225229140 pml lab 6

step 1

In [1]:

```
import pandas as pd
from sklearn.linear_model import LogisticRegression
```

In [2]:

```
data=pd.read_csv('diabetes.csv')
data
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

In [3]:

```
data.head
```

Out[3]:

```
<bound method NDFrame.head of      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age  Outcome
0                       0.627   50        1
1                       0.351   31        0
2                       0.672   32        1
3                       0.167   21        0
4                       2.288   33        1
..                        ...  ...      ...
763                     0.171   63        0
764                     0.340   27        0
765                     0.245   30        0
766                     0.349   47        1
767                     0.315   23        0

[768 rows x 9 columns]>
```

In [4]:

```
data.shape
```

Out[4]:

```
(768, 9)
```

In [5]:

```
data.columns
```

Out[5]:

```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [6]:

```
data.dtypes
```

Out[6]:

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
```

In [7]:

```
data.info
```

Out[7]:

```
<bound method DataFrame.info of      Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6   
1              1       85             66             29        0  26.6   
2              8      183             64              0        0  23.3   
3              1       89             66             23       94  28.1   
4              0      137             40             35      168  43.1   
..           ...      ...            ...            ...      ...   ...   
763           10      101             76             48      180  32.9   
764            2      122             70             27        0  36.8   
765            5      121             72             23      112  26.2   
766            1      126             60              0        0  30.1   
767            1       93             70             31        0  30.4   

     DiabetesPedigreeFunction  Age  Outcome  
0                       0.627   50        1  
1                       0.351   31        0  
2                       0.672   32        1  
3                       0.167   21        0  
4                       2.288   33        1  
..                        ...  ...      ...  
763                     0.171   63        0  
764                     0.340   27        0  
765                     0.245   30        0  
766                     0.349   47        1  
767                     0.315   23        0  

[768 rows x 9 columns]>
```

In [8]:

```
data.Pregnancies.value_counts()
```

Out[8]:

```
1     135
0     111
2     103
3      75
4      68
5      57
6      50
7      45
8      38
9      28
10     24
11     11
13     10
12      9
14      2
15      1
17      1
Name: Pregnancies, dtype: int64
```

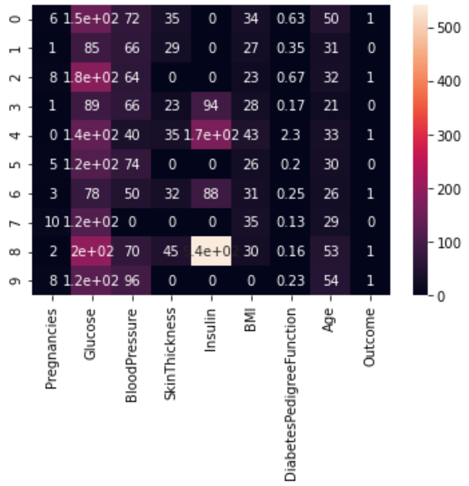
step 2

In [9]:

```
import seaborn as sns
sns.heatmap(data.head(10), annot=True)
```

Out[9]:

```
<AxesSubplot:>
```



step 3

In [10]:

```
X = data[['Age']]
y = data[['Outcome']]
```

In [11]:

```
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=42)
```

In [12]:

```
X_train
```

Out[12]:

	Age
357	44
73	23
352	46
497	25
145	21
...	...
71	26
106	27
270	38
435	29
102	21

576 rows × 1 columns

In [13]:

```
from sklearn import linear_model
logr=linear_model.LogisticRegression()
logr.fit(X,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[13]:

```
LogisticRegression()
```

In [14]:

```
df=logr.predict(X_test)
df
```

Out[14]:

```
array([0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0,
       0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [15]:

```
print("coef_ : ",logr.coef_)
print("intercept_: ",logr.intercept_)
```

```
coef_ :  [[0.04202466]]
intercept_:  [-2.04744865]
```

In [16]:

```
logr.predict([[60]])
```

Out[16]:

```
array([1], dtype=int64)
```

In [17]:

```
lrf=logr.coef_ * 60 + logr.intercept_
from scipy.special import expit
d=expit(lrf)
```

In [18]:

```
if d > 0.5:
    print('Yes, he will become diabetic ')
else:
    print('No, he will not be diabetic')
```

```
Yes, he will become diabetic
```

step 4

In [19]:

```
X1=data[['Glucose','BMI','Age']]
```

In [20]:

```
from sklearn import linear_model

X1_train,X1_test,y1_train,y1_test = train_test_split(X1,y,random_state=42,test_size=0.24)
logr1 =linear_model.LogisticRegression()
logr1.fit(X1_train,y1_train)
logr1.predict(X1_test)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[20]:

```
array([0, 0, 0, 0, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 0,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 1], dtype=int64)
```

In [21]:

```
print("coef_ : ",logr1.coef_)
print("intercept_: ",logr1.intercept_)
```

```
coef_ :  [[0.03292234 0.09635698 0.04398021]]
intercept_:  [-9.39683405]
```

In [22]:

```
lrf1=logr1.coef_ * 150 * 30 * 40+ logr1.intercept_
from scipy.special import expit
expit(lrf1)
```

Out[22]:

```
array([[1., 1., 1.]])
```

In [23]:

```
logr1.predict([[150,30,40]])
```

Out[23]:

```
array([1], dtype=int64)
```

In [24]:

```
logr1.predict_proba([[150,30,40]])
```

Out[24]:

```
array([[0.45228691, 0.54771309]])
```

step 5

In [25]:

```
X2=data.drop(['Outcome'],axis=1)
X2_train,X2_test,y2_train,y2_test = train_test_split(X2,y,test_size=.25,random_state=42)
from sklearn import linear_model
logr2=LogisticRegression()
logr2.fit(X2_train,y2_train)
df1=logr2.predict(X2_test)
df1
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to conve
rge (status=1):
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 (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 (https://scikit-learn.org/stable/modules/
linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[25]:

```
array([0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 1, 0,
       0, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1,
       0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0,
       0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [26]:

```
from sklearn.metrics import roc_auc_score
lor_auc=roc_auc_score(y2_test,df1)
print("Auc:",lor_auc)
```

```
Auc: 0.7122658183103571
```

step 6

In [27]:

```
def get_auc(var,tar,df):
    fx = df[var]
    fy = df[tar]
    logr3=LogisticRegression()
    logr3.fit(fx,fy)
    pred=logr3.predict_proba(fx)[:,1]
    auc_val = roc_auc_score(y,pred)
    return auc_val
get_auc(['Glucose',"BMI"],['Outcome'],data)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[27]:

```
0.8109328358208956
```

In [28]:

```
get_auc(['Pregnancies','BloodPressure','SkinThickness'],['Outcome'],data)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[28]:

```
0.6444962686567164
```

In [29]:

```
def best_next(current,cand,tar,data):
    best_auc=-1
    best_var=None
    for i in cand:
        auc_v = get_auc(current+[i],tar,data)
        if auc_v>=best_auc:
            best_auc=auc_v
            best_var=i
        return best_var
```

In [30]:

```
current=['Insulin','BMI','DiabetesPedigreeFunction','Age']
cand=['Pregnancies','Glucose','BloodPressure','SkinThickness']
tar=['Outcome']
next_var = best_next(current,cand,tar,data)
next_var
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
```

Out[30]:

```
'Pregnancies'
```

In [31]:

```
tar =['Outcome']
current=[]
cand=['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','DiabetesPedigreeFunction','Age']
max_num = 7
num_it = min(max_num,len(cand))
for i in range(0,num_it):
    next_var = best_next(current,cand,tar,data)
    current += [next_var]
    cand.remove(next_var)
    print("variable addd in step "+str(i+1)+' is '+ next_var +" .")
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)

variable addd in step 1 is Pregnancies .
variable addd in step 2 is Glucose .
variable addd in step 3 is BloodPressure .
variable addd in step 4 is SkinThickness .

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to conve
rge (status=1):
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 (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 (https://scikit-learn.org/stable/modules/
linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(

variable addd in step 5 is Insulin .
variable addd in step 6 is BMI .
variable addd in step 7 is DiabetesPedigreeFunction .
```

In [32]:

```
print(current)
```

```
['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction']
```

step 7

In [33]:

```
X2_train,X2_test,y2_train,y2_test = train_test_split(X2,y,stratify=y,test_size=.5,random_state=42)
```

In [34]:

```
prediction=logr2.predict_proba(X2_test)
```

In [35]:

```
train = pd.concat([X2_train,y2_train],axis =1)
test = pd.concat([X2_test,y2_test],axis =1)
def auc_train_test (variables,target, train, test):
    X_train = train[variables]
    X_test = test[variables]
    Y_train =train[target]
    Y_test = test[target]
    Lor=LogisticRegression()
    Lor.fit(X_train,Y_train)
    prediction_train = Lor.predict_proba(X_train)[:,1]
    prediction_test = Lor.predict_proba(X_test)[:,1]
    auc_train = roc_auc_score(Y_train, prediction_train)
    auc_test = roc_auc_score(Y_train,prediction_test)
    return (auc_train,auc_test)
auc_values_train=[]
auc_values_test=[]
variable_evaluate=[]
for v in X2.columns:
    variable_evaluate.append(v)
    auc_train,auc_test = auc_train_test(variable_evaluate,['Outcome'],train,test)
    auc_values_train.append(auc_train)
    auc_values_test.append(auc_test)
```

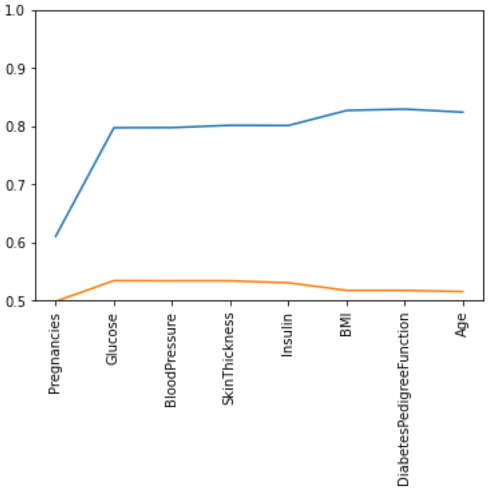
```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:63: DataConversionWarning: A column-vector y was pas
sed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  return f(*args, **kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.py:763: ConvergenceWarning: lbfgs failed to conve
rge (status=1):
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 (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 (https://scikit-learn.org/stable/modules/
linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

In [ ]:

In [ ]:

In [37]:

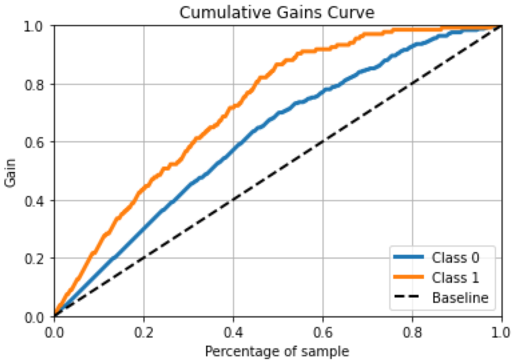
```
import matplotlib.pylab as plt
import numpy as np
x =np.array(range(0,len(auc_values_train)))
my_train = np.array(auc_values_train)
my_test = np.array(auc_values_test)
plt.xticks(x,X2.columns,rotation=90)
plt.plot(x,my_train)
plt.plot(x,my_test)
plt.ylim(0.5,1)
plt.show()
```



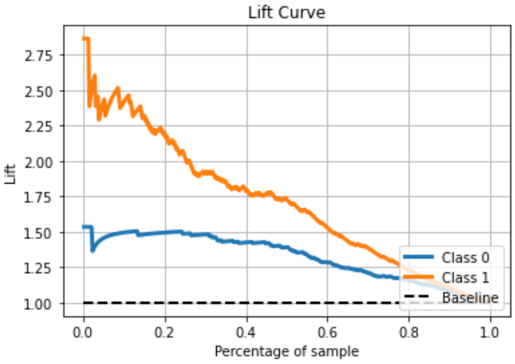
step 8

In [38]:

```
import scikitplot as skplt
skplt.metrics.plot_cumulative_gain(y2_test,prediction)
plt.show()
plt.figure(figsize=(7,7))
skplt.metrics.plot_lift_curve(y2_test, prediction)
plt.show()
```



```
<Figure size 504x504 with 0 Axes>
```



In [ ]: