Naive Bayes with Bernoulli and Binomial Features

1. Problem Statement

The multi feature data set with two classes should be classified using generative learning Naïve Bayes model with Bayes and Binomial features. The data should be trained and tested using the generative learning model. The 10-fold cross validation has to be done to analyze the performance of the model. The accuracy of the model is analyzed with the training and testing error calculated by Mean Square Error.

2. Problem Solution

The solution is to find the parameters using Naïve Bayes model. The parameters are the prior probability and the likelihood for each feature in each class. In Bernoulli's the features are binary and in Binomial the features are discrete. Therefore the way of calculating the parameters are different.

3. Implementation Details

In Bernoulli I have binarized a continuous data set. The steps for implementation is as same as Bayes model except for the difference in the way of find the parameters.

In Binomial the text data is converted to discrete data with no of occurrences of the word. I have shortlisted words for being the features.

I have implemented the code in ipython notebook. The filename has to be mentioned and in the tool bar option "Cell" -> "Run All" will implemented the whole file and the results will be printed.

4. Results and Discussion

Naïve Bayes with Bernoulli Features

The data set used is **spambase.data.**

a) Find the classes and prior probability

```
The Classses are [0, 1]
The Classes Count  {0: 2788, 1: 1813}
The prior probabiliy {0: 0.6059552271245382, 1: 0.39404477287546186}
```

The classes represent the class labels, the classes count is the number of training data in each class. The prior probability is the probability distribution of each class.

b) In order to avoid the probability being zero laplace smoothing is done and the mean of classes are below

```
Smoothed Mean for mulivariate features
0
[ 0.09820789  0.27741935  0.00322581  0.22043011  0.11433692  0.01577061
  0.07383513  0.07849462  0.17060932  0.05125448  0.42401434  0.11935484
  0.04551971  0.01792115  0.090681    0.09569892  0.12580645  0.58064516
  0.0172043   0.34336918  0.00824373  0.02795699  0.01971326  0.37311828
  0.28136201  0.27706093  0.15555556  0.12939068  0.16200717  0.10430108
  0.07311828  0.12365591  0.07383513  0.15770609  0.17491039  0.26129032
  0.01863799  0.11577061  0.09032258  0.05304659  0.11541219  0.10430108
  0.10071685  0.29569892  0.16129032  0.01612903  0.06738351  0.18637993
  0.5516129   0.14336918  0.26810036  0.1046595   0.08243728  0.99964158
  0.99964158  0.99964158]

1
 [ 0.34490358  0.61487603  0.02203857  0.62534435  0.37575758  0.4214876
  0.3415978   0.30633609  0.45619835  0.31294766  0.63030303  0.28705234
  0.12782369  0.15867769  0.54545455  0.384573    0.37961433  0.88650138
  0.20826446  0.80826446  0.05289256  0.3322314   0.37575758  0.02809917
  0.015427    0.00495868  0.01707989  0.00716253  0.01046832  0.00220386
  0.00165289  0.03415978  0.00606061  0.02589532  0.06225895  0.05619835
  0.01818182  0.03471074  0.11184573  0.00110193  0.01157025  0.04738292
  0.02644628  0.26887052  0.03801653  0.01101928  0.00936639  0.14986226
  0.64903581  0.07217631  0.83305785  0.61157025  0.28760331  0.99944904
  0.99944904  0.99944904]
```

The mean for each class is calculated based on the formula stated above.

After find the parameter it's used in membership function to predict the class label. Below is the sample code for membership.

c) Prediction over the entire dataset and below is the sample prediction when the prediction is done over entire training dataset,

```
Predicted Value      True Value
1                              [1]
1                              [1]
1                              [1]
1                              [1]
1                              [1]
1                              [1]
0                              [1]
1                              [1]
1                              [1]
1                              [1]
0                              [1]
```

d) Cross Validation 10 fold

```
Fold 0 Accuracy 0.889370932755
Fold 1 Accuracy 0.89347826087
Fold 2 Accuracy 0.90652173913
Fold 3 Accuracy 0.889130434783
Fold 4 Accuracy 0.889130434783
Fold 5 Accuracy 0.880434782609
Fold 6 Accuracy 0.913043478261
Fold 7 Accuracy 0.882608695652
Fold 8 Accuracy 0.886956521739
Fold 9 Accuracy 0.865217391304
Average Accuracy
0.889589267189
```

e) Cross Validation 10 fold MSE

```
Fold 0 Testing Error [ 0.10629067]
Fold 1 Testing Error [ 0.09565217]
Fold 2 Testing Error [ 0.12608696]
Fold 3 Testing Error [ 0.14130435]
Fold 4 Testing Error [ 0.10869565]
Fold 5 Testing Error [ 0.09130435]
Fold 6 Testing Error [ 0.09565217]
Fold 7 Testing Error [ 0.13043478]
Fold 8 Testing Error [ 0.09782609]
Fold 9 Testing Error [ 0.11304348]
Average Mean Square Error
Training Error        Testing Error
0.110145155025     0.110629067245
```

f) Evaluation over the entire data set

```
Confusion Matrix
[[2594  194]
 [ 314 1499]]
Accuracy 0.889589219735
Precision
[0.89202200825309486, 0.88541051388068515]
Recall
[0.93041606886657102, 0.82680639823496971]
F_score
[0.91081460674157311, 0.85510553337136341]
```

g) Precision through the cross validation

```
Precision
0.85858585858585856   0.92073170731707321
0.90666666666666662   0.89375000000000004
0.91078066914498146   0.89005235602094246
0.88513513513513509   0.92682926829268297
0.90969899665551834   0.86956521739130432
0.88235294117647056   0.87765957446808507
0.89198606271777003   0.8497109826589595
0.8896551724137931   0.85882352941176465
0.89456869009584661   0.87755102040816324
0.89323843416370108   0.87709497206703912
Average Precision
0.892266863     0.884176863
```

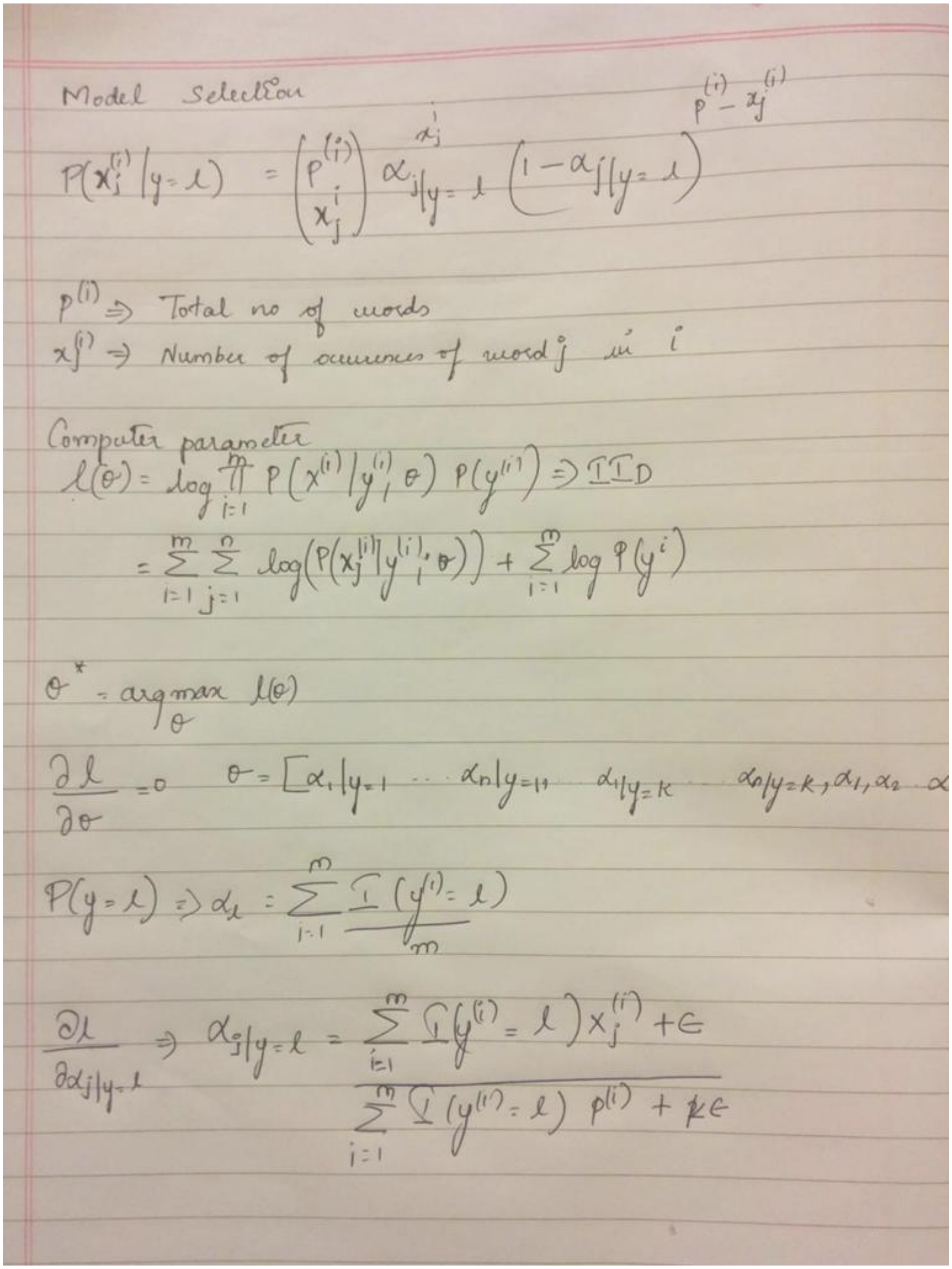
h) Recall from Cross Validation

```
Recall
0.95149253731343286   0.78238341968911918
0.94117647058823528   0.83625730994152048
0.92105263157894735   0.87628865979381443
0.9562043795620438   0.81720430107526887
0.92832764505119458   0.83832335329341312
0.9125475285171103   0.8375634517766497
0.90780141843971629   0.8258426966292135
0.91489361702127658   0.8202247191011236
0.93959731543624159   0.79629629629629628
0.91941391941391937   0.83957219251336901
Recall
0.929250746           0.82699564
```

Naïve Bayes with Binomial Feature

The data set used is **imdb_labelled.txt**

Finding Parameters



Model Selection

$$P(x_j^{(i)} | y = l) = \binom{p^{(i)}}{x_j^{i}} \alpha_{j|y=l}^{x_j^{i}} \left(1 - \alpha_{j|y=l}\right)^{p^{(i)} - x_j^{(i)}}$$

$p^{(i)} \Rightarrow$ Total no of words

$x_j^{(i)} \Rightarrow$ Number of occurences of word j in i

Computer parameter

$$l(\theta) = \log \prod_{i=1}^{m} P(x^{(i)} | y^{(i)}; \theta)\, P(y^{(i)}) \Rightarrow \underline{I}\,\underline{I}D$$

$$= \sum_{i=1}^{m} \sum_{j=1}^{n} \log\left(P(x_j^{(i)} | y^{(i)}; \theta)\right) + \sum_{i=1}^{m} \log P(y^{i})$$

$$\theta^{*} = \arg\max_{\theta} l(\theta)$$

$$\frac{\partial l}{\partial \theta} = 0 \qquad \theta = [\alpha_{1}|y=1 \ldots \alpha_{n}|y=1, \; \alpha_{1}|y=k \ldots \alpha_{n}|y=k, \alpha_1, \alpha_2 \ldots \alpha$$

$$P(y = l) \Rightarrow \alpha_{l} = \sum_{i=1}^{m} \frac{I(y^{(i)} = l)}{m}$$

$$\frac{\partial l}{\partial \alpha_{j|y=l}} \Rightarrow \alpha_{j|y=l} = \frac{\sum_{i=1}^{m} I(y^{(i)} = l)\, x_j^{(i)} + \epsilon}{\sum_{i=1}^{m} I(y^{(i)} = l)\, p^{(i)} + k\epsilon}$$

a) Find the classes and prior probability

```
The Classses are [0, 1]
The Classes Count  {0: 500, 1: 500}
The prior probabiliy {0: 0.5, 1: 0.5}
```

The classes represent the class labels, the classes count is the number of training data in each class. The prior probability is the probability distribution of each class.

Since there are huge number of feature I am printing the mean for all the features

b) Prediction over the entire dataset and below is the sample prediction when the prediction is done over entire training dataset,

```
Predicted Value     True Value
1                           [1]
1                           [1]
1                           [1]
1                           [1]
1                           [1]
0                           [0]
1                           [1]
1                           [1]
1                           [1]
1                           [1]
1                           [1]
```

c) Cross Validation 10 fold

```
Fold 0 Accuracy 0.933333333333
Fold 1 Accuracy 1.0
Fold 2 Accuracy 1.0
Fold 3 Accuracy 0.933333333333
Fold 4 Accuracy 1.0
Fold 5 Accuracy 0.933333333333
Fold 6 Accuracy 1.0
Fold 7 Accuracy 0.933333333333
Fold 8 Accuracy 1.0
Fold 9 Accuracy 1.0
Average Accuracy  0.973333333333
```

d) Cross Validation 10 fold MSE

```
Fold 0 Testing Error [ 0.22]
Fold 1 Testing Error [ 0.17]
Fold 2 Testing Error [ 0.16]
Fold 3 Testing Error [ 0.21]
Fold 4 Testing Error [ 0.17]
Fold 5 Testing Error [ 0.26]
Fold 6 Testing Error [ 0.25]
Fold 7 Testing Error [ 0.22]
Fold 8 Testing Error [ 0.26]
Fold 9 Testing Error [ 0.18]
Average Mean Square Error
Training Error          Testing Error
0.0436666666667    0.21
```

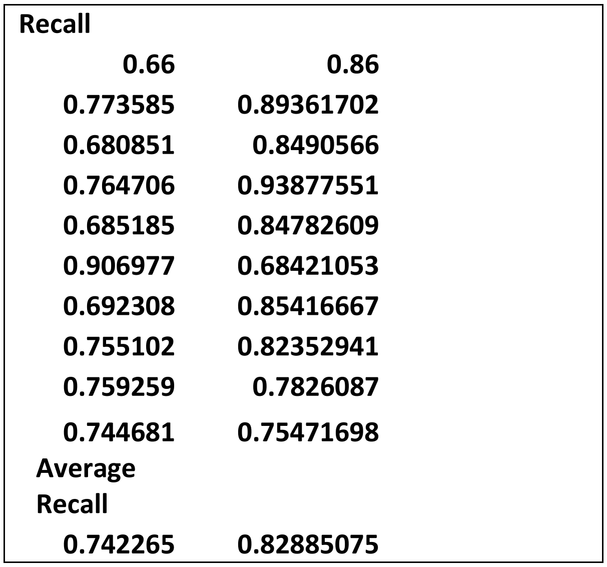
e) Evaluation over the entire data set

```
Confusion Matrix
[[473  27]
 [ 16 484]]
Accuracy 0.957
Precision
 [0.96728016359918201, 0.94716242661448136]
Recall
[0.94599999999999995, 0.96799999999999997]
F_score
 [0.95652173913043481, 0.95746785361028686]
```

f) Precision through the cross validation

```
Precision
  0.825                     0.716666667
  0.891304348               0.777777778
  0.8                              0.75
  0.928571429               0.793103448
  0.840909091               0.696428571
  0.684210526               0.906976744
  0.837209302               0.719298246
  0.804347826               0.777777778
  0.803921569               0.734693878
  0.729166667               0.769230769
  Average Precision
  0.814464076               0.764195388
```

g) Recall from Cross Validation



Recall	
0.66	0.86
0.773585	0.89361702
0.680851	0.8490566
0.764706	0.93877551
0.685185	0.84782609
0.906977	0.68421053
0.692308	0.85416667
0.755102	0.82352941
0.759259	0.7826087
0.744681	0.75471698
Average Recall	
0.742265	0.82885075

5. References

https://en.wikipedia.org/wiki/Generative_model