Practical machine Learning- Project Assignment write up

Summary

This document summarizes the approach followed in formulating a machine learning model for predicting the exercise quality/class based upon Human Activity Reading sensor data. Recursive Partitioning (rpart) model and Random Forest (rf) Model are explored and their predictor parameters are determined and errors and misclassifications are studied. Random Forest Model selected for the final choice and predictors are narrowed down to 30 to get satisfactory accuracy levels. Fitted model is cross validated with test sample and final model is used to get 100% accurate answers of the 20 observation given for the project assignment.

Objective

20 record sets of human exercise activity each having a total of 159 different parameter measurements is given. Objective is to assign each of these observations to one of five classes 'A','B','C','D' and 'E'. These observations are available as CSV file 'PML-testing.csv'.

For building the prediction model, another dataset is given which has large set of observations along with the class they belong to. This data is also available in csv file 'PML-training.csv' and forms the basis for generating training set to work on.

Approach

1. Clean data for completeness
    a. Load Data from CSV file to R (both testing and training data)
    b. Make sure the data available in both testing and training are of same dimension(column names match and NA value columns)
    c. How to deal with NA and NAN value columns – (some have partial NAN where we could use k nearest neighbor but for simplicity of this exercise we will be removing for prediction. Only 60 columns out of 160 were retained with full values available.

2. Data slicing
   The PML-training.csv is split into two test set one for training and one for cross validation of the model (one half each on the basis of classe variable which is the outcome so that classe is distributed uniformly in each group))
3. Study the data and do preprocessing if necessary
    a. Plot feature plots to determine if there are evident patterns such as time stamp or serial number (index).

- X is directly related to classe. user_name,time_stamps.. are taken out to make it predictable for any person. Net result : first 7 predictor variables are excluded.

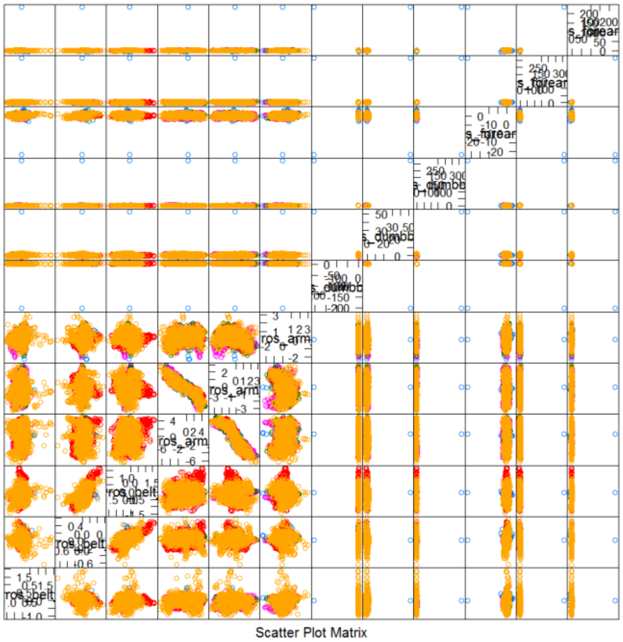


Figure 1: feature plot to identify patterns

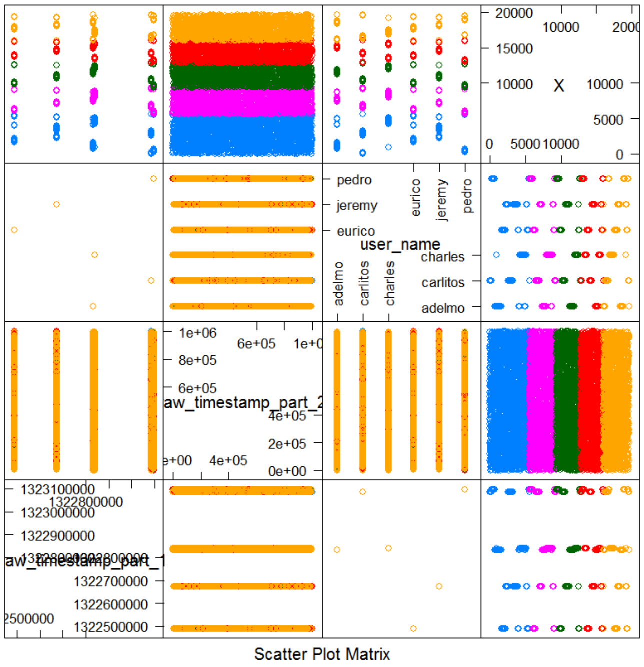


Figure 2: using feature plot it was possible to remove few predictors (like X etc)

4. Choice of Models

   a. Since we are interested in determining the class variable, trees are best choice and recursive partition (“rpart”) and random forecast (“rf”) models are considered.

5. Test Results
   a. Rpart : Train without any preprocessing:

```
CART

9812 samples

  52 predictor

  cp      Accuracy  Kappa   Accuracy SD  Kappa SD

  0.0357  0.520     0.3765  0.0192       0.0268
```

Estimated Miss Classification error

Misclassification Rate =1-sum(diag(tab))/sum(tab)

0.338

Conclusion: with default settings rpart model seems not satisfactory

Applying Random Forest Model

Test results with Random Forest

```
modFit4 <- train(training1$classe ~., data=training1,
method="rf",trControl=trainControl(method="oob",repeats=10))

9812 samples

  52 predictor

   5 classes: 'A', 'B', 'C', 'D', 'E'

Resampling results across tuning parameters:

  mtry  Accuracy  Kappa

   2    0.988     0.984
  27    0.988     0.985
  52    0.978     0.972
```

Accuracy was used to select the optimal model using the largest value.

The final value used for the model was mtry = 27.

This looks fantastic.

Let us run the prediction on training to see the in estimated Sample error and also the out of sample errors:

```
  pred4
       A    B    C    D    E
  A 2785    0    0    0    0
  B    0 1907    0    0    0
  C    0    0 1716    0    0
  D    0    0    0 1633    0
  E    0    0    0    0 1771
```

Estimated error 0%

We will construct the table of actual class versus the predicted class

Cross validate the accuracy and miss classification rate by running the prediction on testing data (other half of training data partition)

```
       A    B    C    D    E
  A 2781   11    1    0    2
  B   26 1851   11    2    0
  C    0   12 1690    4    0
  D    1    0   22 1560    0
  E    0    2    5    8 1821
```

Numbers in the diagonal show the correctly predicted classe and numbers in the other rows show the incorrect predictions. Let us calculated the miss calculation rate:

misclassification Rate: > 1-sum(diag(tab4))/sum(tab4))

```
[1] 0.01090724
```

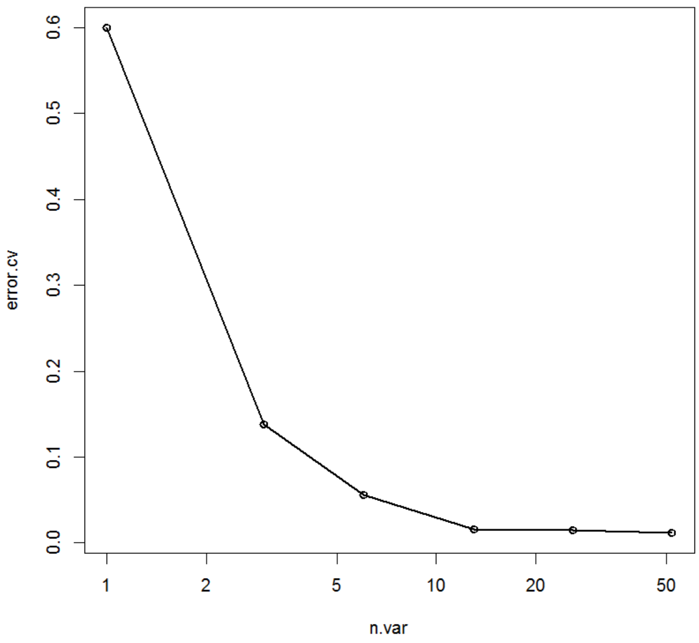
Cross validation: Actual Error 0.0109 is close to (1-0.988) we estimated with training sample

Hopefully we are not doing over fitting. We can check this by running rfcv function on the random forest mod

```
rfcv(training1[,-53], training1[,53], cv.fold=5, scale="log", step=0.5,mtry=function(p)
max(1, floor(sqrt(p))), recursive=FALSE)
```

```
$n.var

[1] 52 26 13  6  3  1

$error.cv

      52         26         13          6          3          1

0.01243375 0.01457399 0.01722381 0.05839788 0.15409702 0.60079494
```

We can also plot the error importance plot



It seems like around 30 predictors contribute to most of the prediction model.

```
rf1 <- randomForest(classe ~ ., data=training1, ntree=500,keep.forest=FALSE,
importance=TRUE)

 importance(rf1)
```

```
                         A        B        C        D        E      MeanDecreaseAccuracy
roll_belt           37.44369 44.66865 43.08343 45.04062 38.30664              52.74945
pitch_belt          28.83358 46.64300 35.90937 29.99252 29.39415              46.81836

……….                ………….
```

So instead of including all predictors we will try to use top 30 predictors for training which results in

```
modFit30 <- train(classe ~
roll_belt+pitch_belt+yaw_belt+total_accel_belt+gyros_belt_x+gyros_belt_y+gyros_belt_z+acc
el_belt_x+accel_belt_y+accel_belt_z+magnet_belt_x+magnet_belt_y+magnet_belt_z+roll_arm+pi
tch_arm+yaw_arm+total_accel_arm+gyros_arm_x+gyros_arm_y+gyros_arm_z+accel_arm_x+accel_arm
_y+accel_arm_z+magnet_arm_x+magnet_arm_y+magnet_arm_z+roll_dumbbell+pitch_dumbbell+yaw_du
mbbell+total_accel_dumbbell, data=training1,
method="rf",trControl=trainControl(method="oob",repeats=5))
```

check our prediction error and miss classification as before.

Resampling results across tuning parameters:

```
  mtry  Accuracy  Kappa

   2    0.973     0.966

  16    0.974     0.967

  30    0.968     0.959
```

The final value used for the model was mtry = 16.

Just a little over a percent loss of accuracy and looks more promising. ***Running on training data estimated error 0%***

Cross validating with test data yields

```
       A    B    C    D    E

  A 2755   13    7   14    6

  B   25 1839   19    4    3
  C    9   21 1661   15    0
  D   10    1   32 1535    5
  E   10   12    3   11 1800
```

For a misclassification rate of 0.0224261 which cross validates with predicted accuracy of 0.974. Note this also compares well with using all predictors 0.0108053.

Final Answer for the 20 observations:

Now we are ready for testing our final model with the real data

```
pred5 <- predict(modFit4,newdata=testing1)

> pred5

 [1] B A B A A E D B A A B C B A E E A B B B
```

These answers were submitted and found to match all 20 cases.