**Optimization of Data Science**

**Introduction**.

We will be going ahead with predictive analytics using churn prediction models that predict customer churn by assessing their propensity of risk to churn. Since these models generate a small prioritized list of potential defectors, they are effective at focusing customer retention marketing programs on the subset of the customer base who are most vulnerable to churn.

**Straight Down into Code**

Loading the dataset for optimization.

Loading up or importing the required libraries.

Implementing the Label Encoder converting the strings values into numerical values, over here we can also make use of the one hot encoding as well. Since Label Encoder takes lead over one hot encoding and is the reason we had chosen and we are now converting the different ages categorical to simple numerical numbers ranging from 0-7 using label encoder.

Over here we are now training and testing datasets with the variable Y\_Train and Y\_Test values i.e., the (exit)output values drafted to the first 7000 rows as training and the rest with testing. Also the training the input data with the variable X\_train and and testing dataset with X\_Test over the same range 7000:3000 where we have got 10000 rows in total.

**Feature Scaling**

Feature Scaling is being carried out here over both the training and the testing datasets inturn converting whatever type the data it is, will be converted to Float by MinMaxScalar to either 0 or 1.

To make sure and cross check the shapes of all our training and testing data sets and in addition we are also checking if we had the MinMax feature has been applied upon successfully.

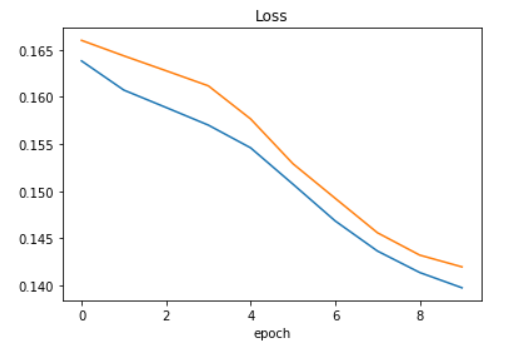
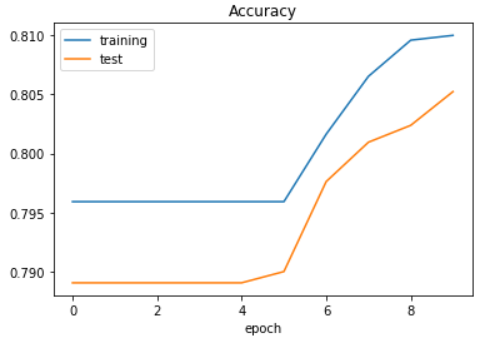
Right after the feature scaling, we split up our training data into another Training and Validation sets to have a self evaluation of how good the training sets had been trained, to be precise with the accuracy of the training datasets we are now shuffling our datasets every iterations and also been batched up in 20.

**Stochastic Gradient Descent**

We make use the Gradient Descent right now, implemented over Advanced Neural Networks trained over 3 layers. 1st layer being the input layer loading the datasets with the Relu function where the negative values wouldn’t be considered and also is half rectified and 2nd will be the hidden layer with the 15 nodes between and then the Softmax function with the sigmoid curve with accurate prediction of output, represent probabilities each number’s value is between 0 and 1 valid value range of probabilities.

Right now is where we make prediction, with the training and testing datasets of both the input and output (X and Y variables) values. Declaring the number of batch size for every iteration to be carried out, and the number of epochs meaning the total number of entire iteration and shuffling it for every iteration.

A graph which clearly represents the curves of Loss over training and testing data sets over every iteration. Similarly a graph on the Accuracy of the training and the testing data sets.

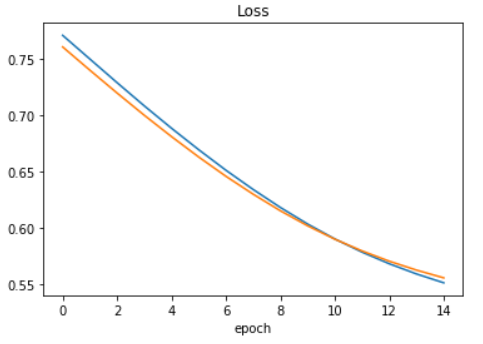
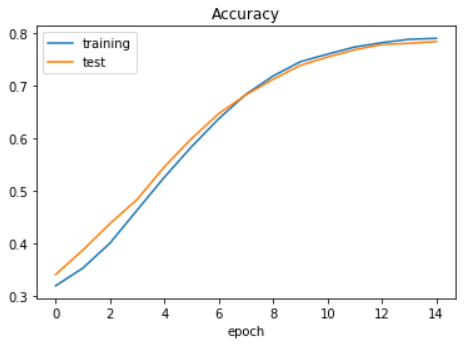
 

**Adadelta Model**

Implemented over the same Neural Network with a new optimizer named Adadelta with the 3 layers in total. 1st layer being the input layer loading the datasets with the Relu function where the negative values wouldn’t be considered and also is half rectified and 2nd will be the hidden layer with the 15 nodes between and then the Softmax function with the sigmoid curve with accurate prediction of output, represent probabilities each number’s value is between 0 and 1 valid value range of probabilities.

Now comes the prediction, with the training and testing datasets of both the input and output (X and Y variables) values. Declaring the number of batch size for every iteration to be carried out, and the number of epochs meaning the total number of entire iteration and shuffling it for every iteration.

A graph which clearly represents the curves of Loss over training and testing data sets over every iteration. Similarly a graph on the Accuracy of the training and the testing data sets.

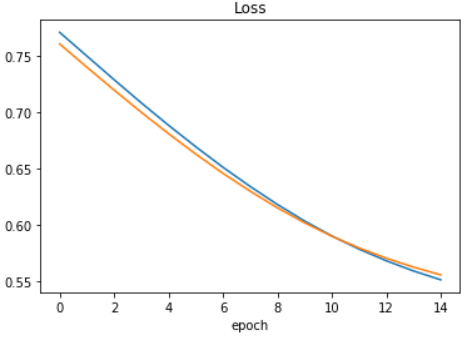
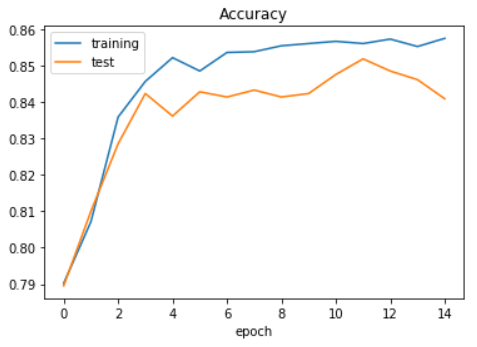
 

**Adam Model**

We have used the Adam optimizer with the same number of layer using the Neural Networks 1st layer being the input layer loading the datasets with the Relu function where the negative values wouldn’t be considered and also is half rectified and 2nd will be the hidden layer with the 15 nodes between and then the Softmax function with the sigmoid curve with accurate prediction of output, represent probabilities each number’s value is between 0 and 1 valid value range of probabilities.

There we go with prediction, with the training and testing datasets of both the input and output (X and Y variables) values. Declaring the number of batch size for every iteration to be carried out, and the number of epochs meaning the total number of entire iteration and shuffling it for every iteration.

A graph which clearly represents the curves of Loss over training and testing data sets over every iteration. Similarly a graph on the Accuracy of the training and the testing data sets.

**Conclusion**

To conclude with we measure the accuracy scores of the prediction for ALL optimizers, and these are with the best precision of our model trained and tested with different functions implemented.

