Name: Murali Kumar R

Roll No: 225229120

Lab.6 Predictive Analytics for hospitals

Step1: import dataset

In [1]:

```
import pandas as pd
f=pd.read_csv("diabetes.csv")
```

In [2]:

```
f.head()
```

Out[2]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunctio
0	6	148	72	35	0	33.6	0.62
1	1	85	66	29	0	26.6	0.35
2	8	183	64	0	0	23.3	0.67
3	1	89	66	23	94	28.1	0.16
4	0	137	40	35	168	43.1	2.28

In [3]:

```
f.shape
```

Out[3]:

```
(768, 9)
```

In [4]:

```
f.columns
```

Out[4]:

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

In [5]:

```
type(f)
```

Out[5]:

```
pandas.core.frame.DataFrame
```

In [6]:

```
f.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   Pregnancies               768 non-null    int64  
 1   Glucose                   768 non-null    int64  
 2   BloodPressure             768 non-null    int64  
 3   SkinThickness             768 non-null    int64  
 4   Insulin                   768 non-null    int64  
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64  
 8   Outcome                   768 non-null    int64  
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

In [7]:

```
f.count()
```

Out[7]:

```
Pregnancies                 768
Glucose                     768
BloodPressure               768
SkinThickness               768
Insulin                     768
BMI                         768
DiabetesPedigreeFunction    768
Age                         768
Outcome                     768
dtype: int64
```

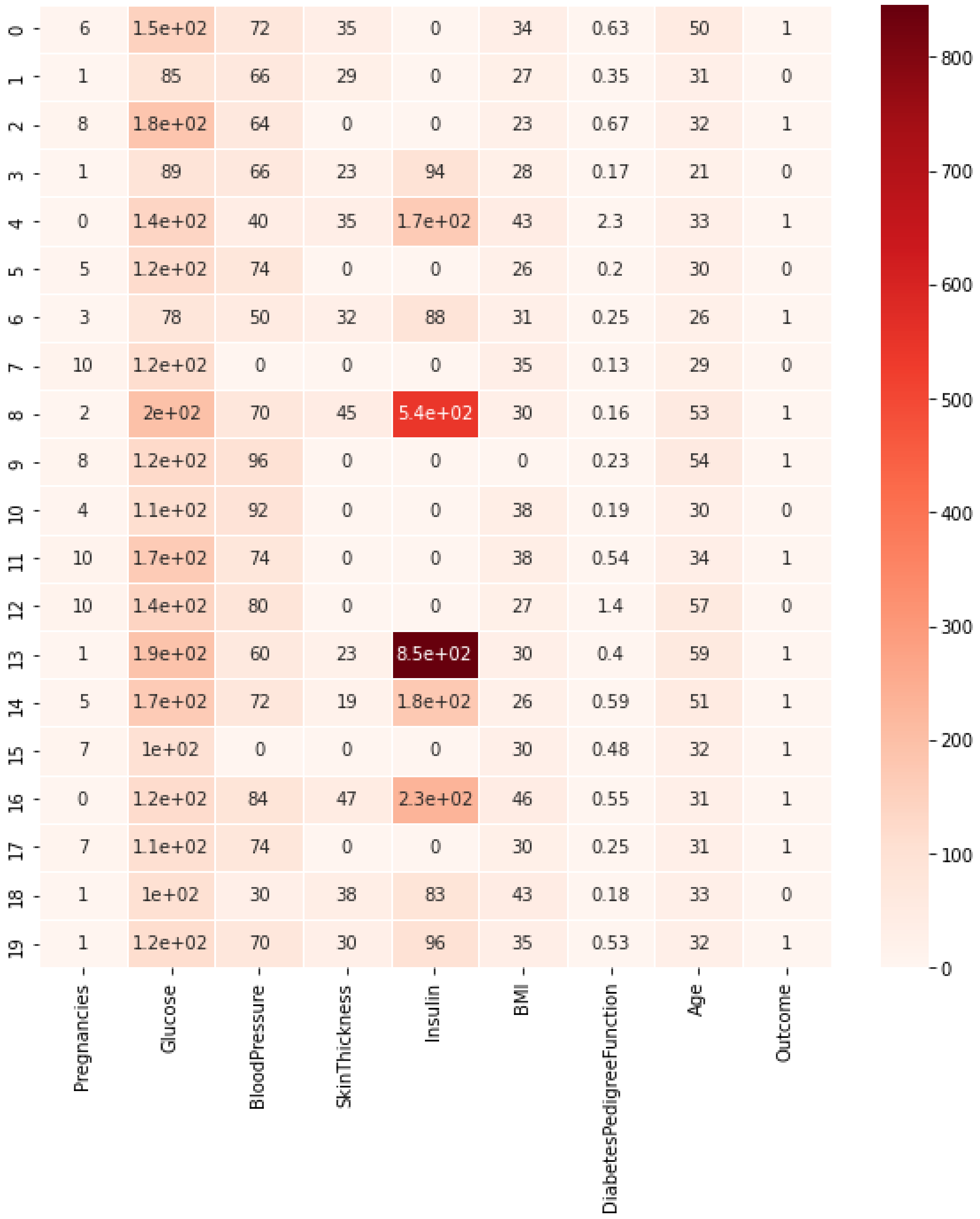
Step2: Identifying relationships between features

In [8]:

```
 import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
sns.heatmap(f.head(20), cmap='Reds',annot=True, linewidth=.5)
```

Out[8]:

```
<AxesSubplot:>
```



Step 3:Prediction using one feature

In [9]:

```
X = f[['Age']]
```

In [10]:

```
y = f['Outcome']
```

In [11]:

```
from sklearn.linear_model import LogisticRegression
lrm1 = LogisticRegression()
lrm1.fit(X, y)
```

Out[11]:

```
LogisticRegression()
```

In [12]:

```
lrm1.coef_
```

Out[12]:

```
array([[0.04202466]])
```

In [13]:

```
lrm1.intercept_
```

Out[13]:

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

In [14]:

```
year_old = [[60]]
lrm1.predict(year_old)
```

```
C:\Users\Dell\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(
```

Out[14]:

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

In [15]:

```
lrf = lrm1.coef_ * 60 + lrm1.intercept_
from scipy.special import expit
if expit(lrf) > 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 [16]:

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

In [17]:

```
lrm2 = LogisticRegression()
```

In [18]:

```
lrm2.fit(X1, y)
```

Out[18]:

```
LogisticRegression()
```

In [19]:

```
lrm2.predict([[150, 30, 40]])
```

```
C:\Users\Dell\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(
```

Out[19]:

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

In [20]:

```
lrm2.predict_proba([[150, 30, 40]])
```

```
C:\Users\Dell\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(
```

Out[20]:

```
array([[0.53053646, 0.46946354]])
```

Step5. [Build LoR model with all features]

In [21]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [22]:

```
X3 = f.drop('Outcome', axis=1)
lrm3 = LogisticRegression()
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X3,y,train_size=0.8,test_size=0.2)
lrm3.fit(X_train, y_train)
```

Out[22]:

```
LogisticRegression()
```

In [23]:

```
y_pred = lrm3.predict(X_test)
y_pred
```

Out[23]:

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

In [24]:

```
from sklearn.metrics import roc_auc_score
print("LoR AUC ", roc_auc_score(y_test, y_pred))
```

```
LoR AUC  0.6821816105082809
```

Step6: Forward selection procedures

In [25]:

```
type(f.columns)
```

Out[25]:

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

In [26]:

```
def get_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
get_auc(["BMI","Glucose"],["Outcome"],f)
```

Out[26]:

```
0.8109328358208956
```

In [27]:

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

Out[27]:

```
0.6444962686567164
```

In [28]:

```
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
```

In [29]:

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

```
Glucose
```

In [30]:

```
tar = ["Outcome"]
current = []
cand = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin','BMI', 'Diabe
max_num = 5
num_it = min(max_num,len(cand))
for i in range(0,num_it):
    next_var = best_next(current,cand,tar,f)
    current = current + [next_var]
    cand.remove(next_var)
print("Variable added in step " + str(i+1) + " is " + next_var + ".")
print(current)
```

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

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

In [31]:

```
X_train,X_test,y_train,y_test = train_test_split(X3,y,test_size = 0.5,stratify =y)
```

In [32]:

```
pred2 = lrm3.predict_proba(X_test)
```

In [33]:

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

In [34]:

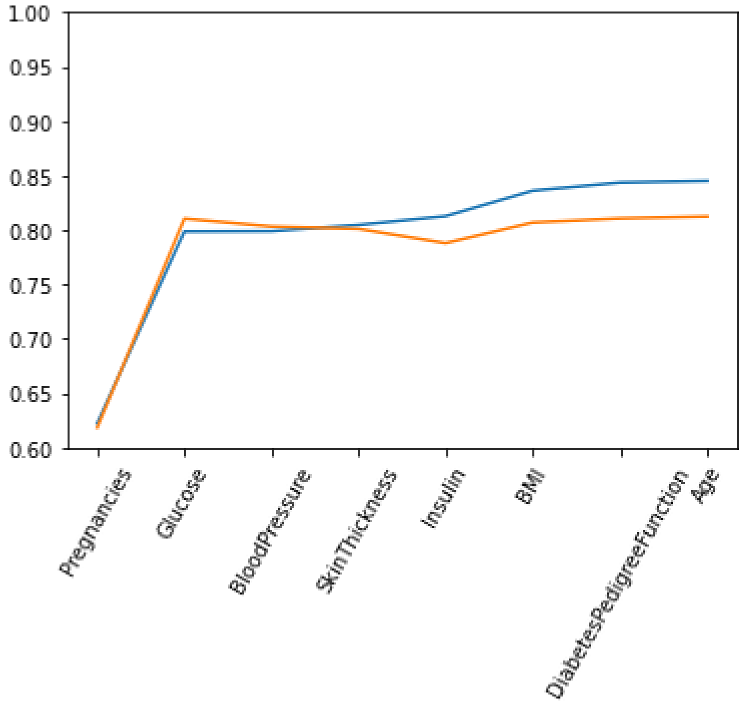
```
def auc_train_test(variables,target,train,test):
    X_train = train[variables]
    X_test = test[variables]
    Y_train = train[target]
    Y_test = test[target]
    logreg = LogisticRegression()
    logreg.fit(X_train, Y_train)
    predictions_train = logreg.predict_proba(X_train)[:,1]
    predictions_test = 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 [35]:

```
auc_values_train = []
auc_values_test = []
variables_evaluate = []
# Iterate over the variables in variables
for v in X3.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)
```

In [36]:

```
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,X3.columns,rotation=60)
plt.plot(x,my_train)
plt.plot(x,my_test)
plt.ylim((0.6,1.0))
plt.show()
```



Step8. [Draw Cumulative Gain Chart and Lift Chart]

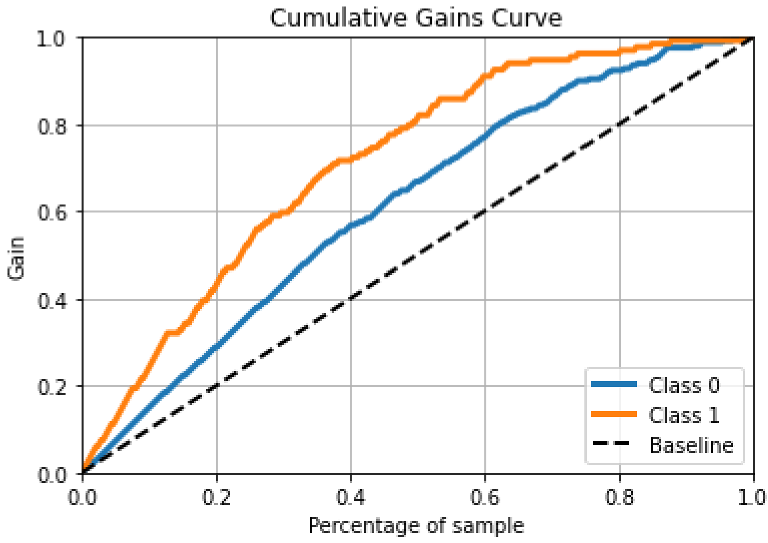
In [37]:

```
!pip install scikit-plot
from scikitplot.estimators import plot_feature_importances
from scikitplot.metrics import plot_confusion_matrix, plot_roc
```

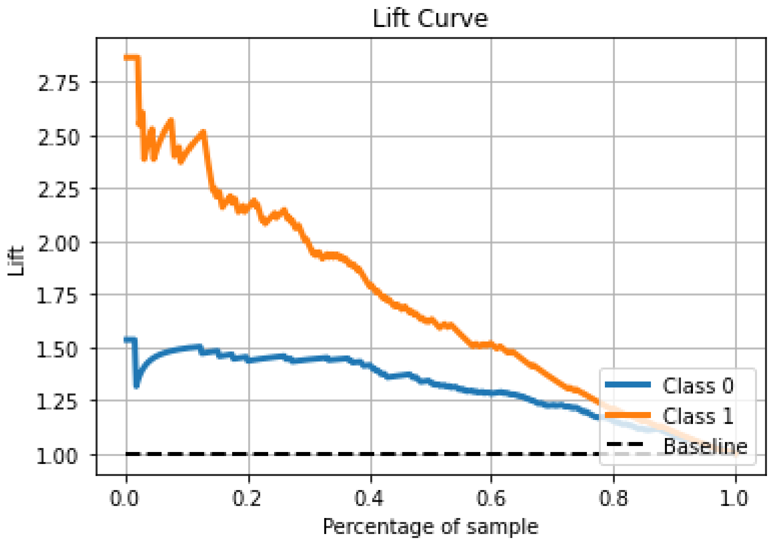
```
Collecting scikit-plot
  Downloading scikit_plot-0.3.7-py3-none-any.whl (33 kB)
Requirement already satisfied: matplotlib>=1.4.0 in c:\users\dell\anaconda3
\lib\site-packages (from scikit-plot) (3.5.1)
Requirement already satisfied: joblib>=0.10 in c:\users\dell\anaconda3\lib\s
ite-packages (from scikit-plot) (1.1.0)
Requirement already satisfied: scipy>=0.9 in c:\users\dell\anaconda3\lib\sit
e-packages (from scikit-plot) (1.7.3)
Requirement already satisfied: scikit-learn>=0.18 in c:\users\dell\anaconda3
\lib\site-packages (from scikit-plot) (1.0.2)
Requirement already satisfied: pillow>=6.2.0 in c:\users\dell\anaconda3\lib
\site-packages (from matplotlib>=1.4.0->scikit-plot) (9.0.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (1.3.2)
Requirement already satisfied: numpy>=1.17 in c:\users\dell\anaconda3\lib\si
te-packages (from matplotlib>=1.4.0->scikit-plot) (1.21.5)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\dell\anacond
a3\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (2.8.2)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\dell\anaconda3
\lib\site-packages (from matplotlib>=1.4.0->scikit-plot) (4.25.0)
Requirement already satisfied: packaging>=20.0 in c:\users\dell\anaconda3\li
b\site-packages (from matplotlib>=1.4.0->scikit-plot) (21.3)
Requirement already satisfied: cycler>=0.10 in c:\users\dell\anaconda3\lib\s
ite-packages (from matplotlib>=1.4.0->scikit-plot) (0.11.0)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\dell\anaconda3\l
ib\site-packages (from matplotlib>=1.4.0->scikit-plot) (3.0.4)
Requirement already satisfied: six>=1.5 in c:\users\dell\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\dell\anacond
a3\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 [38]:

```
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 [ ]: