Machine Learning Versus Statistics

In the context of predictive modeling, what is the difference

between machine learning and statistics? There is not a bright line

dividing the two disciplines. Machine learning tends to be focused

more on developing efficient algorithms that scale to large data in

order to optimize the predictive model. Statistics generally pays

more attention to the probabilistic theory and underlying structure

of the model. Bagging, and the random forest (see “Bagging and

the Random Forest” on page 259), grew up firmly in the statistics

camp. Boosting (see “Boosting” on page 270), on the other hand,

has been developed in both disciplines but receives more attention

on the machine learning side of the divide. Regardless of the history,

the promise of boosting ensures that it will thrive as a technique

in both statistics and machine learning. **B**

While the output of KNN for classification is typically a binary

decision, such as default or paid off in the loan data, KNN routines

usually offer the opportunity to output a probability (propensity)

between 0 and 1. The probability is based on the fraction of one

class in the *K* nearest neighbors. In the preceding example, this

probability of default would have been estimated at 9

20 , or 0.45.

Using a probability score lets you use classification rules other than

simple majority votes (probability of 0.5). This is especially important

in problems with imbalanced classes; see “Strategies for Imbalanced

Data” on page 230. For example, if the goal is to identify

members of a rare class, the cutoff would typically be set below

50%. One common approach is to set the cutoff at the probability

of the rare event **B**

Other Distance Metrics

There are numerous other metrics for measuring distance between

vectors. For numeric data, *Mahalanobis distance* is attractive since

it accounts for the correlation between two variables. This is useful

since if two variables are highly correlated, Mahalanobis will essentially

treat these as a single variable in terms of distance. Euclidean

and Manhattan distance do not account for the correlation, effectively

placing greater weight on the attribute that underlies those

features. Mahalanobis distance is the Euclidean distance between

the principal components (see “Principal Components Analysis”

on page 284). The downside of using Mahalanobis distance is

increased computational effort and complexity; it is computed

using the *covariance matrix* (see “Covariance Matrix” on page 202). **B**

In linear and logistic regression, one hot encoding causes problems

with multicollinearity; see “Multicollinearity” on page 172. In such

cases, one dummy is omitted (its value can be inferred from the

other values). This is not an issue with KNN and other methods

discussed in this book. **B**

*Normalization* in this statistical context is not to be confused with

*database normalization*, which is the removal of redundant data

and the verification of data dependencies **O**

Using the *z*-score is just one way to rescale variables. Instead of the

mean, a more robust estimate of location could be used, such as the

median. Likewise, a different estimate of scale such as the interquartile

range could be used instead of the standard deviation.

Sometimes, variables are “squashed” into the 0–1 range. It’s also

important to realize that scaling each variable to have unit variance

is somewhat arbitrary. This implies that each variable is thought to

have the same importance in predictive power. If you have subjective

knowledge that some variables are more important than others,

then these could be scaled up. For example, with the loan data,

it is reasonable to expect that the payment-to-income ratio is very

important. **G**

Normalization (standardization) does not change the distributional

shape of the data; it does not make it normally shaped if it was not

already normally shaped (see “Normal Distribution” on page 69).**B**

Bias-Variance Trade-off

The tension between oversmoothing and overfitting is an instance

of the *bias-variance trade-off*, a ubiquitous problem in statistical

model fitting. Variance refers to the modeling error that occurs

because of the choice of training data; that is, if you were to choose

a different set of training data, the resulting model would be different.

Bias refers to the modeling error that occurs because you have

not properly identified the underlying real-world scenario; this

error would not disappear if you simply added more training data.

When a flexible model is overfit, the variance increases. You can

reduce this by using a simpler model, but the bias may increase due

to the loss of flexibility in modeling the real underlying situation. A

general approach to handling this trade-off is through *crossvalidation*.

See “Cross-Validation” on page 155 for more details. **B**

You can think of this staged use of KNN as a form of ensemble

learning, in which multiple predictive modeling methods are used

in conjunction with one another. It can also be considered as a

form of feature engineering in which the aim is to derive features

(predictor variables) that have predictive power. Often this involves

some manual review of the data; KNN gives a fairly automatic way

to do this. **B**

Decision Trees in Operations Research

The term *decision trees* has a different (and older) meaning in decision

science and operations research, where it refers to a human

decision analysis process. In this meaning, decision points, possible

outcomes, and their estimated probabilities are laid out in a

branching diagram, and the decision path with the maximum

expected value is chosen. **O**

Gini Coefficient

Gini impurity is not to be confused with the *Gini coefficient*. They

represent similar concepts, but the Gini coefficient is limited to the

binary classification problem and is related to the AUC metric (see

“AUC” on page 226). **O**