**Machine**

**Learning**

**of**

**SpamBase**

**Dataset**

**Research**

**using this dataset:**

**https://archive.ics.uci.edu/ml/datasets/spambase**

**Code URL:**

**https://github.com/DogDogBird/Machine-Learning.git**

**Subject Areas:**

Machine Learning - tensorflow

**Keywords:**

Spambase, MLP, Deep Learning, tensorflow

Author for correspondence:

**Kyubin Kyoung**

**kyubin0704@gmail.com**

|  |
| --- |
| **The Usage of spambase Dataset to get Higher Accuracy and Precision** |

|  |
| --- |
| **Kyubin Kyong[[1]](#endnote-1)** |

|  |
| --- |
| **While using e-mails, we each have spam mail boxes where the spams go. However sometimes, spam that should go into the spam mailbox goes to normal mailbox and the other applies too. To get more accuracy, spam goes to spam mailbox, ham goes to ham mailbox, using ML will be an answer. So far there exists a ML code in github that uses multinominal method.** |

**1. BackGround**

Today there are lots of small websites trying to get users such as gambling or adult site. Even the major companies need more users. The one way they are getting users is by sending e-mails. Users call those unpleasant mails spam mails.

There are so many datasets in UCI Website and there existed a dataset that handles information about spam mail. Consisted of the specific word frequency especially like !, $, #, etc..

On the website, there are some codes that handles this spambase.csv dataset. They handle this data using multinominal method[[2]](#endnote-2). They are just getting a result of accuracy. So the thing I tried to do is using MLP(Multi Layer Perceptron) with 4~6 hidden layers and get precision as well as accuracy.

**2. Dataset Structure[[3]](#endnote-3)**

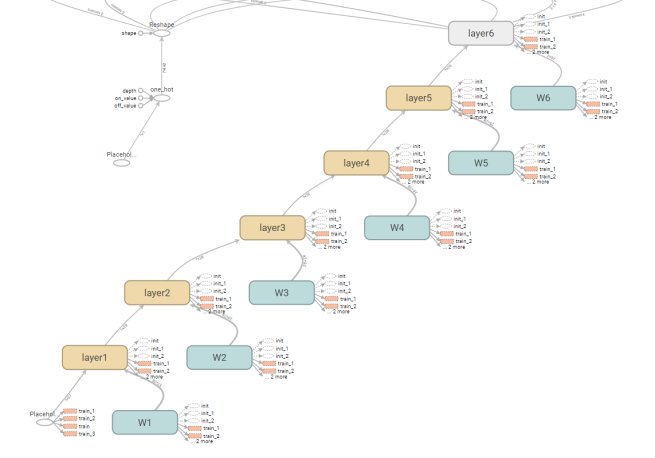
There are 58 rows. Data in 1~54 row means the frequency of the specific term.

Data in 55~57 row means the average, maximum and the total value of the capital run\_length\_encoding.

The last row means if it is Spam or not.

However, there was a problem using this raw dataset. So the dataset needed normalization. Dividing data into the largest number of the dataset has been an answer. I divided 57\*4601 data into 15841 and the data became normalized between 0 to 1.

**3. Why using MLP?**



(Fig. 1) Layers in Tensor Board

MLP is abbreviation of Multi Layer Perceptron. It means as there are a neural network called perceptron, MLP is going to use multiple perceptrons. We call that hidden layer. So I really works like human neural network. And if there are lots of hidden layers this is called Deep Learning.

The reason I am using MLP is because this dataset is frozen. As there are 4601.labels, this number is not too big, and there are just two classes, I didn't need to use CNN.

There are 5 hidden layers and as my data vector is [57,2] I am using [57,28], four [28,28] and [28,2] layer.

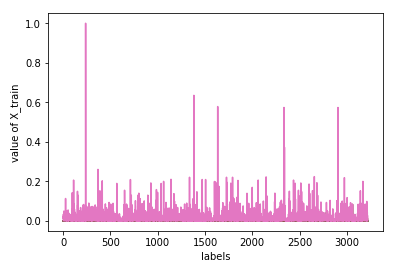


Fig 2. X\_train Data

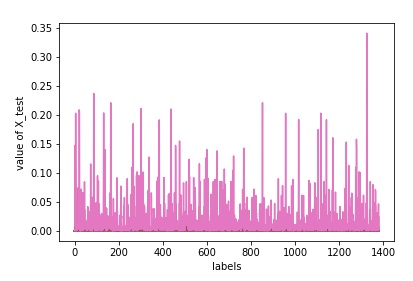


Fig 3, X\_test Data

Training sets and test sets are splitted from one dataset. training sets handle 70% of the data and the remains are test sets. So there are 3220 training sets and 1381 test sets.

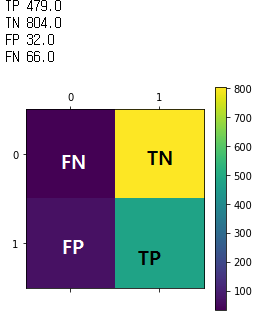
However using only this data caused an error. Cost became zero. Using one\_hot\_encoding was the solution to this problem.

One hot encoding is a process by which categorical variables are converted into a form that could be provided to ML algorithms to do a better job in prediction.

As there are lots of optimizer algorithm, I chose the best working algorithm.

|  |  |
| --- | --- |
| Optimizer | Accuracy |
| AdadeltaOptimizer | 0.605358 |
| AdagradOptimizer | 0.605358 |
| AdamOptimizer | 0.917451 |
| FtrlOptimizer | 0.605358 |
| ProximalGradientDescentOptimizer | 0.605358 |

The AdamOptimizer does the best work making Accuracy up to 91%. So calculating precision with this optimizer I could get the confusion matrix.



(Fig. 4) confusion matrix

**4. Result**

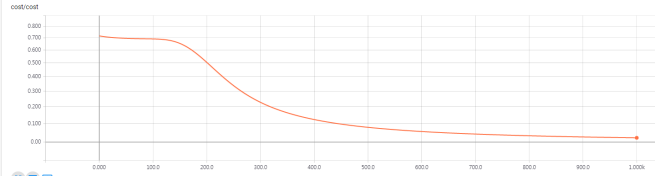
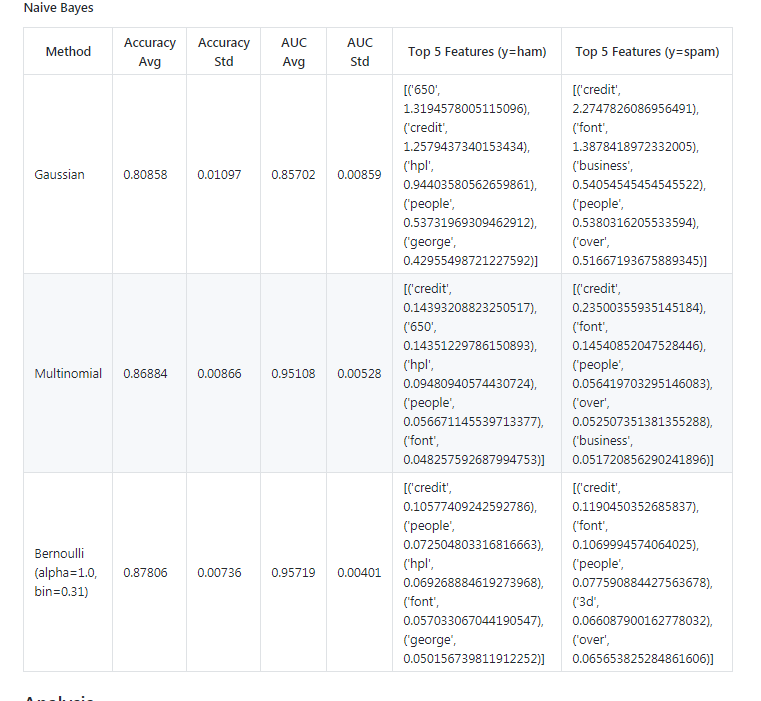


fig 5. cost

Cost gradiently descents while the time flows.

The precision is 0.937378 by the equation

***Precision = TP / TP + FP***



**fig 6. data shown at**

<https://github.com/sampepose/SpamClassifier/blob/master/my_test.py>

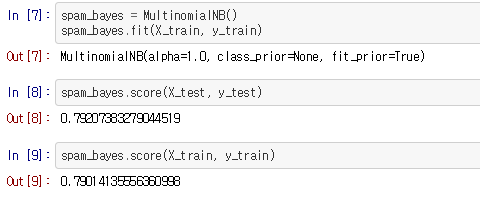


fig 7. data shown at

<https://github.com/JonathanKross/spambase/blob/master/spamalot.ipynb>

The accuracy also increased using MLP than using Multinominal such as in fig.5 and fig.6.

5. Conclusion

This is the best accuracy using the UCI spambase dataset. Using MLP is better than Multinominal.

However to get better result, we need more attribute and labels. Train more data, and testing will be much better. As a result we can defeat spam mails!

1. 3rd Grade. Kyunghee University student. Korea [↑](#endnote-ref-1)
2. <https://github.com/sampepose/SpamClassifier/blob/master/my_test.py>, <https://github.com/JonathanKross/spambase/blob/master/spamalot.ipynb> [↑](#endnote-ref-2)
3. Issued in Readme.txt [↑](#endnote-ref-3)