CS 613 - Machine Learning

Assignment 3 - Classification

Introduction

In this assignment you will perform classification using Naive Bayes and Decision Tree classifiers. You will run your implementations on both a binary class dataset and a multi-class one.

You may **not** use any functions from a ML library in your code. And as always your code should work on any dataset that has the same general form as the provided one.

Grading

Part 1 (Theory)	25pts
Part 2 (Naive Bayes)	25pts
Part 3 (Decision Trees)	25pts
Part 4 (Multi-Class)	25pts
TOTAL	100

Datasets

Spambase Dataset (spambase.data) This dataset consists of 4601 instances of data, each with 57 features and a class label designating if the sample is spam or not. The features are *real valued* and are described in much detail here:

https://archive.ics.uci.edu/ml/machine-learning-databases/spambase/spambase.names

Data obtained from: https://archive.ics.uci.edu/ml/datasets/Spambase

Cartiotocgraphy Dataset (CTG.csv) Download the file CTG.csv from Bblearn. This file contains 2126 instances of 21 feature pertaining to information obtained from Cardiotocography tests. Our task is to determine the fetal state class code given an observation. This code can be one of the 3 values and pertains to the LAST column of the dataset. The second to last column of the dataset can also be used for classification but for our purposes DISCARD it.

Your scripts that use this dataset must be able to run on any dataset where the first two rows contain header information, the 2nd to last column is to be discarded, and the last column contains the target class.
You can read more about the dataset here:

http://archive.ics.uci.edu/ml/datasets/Cardiotocography

1 Theory

1. Consider the following set of training examples for an unknown target function: $(x_1, x_2) \rightarrow y$:

Y	x_1	x_2	Count
+	T	T	3
+	T	F	4
+	F	T	4
+	F	F	1
-	T	T	0
-	T	F	1
-	F	T	3
-	F	F	5

   (a) What is the sample entropy for the class label overall, $H(Y)$ from this training data (using log base 2) (3pts)?

   (b) What are the weighed average entropies for branching on variables x_1 and x_2 (4pts)?

   (c) Draw the deicion tree that would be learned by the ID3 algorithm without pruning from this training data. You may use software to draw this or draw it by hand. But either way the figure should be embedded in your PDF submission. (5pts)

2. We decided that maybe we can use the number of characters and the average word length an essay to determine if the student should get an A in a class or not. Below are five samples of this data:

# of Chars	Average Word Length	Give an A
216	5.68	Yes
69	4.78	Yes
302	2.31	No
60	3.16	Yes
393	4.2	No

   (a) What are the class priors, $P(A = Yes), P(A = No)$? (3pts)

   (b) Find the parameters of the Gaussians necessary to do Gaussian Naive Bayes classification on this decision to give an A or not. Zscore the features first over all the data together so that there is no unfair bias towards the features of different scales (5pts).

   (c) Using your response from the prior question, determine if an essay with 242 characters and an average word length of 4.56 should get an A or not. Show the computations to support your answer. (5pts).

2 Naive Bayes Classifier

Let's train and test a *Naive Bayes Classifier* to classifiy Spam or Not from the Spambase Dataset.

First download the dataset *spambase.data* from Blackboard. As mentioned in the Datasets area, this dataset contains 4601 rows of data, each with 57 continuous valued features followed by a binary class label (0=not-spam, 1=spam). There is no header information in this file and the data is comma separated. As always, your code should work on any dataset that lacks header information and has several comma-separated continuous-valued features followed by a class id $\in 0, 1$.

Write a script that:

1. Reads in the data.
2. Shuffles the observations
3. Selects the first 2/3 (round up) of the data for training and the remaining for validation.
4. Zscores the features using the training data
5. Divides the training data into two groups: Spam samples, Non-Spam samples.
6. Creates Normal models for each feature for each class.
7. Classifies each validation sample using these models and chooses the class label based on which class probability is higher.
8. Computes the following statistics using the validation data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy

Implementation Details

1. Seed the random number generator for reproducability.
2. You may want to consider using the log-exponent trick to avoid underflow issues. Here's a link about it: https://stats.stackexchange.com/questions/105602/example-of-how-the-log-sum-exp-trick-works-in-naive-bayes
3. If you decide to work in log space, realize that there is a potential for $0log(0)$ which will likely cause an issue in your code. To deal with this just have your code identify this situation, and consider it to be a value of zero.
4. Although this is a binary-class dataset, you should write your code to generalize to multi-class (as you will be asked to train and evaluate a multi-class dataset in the last part of this assignment).
5. You also may want to consider removing features that have a very low standard deviation. These features don't add much information but can contribute (numerically) a lot to the computation of the likelihood. You can decide what you think is "very low".

In your report you will need:

1. The statistics requested for your Naive Bayes classifier run.

3 Decision Trees

Let's train and test a *Decision Tree* to classify Spam or Not from the Spambase Dataset.

Everyone taking this class should have implemented at least a **binary search tree** at some point, which can be the starting point for you decision tree impementation. If you're rusty on how to do this, here's a resource:

`https://www.tutorialspoint.com/python_data_structure/python_binary_tree.htm`

.

Write a script that:

1. Reads in the data.
2. Shuffles the observations.
3. Selects the first 2/3 (round up) of the data for training and the remaining for validation.
4. Zscores the features using the training data
5. Trains a decision tree using the ID3 algorithm without any pruning.
6. Classify each validation sample using your trained decision tree.
7. Computes the following statistics using the validation data results:
   (a) Precision
   (b) Recall
   (c) F-measure
   (d) Accuracy

Implementation Details

1. Seed the random number generator for reproducability.
2. Although this is a binary-class dataset, you should write your code to generalize to multi-class (as you will be asked to train and evaluate a multi-class dataset in the last part of this assignment).
3. Depending on your perspective, the features are either continuous or finite discretize. The latter can be considered true since the real-values are just the number of times a feature is observed in an email, normalized by some other count. That being said, for a decision tree we normally use categorical or discretized features. **So for the purpose of this dataset, look at the range of each feature and turn them into binary features by choosing a threshold. I suggest using the median or mean.**

In your report you will need:

1. The statistics requested for your Decision Tree classifier run.

4 Additional Evaluation

Now let's evaluate your two classifiers (Naive Bayes and Decision Trees) on a multi-class dataset.

Download the Cardiotography set provided on Blackboard. Read about how to use it in the Datasets section.

Next train and evaluate (with your validation data) both a Naive Bayes classifier and an ID3 Decision Tree classifier on this dataset. For evaluation we'll use *accuracy*. In order to consider things a "fair" comparison, using the same training set (and therefore same validation set) for both classifiers.

In your report you will need:

1. The accuracy of your Naive Bayes classifier and ID3 Decision Tree classifier.

Submission

For your submission, upload to Blackboard a single zip file containing:

1. PDF Writeup
2. Source Code
3. readme.txt file

The readme.txt file should contain information on how to run your code to reproduce results for each part of the assignment.

The PDF document should contain the following:

1. Part 1:
   (a) Answers to Theory Questions
2. Part 2:
   (a) Requested Classification Statistics
3. Part 3:
   (a) Requested Classification Statistics
4. Part 4:
   (a) Requested Classification Statistics