**CS529 Assignment 2**

[**hegde12@purdue.edu**](mailto:hegde12@purdue.edu)

**Problem 1**

1. **What is the principal assumption in the Naive Bayes’ model, and when is this assumption useful?**

The principal assumption of the Naïve Bayes’ model is that each feature is independent of the others. This assumption becomes useful while designing the classifier as P(x1, x2,⋅⋅⋅, xn | y) becomes P(x1 | y)P(x2 | y)⋅⋅⋅ P(xn | y) so the joint distribution can be found very easily.

1. **When do we expect k-NN to be better than logistic regression?**

K-NN can be expected to be better than Logistic Regression when the solution isn’t linear. Logistic regression only supports linear solutions.

1. **Suppose we have two classes. 150 examples in the + class and 50 examples in the - class. What is the entropy of the class variable (you can leave this in terms of logs)?**

Entropy formula: - where p = probability of each class

P(+) = 150/200 = ¾

P(-) = 50/200 = ¼

Entropy =-(

=-(-31125 -0.5)

=0.81125

1. **Assume that you observed the following number of requests to a web server from two domains, Domain A and Domain B (Domain A: Class1, Domain B: Class2). Number of Requests from Domain A = {2, 3, 4, 3, 4, 3, 3, 4, 5, 3, 2, 3, 4, 3, 2, 2, 3, 4, 5, 6} Number of Requests from Domain B= {22, 23, 24, 23, 24, 23, 23, 24, 25, 23} Using the data, estimate the mean, std. dev. and prior (that is, P(A)/P(A + B) and P(B)/P(A + B)) for each class**

2+3+4+3+4+3+3+4+5+3+2+3+4+3+2+2+3+4+5+6 = 68

22+23+24+23+24+23+23+24+25+23 = 234

|  |  |  |
| --- | --- | --- |
| **A** | **A-** | **(A-** |
| 2 | -1.4 | 1.96 |
| 3 | -0.4 | 0.16 |
| 4 | 0.6 | 0.36 |
| 3 | -0.4 | 0.16 |
| 4 | 0.6 | 0.36 |
| 3 | -0.4 | 0.16 |
| 3 | -0.4 | 0.16 |
| 4 | 0.6 | 0.36 |
| 5 | 1.6 | 2.56 |
| 3 | -0.4 | 0.16 |
| 2 | -1.4 | 1.96 |
| 3 | -0.4 | 0.16 |
| 4 | 0.6 | 0.36 |
| 3 | -0.4 | 0.16 |
| 2 | -1.4 | 1.96 |
| 2 | -1.4 | 1.96 |
| 3 | -0.4 | 0.16 |
| 4 | 0.6 | 0.36 |
| 5 | 1.6 | 2.56 |
| 6 | 2.6 | 6.76 |
| **Sum = 68** |  | **Sum =22.8** |

Prior = P(A)/P(A+B) = 20/20+10 = 2/3

|  |  |  |
| --- | --- | --- |
| **B** | **B-** | **(B-** |
| 22 | -1.4 | 1.96 |
| 23 | -0.4 | 0.16 |
| 24 | 0.6 | 0.36 |
| 23 | -0.4 | 0.16 |
| 24 | 0.6 | 0.36 |
| 23 | -0.4 | 0.16 |
| 23 | -0.4 | 0.16 |
| 24 | 0.6 | 0.36 |
| 25 | 1.6 | 2.56 |
| 23 | -0.4 | 0.16 |
| **Sum = 234** |  | **Sum =6.4** |

Prior = P(B)/P(A+B) = 10/20+10 = 1/3

1. **We’d like to model each conditional probability of the form P(x(i)|y) as a normal distribution for the training data below. We then use the naive Bayes assumption to write an expression for:**

**P(y = 0|x) P(y = 1|x). Please show you answer clearly.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **x1** | **x2** | **y** | **P(x1|y)** | **P(x2|y)** |
| 4 | 7 | 0 | ½ | ½ |
| -4 | 5 | 0 | ½ | ½ |
| 2 | 10 | 1 | ½ | ½ |
| 10 | 4 | 1 | ½ | ½ |
|  |  |  |  |  |

P(y=0) = ½

P(y=1) = ½

**X1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **Mean** | **S.D** |
| **Yes** | 2 | 10 | 6 | 5.65 |
| **No** | 4 | -4 | 0 | 5.65 |

P(X1|y=1) = -----(1)

P(X1|y=0) = -----(2)

**X2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  | **Mean** | **S.D** |
| **Yes** | 10 | 4 | 7 | 4.24 |
| **No** | 7 | 5 | 6 | 1.414 |

P(X2|y=1) = -----(3)

P(X2|y=0) = -----(4)

P(y=1|X) = (1) \* (3) = \*

P(y=0|X) = (2) \* (4) = \*

P(y=0|X) P(y=1|X) =

**Problem 2**

1. **Which attribute would the algorithm choose to use for the root of the tree? Show the details of your calculations**

**1.1** H(Y) = = -(-0.466-0.521) = 0.9875

Split by color

Colors: Yellow- 13, Green-3

H(Y|Yellow) = = =-(-0.431-0.530) = 0.961

H(Y|Green) = -() =-(-0.528-0.39) = 0.918

InfoGain(color) = = 0.9875-(0.780+0.172)=0.0345

Split by size

Size: Small-8, Large-8

H(Y|small) = -( = -(-0.311-0.5) = 0.811

H(Y|large) = -( = -(-0.424-0.530) = 0.955

InfoGain(size) = 0.9875 – ( = 0.9875 – (0.4055+0.4775) = 0.1045

Split by shape

Shape: Round-12, Irregular-4

H(Y|round) = -( = -(-0.5-0.5) = 1

H(Y|irregular) = -( = -(-0.311-0.5) = 0.811

InfoGain(shape) = 0.9875 – ( = 0.9875 – (0.75+0.25) = -0.0125

**Max info gain is for split using size. Thus the root of the decision tree would be size.**

After first split,

Considering the ‘small’ branch,

Total -8 Yes-6, No-2

H(Y) = = 0.811

Split by shape

Size: round-6, irregular-2

H(Y|round) = -( = 0.918

H(Y|irregular) = = -( = 0

InfoGain(shape) = 0.811-( = 0.1225

Split by color

Color: Yellow-6, Green-2

H(Y|yellow) = -( = -(-0.219-0.431) =0.65

H(Y|green) = -( = 1

InfoGain(color) = 0.811 – ( = 0.811 – (0.4875+0.25) =0.811- 0.7375 = 0.0735

**Highest info gain is for split using shape. Thus 2nd split on ‘small’ branch is based on shape.**

Considering the ‘large’ branch,

Total -8, Yes -3, No-5

H(Y) = = 0.955

Split by shape

Size: round-6, irregular-2

H(Y|round) = -( = 0.918

H(Y|irregular) = -( = 1

InfoGain(shape) = 0.955-( = 0.955-(0.6885+0.25) = 0.0165

Split by color

Color: Yellow-7, Green-1

H(Y|Yellow) = -( = -(-0.523-0.461) = 0.984

H(Y|Green) =

InfoGain(color) = 0.955-( = 0.09

**Highest info gain is for split using color. Thus 2nd split on ‘large’ branch is based on color.**

**1.2 Decision Tree:**

**16**

**8**

**8**

**2**

**1**

**7**

**6**

**5**

**1**

**6**

small

round

green

large

yellow

green

irregular

irregular

round

yellow

4-Yes,1-No

Decision - Yes

1-No

Decision - No

2-Yes

Decision - Yes

1-No

Decision - Yes

2-Yes,4-No

Decision - No

1-Yes

Decision - Yes

**1**

**1.3**

For numerical data, a split can be achieved by setting a threshold for a feature value and calculating the info gain by treating the values less than the threshold as one category and greater than the threshold as another. But it is hard to find this threshold if the dataset after sorting has a lot of switches from ‘yes’ to ‘no’ and vise-versa, a decision tree derived this way could cause overfitting and thus give a wrong result on unseen example.

**Problem 3**

Statistics:

Table

Description automatically generated

Accuracy vs K for training:

Chart, line chart

Description automatically generated

K value chosen: 5

Reason: Out of 5,11,12,13 and 14 that gave high accuracies, k>=11 might overfit the model.

**Problem 4**

* After trying with and without satandardization, I found that standardization gives a better result, thus in my implementation of Kmeans++, I have standardized the data
* Clustering Objective: Sum of distances of points from their cluster centre. Greater the sum, more widespread the cluster is

Chart, line chart

Description automatically generated

The above plot tells us that the points are much closer to the cluster centres as k value increases. By this, best k should be 5. But 5 would probably overfit the data. So thes best k should be k=4. This might overfit too. K=3 seems like a better number for clusters.

But let us examine each of them,

Chart, scatter chart

Description automatically generated

On plotting the clusters for k=5, it can be seen that the model is overfitting.

So let’s try k=4

Chart, scatter chart

Description automatically generated

This is better, but it still seems to be overfitting for X1<0.5 and X2<0.5. Let’s try k=3

Chart, scatter chart

Description automatically generated

**It can be seen that k=3 gives the best clusters.**