Untitled

Elizabeth Ash

April 12, 2016

Introduction

Predictive Modeling

- Predictive modeling is a process used to create a statistical model of future behavior.
- A predictive model is made up of a number of **predictors**, which are variable factors that are likely to influence future behaviors or results.
- In predictive modeling, data is collected for the relevant predictors, a statistical model is formulated, predictions are made, and the model is revised as additional data becomes available.
- My research project deals with predictive models and an application of such modeling.

Activity Recognition

- Activity recognition is an increasingly important technology because it can be applied to many real-life problems such as, home-based proactive and preventive healthcare applications.
- The goal of activity recognition is to recognize common human activities in real-life settings.
- One real-life setting example of activity recogniton is physical activity, which is one of the most important things that can be done for overall health.

- ▶ Using other physical activities as predictor variables, a predictive model could be made and used to help determine if an exercise motion is being executed properly.
- ▶ If a predictor model could be made, then the model could be integrated into the weight lift equipment and used to determine if the lift was done correctly or incorrectly. This model could be integrated with other technologies and be used to help reinforce the correct weight lift motion by commending the user for a correct movement or making a comment when the user made an incorrect movement.

Background on Data Used

- ▶ The project data is associated with the following study:

Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H.: Qualitative Activity Recognition of Weight Lifting Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human '13) . Stuttgart, Germany: ACM SIGCHI, 2013

- ▶ http://groupware.les.inf.puc-rio.br/har
- ▶ **Goal of Study**: To provide feedback to weight-lifters using qualitative activity recognition
- ▶ The study involved six male subjects, all in their twenties and with little weight-lifting experience. The subjects were taught how to lift a dumb-bell correctly and how to perform the same movement in four incorrect ways. The Unilateral Dumbbell Bicep Curl was the lift that was taught to the subjects.

- The five categories of lift data collected were:
  - Class A: correct lift movement
  - Class B: throwing the elbows to the front
  - Class C: lifting the dumbbell only halfway
  - Class D: lowering the dumbbell only halfway
  - Class E: throwing the hips to the front

- The subjects repeated each lift ten times and during each lift the researchers recorded a number of inertial measurements (IMUs) from sensors in the users' glove, armband, lumbar belt, and dumbbell (these are pieces of equipment that are commonly used by weight lifters).
- These measurements make up the predictors that the researchers used when determing a correct or incorrect lift. The sensors recorded several data points throughout the lifting motion and the final data set includes 160 variables.
- Some of the variables included are: user_name, num_window, yaw_belt, pitch_belt, and total_accel_belt.

Research Question

- The main question I would like to investigate using the data from Velloso, et. al is "How well would a random forest model work to predict activity type for new lifts performed by the same subjects in the experiment?"

Methods

Classification Trees

- Classification trees are a tree-based model and are used to predict a qualitative response.
- The variables that go into these classification trees can be numerical or categorical.
- A classification tree consists of a set of true/false decision rules.
- We can think of classification trees as a game of 20 questions, where we ask different questions based on the answers to previous questions, and then at the end we make a guess based on all the answers.

- We can visualize a decision tree as a set of nodes (corresponding to true/false questions), each of which has two branches depending on the answer to the question.
- Unlike real trees, we usually draw them with their "root" at the top, and the "leaves" at the bottom. In order to make predictions with the tree, we start at the top (the "root" node), and ask questions, traveling left or right in the tree based on what the answer is (left for true and right for false).
- At each step, we reach a new node, with a new question. Once we reach the bottom (a leaf node), we make a prediction based on the set of answers, just like 20 questions. But unlike 20 questions, the number of questions in a decision tree is not always 20, but can vary (Corso, 2013).
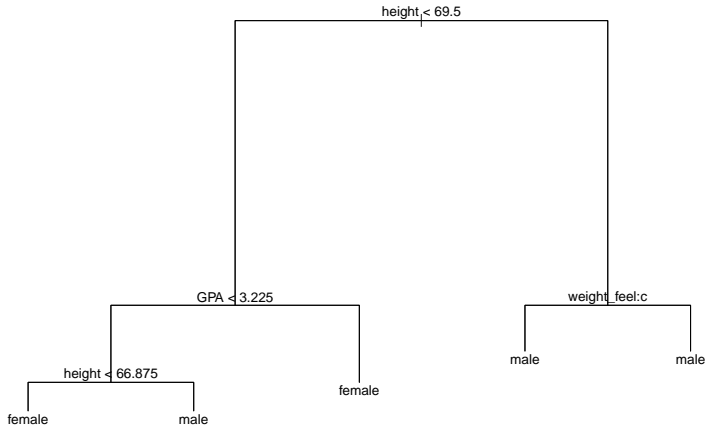
Classification Tree Example

- `mat111survey` data set from the MAT 111 class (Elementary Probability and Statistics) is used
- Variables include:

| height | extra_life | | | —- | — | | | ideal_ht | seat | | sleep | GPA | | fastest | enough_Sleep | | weight_feel | sex | |love_first | diff.ideal.act. |

- ▶ This tree is used to predict the sex of an individual based on the variables of fastest speed ever driven, GPA, height, the amount of sleep the participant got the night before, how the participant feels about their weight, and if the participant believes in love at first sight.
- ▶ The R code for creating a classification tree:

```
set.seed(2020)

m111s.tr <- tree(sex~fastest+GPA+height+sleep+weight_feel+l
                 data=m111survey)
```

height < 69.5

GPA < 3.225

weight_feel:c

height < 66.875

female

male

female

male

female

male

male

- All classification trees have nodes
- The top node is referred to as the *root* node
- Nodes can either split into two *daughter* nodes (or leaves) or they can stop splitting
- A node that does not split any further is known as a *terminal* node
- In this tree example, the majority sex in each terminal node is given under the node. This tree can be used to predict if a new individual is male or female. All we have to do is ask YES or NO questions and follow the nodes to a terminal node.

Here is a more detailed output for the classification tree:

```
node), split, n, deviance, yval, (yprob)
      * denotes terminal node

1) root 70 96.120 female ( 0.5571 0.4429 )
  2) height < 69.5 42 34.450 female ( 0.8571 0.1429 )
    4) GPA < 3.225 18 22.910 female ( 0.6667 0.3333 )
      8) height < 66.875 9  6.279 female ( 0.8889 0.1111 )
      9) height > 66.875 9 12.370 male ( 0.4444 0.5556 ) *
    5) GPA > 3.225 24  0.000 female ( 1.0000 0.0000 ) *
  3) height > 69.5 28 19.070 male ( 0.1071 0.8929 )
    6) weight_feel: 3_overweight 10 12.220 male ( 0.3000 0.
    7) weight_feel: 1_underweight,2_about_right 18  0.000 m
```

We can fine tune the classification tree using the `tree.control` function

```
m111s.tr <- tree(sex~fastest+GPA+height+sleep+weight_feel+1
                 data=m111survey,
                 control = tree.control(
                   nobs = nrow(m111survey),
                   mincut = 5,
                   minsize = 10,
                   mindev = 0.01
                 ))
```