Anubhav Chaturvedi

Globsyn Knowledge Foundation
Kolkata

Document sign date :Aug 6, 2018

globsyn

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

globsyn
finishing school

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Predicting Customer Churn

Group Members:

Karan Patadia, The Heritage Academy, 162131010047

Shreerup Sharma , The Heritage Academy, 162131010103

Rupam Aich , The Heritage Academy, 162131010080

Deepak Kumar Rajak, The Heritage Academy, 162131010030

Parwez Alam , The Heritage Academy, 162131010059

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Table of Contents

Acknowledgement

I take this opportunity to express my profound gratitude and deep regards to my faculty (Anubhav Chaturvedi) for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him/her time to time shall carry me a long way in the journey of life on which I am about to embark.

I am obliged to my project team members for the valuable information provided by them in their respective fields. I am grateful for their cooperation during the period of my assignment.

Karan Patadia
Parwez Alam
Shreerup Sharma
Rupam Aich
Deepak Kumar Rajak

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Project Objective

- **The Business Problem:** Predicting Churn at a Telecom Service Provider
- **Project Objective:** Our objective is to understand which of the factors contribute most to customer churn and to predict which customers will potentially churn based on service-related factors.
- **About the Dataset**
  - It consists of information for 5,000 customers and includes independent variables such as account length, number of voicemail messages, total daytime charge, total evening charge, total night charge, total international charge, and number of customer service calls. The dependent variable in the dataset is whether the customer churned or not, which is indicated by a 1 for "yes" and 0 for "no."

Anubhav Chaturvedi

Document sign date :Aug 6, 2018

Project Scope

Following is the description of all column of the dataset:

- It consists of information for 5,000 customers
- account_length-account length
- number_vmail_messages-number of voicemail messages
- total_day_charge-total daytime charge
- total_eve_charge-total evening charge
- total_night_charge-total night charge
- total_intl_charge-total international charge
- number_customer_service_calls-number of customer service calls.
- churn-The dependent variable in the dataset is whether the customer churned or not, which is indicated by a 1 for "yes" and 0 for "no."

Document sign date :Aug 6, 2018

Data description

	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls	churn
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	7.755200	30.649668	17.054322	9.017732	2.771196	1.570400	0.141400
std	13.546393	9.162069	4.296843	2.273763	0.745514	1.306363	0.348469
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.000000	24.430000	14.140000	7.510000	2.300000	1.000000	0.000000
50%	0.000000	30.620000	17.090000	9.020000	2.780000	1.000000	0.000000
75%	17.000000	36.750000	19.900000	10.560000	3.240000	2.000000	0.000000
max	52.000000	59.760000	30.910000	17.770000	5.400000	9.000000	1.000000

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Model Building

Logistic Regression

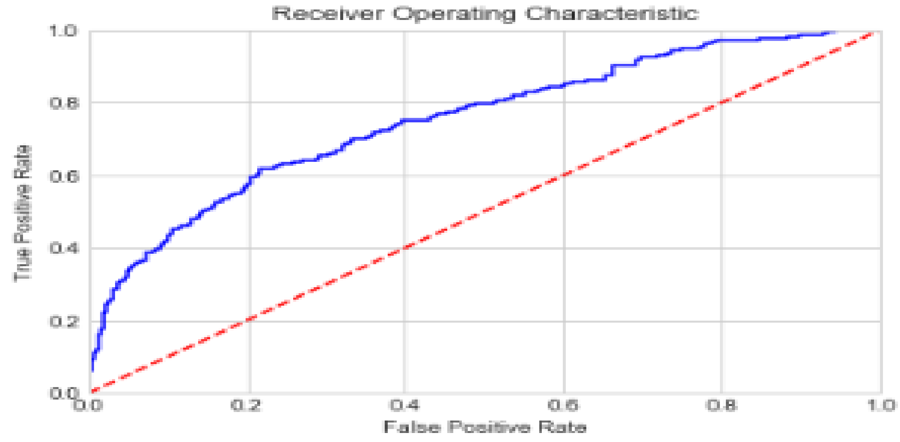
- ✓ Logistic regression is used to describe data and to explain the relationship between one dependent binary variable and one or more nominal, ordinal, interval or ratio-level independent variables.

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

```
             precision    recall  f1-score   support

          0       0.87      1.00      0.93      1297
          1       0.76      0.06      0.12       203

avg / total       0.86      0.87      0.82      1500
```

ROC Curve

	Predicted False	Predicted True
Actual False	1293	4
Actual True	190	13



Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Model Building

KNN

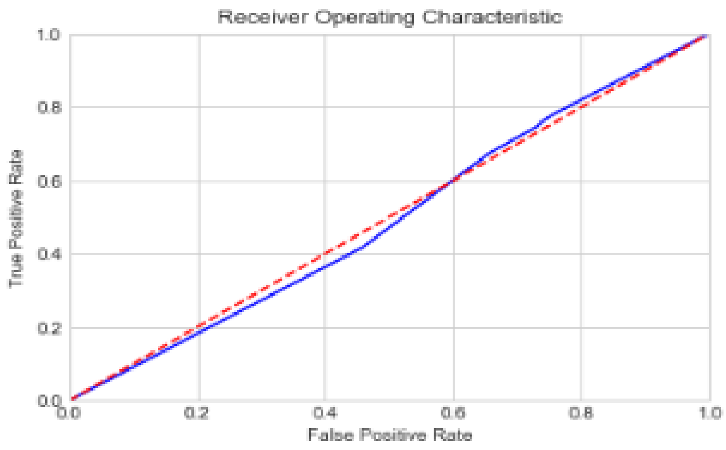
- ✓ K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions).

```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=1, p=2,
           weights='uniform')
```

```
WITH K=8


[[1282    5]
 [ 134   79]]


             precision    recall  f1-score   support

          0       0.91      1.00      0.95      1287
          1       0.94      0.37      0.53       213

avg / total       0.91      0.91      0.89      1500

Accuracy of KNN classifier :  0.9073333333333333
```

ROC Curve

	Predicted False	Predicted True
Actual False	1294	2
Actual True	114	90



Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Model Building

Naïve Bayes

✓ A naive Bayes classifier is an algorithm that uses Bayes' theorem to classify objects. Naive Bayes classifiers assume strong, or naive, independence between attributes of data points.

For Account Length

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1	0.00	0.00	0.00	616
avg / total	0.73	0.86	0.79	4293

number_vmail_messages

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1	0.00	0.00	0.00	616
avg / total	0.73	0.86	0.79	4293

total_day_charge

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1	0.06	0.00	0.00	616
avg / total	0.74	0.85	0.79	4293

total_eve_charge

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1				616
avg / total				4293

total_night_charge

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1	0.00	0.00	0.00	616
avg / total	0.73	0.86	0.79	4293

total_intl_charge

	precision	recall	f1-score	support
0	0.86	1.00	0.92	3677
1	0.00	0.00	0.00	616
avg / total	0.73	0.86	0.79	4293

number_customer_service_calls

	precision	recall	f1-score	support
0	0.86	0.99	0.92	3677
1	0.14	0.01	0.02	616
avg / total	0.75	0.85	0.79	4293

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Code

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import accuracy_score
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_curve
import matplotlib
import matplotlib.pyplot as plt
from IPython.display import display, HTML
```

Preparing the Data

In [2]:
```
ch=pd.read_csv("churn.csv")
ch1=pd.read_csv("churn.csv")
ch.head()
```

Out[2]:

	account_length	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls	churn
0	128	25	45.07	16.78	11.01	2.70	1	0
1	107	26	27.47	16.62	11.45	3.70	1	0
2	137	0	41.38	10.30	7.32	3.29	0	0
3	84	0	50.90	5.26	8.86	1.78	2	0
4	75	0	28.34	12.61	8.41	2.73	3	0

```
In [3]: ch.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5000 entries, 0 to 4999
Data columns (total 8 columns):
account_length                  5000 non-null int64
number_vmail_messages           5000 non-null int64
total_day_charge                5000 non-null float64
total_eve_charge                5000 non-null float64
total_night_charge              5000 non-null float64
total_intl_charge               5000 non-null float64
number_customer_service_calls   5000 non-null int64
churn                           5000 non-null int64
dtypes: float64(4), int64(4)
memory usage: 312.6 KB
```

```
In [4]: ch.describe()
```

Out[4]:

	account_length	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls	churn
count	5000.00000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000	5000.000000
mean	100.25860	7.755200	30.649668	17.054322	9.017732	2.771196	1.570400	0.141400
std	39.69456	13.546393	9.162069	4.296843	2.273763	0.745514	1.306363	0.348469
min	1.00000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	73.00000	0.000000	24.430000	14.140000	7.510000	2.300000	1.000000	0.000000
50%	100.00000	0.000000	30.620000	17.090000	9.020000	2.780000	1.000000	0.000000
75%	127.00000	17.000000	36.750000	19.900000	10.560000	3.240000	2.000000	0.000000
max	243.00000	[illegible]	[illegible]	[illegible]	17.770000	5.400000	9.000000	1.000000

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

EDA

In [5]:
```
#Distribution of churn
print('Distribution of churn : ')
print(ch['churn'].value_counts())
print('\n\nProportion for Distribution of churn : ')
print(ch['churn'].value_counts(normalize=True))
```

```
Distribution of churn :
0    4293
1     707
Name: churn, dtype: int64


Proportion for Distribution of churn :
0    0.8586
1    0.1414
Name: churn, dtype: float64
```

In [6]:
```
#what is the proportion of churned users in our dataframe?
ch['churn'].mean()
```

Out[6]:
```
0.1414
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [7]:
```
#What are average values of numerical variables for churned users?
ch[ch['churn'] == 1].mean()
```

Out[7]:
```
account_length                102.332390
number_vmail_messages           4.496464
total_day_charge               35.338416
total_eve_charge               17.999562
total_night_charge              9.273607
total_intl_charge               2.887426
number_customer_service_calls   2.254597
churn                           1.000000
dtype: float64
```

In [8]:
```
#maximum day,evening,night,international charges who are loyal
print("Maximum Day charge : ",ch[(ch['churn'] == 0)]['total_day_charge'].max())
print("Maximum Evening charge : ",ch[(ch['churn'] == 0)]['total_eve_charge'].max())
print("Maximum Night charge : ",ch[(ch['churn'] == 0)]['total_night_charge'].max())
print("Maximum International charge : ",ch[(ch['churn'] == 0)]['total_intl_charge'].max())
```

```
Maximum Day charge :  53.65
Maximum Evening charge :  30.75
Maximum Night charge :  17.77
Maximum Intern
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [9]:
```
#grouping the data according to the values of the Churn variable and display statistics of all columns in each group:
col= ['number_vmail_messages','total_day_charge','total_eve_charge','total_night_charge','total_intl_charge','number_customer_service_calls']
ch.groupby(['churn'])[col].describe(percentiles=[])
```

Out[9]:

	number_vmail_messages						total_day_charge				...	total_intl_charge				number_customer_service_calls					
	count	mean	std	min	50%	max	count	mean	std	min	...	std	min	50%	max	count	mean	std	min	50%	max
churn																					
0	4293.0	8.291870	13.809408	0.0	0.0	52.0	4293.0	29.877494	8.437810	0.0	...	0.742443	0.0	2.78	5.32	4293.0	1.457722	1.164236	0.0	1.0	8.0
1	707.0	4.496464	11.297719	0.0	0.0	48.0	707.0	35.338416	11.658195	0.0	...	0.754057	0.0	2.86	5.40	707.0	2.254597	1.815956	0.0	2.0	9.0

2 rows × 36 columns

In [10]:
```
#lets see how number of customer service calls effect churn
pd.crosstab(ch['churn'], ch['number_customer_service_calls'], margins=True)
```

Out[10]:

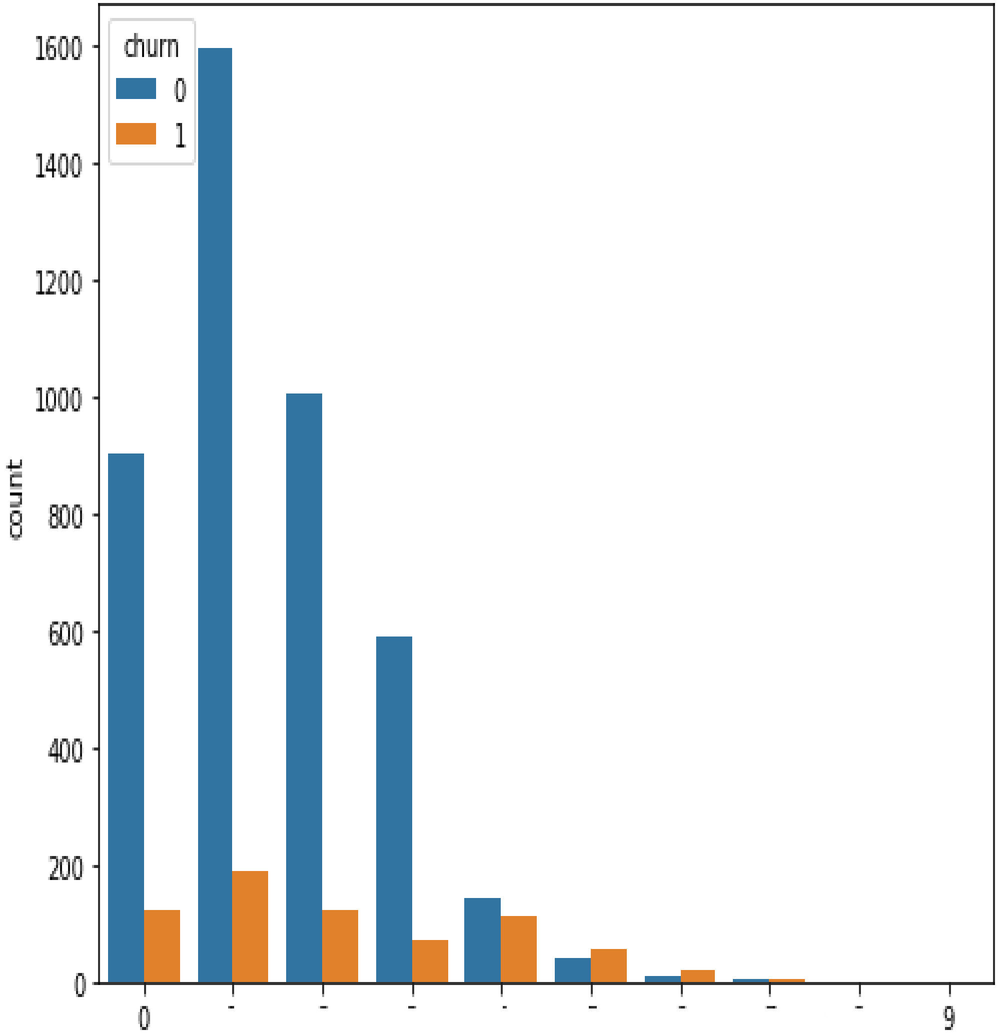
number_customer_service_calls	0	1	2	3	4	5	6	7	8	9	All
churn											
0	902	1596	1005	592	141	38	12	6	1	0	4293
1	121	190	122	73	111	58	22	7	1	2	707
									2	2	5000

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

```
In [11]: plt.rcParams['figure.figsize'] = (8, 6)
         sns.countplot(x='number_customer_service_calls', hue='churn', data=ch);
```

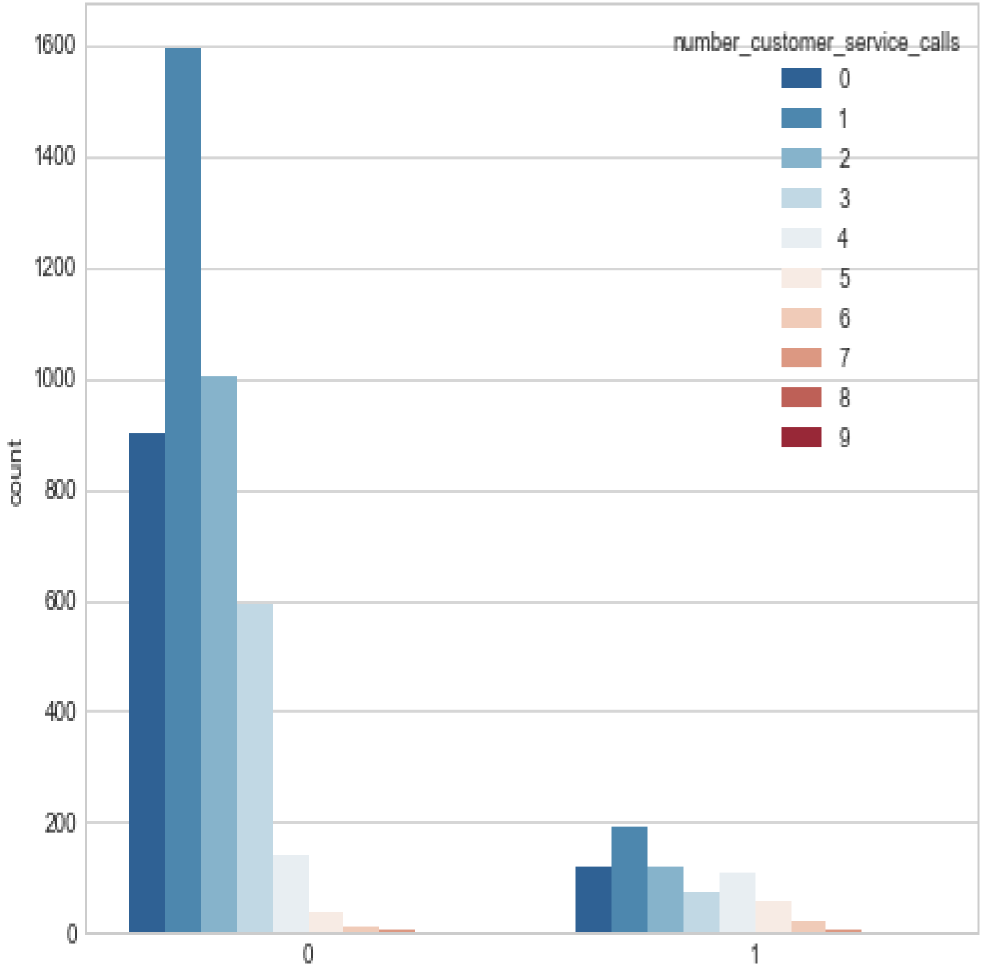


Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

In [12]:
```
sns.set_style('whitegrid')
sns.countplot(x='churn',hue='number_customer_service_calls',data=ch,palette='RdBu_r')
```

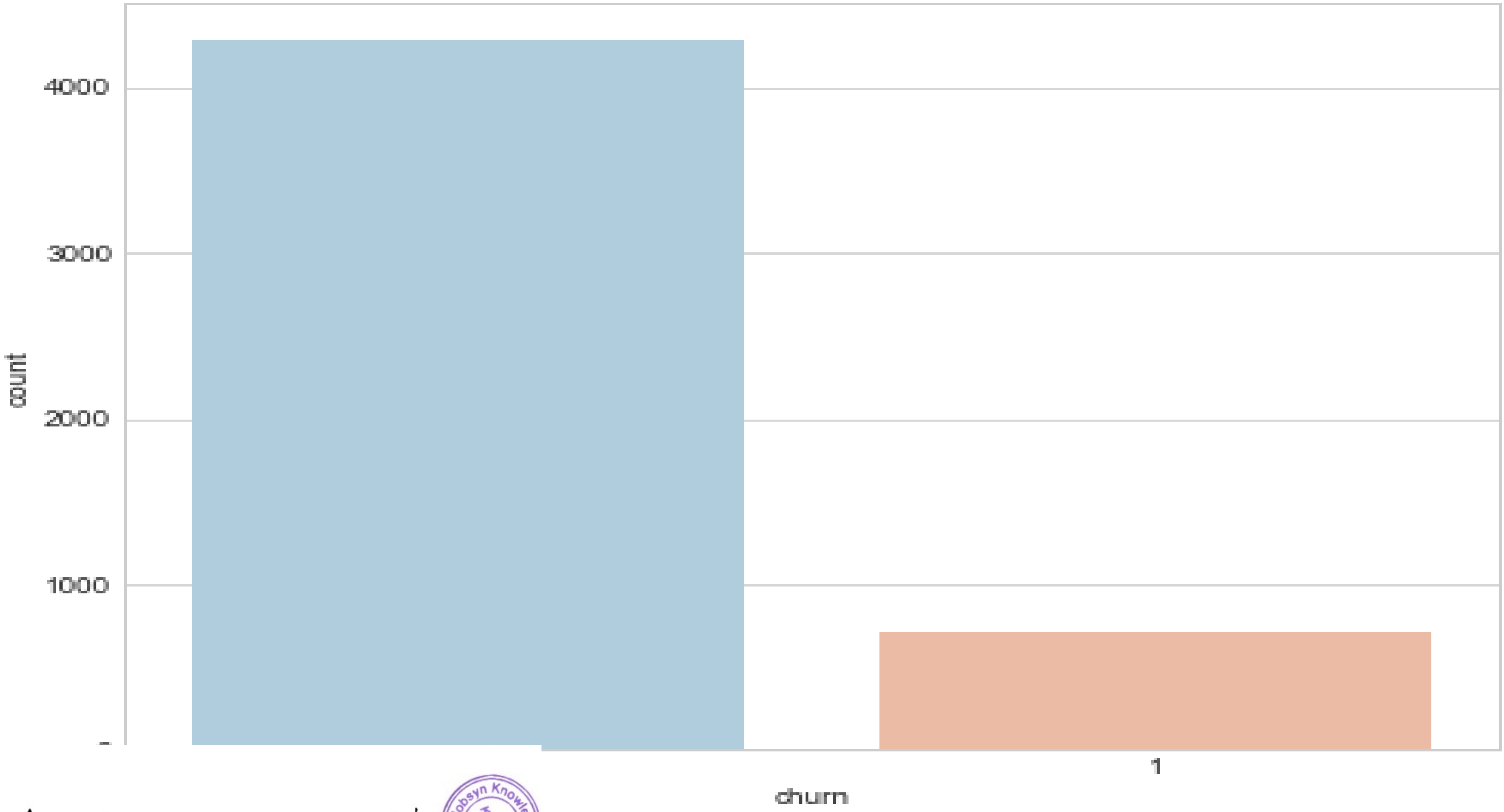
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0xef071c07b8>



Document sign date :Aug 6, 2018

In [13]:
```
sns.set_style('whitegrid')
sns.countplot(x='churn',data=ch,palette='RdBu_r')
```

Out[13]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xef077ceb70>
```



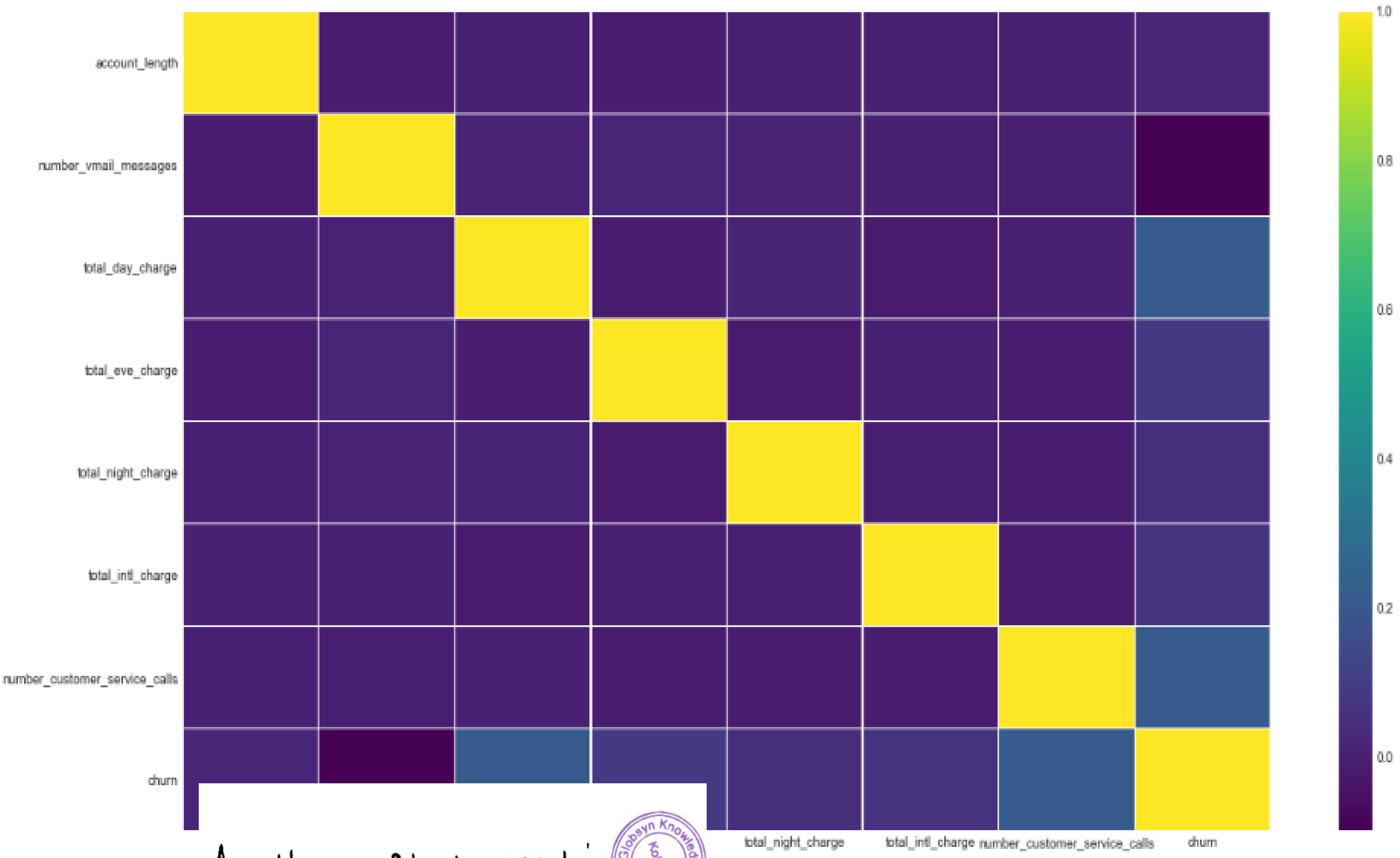
Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [18]:
```
plt.figure(figsize=(20,10))
sns.heatmap(ch.corr(),cmap="viridis",linewidth=0.3)
```

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0xef08113668>



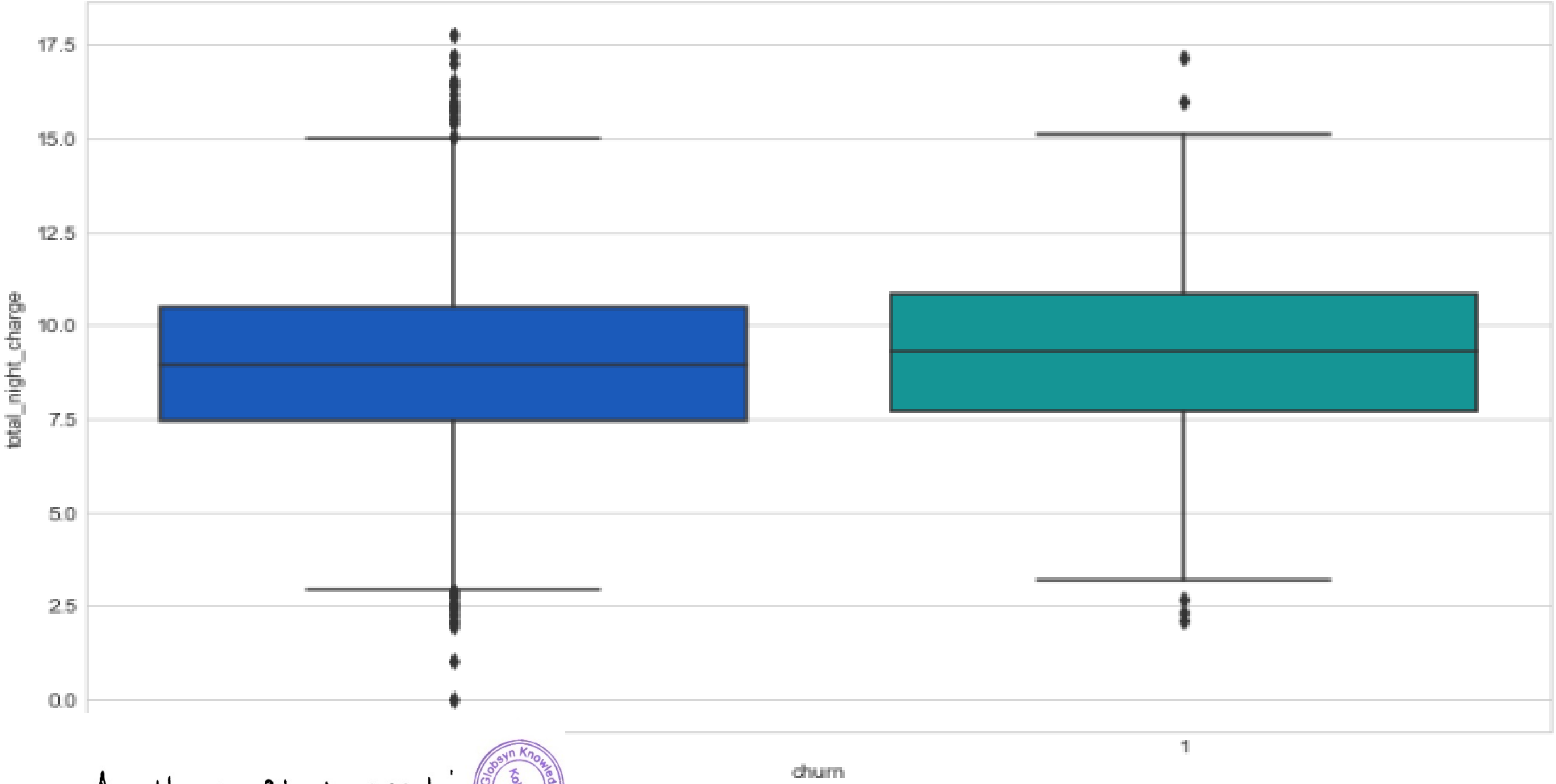
Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [19]:
```
plt.figure(figsize=(12,7))
sns.boxplot(x='churn',y='total_night_charge',data=ch,palette='winter')
```

Out[19]: <matplotlib.axes._subplots.AxesSubplot at 0xef085b5be0>



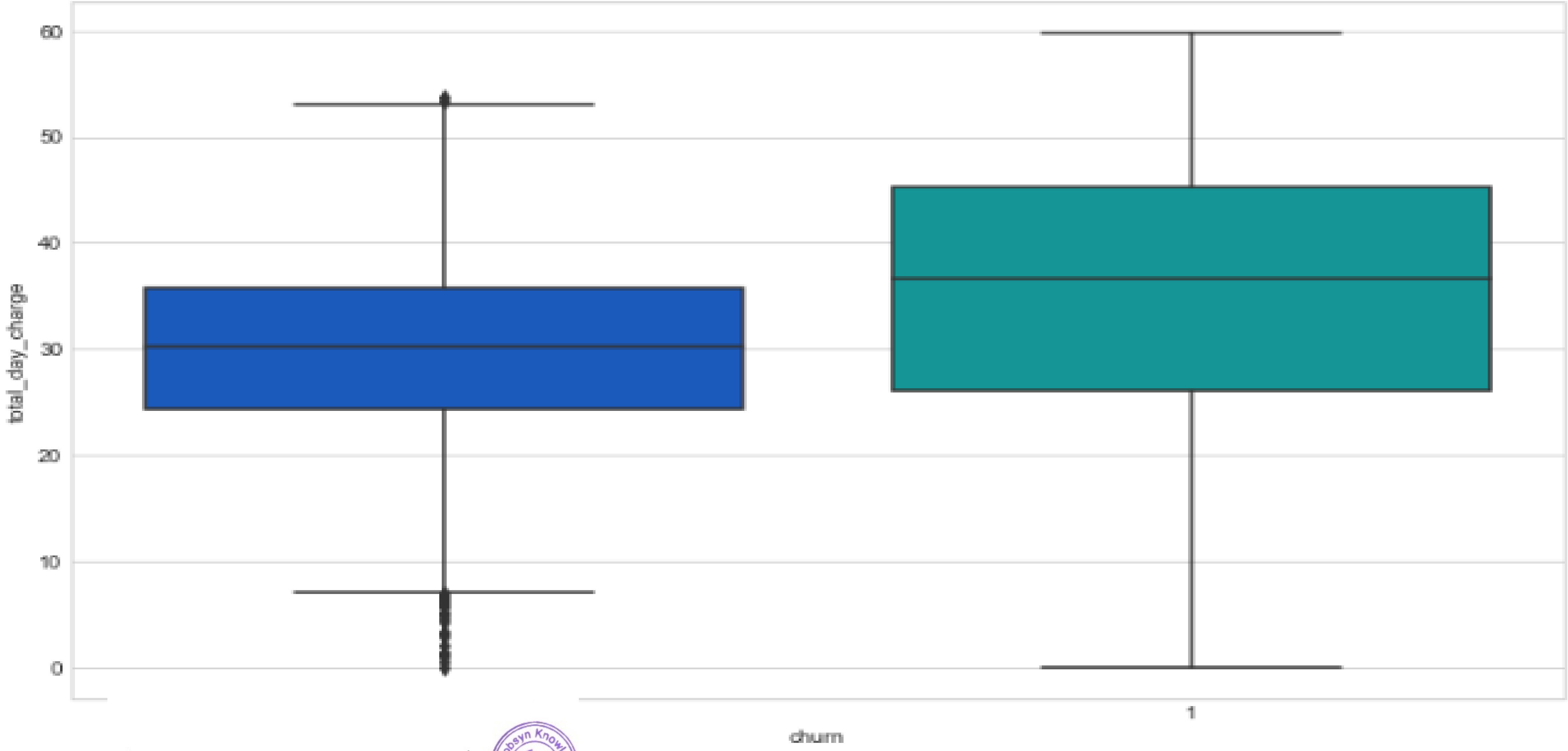
Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

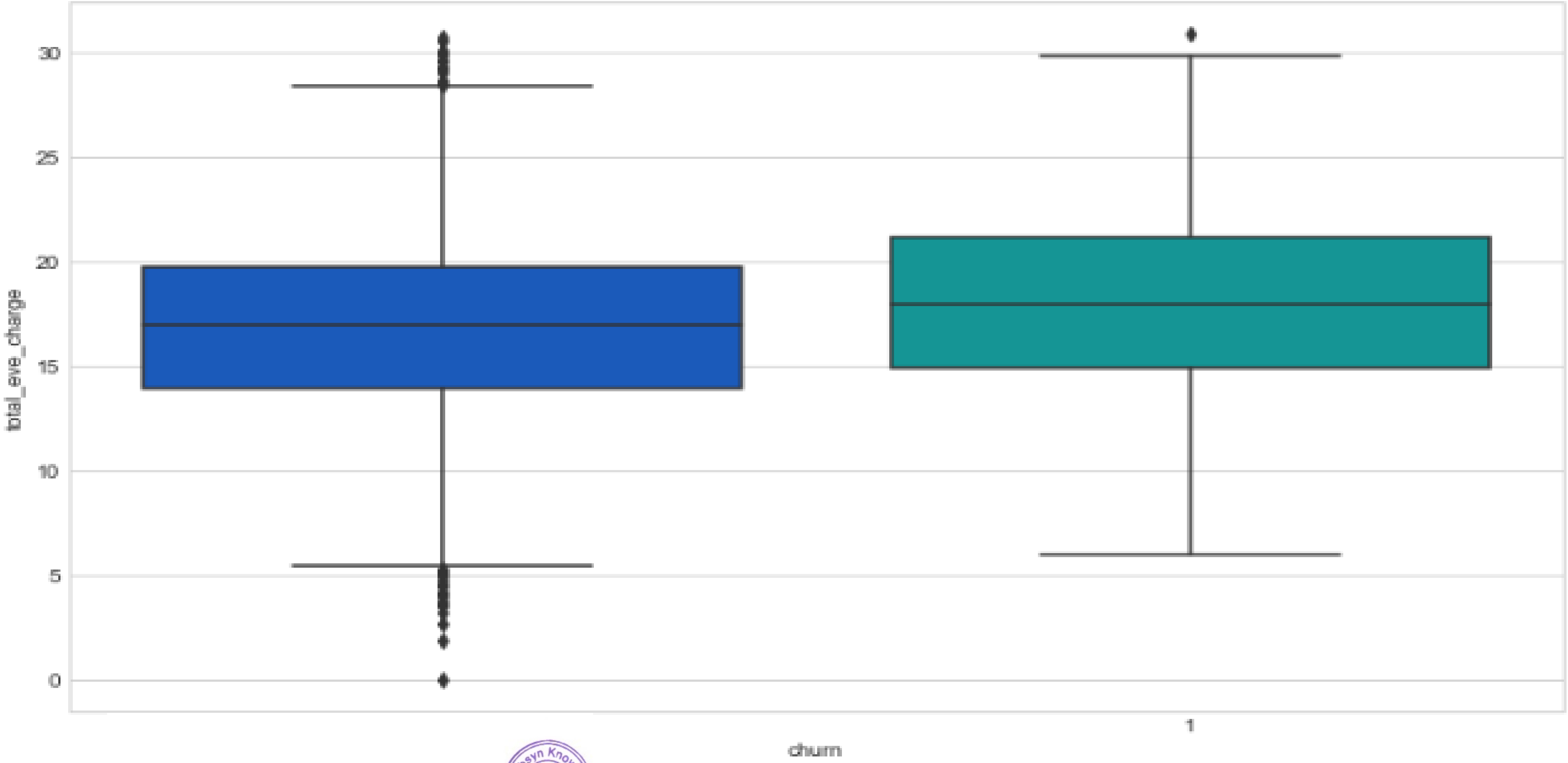
In [20]:
```
plt.figure(figsize=(12,7))
sns.boxplot(x='churn',y='total_day_charge',data=ch,palette='winter')
```

Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0xef08964940>



In [21]:
```
plt.figure(figsize=(12,7))
sns.boxplot(x='churn',y='total_eve_charge',data=ch,palette='winter')
```

Out[21]: <matplotlib.axes._subplots.AxesSubplot at 0xef0778ce10>



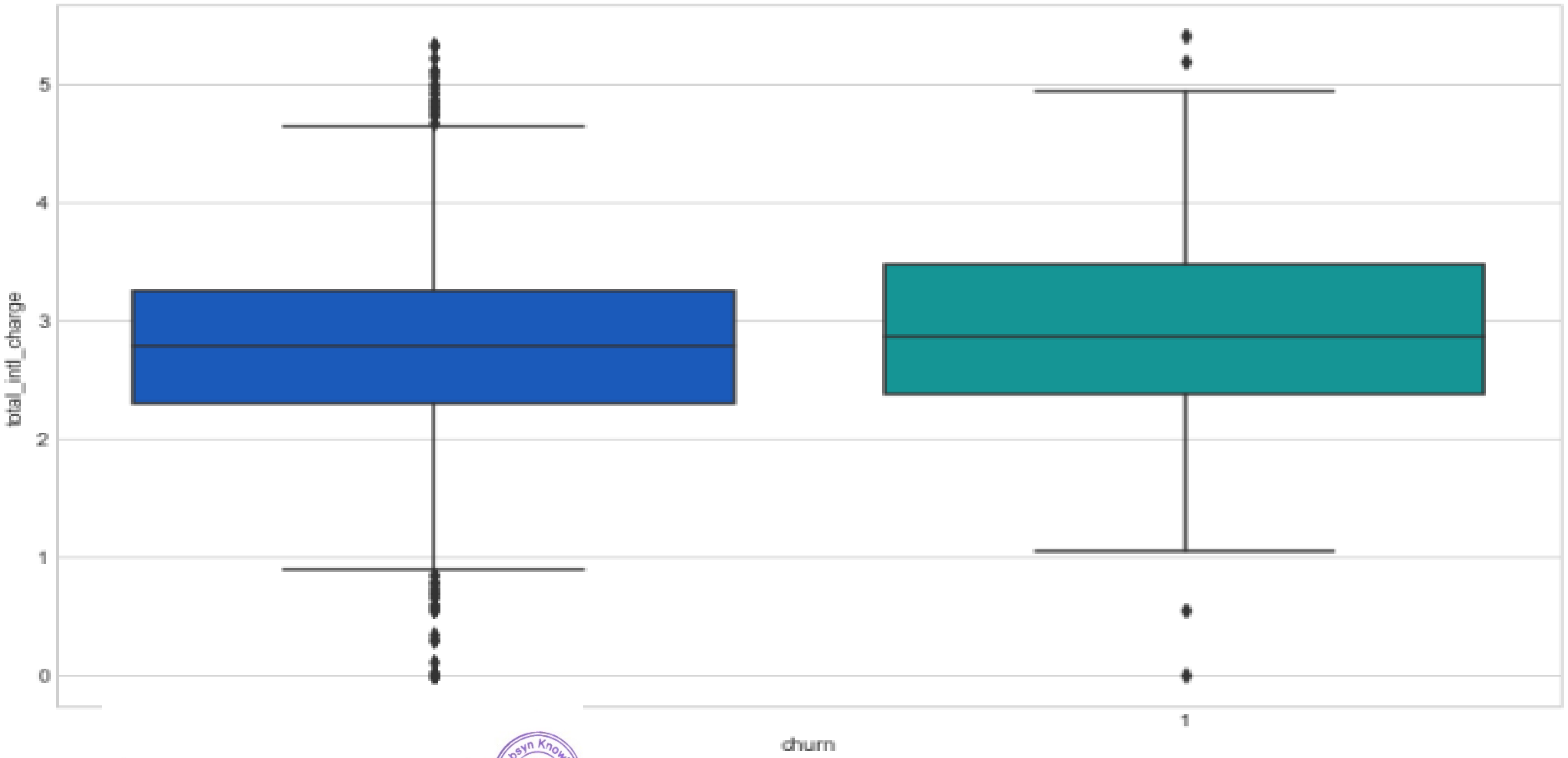
Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

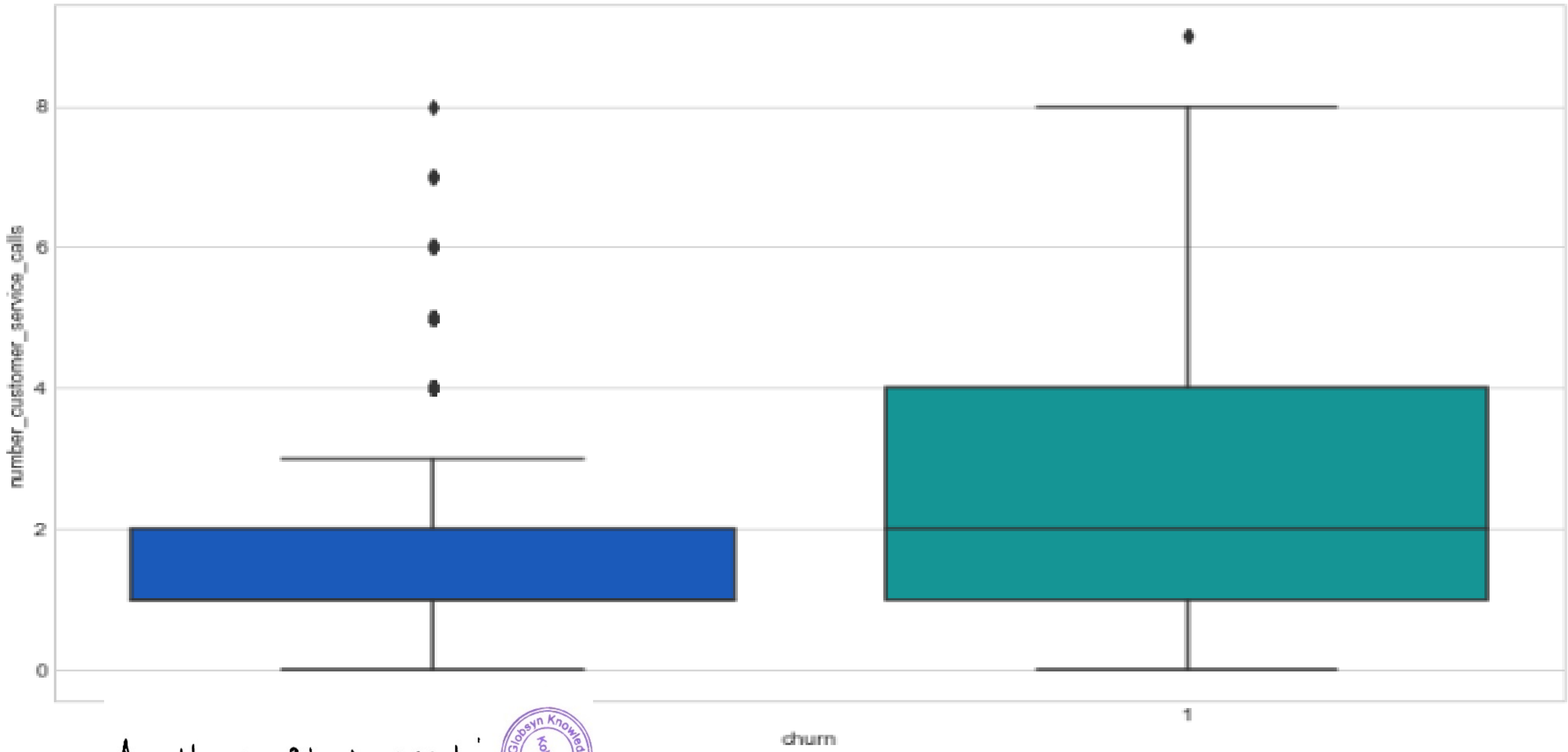
In [22]:
```
plt.figure(figsize=(12,7))
sns.boxplot(x='churn',y='total_intl_charge',data=ch,palette='winter')
```

Out[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0xef0894cef0>
```



In [23]:
```
plt.figure(figsize=(12,7))
sns.boxplot(x='churn',y='number_customer_service_calls',data=ch,palette='winter')
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0xef0816ada0>



Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Logistic Regression

In [24]:
```
ch.drop('account_length',axis=1,inplace=True)
```

In [25]:
```
ch.head()
```

Out[25]:

	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls	churn
0	25	45.07	16.78	11.01	2.70	1	0
1	26	27.47	16.62	11.45	3.70	1	0
2	0	41.38	10.30	7.32	3.29	0	0
3	0	50.90	5.26	8.86	1.78	2	0
4	0	28.34	12.61	8.41	2.73	3	0

In [26]:
```
x=ch[['number_vmail_messages','total_day_charge','total_eve_charge','total_night_charge','total_intl_charge','number_customer_service_calls']]
y=ch['churn']
```

In [27]:
```
from sklearn.model_selection import train_test_split
```

In [28]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30,random_state=101)
```

In [29]:
```
from sklearn.linear_model import LogisticRegression
```

In [30]:
```
logmodel=Log
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [31]:
```
logmodel.fit(x_train,y_train)
```

Out[31]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
          intercept_scaling=1, max_iter=100, multi_class='ovr', n_jobs=1,
          penalty='l2', random_state=None, solver='liblinear', tol=0.0001,
          verbose=0, warm_start=False)
```

In [32]:
```
predictions = logmodel.predict(x_test)
```

In [33]:
```
from sklearn.metrics import classification_report
```

In [34]:
```
print(classification_report(y_test,predictions))
```

```
             precision    recall  f1-score   support

          0       0.87      1.00      0.93      1297
          1       0.76      0.06      0.12       203

avg / total       0.86      0.87      0.82      1500
```

In [35]:
```
from sklearn.metrics import confusion_matrix
```

In [36]:
```
print(confusion_matrix(y_test,predictions))
```

```
[[1293    4]
 [ 190   13]]
```

In [37]:
```
acc_log=accuracy_score(y_test,predictions)
print("Accuracy of Logistic Regression : ",acc_log)
```

```
Accuracy                                    66666666667
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [43]:

```
from sklearn.metrics import confusion_matrix
features = ch.drop(["churn"], axis=1).columns
probs = logmodel.predict_proba(x_test[features])
get_ipython().magic('matplotlib inline')
confusion_matrix = pd.DataFrame(confusion_matrix(y_test, predictions), columns=["Predicted False", "Predicted True"], index=["Actual False", "Actual True"])
display(confusion_matrix)

# Calculate the fpr and tpr for all thresholds of the classification
fpr, tpr, threshold = roc_curve(y_test, probs[:,1])
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

	Predicted False	Predicted True
Actual False	1293	4
Actual True	190	13

Anubhav Chaturvedi

Document sign date :Aug 6, 2018

KNN!

In [45]:
```
from sklearn.preprocessing import StandardScaler
```

In [46]:
```
scaler = StandardScaler()
```

In [47]:
```
scaler.fit(ch.drop('churn',axis=1))
```

Out[47]:
```
StandardScaler(copy=True, with_mean=True, with_std=True)
```

In [48]:
```
scaled_features = scaler.transform(ch.drop('churn',axis=1))
```

In [49]:
```
ch_feat = pd.DataFrame(scaled_features,columns=ch.columns[:-1])
ch_feat.head()
```

Out[49]:

	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls
0	1.273145	1.574074	-0.063849	0.876286	-0.095509	-0.436676
1	1.346973	-0.347082	-0.101089	1.069818	1.245982	-0.436676
2	-0.572549	1.171286	-1.572084	-0.746737	0.695971	-1.202236
3	-0.572549	2.210457	-2.745155	-0.069377	-1.329681	0.328885
4	-0.572549	-0.252115	-1.034426	-0.267307	-0.055264	1.094445

In [50]:
```
from sklearn.model_selection import train_test_split
```

In [51]:
```
X_train, X_test, y_train, y_test = train_test_split(scaled_features,ch['churn'],test_size=0.30)
```

In [52]:
```
from skle assifier
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [53]:
```
knn = KNeighborsClassifier(n_neighbors=1)
```

In [54]:
```
knn.fit(X_train,y_train)
```

Out[54]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
           metric_params=None, n_jobs=1, n_neighbors=1, p=2,
           weights='uniform')
```

In [55]:
```
pred = knn.predict(X_test)
```

In [56]:
```
from sklearn.metrics import classification_report,confusion_matrix
```

In [57]:
```
print(confusion_matrix(y_test,pred))
```

```
[[1197   99]
 [  84  120]]
```

In [58]:
```
print(classification_report(y_test,pred))
```

```
             precision    recall  f1-score   support

          0       0.93      0.92      0.93      1296
          1       0.55      0.59      0.57       204

avg / total       0.88      0.88      0.88      1500
```

In [59]:
```
error_rate = []

for i in range(1,40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train,y_train)
    pred_
    error                                    test))
```

Anubhav Chaturvedi

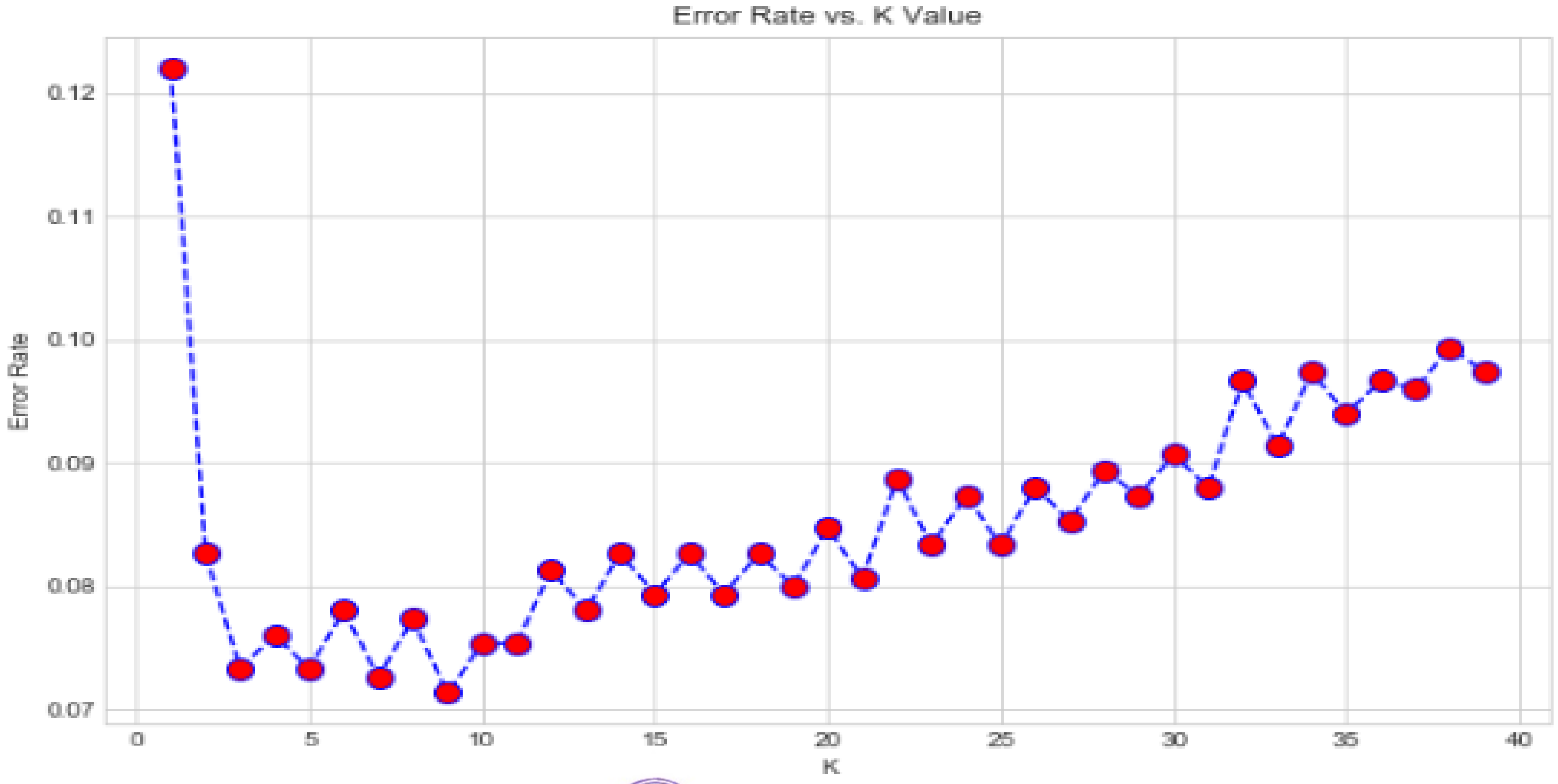
Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [60]:

```
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
```

Out[60]: `Text(0,0.5,'Error Rate')`



In [61]:
```
knn = KNeighborsClassifier(n_neighbors=1)

knn.fit(X_train,y_train)
pred = knn.predict(X_test)

print('WITH K=1')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))
```

```
WITH K=1


[[1197   99]
 [  84  120]]


             precision    recall  f1-score   support

          0       0.93      0.92      0.93      1296
          1       0.55      0.59      0.57       204

avg                          .88      0.88      1500
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

```
In [62]: knn = KNeighborsClassifier(n_neighbors=8)

         knn.fit(X_train,y_train)
         pred = knn.predict(X_test)

         print('WITH K=1')
         print('\n')

         print(confusion_matrix(y_test,pred))
         print('\n')
         print(classification_report(y_test,pred))
         acc_knn=accuracy_score(y_test,pred)
         print("Accuracy of KNN classifier : ",acc_knn)
```

```
WITH K=1


[[1294    2]
 [ 114   90]]


             precision    recall  f1-score   support

          0       0.92      1.00      0.96      1296
          1       0.98      0.44      0.61       204

avg / total       0.93      0.92      0.91      1500

Accurac                                   666666666
```

Anubhav Chaturvedi

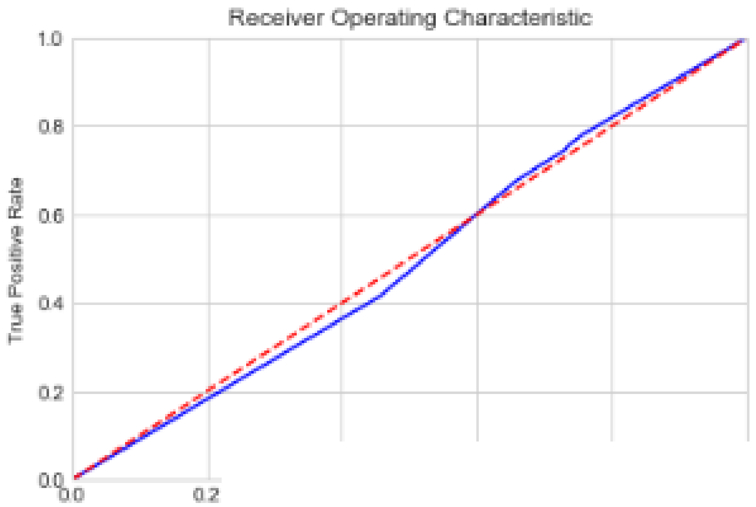
Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [86]:

```
from sklearn.metrics import confusion_matrix
features = ch.drop(["churn"], axis=1).columns
probs = knn.predict_proba(x_test[features])
get_ipython().magic('matplotlib inline')
confusion_matrix = pd.DataFrame(confusion_matrix(y_test, pred), columns=["Predicted False", "Predicted True"], index=["Actual False", "Actual True"])
display(confusion_matrix)

# Calculate the fpr and tpr for all thresholds of the classification
fpr, tpr, threshold = roc_curve(y_test, probs[:,1])
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

	Predicted False	Predicted True
Actual False	1294	2
Actual True	114	90



Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Naive Bayse

In [64]:
```
#Import Library of Gaussian Naive Bayes model
from sklearn.naive_bayes import GaussianNB
```

In [65]:
```
x1=np.array(ch1['account_length'].head(5000)).reshape(-1,1)
x2=np.array(ch1['number_vmail_messages'].head(5000)).reshape(-1,1)
x3=np.array(ch1['total_day_charge'].head(5000)).reshape(-1,1)
x4=np.array(ch1['total_eve_charge'].head(5000)).reshape(-1,1)
x5=np.array(ch1['total_night_charge'].head(5000)).reshape(-1,1)
x6=np.array(ch1['total_intl_charge'].head(5000)).reshape(-1,1)
x7=np.array(ch1['number_customer_service_calls'].head(5000)).reshape(-1,1)

y1=np.array(ch1["churn"].head(5000))
```

In [66]:
```
df=ch1[ch1["churn"]==False]
df.head(5)
```

Out[66]:

	account_length	number_vmail_messages	total_day_charge	total_eve_charge	total_night_charge	total_intl_charge	number_customer_service_calls	churn
0	128	25	45.07	16.78	11.01	2.70	1	0
1	107	26	27.47	16.62	11.45	3.70	1	0
2	137	0	41.38	10.30	7.32	3.29	0	0
3	84	0	50.90	5.26	8.86	1.78	2	0
4	[illegible]	[illegible]	[illegible]	12.61	8.41	2.73	3	0

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [67]:

```
model=GaussianNB()
model.fit(x1,y1)

x10=np.array(df['account_length'].tail(4293)).reshape(-1,1)
x11=np.array(df['number_vmail_messages'].tail(4293)).reshape(-1,1)
x12=np.array(df['total_day_charge'].tail(4293)).reshape(-1,1)
x13=np.array(df['total_eve_charge'].tail(4293)).reshape(-1,1)
x14=np.array(df['total_night_charge'].tail(4293)).reshape(-1,1)
x15=np.array(df['total_intl_charge'].tail(4293)).reshape(-1,1)
x16=np.array(df['number_customer_service_calls'].tail(4293)).reshape(-1,1)
predicted1=(model.predict(x10))
print(predicted1)

model.fit(x2,y1)
predicted2=(model.predict(x11))
print(predicted2)

model.fit(x3,y1)
predicted3=(model.predict(x12))
print(predicted3)

model.fit(x4,y1)
predicted4=(model.predict(x13))
print(predicted4)

model.fit(x5,y1)
predicted5=(model.predict(x14))
print(predicted5)

model.fit(x6,y1)
predicted6=(model.predict(x15))
print(predicted6)

model.fit(x7,y1)
predicted7=(model.predict(x16))
print(predicted7)
```

```
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ... 0 0 0]
[0 0 0 ...
[0 0 0 ...
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [68]:
```
y2=ch["churn"].head(4293)
print("For Account Length\n",classification_report(y2,predicted1))
print("number_vmail_messages\n",classification_report(y2,predicted2))
print("total_day_charge\n",classification_report(y2,predicted3))
print("total_eve_charge\n",classification_report(y2,predicted4))
print("total_night_charge\n",classification_report(y2,predicted5))
print("total_intl_charge\n",classification_report(y2,predicted6))
print("number_customer_service_calls\n",classification_report(y2,predicted7))
```

```
For Account Length
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.00      0.00      0.00       616

avg / total       0.73      0.86      0.79      4293

number_vmail_messages
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.00      0.00      0.00       616

avg / total       0.73      0.86      0.79      4293

total_day_charge
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.06      0.00      0.00       616

avg / t                               0.79      4293
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

```
total_eve_charge
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.00      0.00      0.00       616

avg / total       0.73      0.86      0.79      4293

total_night_charge
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.00      0.00      0.00       616

avg / total       0.73      0.86      0.79      4293

total_intl_charge
             precision    recall  f1-score   support

          0       0.86      1.00      0.92      3677
          1       0.00      0.00      0.00       616

avg / total       0.73      0.86      0.79      4293

number_customer_service_calls
             precision    recall  f1-score   support

          0       0.86      0.99      0.92      3677
          1       0.14      0.01      0.02       616

avg / total       0.75      0.85      0.79      4293
```

In [69]:

```
acc_nb=accuracy_score(y2,predicted1)
print("Accuracy of Naive Bayse Classifier : ",acc_nb)
```

```
Accu                                    er :  0.8565105986489634
```

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Random Forest

```
In [61]: features = ch.drop(["churn"], axis=1).columns
```

```
In [62]: x_train, x_test = train_test_split(ch, test_size=0.25)
```

```
In [63]: # Set up our RandomForestClassifier instance and fit to data
         clf = RandomForestClassifier(n_estimators=30)
         clf.fit(x_train[features], x_train["churn"])

         # Make predictions
         predictions = clf.predict(x_test[features])
         probs = clf.predict_proba(x_test[features])
         display(predictions)

         array([0, 0, 0, ..., 0, 0, 0], dtype=int64)
```

```
In [64]: score = clf.score(x_test[features], x_test["churn"])
         print("Accuracy: ", score)

         Accur
```

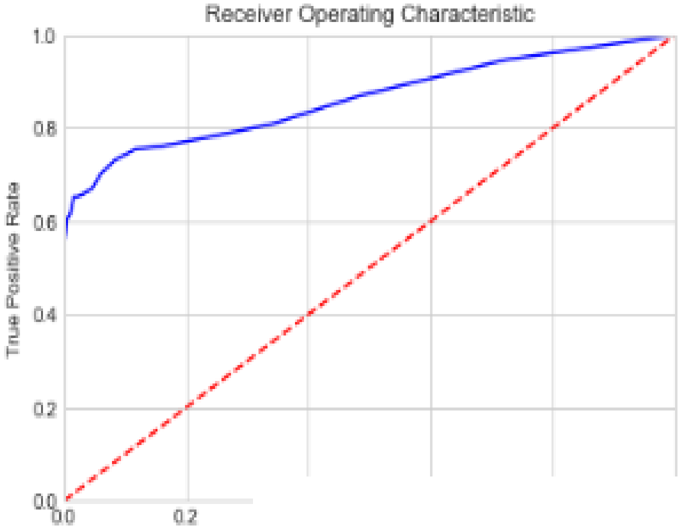
Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [65]:
```
get_ipython().magic('matplotlib inline')
confusion_matrix = pd.DataFrame(confusion_matrix(x_test["churn"], predictions), columns=["Predicted False", "Predicted True"], index=["Actual False", "Actual True"])
display(confusion_matrix)

# Calculate the fpr and tpr for all thresholds of the classification
fpr, tpr, threshold = roc_curve(x_test["churn"], probs[:,1])
plt.title('Receiver Operating Characteristic')
plt.plot(fpr, tpr, 'b')
plt.plot([0, 1], [0, 1],'r--')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.ylabel('True Positive Rate')
plt.xlabel('False Positive Rate')
plt.show()
```

	Predicted False	Predicted True
Actual False	1079	4
Actual True	71	96

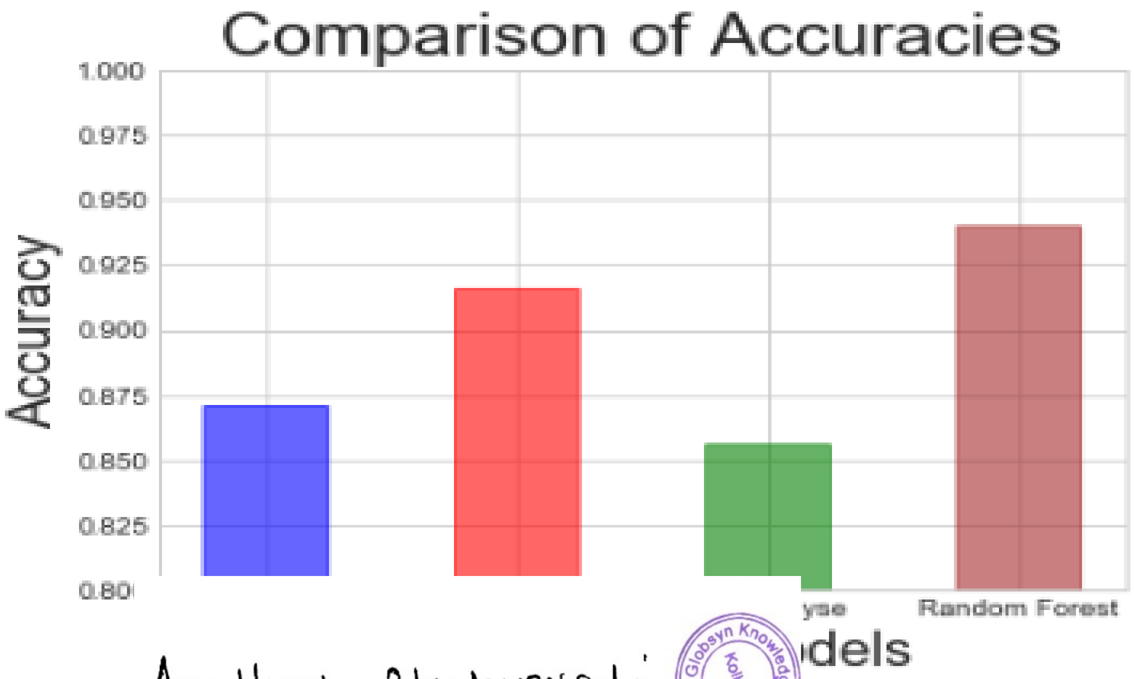


Anubhav Chaturvedi
Globsyn Knowledge Foundation Kolkata
Document sign date :Aug 6, 2018

Comparison of Accuracy

In [66]:

```
mod_compare=['Logistic Regression','K Neighbors','Naive Bayse','Random Forest']
acc=[acc_log,acc_knn,acc_nb,score]
barlist=plt.bar(mod_compare,acc,width=0.5,alpha=0.6)
plt.ylim([0.8,1.0])
barlist[0].set_color('blue')
barlist[1].set_color('red')
barlist[2].set_color('green')
barlist[3].set_color('brown')
plt.xlabel('Classification Models',fontsize=20)
plt.ylabel('Accuracy',fontsize=20)
plt.title('Comparison of Accuracies',fontsize=25)
plt.show();
```



Anubhav Chaturvedi

Document sign date :Aug 6, 2018

Decision Tree

In [122]:
```
from sklearn.tree import DecisionTreeClassifier
```

In [168]:
```
#dtree = DecisionTreeClassifier()
dtree = DecisionTreeClassifier(criterion = "entropy", random_state = 100,
 max_depth=3, min_samples_leaf=5)
```

In [169]:
```
X=ch[['number_vmail_messages','total_day_charge','total_eve_charge','total_night_charge','total_intl_charge','number_customer_service_calls']]
#X=ch[['number_vmail_messages']]
y=ch['churn']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)
dtree.fit(X_train,y_train)
```

Out[169]:
```
DecisionTreeClassifier(class_weight=None, criterion='entropy', max_depth=3,
            max_features=None, max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=5, min_samples_split=2,
            min_weight_fraction_leaf=0.0, presort=False, random_state=100,
            splitter='best')
```

In [170]:
```
from IPython.display import Image
from sklearn.externals.six import StringIO
from sklearn.tree import export_graphviz


features = list(ch.columns[1:])
features
```

Out[170]:
```
['total_day_charge',
 'total_eve_charge',
 'total_night_charge',
 'total_intl_charge',
 'number_customer_service_calls',
 'churn']
```

Anubhav Chaturvedi

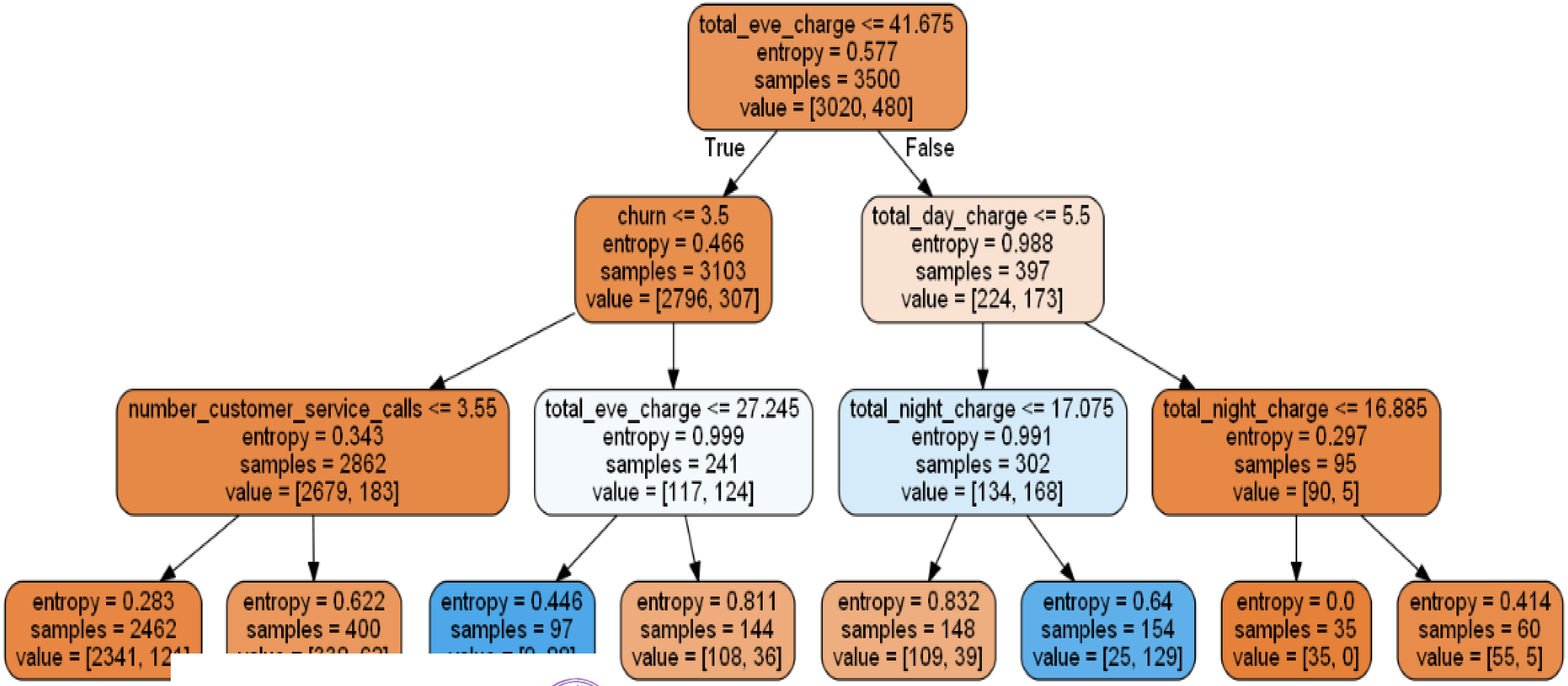
Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

In [171]:

```python
import pydot
dot_data = StringIO()
export_graphviz(dtree, out_file=dot_data,feature_names=features,filled=True,rounded=True)

graph = pydot.graph_from_dot_data(dot_data.getvalue())
Image(graph[0].create_png())
```

Out[171]:



Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Future Scope of Improvements

Based on the EDA and models, an increase in the below variables increases the probability of customer churn:

Number of customer service calls

Total day charge

Total evening charge

Total international charge

Total night charge

Additionally, an increase in number of voice mail messages decreases the probability of customer churn. These insights from the discriminant model can help the business formulate strategies to reduce customer churn. Here's what I would recommend to the business based on what we've learned: First, customer issues should be resolved within the first or second call, as repeated calls to customer service causes customer churn. Second, there should be an organized escalation procedure for issues not resolved within two calls. Lastly, the provider should offer more attractive plans that reduce the cost of day, evening, and international calls based on usage.

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Certificate

This is to certify that Mr [*Karan Patadia*] of [*The Heritage Academy*], registration number: [*162131010047*], has successfully completed a project on [*Predicting Customer Churn*] using [*Machine Learning with Python*] under the guidance of Mr [*Anubhav Chaturvedi*].

[*Anubhav Chaturvedi*]
Globsyn Finishing School

Anubhav Chaturvedi

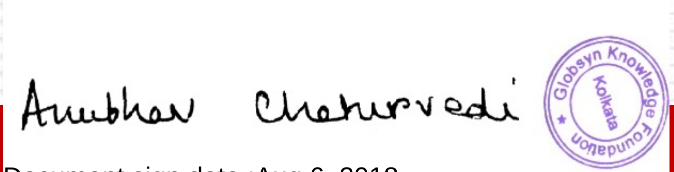
Globsyn Knowledge Foundation
Kolkata

Document sign date :Aug 6, 2018

Certificate

This is to certify that Mr [*Parwez Alam*] of [*The Heritage Academy*], registration number: [*162131010059*], has successfully completed a project on [*Predicting Customer Churn*] using [*Machine Learning with Python*] under the guidance of Mr [*Anubhav Chaturvedi*].

[*Anubhav Chaturvedi*]
Globsyn Finishing School

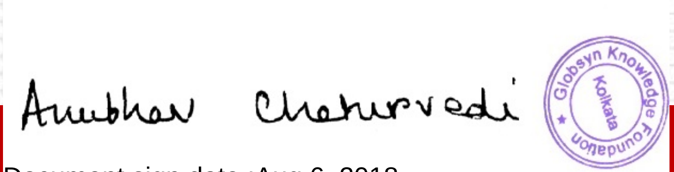


Document sign date :Aug 6, 2018

Certificate

This is to certify that Mr [*Deepak Kumar Rajak*] of [*The Heritage Academy*], registration number: [*162131010030*], has successfully completed a project [*Predicting Customer Churn*] using [*Machine Learning with Python*] under the guidance of Mr [*Anubhav Chaturvedi*].

[*Anubhav Chaturvedi*]
Globsyn Finishing School



Document sign date :Aug 6, 2018

Certificate

This is to certify that Mr [*Shreerup Sharma*] of [*The Heritage Academy*], registration number: [*162131010103*], has successfully completed a project on [*Predicting Customer Churn*] using [*Machine Learning with Python*] under the guidance of Mr [*Anubhav Chaturvedi*].

[*Anubhav Chaturvedi*]
Globsyn Finishing School

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Certificate

This is to certify that Mr [*Rupam Aich*] of *The* [*The Heritage Academy*], registration number: [*162131010080*], has successfully completed a project on [*Predicting Customer Churn*] using [*Machine Learning with Python*] under the guidance of Mr [*Anubhav Chaturvedi*].

[*Anubhav Chaturvedi*]
Globsyn Finishing School

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018

Anubhav Chaturvedi

Globsyn Knowledge Foundation Kolkata

Document sign date :Aug 6, 2018