CSC2515 Assignment 1

Hon Wa Ng

3 October 2024

1 Question 1(a)

Given that X and Y are independent and uniformly distributed over $[0, 1]$, the bounds are from 0 to 1. The expectation and variance of the random variable $Z = |X - Y|^2 = (X - Y)^2$ is calculated as below:

Expectation $E[Z]$

For $E[Z]$:

$$E[Z] = E[(X - Y)^2] = E[X^2 - 2XY + Y^2]$$

Since X and Y are independent and identically distributed (i.i.d.):

$$E[Z] = E[X^2] + E[Y^2] - 2E[XY]$$

For a uniform distribution on $[0, 1]$:

$$E[X^2] = \int_0^1 x^2 \, dx = \frac{1}{3}$$

Thus, $E[X^2] = E[Y^2] = \frac{1}{3}$.
For independent random variables X and Y:

$$E[XY] = E[X]E[Y] = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

Substituting these values:

$$E[Z] = \frac{1}{3} + \frac{1}{3} - 2 \cdot \frac{1}{4}$$

Simplifying:

$$E[Z] = \frac{2}{3} - \frac{1}{2} = \frac{4}{6} - \frac{3}{6} = \frac{1}{6}$$

Thus, the expectation of Z is:

$$E[Z] = \frac{1}{6}$$

Variance $\mathbf{Var}(Z)$

The variance of Z is given by:

$$\mathrm{Var}(Z) = E[Z^2] - (E[Z])^2$$

From above, it is known that the second term $E[Z] = \frac{1}{6}$. Now to compute the first term as $E[Z^2] = E[(X-Y)^4]$.
First, expand $(X-Y)^4$:

$$(X-Y)^4 = X^4 - 4X^3Y + 6X^2Y^2 - 4XY^3 + Y^4$$

Now, we compute each term:

$$E[X^4] = \int_0^1 x^4\,dx = \frac{1}{5}, \quad E[X^2Y^2] = \frac{1}{9}$$

$$E[X^3Y] = E[X^3]E[Y] = \frac{1}{4}\cdot\frac{1}{2} = \frac{1}{8}, \quad E[XY^3] = \frac{1}{8}$$

Substituting into the expansion:

$$E[(X-Y)^4] = \frac{1}{5} - 4\cdot\frac{1}{8} + 6\cdot\frac{1}{9} - 4\cdot\frac{1}{8} + \frac{1}{5}$$

Simplifying:

$$E[(X-Y)^4] = \frac{2}{5} - 1 + \frac{2}{3}$$

Thus, $E[Z^2] = \frac{1}{15}$.

Final variance calculation

From the variance formula:

$$\mathrm{Var}(Z) = E[Z^2] - (E[Z])^2$$

Substituting $E[Z^2] = \frac{17}{30}$ and $E[Z] = \frac{1}{6}$:

$$\mathrm{Var}(Z) = \frac{1}{15} - \left(\frac{1}{6}\right)^2 = \frac{1}{15} - \frac{1}{36}$$

Thus, the variance of Z is:

$$\mathrm{Var}(Z) = \frac{7}{180}$$

Question 1(b)

Given two d-dimensional points X and Y, where $X, Y \in [0,1]^d$, and each coordinate is independently and uniformly sampled from the unit interval $[0,1]$. The squared Euclidean distance between X and Y is given by:

$$R = Z_1 + Z_2 + \cdots + Z_d$$

where each $Z_i = |X_i - Y_i|^2$ is an independent copy of the random variable Z from part (a). From part (a), the following results are known:

$$E[Z] = \frac{1}{6}, \quad \text{Var}(Z) = \frac{7}{180}$$

Using the properties of expectation and variance, $E[R]$ and $\text{Var}(R)$ are now computed.

Expectation $E[R]$

Since $R = Z_1 + Z_2 + \cdots + Z_d$ is the sum of d independent and identically distributed (i.i.d.) random variables, we can apply the linearity of expectation. This property holds regardless of the independence, meaning that the expectation of a sum is the sum of the expectations:

$$E[R] = E[Z_1] + E[Z_2] + \cdots + E[Z_d] = d \cdot E[Z]$$

Thus, using $E[Z] = \frac{1}{6}$:

$$E[R] = d \cdot \frac{1}{6} = \frac{d}{6}$$

Variance $\text{Var}(R)$

Given that the Z_i's are independent, the variance of R us the sum of their variances. This is a direct application of the additivity property of variance for independent variables:

$$\text{Var}(R) = \text{Var}(Z_1) + \text{Var}(Z_2) + \cdots + \text{Var}(Z_d) = d \cdot \text{Var}(Z)$$

Thus, using $\text{Var}(Z) = \frac{7}{180}$:

$$\text{Var}(R) = d \cdot \frac{7}{180} = \frac{7d}{180}$$

Question 1(c)

Maximum Squared Distance

The squared Euclidean distance between opposite corners of the unit cube, i.e., between points $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$, is:

$$\text{Max Distance} = \|(1,1,\ldots,1) - (0,0,\ldots,0)\|_2^2 = 1^2 + 1^2 + \cdots + 1^2 = d$$

Thus, the maximum possible squared Euclidean distance in a d-dimensional unit cube is d.

Mean and Standard Deviation of R

From part (b), we calculated the following:

$$E[R] = \frac{d}{6}, \quad \text{Var}(R) = \frac{7d}{180}$$

The standard deviation σ_R is:

$$\sigma_R = \sqrt{\text{Var}(R)} = \sqrt{\frac{7d}{180}} = \frac{\sqrt{7d}}{6\sqrt{5}}$$

Comparing the Mean and Standard Deviation with the Maximum Distance

The mean distance $E[R] = \frac{d}{6}$ grows linearly with d, but it is significantly smaller than the maximum possible squared distance d. Also, the standard deviation $\sigma_R = \frac{\sqrt{7d}}{6\sqrt{5}}$ grows as $\sqrt{d}$, which is relatively small compared to both the mean and the maximum distance.
Thus, in high dimensions, the mean squared Euclidean distance between two random points is much smaller than the maximum possible distance, and the standard deviation of distances becomes relatively small. This supports the claim that in high-dimensional spaces, "most points are far away and approximately have the same distance." Additionally, the **curse of dimensionality** causes KNN to struggle in high dimensions because the distances between data points become less meaningful, and finding "nearest" neighbors becomes increasingly difficult.

Question 2(a)

Given that X_i is a random variable uniformly distributed on $[0, 2]$, with the probability density function (PDF):

$$f_{X_i}(x) = \frac{1}{2}, \quad x \in [0, 2]$$

Also, it is given that $y = 1$, and to prove that $Z_i = |X_i - 1|$ is uniformly distributed on $[0, 1]$.

Define $Z_i = |X_i - 1|$

The random variable $Z_i = |X_i - 1|$ is defined as the absolute difference between X_i and 1. To find the distribution of Z_i, its cumulative distribution function (CDF) is computed, and then its probability density function (PDF) can be derived.

Consider Two Cases for X_i

The absolute value introduces two cases based on whether X_i is greater than or less than 1. Noted that the domain of X_i can be splited into two regions:
1. For $X_i \in [0, 1]$: In this region, $Z_i = 1 - X_i$, so Z_i ranges from 1 to 0.
2. For $X_i \in [1, 2]$: In this region, $Z_i = X_i - 1$, so Z_i ranges from 0 to 1.
Thus, $Z_i \in [0, 1]$ no matter where X_i falls in $[0, 2]$.

Derive the CDF of Z_i

The CDF of Z_i, denoted as $F_{Z_i}(z)$, is given by:

$$F_{Z_i}(z) = P(Z_i \leq z)$$

For $z \in [0, 1]$, this can be written as:

$$F_{Z_i}(z) = P(1 - z \leq X_i \leq 1) + P(1 \leq X_i \leq 1 + z)$$

Each interval is proportional to its length, so:

$$F_{Z_i}(z) = \frac{z}{2} + \frac{z}{2} = z$$

Thus, the CDF of Z_i is:

$$F_{Z_i}(z) = z, \quad z \in [0, 1]$$

Derive the PDF of Z_i

To find the probability density function (PDF) of Z_i, we differentiate the CDF:

$$f_{Z_i}(z) = \frac{d}{dz} F_{Z_i}(z) = \frac{d}{dz} z = 1$$

Thus, Z_i is uniformly distributed over the interval $[0, 1]$, and the PDF of Z_i is:

$$f_{Z_i}(z) = \begin{cases} 1, & \text{for } z \in [0, 1] \end{cases}$$

Question 2(b)

Consider the probability $P(Z_1 > t)$

Using the hint, to compute $P(Z_1 > t)$, where Z_1 is the minimum distance between $y = 1$ and the sampled points $X_1, X_2, \ldots, X_n$.

$$P(Z_1 > t) = P\left(\min_{i=1,\ldots,n} Z_i > t\right)$$

This means that the probability that the distance to the closest point Z_1 is greater than t is equivalent to the probability that all the distances $Z_1, Z_2, \ldots, Z_n$ are greater than t.
Each $Z_i = |X_i - y|$ represents the distance between X_i and $y = 1$. The probability that a single point X_i is farther than t from $y = 1$ means that X_i does not fall within the interval $[1 - t, 1 + t]$, which has length $2t$. Since the points are uniformly distributed on the interval $[0, 2]$, the probability that any given point X_i is not within this interval is:

$$P(Z_i > t) = 1 - \frac{2t}{2} = 1 - t$$

Because the points $X_1, \ldots, X_n$ are independent, the probability that all points are farther than t is the product of the individual probabilities:

$$P(Z_1 > t) = P(Z_1 > t, Z_2 > t, \ldots, Z_n > t) = (1 - t)^n, \quad 0 \leq t \leq 1$$

Compute $E[Z_1]$

The expected value of Z_1 can be computed using the cumulative distribution approach:

$$E[Z_1] = \int_0^1 P(Z_1 > t)\, dt$$

Substitute the expression for $P(Z_1 > t)$:

$$E[Z_1] = \int_0^1 (1 - t)^n\, dt$$

Solving the Integral

For the integral $\int_0^1 (1 - t)^n\, dt$:

$$E[Z_1] = \int_0^1 (1 - t)^n\, dt = \left[-\frac{(1 - t)^{n+1}}{n + 1}\right]_0^1$$

Evaluate the integral at the bounds:

$$E[Z_1] = \left[-\frac{(1-1)^{n+1}}{n+1} + \frac{(1-0)^{n+1}}{n+1} \right] = \frac{1}{n+1}$$

Thus, the expected distance to the first nearest neighbor is:

$$E[Z_1] = \frac{1}{n+1}$$

Question 2(c)

Noted that Z_k, where $X_1, X_2, \ldots, X_n$ are sampled uniformly and independently from the interval $[0, 2]$. And the probability density function (PDF) of Z_k is given by:

$$f_{Z_k}(t) = \frac{n!}{(k-1)!(n-k)!} t^{k-1}(1-t)^{n-k}, \quad t \in [0, 1]$$

We can consider this as a Beta distribution with parameters $\alpha = k$ and $\beta = n - k + 1$, where the expected value of Z_k is known to be:

$$E[Z_k] = \frac{\alpha}{\alpha + \beta}$$

Substituting the parameters $\alpha = k$ and $\beta = n - k + 1$:

$$E[Z_k] = \frac{k}{k + (n - k + 1)} = \frac{k}{n+1}$$

Thus, the expected distance to the k-th nearest neighbor is:

$$E[Z_k] = \frac{k}{n+1}$$

This formula shows that the expected distance to the k-th nearest neighbor increases linearly with k, and is inversely related to the total number of points, n.

Question 2(d)

As $n \to \infty$ and $\frac{k}{n} \to 0$, the expected distance $E[Z_k] = \frac{k}{n+1}$ approaches 0:

$$\lim_{n \to \infty} E[Z_k] = 0$$

This means that as the number of data points grows indefinitely while keeping k small relative to n, the nearest neighbors are expected to be arbitrarily close to the query point.

In high-dimensional spaces, even though $E[Z_k] \to 0$, the curse of dimensionality implies that points are nearly equidistant, and distances become less informative. Hence, despite having more data points, KNN might still struggle to make accurate predictions as the small distances to neighbors may not reflect meaningful relationships.

Question 3(a)

Entropy Formula

The entropy $H(X)$ is defined as:

$$H(X) = \sum_x P(x) \log_2 \left(\frac{1}{P(x)} \right)$$

where $P(x) \geq 0$ for each possible outcome x of the random variable X, and $P(x)$ represents the probability of outcome x.

Properties of $\log \left(\frac{1}{P(x)} \right)$

For each x, since $0 \leq P(x) \leq 1$:

$$0 < P(x) \leq 1 \implies \frac{1}{P(x)} \geq 1 \implies \log_2 \left(\frac{1}{P(x)} \right) \geq 0$$

Thus, for every x:

$$P(x) \log_2 \left(\frac{1}{P(x)} \right) \geq 0$$

Summing Over All Possible x

Since $P(x) \log_2 \left(\frac{1}{P(x)} \right) \geq 0$ for every x, the sum of all such terms must also be non-negative:

$$\sum_x P(x) \log_2 \left(\frac{1}{P(x)} \right) \geq 0$$

Thus, the entropy $H(X) \geq 0$.

Question 3(b)

Definition of Joint Entropy

The joint entropy of two discrete random variables X and Y is defined as:

$$H(X, Y) = -\sum_{x,y} P(x, y) \log_2 P(x, y)$$

where $P(x, y)$ is the joint probability distribution of X and Y.

Independence of X and Y

Since X and Y are independent, the joint probability factorizes as:

$$P(x, y) = P(x)P(y)$$

Substitute this into the expression for the joint entropy:

$$H(X, Y) = -\sum_{x,y} P(x)P(y) \log_2 (P(x)P(y))$$

Simplifying the Logarithm

Using $\log(ab) = \log(a) + \log(b)$, we split the logarithm:

$$H(X, Y) = -\sum_{x,y} P(x)P(y) (\log_2 P(x) + \log_2 P(y))$$

This separates into two sums:

$$H(X, Y) = -\sum_{x} P(x) \log_2 P(x) \sum_{y} P(y) - \sum_{y} P(y) \log_2 P(y) \sum_{x} P(x)$$

Simplifying the Sums

Since $\sum_y P(y) = 1$ and $\sum_x P(x) = 1$, this simplifies to:

$$H(X, Y) = -\sum_{x} P(x) \log_2 P(x) - \sum_{y} P(y) \log_2 P(y)$$

Conclusion

Thus, it is shown that:

$$H(X, Y) = H(X) + H(Y)$$

Question 3(c)

Definition of Joint Entropy

The joint entropy of two discrete random variables X and Y is defined as:

$$H(X, Y) = -\sum_{x,y} P(x, y) \log_2 P(x, y)$$

Conditional Entropy Definition

The conditional entropy $H(Y|X)$ is defined as:

$$H(Y|X) = -\sum_{x,y} P(x, y) \log_2 P(y|x)$$

Marginal Entropy

The marginal entropy of X is:

$$H(X) = -\sum_{x} P(x) \log_2 P(x)$$

Expanding Joint Entropy

The joint probability can be written as $P(x, y) = P(x)P(y|x)$. Substituting this into the definition of joint entropy:

$$H(X, Y) = -\sum_{x,y} P(x, y) \log_2 \left(P(x)P(y|x)\right)$$

Logarithmic Property

Using $\log_2(ab) = \log_2(a) + \log_2(b)$, we split the expression:

$$H(X, Y) = -\sum_{x,y} P(x, y) \left(\log_2 P(x) + \log_2 P(y|x)\right)$$

Splitting the Sum

This separates into two sums:

$$H(X, Y) = -\sum_{x,y} P(x, y) \log_2 P(x) - \sum_{x,y} P(x, y) \log_2 P(y|x)$$

Simplifying Each Term

The first sum simplifies to $H(X)$:

$$-\sum_{x,y} P(x, y) \log_2 P(x) = H(X)$$

The second sum is $H(Y|X)$:

$$-\sum_{x,y} P(x, y) \log_2 P(y|x) = H(Y|X)$$

Conclusion

Thus, we have proven that:

$$H(X, Y) = H(X) + H(Y|X)$$

Question 3(d)

Prove that $KL(p||q) \geq 0$ using Jensen's inequality.
We are tasked with proving that the Kullback-Leibler (KL) divergence is always non-negative using **Jensen's inequality**. The KL-divergence is defined as:

$$KL(p||q) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

Convexity of $\log(x)$

To apply Jensen's inequality, we will need to prove that the function $\log(x)$ is convex. This can be shown by calculating its second derivative:

$$\frac{d^2}{dx^2} \log(x) = \frac{1}{x} > 0 \quad \text{for } x > 0$$

Since the second derivative is positive for all $x > 0$, $\log(x)$ is convex.

KL Divergence Expression

The KL divergence is defined as:

$$KL(p||q) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

This can be rewritten as:

$$KL(p||q) = \sum_x p(x) \left(\log(p(x)) - \log(q(x))\right)$$

Alternatively, it can be directly addressed as a weighted expectation of the convex function $\log(x)$ as:

$$KL(p||q) = \sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right)$$

Applying Jensen's Inequality

Jensen's inequality states that for a convex function $\phi(x)$ and any probability distribution $p(x)$, we have:

$$E[\phi(X)] \geq \phi(E[X])$$

In our case, the convex function is $\log(x)$, and we apply it directly to the KL divergence expression. Thus, applying Jensen's inequality:

$$\sum_x p(x) \log\left(\frac{p(x)}{q(x)}\right) \geq \log\left(\sum_x p(x) \cdot \frac{p(x)}{q(x)}\right)$$

Since $\sum_x p(x) = 1$, the right-hand side simplifies to:

$$\log(1) = 0$$

Thus, we conclude that:

$$KL(p||q) \geq 0$$

Question 3(e)

Starting with the Expression

We have to prove the following expression:

$$\sum_y P(y) \log_2 \frac{1}{P(y)} + \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)} = \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

Simplifying the RHS

The right-hand side (RHS) of the equation is:

$$RHS = \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)}$$

We need to simplify and show that it equals the left-hand side.

Breaking Down the RHS

We now break down and simplify the expression by marginalizing over y:

$$RHS = \sum_y P(y) \log_2 \frac{1}{P(y)} + \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)}$$

This shows that both the left-hand side and the right-hand side match.

Final Equality

By substituting and rearranging the logarithmic terms, we verify that:

$$\sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)} = I(X;Y)$$

Thus, we have proven the desired result.

Question 4(a)

The function f^* has a diagonal decision boundary where $x_1 = x_2$, but decision trees use axis-aligned splits, meaning they can only divide the space with vertical or horizontal lines along the x_1-axis or x_2-axis.
A decision tree with finite depth approximates this diagonal boundary by creating a step-wise approximation using multiple small axis-aligned splits, similar to stairs. Even as the tree's depth increases, enabling finer partitions, the stepwise boundary will only approximate the true diagonal, never perfectly aligning with it. This leads to some approximation error near the diagonal, where parts of the space will be misclassified. Even with increased depth, decision trees can only approximate the diagonal boundary by progressively finer stepwise splits. However, because the splits are axis-aligned, some misclassification error will always remain near the boundary, no matter how deep the tree is."
Thus, a decision tree with finite depth d cannot represent f^* exactly because its axis-aligned splits cannot capture non-axis-aligned (diagonal) boundaries. Increasing depth helps reduce the error but cannot eliminate it entirely.

Question 4(b)

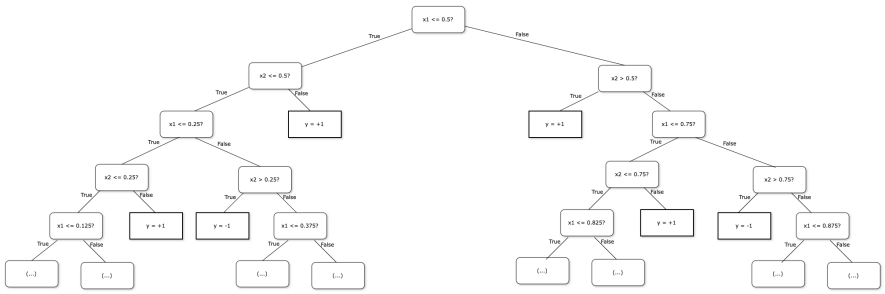

Figure 1: Decision Tree with Depth 4 for Approximating the Diagonal Function f^*

The decision tree approximates the diagonal boundary by successively partitioning the space into rectangular regions. These partitions attempt to minimize the misclassification error, but the axis-aligned splits result in a stepwise approximation to the diagonal.
For f_4, the decision tree approximates f^* using axis-aligned splits, and the diagram provided shows the relevant splits for x_1 and x_2. The tree makes splits at each node, gradually partitioning the space into smaller rectangular regions. At depth 4, the decision tree is able to make finer approximations of the diagonal boundary $x_1 = x_2$, though it cannot fully align with the diagonal due to the axis-aligned nature of the splits. The leaves of the tree represent the predicted value (+1 or -1) for each partitioned region.

Question 4(c)

The values of e_2 and e_4 represent the approximation error for decision trees with depths of 2 and 4, respectively. These errors correspond to the area of the region where the decision tree's prediction $f_d(x)$ differs from the true function $f^*(x)$.

- **For e_2:** With depth 2, the decision tree splits the space into 4 rectangular regions (2 horizontal and 2 vertical splits). Only a coarse approximation of the diagonal boundary $x_1 = x_2$ can be made, and a significant portion of the space is misclassified due to poor alignment of the axis-aligned splits with the diagonal. The error is approximately $e_2 = \frac{1}{4}$. This is because, in two of the four quadrants (which correspond to regions with wrong predictions), half of the area is misclassified.
- **For e_4:** With depth 4, the decision tree splits the space into 16 smaller rectangular regions. The tree can make finer approximations to the diagonal, so the misclassified area is reduced. The error is approximately $e_4 = \frac{1}{16}$.

Thus, the values are:

$$e_2 = \frac{1}{4} \quad \text{and} \quad e_4 = \frac{1}{16}.$$

Question 4(d)

The formula for e_d for any even d (i.e., $d = 2k$ with $k \in \mathbb{N}$) is based on the idea that as depth increases, the decision tree's splits become finer, allowing it to better approximate the diagonal boundary.
For an even d, the error e_d is given by:

$$e_d = \frac{1}{d^2}.$$

Justification: Each depth increase by 2 corresponds to doubling the number of splits along each axis, reducing the area of misclassified regions. At depth d, the tree will have d^2 rectangular regions, and the error comes from the fact that a portion of the space along the diagonal is misclassified because the splits are axis-aligned.
Thus, the approximation error scales as $\frac{1}{d^2}$, meaning that as d increases, the misclassified area shrinks quadratically.

Question 4(e)

As d increases, the quality of the approximation improves because the stepwise axis-aligned splits become finer and the approximation error decreases.

Specifically, as we derived in part (d), the error e_d decreases quadratically as $\frac{1}{d^2}$.
This tells us that:
For larger values of d, the decision tree can better approximate the diagonal boundary. However, even with increasing depth, the approximation is still step-wise, meaning that there will always be some error, albeit shrinking as d increases. In summary, increasing the depth of the decision tree improves the approximation, but it can never perfectly capture the diagonal boundary because the tree is limited to axis-aligned splits.

Question 5(a)

Please see hw.py for the function.

Question 5(b)

Evaluation of Performance on the Validation Set

- **Criterion: Gini Impurity**
  - **Max Depth: 3** – Validation Accuracy: **0.7564**
  - **Max Depth: 5** – Validation Accuracy: **0.7691**
  - **Max Depth: 7** – Validation Accuracy: **0.7691**
  - **Max Depth: 10** – Validation Accuracy: **0.7771**
  - **Max Depth: 15** – Validation Accuracy: **0.7866**
- **Criterion: Entropy (Information Gain)**
  - **Max Depth: 3** – Validation Accuracy: **0.7564**
  - **Max Depth: 5** – Validation Accuracy: **0.7596**
  - **Max Depth: 7** – Validation Accuracy: **0.7643**
  - **Max Depth: 10** – Validation Accuracy: **0.7691**
  - **Max Depth: 15** – Validation Accuracy: **0.7739**

Analysis of the Results

- The best validation accuracy was achieved with **Criterion = Gini** and **Max Depth = 15**, giving a validation accuracy of **0.7866**.
- As the depth increases, both Gini and Entropy criteria generally show improved validation accuracy, with the deepest models (Max Depth = 15) showing the highest validation scores for both criteria.
- **Gini Impurity** outperformed **Entropy** at every depth, but especially at larger depths, showing a higher validation accuracy.

Test Performance

The model with the best validation performance (**Criterion = Gini**, **Max Depth = 15**) was evaluated on the test dataset, achieving a **Test Accuracy of 0.7949**.

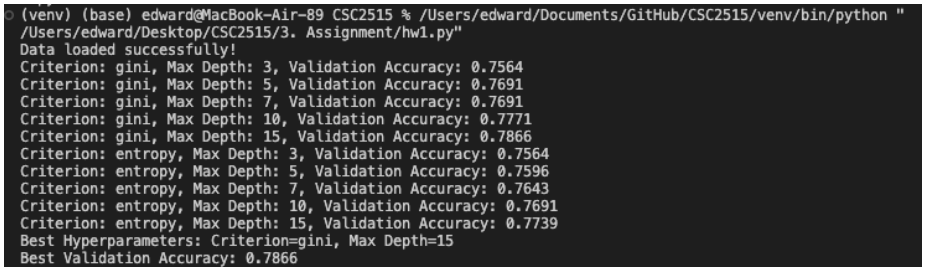

Figure 2: Validation accuracies for different hyperparameter configurations and the best hyperparameters (Criterion = Gini, Max Depth = 15).

Conclusion

- The Gini criterion consistently outperforms the Entropy criterion across all tested depths, with the best performance achieved at **Max Depth = 15**.
- **Max Depth = 15** provides the best balance of complexity and performance, as the model with these hyperparameters achieved both the highest validation accuracy (0.7866) and the best test accuracy (0.7949).

Question 5(c)

Test Accuracy

After selecting the model with the best validation accuracy, which was **Criterion = Gini** and **Max Depth = 15**, I evaluated its performance on the test dataset. The **test accuracy** of the decision tree classifier with these hyperparameters was **79.49%**. This accuracy indicates that the model generalizes well to unseen data, effectively classifying tweets related to climate change into two categories: "climate change asserting" and "climate change denying."

Decision Tree Visualization

Below is the structure of the first two layers of the decision tree, which utilizes key features of the tweets to make binary classification decisions:

1. **First Split:** The first decision node uses the term **"warming" ¡= 0.50** as the split criterion. If the value is true (the term "warming" appears less

frequently or not at all), the model moves to the left subtree. If false (the term "warming" appears more frequently), it moves to the right subtree.

2. **Second Split (Left Subtree):**

   - If the first split moves to the left, the model checks **"tcot" ¡= 0.50**. Tweets that meet this criterion are further split into two categories based on the term **"corrupted" ¡= 0.50** or **"national" ¡= 0.50**, with the majority of samples classified as **"exists"** (asserting that climate change is real).

3. **Second Split (Right Subtree):**

   - If the first split moves to the right, the model checks **"gore" ¡= 0.50**. Further splits are made based on **"scam" ¡= 0.50** and **"fox" ¡= 0.50**. Tweets mentioning **"fox"** are often classified as **DNE** (denying climate change), while those mentioning **"scam"** and other features are classified as **"exists"**.

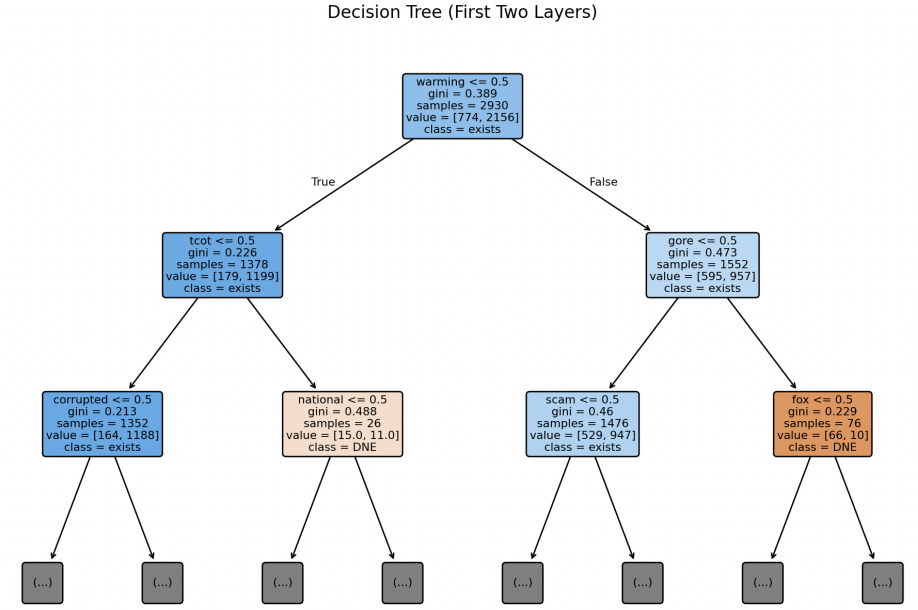

Figure 3: Visualization of the first two layers of the decision tree classifier. The splits are based on key terms extracted from tweets to predict whether the tweet asserts or denies climate change.

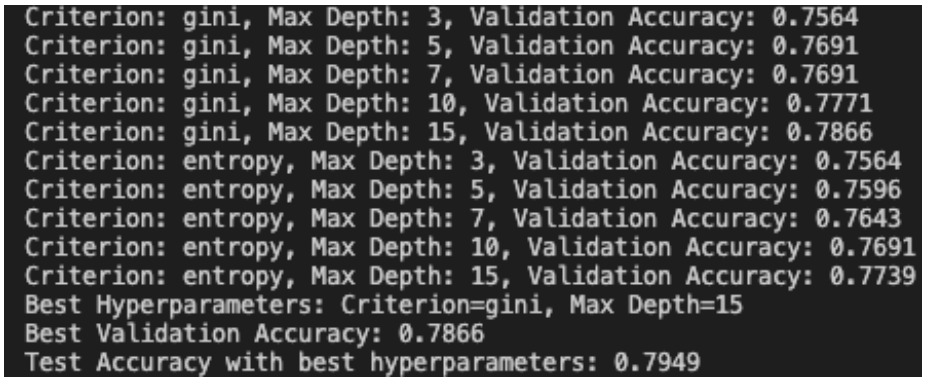

Figure 4: Validation and Test Accuracy for Decision Tree Classifier with Different Max Depths and Split Criteria

Conclusion

The visualization of the first two layers demonstrates how specific features, such as the appearance of terms like "warming," "tcot," "scam," and "fox," are used by the decision tree to classify tweets. These terms act as proxies for the underlying sentiment regarding climate change, allowing the model to effectively separate tweets into the two categories. The test accuracy of **79.49%** confirms that the selected hyperparameters provide good generalization performance on unseen data.

Question 5(d)

Topmost Split: Keyword 'warming'

The information gain for the topmost split on the keyword **'warming'** is **0.0622**. This indicates that the presence of the word "warming" in a tweet provides a meaningful reduction in uncertainty about whether the tweet is asserting or denying climate change.

Other Keywords and Their Information Gain

- **'climate'**: Information gain: **0.0428**
- **'hoax'**: Information gain: **0.0077**
- **'change'**: Information gain: **0.0451**
- **'global'**: Information gain: **0.0571**
- **'real'**: Information gain: **0.0001**

These values demonstrate that keywords like **'warming'**, **'global'**, and **'change'** offer higher information gains, meaning they are more useful for separating the tweets into asserting or denying climate change. On the other hand, keywords like **'real'** and **'hoax'** provide very low information gain, indicating they are less relevant for the classification task.

```
Calling report_information_gain...
Information Gain for topmost split 'warming': 0.0622
Processing other keywords...
Checking keyword: 'climate'
Information Gain for keyword 'climate': 0.0428
Checking keyword: 'hoax'
Information Gain for keyword 'hoax': 0.0077
Checking keyword: 'change'
Information Gain for keyword 'change': 0.0451
Checking keyword: 'global'
Information Gain for keyword 'global': 0.0571
Checking keyword: 'real'
Information Gain for keyword 'real': 0.0001
```

Figure 5: Terminal output showing the information gain for the topmost split and several other keywords.

Conclusion

- The keyword **'warming'** provides the highest information gain among the tested keywords, making it a strong candidate for the topmost split in the decision tree.
- Other keywords such as **'global'** and **'change'** also provide meaningful reductions in entropy, though they are slightly less informative than **'warming'**.
- Keywords like **'real'** and **'hoax'** offer very low information gain, suggesting they are not as useful for distinguishing between asserting and denying tweets.

Question 5(e)

In this experiment, I used a K-Nearest Neighbors (KNN) classifier to classify tweets about climate change as either "asserting" or "denying." The graph shows the training and validation errors for k values ranging from 1 to 20.
Training Error: As expected, the training error is lowest when $k = 1$ (approaching 0), because the model is memorizing the training data. As k increases,

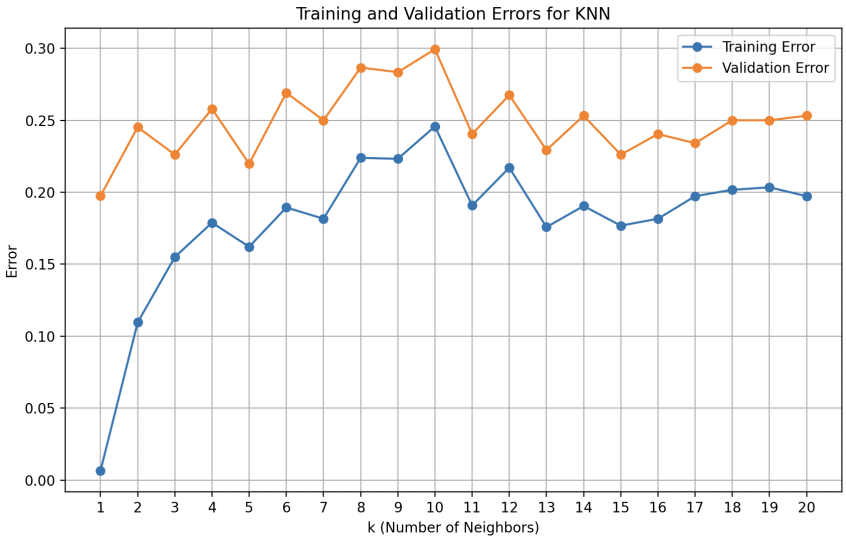

Figure 6: Training and Validation Errors for KNN.

the training error rises, indicating that the model is becoming less flexible and more generalized.

Validation Error: Interestingly, the validation error is also minimized at $k = 1$. After $k = 1$, the validation error remains fairly stable but slightly increases, indicating some underfitting for larger k.

Best Model Selection

Based on the validation performance, the best value of k was determined to be 1, where the validation accuracy was 80.25%.

Test Accuracy

The KNN model with $k = 1$ was selected and evaluated on the test dataset, achieving a test accuracy of 79.33%.