PML_lab-6_205229118_Mahalakshmi.S

April 26, 2021

0.0.1 Lab6. Predictive Analytics for Hospitals

0.0.2 Step1. [Import dataset]

Using Pandas, import "diabetes.csv" file and print properties such as head, shape, columns, dtype, info and value_counts.

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

[2]:
```
diab=pd.read_csv("diabetes.csv")
diab
```

[2]:
```
     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]
```

[3]:
```
diab.head()
```

[3]:
```
   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

   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
```

[4]:
```
diab.shape
```

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

[5]:
```
df = pd.read_csv("diabetes.csv")
df
```

[5]:
```
     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]
```

```
[6]: dia1 = df.columns
     dia1
```

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

```
[7]: df.dtypes
```

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

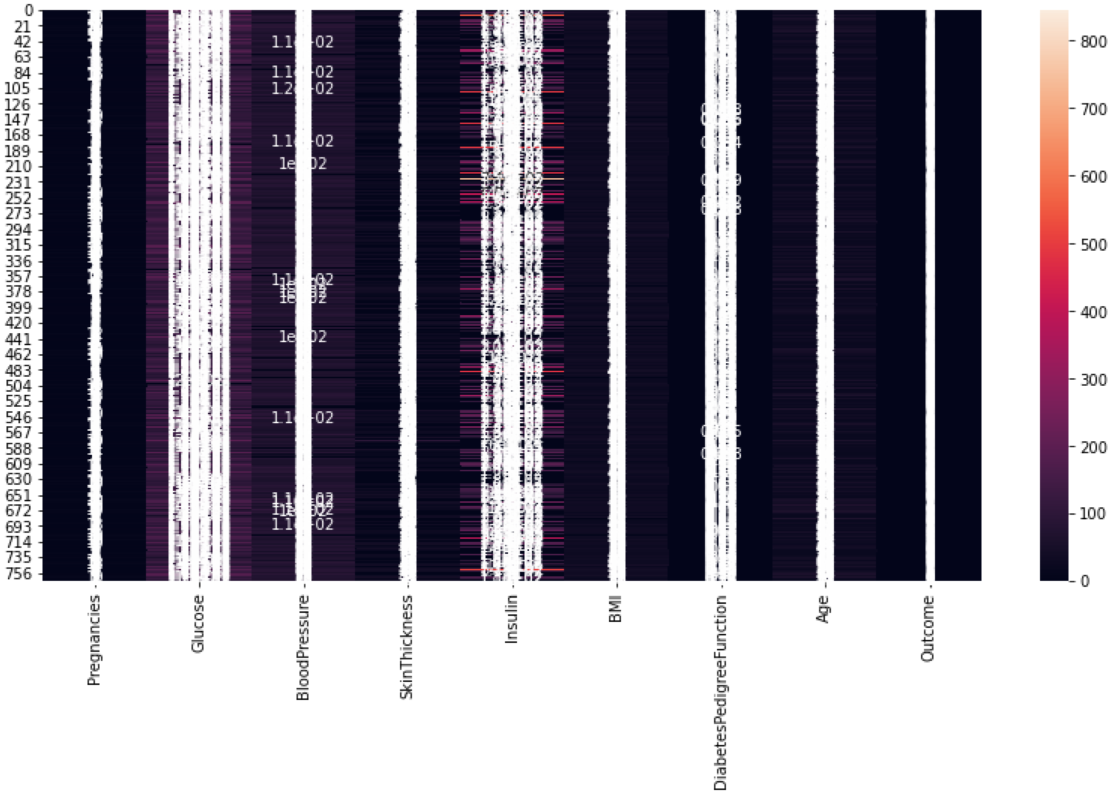
0.0.3 Step2. [Identify relationships between feature]

Create a Heatmap for the dataset and understand the data

```
[8]: import matplotlib.pyplot as plt
     import seaborn as sns

     %matplotlib inline
```

```
[9]: plt.figure(figsize=(14,7))
     sns.heatmap(data=df,annot=True)
```

```
[9]: <AxesSubplot:>
```



0.0.4 Step3. [Prediction using one feature]

Will older people become diabetic?

Create LogisticRegression model, train with "Age" as X and "Outcome" feature as y.

[10]:
```
X=df[["Age"]]
y=df[["Outcome"]]
```

[11]:
```
X
```

[11]:
```
     Age
0     50
1     31
2     32
3     21
4     33
..   …
763   63
764   27
765   30
766   47
767   23
```

```
[768 rows x 1 columns]
```

[12]:
```
y
```

[12]:
```
     Outcome
0          1
1          0
2          1
3          0
4          1
..       …
763        0
764        0
765        0
766        1
767        0

[768 rows x 1 columns]
```

[13]:
```
from sklearn.linear_model import LogisticRegression
```

[14]:
```
logmodel = LogisticRegression()
```

[15]:
```
logmodel.fit(X,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

[15]:
```
LogisticRegression()
```

0.0.5 Print model parameter values: coef_ and intercept_

0.0.6 coef_

[16]:
```
logmodel.coef_
```

[16]:
```
array([[0.04202466]])
```

0.0.7 intercept_

[17]:
```
logmodel.intercept_
```

[17]:
```
array([-2.04744865])
```

0.0.8 Query: A person is 60 years old. Will he be diabetic?

[24]:

```
log = logmodel.coef_ * 60 + logmodel.intercept_
from scipy.special import expit
if expit(log) > 0.5:
    print(expit(log))
    print('YES, he will become diabetic')
else:
    print("NO, he will not be diabetic")
```

```
[[0.61633741]]
YES, he will become diabetic
```

[25]:

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

[25]:

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

0.0.9 Step4. [Prediction using many features]

Will Glucose, BMI and Age values make someone diabetic?

0.0.10 Select the three features 'Glucose', 'BMI' and 'Age' from your dataset, call it as X

[39]:

```
_X=df[["Glucose","BMI","Age"]]
_X
```

[39]:

```
     Glucose   BMI  Age
0        148  33.6   50
1         85  26.6   31
2        183  23.3   32
3         89  28.1   21
4        137  43.1   33
..       …     …    …
763      101  32.9   63
764      122  36.8   27
765      121  26.2   30
766      126  30.1   47
767       93  30.4   23

[768 rows x 3 columns]
```

0.0.11 Create a new LogisticRegression model, train with X and 'Outcome' as y.

[40]:

```
y=df[["Outcome"]]
y
```

[40]:

```
     Outcome
0          1
```

```
1          0
2          1
3          0
4          1
..       …
763        0
764        0
765        0
766        1
767        0

[768 rows x 1 columns]
```

[41]:
```
logmodel1 = LogisticRegression()
```

[42]:
```
logmodel1.fit(_X,y)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

[42]:
```
LogisticRegression()
```

[43]:
```
logmodel1.predict([[40,30,150]])
```

[43]:
```
array([0], dtype=int64)
```

[44]:
```
logmodel1.coef_
```

[44]:
```
array([[0.03251154, 0.08157404, 0.03015421]])
```

[45]:
```
logmodel1.intercept_
```

[45]:
```
array([-8.39311252])
```

[52]:
```
log1 = logmodel1.coef_[0][0] * 40 + logmodel1.coef_[0][1] *30 + logmodel1.
 ↪coef_[0][2] *150 + logmodel1.intercept_
from scipy.special import expit
if expit(log1) > 0.5:
    print(expit(log1))
    print('YES, he will become diabetic')
else:
    print("NO, he will not be diabetic")
```

```
NO, he will not be diabetic
```

[49]:
```
logmodel1.predict_proba([[150,30,40]])
```

```
[49]: array([[0.46580162, 0.53419838]])
```

0.0.12 Step5. [Build LoR model with all features]

Create LoR model, train it with X_train and y_train values

```
[53]: X_lor = df.drop('Outcome',axis=1)
```

```
[54]: logmodel2 = LogisticRegression()
```

```
[55]: from sklearn.model_selection import train_test_split
```

```
[57]: X_train,X_test,y_train,y_test = train_test_split(X_lor,y,train_size=0.
      ↪8,test_size=0.2)
```

```
[58]: logmodel2.fit(X_train,y_train)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:762: 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
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

```
[58]: LogisticRegression()
```

```
[59]: y_test
```

```
[59]:      Outcome
      143        1
      579        1
      764        0
      313        0
      426        0
      ..       …
      437        0
      285        0
      62         0
      513        0
```

```
389        0

[154 rows x 1 columns]
```

[60]:

```
y_predict = logmodel2.predict(X_test)
y_predict
```

[60]:

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

Now, compute and print its AUC value

[61]:

```
from sklearn.metrics import roc_auc_score
```

[63]:

```
print("LoR AUC  ",roc_auc_score(y_test,y_predict))
```

```
LoR AUC   0.7733785822021116
```

0.0.13 Step6. [Forward Selection Procedure]

[64]:

```
type(df.columns)
```

[64]:

```
pandas.core.indexes.base.Index
```

[65]:

```
def auc(var,tar,df):
    fX = df[var]
    fy = df[tar]
    logreg = LogisticRegression()
    logreg.fit(fX,fy)
    pred=logreg.predict_proba(fX)[:,1]
    auc_val = roc_auc_score(y,pred)
    return auc_val
```

[66]:

```
auc(["BMI","Glucose"],["Outcome"],df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

[66]:

```
0.8109328358208956
```

[67]:

```
auc(['Pregnancies', 'BloodPressure', 'SkinThickness'],["Outcome"],df)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

[67]:

```
0.6444962686567164
```

[68]:

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

[70]:

```
tar = ["Outcome"]
current = ['Insulin','BMI', 'DiabetesPedigreeFunction', 'Age']
cand = ['Pregnancies', 'BloodPressure', 'SkinThickness']
next_var = next_best(current,cand,tar,df)
print(next_var)
```

```
SkinThickness
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

[72]:

```
tar = ["Outcome"]
current = []
cand = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness',␣
 ↪'Insulin','BMI', 'DiabetesPedigreeFunction', 'Age']
max_num = 5
```

```python
num_it = min(max_num,len(cand))
for i in range(0,num_it):
    next_var = next_best(current,cand,tar,df)
    current = current + [next_var]
    cand.remove(next_var)
    print("Variable added in step " + str(i+1)  + " is " + next_var + ".")
print(current)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

```
Variable added in step 1 is Glucose.
Variable added in step 2 is BMI.
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

```
Variable added in step 3 is Pregnancies.
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
```

```
Variable added in step 4 is DiabetesPedigreeFunction.

C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)


Variable added in step 5 is BloodPressure.
['Glucose', 'BMI', 'Pregnancies', 'DiabetesPedigreeFunction', 'BloodPressure']
```

0.0.14 Step7. [Plot Line graph of AUC values and select cut-off]

[73]:
```
X_train,X_test,y_train,y_test = train_test_split(X_lor,y,test_size = 0.
↪5,stratify =y)
```

[74]:
```
predict = logmodel2.predict_proba(X_test)
```

[75]:
```
train = pd.concat([X_train,y_train], axis=1)
test = pd.concat([X_test,y_test], axis=1)
```

[76]:
```
def auc_train_test(variables,target,train,test):
    X_train = train[variables]
    X_test = test[variables]
    Y_train = train[target]
    Y_test = test[target]
    log_reg = LogisticRegression()

    # Fit the model on train data
    log_reg.fit(X_train, Y_train)
    # Calculate the predictions both on train and test data
    predictions_train = log_reg.predict_proba(X_train)[:,1]
    predictions_test = log_reg.predict_proba(X_test)[:,1]

    # Calculate the AUC both on train and test data
    auc_train = roc_auc_score(Y_train, predictions_train)
    auc_test = roc_auc_score(Y_test,predictions_test)
    return(auc_train, auc_test)
```

[77]:

```
auc_values_train = []
auc_values_test = []
variables_evaluate = []

# Iterate over the variables in variables
for v in X_lor.columns:

    # Add the variable
    variables_evaluate.append(v)

    # Calculate the train and test AUC of this set of variables
    auc_train, auc_test =␣
 ↪auc_train_test(variables_evaluate,["Outcome"],train,test)

    # Append the values to the lists
    auc_values_train.append(auc_train)
    auc_values_test.append(auc_test)
```

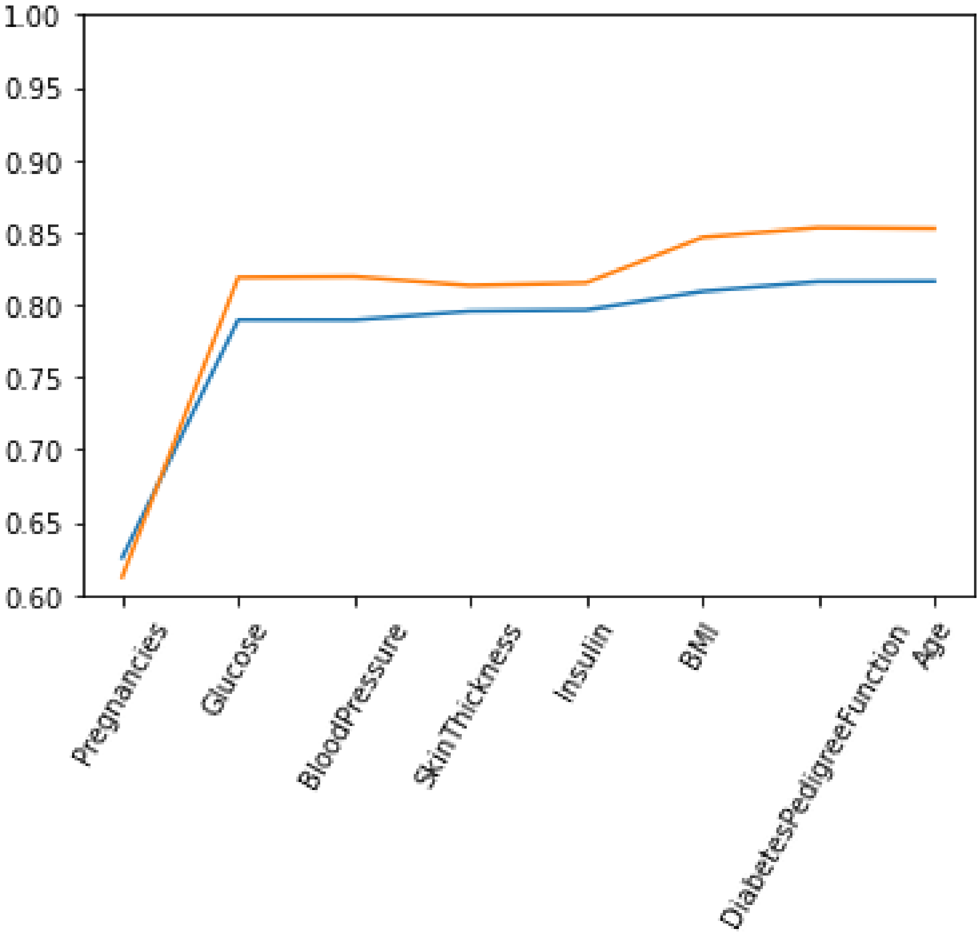
```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
```

```
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:762: 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
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\utils\validation.py:72:
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().
  return f(**kwargs)
C:\ProgramData\Anaconda3\lib\site-
packages\sklearn\linear_model\_logistic.py:762: 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
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[79]:

```python
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_lor.columns,rotation=60)
plt.plot(x,my_train)
plt.plot(x,my_test)
plt.ylim((0.6,1.0))
plt.show()
```



0.0.15 Step8. [Draw Cumulative Gain Chart and Lift Chart]

[83]:
```
!pip install scikit-plot
```

```
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: scipy>=0.9 in c:\programdata\anaconda3\lib\site-
packages (from scikit-plot) (1.5.2)
Requirement already satisfied: joblib>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from scikit-plot) (0.17.0)
```

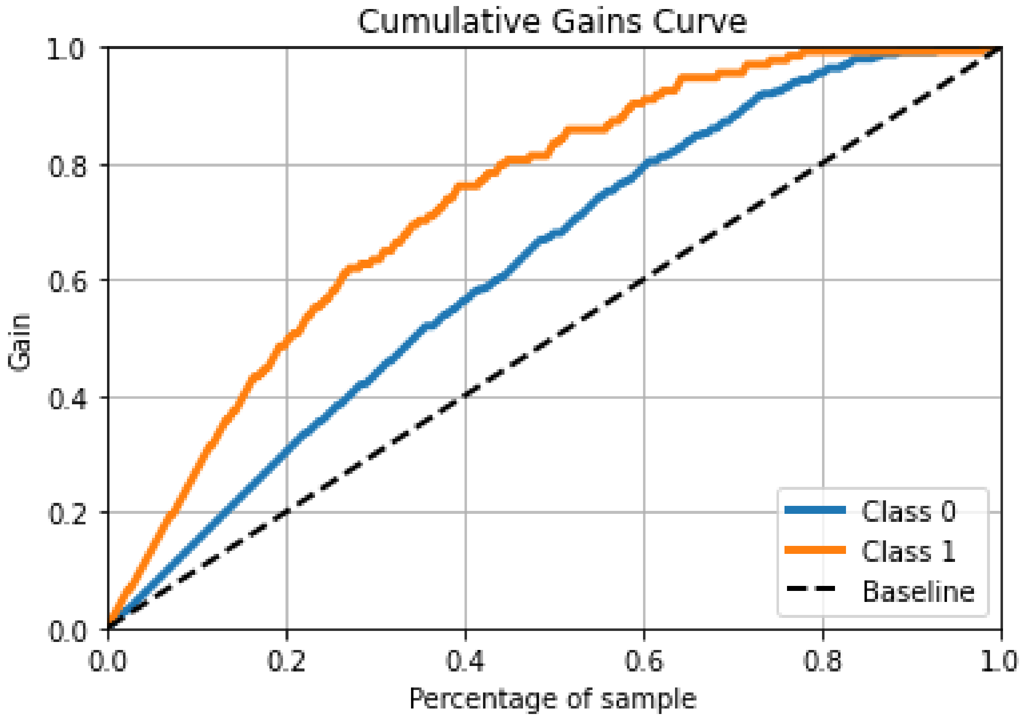
```
WARNING: Value for scheme.headers does not match. Please report this to
<https://github.com/pypa/pip/issues/9617>
distutils: c:\programdata\anaconda3\Include\UNKNOWN
sysconfig: c:\programdata\anaconda3\Include
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Value for scheme.headers does not match. Please report this to
<https://github.com/pypa/pip/issues/9617>
distutils: c:\programdata\anaconda3\Include\UNKNOWN
sysconfig: c:\programdata\anaconda3\Include
WARNING: Additional context:
user = False
home = None
root = None
prefix = None
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
WARNING: Ignoring invalid distribution -umpy (c:\programdata\anaconda3\lib\site-
packages)
```

```
Requirement already satisfied: matplotlib>=1.4.0 in
c:\programdata\anaconda3\lib\site-packages (from scikit-plot) (3.3.2)
Requirement already satisfied: scikit-learn>=0.18 in
c:\programdata\anaconda3\lib\site-packages (from scikit-plot) (0.23.2)
Requirement already satisfied: cycler>=0.10 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(2.4.7)
Requirement already satisfied: certifi>=2020.06.20 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(2020.6.20)
Requirement already satisfied: python-dateutil>=2.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(2.8.1)
```

```
Requirement already satisfied: pillow>=6.2.0 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(8.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\programdata\anaconda3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot)
(1.3.0)
Requirement already satisfied: numpy>=1.15 in c:\programdata\anaconda3\lib\site-
packages (from matplotlib>=1.4.0->scikit-plot) (1.20.2)
Requirement already satisfied: six in c:\programdata\anaconda3\lib\site-packages
(from cycler>=0.10->matplotlib>=1.4.0->scikit-plot) (1.15.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\programdata\anaconda3\lib\site-packages (from scikit-learn>=0.18->scikit-
plot) (2.1.0)
Installing collected packages: scikit-plot
Successfully installed scikit-plot-0.3.7
```

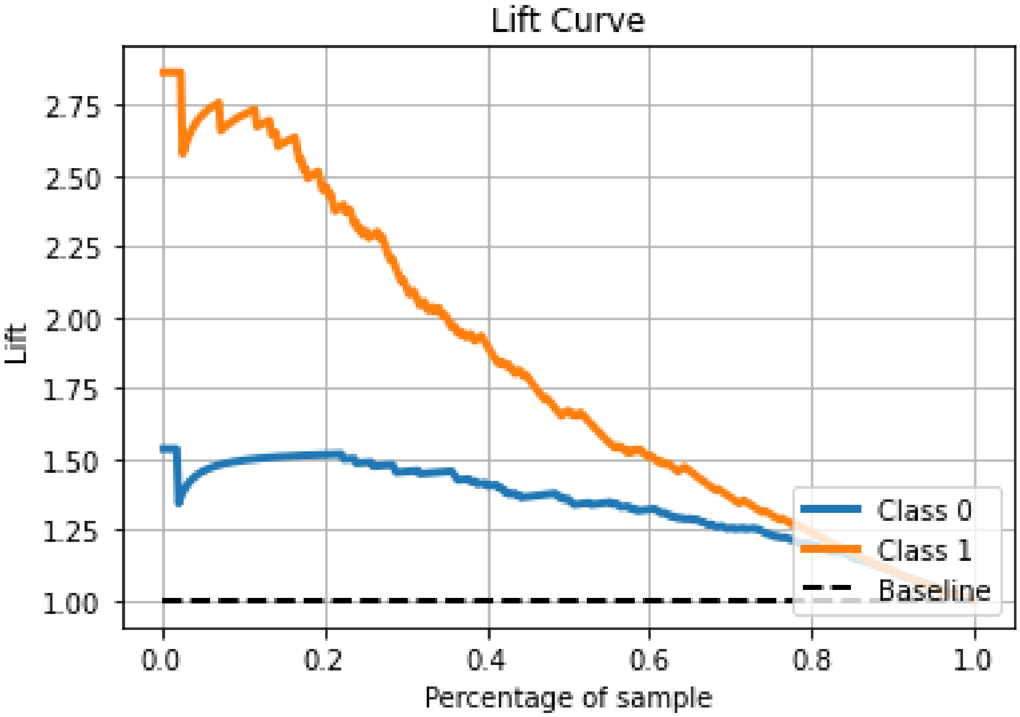
```
[84]: import scikitplot as skplt
```

```
[85]: skplt.metrics.plot_cumulative_gain(y_test,predict)
      plt.show()
```



```
[86]: plt.figure(figsize=(7,7))
      skplt.metrics.plot_lift_curve(y_test,predict)
      plt.show()
```

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


Lift Curve
Lift
2.75
2.50
2.25
2.00
1.75
1.50
1.25
1.00
0.0
0.2
0.4
0.6
0.8
1.0
Percentage of sample
Class 0
Class 1
Baseline

```
[ ]:
```