# 07 Decision Trees (Classification)

Created Date: 2022-10-25

> Metadata 🗃️
>
> - Title: Decision Trees (Classification)
> - Author: Andrew Jones
> - Reference: Data Science Infinity

> Links & Tags 🔗
>
> - Index: Course Note Index
> - Atomic Tag: #datascience
> - Subatomic Tags: #machinelearning #decisiontrees #classification

---

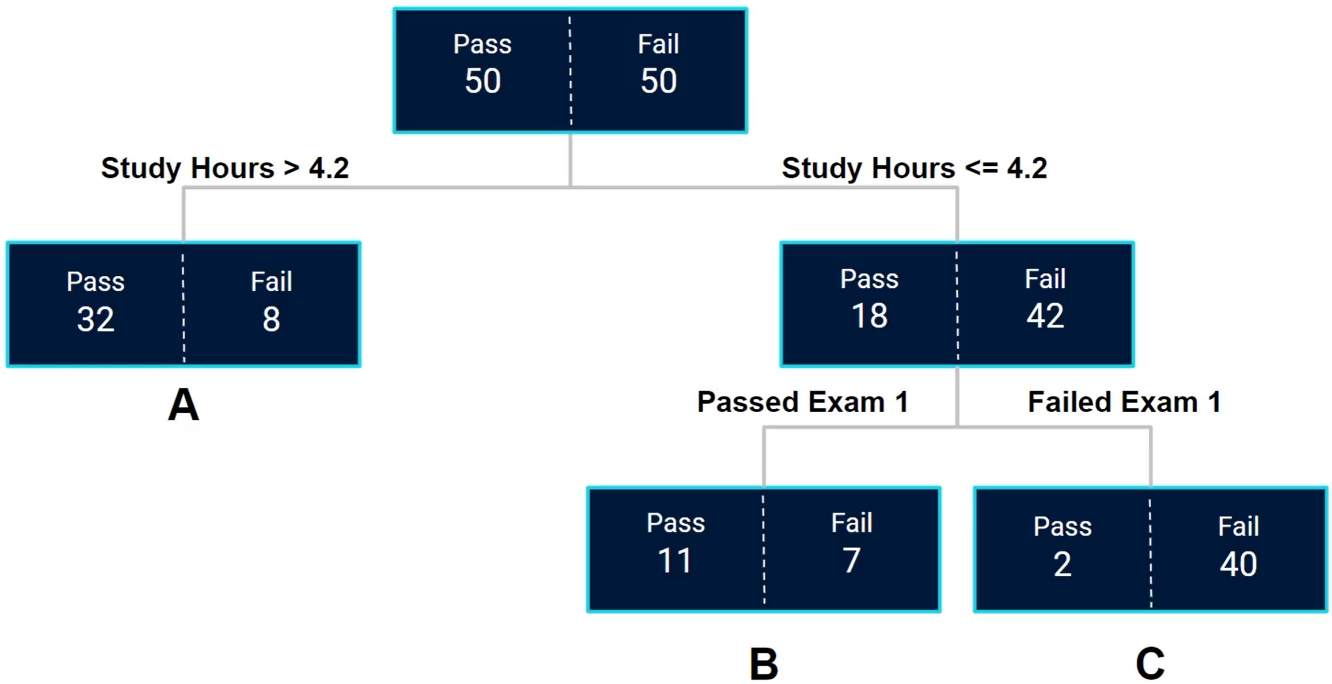
## High-Level Overview

Jupyter Notebook: Basic Classification Tree Template

> Decision Tree is a model that splits the data into distinct buckets using the input variables, with split decisions being based on how well each potential split explains differences in the output variable.

- Classification Tree is commonly used for classification modeling
- The output variables are binary (true or false)
  - Did the event take place or not
- All of the data starts at the root node with a classification threshold (typically 50/50)
- The model will assess each possible split and pick the split with the biggest influence on the output variable to a left and right leaf node

- The model will then repeat this process for each node, spitting again (if appropriate)
- Within a node, we can calculate the success rate by dividing $\frac{Pass\ Outcomes\ in\ Node}{Total\ Outcomes\ in\ Node}$
- All nodes with a pass rate greater than the threshold are classified as pass nodes




# Advanced Theory

Jupyter Notebook: Advanced Classification Tree Template

## Splitting Criteria

> Gini Impurity is a metric used to compare potential splits in classification trees.

- The Gini Impurity shows the probability of misclassifying an observation
- A lower Gini score indicates a lower likelihood of classification
- The model looks to find the split point with the lowest weighted Gini score

- The weighted averages of Gini scores for a split (each node) are calculated as the Total Gini score
- If the split variable is numeric, the model will take the middle point between the two values with the lowest Total Gini score

$$Gini = 1 - \sum_{i=1}^{n}(p_{i)^2}\ Gini = 1 - (p\,of\,Passing)^2 - (p\,of\,Failing)^2$$
$$Total\ Gini = \sum(Node\ Gini)(Percentage\ of\ Observations\ in\ Node)$$

- $n$ Number of classes present in the node
- $p$ Probability

## Stopping Criteria

- The model will continue to split until one of the following criteria are met;
  - There is only one data point in each leaf node
  - It cannot find a split point that will reduce the Gini score
  - It's told to stop
- Models that are allowed to go too deep are often overfitted
- Setting a stopping criteria is achieved by setting a maximum depth
- This doesn't guarantee n leaf nodes, the model will split before the maximum depth if it meets one of the two conditions above
- Alternatively, we can assign a minimum number of data points required to split

## Evaluating Classification Accuracy

- $Classification\ Accuracy = \frac{n\ Outcomes\ Classified\ Correctly}{Total\ Outcomes}$

- $n$ Number of Outcomes Classified Correctly

- - Type I Error: False Positive - Type II Error: False Negative - Diagonal (True Positive & True Negative): Outcomes Correctly Classified

| | | Predicted Class | |
|---|---|---|---|
| | | Pass | Fail |
| Actual Class | Pass | True Positive | False Negative |
| | Fail | False Positive | True Negative |

## Advanced Evaluation Techniques

- When there is a large bias towards one of the classes we have an imbalanced data set
- Advanced techniques help evaluate models when we have imbalanced data
    - Precision evaluates how many observations were predicted as positive who were actually positive
    - Recall (Sensitivity) evaluates how many observations were predicted as positive who were actually positive (also referred to as the True Positive Rate)

- False Positive Rate evaluates how many observations were predicted as positive who were actually negative
- F1-Score evaluates the harmonic mean of Precision and Recall
  - A good F1-Score comes when there is a balance between Precision & Recall, rather than a disparity between them
- Precision & Recall can not be optimized together, sometimes it makes sense to adapt a model to optimize one of these metrics
  - As an example, in a disease diagnoses model this would evaluate observations that were not predicted to have a disease who actually have the disease
  - In this example, it may make sense to optimize Recall while still being cognizant that we don't want to misdiagnose people as positive when they are in fact negative

| Precision | Recall | Meaning |
| --- | --- | --- |
| High | High | The model is differentiating between classes well |
| High | Low | The model is struggling to detect the class, but when it does it is very trustworthy |
| Low | High | The model is identifying most of the class, but is also incorrectly including a high number of data points from another class |
| Low | Low | The model is struggling to differentiate between classes |

## Advanced Evaluation Metrics

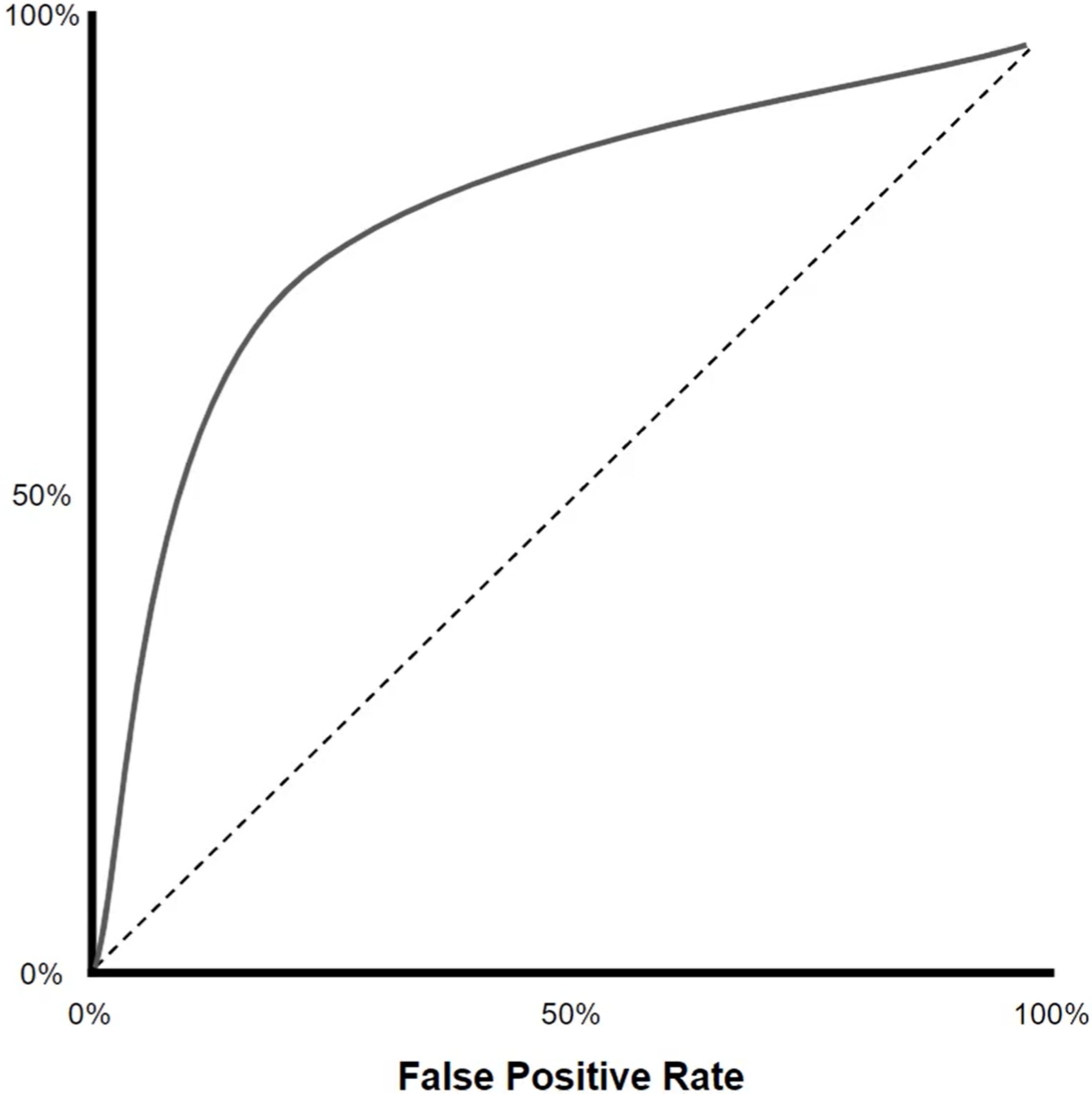
- $Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$
- $True\ Positive\ Rate\ (Recall\ Sensitivity) = \frac{True\ Positive}{True\ Positive + False\ Negative}$
- $False\ Positive\ Rate = \frac{False\ Positive}{False\ Positive + TrueNegative}$

- $F1\ Score = \frac{2 * (Recall * Precision)}{Recall + Precision}$

## Changing the Classification Threshold

- The default classification threshold is 50%
- A low threshold will classify more observations as positive, while a high threshold will classify more as negative
- Changing the threshold will impact the Precision and Recall evaluation metrics
- We can visualize the impact of changing the threshold on the TPR and FPR metrics using an ROC curve
  - The dashed lines would represent observations that had equal TPR and FPR results
  - The solid line represents the actual results of the TPR and FPR metrics calculated for varying thresholds
  - Observations to the left of the dashed line are good, as they infer the model has proportionately lower incorrect classifications (false positives)
  - We can optimize the threshold by picking a threshold that results in the furthest point from the dashed line
  - ROC curves can also be used to compare the accuracy of different classification models by calculating the area under the curve (AUC)
    - A larger AUC is considered to be a better performing model
  - The ROC curve can be misleading when we have an imbalanced data set
    - In this case, we aim to optimize the F1 Score

> ROC (Receiver Operator Characteristic) Curve visualizes the trade-off between the *True Positive Rate* and the *False Positive Rate* across varying classification thresholds.


100%
50%
0%
0%
50%
100%
False Positive Rate