EDA No. 7 AAA Project Martin George mgeorgevienna@gmail.com

In [1]:
```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
df = pd.read_csv('member_sample.csv', index_col = 0)
```

In [ ]:

Application of using GAIN LIFT on a model on AAA data

In [2]:
```
df.head()
df.info()
df.columns
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 21344 entries, 0 to 99998
Columns: 112 entries, Individual Key to Was Towed To AAR Referral
dtypes: float64(35), object(77)
memory usage: 18.4+ MB
```

Out[2]:
```
Index(['Individual Key', 'Household Key', 'Member Flag', 'City',
       'State - Grouped', 'ZIP5', 'ZIP9', 'FSV CMSI Flag',
       'FSV Credit Card Flag', 'FSV Deposit Program Flag',
       ...
       'SC Vehicle Manufacturer Name', 'SC Vehicle Model Name',
       'SVC Facility Name', 'SVC Facility Type', 'Total Cost',
       'Tow Destination Latitude', 'Tow Destination Longitude',
       'Tow Destination Name', 'Was Duplicated', 'Was Towed To AAR Referral'],
      dtype='object', length=112)
```

In [3]:
```python
df.head()
```

Out[3]:

	Individual Key	Household Key	Member Flag	City	State - Grouped	ZIP5	ZIP9	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	...	SC Vehicle Manufacturer Name	SC Vehicle Model Name	SVC Facility Name	F
0	10000003.0	10462590.0	Y	NEW HAVEN	CT	6511.0	65111349.0	N	N	N	...	NaN	NaN	NaN	
1	52211550.0	4500791.0	Y	WEST WARWICK	RI	2893.0	28933850.0	N	Y	N	...	TOYOTA	CAMRY	ASTRO WRECKER SERVICE	indepe
2	52211550.0	4500791.0	Y	WEST WARWICK	RI	2893.0	28933850.0	N	Y	N	...	TOYOTA	CAMRY	Astro Wrecker Service	indepe
3	52211550.0	4500791.0	Y	WEST WARWICK	RI	2893.0	28933850.0	N	Y	N	...	TOYOTA	CAMRY	ASTRO WRECKER SERVICE	indepe
4	52211550.0	4500791.0	Y	WEST WARWICK	RI	2893.0	28933850.0	N	Y	N	...	TOYOTA	CAMRY	ASTRO WRECKER SERVICE	indepe

5 rows × 112 columns

In [4]:
```python
df.groupby('FSV CMSI Flag').mean()
```

Out[4]:

	Individual Key	Household Key	ZIP5	ZIP9	Length Of Residence	Do Not Direct Mail Solicit	Email Available	ERS ENT Count Year 1	ERS ENT Count Year 2	ERS ENT Count Year 3	...	Member Match Flag	Me Numbe Associ
FSV CMSI Flag													
N	3.403291e+07	1.600860e+07	2947.671848	2.948020e+07	11.552839	0.054041	0.52604	0.517824	0.921864	0.952447	...	1.0	1.09198
Y	2.398762e+07	1.515128e+07	2885.457413	2.885794e+07	11.088766	0.027340	0.75184	0.531746	1.193878	1.090703	...	1.0	1.07118

2 rows × 35 columns

Consider a classification problem.

In [5]:
```
def yn(x):
    return x.replace('N',0).replace('Y',1)
```

In [6]:
```
 products_c= [i for i in df.columns if i.startswith('FSV')]
```

In [7]:
```
products = df[[i for i in df.columns if i.startswith('FSV')]]
```

In [8]:
```
products
```

Out[8]:

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
0	N	N	N	N	N	N
1	N	Y	N	N	N	N
2	N	Y	N	N	N	N
3	N	Y	N	N	N	N
4	N	Y	N	N	N	N
5	N	Y	N	N	N	N
6	N	Y	N	N	N	N
7	N	Y	N	N	N	N
8	N	Y	N	N	N	N
9	N	Y	N	N	N	N
10	N	N	N	N	N	N
11	N	N	N	N	N	N
12	N	N	N	N	N	N
13	N	N	N	N	N	N
14	N	N	N	N	N	N
15	N	N	N	N	N	N
16	N	N	N	N	N	N
17	N	N	N	N	N	N
18	N	N	N	N	N	N
19	N	N	N	N	N	N
20	N	N	N	N	N	N
21	N	N	N	N	N	N
22	N	N	N	N	N	N
23	N	N	N	N	N	N
24	N	N	N	N	N	N
25	N	N	N	N	N	N
26	N	N	N	N	N	N
27	N	N	N	N	N	N
28	N	N	N	N	N	N

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
29	N	N	N	N	N	N
...	...	...	...	...	...	...
99968	N	N	N	N	N	N
99969	N	N	N	N	N	N
99970	N	N	N	N	N	N
99971	N	N	N	N	N	N
99972	N	N	N	N	N	N
99973	N	N	N	N	N	N
99974	N	N	N	N	N	N
99975	N	N	N	N	N	N
99976	N	N	N	N	N	N
99977	N	N	N	N	N	N
99979	N	N	N	N	N	N
99980	N	N	N	N	N	N
99981	N	N	N	N	N	N
99982	N	N	N	N	Y	N
99983	Y	N	N	N	N	N
99984	Y	N	N	N	N	N
99985	Y	N	N	N	N	N
99986	Y	N	N	N	N	N
99987	Y	N	N	N	N	N
99988	N	Y	N	N	N	N
99989	Y	N	N	N	N	N
99990	N	N	N	N	N	N
99991	N	N	N	N	N	N
99992	N	N	N	N	N	N
99993	N	N	N	N	N	N
99994	N	N	N	N	N	N
99995	N	N	N	N	N	N
99996	N	N	N	N	N	N

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
99997	N	N	N	N	N	N
99998	N	N	N	N	N	N

21344 rows × 6 columns

In [ ]:
```
#products['FSV CMSI Flag'].apply(yn)
```

In [10]:

```
products
```

Out[10]:

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
0	N	N	N	N	N	N
1	N	Y	N	N	N	N
2	N	Y	N	N	N	N
3	N	Y	N	N	N	N
4	N	Y	N	N	N	N
5	N	Y	N	N	N	N
6	N	Y	N	N	N	N
7	N	Y	N	N	N	N
8	N	Y	N	N	N	N
9	N	Y	N	N	N	N
10	N	N	N	N	N	N
11	N	N	N	N	N	N
12	N	N	N	N	N	N
13	N	N	N	N	N	N
14	N	N	N	N	N	N
15	N	N	N	N	N	N
16	N	N	N	N	N	N
17	N	N	N	N	N	N
18	N	N	N	N	N	N
19	N	N	N	N	N	N
20	N	N	N	N	N	N
21	N	N	N	N	N	N
22	N	N	N	N	N	N
23	N	N	N	N	N	N
24	N	N	N	N	N	N
25	N	N	N	N	N	N
26	N	N	N	N	N	N
27	N	N	N	N	N	N
28	N	N	N	N	N	N

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
29	N	N	N	N	N	N
...	...	...	...	...	...	...
99968	N	N	N	N	N	N
99969	N	N	N	N	N	N
99970	N	N	N	N	N	N
99971	N	N	N	N	N	N
99972	N	N	N	N	N	N
99973	N	N	N	N	N	N
99974	N	N	N	N	N	N
99975	N	N	N	N	N	N
99976	N	N	N	N	N	N
99977	N	N	N	N	N	N
99979	N	N	N	N	N	N
99980	N	N	N	N	N	N
99981	N	N	N	N	N	N
99982	N	N	N	N	Y	N
99983	Y	N	N	N	N	N
99984	Y	N	N	N	N	N
99985	Y	N	N	N	N	N
99986	Y	N	N	N	N	N
99987	Y	N	N	N	N	N
99988	N	Y	N	N	N	N
99989	Y	N	N	N	N	N
99990	N	N	N	N	N	N
99991	N	N	N	N	N	N
99992	N	N	N	N	N	N
99993	N	N	N	N	N	N
99994	N	N	N	N	N	N
99995	N	N	N	N	N	N
99996	N	N	N	N	N	N

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
99997	N	N	N	N	N	N
99998	N	N	N	N	N	N

21344 rows × 6 columns

In [24]:

```
products = products.apply(yn)
```

In [25]:
```
products
```

Out[25]:

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
0	0	0	0	0	0	0
1	0	1	0	0	0	0
2	0	1	0	0	0	0
3	0	1	0	0	0	0
4	0	1	0	0	0	0
5	0	1	0	0	0	0
6	0	1	0	0	0	0
7	0	1	0	0	0	0
8	0	1	0	0	0	0
9	0	1	0	0	0	0
10	0	0	0	0	0	0
11	0	0	0	0	0	0
12	0	0	0	0	0	0
13	0	0	0	0	0	0
14	0	0	0	0	0	0
15	0	0	0	0	0	0
16	0	0	0	0	0	0
17	0	0	0	0	0	0
18	0	0	0	0	0	0
19	0	0	0	0	0	0
20	0	0	0	0	0	0
21	0	0	0	0	0	0
22	0	0	0	0	0	0
23	0	0	0	0	0	0
24	0	0	0	0	0	0
25	0	0	0	0	0	0
26	0	0	0	0	0	0
27	0	0	0	0	0	0
28	0	0	0	0	0	0

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
29	0	0	0	0	0	0
...	...	...	...	...	...	...
99968	0	0	0	0	0	0
99969	0	0	0	0	0	0
99970	0	0	0	0	0	0
99971	0	0	0	0	0	0
99972	0	0	0	0	0	0
99973	0	0	0	0	0	0
99974	0	0	0	0	0	0
99975	0	0	0	0	0	0
99976	0	0	0	0	0	0
99977	0	0	0	0	0	0
99979	0	0	0	0	0	0
99980	0	0	0	0	0	0
99981	0	0	0	0	0	0
99982	0	0	0	0	1	0
99983	1	0	0	0	0	0
99984	1	0	0	0	0	0
99985	1	0	0	0	0	0
99986	1	0	0	0	0	0
99987	1	0	0	0	0	0
99988	0	1	0	0	0	0
99989	1	0	0	0	0	0
99990	0	0	0	0	0	0
99991	0	0	0	0	0	0
99992	0	0	0	0	0	0
99993	0	0	0	0	0	0
99994	0	0	0	0	0	0
99995	0	0	0	0	0	0
99996	0	0	0	0	0	0

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag
99997	0	0	0	0	0	0
99998	0	0	0	0	0	0

21344 rows × 6 columns

In [26]:
```
model_df = pd.concat([products , df[['Household Key','Total Cost']]], axis=1)
```

In [27]:
```
model_df
```

Out[27]:

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Household Key	Total Cost
0	0	0	0	0	0	0	10462590.0	NaN
1	0	1	0	0	0	0	4500791.0	32.50
2	0	1	0	0	0	0	4500791.0	30.00
3	0	1	0	0	0	0	4500791.0	32.50
4	0	1	0	0	0	0	4500791.0	30.00
5	0	1	0	0	0	0	4500791.0	53.00
6	0	1	0	0	0	0	4500791.0	30.00
7	0	1	0	0	0	0	4500791.0	32.00
8	0	1	0	0	0	0	4500791.0	32.00
9	0	1	0	0	0	0	4500791.0	32.50
10	0	0	0	0	0	0	4317516.0	NaN
11	0	0	0	0	0	0	11622991.0	58.85
12	0	0	0	0	0	0	11622991.0	53.00
13	0	0	0	0	0	0	11622991.0	53.00
14	0	0	0	0	0	0	11622991.0	NaN
15	0	0	0	0	0	0	11622991.0	NaN
16	0	0	0	0	0	0	11622991.0	53.00
17	0	0	0	0	0	0	11622991.0	53.00
18	0	0	0	0	0	0	11622991.0	53.00
19	0	0	0	0	0	0	11622991.0	29.00
20	0	0	0	0	0	0	579810.0	28.00
21	0	0	0	0	0	0	7187017.0	53.00
22	0	0	0	0	0	0	7187017.0	53.00
23	0	0	0	0	0	0	7187017.0	53.00
24	0	0	0	0	0	0	7187017.0	53.00
25	0	0	0	0	0	0	7187017.0	53.00
26	0	0	0	0	0	0	7187017.0	58.85
27	0	0	0	0	0	0	7728088.0	53.00
28	0	0	0	0	0	0	7728088.0	53.00

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Household Key	Total Cost
29	0	0	0	0	0	0	7728088.0	53.00
...	...	...	...	...	...	...	...	...
99968	0	0	0	0	0	0	1588987.0	NaN
99969	0	0	0	0	0	0	16604128.0	47.00
99970	0	0	0	0	0	0	16604128.0	58.85
99971	0	0	0	0	0	0	16604128.0	58.85
99972	0	0	0	0	0	0	16604128.0	53.00
99973	0	0	0	0	0	0	16604128.0	53.00
99974	0	0	0	0	0	0	16604128.0	58.85
99975	0	0	0	0	0	0	16604128.0	NaN
99976	0	0	0	0	0	0	45466286.0	NaN
99977	0	0	0	0	0	0	45466286.0	34.00
99979	0	0	0	0	0	0	15397653.0	NaN
99980	0	0	0	0	0	0	15397653.0	NaN
99981	0	0	0	0	0	0	15397653.0	NaN
99982	0	0	0	0	1	0	15397653.0	NaN
99983	1	0	0	0	0	0	13735475.0	29.00
99984	1	0	0	0	0	0	13735475.0	44.00
99985	1	0	0	0	0	0	13735475.0	29.00
99986	1	0	0	0	0	0	13735475.0	47.00
99987	1	0	0	0	0	0	13735475.0	44.00
99988	0	1	0	0	0	0	13735475.0	82.00
99989	1	0	0	0	0	0	5462399.0	29.43
99990	0	0	0	0	0	0	5462399.0	NaN
99991	0	0	0	0	0	0	5462399.0	NaN
99992	0	0	0	0	0	0	5462399.0	NaN
99993	0	0	0	0	0	0	20330346.0	53.00
99994	0	0	0	0	0	0	20330346.0	36.00
99995	0	0	0	0	0	0	20330346.0	53.00

	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Household Key	Total Cost
99996	0	0	0	0	0	0	8325571.0	58.85
99997	0	0	0	0	0	0	8325571.0	58.85
99998	0	0	0	0	0	0	8325571.0	NaN

21344 rows × 8 columns

In [28]:
```
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier
```

In [29]:
```
lgr = LogisticRegression()
gbr = GradientBoostingClassifier()
```

In [30]:
```
#model_df_g = model_df.groupby(['Household Key'])['FSV CMSI Flag'].sum()
```

In [31]:
```
#model_df_g
```

In [32]:
```
model_df_g = model_df.groupby(['Household Key']).sum()
```

In [33]:
```
model_df_g = model_df.groupby(['Household Key']).sum()
```

In [34]:
```
mg = model_df_g.dropna()
```

In [35]:
```
mg.shape
```

Out[35]:
```
(5241, 7)
```

In [36]:

```
model_df_g
```

Out[36]:

Household Key	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Total Cost
875.0	0	1	0	0	0	0	1063.20
969.0	0	0	0	0	0	0	226.10
3338.0	0	0	0	0	0	0	0.00
8718.0	0	0	0	0	0	0	0.00
11524.0	0	0	0	0	0	0	294.25
13422.0	0	0	0	0	0	0	118.85
19747.0	0	0	0	0	0	0	0.00
20469.0	1	0	0	0	0	0	537.25
20850.0	0	0	0	0	0	0	0.00
25365.0	0	0	0	0	0	0	0.00
30007.0	0	0	0	0	0	0	34.00
37468.0	0	0	0	0	0	0	0.00
38093.0	1	0	0	0	1	0	555.85
41756.0	6	0	0	0	0	0	518.35
43381.0	0	0	0	0	0	0	102.00
49578.0	0	0	0	0	0	0	30.00
55047.0	0	0	0	2	2	0	60.00
55295.0	0	0	0	0	0	0	0.00
73421.0	0	0	0	0	1	0	0.00
93896.0	0	1	0	0	0	0	130.00
94927.0	0	0	0	0	0	0	0.00
103545.0	0	9	0	0	0	0	390.35
106487.0	0	0	0	0	0	0	178.70
115289.0	0	0	0	0	0	0	0.00
115306.0	0	0	0	0	0	0	0.00
115346.0	0	0	0	0	0	0	0.00
115351.0	0	2	0	0	0	0	38.00

Household Key	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Total Cost
115430.0	0	0	0	0	0	0	53.00
116806.0	15	1	0	0	0	0	2189.60
117430.0	0	0	0	0	0	0	165.00
...	...	...	...	...	...	...	...
99800577.0	0	0	0	0	0	0	454.85
99817387.0	0	0	0	0	0	0	270.85
99817390.0	0	0	0	0	0	0	0.00
99839301.0	0	0	0	0	0	0	0.00
99843098.0	0	0	0	0	0	0	117.70
99851820.0	0	0	0	0	0	0	147.50
99873114.0	0	0	0	0	0	0	318.85
99881116.0	0	0	0	0	0	0	106.00
99953012.0	0	0	0	0	0	0	58.85
99987696.0	2	0	0	0	0	0	53.00
99991498.0	0	0	0	0	0	0	0.00
99992624.0	0	0	0	0	0	0	276.00
99992663.0	0	0	0	0	0	0	122.00
99993288.0	0	0	0	0	0	0	229.55
99996562.0	0	0	0	0	0	0	265.00
100004477.0	0	0	0	0	0	0	0.00
100016608.0	0	0	0	0	0	0	323.39
100020029.0	0	0	0	0	0	0	53.00
100022741.0	0	0	0	0	0	0	0.00
100023243.0	0	0	0	0	0	0	0.00
100035899.0	1	0	0	0	0	0	0.00
100053546.0	0	0	0	0	0	0	53.00
100064720.0	0	2	0	0	0	0	54.00
100065197.0	0	0	0	0	0	0	297.35

Household Key	FSV CMSI Flag	FSV Credit Card Flag	FSV Deposit Program Flag	FSV Home Equity Flag	FSV ID Theft Flag	FSV Mortgage Flag	Total Cost
100067809.0	0	0	0	0	0	0	212.00
100069201.0	0	0	0	0	0	0	106.00
100070004.0	0	0	0	0	0	0	60.00
100071861.0	0	0	0	0	0	0	447.40
100071870.0	0	0	0	0	0	0	211.00
100079136.0	14	0	0	0	0	0	771.75

5241 rows × 7 columns

Apply classificaiton model (Logistics and GredeintBoost) on 'FSV CMSI Flag' and ' FSV Credit Card Flag' and compare the gain lift in each model.

In [37]:
```
X = model_df_g[['Total Cost']]
```

In [38]:
```
y = model_df_g[['FSV CMSI Flag']]
```

In [39]:
```
y = np.where(y>0,1,0)
```

In [41]:
```
from sklearn.model_selection import train_test_split
```

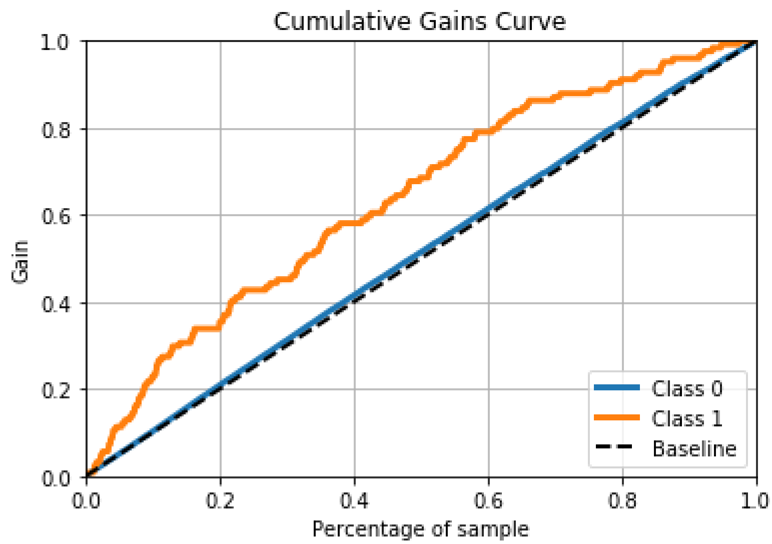
In [42]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

In [43]:
```
lgr.fit(X_train, y_train)

predicted_probas = lgr.predict_proba(X_test)
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarn
ing: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exa
mple using ravel().
  return f(**kwargs)
```

In [44]:
```
# The magic happens here

import matplotlib.pyplot as plt

import scikitplot as skplt

skplt.metrics.plot_cumulative_gain(y_test, predicted_probas)

plt.show()
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:424: MatplotlibDeprec
ationWarning:
Passing one of 'on', 'true', 'off', 'false' as a boolean is deprecated; use an actual boolean (True/False) instead.
  warn_deprecated("2.2", "Passing one of 'on', 'true', 'off', 'false' as a "
```

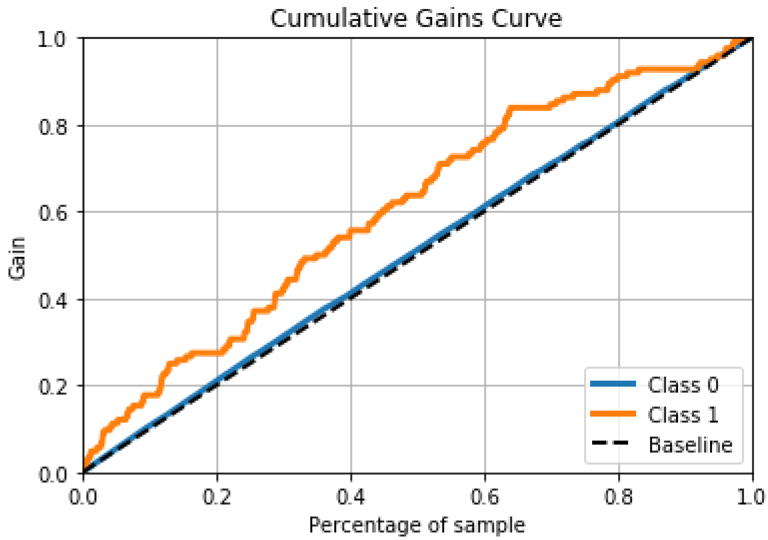


In [45]:
```
gbr.fit(X_train, y_train)

predicted_probas = gbr.predict_proba(X_test)
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarn
ing: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exa
mple using ravel().
  return f(**kwargs)
```

In [46]:
```
skplt.metrics.plot_cumulative_gain(y_test, predicted_probas)

plt.show()
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:424: MatplotlibDeprec
ationWarning:
Passing one of 'on', 'true', 'off', 'false' as a boolean is deprecated; use an actual boolean (True/False) instead.
  warn_deprecated("2.2", "Passing one of 'on', 'true', 'off', 'false' as a "
```



In [47]:
```
lgr.score(X,y)
```

Out[47]:
```
0.9297843922915474
```

In [48]:
```
gbr.score(X,y)
```

Out[48]:
```
0.9328372448006106
```

In [49]:
```
y = model_df_g[['FSV Credit Card Flag']]
```

In [50]:
```
y = np.where(y>0,1,0)
```

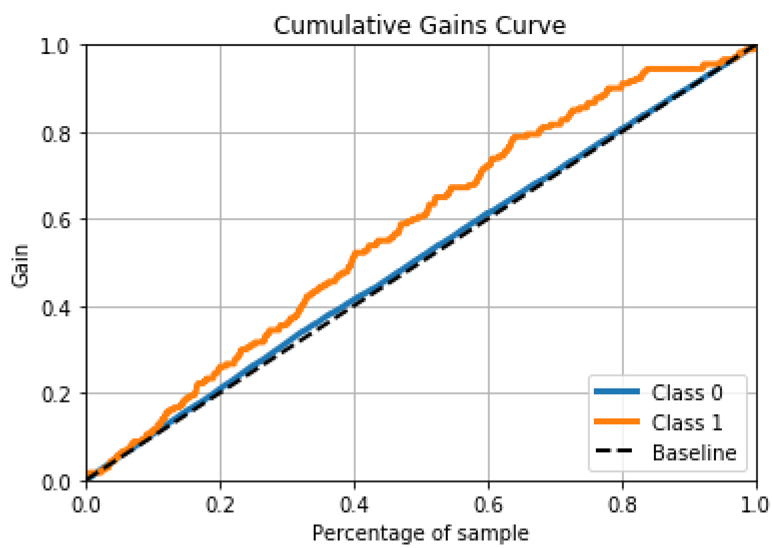
In [51]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33)
```

In [52]:
```
lgr.fit(X_train, y_train)

predicted_probas = lgr.predict_proba(X_test)
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarn
ing: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exa
mple using ravel().
  return f(**kwargs)
```

In [53]:
```
skplt.metrics.plot_cumulative_gain(y_test, predicted_probas)

plt.show()
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:424: MatplotlibDeprec
ationWarning:
Passing one of 'on', 'true', 'off', 'false' as a boolean is deprecated; use an actual boolean (True/False) instead.
  warn_deprecated("2.2", "Passing one of 'on', 'true', 'off', 'false' as a "
```



In [54]:
```
lgr.score(X,y)
```

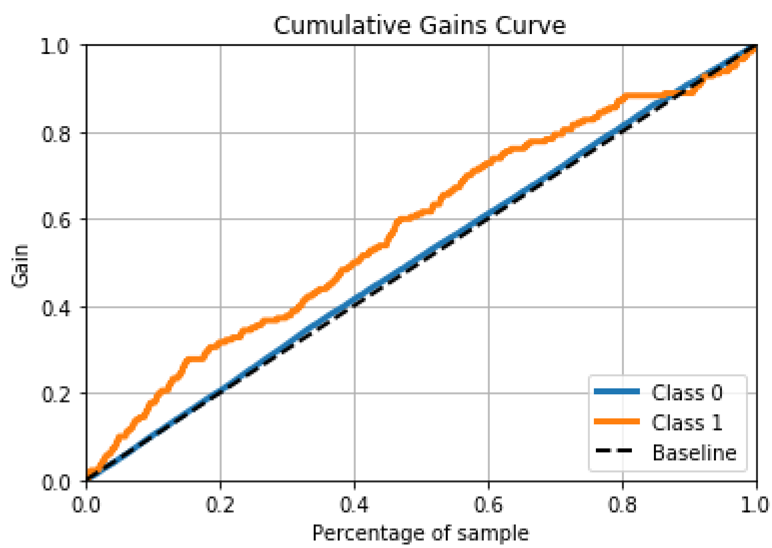
Out[54]: 0.8895248998282771

In [55]:
```
gbr.fit(X_train, y_train)

predicted_probas = gbr.predict_proba(X_test)
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\sklearn\utils\validation.py:73: DataConversionWarn
ing: A column-vector y was passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for exa
mple using ravel().
  return f(**kwargs)
```

In [56]:
```
skplt.metrics.plot_cumulative_gain(y_test, predicted_probas)

plt.show()
```

```
C:\Users\george\AppData\Local\Continuum\anaconda3\lib\site-packages\matplotlib\cbook\__init__.py:424: MatplotlibDeprec
ationWarning:
Passing one of 'on', 'true', 'off', 'false' as a boolean is deprecated; use an actual boolean (True/False) instead.
  warn_deprecated("2.2", "Passing one of 'on', 'true', 'off', 'false' as a "
```



In [57]:
```
gbr.score(X,y)
```

Out[57]: 0.8912421293646251

In [63]:
```
model_df_g['FSV Credit Card Flag'].value_counts()
```

Out[63]:
```
0     4663
1      341
2       84
3       65
4       29
6       15
5       12
7        8
9        8
8        4
11       3
10       2
13       2
12       2
15       1
14       1
17       1
Name: FSV Credit Card Flag, dtype: int64
```

In [65]:
```
model_data = np.where(model_df_g>0,1,0)
```

In [68]:
```
model_data
```

Out[68]:
```
array([[0, 1, 0, ..., 0, 0, 1],
       [0, 0, 0, ..., 0, 0, 1],
       [0, 0, 0, ..., 0, 0, 0],
       ...,
       [0, 0, 0, ..., 0, 0, 1],
       [0, 0, 0, ..., 0, 0, 1],
       [1, 0, 0, ..., 0, 0, 1]])
```

Drawing ROC curve

One of the most commonly used metrics nowadays is AUC-ROC (Area Under Curve - Receiver Operating Characteristics) curve. ROC curves are pretty easy to understand and evaluate once there is a good understanding of confusion matrix and different kinds of errors.

In [87]:
```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

In [88]:
```
def plot_roc_curve(fpr, tpr):
    plt.plot(fpr, tpr, color='orange', label='ROC')
    plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic (ROC) Curve')
    plt.legend()
    plt.show()
```

In [96]:
```
probs = lgr.predict_proba(X_test)
```

In [97]:
```
probs
```

Out[97]:
```
array([[0.89739947, 0.10260053],
       [0.86989481, 0.13010519],
       [0.89321462, 0.10678538],
       ...,
       [0.89739947, 0.10260053],
       [0.89479453, 0.10520547],
       [0.89739947, 0.10260053]])
```

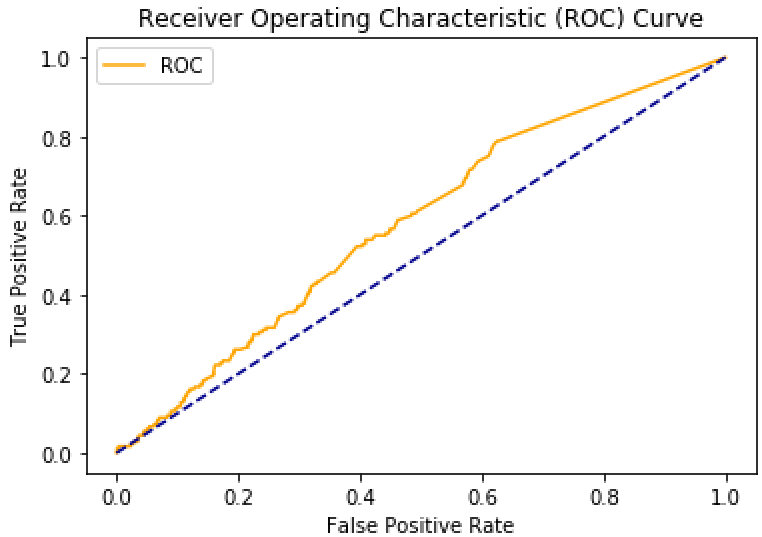
In [98]:
```
probs = probs[:, 1]
```

In [99]:
```
auc = roc_auc_score(y_test, probs)
print('AUC: %.2f' % auc)
```

```
AUC: 0.58
```

In [100]:
```
fpr, tpr, thresholds = roc_curve(y_test, probs)
```

In [101]:
```
plot_roc_curve(fpr, tpr)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [89]:
```
probs = gbr.predict_proba(X_test)
```

In [90]:
```
probs
```

Out[90]:
```
array([[0.93032273, 0.06967727],
       [0.81500049, 0.18499951],
       [0.85242294, 0.14757706],
       ...,
       [0.93032273, 0.06967727],
       [0.81036009, 0.18963991],
       [0.93032273, 0.06967727]])
```

In [91]:
```
probs = probs[:, 1]
```

In [92]:
```
probs
```

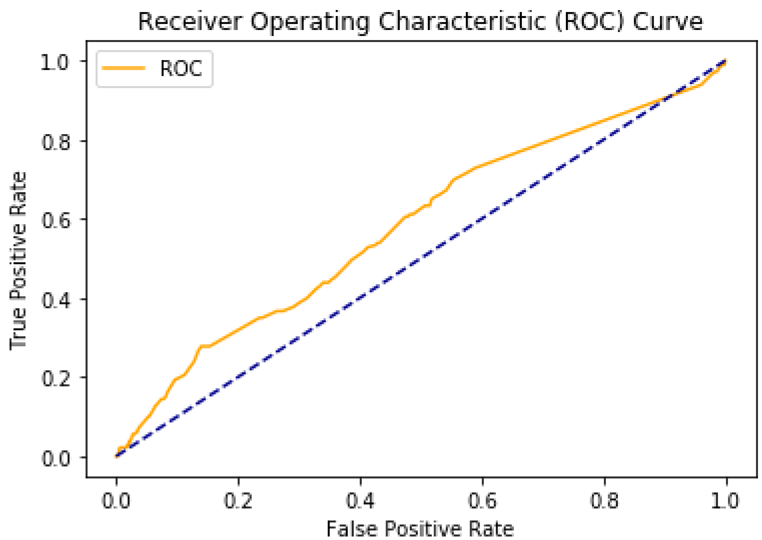
Out[92]:
```
array([0.06967727, 0.18499951, 0.14757706, ..., 0.06967727, 0.18963991,
       0.06967727])
```

In [93]:
```
auc = roc_auc_score(y_test, probs)
print('AUC: %.2f' % auc)
```

```
AUC: 0.58
```

In [94]:
```
fpr, tpr, thresholds = roc_curve(y_test, probs)
```

In [95]:
```
plot_roc_curve(fpr, tpr)
```



We can see that ROC AUC score of both classifiers are same. But the ROC curve of Gradient Boosting Regression has advantage as there is a clear improvement of ROC before 20% in True positive cases.

AUC-ROC curve is one of the most commonly used metrics to evaluate the performance of machine learning algorithms particularly in the cases where we have imbalanced datasets.

In [ ]:

```

```

In [ ]:

```

```