

Machine Learning - Assignment 2 (Spotify)

Conor Heffron (23211267)

Load R libraries

```
library(tidyverse)

-- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
v dplyr     1.1.3     v readr     2.1.4
v forcats   1.0.0     v stringr   1.5.0
v ggplot2   3.4.4     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
v purrr     1.0.2

-- Conflicts ----- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become non-conflicting

library(readr)
library(dplyr)
library(ggplot2)
library(dendextend)

-----
Welcome to dendextend version 1.17.1
Type citation('dendextend') for how to cite the package.

Type browseVignettes(package = 'dendextend') for the package vignette.
The github page is: https://github.com/talgalili/dendextend/
Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
```

You may ask questions at stackoverflow, use the r and dendextend tags:
<https://stackoverflow.com/questions/tagged/dendextend>

To suppress this message use: suppressPackageStartupMessages(library(dendextend))

Attaching package: 'dendextend'

The following object is masked from 'package:stats':

cutree

```
library(formatR)
library(knitr)
opts_chunk$set(tidy.opts=list(width.cutoff=60), tidy=TRUE)
```

Load Spotify Data

```
path = "/Users/conorheffron/Library/CloudStorage/GoogleDrive-conor.heffron@ucdconnect.ie/Music/Spotify/Spotify Data/2023-01-26/spotify_23211267.csv"
spotify_23211267 <- read_csv(paste(path, "spotify_23211267.csv",
sep = "/"), na = "NA")
```

Rows: 21812 Columns: 13

-- Column specification -----

Delimiter: ","

chr (1): playlist_genre

dbl (12): danceability, energy, key, loudness, mode, speechiness, acousticne...

i Use `spec()` to retrieve the full column specification for this data.

i Specify the column types or set `show_col_types = FALSE` to quiet this message.

```
# View(spotify_23211267)
```

Show Data Dimensions, Structure, Summary

```
dim(spotify_23211267)

[1] 21812     13

str(spotify_23211267)

spc_tbl_ [21,812 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
$ danceability    : num [1:21812] 0.533 0.66 0.812 0.654 0.605 ...
$ energy          : num [1:21812] 0.703 0.947 0.666 0.887 0.903 0.562 ...
$ key             : num [1:21812] 2 0 1 1 7 0 5 9 5 11 ...
$ loudness        : num [1:21812] -6.36 -7.58 -7.1 -2 -4.54 ...
$ mode            : num [1:21812] 1 1 1 1 1 0 1 1 0 1 ...
$ speechiness     : num [1:21812] 0.0423 0.0707 0.0485 0.115 0.0637 ...
$ acousticness    : num [1:21812] 0.184 0.0367 0.000341 0.0588 0.188 0.08 ...
$ instrumentalness: num [1:21812] 0 0.0316 0.862 0.000103 0.00814 0 0 0.000569 ...
$ liveness         : num [1:21812] 0.101 0.419 0.12 0.0849 0.0989 0.0806 0.218 0.26 0.153 ...
$ valence          : num [1:21812] 0.612 0.783 0.178 0.691 0.55 0.683 0.673 0.473 0.695 0.647 ...
$ tempo            : num [1:21812] 129 123 125 135 120 ...
$ duration_ms      : num [1:21812] 186093 201110 485760 270222 307653 ...
$ playlist_genre   : chr [1:21812] "rock" "edm" "edm" "edm" ...
- attr(*, "spec")=
.. cols(
..   danceability = col_double(),
..   energy = col_double(),
..   key = col_double(),
..   loudness = col_double(),
..   mode = col_double(),
..   speechiness = col_double(),
..   acousticness = col_double(),
..   instrumentalness = col_double(),
..   liveness = col_double(),
..   valence = col_double(),
..   tempo = col_double(),
..   duration_ms = col_double(),
..   playlist_genre = col_character()
.. )
- attr(*, "problems")=<externalptr>
```

```
summary(spotify_23211267)
```

danceability	energy	key	loudness
Min. :0.0000	Min. :0.000175	Min. : 0.000	Min. :-46.448
1st Qu.:0.5590	1st Qu.:0.613000	1st Qu.: 2.000	1st Qu.: -7.836
Median :0.6670	Median :0.745000	Median : 6.000	Median : -5.895
Mean :0.6514	Mean :0.720189	Mean : 5.359	Mean : -6.489
3rd Qu.:0.7580	3rd Qu.:0.856000	3rd Qu.: 8.000	3rd Qu.: -4.473
Max. :0.9830	Max. :1.000000	Max. :11.000	Max. : 1.275
mode	speechiness	acousticness	instrumentalness
Min. :0.0000	Min. :0.0000	Min. :0.0000	Min. :0.000000
1st Qu.:0.0000	1st Qu.:0.0408	1st Qu.:0.0116	1st Qu.:0.000000
Median :1.0000	Median :0.0614	Median :0.0670	Median :0.000022
Mean :0.5788	Mean :0.1048	Mean :0.1582	Mean :0.096463
3rd Qu.:1.0000	3rd Qu.:0.1270	3rd Qu.:0.2230	3rd Qu.:0.007622
Max. :1.0000	Max. :0.8770	Max. :0.9940	Max. :0.994000
liveness	valence	tempo	duration_ms
Min. :0.0000	Min. :0.0000	Min. : 0.0	Min. : 4000
1st Qu.:0.0931	1st Qu.:0.3240	1st Qu.:101.0	1st Qu.:186619
Median :0.1300	Median :0.5060	Median :124.0	Median :213507
Mean :0.1943	Mean :0.5055	Mean :122.2	Mean :223585
3rd Qu.:0.2560	3rd Qu.:0.6890	3rd Qu.:134.5	3rd Qu.:249590
Max. :0.9960	Max. :0.9910	Max. :220.3	Max. :517810
playlist_genre			
Length:21812			
Class :character			
Mode :character			

Hierarchical Cluster via Euclidean Distance of Spotify Data

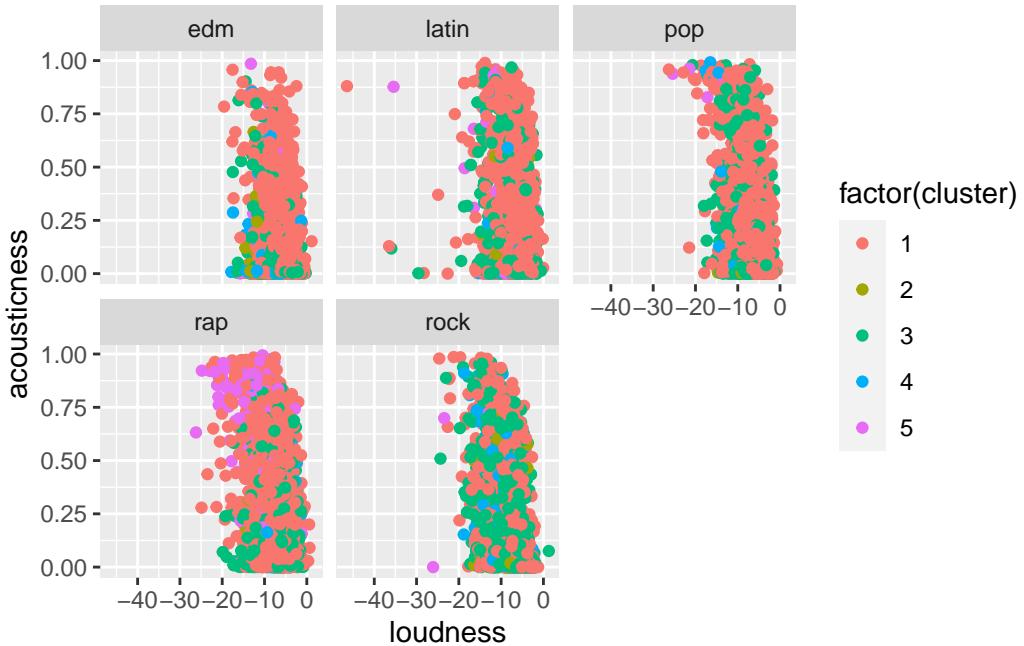
```
hc_spotify <- hclust(dist(spotify_23211267, method = "euclidean"),  
method = "average")
```

```
Warning in dist(spotify_23211267, method = "euclidean"): NAs introduced by  
coercion
```

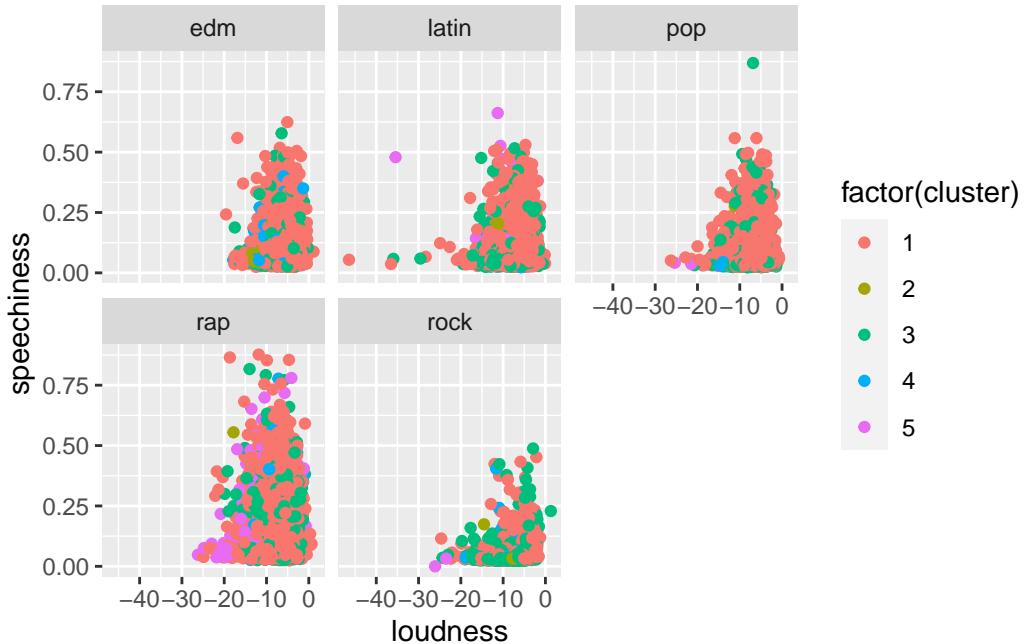
Segment Spotify Data into 5 clusters / groups

```
grp5 <- cutree(hc_spotify, k = 5)
segmented_spotify <- mutate(spotify_23211267, cluster = grp5)

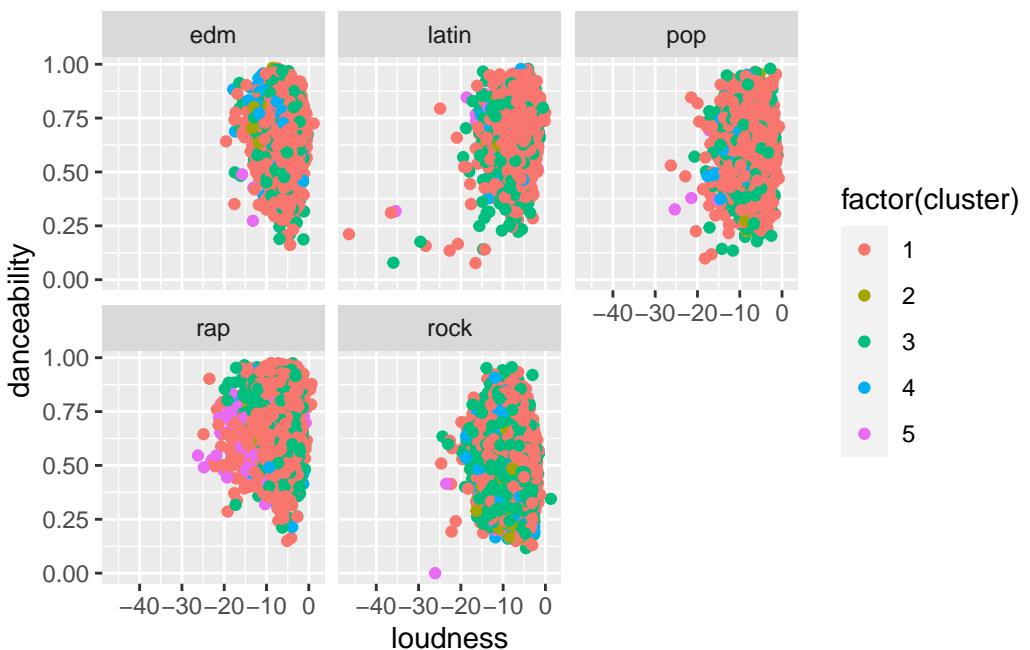
# Plots
ggplot(segmented_spotify, aes(x = loudness, y = acousticness,
                               color = factor(cluster))) + geom_point() + facet_wrap(~playlist_genre)
```



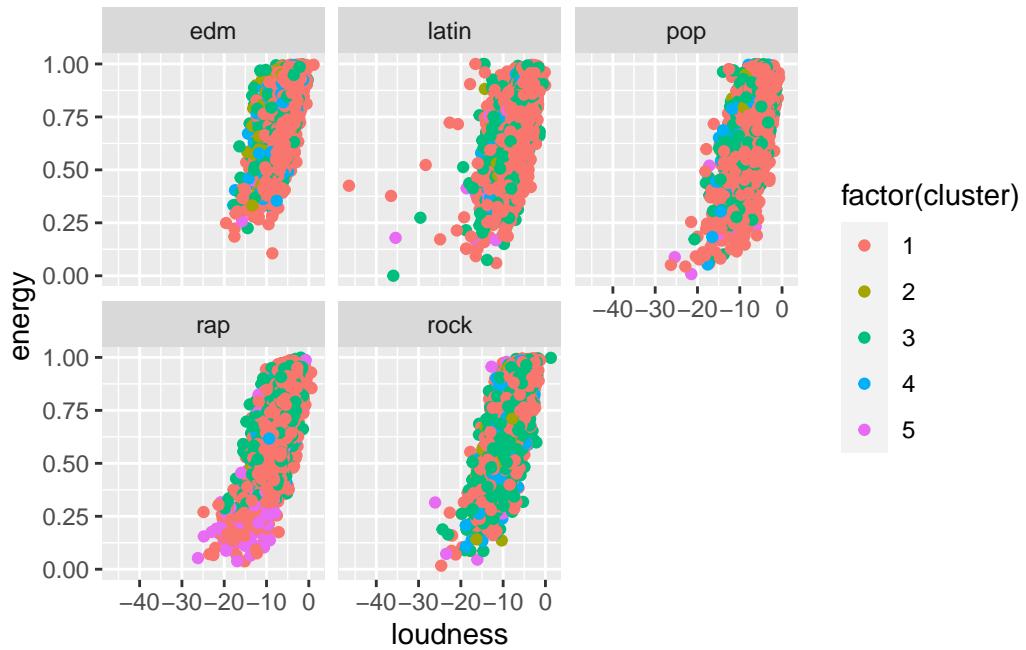
```
ggplot(segmented_spotify, aes(x = loudness, y = speechiness,
                               color = factor(cluster))) + geom_point() + facet_wrap(~playlist_genre)
```



```
ggplot(segmented_spotify, aes(x = loudness, y = danceability,
color = factor(cluster))) + geom_point() + facet_wrap(~playlist_genre)
```



```
ggplot(segmented_spotify, aes(x = loudness, y = energy, color = factor(cluster))) +
  geom_point() + facet_wrap(~playlist_genre)
```



```
avg_dend_obj <- as.dendrogram(hc_spotify)
avg_col_dend <- color_branches(avg_dend_obj, h = 16000)

# Create PDF for plotting
pdf("plots/plots_avg_col_dendrogram.pdf", width = 40, height = 15)

# Plotting
plot(avg_col_dend)
plot(cut(avg_col_dend, h = 17000)$upper, main = "Upper tree of cut at h=16000")
plot(cut(avg_col_dend, h = 17000)$lower[[2]], main = "Second branch of lower tree with cut")

# Close the PDF file's associated graphics device
dev.off()
```

pdf
2

```
path_weka <- "/Users/conorheffron/Library/CloudStorage/GoogleDrive-conor.heffron@ucdconnect.ie/Downloads/weka_acc.csv"
weka_acc <- read_csv(paste(path_weka, "weka_acc.csv", sep = "/"))
```

```
Rows: 76 Columns: 5
-- Column specification -----
Delimiter: ","
chr (2): Accuracy, Type
dbl (3): n, %, Bag

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Get Data Dimensions & Summary, Print Data Frame

```
dim(weka_acc)
```

```
[1] 76 5
```

```
summary(weka_acc)
```

	n	%	Type
Accuracy			
Length:76	Min. : 8815	Min. :40.41	Length:76
Class :character	1st Qu.: 9730	1st Qu.:44.61	Class :character
Mode :character	Median :10906	Median :50.00	Mode :character
	Mean :10906	Mean :50.00	
	3rd Qu.:12082	3rd Qu.:55.39	
	Max. :12997	Max. :59.59	
Bag			
	Min. : 0.000		
	1st Qu.: 2.000		
	Median : 8.000		
	Mean : 8.684		
	3rd Qu.:14.000		
	Max. :20.000		

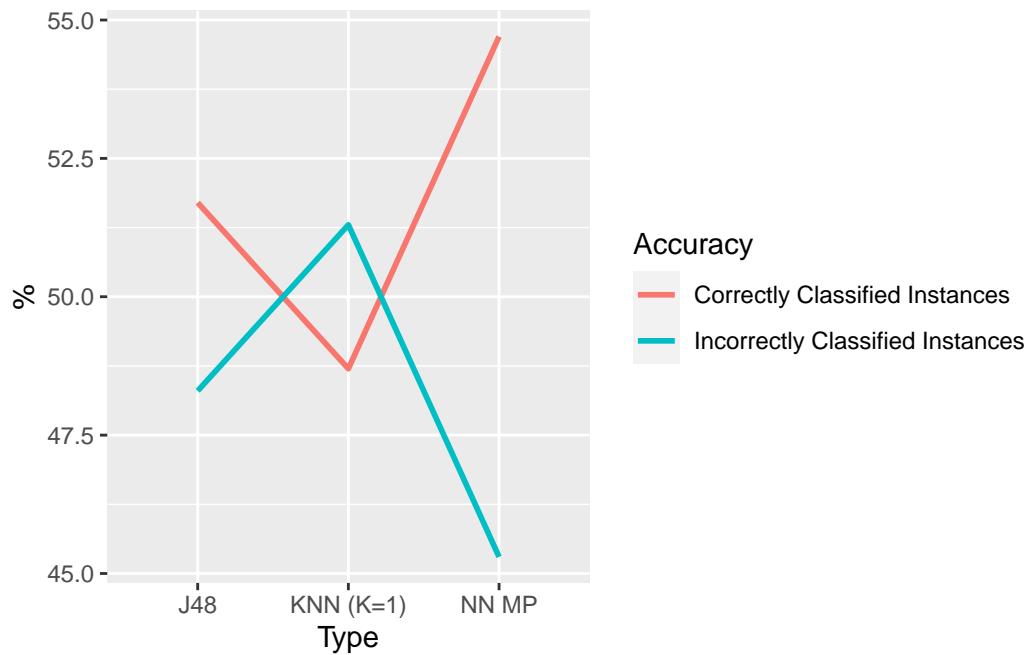
```
weka_acc
```

```
# A tibble: 76 x 5
#> # ... with 66 more rows
#> # ... with 10 variables: Accuracy <chr>, n <dbl>, `%` <dbl>, Type <chr>, Bag <dbl>
#> #   1 Correctly Classified Instances 12102 55.5 Vote(AVG) 0
#> #   2 Incorrectly Classified Instances 9710 44.5 Vote(AVG) 0
#> #   3 Correctly Classified Instances 12248 56.2 Vote(Majority) 0
#> #   4 Incorrectly Classified Instances 9564 43.8 Vote(Majority) 0
#> #   5 Correctly Classified Instances 10700 49.1 Vote(MAX) 0
#> #   6 Incorrectly Classified Instances 11112 50.9 Vote(MAX) 0
#> #   7 Correctly Classified Instances 10509 48.2 Vote(MIN) 0
#> #   8 Incorrectly Classified Instances 11303 51.8 Vote(MIN) 0
#> #   9 Correctly Classified Instances 10938 50.2 Vote(PRODUCT) 0
#> #  10 Incorrectly Classified Instances 10874 49.8 Vote(PRODUCT) 0
```

Plot for voting with 3x Combination Rules(KNN K=1, Multilayer Perceptron Neural Network (MP NN), JV8 (Decision Tree))

```
ggplot(weka_acc %>%
  filter(substr(Type, 1, 4) != "Vote" & Bag == 0), aes(x = Type,
  y = `%`, color = Accuracy, group = Accuracy)) + geom_line(size = 1)
```

Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
 i Please use `linewidth` instead.



Plot Ensembles by Bagging (Bag == 2->20 in increments of 2)

```
ggplot(weka_acc %>%
  filter(Bag != 0), aes(y = `%`, x = Bag, color = Accuracy,
  group = Accuracy)) + geom_point() + geom_line(size = 0.31) +
  facet_wrap(~Type)
```

