Roll no : 225229113

Lab6.Predictive Analysis for Hospitals

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

Step1. Import dataset

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

Out[2]:

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

768 rows × 9 columns

In [3]:
```
#head
df.head(5)
```

Out[3]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	A
0	6	148	72	35	0	33.6	0.627	
1	1	85	66	29	0	26.6	0.351	
2	8	183	64	0	0	23.3	0.672	
3	1	89	66	23	94	28.1	0.167	
4	0	137	40	35	168	43.1	2.288	

In [4]:
```
#shape
df.shape
```

Out[4]:
```
(768, 9)
```

In [5]:
```
#columns
df.columns
```

Out[5]:
```
Index(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin',
       'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome'],
      dtype='object')
```

In [6]:
```
#dtype
df.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]:
```
#info
df.info
```

Out[7]:
```
<bound method DataFrame.info of      Pregnancies  Glucose  BloodPressure  SkinT
hickness  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]:
```
#value_counts
df.Glucose.value_counts
```

Out[8]:
```
<bound method IndexOpsMixin.value_counts of 0      148
1       85
2      183
3       89
4      137
      ...
763    101
764    122
765    121
766    126
767     93
Name: Glucose, Length: 768, dtype: int64>
```

Step2. Identify relationships between feature

In [9]:
```
df.style.background_gradient(cmap = 'GnBu')
```

Out[9]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigree
0	6	148	72	35	0	33.600000	
1	1	85	66	29	0	26.600000	
2	8	183	64	0	0	23.300000	
3	1	89	66	23	94	28.100000	
4	0	137	40	35	168	43.100000	
5	5	116	74	0	0	25.600000	
6	3	78	50	32	88	31.000000	
7	10	115	0	0	0	35.300000	
8	2	197	70	45	543	30.500000	
9	8	125	96	0	0	0.000000	
10	4	110	92	0	0	37.600000	

Step3. Prediction using one feature

In [10]:
```
X=df[['Age']]
Y=df[['Outcome']]
```

In [11]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [12]:
```
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=.25,random_state=1
```

In [13]:
```
logr = LogisticRegression()
logr.fit(X_train,Y_train)
```

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

Out[13]:
```
LogisticRegression()
```

In [14]:
```
logr.predict(X_test)
```

Out[14]:
```
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, 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, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0], dtype=int64)
```

In [15]:
```
print("Coef :",logr.coef_)
print("Intercept :",logr.intercept_)
```

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

In [16]:
```
logr.predict([[60]])
```

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

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
```

Step4.Prediction using many features

In [19]:
```
X1=df[['Glucose','BMI','Age']]
X1_train,X1_test,Y1_train,Y1_test = train_test_split(X1,Y,test_size=.25,random_st
logr1= LogisticRegression()
logr1.fit(X1_train,Y1_train)
```

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

Out[19]:
```
LogisticRegression()
```

In [20]:
```
logr1.predict(X1_test)
```

Out[20]:
```
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, 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, 0, 1, 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 [21]:
```
print("Coef :",logr1.coef_)
print("Intercept :",logr1.intercept_)
```

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

In [22]:
```
lrf1=logr1.coef_*150*40*30+logr1.intercept_
from scipy.special import expit
expit(lrf1)
```

Out[22]:
```
array([[1., 1., 1.]])
```

In [23]:
```
logr1.predict([[150,40,30]])
```

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

Out[23]:
```
array([1], dtype=int64)
```

In [24]:
```
logr1.predict_proba([[150,40,30]])
```

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

Out[24]:
```
array([[0.34632141, 0.65367859]])
```

Step5. Build LoR model with all features

In [25]:
```
X2=df.drop(['Outcome'],axis=1)
X2_train,X2_test,Y2_train,Y2_test = train_test_split(X2,Y,test_size=.25,random_st
logr2=LogisticRegression()
logr2.fit(X2_train,Y2_train)
logr2.predict(X2_test)
```

```
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:993:
DataConversionWarning: A column-vector y was passed when a 1d array was expecte
d. Please change the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\linear_model\_logistic.p
y:814: ConvergenceWarning: lbfgs failed to converge (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-regressi
on (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regressi
on)
  n_iter_i = _check_optimize_result(
```

Out[25]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 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, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 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,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 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, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [26]:
```
Y2_pred=logr2.predict(X2_test)
Y2_pred
```

Out[26]:
```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 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, 1, 0, 0, 0,
       0, 0, 0, 0, 1, 0, 0, 0, 0, 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,
       0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 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, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0,
       0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0,
       0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0], dtype=int64)
```

In [27]:
```
from sklearn.metrics import roc_auc_score
lor_auc=roc_auc_score(Y2_test,Y2_pred)
print("Auc:",lor_auc)
```

```
Auc: 0.7142857142857143
```

Step6. Forward selection prodcure

In [28]:
```
#get_auc

def get_auc(var,tar,df):
    fx = df[var]
    fy = df[tar]
    logr4=LogisticRegression()
    logr4.fit(fx,fy)
    pred=logr4.predict_proba(fx)[:,1]
    auc_val = roc_auc_score(Y,pred)
    return auc_val
get_auc(['Glucose',"BMI"],['Outcome'],df)
```

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

Out[28]:
```
0.8109328358208956
```

In [29]:
```
get_auc(['Pregnancies','BloodPressure','SkinThickness'],['Outcome'],df)
```

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

Out[29]:
```
0.6444962686567164
```

In [30]:

```
#fuction of best_next

def best_next(current,cand,tar,df):
    best_auc=-1
    best_var=None
    for i in cand:
        auc_v = get_auc(current+[i],tar,df)
        if auc_v>=best_auc:
            best_auc=auc_v
            best_var=i
    return best_var
current=['Insulin','BMI','DiabetesPedigreeFunction','Age']
cand=['Pregnancies','Glucose','BloodPressure','SkinThickness']
tar=['Outcome']
next_var = best_next(current,cand,tar,df)
next_var
```

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

Out[30]: 'Glucose'

In [31]:
```
tar =['Outcome']
current=[]
cand=['Pregnancies','Glucose','BloodPressure','SkinThickness','Insulin','BMI','Di
max_num=7
num_it = min(max_num,len(cand))
for i in range(0,num_it):
    next_var = best_next(current,cand,tar,df)
    current += [next_var]
    cand.remove(next_var)
    print("variable added in step "+str(i+1)+' is '+ next_var +" .")
```

```
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
```

In [32]:
```
current
```

Out[32]:
```
['Glucose',
 'BMI',
 'Pregnancies',
 'DiabetesPedigreeFunction',
 'BloodPressure',
 'Age',
 'SkinThickness']
```

step-7 Plot line graph of AUC values and select cut-off

In [33]:
```
X2_train,X2_test,Y2_train,Y2_test = train_test_split(X2,Y,stratify=Y,test_size=.5
```

In [34]:
```
prediction=logr2.predict_proba(X2_test)
```

In [36]:
```
#plot auc values
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:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
```

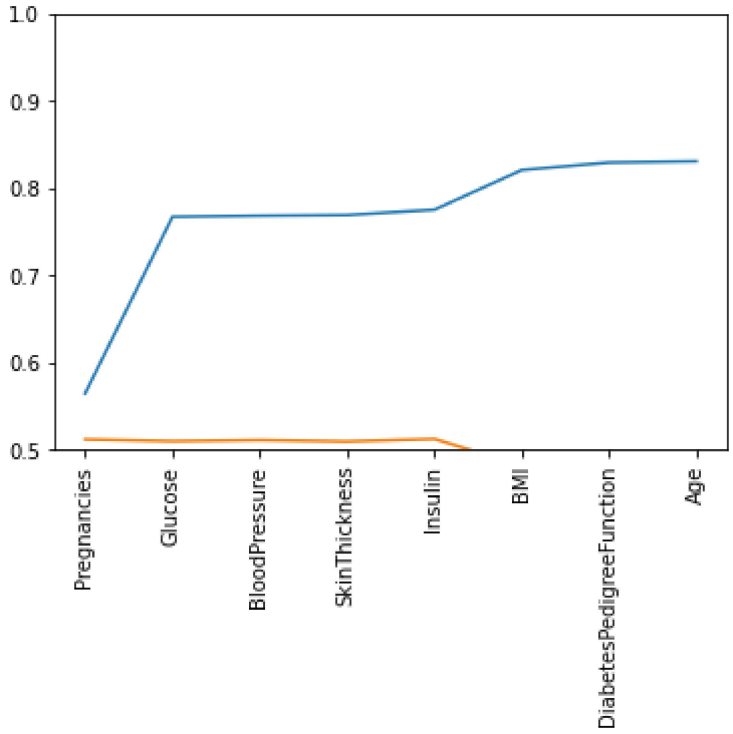
```
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\linear_model\_logisti
c.py:814: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sciki
t-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-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\utils\validation.py:99
3: DataConversionWarning: A column-vector y was passed when a 1d array was ex
pected. Please change the shape of y to (n_samples, ), for example using rave
l().
  y = column_or_1d(y, warn=True)
C:\Users\Harsmitha\anaconda3\lib\site-packages\sklearn\linear_model\_logisti
c.py:814: ConvergenceWarning: lbfgs failed to converge (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://sciki
t-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-regres
sion (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regr
ession)
  n_iter_i = _check_optimize_result(
```

In [37]:
```
import matplotlib.pyplot as plt
```

In [38]:
```
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()
```



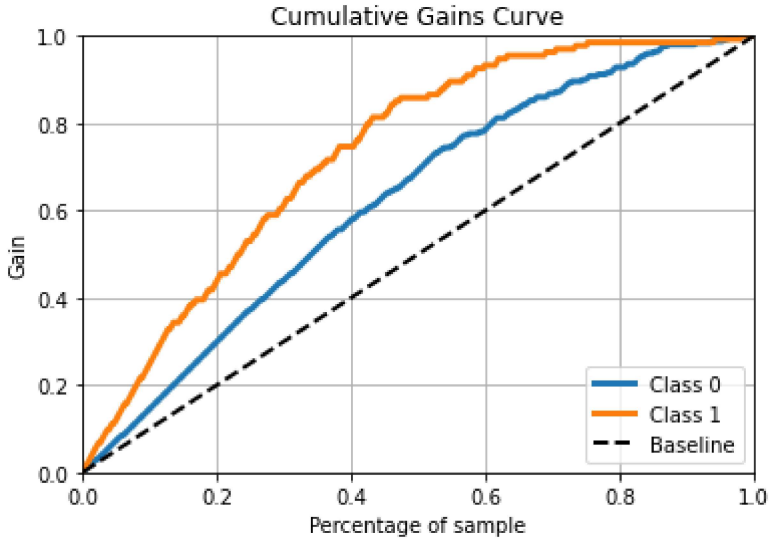
step 8 Draw cumulative gain chart and lift chart

In [41]:
```
pip install scikit.plot
```

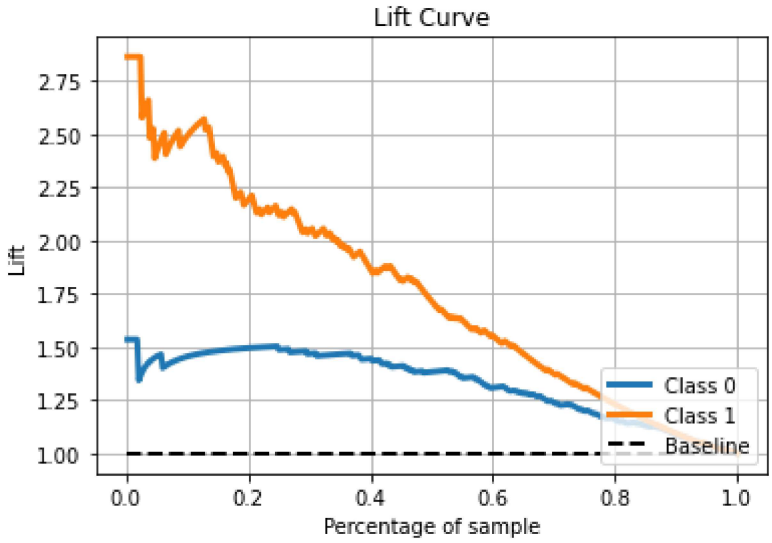
```
Collecting scikit.plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\harsmitha\anacond
a3\lib\site-packages (from scikit.plot) (1.0.2)
Requirement already satisfied: scipy>=0.9 in c:\users\harsmitha\anaconda3\lib\s
ite-packages (from scikit.plot) (1.7.3)
Requirement already satisfied: matplotlib>=1.4.0 in c:\users\harsmitha\anaconda
3\lib\site-packages (from scikit.plot) (3.5.1)
Requirement already satisfied: joblib>=0.10 in c:\users\harsmitha\anaconda3\lib
\site-packages (from scikit.plot) (1.1.0)
Requirement already satisfied: numpy>=1.17 in c:\users\harsmitha\anaconda3\lib
\site-packages (from matplotlib>=1.4.0->scikit.plot) (1.21.5)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\harsmitha\anaconda
3\lib\site-packages (from matplotlib>=1.4.0->scikit.plot) (4.25.0)
Requirement already satisfied: packaging>=20.0 in c:\users\harsmitha\anaconda3
\lib\site-packages (from matplotlib>=1.4.0->scikit.plot) (21.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\harsmitha\anaconda3
\lib\site-packages (from matplotlib>=1.4.0->scikit.plot) (3.0.4)
Requirement already satisfied: cycler>=0.10 in c:\users\harsmitha\anaconda3\lib
\site-packages (from matplotlib>=1.4.0->scikit.plot) (0.11.0)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\harsmitha\anaco
nda3\lib\site-packages (from matplotlib>=1.4.0->scikit.plot) (2.8.2)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\harsmitha\anaconda
3\lib\site-packages (from matplotlib>=1.4.0->scikit.plot) (1.3.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\harsmitha\anaconda3\li
b\site-packages (from matplotlib>=1.4.0->scikit.plot) (9.0.1)
Requirement already satisfied: six>=1.5 in c:\users\harsmitha\anaconda3\lib\sit
e-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\harsmitha\anaco
nda3\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
Note: you may need to restart the kernel to use updated packages.
```

In [42]:
```
import scikitplot as skplt
```

In [45]:
```
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 [ ]:
```

```