CS 461 Homework Three
November 17th, 2024

Matthew McCaughan

Problem 1

1 Decision Tree

1.1 Finding the Root

The variable Y (Play) has two possible values: Yes and No. The entropy $H(Y)$ is calculated as:

$$H(Y) = -\left(\frac{5}{10}\log_2\frac{5}{10}\right) - \left(\frac{5}{10}\log_2\frac{5}{10}\right)$$

$$H(Y) = -\left(0.5 \cdot -1 + 0.5 \cdot -1\right) = 1$$

The entropy of Y is:

$$H(Y) = 1$$

The information gain $IG(X_k)$ for each feature X_k is calculated using:

$$IG(X_k) = H(Y) - H(Y|X_k)$$

Feature: Weather
The possible values for Weather are Sunny, Cloudy, and Rainy.
1. **Sunny**: - Entropy:

$$H(Y|\text{Sunny}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

2. **Cloudy**: - Entropy (Pure):

$$H(Y|\text{Cloudy}) = 0$$

3. **Rainy**: - Entropy:

$$H(Y|\text{Rainy}) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right) = 1$$

Weighted entropy for Weather:

$$H(Y|\text{Weather}) = \frac{3}{10} \cdot 0.918 + \frac{3}{10} \cdot 0 + \frac{4}{10} \cdot 1 = 0.55$$

$$IG(\text{Weather}) = H(Y) - H(Y|\text{Weather}) = 1 - 0.55 = 0.45$$

Feature: Temperature
The possible values for Temperature are Hot, Mild, and Cool.
1. **Hot**: - Entropy:

$$H(Y|\text{Hot}) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

2. **Mild**: - Entropy:

$$H(Y|\text{Mild}) = -\left(\frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5}\right) \approx 0.971$$

3. **Cool**: - Entropy (Pure):

$$H(Y|\text{Cool}) = 0$$

Weighted entropy for Temperature:

$$H(Y|\text{Temperature}) = \frac{3}{10}\cdot 0.918 + \frac{5}{10}\cdot 0.971 + \frac{2}{10}\cdot 0 = 0.774$$

$$IG(\text{Temperature}) = H(Y) - H(Y|\text{Temperature}) = 1 - 0.774 = 0.226$$

Feature: Humidity
The possible values for Humidity are High and Normal.
1. **High**: - Entropy:

$$H(Y|\text{High}) = -\left(\frac{4}{7}\log_2\frac{4}{7} + \frac{3}{7}\log_2\frac{3}{7}\right) \approx 0.985$$

2. **Normal**: - Entropy:

$$H(Y|\text{Normal}) = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) \approx 0.918$$

Weighted entropy for Humidity:

$$H(Y|\text{Humidity}) = \frac{7}{10}\cdot 0.985 + \frac{3}{10}\cdot 0.918 = 0.965$$

$$IG(\text{Humidity}) = H(Y) - H(Y|\text{Humidity}) = 1 - 0.965 = 0.035$$

Feature: Wind
The possible values for Wind are Weak and Strong.
1. **Weak**: - Entropy:

$$H(Y|\text{Weak}) = -\left(\frac{1}{4}\log_2\frac{1}{4} + \frac{3}{4}\log_2\frac{3}{4}\right) \approx 0.811$$

2. **Strong**: - Entropy:

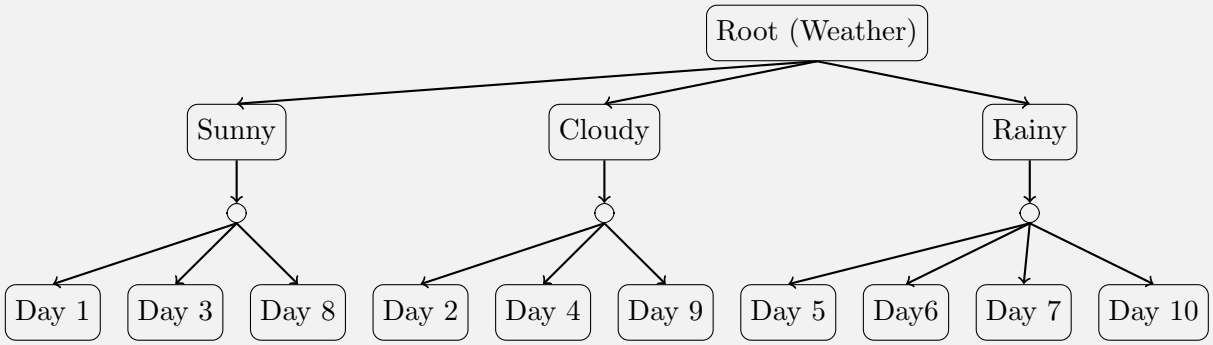
$$H(Y|\text{Strong}) = -\left(\frac{4}{6}\log_2\frac{4}{6} + \frac{2}{6}\log_2\frac{2}{6}\right) \approx 0.918$$

Weighted entropy for Wind:

$$H(Y|\text{Wind}) = \frac{4}{10}\cdot 0.811 + \frac{6}{10}\cdot 0.918 = 0.876$$

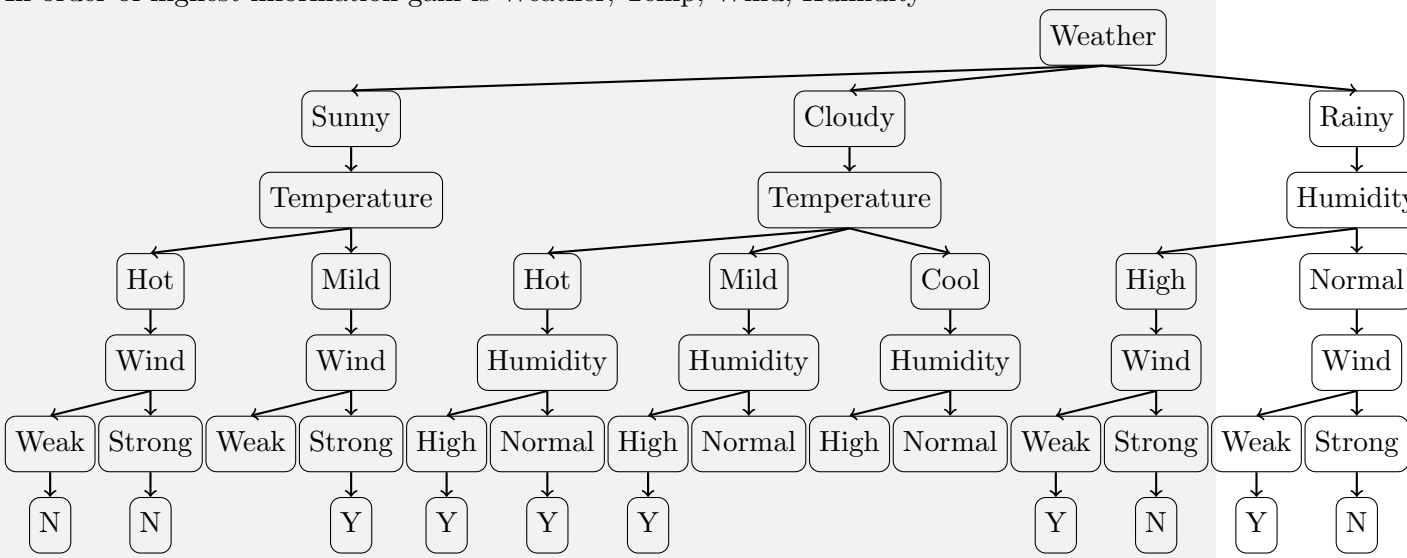
$$IG(\text{Wind}) = H(Y) - H(Y|\text{Wind}) = 1 - 0.876 = 0.124$$

Based on these information gains calculated, Weather would be the best to select as the root node, given its highest calculated information gain.



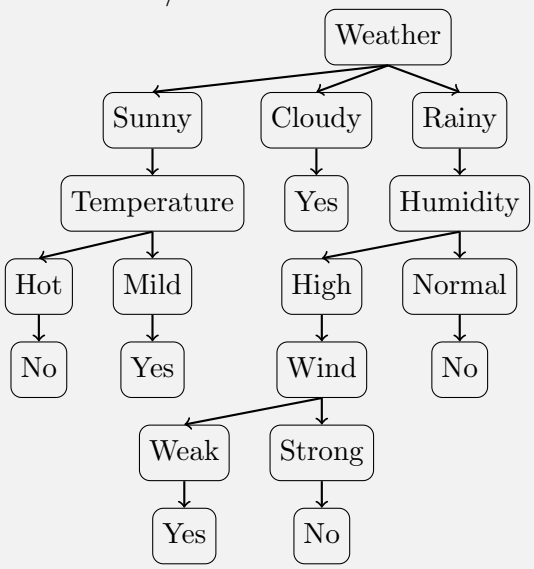
1.2 Splitting Data

In order of highest information gain is Weather, Temp, Wind, Humidity



1.3 Extra: Tree Pruning

The tree has been pruned to remove redundant features being used for the decision tree. Features are removed bottom up until all data points within the removed features have a common Yes/No decision.



Problem 2

2 Perceptron

2.1 Iterations

Given a single data point (x_1, x_2) with label $+1$: $w = (w_1, w_2) = (0, 0)$ For each misclassified point, the weights are updated as:

$$w \leftarrow w + \eta \cdot y \cdot x$$

Since its starting with $(w_1, w_2) = (0, 0)$ and only has one data point with label $+1$,The decision boundary for the initial weights $(w_1, w_2) = (0, 0)$ is:

$$w_1 x_1 + w_2 x_2 = 0$$

which does not classify any point as $+1$. The point is misclassified.
Since the point is misclassified, the weight is updated:

$$w = w + \eta \cdot y \cdot x = (0, 0) + 1 \cdot (+1) \cdot (x_1, x_2) = (x_1, x_2)$$

After this update, the weights become $w = (w_1, w_2) = (x_1, x_2)$.
With the updated weights, if the point is now correctly classified:

$$w_1 x_1 + w_2 x_2 = x_1^2 + x_2^2$$

Then this value should be greater than 1, which it is, and classifies the point correctly. Only 1 iteration is required to correctly classify the point with a decision rule.

2.2 Random Initialization

If the initial weight vector w_0 is initialized randomly, the number of iterations required for the Perceptron algorithm to correctly classify depends on if the initialization will correctly classify the point.
If the initial random weights $w_0 = (w_1, w_2)$ satisfy the condition:

$$w_1 x_1 + w_2 x_2 > 0$$

then the data point is correctly classified without requiring any updates. In this case, 0 iterations are required.
If the initial random weights misclassify the point, meaning:

$$w_1 x_1 + w_2 x_2 \leq 0$$

Then the weights must be updated After one update, the weight vector generally is updated to:

$$w = w_0 + \eta \cdot y \cdot x = w_0 + 1 \cdot (+1) \cdot (x_1, x_2) = w_0 + (x_1, x_2)$$

With this updated weight vector, the decision rule is now generally:

$$(w_1 + x_1)x_1 + (w_2 + x_2)x_2 = w_1 x_1 + w_2 x_2 + x_1^2 + x_2^2$$

Since $x_1^2 + x_2^2 > 0$, the updated rule should correctly classify the data point in 1 iteration.
Overall, it should take either 0 or 1 iteration for the perception to correctly find a decision rule.

2.3 Iterative Updates for Algorithm

Iteration 0: $w_0 = (0, 0)$
Iteration 1: Process $(0, 1), +1$

$$w_1 = w_0 + \eta \cdot y \cdot x = (0, 0) + 1 \cdot (+1) \cdot (0, 1) = (0, 1)$$

Iteration 2: Process $(1, 1), +1$

$$w_1 \cdot x_1 + w_1 \cdot x_2 = 0 \cdot 1 + 1 \cdot 1 = 1 \quad \text{(correctly classified)}$$

No update.
Iteration 3: Process $(1, 0.5), -1$

$$w_1 \cdot x_1 + w_1 \cdot x_2 = 0 \cdot 1 + 1 \cdot 0.5 = 0.5 \quad \text{(misclassified)}$$

$$w_2 = w_1 + \eta \cdot y \cdot x = (0, 1) + 1 \cdot (-1) \cdot (1, 0.5) = (-1, 0.5)$$

Iteration 4: Process $(0, 1), +1$

$$w_2 \cdot x_1 + w_2 \cdot x_2 = -1 \cdot 0 + 0.5 \cdot 1 = 0.5 \quad \text{(correctly classified)}$$

No update.
Iteration 5: Process $(1, 1), +1$

$$w_2 \cdot x_1 + w_2 \cdot x_2 = -1 \cdot 1 + 0.5 \cdot 1 = -0.5 \quad \text{(misclassified)}$$

$$w_3 = w_2 + \eta \cdot y \cdot x = (-1, 0.5) + 1 \cdot (+1) \cdot (1, 1) = (0, 1.5)$$

Iteration 6: Process $(1, 0.5), -1$

$$w_3 \cdot x_1 + w_3 \cdot x_2 = 0 \cdot 1 + 1.5 \cdot 0.5 = 0.75 \quad \text{(misclassified)}$$

$$w_4 = w_3 + \eta \cdot y \cdot x = (0, 1.5) + 1 \cdot (-1) \cdot (1, 0.5) = (-1, 1)$$

Iteration 7: Process $(0, 1), +1$

$$w_4 \cdot x_1 + w_4 \cdot x_2 = -1 \cdot 0 + 1 \cdot 1 = 1 \quad \text{(correctly classified)}$$

No update.
Iteration 8: Process $(1, 1), +1$

$$w_4 \cdot x_1 + w_4 \cdot x_2 = -1 \cdot 1 + 1 \cdot 1 = 0 \quad \text{(misclassified)}$$

$$w_5 = w_4 + \eta \cdot y \cdot x = (-1, 1) + 1 \cdot (+1) \cdot (1, 1) = (0, 2)$$

Iteration 9: Process $(1, 0.5), -1$

$$w_5 \cdot x_1 + w_5 \cdot x_2 = 0 \cdot 1 + 2 \cdot 0.5 = 1 \quad \text{(misclassified)}$$

$$w_6 = w_5 + \eta \cdot y \cdot x = (0, 2) + 1 \cdot (-1) \cdot (1, 0.5) = (-1, 1.5)$$

Iteration 10: Process $(0, 1), +1$

$$w_6 \cdot x_1 + w_6 \cdot x_2 = -1 \cdot 0 + 1.5 \cdot 1 = 1.5 \quad \text{(correctly classified)}$$

No update.
Iteration 11: Process $(1, 1), +1$

$$w_6 \cdot x_1 + w_6 \cdot x_2 = -1 \cdot 1 + 1.5 \cdot 1 = 0.5 \quad \text{(correctly classified)}$$

No update.
Iteration 12: Process $(1, 0.5), -1$

$$w_6 \cdot x_1 + w_6 \cdot x_2 = -1 \cdot 1 + 1.5 \cdot 0.5 = -0.25 \quad \text{(correctly classified)}$$

No update.
Final weight vector: $w_6 = (-1, 1.5)$

Problem 3

3 Gaussian Discriminant Analysis

3.1 Statistic Estimation

Class + Statistics:
Mean: -0.0721922106722285 Variance: 1.3031231465734459
Class - Statistics:
Mean: 0.9401561132214228 Variance: 1.9426265036964034

3.2 Test Accuracy

Test Accuracy: 61.00 Percent

3.3 Improvement/Optimality

Test Accuracy with MAP rule: 90.00 Percent

3.4 2D Data Statistics

Mean vector for class +1: [0.0130754 0.06295251]
Covariance matrix for class +1:
[[0.98285498 0.00612046] [0.00612046 1.05782804]]
Mean vector for class -1: [-0.02313942 -0.02114952]
Covariance matrix for class -1:
[[1.00329037 -0.01142356] [-0.01142356 4.97693356]]

3.5 2D Test Accuracy

Test Accuracy: 84.00 Percent

3.6 Extra: Accuracies based on Density

Test Accuracy based on densities: 85.00 Percent.
This improvement is minuscule but notable, showing that more complex models can be utilized to reasonably improve test accuracy in the model. Modeling the second class as a mixture of two Gaussians allows for slightly better distinctions and classification.

Problem 4

4 Logistic Regression

4.1 Text Vectorization

The text vectorization process of Term Frequency-Inverse Document Frequency is the combination of the two components the method is composed of, Term Frequency and Inverse Document Frequency. Term Frequency quantifies the usage of a particular word in the documents, allowing for a numerical weight to the words that compose the text. Inverse Document Frequency measures words according to their importance in relation to the text by having more frequent words being considered less than less frequent words, as less frequent words have a tendency to make the text distinguishable. This method takes both of these weights into consideration.

4.2 Dimensionality Reduction

4.3 Regression Training

Iteration 1: NLL = 0.6931276505275443
Iteration 2: NLL = 0.6931081214802107
Converged after 2 iterations.
Learned weights:
[1.62633800e-06 -2.52656339e-05 -4.75698052e-05 -2.59351229e-05 -3.08800215e-06 -3.83149660e-05 1.07296102e-05 -3.40676488e-05 -8.87621495e-06 4.36117348e-06 5.19996240e-06 -8.24762073e-07 1.35371214e-05 1.41644200e-05 -1.52619033e-06 6.10932677e-06 6.24050750e-06 7.99756229e-06 -5.54731080e-07 1.98308328e-06 1.87736165e-07 1.00324675e-

05 -1.31725226e-05 -1.37350121e-05 -4.14657603e-06 -3.24245125e-06 6.05343084e-07 1.07806232e-06 5.31854415e-06 -6.31248851e-06 -6.92846928e-07 -1.26990546e-07 -2.02299915e-06 2.94053803e-06 3.84406177e-07 8.34239867e-06 8.45965879e-07 -3.19278669e-06 2.23569722e-07 4.28834973e-06 6.93646181e-06 -2.59916567e-06 2.77975884e-06 -6.18944337e-06 -2.62167852e-06 -4.60887767e-07 2.29326225e-06 4.08024940e-06 4.33849596e-06 2.14855839e-06]

4.4 Train and Test Accuracy

Train Accuracy: 48.86 Percent
Test Accuracy: 51.60 Percent

4.5 Extra: Testing mail.txt

The spam email from "mail.txt' is classified as not spam (ham).