Ivani Patel
11809154

Assignment 7

Q1. [20 pts] Entropy

Given the following sets of 2-class values (T/F), calculate the entropy on each of the below set. Note: Use the entropy definition in the lecture slides which uses $\log_2$ (base 2) instead of the log operator with natural base (base *e*).

(a) [5 pts] {*T,T,T,T*}

Answer:

Number of T's = 4,
Number of F's = 0
Total count = 4
Probability of T = 4/4 = 1,
Probability of F = 0/4 = 0
Entropy = - (1log2(1) + 0log2(0)) = 0

(b) [5 pts] {*T,T,T,F*}

Answer:
Number of T's = 3,
Number of F's = 1
Total count = 4
Probability of T = 3/4,
Probability of F = 1/4
Entropy = - (3/4log2(3/4) + 1/4log2(1/4)) = 0.8113

(c) [5 pts] {*T,T,F,F*}

Answer:
Number of T's = 2,
Number of F's = 2
Total count = 4
Probability of T = 2/4 = 0.5,
Probability of F = 2/4 = 0.5
Entropy = - (0.5log2(0.5) + 0.5log2(0.5)) = 1

(d) [5 pts] {*T,F,F,F*}

Answer:
Number of T's = 1,
Number of F's = 3
Total count = 4
Probability of T = 1/4,
Probability of F = 3/4
Entropy = - (1/4log2(1/4) + 3/4log2(3/4)) = 0.8113

Ivani Patel
11809154

Assignment 7

Q2. [40 pts] Decision Tree

Given the following training dataset about exotic dishes, we want to predict whether or not a dish is *Appealing* based on the input attributes *Temperature*, *Taste* and *Size*.

ID	Temperature	Taste	Size	Appealing
1	Hot	Salty	Small	No
2	Cold	Sweet	Large	No
3	Cold	Sweet	Large	No
4	Cold	Sour	Small	Yes
5	Hot	Sour	Small	Yes
6	Hot	Salty	Large	No
7	Hot	Sour	Large	Yes
8	Cold	Sweet	Small	Yes
9	Cold	Sweet	Small	Yes
10	Hot	Salty	Large	No

(a) [10 pts] What is the information gain Gain(Taste) at the root node of the decision tree?

Answer:
To calculate the information gain of Taste at the root node, we first need to calculate the entropy of the target variable, Appealing, for the entire dataset:

Number of positive (Yes) instances = 5
Number of negative (No) instances = 5
Total count = 10
Probability of positive instances = 5/10 = 0.5,
Probability of negative instances = 5/10 = 0.5
Entropy of target variable = - (0.5 * log2(0.5) + 0.5 * log2(0.5))
= 1
Next, we need to calculate the entropy of the target variable after splitting the data on the Taste attribute:

For Taste = Sweet, there are 4 instances, with 2 positive and 2 negative instances
For Taste = Salty, there are 3 instances, with 0 positive and 3 negative instances
For Taste = Sour, there are 3 instances, with 3 positive and 0 negative instances
The entropy for each subset is calculated as follows:

Entropy for Taste = Sweet subset: - (0.5 * log2(0.5) + 0.5 * log2(0.5)) = 1
Entropy for Taste = Salty subset: - (0/3 * log2(0/3) + 3/3 * log2(3/3)) = 0
Entropy for Taste = Sour subset: - (3/3 * log2(3/3) + 0/3 * log2(0/3)) = 0
Finally, we calculate the information gain of Taste as follows:

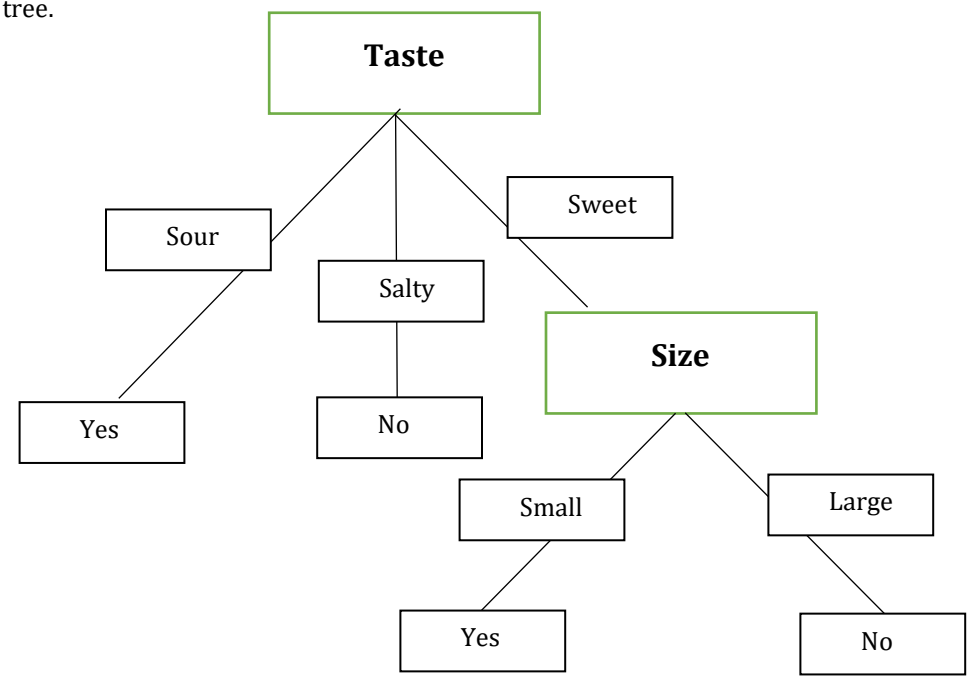
Information gain of Taste = entropy of target variable - (weighted average of entropies of Taste subsets)
Information gain of Taste = 1 - (4/10 * 1 + 3/10 * 0 + 3/10 * 0) = 0.6
Therefore, the information gain of Taste at the root node is 0.6

Ivani Patel
11809154

Assignment 7

(b) [10 pts] Suppose we build a decision tree with Taste as the attribute to split at the root. How many children does the root have? Which of them requires further splitting and which attribute to use next? Draw this tree.

Answer:



Taste have 3 children Sour, Salty and Sweet in which Sweet requires further splitting in small and large parts.

(c) [10 pts] Use the decision tree to predict the class value for the two records given by

ID	Temperature	Taste	Size
11	Hot	Salty	Small
12	Cold	Sweet	Large

Answer:

ID 11 (Hot, Salty, Small): We start at the root node and follow the path Taste -> Salty -> No. Therefore, the predicted class value is No.

ID 12 (Cold, Sweet, Large): We start at the root node and follow the path Taste -> Sweet -> Size -> Large-> No. Therefore, the predicted class value is No.

Ivani Patel
11809154

Assignment 7

(d) [10 pts] Explain why the decision tree would never choose the same attribute twice along a single path in a decision tree. Note: A single path is a path that starts from the root node and ends at a leaf node.

Answer:

The decision tree would never choose the same attribute twice along a single path in a decision tree because if an attribute was chosen twice, it would not provide any additional information to help classify the data. In other words, if an attribute was already used to split the data at a previous node along the path, then any further splitting based on that attribute would not provide any additional information or improve the classification accuracy. Therefore, it is more efficient to choose a different attribute that has not been used before along that path to maximize the amount of information gained from each split.

Ivani Patel
11809154

Assignment 7

Q3. [30 pts] Naive Bayes Classifier

The loan department of a bank has the following past loan processing records, each of which contains an applicant's income, credit history, debt and the final approval decision. These records can serve as training examples to build a decision-making software for a loan advisory system.

ID	Income	Credit History	Debt	Decision
1	0-5K	Bad	Low	Reject
2	0-5K	Good	Low	Approve
3	0-5K	Unknown	High	Reject
4	0-5K	Unknown	Low	Approve
5	0-5K	Unknown	Low	Approve
6	0-5K	Unknown	Low	Reject
7	5-10K	Bad	High	Reject
8	5-10K	Good	High	Approve
9	5-10K	Unknown	High	Approve
10	Over 10K	Unknown	Low	Approve
11	Over 10K	Bad	Low	Reject
12	Over 10K	Good	Low	Approve

We will build a Naive Bayes classifier in this question.

Recall that Naive Bayes inference (Lecture 9, slides 49-50) is based on computing $P(\text{Cause}|\text{Effect}_1,\text{Effect}_2,\text{Effect}_3)$ while assuming that:

(1) $\text{Effect}_1,\text{Effect}_2$ and Effect_3 are conditionally independent given Cause; and

(2) $P(\text{Effect}_i \mid \text{Cause})$ is given for each effect Effect_i as well as $P(\text{Cause})$.

To apply Naive Bayes inference here, we can set Cause = Decision, Effect_1 = Income, Effect_2 = Credit History and Effect_3 = Debt. However, we are not explicitly provided $P(\text{Effect}_i \mid \text{Cause})$ and $P(\text{Cause})$.

So we need to estimate these probabilities from the data.

(a) [5 pts] Estimate $P(\text{Cause} = \text{Approve})$ = **(no. of approved cases / no. of cases)** and $P(\text{Cause} = \text{Reject})$ = **(no. of rejected cases / no. of cases)** from the data.

Answer:

Using the training data, we have:

Number of Approve cases = 7

Number of Reject cases = 5

Total number of cases = 12

Therefore, we can estimate:

P(Cause = Approve) = 7 / 12 = 0.5833

P(Cause = Reject) = 5 / 12 = 0.4167

Ivani Patel
11809154

Assignment 7

(b) [5 pts] Estimate *P*(Income = *u* | Decision = Approve) = **[no. of approved cases where (**Income = *u***) / no. of approved cases]**; and *P*(Income = *u* | Decision = Reject) = **[no. of rejected cases where (**Income = *u***) / no. of rejected cases]**. Do that for each value of *u* ∈ {0−5K, 5−10K, Over10K}.

Answer:

Using the data given, we can compute these probabilities as follows:

P(Income=0-5K|Decision=Approve) = 3/6 = 0.5

P(Income=0-5K|Decision=Reject) = 3/6 = 0.5

P(Income=5-10K|Decision=Approve) = 2/3 = 0.666

P(Income=5-10K|Decision=Reject) = 1/3 = 0.333

P(Income=Over10K|Decision=Approve) = 2/3 = 0.666

P(Income=Over10K|Decision=Reject) = 1/3 = 0.333

(c) [5 pts] Estimate *P*(Credit History = *u* | Decision = Approve) = **[no. of approved cases where (**Credit History = *u***) / no. of approved cases]**; and *P*(Credit History = *u* | Decision = Reject) = **[no. of rejected cases where (**Credit History = *u***) / no. of rejected cases]**. Do that for each value of *u* ∈ {Bad, Good, Unknown}.

Answer:

P(Credit History = Bad | Decision = Approve) = 0/3 = 0

P(Credit History = Bad | Decision = Reject) = 3/3 = 1

P(Credit History = Good | Decision = Approve) = 3/3 = 1

P(Credit History = Good | Decision = Reject) = 0/3 = 0

P(Credit History = Unknown | Decision = Approve) = 4/6 = 0.666

P(Credit History = Unknown | Decision = Reject) = 2/6 = 0.333

(d) [5 pts] Estimate *P*(Debt = *u* | Decision = Approve) = **[no. of approved cases where (**Debt = *u***) / no. of approved cases]**; and *P*(Debt = *u* | Decision = Reject) = **[no. of rejected cases where (**Debt = *u***) / no. of rejected cases]**. Do that for each value of *u* ∈ {Low, High}.

Answer:

P(Debt = Low | Decision = Approve) = 5/8 = 0.625

Ivani Patel
11809154

Assignment 7

P(Debt = Low | Decision = Reject) = 3/8 = 0.375

P(Debt = High | Decision = Approve) = 2/4 = 0.5

P(Debt = High | Decision = Reject) = 2/4 = 0.5

(e) [10 pts] What is the Naive Bayes decision for an applicant who has 4K annual income, a good credit and a high amount of debt?

Answer:

P(Cause = Approve) = 7 / 12 = 0.5833

P(Cause = Reject) = 5 / 12 = 0.4167

P(Income = 4K | Decision = Approve) = 3/6 = 0.5

P(Income = 4K | Decision = Reject) = 3/6 = 0.5

P(Credit History = Good | Decision = Approve) = 3/3 = 1

P(Credit History = Good | Decision = Reject) = 0/3 = 0

P(Debt=High | Decision = Approve) = 2/4 = 0.5

P(Debt=High | Decision = Reject) = 2/4 = 0.5

P(Approve | Income=4K, Credit History=Good, Debt=High) = 0.5833 * 0.5 * 1 * 0.5
= 0.1458

P(Reject | Income=4K, Credit History=Good, Debt=High) = 0.4167 * 0.5 * 0 * 0.5
= 0

P(Approve) > P(Reject). The decision approves the application.

Ivani Patel
11809154

Assignment 7

Q4. [10 pts] Perceptron

Consider two perceptron units $\theta_1(\mathbf{x}) = w_1 \cdot \tanh(\mathbf{a}^\top \mathbf{x} + b) + h_1$ and $\theta_2(\mathbf{x}) = w_2 \cdot \text{sigmoid}(2\mathbf{a}^\top \mathbf{x} + 2b) + h_2$ where $\text{sigmoid}(z) = 1/(1 + e^{-z})$ and $\tanh(z) = (e^z - e^{-z})/(e^z + e^{-z})$.

(a) [5 pts] Show that $\tanh(z) = 2\text{sigmoid}(2z) - 1$.

Answer:

Starting with the right-hand side of the equation we want to prove:

2sigmoid(2z) - 1

= 2/(1 + e^(-2z)) - 1 (by the definition of sigmoid function)

= (2 - 1 - e^(-2z)) / (1 + e^(-2z))

= (1 - e^(-2z)) / (1 + e^(-2z))

Now, we can use the identity:

tanh(z) = (e^z - e^(-z)) / (e^z + e^(-z))

Multiplying both the numerator and denominator by e^z, we get:

tanh(z) = (e^2z - 1) / (e^2z + 1)

Dividing both the numerator and denominator by e^2z, we get:

tanh(z) = (1 - e^(-2z)) / (1 + e^(-2z))

which is the same as the right-hand side of the equation we wanted to prove. Therefore,

tanh(z) = 2sigmoid(2z) – 1 is true, and we have shown it.

(b) [5 pts] Given $w_1 = 2$ and $h_1 = 5$, find the values of w_2 and h_2 such that for any input $\mathbf{x}$, the outputs produced by θ_1 and θ_2 are always the same (regardless of how we set $\mathbf{a}$ and b).

Answer:

We want to find the values of w2 and h2 such that θ1(x) = θ2(x) for any input x. Substituting the given values for w1 and h1, we have:

w1 · tanh(aTx + b) + h1 = w2 · sigmoid(2aTx + 2b) + h2

Substituting the identity from part (a), we have:

w1 · (2sigmoid(2(aTx + b)) - 1) + h1 = w2 · sigmoid(2aTx + 2b) + h2

2w1 · sigmoid(2(aTx + b)) – w1 + h1 = w2 · sigmoid(2aTx + 2b) + h2

4 · sigmoid(2(aTx + b)) – 2 + 5 = w2 · sigmoid(2aTx + 2b) + h2

w2 = 4

h2 = 5 - 2

h2 = 3