In [1]:

```
import pandas as pd
import numpy as np
from plotnine import *
import matplotlib
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error

from sklearn.linear_model import LogisticRegression

from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.metrics import plot_confusion_matrix

%matplotlib inline
```

In [2]:

```
Q4 = pd.read_csv("/Users/irenehuang/Desktop/MGSC410/RSFinal/Q4.csv")
```

In [3]:

```
Q4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 26529 entries, 0 to 26528
Data columns (total 28 columns):
ID                    26529 non-null int64
Language              26529 non-null object
Subscription_Type     26529 non-null int64
Event_Type            26529 non-null int64
Purchase_Store        26529 non-null int64
Quiter                26529 non-null int64
Duration              26529 non-null int64
Demo_User             26529 non-null int64
Free_Trial_User       26529 non-null int64
Auto_Renew            26529 non-null int64
Country               26529 non-null int64
User_Type             26529 non-null int64
Lead_Platform         26529 non-null int64
Email_Subscription    26529 non-null int64
Push_Notification     26529 non-null int64
Open/Send             26529 non-null float64
Click/Send            26529 non-null float64
Send_Count            26529 non-null int64
Open_Count            26529 non-null int64
Click_Count           26529 non-null int64
Unique_Open_Count     26529 non-null int64
Unique_Click_Count    26529 non-null int64
Start                 26529 non-null int64
Other                 26529 non-null int64
Completed             26529 non-null int64
NULL                  26529 non-null int64
Onboarding            26529 non-null int64
App_Launch_Times      26529 non-null int64
dtypes: float64(2), int64(25), object(1)
memory usage: 5.7+ MB
```

In [4]:

```
Q4.head()
```

Out[4]:

	ID	Language	Subscription_Type	Event_Type	Purchase_Store	Quiter	Duration	Demo_
0	24189	ENG	0	0	0	0	31	
1	27676	ENG	0	0	1	0	31	
2	33391	FRA	0	0	1	0	31	
3	35553	JPN	0	0	1	0	31	
4	36632	ESP	0	0	0	0	31	

5 rows × 28 columns

In [5]:

```
Q4DF = pd.DataFrame(Q4)
print(Q4DF.columns)
```

```
Index(['ID', 'Language', 'Subscription_Type', 'Event_Type', 'Purchas
e_Store',
       'Quiter', 'Duration ', 'Demo_User ', 'Free_Trial_User', 'Auto
_Renew',
       'Country', 'User_Type', 'Lead_Platform', 'Email_Subscriptio
n',
       'Push_Notification', 'Open/Send', 'Click/Send', 'Send_Count',
       'Open_Count', 'Click_Count', 'Unique_Open_Count', 'Unique_Cli
ck_Count',
       'Start', 'Other', 'Completed', 'NULL', 'Onboarding',
       'App_Launch_Times'],
      dtype='object')
```

In [6]:

```
z = StandardScaler()
```

In [7]:

```
Q4_feature = Q4.columns[2:27]
```

In [8]:

```
Q4_feature = Q4_feature.drop(["Subscription_Type","Duration ","Quiter","Send_Cou
nt","Open_Count",
                              "Click_Count","Unique_Open_Count","Unique_Click_Co
unt"])
```

In [9]:

```
Q4_predictors = Q4[Q4_feature]
Q4_y = Q4["Quiter"]
```

In [10]:

```
Q4_X_train, Q4_X_test, Q4_y_train, Q4_y_test = train_test_split(Q4_predictors, Q
4_y, test_size=0.2)
z.fit(Q4_X_train)
Xz_train = z.transform(Q4_X_train)
Xz_test = z.transform(Q4_X_test)
```

In [11]:

```
myLogit = LogisticRegression(penalty = "none")
```

In [12]:

```
myLogit.fit(Xz_train,Q4_y_train)
```

Out[12]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_interce
pt=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='none',
                   random_state=None, solver='lbfgs', tol=0.0001, ve
rbose=0,
                   warm_start=False)
```

In [13]:

```
predictedVals = myLogit.predict(Xz_test)
```

In [14]:

```
accuracy_score(Q4_y_test,predictedVals)
```

Out[14]:

```
0.9046362608367885
```

In [15]:

```
coef = pd.DataFrame({"Coefs": myLogit.coef_[0], "Names": Q4_predictors})
coef = coef.append({"Coefs": myLogit.intercept_[0], "Names": "intercept"}, ignor
e_index = True)
coef
```

Out[15]:

	Coefs	Names
0	0.502395	(E, v, e, n, t, _, T, y, p, e)
1	-0.073433	(P, u, r, c, h, a, s, e, _, S, t, o, r, e)
2	-0.180658	(D, e, m, o, _, U, s, e, r,)
3	0.235723	(F, r, e, e, _, T, r, i, a, l, _, U, s, e, r)
4	0.140433	(A, u, t, o, _, R, e, n, e, w)
5	-0.300336	(C, o, u, n, t, r, y)
6	0.215688	(U, s, e, r, _, T, y, p, e)
7	-0.361991	(L, e, a, d, _, P, l, a, t, f, o, r, m)
8	0.025937	(E, m, a, i, l, _, S, u, b, s, c, r, i, p, t, ...
9	0.438137	(P, u, s, h, _, N, o, t, i, f, i, c, a, t, i, ...
10	-0.003396	(O, p, e, n, /, S, e, n, d)
11	0.028320	(C, l, i, c, k, /, S, e, n, d)
12	0.021243	(S, t, a, r, t)
13	0.128886	(O, t, h, e, r)
14	-0.016689	(C, o, m, p, l, e, t, e, d)
15	0.009699	(N, U, L, L)
16	0.005515	(O, n, b, o, a, r, d, i, n, g)
17	-2.417842	intercept

In [16]:

```
coef["Odds Coefs"] = np.exp(coef["Coefs"])
coef
```

Out[16]:

	Coefs	Names	Odds Coefs
0	0.502395	(E, v, e, n, t, _, T, y, p, e)	1.652674
1	-0.073433	(P, u, r, c, h, a, s, e, _, S, t, o, r, e)	0.929198
2	-0.180658	(D, e, m, o, _, U, s, e, r,)	0.834721
3	0.235723	(F, r, e, e, _, T, r, i, a, l, _, U, s, e, r)	1.265824
4	0.140433	(A, u, t, o, _, R, e, n, e, w)	1.150772
5	-0.300336	(C, o, u, n, t, r, y)	0.740569
6	0.215688	(U, s, e, r, _, T, y, p, e)	1.240715
7	-0.361991	(L, e, a, d, _, P, l, a, t, f, o, r, m)	0.696289
8	0.025937	(E, m, a, i, l, _, S, u, b, s, c, r, i, p, t, ...	1.026276
9	0.438137	(P, u, s, h, _, N, o, t, i, f, i, c, a, t, i, ...	1.549818
10	-0.003396	(O, p, e, n, /, S, e, n, d)	0.996610
11	0.028320	(C, l, i, c, k, /, S, e, n, d)	1.028725
12	0.021243	(S, t, a, r, t)	1.021470
13	0.128886	(O, t, h, e, r)	1.137560
14	-0.016689	(C, o, m, p, l, e, t, e, d)	0.983449
15	0.009699	(N, U, L, L)	1.009746
16	0.005515	(O, n, b, o, a, r, d, i, n, g)	1.005530
17	-2.417842	intercept	0.089114