Istanbul Technical University- Spring 2019-2020
BLG454E Learning from Data, Homework 1

Purpose: Bayes Classifier
Total worth: 100 points
Handed out: Monday, March 09, 2020.
Due: Sunday, March 22, 2020 23:59. (through Ninova!)
Instructor: Islem Rekik (irekik@itu.edu.tr)
Assistants: Emrullah Gazioğlu (egazioglu@itu.edu.tr), Doğay Kamar (kamard@itu.edu.tr)

Please contact egazioglu@itu.edu.tr if you have any questions with the "LfD HW1" subject on email (for easy filtering)

Video blinks related to this homework: watch Machine Learning Blink 3 while focusing on parts:

https://www.youtube.com/watch?v=WENZIdjcMzc&list=PLug43ldmRSo1LDlvQOPzgoJ6wKnfmzimQ&index=17

https://www.youtube.com/watch?v=UIQ_Fvqmi80&list=PLug43ldmRSo1LDlvQOPzgoJ6wKnfmzimQ&index=18

For easy reference, the mathematical notation summary table of Lecture 3 is also attached in the last page of this homework.

IMPORTANT: Cheating from each-other or copying from internet will be penalized.

We also would like to inform you that we already collected more than 25+ source code from internet and your work will be compared to them as well. So please do your own work.

DATASETS

You are given 2 mainly different datasets and each has train and test set separately.

You are also given an extra modified dataset for the job that you will find in **PART D**.

Summary:

Number of samples	**Dataset 1**	**Dataset 2**	**Dataset2 (modified)**
Training set	200	1400	1410
Test set	100	600	600
TOTAL	**300**	**2000**	**2010**

PART A (10 Points) (Apply on: Dataset1, Dataset2)

Examine the two training **datasets 1 and 2** (here we exclude Dataset2_modified):

A.1) Import your training set (but only the training set for now) to the **Pandas** data frame.

- You are not going to touch test set until you create your model!

A.2) For each class, calculate and visualize the covariance matrices of datasets 1 and 2 and interpret them (via comment outs) with your own words. (4 points)

A.3) For each dataset, plot two overlaid transparent histograms for each feature (red for class 0 and blue for class 1) and interpret (via comment outs in Jupyter Notebook) them as well. This will allow you to examine the feature distribution within each class and compare them across classes.

  - In total you should generate 4 plots (each plot with 2 overlaid histograms): 2 datasets × 2 features each. (4 points)

A.4) Plot each training dataset (feature 1 on x-axis and feature 2 on y-axis). Example plots can be seen in Figure 1. (2 points)

PART B (45 Points) (Apply on: Dataset1, Dataset2)

Experiment 1: Evaluation of Bayes classifier using random single dataset split into train and test sets

Assume that each class is generated from a multivariate Gaussian distribution.

B.1) Estimate the mean vectors for each class using the training data and consider the covariance matrices that you have calculated in Part A.: (just `print()` them)

def mean(args)
or use **numpy**'s built-in method.

c1 (class 1)	c2 (class 2)
Mean Vector ($\mu_1 = [?\ \ ?]$)	Mean Vector ($\mu_2 = [?\ \ ?]$)
Covariance Matrix $\Sigma_1 = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$	Covariance Matrix $\Sigma_2 = \begin{bmatrix} ? & ? \\ ? & ? \end{bmatrix}$

B.2) Compute the discriminant function *gi(x)* for each class regarding the following cases (40 points):

- Case 1 (10 points): Different means but equal covariance matrices ($\mu_1 \neq \mu_2$ and $\Sigma_1 = \Sigma_2$).
- Case 2 (10 points): For each class, the covariance matrix has constant variances across features within each class and satisfies:
$$\Sigma_1 = \sigma_1^2 I;\ \Sigma_2 = \sigma_2^2 I$$
- Case 3 (20 points): General case where the covariance matrix of each class is arbitrary.

You may use the following code block for this job:

```
def trainBayes(trainingSamples,trainingLabels)
    if means are different but covariance matrices are equal:
        gi(x)= …
    elif covariance matrix has constant variances across features within each class
        gi(x)= …
    else: #general case
        gi(x)= …
    return [gi(x)]
```

B.3) **For each dataset 1 and 2**, test your Bayes classifier (5 points):

Apply the model to the test set after you import it and save the results in a vector.

Note that, you should pretend like you don't know the labels (classes) of the test set while predicting.

Compare the predicted labels by Bayes classifier you have obtained with the actual (ground truth) labels of the test set and calculate the classification error rate as follows:

$$e = \frac{\#misClassification}{\#testSamples} x100$$

PART C (25 Points) (Apply on: Dataset1, Dataset2)

Experiment 2: Evaluation of Bayes classifier using 5-fold cross-validation

C.1) Combine the training set and the test set row-wise (concat) (1 Point)

C.2) Apply 5-fold cross validation on this new dataset: (20 Points)

a. Train your classifier and follow the same steps that given in Part B.2, B.3 and B.4
   i. Compute means and covariance matrices
   ii. Compute the $g_i(x)$
   iii. Find the classification error rate by applying your model to each left-out fold for testing
b. Save your result

C.3) Compute the avg. of your testing results (1 Point)

C.4) Compare and interpret the avg. result to the result that you have obtained in Part B (i.e., using the random single split of the dataset). Which evaluation strategy is most reliable? Discuss and justify. (3 Points)

Warning 1: There might be an issue with CV since it perturbs the covariance of the data as we shuffle the samples. For this reason, use the same random seed in Python: random.seed(1).

Warning 2: Note also that the covariance matrix of each class will randomly change. In case, the covariance matrix becomes invertible, you can use a pseudo-inverse. There are Python codes (strategies) which estimate a pseudo-inverse of a non-invertible matrix. In case you use such codes, include the references in your Jupyter notebook and explain in brief how you generated the matrix pseudo-inverse. This can be added as a test in your code (i.e., whether the covariance matrix is invertible or not).

PART D (20 Points) (Apply on: Dataset2_modified)

D.1) Plot training Dataset2_modified (feature 1 on x-axis and feature 2 on y-axis). What do you notice in comparison with Dataset2 plot? (5 points)

D.2) Apply the Experiment 1 above on the Dataset2_modified and compute the classification error on the test set of Dataset2_modified. (5 points)

D.3) Compare the results for Dataset2 by Experiment 1 (i.e., random single partition of the data) and those of Dataset2_modified. What can you conclude about Bayes classifier? (10 points)

EXAMPLE PLOTS

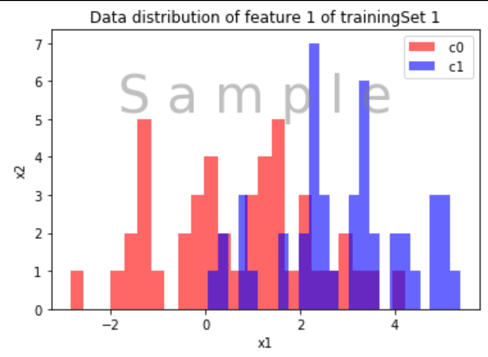
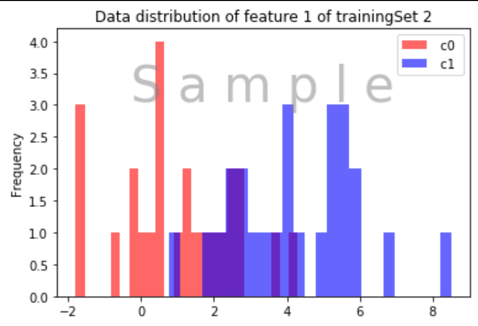
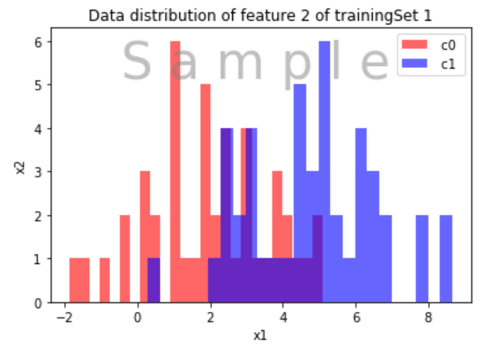
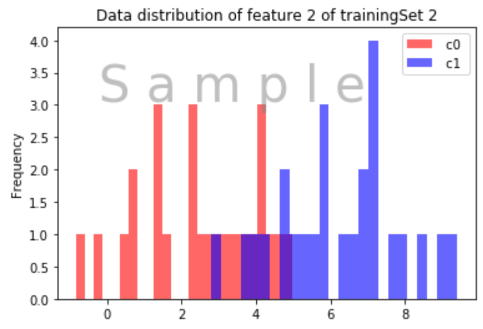
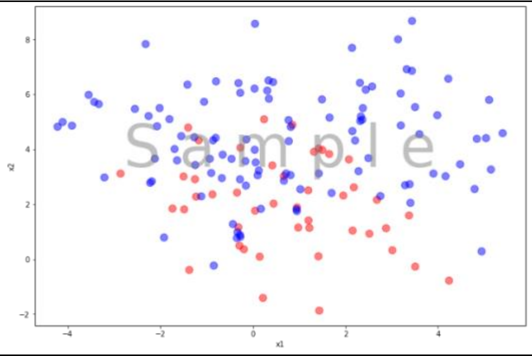
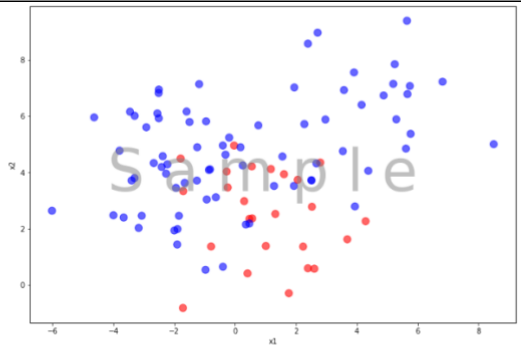
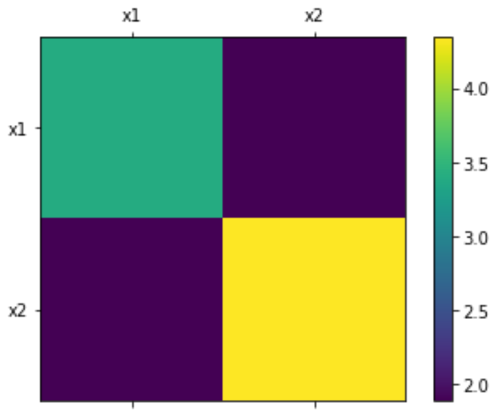
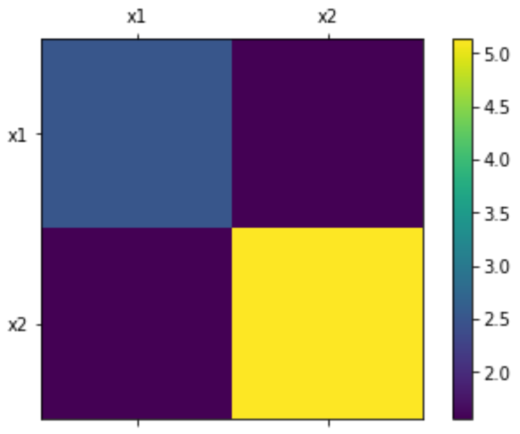
DATASET 1	DATASET 2
	
	
	
	

Figure 1

```
The Modules that You Can Use:
-> pandas, numpy, math, random, matplotlib, csv, etc…
The Modules that You CANNOT Use:
-> scipy, scikit-learn, Bio, Theano, Tensorflow, Keras, PyTorch and any other similar machine learning modules.
```

NOTE: <u>You are not allowed to use any Python built-in functions for Bayes classifier model building.</u>

SUBMISSION INSTRUCTIONS

- Only electronic submissions through Ninova will be accepted.
- Late submissions or those submitted otherwise than according to instructions will not be accepted.
- Submit a your Jupyter Notebook file: **yourStudentID.ipynb** file. Note that the discussions should be included in the same file.

INSTALLING AND USING JUPYTER NOTEBOOK

Windows:

On command prompt (cmd.exe with admin mode):

```
C:\**path**> python -m pip install jupyter
```

After changing your current folder to the folder which you want to work on it (see 'cd' command):

```
C:\**your_working_folder**> jupyter notebook
```

Then it will be launched on your default browser

Ubuntu / Linux / Unix / Mac:

On Terminal:

```
$ pip install notebook
```

And then launch with:

```
$ jupyter notebook
```

For more information: https://jupyter.org/install

Table 2: *Major mathematical notations used in lecture 3.*

Mathematical notation	Definition
$\mathcal{D}$	dataset
n	number of samples in a dataset $\mathcal{D}$
d	number of features
$\mathbf{x} \in \mathbb{R}^{d\times 1}$	feature vector or data point (sample)
$\mathbf{x}_{feature}^{sample}$	–
$\mathbf{x}^i \in \mathbb{R}^{d\times 1}$	i^{th} sample in the population
$\mathbf{x}_j^i \in \mathbb{R}$	j^{th} feature of i^{th} sample in the population
$\mathbf{\Sigma} \in \mathbb{R}^{d\times d}$	covariance matrix of data population $\{\mathbf{x}^i\}_{i=1}^n$
$\lvert\mathbf{A}\rvert \in \mathbb{R}$	determinant of matrix A
$p(x) = \frac{1}{\sqrt{2\pi\sigma}} exp(\frac{-1}{2}\frac{\lvert\lvert x-\mu\rvert\rvert_2^2}{\sigma^2})$	probability density function of a variable $x \in \mathbb{R}$
$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}\lvert\mathbf{\Sigma}\rvert^{1/2}} exp[\frac{-1}{2}(\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu)]$	probability density function of a multidimensional variable $\mathbf{x} \in \mathbb{R}^{d\times 1}$
$\mu \in \mathbb{R}^{d\times 1}$	sample mean $\mu = \frac{1}{n}\sum_{i=1}^n \mathbf{x}^i$
$\lvert\lvert\mathbf{x}-\mu\rvert\rvert_{\mathbf{\Sigma}^{-1}} = (\mathbf{x}-\mu)^T\mathbf{\Sigma}^{-1}(\mathbf{x}-\mu) \in \mathbb{R}$	Mahalanobis distance between $\mathbf{x}$ and μ
$\mathbf{I}_{d\times d} \in \mathbb{R}^{d\times d}$	identify matrix of size d ×d
$\lvert\lvert\mathbf{x}-\mu\rvert\rvert_{\mathbf{I}_{d\times d}} = (\mathbf{x}-\mu)^T(\mathbf{x}-\mu) \in \mathbb{R}$	Euclidean distance between $\mathbf{x}$ and μ
	also noted as L_2 norm $\lvert\lvert\cdot\rvert\rvert_2$
$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	discriminant Bayes function for class i when $\mathbf{\Sigma}_i = \sigma_i^2\mathbf{I}$
	general case: when $\sigma_i^2 \neq \sigma_j^2$ for classes i and j (i.e., different means $\mu_i \neq \mu_j$
	but constant variance for all data features in each class)
	special case: when $\sigma_i^2 = \sigma_j^2$ for classes i and j (i.e., different means $\mu_i \neq \mu_j$
	but constant variances across all classes)
	(i.e., lines connecting means of different classes are perpendicular to decision boundaries)
	if $ln(p(c_i)) = ln(p(c_j))$, $g_i(\mathbf{x}) = -\lvert\lvert\mathbf{x}-\mu_i\rvert\rvert_2^2$
$g_i(\mathbf{x}) = \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	discriminant Bayes function for class i when $\mathbf{\Sigma}_i = \mathbf{\Sigma}_j = \mathbf{\Sigma}$
	(i.e., constant data feature covariance $\mathbf{\Sigma}$ across classes)
	(i.e., lines connecting means of different classes are not perpendicular to decision boundaries)
	if $ln(p(c_i)) = ln(p(c_j))$, $g_i(\mathbf{x}) = -\frac{1}{2}\lvert\lvert\mathbf{x}-\mu_i\rvert\rvert_{\mathbf{\Sigma}^{-1}}^2$
$g_i(\mathbf{x}) = \mathbf{x}^T\mathbf{W}\mathbf{x} + \mathbf{w}_i^T\mathbf{x} + \mathbf{w}_{i0}$	quadratic discriminant function (decision boundaries are nonlinear)