Homework 2: Tree Models

(Due Date: Jun 24, 2025)

MATH QUESTIONS

Problem 1: Information Gain

We are given 6 training examples with two binary features X_1 and X_2, and a 3-class target variable $Y \in \{1, 2, 3\}$:

X_1	X_2	Y
1	1	1
1	1	1
1	1	2
1	0	3
0	0	2
0	0	3

1. To calculate the information gain, we calculate the following:

Step 1: Compute Entropy at the Root

We compute the entropy of the dataset before any split:

$$P(Y=1) = \frac{2}{6}, \quad P(Y=2) = \frac{2}{6}, \quad P(Y=3) = \frac{2}{6}$$

$$H(S) = -\sum_{i=1}^{3} P(Y=i)\log_2 P(Y=i) = -3 \cdot \frac{2}{6} \cdot \log_2\left(\frac{2}{6}\right) = \log_2 3 \approx 1.585$$

Step 2: Compute Conditional Entropy for X_1

$X_1 = 1 : Y = [1,1,2,3] \rightarrow Counts : [2,1,1] \Rightarrow Probabilities : \left[\frac{2}{4}, \frac{1}{4}, \frac{1}{4}\right]$

$$H(X_1 = 1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \cdot\frac{1}{4}\log_2\frac{1}{4} + \cdot\frac{1}{4}\log_2\frac{1}{4}\right) = (0.5 \cdot 1 + 2 \cdot 0.25 \cdot 2) = 1.5$$

$X_1 = 0 : Y = [2,3] \rightarrow Counts : [1,1] \Rightarrow Probabilities : \left[\frac{1}{2}, \frac{1}{2}\right]$

$$H(X_1 = 0) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

Weighted average:

$$H(Y|X_1) = P(X_1=1) \cdot H(X_1=1) + P(X_1=0) \cdot H(X_1=0) = \frac{4}{6} \cdot 1.5 + \frac{2}{6} \cdot 1.0 = 1.333$$

Information Gain:

$$IG(X_1) = H(S) - H(Y|X_1) = 1.585 - 1.333 = 0.252$$

Step 3: Compute Conditional Entropy for X_2

$X_2 = 1 : Y = [1,1,2] \rightarrow Counts : [2,1] \Rightarrow Probabilities : \left[\frac{2}{3}, \frac{1}{3}\right]$

$$H(X_2 = 1) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

$$X_2 = 0 : Y = [3, 2, 3] \rightarrow Counts : [1, 2] \Rightarrow Probabilities : \left[\tfrac{1}{3}, \tfrac{2}{3}\right]$$

$$H(X_2 = 0) = -\left(\frac{1}{3}\log_2\frac{1}{3} + \frac{2}{3}\log_2\frac{2}{3}\right) \approx 0.918$$

Weighted average:

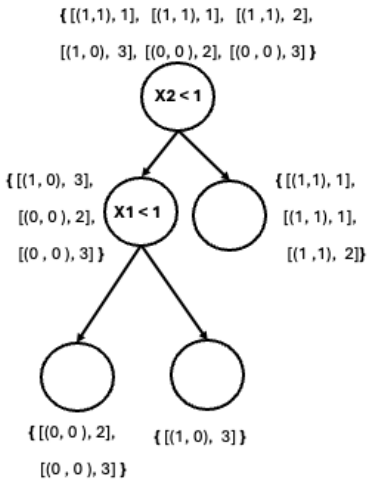
$$H(Y|X_2) = \frac{3}{6} \cdot 0.918 + \frac{3}{6} \cdot 0.918 = 0.918$$

Information Gain:

$$IG(X_2) = 1.585 - 0.918 = 0.667$$

2. Since $IG(X_2) > IG(X_1)$, we split on X_2. Figure 2 depicts the decision tree.

Figure 1: Classification tree with split on X_2



3. Here, we classify the test example $X_1 = 0$, and $X_2 = 1$

- $X_2 = 1$, we follow the right branch.
- $X_1 = 0$, no matching training data.
- Thus, we use the majority class from $X_2 = 1$ branch: Y = [1, 1, 2] $\rightarrow$ majority is Y = 1. Therefore, the classification for test example $X_1 = 0$, and $X_2 = 1$ is $Y = 1$.

Problem 2: Entropy

1. In this problem, we compute the conditional entropy of each attribute (X_1 and X_2) and choose the attribute that results in the lowest entropy after the split.

Step 1: Entropy Before Splitting

As in Problem 1:

$$H(Y) = -\sum_{i=1}^{3} \frac{2}{6} \log_2 \left(\frac{2}{6}\right) = \log_2(3) \approx 1.585$$

Step 2: Conditional Entropy for X_1

$X_1 = 1$: Y = [1, 1, 2, 3] → counts: [2, 1, 1]

$$H(X_1 = 1) = -\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{1}{4}\log_2\frac{1}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) = 1.5$$

$X_1 = 0$: Y = [2, 3] → counts: [1, 1]

$$H(X_1 = 0) = -\left(\frac{1}{2}\log_2\frac{1}{2} + \frac{1}{2}\log_2\frac{1}{2}\right) = 1.0$$

Weighted entropy:

$$H(Y|X_1) = \frac{4}{6} \cdot 1.5 + \frac{2}{6} \cdot 1.0 = 1.333$$

Step 3: Conditional Entropy for X_2

$X_2 = 1$: Y = [1, 1, 2] → counts: [2, 1]

$$H(X_2 = 1) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

$X_2 = 0$: Y = [3, 2, 3] → counts: [1, 2]

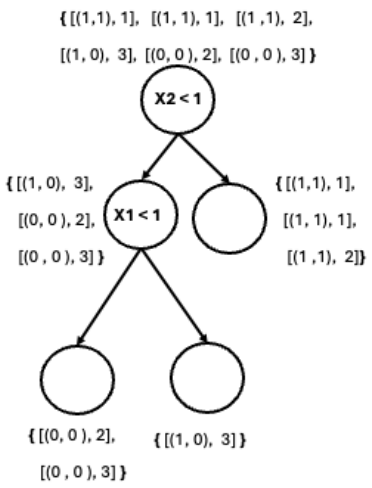
$$H(X_2 = 0) = -\left(\frac{2}{3}\log_2\frac{2}{3} + \frac{1}{3}\log_2\frac{1}{3}\right) \approx 0.918$$

Weighted entropy:

$$H(Y|X_2) = \frac{3}{6} \cdot 0.918 + \frac{3}{6} \cdot 0.918 = 0.918$$

2. Since $H(Y|X_2) < H(Y|X_1)$, We choose to split first on attribute X_2.

Figure 2: Classification tree with split on X_2



3. Classification for test example $X_1 = 0$, $X_2 = 1$.

- $X_2 = 1$, we follow the right branch.
- $X_1 = 0$, but there is no training sample with $X_2 = 1$ and $X_1 = 0$.
- Therefore, we default to the majority class from the $X_2 = 1$ group, which is $Y = 1$. Therefore, the classification for test example $X_1 = 0$, and $X_2 = 1$ is $Y = 1$.

PROGRAMMING QUESTIONS

Part A: Classification Tree

1. Data Processing and EDA

(a) An 80/20 split of the dataset is performed into training and validation sets, ensuring that the class distribution of the target variable is preserved via stratified sampling.

(b) The dataset used in this project is the *Bank Marketing* dataset. It contains information related to direct marketing campaigns (phone calls) of one of the banks. The goal is to predict whether a client will subscribe to a term deposit (`y` = `yes`/`no`).

The dataset contains 16 input features derived from both numerical and categorical variables. Some key variables include:

- `age`, `balance`, `duration`, `pdays`: numerical features.
- `job`, `marital`, `education`, `month`, `poutcome`: categorical variables. The categorical variables are mapped to numeric values.

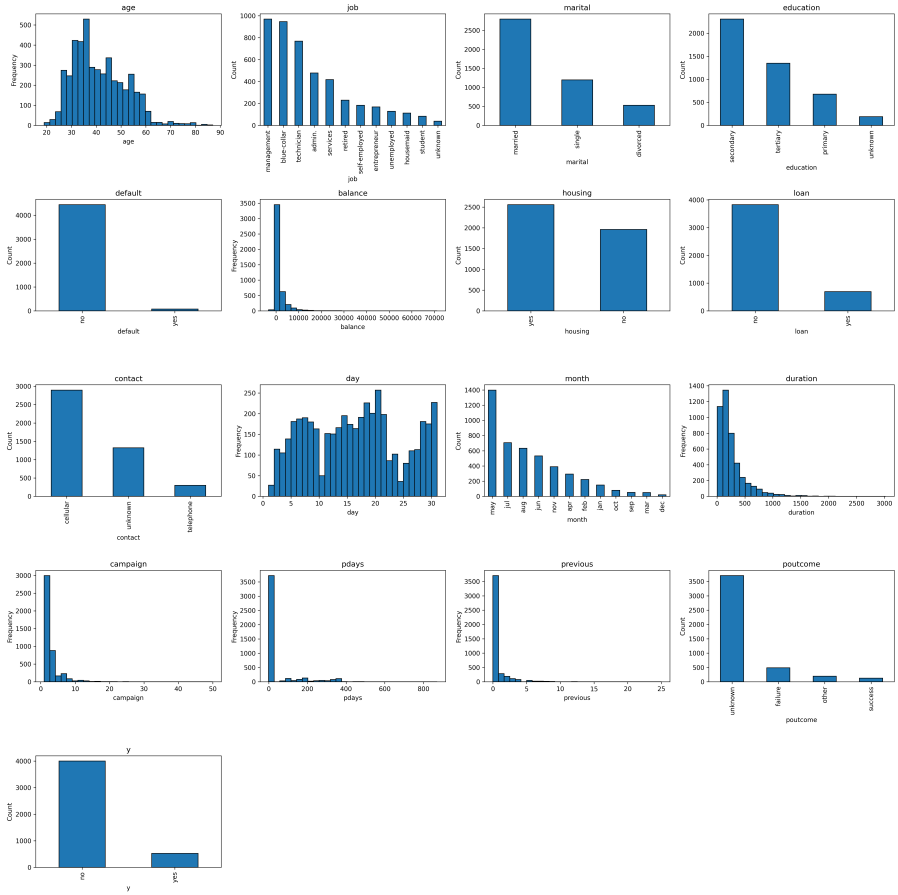
There are no missing values in the dataset; however, several features contain entries labeled as `"unknown"`, which may indicate missing or ambiguous information. These `"unknown"` values are not imputed.

The target variable `y` is binary (1 for "yes", 0 for "no") and is **highly imbalanced**, with the majority class being "no". As a result, we applied stratified sampling to preserve class distribution in the training and validation splits, and used upsampling on the training data to address the imbalance during model training.

(c) The features and the labels are extracted.

(d) The histograms (in Figure 3) below provide a visual overview of the distribution of both numerical and categorical features in the dataset. The last barplot is the barplot of the target y.

Figure 3: Histogram of all numerical features and barplot of all categorical features.

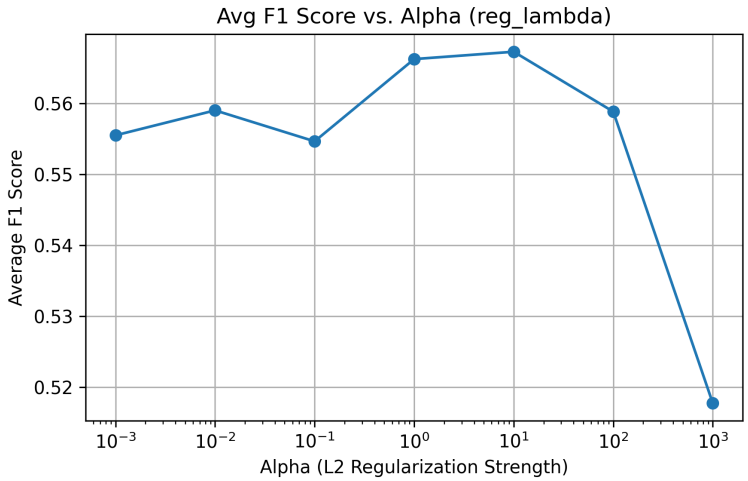


- **age:** Right-skewed, with most clients between 30 and 60 years old.
- **balance:** Highly skewed, with the majority of balances concentrated near zero and a few extreme positive outliers.
- **duration:** Strongly right-skewed. Many calls are short, but a small number last over 1000 seconds. This variable could be highly predictive.
- **job, marital, education:** These categorical features show imbalanced distributions. For example, "blue-collar" and "management" dominate the `job` category, while "married" dominates `marital`.
- **default, loan, housing:** Most clients do not have credit default, or a personal loan, but do have a housing loan.
- **contact:** The majority of contacts were made via cellular phones, with few via telephone. A notable number of values are labeled "unknown".
- **month:** Marketing calls are concentrated in certain months, particularly May, July, and August.
- **campaign, pdays, previous:** These features are right-skewed and include many zeros, suggesting many first-contact clients or recently inactive ones.
- **poutcome:** Most values are "unknown", though "failure" and "success" appear in smaller proportions, reflecting the outcomes of previous campaigns.
- **Target variable (y):** The dataset is highly imbalanced, with the majority class being "no" for subscription.

Part B: Boosting

1. XGBoost models with L2 regularization and bootstrapping (100 iterations are implemented and evaluated for different values of α. The plot in Figure 4 shows the average F1 score versus α, the strength of L2 regularization. Based on the plot, the optimal α is 10 and is used for `my_best_model` in part 2 of this problem.

Figure 4: Average F1 score versus α



3. The ROC (Receiver Operating Characteristic) curve for `my_best_model` is shown in Figure 5. It illustrates the trade-off between the True Positive Rate (sensitivity) and the False Positive Rate (1 - specificity) across various threshold settings.

- The curve consistently stays above the diagonal reference line, indicating that the model performs significantly better than random guessing.
- The Area Under the Curve (AUC) is **0.9117**, which reflects a strong discriminatory ability of the classifier.
- AUC values closer to 1.0 represent better performance. In this case, an AUC above 0.9 is considered excellent, suggesting that the model is very effective in distinguishing between the positive and negative classes.

Figure 5: ROC Curve for `my_best_model` with AUC = 0.9117