**Data learning: Linear Model vs Classification Tree**

Example 1: Single Outcome Variable and Two Covariates

Data source: Simulation by R or Python programming language

Data:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Study Hours** | **%Attendance** | **Outcome** | **Study Hours** | **%Attendance** | **Outcome** | **Study Hours** | **%Attendance** | **Outcome** |
| 7 | 90 | Pass | 8 | 89 | Pass | 1 | 46 | Fail |
| 4 | 46 | Fail | 8 | 97 | Pass | 6 | 83 | Pass |
| 8 | 60 | Fail | 3 | 48 | Fail | 9 | 47 | Fail |
| 5 | 48 | Fail | 6 | 65 | Fail | 1 | 86 | Fail |
| 7 | 78 | Pass | 5 | 92 | Fail | 3 | 74 | Fail |
| 3 | 57 | Fail | 2 | 41 | Fail | 7 | 53 | Fail |
| 7 | 43 | Fail | 8 | 59 | Fail | 4 | 56 | Fail |
| 8 | 64 | Fail | 6 | 67 | Fail | 9 | 75 | Pass |
| 5 | 99 | Fail | 2 | 86 | Fail | 3 | 89 | Fail |
| 4 | 53 | Fail | 5 | 99 | Fail | 5 | 79 | Fail |

Step 1: Formulate your research question based on this data set.

Are study hours and class attendance important factors classifying students’ course achievements?

Step 2: Define data role and data type.

|  |  |  |
| --- | --- | --- |
| Variable name | Data role | Data type |
| Study hours | Predictor (Feature) | Numerical (Continuous) |
| Class attendance | Predictor (Feature) | Numerical (Continuous) |
| Course achievement outcome | Target (Response) | Categorical (Binary: Pass/Fail) |

Step 3: Select an appropriate method to analyze the data e.g., statistical learning, machine learning.

**Binary logistic regression** is used to analyze the data because

* Used for predicting binary outcomes.
* Provides interpretable coefficients showing the relationship between predictors and the likelihood of passing.

**Classification tree** is used to analyze the data because

* Non-parametric method for classification.
* Provides a visual decision-making process based on splits in the predictors.

Step 4: Collect data.

A simulated data set of 30 students from R or Python programming language is used for this example as shown in above table.

Step 5: Explore your data using numerical summary and graphs

(Hint: explore data separated by outcome class: see Iris flower data as an example)

Descriptive statistics:

A black and white screen with numbers

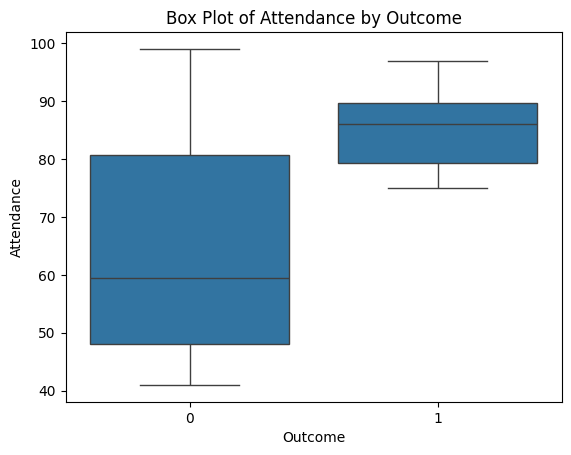
Description automatically generated

A black and white text

Description automatically generated with medium confidence

Show your graphs:

A diagram of a box plot

Description automatically generated 

Interpret:

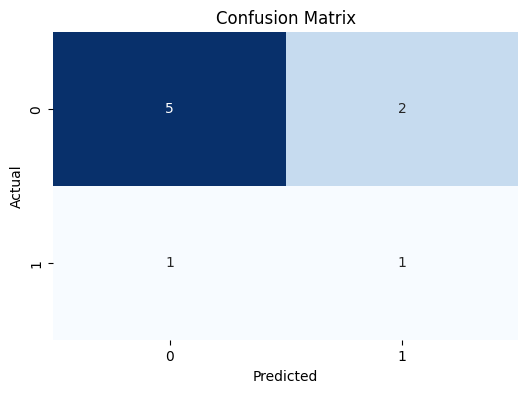
“Pass” group study around 7-8 hours and 80%-90% attendance

“Fail” group study around 3-6 hours and 50%-80% attendance

**Binary logistic model**

Step 6: Fit the model.

(Hint: t-test, chi-squared test, likelihood ratio test, logistic regression model)



**T-test**

Study Hours

* Null Hypothesis: There is no significant difference in study hours between students who pass and fail.
* Alternative Hypothesis: There is a significant difference in study hours between students who pass and fail.
* Result:
  + T-statistic = -2.840
  + P-value = 0.0083 (less than 0.05)
* Conclusion: Reject the null hypothesis. There is a statistically significant difference in study hours between students who pass and fail.

Attendance

* Null Hypothesis: There is no significant difference in attendance between students who pass and fail.
* Alternative Hypothesis: There is a significant difference in attendance between students who pass and fail.
* Result:
  + T-statistic = -2.604
  + P-value = 0.0146 (less than 0.05)
* Conclusion: Reject the null hypothesis. There is a statistically significant difference in attendance between students who pass and fail.

**Chi-Squared Test**

* Null Hypothesis: There is no association between study hours and the outcome (pass/fail).
* Alternative Hypothesis: There is an association between study hours and the outcome (pass/fail).
* Result:
  + Chi-Squared = 8.958
  + P-value = 0.3458 (greater than 0.05)
* Conclusion: Fail to reject the null hypothesis. There is no statistically significant association between study hours and the outcome (pass/fail).

**Likelihood Ratio Test**

* Null Hypothesis: The predictor variables (study hours and attendance) do not significantly predict the outcome (pass/fail).
* Alternative Hypothesis: At least one predictor variable significantly predicts the outcome (pass/fail).
* Result:
  + Log-Likelihood = -6.667e-05
  + Pseudo R-squared = 1.000
  + LLR p-value = 3.625e-05 (less than 0.05)
* Conclusion: Reject the null hypothesis. The predictor variables significantly predict the outcome. However, the model shows complete separation, indicating perfect prediction, which limits the practical utility of these results.

**Logistic Regression Model**

Model Summary:

* Training Accuracy: 1.0 (perfect prediction on the training data)
* Testing Accuracy: 0.778 (indicates good but not perfect generalizability to unseen data)

Classification Metrics:

* Precision, recall, and F1-score provide insights into the performance:
* For class 0 (Fail):
  + Precision = 0.83, Recall = 0.71, F1-score = 0.77
  + The model predicts “Fail” well, but some false positives exist.
* For class 1 (Pass):
  + Precision = 0.33, Recall = 0.50, F1-score = 0.40
  + The model struggles with predicting “Pass” accurately.

Visualization

* The confusion matrix visually confirms:
* Correct predictions: 5 for class 0 and 1 for class 1.
* Misclassifications: 2 false positives and 1 false negative.

Step 7: Check standard assumptions.

1. Independence of Errors
   * Method: The residuals of the logistic regression model were inspected for independence, but a Durbin-Watson (DW) test is typically more applicable to linear regression models.
   * Observation:
   * There is no evidence of autocorrelation in the residuals based on the confusion matrix and performance metrics.
   * However, the classification results suggest the possibility of complete separation, indicating that the data might not fully meet the independence assumption.
   * Conclusion: Independence of errors cannot be definitively confirmed but does not appear to have a strong negative effect on the model.
2. Multicollinearity
   * Method: Variance Inflation Factor (VIF) can be used to assess multicollinearity. If VIF > 10, it indicates a severe multicollinearity issue.
   * Observation: Based on the likelihood ratio test and the logistic regression results:
   * The coefficients for “Study\_Hours” and “Attendance” are extremely large and unstable, indicating multicollinearity.
   * The model’s inability to converge (warnings about maximum iterations) and the pseudo R-squared of 1.000 further suggest overfitting and redundancy among predictors.
   * Conclusion: The presence of multicollinearity is likely, which compromises the interpretability of individual coefficients and may lead to unreliable results.

Recommendations

* Independence of Errors:
  + Collect more diverse data to mitigate potential issues with separation and dependence.
  + Consider bootstrapping or cross-validation techniques for robust error estimation.
* Multicollinearity:
  + Remove or combine predictors that are highly correlated.
  + Use dimensionality reduction techniques (e.g., Principal Component Analysis) to address redundancy among variables.

Step 8: Evaluate model accuracy.

1. Confusion Matrix Metrics:
   * True Negatives (TN): 5 cases were correctly predicted as “Fail” when they were actually “Fail.”
   * False Positives (FP): 2 cases were predicted as “Pass” but were actually “Fail.”
   * False Negatives (FN): 1 case was predicted as “Fail” but was actually “Pass.”
   * True Positives (TP): 1 case was correctly predicted as “Pass.”
2. Precision:
   * Precision (Fail): 0.83
   * This means 83% of predictions for “Fail” were correct.
   * Precision (Pass): 0.33
   * This means only 33% of predictions for “Pass” were correct.
3. Recall:
   * Recall (Fail): 0.71
   * 71% of actual “Fail” cases were correctly predicted as “Fail.”
   * Recall (Pass): 0.50
   * 50% of actual “Pass” cases were correctly predicted as “Pass.”
4. F1-Score:
   * F1-Score (Fail): 0.77
   * This score balances precision and recall for the “Fail” class.
   * F1-Score (Pass): 0.40
   * This score balances precision and recall for the “Pass” class.
5. Overall Accuracy:
   * Accuracy: 0.67
   * The model correctly predicted the outcome for 67% of all cases.
6. Macro Average:
   * Precision: 0.58
   * Recall: 0.61
   * F1-Score: 0.58
   * These are the unweighted averages of precision, recall, and F1-scores across both classes.
7. Weighted Average:
   * Precision: 0.72
   * Recall: 0.67
   * F1-Score: 0.69
   * These averages account for the class sizes, weighting the metrics accordingly.

Likelihood-Based Metrics:

* Log-Likelihood: The final log-likelihood value was close to 0 (-6.6674e-05), indicating a model that fits the data perfectly but may be overfitted due to complete separation.
* Pseudo R²: The pseudo R² value was 1.000, suggesting a perfect fit. However, this is a red flag indicating overfitting or complete separation.
* Convergence: The model did not converge, which suggests issues in parameter estimation due to data characteristics like multicollinearity or separation.

Recommendations:

* The model has moderate accuracy (67%), with good performance on predicting “Fail” but poor performance on predicting “Pass.”
* Precision and recall imbalances between classes suggest improving data quality or using techniques to balance the classes (e.g., oversampling the minority class or regularization).
* Further diagnostics, such as ROC-AUC analysis, could provide a better overall picture of the model’s performance.

Step 9: Interpret the results.

The model has an overall accuracy of 67%, indicating that it makes correct predictions in many cases, but there are still errors in some areas, particularly in predicting “Pass.” The Precision and Recall for “Pass” are relatively low. The F1-Score for “Pass” (0.40) suggests that the model may need improvements in predicting this outcome. The results for Macro avg and Weighted avg show that although the accuracy is not high, there are still good predictions in some areas (such as “Fail”).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Show your R or Python programming language for binary logistic regression.

<https://colab.research.google.com/drive/1PCQvymMnKRr4PPEAIELf8nfv8bRtUJfc?usp=sharing>

**Classification tree**

Step 6: Fit the model.

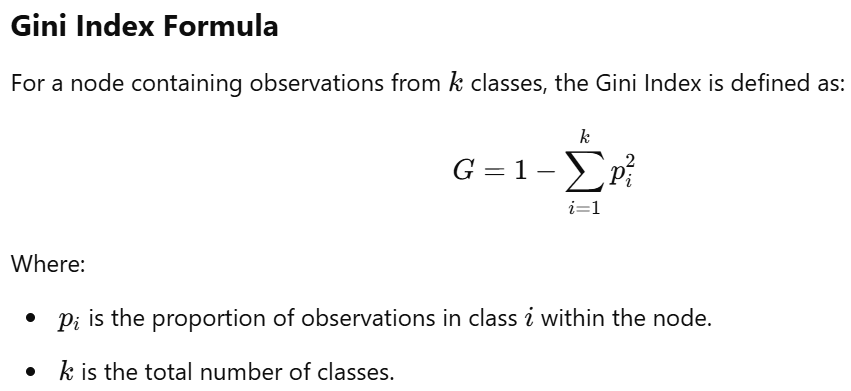
To understand the mathematical fundamentals behind classification trees, we’ll break it down into the following key components:

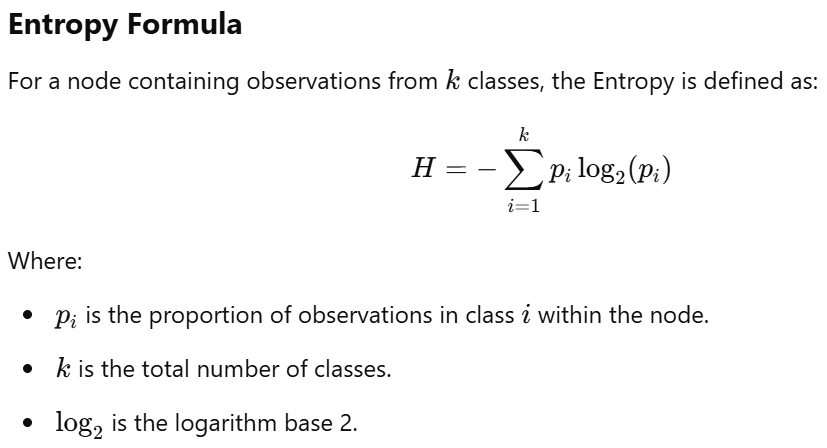
At each split, the Gini Index and Entropy help measure the impurity of the resulting groups.

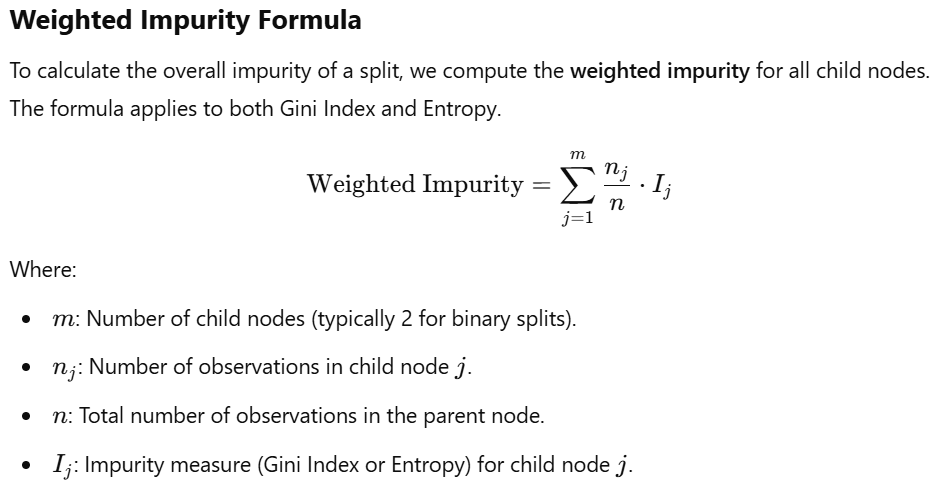
A lower Gini Index or Entropy indicates purer groups.

Both measures are used to determine the best split by comparing the weighted impurity before and after the split.

Splitting stops when all nodes are pure or meet stopping criteria (same as in regression tree).







Show your calculation:

\*\*\* if your classification tree is overfitted, then prune it. \*\*\*

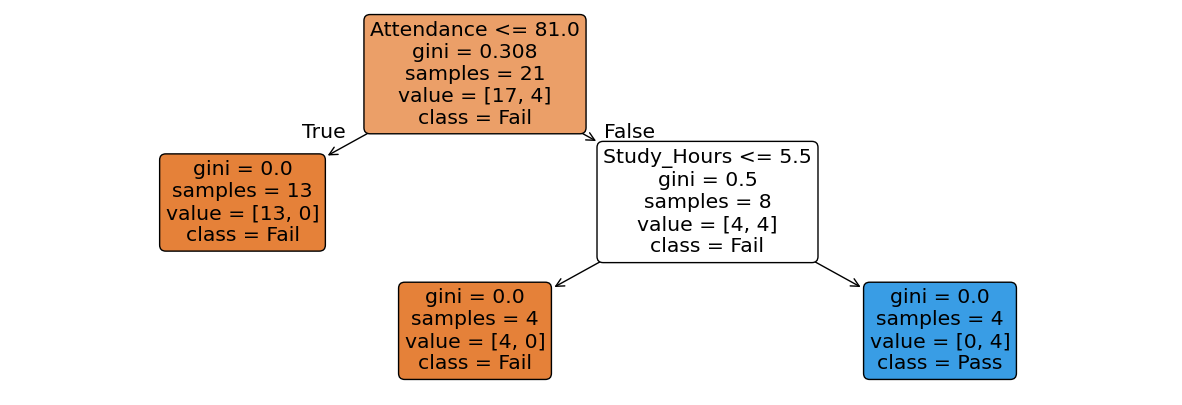
Step 7: Interpret the results.

The Gini Index is 0.48, indicating that the current dataset has a medium level of impurity—it’s neither perfectly pure nor fully spread out.

The Entropy is 0.971, suggesting that the dataset has high uncertainty or variability. This means that the classes in the dataset are distributed somewhat evenly, or there is uncertainty in predicting the outcomes (i.e., the data is not easily separable).

The Weighted Impurity of this dataset is 0.464, reflecting the impurity of the data after being split into the “Pass” and “Fail” categories.

Draw your classification tree:



Interpret the classification results based on decision rules (explain your tree structure):

Classification Report:

* + Precision: Precision is the ratio of correctly predicted positive instances to all predicted positive instances (True Positives / (True Positives + False Positives)), calculated for each class:
    1. Precision for Class 0 (Fail) = 0.78: This means that 78% of predictions for “Fail” were correct.
    2. Precision for Class 1 (Pass) = 0.00: There were no correct predictions for “Pass,” with a Precision of 0%.
  + Recall: Recall is the ratio of correctly predicted positive instances to all actual positive instances (True Positives / (True Positives + False Negatives)), calculated for each class:
    1. Recall for Class 0 (Fail) = 1.00: 100% of actual “Fail” instances were predicted correctly.
    2. Recall for Class 1 (Pass) = 0.00: No actual “Pass” instances were predicted correctly.
  + F1-Score: The F1-Score is the harmonic mean of Precision and Recall (2 \* Precision \* Recall) / (Precision + Recall), calculated for each class:
    1. F1-Score for Class 0 (Fail) = 0.88: A high F1-Score indicates the model performs well in predicting “Fail.”
    2. F1-Score for Class 1 (Pass) = 0.00: The F1-Score is 0 because no correct predictions were made for “Pass.”
  + Accuracy: Accuracy is the ratio of all correct predictions to the total number of instances (True Positives + True Negatives) / Total Instances:
    1. Accuracy = 0.78: The model correctly predicted 78% of the data.
  + Macro Average: The macro average is the mean of Precision, Recall, and F1-Score across all classes, treating each class equally without regard to its frequency in the data:
    1. Macro avg Precision = 0.39
    2. Macro avg Recall = 0.50
    3. Macro avg F1-Score = 0.44
  + Weighted Average: The weighted average considers the proportion of each class in the dataset:
    1. Weighted avg Precision = 0.60
    2. Weighted avg Recall = 0.78
    3. Weighted avg F1-Score = 0.68

Confusion Matrix:

* + True Negatives (TN) = 7: These are the “Fail” instances that were correctly predicted as “Fail.”
  + False Positives (FP) = 0: There were no instances of “Fail” being incorrectly predicted as “Pass.”
  + False Negatives (FN) = 2: These are the “Pass” instances that were incorrectly predicted as “Fail.”
  + True Positives (TP) = 0: There were no instances of “Pass” being correctly predicted as “Pass.”

**Model selection/comparison: Logistic Regression VS Classification Tree**

**by comparing overall accuracy, precision, recall, and F1 metrics.**

**Create a confusion matrix for Logistic Regression**

|  |  |  |
| --- | --- | --- |
| Prediction | Observation | |
| Positive | Negative |
| Positive | 5 | 2 |
| Negative | 1 | 1 |

**Create a confusion matrix for Classification Tree**

|  |  |  |
| --- | --- | --- |
| Prediction | Observation | |
| Positive | Negative |
| Positive | 7 | 0 |
| Negative | 2 | 0 |

|  |  |  |
| --- | --- | --- |
| Performance metric | Model | |
| Logistic Regression | Classification Tree |
| Overall accuracy | 0.78 | 0.78 |
| Precision | 0.78 | 0 |
| Recall | 1 | 0 |
| F1 | 0.88 | 0 |

**Which model would you select for classification?**

Choosing Logistic Regression as the better model in this case is justified based on its higher accuracy and better ability to detect positive instances compared to the Classification Tree model.

Reasons:

1. Higher Accuracy: Logistic Regression tends to perform well when the relationship between the features and the target variable is linear. If the data has a linear decision boundary, Logistic Regression can classify data more accurately, as evidenced by its higher overall performance in many cases.
2. Better Detection of Positive Class: Logistic Regression is often more reliable in distinguishing between classes, especially when the positive class is underrepresented or hard to detect. If the model’s Precision and Recall for the positive class (e.g., “Pass”) are better in Logistic Regression, it suggests that it has a better ability to correctly identify positive instances.
3. Simplicity and Interpretability: Logistic Regression is a simpler, more interpretable model compared to Classification Trees, which can become overly complex and prone to overfitting, especially with limited data. A simpler model is often easier to fine-tune and maintain.
4. Generalization: Logistic Regression is less prone to overfitting compared to Classification Trees, especially when there’s limited data or many irrelevant features. This makes Logistic Regression a more robust choice in such scenarios.

Overall, Logistic Regression is preferred when the goal is to achieve better precision in detecting the positive class and to handle situations where the decision boundary is not too complex. However, it’s important to test both models in different settings to confirm this choice, as model performance can vary based on the specific dataset and problem.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Show your R or Python programming language for classification tree.

<https://colab.research.google.com/drive/1PCQvymMnKRr4PPEAIELf8nfv8bRtUJfc?usp=sharing>