Problem assignment 5

Due: Thursday, February 22, 2018

In this problem we shall investigate the ”Pima” dataset and learn classification models for it. Recall we performed some exploratory analysis of the Pima dataset in Problem set 1. You can download the dataset (pima.txt) and its description (pima desc.txt) from the course web page. In addition to the complete dataset pima.txt, you have pima train.txt and pima test.txt you will need to use for training and testing purposes. The dataset has been obtained from the UC Irvine machine learning repository: <http://www1.ics.uci.edu/_mlearn/MLRepository.html>.

**Problem 1. Logistic regression model**

First we try the logistic regression model in combination with gradient methods. Give solutions to the following tasks:

(a) Use functions from assignment 4 to normalize the inputs in the pima dataset based on the data in the training set. Generate two new files *pima­\_train\_norm.txt* and *pima\_test\_norm.txt*.

*Part1.m*

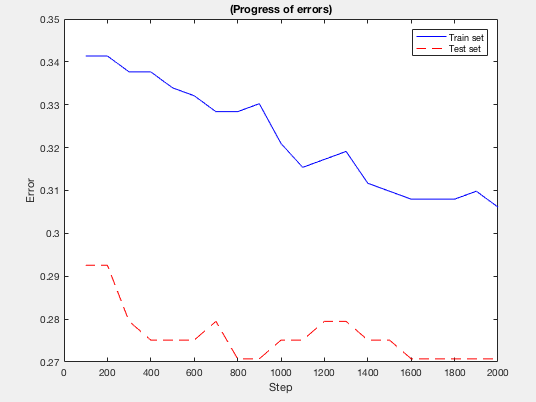
(b) Familiarize yourself with a batch-mode gradient procedure in file *Log\_regression.m*, in which all data points are considered at the same time.

(c) Implement and submit a program ***main1.m*** that runs the gradient procedure on the training dataset for 2000 iteration steps (also called epochs). Initialize all weights to 1 at the beginning. Use 2/sqrt(i) learning rate schedule.

(d) Include graph functions for monitoring the progress of misclassification errors (for both training and testing data) in ***main1.m*** as used in the previous problem set (HW-5).

***Main1.m***

*Progress of the error plot:*



In the report include final:

– Training and test misclassification errors

*The final misclassification error for training/testing set is*

*Train\_error = 0.3061*

*Test\_error = 0.2707*

– Confusion matrices for the train and test sets

*Test:*

|  |  |  |
| --- | --- | --- |
|  | *Target-1* | *Target-0* |
| *Predict-1* | *46* | *22* |
| *Predict-0* | *40* | *121* |

*Train:*

|  |  |  |
| --- | --- | --- |
|  | *Target-1* | *Target-0* |
| *Predict-1* | *111* | *89* |
| *Predict-0* | *76* | *264* |

– Sensitivity and specificity of the model on the test set.

*test\_sensit = 0.5349*

*test\_specif = 0.8462*

(e) Experiment with the learning algorithm by changing initial weights, learning schedule, number of epochs. Report training and test misclassification errors. What was the best result you could get?

*Initial weight as ones:*

*iteration steps = 2000, alpha = 0.5*

*misclass\_train = 0.2913*

*misclass\_test = 0.3013*

*Increase the iteration steps = 5000, alpha = 2/sqrt(n)*

*misclass\_train = 0.2913*

*misclass\_test = 0.2576*

*initial weight as zeros:*

*iteration steps = 2000, alpha = 2/sqrt(n)*

*misclass\_train = 0.2542*

*misclass\_test = 0.2358*

*this test has the best misclassification error on both train and test*

**Problem 2. Naive Bayes model**

The Naive Bayes model defines a generative classifier model in which all features are independent given the class label. In such a case the class-conditional densities over many input variables can be decomposed into a set of independent class-conditional densities, one for every input variable.

One important concern is the choice of an appropriate parameterization of class-conditional densities. Typically we do not choose the distributions arbitrarily, instead we want to make a good educated guess. Exploratory data analysis can help us greatly to recognize types of densities that appear to match the data the best.

**Problem 2.1. Exploratory data analysis**

We have performed the exploratory analysis of the Pima dataset in Problem set 1. Here we reuse the programs created there and apply them to study the density models we choose to parameterize our Naive Bayes model.

**Part a. Write and submit a program (main2\_1.m) that:**

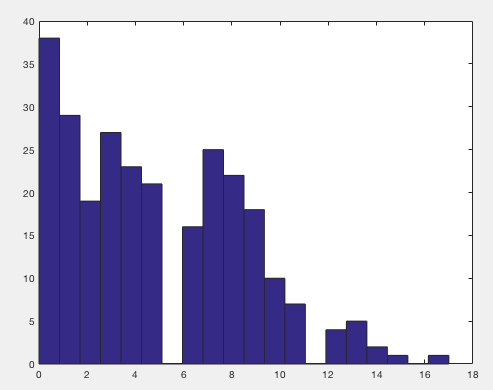
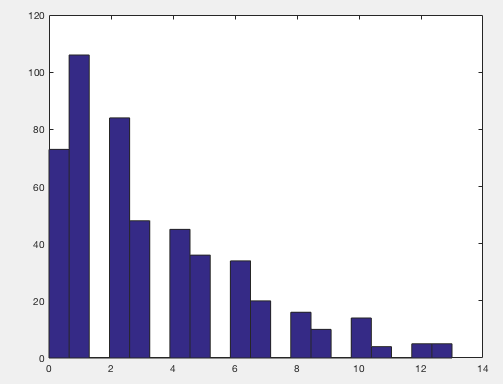
1. Divides ”pima.txt” data into two subsets - one with all examples with class ”0”, and another with all examples with class ”1”.
2. Analyzes examples in two subsets using histograms. Histograms should give you more information about the shape of the distribution of attributes. You can use the function ***histogram\_analysis.m*** for this purpose.

**Part b.** What distribution/density would you use to fit the values of attributes 1 to 8 in the pima dataset? Choices one typically considers are Bernoulli, Binomial, Multinomial, Normal, Poisson, Gamma, exponential distributions.

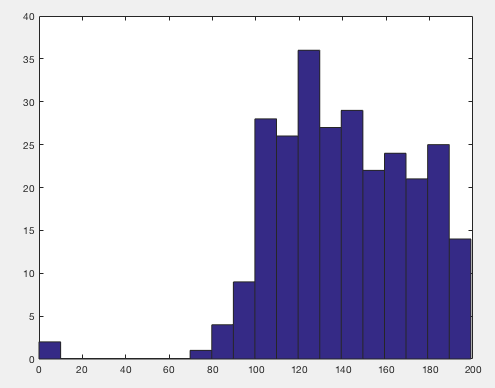
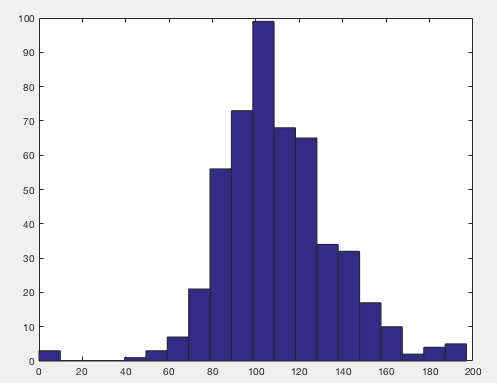
*left graph: label=0; right graph: label=1*

*Attribute 1:*

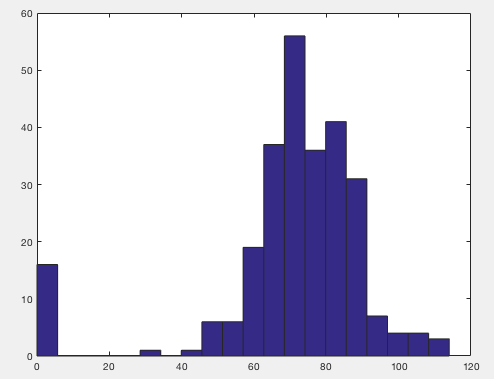
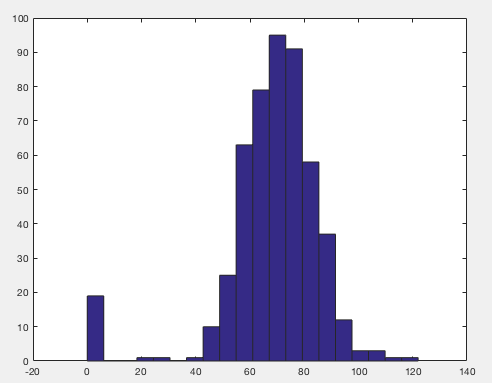
*Integers, shape best modeled by* ***exponential distribution***



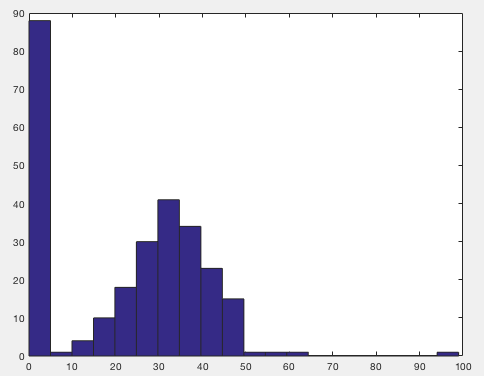
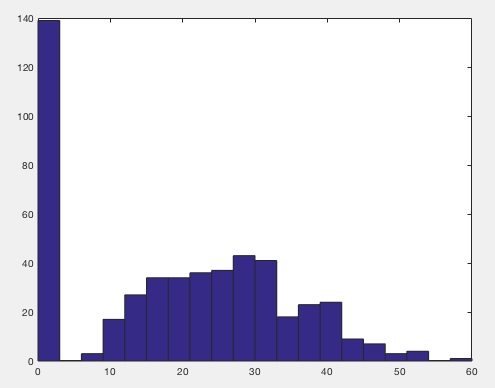
*Attribute 2, integers, shapes best modeled by* ***normal distribution,*** *especially for the class0 data*



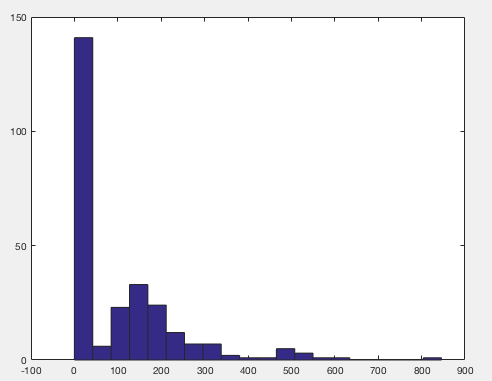
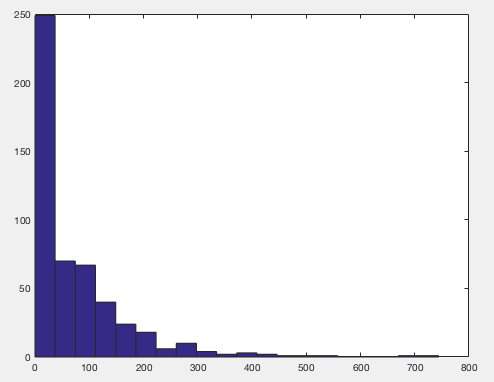
*Attribute 3: seem to be modeled by* ***normal distribution****, but data also has spikes at 0*



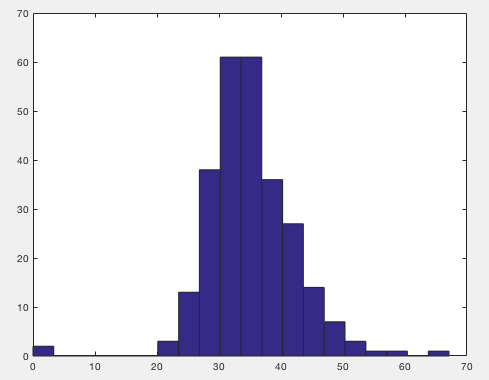
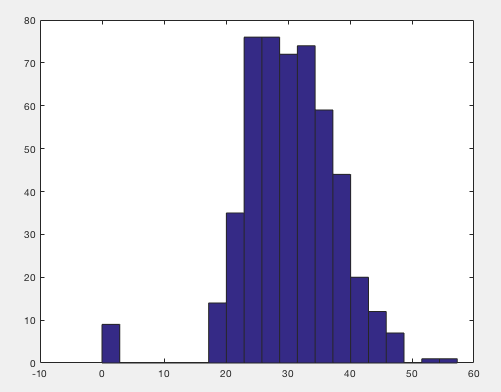
*Attribute 4: not as simple as normal distribution, there is huge spikes in 0*



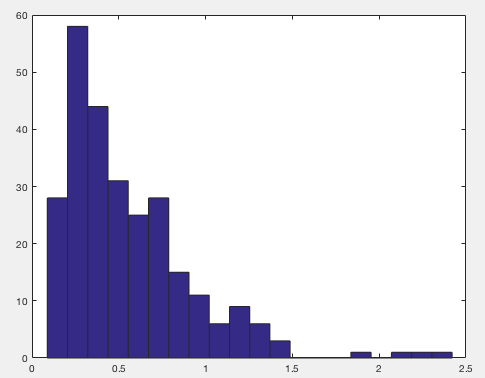
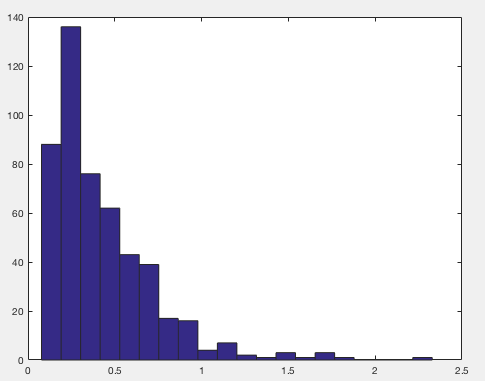
*Attribute 5: best modeled by* ***exponential distribution***



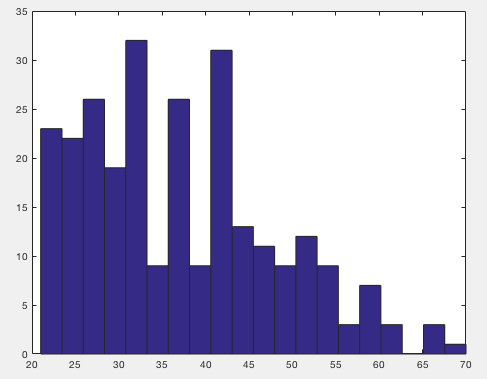
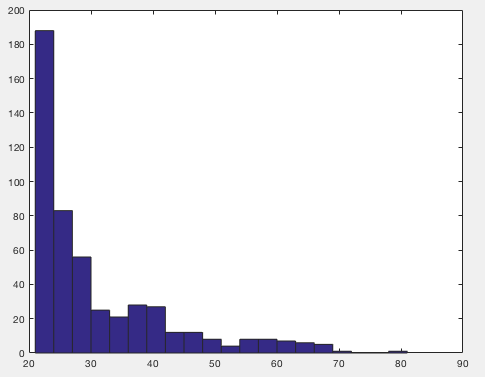
*Attribute 6: normal distribution*



*Attribute 7:* ***exponential distribution***



*Attribute 8:* ***exponential distribution***



**Problem 2.2. Learning of the Naive Bayes classifier**

Assume that class-conditional densities for pima dataset have the following form:

1. Class-conditionals for inputs [1 5 7 8] take the form of exponential distribution.
2. Class-conditionals for inputs [2 3 4 6] follow univariate normal distributions
3. In addition assume that priors on classes follow a Bernoulli distribution

**Part a.** Write and submit a program ***main2\_2.m*** that computes and returns the estimates of the parameters of the Naive Bayes model using the training set *pima­\_train.txt*.

The parameters include priors on classes, 16 class-conditionals (8 \* 2 = 16), one for every input

component and class label.

*main2\_2.m*

Part b. List parameters found by your program in the report.

*exp\_0\_1\_muhat = 3.2419; exp\_1\_1\_muhat = 4.7100;*

*exp\_0\_5\_muhat = 67.7168; exp\_1\_5\_muhat = 103.7200;*

*exp\_0\_7\_muhat = 0.4164; exp\_1\_7\_muhat = 0.5491;*

*exp\_0\_8\_muhat = 31.1032; exp\_1\_8\_muhat = 37.1200;*

*norm\_0\_2\_mu = 109.6254; norm\_0\_2\_sigma = 26.2304;*

*norm\_0\_3\_mu = 67.5339; norm\_0\_3\_sigma = 18.6683;*

*norm\_0\_4\_mu = 19.7316; norm\_0\_4\_sigma = 14.5828;*

*norm\_0\_6\_mu = 30.3059; norm\_0\_6\_sigma = 7.7258;*

*norm\_1\_2\_mu = 141.3950; norm\_1\_2\_sigma = 33.6655;*

*norm\_1\_3\_mu = 70.1900; norm\_1\_3\_sigma =21.6213;*

*norm\_1\_4\_mu = 22.9350; norm\_1\_4\_sigma =17.8275;*

*norm\_1\_6\_mu = 35.2580; norm\_1\_6\_sigma = 7.3286;*

*p\_y\_1 = 0.3711; p\_y\_0 = 0.6289;*

**Problem 2.3. Classification with the Naive Bayes model**

Once the parameters of the Naive Bayes model are learned (estimated) , the decision about the class for a specific input x can be made by designing the appropriate discriminant functions. Typically, there are based on class posteriors, thus a classification problems boils down to the problem of comparison of posteriors of classes for x.

Part a. Write and submit a program ***main2\_3.m*** that:

* Calls a function predict NB that predicts class labels for inputs based on class posterior. The discriminant functions you need to use here are given in expressions 1 and 2 and use parameters obtained in Problem 2.2.
* Uses predict NB to compute the misclassification error of the Naive Bayes classifier on both training and test datasets. Report the errors.
* Calculates and reports a confusion matrix for the test and training sets

Part b. In your report include:

* Training and test misclassification errors.
* Confusion matrices for the train and test sets.
* Sensitivity and specificity of the model on the test set.

***Main2\_3.m***

*misclass\_train = 0.2393*

*conf\_mat\_train =*

*121 79*

*50 289*

*misclass\_test = 0.4454*

*conf\_mat\_test =*

*21 47*

*55 106*

*test\_sensitivity = 0.2763*

*test\_specificity = 0.6928*

**Part c.** Compare results for the mean misclassification errors for the logistic regression model to the Naive Bayes classifier.

*The result from Naïve Bayes has higher testing error rate*

**Problem 3. ROC analysis**

The ROC analysis let us explore the ability of the classification model to discriminate in between the two classes including possible sensitivity and specificity trade-offs.

In the ROC analysis we assume a changing threshold for calling class 1 based on the projection defined by the model. This can be P(y = 1|x) for the logistic regression or the Naive Bayes.

Part a: Familiarize yourself with the function *perfcurve* in matlab that lets you calculate coordinates of points defining the ROC curve, as well as the area under the ROC curve (AUROC).

Part b. Use the function *perfcurve* to plot the ROC curve and calculate AUC on the testing set for the models you build in Problems 1 and 2. All models should be trained on the training set.

Part c. Please include the ROC curves and the AUC statistics in the report. Compare the ROC curves and their AUC statistics. What do you think, which model is better?

Logistic regression

lg\_AUC\_test = 0.7695

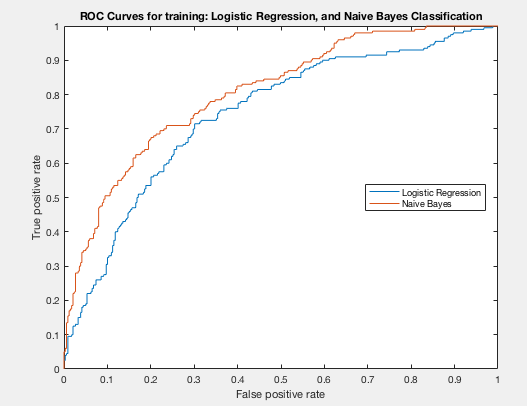
lg\_AUC\_train = 0.7430

Naïve Bayes

NB\_AUC\_test = 0.5134

NB\_AUC\_train = 0.8060

ROC curve for training dataset shows that Naïve bayes outperform the Logistic regression



ROC curve for testing dataset showing that Logistic regression outperform the Naïve Bayes on testing

