ESTIMATING THE RISK OF HEART DISEASE BASED ON CLINICAL DATA SETS OF PATIENTS.

Name: Palliyaguru S.T

Student Number: IT15099846

Batch: Weekend-SE

Contents

Introduction

The healthcare industry today generates large amounts of complex data concerning patients, hospitals resources, disease diagnosis, electronic patient records, medical devices, etc. The large amount of data is a key resource to be processed and analyzed for knowledge extraction and enabling support for cost-savings and decision-making.

Coronary heart disease is considered a fatal illness that causes death to over a million patients every year. Nearly half the patients diagnosed with CAD will eventually die from the disease

This study is based on the clinical data sets on patients, who were examined for possibility of presence of heart diseases. The original dataset was published by Robert Detrano, M.D., Ph.D. V.A. Medical Center, Long Beach and Cleveland Clinic Foundation.

In this study I have referred to the Cleveland data set data set available from the UC Irvine Machine Learning Repository. (Reference: https://archive.ics.uci.edu/ml/datasets/Heart+Disease)

The original Cleveland data set contain 303 instances and nearly 76 attributes. The dataset available in UCI repository contains only 14 attributes and 303 instances.

This data set have been subjected to many machine learning based studies, in order to find the possibility of presence of a heart disease , with the factors affecting patients heart health.

Description on the dataset set.

Following are the attributes presented in Cleveland data set.

Attribute	Description	Type
age	Age in years	int
sex	Female or male	bin
cp	Chest pain type (typical angina, atypical angina, non-angina, or asymptomatic angina)	cat
trestbps	Resting blood pressure (mm Hg)	con
chol	Serum cholesterol (mg/dl)	con
fbs	Fasting blood sugar (< 120 mg/dl or > 120 mg/dl)	bin
restecg	Resting electrocardiography results (normal, ST-T wave abnormality, or left ventricular hypertrophy)	cat
thalach	Max. heart rate achieved during thallium stress test	con
exang	Exercise induced angina (yes or no)	bin
oldpeak	ST depression induced by exercise relative to rest)	con
slope	Slope of peak exercise ST segment (up sloping, flat, or down sloping)	cat
ca	Number of major vessels colored by fluoroscopy	int
thal	Thallium stress test result (normal, fixed defect, or reversible defect)	cat
num	Heart disease status: number of major vessels with >50% narrowing (0,1,2,3, or 4)	int

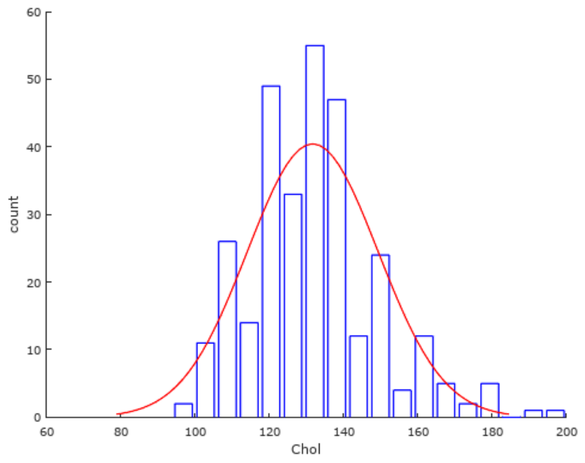
The last attribute "num" used to gain some insights on probability of a heart disease, as num=0 for no disease and num= 1,2,3,4 as presence of heart disease.

First of all I plotted some graphs in order to identify the weight of contribution of each of these attributes for the positive of heart disease.

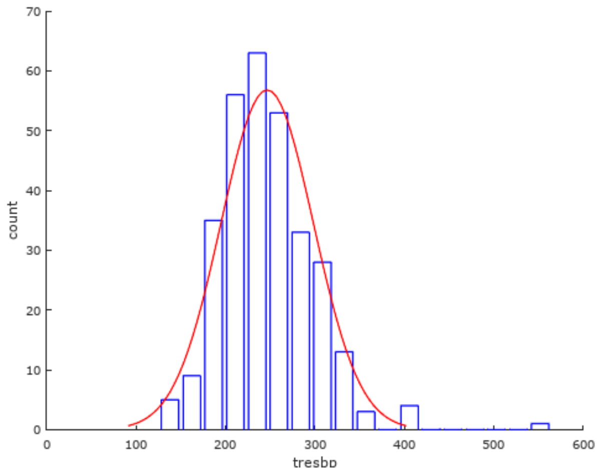
Heart disease data.

In my study, I preferred to identify the relationship between blood cholesterol level and resting blood pressure to the probability of heart disease, so that if the blood pressure and cholesterol level has given, then we can predict the possibility of a heart disease in that patient.

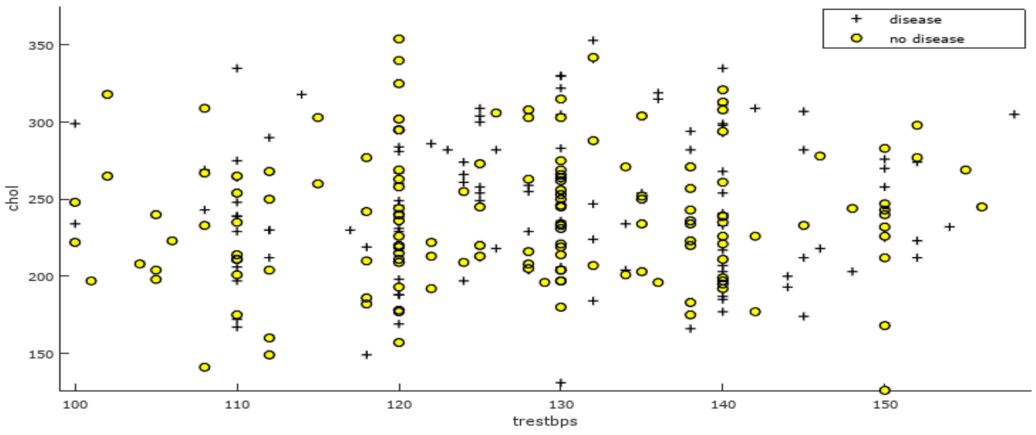
- Figure: 1 – distribution of cholesterol level data.



- Figure:2 – distribution of resting blood pressure data.

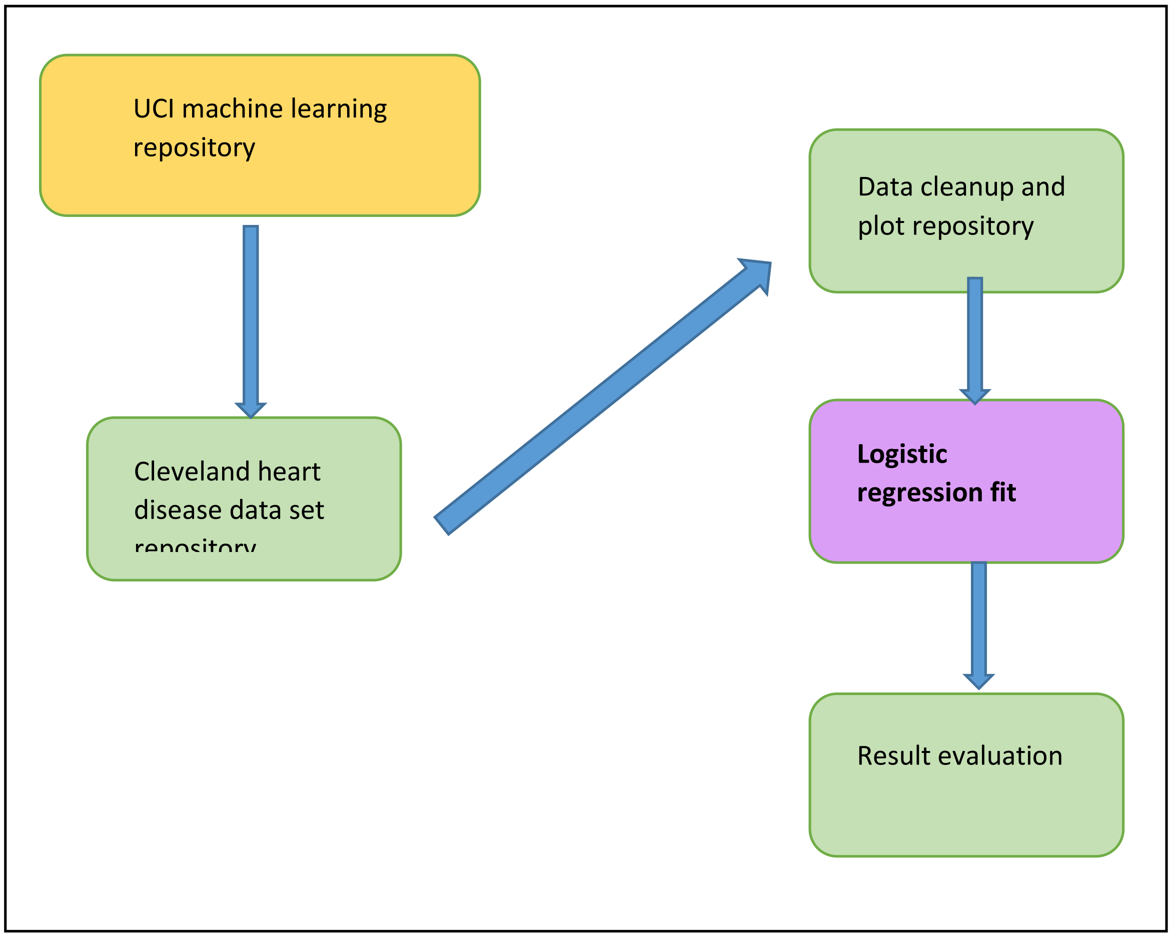


- Figure:3 - Scatter plot of chol and tresbp data- relation to the disease status.



Methodology.

I have used a logistic regression approach , where the end result will be classified to 5 classes : no disease (num =0) and disease (num =1,2,3,4).


UCI machine learning repository
Cleveland heart disease data set repository
Data cleanup and plot repository
Logistic regression fit
Result evaluation

Algorithm applied.

Logistic regression.

I have used the logistic regression approach to identify the relationships between blood cholesterol level and the resting blood pressure to the heart disease status.

In order to perform the multiclass classification, I needed to generate the confusion matrix for the column "num", as the categorical data couldn't identified by the algorithm.

num	C0	C1	C2	C3	C4
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1

The confusion matrix for "num" column

The cost function.

The aim of selecting the weight values for parameters: "chol" and "trestbps" is to minimize the error of predicted result to the actual result. In order to do this, the module is trained in a way to maximize the cost of a fault prediction. The cost function for a binary classification module is as below.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\begin{aligned} \mathrm{Cost}(h_\theta(x), y) &= -\log(h_\theta(x)) && \text{if } y = 1 \\ \mathrm{Cost}(h_\theta(x), y) &= -\log(1 - h_\theta(x)) && \text{if } y = 0 \end{aligned}$$

The **$h_\Theta(x^i)$** is the predicted result, and **y^i** is the actual result.

Sigmoid function.

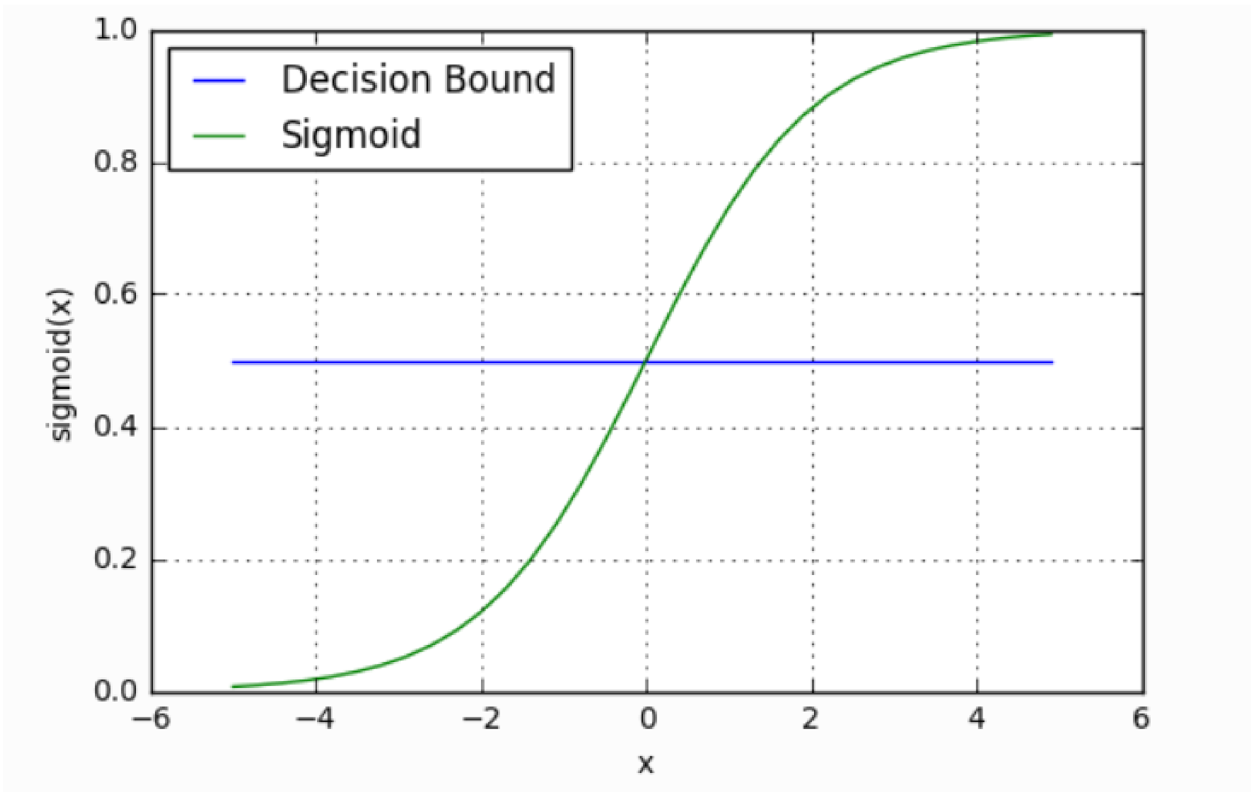
The predicted values are mapped into probabilities between 0-1 in order to identify the discrete class of the input parameters.

$$S(z) = \frac{1}{1 + e^{-z}}$$

The **z**, represents:

$$g\left(\boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{x}\right)$$

The sigmoid function classifies the probabilities of each observation, into positive (value closer to 1) or negative (values closer to zero) classes:



Implementation. [Appendix]

The solution was implemented using octave as main IDE.

The implementation process was divided as below.

1. Import data.
2. Plot data (chol- num and trestbps-num).
3. Add the intercept term.
4. Set initial values for Theta and Lambda.
5. Compute cost and gradient for logistic regression with regularization.
6. Get optimal theta values and the cost.
7. Plot decision boundary.

1. Import data.

```
%% ============= Load data =============
data=dlmread("clevlandedited.txt","\t",0,0);

X = data(:, [4, 5]); y = data(:, 14);
```

2. Plot data.

```
%plot chol and trestbps with num(heart disease status)
plotData(X, y);

% Labels and Legend
hold on;
xlabel('trestbps')
ylabel('chol')

legend('disease ', 'no disease ')
hold off;
```

3. Add the intercept term.

```
[m, n] = size(X);
X = [ones(m, 1) X];
```

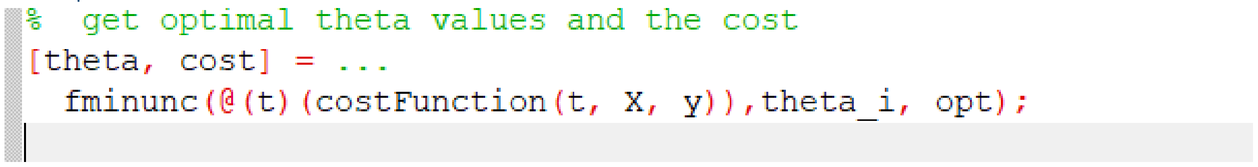
4. Set initial values for Theta.

```
theta_i = zeros(n + 1, 1);
```

5. Compute cost and gradient for logistic regression with regularization.

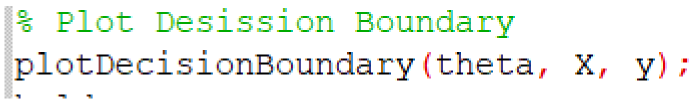
```
% Calculate the initial cost and gradient
[cost, gradient] = costFunction(theta_i, X, y);

temp1 = -1 * (y .* log(sigmoid(X * theta)));
temp2 = (1 - y) .* log(1 - sigmoid(X * theta));

J = sum(temp1 - temp2) / m;



grad = (X' * (sigmoid(X * theta) - y)) * (1/m);
```

6. Get optimal theta values and the cost.



```
%  get optimal theta values and the cost
[theta, cost] = ...
  fminunc(@(t)(costFunction(t, X, y)),theta_i, opt);
```

7. Plot decision boundary.



```
% Plot Desission Boundary
plotDecisionBoundary(theta, X, y);
```

Results.

The used model could predict a probability of 0.464210 to the values:

For a level of Cholesterol 145 mm Hg and resting blood pressure of 286 mg/dl with a train accuracy

Of: 54.125413.

Future Works.

The study was based on logistic regression, in which the end result will be classified among two discrete classes (disease, no disease) considering only the attributes of blood cholesterol level and resting blood pressure. And the accuracy for a particular two input values was nearly 54%.

In order to increase the accuracy level of the model, I hope to integrate more input parameters that would be co-relate with heart diseases. (Ex: age, fasting blood sugar, max heart rate during thallium test, more categorical variables like chest pain type, resting electrocardiography results.. etc.).

And rather than the logistic regression, the model accuracy and fitness will be improved if used Decision tree model, Support Vector Machine etc. So I would improve the solution by implementing mentioned algorithms to the scenario in future.

Appendix:

1. **clevlandTestData.m**

```
%%----------Cleveland Heart disease data analysis----------

% this file contains the implementation for logistic regression classification

% of Cleveland heart disease data.

% the blood cholesterol level (chol) and the resting blood pressure (tresbps) have been considered

% as attributes for the classification.


%% Initialization

clear ; close all; clc


%% ============ Load data ============

data=dlmread("clevlandedited.txt","\t",0,0);


X = data(:, [4, 5]); y = data(:, 14);


  %% ============ Plotting data============

  %plot Cholesterol data distribution.

  fprintf('\n----plotting Cholesterol data distribution..\n');

  figure; hold on;

  histfit(X(:,1));

  xlabel("Chol");

  ylabel("count");

  hold off;

  fprintf('\nProgram paused. Press enter to continue.\n');

  pause;


  %plot tresbp data distribution.

  fprintf('\n----plotting trebp data distribution..\n');

  figure; hold on;
```

```
   histfit(X(:,2));
   xlabel("tresbp");
   ylabel("count");
   hold off;
   fprintf('\nProgram paused. Press enter to continue.\n');
   pause;

   %plot chol and trestbps with num(heart disease status)
   fprintf('\n----plotting scatter plot of X and Y matrixes\n');
   plotData(X, y);
   % Labels and Legend
   hold on;
   xlabel('trestbps')
   ylabel('chol')
   legend('disease ', 'no disease ')
   hold off;

   fprintf('\nProgram paused. Press enter to continue.\n');
   pause;

   %% =========== Compute Cost and Gradient ===========

   %  setting up data matrix and add ones to intercept term
   [m, n] = size(X);
   X = [ones(m, 1) X];

% set initial Theta
theta_i = zeros(n + 1, 1);

% Calculate the initial cost and gradient
[cost, gradient] = costFunction(theta_i, X, y);
```

```
   fprintf('Cost at initial theta (zeros): %f\n', cost);
   fprintf('Gradient at initial theta (zeros): \n');
   fprintf(' %f \n', gradient);

   fprintf('\nProgram paused. Press enter to continue.\n');
   pause;

   %% ============use fminunc built in function to find optimized theta values===========

   %  Set options for fminunc function
   opt = optimset('GradObj', 'on', 'MaxIter', 400);

   %  Run fminunc
   %  get optimal theta values and the cost
   [theta, cost] = ...
   fminunc(@(t)(costFunction(t, X, y)),theta_i, opt);

   % display theta and cost by fminunc function
   fprintf('Cost at theta found by fminunc: %f\n', cost);
   fprintf('theta: \n');
   fprintf(' %f \n', theta);

   % Plot Desission Boundary
   plotDecisionBoundary(theta, X, y);
hold on;
% Labels and Legend
xlabel('Dicease status')
ylabel('trestbps')

legend('disease ', 'no disease ')
hold off;
```

```
fprintf('\nProgram paused. Press enter to continue.\n');

pause;


%% ============= Predict and display Accuracies ==============


probability = sigmoid([1 145 286] * theta);

fprintf(['For a student with scores 145 mm Hg and 286 mg/dl,  predicted status ' ...

         'probability of %f\n\n'], probability);


% Compute accuracy on our training set

p = predict(theta, X);


fprintf('Train Accuracy: %f\n', mean(double(p == y)) * 100);


fprintf('\nProgram paused. Press enter to continue.\n');

pause;
```

2. **costFunction.m**

```
function [J, grad] = costFunction(theta, X, y)
%this function computes the cost of selecting theta as parametr for logistic regression.

%m= number of traing examples
m = length(y);

%
temp1 = -1 * (y .* log(sigmoid(X * theta)));
temp2 = (1 - y) .* log(1 - sigmoid(X * theta));

J = sum(temp1 - temp2) / m;



grad = (X' * (sigmoid(X * theta) - y)) * (1/m);

end
```

3. **sigmoid.m**

```
function g = sigmoid(z)
%This function computes the sigmoid of z

g = zeros(size(z));

% denom=denominator
denom = 1 + exp(-1 * z);
g = 1 ./ denom;

end
```

4. **predict.m**

```
function p = predict(theta, X)
%% this function predic the lables 0 or 1 for the output result using theat
%  and X values

%  if sigmoid(theta'*x) >= 0.5, predict 1) etc..

%m=number of trainig example
m = size(X, 1);

p = zeros(m, 1);

result = sigmoid(X * theta);
p = round(result);

end
```

5. **mapFeature.m**

```
function out = mapFeature(X1, X2)
% this function maps polynomial features in X1,X2

deg = 10;
out = ones(size(X1(:,1)));
for i = 1:deg
   for j = 0:i
      out(:, end+1) = (X1.^(i-j)).*(X2.^j);
   end
end

end
```

6. **plotDecisionBoundary.m**

```
function plotDecisionBoundary(theta, X, y)
%this function plots data points X,Y into a new figure with the decision boundary.
% the input param, X assumed either Mx1 matrix or MxN matrix

% Plot Data
plotData(X(:,1:2), y);
hold on

if size(X, 2) <= 3
   % choose two endpoints to plot the line
   x_mark = [min(X(:,1))-2,  max(X(:,2))+2];

   % Calculate the decision boundary line
   y_mark = (-1./theta(2)).*(theta(1).*x_mark + theta(1));

   % Plot, and adjust axes
   plot(y_mark, x_mark)
   axis([80, 200, 80, 200])

   % Legend
   legend('dicease', 'No dicease', 'Decision Boundary')

else
   % define grid range
   u = linspace(-1, 1.5, 30);
   v = linspace(-1, 1.5, 30);

   z = zeros(length(u), length(v));
   % Evaluate z = theta*x over the grid
   for i = 1:length(u)
      for j = 1:length(v)
         z(i,j) = mapFeature(u(i), v(j)).*theta;
      end
   end
```

```
    z = z';

    % Plot z = 0
    contour(u, v, z, [0, 0], 'LineWidth', 2)
  end
  hold off

  end
```

7. **plotData.m**

```
function plotData(X, y)
%this function plots the data points X and y into a new figure

% Create New Figure
figure; hold on;

positive = find(y == 1);
negative = find(y == 0);

plot(X(positive, 1), X(positive, 2), 'k+', 'LineWidth', 2, 'MarkerSize', 7);
plot(X(negative, 1), X(negative, 2), 'ko', 'MarkerFaceColor', 'y', 'MarkerSize', 7);


hold off;

end
```

--End--