ANALYZING SPOTIFY DATA

EXPLORING THE POSSIBILITIES OF USER DATA FROM A SCIENTIFIC AND BUSINESS PERSPECTIVE

By Jeroen van den Hoven

Supervised by Sandjai Bhulai

VU University, August 2015.

SUMMARY

Spotify is the largest music streaming service available. The company started in 2006 in a time when piracy caused considerable losses to the music industry. In January 2015 they had 60 million users in total of which 15 million premium users (1) and these numbers seem to be increasing. Spotify offers free streaming of music to its users, though one can purchase a premium membership for added benefits, such as no advertisements and being able to listen to music offline.

The large number of users and content of Spotify create a large database of users and songs that users listened to that could hold interesting patterns and information for related companies, such as Spotify themselves, record companies or radio stations. The dataset in question has been provided by <Undisclosed company>, so we first look for general applications of the data and then focus on possibilities that will also be useful to <Undisclosed company>, but will also be interesting from a scientific perspective.

By performing some statistics on the entire dataset, we try to determine the worth of the Spotify data for both <Undisclosed company> and for scientific purposes. We will answer a few relatively simple questions regarding interesting patterns found in the data and try to formulate a good model that can be used with this data. After that, we will try to overcome the problems that arise when applying our model of choice to this dataset and deliver a way to create such a model for this database.

In the end we decided to try to perform a clustering on this dataset. This presented some challenges, such as a dataset of mixed variables, containing both continuous and nominal variables. Deciding that we did not want to use basic techniques to solve, we looked further for solutions and found two possible candidates: the cluster ensemble approach and Gower's distance metric. The metric used to evaluate whether or not a clustering was good was the cophenetic correlation coefficient.

After some trials, the cluster ensemble approach appeared to be an ineffective way of tackling the issue of mixed variables since it resulted in clusterings with poor fitness, with a maximum of 0.4 on a scale of 0 to 1. We expect this has something to do with the amount of unnecessary information loss that will be lost in the process of using the cluster ensemble approach.

Gower's distance metric performed considerably better with an initial fitness of 0.68. After some optimization we ended with a fitness of 0.94. We decided to base our final clustering on this model. However, upon analysing this final clustering we found that the set of weights used for this fitness resulted in the dataset only being split on two variables. It seems that, though Gower's metric does have the potency to reach high fitness, it may have a tendency to be biased, depending on the chosen weights and the underlying data.

In the end we will also propose a few changes to the methods that could be used to improve the effectiveness of both clustering methods.

CONTENTS

INTRODUCTION

Spotify is the largest music streaming service available. The company started in 2006 in a time when piracy caused considerable losses to the music industry. In January 2015 they had 60 million users in total of which 15 million premium users (1) and these numbers seem to be increasing. Spotify offers free streaming of music to its users, though one can purchase a premium membership for added benefits, such as no advertisements and being able to listen to music offline.

The large number of users and content of Spotify create a large database of users and songs that users listened to that could hold interesting patterns and information for related companies, such as Spotify themselves, record companies or radio stations. The dataset in question has been provided by <Undisclosed company>, so we will first be looking for general applications of the data and then focus on possibilities that will also be useful to <Undisclosed company>, but will also be interesting from a scientific perspective.

This paper will try to determine the worth of the Spotify data. This will be done according to the following procedure:

- Exploration of the data.
- Evaluation of the usefulness of different variables in different models.
- Choosing the most promising model based on the evaluation we just did, with proper argumentation. This argumentation will be based on how interesting the problem is from a scientific point of view and a business view, coupled with how likely it will be that the model can actually be build.
- Building a prototype of said model.

By combining all the information we just provided, the following research question is an obvious choice:

What are the possibilities of the Spotify dataset for <Undisclosed company> and which ones will be most interesting, both from a business perspective and a scientific perspective?

LITERATURE AND BACKGROUND

In this section we will be discussing some of the necessary definitions, techniques, and background information for this paper.

TERMS

Before we get to the technical details of the paper, it will be useful to define some of the terms that will be used regularly:

- Instance : One measurement: one song that was listened to at some time by someone. This corresponds to one row in the database.
- Nominal variable : A variable that takes values from a finite range of possibilities, for instance gender or device type.
- Continuous variable : A variable that takes values from a range on the real axis.

CLUSTERING METHODS

In the end this paper will focus mainly on clustering the Spotify data. To do this we need a good clustering method. There is a large selection to choose from, starting with hierarchal or non-hierarchal clustering and different clustering methods for both categories. Another important question is related to the type of data that we have available. We will be getting more into detail regarding this later on, but for now the most important piece of information is that the dataset has mixed variables with both nominal and continuous variables. This creates a problem, since most distance metrics and some clustering methods do not work well with this type of dataset. Simply replacing the nominal variables with dummy binary variables would normally be an option; however there is one interesting variable that has almost 700 different values, which would probably lead to a significant decrease in performance if they were converted to binary variables. To solve this problem we will be looking at two methods: Gower's metric and a cluster ensemble approach (2), which will be explained below.

NON-HIERARCHAL CLUSTERING

For non-hierarchal clustering we will be looking at k-means. K-means clustering is one of the better-known non-hierarchal clustering methods that chooses K centres for K clusters and assigns each instance to the closest cluster. It then recomputes the centre of each cluster by taking the average for each variable of all instances that are part of the cluster and repeats the process. So, in essence we will do the following:

1. Initialise K vectors M_i, i ε {1,2,…,K}, representing our K clusters. This can be done at random or by choosing K different instances from our original dataset.
2. Until we achieve convergence, do the following:
    2.1 Assign each instance X_i to its nearest cluster centre.
    2.2 Recompute the cluster centre for each cluster by averaging all instances in that cluster.

The assignment of instances to clusters is done in the following way: (3)

$$b_{i,j} = \begin{cases} 1 \; if \; d(X_i, M_j) = \min\limits_{k \in \{1,\dots,K\}} d(X_i, M_k) \\ 0 \; otherwise \end{cases}$$

where $d(X_i, M_j)$ is the distance between cluster M_j and instance X_i. Recomputing the new clusters is done in the following way: (3)

$$M_j = \frac{\sum_{i=0}^{K} b_{i,j} X_i}{\sum_{i=0}^{K} b_{i,j}}$$

Convergence will be achieved when M_j has stabilized for each $j \in \{1,\ldots,K\}$, or when a maximum number of iterations have been completed.

HIERARCHAL CLUSTERING

As for the hierarchal clustering, we will only be looking at agglomerative clustering (4) with complete linkage, which, at each combining step, combines the two clusters of which the furthest members are the closest to each other, until only one large cluster remains. By tracing the way in which the final cluster was formed, we can see how different instances are related to each other. The formula used to perform the clustering is the following, which will be repeated until only one cluster remains:

$$d(G_i, G_j) = \max_{X^r \in G_i, X^s \in G_j} d(X^r, X^x)$$

Where:

- D(x,y) : The distance between instance x and y, depending on some distance measure.
- G_i : Cluster i
- X_i : Instance i

GOWER'S DISTANCE METRIC

An important part of finding a clustering on a dataset with variables of mixed types is finding a distance metric that is capable of handling both continuous variables as well as nominal (or categorical) variables. Gower's similarity coefficient is capable of doing this by calculating the components of the distance between two instances X_i and X_j differently for each variable. For instance, take two instances X_i and X_j with both two variables, denoted by X_{ik} and X_{ik} for $k \in \{1, 2\}$. Assume the first variable is nominal and the second is continuous. For the first variable, the nominal variable, the difference between the values of X_{ik} and X_{jk} is defined as a simple indicator function (5):

$$s_{ijk} = \begin{cases} 1 \; if \; X_{ik} \neq X_{jk} \\ 0 \; if \; X_{ik} = X_{jk} \end{cases}$$

For the second variable, the continuous variable, the difference between the values of X_{ik} and X_{jk} is defined as (5):

$$s_{ijk} = 1 - \frac{|x_{ik} - x_{jk}|}{r_k}$$

r_k is defined here as the range of variable k, $\max(x_{.k}) - \min(x_{.k})$. These two types of variables are the only ones we will discuss here, since they are the only relevant ones for this paper. The Gower similarity coefficient is capable of dealing with other types of variables, though.

The only question that remains regarding the Gower coefficient now is how these S_{ijk} values are combined in a metric. This is done in the following way (5):

$$S_{ij} = \sum_{k=1}^{N} \frac{w_k s_{ijk}}{w_k}$$

Where:

- S_{ij} := the distance between observations X_i and X_j.
- w_k := the weight for variable k.
- S_{ijk} := the difference between X_{ik} and X_{jk}.

In the original formula the W_k is replaced by W_{ijk}, but we will be using the same weights for each pair of observations.

CLUSTER ENSEMBLE APPROACH

The basic principle of the cluster ensemble approach is a divide and conquer technique: it focuses on dividing the dataset in two datasets: one with all the nominal variables and one with all the continuous variables. The individual datasets are then clustered like normal datasets, which is possible since they only contain variables of one type. Once both datasets have been clustered, the results are combined into a new dataset of nominal variables, which is clustered again, resulting in the final clustering (6). The advantage of this technique is that one can use existing techniques to cluster the separate datasets and the final dataset.

COPHENETIC CORRELATION COEFFICIENT

One problem of clustering with this dataset is that there are no predefined clusters. This makes the process of determining whether or not a clustering is a good fit difficult. In order to be able to distinguish a good clustering from a bad clustering, we need a different evaluation method.

For non-hierarchical clustering methods we will be using the cophenetic correlation coefficient (7). This is a measure of how well a dendrogram matches the underlying distance matrix. It is defined as the correlation between the Euclidian distance and the distance in the dendrogram (8), or in our case, between Gower's distance metric and the distance in the dendrogram. The distance between two instances in the dendrogram is defined as the height in the dendrogram where two instances are joined for the first time. The resulting formula is as follows:

$$c = \frac{\sum_{i<j}(x(i,j) - \bar{x})(t(i,j) - \bar{t})}{\sqrt{(\sum_{i<j}(x(i,j) - \bar{x})^2)\,(\sum_{i<j}(t(i,j) - \bar{t})^2)}}$$

Where:

- c : the cophenetic correlation coefficient
- $x(i,j)$: the Euclidian / Gower's distance between instances i and j.
- $t(i,j)$: the distance between instances i and j in the dendrogram, defined as the height in the dendrogram where the two instances are joined for the first time.
- $\bar{x}$: the average Euclidian / Gower's distance between instance.
- $\bar{t}$: the average distance between instances in the dendrogram.

The fit is deemed reasonably good if the cophenetic correlation coefficient lies between 0.7 and 0.8 on a scale from 0 to 1, good when it is in the range (0.8,0.9] and very good for any value larger than 0.9 (9)

SILHOUETTE

A more visual criteria to decide whether or not a clustering is good is the (average) silhouette (10). The silhouette is a measure of how well an instance is matched to its own cluster compared to the closest other cluster. By looking at the average silhouette over all instances, we can get a good idea whether or not the current clustering is appropriate. By doing this for multiple different numbers of clusters, we can determine a good value for the number of clusters.

Before we continue, we need to define a few variables:

- a_i : average dissimilarity of instance *i* to all other objects in *a*. This variable has value 0 for a cluster of size 1.
- $d_{i,c}$: average dissimilarity of instance *i* to all other objects in *c*.
- b_i : $\min_{C \neq A} d_{i,C}$

We will then be looking at the silhouette s_i of instance i: (10)

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}}$$

We can see that:

$$-1 \leq s_i \leq 1$$

Now we can get an idea about what s_i *represents*: (10)

1. if s_i is close to 1, then a_i is much lower than b_i, indicating that instance *i* is assigned to the proper cluster.
2. if s_i is close to -1, then a_i is much higher than b_i, indicating that instance *i* is assigned to the wrong cluster.
3. if s_i is close to 0, then a_i is approximately equal to b_i, indicating that it is unclear to which cluster instance *i* should be allocated.

By looking at the average silhouette *S* we can determine whether or not instances have been properly assigned to a cluster. This can be used to determine the number of clusters by computing multiple different clusters and their average silhouettes and plotting these in a simple graph. We can then select a cluster based on the value of the average silhouette and the number of clusters. For instance, a value of *K = 2* clusters might have a high silhouette, but not enough clusters for us to actually work with.

METHODS

The goal of this paper is to determine the most potent application for the Spotify dataset. In order to do this, we will be following these steps:

- Exploration of the data.
- Evaluation of the usefulness of different variables in different models, as well as the models themselves.
- Choosing the most potent model based on said evaluation, with proper argumentation.
- Building a prototype of said model.

DATA EXPLORATION

An important part of constructing any model is the data exploration that precedes it. It gives one insight into what kind of data is available and which parts of it are of interest. Apart from studying the variables themselves, the data exploration also allows one to get an idea of which types of models can be used and which parts can be used to form a model. A description of the variables can be found in the appendix. For the data exploration, we will be using some standard histograms to get an idea about how some variables are distributed across instances and across users. Since the dataset contains 52 different variables, we will not be looking extensively at most of them, but only at the ones that will be used in the models. We will comment shortly on why some variables were not selected in the evaluation of models.

EVALUATION OF MODELS

After the basic data exploration, we will be focussing on thinking of possible models that we can use with this data. Furthermore, we will be looking at some interesting questions that we believe can be answered using this dataset. We plan on choosing one larger model to develop, chosen with proper argumentation, and answer multiple of the smaller questions.

TRANSFORMATION OF THE DATABASE

The current dataset is presented in an instance focussed way. This means that each row corresponds to one listening of a song by some user. Looking at what a specific user does becomes difficult in such a database, so we will be transforming the current database into a user focussed database, where each row corresponds to information for one user. Some information will be lost in this transformation, such as all the songs that a user listened to or all the times on which he / she listened to that song. We want to keep some information regarding these variables. This will be done in two ways, depending on the type of variable in question:

- For nominal variables we will be taking the most often occurring value as the value for this user.
- For continuous variables the mean value will be used.

This transformation will allow us to study the individual users, removing the number of times someone listened to music during March from the data. This allows us to give a better estimation of the number of premium users, the number of users from each region of the Netherlands, and perform analytics on the gender of users. Furthermore, if we want to perform any form of clustering we need to generate a matrix of distances between different users. When we want to use the instance-focussed database for this, this will probably become a tedious and time consuming task, but with a user-focussed database we can plug this database into the correct functions and end up with a distance matrix. Plugging the instance-focussed database into a clustering algorithm will also work, though it would create a clustering based on the instances. This could be interesting in its own right but we are more interested in the clusters underlying the users than the clusters of the instances.

RESULTS

DATA EXPLORATION

This dataset is not the entire dataset, but just a sample from the main database of Spotify for the month of March from the Netherlands. It contains data from 969 different users. Approximately 18.000 songs can be found in the ± 113.000 instances. The dataset contains 52 columns, corresponding to 52 different variables. The first 16 variables contain information about the user, whilst the other 36 variables describe the song. We will focus mainly on the first 16 variables. This is because we deemed the music related variables to be of no use for larger models with such a small dataset. With 18.000 different songs and 113.000 instances, we have an average of ± 6 instances / song, which will not be sufficient data to construct a model with. Furthermore, only 0.36% of the songs have been listened to at least 100 times.

We will be looking at the following variables:

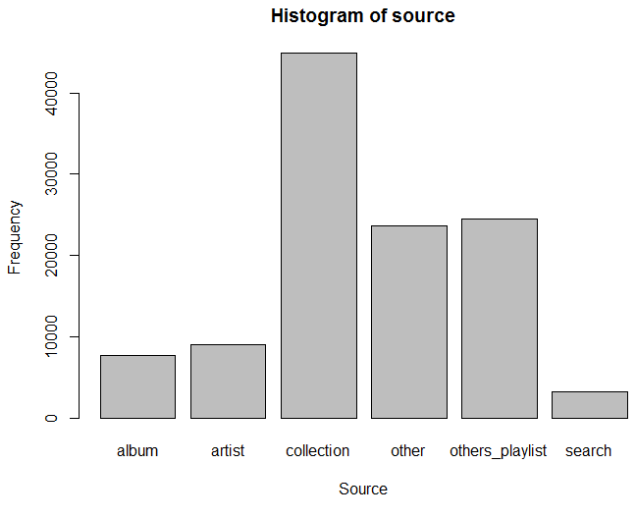
- Source
- Device type and OS type
- Gender
- Region
- Age

SOURCE

One of the interesting variables is the source variable. It describes how the song was found:

Source:	*Album*	*Artist*	*Collection*	*Other*	*Others_playlist*	*Search*
Number:	7731	9038	44867	23628	24505	3248

Table 1 and Figure 1: The number of instances found through each source.



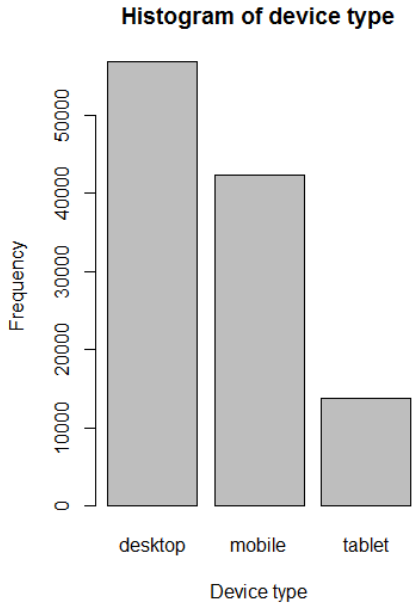
As we can see, it's clear that *collection* is the most popular choice of finding a song on Spotify. *Others_playlist* and *other* also provide a sizable portion of the methods for finding a song. Surprisingly, the *search* function is ranked as the function that is used the least to find a song. We are not sure why this is the case; this might have something to do with how the variable is recorded, but this is just speculation.

DEVICE TYPE & OPERATING SYSTEM TYPE

Another interesting variable is the device type. As the name implies, it records on which type of device the song was listened to:

Source:	*Desktop*	*Mobile*	*Tablet*
Number:	56866	42347	13804
Percentage originating from a premium account:	51%	78%	51%

Table 2 and Figure 2: The number of instances on each device type.



Desktop leads as the favourite device for using Spotify with just over half of all recorded instances. *Mobile* devices follow in a reasonably close second position with 42.347 recorded instances. It seems that the *tablet* is not a very popular device for listening to Spotify, but to make a better decision about this we would need data from a longer period of time. The large majority of instances from mobile platforms seem to come from premium users, though desktop and tablet do not perform poorly, since both have a premium usage percentage of 51%.

It is interesting to note that the smart TV is not a separate device, though Spotify does support an application for these TV's. Either smart TV's are added to another device type, or the application came out after March 2015, or no one in the Netherlands uses this application, which seems very unlikely. There are probably more possible reasons why this device is not shown in this dataset, but since we do not have any means of confirming any of them, we will not speculate any further.

Another variable, the operating system type, is also interesting in combination with the device type:

Source:	*Android*	*Browser*	*iOS*	*Linux*	*Mac*	*Other*	*Windows*	*W. Phone*
Number:	19962	995	35690	243	12009	10691	32928	499
Percentage from premium users:	65.1%	30.2%	75.1%	46.9%	57.3%	98.8%	33.2%	76.2%

Table 3: The number of instances per operating systems. As expected, the number of instances on mobile and tablet from the device type variable equal the number of instances on Android, iOS and Windows Phone (W. Phone). The same applies to desktop and browser, Linux, Mac, other and Windows.

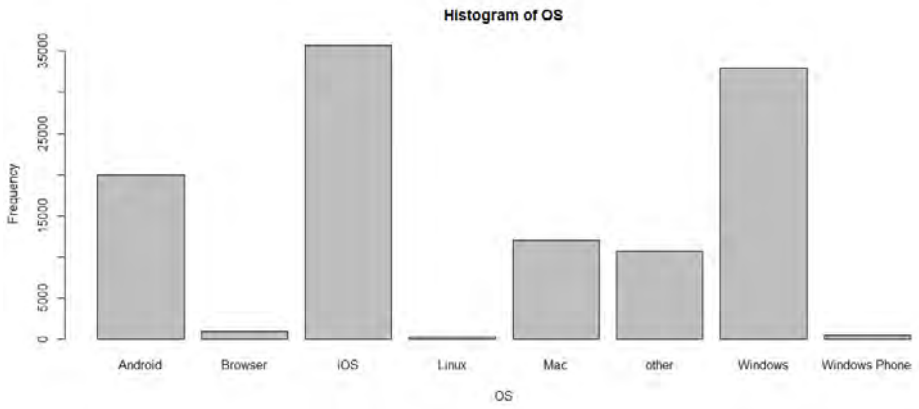


Figure 3: The number of instances per operating systems.

Interestingly, *iOS* is now the largest contributor to the number of instances with 35.690 instances. This causes *iOS* to make up 84% of all *mobile* usage on Spotify, a significant portion of the Spotify market. *Windows* is also a large contributor with just short of 33.000 instances, which is 58% of all *desktop* instances. Since *other* belongs in the *desktop* category according to the counting of instances, we can only assume that this encompasses other operating systems that fall outside of the other categories. What falls in the *other* category is quite interesting though, since almost 99% of all instances from *other* operating systems have a premium account associated with them. Furthermore, *Mac* and *iOS* users also seem to have a high percentage of premium users, compared to others in their respective device genre.

GENDER

The gender of the users is always an interesting variable.

Source:	*Female*	*Male*	*Other*
Number of instances:	43663	69240	114
Number of users:	391	566	1
Average number of instances per user	111.67	122.33	114
Percentage premium user	32%	52%	-

Table 4: The number of instances created per gender, the number of users per gender and the average number of instances per user per gender.

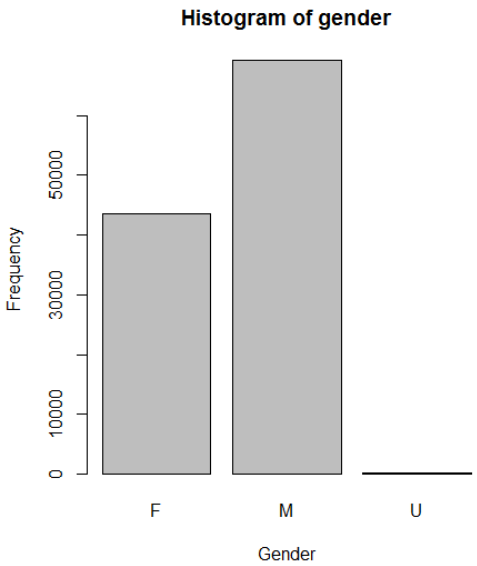


Figure 4: The number of instances created, sorted by gender.

From the data it is apparent that Spotify has significantly more male than female users, since men account for 59% of the users and women account for 40%. Women also seem to have a higher chance of being a free user with 32% of women being a premium member versus 52% of men. This might be something thing Spotify can exploit by targeting women more in their advertisements. Men also seem to tend to listen to more songs on average, with 10% more instances than women on average.

It can also be quite interesting to see from which areas of the Netherlands the users come.

Province	*Number of instances*	*Number of users*	*Average instances per user per province*	*Percentage premium users*
Unknown	4050	9	450	11.1%
Drenthe	1961	66	29.7	42.7%
Flevoland	1353	15	90.2	37.5%
Friesland	1373	21	65.4	23.3%
Gelderland	8105	81	100.1	36.7%
Groningen	3600	27	133.3	30.6%
Limburg	3343	41	81.5	36.0%
Noord Brabant	18989	133	142.8	49.3%
Noord Holland	25063	199	125.9	42.8%
Overijssel	8824	56	157.6	35.4%
Utrecht	12118	108	112.2	41.0%
Zeeland	1616	14	115.4	30.4%
Zuid Holland	22622	188	120.3	44.2%

Table 5: The number of instances, the number of users, the average number of instances, and the percentage of premium users per province.

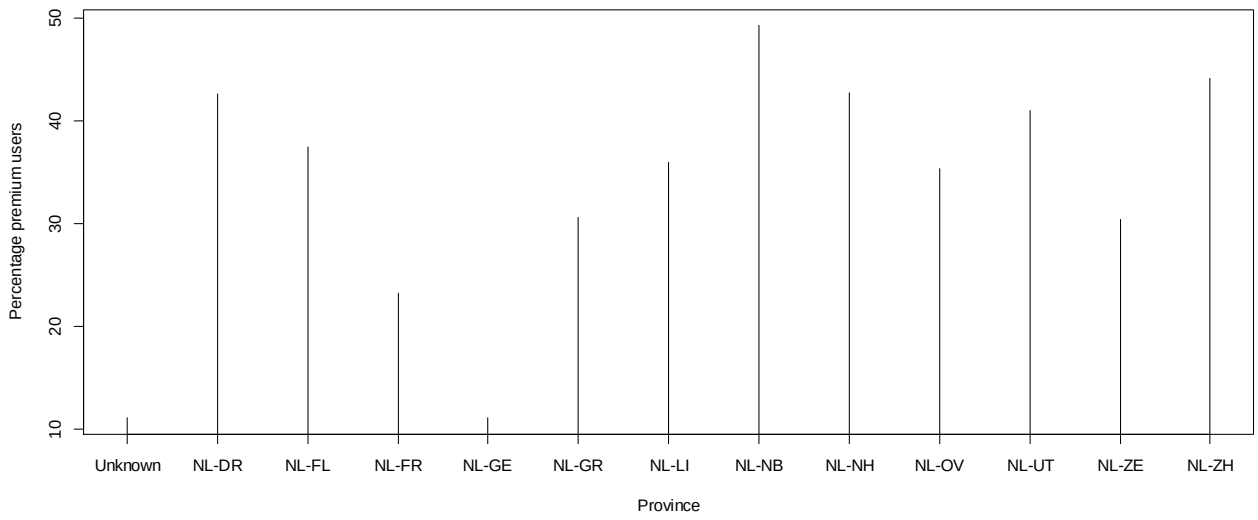


Figure 5: The percentage of premium users per province.

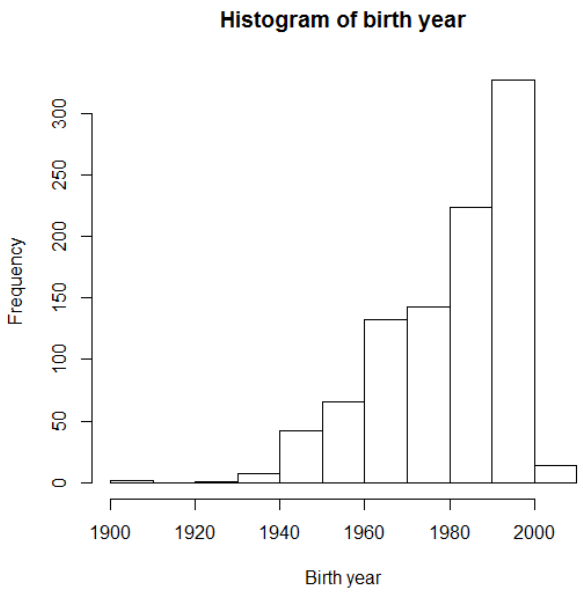
Table 5 and Figure 5 show quite well how types of users and number of users differ per province. Noord Brabant, Noord Holland, Zuid Holland, and Utrecht for instance have a relative high percentage of premium users, a large number of instances and high usage per user. Drenthe has a surprisingly low average number of instances per user with just 29.7, especially compared to Overijssel, where this number rests at 157.6. Drenthe, Flevoland, Friesland, and Zeeland have a very low usage with all of these provinces providing a maximum of around 2000 instances for a database with a total of approximately 113.000 instances. It seems that Spotify still has a potential market that is spread over the Netherlands and not necessarily focussed on a few provinces. The nine users whose location is unknown provide for a significant portion of the recorded instances with an average of 450 instances per user.

AGE

As the second to last variable we will be looking more closely at the age of the users.

Time period	*1900 – 1930*	*1931 – 1940*	*1941 – 1950*	*1951 – 1960*	*1961 – 1970*	*1971 – 1980*	*1981 – 1990*	*1991 – 2002*
Number of users	3	7	42	66	132	143	224	341
Percentage of premium users	0%	14.3%	35.7%	65.2%	55.3%	61.5%	54.9%	23.5%

Table 6 and Figure 6: The birth year of users.



Spotify seems to have a relatively young user base. The correlation between the first lowest boundary for each time period and the number of user is 0.89, which indicates a strong positive correlation between the number of users for a specific age bracket and the minimum age of that bracket. This in turn implies that there is a strong correlation between the age and the number of users of that age.

Interestingly though, the younger users do not contribute the most to the number of premium users on a percentage basis. The age group 25 – 65 contributes the most on a percentage basis, ranging from 55% to 65%, whilst the age range of 13-25 only contributes 23.5%. This could be money related, since older people tend to have more money.

PREMIUM USERS

The last variable that we will be looking at is most likely the most important one to Spotify. It tells us what type of user created the instance as well as telling us how they got a certain kind of membership:

Kind \ Type	*Basic-Desktop*	*Free*	*Premium*
Ad	0	42060	0
Paid	1945	0	38570
Partner	0	0	27445
Trial	0	33	2964

Table 7: The number of instances created by each type / kind pair. Type represents the type of user; free, premium or basic-desktop. Kind represents how the user got this status: through having ads on their Spotify, they paid for it, through a partner, or through a trial version.

Kind \ Type	*Basic-Desktop*	*Free*	*Premium*
Ad	0	524	0
Paid	6	0	182
Partner	0	0	206
Trial	0	4	33

Table 8: The number of users for each type / kind pair. Type represents the type of user; free, premium or basic-desktop. Kind represents how the user got this status: through having ads on their Spotify, they paid for it, through a partner, or through a trial version.

Something that becomes apparent when analysing this data is those premium Spotify members use the service more on average. This might be expected, since users who use a service more may be more inclined to pay for that service. It also shows that a significant amount of people has a *premium* account on Spotify thanks to *partners*: more people own a *partner* account than a *paid* account. In the Netherlands, people can get a Spotify membership as a package bonus at some companies, which may explain this figure. However, *paid* users tend to use the service significantly more than *partner* users.

EVALUATION OF MODELS

Given what we have learned from the data so far, we devised some possible models that could be used to turn this data into something interesting and useful:

- A recommender system, either for playlists or for songs.
- Clustering of users to create profiles.

Some smaller and more specific questions that the data exploration raised were:

- When do most people listen to Spotify?
- When do people listen to a specific playlist?
- What are the top 10 songs?
- Do older people listen to older music?

These questions were of such a nature that they would not cost a tremendous amount of time to answer. This is why we decided to answer all of them, or at least provide a way to answer them. However, constructing a good model for the two larger models would require a significant amount of time, which allows us to construct just one of them. In order to determine which model will be built, we will look at certain aspects of the model, such as feasibility and value in practice. We will first report the answers to the smaller questions and report our choice of model after that.

WHEN DO MOST PEOPLE LISTEN TO SPOTIFY?

A relatively easy question to answer, since we have the stream time available in our dataset:

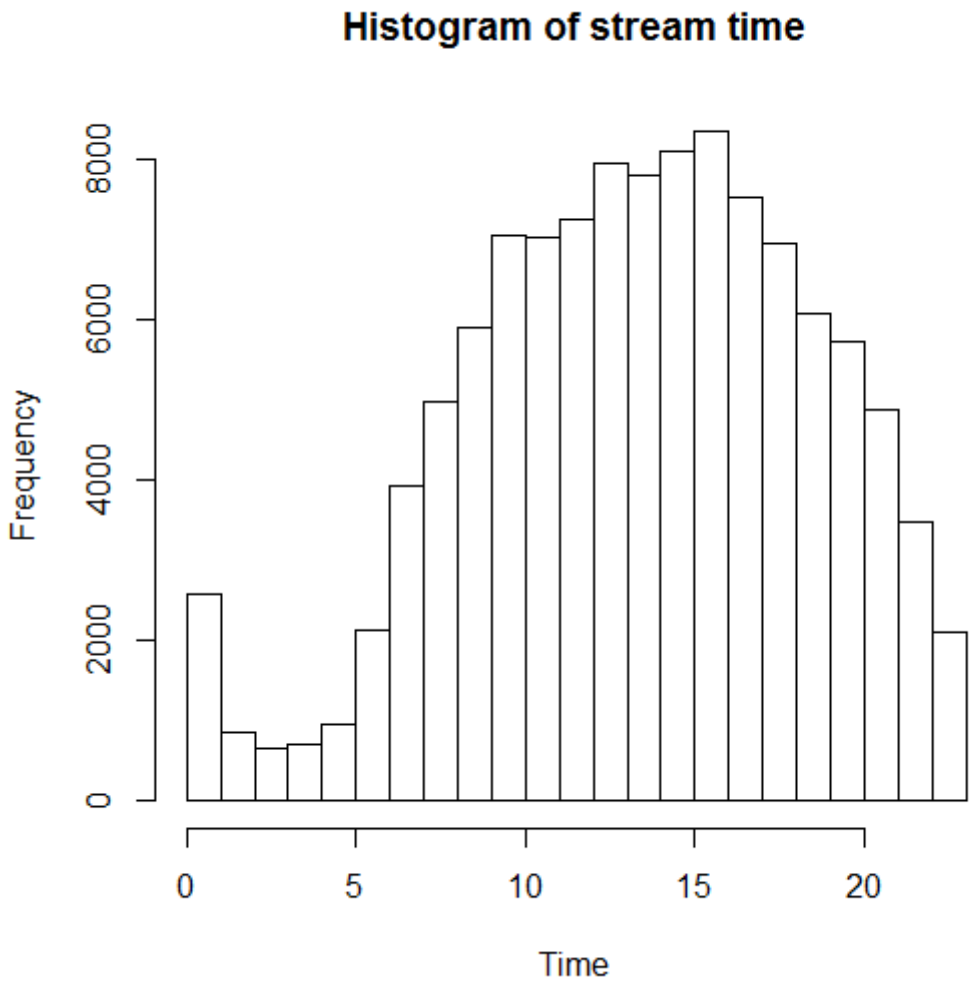


Figure 7: A histogram of when instances were streamed.

It is quite obvious that people tend to listen less to music on Spotify during the night. The number of instances starts to increase around 5:00 – 6:00 in the morning, growing steadily until around 9:00. After that the number of instances increases, though at a slower rate, until 15:00, when it slowly starts to decline at a reasonably steady pace until midnight. This information could be useful when planning server management or for planning the maximum capacity throughout the day. However, when we start looking at when a specific playlist is listened to, things can look quite different.

WHEN DO PEOPLE LISTEN TO A SPECIFIC PLAYLIST?

When looking at a histogram of the average time on which a playlist is listened to, we find a similar pattern as we found when looking at when songs were streamed:

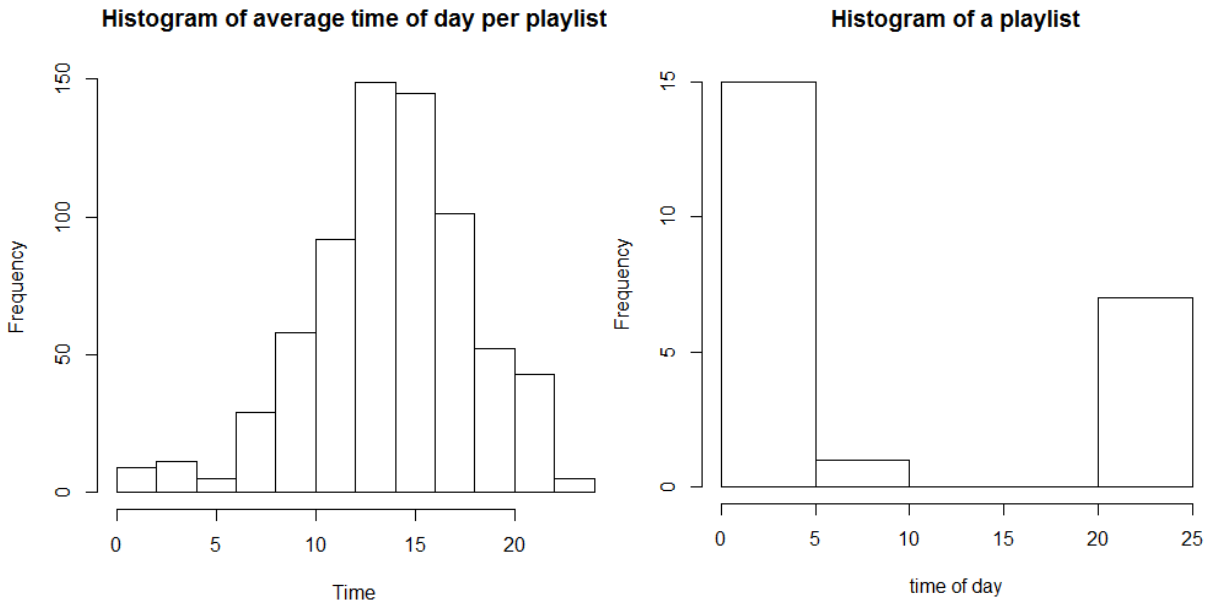


Figure 8 a (left) and b (right). Left: A histogram of the average time of day when playlists were listened to. Right: A histogram of one of the playlists, created by a user with id 11155891456.

It is clear that the histogram for the average time of day per playlist resembles the histogram for when instances were streamed. However, when we construct histograms for individual playlists, we can see clear differences in playlists. One example is shown in figure 8b, where we can clearly see that this playlists is listened only between 20:00 and 10:00. There are also playlists to which people listen to at every point of the day, those that are only listened to during working hours, or those that are listened to significantly more during 20:00 – 24:00, but not as much during the rest of the day. This shows that we cannot assume that all playlists are similar in nature, which probably is not a significant shock to anyone. However, this could perhaps be used to advertise different playlists to people at different times of the day.

An old question, and one that we personally found interesting: whether older people listen to older music.

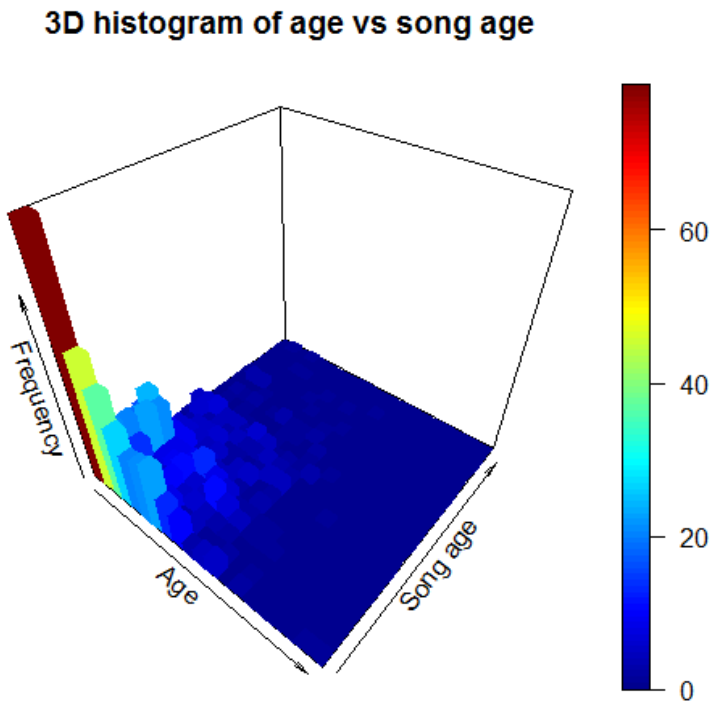


Figure 9: The relation between birth year and song year.

Figure 9 seems to indicate that younger songs are listened to a lot more. However, there does not seem to be any apparent correlation. To be sure if there is no correlation, we can do a correlation test. Using a Pearson's product moment correlation coefficient test, we determined that the p-value for the H_0 hypothesis *'There is no correlation between birth year and song year'* equals $2.7 * 10^{-9}$. This means that there is a significant indicator that there is a correlation between birth year and song year, with a value of 0.19. So although the correlation is significant, it is not exceptionally large. To see how this would work in a model, we performed a simple linear regression model on *age* and *song age*. This yields a model with significant parameters for the intercept ($<2 * 10^{-16}$) and the *age* variable ($2.7 * 10^{-9}$). However, the *age* variable has an ever lower estimate for the weight at just 0.12, compared to 0.19 for the correlation. This is a counterintuitive result, since one would not expect such a low correlation to be significant. This effect might be related to the large number of young songs (0-3 years) compared to the number of older songs. Since there are many of these younger songs, it seems logical that people encounter and listen to these songs more than the other songs, independent of their age. A short study of the data finds that around 54% of the instances are from the age bracket of [0,3] years.

We will start by looking at the recommender system. A recommender system in this situation would try to predict which playlists / songs would be liked by a certain user based on the history of that user and the history of other users. Based on what other users with a similar history listened to it will recommend playlists or songs to the user. Such an analysis requires a vast amount of data, especially when the number of possible playlists / songs can be very large. Since Spotify managed to generate just over 100.000 instances just in the Netherlands with a subset of its users, we believe that, using the entire dataset of Spotify, a recommender system would be possible, though it would require a significant amount of computing power. However, since we have a small subset of the data available, we do not have enough data available to create such a recommender system. As mentioned before, we have 18.000 songs in this playlist, which means each song is listened to approximately 6 times on average. Combined with the fact that just 0.36% of the songs has been listened to at least 100 times, it is clear that no accurate recommender system can be built. Furthermore, Spotify already has a recommender system, so the added value of another system would be small and probably not very useful to <Undisclosed company>. In our opinion, the recommender system is not a very viable idea for this subset, but it could be an interesting tool if it can be run on the entire dataset.

As for the clustering of users, we would try to find patterns in the dataset that we could exploit to identify different groups of users. This might then be used to target different groups of users in different ways and might help with sales. We have almost 1000 users available for clustering, which is a reasonable number. Furthermore, we have 16 descriptive variables for each of these users such as age, gender, and membership type, giving us a reasonable number of variables to perform the clustering on. A tricky part of this would be that it would be unsupervised learning without any previous information on what type of clusters / people could be expected to show up, so deciding what clustering is a good clustering might become difficult. However, if we do manage to find criteria to determine a good fit for a clustering through some alternative means, we might discover some interesting consumer groups in the Spotify database. To be able to work properly with users instead of instances, we would have to transform the database from an instance-focussed database to a user-focussed database, though this should not be too difficult. In our opinion, the clustering of users is a viable model for this subset of the data and possible an even more potent model if used on the entire dataset.

Our choice for a model is quite obvious: we prefer to try the clustering of users above the recommender system, at least for this dataset.

CLUSTERING

When we want to perform a clustering, we first need to determine a few key factors such as the clustering method that we use and, in our case, how to handle the different types of variables. In our database we have nominal and continuous variables that we would like to use for the clustering. However, most clustering algorithms can only work with either nominal or continuous variables, unless some transformation is used on the database. We will discuss the choices that we made regarding these factors after choosing the variables used for the clustering.

SELECTION OF VARIABLES

We will be using the following variables for our clustering:

Name	*Type*	*Description*
Hour of streaming	Nominal	Time of streaming
Stream length	Continuous	Duration of stream
Source	Nominal	Denotes how the song was found, for instance through artist or collection.
Source uri	Nominal	If source was a playlist, then this denotes the specific playlist.
Device type	Nominal	Desktop, mobile, or tablet. No smart TV.
Operating system	Nominal	The operating system of the device.
Region	Nominal	The province where the song was listened to.
Gender	Nominal	Gender
Birth year	Continuous	Birth year
Access	Nominal	Type of account. Basic desktop, free or premium
Type	Nominal	Ad, paid, partner, trial. Gives extra information about access.

Table 10: the chosen variables.

As discussed before, the music specific variables will not be used for clustering due to the many different songs present in the database. This leaves us with 16 possible variables for clustering. We removed five more variables:

- The user id. If we want to cluster different users, then this will be of no use.
- The metadata id. We do not know where this is for, so we will not be using it.
- The date of streaming. We do not use this for the same reason as the user id.
- Cached. This variable is always false, so it does not give us any information.
- Stream territory: For us this is always NL, for the Netherlands.

This leaves us with the above-mentioned set of eleven variables.

HIERARCHAL OR NON-HIERARCHAL

In order to be able to determine which clustering is the best one for our problem we need to be able to rate different clusterings. In supervised learning cases this would not be a problem, since one could do some basic tests to determine how well a clustering fits the data. We, on the other hand, have an unsupervised case, since we do not know in advance to which cluster an individual belongs. Since the data has twelve variables, using simple visual test would be difficult

and slow as well. We need a simple way to express the quality of the fit of the clustering to the actual data. For hierarchal clustering there exists a way to do this: the cophenetic correlation coefficient (CPCC). As mentioned in the literature section, the CPCC indicates how well a clustering fits the underlying dataset by computing the correlation between the distance between pairs of instances in the dataset and the distance in the dendrogram. Since we are uncertain whether such a similar coefficient exists for non-hierarchal clustering algorithms, we decided to use a hierarchal clustering algorithm.

HANDLING OF MIXED VARIABLES

As mentioned earlier, most clustering algorithms use either nominal or continuous variables. The reason for this is that in order to be able to perform a clustering on data, one needs to determine a distance measure so that one can define the distance between two instances. For a pure continuous / nominal database this is not an issue, since there are more than enough distance measures that work for pure continuous / nominal instances. For databases where there are just a few variables that are nominal or continuous compared to the other type, a transformation can be done to the fewer present variables to ensure a pure nominal / continuous dataset.

For instance, one could transform a continuous variable to a nominal one by breaking up the range of the continuous variable in multiple smaller ranges and assigning a number to each smaller range. By doing this you would have replaced the continuous variable with a nominal one. For nominal variables on the other hand one could replace the current nominal variable with *N* binary variables, where *N* is the number of values that the nominal variable could be. Only one of the binary variables will be equal to 1 for each instance and the rest would be 0. The binary variable that would be equal to 1 would be the variable that corresponds to the value that the instance had in the original nominal variable. The binary variables can be regarded as regular continuous variables in the distance measurement, though one might want to normalize all variables in the database (or do something similar) to ensure that the binary variables do not outweigh the other variables or vice versa.

Looking at our database, we can see that the first type of transformation could provide the solution that we need. *Age* and *streamlength*, the two continuous variables that we will be using, have a small enough finite range of values that they attain, meaning that we can transform them into nominal variables. The second type of transformation would be less feasible, since some of the nominal variables in our dataset can attain almost 700 different values, resulting in the creation of almost 700 new variables, which would probably slow down calculations considerably.

However, we would like to try some methods that do not require us to transform the variables, since this causes some information loss. For this we will be using two methods:

- Cluster ensemble approach.
- Gower's distance metric

CLUSTER ENSEMBLE APPROACH

As mentioned in the literature section, the basic idea of the cluster ensemble approach is the following (6):

1. Divide the database in a pure nominal and a pure continuous database.

2. Cluster both the pure nominal and the pure continuous databases. This will result in a cluster index for each user for each database.
3. Combine the indices from step 2 into a single database. Each row represents one user and each column contains the result from one of the clustered databases.
4. Perform a clustering on this final database.

In our case, we would form a database for S*treamlength* and *Age* and another database for the rest of the variables, perform a clustering on both, and combine the results for one final clustering. This technique allows us to use basic clustering techniques to perform all the clustering. For the continuous dataset a simple Euclidian distance measure can be used, and for the other two datasets (the nominal and final sets) we can use a distance measurement for nominal variables, such as the number of variables that have different values. Since *Age* and *Streamlength* have different ranges, it might be better to first normalize these and then perform the clustering, so that neither one of the variables has a larger impact on the clustering than the other one.

This is a technique that we had not heard about before, so we are eager to see whether or not it works properly. If it does, it would make working with databases of mixed variables significantly easier, but probably at the cost of a longer runtime, since we need to do the clustering three times. However, the first two times will be done on a subset of the data and the final one on a dataset with just two columns, so each clustering will probably require less time than a clustering on the entire dataset.

When performing the clustering using the cluster ensemble approach we discovered that the resulting CPCC on the final cluster was of a surprising low value at around 0.19 for up to 10 clusters and around 0.4 for 20 or more clusters. An initial study of the dendrogram showed us the reason. When the nominal and continuous datasets are clustered, the resulting indicator is nominal. If one wants to cluster this, one will be clustering on the basis of two nominal variables. Since the distance metrics for nominal variables tend to work on a basis of a distance of 0 when two variables are the same and a distance of 1 when they are not, we ended up with only three possible distances: 0, 1 and 2.

For users that have distance zero this is not a problem: they were clustered in the same clusters in both the nominal and continuous datasets, so it would seem obvious that they are similar. However, problems emerge for the users that have a distance greater than zero. Since the algorithm is incapable of using the previous datasets, it cannot determine the difference between two sets of two users that both have distance one (or two for that matter); though in the actual datasets there could be a significant difference in the distance. This results in a clustering of poor quality. Increasing the number of clusters in the clusterings on the pure datasets resulted in higher CPCC's since the algorithm could make a better distinction between clusters, but this still did not raise the CPCC significantly above 0.4. When trying 80 clusters for the pure datasets clusterings we found that the CPCC dropped back to 0.3, indicating that this again gives problems with the final clustering, probably due to the fact that there are now too many small distinctions between clusters that are still quite similar.

GOWER'S DISTANCE METRIC

Another solution is to use a distance metric that is capable of combining different types of variables. As mentioned in the Literature section, one way to do this is by using Gower's distance metric. Gower's distance metric works by applying a different function to each variable depending on its type and computing a weighted sum of these values. This sum is then used as the distance between two instances. By using this distance metric we were able to perform a

hierarchal clustering on the user database. The first try, with all weights set to one, resulted in a CPCC of 0.68, a significant increase compared to the cluster ensemble approach.

With the rule of thumb being that a CPCC of 0.7 or higher indicates a strong fit, we decided to try different sets of weights for the variables to increase this value. Since there are no known methods yet for choosing good weights for as far as we know, we decided to do a simple though time intensive random search. We choose our weights at random from the range [0, 4] uniformly, ran the clustering and computed the CPCC. By choosing one weight for the nominal variables and one for the continuous variables we were able to raise the CPCC to approximately 0.73. However, by choosing a different weight for each individual variable, we were able to raise the CPCC to 0.94, indicating a very strong fit. The results are summarized in Table 11:

Method	*CPCC*
All weights equal 1	0.68
One per variable type	0.73
One weight for each variable	0.94

Table 11: the CPCC for different sets of weights.

The resulting weight set for the case where the CPCC was 0.94 was as follows:

Variable name	*Weight*
Type	3.90
Access	3.11
Birth year	2.33
Stream length	1.78
Source	1.19
OS	1.08
Source uri	0.77
Hour of streaming	0.73
Gender	0.61
Region	0.15
Device type	0.10

Table 12: the weight set for the optimal case.

It is clear that *Type*, *Access*, and *Birth year* have a relatively high weight in the calculation of the distance between users. This might indicate that these variables are good for splitting the dataset into different clusters. *Device type* has the lowest weight, but the reason for this may be the fact that the *OS* variable has similar but more detailed information. Interestingly, even though *Type* and *Access* should contain similar information, both are apparently useful in dividing the users. The reason that *Type* and *Access* both have high weights and *Device type* and *OS* do not may have something to do with how they are related. Each possible value of *OS* can only be in one category in *Device type*, but this does not hold for *Type* and *Access*.

THE FINAL CLUSTERING

One final step to overcome when using hierarchal clustering is to determine the number of clusters. A hierarchal algorithm creates a dendrogram that represents how different instances are connected. The dendrogram of the final clustering can be found in Figure 10:

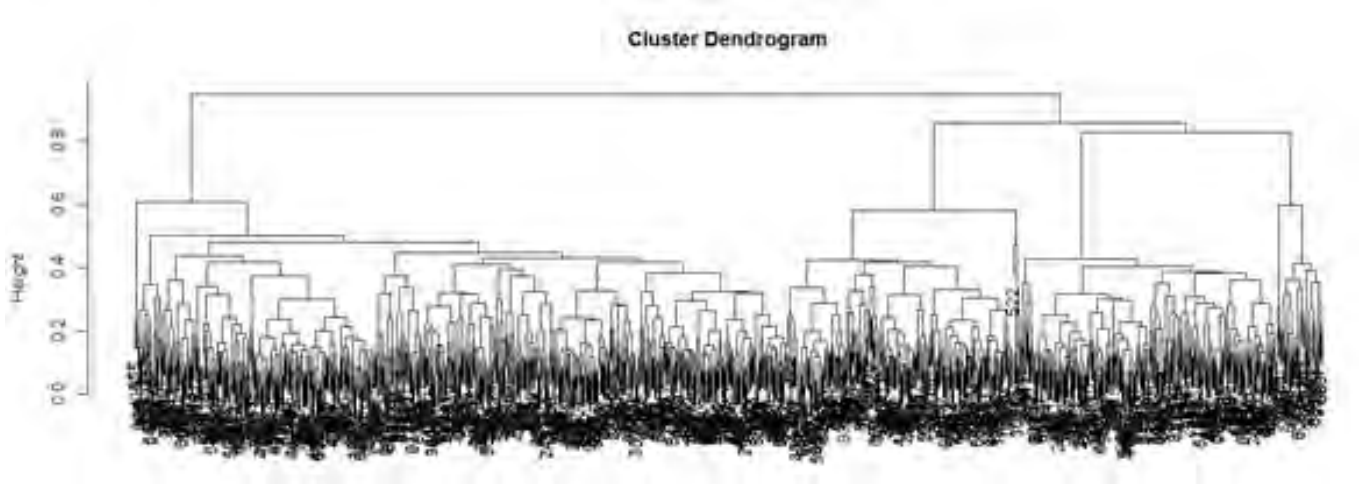
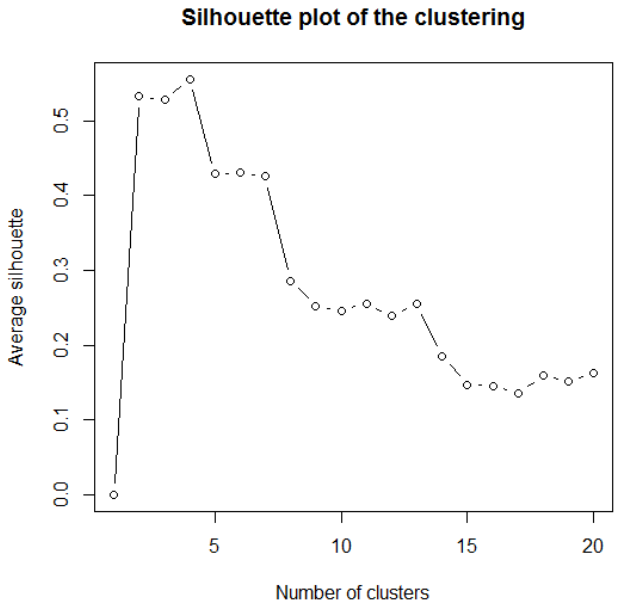


Figure 10: the dendrogram of the final clustering.

The clustering seems to suffer from a few outlying clusters, indicated by a small selection of instances that are only joined to the rest of the dendrogram at a reasonably high height. In our final clustering we might want to exclude these from the rest of the clusters to make sure that they do not temper with the final result. Apart from this, the lower level merges seem to be reasonably balanced. Combined with the high CPCC, we believe that this is a reasonably good dendrogram to continue working with to determine the optimal number of clusters. To create clusters we need to cut the dendrogram at a certain height or after a predetermined number of merges, starting at the top of the dendrogram. Then each instance in one of the separate subtrees will be classified as belonging to the cluster of that subtree.

To determine the number of clusters, we will be looking at the average silhouette of the clustering (10). The silhouette is a measure of how well an instance is matched to its own cluster compared to the closest other cluster. By looking at the average silhouette over all instances, we can get a good idea whether or not the current clustering is appropriate. By doing this for multiple different numbers of clusters, we can determine a good value for the number of clusters:

Figure 11: the silhouette plot for the dendrogram in figure 10.



As we can clearly see, the average silhouette, the average of all silhouettes of all clusters, decreases as the number of clusters increases. This seems logical, since an increase in the number of clusters will most likely decrease the distance of instances to their closest neighbouring clusters, decreasing the value of the average silhouette. In our case, K=4 has the highest average silhouette. However this would result in very few clusters, so one might also be tempted to choose K=7 as the final number of clusters, depending one ones preferences. For instance, in the case of seven clusters, one might be able to

target the audience in each cluster more efficiently compared to only four clusters.

To see the difference between four and seven clusters, we will be looking at the number of instances in each cluster:

K	*1*	*2*	*3*	*4*	*5*	*6*	*7*
4	526	206	189	37	-	-	-
7	524	206	182	33	7	4	2

Table 13: the number of instances in each cluster, sorted by size.

One can see that the difference between *K*=4 and *K*=7 is that the outlying clusters that we detected at figure 10 have been cut from the rest of the clusters. As we can see, almost every cluster seemed to have suffered from these outliers, since the three new clusters came from different original clusters. From seven clusters on the cutting will result in larger clusters being broken down evenly, indicating that we are not breaking up clusters from their outliers. Taking this into account, we believe that it would be better to select seven clusters instead of four, even though four clusters have a higher average silhouette. In our opinion, the higher silhouette does not weigh up to the inclusion of outlier clusters in other clusters.

STATISTICS FOR THE FINAL CLUSTERING

Lastly, we will be looking at some basic statistics for the final clustering with *K*=7:

Cluster ID	*Size of cluster*	*Median age*	*MAD of age*	*% male users*
1	206	40	16.3	70.9%
2	524	25	11.9	49.4%
3	182	34	13.3	70.3%
4	4	26.5	8.9	100%
5	7	32	16.3	100%
6	33	25	10.4	60.6%
7	2	31.5	20.0	100%

Table 14: statistics for the final clustering. MAD stands for Mean Absolute Deviation and is a more robust version to estimate the standard deviation. The low number of users in clusters 4, 5, and 7 is why it is not entire illogical they can be 100% male.

The largest cluster, cluster 2, seems to be mainly made of young people with an almost 50/50 distribution of male / female users. Furthermore, the estimate for the standard deviation, the MAD, is lower than the two other large clusters, indicating that the spread of this cluster is relatively low and thus that it contains reasonably young people. Furthermore, this cluster contains only free users who have ads. This indicates that younger people tend to be free users, which corroborates the results found earlier in this chapter. Cluster 6 contains our *Trial* premium members, which are also relatively young. The spread of their age is also quite low, similar to cluster 2.

Clusters 1 and 3, the two other large clusters, mainly contain older users that tend to be male. When looking at the type of membership that they have, both groups are exclusively premium users. However, cluster 1 only contains *partner* type premium memberships, whereas cluster 3 only contains *paid* premium memberships. This should come as no surprise, since *Type* and *Access* were weighed heavily in the clustering algorithm, making the algorithm favour putting emphasis on clustering these types of users.

CONCLUSION & DISCUSSION

By performing some basic statistics on the entire dataset, we found out that most of the music related data would not be very useful. Some of the music data could be useful for some small insights into the data, such as creating a top 10 and doing a small test to see if there is a correlation between the age of the user and the age of a song. This showed that there was a statistically significant correlation, but not a very strong correlation. Fortunately, the user specific data would provide enough variance to be of interest, but not too much, allowing us to still use if for some models. Using this data we converted the music focussed database into a user-focussed database.

We concluded that performing a clustering on this user database would be the most valuable option for Spotify and <Undisclosed company>, since we had enough users to cluster on as well as enough variables to perform a proper selection and clustering. Our other option, the recommender system, would not work well with the dataset that we currently have since not enough songs are listened to a significant amount of time, though we believe it will work with the complete database.

In the end the best clustering setup was a hierarchal clustering algorithm on the entire database using the Gower's distance metric to cope with the mixed variables problem. Using this method we were able to reach a CPCC of 0.94, which indicates a very strong fit between the clustering and the underlying data. By looking at the resulting dendrogram and the average silhouette we determined that seven clusters was a good start for the number of clusters. By analysing this final clustering one might be able to identify specific groups of users, which may be used for targeted advertising.

This clustering results in the dataset being split mainly on the variables *Type* and *Access*. If we had more time, it might be useful to look more closely at clusterings with more clusters and analyse the statistics for more of these clusterings. This might help us to differentiate further between clusters so that we do not just split the database on one variable. Another way that we might prevent this in the future would be to assign a weight to each type of variable; however, as we have seen in the section *Gower's distance metric*, this will result in an initially lower CPCC. A third solution may be to not compute the CPCC between the clustering and the distance matrix on which it was build, but instead on the distance matrix that is created when using Gower's distance metric with a weight of (for instance) one for each variable.

When looking at the algorithm for the cluster ensemble approach, we believe that in its current form it will not be useful. The cluster steps performed on the pure nominal / continuous databases cause a loss of information that is too great for the clustering algorithm to work with. A solution might be to cluster just the continuous database and add the resulting clustering as a nominal variable to the nominal database. This will cause some information loss to occur in the clustering of the continuous variables but will preserve the information in the nominal database, resulting in a larger database that can be used for the final clustering. For instance, in our case we would end up with a final database of ten variables instead of just two, giving the distance measures for nominal variables a much better way to differentiate between users.

BIBLIOGRAPHY

1. *Tweakers.* [Online] 12 January 2015. [Cited: 15 August 2015.] http://tweakers.net/nieuws/100754/spotify-heeft-vijftien-miljoen-betalende-gebruikers.html.

2. *Clustering Mixed Numeric and Categorical Data: A cluster Ensemble Approach.* **Zengyou He, Xiaofe i Xu, Shengchun Deng.** 2005.

3. **Alpaydin, Ethem.** Introduction to Machine Learning. *Introduction to Machine Learning.* London : The MIT Press, p. 147.

4. —. Introduction to Machine Learning. *Introduction to Machine Learning.* London : TheMIT Press, p. 157.

5. *A General Coefficient of Similarity and Some of Its Properties.* **Gower, J. C.** 1971, Biometrics, p. 859.

6. *Clustering Mixed Numeric and Categorical Data: A cluster Ensemble Approach.* **Zengyou He, Xiaofe i Xu, Shengchun Deng.** 2005, p. 5.

7. *The Comparison of Dendrograms by Objective Methods.* **Rohlf, Robert R. Sokal and F. James.** s.l. : International Association for Plant Taxonomy, 1962.

8. *Numerical Taxonomy: The Principles and Practice of Numerical Classification.* **Sokal, P.H.A. Sneath and R. R.** San Francisco : Freeman, 1973.

9. *Numerical taxonomy and multivariate analysis system.* **FJ, Rohlf.** New York : Exeter Publishing, 1988.

10. **Rousseeuw, Peter J.** Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics.* 1987, 20.