What Is Churn First of All:

Churn is the Number of subscribers to a service that discontinue their subscription to that service in a given time period. In order for a company to expand its Client Base, its growth rate (i.e. its number of new customers) must exceed its churn. Churn is an important consideration in the telephone and cell phone services industry.

Why Companies is Much worried about the Churn:

> Churn is used to indicate the strength of a company's customer division and its overall growth prospects.The less the Churn the more the company can make revenue out of Them.

> High Churn means the company need to again spend money to acquire new customer Base.

> Thats why companies are much worried about churn because its always difficult to acquire new customers and its mostly easy to retain them but the important quetion is how we know who will churn.

Thats where we find our Business Problem.

Business Problem:

Every Day to day passing by the competion is high in the market for the Telecom Industry and lossing the customers from its customer base gives a lot of loss to the company and other the other hand acquiring new customers is difficult and costly.The Telecom Company wants to know the customers who gonna churn and want a Model which classifies the customers which is going to churn so that the company can run measures to retain them.

Classification Description:

I will be classifying the customers based on the various features we collected from the Telecom Company and will be given output if the customer will churn or not.

Data Discription:

The data is from the IBM Watson Of Telecom Churn,Thanks to IBM providing real life senario data so that like me aspiring Data scientist can learn and perform task which can be in future replicated in Real Industry.

Attribute Information:

customerID : Customer Identification

Gender : customer is a male or a female

SeniorCitizen : customer is a senior citizen or not (1, 0)

Partner : customer a partner or not (Yes, No)

Dependents : customer dependents or not (Yes, No)

Tenure : Number of months the customer stayed with the company

PhoneService : a phone service or not (Yes, No)

MultipleLines : customer multiple lines or not (Yes, No, No phone service)

InternetService : Customer's internet service provider (DSL, Fiber optic, No)

OnlineSecurity : customer online security or not (Yes, No, No internet service)

OnlineBackup :customer online backup or not (Yes, No, No internet service)

DeviceProtection :customer device protection or not (Yes, No, No internet service)

TechSupport : customer tech support or not (Yes, No, No internet service)

StreamingTV : customer streaming TV or not (Yes, No, No internet service)

StreamingMovies :customer streaming movies or not (Yes, No, No internet service)

Contract : The contract term of the customer (Month-to-month, One year, Two year)

PaperlessBilling :the customer has paperless billing or not (Yes, No)

PaymentMethod : The customer's payment method (Electronic check, Mailed check, Bank transfer (automatic), Credit card (automatic))

MonthlyCharges : The amount charged to the customer monthly

TotalCharges : The total amount charged to the customer

Churn : Whether the customer churned or not (Yes or No)

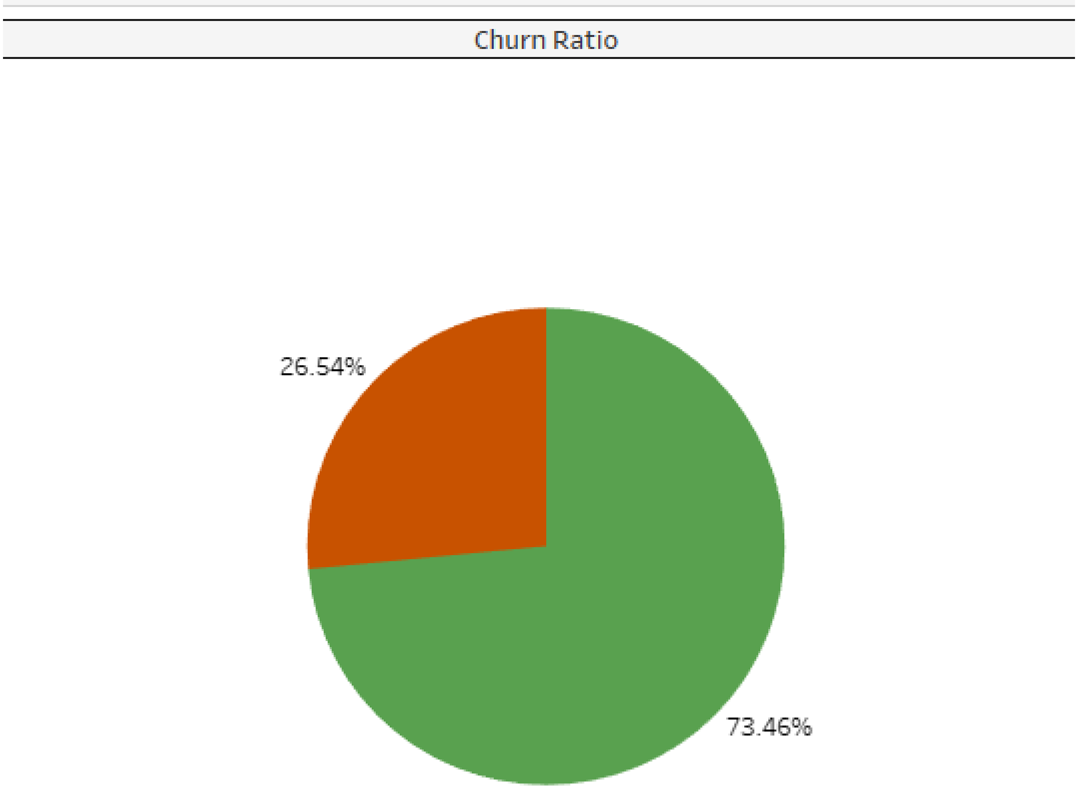
We have Succefully defined our business Problem and now we will solve the Problem Using our Business Understading First with approching the Problem Solving by Explorartory Data Analysis and then using Machine learning to Classify the Churn customers.

Explorartory Data Analysis:

This time i will doing the EDA in the Tableau,as is a very powerfull tool.i have created a dedicated dashboard regarding the same and have depicted a powerfull Data story of Telcom churn.

Tableau Dashboard Link:
https://public.tableau.com/profile/shubham.pundir#!/vizhome/TelecomChurnEDAAndInsightStory/TelecomChurnEDAAndInsightStory?publish=yes .

I will be linking the snips here for better understanding.



> Our data contains 26% of the churn people and 73% of the people who did not churn.

Insights: Contracts

In single view we will be looking at Gender ratio among the Churn people in Contract and there charging pattern.

The above figures shows The combination of Contract and the average Total ,Monthly charges with the tenure.

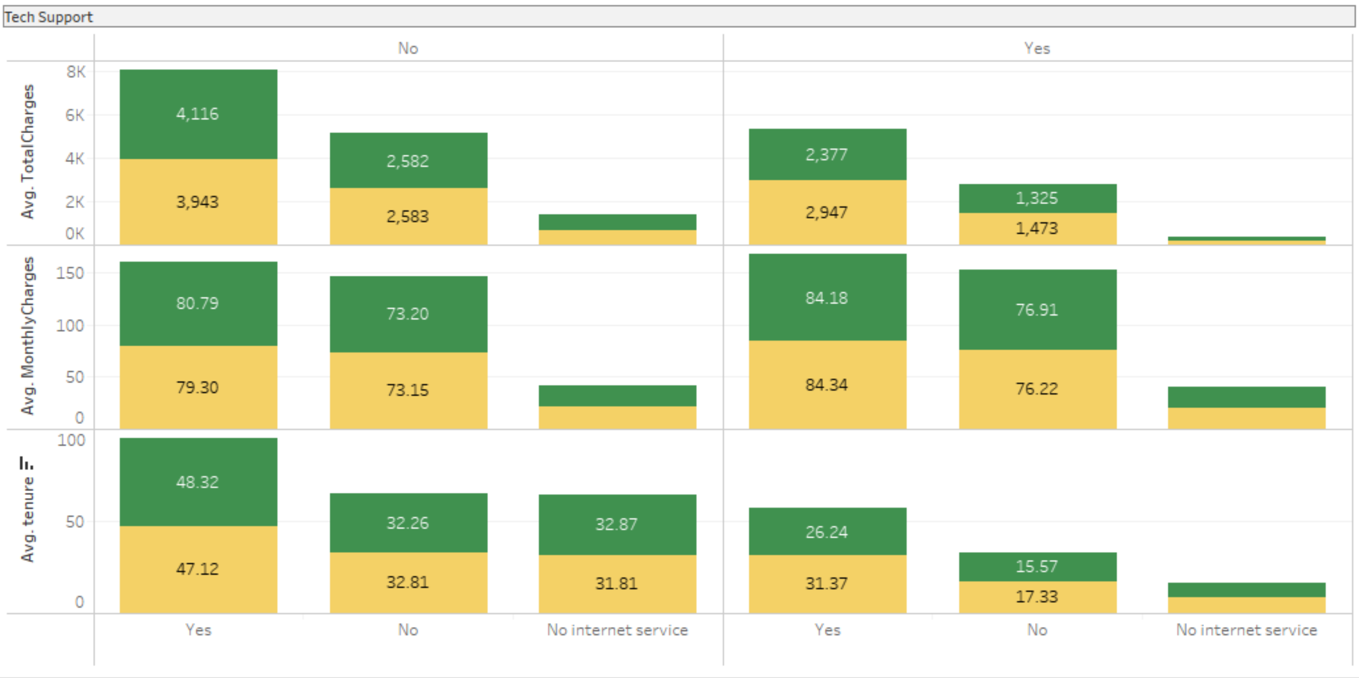
The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned yes have a greater Money generation in the two year and one year Contracts it means when these customer leave the company it generated a huge loss.

> Most of the revenue Generation is from the long Contracts ,company should concentrate more on retaining the longer contracts.



Insights: Tech Support

In single view we will be looking at Gender ratio among the Churn people in Tech Support and there charging pattern.

The above figures shows The combination of Tech Support and the average Total ,Monthly charges with the tenure.

The Green is :**Female**

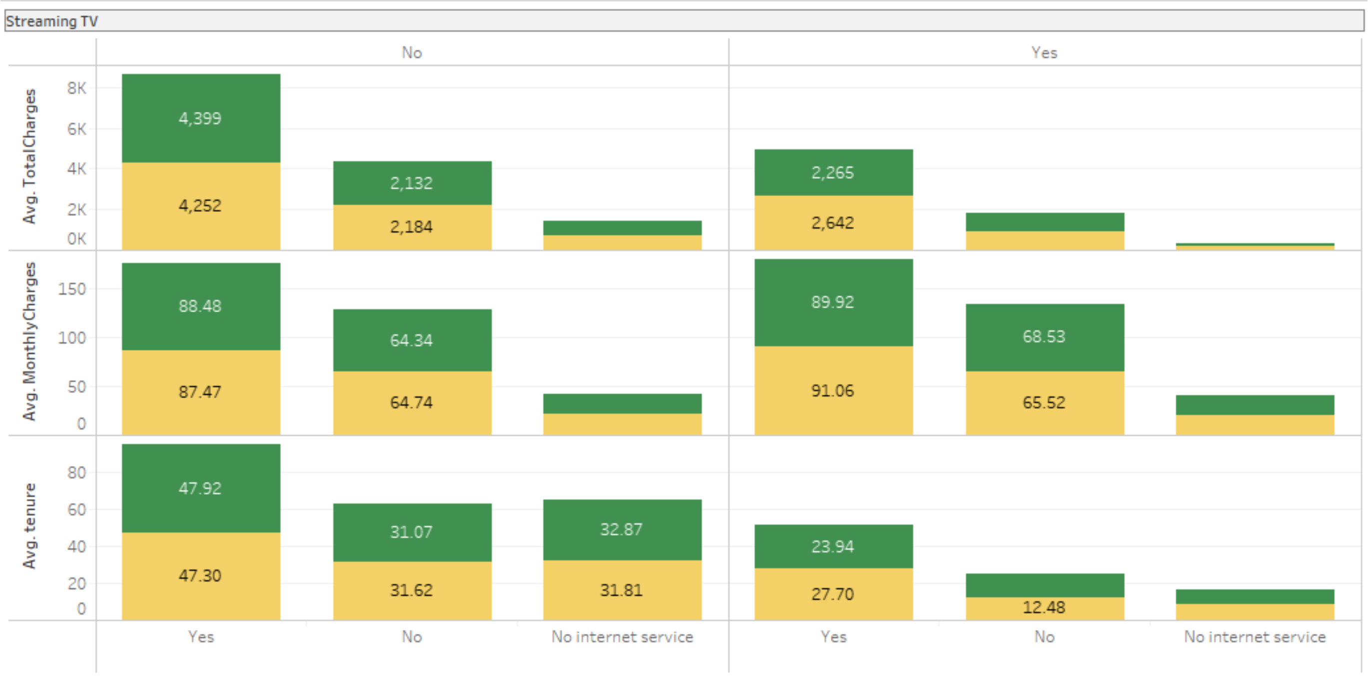
The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES have a less charging in the Tech Support ,we can conclude that most of the unavilability of Tech support the People are leaving the Company.

> We can also see that in the monthly basis the average charging is same means that people are not much satisfied with the service hence Churn YES.

> The Tenure section says it all, validates it as we can see the average tenure of the People with Tech support is less and should be taken into considersation by the company.



Insights: Streaming TV

In single view we will be looking at Gender ratio among the Churn people in Streaming TV and there charging pattern.

The above figures shows The combination of Tech Support and the average Total ,Monthly charges with the tenure.

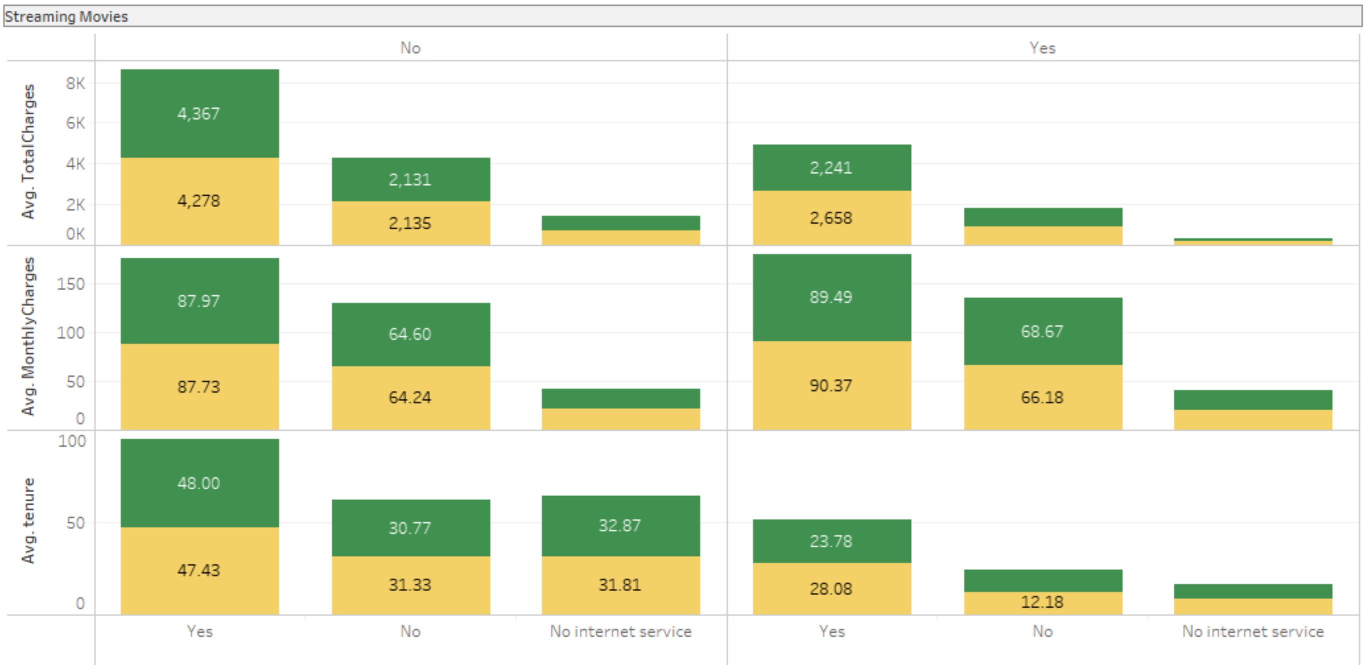
The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue still they have subscribed to the Steaming TV ,their should be more customer centric plans to increase the revenue and to retain them.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more yearly plans should come up with customer centric mindset.



Insights: Streaming Movies

In single view we will be looking at Gender ratio among the Churn people in Streaming Movies and there charging pattern.

The above figures shows The combination of Streaming Movies and the average Total ,Monthly charges with the tenure.

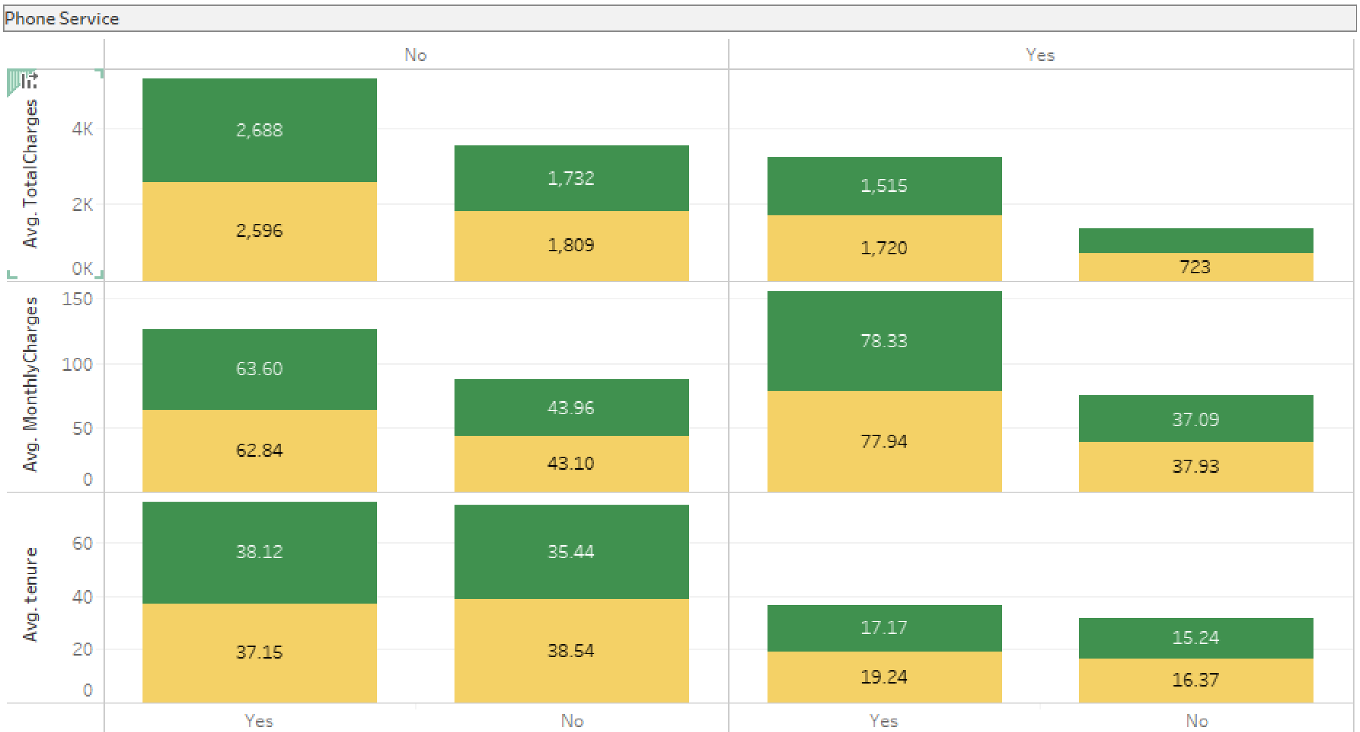
The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue still they have subscribed to the Steaming Movies ,their should be more customer centric plans to increase the revenue and retain them.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more yearly Movie plans should come up with customer centric mindset.



Insights: Phone Service

In single view we will be looking at Gender ratio among the Churn people in Phone Service and there charging pattern.

The above figures shows The combination of Phone Service and the average Total ,Monthly charges with the tenure.

The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue as there are signficantly many in the monthly charges who have left the service means there is wrong in the service provided by the company the phone service should be more customer centric.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more yearly Phone service plans should come up with customer centric mindset.

Insights: Online Security

In single view we will be looking at Gender ratio among the Churn people in Online Security and there charging pattern.

The above figures shows The combination of Online Security and the average Total ,Monthly charges with the tenure.

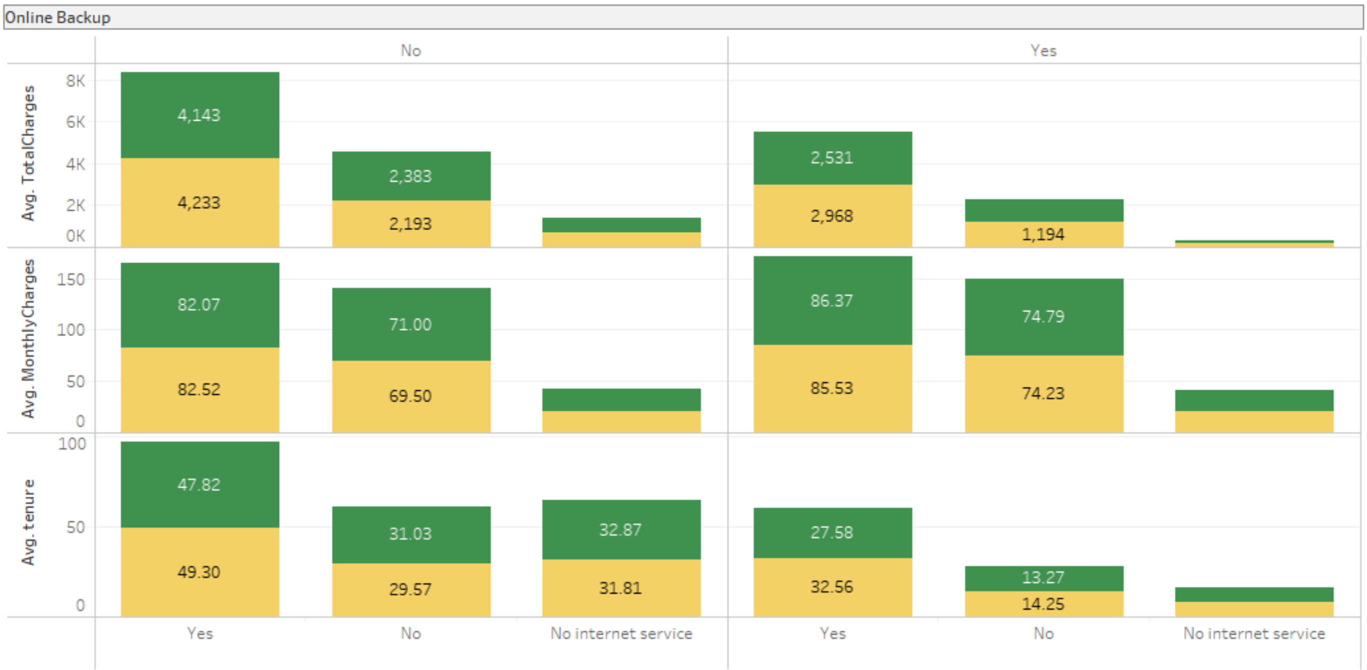
The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue we should deep dive more into the Online security measures and should one to one clear the online security problems people are facing and customers can be retained.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more Online Security Measures plans should come up with customer centric mindset.



Insights: Online Backup

In single view we will be looking at Gender ratio among the Churn people in Online Backup and there charging pattern.

The above figures shows The combination of Online Backup and the average Total ,Monthly charges with the tenure.

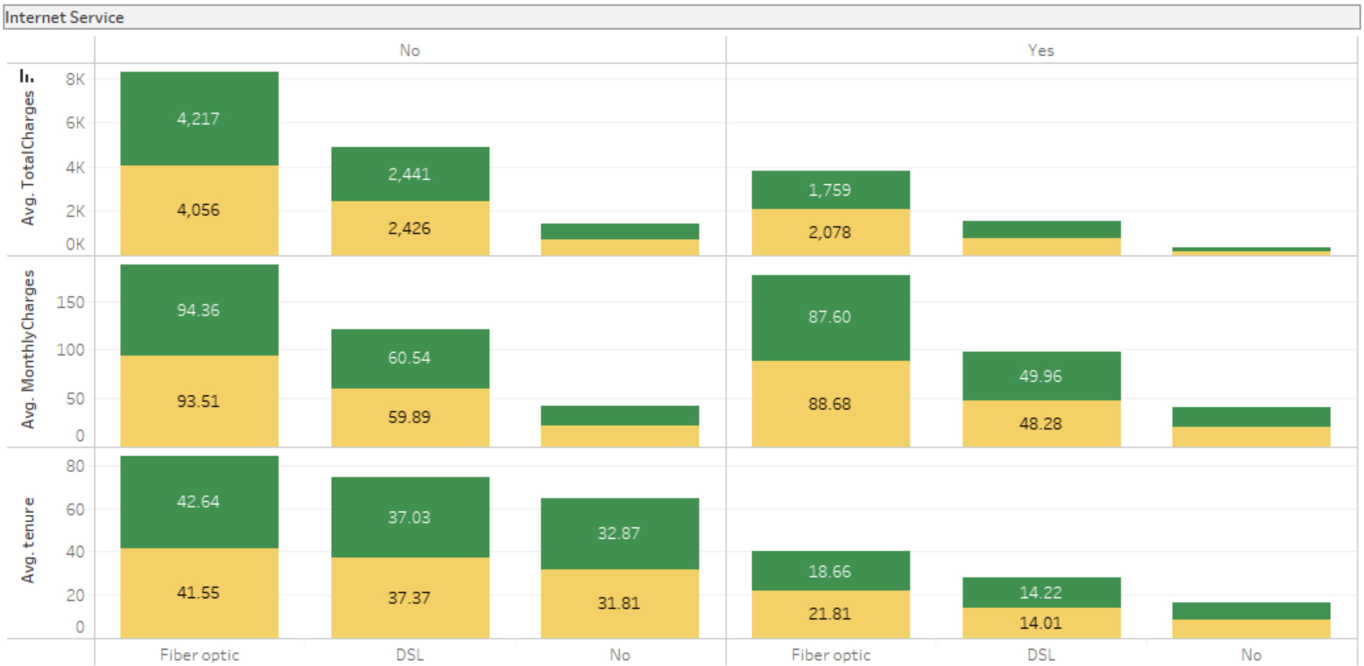
The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue we should deep dive more into the Online Backup measures and should one to one clear the online Backup problems people are facing and customers can be retained.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more Online Backup Measures plans should come up with customer centric mindset.



Insights: Internet Service

In single view we will be looking at Gender ratio among the Churn people in Internet Service and there charging pattern.

The above figures shows The combination of Internet Service and the average Total ,Monthly charges with the tenure.

The Green is :**Female**

The Gold is : **Male**

The Values in The Bars is the average Charges and tenure in months and year.

> We can see that the people who have churned YES are generating less revenue we can even see that there is less amount of people opting for Fiber optics and DSL company should come up with customer centric flexible plans to provide least Internet as the other steaming TV and movies is more dependent on those.

> The average tenure is also very less means they are not even using it for straight a year and dropping it before that,more Internet Service flexible plans should come up with customer centric mindset.

Insights: Contract pattern over all Data

In single view we will be looking at COntract pattern among the Churn people in all data and there charging pattern.

The above figures shows The combination of Contract pattern and the average Total ,Monthly charges with the tenure.

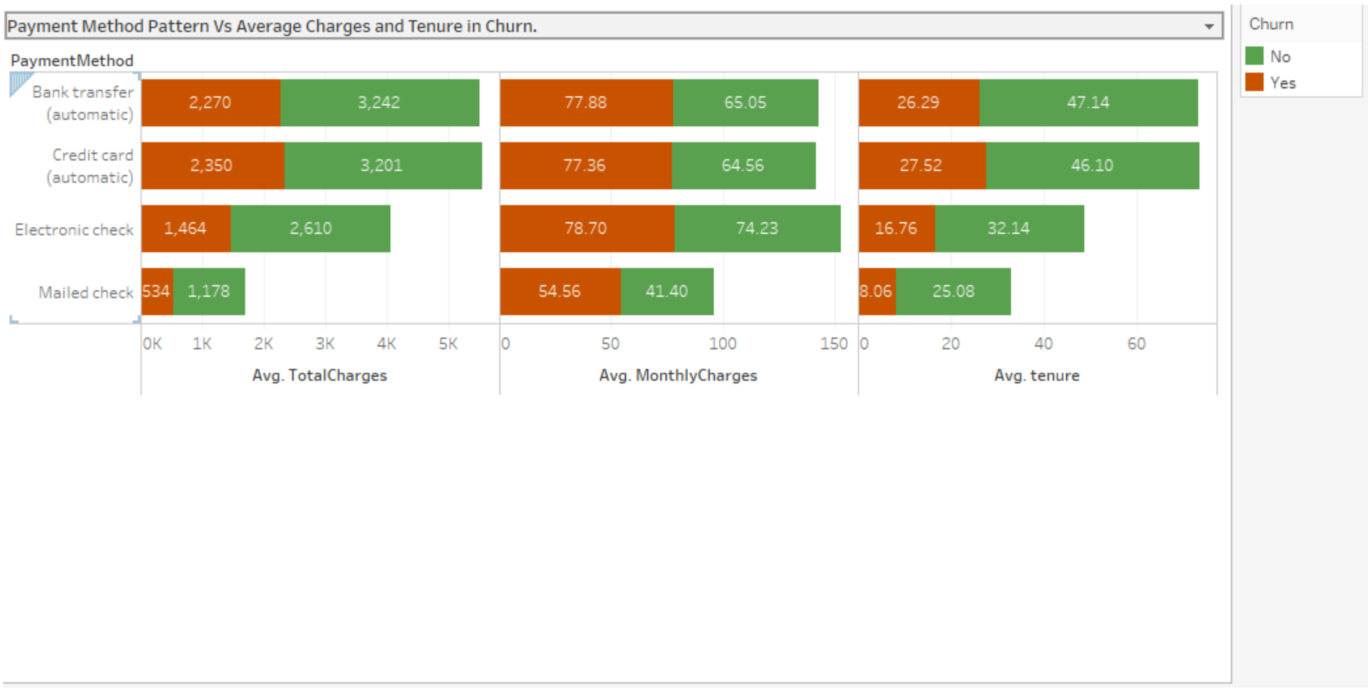
The Green is :**Churn NO**

The Red is : **Churn Yes**

The Values in The Bars is the average Charges and tenure in months and year.

> In the Total charges we can see that The Churn No is creating a huge loss for the company,but in month to month is not that much to be worried about.

> The monthly customers are not showing much of patter In the Contract,but Average tenure is worth looking into for the contract in the Company.



Insights: Payment Method pattern over all Data

In single view we will be looking at Payment Method pattern among the Churn people in all data and there charging pattern.

The above figures shows The combination of Payment Method and the average Total ,Monthly charges with the tenure.

The Green is :**Churn NO**

The Red is : **Churn Yes**

The Values in The Bars is the average Charges and tenure in months and year.

Pattern In Total Charges Among Various Paying Method:

> The people who are paying vai Bank tranfer less then 23000 are likely to be Churn YES.

> The people who are paying vai Credit Card less then 24000 are likely to be Churn YES.

> The people who are paying vai Elctronic Check less then 15000 are likely to be Churn YES.

> The people who are paying vai Mail Check less then 500 are likely to be Churn YES.

Pattern In Monthly Charges Among Various Paying Method:

> On an average who are paying less then 80 they are likely to get Churn YES.

Pattern In Tenure Among Various Paying Method:

> If a customer is with the company for less then 30 months and is paying via Bank Transfer and Credit Card they are likely to get Churn Yes.

> If a customer is with the company for less then 17 months and is paying via Electronic Check they are likely to get Churn Yes.

> If a customer is with the company for less then 9 months and is paying via Mailed Check they are likely to get Churn Yes.

Company should try to increase the tenure of the payers and move them to automatic Paying via options by giving more attractive cash back options and This will help in Less Churn Yes.

Lest kick in our Machine Learning and apply the All best XGboost and tune The model to reach our best accuracy Score(Using Confusion Matrix).

When ever the Imbalanced data set comes up to mind The XGboost performs really well.i use xgboost in imbalance data set because i dont want to opt for the Upsampling and Downsampling as it creates a baise if i upsample and loss of Valuable infomation when we do downsample.

Ther is a hyperparameter Scale_pos_weight which let the Xgboost penalise each time it classify wrong the class and it helps to reach a better accuracy other algorythms fail to.

XGboost Total understanding.

```
import numpy as np # For data manipluation
import pandas as pd # for data manipulation
import matplotlib.pyplot as plt #plot libary
%matplotlib inline
import xgboost as xgb
from sklearn.model_selection import train_test_split
from sklearn.metrics import balanced_accuracy_score,roc_auc_score,make_scorer # for scorin
from sklearn.model_selection import GridSearchCV  # for cross validation
from sklearn.metrics import confusion_matrix
from sklearn.metrics import plot_confusion_matrix
```

```
from google.colab import files
uploaded=files.upload()
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving WA_Fn-UseC_-Telco-Customer-Churn.csv to WA_Fn-UseC_-Telco-Customer-Churn.csv
```

```
!ls
```

```
sample_data  WA_Fn-UseC_-Telco-Customer-Churn.csv
```

```
df=pd.read_csv("WA_Fn-UseC_-Telco-Customer-Churn.csv")
```

```
df.head()
```

	customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	Mul
0	7590-VHVEG	Female	0	Yes	No	1	No	
1	5575-GNVDE	Male	0	No	No	34	Yes	
2	3668-QPYBK	Male	0	No	No	2	Yes	
3	7795-CFOCW	Male	0	No	No	45	No	
4	9237-HQITU	Female	0	No	No	2	Yes	

```
df.shape
```

```
(7043, 21)
```

```
df.columns
```

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

```
df["Contract"].unique()
```

```
array(['Month-to-month', 'One year', 'Two year'], dtype=object)
```

```
df["gender"].unique()
```

```
array(['Female', 'Male'], dtype=object)
```

```
df["SeniorCitizen"].unique()
```

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

```
df["Partner"].unique()
```

```
array(['Yes', 'No'], dtype=object)
```

```
df.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
```

```
df.columns
```

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

```
df.drop(["customerID"],axis=1,inplace=True)
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	Female	0	Yes	No	1	No	No phone service
1	Male	0	No	No	34	Yes	No
2	Male	0	No	No	2	Yes	No
3	Male	0	No	No	45	No	No phone service
4	Female	0	No	No	2	Yes	No

```
df.columns=df.columns.str.replace(" ","_")
```

```
df.columns[(df.isnull().any())].tolist()  # doesnt mean there is no Blank Values they can
```

```
[]
```

```
df.columns
```

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

```
df["TotalCharges"].unique()
```

```
array(['29.85', '1889.5', '108.15', ..., '346.45', '306.6', '6844.5'],
      dtype=object)
```

```
len(df.loc[df["TotalCharges"]==" "])
```

```
11
```

The total 11 places the total charges have the blank column which we were unable to see.

```
df.loc[df["TotalCharges"]==" "]  # these are the rows where we have black in total charges
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLine
488	Female	0	Yes	Yes	0	No	No phon servic
753	Male	0	No	Yes	0	Yes	N
1082	Male	0	Yes	Yes	0	Yes	Ye

Above we can see that the months of all the customer is 0 thats means that these are new customers and have not been charged anything yet so this is the reason of misisng data .

```
#lets make the charges 0

df.loc[(df["TotalCharges"]== " "),"TotalCharges"]=0
```

We will check for the tenure of months zero as there may be people who did not have paid the bills and showing 0 so we dont need them.

```
df.loc[df["tenure"]== 0]
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLine
488	Female	0	Yes	Yes	0	No	No phon servic
753	Male	0	No	Yes	0	Yes	N
936	Female	0	Yes	Yes	0	Yes	N
1082	Male	0	Yes	Yes	0	Yes	Ye
1340	Female	0	Yes	Yes	0	No	No phon servic
3331	Male	0	Yes	Yes	0	Yes	N
3826	Male	0	Yes	Yes	0	Yes	Ye
4380	Female	0	Yes	Yes	0	Yes	N
5218	Male	0	Yes	Yes	0	Yes	N
6670	Female	0	Yes	Yes	0	Yes	Ye
6754	Male	0	No	Yes	0	Yes	Ye

```
# lets change the data type to numeric as xgboost dont take objects or strings

df["TotalCharges"]=pd.to_numeric(df["TotalCharges"])
```

```
df.dtypes
```

```
gender               object
SeniorCitizen         int64
Partner              object
Dependents           object
tenure                int64
PhoneService         object
MultipleLines        object
InternetService      object
OnlineSecurity       object
OnlineBackup         object
DeviceProtection     object
TechSupport          object
StreamingTV          object
StreamingMovies      object
Contract             object
PaperlessBilling     object
PaymentMethod        object
MonthlyCharges      float64
TotalCharges        float64
Churn                object
dtype: object
```

```
df.replace(" ","_",regex=True,inplace=True)
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	Female	0	Yes	No	1	No	No_phone_service
1	Male	0	No	No	34	Yes	No
2	Male	0	No	No	2	Yes	No
3	Male	0	No	No	45	No	No_phone_service
4	Female	0	No	No	2	Yes	No

```
df["Churn"]=df["Churn"].apply(lambda x: 0 if x=="No" else 1)
df.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines

Formating Data:

X=Indipendent and

y=Dependent Variable

```
X= df.drop("Churn",axis=1).copy()
X.head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	Female	0	Yes	No	1	No	No_phone_service
1	Male	0	No	No	34	Yes	No
2	Male	0	No	No	2	Yes	No
3	Male	0	No	No	45	No	No_phone_service
4	Female	0	No	No	2	Yes	No

```
y=df["Churn"].copy()
y.head()
```

```
0    0
1    0
2    1
3    0
4    1
Name: Churn, dtype: int64
```

```
pd.get_dummies(X,columns=["PaymentMethod"],drop_first=True).head()
```

	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines
0	Female	0	Yes	No	1	No	No_phone_service
1	Male	0	No	No	34	Yes	No
2	Male	0	No	No	2	Yes	No
3	Male	0	No	No	45	No	No_phone_service
4	Female	0	No	No	2	Yes	No

```
X.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7043 entries, 0 to 7042
Data columns (total 19 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 
 16  PaymentMethod     7043 non-null   object 
 17  MonthlyCharges    7043 non-null   float64
 18  TotalCharges      7043 non-null   float64
dtypes: float64(2), int64(2), object(15)
memory usage: 1.0+ MB
```

```
X_encoded=pd.get_dummies(X,drop_first=True)
```

```
X_encoded.shape
```

```
(7043, 30)
```

```
y.unique()
```

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

```
#lets check if the data is our dependent variable is Balanced

sum(y)/len(y)
```

```
0.2653698707936959
```

Only the 26.5 % of the people actually left the company.

We will use statify the sample so that it remains same in both training and testing.

```
X_train, X_test, y_train, y_test = train_test_split(X_encoded,y,random_state=42,stratify=y
```

```
#now lets check if it got statified or not
```

```
sum(y_train)/len(y_train)
```

```
0.2654297614539947
```

```
sum(y_test)/len(y_test)
```

```
0.26519023282226006
```

its same and have been statified correctly.

```
clf_xgb=xgb.XGBClassifier(objective="binary:logistic",missing=None,seed=42)
clf_xgb.fit(X_train,
            y_train,
            verbose=True,
            early_stopping_rounds=10,
            eval_metric="aucpr",
            eval_set=[(X_test,y_test)]
            )
```

```
[0]	validation_0-aucpr:0.596821
Will train until validation_0-aucpr hasn't improved in 10 rounds.
[1]	validation_0-aucpr:0.596821
[2]	validation_0-aucpr:0.616819
[3]	validation_0-aucpr:0.622354
[4]	validation_0-aucpr:0.625807
[5]	validation_0-aucpr:0.63018
[6]	validation_0-aucpr:0.628641
[7]	validation_0-aucpr:0.630951
[8]	validation_0-aucpr:0.630587
[9]	validation_0-aucpr:0.636708
[10]	validation_0-aucpr:0.637894
[11]	validation_0-aucpr:0.639997
[12]	validation_0-aucpr:0.638768
[13]	validation_0-aucpr:0.640351
[14]	validation_0-aucpr:0.643511
[15]	validation_0-aucpr:0.642886
[16]	validation_0-aucpr:0.643331
[17]	validation_0-aucpr:0.643869
[18]	validation_0-aucpr:0.644045
[19]	validation_0-aucpr:0.644829
[20]	validation_0-aucpr:0.644679
[21]	validation_0-aucpr:0.644612
[22]	validation_0-aucpr:0.643908
[23]	validation_0-aucpr:0.643668
[24]	validation_0-aucpr:0.645144
[25]	validation_0-aucpr:0.646713
[26]	validation_0-aucpr:0.645666
[27]	validation_0-aucpr:0.646645
[28]	validation_0-aucpr:0.64798
[29]	validation_0-aucpr:0.648492
[30]	validation_0-aucpr:0.650203
[31]	validation_0-aucpr:0.650393
[32]	validation_0-aucpr:0.651138
[33]	validation_0-aucpr:0.650801
[34]	validation_0-aucpr:0.651762
[35]	validation_0-aucpr:0.652429
[36]	validation_0-aucpr:0.651653
[37]	validation_0-aucpr:0.651695
[38]	validation_0-aucpr:0.652438
[39]	validation_0-aucpr:0.652847
[40]	validation_0-aucpr:0.652991
[41]	validation_0-aucpr:0.653017
[42]	validation_0-aucpr:0.652628
```

We have trained the model now ,we have stopped the model when the auc scores are not getting better so at the 58 step we got our best model.

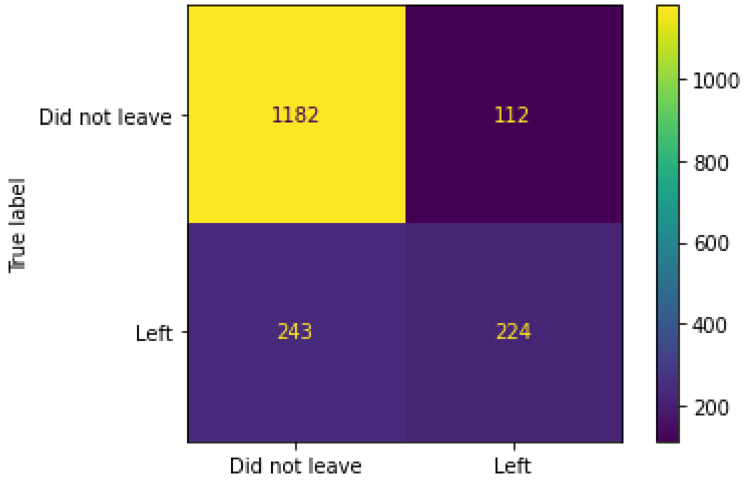
```
[47]	validation_0-aucpr:0.6552
```

Now lets see how well it performs on the Testing data:

```
[50]	validation_0-aucpr:0.655704
```

```
plot_confusion_matrix(clf_xgb,
                      X_test,
                      y_test,
                      values_format="d",
                      display_labels=["Did not leave","Left"])
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f171c17aef0>
```



Here we can se that we are only able to classify 50% of the people leaving the company and as we know this cost a company a lot so we try to optimise it

The hyperparameter **Scale_post_weight** help in when data is inbalance and act like a penalty and make model classfy the labels correctly

```
#Round 1
param_grid={
    "max_depth":[3,4,5],
    "learning_rate":[0.1,0.01,0.05],
    "gamma":[0,0.25,1],
    "reg_lambda":[0,1.0,10.0],
    'scale_pos_weight':[1,3,5]
}


grid_search=GridSearchCV(clf_xgb,param_grid=param_grid,n_jobs=-1,cv=2,scoring="accuracy")
grid_search.fit(X_train,y_train)
```

```
GridSearchCV(cv=2, error_score=nan,
             estimator=XGBClassifier(base_score=0.5, booster='gbtree',
                                     colsample_bylevel=1, colsample_bynode=1,
                                     colsample_bytree=1, gamma=0,
                                     learning_rate=0.1, max_delta_step=0,
                                     max_depth=3, min_child_weight=1,
                                     missing=None, n_estimators=100, n_jobs=1,
                                     nthread=None, objective='binary:logistic',
                                     random_state=0, reg_alpha=0, reg_lambda=1,
                                     scale_pos_weight=1, seed=42, silent=None,
                                     subsample=1, verbosity=1),
             iid='deprecated', n_jobs=-1,
             param_grid={'gamma': [0, 0.25, 1],
                         'learning_rate': [0.1, 0.01, 0.05],
                         'max_depth': [3, 4, 5], 'reg_lambda': [0, 1.0, 10.0],
                         'scale_pos_weight': [1, 3, 5]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='accuracy', verbose=0)
```

```
print(grid_search.best_params_)
```

```
{'gamma': 1, 'learning_rate': 0.05, 'max_depth': 3, 'reg_lambda': 10.0, 'scale_pos_we
```

```
#Round 1
param_grid_2={
    "max_depth":[3,4,5],
    "learning_rate":[0.1,0.01,0.05],
    "gamma":[0,0.25,1],
    "reg_lambda":[0,1.0,10.0],
    'scale_pos_weight':[1,3,5]
}


grid_search=GridSearchCV(estimator=xgb.XGBClassifier(objective="binary:logistic",
                                                     missing=None,
                                                     seed=42,
                                                     subsample=0.9,
                                                     colsample_bytree=0.5
                                                     ),param_grid=param_grid_2,n_jobs=-1,c
grid_search.fit(X_train,y_train)
```

```
GridSearchCV(cv=10, error_score=nan,
             estimator=XGBClassifier(base_score=0.5, booster='gbtree',
                                     colsample_bylevel=1, colsample_bynode=1,
                                     colsample_bytree=0.5, gamma=0,
                                     learning_rate=0.1, max_delta_step=0,
                                     max_depth=3, min_child_weight=1,
                                     missing=None, n_estimators=100, n_jobs=1,
                                     nthread=None, objective='binary:logistic',
                                     random_state=0, reg_alpha=0, reg_lambda=1,
                                     scale_pos_weight=1, seed=42, silent=None,
                                     subsample=0.9, verbosity=1),
             iid='deprecated', n_jobs=-1,
             param_grid={'gamma': [0, 0.25, 1],
                         'learning_rate': [0.1, 0.01, 0.05],
                         'max_depth': [3, 4, 5], 'reg_lambda': [0, 1.0, 10.0],
                         'scale_pos_weight': [1, 3, 5]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring='roc_auc', verbose=0)
```

```
print(grid_search.best_params_)
```

```
{'gamma': 1, 'learning_rate': 0.1, 'max_depth': 4, 'reg_lambda': 10.0, 'scale_pos_wei
```

Now as we know the best hyperparameters lets put them and train our model.

```
clf_xgb=xgb.XGBClassifier(objective="binary:logistic",missing=None,seed=42,
                          gamma=1,
                          learning_rate=0.1,
                          max_depth=4,
                          reg_lambda=10,
                          scale_pos_weight=5
```

```
                             )
```

```
clf_xgb.fit(X_train,
            y_train,
            verbose=True,
            early_stopping_rounds=10,
            eval_metric="aucpr",
            eval_set=[(X_test,y_test)]
            )
```

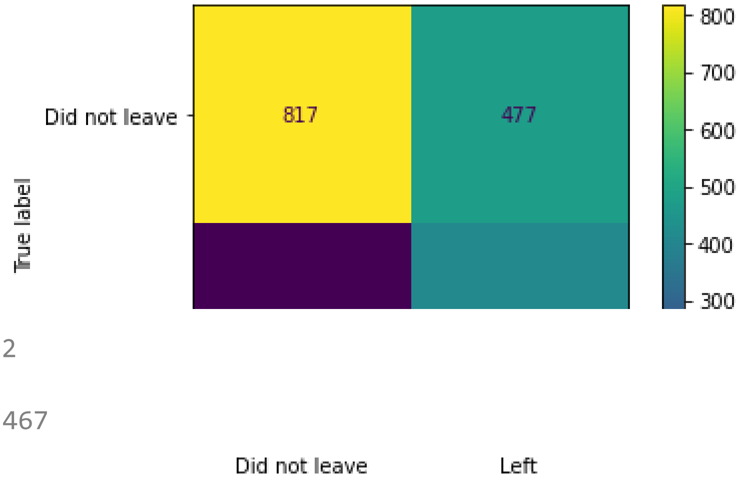
```
[0]     validation_0-aucpr:0.574162
Will train until validation_0-aucpr hasn't improved in 10 rounds.
[1]     validation_0-aucpr:0.578479
[2]     validation_0-aucpr:0.579897
[3]     validation_0-aucpr:0.59184
[4]     validation_0-aucpr:0.591796
[5]     validation_0-aucpr:0.587404
[6]     validation_0-aucpr:0.5927
[7]     validation_0-aucpr:0.592849
[8]     validation_0-aucpr:0.603349
[9]     validation_0-aucpr:0.609321
[10]    validation_0-aucpr:0.610095
[11]    validation_0-aucpr:0.610061
[12]    validation_0-aucpr:0.616711
[13]    validation_0-aucpr:0.616246
[14]    validation_0-aucpr:0.616892
[15]    validation_0-aucpr:0.617999
[16]    validation_0-aucpr:0.633337
[17]    validation_0-aucpr:0.633142
[18]    validation_0-aucpr:0.633891
[19]    validation_0-aucpr:0.635327
[20]    validation_0-aucpr:0.634329
[21]    validation_0-aucpr:0.636379
[22]    validation_0-aucpr:0.636442
[23]    validation_0-aucpr:0.637724
[24]    validation_0-aucpr:0.642557
[25]    validation_0-aucpr:0.64244
[26]    validation_0-aucpr:0.641888
[27]    validation_0-aucpr:0.641636
[28]    validation_0-aucpr:0.643076
[29]    validation_0-aucpr:0.642962
[30]    validation_0-aucpr:0.640516
[31]    validation_0-aucpr:0.644987
[32]    validation_0-aucpr:0.645373
[33]    validation_0-aucpr:0.645845
[34]    validation_0-aucpr:0.645468
[35]    validation_0-aucpr:0.645989
[36]    validation_0-aucpr:0.64689
[37]    validation_0-aucpr:0.647634
[38]    validation_0-aucpr:0.648718
[39]    validation_0-aucpr:0.648079
[40]    validation_0-aucpr:0.648275
[41]    validation_0-aucpr:0.647636
[42]    validation_0-aucpr:0.648527
```

Lets see the confusion matrix now if we have impoved .

```
[45]    validation_0-aucpr:0.649258
```

```
plot_confusion_matrix(clf_xgb,
                      X_test,
                      y_test,
                      values_format="d",
                      display_labels=["Did not leave","Left"])
```

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7f171a947898>
```




```
405+62
```

```
467
```

```
414/467
```

```
0.8865096359743041
```

We were Succefully able to classify 86% correctly the Churn Yes customers.

```
learning_rate=0.1, max_delta_step=0, max_depth=4,
```

```
851+443
```

```
1294
```

```
851/1294
```

```
0.6576506955177743
```

```
node_param={"shape":"box",
            "style":"filled"
}
```

Lets plot the first decession Tree to have a idea how the fucntionalty is Happening.

```
xgb.to_graphviz(clf_xgb,num_trees=0)
```