Course : EECS 349. Machine Learning

Name : Sangrin Lee

Student ID(netID) : 2999428(slk2940)

* Homework 4.

(student I worked with : Pradyoth Hegde)

Part1.

Q1. 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.data in your zip file.

- I generated the dataset by using Weka's RDG1 generator with setting the options – numAttributes with 200, seed with 2, and numExamples with 1000.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         967               96.7    %
Incorrectly Classified Instances        33                3.3    %
Kappa statistic                          0.9236
Mean absolute error                      0.036
Root mean squared error                  0.176
Relative absolute error                  8.2957 %
Root relative squared error             37.7983 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.979    0.060    0.972      0.979    0.976      0.924    0.970     0.976     c0
               0.940    0.021    0.955      0.940    0.948      0.924    0.970     0.937     c1
Weighted Avg.  0.967    0.047    0.967      0.967    0.967      0.924    0.970     0.964

=== Confusion Matrix ===

   a   b   <-- classified as
 668  14 |   a = c0
  19 299 |   b = c1
```

[Decision tree (J48)]

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         656               65.6    %
Incorrectly Classified Instances       344               34.4    %
Kappa statistic                          0.1765
Mean absolute error                      0.3582
Root mean squared error                  0.5806
Relative absolute error                 82.5639 %
Root relative squared error            124.6778 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               0.780    0.610    0.733      0.780    0.756      0.178    0.595     0.733     c0
               0.390    0.220    0.453      0.390    0.419      0.178    0.595     0.385     c1
Weighted Avg.  0.656    0.486    0.644      0.656    0.649      0.178    0.595     0.622

=== Confusion Matrix ===

   a   b   <-- classified as
 532 150 |   a = c0
 194 124 |   b = c1
```

[Nearest neighbor (IBk)]

Accuracy of decision tree (J48)	96.7%
Accuracy of nearest neighbor (IBk)	65.6%
Difference in accuracy	31.1%

Decision tree works better than nearest neighbors.

Nearest neighbor works better especially for clustering application. Basically, larger number of attributes and dataset leads to lower accuracy for nearest neighbor. The dataset being tested has 200 attributes and a majority class, and this is why nearest neighbor does not work well, because during voting, the majority class dominates.

In contrast, decision tree is more complicated in terms of classification rules compared to the nearest neighbors. This means decision tree can be used to classify correctly with more details if they have lots of attributes and large dataset. The dataset being tested has 200 attributes and 1000 examples, which is complicated but provides better performance. This is why decision tree works better than nearest neighbors.

Q2. 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.data in your zip file.

- I generated the dataset by randomly creating two input values(0 or 1), and then got target value by applying the XOR operator over those two input values in excel.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         979               97.9    %
Incorrectly Classified Instances        21                2.1    %
Kappa statistic                          0.9577
Mean absolute error                      0.0264
Root mean squared error                  0.1038
Relative absolute error                  5.3015 %
Root relative squared error             20.7977 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

               TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
               1.000    0.045    0.962      1.000    0.981      0.959    0.998     0.998     FALSE
               0.955    0.000    1.000      0.955    0.977      0.959    0.998     0.998     TRUE
Weighted Avg.  0.979    0.024    0.980      0.979    0.979      0.959    0.998     0.998

=== Confusion Matrix ===

   a   b   <-- classified as
 530   0 |   a = FALSE
  21 449 |   b = TRUE
```

[multi-layer perceptrons]

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         530               53      %
Incorrectly Classified Instances       470               47      %
Kappa statistic                          0
Mean absolute error                      0.4984
Root mean squared error                  0.4995
Relative absolute error                100.0413 %
Root relative squared error            100.0784 %
Total Number of Instances             1000

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 1.000    1.000    0.530      1.000    0.693      0.000    0.483     0.506     FALSE
                 0.000    0.000    0.000      0.000    0.000      0.000    0.483     0.536     TRUE
Weighted Avg.    0.530    0.530    0.281      0.530    0.367      0.000    0.483     0.520

=== Confusion Matrix ===

   a   b   <-- classified as
 530   0 |   a = FALSE
 470   0 |   b = TRUE
```

[Naïve Bayes]

Accuracy of multi-layer perceptrons	97.9%
Accuracy of Naïve Bayes	53%
Difference in accuracy	44.9%

Naive Bayes models independent events. So, given the two values x and y, it doesn't model any relation between two values. So, making a prediction based on x without taking into account y can't lead to a model with high performance in this situation. However, multilayer perceptron doesn't have these limitations.

Q3.

(a) 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. (1/3pt)

- 9 * 3 * 3 * 3 – 1 = 242

(b) Now assume you want to model the conditional distribution, P(ExamScore | HoursOfSleep, Studied, LikesMaterial). How many independent parameters does that conditional distribution contain? (1/3pt)

- 3 * 3 * 9 * 2= 162

(c) 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? (1/3pt)

- 3 * 8 + 3 * 2 + 3 * 2 + 2 = 38

Q5. 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? (1 pt)

- Pixel Representation

Similarity between mj1 and mj2 : 0.370861946454
Similarity between mj2 and cat : 0.619743990598
Similarity between cat and mj1 : 0.473088523579

- VGG Representation

Similarity between mj1 and mj2 : 0.960011026178
Similarity between mj2 and cat : 0.141834212136
Similarity between cat and mj1 : 0.152535112667

- mj2 and cat are the most similar in Pixel representation.
- mj1 and mj2 are the most similar in VGG representation.

Part2.

Q6. Based on the result in #5 above, what problem does pixel representation have and why does the convolutional neural network fix it? (1 pt)

- The result in #5 indicates that mj2 and cat are the most similar in Pixel representation. This is because the images are only represented as RGB values, and the model cannot take any object information about the image. The pixel representation between mj2 and cat has similar values of pixels in the same regions. This is why it's matched better than the pixel representation between mj1 and mj2. Even if mj1 and mj2 has same person in it, different position of the person(the person in mj1 is moved more to the right than the person in mj2) makes big difference. The position of the pixel value has a big influence on the pixel representation. Convolutional neural network can fix this problem by adjusting pixel information, among which features are more stable. The result indicates that m1j and mj2 are the most similar in VGG representation because the position of same image has been moved horizontally. This means that it contains same images even if it's moved. So, Convolutional neural network allows the changes in position of the image.

Q7. Compare the two results from VGG representation and pixel representation –
is one better than the other? Justify your answer in 1-2 sentences. (1pt)
- VGG representation is better than Pixel representation. Pixel representation only checks each pixels and returns the nearest neighbors based on pixel value, so if the positions of the pixels in images are different, those images could be considered to be very different. On the other hand, VGG representation takes the object information of the images and returns the nearest neighbors taking the entire images into consideration, so even if the positions of the images are different, those images could be considered to be quite similar.

Q8. 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. (1 pt)
- VGG representation(COCO_val2014_000000188308.jpg)

[Testing image]

[Most similar image]

Caption : A pair of motorcycles parked in a bike slot.
Through the observation, in the testing image(image above) and most similar image by VGG representation, top-left sides have green tree, and bottom sides have grey-colored roads. From this information, it leads to highest similarity. VGG representation didn't predict accurate captions, but all the prediction VGG representation generated makes sense. In VGG representation, object information was considered in the testing image compared with object

information in the training set. Nearest neighbor returned the image with the closest object information as that of the testing image and that is why the caption generated in this case makes more sense when compared to the pixel representation

- Pixel representation(COCO_val2014_000000183889.jpg)

[Testing image]

[Most similar image]

Caption : Darth Vader in a bathroom holding a toothbrush in one hand and staring at it.
Through the observation, in the testing image(image above) and most similar image by pixel representation, left sides have darker color, and right sides have lighter color. From this information, it leads to highest similarity. This explains why this makes wrong prediction for the caption, and it is a limitation of the pixel representation which only checks each pixels. In Pixel representation, given the testing image, the model checked which training image has similar pixel values in each position and returned the nearest neighbor. The testing image and the nearest neighbor image have similar pixel values throughout the image, that is why that caption was generated for this testing image.