**Q-1. Max Posteriori Hypothesis**

When a test for steroids is given to soccer players, 98% of the players taking steroids test positive and 12% of the players are not taking steroids test positive. Suppose that 5% of soccer players take steroids. What is the maximum posteriori hypothesis for a soccer player who tests positive? What are the exact posterior probabilities?

Solution:

P(+ | ¬steroids)=0.12(False Positive), P(−|¬ Steroids)=0.88(True Positive)

P(+ | Steroids) = 0.98 (True Positive), P(−| Steroids)=0.02 (False Negative)

P(+| Steroids) ∗ P(Steroids)=(0.98) \* (0.05) = 0.049 ≈ 5%

P(+|¬ Steroids) ∗ P(¬ Steroids)= (0.12) \*(0.95) = 0.114 ≈ 11%

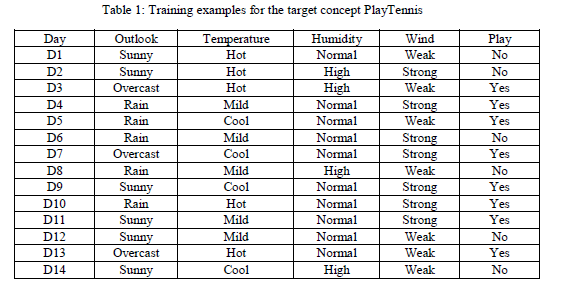
So, hMAP = argmaxhi∈H {5%, 11%} = 11%

* hMAP = ¬ Steroids

**Q-2. Naive Bayes Classifier**

**2.1** Consider the hypothesis space defined over these instances (Table 1), in which each hypothesis is represented by a pair of 4-tuples. Using the native Bayes classifier to predict the target value PlayTennis = Yes/No to the following instance.

1. Compute <Sunny, Mild, Normal, Weak>
2. Compute <Rain, Cool, High, Strong>



Solution:

(A) Overall Distribution Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind | Play |
| Sunny (6/14)  Overcast (3/14)  Rain (5/14) | Hot (5/14)  Mild (5/14)  Cool (4/14) | Normal (10/14)  High (4/14) | Weak (7/14)  Strong (7/14) | Yes (8/14)  No (6/14) |

(B) Play/ No Play Distribution Table

|  |  |  |  |
| --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind |
| Sunny / Yes (2/8)  Sunny / No (4/6) | Hot / Yes (3/8)  Hot /No (2/6) | Normal / Yes (7/8)  Normal /No (3/6) | Weak / Yes (3/8)  Weak /No (4/6) |
|
|
| Overcast / Yes (2/8)  Overcast / No (1/6) | Mild / Play (2/8)  Mild /No (3/6) | High / Yes (1/8)  High / No (3/6) | Strong / Yes (5/8)  Strong /No (2/6) |
|
|
| Rain / Yes (3/8)  Rain / No (2/6) | Cool / Yes (3/8)  Cool / No (1/6) |  |  |
|
|
|

We know, **Naive Bayes Classifier [**P(c|x)

P(c | x) = (P(x | c )\* P(c))/ P(x)

Where, C =Play|¬Play

Here, P(Play) = 8/14, and P(No) 6/14

P(Play | X)=(Play | <Sunny, Mild, Normal, Weak>)

= P(Sunny | Play) \*P (Mild | Play) \*P (Normal | Play) \*P (Weak | Play) \*P(Play)

= (2/8) \* (2/8) \* (7/8) \* (3/8) \* (8/14)

= 0.0117

≈ 1.17%

P(Not Play | X)=(No Play| <Sunny, Mild, Normal, Weak>)

= P(Sunny | No) \*P (Mild | No) \*P (Normal | No) \*P (Weak | No) \*P( No)

= (4/6) \* (3/6) \* (3/6) \* (6/6) \* (6/14)

= 0.0476

≈ 4. 76%

Then, (Play | Yes) = 0.0117 / (0.0117 + 0.0476 ) = 0.0117 / 0.0593

≈ 0.1973 ≈ 20 %

Then, (Play | No) = 0.047 / (0.0117 + 0.0476 ) = 0.0476 / 0.0593

≈ 0.8026 ≈ 80 %

**Since (Play | Yes) < (Play | No) => No Play with this condition.**

1. **Compute <Rain, Cool, High, Strong>**

P(Play | X)=(Play | <Rain, Cool, High, Strong>)

= P(Rain | Play) \*P (Cool | Play) \*P (High | Play) \*P (Strong | Play) \*P(Play)

= (3/8) \* (3/8) \* (1/8) \* (5/8) \* (8/14)

= (3\*3\*1\*5\*8)/(8\*8\*8\*\*814)

= 360 / 7168 = 0.006277

≈ 0.63%

P(Not Play | X)=(No Play| <Rain, Cool, High, Strong>)

= P(Rain | No) \*P (Cool | No) \*P (High | No) \*P (Strong | No) \*P( No)

= (2/6) \* (1/6) \* (3/6) \* (2/6) \* (6/14)

= (2\*1\*3\*2\*6)/ (6\*6\*6\*6\*14)

= 72 / 18144

= 0.00396

≈ 0.4%

Then, (Play | No) = 0.00627 / (0.00627 + 0.00396) = 0.00627 / 0.01023

≈ 0.61290 ≈ 61 %

Then, (Not Play | No) = 0.00396 / (0.00627 + 0.00396) = 0.00396 / 0.01023

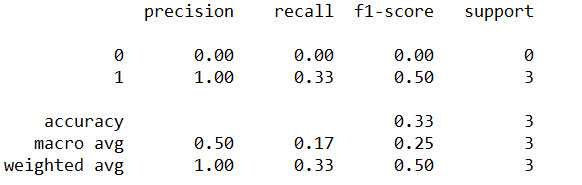
≈ 0.38709 ≈ 39 %

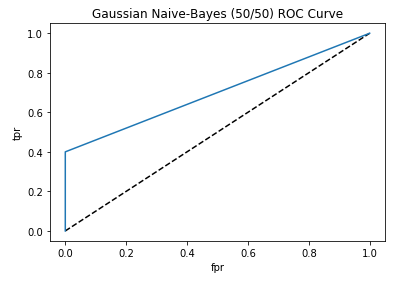
**Since (Play | No) > (Not Play | No) => Play with this condition.**

Q-2.2: Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

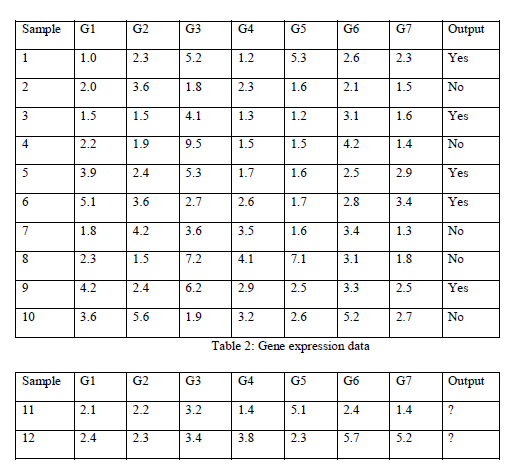
**Output in Brief:**





**Q-3. KNN (20 Points + 10 Points)**

**3.1** Table 2 is a gene expression microarray data, where each row represents a sample for a person, each column represents a gene, and each entry represents gene expression value of a gene over a sample. The output of each sample indicates whether a specific disease exists in this sample. Please use Euclidean Distance-Weighted KNN (K=3) to predict the output of the sample 11 and sample 12.



Solution:

**We know that the Euclidean Distance [**D(x,y)D(x,y)**]**

D(x,y)= √[ (x1−y1)2+(x2−y2)2+...+(xn−yn)2 ]

**Let’s suppose:**

x = sample <G1, G2, G3, G4, G5, G6, G7>

y1 = sample11<2.1, 2.1, 3.2, 1.4, 5.1, 2.4, 1.4>

y2 = sample12<2.4, 2.3, 3.4, 3.8, 2.3, 5.7, 5.2>

**For the KNN Distance:** K =

D(1, 11) = √ [ (<1,G1>−<11,G1>)2+(<1,G2>−<11,G2>)2+...+(<1,G7>−<11,G7>)2]

≈ 2.489

D(2, 11)= √ [(<2, G1>−<11,G1>)2 + (<2,G2>−<11,G2>)2+...+(<2,G7>−<11,G7>)2 ]≈4.134

Similarly,

D(3, 11) ≈ 4.172

D(4, 11) ≈ 7.476

D(5, 11) ≈ 4.723

D(6, 11) ≈ 5.33

D(7, 11) ≈ 4.681

D(8, 11) ≈ 5.336

D(9, 11) ≈ 4.958

D(10, 11) ≈ 5.875

For K = 3,

3 Nearest Neighbors of sample-11 = <sample1, sample2, sample3>

Which as Yes, No, and Yes, respectively.

* Sample11 Output should be Yes

Similarly, for the given Sample12,

D(1, 12) = √ [ (<1,G1>−<11,G1>)2+(<1,G2>−<11,G2>)2+...+(<1,G7>−<11,G7>)2]≈ 6.2436

D(2, 12)= √ [(<2, G1>−<11,G1>)2 + (<2,G2>−<11,G2>)2+...+(<2,G7>−<11,G7>)2 ] ≈5.8129

Similarly,

D(3, 12) ≈ 5.395

D(4, 12) ≈ 7.747

D(5, 12) ≈ 5.129

D(6, 12) ≈ 4.789

D(7, 12) ≈ 4.008

D(8, 12) ≈ 5.518

D(9, 12) ≈ 4.998

D(10, 12) ≈ 4.604

For K = 3,

3 Nearest Neighbors of the Sample-12 = < Sample-6, Sample-7, Sample-10>

Which as Yes, No, and No, respectively.

* Sample12 Output should be No

**I found same output using program. Please see in the attached program file.**

Q-3.2: Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

**Output in Brief:**

Train Accuracy Score: 0.91

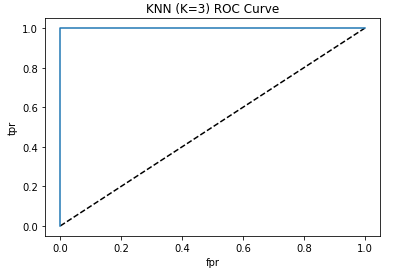
Test Accuracy Score: 0.3333333333333333

Accuracy Score 0.3333333333333333

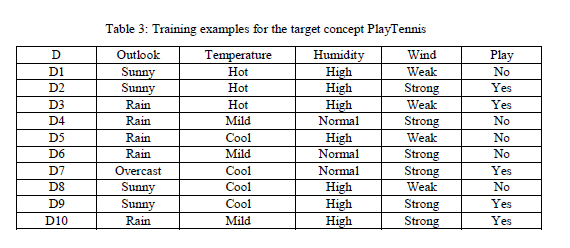
Precision Score 1.0

Recall Score 0.3333333333333333

F1 Score 0.5



Q-4.1. Consider the hypothesis space defined over these instances (Table 3), in which each hypothesis is represented by a pair of 5-tuples. Please provide a hand trace of the ID3 algorithm to build a Decision Tree Classifier.



Solution:

(A) Overall Distribution Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind | Play |
| Sunny (4/10)  Overcast (1/10)  Rain (5/10) | Hot (3/10)  Mild (3/10)  Cool (4/10) | Normal (3/10)  High (7/10) | Weak (4/10)  Strong (6/10) | Yes (5/10)  No (5/10) |

(B) Play/ No Play Distribution Table

|  |  |  |  |
| --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind |
| Sunny / Yes (2/5)  Sunny / No (2/5) | Hot / Yes (2/5)  Hot /No (1/5) | Normal / Yes (1/5)  Normal /No (2/5) | Weak / Yes (1/5)  Weak /No (3/5) |
|
|
| Overcast / Yes (1/5)  Overcast / No (0/5) | Mild / Play (1/5)  Mild /No (2/5) | High / Yes (4/5)  High / No (3/5) | Strong / Yes (4/5)  Strong /No (2/5) |
|
|
| Rain / Yes (2/5)  Rain / No (3/5) | Cool / Yes (2/5)  Cool / No (2/5) |  |  |
|
|
|

**Entropy (Single Attribute):**

E(S)= − ∑pilog‑pi

**Entropy (Multi Attributes):**

E(T,X)=∑c∈XP(c)E(c)=P(X1)⋅c1log2c1+P(X2)⋅c2log2c2+...+P(Xn)⋅cnlog2cn

**Information Gain**

G(T,X)=E(T)−E(T,X)

**Best Split Point:**

SplitInfoA(D)=∑ |Dj|/ |D|⋅log2(|Dj|/|D|)

**Gain Ratio:**

GainRatio(A)=Gain(A)/SplitInfo(A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind | Play |
| Sunny (4/10)  Overcast (1/10)  Rain (5/10) | Hot (3/10)  Mild (3/10)  Cool (4/10) | Normal (3/10)  High (7/10) | Weak (4/10)  Strong (6/10) | Yes (5/10)  No (5/10) |

|  |  |  |  |
| --- | --- | --- | --- |
| Outlook | Temperature | Humidity | Wind |
| Sunny / Yes (2/5)  Sunny / No (2/5) | Hot / Yes (2/5)  Hot /No (1/5) | Normal / Yes (1/5)  Normal /No (2/5) | Weak / Yes (1/5)  Weak /No (3/5) |
|
|
| Overcast / Yes (1/5)  Overcast / No (0/5) | Mild / Play (1/5)  Mild /No (2/5) | High / Yes (4/5)  High / No (3/5) | Strong / Yes (4/5)  Strong /No (2/5) |
|
|
| Rain / Yes (2/5)  Rain / No (3/5) | Cool / Yes (2/5)  Cool / No (2/5) |  |  |
|
|
|

**Now, Let’s draw the ID3 Algorithm**

**Play, Outlook**

E(Play, Outlook)=P(Outlook)\*E(Outlook,¬p)

=P(Sunny) \* E(Sunnyp, Sunny¬p) + P(Overcast) \* E(Overcastp, Overcast¬p) + P(Rain)\* E(Rainp, Rain¬p)

=(4/10) \* E(2/5, 2/5) + (1/10) \* E(1/5, 0/5) + (5/10) \* E(2/5, 3/5)

=(0.4)\*(1) + (0.1)\*0 + (.5)\*(0.97095) ≈ 0.8855

G(Play, Outlook) = E(Play)−E (Play, Outlook)

= E (5/10, 5/10) − 0.8855

= 1−0.8855 ≈ 0.1145 ≈ 11%

**Play, Temperature**

E(Play, Temperature)=P(Temperature)\*E(Temperature,¬p)

=P(Hot) \* E(Hotp, Hot¬p) + P(Mild) \* E(Mildp, Mild¬p) + P(Cool)\* E(Coolp, Cool¬p)

=(3/10) \* E(2/5, 1/5) + (3/10) \* E(1/5, 2/5) + (4/10) \* E(2/5, 2/5)

=(0.3)(0.9932)+(0.3)(0.9932)+(.4)(1) ≈ 0.8

G(Play, Temperature) = E(Play)−E (Play, Temperature)

= E (5/10, 5/10) − 0.8

= 1−0.8 ≈ 0.2 ≈ 20%

**Play, Humidity**

E(Play, Humidity)=P(Humidity)\*E(Humidity,¬p)

=P(Normal) \* E(Normalp, Normal¬p) + P(High) \* E(Highp, High¬p)

=(3/10) \* E(1/5, 2/5) + (7/10) \* E(4/5, 3/5)

=(0.3)(0.9932)+(0.7)(0.6997)

= 0.2979 + 0.4898 ≈ 0.7877

G(Play, Temperature) = E(Play)−E (Play, Humidity)

= E (5/10, 5/10) − 0.7877

= 1−0.8 ≈ 0.2123 ≈ 21%

**Play, Wind**

E(Play, Wind)=P(Wind)\*E(Wind,¬p)

=P(Weak) \* E(Weakp, Weak¬p) + P(Strong) \* E(Strongp, Strong¬p)

=(4/10) \* E(1/5, 3/5) + (6/10) \* E(4/5, 2/5)

=(0.4)(0.9066)+(0.6)(0.7863)

= 0.3626 + 0.4718 ≈ 0.8344

G(Play, Wind) = E(Play)−E (Play, Wind)

= E (5/10, 5/10) − 0. 8344

= 1−0.8344 ≈ 0.1655 ≈ 17%

**Thus, the largestInfoGain is for the Wind attribute. So Wind will be the root node.**

**For Wind = Strong**

| **Outlook** | **Temperature** | **Humidity** | **Play** |
| --- | --- | --- | --- |
| Sunny | Hot | High | Yes |
| Sunny | Cool | High | Yes |
| Rain | Mild | High | Yes |
| Rain | Mild | Normal | No |
| Rain | Mild | Normal | No |
| Overcast | Cool | Normal | Yes |

**Play, Outlook**

E(Play, Outlook)=P(Outlook)\*E(Outlook,¬p)

=P(Sunny) \* E(Sunnyp, Sunny¬p) + P(Overcast) \* E(Overcastp, Overcast¬p) + P(Rain)\* E(Rainp, Rain¬p)

=(2/6) \* E(2/4, 0/2) + (1/6) \* E(1/4, 0/2) + (3/6) \* E(1/3, 2/3)

=(0.3333)\*(0) + (0.1667)\*0 + (.5)\*(0.9193) ≈ 0.46

G(Play, Outlook) = E(Play)−E (Play, Outlook)

= E (4/6, 2/6) − 0.46

= 0.9193 1− 0.46 ≈ 0. 4593 ≈ 46%

**Play, Temperature**

E(Play, Temperature)=P(Temperature)\*E(Temperature,¬p)

=P(Hot) \* E(Hotp, Hot¬p) + P(Cool) \* E(Coolp, Cool¬p) + P(Mild)\* E(Mildp, Mild¬p)

=(2/6) \* E(2/4, 0/2) + (1/6) \* E(1/4, 0/2) + (3/6) \* E(1/4, 2/2)

=(0.3333)\*(0) + (0.1667)\*0 + (0.5)\*(0.5) ≈ 0.25

G(Play, Temperature) = E(Play)−E (Play, Temperature)

= E (4/6, 2/6) − 0.25

= 0.9193 1− 0.25 ≈ 0. 6693 ≈ 67%

**Play, Humidity**

E(Play, Humidity)=P(Humidity)\*E(Humidity,¬p)

=P(High) \* E(High, High ¬p) + P(Normal) \* E(Normal, Normal ¬p)

=(3/6) \* E(3/4, 0/2) + (3/6) \* E(1/3, 2/3)

=(0.5)\*(0) + (0.5)\*(0.9183) ≈ 0.46

G(Play, Humidity) = E(Play)−E (Play, Humidity)

= E (4/6, 2/6) − 0.46

= 0.9193 1− 0.46 ≈ 0. 46 ≈ 46%

**Thus, the largestInfoGain is for the Wind = Strong attribute is the Temperature. So the Temperature will be used for the tree’s second level**

**So, the next nodes will be Humidity => Outlook**

Now for (Wind = Strong) and (Humidity = Normal)

**Normal => Rain => No**

**Normal => Overcast => Yes**

And for (Wind = Strong) and (Humidity = High) => Yes

**For Wind = Weak**

**Wind = Weak**

| **Outlook** | **Temperature** | **Humidity** | **Play** |
| --- | --- | --- | --- |
| Sunny | Cool | High | No |
| Sunny | Hot | High | No |
| Rain | Hot | High | Yes |
| Rain | Cool | High | No |

**Play, Outlook**

E(Play, Outlook)=P(Outlook)\*E(Outlook,¬p)

=P(Sunny) \* E(Sunnyp, Sunny¬p) + P(Rain)\* E(Rainp, Rain¬p)

=(2/4) \* E(0/1, 2/2) + (2/4) \* E(1/1, 1/3)

=(0.5)\*(0) + (0.5)\*0.5283 ≈ 0.264

G(Play, Outlook) = E(Play)−E (Play, Outlook)

= E (1/4, 3/4) − 0.264

= 0.8112− 0.264 ≈ 0. 5470 ≈ 54%

**Play, Temperature**

E(Play, Temperature)=P(Temperature)\*E(Temperature,¬p)

=P(Hot) \* E(Hotp, Hot¬p) + P(Cool) \* E(Coolp, Cool¬p)

=(2/4) \* E(1/1, 1/3) + (2/4) \* E(0/1, 2/3)

=(0.5)\*(0.5283) + (0.5)\*0 ≈ 0.264

G(Play, Temperature) = E(Play)−E (Play, Temperature)

= E (4/6, 2/6) − 0.264

= 0.9193 1− 0.264 ≈ 0. 655 ≈ 65%

**Play, Humidity**

E(Play, Humidity)=P(Humidity)\*E(Humidity,¬p)

=P(High) \* E(High, High ¬p)

=(4/4) \* E(1/4, 3/4)

=(1)\*(0.8112) ≈ 0.8112

G(Play, Humidity) = E(Play)−E (Play, Humidity)

= E (1/4, 3/4) − 0.8112

= 0.8112− 0.8112 ≈ 0. 0 ≈ 0%

**Thus, the largestInfoGain is for the Wind = Weak attribute is the Temperature. So the Temperature will be used for the tree’s second level**

**So, the next nodes will be Outlook and Temperature**

Now for (Wind = Weak) and (Outlook = Sunny)

**Sunny => Play = No**

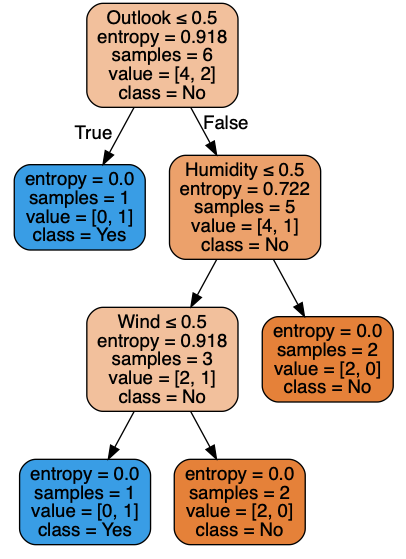
**Normal => Overcast => Yes**

And for (Wind = Weak) and (Outlook = Rain)

Rain => Hot => Play = Yes

Rain => Cool => Play = No

**Major Output:**



**4.2** Consider the following example and calculate the accuracy of the classifier with precision, recall, F1-score, sensitivity, specificity and ROC curve using Python.

[Please see Program codes in the attached file]

**Output in Brief:**

Train Accuracy Score: 1.0

Test Accuracy Score: 0.5

Accuracy Score 1.0

Precision Score 1.0

Recall Score 1.0

F1 Score 1.0

