**Information – SL Assignment (Naïve bayes)**

**OptDigits Dataset:**

* Status - Completed.
* It was an easy implementation of Naïve Bayes on the OptDigits dataset. I have first given indices to all the attributes and labelled the class attribute as Target so that WEKA can understand it.
* I have used multiple algorithms in the Naïve Bayes domain.
* **Naïve Bayes:** Class for a Naive Bayes classifier using estimator classes. Numeric estimator precision values are chosen based on analysis of the training data. For this reason, the classifier is not an UpdateableClassifier (which in typical usage are initialized with zero training instances) -- if you need the UpdateableClassifier functionality, use the NaiveBayesUpdateable classifier. The NaiveBayesUpdateable classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.
* **Naïve Bayes Updateable:** Class for a Naive Bayes classifier using estimator classes. This is the updateable version of NaiveBayes. This classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.
* **Naïve Bayes Multinomial:** The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.
* **Naïve Bayes Multinomial Updateable:** Class for building and using a multinomial Naive Bayes classifier. For more information see, Andrew Mccallum, Kamal Nigam: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI-98 Workshop on 'Learning for Text Categorization', 1998. The core equation for this classifier:
  + P[Ci|D] = (P[D|Ci] x P[Ci]) / P[D] (Bayes rule) where Ci is class i and D is a document.
* **Bayes Network:** Bayes Network learning using various search algorithms and quality measures. Base class for a Bayes Network classifier. Provides datastructures (network structure, conditional probability distributions, etc.) and facilities common to Bayes Network learning algorithms like K2 and B.
  + For more information see:
  + [http://sourceforge.net/projects/weka/files/documentation/WekaManual-3-7-0.pdf /download](http://sourceforge.net/projects/weka/files/documentation/WekaManual-3-7-0.pdf%20/download)
* Results and Observations:
* Naïve Bayes results:

=== Summary ===

Correctly Classified Instances 1607 89.4268 %

Incorrectly Classified Instances 190 10.5732 %

Kappa statistic 0.8825

Mean absolute error 0.0214

Root mean squared error 0.1425

Relative absolute error 11.9131 %

Root relative squared error 47.5052 %

Total Number of Instances 1797

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

173 0 0 0 3 1 0 0 1 0 | a = 0

0 140 14 0 2 0 5 0 8 13 | b = 1

0 5 158 0 1 0 0 0 12 1 | c = 2

0 0 3 142 0 3 0 6 23 6 | d = 3

0 1 0 0 172 1 1 2 3 1 | e = 4

0 0 1 2 2 171 1 1 3 1 | f = 5

0 2 0 0 4 0 174 0 1 0 | g = 6

0 0 1 0 3 0 0 173 1 1 | h = 7

0 15 1 0 0 0 0 1 148 9 | i = 8

0 3 0 9 2 1 0 1 8 156 | j = 9

* Naïve Bayes Multinomial

=== Summary ===

Correctly Classified Instances 1598 88.926 %

Incorrectly Classified Instances 199 11.074 %

Kappa statistic 0.877

Mean absolute error 0.022

Root mean squared error 0.1456

Relative absolute error 12.2393 %

Root relative squared error 48.5294 %

Total Number of Instances 1797

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

174 0 0 0 4 0 0 0 0 0 | a = 0

0 132 19 0 0 1 6 0 9 15 | b = 1

1 7 155 0 0 0 0 1 11 2 | c = 2

1 0 2 154 0 2 0 6 9 9 | d = 3

0 1 0 0 173 0 0 3 3 1 | e = 4

0 0 0 0 1 165 1 0 2 13 | f = 5

0 4 0 0 4 1 172 0 0 0 | g = 6

0 1 0 0 1 0 0 173 3 1 | h = 7

0 21 1 0 1 0 1 1 140 9 | i = 8

0 1 0 6 4 1 0 1 7 160 | j = 9

* Naïve Bayes Updateable

=== Summary ===

Correctly Classified Instances 1607 89.4268 %

Incorrectly Classified Instances 190 10.5732 %

Kappa statistic 0.8825

Mean absolute error 0.0214

Root mean squared error 0.1425

Relative absolute error 11.9131 %

Root relative squared error 47.5052 %

Total Number of Instances 1797

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

173 0 0 0 3 1 0 0 1 0 | a = 0

0 140 14 0 2 0 5 0 8 13 | b = 1

0 5 158 0 1 0 0 0 12 1 | c = 2

0 0 3 142 0 3 0 6 23 6 | d = 3

0 1 0 0 172 1 1 2 3 1 | e = 4

0 0 1 2 2 171 1 1 3 1 | f = 5

0 2 0 0 4 0 174 0 1 0 | g = 6

0 0 1 0 3 0 0 173 1 1 | h = 7

0 15 1 0 0 0 0 1 148 9 | i = 8

0 3 0 9 2 1 0 1 8 156 | j = 9

* Naïve Bayes Multinomial Updateable

=== Summary ===

Correctly Classified Instances 1598 88.926 %

Incorrectly Classified Instances 199 11.074 %

Kappa statistic 0.877

Mean absolute error 0.022

Root mean squared error 0.1456

Relative absolute error 12.2393 %

Root relative squared error 48.5294 %

Total Number of Instances

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

174 0 0 0 4 0 0 0 0 0 | a = 0

0 132 19 0 0 1 6 0 9 15 | b = 1

1 7 155 0 0 0 0 1 11 2 | c = 2

1 0 2 154 0 2 0 6 9 9 | d = 3

0 1 0 0 173 0 0 3 3 1 | e = 4

0 0 0 0 1 165 1 0 2 13 | f = 5

0 4 0 0 4 1 172 0 0 0 | g = 6

0 1 0 0 1 0 0 173 3 1 | h = 7

0 21 1 0 1 0 1 1 140 9 | i = 8

0 1 0 6 4 1 0 1 7 160 | j = 9

* Bayes Network

=== Summary ===

Correctly Classified Instances 1621 90.2059 %

Incorrectly Classified Instances 176 9.7941 %

Kappa statistic 0.8912

Mean absolute error 0.0202

Root mean squared error 0.1315

Relative absolute error 11.2468 %

Root relative squared error 43.8352 %

Total Number of Instances 1797

=== Confusion Matrix ===

a b c d e f g h i j <-- classified as

174 0 0 0 3 1 0 0 0 0 | a = 0

0 150 15 1 0 0 1 0 3 12 | b = 1

0 7 154 2 0 0 0 1 9 4 | c = 2

0 1 2 157 0 1 0 8 5 9 | d = 3

0 3 0 0 172 0 0 2 3 1 | e = 4

0 0 0 1 2 171 1 0 1 6 | f = 5

2 4 0 0 1 1 173 0 0 0 | g = 6

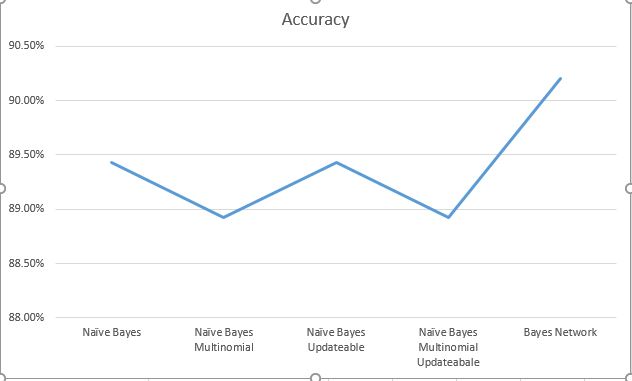
0 0 0 0 9 0 0 166 2 2 | h = 7

0 14 1 1 1 1 0 1 145 10 | i = 8

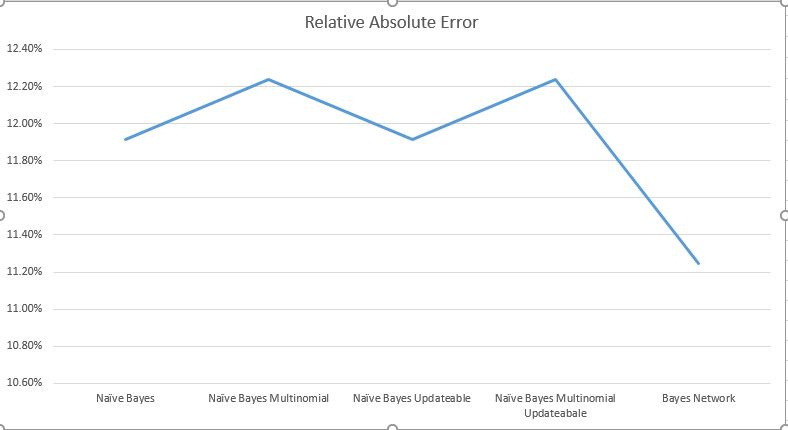
0 1 0 5 6 3 0 2 4 159 | j = 9

* The following graphs show us the accuracy and relative absolute error for all the algorithms used.

**Accuracy:**



**Relative Absolute Error:**

****

* We can say that almost all the algorithms work similarly but Bayes network stands out with the highest accuracy and least relative absolute error rates.

**Amazon Reviews Dataset:**

* Status – Completed.
* The challenging part dealing with Amazon Reviews Dataset is to understand how to classify text data.
* Implemented Bag of Words algorithm to the reviews column of the dataset. Cleaned the data from stopwords (using NlTK’s corpus – stopwords), numbers and converted to a set of meaningful words.
* Count Vectorizer in scikit learn was used to create a bag of words model which will return binary value if that is word present in each review of the data set or not. The same will be iterative for all the words in the bag of words.
* Later I learned of something called TF-IDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval, text mining, and user modelling.
* Instead of using count vectorizer I used TF-IDF vectorizer in scikit-learn to give a vector for each review with which I performed decision tree classification.
* I used Naive Bayes classifier for multinomial models - MultinomialNB by scikit-learn.
* The multinomial Naive Bayes classifier is suitable for classification with discrete features (e.g., word counts for text classification). The multinomial distribution normally requires integer feature counts. However, in practice, fractional counts such as tf-idf may also work.
* The Gaussian Naïve Bayes doesn’t work well with the text-vectorised data and hence showed us really less accuracy on cross validation.
* The cross validation scores for the Multinomial NB on the train dataset were:

[ 0.58573217

0.58720201

0.58720201

0.58720201

0.58720201

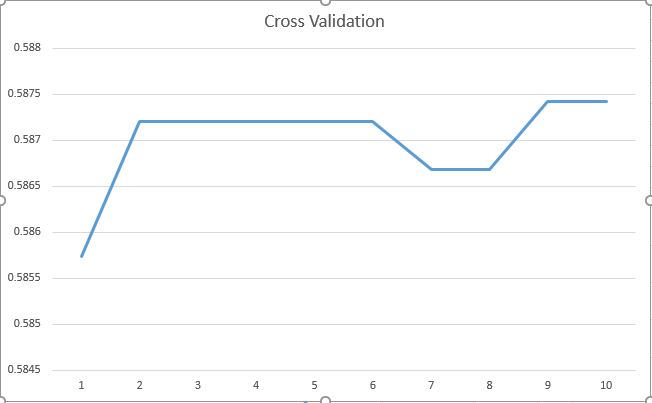
0.58720201

0.58668342

0.58668342

0.58742138

0.58742138]



* Accuracy and Confusion matrix for a smaller dataset where train contained 8000 instances and test contained 2000 instances:

0.601005025126

[[ 3 0 0 0 160]

[ 0 0 0 2 118]

[ 0 0 0 2 172]

[ 0 0 0 4 340]

[ 0 0 0 0 1189]]

* Accuracy and Confusion matrix for the entire train and test dataset:

0.599692788765

[[ 570 7 4 51 2387]

[ 130 6 7 82 2028]

[ 60 8 10 165 3153]

[ 20 6 1 270 6369]

[ 31 0 0 86 21006]]

* We have a class imbalance issue when we take a smaller dataset, since it contains more number of class values as 5’s, the bias in the naïve bayes algorithm makes all the predictions in favour of most likely class.
* One way to deal with this was to take more amount of data, I have taken the entire dataset and we can see an improvement in the class values being predicted among 1,2,3 and 4 as well.
* Even though Naïve bayes works pretty well with OPT dataset, it has the problem of class imbalance when we take the amazon reviews dataset.