Assignment 5.

Q1[8]. Consider a classifier that classified documents as follows:
`C1 :{1,3,5} C2:{2,4,6}, C3{7,8}.` Suppose the correct classification, as determined by humans, is: `C1:{1,3,6} C2:{2,4,5},C3{7,8}`. What is the precision, recall, and f-score of the algorithm on the training set? Please use the weighted average of p/r of each cluster in your calculations. For example, multiply the p/r for cluster by 3/8 because it contains 3 out of the 8 tuples in the correct classification.

Note:

- Precision = TP / TP + FP
- Recall = TP / TP + FN
- F - Score = [ 2 * Precision * Recall ] / [ Precision + Recall ]

**Cluster #1 (C1)**
TP (True Positive): 2 (correct numbers being 1 and 3)
FP (False Positive): 1 (wrong number being 5)
FN (False Negative): 1 (number needed being 6)

**Cluster #2 (C2)**
TP (True Positive): 2 (correct numbers being 2 and 4)
FP (False Positive): 1 (wrong number being 6)
FN (False Negative): 1 (number needed being 5)

**Cluster #3 (C3)**
TP (True Positive): 2 (correct numbers being 7 and 8)
FP (False Positive): 0 (no wrong numbers)
FN (False Negative): 0 (no number needed)

**Overall Algorithm Precision**
TP (True Positive): 6 (TP C1 + TP C2 + TP C3)
FP (False Positive): 2 (FP C1 + FP C2)
FN (False Negative): 2 (FP C1 + FP C2)
Precision: 6 / (6 + 2) = 0.75

**Overall Algorithm Accuracy**
TP (True Positive): 6 (TP C1 + TP C2 + TP C3)
FP (False Positive): 2 (FP C1 + FP C2)
FN (False Negative): 2 (FP C1 + FP C2)
Accuracy: 6 / (6 + 2) = 0.75

**Overall Algorithm F - Score**
P/R Weighted Average: (3/8 * 0.667) + (3/8 * 0.667) + (2/8 * 1) = 0.75
F-Score: ( 2 * 0.75 * 0.75) / (0.75 + 0.75) = 0.75

Q2[2]. Suppose that the training set is small and there is not enough data for a validation set. How can you train the hyperparameters?

One way you could train the hyperparameters is by doing cross-validation. Here, you would split the training data into a specific number of subsets/folds, taking turns altering one subset used for “testing” and the rest for “training”. This ensures that all of the data is present for both training and testing, avoiding overfitting and providing a more accurate assessment of the performance.

Q3[8]. Consider the following dataset.

| Age | Income | Owns Car (classification attribute) |
|---|---|---|
| Young | Middle class | Yes |
| Old | High | Yes |
| Young | Low | No |
| Middle age | High | Yes |
| Young | High | No |
| Old | Middle class | No |
| Middle age | Low | Yes |
| Young | Low | Yes |
| Old | Middle class | No |
| Old | High | Yes |

Use the Naïve Bayesian model to classify someone who is middle age and middle class. Is it more probable that they own a car or that they don’t own a car? **Show all calculations!** Use the formula with λ for Laplace smoothing.

**P (Yes / Car)**: 6/10<br>
**P (No / No Car)** : 4/10

| **Age** | **Yes (Car)** | **No (No Car)** |
|---|---|---|
| Young | 2/6 | 2/4 |
| Mid | 2/6 | 0/4 |
| Old | 2/6 | 2/4 |

| **Income** | **Yes (Car)** | **No (No Car)** |
|---|---|---|
| Low | 2/6 | 1/4 |
| Mid | 1/6 | 2/4 |
| High | 3/6 | 1/4 |

P( Car |Middle Age, Middle Class)
⟶ P(Car) * p(Middle Age | Car) * p(Middle Class | Car)
⟶ P(Car) * [(# Middle Age & Car) + 1 / # total yes + 1 * 3 options] * [(# Middle Class & Car) + 1 / # total yes + 1 * 3 options]
= (6/10) * [(2 + 0.1) / (6 + 0.1 * 3)] * [(1 + 0.1) / (6 + 0.1 * 3)]
= 0.6 * 0.334 * 0.174
= 0.0349

P( No Car |Middle Age, Middle Class)
⟶ P(No Car) * p(Middle Age | No Car) * p(Middle Class | No Car)
⟶ P(No Car) * [(# Middle Age & No Car) + 1 / # total no + 1 * 3 options] * [(# Middle Class & No Car) + 1 / # total no + 1 * 3 options]
= (4/10) * [(0 + 0.1) / (4 + 0.1 * 3)] * [(2 + 0.1) / (4 + 0.1 * 3)]
= 0.4 * 0.023 * 0.488
= 0.004

**Since 0.0349 > 0.004 then it is more probable that someone who is both Middle Age and Middle Class will own a car**

Q4[2]. What must be true about a dataset in order for the linear SVM classifier to work and produce good results?

In order for the linear SVM classifier to work and produce good results, the data must be perfectly linearly separable. This means that the data must only have two classification groups that can be separated using a single straight line. If the data has more than two classification groups, it must be separated using a dimensional hyperplane.

Q5[3]. Explain in detail three applications that you believe are good candidates for using Neural Networks. Please do not use any of the examples that were discussed in class.

Three applications I believe are good candidates for using Neural Networks are:

a. Bioinformatics - Convolutional neural networks can be used to predict three-dimensional protein structures from their given amino acid sequences in order to understand their functions in drug discovery.
b. Financial Gain - Recurrent neural networks can be used to analyze time series data, such as stock prices or economic indicators, in order to make informed predictions and decisions about market trends to maximize profit.
c. Climate Change - Recurrent neural networks can be used to analyze satellite imagery, weather temperature patterns and air pollution levels from various sensors in order to analyze changes to the atmosphere over time.

Q6[6]. Consider again the dataset from Q3. Convert the dataset in numeric data (e.g, young =1, middle age = 2, old = 3, apply the same for income). Apply the kNN approach to classify someone who is middle age and middle class. Use k = 3. Use the L2 Norm to compute the distance between two samples.

Note: For all questions, break ties by rolling a dice.

| Age | Income | Owns Car (classification attribute) |
|---|---|---|
| 1 | 2 | Yes |
| 3 | 3 | Yes |
| 1 | 1 | No |
| 2 | 3 | Yes |
| 1 | 3 | No |
| 3 | 2 | No |
| 2 | 1 | Yes |
| 1 | 1 | Yes |
| 3 | 2 | No |
| 3 | 3 | Yes |

Data Points –
(1, 2), (3,3), (1,1), (2,3), (1,3), (3,2), (2,1), (1,1), (3,2), (3,3)

Normalized Data Points –
(1, 2), (3,3), (1,1), (2,3), (1,3), (3,2), (2,1), (1,1), (3,2), (3,3)

Euclidean Distance for (2, 2) –
A: (1, 2), B: (3,3), C: (1,1), D: (2,3), E: (1,3), F: (3,2), G: (2,1), H: (1,1), I: (3,2), J: (3,3)

A. sqrt( (2 - 1)^2 + (2 - 2)^2 ) = 1
B. sqrt( (2 - 3)^2 + (2 - 3)^2 ) = 1.414
C. sqrt( (2 - 1)^2 + (2 - 1)^2 ) = 1.414
D. sqrt( (2 - 2)^2 + (2 - 3)^2 ) = 1
E. sqrt( (2 - 1)^2 + (2 - 3)^2 ) = 1.414
F. sqrt( (2 - 3)^2 + (2 - 2)^2 ) = 1
G. sqrt( (2 - 2)^2 + (2 - 1)^2 ) = 1
H. sqrt( (2 - 1)^2 + (2 - 1)^2 ) = 1.414
I. sqrt( (2 - 3)^2 + (2 - 2)^2 ) = 1
J. sqrt( (2 - 3)^2 + (2 - 3)^2 ) = 1.414

Three Closest Neighbors –

1. (1, 2) , Yes Category
2. (2,3) , Yes Category
3. (3,2) , No Category