# Assignment – 2 :

**You will have to build a logistic regression model and interpret the result. Make sure you partition the data set by allocating 70% -for training data and 30% -for validating the results.**

**Validation steps should be clear and also interpret in terms of business consequence as well.**

**Data set: Data set -Churn modelling**

## Importing the Packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.linear\_model import LogisticRegression

from sklearn import preprocessing

import warnings

warnings.filterwarnings("ignore")

## Importing the data

xls = pd.ExcelFile('Dataset-Churn modelling.xlsx')

df1 = pd.read\_excel(xls, 'Description')

data = pd.read\_excel(xls, 'Data')

## Checking the Null values

data.isnull().sum()

OUT[5]:

Churn 0

AccountWeeks 0

ContractRenewal 0

DataPlan 0

DataUsage 0

CustServCalls 0

DayMins 0

DayCalls 0

MonthlyCharge 0

OverageFee 0

RoamMins 0

dtype: int64

There are No null values in the dataset

## Let's get a sense of the numbers across the two classes

data.groupby('Churn').mean()

### Observations:

1.The average Datausage of customers who Cancelled the service is less than who don't

2.The AVerage CustServCalls are more for the service cancelled customers

3.The Average Monthly bill is more for cancelled service customers

4.The OverageFee is more for the Cancelled customers

## Applying the VIF factor for all the numerical variables in the data

from statsmodels.stats.outliers\_influence import variance\_inflation\_factor

def VIFfunc(x):

data\_mat = x.as\_matrix()

vif = [variance\_inflation\_factor(data\_mat,i) for i in range(data\_mat.shape[1])]

vif\_factors = pd.DataFrame()

vif\_factors['columns'] = x.columns

vif\_factors['vif'] = vif

return vif\_factors

VIFfunc(X)

## Dropped these columns because having high VIF vale ['OverageFee','MonthlyCharge','DayMins','DataUsage', 'DayCalls',]

# Applied Logit Function to the data to see the Variables having p > 0.05 (Insignificant variables)

import statsmodels.api as sm

logit\_model=sm.Logit(Y,X)

result=logit\_model.fit()

print(result.summary())

## 'RoamMins' got more Pvalue as 0.5(greater than 0.05) so its dropped from the data

# Spliting the data in the ratio 70:30

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,Y, test\_size=0.3, random\_state=0)

# And Fitting the Logistic Regression

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

logreg = LogisticRegression()

logreg.fit(X\_train, y\_train)

# Predicting the test set results and caculating the accuracy

y\_pred = logreg.predict(X\_test)

print('Accuracy of logistic regression classifier on test set: {:.2f}'.format(logreg.score(X\_test, y\_test)))

And got Accuracy of logistic regression classifier on test set: 0.86

# Confusion Matrix

from sklearn.metrics import confusion\_matrix

confusion\_matrix = confusion\_matrix(y\_test, y\_pred)

print(confusion\_matrix)

[[843 19]

[122 16]]

from sklearn.metrics import classification\_report

print(classification\_report(y\_test, y\_pred))

precision recall f1-score support

0.87 0.98 0.92 862

0.46 0.12 0.18 138

from sklearn.metrics import roc\_auc\_score

from sklearn.metrics import roc\_curve

logit\_roc\_auc = roc\_auc\_score(y\_test, logreg.predict(X\_test))

fpr, tpr, thresholds = roc\_curve(y\_test, logreg.predict\_proba(X\_test)[:,1])

plt.figure()

plt.plot(fpr, tpr, label='Logistic Regression (area = %0.2f)' % logit\_roc\_auc)

plt.plot([0, 1], [0, 1],'r--')

plt.xlim([0.0, 1.0])

plt.ylim([0.0, 1.05])

plt.xlabel('False Positive Rate')

plt.ylabel('True Positive Rate')

plt.title('Receiver operating characteristic')

plt.legend(loc="lower right")

plt.savefig('Log\_ROC')

plt.show()

## Area Under the curve got 0.55

# **Validation steps should be clear and also interpret in terms of business consequence as well.**

'AccountWeeks', 'ContractRenewal', 'DataPlan', 'CustServCalls'

AccountWeeks number of weeks customer has had active account

ContractRenewal 1 if customer recently renewed contract, 0 if not

DataPlan 1 if customer has data plan, 0 if not

CustServCalls number of calls into customer service

The model fitted on only 4 variables ( 'AccountWeeks', 'ContractRenewal', 'DataPlan', 'CustServCalls')

These 4 variables are giving more information about the Churn of the customer.