WINTER 2021, ALY6020

Predictive Analytics

INSTRUCTOR – Dr. Marco Montes de Oca

****

 Logistic Regression

Submitted by:

Reetika Chaturvedi

**OVERVIEW**

In this task we were asked to implement a basic image recognition system using logistic regression. We were given a dataset that contained instances of 10 digits available (0-9) and were asked to train and test ten different logistic regression models. Two datasets were provided (training and test data) that contained grayscale images. There were 785 columns in the dataset. The first column represented the digit labels (0-9) and the remaining columns represented pixel values of the image

**APPROACH**

Logistic regression is useful to predict a binary outcome from a set of repeated predictor variables.

To complete the given task, caret and dummies package were used. After loading the data, I viewed the image with the help of code given in the assignment. Thereafter, binary encoding is done to remove the weights to replace with a Boolean value.

In order to complete the task, I performed data sampling and selected 10000 observations from the training data. A model was run for each training set (0-9) for prediction of each class of the test data. The outcome of these models was converted to a probability distribution of ten classes in order to change the probabilities that were not related. For this, I used the softmax function in R.

The class that will have the highest probability is considered as the best prediction.

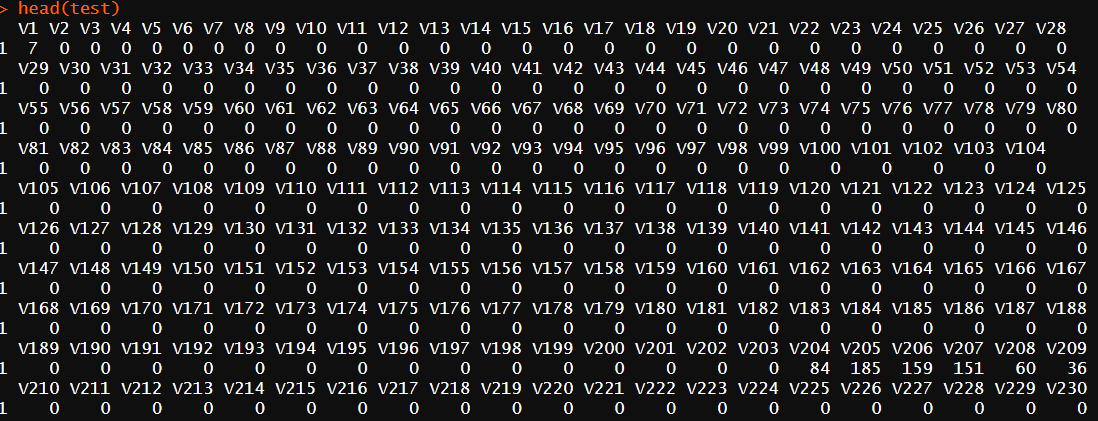
**USE OF SOFTMAX FUNCTION**

This is a normalized exponential function that helped to change the vector to binary form. It was useful as it allowed to interpret the values as probabilities.

**TRAINING DATA**

The training data has 785 variables. The first column is the label of the image which is dummy coded. It will be 1 when it will be from a particular label else it will be a zero. A random sample of 10,000 was taken for modeling and dummy variables were concatenated with the train data.

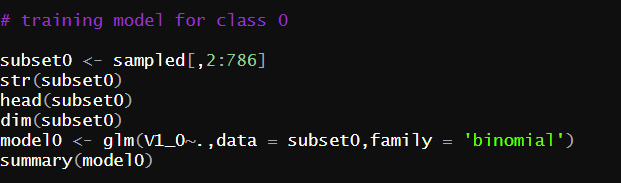
The structure of training data looked like this:



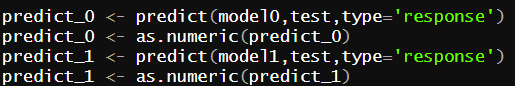
After this binary coding was done to create dummy variables.

**TRAINING THE MODEL**

The model was trained using glm function. The concatenated data was subset for fitting the model for every class so that our data contains the labels in the binary form and the pixels.



After training the data for label 0, predictions were made. The probabilities vector was created for the test set using the following code.



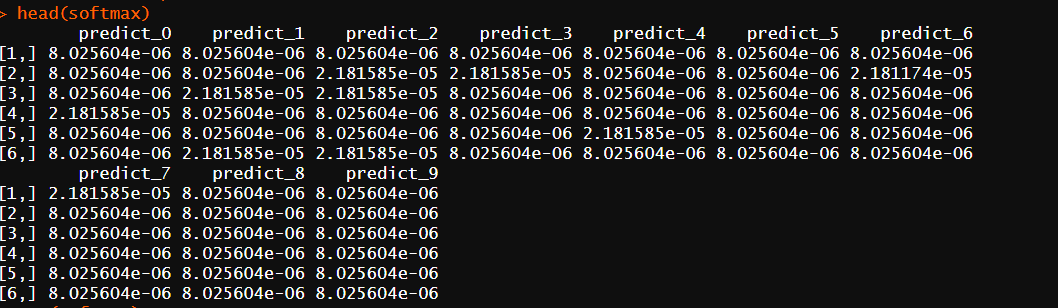
The predicted probabilities vectors were concatenated using cbind.

z=cbind(predict\_0,predict\_1,predict\_2,predict\_3,predict\_4,predict\_5,predict\_6,predict\_7,predict\_8,predict\_9)

dim(z)

str(z)

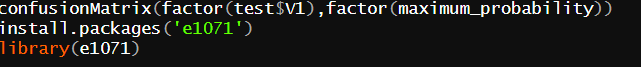
**RESULT AFTER USING SOFTMAX FUNCTION**

Softmax= exp(z)/sum(exp(z))

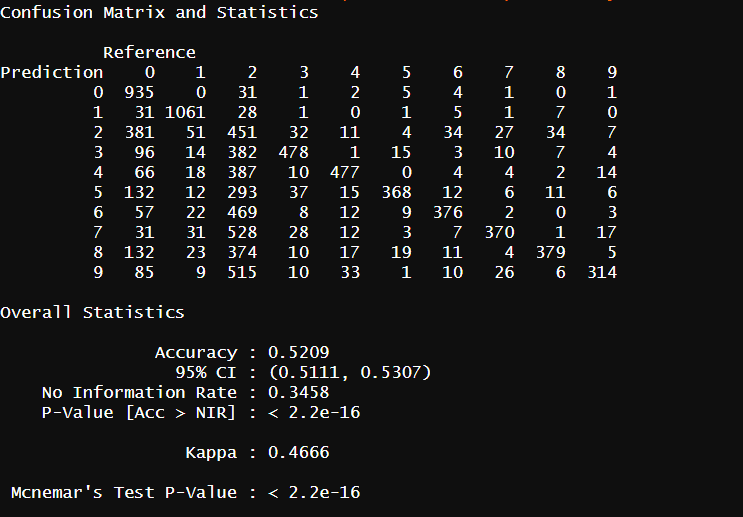
Column names with the maximum row values were extracted in the following way:



After this confusion matrix was created between the actual and predicted data.



The confusion matrix represents that how our predictions match the actual data. In this case our model yielded an accuracy of 52%. It had been better if complete training data would have been used.



**REFERENCES**

* <https://stackoverflow.com/questions/17735859/for-each-row-return-the-column-name-of-the-largest-value>
* <https://www.pluralsight.com/guides/encoding-data-with-r>