Extract from *Data Mining for Business Anlaytics*, Shmueli, Bruce & Patel, Wiley, 2016

PANDORA

Pandora is an Internet music radio service that allows users to build customized "stations" that play music similar to a song or artist that they have specified. Pandora uses a k-NN style clustering/classification process called the Music Genome Project to locate new songs or artists that are close to the user-specified song or artist.

Pandora was the brainchild of Tim Westergren, who worked as a musician and a nanny when he graduated from Stanford in the 1980s. Together with Nolan Gasser, who was studying medieval music, he developed a "matching engine" by entering data about a song's characteristics into a spreadsheet. The first result was surprising—a Beatles song matched to a Bee Gees song, but they built a company around the concept. The early days were hard—Westergren racked up over $300,000 in personal debt, maxed out 11 credit cards, and ended up in the hospital once due to stress-induced heart palpitations. A venture capitalist finally invested funds in 2004 to rescue the firm, and as of 2013, it is listed on the NY Stock Exchange.

In simplified terms, the process works roughly as follows for songs:

1. Pandora has established hundreds of variables on which a song can be measured on a scale from 0–5. Four such variables from the beginning of the list are
   - Acid Rock Qualities
   - Accordion Playing
   - Acousti-Lectric Sonority
   - Acousti-Synthetic Sonority
2. Pandora pays musicians to analyze tens of thousands of songs, and rate each song on each of these attributes. Each song will then be represented by a row vector of values between 0 and 5, for example, for Led Zeppelin's Kashmir:

   Kashmir 4 0 3 3 … (high on acid rock attributes, no accordion, etc.)

   This step represents a costly investment, and lies at the heart of Pandora's value because these variables have been tested and selected because they accurately reflect the essence of a song, and provide a basis for defining highly individualized preferences.
3. The online user specifies a song that s/he likes (the song must be in Pandora's database).
4. Pandora then calculates the statistical distance[1] between the user's song, and the songs in its database. It selects a song that is close to the user-specified song and plays it.
5. The user then has the option of saying "I like this song," "I don't like this song," or saying nothing.
6. If "like" is chosen, the original song, plus the new song are merged into a 2-song cluster[2] that is represented by a single vector, comprised of means of the variables in the original two song vectors.

7. If "dislike" is chosen, the vector of the song that is not liked is stored for future reference. (If the user does not express an opinion about the song, in our simplified example here, the new song is not used for further comparisons.)
8. Pandora looks in its database for a new song, one whose statistical distance is close to the "like" song cluster,[3] and not too close to the "dislike" song. Depending on the user's reaction, this new song might be added to the "like" cluster or "dislike" cluster.

Over time, Pandora develops the ability to deliver songs that match a particular taste of a particular user. A single user might build up multiple stations around different song clusters. Clearly, this is a less limiting approach than selecting music in terms of which "genre" it belongs to.

While the process described above is a bit more complex than the basic "classification of new data" process described in this chapter, the fundamental process—classifying a record according to its proximity to other records—is the same at its core. Note the role of domain knowledge in this machine learning process—the variables have been tested and selected by the project leaders, and the measurements have been made by human experts.

Further reading: See www.pandora.com, Wikipedia's article on the Music Genome Project, and Joyce John's article "Pandora and the Music Genome Project," *Scientific Computing*, vol. 23, no. 10: 14, p. 40–41, Sep. 2006.

[1] See Section 12.5 in Chapter 12 for an explanation of statistical distance.

[2] See Chapter 15 for more on clusters.

[3] See Case 21.6 "Segmenting Consumers of Bath Soap" for an exercise involving the identification of clusters, which are then used for classification purposes.

7.3 Advantages and Shortcomings of k-NN Algorithms

The main advantage of k-NN methods is their simplicity and lack of parametric assumptions. In the presence of a large enough training set, these methods perform surprisingly well, especially when each class is characterized by multiple combinations of predictor values. For instance, in real-estate databases, there are likely to be multiple combinations of {home type, number of rooms, neighborhood, asking price, etc.} that characterize homes that sell quickly vs. those that remain for a long period on the market.

There are three difficulties with the practical exploitation of the power of the k-NN approach. First, although no time is required to estimate parameters from the training data (as would be the case for parametric models such as regression), the time to find the nearest neighbors in a large training set can be prohibitive. A