NAIVE BAYES VS. MAX ENTROPY

I HAVE MERGED AND COLLECTED MOST OF THE NEEDED READMES IN THIS PDF IF YOU CAN NOT FIND A README IN FOLDERS YOU CAN SEE THE EXPLANATION OF THAT README HERE ...

The results of naive bayes can be found in the directory shown below:
NLP97982\P2\ClsModel\NaiveBayes\Test.report.txt

And the results are:

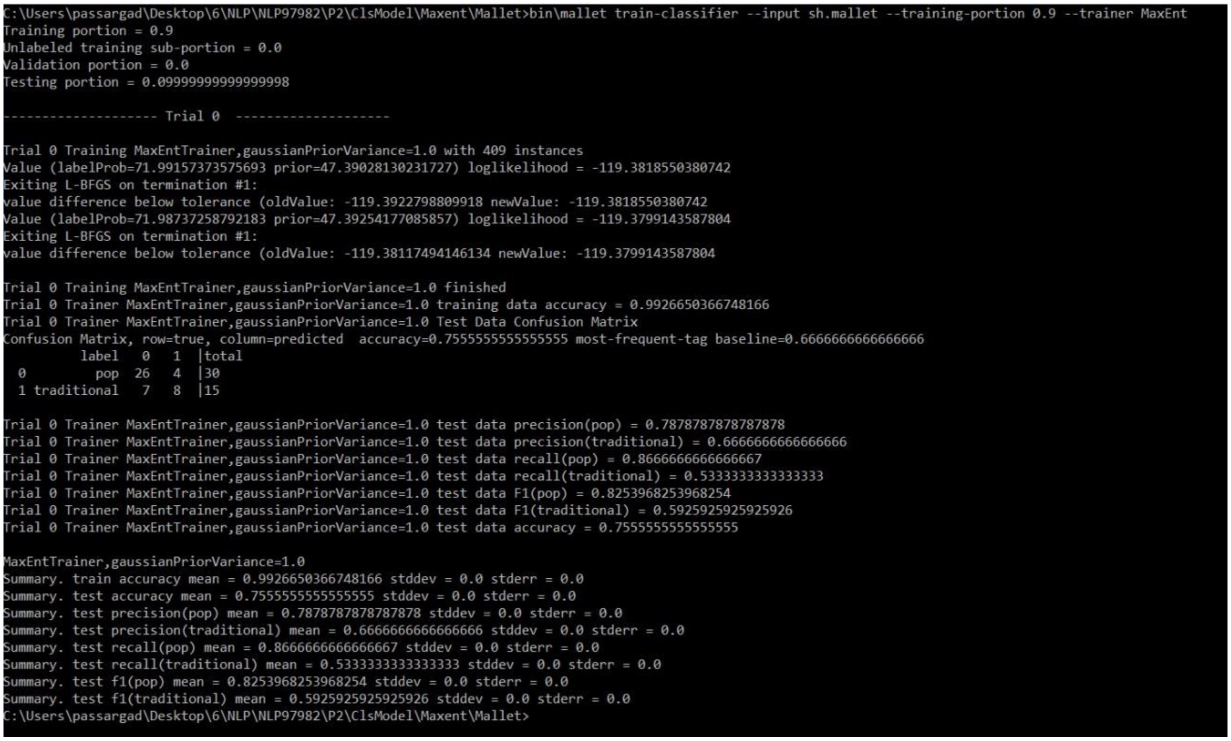
Pop accuracy percentage : 93.17180616740089
Pop Precision percentage : 93.38235294117648
Pop Recall percentage : 95.13108614232209
Pop F1 percentage : 94.24860853432281

Traditional accuracy percentage : 93.17180616740089
Traditional Precision : 92.85714285714286
Traditional Recall : 90.37433155080214
Traditional F1 : 91.59891598915989

As MaxEnt needs features first of all i have used unigram as feature and made a file in format of

--label feature:value --
that label is one of the {pop-traditional} labels
feature contains every unigram word in each sentence
value: for each word appearing in a sentence the value is 1 and if sentence includes a word twice that word comes twice with value of 1.

By using mallet, making model, training and testing it by mallet commands the results became:



```
C:\Users\passargad\Desktop\6\NLP\NLP97982\P2\ClsModel\Maxent\Mallet>bin\mallet train-classifier --input sh.mallet --training-portion 0.9 --trainer MaxEnt
Training portion = 0.9
Unlabeled training sub-portion = 0.0
Validation portion = 0.0
Testing portion = 0.09999999999999998

-------------------- Trial 0  --------------------

Trial 0 Training MaxEntTrainer,gaussianPriorVariance=1.0 with 409 instances
Value (labelProb=71.99157373575693 prior=47.39028130231727) loglikelihood = -119.3818550380742
Exiting L-BFGS on termination #1:
value difference below tolerance (oldValue: -119.3922798809918 newValue: -119.3818550380742
Value (labelProb=71.98737258792183 prior=47.39254177085857) loglikelihood = -119.3799143587804
Exiting L-BFGS on termination #1:
value difference below tolerance (oldValue: -119.38117494146134 newValue: -119.3799143587804

Trial 0 Training MaxEntTrainer,gaussianPriorVariance=1.0 finished
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 training data accuracy = 0.9926650366748166
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 Test Data Confusion Matrix
Confusion Matrix, row=true, column=predicted  accuracy=0.7555555555555555 most-frequent-tag baseline=0.6666666666666666
          label   0   1  |total
  0         pop  26   4  |30
  1 traditional   7   8  |15

Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data precision(pop) = 0.7878787878787878
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data precision(traditional) = 0.6666666666666666
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data recall(pop) = 0.8666666666666667
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data recall(traditional) = 0.5333333333333333
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data F1(pop) = 0.8253968253968254
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data F1(traditional) = 0.5925925925925926
Trial 0 Trainer MaxEntTrainer,gaussianPriorVariance=1.0 test data accuracy = 0.7555555555555555

MaxEntTrainer,gaussianPriorVariance=1.0
Summary. train accuracy mean = 0.9926650366748166 stddev = 0.0 stderr = 0.0
Summary. test accuracy mean = 0.7555555555555555 stddev = 0.0 stderr = 0.0
Summary. test precision(pop) mean = 0.7878787878787878 stddev = 0.0 stderr = 0.0
Summary. test precision(traditional) mean = 0.6666666666666666 stddev = 0.0 stderr = 0.0
Summary. test recall(pop) mean = 0.8666666666666667 stddev = 0.0 stderr = 0.0
Summary. test recall(traditional) mean = 0.5333333333333333 stddev = 0.0 stderr = 0.0
Summary. test f1(pop) mean = 0.8253968253968254 stddev = 0.0 stderr = 0.0
Summary. test f1(traditional) mean = 0.5925925925925926 stddev = 0.0 stderr = 0.0
C:\Users\passargad\Desktop\6\NLP\NLP97982\P2\ClsModel\Maxent\Mallet>
```

As p result is not very good and accuracy is about 75% which is not as we expect from MaxEnt. So MaxEnt needed mp