EEE 498 ML with application to FPGAs

All homework is handed in online. Don’t hand in pictures of the code files, hand in the actual code files so I can see if they work. Hand in images of outputs of your code (displays, plots …) converted to pdf. If there are questions to be answered answer them in PowerPoint, Word, or on paper and convert to pdf (It doesn’t actually matter how you get to the pdf).

Organize your work so that the answer to each problem can be easily identified. You will not get credit if your submission is poorly organized, or if the grader cannot find the answers.

You can use any language python, c, C++, MATLAB,…, to do coding though Python is likely the easiest and easiest to translate from lectures.

Homework 5 Naïve Bayes (20 points each)

1) Write a Naïve Bayes algorithm to classify the iris database. Use all 150 observations and all 3 classes.

```
## mu is the mean, sig is the standard dev, x is the feature value
def PGauss(mu, sig, x):
    return np.exp(-np.power(x - mu, 2.) / (2 * np.power(sig, 2.) + 1e-300) )
```

find the distribution (mu and stdev) for each feature for each class

```
from sklearn import datasets
iris = datasets.load_iris()
X = iris.data[:,0:4]   ## from this only take features 0,1,2,3
y = iris.target
```

Using this distribution find the probability of each class for each observation (product of probability of each feature given that class and the probability of the class). This may be useful code

```
self.u[f,c] = X[np.where(t==c),f].mean()
self.s[f,c] = X[np.where(t==c),f].std()
```

The class that has the highest probability is the predicted class for that observation

2) Run it for the iris database and determine

You don’t need to do standard scaling, but split into training and test (or validation) as usual. 30% test should be fine.

- Print out the number of samples (observations) in the training(or test) and the number of features used
- the number of miss classifications,
- the accuracy in the typical way and also
- dump out the indices of cases that were incorrectly classified, you can use this code

  ```
  ## where were there errors
  err=np.where(y_train!=ypred)
  ```

```
print('errors at indices ', err, 'actual classificiton ',
y_train[err],' pred myNB ', ypred[err])
```

capture all output in screen grabs

3) Train and test using the GaussianNB Naïve Bayes algorithm in Sklearn. Compare results with (2) and comment on differences.

Hand in all code, your results, use pdfs for documents other than code, and be well organized so that the grader can find your work.

Hand in all output as a screen grab for each problem converted to pdf. It is a good practice to paste well organized and labeled images into PowerPoint, answer all questions in the PowerPoint file, and then convert the PowerPoint file to pdf. Hand in all your code files. If code files are missing you will not get credit. It is typically easier to grade individual files rather than a zipped file, but either is accepted.