Weight Lifting: Machine Learning Analysis

Eric Solano

October 21, 2015

1. Data cleaning and exploration

The training dataset was loaded and cleaned. First, all columns with multiple missing values were removed. All variables whose name start with 'kurtosis_', 'skewness_', 'max_', 'min_', 'stddev_', 'var_', 'avg_' and 'amplitude_' were removed.
Next, a correlation analysis was performed using the 'cor' function from the 'caret' package.
Highly correlated variables with correlation > 0.75 were identified and removed.

The removed highly correlated variables are: accel_belt_z, roll_belt, accel_belt_y, accel_arm_y, total_accel_belt, accel_dumbbell_z, accel_belt_x, pitch_belt, magnet_dumbbell_x, accel_dumbbell_y, magnet_dumbbell_y, accel_arm_x, accel_dumbbell_x, accel_arm_z, magnet_arm_y, magnet_belt_z, accel_forearm_y, gyros_forearm_y, gyros_dumbbell_x, gyros_dumbbell_z, gyros_arm_x.

2. Creation of folds for cross validation using 10-fold

Function 'createFolds' from package 'caret' was used for a 10-fold cross-validation analysis.
The 10-fold methodology breaks data into 10 sets of size n/10; trains on 9 datasets and tests on 1; and repeats 10 times. An accuracy value is calculated for each run and a mean accuracy is calculated at the end.

3. Supervised Learning using Classification

3.1. Use Naive-Bayes classifier to train data

The Naive-Bayes classifier was used to train the training sets and to test the testing sets formed by using the 10 folds.
Accuracy was calculated for each one of the 10 model training/testing runs. Accuracy is defined as the number of correct predictions divided by the number of total data points.
The mean accuracy value was calculated for the 10 accuracy values obtained.

3.2. Use knn classifier to train data

The knn classifier was used to train the training sets and to test the testing sets formed by using the 10 folds.
The mean accuracy was calculated in a similar way as with the Naive-Bayes classifier.

3.3. Compare results

The confusion matrix for the Naive-Bayes classifier from one of the runs is shown below:

	A	B	C	D	E
A	285	56	134	69	14
B	29	214	79	26	32

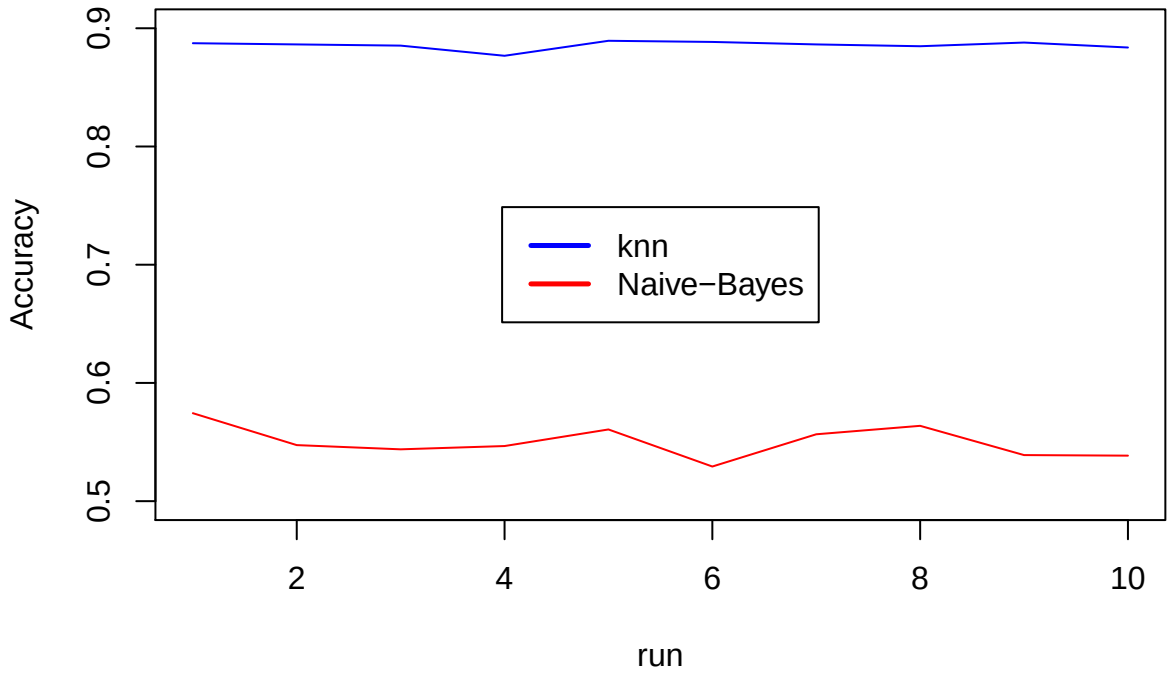
	A	B	C	D	E
C	11	39	245	30	17
D	4	16	102	154	45
E	8	95	52	47	158

The confusion matrix for the knn classifier from one of the runs is shown below:

	A	B	C	D	E
A	522	10	9	13	4
B	15	324	15	7	19
C	0	11	311	13	7
D	4	5	27	279	6
E	12	19	14	18	297

Figure 1 shows the comparison of those 2 methods using accuracy as the performance metric.
The knn classifier (in blue) performs much better with mean accuracy = 0.89 than the Naive-Bayes classifier with mean accuracy = 0.55 (in red).
Accuracy is a good performance indicator when the distribution of class labels is not skewed. Out-of-sample error can be approximated as 1-accuracy (other proposed metrics from the literature mention 1 - average recall). The higher the accuracy, the lower the out-of-sample error.

Figure 1: Accuracy comparison for 2 algorithms



method	OOB_error
Naive-Bayes	0.45

method	OOB_error
knn	0.11

4. Predictions for new data

A test dataset with 20 cases was used to find the predicted response from both algorithms.
The following table sumarizes the findings:

case	Naive.Bayes	knn
1	D	B
2	C	A
3	C	B
4	A	A
5	B	A
6	A	E
7	C	D
8	D	B
9	A	A
10	A	A
11	C	B
12	C	C
13	B	E
14	A	A
15	B	E
16	B	E
17	C	A
18	B	B
19	D	B
20	B	B