Module-Lab 6: Probabilistic Machine Learning

Exercise 1 This hands-on exercise explore simple supervised classification algorithms, namely the "Bayesian" classifier and the Naïve-Bayes Classifier (NBC). In this Lab session we shall use a dataset from UCI.

a) Go to the UCI repository (https://archive.ics.uci.edu/ml/datasets.php) and then search for the "**Ionosphere** Data Set". This is a binary (ie, 2-class: {"Good" , "Bad"}) classification dataset, containing 34 attributes (features) and 351 instances (or examples). Download the dataset ie, ionosphere.data and ionosphere.names.names.

Description of the Dataset (you can find details of the dataset in the file "ionosphere.names"): this radar data was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not ie, their signals pass through the ionosphere.

b) The ground-truth variable is the last value (row 35th) giving by a categorical (non-numeric) variable 'g' or 'b'. Using Matlab or Python, or both, separate the Dataset in a training set (the first 200 instances) and a test (the remaining instances ie, the number of test examples is 151). **Plot histogram graphs** for the training and test sets of the number of positives ("good") and negatives. For example (in Matlab),

```
figure(1);  histogram(LabelTraining,2)
figure(2);  histogram(LabelTest,2)
```

Hint1: this piece of code may be usefull

```
% % % Ionosphere dataset (in MATLAB)
D = importdata('ionosphere.data');
N = size(D,1); %351
X = zeros(N,35); %34 features plus the GT
for i=1:N
    n = length(D{i});
    X(i,1:34) = str2num(D{i}(1:n-1));
    if D{i}(n) == 'g' X(i,35) = 1; end
end
```

Hint2: we can load the Ionosphere directly in Matlab, by simply **>> load ionosphere**

c) Amongst the 34 features, there is one feature/attribute that is clearly **irrelevant**. Which one? Furthermore, there is another feature in the feature vector of class "g" that is redundant" (linearly-dependent) thus, remove features 1 and 2.

Note: the **rank** of a matrix X is the number of independent features/attributes of X. If the **determinant** of X is zero, there are linearly dependent features (which are **redundant**) and the matrix is *not full rank*.

d) Make a code to calculate the mean of the feature vector (now comprising 32 features) for the training and test sets. Note that we should calculate, for the **training set**, the mean-vector for "class0" and "class1" separately.

Hint:
(mean for class 'g'): 0.8816 0.0510 0.8456 0.1243 0.7816 …
(mean for class 'b'): 0.3187 -0.0254 0.2455 0.0040 0.2625 …

e) Now, make a code to calculate the simple variance (ie, assuming the pdf is uniform) of the features. Again, on the **training set**, the variances should be calculated per class.

Hint:
(variance for class 'g'): 0.0400 0.0607 0.0429 0.1062 0.0861 …
(variance for class 'b'): 0.4534 0.4429 0.4767 0.4091 0.3752 …

f) Assuming the features are uncorrelated, obtain the **diagonal** covariance matrix $\mathbf{\Sigma}=\sigma^2\mathbf{I}$. It should be a 32x32 matrix where the off-diagonal elements are zero.

Exercise 2

a) Implement a multivariate Normal Bayes classifier ie, consider the class conditional density is modelled by Gaussians.
Hint: see **Lec5** and/or page 6 of the **Lec6_BayesianInference.pdf** slides.

b) Performance measures on the **TEST set**: assuming identical priors for both classes (ie, priors = 0.5) and using the maximum a-posteriori decision, calculate the TPrate, FPrate, the Balanced Acc, F1-score, Precision, and the Recall on the **test set**.

c) Implement the Naïve Bayes classifier (NBC) and compare the results obtained in (b).
Hint: see page 17 of the **Lec5_Classifier.pdf** or page 12 of **Lec6** slides.

Hint: to avoid the underflow problem (due to the product of 33 likelihood terms tending to zero), it is preferable to use the sum of the logarithms. Thus, the NBC expression becomes

$$\hat{y} = \underset{k\in\{1,\ldots,|C|\}}{\operatorname{argmax}} \left(\log(p(C_k)) + \sum_{i=1}^{n} \log(p(x_i|C_k)) \right)$$

Some results using Matlab implementation:
>> ********** Classifier Normal Bayes
Acc = 0.70
BAcc = 0.79
Pre = 0.98
Rec = 0.65
F1 = 0.78

>> ********** NBC
Acc = 0.49
BAcc = 0.69
Pre = 1.00
Rec = 0.38
F1 = 0.55