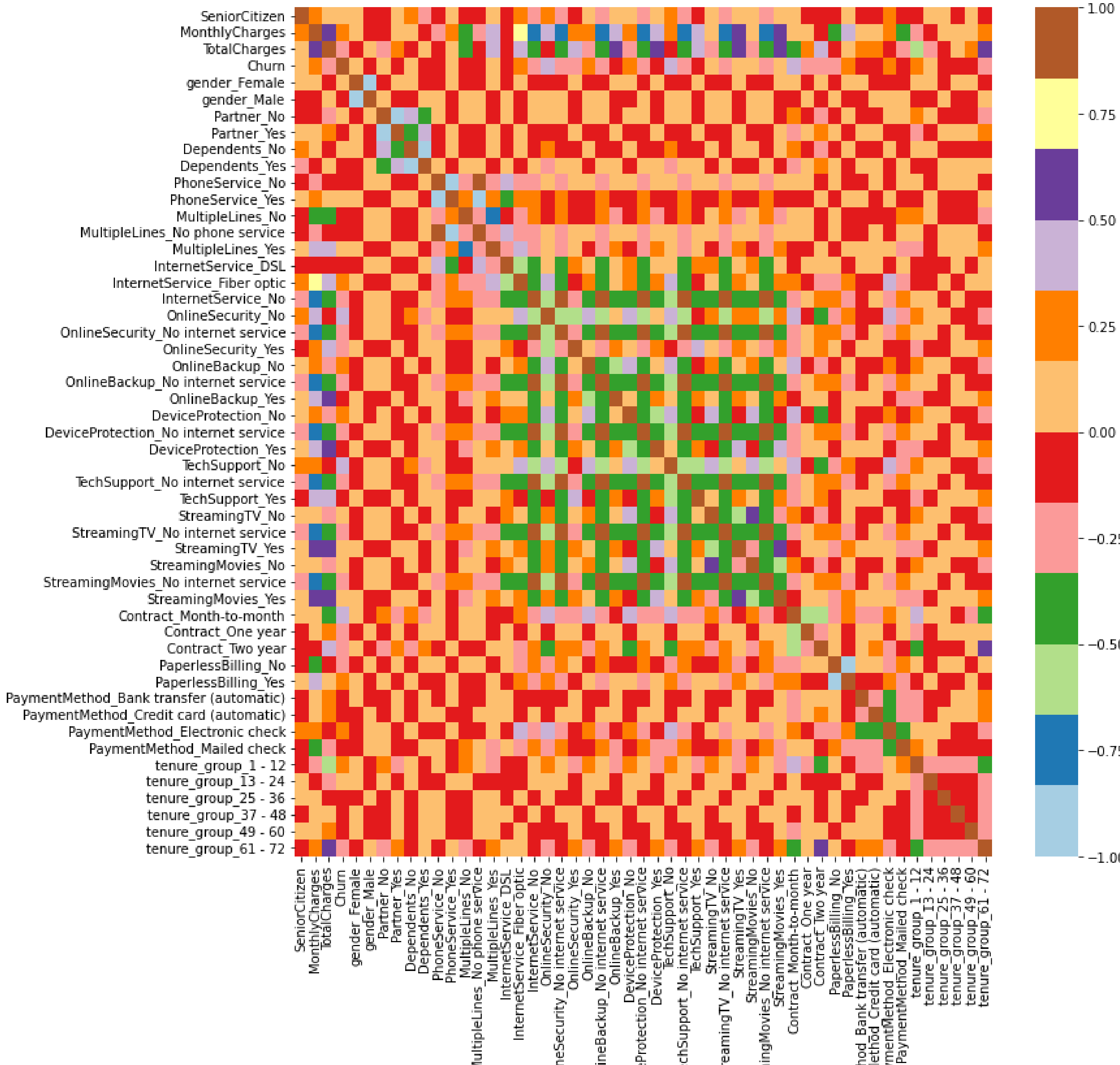
However if we combine the insights of 3 parameters i.e. Tenure, Monthly Charges & Total Charges then the picture is bit clear :- Higher Monthly Charge at lower tenure results into lower Total Charge. Hence, all these 3 factors viz **Higher Monthly Charge**, **Lower tenure** and **Lower Total Charge** are linked to **High Churn**.

HIGH Churn seen in case of **Month to month contracts**, **No online security**, **No Tech support**, **First year of subscription** and **Fibre Optics Internet**

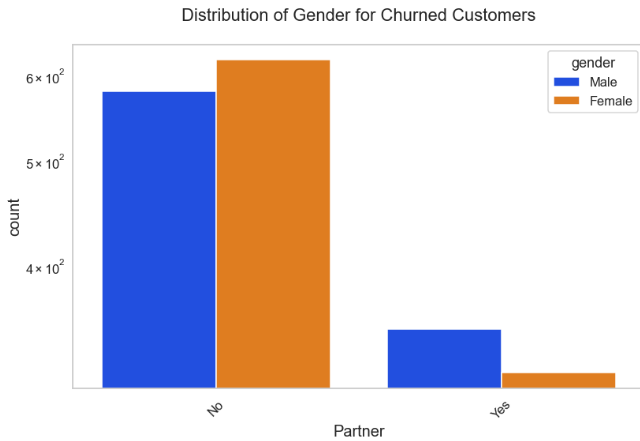
LOW Churn is seen in case of **Long term contracts**, **Subscriptions without internet service** and **The customers engaged for 5+ years**

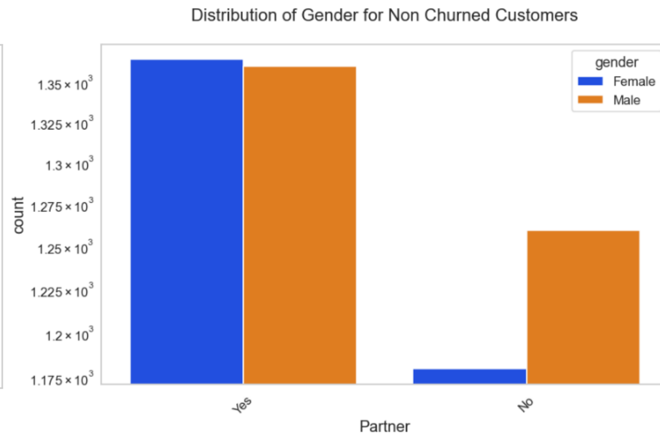
Factors like **Gender**, **Availability of Phone Service** and **# of multiple lines** have almost **NO** impact on Churn

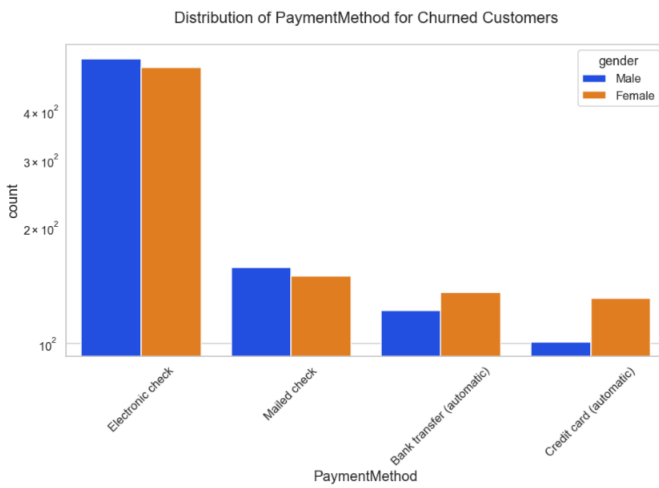
This is also evident from the **Heatmap** below:

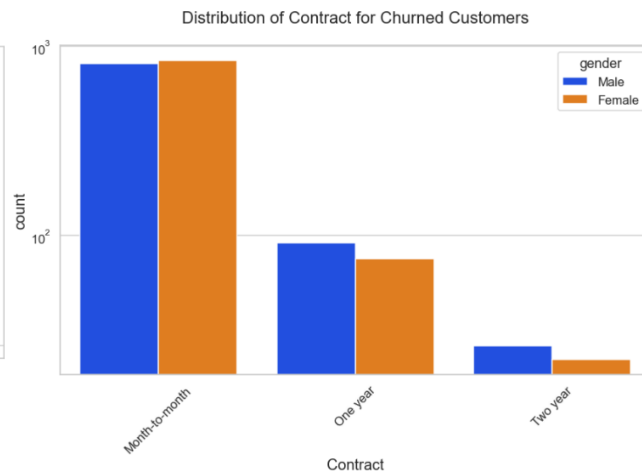


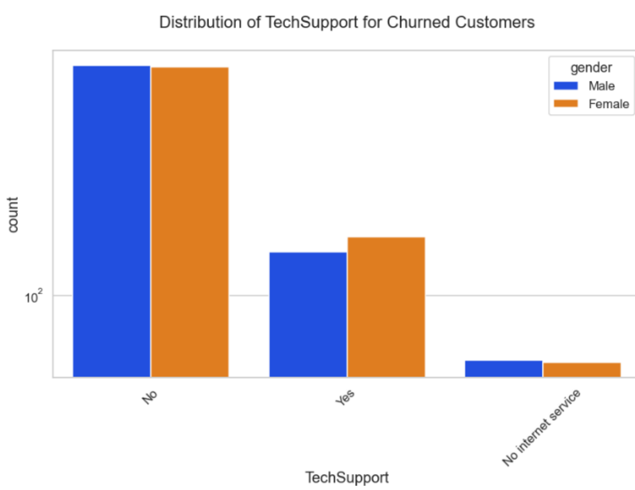
2. Bivariate Analysis:

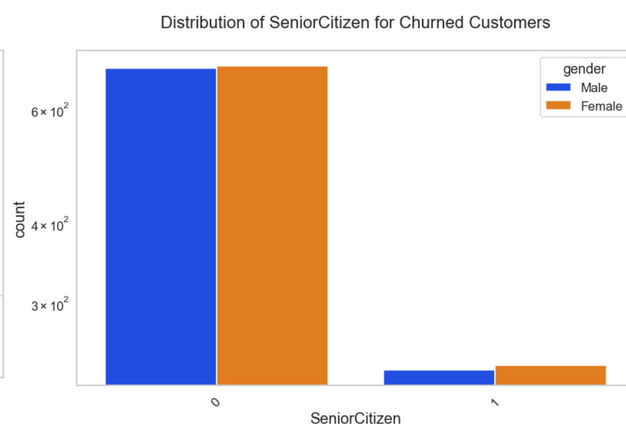












These are some of the quick insights from this exercise:

1. Electronic check medium are the highest churners
2. Contract Type - Monthly customers are more likely to churn because of no contract terms, as they are free to go customers.
3. No Online security, No Tech Support category are high churners
4. Non senior Citizens are high churners

2. <u>ML Modelling(Jupyter Notebook):</u>

```
import pandas as pd
from sklearn import metrics
from sklearn.model_selection import train_test_split
from sklearn.metrics import recall_score
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.tree import DecisionTreeClassifier
from imblearn.combine import SMOTEENN

df=pd.read_csv("tel_churn.csv")
df.head()

df=df.drop('Unnamed: 0',axis=1)
df.head(5)

#Creating X and Y variables - Input and Output Variables
x=df.drop('Churn',axis=1)

y=df['Churn']

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)
```

#**<u>Decision Tree Classifier:</u>**

```
model_dt=DecisionTreeClassifier(criterion = "gini",random_state = 100,max_depth=6,
min_samples_leaf=8)
model_dt.fit(x_train,y_train)
y_pred=model_dt.predict(x_test)
model_dt.score(x_test,y_test)
print(classification_report(y_test, y_pred, labels=[0,1]))
```

	precision	recall	f1-score	support
0	0.83	0.88	0.85	1037
1	0.60	0.49	0.54	370
accuracy			0.78	1407
macro avg	0.71	0.69	0.70	1407
weighted avg	0.77	0.78	0.77	1407

As you can see that the accuracy is quite low, and as it's an imbalanced dataset, we shouldn't consider Accuracy as our metrics to measure the model, as Accuracy is cursed in imbalanced datasets.

Hence, we need to check recall, precision & f1 score for the minority class, and it's quite evident that the precision, recall & f1 score is too low for Class 1, i.e. churned customers.

Hence, moving ahead to call SMOTEENN (UpSampling + ENN)

```
sm = SMOTEENN()
X_resampled, y_resampled = sm.fit_resample(x,y)

xr_train,xr_test,yr_train,yr_test=train_test_split(X_resampled,y_resampled,test_size=0.2)

model_dt_smote=DecisionTreeClassifier(criterion = "gini",random_state =
100,max_depth=6, min_samples_leaf=8)

model_dt_smote.fit(xr_train,yr_train)
yr_predict = model_dt_smote.predict(xr_test)
model_score_r = model_dt_smote.score(xr_test, yr_test)
print(model_score_r)
print(metrics.classification_report(yr_test, yr_predict))
```

0.9368600682593856

	precision	recall	f1-score	support
0	0.93	0.93	0.93	526
1	0.94	0.94	0.94	646
accuracy			0.94	1172
macro avg	0.94	0.94	0.94	1172
weighted avg	0.94	0.94	0.94	1172

Now we can see quite better results, i.e. Accuracy: 92 %, and a very good recall, precision & f1 score for minority class.

Let's try with some other classifier.

Random Forest Classifier:

from sklearn.ensemble import RandomForestClassifier

model_rf=RandomForestClassifier(n_estimators=100, criterion='gini', random_state = 100,max_depth=6, min_samples_leaf=8)

model_rf.fit(x_train,y_train)

y_pred=model_rf.predict(x_test)

model_rf.score(x_test,y_test)

print(classification_report(y_test, y_pred, labels=[0,1]))

```
              precision    recall  f1-score   support

           0       0.82      0.92      0.87      1037
           1       0.67      0.45      0.54       370

    accuracy                           0.80      1407
   macro avg       0.75      0.68      0.70      1407
weighted avg       0.78      0.80      0.78      1407
```

sm = SMOTEENN()
X_resampled1, y_resampled1 = sm.fit_resample(x,y)

xr_train1,xr_test1,yr_train1,yr_test1=train_test_split(X_resampled1, y_resampled1,test_size=0.2)

model_rf_smote=RandomForestClassifier(n_estimators=100, criterion='gini', random_state = 100,max_depth=6, min_samples_leaf=8)

model_rf_smote.fit(xr_train1,yr_train1)

yr_predict1 = model_rf_smote.predict(xr_test1)

model_score_r1 = model_rf_smote.score(xr_test1, yr_test1)

print(metrics.confusion_matrix(yr_test1, yr_predict1))

```
[[492  59]
 [ 14 611]]
```

print(model_score_r1)
print(metrics.classification_report(yr_test1, yr_predict1))

```
0.9379251700680272
              precision    recall  f1-score   support

           0       0.97      0.89      0.93       551
           1       0.91      0.98      0.94       625

    accuracy                           0.94      1176
   macro avg       0.94      0.94      0.94      1176
weighted avg       0.94      0.94      0.94      1176
```

With RF Classifier, also we are able to get quite good results, infact better than Decision Tree.

let's finalize the model which was created by RF Classifier, and save the model so that we can use it in a later stage.

#**Pickling the Model:**

```
import pickle
filename = 'model.sav'
pickle.dump(model_rf_smote, open(filename, 'wb'))
load_model = pickle.load(open(filename, 'rb'))
model_score_r1 = load_model.score(xr_test1, yr_test1)
model_score_r1
```

```
Out[43]: 0.9379251700680272
```

Our final model i.e. RF Classifier with SMOTEENN, is now ready and dumped in model.sav, which we will use and prepare API's so that we can access our model from UI.

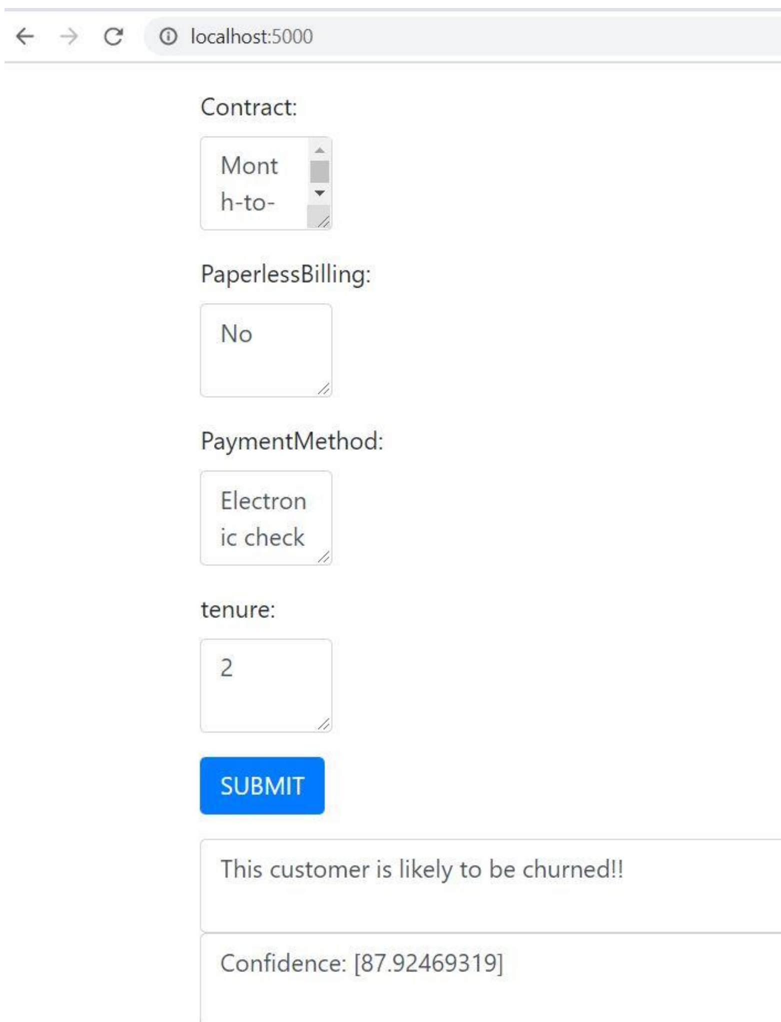
3. Model Deployment(Pycharm):

```
import pandas as pd
import sklearn
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics
from flask import Flask, request, render_template
import pickle

app = Flask("__name__")

df_1 = pd.read_csv("first_telc.csv")
```

```python
q = ""


@app.route("/")
def loadPage():
    try:
        return render_template('home.html', query="")
    except Exception as e:
        return str(e)



@app.route("/", methods=['POST'])
def predict():
    '''
    SeniorCitizen
    MonthlyCharges
    TotalCharges
    gender
    Partner
    Dependents
    PhoneService
    MultipleLines
    InternetService
    OnlineSecurity
    OnlineBackup
    DeviceProtection
    TechSupport
    StreamingTV
    StreamingMovies
    Contract
    PaperlessBilling
    PaymentMethod
    tenure
    '''

    inputQuery1 = request.form['query1']
    inputQuery2 = request.form['query2']
    inputQuery3 = request.form['query3']
    inputQuery4 = request.form['query4']
    inputQuery5 = request.form['query5']
    inputQuery6 = request.form['query6']
    inputQuery7 = request.form['query7']
    inputQuery8 = request.form['query8']
    inputQuery9 = request.form['query9']
    inputQuery10 = request.form['query10']
    inputQuery11 = request.form['query11']
    inputQuery12 = request.form['query12']
    inputQuery13 = request.form['query13']
    inputQuery14 = request.form['query14']
    inputQuery15 = request.form['query15']
    inputQuery16 = request.form['query16']
    inputQuery17 = request.form['query17']
    inputQuery18 = request.form['query18']
    inputQuery19 = request.form['query19']

    model = pickle.load(open("model.sav", "rb"))
```

```
    data = [[inputQuery1, inputQuery2, inputQuery3, inputQuery4,
inputQuery5, inputQuery6, inputQuery7,
             inputQuery8, inputQuery9, inputQuery10, inputQuery11,
inputQuery12, inputQuery13, inputQuery14,
             inputQuery15, inputQuery16, inputQuery17, inputQuery18,
inputQuery19]]

    new_df = pd.DataFrame(data, columns=['SeniorCitizen',
'MonthlyCharges', 'TotalCharges', 'gender',
                                         'Partner', 'Dependents',
'PhoneService', 'MultipleLines', 'InternetService',
                                         'OnlineSecurity',
'OnlineBackup', 'DeviceProtection', 'TechSupport',
                                         'StreamingTV',
'StreamingMovies', 'Contract', 'PaperlessBilling',
                                         'PaymentMethod', 'tenure'])

    df_2 = pd.concat([df_1, new_df], ignore_index=True)
    # Group the tenure in bins of 12 months
    labels = ["{0} - {1}".format(i, i + 11) for i in range(1, 72, 12)]

    df_2['tenure_group'] = pd.cut(df_2.tenure.astype(int), range(1, 80,
12), right=False, labels=labels)
    # drop column customerID and tenure
    df_2.drop(columns=['tenure'], axis=1, inplace=True)

    new_df__dummies = pd.get_dummies(df_2[['gender', 'SeniorCitizen',
'Partner', 'Dependents', 'PhoneService',
                                           'MultipleLines',
'InternetService', 'OnlineSecurity', 'OnlineBackup',
                                           'DeviceProtection',
'TechSupport', 'StreamingTV', 'StreamingMovies',
                                           'Contract',
'PaperlessBilling', 'PaymentMethod', 'tenure_group']])

    # final_df=pd.concat([new_df__dummies, new_dummy], axis=1)

    single = model.predict(new_df__dummies.tail(1))
    probablity = model.predict_proba(new_df__dummies.tail(1))[:, 1]

    if single == 1:
        o1 = "This customer is likely to be churned!!"
        o2 = "Confidence: {}".format(probablity * 100)
    else:
        o1 = "This customer is likely to continue!!"
        o2 = "Confidence: {}".format(probablity * 100)

    return render_template('home.html', output1=o1, output2=o2,
                           query1=request.form['query1'],
                           query2=request.form['query2'],
                           query3=request.form['query3'],
                           query4=request.form['query4'],
                           query5=request.form['query5'],
                           query6=request.form['query6'],
                           query7=request.form['query7'],
                           query8=request.form['query8'],
```

```
                                    query9=request.form['query9'],
                                    query10=request.form['query10'],
                                    query11=request.form['query11'],
                                    query12=request.form['query12'],
                                    query13=request.form['query13'],
                                    query14=request.form['query14'],
                                    query15=request.form['query15'],
                                    query16=request.form['query16'],
                                    query17=request.form['query17'],
                                    query18=request.form['query18'],
                                    query19=request.form['query19'])


app.run()
```



Github link to the code implementation:

Find below the link to the github repository for the code implementation and more information about the code and the dataset.

https://github.com/Akshaj1017/Customer-Churn-Analysis

Conclusion:

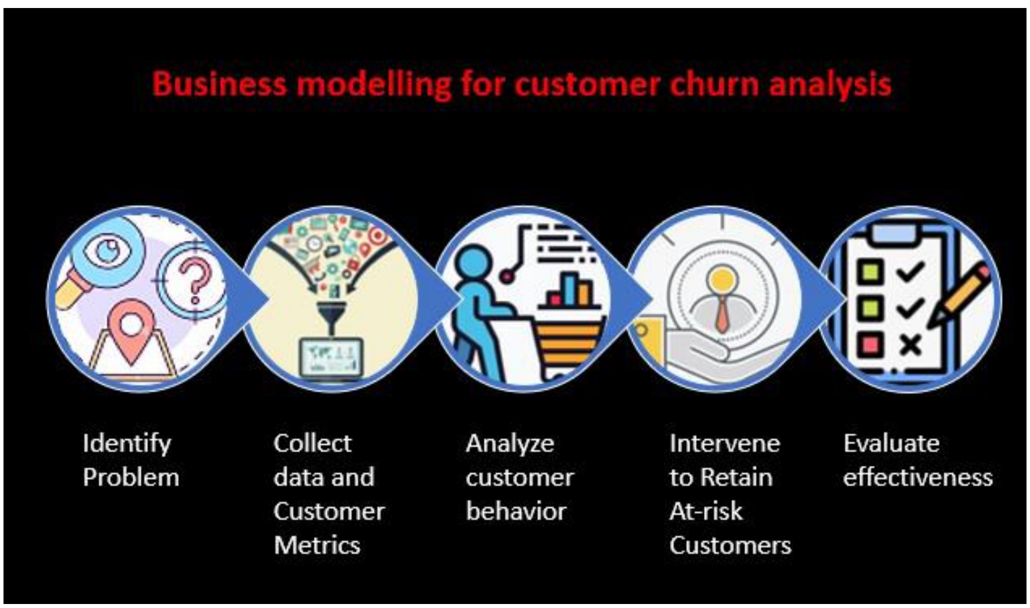
In conclusion, customer churn analysis is a crucial task for businesses looking to retain their customers and increase their revenue. Through the use of machine learning techniques, we can predict which customers are most likely to churn and take proactive steps to prevent it.

In this project, we used a variety of machine learning algorithms to build a predictive model for customer churn. We explored different feature engineering techniques, evaluated model performance, and ultimately selected the decision tree classifier model with an accuracy of 94%.

By analyzing the data and identifying the key drivers of customer churn, we were able to provide actionable insights to the business. These insights can be used to develop targeted retention strategies and improve customer satisfaction.

Overall, customer churn analysis is a powerful tool for businesses looking to improve their customer retention rates and increase their bottom line. With the help of machine learning, we can gain valuable insights into customer behavior and take proactive steps to prevent churn.

Step 3: Business Modelling



1. **Identify the problem:** The first step is to identify the problem that the business wants to solve. In this case, the problem is customer churn, or the rate at which customers stop doing business with the company.
2. **Data collection:** The next step is to collect data on customer behavior. This data may include purchase history, customer demographics, and other relevant information.
3. **Analysis:** Once the data has been collected, it is analyzed to identify trends and patterns. The analysis may involve the use of machine learning algorithms or other statistical techniques to predict which customers are at risk of churning.
4. **Intervention:** Based on the analysis, the company can implement interventions to reduce churn. This may include personalized marketing campaigns, loyalty programs, or other incentives to retain customers.
5. **Evaluation:** Finally, the effectiveness of the interventions should be evaluated using metrics such as customer retention rates, customer satisfaction scores, and other relevant measures.

Implementing the Subscription Based model based on the following steps:

1. **Identify the problem:** The problem is that customers are churning out of the subscription-based model. In this case, churn is defined as the percentage of customers who cancel their subscription during a given period.
2. **Data collection:** Subscription-based businesses typically have access to a wealth of customer data, such as purchase history, subscription length, renewal dates, and other relevant information. This data can be collected and analyzed to identify patterns in customer behavior.
3. **Analysis:** Using the collected data, businesses can analyze customer behavior to identify trends and patterns that indicate a risk of churn. For example, customers who have been subscribed for a long time but have not made a purchase in the last few months may be at risk of churning.
4. **Intervention:** Once at-risk customers have been identified, businesses can implement interventions to retain them. These may include targeted marketing campaigns, loyalty programs, or personalized offers to incentivize customers to stay subscribed.
5. **Evaluation:** Finally, the effectiveness of the interventions can be evaluated using metrics such as customer retention rates and revenue generated by retained customers.

By following these steps, businesses can improve their customer retention rates and maximize revenue from subscription-based models. This can be monetized by charging a subscription fee or taking a commission on customer transactions.

It's important to note that the specific interventions will depend on the nature of the subscription based business, the target customer demographic, and the available data. A successful subscription-based business model requires continuous monitoring and adjustment to ensure maximum customer retention and revenue generation.

Implementing the Pay-Per-Use model based on the following steps:

1. **Identify the problem:** The problem is that customers are churning out of the pay-per-use model. In this case, churn is defined as the percentage of customers who stop using the product or service during a given period.
2. **Data collection:** Pay-per-use businesses typically have access to a wealth of usage data, such as the number of times a product or service is used, the duration of usage, and other relevant information. This data can be collected and analyzed to identify patterns in customer behavior.
3. **Analysis:** Using the collected data, businesses can analyze customer behavior to identify trends and patterns that indicate a risk of churn. For example, customers who have not used the product or service in the last few weeks may be at risk of churning.
4. **Intervention:** Once at-risk customers have been identified, businesses can implement interventions to retain them. These may include personalized offers, promotional discounts, or improved customer service to encourage customers to continue using the product or service.
5. **Evaluation:** Finally, the effectiveness of the interventions can be evaluated using metrics such as customer retention rates and revenue generated by retained customers.

By following these steps, businesses can improve their customer retention rates and maximize revenue from pay-per-use models. This can be monetized by charging a fee for each usage or taking a commission on customer transactions.

It's important to note that the specific interventions will depend on the nature of the pay-per-use business, the target customer demographic, and the available data. A successful pay-per-use business model requires continuous monitoring and adjustment to ensure maximum customer retention and revenue generation.

Step 4: Financial Modelling (equation) with Machine Learning & Data Analysis:

This model can be provided to different product-based companies for prediction their customer behaviour. The market profit earned by this model can be calculated as:

1. **PROFIT EARNED BY THE MODEL:**

   Market profit = (Number of Customers * Average Revenue per Customer * Subscription Duration) - (Acquisition Cost * Number of New Customers + expenditure cost)

   Number of Customers is the total number of customers in each period
   Average Revenue per Customer is the average revenue generated per customer in a given period
   Subscription duration is the duration for which customer has taken the subscription.
   Acquisition Cost is the cost of acquiring a new customer
   Number of New Customers is the number of new customers acquired during the given period
   Expenditure cost is the cost spent in miscellaneous like salary, maintenance etc.
   Now, let us assume that number of customers be 1000, average revenue per customer be Rs. 10000 per month, acquisition cost be 3000 with new customers be 200 and expenditure cost be 2000 and let T be 1 year.

Market profit = (1000*10000* T) – (3000*200 + 2000)
Market profit= (10000000*T)- 602000
Market Profit= (10000000*1)-602000
Market profit= 9,398,000

2. PROFIT EARNED BY INDUSTRY USING MODEL:

Here, the profit earned by the industries using our customer churn model is been elaborated.
We can derive the equations for the profit calculation of the basis of two models:

Linear financial model:
If the cost of acquiring new customers is constant, the total profit for a given period can be calculated as follows:

Total profit = Total revenue - Total costs
Total revenue = Price per customer * Total number of customers
Total costs = Production costs + Maintenance costs + Cost of retaining existing customers + Cost of acquiring new customers

Assuming that the churn rate is a linear function of time, we can use the following equation to estimate the number of customers who will churn:

Churn rate = a * t + b

Where t is the time period (in months or years), a and b are coefficients that can be estimated using linear regression.

Using this churn rate, we can estimate the number of customers who will churn in the given period:

Number of churned customers = Churn rate * Total number of customers

Substituting this into the equation for total revenue and costs, we get the following equation for total profit:

Total profit = (Price per customer * Total number of customers) - (Production costs + Maintenance costs + Cost of retaining existing customers + Cost of acquiring new customers) - (Churn rate * Total number of customers * Cost of retaining existing customers)

Exponential financial model:

If the churn rate follows an exponential decay function, we can use the following equation to estimate the number of customers who will churn:

Churn rate = a * exp (-b * t)

Where t is the time period (in months or years), a and b are coefficients that can be estimated using non-linear regression.
Using this churn rate, we can estimate the number of customers who will churn in the given period:

Number of churned customers = Churn rate * Total number of customers

Substituting this into the equation for total revenue and costs, we get the following equation for total profit:

Total profit = (Price per customer * Total number of customers) - (Production costs + Maintenance costs + Cost of retaining existing customers + Cost of acquiring new customers) - (Churn rate * Total number of customers * Cost of retaining existing customers)

These equations can be used to estimate the financial impact of customer churn and to identify strategies to reduce churn and increase profits.