Kinematic Features Classification

Andrew Boschee

4/26/2020

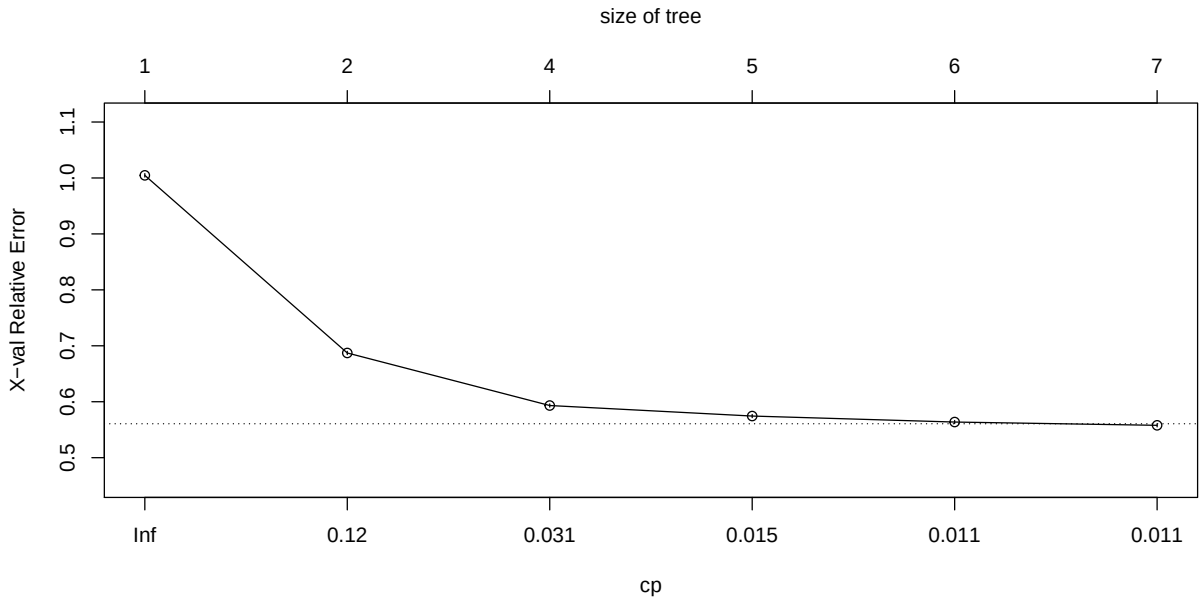
Introduction

Among the three objectives of this project, I believe that the practicality of certain classifications are much more legitimate than others. To begin with, given the type of independent variables in the dataset, I believe that the cursive vs print classification is a reasonable request when it comes to the duration of contact, angles, and other attributes of movement. The other two dependent variables I am not so sure about. While letters are a legitimate object to classify from image recognition, I am not sure about the legitimacy from the data given. Finally, when looking at legitimacy of them as a combined response variable, there will most likely be a struggle.

Group Classification - Logistic Regression, Decision Tree, KNN

With a higher level of confidence in classifying the difference between cursive and print, the first thing I would like to find is variable importance. I expect duration and possibly the angle to be important for this dependent variable. We can see from the plot that error rate flattens out aroud the fourth split. I'm a little surprised that we only come away with about 50% accuracy.

Decision Tree



Regarding group as a response variable, we can see the most important variables in the tree diagram below as well as the table containing all predictor variables. While the predictions were not impressive, it is understandable why these variables such as duration and jerk are seen as important when comparing cursive writing to print.

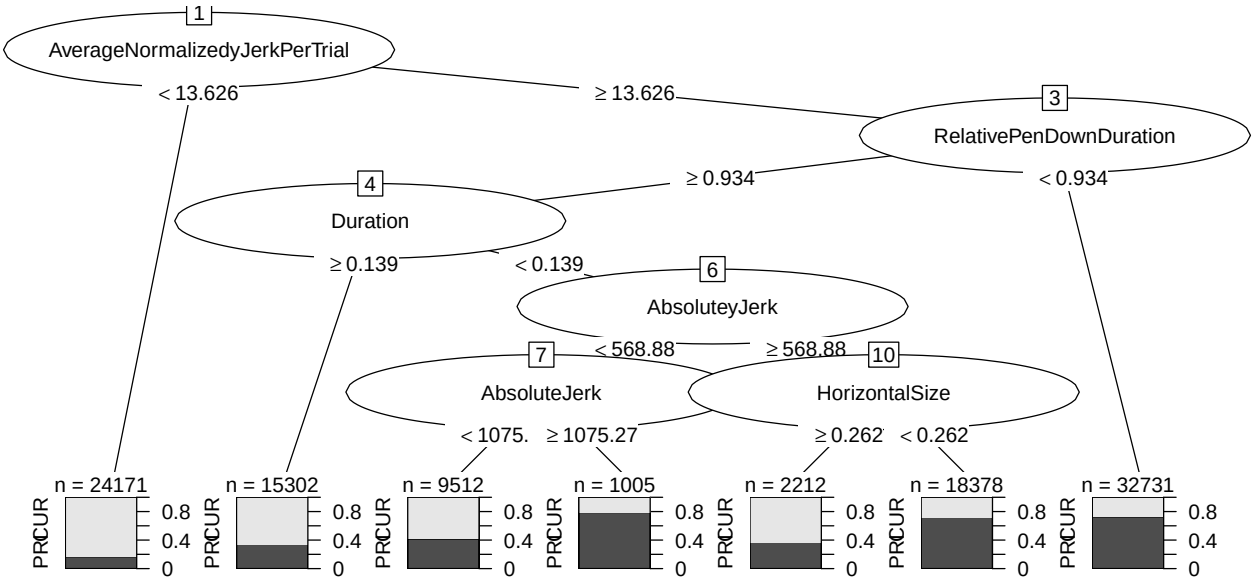
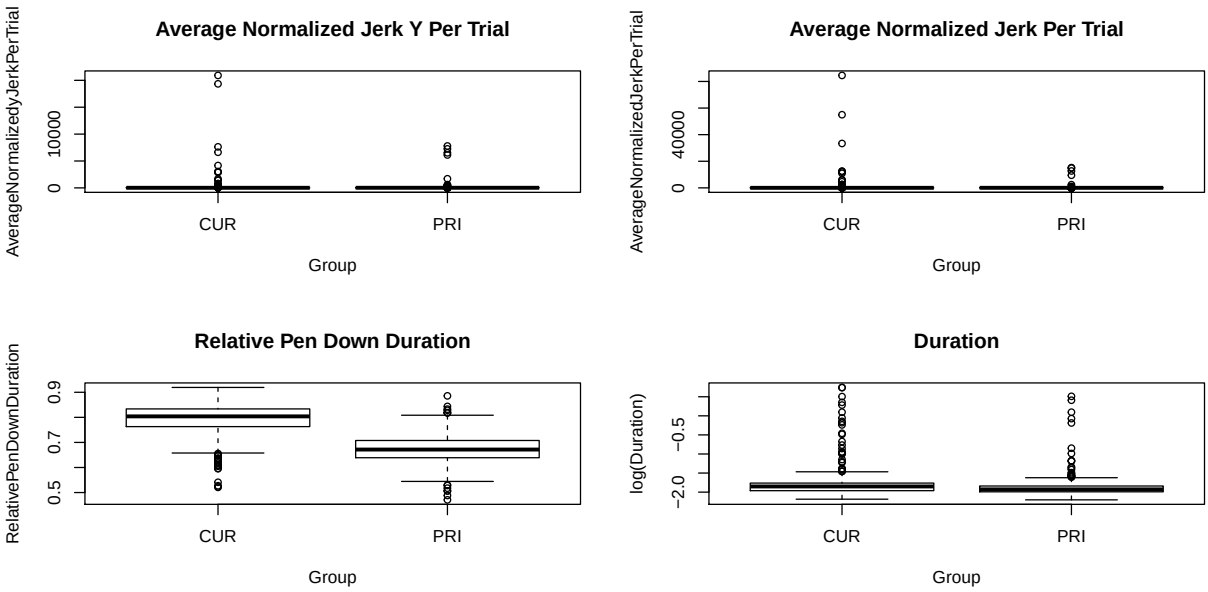


Table 1: Variable Importance with Group as Response

AverageNormalizedyJerkPerTrial	7054.9079261
AverageNormalizedJerkPerTrial	4606.9532376
Duration	1801.3815779
RelativePenDownDuration	1681.0929877
AveragePenPressure	1261.5259529
NormalizedJerk	1029.4731740
NormalizedyJerk	1025.7912470
NumberOfPeakAccelerationPoints	812.7944435
AbsoluteyJerk	744.8574796
AbsoluteJerk	642.2037073
Roadlength	578.1504301
HorizontalSize	490.9543528
AbsoluteSize	478.7218303
AverageAbsoluteVelocity	266.4368691
StraightnessError	5.8033233
PeakVerticalVelocity	1.3098015
VerticalSize	1.0915013
RelativeInitialSlant	0.5837498
PeakVerticalAcceleration	0.2918749



Logistic Regression - All Features Against Only Top Three From Variable Importance

To see how well a common logistic regression model performs with all features in comparison to the top variables from the variable importance the prior step, there are two confusion matrices and table comparing the accuracy.

```
##
##         CUR   PRI
##   CUR 36311 15318
##   PRI 19599 32083
```

```
##
##         CUR   PRI
##   CUR 30430 21199
##   PRI 21447 30235
```

Table 2: Accuracy of Predictions

All Features	Top 3 Features
0.6620205	0.5872076

Subject Classification - Decision Tree

Moving on to the subject variable, it is interesting to compare the similarities and differences of variable importance against the group variable. Both models look at Duration and AverageNormalizedJerkPerTrial. However, the top predictor for subject is AveragePenPressure which makes sense since it's easy to see that pen pressure can easily vary between people.

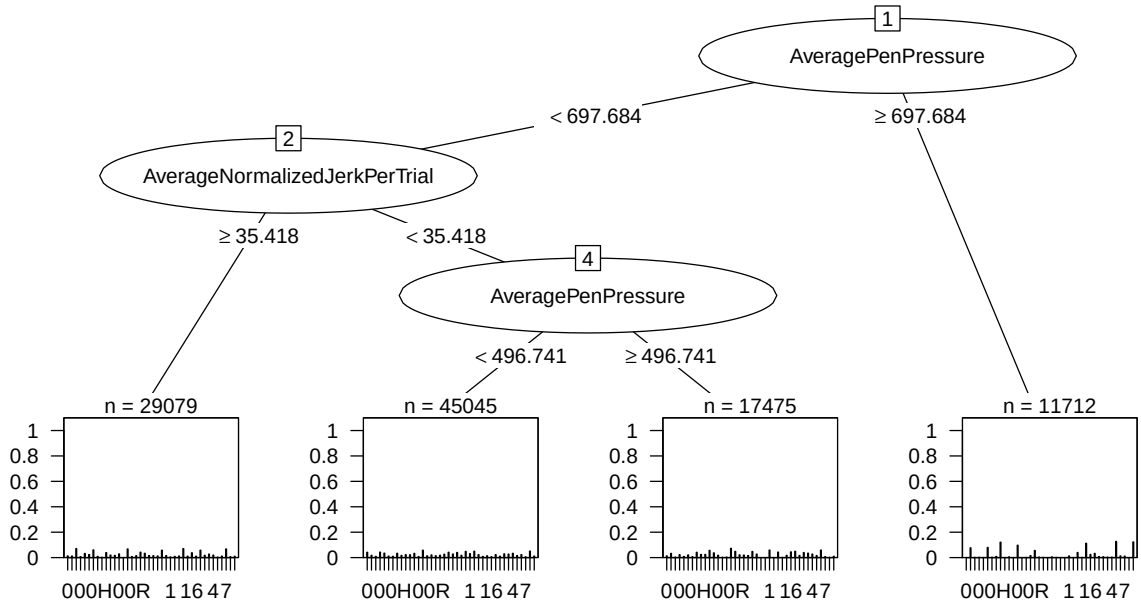
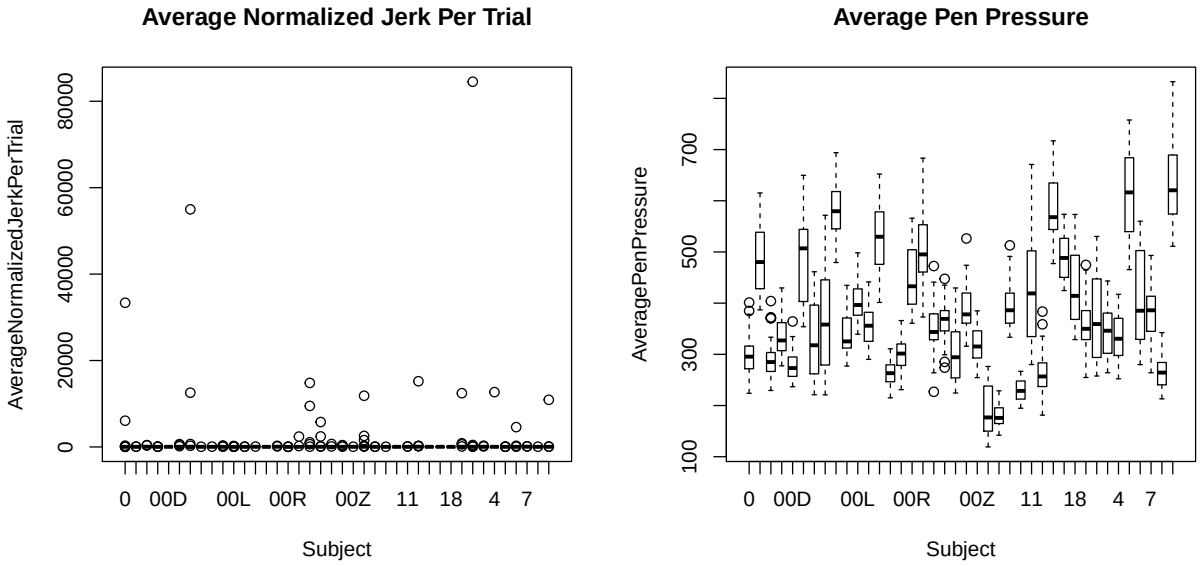


Table 3: Variable Importance Subject as Response

AveragePenPressure	1141.5066041
AverageNormalizedJerkPerTrial	698.8292099
AverageNormalizedyJerkPerTrial	492.0147838
NormalizedJerk	23.8787925
NormalizedyJerk	23.4704026
Duration	20.9239721
NumberOfPeakAccelerationPoints	15.9031797
PeakVerticalAcceleration	0.0887248
Roadlength	0.0355237
AbsoluteSize	0.0177618



Condition Classification - Decision Tree

Lastly regarding feature importance, we will look at the Condition response variable. With the given data and models being used at the moment this isn't ideal. An easy thing to notice hear is how all three nodes are regarding some sense of the average normalized jerk per trial. This is concerning and again makes me want to look more towards princial components.

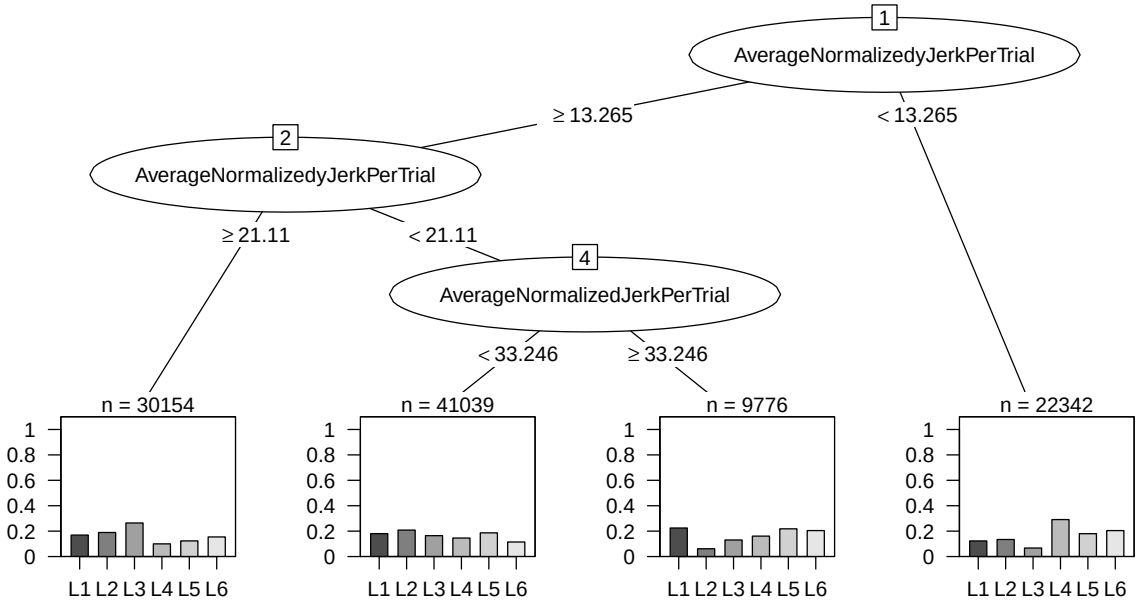
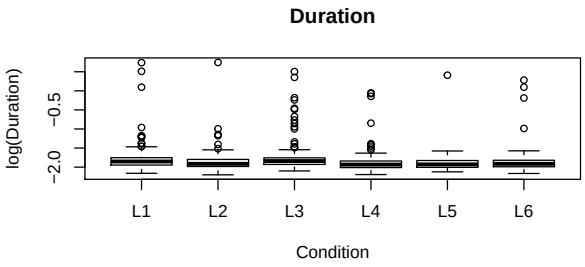
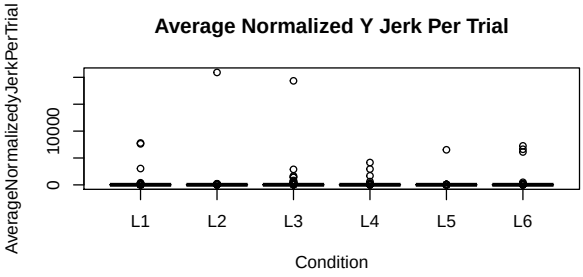
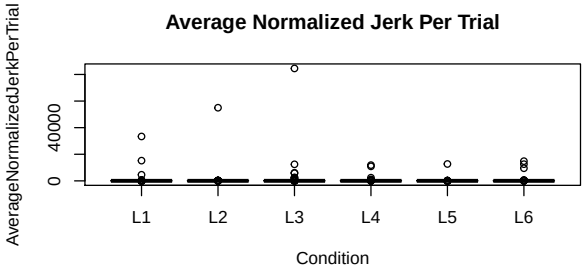


Table 4: Variable Importance Condition as Response

AverageNormalizedyJerkPerTrial	1321.4150482
AverageNormalizedJerkPerTrial	1111.9048960
Duration	14.0817288
NormalizedyJerk	13.9152328
NormalizedJerk	12.1021537
NumberOfPeakAccelerationPoints	9.7752400
AbsoluteJerk	0.4125625
AverageAbsoluteVelocity	0.2200333
RelativeInitialSlant	0.1929924
PeakVerticalAcceleration	0.0414879

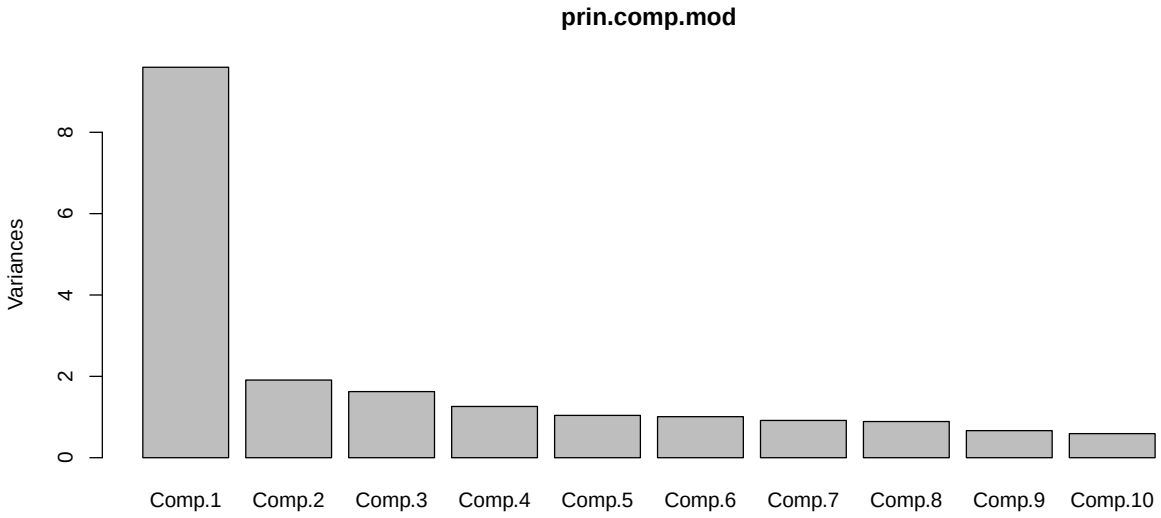






Methods Using PCA

Moving on to principal comonents, I kept the use down to 10 or less components and examined the accuracy using various numbers of principal components. My first instinct was to keep at down around three variables, but results coming up shortly changed my opinion and went up to using ten in most cases.



Linear Discriminant Analysis

The starting point of making predictions with the principal components began with linear discriminant analysis. The table below shows the accuracy rate for each dependent variable individually and the 'Joint' variable which is having every unique combination of those three variable as the reponse.

Table 5: LDA with PCA Classification Summary

Group	Subject	Condition	Joint
0.8708333	0.3881944	0.3097222	0.2819444

KNN

With poor performance so far with previous modeling methods, I am trying a very simple method using K-Nearest Neighbors with various arguments for K and the number of Principal Components used. With a 75/25 train/test split, there are some surprising and concerning results from these models.

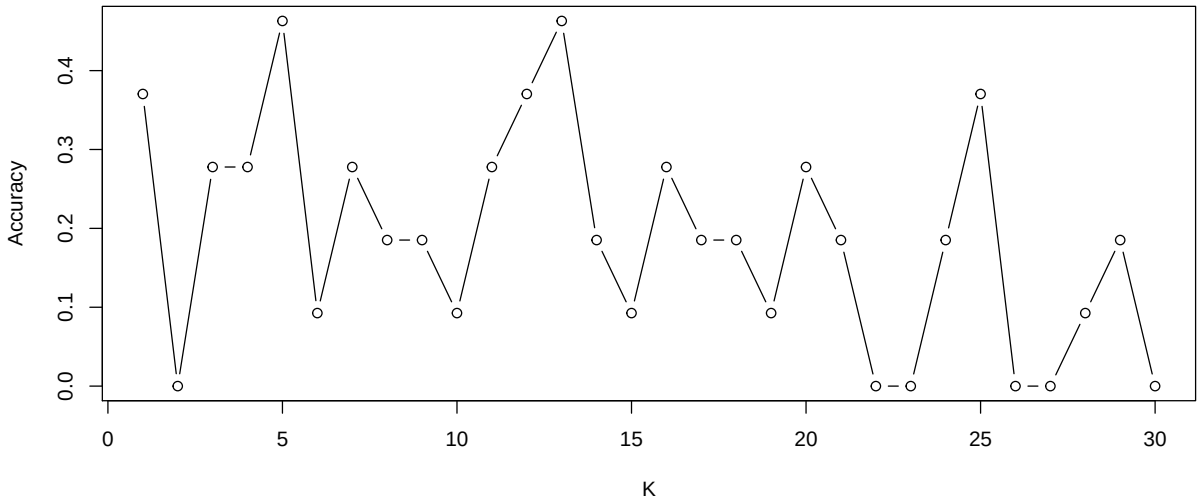
We begin with the Group variable that I feel most confident in and get decent accuracy in the 80-90 percent range with just the first three principal components.

Table 6: KNN Accuracy Comparison - Group Variable

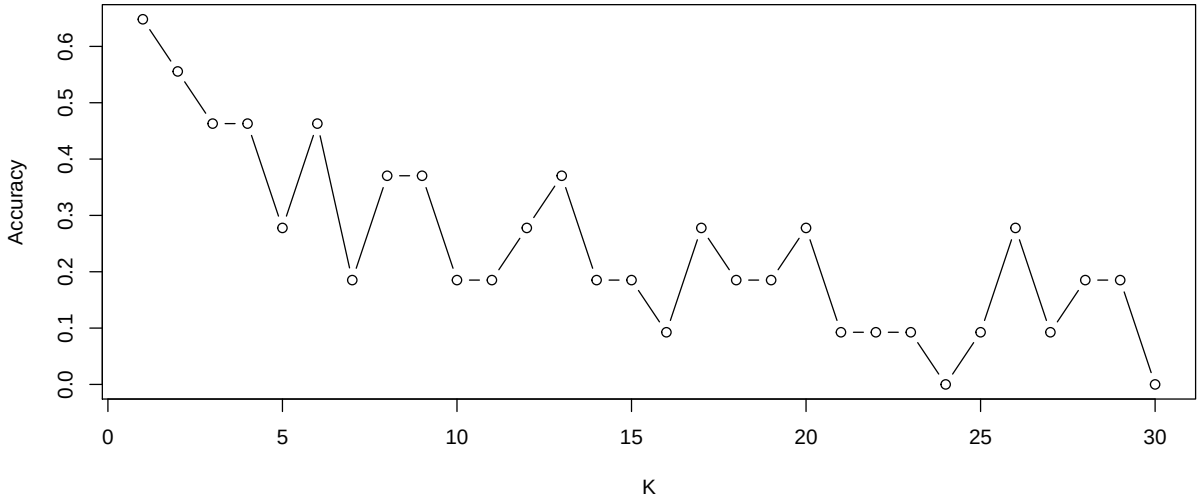
k=3	k=5	k=5	k=10
0.8138889	0.8583333	0.8333333	0.8638889

Next, for quicker analysis, I created a function to iterate through the model 30 times using varying numbers of principal components. The argument given determined the range of principal components with the plots below showing the drastic variance of accuracy when changing the value of K and number of principal components.

Three Principal Components Used

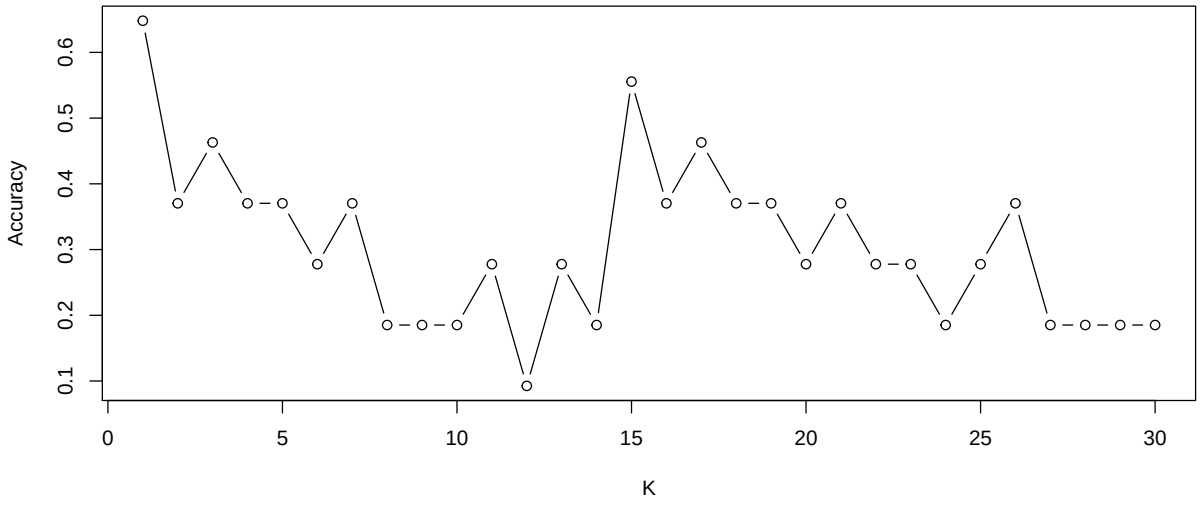


Eight Principal Components Used



Ten Principal Components Used

This is where we actually get an accuracy above 40% when K is between 15 and 20 using the first ten principal components.



Mclust and PCA

The final approach is using ten principal components with Mclustcv() function. This method gives similar output to LDA from earlier ending up with accuracy between 25-30%.

Using Ten Principal Components

Table 7: Mclust Using Principal Components - 10 Fold CV

x
0.2534722

Hierarchical Clustering (Complete Method) Vs Kmeans Clustering

The last phase of comparing reasonability of being able to predict each dependent variable, I used hclust() and kmeans() on each variable and compared them in tables. hclust() was not very impressive at all when looking at any variables. While Kmeans was nothing special, it did significantally better when it came to identifying cursive vs print. Surprisingly, hclust only had 3 samples in the second cluster with the remaining 1397 in the first cluster. This seems like one of the better ways to look at separating condition variable with given data.

Hierarchical Clustering

```
##
## hclustClustersGroup CUR PRI
##                   1 717 720
##                   2   3   0
```

```
##
## hclustClustersCondition  L1  L2  L3  L4  L5  L6
##                       1 238 239 237 238 239 237
##                       2   1   0   0   0   0   0
##                       3   1   0   1   2   1   2
##                       4   0   0   1   0   0   0
##                       5   0   1   0   0   0   0
##                       6   0   0   1   0   0   1
```

KMeans Clustering

```
##
## kmClustersGroup CUR PRI
##               1  74 538
##               2 646 182
```

```
##
## kmClustersCondition L1 L2 L3 L4 L5 L6
##                   1 41 13 28 33 51 24
##                   2 41 58 40 43 20 36
##                   3 50 82 83 28 68 53
##                   4 48 18 38 69 52 49
##                   5 58 68 49 65 48 76
##                   6  2  1  2  2  1  2
```

Kmeans vs Hclust Comparison

```
##                hclustClustersGroup
## kmClustersGroup   1   2
##               1 609   3
##               2 828   0
```

```
##                     hclustClustersCondition
## kmClustersCondition   1   2   3   4   5   6
##                   1 189   0   0   0   0   1
##                   2 238   0   0   0   0   0
##                   3 363   0   0   0   0   1
##                   4 274   0   0   0   0   0
##                   5 364   0   0   0   0   0
##                   6   0   1   7   1   1   0
```

Conclusion

The main goal here was to find whether it is practical or not to classify a joint dependent variable composed of the variables Group, Subject, and Condition. After observing outcomes from models given above, while the accuracy is greater than a random gues of the 480 possible objects, it doesn't seem like a task that can be completed with a high level of confidence. My main concern after all is said and done remains with the condition variable. I would further investigate the clustering of the phrases to see more in depth on consistency across subjects making it more feasible to classify those phrases.

The similarity of variable importance across the response variables gives a slight glimmer of hope that there may be potential to increase the accuracy. Further investigation with the use of Support Vector methods and Random Forest modeling as well as continued modification of principal components, I could see potential improvement.

Outside Resources - cran.r-project.org, edureka.co, statmethods.net, A Handbook of Statistical Analysis Using R, Introduction to Statistical Learning