**Name: Lianhan Huang 3700459; Qu Wang 3700666**

**Task 1**

**X**={color, shape, texture}

**Y**={banana, grapes, apple, orange}

**D**={**banana**:[yellow, curved, smooth]; **apple**:[red, round, smooth]; **orange**:[orange, round, rough]; **banana**:[brown, curved, smooth]; **grapes**:[green, oval, smooth]; **apple**:[green, round, smooth]}

**1.2**

Distance between l0, l1 is |0-1|=1

Distance between l0, l2 is |0-2|=2

Distance between l0, l3 is |0-3|=3

Distance between l1, l2 is |1-2|=1

Distance between l1, l3 is |1-3|=2

Distance between l2, l3 is |2-3|=1

Problem: distance not evenly distributed. Mis-classify l1 to l3 should not be more wrong than classify l1 to l2.

**1.3**

True labels y={0,2,3,0,1,2}

Predict labels by classifier ^y={0,2,3,3,1,1}

Loss = 0+0+0+3+0+1 = 4

*\*\* Q: what should the loss function be like?*

**1.4**

To be uniform means to have equal distance between each pair of labels. Since in two-dimension coordinate system, the max number of uniformed labels is 3, we need three dimension.

L0 = (1,1,1)

L1 = (1,-1,-1)

L2 = (-1,1,-1)

L3 = (-1,-1,1)

Distance = 8\*\*0.5

Normalization:

L0 = (,,)

L1 = (,,)

L2 = (,,)

L3 = (,,)

**1.5**

Distance = 1

Loss = 0+0+0+1+0+1=2

**1.6**

To encode all the features, for each feature, the distance should be the same, so:

Color: [yellow, red, orange, brown, green]

Yellow=(1,1,1,1)

Red=(1,-1,-1,-1)

Orange=(-1,1,-1,-1)

Brown=(-1,-1,1,-1)

Green=(-1,-1,-1,1)

Shape: [curved, round, oval]

Curved=(1,1)

Round=(1,-1)

Oval=(-1,1)

Texture: [smooth, rough]

Smooth=(1)

Rough=(-1)

*\*\* Q: do we need normalize all these values?*

**Task2**

**2.1.1**

F1-score is harmonic mean of recall and precious.

Precious: in true class i, how many samples are successfully counted.

Precious = *T\_positive / (T\_positive + T\_negative)*

Recall: in predicted class i, how many prediction is right.

Recall = *T\_positive / (T\_positive + F\_positive)*

F1-score = *2 \* precious \* recall / (precious + recall)*

So:

Minimal F1-score = 0

When a classifier predict all instance as negative, but there is actually instance positive.

Maximal F1-score = 1

When the classifier perfectly predict all instance.

**2.1.2**

Pre = Recall = 0 means the classifier predict all instance in True\_class1 as class2, and all instance in True\_class2 as class1.

**2.1.3**

In data set D with *n* samples, the number of positive sample is *m*, negative sample is *n-m*.

If classifier pick class randomly:

True\_Positive = *m/2*, True\_Negative = *(n-m)/2*

False\_positive = *m/2*, False\_negative = *(n-m)/2*

Precious = *0.5m / 0.5n = m/n*

Recall = *0.5m / m = 0.5*

F1 = *2m/(2m+n)*

If n-m = m:

Precious = 0.5

Recall = 0.5

F1 = 0.5

**2.2.1**

For C1: P1=250/250=1, P2=250/250=1; R1=250/250=1, R2=250/250=1 -> F=1

For C2: P1=125/250=0,5; P2=125/250=0,5; R1=125/250=0,5; R2=125/250=0,5 -> F=0,5

Choose C1

**2.2.2**

For C1: P1=0, P2=0; R1=0, R2=0 -> F=0

For C2: P1=125/250=0,5; P2=125/250=0,5; R1=125/250=0,5; R2=125/250=0,5 -> F=0,5

Choose C2

**2.2.3**

For C1: P1=200/225=0,89, P2=200/275=0,73; R1=200/275=0,73, R2=200/225=0,89 -> F=0,8

For C2: P1=200/275=0,73; P2=200/225=0,89; R1=200/225=0,89; R2=200/275=0,73 -> F=0,8

Choose C2, because higher true positive prediction matters more.

**2.2.4**

Choose C1, because lower false positive prediction matters.

**Task3**

**3.1**

Predict

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Apple | Grapes | Orange | Total |
| Apple | **5** | 1 | 2 | 8 |
| Grapes | 1 | **7** | 1 | 9 |
| Orange | 1 | 1 | **6** | 8 |
| Total | 7 | 9 | 9 | 25 |

**3.2**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Precious | Recall | F1 |
| Apple | 5/7=0,71 | 5/8=0,625 | 0,8875/1,335≈0.665 |
| Grapes | 7/9=0,78 | 7/9=0,78 | 1,2168/1,56≈0.78 |
| Orange | 6/9=0,67 | 6/8=0,75 | 1,005/1,42≈0.708 |

**3.3**

All Precious = 18/25 = 0,72

All Recall = 18/25 = 0.72

Micro F1-score = 0.72

**3.4**

Macro F1-score ≈ 0.718

**3.5**

Small number of samples.

Many samples in data set belong to a certain class, not evenly distributed.

Classifier has different performance with different class.