Modelling the data

Selecting the features

The features of the audio-data consisted of continuous and categorical values. We decided to focus on continuous and semi-continuos (like tempo as integer) values because it made the task more reasonable considering the scale of the project. The categorical values also often where somewhat misleading as even though they where quite easily understood, they made oversimplified presentations of the songs. For example it is quite often that one pop song has many different keys and modes in different parts of the song and not only one as the data suggested. The same goes of course with continuous features also, but in their case it is more reasonable to express some kind of average value. Tempo can change totally inside a song but this happens very rarely in pop-music. Some extreme outliers are for example “Bohemian Rhapsody” by Queen an “Paranoid Android” by Radiohead. But this is evil data-science aimed at minimising risks for someone who already lives a safe life and not analysing interesting historically revolutionary songs.

Understanding the data

1. There where lot's of challenges in order to understand what kind of data we had in hand and how it could be interpreted. Firstly it was not obvious what all those features actually were. How is liveness or acousticness measured in music? We had to take a somewhat blind leap here and take them as given because we found no open access to Spotify's analysis -function. Given more time we could have tried to make our own analysis based on cloning Spotify's function by machine learning. For now we had to accept that one must first make at least some kind of publish in Spotify in order to get the features out of a song. Fortunately two extremely important features in the data don't need any complex analysis, because song writers, record companies and other target groups certainly now what “tempo” value stands for and how it can be measured and “length” feature only requires a clock and someone to measure the length of the song.

2. Secondly we had to realize that what we have in the data is a bunch of already successful songs. So we do have some kind of posterior distribution of the successful songs but what we don't have is the priori distribution. Where in the domain do the rest of the data lie? Even though you can find what kind of songs are usually successful songs, you can also take guess that there are actually very much songs just like these which are not successful. So you shouldn't interpret the result of our model as some kind of prediction machine which tells how successful your song is gonna be. What you however can safely assume is that if our model gives very low score for your song, it will have very small chance of being a hit song as the successful songs usually just aren't like your song. Still music and the behaviour of pop-culture is a mystery and every now and then things change radically like was mentioned earlier.

3. The data has a lot of outliers which is also related to the behaviour of pop-culture. Certain factors which don't relate to the features of the song in predictable way can make a song very popular. The death of a musician or a political icon like Freddie Mercury and princess Diana can cause radical outliers in currently popular songs. A revolutionary hit movie like “The Exorcist” can cause weird long instrumental song like Mike Oldfield's “Tubular Bells: Part One” the most played radio song in Britain for a short period even if the the rest of the popular songs for that period doesn't remind it at all. These are the interesting parts of pop music but once again, our application does not try to assist you making interesting music but minimize the risks for professional music makers. Sounds awful, doesn't it! We wish that data science would usually be used for something good but the reality is dark and we are the

Devil's advocates. Deal with it.

4. The data is also highly non linear and even though it might seem that there are a lot of data-points when trying to make distinction between over 50 regions of which some all very small, it turns out that 70000 points is not at all enough. The model is quite complicated and this is bad news for small data and yet there seems to be no easy way to make it simpler. Also it is not a good idea to use simple dimensionality reduction as for example PCA produces only one significantly small eigen value. So simply rotating the space only makes the features even more hard to understand. Obviously Spotify has done it best not to use redundant features. These observations still don't mean that the features would be independent from each other. The relations just aren't linear so they cannot be seen in covariance matrix.

5. The data is from one year and no time progression is taken account. So with this data only one shouldn't make false conclusions about the future as the data in this year can have changed. More data is needed in order to produce a tool to be taken seriously.

6. Finally we are studying purely dependencies of data one one shouldn't either make blind conclusions about cause-effect relations. However in this situation an educated guess may be made for those keeping in mind that some of the data could very well have strong relations to some hidden variables and knowing that hidden variable would cause the feature becoming totally irrelevant. That could be the case for example with the tendency of an instrument to affect to the key of the song. With guitar you want to play from E or A, so the actual key does not have much effect if you know that it is the sound of guitar that makes the song successful. You can try to tune your guitar lower and get Eb and Ab and still have as many listeners because most of us can't even notice any change. Fortunatelly we don't consider the key at all (even if we happen to know that it is quite important).

The actual model

We decided to go with somewhat strong assumptions of the data as we had quite a lot hands on experience about music in our group. It is very important to remind here that because the application is strongly in Beta-phase there are even assumptions that we knew that where totally wrong. Better assumptions for example the distribution of the data made the machine learning part so complex that we decided to leave it for later and just present the Beta version which demos the models behaviour.

All the features where first of all considered independent even if we know that they are not. We know that tempo is indeed very much related to danceability but not in linear way. This does not matter so much for the model as this kind of dependency is caused by the music inherently. So if you make a song and let Spotify analyze it, you just can't get those weird values. This hasn't got anything to do with the success of your song. It just makes restrictions to the problem space. In our application you can test these impossible values, because you do it manually.

Much bigger problem is that the features distributions are not at all similar to each other so one model fits for one feature and another model should be used for another feature. It causes quite a lot of extra work which we weren't able to do within this period. For now the model is as follows:

Each feature is first considered an independent variable which affects to the amount of streams. We calculate the distribution of this feature by making it a one dimensional problem. If feature f has value n at point x, we have $f(x)=n$, where x is the value of the feature and n is the amount of streams

in thousands. But we don't want to make curve fitting as it turned out to be very hard to complete. Instead we take the discrete value n and copy the data-point x n times (maybe with a small variance). It seems a reasonable thing to do as we can think that if a song with feature value has been downloaded a million times, it can be understood that it stands for million points in the density of x. What we have after this transformation is a big data consisting of real values. We then make a Gaussian mixture-model regression for it in order to estimate relative n with the features pdf.

Former causes a problem with outliers and at the Beta-phase it hasn't been solved. However this problem is relatively easy to solve so it can be done later. More urgent problem is that some of the features indeed are not distributed as mixture-gaussians. They clearly have some kind of mixtures of one gamma distribution and possibly a few gaussians or alike. And for more even within one feature, the distribution within different regions can vary a lot. So with 8 features and 50 regions you will get 400 different distributions. This kind of model which would be reasonable is indeed possible to produce but naturally it takes a lot of time and tuning. At the beta-phase we just took sklearn.mixture.GaussianMixture as our general model. Ilmari ha d a trick in mind in order to make the gamma distribution be handled as a Laplace-distribution by mirroring the data through y-axis. Laplace distribution could have then be estimated as Gaussian. However there was not enough time for that so now there is a major flaw in the actual model, which just waits to be fixed.

After the distributional transformation of the features it is obvious that we can use linear combination of these models to make a final estimate. If the different distributions are good enough, all the coefficients of the linear regression should be positive, because basically tuning some value to it's peak increases the density of the downloads so it is a positive change in real value. However as we noticed that some of the coefficients though not many in certain features and certain countries were negative. This implies that the distribution of that feature in that region is calculated totally wrong. So once again, the project is in beta-phase.

Basically the whole model can be understood as a feature-independent mixture-model presenting the pdf of streams-amount in data space. We believe that if the independent distributions are better calculated and more data will be gathered, the model will become powerful in estimating the suitability for a song to be a hit for a short period of time. However the model should be updated at least once in a year and probably whenever there is something “fresh” happening in music business.