# Decision Tree Project Report

Your Name

## 1 Overview

This project involves building a decision tree model entirely from scratch for academic purposes, specifically as part of an Artificial Intelligence course. The primary objective is to implement the decision tree algorithm manually, understand the classification process, and apply it to real-world data. This project helps in reinforcing the theoretical concepts of machine learning and decision-making techniques, aimed at classifying data based on its features.

## 2 Preprocessing

- The dataset contained missing values, which were handled by either filling or removing incomplete data.
- Numerical columns were transformed into categorical ones, and high-cardinality categorical columns were reduced to fewer categories to improve efficiency.
- Label encoding was applied, where the target labels (`y`) were transformed into binary labels (0 and 1).

## 3 Decision Tree Implementation

- Two classes, `DecisionTree` and `Node`, were implemented from scratch to build the decision tree model. Separate functions were created for calculating entropy and Gini index.
- A small weather-related dataset was used to debug the code and ensure correctness.

## 4 Training and Testing

- Preprocessed data was divided into training and test sets. Two models were built using the `DecisionTree` class: one utilizing entropy and the other using the Gini index.

- Precision and accuracy were calculated for both models.
- A threshold of 0.05 was used for the Gini index to construct better-performing trees. The Gini-based tree performed slightly worse than the entropy-based tree but was more efficient in terms of computation.

## 5 Challenges Encountered

- In the training data, some rows had inconsistent values that made it challenging for the model to predict correctly. For these cases, majority voting was used as a fallback method to assign a label to the problematic rows.
- This issue did not significantly affect the test data, as only a small fraction of rows in the test set exhibited this problem.

## 6 Final Thoughts

- Majority voting was employed when a node couldn't make a clear decision during training. This fallback mechanism ensured better stability in the output tree.
- The preprocessing techniques, especially label encoding and category reduction, significantly improved the decision tree's performance.
- Through iterative improvements and debugging, the decision tree models were optimized to yield satisfactory results.