CS-584 MACHINE LEARNING

ASSIGNMENT – 2

GENERATIVE LEARNING

By

JAIMIN SANGHVI(A20344798)

PROBLEM STATEMENT

- A generative model is used for randomly generating observable data values.
- A generative model is used in machine learning for either modeling data directly or as an intermediate step to forming a conditional probability density function.
- A conditional distribution can be formed from generative mode through Bayes' rule
- The main objective of these assignment is to classify different dataset using generative learning methods.
- In the given assignment2, I have implemented techniques for generative learning. I have used three different continuous and discrete datasets to analyze these two models in different way.
- I have used three different sets from UCI Machine Learning Repositories named as data_banknote_authentication.txt, iris.data, spambase.data
- I have implemented and evaluated algorithm for two models named as Gaussian discriminant analysis and naïve Bayes analysis. In addition, I have implemented k-fold cross validation to evaluate performance and derive more accurate results for these models.

PROPOSED SOLUTION

- I have implemented generative learning algorithms from the scratch using core logic of mathematics and python.
- I have implemented five separate program files for 1D-2class GDA, nD-2class GDA, nD-kclass GDA, Naïve Bayes Bernoulli and Naïve Bayes Binomial models.
- I have implemented generalize model for Gaussian Discriminant Analysis in the following steps:
  - ✓ Load dataset and store it in data and target matrix
  - ✓ Distinguish training and testing data according to k-fold
  - ✓ Calculate mean value for each class
  - ✓ Calculate sigma(variance) value for each class
  - ✓ Calculate membership function for each class and store it in dictionary
  - ✓ Perform discriminant function on data and calculate predicted Y
  - ✓ Find confusion matrix and evaluate accuracy, precision, recall and f-measure error
  - ✓ Print the maximum accuracy for a given data set
- I have implemented Naïve Bayes Bernoulli model and Naïve Bayes Binomial model in the following steps:
  - ✓ Load dataset and store it in data and target matrix
  - ✓ Distinguish training and testing data according to k-fold
  - ✓ Distinguish training and testing data according to class
  - ✓ Find the prior values and alpha for each class

- ✓ Find membership function for each class
- ✓ Perform discriminant function on data and calculate predicted Y
- ✓ Find confusion matrix and evaluate accuracy, precision, recall and f-measure error
- ✓ Print the maximum accuracy for a given data set

IMPLEMENTATION DETAILS

INSTRUCTIONS

- I have used online datasets for model evaluations. Hence it must need internet connectivity.
- The given implementation is data-oriented. User may require necessary changes to run same model for different datasets.
- I have implemented model for K-folds and print the single fold output with maximum accuracy at the end of code. For nD 2-Class GDA, I have generated precision recall curve graph which will be generated automatically.

DESIGN ISSUES

- The spam-base dataset contains number of features. Hence, The Binomial Naïve Bayes model takes too much time to find the optimized value of membership function.
- The computation of nCr was a typical issue during the implementation. However, I have resolve it by using core mathematic functions.

INSTRUCTION TO RUN

- I have implemented given problems solution in jupyter notebook.
- Instruction to run given project files:
  - ✓ Load *.ipynb file in jupyter notebook (*- 1D-2class_GDA, nD-2class_GDA, nD-kclass_GDA, Naïve_Bayes_Bernoulli, Naïve_Bayes_Binomial)
  - ✓ **Run 1D-2class_GDA.ipynb** file for Gaussian discriminant analysis. It will run program for 1 dimensional 2 class model. I have used bank note dataset to evaluate this model. Note: For GDA analysis, I have fetched dataset from URL. Hence it must need internet connectivity.
  - ✓ **Run nD-2class_GDA.ipynb** file for Gaussian discriminant analysis. It will run program for n dimensional 2 class model. I have used bank note dataset to evaluate this model. Note: For GDA analysis, I have fetched dataset from URL. Hence it must need internet connectivity.
  - ✓ **Run nD-kclass_GDA.ipynb** file for Gaussian discriminant analysis. It will run program for n dimensional k class model. I have used iris dataset to evaluate this model. Note: For GDA analysis, I have fetched dataset from URL. Hence it must need internet connectivity.

- ✓ **Run Naïve_Bayes_Bernoulli.ipynb** file for Naïve Bayes analysis. It will run program for Bernoulli model. I have used spam dataset to evaluate this model. Note: For GDA analysis, I have fetched dataset from URL. Hence it must need internet connectivity.
- ✓ **Run Naïve_Bayes_Binomial.ipynb** file for Naïve Bayes analysis. It will run program for Binomial model. I have used spam dataset to evaluate this model. Note: For GDA analysis, I have fetched dataset from URL. Hence it must need internet connectivity.

RESULTS AND DISCUSSIONS

❖ 1D, 2-Class Gaussian Discriminant Analysis

Dataset: "archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt"

Confusion matrix: [[61 9]
[5 62]]

Accuracy: 0.8978 (89%)

Errors:

Error/Class	**Class 0**	**Class 1**
Precision	0.8714	0.9253
Recall	0.9242	0.8732
F-Measure	0.8970	0.8985

Conclusion: Above results conclude that the implemented model is provide the 90% accurate results.

❖ nD, 2-Class Gaussian Discriminant Analysis

Dataset: "archive.ics.uci.edu/ml/machine-learning-databases/00267/data_banknote_authentication.txt"

Confusion matrix: [[64 15]
[27 31]]

Accuracy: 0.6954 (70%)

Errors:

Error/Class	**Class 0**	**Class 1**
Precision	0.8101	0.5345
Recall	0.7032	0.6739
F-Measure	0.7529	0.5962

According to above results, I can conclude that the model accuracy for nD 2-class model is lower than the model accuracy of 1D 2-class model.

Precision-recall error graph for nD, 2-class model is as below:

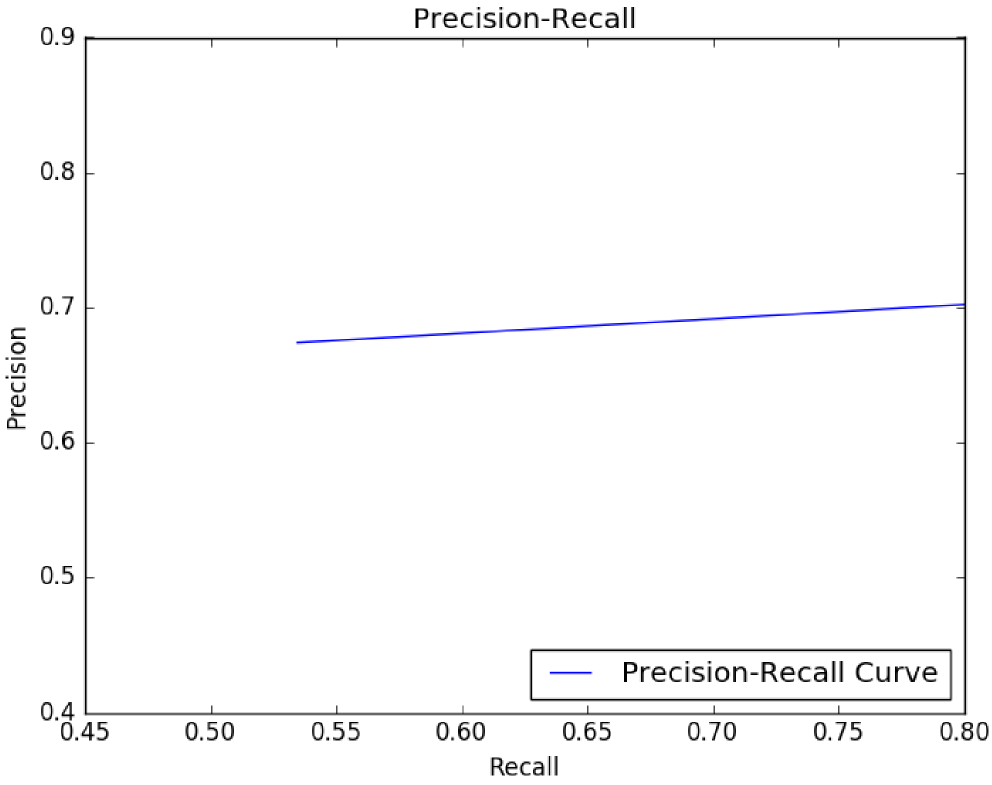

Figure. Precision-Recall Curve

❖ nD, k-Class Gaussian Discriminant Analysis

Dataset: "mlr.cs.umass.edu/ml/machine-learning-databases/iris/iris.data"

Confusion matrix:

```
[[6 0 0]
 [0 4 0]
 [0 1 4]]
```

Accuracy: 0.9333 (93%)

Errors:

Error/Class	**Class 0**	**Class 1**	**Class 2**
Precision	1.0	1.0	0.75
Recall	1.0	0.875	1.0
F-Measure	1.0	0.9333	0.8571

For above results, I conclude that the accuracy of nD, k-Class GDA is exactly match with the accuracy that I have achieved using inbuilt functions.

❖ Naïve Bayes Bernoulli Model

- In Bernoulli NB, I have implemented naïve Bayes training and classification algorithms for the data which is distributed according to multivariate Bernoulli distributions.
- It consists multiple feature but I have assumed each one as binary valued variable. Hence, this class requires samples to be represented as binary-valued feature vector.

Dataset: "archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data"

Confusion matrix: [[443 17]
[0 0]]

Accuracy: 0.9630 (96%)
Errors:

Error/Class	**Class 0**	**Class 1**
Precision	0.9630	0.0
Recall	1.0	0.0
F-Measure	0.9812	0.0

❖ Naïve Bayes Binomial Model

- Bernoulli NB is another classic naïve Bayes algorithm which is used for text classification.
- I have implemented naïve Bayes training and classification algorithms for the data which is distributed according to multivariate distributions.
- In Bernoulli NB classification, I have considered actual count of words for prediction. The given dataset consists the word frequency instead of actual count, therefore I have assumed document length as 100.

Dataset: "archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.data"

Confusion matrix:

```
[[0   0]
 [17 444]]
```

Accuracy: 0.9631 (96%)

Errors:

Error/Class	Class 0	Class 1
Precision	0.0	0.9631
Recall	0.0	0.1
F-Measure	0.0	0.9812

As per given in assignment2, I have written likelihood function, compute derivative, equate the derivative to zero and solve the required parameters.

I have attached e-copy (named as JAIMIN_A20344798_ASS2.pdf) of solution 5(a) in directory named as report.

REFERENCES

[1] http://www.astro.ufl.edu/~warner/prog/python.html

[2] http://scikit-learn.org/stable/modules/naive_bayes.html

[3] https://archive.ics.uci.edu/ml/index.html

[4] http://nlp.stanford.edu/IR-book/html/htmledition/the-bernoulli-model-1.html

[5] https://en.wikipedia.org/wiki/Naive_Bayes_classifier