```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
```

```
data = pd.read_csv("BCG_Modelling Dataset.csv")
```

```
data.head()
```

ed: 0	cons_12m	cons_gas_12m	cons_last_month	date_activ	date_end	date_modif_prod	date
0	309275	0	10025	2012-11-07	2016-11-06	2012-11-07	2
1	4660	0	0	2009-08-21	2016-08-30	2009-08-21	2
2	544	0	0	2010-04-16	2016-04-16	2010-04-16	2
3	1584	0	0	2010-03-30	2016-03-30	2010-03-30	2
4	121335	0	12400	2010-04-08	2016-04-08	2010-04-08	2

```
drop(['Unnamed: 0', 'date_activ', 'date_end', 'date_modif_prod', 'date_renewal', 'contract_mo
```

```
Y = data['churn']
```

```
Y = Y.replace({'Churned':1 ,'Stayed':0})
```

```
['cons_12m',  'cons_gas_12m', 'cons_last_month', 'forecast_cons_12m', 'forecast_cons_year', '
```

```
#Splitting the Dataset
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

```
X_train = X_train.drop(['churn'], axis = 1)
```

```
X_test = X_test.drop(['churn'], axis = 1)
```

Logistic Regression Model

```
from sklearn.linear_model import LogisticRegression
```

```
log_reg = LogisticRegression()
log_reg.fit(X_train, Y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:940: Convergen
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG)
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=None, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

```
Y_pred_log = log_reg.predict(X_test)
Y_pred_log
```

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

```
Y_train
```

```
10626    0
4075     0
7674     0
10447    1
9206     0
        ..
13123    0
3264     0
9845     0
10799    0
2732     0
Name: churn, Length: 11755, dtype: int64
```

```
from sklearn.metrics import confusion_matrix, accuracy_score
```

```
accuracy_score_log = accuracy_score(Y_pred_log, Y_test)
```

```
accuracy_score_log
```

```
     0.897933146210768
```

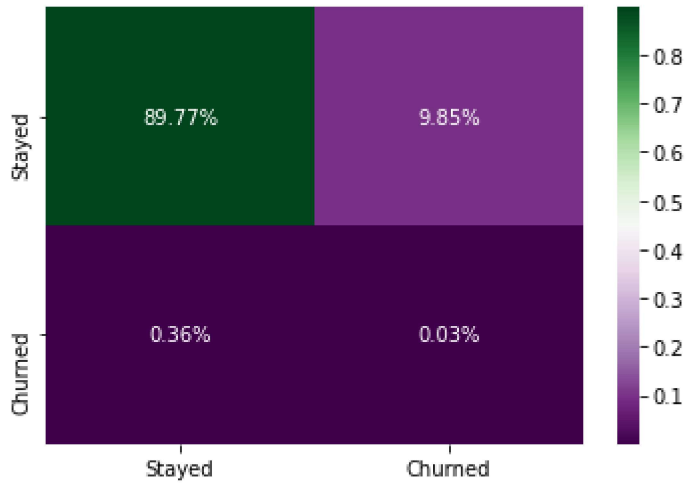
```
cm1 = confusion_matrix(Y_pred_log, Y_test)
```

```
cm1
```

```
     array([[3518,  386],
            [  14,    1]])
```

```
sns.heatmap(cm1/np.sum(cm1), annot=True, xticklabels = ['Stayed', 'Churned'], yticklabels = [
```

```
     <matplotlib.axes._subplots.AxesSubplot at 0x7f27c75a5e90>
```



Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
```

```
rfc = RandomForestClassifier(n_estimators = 1000)
rfc.fit(X_train, Y_train)
```

```
     RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                            criterion='gini', max_depth=None, max_features='auto',
                            max_leaf_nodes=None, max_samples=None,
                            min_impurity_decrease=0.0, min_impurity_split=None,
                            min_samples_leaf=1, min_samples_split=2,
                            min_weight_fraction_leaf=0.0, n_estimators=1000,
                            n_jobs=None, oob_score=False, random_state=None,
                            verbose=0, warm_start=False)
```

```
Y_pred_rfc = rfc.predict(X_test)
```

```
Y_pred_rfc
```

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

```
accuracy_score_rfc = accuracy_score(Y_pred_rfc, Y_test)
```

```
accuracy_score_rfc
```

```
    0.9038019903036489
```

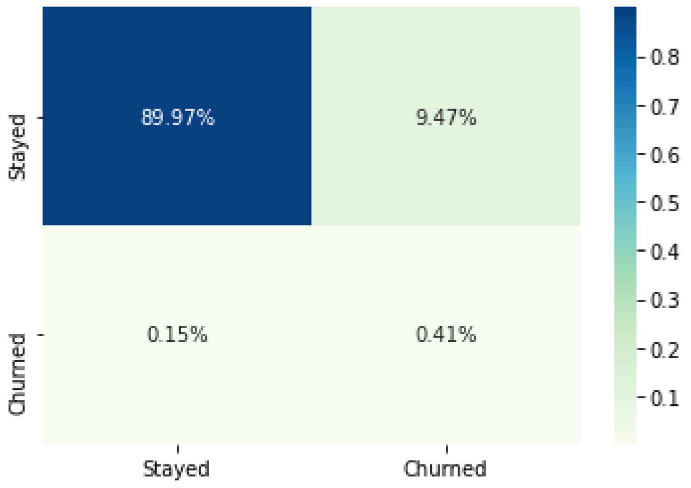
```
cm2 = confusion_matrix(Y_pred_rfc, Y_test)
```

```
cm2
```

```
    array([[3526,  371],
           [   6,   16]])
```

```
sns.heatmap(cm2/np.sum(cm2), annot=True, xticklabels = ['Stayed', 'Churned'], yticklabels = [
```

```
    <matplotlib.axes._subplots.AxesSubplot at 0x7f27c77861d0>
```



0s completed at 12:59 AM