FINAL ASSIGNMENT - UNSUPERVISED MACHINE LEARNING

Project Background & Objective

Using devices such as Jawbone Up, Nike FuelBand, and Fitbit it is now possible to collect a large amount of data about personal activity relatively inexpensively. These type of devices are part of the quantified self movement – a group of enthusiasts who take measurements about themselves regularly to improve their health or to find patterns in their activity.

One thing that people regularly do is quantify how much of a particular activity they do, but they rarely quantify how well they do it. In this project, my goal was to use data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants. They were asked to perform lifts correctly and incorrectly in 5 different ways.

The goal of the exercise of the exercise is to cluster the contents of the dataset into clusters to see if I can identify if there any clear distinctions amongst the instances in the data set which may help me highlight distinct categories within in it.

Given the quality of the dataset I was also able to run a separate, parallel supervised learning task to see if the results of my unsupervised learning task were consistent with a supervised learning task designed to bucket partcipants into discrete categories – specifically the classifications of the weightlifters in a test set based off of data provided as part of a training set after dividing the dataset appropriately.

About the Dataset

The underlying dataset can be found within the UC-Irvine Machine Learning Repository. More information is available from the website here:

https://archive.ics.uci.edu/dataset/273/weight+lifting+exercises+monitored+with+inertial+measurement+units

Weightlifters in the dataset were categorized into five qualitative categories, A-E but not in any specific ordinal order (i.e. A isn't necessarily better than E

Exploratory Data Analysis & Pre-Processing

The dataset had a large number of columns which displayed a high degree of co-linearity – representing possible redundancy and unnecessary dimensionality - as well as some instances of typos and "N/A" data fields. Because of this I had to clean and preprocess the training and testing datasets.

This involved handling many missing values, encoding categorical variables, and scaling numerical features where necessary to make the dataset more efficient to process. There were a number of columns where data points were missing – essentially all of them. I felt that the amount of data missing for those columns was too high to safely interpolate values based on the limited data available for the isolated number of rows that did have data. So, I chose to exclude those from the analysis where necessary.

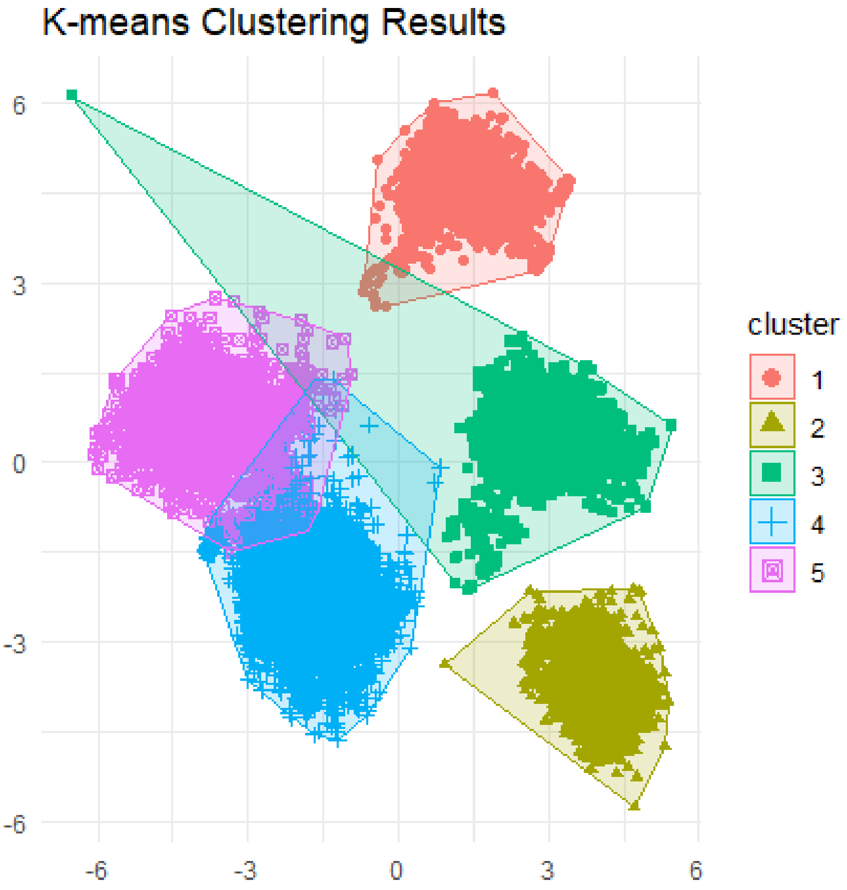
I experimented with several combinations of feature columns using feature engineering. I was able to successfully condense the number of features from 159 to 53 which greatly enhanced the computational speed when the model was executed in R Studio using the R programming language and later in Keras using the Python programming language.

Model Evaluation

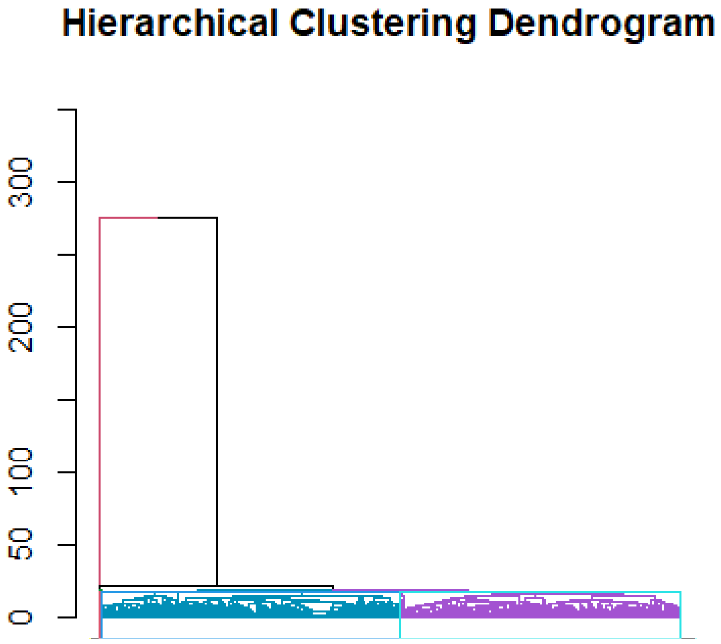
I used three different approaches to conduct the unsupervised learning project: k-means, hierarchical classification and DB-Scan. Each have benefits and drawbacks and although I did not run an ensemble approach per se, I ran them successively to see if different approaches allowed me to tease out different bits of information about the dataset. The unsupervised models I applied are best viewed in aggregate. The K-means approach was by far the most effective as it yielded clear results with a comparatively light computational load. Hierarchical clustering was the least useful on my dataset as the resulting dendogram was so complex that the incremental insights provided were outweighed by the computational intensity and relatively dense results making the plot difficult to decipher in more detail.

Key Findings & Insights

The results of the K-means analysis were clear and concise, easily bucketing the clusters into five distinct groupings with one unique outlier that despite my best efforts I was unable to explain why it had been bucketed in a certain set. Regardless, the outcomes are easy to understand visually.

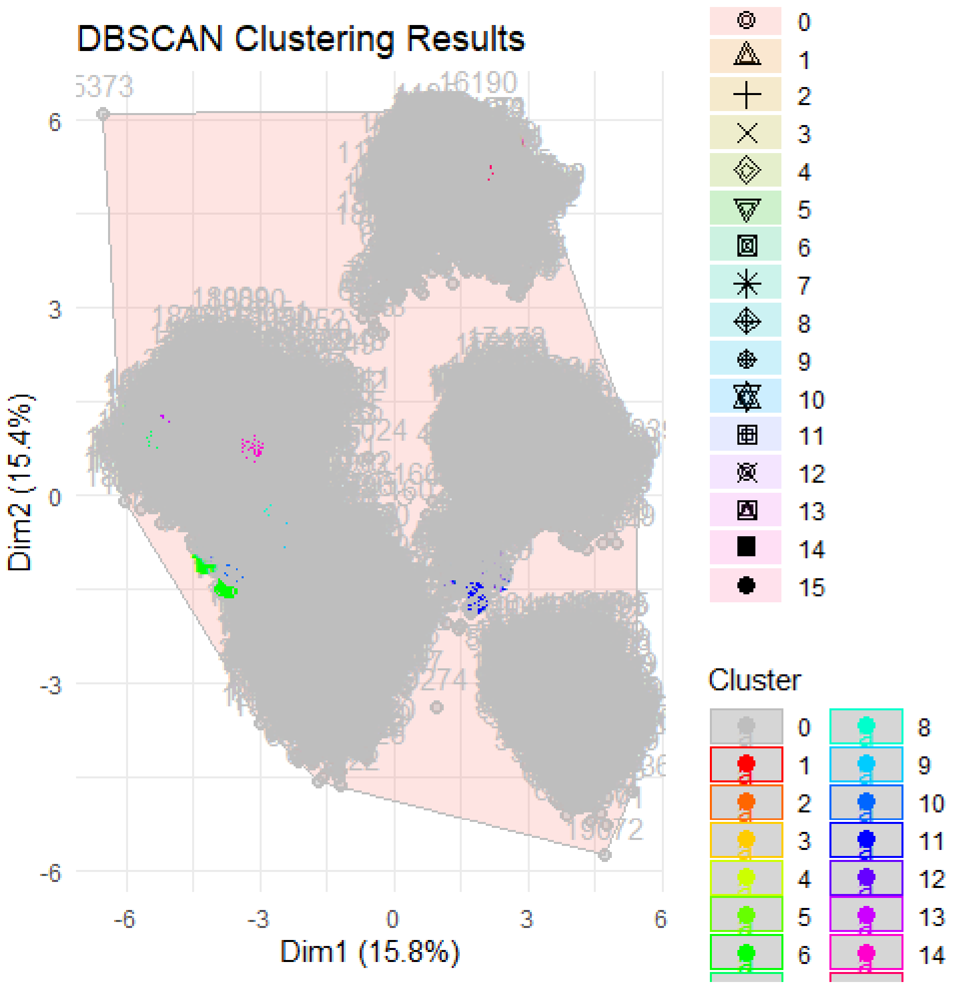


As explained earlier the hierarchical clustering approach had limited efficacy in this particular case.



Key Findings & Insights (continued)

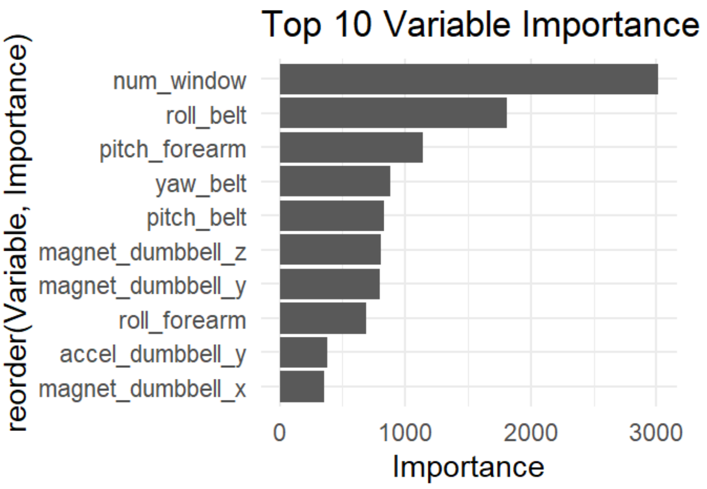
Applying DB Scan I was able to mimic the K-Means results pretty closely as there were five distinct agglomerations of data. I think the value of DB-Scan in this case is that this methodology does not require the user to specify the number of clusters beforehand yet arrived at a broadly similar outcome, supporting my assertion that there we five distinct groups which was subsequently reinforced with applying a supervised learning technique to confirm my assumptions. DB Scan was able to draw out more distinctive clusters although one can see clearly in the chart that while this clusters were unique, they were not necessarily significant and they were subsumed within the five broader buckets found when using the K-means approach.

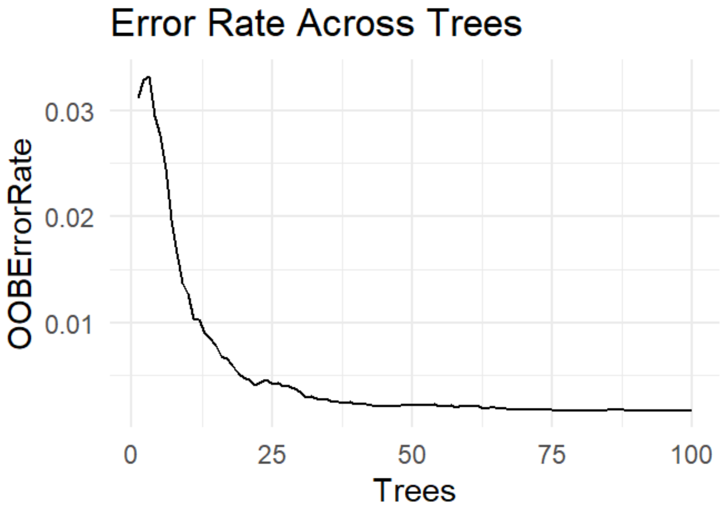


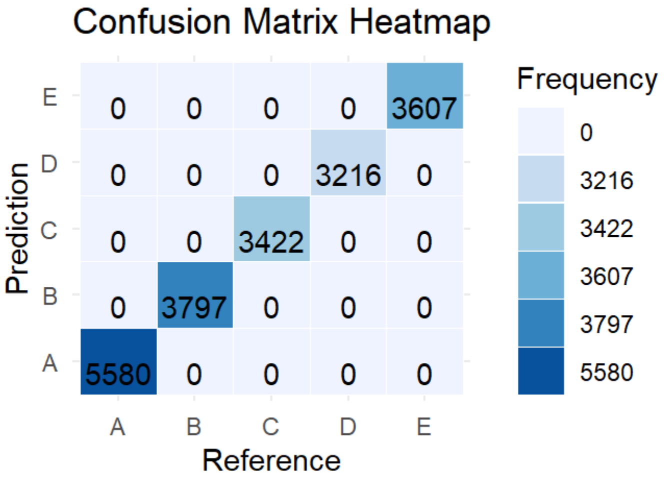
To cross-reference the results of the unsupervised learning approach, I conducted a random forest analysis to see if the results were internally consistent with the k-means and DB-Scan approaches. K-means requires the user to provide a number of clusters before running the analysis. On a hunch I chose the number 5 which as you can see below is supported by the X axis on the confusion matrix heatmap which has five corresponding categories labelled A through E.

After training the supervised learning model using cross-validation, I evaluated its performance using various metrics such as accuracy, precision, recall, and F1-score. The random forest method appears to have been highly effective in accurately predicting the results of the test set. The error rate improvement appeared to cease at around 27 trees. The expected out-of-sample error estimate: 0.001121253 I wanted to emphasize some degrees of explainability in the results, which is why I chose to move ahead with more simplistic machine learning models over models with lots of hidden layers. As I was able to get highly accurate results while also using a simpler approach I chose to keep it simple over making it complex for the sake of it. Specifically I was able to extract the top 10 most important variables which influenced the correct classifications.As expected, there was notable improvement in the error rate when I added more trees but there was a rapid drop-off from approximately tree 30 onwards.

Key Findings & Insights (continued)







Prediction	A	B	C	D	E
E	0	0	0	0	3607
D	0	0	0	3216	0
C	0	0	3422	0	0
B	0	3797	0	0	0
A	5580	0	0	0	0

More details on my results and the source code can be found here: https://rpubs.com/nhartley1986/1161664

Suggested Next Steps

The initial dataset had some challenges, specifically related to over-representation of certain classes. The outputs of the model could be made more robust by ensuring that the initial dataset was a) more diverse; b) had more samples; and c) fewer co-linear features. In this sense the "shape" of the dataset was quite wide and I think it could have benefited from additional rows of data at the very least.