**Report**

Newton Raphson

A blue and orange math equations

Description automatically generated with medium confidence

A diagram of a method

Description automatically generated

I started to implement the logistic regression function with newton. Initially as stated in the problem we split the data into 10 folds.

The data is split using the sklearn function test\_train\_split.

Then the training and testing data is generated randomly since we used the random\_state=None. The splitting of data is random.

Then we need to derive theta from the newton Raphson method.  
As in the above images I calculated the gradient and hessian to update the theta in every iteration.

If the theta is converging too small, then we automatically break the iteration.

This algorithm is given in the Pdf provided as in the above images.  
The theta that is derived from the newton function is used in the model to test on the testing data.

The accuracy can also be calculated using accuracy\_score method from sklearn.metrics package.

We got the accuracy of 90%+

References:  
<https://www.cs.cmu.edu/~mgormley/courses/10701-f16/slides/lecture5.pdf>

Stack overflow for error corrections:

<https://stackoverflow.com/questions/24560298/python-numpy-valueerror-operands-could-not-be-broadcast-together-with-shapes>

[python - Convert numpy array of shape (n,1) to shape (n,) - Stack Overflow](https://stackoverflow.com/questions/29735340/convert-numpy-array-of-shape-n-1-to-shape-n)