Data Mining I

Classification
- Part 3 -
Outline

1. What is Classification?
2. K-Nearest-Neighbors
3. Decision Trees
4. Model Evaluation
5. Rule Learning
6. Naïve Bayes
7. Artificial Neural Networks
8. Support Vector Machines
9. Parameter Tuning
6. Bayesian Classifiers

- Classifiers based on Bayes Theorem
- Thomas Bayes (1701-1761)
  - British mathematician and priest
  - tried to formally prove the existence of God

\[
P(C|A) = \frac{P(A|C)P(C)}{P(A)}
\]

- Computes one conditional probability \(P(C|A)\) out of another \(P(A|C)\)
  - given that the base probabilities \(P(A)\) and \(P(C)\) are known

- Useful in situations where \(P(C|A)\) is unknown
  - while \(P(A|C)\), \(P(A)\) and \(P(C)\) are known or easy to determine/estimate
Example of Bayes Theorem

- Given a symptom, what's the probability that I have a certain disease?
- Given:
  - A doctor knows that meningitis causes stiff neck 50% of the time $\Rightarrow P(S|M)$
  - Prior probability of any patient having meningitis is $1/50,000 \Rightarrow P(M)$
  - Prior probability of any patient having stiff neck is $1/20 \Rightarrow P(S)$
- If a patient has stiff neck, what's the probability he/she has meningitis?

$$P(M|S) = \frac{P(S|M)P(M)}{P(S)} = \frac{0.5 \times 1/50000}{1/20} = 0.0002$$
Bayes Rule: Evidence Formulation

- Probability of event $H$ given evidence $E$ :

$$
\Pr[H \mid E] = \frac{\Pr[E \mid H] \Pr[H]}{\Pr[E]}
$$

- A priori probability of $H$ :
  - Probability of event before evidence is seen.

- A posteriori probability of $H$ :
  - Probability of event after evidence is seen.
Bayesian Classifiers

- consider each attribute and class label as random variables.
- Given a record with attributes \((A_1, A_2, \ldots, A_n)\), the goal is to predict class \(C\).
- Specifically, we want to find the value of \(C\) that maximizes:

\[
P(C| A_1, A_2, \ldots, A_n)
\]
Classifying an unseen Record

1. compute the posterior probability \( P(C \mid A_1, A_2, \ldots, A_n) \) for all values of \( C \) using Bayes theorem

\[
P(C \mid A_1 A_2 \ldots A_n) = \frac{P(A_1 A_2 \ldots A_n \mid C) P(C)}{P(A_1 A_2 \ldots A_n)}
\]

2. Choose value of \( C \) that maximizes
\[
P(C \mid A_1, A_2, \ldots, A_n)
\]

Equivalent to choosing value of \( C \) that maximizes \( P(A_1, A_2, \ldots, A_n \mid C) P(C) \) as \( P(A_1, A_2, \ldots, A_n) \) is the same for all \( C \).

• How to estimate \( P(A_1, A_2, \ldots, A_n \mid C) \)?
Naïve Bayes Classifiers

- Naïve Bayes is a widely used classification technique
  - Especially successful at classifying texts

- Naïve Bayes Classifiers are based on two assumptions:
  1. All attributes are *equally important*
  2. All attributes are *statistically independent*
     - knowing the value of one attribute says nothing about the value of another
     - the independence assumption is almost never correct!
     - but … this scheme works well in practice.

- Reformulation using the independence assumption:
  \[
P(A_1, A_2, \ldots, A_n \mid C) = P(A_1 \mid C_j) \cdot P(A_2 \mid C_j) \cdot \ldots \cdot P(A_n \mid C_j)
\]

- Result: We can estimate \(P(A_i \mid C_j)\) for all \(A_i\) and \(C_j\) as well as \(P(C)\) directly from the data (fractions).
Estimating the Probabilities for the Weather Data

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sunny</td>
<td>2   3</td>
<td>Hot</td>
<td>2   2</td>
<td>High</td>
</tr>
<tr>
<td>Overcast</td>
<td>4   0</td>
<td>Mild</td>
<td>4   2</td>
<td>Normal</td>
</tr>
<tr>
<td>Rainy</td>
<td>3   2</td>
<td>Cool</td>
<td>3   1</td>
<td></td>
</tr>
<tr>
<td>Sunny</td>
<td>2/9 3/5</td>
<td>Hot</td>
<td>2/9 2/5</td>
<td>High</td>
</tr>
<tr>
<td>Overcast</td>
<td>4/9 0/5</td>
<td>Mild</td>
<td>4/9 2/5</td>
<td>Normal</td>
</tr>
<tr>
<td>Rainy</td>
<td>3/9 2/5</td>
<td>Cool</td>
<td>3/9 1/5</td>
<td></td>
</tr>
</tbody>
</table>

Probabilities are estimated by counting the values of each attribute for each possible value of the class attribute:

2 times “Yes” together with “Outlook=sunny” out of altogether 9 “Yes” examples

⇒ p(Outlook=sunny|Yes) = 2/9
Classifying a New Day

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

\[ \Pr[yes \mid E] = \Pr[Outlook = Sunny \mid yes] \times \Pr[Temperatur = Cool \mid yes] \times \Pr[Humidity = High \mid yes] \times \Pr[Windy = True \mid yes] \times \frac{\Pr[yes]}{\Pr[E]} \]

\[ = \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{\Pr[E]} \]
### Classifying a New Day: Weigh the evidence!

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>2/9</td>
<td>3/9</td>
<td>2/9</td>
<td>3</td>
</tr>
<tr>
<td>Overcast</td>
<td>4/9</td>
<td>0/5</td>
<td>2/5</td>
<td>3/9</td>
</tr>
<tr>
<td>Rainy</td>
<td>3/9</td>
<td>2/5</td>
<td>3/9</td>
<td>1/5</td>
</tr>
</tbody>
</table>

#### A new day:

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

#### Likelihood of the two classes

- A priori
  - Prejudice
  - Evidence

#### Choose Maximum

Let's calculate the likelihood for each class:

- For "yes":
  \[ P(\text{"yes"}) = \frac{2/9 \times 3/9 \times 3/9 \times 3/9 \times 9/14}{0.0053} \]

- For "no":
  \[ P(\text{"no"}) = \frac{3/5 \times 1/5 \times 4/5 \times 3/5 \times 5/14}{0.0206} \]

Conversion into a probability by normalization:

- \[ P(\text{"yes"}) = 0.0053 / (0.0053 + 0.0206) = 0.205 \]
- \[ P(\text{"no"}) = 0.0206 / (0.0053 + 0.0206) = 0.795 \]
The Zero-Frequency Problem

What if an attribute value doesn’t occur with every class value? (e.g. “Outlook = overcast” for class “no”)

- Probability will be zero!
  \[ \Pr[\text{Outlook} = \text{overcast} \mid \text{no}] = 0 \]

- Problem: A posteriori probability will also be zero! (No matter how likely the other values are!)
  \[ \Pr[\text{yes} \mid E] = 0 \]

Remedy: Add 1 to the count for every attribute value-class combination (Laplace Estimator)

Result: Probabilities will never be zero! (also: stabilizes probability estimates)

Original: \[ P(A_i \mid C) = \frac{N_{ic}}{N_c} \]

Laplace: \[ P(A_i \mid C) = \frac{N_{ic} + 1}{N_c + c} \]

\[ c: \text{number of classes} \]
Handling Missing Values

- **Training**: Instance is not included in frequency count for attribute value-class combination.
- **Classification**: Attribute will be omitted from calculation.
- **Example:**

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>Cool</td>
<td>High</td>
<td>True</td>
<td>?</td>
</tr>
</tbody>
</table>

Likelihood of “yes” = $\frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14} = 0.0238$

Likelihood of “no” = $\frac{1}{5} \times \frac{4}{5} \times \frac{3}{5} \times \frac{5}{14} = 0.0343$

$P(“yes”) = \frac{0.0238}{0.0238 + 0.0343} = 41\%$

$P(“no”) = \frac{0.0343}{0.0238 + 0.0343} = 59\%$
Handling Numeric Attributes

- Option 1: Discretize data before learning classifier.
- Option 2: Make assumption that numeric attributes have a normal distribution (given the class).
- For Option 2, the probability density function for the normal distribution is defined by two parameters:
  - Sample mean $\mu$
    $$\mu = \frac{1}{n} \sum_{i=1}^{n} x_i$$
  - Standard deviation $\sigma$
    $$\sigma = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \mu)^2$$
  - The density function $f(x)$ is
    $$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$
### Statistics for Weather Data

<table>
<thead>
<tr>
<th></th>
<th><strong>Outlook</strong></th>
<th><strong>Temperature</strong></th>
<th><strong>Humidity</strong></th>
<th><strong>Windy</strong></th>
<th><strong>Play</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>Sunny</td>
<td>64, 68, ..</td>
<td>65, 71,</td>
<td>Yes</td>
<td>6</td>
</tr>
<tr>
<td>No</td>
<td>Overcast</td>
<td>69, 70, ..</td>
<td>72, 80,</td>
<td>True</td>
<td>3</td>
</tr>
<tr>
<td>Yes</td>
<td>Rainy</td>
<td>72, ..</td>
<td>85, ..</td>
<td>False</td>
<td>9/14</td>
</tr>
<tr>
<td>No</td>
<td>Sunny</td>
<td>65, 70, ..</td>
<td>70, 85,</td>
<td>True</td>
<td>6/9</td>
</tr>
<tr>
<td>Yes</td>
<td>Overcast</td>
<td>72, 80, ..</td>
<td>70, 75,</td>
<td>False</td>
<td>2/5</td>
</tr>
<tr>
<td>Yes</td>
<td>Rainy</td>
<td>85, ..</td>
<td>90, 91,</td>
<td>True</td>
<td>3/9</td>
</tr>
<tr>
<td>No</td>
<td>Sunny</td>
<td>65, 71,</td>
<td>70, 85,</td>
<td>False</td>
<td>6/9</td>
</tr>
<tr>
<td>No</td>
<td>Overcast</td>
<td>72, 80, ..</td>
<td>70, 75,</td>
<td>True</td>
<td>3/9</td>
</tr>
<tr>
<td>Yes</td>
<td>Rainy</td>
<td>85, ..</td>
<td>90, 91,</td>
<td>True</td>
<td>3/9</td>
</tr>
</tbody>
</table>

#### Example density value:

\[
f(\text{temp} = 66 \mid \text{yes}) = \frac{1}{\sqrt{2\pi \sigma^2}} e^{-\frac{(66-73)^2}{2\sigma^2}} = 0.0340
\]
Classifying a new day

- A new day:

<table>
<thead>
<tr>
<th>Outlook</th>
<th>Temp.</th>
<th>Humidity</th>
<th>Windy</th>
<th>Play</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sunny</td>
<td>66</td>
<td>90</td>
<td>true</td>
<td>?</td>
</tr>
</tbody>
</table>

Likelihood of “yes” = \( \frac{2}{9} \times 0.0340 \times 0.0221 \times \frac{3}{9} \times \frac{9}{14} = 0.000036 \)

Likelihood of “no” = \( \frac{3}{5} \times 0.0291 \times 0.0380 \times \frac{3}{5} \times \frac{5}{14} = 0.000136 \)

\[ P(\text{“yes”}) = \frac{0.000036}{0.000036 + 0.000136} = 20.9\% \]

\[ P(\text{“no”}) = \frac{0.000136}{0.000036 + 0.000136} = 79.1\% \]

- But note: Many numeric attributes are not normally distributed and you may thus need to choose a different probability distribution function.
Naïve Bayes Discussion

- Naïve Bayes works surprisingly well.
  - even if independence assumption is clearly violated
  - Why? Because classification doesn’t require accurate probability estimates as long as maximum probability is assigned to correct class

- However: Adding too many redundant attributes will cause problems.
  - Solution: Select attribute subset as Naïve Bayes often works as well or better with just a fraction of all attributes.

- Technical advantages:
  - Learning Naïve Bayes classifiers is computationally cheap as probabilities can be estimated doing one pass over the training data.
  - Storing the probabilities does not require a lot on memory.
Naive Bayes in RapidMiner
Naive Bayes in RapidMiner

Table:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Parameter</th>
<th>no</th>
<th>yes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outlook</td>
<td>value=rain</td>
<td>0.392</td>
<td>0.331</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=overcast</td>
<td>0.014</td>
<td>0.438</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=sunny</td>
<td>0.581</td>
<td>0.223</td>
</tr>
<tr>
<td>Outlook</td>
<td>value=unknown</td>
<td>0.014</td>
<td>0.008</td>
</tr>
<tr>
<td>Temperature</td>
<td>mean</td>
<td>74.600</td>
<td></td>
</tr>
<tr>
<td>Temperature</td>
<td>standard deviation</td>
<td>7.893</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>mean</td>
<td>84</td>
<td></td>
</tr>
<tr>
<td>Humidity</td>
<td>standard deviation</td>
<td>9.618</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=true</td>
<td>0.589</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=false</td>
<td>0.397</td>
<td></td>
</tr>
<tr>
<td>Wind</td>
<td>value=unknown</td>
<td>0.014</td>
<td></td>
</tr>
</tbody>
</table>

Graph:

- Distribution of humidity for 'no' and 'yes' classes.
- The graph shows the density distribution of humidity with two peaks, one for 'no' and one for 'yes' classes.
Naïve Bayes in RapidMiner

<table>
<thead>
<tr>
<th>Row No.</th>
<th>Play</th>
<th>confidence(no)</th>
<th>confidence(yes)</th>
<th>prediction(Play)</th>
<th>Outlook</th>
<th>Temperature</th>
<th>Humidity</th>
<th>Wind</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>yes</td>
<td>0.711</td>
<td>0.289</td>
<td>no</td>
<td>sunny</td>
<td>85</td>
<td>85</td>
<td>false</td>
</tr>
<tr>
<td>2</td>
<td>no</td>
<td>0.058</td>
<td>0.942</td>
<td>yes</td>
<td>overcast</td>
<td>80</td>
<td>90</td>
<td>true</td>
</tr>
<tr>
<td>3</td>
<td>yes</td>
<td>0.014</td>
<td>0.986</td>
<td>yes</td>
<td>overcast</td>
<td>83</td>
<td>78</td>
<td>false</td>
</tr>
<tr>
<td>4</td>
<td>yes</td>
<td>0.412</td>
<td>0.588</td>
<td>yes</td>
<td>rain</td>
<td>70</td>
<td>96</td>
<td>false</td>
</tr>
<tr>
<td>5</td>
<td>yes</td>
<td>0.460</td>
<td>0.540</td>
<td>yes</td>
<td>rain</td>
<td>68</td>
<td>80</td>
<td>true</td>
</tr>
<tr>
<td>6</td>
<td>no</td>
<td>0.336</td>
<td>0.664</td>
<td>no</td>
<td>rain</td>
<td>65</td>
<td>70</td>
<td>true</td>
</tr>
<tr>
<td>7</td>
<td>yes</td>
<td>0.010</td>
<td>0.990</td>
<td>yes</td>
<td>rain</td>
<td>65</td>
<td>70</td>
<td>true</td>
</tr>
<tr>
<td>8</td>
<td>no</td>
<td>0.596</td>
<td>0.404</td>
<td>no</td>
<td>overcast</td>
<td>85</td>
<td>75</td>
<td>true</td>
</tr>
<tr>
<td>9</td>
<td>yes</td>
<td>0.248</td>
<td>0.752</td>
<td>yes</td>
<td>sunny</td>
<td>69</td>
<td>70</td>
<td>false</td>
</tr>
<tr>
<td>10</td>
<td>no</td>
<td>0.407</td>
<td>0.593</td>
<td>yes</td>
<td>sunny</td>
<td>75</td>
<td>80</td>
<td>true</td>
</tr>
<tr>
<td>11</td>
<td>yes</td>
<td>0.496</td>
<td>0.504</td>
<td>yes</td>
<td>overcast</td>
<td>81</td>
<td>75</td>
<td>true</td>
</tr>
<tr>
<td>12</td>
<td>yes</td>
<td>0.038</td>
<td>0.962</td>
<td>yes</td>
<td>overcast</td>
<td>81</td>
<td>75</td>
<td>true</td>
</tr>
<tr>
<td>13</td>
<td>no</td>
<td>0.027</td>
<td>0.973</td>
<td>yes</td>
<td>overcast</td>
<td>81</td>
<td>75</td>
<td>true</td>
</tr>
<tr>
<td>14</td>
<td>yes</td>
<td>0.453</td>
<td>0.547</td>
<td>yes</td>
<td>rain</td>
<td>71</td>
<td>80</td>
<td>true</td>
</tr>
</tbody>
</table>

- classifier is quite sure
- classifier is not sure
Alternative Classification Methods

• There are various methods of classification
  – e.g., 53 methods in basic RapidMiner edition

• So far, we have seen
  1. k-NN
  2. Decision Trees
  3. C4.5 and Ripper
  4. Naive Bayes

• Now: Brief Introduction into
  1. Artificial Neural Networks
  2. Support Vector Machines
7. Artificial Neural Networks (ANN)

• Inspiration
  – one of the most powerful super computers in the world
Artificial Neural Networks (ANN)

Example:
Output $Y$ is 1 if at least two of the three inputs are equal to 1.
Artificial Neural Networks (ANN)

\[ Y = I \left( 0.3X_1 + 0.3X_2 + 0.3X_3 - 0.4 > 0 \right) \]

where \( I(z) = \begin{cases} 1 & \text{if } z \text{ is true} \\ 0 & \text{otherwise} \end{cases} \)
Artificial Neural Networks (ANN)

- Model is an assembly of inter-connected nodes and weighted links
- Output node sums up each of its input value according to the weights of its links
- Classification decision: Compare output node against some threshold $t$

Perceptron Model

$$Y = I \left( \sum_{i} w_i X_i - t \right) \quad \text{or} \quad Y = \text{sign} \left( \sum_{i} w_i X_i - t \right)$$
Multi-Layer Artificial Neural Networks

Training ANN means learning the weights of the neurons.
Algorithm for learning ANN

- Initialize the weights \( (w_0, w_1, \ldots, w_k) \), e.g., all with 1

- Adjust the weights in such a way that the output of ANN is consistent with class labels of training examples

  - Objective function:
    \[
    E = \sum_i \left[ Y_i - f(w_i, X_i) \right]^2
    \]

  - Find the weights \( w_i \)'s that minimize the above objective function
    - e.g., back propagation algorithm (see Tan Steinbach, Chapter 5.4)
ANN in RapidMiner
8. Support Vector Machines

- Support vector machines (SVMs) are algorithms for learning linear classifiers for
  - two class problems (a positive and a negative class)
  - from examples described by continuous attributes.

- SVMs achieve very good results especially for high dimensional data.

- SVMs were invented by V. Vapnik and his co-workers in 1970s in Russia and became known to the West in 1992.
Support Vector Machines

- SVMs find a linear hyperplane (decision boundary) that will separate the data.
Support Vector Machines

• Which one is better? B1 or B2?
• How do you define “better”? 
In order to avoid overfitting and to generalize for unseen data, SVMs find the hyperplane that maximizes the margin to the closest points (support vectors).

Visual solution: B1 is better than B2.

Mathematical solution: Constrained optimization that can be solved using quadratic programming.

See Tan/Steinbach/Kumar Chapter 5.5
Support Vector Machines

- Let the set of training examples $D$ be $\{(x_1, y_1), (x_2, y_2), \ldots, (x_r, y_r)\}$, where
  
  - $x_i = (x_1, x_2, \ldots, x_n)$ is an input vector in a real-valued space
  - $y_i$ is its class label ($y_i \in \{1, -1\}$, 1: positive class and -1: negative class)

- SVM finds a linear function (hyperplane) of the form

  $$f(x) = \langle w \cdot x \rangle + b$$

  where $w$ is a weight vector

- Classifying data using the hyperplane:

  $$y_i = \begin{cases} 
  1 & \text{if } \langle w \cdot x_i \rangle + b \geq 0 \\
  -1 & \text{if } \langle w \cdot x_i \rangle + b < 0 
  \end{cases}$$
Application Areas of SVMs

- SVM are often the most successful classification technique for high dimensional data.
- Application areas of SVM include
  - Text classification
  - Machine vision, e.g. face identification
  - Handwritten digit recognition
  - SPAM detection
  - Bioinformatics
SVMs in RapidMiner

The diagram illustrates the process of using SVM (Support Vector Machine) in RapidMiner. The SVM component is connected to other data flow components, which are likely to be responsible for data retrieval and transformation.

The SVM parameters are set as follows:
- **Svm type**: C-SVC
- **Kernel type**: rbf
- **Gamma**: 0.0
- **C**: 0.0
- **Cache size**: 80
- **Epsilon**: 0.001

Additionally, the SVM settings include options for shrinking and calculating confidences, with the latter being enabled.
9. Parameter Tuning

- Many learning methods require parameters
  - $k$ for k-nearest-neighbors
  - pruning thresholds for trees and rules
  - hidden layer configuration for ANN
  - epsilon and gamma for SVM
  - ...

- Some methods often work rather poorly with default parameters.

- How to determine the optimal parameters?
  - Play around with different parameters in RapidMiner yourself
  - Alternative: Let RapidMiner test different parameter settings for you
Parameter Optimization

– Let Rapidminer test various parameter combinations for you.
Optimization approaches that are more efficient than brute force GridSearch are discussed in Data Mining II (Beam Search, Evolutionary Algorithms)
Attribute Selection Optimization

- Let Rapidminer select the optimal subset of the available attributes.

Forward selection: Find best single attribute, add further attributes, test again
Backward selection: Start with all attribute, remove attributes, test again
Outputing Interim Results

- Using the Log Operator

![Log Operator Diagram]

**Column Name**

**Operator**

**parameter or port**

**Selection of output**

<table>
<thead>
<tr>
<th>Column Name</th>
<th>Operator</th>
<th>parameter or port</th>
<th>Selection of output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
<td>Neural Net</td>
<td>parameter</td>
<td>shuffle</td>
</tr>
<tr>
<td>Training</td>
<td>Neural Net</td>
<td>parameter</td>
<td>training_cycles</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Validation</td>
<td>value</td>
<td>performance</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Log (22 rows, 3 columns)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shuffle</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>false</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>false</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>false</td>
</tr>
<tr>
<td>true</td>
</tr>
<tr>
<td>false</td>
</tr>
</tbody>
</table>
Further Information

- Parameter Optimization YouTube Video:
  - [http://www.youtube.com/watch?v=R5vPrTLMzng](http://www.youtube.com/watch?v=R5vPrTLMzng)

- Attribute Selection Optimization YouTube Video:
  - Part 1: [http://www.youtube.com/watch?v=7IC3IQEdWxA](http://www.youtube.com/watch?v=7IC3IQEdWxA)
  - Part 2: [http://www.youtube.com/watch?v=j5vhwbLlZWg](http://www.youtube.com/watch?v=j5vhwbLlZWg)