Untitled

Abhinav Sehgal

24/10/2020

Introduction

Took the data generated by the following study (http://groupware.les.inf.puc-rio.br/har. I'll include the following things: question, data input, features, algorithm, parameters and evaluation.

Question Statement

According to the study afore mentioned: "Six young health participants were asked to perform one set of 10 repetitions of the Unilateral Dumbbell Biceps Curl in five different fashions: exactly according to the specification (Class A), throwing the elbows to the front (Class B), lifting the dumbbell only halfway (Class C), lowering the dumbbell only halfway (Class D) and throwing the hips to the front (Class E). Class A Corresponds to the specified execution of the exercise, while the other 4 classes Correspond to common mistakes." Our goal is to predict the manner in which they did the exercise.

cleaning Data

```
library(data.table)
library(rpart)


library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
TrainD <- read.csv("https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv", header=T, na
TestD <- read.csv("http://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv", header=T, na.st
dim(TrainD)
```

```
## [1] 19622   160
```

```
dim(TestD)
```

```
## [1]  20 160
```

TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

```r
names(TrainD)
```

```
##   [1] "X"                        "user_name"
##   [3] "raw_timestamp_part_1"     "raw_timestamp_part_2"
##   [5] "cvtd_timestamp"           "new_window"
##   [7] "num_window"               "roll_belt"
##   [9] "pitch_belt"               "yaw_belt"
##  [11] "total_accel_belt"         "kurtosis_roll_belt"
##  [13] "kurtosis_picth_belt"      "kurtosis_yaw_belt"
##  [15] "skewness_roll_belt"       "skewness_roll_belt.1"
##  [17] "skewness_yaw_belt"        "max_roll_belt"
##  [19] "max_picth_belt"           "max_yaw_belt"
##  [21] "min_roll_belt"            "min_pitch_belt"
##  [23] "min_yaw_belt"             "amplitude_roll_belt"
##  [25] "amplitude_pitch_belt"     "amplitude_yaw_belt"
##  [27] "var_total_accel_belt"     "avg_roll_belt"
##  [29] "stddev_roll_belt"         "var_roll_belt"
##  [31] "avg_pitch_belt"           "stddev_pitch_belt"
##  [33] "var_pitch_belt"           "avg_yaw_belt"
##  [35] "stddev_yaw_belt"          "var_yaw_belt"
##  [37] "gyros_belt_x"             "gyros_belt_y"
##  [39] "gyros_belt_z"             "accel_belt_x"
##  [41] "accel_belt_y"             "accel_belt_z"
##  [43] "magnet_belt_x"            "magnet_belt_y"
##  [45] "magnet_belt_z"            "roll_arm"
##  [47] "pitch_arm"                "yaw_arm"
##  [49] "total_accel_arm"          "var_accel_arm"
##  [51] "avg_roll_arm"             "stddev_roll_arm"
##  [53] "var_roll_arm"             "avg_pitch_arm"
##  [55] "stddev_pitch_arm"         "var_pitch_arm"
##  [57] "avg_yaw_arm"              "stddev_yaw_arm"
##  [59] "var_yaw_arm"              "gyros_arm_x"
##  [61] "gyros_arm_y"              "gyros_arm_z"
##  [63] "accel_arm_x"              "accel_arm_y"
##  [65] "accel_arm_z"              "magnet_arm_x"
##  [67] "magnet_arm_y"             "magnet_arm_z"
##  [69] "kurtosis_roll_arm"        "kurtosis_picth_arm"
##  [71] "kurtosis_yaw_arm"         "skewness_roll_arm"
##  [73] "skewness_pitch_arm"       "skewness_yaw_arm"
##  [75] "max_roll_arm"             "max_picth_arm"
##  [77] "max_yaw_arm"              "min_roll_arm"
##  [79] "min_pitch_arm"            "min_yaw_arm"
##  [81] "amplitude_roll_arm"       "amplitude_pitch_arm"
##  [83] "amplitude_yaw_arm"        "roll_dumbbell"
##  [85] "pitch_dumbbell"           "yaw_dumbbell"
##  [87] "kurtosis_roll_dumbbell"   "kurtosis_picth_dumbbell"
##  [89] "kurtosis_yaw_dumbbell"    "skewness_roll_dumbbell"
##  [91] "skewness_pitch_dumbbell"  "skewness_yaw_dumbbell"
##  [93] "max_roll_dumbbell"        "max_picth_dumbbell"
##  [95] "max_yaw_dumbbell"         "min_roll_dumbbell"
```

```
##  [97] "min_pitch_dumbbell"       "min_yaw_dumbbell"
##  [99] "amplitude_roll_dumbbell"  "amplitude_pitch_dumbbell"
## [101] "amplitude_yaw_dumbbell"   "total_accel_dumbbell"
## [103] "var_accel_dumbbell"       "avg_roll_dumbbell"
## [105] "stddev_roll_dumbbell"     "var_roll_dumbbell"
## [107] "avg_pitch_dumbbell"       "stddev_pitch_dumbbell"
## [109] "var_pitch_dumbbell"       "avg_yaw_dumbbell"
## [111] "stddev_yaw_dumbbell"      "var_yaw_dumbbell"
## [113] "gyros_dumbbell_x"         "gyros_dumbbell_y"
## [115] "gyros_dumbbell_z"         "accel_dumbbell_x"
## [117] "accel_dumbbell_y"         "accel_dumbbell_z"
## [119] "magnet_dumbbell_x"        "magnet_dumbbell_y"
## [121] "magnet_dumbbell_z"        "roll_forearm"
## [123] "pitch_forearm"            "yaw_forearm"
## [125] "kurtosis_roll_forearm"    "kurtosis_picth_forearm"
## [127] "kurtosis_yaw_forearm"     "skewness_roll_forearm"
## [129] "skewness_pitch_forearm"   "skewness_yaw_forearm"
## [131] "max_roll_forearm"         "max_picth_forearm"
## [133] "max_yaw_forearm"          "min_roll_forearm"
## [135] "min_pitch_forearm"        "min_yaw_forearm"
## [137] "amplitude_roll_forearm"   "amplitude_pitch_forearm"
## [139] "amplitude_yaw_forearm"    "total_accel_forearm"
## [141] "var_accel_forearm"        "avg_roll_forearm"
## [143] "stddev_roll_forearm"      "var_roll_forearm"
## [145] "avg_pitch_forearm"        "stddev_pitch_forearm"
## [147] "var_pitch_forearm"        "avg_yaw_forearm"
## [149] "stddev_yaw_forearm"       "var_yaw_forearm"
## [151] "gyros_forearm_x"          "gyros_forearm_y"
## [153] "gyros_forearm_z"          "accel_forearm_x"
## [155] "accel_forearm_y"          "accel_forearm_z"
## [157] "magnet_forearm_x"         "magnet_forearm_y"
## [159] "magnet_forearm_z"         "classe"
```

Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

```
LNA<- sapply(TrainD, function (x) any(is.na(x)))
NTD <- subset(TrainD, select=c("classe", names(LNA)[!LNA & grepl("belt|[^(fore)]arm|dumbbell|forearm", n
```

convert classe to a Factor data type, so that caret builds a classification rather than of a regression model.

Train d - 60 test d = 40

```
inTrain <- createDataPartition(NTD$classe, p=0.6, list=FALSE)
Train_data <-  NTD[inTrain, ]
Test_data <- NTD[-inTrain, ]
```

Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

```
NZV <- nearZeroVar(Train_data, saveMetrics=TRUE)
NZV
```

```
##                      freqRatio percentUnique zeroVar   nzv
## classe                1.469065    0.04245924   FALSE FALSE
## roll_belt             1.032143    8.74660326   FALSE FALSE
## pitch_belt            1.018349   13.68036685   FALSE FALSE
## yaw_belt              1.133758   14.40217391   FALSE FALSE
## total_accel_belt      1.100166    0.23777174   FALSE FALSE
## gyros_belt_x          1.113466    1.06997283   FALSE FALSE
## gyros_belt_y          1.190398    0.53498641   FALSE FALSE
## gyros_belt_z          1.068901    1.34171196   FALSE FALSE
## accel_belt_x          1.066955    1.32472826   FALSE FALSE
## accel_belt_y          1.106724    1.15489130   FALSE FALSE
## accel_belt_z          1.086207    2.37771739   FALSE FALSE
## magnet_belt_x         1.035398    2.53057065   FALSE FALSE
## magnet_belt_y         1.106267    2.32676630   FALSE FALSE
## magnet_belt_z         1.035587    3.59205163   FALSE FALSE
## roll_arm             49.756098   19.49728261   FALSE FALSE
## pitch_arm            85.041667   22.29110054   FALSE FALSE
## yaw_arm              30.909091   21.39945652   FALSE FALSE
## total_accel_arm       1.009208    0.56046196   FALSE FALSE
## gyros_arm_x           1.026059    5.30740489   FALSE FALSE
## gyros_arm_y           1.400651    3.07404891   FALSE FALSE
## gyros_arm_z           1.141956    1.97010870   FALSE FALSE
## accel_arm_x           1.035088    6.41983696   FALSE FALSE
## accel_arm_y           1.166667    4.44123641   FALSE FALSE
## accel_arm_z           1.092105    6.43682065   FALSE FALSE
## magnet_arm_x          1.054545   11.15828804   FALSE FALSE
## magnet_arm_y          1.120000    7.20957880   FALSE FALSE
## magnet_arm_z          1.156250   10.57235054   FALSE FALSE
## roll_dumbbell         1.065789   87.55095109   FALSE FALSE
## pitch_dumbbell        2.565789   85.34307065   FALSE FALSE
## yaw_dumbbell          1.041096   86.65930707   FALSE FALSE
## total_accel_dumbbell  1.147541    0.34816576   FALSE FALSE
## gyros_dumbbell_x      1.070225    1.91915761   FALSE FALSE
## gyros_dumbbell_y      1.216066    2.24184783   FALSE FALSE
## gyros_dumbbell_z      1.023684    1.62194293   FALSE FALSE
## accel_dumbbell_x      1.004902    3.37975543   FALSE FALSE
## accel_dumbbell_y      1.068493    3.83831522   FALSE FALSE
## accel_dumbbell_z      1.183099    3.35427989   FALSE FALSE
## magnet_dumbbell_x     1.210000    8.89945652   FALSE FALSE
## magnet_dumbbell_y     1.050000    6.81046196   FALSE FALSE
## magnet_dumbbell_z     1.088235    5.54517663   FALSE FALSE
## roll_forearm         12.062500   14.53804348   FALSE FALSE
## pitch_forearm        66.114286   21.14470109   FALSE FALSE
## yaw_forearm          15.124183   14.09646739   FALSE FALSE
## total_accel_forearm   1.150943    0.56895380   FALSE FALSE
## gyros_forearm_x       1.130435    2.32676630   FALSE FALSE
## gyros_forearm_y       1.074561    6.00373641   FALSE FALSE
## gyros_forearm_z       1.125448    2.38620924   FALSE FALSE
## accel_forearm_x       1.094340    6.56419837   FALSE FALSE
## accel_forearm_y       1.015385    8.18614130   FALSE FALSE
## accel_forearm_z       1.063158    4.59408967   FALSE FALSE
## magnet_forearm_x      1.000000   11.94802989   FALSE FALSE
## magnet_forearm_y      1.094340   15.18342391   FALSE FALSE
## magnet_forearm_z      1.025000   13.30672554   FALSE FALSE
```

I use a plot to display the result in a best manner.

The plot show the relationship between the number of PLS components and the resampled estimate of the area under the ROC curve. And then, finally, I take a look to the confusion matrix and associated statistics.

I can apply another model like the "regularized discriminant analysis" model

The accuracy of this model is 99.5% I look closely to the final model, i can extract the variables that compone the model and see the confusion matrix of this model with the class.error. The class error is less tha 1% TensorFlow Install Learn Introduction New to TensorFlow? TensorFlow The core open source ML library For JavaScript TensorFlow.js for ML using JavaScript For Mobile & IoT TensorFlow Lite for mobile and embedded devices For Production TensorFlow Extended for end-to-end ML components Swift for TensorFlow (in beta) API TensorFlow (r2.3) r1.15 Versions… TensorFlow.js TensorFlow Lite TFX Resources Responsible AI Resources and tools to integrate Responsible AI practices into your ML workflow Models & datasets Pre-trained models and datasets built by Google and the community Tools Ecosystem of tools to help you use TensorFlow Libraries & extensions Libraries and extensions built on TensorFlow TensorFlow Certificate program Differentiate yourself by demonstrating your ML proficiency Learn ML Educational resources to learn the fundamentals of ML with TensorFlow Community Why TensorFlow About Case studies Trusted Partner Program Search Language GitHub Sign in Google is committed to advancing racial equity for Black communities. See how.

Why TensorFlow TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

About

Easy model building Build and train ML models easily using intuitive high-level APIs like Keras with eager execution, which makes for immediate model iteration and easy debugging.

Robust ML production anywhere Easily train and deploy models in the cloud, on-prem, in the browser, or on-device no matter what language you use.

Powerful experimentation for research A simple and flexible architecture to take new ideas from concept to code, to state-of-the-art models, and to publication faster.

Solutions to common ML problems Simple step-by-step walkthroughs to solve common ML problems with TensorFlow.

For beginners Your first neural network Train a neural network to classify images of clothing, like sneakers and shirts, in this fast-paced overview of a complete TensorFlow program.

For experts Generative adversarial networks Train a generative adversarial network to generate images of handwritten digits, using the Keras Subclassing API.

For experts Neural machine translation with attention Train a sequence-to-sequence model for Spanish to English translation using the Keras Subclassing API.

Powered by TensorFlow A diverse community of developers, enterprises and researchers are using ML to solve challenging, real-world problems. Learn how their research and applications are being #PoweredbyTF and how you can share your story.

Developers Helping doctors detect respiratory diseases using machine learning Watch the video

Enterprise Categorizing Airbnb listing photos using TensorFlow Watch the video

Research Enabling medical staff to prescribe the right antibiotics with TensorFlow Watch the video
Companies using TensorFlow See case studies
News & announcements Check out the TensorFlow blog for additional updates, and subscribe to our monthly TensorFlow newsletter to get the latest announcements sent directly to your inbox.

Sign up

June 29, 2020 Integrate Responsible AI practices into your ML workflow TensorFlow is committed to helping make progress in the responsible development of AI by sharing a collection of resources and tools with the ML community.

Learn more

Mar 16, 2020 Google Developers Machine Learning Crash Course The Machine Learning Crash Course is a self-study guide for aspiring machine learning practitioners featuring a series of lessons with video lectures, real-world case studies, and hands-on practice exercises.

Learn more

Mar 12, 2020 TensorFlow Dev Summit 2020 Recap Our virtual Dev Summit brought announcements of TensorFlow 2.2 plus many new features and additions to the ecosystem! Read the recap on our blog to learn about the updates and watch video recordings of every session.

Learn more

Mar 10, 2020 Differentiate yourself with the TensorFlow Developer Certificate Check out our TensorFlow Certificate program for practitioners to showcase their expertise in machine learning in an increasingly AI-driven global job market.

Learn more
Resources and community support We are committed to fostering an open and welcoming ML community. Join the TensorFlow community and help grow the ecosystem.

Community
Responsible AI DevPost Challenge Use TensorFlow 2.2 to build a model or application with AI Principles in mind. As you build, ask questions related to fairness, privacy, and security.