Università
della
Svizzera
italiana

Faculty
of Informatics

May 30, 2018

Stochastic Methods

Assignment 6

Sumeet Gyanchandani

1 K-Means Clustering

The figure below depicts the confusion matrix for the Y_{kmeans} values. Considering Y value 1, i.e., *Yes* as positive and value 0, i.e., *No* as negative. The top right quadrant represents **False Positives**, bottom right quadrant represents **True Positives**, top left quadrant represents **True Negatives** and bottom left quadrant represents **False Negatives**.

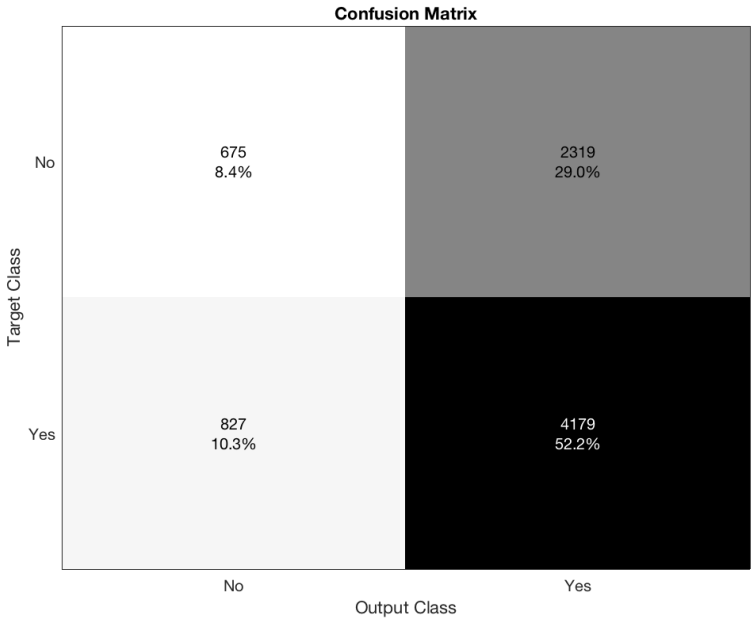

Figure 1. Confusion Matrix for K-Means Clustering

The Accuracy of these clustering is calculated by is the formula:

$$\text{Accuracy} = \frac{TN + TP}{FN + FP + TN + TP}$$

The Accuracy of K-Means Clustering is **60.68%**.

2 Support Vector Machine

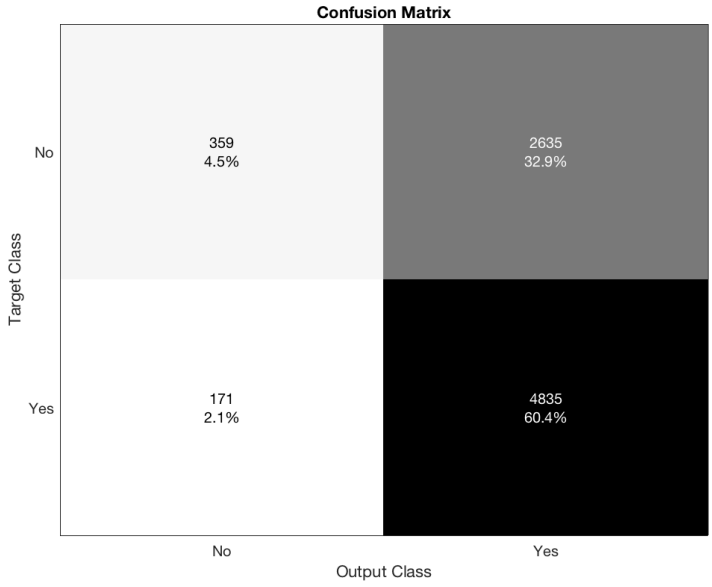

Figure 2. Confusion Matrix for SVM

The Accuracy of Support Vector Machine model is **64.93%**.

3 Logit Regression

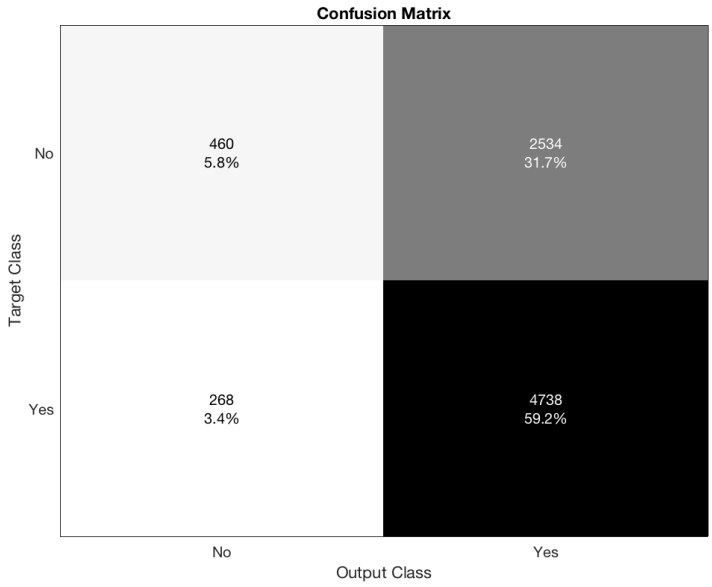

Figure 3. Confusion Matrix for Logit Regression

The Accuracy of Logit Regression is **64.98%**.

The prediction of supervised learning methods were slightly better than the K-Means Clustering. The number of False Positive was much higher than the number of False Negatives for all the algorithms.

Listing 1. clustering.m

```
clear all;
close all;

% Loading data
load('swissmetro.mat')

%Parameters
K = 2;
N = size(Y,1);
elements = 1:N;

% Finding the number of Yes
yes = sum(Y);

% Kmeans Clustering
rng(4);
[Y_kmeans,C] = kmeans(X,K);
Y_kmeans(Y_kmeans==2)=0;

C_kmeans = confusionmat(Y, Y_kmeans);
Accuracy_kmeans = plot_cm(C_kmeans, N);

%Support Vector Machine
svmStruct = fitcsvm(X,Y,'KernelFunction','RBF');
Y_svm = svmStruct.predict(X);

C_svm = confusionmat(Y, Y_svm);
Accuracy_svm = plot_cm(C_svm, N);

%Logit Regression
[b, dev, stats] = glmfit(X, Y,'binomial','link','logit');
Y_logit = round(glmval(b, X,'logit'));

C_logit = confusionmat(Y, Y_logit);
Accuracy_logit = plot_cm(C_logit, N);

function acc = plot_cm(confmat,N)
figure;
numlabels = size(confmat, 1); % number of labels
labels = {'No', 'Yes'};

% calculate the percentage accuracies
confpercent = 100*confmat./N;

% plotting the colors
imagesc(confpercent);
title('Confusion Matrix');
xlabel('Output Class'); ylabel('Target Class');

% set the colormap
colormap(flipud(gray));

% Create strings from the matrix values and remove spaces
textStrings = num2str([confmat(:), confpercent(:)], '%d\n%.1f%%\n');
textStrings = strtrim(cellstr(textStrings));

% Create x and y coordinates for the strings and plot them
[x,y] = meshgrid(1:numlabels);
hStrings = text(x(:),y(:),textStrings(:), ...
'HorizontalAlignment','center');

% Get the middle value of the color range
```

```
midValue = mean(get(gca,'CLim'));

% Choose white or black for the text color of the strings so
% they can be easily seen over the background color
textColors = repmat(confpercent(:) > midValue,1,3);
set(hStrings,{'Color'},num2cell(textColors,2));

% Setting the axis labels
set(gca,'XTick',1:numlabels,...
'XTickLabel',labels,...
'YTick',1:numlabels,...
'YTickLabel',labels,...
'TickLength',[0 0]);

%Calculate the accuracy for the model
acc = (confmat(1,1)+confmat(2,2))/N;
end
```
