CAN YOU PREDICT A HIT?

FINDING POTENTIAL HIT SONGS THROUGH LAST.FM PREDICTORS

BACHELOR THESIS
MARVIN SMIT (0720798)
DEPARTMENT OF ARTIFICIAL INTELLIGENCE
RADBOUD UNIVERSITEIT NIJMEGEN

SUPERVISORS:
DR. L.G. VUURPIJL
DR. F. GROOTJEN

Abstract

In the field of Music Information Retrieval, studies on Hit Song Science are becoming more and more popular. The concept of predicting whether a song has the potential to become a hit song is an interesting challenge. This paper describes my research on how hit song predictors can be utilized to predict hit songs. Three subject groups of predictors were gathered from the Last.fm service, based on a selection of past hit songs from the record charts. By gathering new data from these predictors and using a term frequency-inversed document frequency algorithm on their listened tracks, I was able to determine which songs had the potential to become a hit song. The results showed that the predictors have a better sense on listening to potential hit songs in comparison to their corresponding control groups.

TABLE OF CONTENTS

1 INTRODUCTION

All around the world, people listen to music every day. It is a language we can all understand. Music travels intercontinental and in all kinds of genres, so that everyone can listen to the songs they love most. Nowadays there are many online services that enable us to do so, but some of these services offer much more. Some allow people to interact with each other by sharing their playlists, while others provide the option to share your own music. Last.fm is one of those online services, where everyone establishes their own musical profile by registering every song one listens to. It enables us to find other people that have somewhat the same taste in music as yourself, such that you can share recommendations. There is no doubt that everyone has their own unique taste in music.

However, some music is more popular than other and the record charts tell us what today's most popular songs are. It can make you wonder why certain songs become a hit song while others do not. On top of that, some hit songs become acknowledged masterpieces, while other hit songs fall into oblivion as so called 'one-hit-wonders'. The work presented in this thesis is inspired by this observation and pursues the following question: "is there any way to make predictions about which songs have the potential to become hit songs?". A variety of studies have been done on the topic of Hit Song Science and some of these brought interesting results, providing lots of options for further research.

Now, suppose that there are people who have some sort of 'sixth sense' for hit songs, such that they listen to hit songs even before they actually climb to the top of the record charts. What if I could find a number of these people through the Last.fm online services? Perhaps there is a way to construct a group of predictive listeners who might succeed in producing fair predictions about which songs could actually become hit songs in the (near) future. This road has not yet been followed within the field of Hit Song Science, and so the goal of this research is to provide a foundation for this novel approach.

This paper describes a method in which a predictive model is constructed with users from the Last.fm service. First, I will give an introduction to the fields of (Music) Information Retrieval and Hit Song Science. Additionally, I will give a short introduction to the Last.fm service. Then, I will elaborate on the followed approach: this section provides information on how the subjects were gathered, how the hits were defined and how the data was retrieved. The next section explains how the data was processed and which methods were used to do the hit song predictions. Before analysing the results, we will take a quick look at the (raw) data that has been gathered. The final sections will conclude the research and provide some pointers on future research.

2 BACKGROUND

2.1 MUSIC INFORMATION RETRIEVAL

Information Retrieval (IR) has become more and more important as the amount of digital information keeps growing. Because of the enormous quantities of available information, there is a need for effective methods of automated information retrieval (E. Greengrass, 2000). Information retrieval systems are designed to analyse, process and store (sometimes unstructured) sources of information and retrieve those that are relevant to the needed information (G. Chowdhury, 2010). The field of information retrieval originates from the late 1940s, when the innovative article *As We May Think* (V. Bush, 1945) introduced the concept of accessing large amounts of stored knowledge in an automated fashion. Over the last couple of decades, the field has matured considerably. Several IR systems are used by a wide variety of users on an everyday basis nowadays such as web search engines, junk-mail filters and news clipping services (A. Singhal, 2001). However, information retrieval techniques can be utilized on many other fields as well, one of which will be introduced in the next section.

Music Information Retrieval (MIR) is an interdisciplinary science that has its roots in the field of information retrieval, musicology and music psychology. It is an emerging field of research that aims to satisfy users' *music* information needs (N. Orio, 2006). Many people find themselves in the situation where they are not able to recall the name of a song of which they can only remember fragments. One option to tackle this is to query an online search engine with (parts of) the lyrics. Other services allow melodic queries, which in turn present a collection of songs that might include the right one. These are just a few examples of MIR techniques in practice, for people trying to find the name of a song are only a small group of users of MIR technology. For example, composers and songwriters may question where their inspiration has come from, musicians might be interested in finding alternative arrangements of a particular piece and forensic musicologists would analyse songs for copyright infringement lawsuits (A.L. Uitdenbogerd, J. Zobel, 2004).

MIR systems can utilize various approaches. One popular approach is content-based music information retrieval. Content-based music information retrieval works with actual music pieces: given a piece of music, the system should be able to retrieve similar music from a database only based on the content of that musical piece. In recent years, more and more researchers are working with this approach (C. Wang,, J. Li, S. Shi, 2002). Interesting as it is, this approach can be rather difficult, since the content of music consists of numerous aspects such as pitch, key, duration, temporal-, harmonic- and textual- aspects (Downie, 2003). Another approach is to retrieve music information by means of *tags*. In contrast to content-based MIR, these systems make use of notated information on music, such as the performing artist, song title, album name, year of release.

2.1 HIT SONG SCIENCE

In the field of Hit Song Science, the intention is to detect, or rather predict, whether a song has the potential to become a commercial hit song, thus reaching the top of the record charts. Quite a few studies have already been done on this topic, some of which focused on

extracting general acoustic and lyrical features from songs and then use standard classifiers to separate hits from non-hits (R. Dhanaraj, B. Logan, 2005). Another approach was to predict potential hit songs by using data mined from a music social network and the relationships between tracks, artists and albums (K. Bischoff, C.S. Firan et al., 2009). These studies produced some promising results, but there are still plentiful other approaches to follow. For instance, one could attempt a combination of the acoustic/lyrical features and social features.

The ability to detect potential hit songs would have an incredible impact on the music industry. There are online services, such as uPlaya (http://www.uplaya.com), which claim to have already succeeded in Hit Song Science, promising that *Hit Song Science™ provides immediate feedback on your song's potential for commercial success and instant legitimization in the market for high-scoring music.* This particular online service states that their rating system provides immediate feedback on the quality of music, its competitive edge in the music industry, and its reception among professionals and music lovers. However, even though uPlaya provides objective feedback on a song's hit potential by comparing it against hit songs from the past, it does not *predict* whether a song will become a hit: it returns an evaluation based on certain 'hit characteristics'.

There is still lots of room for improvement on hit song prediction, from which there is much to gain. First of all, it would greatly support record companies in pinpointing songs and artists that have the most potential of scoring a hit song. Furthermore, it could enhance music recommendation services in determining which songs and artists to recommend.

2.2 LAST.FM

> **scrobble**: skrob·bul
>
> [verb] To automatically add the tracks you play to your Last.fm profile with a piece of software called a Scrobbler
>
> 1. *If I'm not scrobbling the music I hear, it doesn't count!*

Founded in the United Kingdom in 2002, Last.fm is a music recommendation website that allows its users to maintain a music profile. The service is free to use by signing up on the website and downloading the Last.fm Scrobbler. With this little program, the Last.fm website is able to build a detailed profile of one's musical preferences by scrobbling every song the user listens to. A scrobble is a little note the Scrobbler sends to Last.fm, containing details of the song the user is listening to such as artist, title and a timestamp. This process of scrobbling helps Last.fm tell its user what songs are listened to the most, which artists have been listened to for a certain period of time, which other users have the most similar taste and many more features (http://www.last.fm). All of these features are displayed on one´s profile page, found within the Last.fm website. The profile pages enable users to browse through their own, or each other´s music profile such that a user can find other users that have a similar taste, from which the user may extract ideas for other artists and songs to listen to.

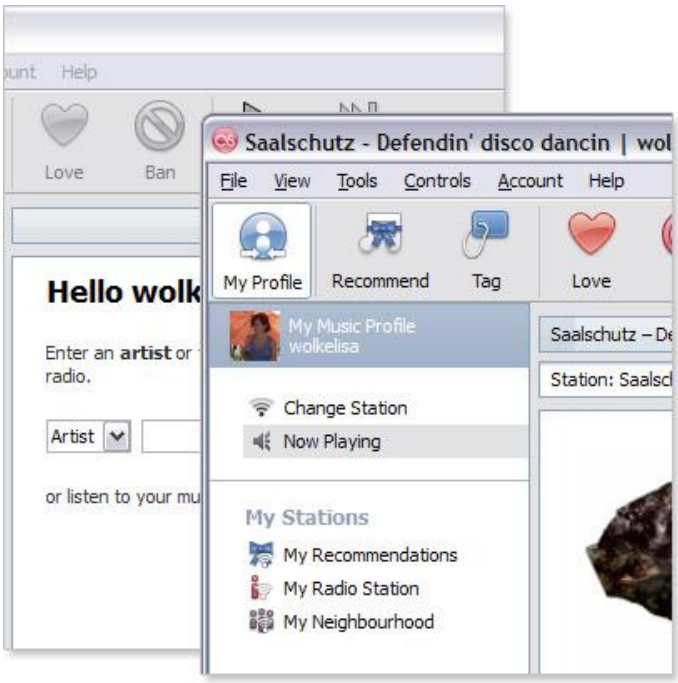

Figure 1. The Last.fm Scrobbler

The notion of 'musical profile' can be interpreted in various ways. Last.fm seems to define this concept with the following key features: *Recently Listened Tracks*, which displays every single track you have ever scrobbled including a timestamp; a personal *Library*, which shows all the artists that are included in the user's profile, including a count of how many times an artist was listened to; and the *Top Tracks* display the songs that have been listened to the most. Along with these key features, the profile pages provides additional information such as personal information, upcoming events that the user is attending, and other users that have a similar music taste called 'neighbours'.

The profile page is just the beginning however. In addition, Last.fm provides a dashboard, customized to the user, which contains personal recommendations for artists, upcoming events, groups and neighbours. These recommendations allow users to browse for other music and listen to previews. Last.fm even provides an online 'recommendation radio station' which plays songs you might like based on your personal taste.

Over the years, the Last.fm community grew vastly and by March 2009, Last.fm claimed to have 30 million active users. Each minute, thousands of songs are scrobbled by numerous users and so the Last.fm database is getting bigger by the day. By April 2011 Last.fm reported to have more than 50 billion scrobbles (B. McCarty, 2011). With all this information freely available, Last.fm could prove to be an excellent source for performing Hit Song Science studies.

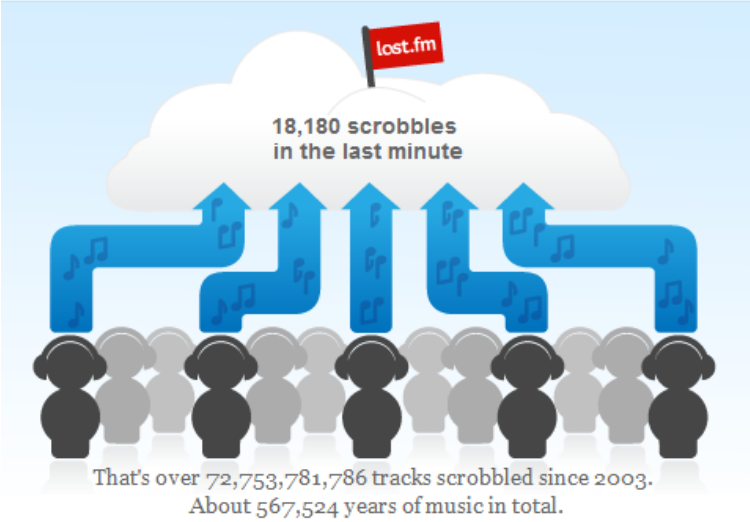

Figure 2. Last.fm's growing database

2.3 RESEARCH

In the field of Hit Song Science, various studies tried to achieve hit song prediction in multiple ways: using acoustic and lyrical features, or from a social-driven perspective. For my research I will continue this search for a hit song prediction method from a somewhat social point of view. My approach is to search for people who listen to hit songs *before* these songs actually reach the top of the record charts. This approach can be characterized by the following steps:

1. First, a number of hit songs, coming from the record charts, will be identified.
2. The next step is to retrieve all scrobbles from 900 users (divided over three subject groups) from the Last.fm database over the period of four weeks preceding the date that the specific song entered the record charts to become a hit song
3. Subsequently, I hope to find scrobbles of that song, indicating that some users listened to it before it actually became a hit song, labelling that user as a 'predictor'.
4. From this information, I will attempt to construct a model that represent the characteristics of these users, such that I might formulate predictions about which songs have the potential to become hit songs in the near future.

From this approach, I have come to formulate the following research question:

To what extent can hit songs be predicted
through the listening behaviour of Last.fm 'predictors'?

For this research, I will look into hit songs and users from the Netherlands, the United Kingdom and the United States separately. The hit songs will be defined through specific national record charts, and the users will be filtered from the Last.fm community. From this approach, I hope to find a viable indication that Last.fm listeners can predict potential hit songs to some degree. My expectation is that this might turn out to be quite difficult. Regardless of the outcome, this research will lay down a foundation for further research on this approach.

3 APPROACH

3.1 THE LAST.FM API

In order to retrieve the information needed for my research, I will need to collect a lot of data from the Last.fm website. Last.fm provides an API (Application Programming Interface) that supports requests for web services, which allows anyone to build programs using data from the Last.fm database. In turn, the Last.fm API allows anyone to call a wide variety of methods that respond in a Last.fm-idiom XML.

To get started, you need to sign up to get an API account, which includes an API key that is required to use the Last.fm web services. The next thing you need to know is that the API root URL is located at:

http://ws.audioscrobbler.com/2.0/

Then, in order to send a request, you will send a method parameter for the specific method you want to call, as well as the API key assigned to your account. Some methods require additional arguments for their own, and some have a few optional arguments. For example, the method `artist.getSimilar` requires the additional argument *artist*, which is the name of the artist you wish to request similar artists of. For example, the request for calling this method for the artist Coldplay would look as follows:

http://ws.audioscrobbler.com/2.0/?method=artist.getsimilar&artist=coldplay&api_key=...

And for this example, the XML response would look as follows:

```
<similarartists artist="Coldplay">
      <artist>
            <name>Keane</name>
            <mbid>c7020c6d-cae9-4db3-92a7-e5c561cbad50</mbid>
            <match>1</match>
            <url>www.last.fm/music/Keane</url>
            <image size="small">http://userserve-
            ak.last.fm/serve/34/74942332.png</image>
            <image size="medium">http://userserve-
            ak.last.fm/serve/64/74942332.png</image>
            <image size="large">http://userserve-
            ak.last.fm/serve/126/74942332.png</image>
            <image size="extralarge">http://userserve-
            ak.last.fm/serve/252/74942332.png</image>
            <streamable>1</streamable>
      </artist>
      ...
</similarartists>
```

These XML responses contain the information you requested with a specific method. For this example, the XML shows us a list of artists that are found to be similar to Coldplay, indicated by a similarity value between 0 (not similar) and 1 (very similar) for each given artist. According to the response by Last.fm, the artist named Keane should be the most similar to Coldplay, scoring the highest similarity value. In addition, the XML returns extra information such as the URL on which the artist page can be found, a unique id for the artist, multiple images in various sizes and some statistics such as playcount or number of listeners. Naturally, the amount of information will vary among the methods.

3.1.1 API METHODS

The example above shows just one of the many methods provided by the Last.fm API. All of the methods are organized in categories such as *artist*, *track*, *group* and *user*. Each category then contains a number of available methods such as *artist.getTags*, *group.getMembers* and *user.getFriends*. For this research, only a handful of methods are needed for retrieving the essential data. These methods are:

- *user.getInfo*
  - Requests the information for a given username. The information includes the user's real name, gender, age, country and playcount (total number of scrobbles made by the user).
- *track.getInfo*
  - Requests the information for a given track. The information includes the artist, track name, duration and playcount (total number of scrobbles made for this specific track).
- *user.getRecentTracks*
  - Requests the scrobbles for a given user. For this method, the optional parameters *from* and *to* allow you to request the scrobbles for a certain period. The response consists of the tracks scrobbled by the user, with a maximum of 200 tracks.

3.1.2 LAST.FM API JAVA BINDINGS

These methods allow me to request the data needed for this research. However, it still leaves me to decipher the response, by means of parsing the XML. Conveniently, there are numerous unofficial API Tools that facilitate the use of the Last.fm API, although they are not supported by the Last.fm website. Among these tools are the Last.fm API bindings for Java (http://www.u-mass.de/lastfm), which provides classes and methods to invoke the Last.fm API methods in a Java interface. This allows you to request all the information from Last.fm through the Java interface, returning object-oriented results instead of XML responses.

3.2 GATHERING THE SUBJECTS

The data for this research will be gathered from three groups, being: users from the Netherlands (NL), users from the United Kingdom (UK) and users from the United States (US). Each group consists of 300 random users from the Last.fm website. Although the Last.fm API provides a wide variety of methods to retrieve information, there is no method

available to retrieve random users, let alone to retrieve random users from a specific country.

One option to solve this matter is to take a look into *groups*. Last.fm allows its community to create groups. Generally, these groups have a specific background and are free to join by any user on Last.fm. For instance, there is a group called *I Still Buy CDs*, which holds more than 75 thousand members and is 'meant' for people who actually still buy CDs. Within the Last.fm API, there is method called *group.getMembers* which allows you to retrieve all the users that have joined the specific group. Coincidentally, there is a group called *Nederlanders!* which is meant to be *a group for all the Dutch scrobblers*. There are a few catches for this approach however. First, as I mentioned, groups are free to join by anyone, and thus the group *Nederlanders!* could very well hold members from outside of the Netherlands. Second, a group does not provide information on a user's recent activity, so you might be gathering users that stopped using Last.fm a long time ago. Conclusively, the risks for this approach are too big to take the chance.

Alternatively, another option is to make use of the *Recently Active Users* (http://www.last.fm/community/users/active) pages on the Last.fm website. These ten pages contain twenty random users each, who have been scrobbling tracks recently. Thus, users that are displayed on these pages have a fair chance of being considerable active users. Then again, the Last.fm API does not provide a method to retrieve these users. Luckily, Dr. L. G. Vuurpijl provided me with a script that collected all the users from these pages every fifteen minutes, which resulted in a total of more than 15 thousand users. From this vast amount of random users, I managed to filter 300 users for each of the three groups separately.

Taking a closer look at the subjects, we can learn a couple of things on the distribution in age and gender. Although the majority of the subjects have filled out their age, it seems highly improbable that there are Last.fm users aged 102 years old. In order to make the distribution more probable, I have filtered out these anomalies among the ages, taking only ages from 15 to 65 into account. This resulted in the following distribution:

AGE			
	US	**UK**	**NL**
Min	16	16	16
Avg.	25	27	24
Max	64	61	63
Undef.	84	82	58

Table 1. Age distribution

GENDER			
	US	**UK**	**NL**
Male	189	200	193
Female	82	67	86
Ratio (F:M)	1 : 2.30	1 : 2.98	1 : 2.24
Undef.	29	33	21

Table 2. Gender distribution

Looking at these tables, we can see that that the ages are still quite varied, ranging from 16 to 64. For each of the three groups, the average age lays around 25 years, but there are still a fair amount of users who did not specify their age, or filled out an age that does not seem right. The graph below provides a clear image on the age distribution. Taking a look at the gender distribution, it becomes obvious that the male subjects out-represent the female subjects. However, considering these groups consist of 300 random subjects each, they should be a legitimate representation of users from the Last.fm community.

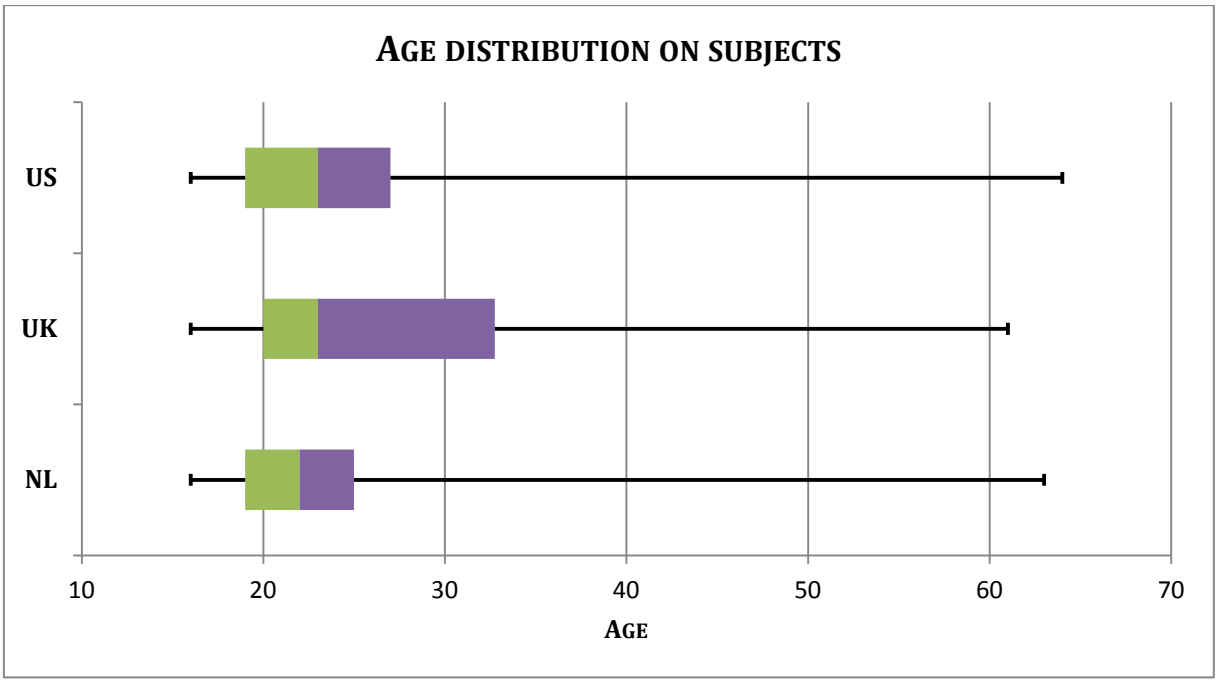

Figure 3. Box plots on age distribution

The box plots provide a clear illustration on the age distribution within each of the three groups. It becomes evident that most of the users are somewhere between twenty and thirty years old. Although it is quite probable that these distributions are slightly off, due to some users not filling out their age, or filling out an age that does not seem valid, these plots do give a valid indication on age distribution.

3.3 DEFINING THE HITS

The next step is to define the hit songs I want to search for among the subjects. But then first I will have to define the term 'hit song'. The Guinness Book of British Hit Singles, which was first published in the 1970s, states that "*a single is usually considered to be a music hit (hit song) when it has reached the official Billboard Magazine's Hot 100 or the UK Singles Chart Top 75 and stayed there for at least one week*". This is a rather arguable definition for my specific research, because from my point of view, a single that has lingered in the bottom of a record chart for 5 weeks should be considered 'less of a hit song than a single that reached the top 5 of a record chart and stayed there for 10 weeks.

This 'degree' in hit songs is something to take into account when picking the hit songs I intend to use for this research. Considering that the goal is to predict actual 'top-of-the-charts' hit songs, I will hold on to a few criteria when defining the hit songs. These criteria are:

- the single held a top 5 position for a minimum of 5 weeks
- the single has been in the record chart for a minimum of 10 weeks

Furthermore, whether or not a single becomes a hit song can differ per continent and even per country. It happens that an artist from the US is immensely more popular in the UK than in its native country, which means record charts in the US can be completely different

than those in the UK. For this reason, I will have to make a clear distinction in hit songs between the three subject groups and so I will take on the most renowned national record chart for each country.

The hit songs will be defined through the *Nederlandse Top 40* (Stichting Nederlandse Top 40, NL), *the Official UK Singles Top 40* (The Official Charts Company, UK) and the *Billboard Hot 100* (Billboard Magazine, US) respectively. Both the charts for the Netherlands and the UK compile their listing through record sales, airplay and download sales. However, the US chart does not take download sales into account. Even though this distinction seems a bit warped, it does not affect my definition of a hit song.

With a clear definition on the term hit song, and specific record charts assigned to each country, the following hit songs have been chosen for this research:

NL – NEDERLANDSE TOP 40				
Artist	**Title**	**Date of Entry**	**P.P**	**W.C.**
PSY	Gangnam Style	01-09-2012	1	18
Gusttavo Lima	Balada	28-04-2012	1	23
Carly Rae Jepsen	Call Me Maybe	24-03-2012	2	29
Lykke Li	I Follow Rivers	24-12-2011	2	28
Gers Pardoel	Ik Neem Je Mee	15-10-2011	1	33

Table 3. Hit Songs for the Netherlands, based on the Nederlandse Top 40

UK – OFFICIAL UK SINGLES TOP 40				
Artist	**Title**	**Date of Entry**	**P.P**	**W.C.**
PSY	Gangnam Style	15-09-2012	1	16
Maroon 5 & Wiz Khalifa	Payphone	23-06-2012	1	12
Flo Rida	Whistle	09-06-2012	2	14
Jessie J	Domino	31-12-2011	1	23
Rihanna & Calvin Harris	We Found Love	08-10-2011	1	33

Table 4. Hit Songs for the UK, based on the Official UK Singles Top 40

US – BILLBOARD HOT 100				
Artist	**Title**	**Date of Entry**	**P.P**	**W.C.**
Bruno Mars	Locked Out Of Heaven	20-10-2012	1	13
Maroon 5	One More Night	18-08-2012	1	22
Ellie Goulding	Lights	12-05-2012	2	32
Carly Rae Jepsen	Call Me Maybe	10-03-2012	1	36
Rihanna & Calvin Harris	We Found Love	08-10-2011	1	35

Table 5. Hit Songs for the US, based on the Billboard Hot 100

In these tables, P.P. is the peak position (the highest position that has been reached during the period it was in the record chart) and W.C. is the week count (the number of weeks it has been in the record chart, on any position). Each hit has been chosen carefully, making sure they meet the defined criteria. With a total of fifteen hits, I hope to be able to gather

enough data which would lead me to interesting results. And so, the next step is to gather that data.

3.4 DATA RETRIEVAL

Now that the hit songs have been defined and the subject groups have been gathered, all the prerequisites for collecting the data are met. With the use of the Last.fm API Java Bindings, I was able to write a fairly straightforward program that would collect the necessary data. Since I have defined five hit songs for each of the three subject groups, the program will need to run a total of fifteen batches, because the information needs to be retrieved over specific periods of time. In pseudo-code, the program looks as follows:

```
FOR each user in the group (#300)
{
	REQUEST the information for the current user

	WRITE user.name
	WRITE user.age
	WRITE user.country
	WRITE user.gender
	WRITE user.playcount

	FOR each day in the specified period of four weeks (#28)
	{
		REQUEST the list of tracks for current user on current day

		FOR each track in the list of tracks (#??)
		{
			REQUEST the information for the current track

			WRITE track.artist
			WRITE track.title
			WRITE track.date&time
		}
	}
}
```

The pseudo-code shows that the program uses three specific methods from the Last.fm API, as I mentioned earlier. This is what happens each batch: it loops through each of the users (with a total of 300 users within a subject group). First, it requests the information for the current user through the *user.getInfo* method. The information from this request is written to a text file. Then, the program loops through the specified period of four week with a frame of 24 hours (thus, having a total of 28 frames per user). For each frame, the method *user.getRecentTracks* requests the list of all tracks that have been scrobbled by the specific user (within the current frame). Finally, for every track in the list (which is an uknown amount of tracks, with a maximum of 200), the information is requested through the *track.getInfo* method. This information is then added to the text file. When a batch is completed, all of the data for the specific hit song is stored in 300 separate text files, one for each user. The collection of retrieved data is discussed in section 5.

4 METHODS

4.1 DATA PROCESSING

Before I can commence on the analysis, I will need to prepare the information by means of data processing. The raw data is distributed over 300 files per hit song, of which there are fifteen in total. Each of these files contains an unknown number of scrobbles, with a maximum of 200 scrobbles per day (from a period of 28 days). The idea is to find the targeted hit song within the data.

The first step is to find those subjects that have indeed scrobbled the hit song I am looking for, which indicates that they have listened to the hit song before it reached the record charts. A rather simple program, written in Java, crawls through each of the 300 files, processing each track by artist, title and date & time. Whenever the crawler finds the specific hit song, the information from that scrobble is written to a new text file, which keeps track of all the hit scrobbles. This information can be represented in graphs to give a clear representation on the amount of scrobbles of the hit song. With a little luck, the crawler will be able to find a number of subjects that have listened to the specific hit song. These subjects will be labelled as *predictor*. These predictors will be combined, forming a group of predictors from each of the original subject groups. These groups of predictors will function as the target subject groups in the analysis.

4.2 ANALYSIS

Once the data has been processed, and the groups of predictors have been composed, the next step is to make preparations for the analysis. First, in order to be able to evaluate the predictions made by the groups of predictors, it is essential to compose control groups. For each group of predictors (US, UK & NL), there will be formed four control groups, containing random users from the original subject group of 300 users. In composing the groups, it will be made sure that the groups do not contain any of the subjects that are labelled as predictor. On top of that, the control groups will be equal in size to its designated group of predictors and it will made sure that the random users provide data to work with (to prevent gathering random users that have no scrobbled tracks at all).

The analysis will be done on a new batch of data, regarding a new period of time. The idea is to gather data for each group for the period of January 1st through January 31th (2013). This way, the analysis is conducted on data from a period of time that has not been looked upon before, making it a *test set*. The first step in the analysis is done by means of pre-processing the data. Through a simple program, each unique track is indexed throughout all of the data from a subject group. The program indexes each unique track, counting the number of times it was scrobbled by all of the users and the number of users that scrobbled the track at least once. Once all of the tracks are indexed, each track will be given a prediction value by calculating its *term frequency-inverse document frequency* value, and find the maximum value by combining the tf-idf values for each unique track.

4.2.1 TERM FREQUENCY-INVERSED DOCUMENT FREQUENCY

The term frequency-inversed document frequency reflects the importance of a term with reference to its collection of documents. Tf-idf is a numerical statistic that is often used in information retrieval and text mining, and proved to serve well in predictive models (F. Sebastiani , 1999). In essence, tf-idf works by determining the relative frequency of terms in a specific document compared to the inverse proportion of that word over the entire collection of documents. (J. Ramos, 2000). The tf-idf function can have some minor differences per application, but the overall approach works as follows. Given a document collection D, a term t, and an individual document $d \in D$, we calculate

$$t_d = f_{t,d} * \log(|D| \,/\, f_{t,D})$$

where $f_{t,d}$ is the number of times t appears in document d divided by the total amount of terms within the document, $|D|$ equals the size of the collection of documents, and $f_{t,D}$ is the number of documents in which t appears in D (G. Salton, C. Buckley, 1998)(A. Berger et al, 2000).

Translating this into the current research: each scrobble represents a term t, each subject (stored in a separate file) represents a document d, and all of the subjects (files) combined represent the collection of documents D. When calculating the tf-idf value for each of the scrobbles, we basically determine the relevancy of that scrobble in relation to the collection of documents. Once all the tf-idf values are calculated, we can return those scrobbles that maximize the following equation:

$$\text{MAX} \sum_i t_{i,d}$$

Using this traditional implementation of the tf-idf, the returned scrobbles serve as the predictions from each of the subject groups. These predicted tracks will be evaluated through the official record charts, checking to what extent the prediction is correct. The next section provides some more information on how the tf-idf algorithm computes the predictions for this analysis.

4.2.2 PREDICTION

As explained in the previous section, the tf-idf values determine the relevancy of a song in relation to the entire collection of scrobbles from all of the users. In essence, this is how the prediction value is calculated, illustrated in a more practical example:

1. First, the number of occurrences of a song (for the current user) is determined
   - For instance, user *x* has listened to *I Follow Rivers* by Lykke Li twelve times
2. Then, the term-frequency is calculated by dividing the number of occurrences by the total amount of song in the collection
   - For instance, if user *x* has scrobbled 150 tracks, the tf value will be `12/150 = 0.080`
3. Next, the inversed document frequency is calculated by taking the logarithm of [the total number of documents / the number of documents in which the scrobble occurs]
   - For instance, user *x* is part of a subject group that holds 100 users. On top of that, say there are a total of 7 users in this group that have listened to the song *I Follow Rivers*. The idf value will then be `log(100/7) = 1.155`

4. The following step is to complete the function by multiplying the tf value with the idf value.
   - The tf-idf value for *I Follow Rivers* on user *x* would be `0.080 * 1.155 = 0.092`
5. This calculation is done for each song from each user in the subject group. The last step is to combine the corresponding songs among the users by summing their tf-idf values. This is the final predictive value. The actual prediction is made by taking the songs with the highest predictive values.

One of the advantages of the tf-idf algorithm is that counterbalances scenarios in which one user listens a specific track over and over. This is compensated by dividing the number of occurrences by the total number of song in the collection. Another advantage is that it calculates the corresponding occurrences of a track within the collection of users, such that tracks that are listened to by a greater number of users will have a higher idf value. I think this method will prove to be a good first attempt to predict hit songs in this fashion because of these features.

5 DATA

5.1 RAW DATA

The collection of data was retrieved by running fifteen batches, one for each of the defined hit songs. In order to retrieve this data, the program sent a total of 126,000 requests and considering each request could retrieve up to 200 scrobbles, the amount of data to be processed seems reasonable. The graphs below should provide some indication on the size of the entire raw data collection.

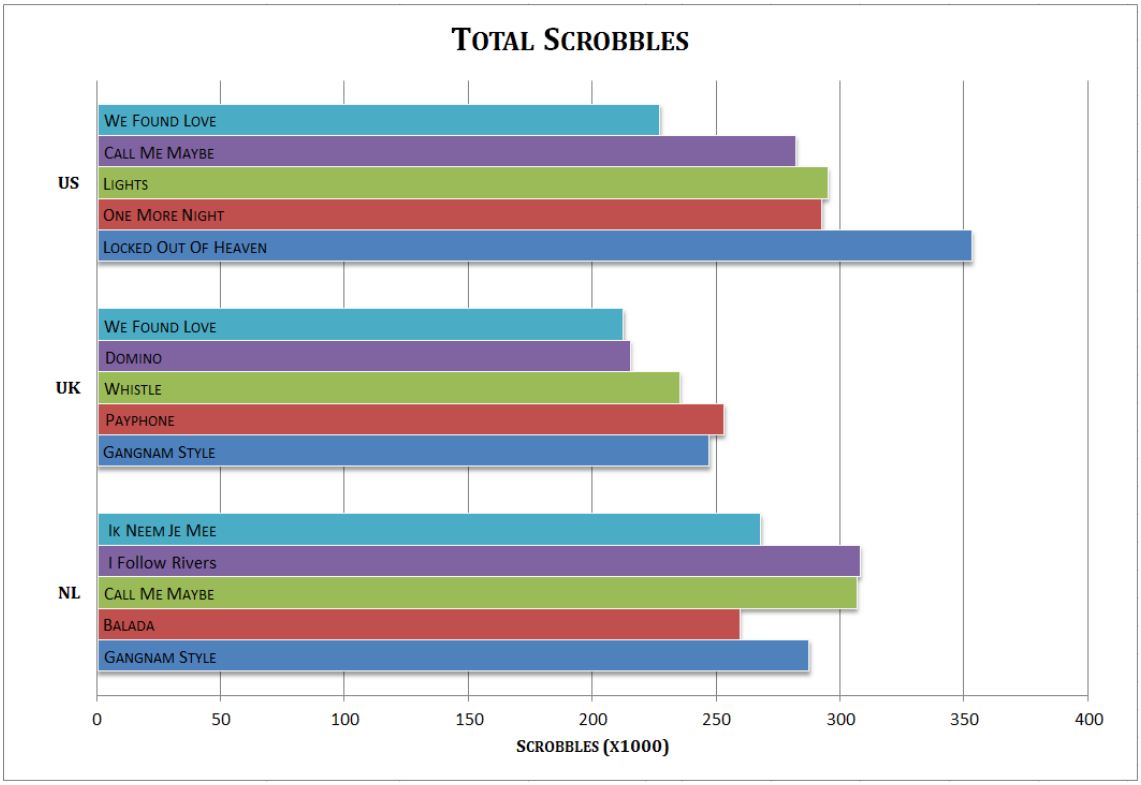

Figure 4. Total amount of scrobbles per hit song

This graph shows the combined amounts of scrobbles for each group. Each bar shows the total amount of scrobbles (for the complete subject group of 300 users) for one of the fifteen hit songs. We can see that the UK group scrobbled the least amount of tracks (approximately 1,16 million scrobbles altogether), while the amount of scrobbles for the US group and the NL group are reasonably higher (about 1,45 million scrobbles for the US group, and 1,43 million scrobbles for the NL group). If we take the sum of all the scrobbles, it comes to a total of almost 4,04 million gathered scrobbles.

It shows that there is a decent quantity of data to work with, considering *each* user scrobbled somewhere between 29 and 43 tracks *per day* on average. However, it must be taken into account this is an *average*: the subject groups will surely include a number of listeners that may very well have scrobbled no tracks at all. Once the data has been processed, the distribution of 'hit'-listeners should become more clear.

5.2 PROCESSED DATA

In order to find the subjects that have listened to the specific hit song, the subject groups are filtered based on tracks they have scrobbled for each of the fifteen hit songs separately. First, the subjects are filtered on whether they have listened to the *artist* of the targeted hit song. This should give a first indication on the number of subjects that actually listen to the target music genre. From there, the next step is to find out how many subjects actually listened to the target hit song. This is achieved through another round of filtering.

US – NUMBER OF SUBJECTS FILTERED BY ARTIST/TRACK		
Hit Song	**Number of subjects that scrobbled the ARTIST**	**Number of subjects that scrobbled the TRACK**
We Found Love	15	3
Call Me Maybe	1	1
Lights	17	8
One More Night	24	1
Locked Out Of Heaven	12	1

Table 6. The number of subjects filtered from scrobbles on artist/track (US)

UK – NUMBER OF SUBJECTS FILTERED BY ARTIST/TRACK		
Hit Song	**Number of subjects that scrobbled the ARTIST**	**Number of subjects that scrobbled the TRACK**
We Found Love	24	7
Domino	33	4
Whistle	12	1
Payphone	23	4
Gangnam Style	0	0

Table 7. The number of subjects filtered from scrobbles on artist/track (UK)

NL – NUMBER OF SUBJECTS FILTERED BY ARTIST/TRACK		
Hit Song	**Number of subjects that scrobbled the ARTIST**	**Number of subjects that scrobbled the TRACK**
Ik Neem Je Mee	5	5
I Follow Rivers	37	21
Call Me Maybe	11	11
Balada	3	3
Gangnam Style	0	0

Table 8. The number of subjects filtered from scrobbles on artist/track (NL)

These tables show the results from the data processing by means of filtering. It shows that from each group of 300 subjects, there seem to be a handful of subjects that listen to the designated hit song before it entered the record charts. On average, there are just under eight subjects per hit that deserve the label of *predictor*, while the number of subjects that listen to the artist of interest is even higher. However, each of the five hit songs per subject group are taken from the same 300 subjects (per subject group), which means that there might be duplicates: subjects that have listened to two or more of the targeted hit songs

before these became a hit song in the record charts. Thus, the final step is to combine the found predictors and to remove these duplicates. The remaining subjects will form the final group of predictors for each subject group separately.

	TOTAL AMOUNT OF UNIQUE PREDICTORS
US	13
UK	8
NL	29

Table 9. The total amount of unique subjects that scrobbled a target hit song

The table above shows us the final amount of unique subjects that have been labeled as *predictors*. Coming from groups of 300 subjects, these numbers may seem rather low, but we have to take into account that there are countless users that listen to other music genres.

5.3 NEW DATA

Now that the original data has been processed, it is essential to gather a new collection of data. This new data will regard a novel period of time, which is from the 1st of January to the 31st of January. For each subject group, new data will be collected for both the group of predictors as well as the four control groups. This data will have the same structure as the initial data, and will be analyzed by means of the tf-idf function. Through this algorithm and the maximization function, as explained in section 4.2.1, will produce at ranked list of predictions. For the results, the top three will be evaluated to construct an answer to the research question.

6 RESULTS

6.1 HIT PREDICTION FOR NL

PREDICTORS TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Tom Odell – Another Love	0.38805	15-12-2012	6	12-01-2013
Villagers – Nothing Arrived	0.26860	NO RESULTS	NO RESULTS	NO RESULTS
Darin – Playing With Fire	0.26021	08-02-2013*	28*	08-02-2013*

Table 10.1 Top three predictions for the group of predictors (NL)

CONTROL GROUP #1 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Solange Knowles – Losing You	0.12981	18-01-2013*	9*	25-01-2013*
Pink Floyd – Comfortably Numb	0.11286	NO RESULTS	NO RESULTS	NO RESULTS
Bat For Lashes – Laura	0.11111	NO RESULTS	NO RESULTS	NO RESULTS

Table 10.1 Top three predictions for control group #1 (NL)

CONTROL GROUP #2 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Kodaline – All I Want	0.16087	10-11-2012	26	10-11-2012
M83 – Midnight City	0.07907	17-06-2012	8	09-09-2102
Guy Farley – Modigliani Suite	0.07239	NO RESULTS	NO RESULTS	NO RESULTS

Table 10.3 Top three predictions for control group #2 (NL)

CONTROL GROUP #3 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Avril Lavigne – Fall To Pieces	0.33595	NO RESULTS	NO RESULTS	NO RESULTS
While She Sleeps – Our Courage, Our Cancer	0.10720	NO RESULTS	NO RESULTS	NO RESULTS
Ben Howard – Keep Your Head Up	0.09982	23-06-2012	14	25-08-2012

Table 10.4 Top three predictions for control group #3 (NL)

CONTROL GROUP #4 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Binford – Slate	0.20171	NO RESULTS	NO RESULTS	NO RESULTS
Alex Clare – Hands Are Clever	0.18948	NO RESULTS	NO RESULTS	NO RESULTS
Redlight – Lost In Your Love	0.15061	11-08-2012*	5*	11-08-2012*

Table 10.5 Top three predictions for control group #4 (NL)

Tables 10.1 through 10.5 show the results from the analysis on the groups from the Netherlands. Each table contains the top three tracks that maximized the tf-idf function, which is interpreted as the most relevant track among the collection of documents. The tf-idf values are shown as well. Then, for each track, the record chart information is displayed, which incorporates the date of entry, the peak position, and the date on which the track was on this peak position. Tracks that have no entry on the record charts are indicated with 'NO RESULTS'. Tracks that have an entry in a record chart *other* than the targeted record charts (i.e. a record chart from another country) are indicated by the * symbol.

Starting off with the group of predictors, it shows that the highest scoring track, which is the song *Another Love* by *Tom Odell* entered the record charts on the 15th December of 2012, which precedes the targeted period. However, it did not reach its peak position (no. 6) until the 12th of January of 2013, which is roughly halfway the target period. Another noteworthy track is the song *Playing With Fire* by *Darin*. Although this track entered the record charts a week after the target period, it is not necessarily a hit, because its peak position was only no. 28. Furthermore, indicated by the * symbol, it did not have an entry in the Dutch record charts.

When we look upon the results for the control groups, many tracks did not even have an entry in record charts. For the tracks that did have an entry in record charts, only one track jumps out, which is the song *Losing You* by *Solange Knowles* (in control group #1). This track did become some degree of a hit late in the target period (18th – 25th of January), although it only appeared in Danish record charts instead of Dutch record charts.

6.2 HIT PREDICTION FOR US

PREDICTORS TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Marina & The Diamonds – Primadonna	0.05905	21-04-2012*	6*	28-04-2012
Sky Ferreira – Everything Is Embarrassing	0.05418	NO RESULTS	NO RESULTS	NO RESULTS
Rihanna – Pour It Up	0.04973	09-02-2013	9	23-03-2013

Table 11.1 Top three predictions for the group of predictors (US)

CONTROL GROUP #1 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
One Direction – Magic	0.06602	NO RESULTS	NO RESULTS	NO RESULTS
Rascal Flatts – Banjo	0.03978	NO RESULTS	NO RESULTS	NO RESULTS
Leona Lewis – Glassheart	0.03375	NO RESULTS	NO RESULTS	NO RESULTS

Table 11.2 Top three predictions for control group #1 (US)

CONTROL GROUP #2 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
The Surfaris – Wipe Out	0.07724	NO RESULTS	NO RESULTS	NO RESULTS
Dave Matthews Band – Mercy	0.05091	NO RESULTS	NO RESULTS	NO RESULTS
Dave Matthews Band – Crash Into Me	0.04967	21-07-1997	24	08-09-1997

Table 11.3 Top three predictions for control group #2 (US)

CONTROL GROUP #3 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Ronald Jenkees – Throwing Fire	0.03251	NO RESULTS	NO RESULTS	NO RESULTS
Donnie McClurkin – Holy	0.02970	NO RESULTS	NO RESULTS	NO RESULTS
Darwin Hobbs – Glorify Him	0.02970	NO RESULTS	NO RESULTS	NO RESULTS

Table 11.4 Top three predictions for control group #3 (US)

CONTROL GROUP #4 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Chelsea Grin – S.H.O.T. (1)	0.22278	NO RESULTS	NO RESULTS	NO RESULTS
Attila – Make It Sick	0.22278	NO RESULTS	NO RESULTS	NO RESULTS
Attila – Outlawed	0.22278	NO RESULTS	NO RESULTS	NO RESULTS

Table 11.5 Top three predictions for control group #4 (US)

Tables 11.1 through 11.5 show the results from the analysis on the groups from the United States. Looking upon the results for the group of predictors, the song *Pour It Up* from *Rihanna* seems promising. Considering it entered the record charts on the 9th of February, and reached its peak position (no. 9) on the 23rd of May, you could say that the track was actually correctly predicted. As can be seen in the remaining tables, most of the other tracks showed no results in the record charts, indicating that the control groups had no success in predicting a hit song.

(1) One remark that should be made is the occurrence of corresponding tf-idf values for the tracks from control group #4. This can be blamed on the fact that the subject groups counted only 13 subjects per group. In the case where one of the subjects listened to a small number of track, each of these tracks receive a rather high tf-idf values, because its term frequencies remain considerably high.

6.3 HIT PREDICTION FOR UK

PREDICTORS TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Funeral For A Friend – Nails [2]	0.07776	NO RESULTS	NO RESULTS	NO RESULTS
Haim – Don't Save Me	0.02110	12-01-2013	28	12-01-2013
The Saturdays – What About Us	0.02103	23-03-2013	1	23-03-2013

Table 12.1 Top three predictions for the group of predictors (UK)

CONTROL GROUP #1 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Skarlett Riot – Villain	0.06673	NO RESULTS	NO RESULTS	NO RESULTS
Squeeze – Is That Love?	0.03283	NO RESULTS	NO RESULTS	NO RESULTS
Squeeze – Up The Junction	0.03283	NO RESULTS	NO RESULTS	NO RESULTS

Table 12.2 Top three predictions for control group #1 (UK)

CONTROL GROUP #2 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
We Came As Romans – Glad You Came	0.05043	NO RESULTS	NO RESULTS	NO RESULTS
Sleeping With Sirens – Fuck You	0.04399	NO RESULTS	NO RESULTS	NO RESULTS
Sleeping With Sirens – Scene Two	0.04227	NO RESULTS	NO RESULTS	NO RESULTS

Table 12.3 Top three predictions for control group #2 (UK)

CONTROL GROUP #3 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Mindless Behavior – Mrs. Right	0.04511	NO RESULTS	NO RESULTS	NO RESULTS
Taylor Swift – I Knew You Were Trouble	0.03458	13-10-2012	2	12-01-2013
Will.I.Am – Scream & Shout	0.03457	15-12-2012	1	12-01-2013

Table 12.4 Top three predictions for control group #3 (UK)

CONTROL GROUP #4 TRACK	TF-IDF	DATE OF ENTRY	PEAK POSITION	DATE OF P.P
Massive Attack – Angel	0.16419	NO RESULTS	NO RESULTS	NO RESULTS
Tony Wakeford – A Rose In Hell	0.08209	NO RESULTS	NO RESULTS	NO RESULTS
Fire + Ice – Take My Hand	0.08209	NO RESULTS	NO RESULTS	NO RESULTS

Table 12.5 Top three predictions for control group #4 (UK)

Tables 12.1 through 12.5 show the results from the analysis on the groups from the United Kingdom. Like the other subject groups, the group of predictors shows a promising track, which is *What About Us* by *The Saturdays*. This track entered the UK record charts on no. 1 on the 23rd of May. From the analysis, this track was correctly predicted to become a hit song. The control groups show no specifically interesting results, since most of the tracks did not have an entry in the record charts. This is probably due to the fact that the subject groups were too small in size, which is because there were found just 8 predictors from the initial data.

(2) A remark that should be made is that the results from the maximization function showed a top eight tracks by the artist *Funeral For A Friend*. As stated in remark (1), this is probably due to biased tf-idf values from a subject that listened to nothing else but this artist, which biases the results. For this reason, I left out the remaining tracks from this artist, with the tracks from *Haim* and *The Saturdays* actually being the 9th and 10th tracks from the ranking.

7 CONCLUSION

The analyses show some interesting results since each group of predictors from each of the subject groups seem to be able to predict a hit song to some extent. Even though the track *Another Love* by *Tom Odell*, from the group of predictors from the Netherlands, already entered the record charts during the target period, it did not became a hit until two weeks into the target period. On top of that, the other groups of predictors, from both the United Kingdom and the United States, managed to predict a hit song in the near future. One could debate to what extent the song *Pour It Up* by *Rihanna* can be considered a hit, since it only reached no. 9 in the record charts.

The results also show that the control groups have no success whatsoever in predicting hit songs. Most of the tracks that resulted from the analysis on the control groups do not even have an entry in the record charts, and the ones that do cannot be considered to be hits in the near future.

This aim of this research was to question *to what extent hit songs could be predicted through the listening behaviour of Last.fm 'predictors'*. Based on the results, it can be said that predictors have a better sense for listening to potential hit songs in comparison to random listeners. Although the analyses did not pinpoint the perfect tracks we could have hoped for, the results were rather promising, and provide a clear indication that this research could be taken to a next level.

8 DISCUSSION

The results from the analysis seem promising, for the model has proven to be able to predict potential hit songs to some degree. However, there is still a lot of room for improvement, since there are numerous factors that can influence the hit potential of a song. This section provides some elaboration on these factors, and how it could prove quite difficult to take these factors into account.

8.1 INFLUENCING FACTORS ON HIT SONG POTENTIAL

As proven in this paper, it is sometimes possible to predict potential hit songs based on the listening behavior of so-called predictors. Nonetheless, there are various other factors that could play a role in the process of a song becoming a hit song.

One factor, for example, is the notion that a song can become immensely popular through advertisement. Commercials on radio and television often utilize music to assist in expressing the message. Since a lot of people get to see or hear these commercial, it every so often happens that a song becomes so popular such that it reaches the top of the record charts. This flow of events can occur so fast that it would be troublesome to keep track of this process, making it somewhat impossible to predict the song.

Another factor worth mentioning is airplay on radio stations. For most record charts, airplay is taken into account when it comes to composing the chart. Several radio stations, however, have a weekly feature in which, for example, they pick a specific song that gets extra airplay for the next seven days. This can extensively improve the popularity of the song, which in turn could boost its potential to become a hit song. For this research, radio airplay was not taken into consideration when constructing the prediction model, but my guess is that it could be an important factor in studies on hit song prediction.

Furthermore, another factor is the case where old songs re-enter the record charts. Contemporary artists occasionally cover a song that was written and released five, twenty of maybe fifty years ago. In the event where such a cover becomes popular, it occasionally happens that the original song regains popularity as well. In extreme cases, the original song could actually reach the top of the record charts (again). Obviously, these cases are hard to foresee, thus making it difficult to take into account when constructing a prediction model on hit songs.

8.2 FURTHER RESEARCH

As mentioned earlier, there is lots of room for improvement on this approach in hit song prediction. In the introduction, I explained that one of the goals for this research was to lay a foundation for further research. In this section, I will provide some pointers on which future studies with this approach could be improved.

The first, perhaps most obvious, pointer is to enlarge the subject groups on which the research is based. For this study, the size of each subject group was 300 subjects, which is not all that bad since it provided some interesting results. However, considering there

were found just a handful of predictors (13 for the US, 8 for the UK and 29 for the NL), the results could probably be improved when more predictors are gathered.

Another suggestion would be to increase the variety on which predictors were found. For this research, the predictors were gathered through a selection of five hit songs per subject group. But naturally, there are numerous of other hit songs that can be added to the collection when looking for predictors. By doing this, the subject group of predictors will surely get bigger, providing more data to do the analysis on.

As a last pointer, the algorithm with which the analysis is done can be improved in various ways. The TF-IDF is just one of many methods that can be utilized in information retrieval, and even this function has lots of variants. Alternatively, one could use Artificial Neural Networks, or perhaps Support Vector Machines to calculate the hit potential But still, as said before, there are numerous of other options.

9 REFERENCES

[1] J.S. Downie. Stephen: *Music Information Retrieval*. In *Annual Review of Information Science and Technology 37*, 2003, pp. 295-340.

[2] R. Typke, F. Wiering, R. Veltkamp: *A Survey of Music Information Retrieval Systems*. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2005, pp. 153–160.

[3] G. Tzanetakis, P. Cook: *Music genre classification of audio signals*. In *IEEE Transactions on Speech and Audio Processing*, 2002, pp. 293–302.

[4] R. Dhanaraj, B. Logan: *Automatic Prediction of Hit Songs*. In *Proceedings of International Conference on Music Information Retrieval (ISMIR)*, 2004, pp. 488–491.

[5] K. Bischoff, C.S. Firan, M. Georgescu, W. Nejdl, R. Paiu: *Social Knowledge-Driven Music Hit Prediction*. In *Proceedings of the 5th International Conference on Advanced Data Mining and Applications*, 2009, pp. 43-54.

[6] *Could your song be a hit? Find out now and share it with the world.* Found at *http://www.uplaya.com*.

[7] *The world's largest online music catalogue, powered by your scrobbles.* Found at *http://www.last.fm*.

[8] B. McCarthy: *Last.fm: 50 billion scrobbles and the return of the mix tape*. Found at *http://www.thenextweb.com*, 2011.

[9] V. Bush: *As We May Think*. In *Atlantic Monthly*, 1945, 176: pp. 101–108.

[10] *A BSD-licensed wrapper for the new last.fm api and the last.fm submission service*. Found at *http://www.u-mass.de/lastfm/*.

[11] W.W. Cohen and W. Fan: *Web-collaborative filtering: Recommending music by crawling the web*. In *WWW9 / Computer Networks*, 2000, vol. 33, pp. 685–698.

[12] G. Chowdhury: *Introduction to Modern Information Retrieval*, Third Edition, 2010.

[13] E. Greengrass: *Information retrieval: A survey*. In *Tech. Rep*, 2000, TR-R52-008-001, *Center for Architectures for Data-Driven Information Processing (CADIP)*.

[14] A. Singhal: *Modern Information Retrieval: A Brief Overview*. In *IEEE Computer Society Technical Committee on Data Engineering*, 2001, pp. 35–43.

[15] N. Orio: *Music retrieval: A tutorial and review*. In *Foundations and Trends in Information Retrieval 1*, 2006, pp. 1–90.

[16] A.L. Uitdenbogerd, J. Zobel: *An architecture for effective music information retrieval*. In *Journal of the American Society for Information Science and Technology,* 2004, pp. 1053-1057.

[17] C. Wang, J. Li, S. Shi: *A kind of content-based music information retrieval method in a peer-to-peer environment.* In *Third International Conference on Music Information Retrieval. Paris*, 2002, pp. 178–186.

[18] F. Sebastiani: *A Tutorial on Automated Text Categorisation*. In *Proceedings of the 1st Argentinean Symposium on Artificial Intelligence*, 1999, pp. 7-35.

[19] J. Ramos: *Using TF-IDF to Determine Word Relevance in Document Queries*. In *First International Conference on Machine Learning*, 2000.

[20] G. Salton, C. Buckley: *Term-weighing approache sin automatic text retrieval*. In *Information Processing & Management*, 1988, 24(5): 513-523.

[21] A. Berger et al: *Bridging the Lexical Chasm: Statistical Approaches to Answer Finding*. In *Proc. Int. Conf. Research and Development in Information Retrieval*, 2000, 192-199.