Advanced Regression, Tree-Based Models

Thursday, September 26, 2024 10:48 AM

CART Introduction

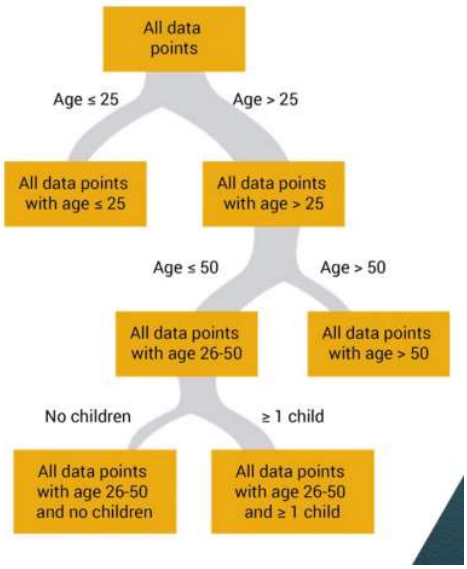
CART: Classification and Regression Trees. Tree models can be used for classification and regression models. When a tree model is used for decision making, it is known as a Decision Tree. Types of tree models include:

- Classification Trees: used to make categorical predictions, where the most common classification for each node is the prediction
- Logistic Regression Trees: used to make numerical predictions, where the fraction of the node's data points with the "true" response is the percentage prediction
- Decision Trees: Each leaf represents a decision to make.

Suppose we want to make a Single Regression Model to show the impact of a marketing email on recipient spending. The predictors include:

- Demographics: age, sex, number of children, income, etc.
- Purchasing factors: average amount spent per month on website
- Binary factors: if the email was received

The data can be split into multiple regression models based on how differing the different groups of people behave with regards to marketing emails, and then more targeted predictions can be made towards each group. For example, if we want to predict the impact of the marketing email on a 37 year old with two kids:



This approach has the added benefit of informing us which of the models are performing well and which may need more work. If the model for the group Age > 50 has low evaluation metrics, we know that additional work to make the model perform better is needed.

However, the downside is that this approach requires making a regression model at every leaf of the tree, which is time consuming and computationally expensive. In practice, rather than doing a full regression model at every leaf, we instead make the simplest regression model for every leaf: $y = a_0$

- $a_0 = \frac{\sum_i in\ node\ y_i}{\#\ of\ data\ points\ in\ node} = average\ response\ in\ node$

Branching

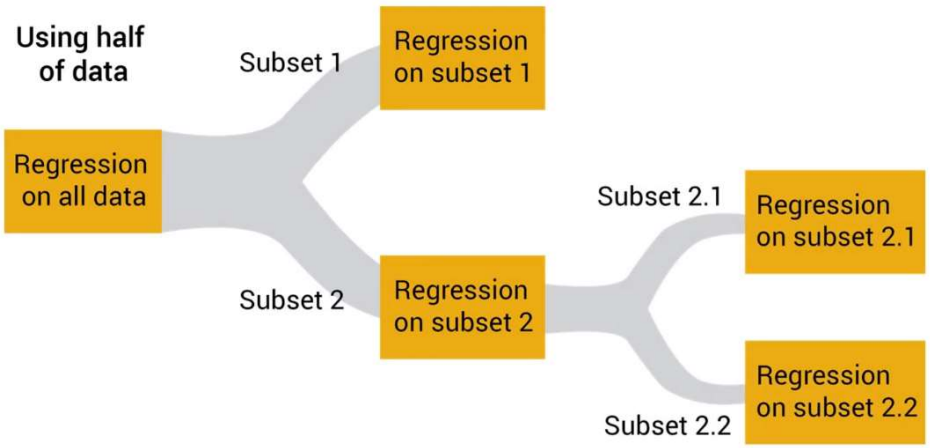
Which factor(s) should be considered when deciding the branches? How should the data be split into branches? There isn't a good algorithm than can be used to make this decision. Common practice is to split on one factor at a time and observe how the models perform.

Branching method using half of the data:

1. Make a regression model on half of the data
2. Split the data into two equal subsets and make regression models on both subsets ("branching")
3. For each leaf:
    a. Calculate the variance
    b. Split on each factor and find the biggest variance decrease.
    c. Make the split where there is the biggest variance decrease (if it is more than the threshold).
    d. Repeat until no split decreases the variance more than the threshold.
4. Use the other half of the data to evaluate the models and remove unneeded branches ("pruning"):
    a. For each branching point, calculate the estimation error with and without branching.
    b. If branching increases the error, remove that branching.

When considering if a branch ought to be pruned or not:

- Check if the benefit of the branching is very low
- Check if one side of the branch has too few data points (rule of thumb: each leaf contains at least 5% of the original data).



This is just one method for branching, but in general: use a metric related to the model's quality, find the "best factor" to branch with, and then check if the branching really improved the model (and prune if not).

Random Forests

Random Forests: a machine learning algorithm that combines multiple trees based on various subsets of a dataset (each with their own strengths and weaknesses) to produce a single, averaged result. This averaged approach may have more strengths and perform better than a single tree by itself. Since each tree in the forest has slightly different data in it and since we do this procedure many times over, we generally end up with a large (usually 500-1000) number of trees.

- Regression Trees Random Forest: use the average predicted response
- Classification Trees Random Forest: use the most common predicted response

Randomness is introduced in this process at a few different points:

1. Via the Bootstrapping process, which is a resampling procedure that uses data from one sample to generate a sampling distribution by repeatedly taking random samples from the known sample with replacement.
2. During the branching process, where we randomly choose a small number of factors (set X) and select the best factor within that set to branch on.
    a. A common number of factors to use for set X is $1 + log(n)$, where n is the number of factors in the data set.
3. Trees are not pruned in the random forest

Random Forest benefits and drawbacks:

- Better overall estimates than a single tree
- The averages between trees somewhat neutralizes the effects of overfitting
- The output is harder to explain or interpret
- Doesn't provide a specific regression or classification model from the data

Thus Random Forests are good as a default model or black-box predictor, but not good for detailed insight.

Explainability / Interpretability

Linear Regression models are generally easy to explain to a layperson, but trees are more difficult. This difficulty increases when the model is actually a random forest. The forest does provide relative branching importance of each variable, but not how or why - so it isn't precise for explainability.

Explainability helps us toward an understanding of why and how the model works. This helps decision-makers who don't have the technical modeling background understand and appropriately use the model's results to make decisions. Sometimes, explainability is a legal requirement. On the other hand, less-explainable models sometimes give better results and so we are able to identify and model more complex patterns.

It's important to keep in mind this trade off when suggesting a model:

- Less explainable models can give more value by fitting to more complex patterns
- More explainable models can be more likely to be adopted since they're easier for decision-makers to believe in and are sometimes legally required (in certain industries)

Logistic Regression

In cases where we need to estimate a probability, Linear Regression will not work (since the output of this model is continuous and can generate output outside of the required [0,1] range of a percentage). Recall the Standard Linear Regression equation: $y = a_0 + a_1x_1 + a_2x_2 + \cdots + a_jx_j$

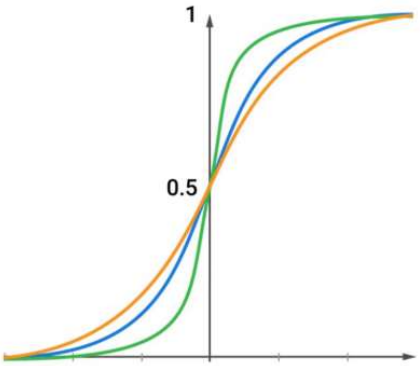
Logistic Regression takes the standard linear regression equation and converts it to an exponential, so the output is strictly between [0,1]. Logistic Regression is sometimes referred to as the logit function, which is the natural logarithm of the probability of an event occurring ($p/1-p$), also knowing as log-odds. Let p be the probability of the event, then:

$$\log\frac{p}{1-p} = a_0 + a_1x_1 + a_2x_2 + \cdots + a_jx_j$$

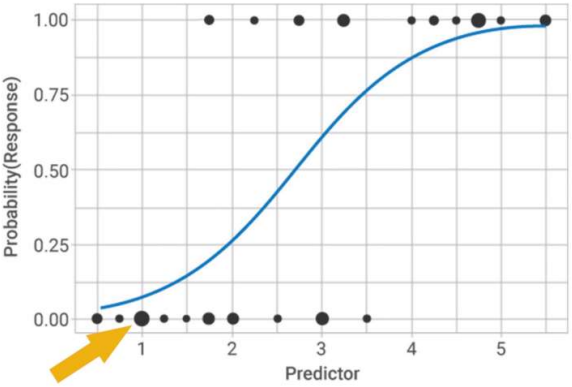
$$p = {}^{1}/_{1+e^{-(a_0+a_1x_1+a_2x_2+\cdots+a_jx_j)}}$$

- If $(a_0+a_1x_1 + a_2x_2 + \cdots + a_jx_j) = -\infty$, p=0
- If $(a_0+a_1x_1 + a_2x_2 + \cdots + a_jx_j) = \infty$, p=1

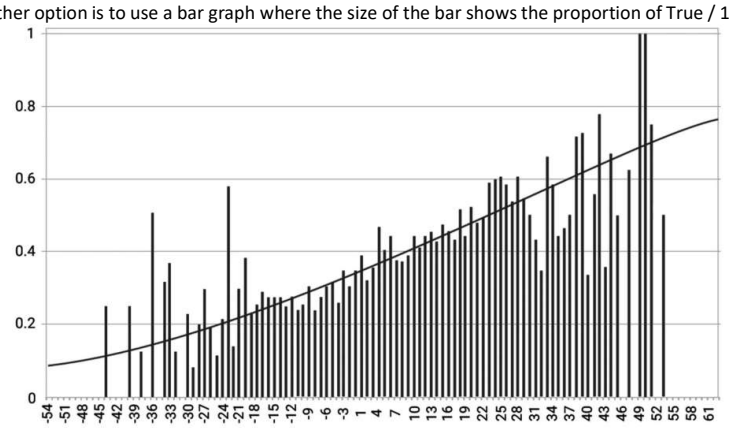
The values of the coefficients can change how steep the curve of the Logistic Regression is, but the curve will always stay within the bounds of [0,1].



Graphically, since the data points may sit on top of one another, it may be difficult to interpret with a scatterplot. One method to handle this could be to increase the size of the points in relation to how many data points there are:



Another option is to use a bar graph where the size of the bar shows the proportion of True / 1 data points and conveniently mirrors the shape of the regression graph:



Logistic Regression vs Linear Regression. Most things that can be done with Linear regression can also be done with Logistic regression in a similar way.

- Similarities:
  - Use transformations of input data
  - Consider interaction terms
  - Use variable selection methods
  - Build logistic regression trees
  - Build random logistic regression forests
- Differences:
  - Logistic Regression takes longer to compute due to there being no closed-form solution, or a unique solution to a single equation
  - Logistic Regression and Linear Regression have different measures of model quality:
    - Linear: R-Squared Value (the fraction of the variance explained by the model)
    - Logistic: Pseudo R-Squared value (good for comparing models, but does not explain the fraction of the variance explained by the model)
  - Logistic Regression is sometimes used for classification, where it takes some threshold and answers "yes" if probability *p* is greater than the threshold

Logistic Regression for classification can be visually evaluated with what is known as a Receiver Operating Characteristic (ROC) Curve, which graphs the sensitivity versus the specificity of the output - a quick estimate of the model quality. The Area Under the Curve (AUC) is the probability that the model estimates a random "yes" point higher than a random "no" point. An AUC of 0.5 is just random guessing. The ROC curve does not differentiate between the cost of false negatives and false positives, though.

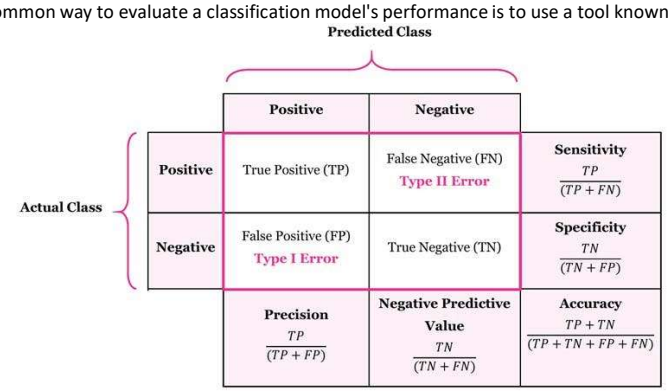
Confusion Matrices

Common classification-type models:

- Support Vector Machines
- k-Nearest-Neighbors

- Logistic Regression

A common way to evaluate a classification model's performance is to use a tool known as a Confusion Matrix:



- Sensitivity: the number of correct positive predictions divided by the total number of positives. Also known as Recall or True Positive Rate.
- Specificity: the number of correct negative predictions divided by the total number of negatives. Also known as the True Negative Rate.

Situationally-Driven Comparison

Let's say we have a spam detection model with the below confusion matrix:

	REAL	SPAM
REAL	490	10
SPAM	100	400

In addition to the metrics we can calculate from the matrix, we also need to consider the cost of lost productivity. Let's say the cost of lost productivity is calculated as the below:

- Correct classification (TP, TN)s: $0.00
- Cost of reading spam (FP): $0.04
- Cost of missing a real message (FN): $1.00

If we have 1000 emails and we think 50% of that is spam, then the total expected cost per email is $0.14: $E[cost] = (490 * 0) + (10 * 1) + (100 * 0.04) + (400 * 0) = \$14.00$

This is a useful metric for comparing, since we can design a model that more strictly filters out spam but at the cost of also filtering out more real emails. The model below accurately classifies more emails, but the cost of lost productivity increases to $0.52.

	REAL	SPAM
REAL	450	50
SPAM	50	450

Always consider the tradeoff between the different types of errors.

Advanced Topics in Regression

This section briefly covers some topics that may be covered in later classes in the program

Poisson Regression: used when the response follows a Poisson Distribution, a discrete distribution used to model the occurrences of an event during a fixed interval, where the occurrences in disjoint intervals are independent. The interval is usually time, but it can be some other unit of measurement.

Regression Splines: functions of polynomials that connect to each other at a "knot". Fits different functions to different parts of the data set with smooth connections between the parts.

Bayesian Regression: starts with data and an estimate of how the regression coefficients and the random error is distributed. Then Bayes theorem can be used to update the estimate. Most helpful when there isn't much data to use.

k-Nearest-Neighbors Regression: no estimate of prediction function, instead we plot all of the data and predict a response for a new data point (average response of the k closest data points).