**EECS/MSAI 349 Problem Set 4**

**1. Report how you generated the data set (including the code if applicable) and a table giving the 10-fold CV accuracy of each algorithm on your data. Include a 2-3 sentence explanation for why the classifiers perform as they do on your data set. Include your data set as q1.csv in your zip file**

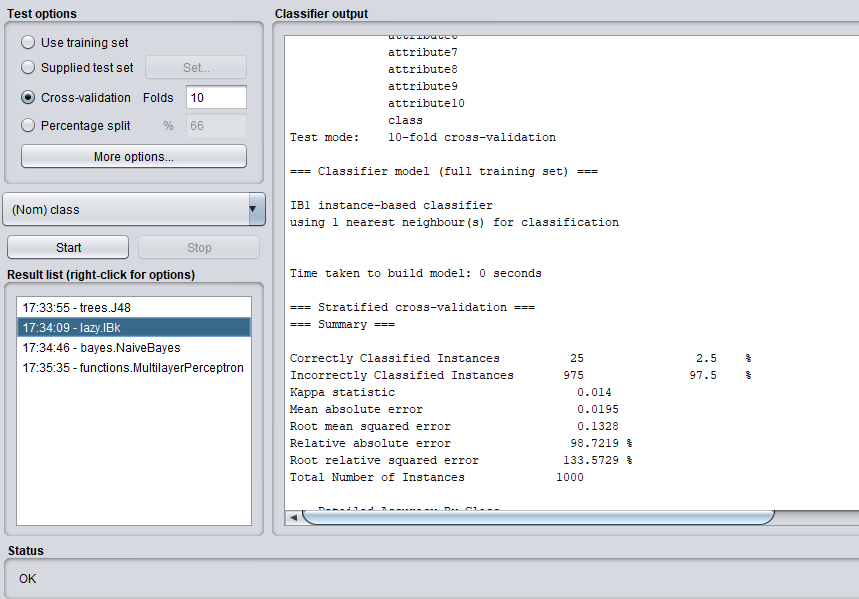
**Answer:-**

Please find the dataset in q1.csv file and the code to generate the same is in “q1csvgen.java” file.

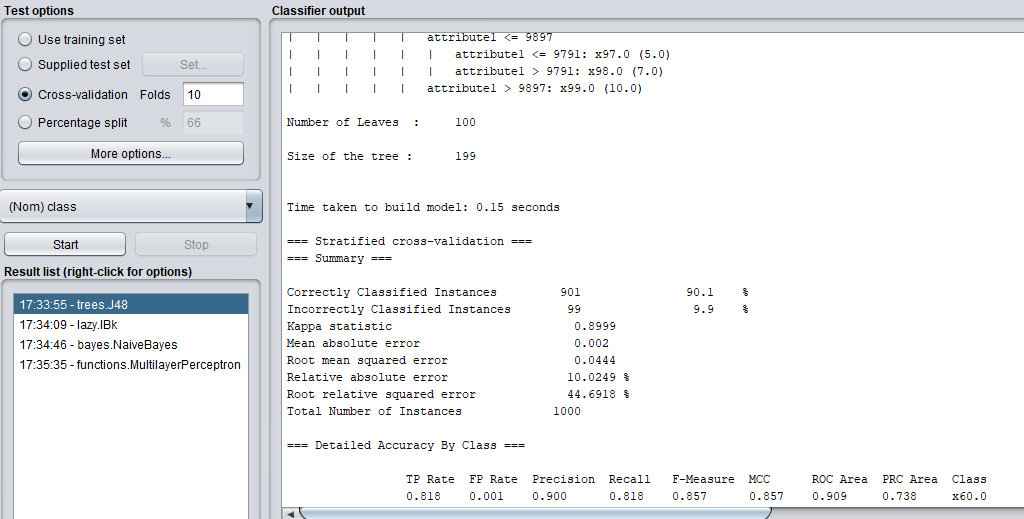
I have created a data set of 1000 with 10 attribute and one class attribute. The trick I used while creating my data is that I generated some random number between 1000 to 9999 for each of the attribute but my class attribute data was created by using only attribute1. Class attribute is some random string (x) appended by attribut1/100.

This is working well with decision tree and not with nearest neighbors because nearest neighbor calculate the distance of each and every attributes whether that attribute affect the class attribute or not. Decision tree is working well because it calculate entropy for each attributes which says that which attribute is important and which is not.

nearest neighbor (IBk) = 2.5%



decision trees (J48) = 90.1%



The absolute differences is |J48 accuracy – Ibk accuracy| = |90.1-2.5| = 87.6

**2. Repeat Question 1, except using multi-layer perceptrons (found under "functions" in Weka) vs. Naive Bayes classifiers (found under "bayes" in Weka). Submit your data set as q2.csv in your zip file. Yes, you can use the same data set in Question 1 as in Question 2, if you like.**

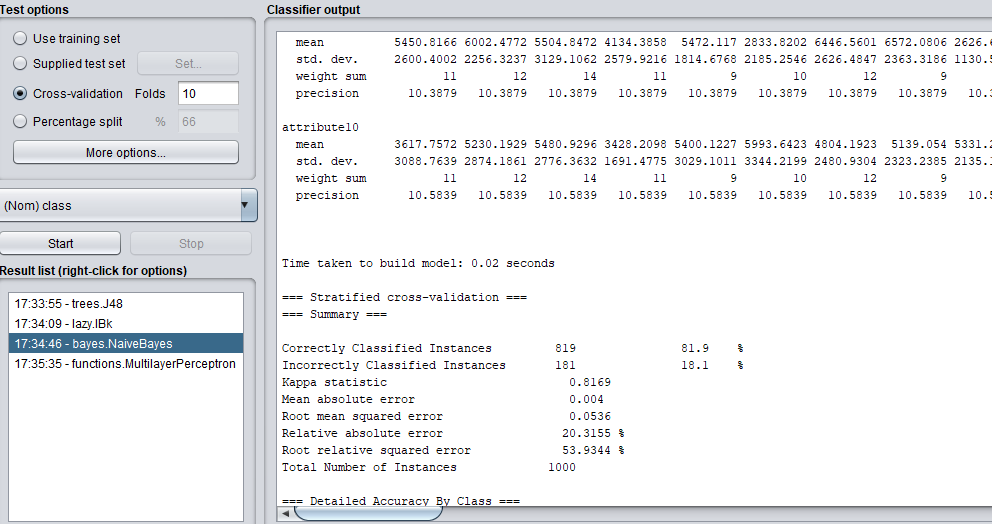
Answer:-

I used the same dataset (q1.csv) for the same and the results are as follows:

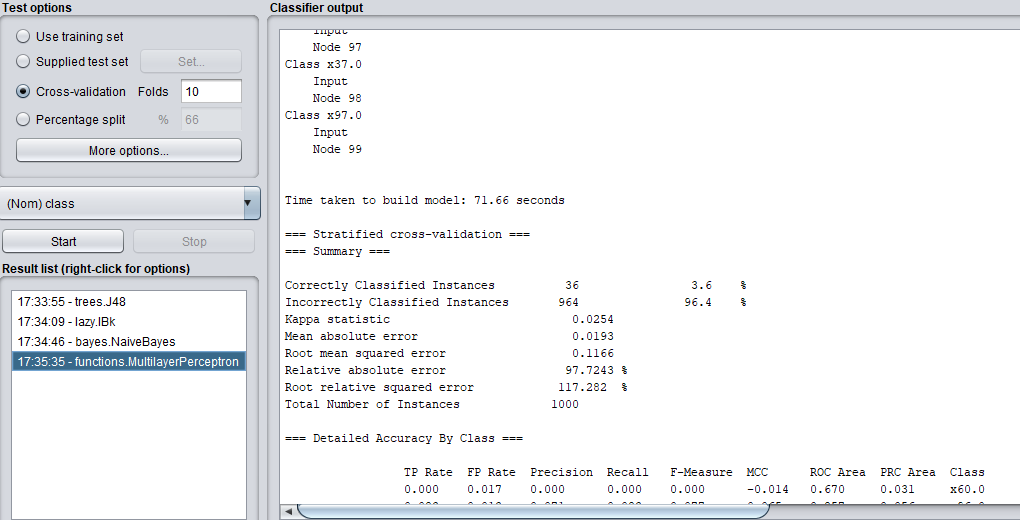
Naïve Bayes is working well in this case as all the attributes are independent of each other and naïve Bayes works well when all the attributes are independent to each other.

Multilayer preceptron is taking lots of time to run and not able to give good accuracy because all the data of each attributes are random numbers. There is no pattern in data and no attributes are dependent on any attributes.

Naïve Bayes = 81.9%



multi-layer perceptrons =3.6%

****

The absolute differences is |NaivesBayes – MultiLayer Perceptrons| = |81.9-3.6| = 78.3

**3. Define a data set with four binary attributes and a binary output, such that Naïve Bayes achieves only 2/3 training accuracy but Logistic Regression achieves 1.0 training accuracy. You should use add-one smoothing for Naïve Bayes (for both P(Class) and P(Attribute\_i | Class)), and no regularization for logistic regression. Justify your answer.**

**Answer:-**

Please find the dataset below (also in the assignment submission zip file with the name of q3.csv):

**a1,a2,a3,a4,class**

1,1,1,1,T

1,1,1,1,T

1,0,1,1,F

0,1,1,1,F

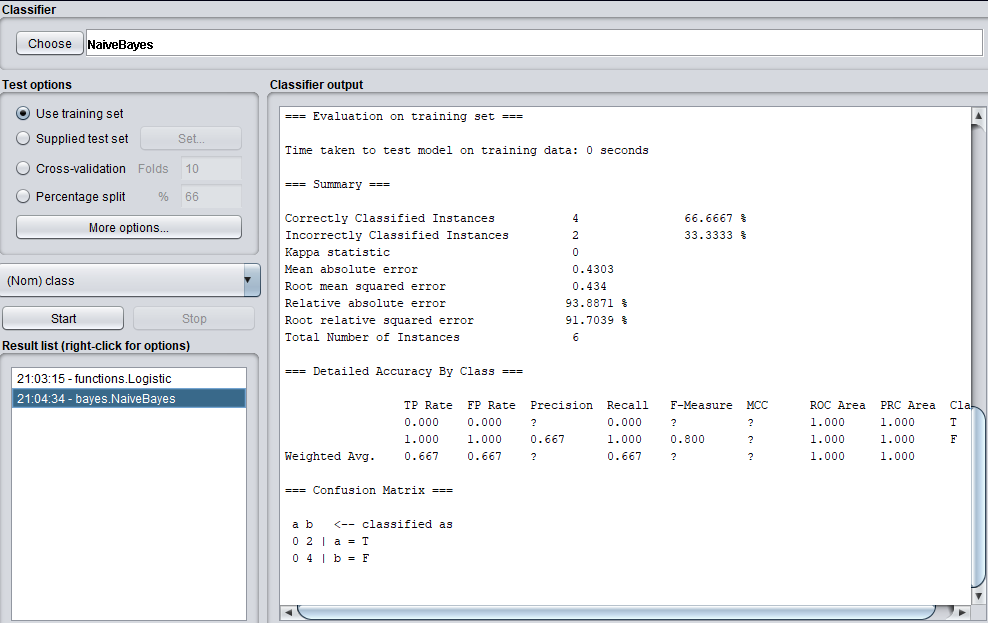
1,1,1,0,F

1,1,0,1,F

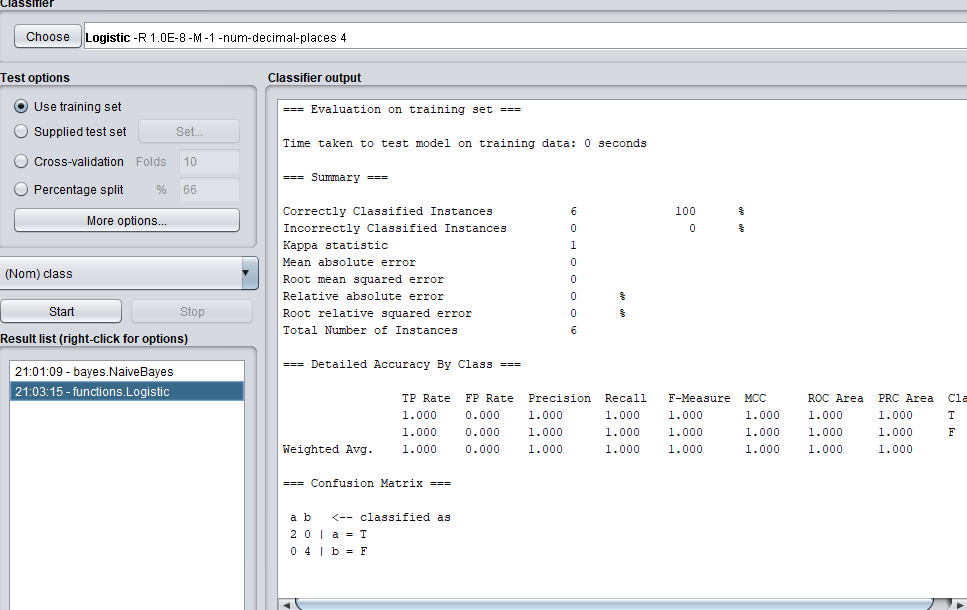
With above data set I was getting accuracy of 2/3 (66.67) for naïve Bayes and 1.0 for Logistic regression.

I used AND for finding class attribute values. Naïve Bayes is able to give only 2/3 accuracy because Naïve bayes runs on the assumption that attributes are independent on each other while in my dataset all attributes are dependent on each other. While the logistic regression fix this issue and works well even if all attributes are dependent on each other.

Naïve Bayes:



Logistic regression:



**4. Consider the following four random variables, which can take values in the given sets:**

**Hours of Sleep: {0, 1, 2, 3, 4, 5, 6, 7, 8}**

**Studied: {None, Some, Lots}**

**LikesMaterial: {None, Some, Lots}**

**ExamScore: {A, B, C}**

1. **How many independent parameters are needed to specify the joint distribution over these four variables? A joint distribution defined over k disjoint events requires k-1 independent parameters.**

Answer: The number of independent parameters needed to specify the joint distribution over these four variables = 9\*3\*3\*3 -1 =242

1. **Now assume you want to model the conditional distribution, P(ExamScore | HoursOfSleep, Studied, LikesMaterial). How many independent parameters does that conditional distribution contain?**

Answer: The number of independent parameters that the conditional distribution contains = 9\*3\*3\*(3-1) = 162

1. **Now imagine modeling P(ExamScore | HoursOfSleep, Studied, LikesMaterial) with the Naïve Bayes assumption, that each of the three variables on the right-hand-side of the pipe is conditionally independent given ExamScore. How many independent parameters are needed to specify the conditional distribution under the Naïve Bayes assumption?**

Answer: Independent parameter needed to specify the conditional distribution under the Naïve Bayes assumption = (3\*8) + (3\*2) + (3\*2) +2=38

**5. Implement a function named “cosine\_similarity” that computes the cosine similarity (a number in [-1.0, 1.0]) between two vectors (represented as numpy arrays or list of numbers). Turn in your code in a file ps4.py.**

Code is in ps4.py

**6. Report the similarity between each pair using VGG representation and pixel representation respectively (so you should report 6 numbers in total). Which pair is the most similar in VGG representation and which pair is the most similar in pixel representation?**

**Answer:**

Cosine similarity between mj1 and mj2 VGG: 0.9600110261784119

Cosine similarity between mj1 and cat VGG: 0.15253511266679187

Cosine similarity between mj2 and cat VGG: 0.14183421213603353

Cosine similarity between mj1 and mj2 pixel: 0.3708619464542405

Cosine similarity between mj1 and cat pixel: 0.4730885235789724

Cosine similarity between mj2 and cat pixel: 0.6197439905980221

The most similar pair in VGG is mj1 and mj2 while the most similar pair in pixel is mj2 and cat.

**7. Based on the result in #5 above, what problem does pixel representation have and why does the convolution neural network fix it?**

**Answer:**

The problems with pixel representation compared with Convolution neural networks are mentioned as follow:

* The pixel representation is only comparing two pixel of any image. Basically the RGB values of both the pixels, which contains just its color information and their distributions, that is why in above experiments the cosine similarity of mj2 (Michael Jordan) and mj1 with cat is higher compare to mj1 with mj2 image.
* In case of convolution neural networks, it uses many hidden layer to depict the overall image similarity. It is able to divide the image into many parts and each part describes some pattern with neighboring pixels. This is the reason why in above case mj1 and mj2 image shows the good amount of cosine similarity. It is able to fix the problem with pixel representation.

**8. Save your prediction result of the test set in two text files (one caption per line, in the same order that the image name appears in “test” field of “dataset.json”), vgg.txt and pixel.txt. Include the code you wrote to find the nearest neighbor images and captions in ps4.py. Compare the two results from VGG representation and pixel representation – is one better than the other? Justify your answer in 1-2 sentences**.

**Answer:**  Code is added in ps4.py

The files “vgg.txt”, “pixel.txt” and code “ps4.py”are included in zip file.

After checking each captions and their respective images, I found that VGG representation captions are better than the pixel representations. Although the vgg.txt has not 100 percent accurate caption as per the image but it is able to depict the image as much correct as possible.

This is because the captions were according to the train images and test images are not exactly similar to train images. The test image with the maximum cosine similarity with train image’s caption was chosen to label the caption of test image.

**9. A machine learning model is usually not perfect, and analyzing its errors can usually tell us more about how the model works. From the captions that doesn’t describe the image well, pick one example that uses VGG representation and one that uses pixel representation. Include the image and the generated caption, and describe why you think it generates that wrong caption.**

**Answer:**

VGG:

Caption: Two sheep in a field with long grass.

Image: 

I feel this image’s caption is not correctly classified as per the image because there are no two sheep and long grass in the image. Although the VGG did not predicted the image completely correct it tried to predict it partially correct like there is some greenery (long grass) in the image and there are 2 animals (although not 2 sheep) in the picture. The main reason behind this is that, none of the training image are exact similar to the test images. VGG tried to give the caption of trained images on the basis of the maximum cosine similarity of train image.

Pixel:

Caption: A person riding a snowboard down a snowy slope.

Image:

****

The captions and the images differ a lot in this case as you can see that there is no person, no snow board and there is no snowy slope. It generates wrong caption because, the cosine similarity of this image came maximum with one of the image of train image which actually has a person, a snowboard and he is riding the snowboard down the snowy slope.

Cosine similarity came maximum in this case with that train image (A person riding a snowboard down a snowy slope) because the pixel in the train image and in the above test image has similar RGB values (may because both images is almost white) and pixel representation only uses pixel by pixel values for both the images.