Sane's STATS Academy of Statistics

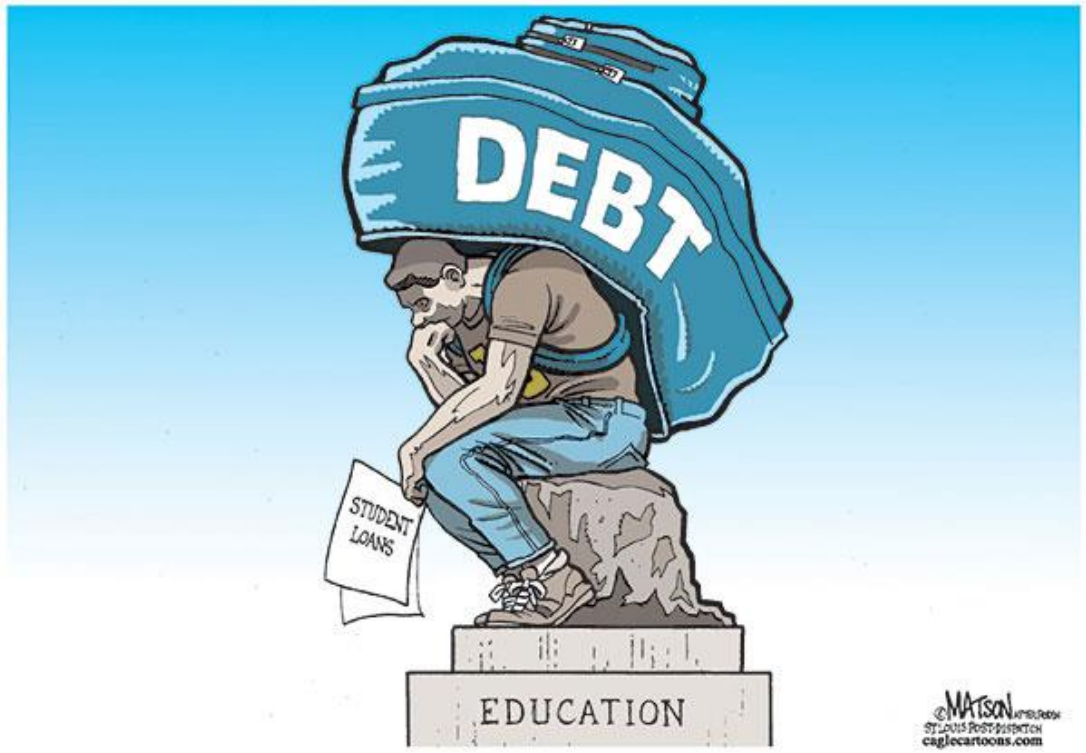
Classification Trees

Decision Trees

- Decision Trees create a set of binary splits on the predictor variables
- These splits are used to classify new observations into one of the two groups
- There are two categories of decision trees:
  - Classification Tree for Categorical Response Variable
  - Regression Tree for Numerical Response Variable

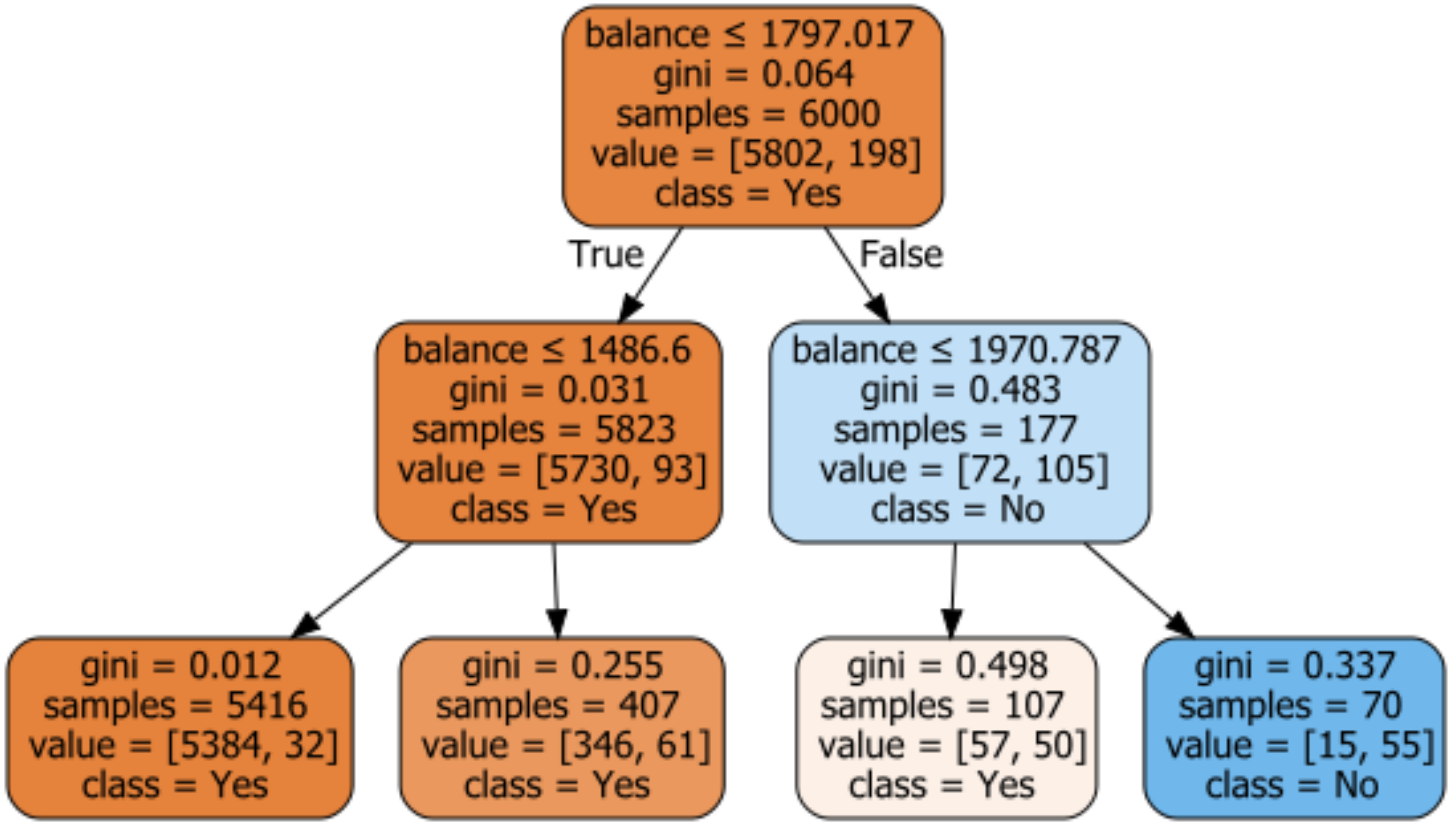
Example 1: Loan Defaulter

- Consider the data on loan defaulters.
- For analysis, we have taken non-defaulter + defaulters
- Data has been recorded in the file Default.csv

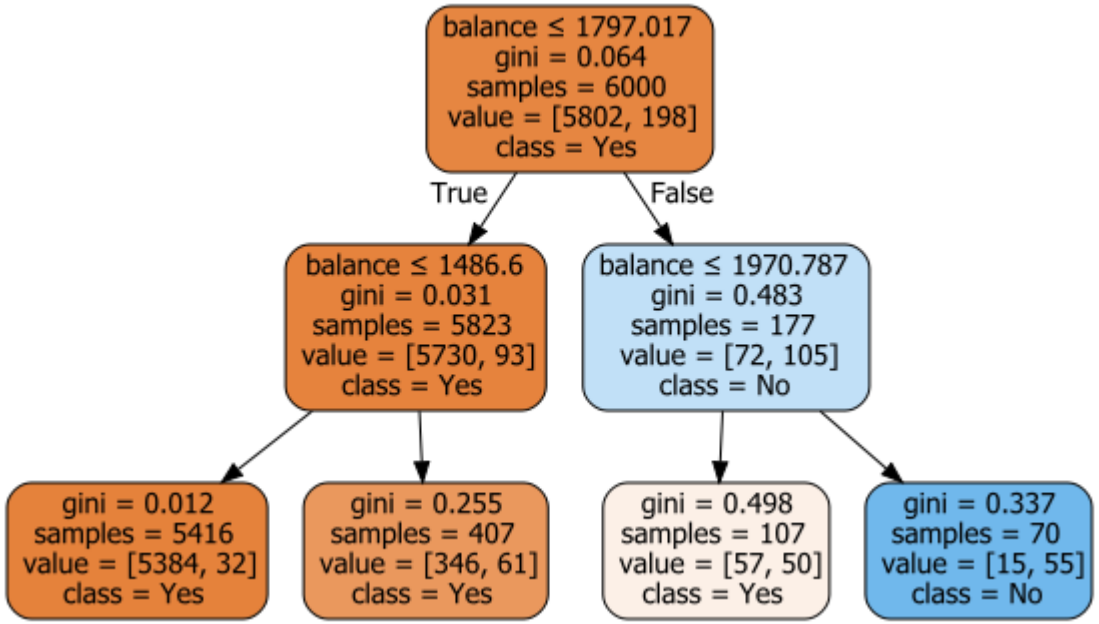


Sane's STATS Academy of Statistics

Classification Tree



What did we gain from classification?



- For the whole training data, we had n(Yes)=198 and n(Not Bought)=5802 with proportions as 0.033 and 0.967 respectively
- By splitting on balance cut-off as 1797.017, we got two partitions with proportions as [P(Yes)=0.016, P(No)=0.98] and [P(Yes)=0.41, P(No)=0.59] respectively
- Hence we gained a **purity** or **homogeneity** in one case and lost it in other case
- The splits get proceeded for increasing purity

Sane's STATS Academy of Statistics

Types of Decision Tree Algorithms

- There are many specific decision-tree algorithms. Notable ones include:
  - ID3 (Iterative Dichotomiser 3)
  - C4.5 (successor of ID3)
  - CART (Classification And Regression Tree)
  - CHAID (CHi-squared Automatic Interaction Detector). Performs multi-level splits when computing classification trees.
  - MARS: extends decision trees to handle numerical data better.
  - Conditional Inference Trees: Statistics-based approach that uses non-parametric tests as splitting criteria, corrected for multiple testing to avoid overfitting. This approach results in unbiased predictor selection and does not require pruning.
- We will be covering CART

Classical Decision Trees Algorithm

- Response : Categorical outcome variable
- Predictors : Categorical and/or Continuous Variables.
- Steps are as follows:
  - Predictor variable gets chosen in such a way that it best splits the data into two groups with maximized purity.
    - If the predictor is continuous, then cut-point is chosen for maximizing the purity
    - If the predictor is categorical, then the categories are combined together to obtain two groups with maximized purity
  - The data is separated into two groups and the process for each subgroup is continued
  - Steps 1 and 2 are repeated until a subgroup is obtained containing fewer number of observations than a minimum number specified or the algorithm may terminate if further splitting doesn't increase purity beyond a specific threshold

Classical Decision Trees Algorithm

- The subgroups in the lowest branch of the tree are called terminal nodes or leaf nodes.
- Each leaf node is classified as one single category of the outcome
- For classifying any observation in the validation / test data set, that observation is traversed through the branches of tree and leaf node at which the traversal stops is predicted as the outcome of the observation.

Gini's Impurity Index

- Gini impurity can be computed by summing the probability of each item being chosen times the probability of a mistake in categorizing that item.
- It reaches its minimum (zero) when all cases in the node fall into a single target category.
- It is used by CART Algorithm, by package rpart by default

$$I_G(f) = \sum_{i=1}^{m} f_i(1 - f_i) = \sum_{i=1}^{m} (f_i - f_i^2) = \sum_{i=1}^{m} f_i - \sum_{i=1}^{m} f_i^2 = 1 - \sum_{i=1}^{m} f_i^2 = \sum_{i \neq k} f_i f_k$$

Package tree from scikit-learn

- We need to instantiate the DecisionTreeClassifier first
- Then, call fit function on training data
- Finally, we predict on test data

```
clf = tree.DecisionTreeClassifier(max_depth=2)
clf2 = clf.fit(X_train, y_train)
```

```
y_pred = clf2.predict(X_test)
```

DecisionTreeClassifier( )

Syntax :

sklearn.tree.DecisionTreeClassifier(criterion='gini', max_depth=None, …)

Where

criterion : split criterion; "gini" or "entropy"

max_depth : The maximum depth of the tree

Program and Output

```
# Create training and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.4,
                                                    random_state=42)

clf = tree.DecisionTreeClassifier(max_depth=2)
clf2 = clf.fit(X_train, y_train)
```

```
y_pred = clf2.predict(X_test)

print(confusion_matrix(y_test, y_pred))
print(classification_report(y_test, y_pred))
```

```
In [111]: y_pred = clf2.predict(X_test)
     ...:
     ...: print(confusion_matrix(y_test, y_pred))
     ...: print(classification_report(y_test, y_pred))
[[3850   15]
 [ 100   35]]
             precision    recall  f1-score   support

          0       0.97      1.00      0.99      3865
          1       0.70      0.26      0.38       135

avg / total       0.97      0.97      0.96      4000
```

Sane's STATS Academy of Statistics

Sane's STATS Academy of Statistics

Questions?