Report for assignment 1:

1. Problem statement: Implementing the logical regression model, which will determine will the patient take part in the rehabilitation program or not.

2. There are five input categories-Reason, Gender, Age, Mobility and Distance, and for each patient there is one output value, determining did the patient take part in the rehabilitation program. Our model should use statistics from the provided data to determine probability will the new patient, regarding his data, take part in the program.

3.For the non-numerical categories (reason gender and mobility) I will use classification model, presenting the female gender as 0, and the male as 1, mobility for no car as 0, and with car with 1, and for the 10 different reasons I will use numbers from 0 to 9. I will use the data in Newton's model, and calculate the theta values from the given data, which I will use in order to make a function to determine the probability of the new person's acceptance.

4. First, I made a script called newton.m in which I'll implement Newton's algorithm for logical regression. Before coding, I had changed the Data.csv file into the Data.txt file, and performed substitutions as explained in figure 3.

In my newton.m script I first read the data from the text file-which gives me the matrix in which all columns represent the data of the one patient. Next I form vectors x (with the input) and y (with the output data) from that matrix. I add ones in the input data as a first column, so when theta is calculated, the zero value stays the same- multiplicities by 1, as in the gradient decent model.

After that I prepare all the matrixes I am going to use:

Theta-matrix that will in the end contain the theta needed for calculating the probabilty,

Old_th-matrix that will save the previous theta values, so we can compare the changes.

In-variable that will calculate have we reached the optimal numbers, according to threshold.

Then I make a while statement, which will work until the given threshold.

In while statement I:

add 1 to number of iterations.

Calculate the gradient for each category with, from the formula:

$$\nabla_\theta J = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

h=(1/(1 + exp(-theta'*x(j,:)')));

gradient(i,1) = gradient(i,1) + (-1)*[y(j,1)-h]*x(j,i);

I calculate Hessian as:

$$H = \frac{1}{m}\sum_{i=1}^{m}\left[h_\theta(x^{(i)})\left(1 - h_\theta(x^{(i)})\right)x^{(i)}\left(x^{(i)}\right)^T\right]$$

h=(1/(1 + exp(-theta'*x(j,:)')));

gradient(i,1) = gradient(i,1) + [y(j,1)-h]*x(j,i);

Then I save the values of old theta for comparison , and calculate new theta:

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla_\theta J$$

theta = theta - inv(H)*gradient;

In the end, I calculate the difference between old and new theta.

In the end of while statement, I have the important theta values in theta vector.

In other script, called calculate.m I implement the function which works with the theta vector made in previous script, and 5 data inputs about the new patient.

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$
$$= P(y = 1|x; \theta)$$

This function turns non-numerical data into numerical values-as explained in 3. and then counts the dot product of theta vector and vector of new data of the new vector, and then performs sigma function to the product, which tells us the result.

5.The result Octave presents shows the probability of the patient's acceptance.

6. I have put the counter called number_of_iterations into the while statement, so after the calculations it shows that the calculation is done in 5 iterations, which is a great optimization if compared to gradient descent which takes hundreds of iterations. So, I can evaluate my solution as a good one, it finishes in a good number of iterations and provides the solution for given data in few seconds.

7. Using the Newton's model for the logical regression issues simplifies the calculations, and using simple sigmoid function, you can easily read the results provided for the new data.

8. How to work with the solution:

Open the folder containing data.txt, newton.m and calculate.m in Octave.

Type newton

Call calculation()-> for example (theta, "Forgot", "F", 76, "Car", 61,5)

Read the result.