Name : Arul kumar ARK

Roll No. : 22522913

Lab : 6

```
Predictive Analytics for Hospitals
```

Step : 1

In [1]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv('diabetes.csv')
```

In [3]:
```
data.head(10)
```

Out[3]:

	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
5	5	116	74	0	0	25.6	0.201	30	0
6	3	78	50	32	88	31.0	0.248	26	1
7	10	115	0	0	0	35.3	0.134	29	0
8	2	197	70	45	543	30.5	0.158	53	1
9	8	125	96	0	0	0.0	0.232	54	1

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.DiabetesPedigreeFunction.value_counts
```

Out[7]:
```
<bound method IndexOpsMixin.value_counts of 0      0.627
1      0.351
2      0.672
3      0.167
4      2.288
       ...
763    0.171
764    0.340
765    0.245
766    0.349
767    0.315
Name: DiabetesPedigreeFunction, Length: 768, dtype: float64>
```

In [8]:
```
1 data.describe()
```

Out[8]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	120.894531	69.105469	20.536458	79.799479	31.992578	0.471876	33.240885	0.348958
std	3.369578	31.972618	19.355807	15.952218	115.244002	7.884160	0.331329	11.760232	0.476951
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	62.000000	0.000000	0.000000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	30.500000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

step : 2

In [9]:
```
1 import numpy as np
2 import seaborn as sns
3 import matplotlib.pyplot as plt
```

In [10]:
```
1 data.drop_duplicates()
2
```

Out[10]:

	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 [54]:
```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
sns.heatmap(data.head(10), cmap='BuPu',annot=True, linewidth=.10)
```

Out[54]: <AxesSubplot:>

step : 3

In [12]:
```
X=data["Age"]
y=data['Outcome']
```

In [13]:
```
X = np.array(X).reshape(-1, 1)
```

In [14]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [15]:
```
X_train,x_test,Y_train,y_test = train_test_split(X,y,test_size=.25,random_state=11)
```

In [16]:
```
logr = LogisticRegression()
logr.fit(X_train,Y_train)
```

Out[16]: LogisticRegression()

In [17]:
```
logr.predict(x_test)
```

Out[17]:
```
array([0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 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, 1, 1, 0, 0, 0, 0, 0, 1, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,
       1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 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, 1, 0, 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, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0], dtype=int64)
```

In [18]:
```
print("Coef :",logr.coef_)
print("Intercept :",logr.intercept_)
```

```
Coef : [[0.04278121]]
Intercept : [-2.06807344]
```

In [19]:
```
logr.predict([[60]])
```

Out[19]:
```
array([1], dtype=int64)
```

In [20]:
```
lrf = logr.coef_*60 + logr.intercept_
```

In [21]:
```
from scipy.special import expit
```

In [22]:
```
output=expit(lrf)
```

In [23]:
```
output
```

Out[23]:
```
array([[0.62217709]])
```

In [24]:
```
if output > 0.5 :
    print(" YES he will become diabetic")
else :
     print(" NO he will not be diabetic")
```

```
 YES he will become diabetic
```

step : 4

In [25]:
```
X=data[['Glucose','BMI','Age']]
y=data['Outcome']
```

In [26]:
```
X_train,x_test,Y_train,y_test = train_test_split(X,y,test_size=.25,random_state=11)
```

In [27]:
```
logr = LogisticRegression()
logr.fit(X_train,Y_train)
```

Out[27]:
```
LogisticRegression()
```

In [28]:
```
logr.predict(x_test)
```

Out[28]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [29]:
```
print("Coef :",logr.coef_)
print("Intercept :",logr.intercept_)
```

```
Coef : [[0.03358049 0.07889299 0.02722911]]
Intercept : [-8.37441801]
```

In [30]:
```
logr.predict([[150,30,40]])
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but Logist
icRegression was fitted with feature names
  warnings.warn(
```

Out[30]:
```
array([1], dtype=int64)
```

In [31]:
```
lrf = logr.coef_*150*30*40 + logr.intercept_
```

In [32]:
```
output=expit(lrf)
```

In [33]:
```
output
```

Out[33]:
```
array([[1., 1., 1.]])
```

In [34]:
```
output1=logr.predict_proba([[150,30,40]])
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LogisticRegression was fitted with feature names
  warnings.warn(
```

In [35]:
```
output1[0] > 0.5
```

Out[35]:
```
array([False,  True])
```

In [36]:
```
output1
```

Out[36]:
```
array([[0.47038229, 0.52961771]])
```

Step : 5

In [37]:
```
X1=data.drop(['Outcome'],axis=1)
```

In [38]:
```
X_train,x_test,Y_train,y_test = train_test_split(X,y,test_size=.25,random_state=11)
```

In [39]:
```
LOR=LogisticRegression()
LOR.fit(X_train,Y_train)
```

Out[39]:
```
LogisticRegression()
```

In [40]:
```
LOR.predict(x_test)
```

Out[40]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 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, 1, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [41]:
```
y_pred=logr.predict(x_test)
y_pred
```

Out[41]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 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, 1, 1, 0, 1, 0, 1, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,
       1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1,
       0, 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [42]:
```
from sklearn.metrics import roc_auc_score
lor_auc=roc_auc_score(y_test,y_pred)

```

In [43]:
```
print("Auc:",lor_auc)
```

```
Auc: 0.6756854256854258
```

step : 6

In [44]:
```
def get_auc(var,tar,data):
    fx=data[var]
    fy=data[tar]
    LOR4=LogisticRegression()
    LOR4.fit(fx,fy)
    pred=LOR4.predict_proba(fx)[:,1]
    auc_val = roc_auc_score(y,pred)
    return auc_val
```

In [45]:
```
get_auc(['Glucose','BMI'],['Outcome'],data)
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
```

Out[45]: 0.8109328358208956

In [46]:
```
get_auc(['Pregnancies','BloodPressure','SkinThickness'],['Outcome'],data)
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
```

Out[46]: 0.6444962686567164

In [47]:
```
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 [48]:
```
current=['Insulin','BMI','DiabetesPedigreeFunction','Age']
cand=['Pregnancies','Glucose','BloodPressure','SkinThickness']
tar=['Outcome']
next_var = best_next(current,cand,tar,data)
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
```

In [49]:
```
next_var
```

Out[49]: 'Glucose'

In [50]:

```python
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 add in step "+str(i+1)+' is '+ next_var +" .")
print(current)
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

variable add in step 3 is Pregnancies .

C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
```

step : 7

In [58]:

```python
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.5,stratify =y)
```

In [60]:

```python
pred2 = logr.predict_proba(X_test)
```

In [61]:

```python
train = pd.concat([X_train,y_train], axis=1)
test = pd.concat([X_test,y_test], axis=1)
```

In [62]:

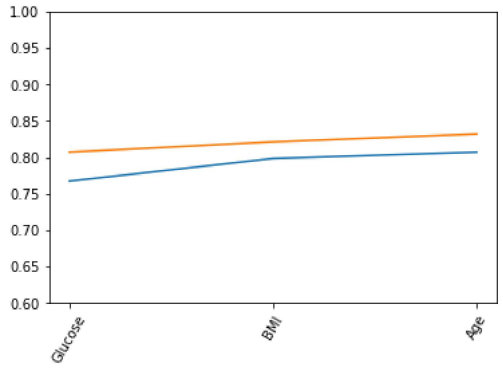
```python
def auc_train_test(variables,target,train,test):
    X_train = train[variables]
    X_test = test[variables]
    Y_train = train[target]
    Y_test = test[target]
    logreg = LogisticRegression()
    logreg.fit(X_train, Y_train)
    predictions_train = logreg.predict_proba(X_train)[:,1]
    predictions_test = logreg.predict_proba(X_test)[:,1]
    auc_train = roc_auc_score(Y_train, predictions_train)
    auc_test = roc_auc_score(Y_test,predictions_test)
    return(auc_train, auc_test)
```

In [64]:

```python
auc_values_train = []
auc_values_test = []
variables_evaluate = []
for v in X.columns:
    variables_evaluate.append(v)
    auc_train, auc_test = auc_train_test(variables_evaluate,["Outcome"],train,test)
    auc_values_train.append(auc_train)
    auc_values_test.append(auc_test)
```

```
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
C:\Users\arulk\anaconda3\lib\site-packages\sklearn\utils\validation.py:993: 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().
  y = column_or_1d(y, warn=True)
```

In [69]:
```
import matplotlib.pyplot 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,X.columns,rotation=60)
plt.plot(x,my_train)
plt.plot(x,my_test)
plt.ylim((0.6,1.0))
plt.show()
```

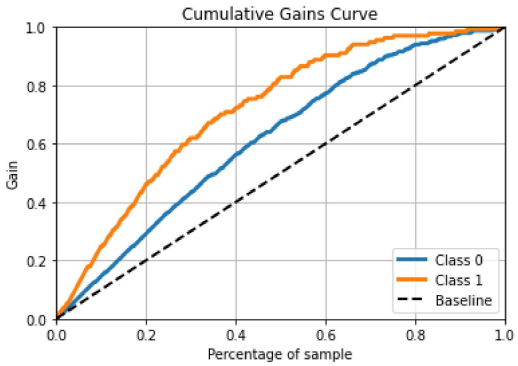


Step : 8

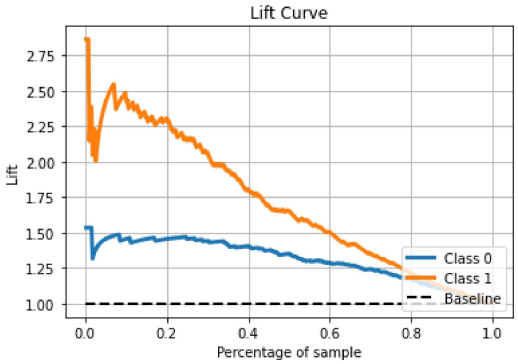
In [70]:
```
!pip install scikit-plot
from scikitplot.estimators import plot_feature_importances
from scikitplot.metrics import plot_confusion_matrix, plot_roc
```

```
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: matplotlib>=1.4.0 in c:\users\arulk\anaconda3\lib\site-packages (from scikit-plot) (3.5.1)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\arulk\anaconda3\lib\site-packages (from scikit-plot) (1.0.2)
Requirement already satisfied: scipy>=0.9 in c:\users\arulk\anaconda3\lib\site-packages (from scikit-plot) (1.7.3)
Requirement already satisfied: joblib>=0.10 in c:\users\arulk\anaconda3\lib\site-packages (from scikit-plot) (1.1.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->s
cikit-plot) (2.8.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-p
lot) (9.0.1)
Requirement already satisfied: numpy>=1.17 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plo
t) (1.21.5)
Requirement already satisfied: packaging>=20.0 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit
-plot) (21.3)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scik
it-plot) (1.3.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scik
it-plot) (4.25.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->sciki
t-plot) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\arulk\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-pl
ot) (0.11.0)
Requirement already satisfied: six>=1.5 in c:\users\arulk\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib
>=1.4.0->scikit-plot) (1.16.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\arulk\anaconda3\lib\site-packages (from scikit-learn>=0.18->
scikit-plot) (2.2.0)
Installing collected packages: scikit-plot
Successfully installed scikit-plot-0.3.7
```

In [71]:
```
import scikitplot as skplt
skplt.metrics.plot_cumulative_gain(y_test,pred2)
plt.show()
plt.figure(figsize=(7,7))
skplt.metrics.plot_lift_curve(y_test,pred2)
plt.show()
```




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



In [ ]:
```
1
```