ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ
ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

ΣΧΟΛΗ
ΔΙΟΙΚΗΣΗΣ
ΕΠΙΧΕΙΡΗΣΕΩΝ
SCHOOL OF
BUSINESS
MSc IN
BUSINESS ANALYTICS

Department of Management Science & Technology

MSc in Business Analytics

Applications of Machine learning on Spotify Data

By:
Georgia Iliaki

Student ID Number: f2821902

Name of Supervisor: Panagiotis Louridas

July 29, 2021

Athens, Greece

Contents

List of Tables

List of Figures

Abstract

This study refers to machine learning applications on data scraped from the Spotify API website. It is divided in two sections based on different data provided by the company. The first section the data handled are the musical features of the songs and an effort is made to classify over 2000 songs based on the emotion they convey to the listener using different classification methods such us Neural Networks, Random Forest, LightGBM, XGboost. Also two regression methodologies are used (Neural Network Regressor and Random Forest Regressor) in order to predict the "valence" value of the songs (how happy or not a song is). On the second part of the analysis the structural layers of the songs are used to create 5 different Neural Network model, one for each layer (Sections, Segments, Tatums, Beats and Bars) to figure out how deep the emotion can be traced on a song. On the first part the most effective method appeared to be the Random Forests. On the second part of the study, the results indicated that the emotions of the songs were better identified on the deepest structural levels of the songs, on the segments data set.

1 Introduction

Emotions are an integral part of the human nature. They affect all living life daily and one can imagine how dull and colorless life would be without it. They influence all people and the way they interact and behave. Many things cause people to feel emotions from simple interactions to audiovisual material. One of those is music. Music is so inter-wined with emotion that there is a whole field of psychology dedicated to it. Every instrumental piece and song can bring an emotional response from everyone such as joy, sadness, disdain or even indifference. The benefits of music in ones life are undeniable. Listening to ones beloved song can improve their mood, exercise routine, lessen anxiety and stress and many more thing.

Fortunately for everyone, in this day and age music is widely available to everyone. One can simply open his mobile or computer and instantly find every kind of instrumental or not piece they desire. Many companies are gathering artists' portfolios to create music libraries available in one place. One of those companies is Spotify Technologies S.A. They have created an application called *Spotify* that has over 70 million songs available from all over the world. Unfortunately for them, there are not the only ones within the entertainment industry trying to offer in demand music to consumers. In order to have a viable future in this competitive, constantly changing market who is overflowing with new technological innovations, businesses must differentiate in order to keep their the interest of their customers and avoid the danger of becoming irrelevant and fade silently in the background. That is why companies heavily invest in data analysis and machine learning techniques that will help them bring better experiences to their customers.

This document examines data made available by Spotify about the musical structure of their songs. The goals are to firstly identify the "emotion" that the song conveys to the listener as it has been labeled by Spotify and in a second note to try and predict the value "Valence". Valence is a variable that measures how happy or sad a song is and it has been calculated by a company called "Echo-Nest" that is owned by Spotify. The above are achieved by using different models of machine learning for classification and regression.

The first chapter is the current containing the introduction and the abstract of the thesis. The second chapter focuses on the whole background of the document. Firstly, a brief history of the company in question and how it has expanded in many fields related to data science and analytics. Then an analysis on the techniques that were used for the prediction of the response variables (emotion and valence). Those techniques draw from traditional statistics (regression) and from new innovations in machine learning,

The third chapter focuses on the literature review of the subject. It is about researches published about prediction of emotion in music. Those papers can be theoretical, or literature reviews themselves, method comparisons. The results of their researches are presented and also their different approaches on the subject.

The next chapter focuses on the data. It begins by explaining the background behind the data selection, the labeling of the emotions of the songs and the collection process. Then, the many variables are explained and how they contribute to the description of a song's structure. Also, the processing of the data is presented and some analysis is done for the relationship between them. Finally the importance of the response variable "valence" is explained and how it contributes to the emotion label of the songs.

Following the data analysis, are the methods, model building and results. The specifics of the techniques used are mentioned, how they were implemented, the challenges faced and the the attempts of different methods. The results of the model training are also presented and the metrics produced such as classification reports and accuracy. Finally a last chapter with the conclusions and some suggestions for future research are presented.

2 Background

2.1 About Spotify

Spotify is an application owned by a company currently called Spotify Technology S.A.[1]. It was founded in 2006 by Daniel Ek and Martin Lorentzon in Stockholm[2]. It is an audio streaming service and according to their website "*the world's most popular audio streaming subscription service with 345 million users, including 155 million paying subscribers (as at December 31, 2020), across 178 markets*"[3]. It offers in it's wide ranged catalogues over 70 million tracks and 2.2 million podcast titles.[4]. In the first trimester of 2020 it was announced that an opening in 85 new countries was going to happen and that the languages that the application is offered would increase by 36 new ones[5]. The general business model is two-sided (artists and consumers) and is based primarily on the paid subscriptions from the users. The company offers information about the artists, the songs and music genres and for developers data about the song's structure as a musical piece. The company's success was not an overnight event. The competition in the market is steep with many streaming services (Amazon, Apple, Google) trying to get the customer's attention. Today music is available everywhere in the internet from YouTube, to websites to even small pieces of songs when you want to buy a song.

The massive portfolio of the application offers an abundance of choices to the user but it tried to evolve by acquiring a series of companies related to the discovery music industry. First it was the music discovery & playlist app start up company "Tunigo" in 2013[6]. The application was already inside the spotify API and working with Spotify. The application created various playlists for the users based on the genre/mood/artists etc with a higher goal to do so with real time data. Much like today's spotify playlists work today. The spotify playlists will be discussed in more detail further in the document. A year later it bought the music data firm "The Echo-Nest". According

---

[1] https://lei.report/LEI/549300B4X0JHWV0DTD60

[2] https://www.sec.gov/Archives/edgar/data/1639920/000163992021000006/0001639920-21-000006-index.htm

[3] https://newsroom.spotify.com/company-info/

[4] https://newsroom.spotify.com/company-info/

[5] https://techcrunch.com/2021/02/22/spotify-to-expand-international-footprint-across-85-new-markets/

[6] https://venturebeat.com/2013/05/03/spotify-buys-tunigo/

to an article from The Guardian "*The Echo Nest uses a database of more than 30m songs to provide music recommendation, audio fingerprinting, and other services*"[7] which seems almost indistinguishable from the "Tunigo" mission. The data used in this thesis are mostly created by Echo-Nest. Their algorithm can detect characteristics of the music as a sound entity like acousticness, loudness etc.

In the course of the last seven years Spotify Technology S.A. has acquired 15 companies related to many new technologies, data, music production and licensing. The strategy of expansion was in many directions such as advanced analytics and consultancy (Seed Scientific[8]), social and messaging startups, cloud-based platforms for subscriptions, audio detection, content recommendation and even blockchain related companies. They also seem to want to draw more artists in the platform by acquiring tools that help the artist collaborate (SoundBetter[9]) license (Loudr [10]) and create music (Soundtrap[11]) and boost their marketing image (CrowdAlbum [12]). In addition, they expanded their podcast original content by buying three podcast production companies.[13] and launching exclusive podcasts with creators.

One of the strong characteristics of the Spotify applications are the abundance of playlists. Those can be created either by the users or by the application itself. The previous-mentioned are constantly changing according to their popularity. The less popular ones are mixed with new songs in order to become more appealing. All of the above are achieved by consumer profiles that the company has created from the listeners data, called "Taste Profiles"[14]. These profiles measure the listener's "taste" in music by unexpected metrics such as how different are the songs you listen sound-wise, how

---

[7] https://www.theguardian.com/technology/2014/mar/06/spotify-echo-nest-streaming-music-deal

[8] https://www.billboard.com/articles/business/6612600/spotify-seed-scientific-acquisition

[9] https://artists.spotify.com/blog/spotify-for-artists-and-soundbetter

[10] https://techcrunch.com/2018/04/12/spotify-acquires-music-licensing-platform-loudr/

[11] https://www.engadget.com/2017-11-17-spotify-acquires-soundtrap-music-production-studio.html

[12] https://techcrunch.com/2016/04/27/spotify-buys-photo-aggregator-crowdalbum-to-build-more-marketing-tools-for-artists/

[13] https://techcrunch.com/2019/03/26/spotify-acquires-true-crime-studio-parcast-to-expand-its-original-podcast-content/

[14] https://techcrunch.com/2014/10/19/the-sonic-mad-scientists/

popular are the songs, if you stick to an artist or not. These make spotify more personal. Daily, the users have updated playlists called“Daily Mix” or “More of what you like”. Also it can help push already created playlists to the user. If the listener enjoys new songs , he will probably get in his feed the “New releases” playlist.

The *Taste profiles* are not the only way the company uses data. They also have “clusters”, who are similar to the ones in classical Statistics. There are groups of songs with similar traits such as the genre of the song or the emotional content of the song (see chapter 4). For example, the hip-hop songs are a cluster. If the user keeps listening to hip-hop artist, his Daily mix will include a variety of new and already familiar songs to him from that genre. Also he may be suggested to listen a playlist called “00s Hip-Hop”. The playlist are definitely a vital piece of the company’s strategy as Spotify’s VP Of Product Charlie Hellman says “*There’s no one taking playlisting more seriously than Spotify*”[15].

One other import aspect of spotify analytics are the data that are “scrapped” from the sound of each song by “Echo-Nest”. They will be discussed in detail in chapter 4. One very import attribute that their data provide is a value called “valence”. According to the dictionary of the *American Psychological Association* “*an entity that attracts the individual has positive valence, whereas one that repels has negative valence*” or “*in certain theories of motivation, the anticipated satisfaction of attaining a particular goal or outcome*”[16]. The team of the company translated this to the emotion a sound conveys to the listener, if it brings happy or negative emotions or something in between. In their website from 2013 they are claiming that valence is an experimental way to analyze sound and for the computer to be able to recognize the emotional valence in songs is a very complicated subject that requires many aspects. They do not have fully disclosed how they do it only that a music expert is helping them classify a sample of the songs by valence[17].

---

[15] https://musically.com/2014/10/22/truffle-pig-theres-one-taking-playlisting-seriously-spotify/

[16] https://dictionary.apa.org/valence

[17] https://blog.echonest.com/post/66097438564/plotting-musics-emotional-valence-1950-2013

2.2 The methods

> “*A machine-learning model transforms its input data into meaningful outputs, a process that is “learned” from exposure to known examples of inputs and outputs. (François Chollet 2017)*”

2.2.1 Neural networks

Neural networks are computing systems that mimic the human neural system. It is a group of artificial neurons who are connected and are processing almost any kind of input. The first neural network was created by McCulloch and Pitts 1943. There are many types of neural networks used in bibliography throught the years. In this document two of them were utilized on the available data. The first one was *Multi-layer Perceptron (MLP) Neural network* and the second was the *Recurrent neural network (RNN)*. Both techniques were used on the classification tasks of the research and the first one was also utilized as a regression technique as well. The use of NN’s has exploded in the last years and new applications are implemented all the time. It is considered a very powerful tool in the machine learning world with endless possibilities, like speech recognition, text generating, image classification, regression problems and human or object identification.

Neural networks are not a technique with no faults. In order to train a good model usually a large data set is required. This is in contrast to more traditional machine learning and statistical methods. If the data are not enough or the structure too complicated there is the danger of over-fitting. This causes the network to learn the training data set too well and as result is almost worthless on never seen data. Another minus of the networks is that it produces result but not why and how.

The first method that was used on the data set was a Multi-layer Perceptron (MLP) Neural network. It is the most common method of Neural Networks (NN) used. It is widely accepted that the foundations for the flourish of this technique (MLP) were laid by Rosenblatt 1958 with his *Perceptrons*. Just like “modern” neural networks, Perceptrons were mimicking the biology of the human brain. An MLP is class of feed-forward artificial neural network (ANN), meaning that all neurons of a layer are connected to the neurons of another layer but with a unique connection (see figure 1). The neurons of a layer are not connected with each other. Each connection between the neurons is rated with a “weight”. This indicates how possible

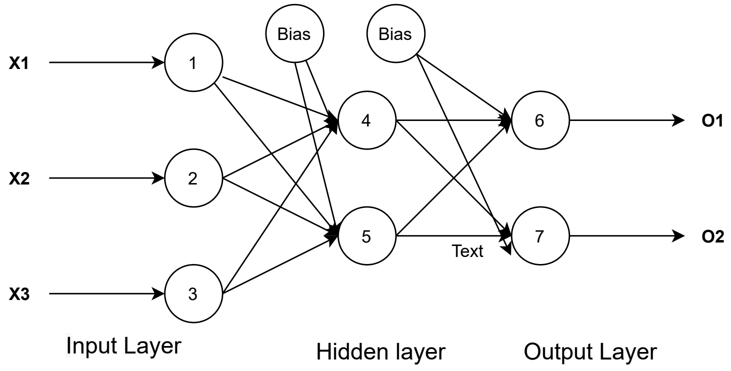


Figure 1: Basic example of MLP architecture
(Graph created using: https://app.diagrams.net/)

it is to “go” from one of the neurons to the other and basically it is the value of the connection. In each layer a bias is added which represent the intercept of the the output of the neurons. These networks are trained using back-propagation. When the data have passed through the network and a prediction has been made about each data point the results are evaluated. This evaluation depends on the loss function. The weights are re-evaluated and the data pass thought the network again. A “pass” of the data through the network is called an “epoch”.

The field of neural networks has been into focus in the last decade, especially with the rise of data analytics and there many new applications and implementation of NNs. Those require various types of data inputs. Despite all this, one thing that is constant in all NN’s and it is the activation function. This mathematical function defines the output of each neuron. The input is used as a value, passes through the function and produces the output of each neuron to pass to the next layer. Many activation functions have been used and the choice depends on the application. An activation function can exist in every layer of the network. It’s most common use is in the first and last layers. The most popular activation function used for an input layer is the ReLu function of *rectified linear unit*[18] (see figure 2). It is widely use because it offers low computational cost, is fast, effective and does not activate all the neurons of each layer. The last-mentioned occurs because of the nature

---

[18] https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7

of the function itself. The ReLu function takes zero for all negative inputs and is linear for all positive values (Equation 1). Certainly it is not the only function in the NN spectrum.

$$f(x) = max(0, x) = \begin{cases} x_i, & if \quad x_i \geq 0 \\ 0, & if \quad x_i < 0 \end{cases} \tag{1}$$

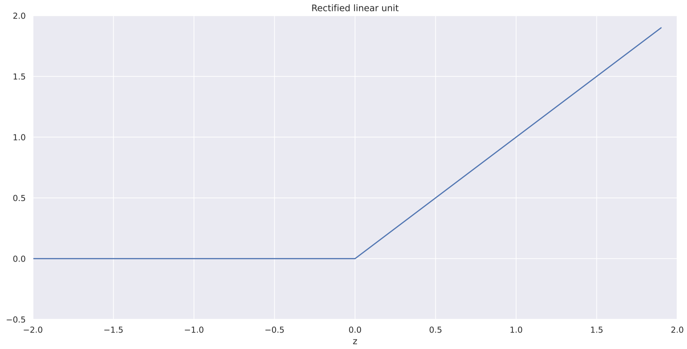


Figure 2: Graphical representation of the ReLu activation function

Some other activation functions used in NN applications are the sigmoid and the softmax. The sigmoid function (equation 2) is sometimes used in binary classification problems because of it's dichotomous nature. It is used in the output layer in order to compute the probabilities of each data point belonging to either category.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2}$$

This function's graphical representation is the typical "S" curve of the sigmoid curve (see figure 3). For each data point the more it's probability is closer to 0 or 1 then the more it belongs to the corresponding category. As the probability moves closer to 0.5 from either side it is more ambiguous to classify the data. The softmax function (Equation 3) just like the sigmoid is used the the output layer as well and it computes probability distribution. They differ in the sense that the first is recommended for multi class classification problems and the latter for binary ones.

$$f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{3}$$

Another type of functions that is necessary for the neural network models is the loss function. It is an essential part of the training of the network and the optimization of the results, no matter the type of architecture it has. This functions calculate how much the networks guess is off from reality. It gives the network a grade on it's overall performance. These functions double as footsteps for the optimization of the result. Depending on the results, the weights of the connections between neurons are changed accordingly by the optimizer. The whole procedure as previously mentioned is called back propagation. One popular loss function is the "categorical crossentropy" function which measure the distance of the probability distribution of the output of the network to the corresponding one of the truth. The goal is the minimization of this distance so the network has the chance to reach the true values. A variation of this function is the "sparse categorical crossentropy" which is practically used on classification problems where the to-be-predicted classes has been turned into integers. Furthermore, on *regression* problems (see 2.2.5) there are two very popular ones that are the "mean absolute error" and the "mean squared error". The first measures absolute differences between the output and the truth and the latter measures the sum of squared distances between the output and the truth. The first one is more robust for data that have outliers.

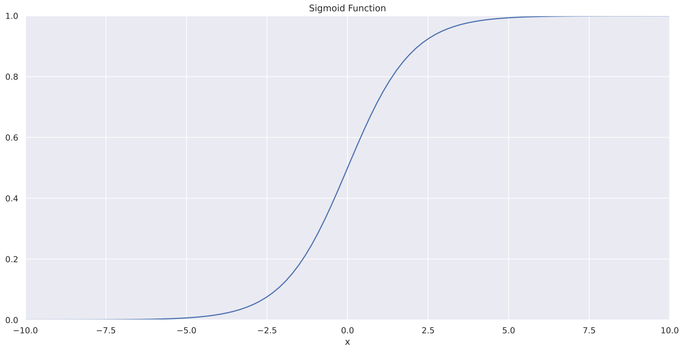


Figure 3: Graphical representation of the sigmoid activation function

As previously mentioned, one thing that stands between each epoch of neural networks is the optimizer. It is the function responsible for the weight adjustments that happen between epochs. Two of the most common functions used are the *Stochastic Gradient Descent* (SGD) and *Adaptive Moment Estimation* (Adam). The first algorithm has a goal to reach the lower point possible by descending from a higher one. The stochastic element in the algorithm has proven to be very useful because instead of choosing all possible descending points, those are chosen in a stochastic manner. This offers computational advantages. On the other hand Adam is a newer function first presented in 2014 on a deep learning conference[19]. It uses squared gradients and contrary to SGD it uses the moving average of the moments of the gradient.

The other neural network architecture that was implemented on the data of this document is the Recurrent Neural Networks (RNN). This type of networks differ from the MLP architecture because they have “memory”. These networks are widely used in problems where the data are a sequence (text) or there is a historicity between each data point or the data are dependant or even descendants of each other. In contrast with the feed forward networks who take an independence hypothesis between the data, RNN’s on each epoch keep in their “memory” information they have from previous ones and using it to influence the current or feature outputs. The have used what the have “learned” to determine what the next input/output will be.

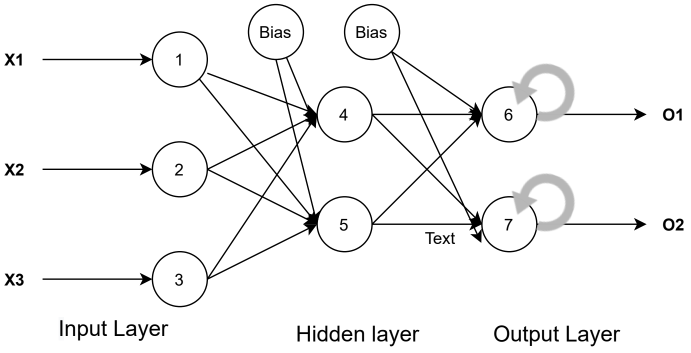


Figure 4: Basic example of a Recurrent network architecture (Graph created using: https://app.diagrams.net/)

Two of the most common types of RNNs are the Bidirectional recurrent

---

[19] https://www.iclr.cc/archive/www/2015.html

neural networks (BRNN) and the Long short-term memory (LSTM). The first one is an expand on the RNN dogma because the “memory” it uses can not only be from past epochs but from data it have not even seen yet. It can “see” all of the data to come in order to achieve more impressive accuracy results. This is very helpful in text generation processes as some words are known to be more frequently before or after certain ones. An example is that the phrase “*Give someone the cold shoulder*”. If the network possess prior knowledge that the word “shoulders” will come next it is easier to predict the word “cold”. On the the hand, LSTM networks keep their memory “clean” and not influenced by long past data have three memory gates. One for the input in their memory, one for output and one to forget what is far behind in the past of the data. The term Long-short in their name is to indicate this procedure.

2.2.2 Random Forests

Random Forest is a supervised machine learning method[20] that, as it’s name suggests, is a forest (ensemble) of decision trees that work together (but they do not actually interact) to achieve their goals. It is a technique that is based on “*The wisdom of the crowds*”[21] and it is simple and versatile as it can be used for classification and regression problems as it’s principles are applicable to both (Breiman 2001). Each tree produces an outcome and the majority wins. The number of decision trees that the forest is consisted from can vary from a few to thousands depending on the needs of the user. A decision tree searches for the most important features and thresholds of the data and splits it’s nodes accordingly. In contrast, random forests pick the most important feature again but from a random bunch of features. This ensures an unbiased model, that has both diversity and randomness to produce the best possible results and avoid overfitting and correlation cases in the same time. Moreover it is a widely used technique that outperforms most methods in terms of accuracy without issues of overfitting (Misra and H. Li 2020). The technique has many advantages and desirable features for a machine learning algorithm. As previously mentioned, they usually display high accuracy scores without overfitting. Misra and H. Li 2020 gathered all the advantages of the technique and presented them. Adding to the accuracy, there was also a robustness to nose and outliers, simplicity and

---

[20] https://en.wikipedia.org/wiki/Supervised_learning

[21] https://towardsdatascience.com/understanding-random-forest-58381e0602d2

ability to be parallelized. They can also be used as a feature selection tool (variable importance) and offer data on errors and correlation between the variables. Finally they claim that the algorithm is better than Adaboost[22] and faster than bagging (merging the same type of predictions, also decreases variance[23]) and boosting techniques (aims to solve overfitting problems by decreasing variance).

Many parameters are available for hyper-tuning the Random Forest. One of the most important ones is the criteria to rank the feature importance. It can be done either with *Entropy* (equation 4) or the *Gini index* (equation 5). The first one is a statistical measure of impurity that measures between 0 and 1. The closer it is to one, the higher the "disorder" on the data is. The latter is an "impurity" measure and ranks the features by their contribution to the homogeneity of the data .

$$E(S) = \sum_{i=1}^{c} -p_i \log_2 p_i \tag{4}$$

The Gini index is calculated each time a feature is used to split the data. It represents the homogeneity and is 0 for completely homogeneous data and 1 for completely heterogeneous data. The computation formula subtracts from 1 the sum the squared class probabilities. High values of the index mean the feature is important and vice versa. Furthermore, other parameters dictate how each tree of the forest can either be built from the whole data set or from a bootstrap sample (random sampling with replacement). More parameters can be manipulated such as the number of trees, the maximum and minimum number of trees, the learning rate and even weights for each estimator.

$$Gini = 1 - \sum_{i=1}^{n} pi^2 \tag{5}$$

2.2.3 XGboost

Many classification algorithms are now widely available for developers and researchers alike. One that has been a constant recently is *eXtreme Gradient Boosting* or XGboost for short. It was created by Tianqi Chen

---

[22] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[23] https://www.upgrad.com/blog/bagging-vs-boosting/

(2014)[24] originally as an open source scalable and flexible Gradient Boosting algorithm and it has now reached availability for all systems. Like Random Forests, it is considered an ensemble supervised classifier, and it has became very popular due to it's execution speed and computational savings. It is a boosting classification tree method that penalizes complexity and Just like other tree models, it offers an automatic feature selection.

The technique in general mimics in some ways the gradient boosting algorithm but with some improvements. The boosting methods in general, build models in sequence and computing their errors. Each time a new model is to be added to the ensemble all previous ones are evaluated and the impact of the ones with higher error is minimized and the other way around. Only the models with lower error values have an impact in the ensemble. In the gradient boosting approach the models to be added predict the residuals of errors of the existing models . Then all models together provide the output. In a similar way with neural networks the loss is minimized by the gradient descent algorithm. The XGboost algorithm takes all the aforementioned and combines it with parallel processing, tree pruning, optimal node spits, tries to avoid biases and overfitting and it even handles missing values[25].

All the above features make it a powerful algorithm. From a system's view, parallelization and optimal tree pruning offer computational gains. Moreover the algorithm is hardware friendly as it distributes in a smart way the buffering processes. The performance of the XGboost is also boosted with many features from a statistical/mathematical point of view. As previously mentioned the trees are penalised with other classic statistical methods like LASSO[26]. This makes the model more robust and resilient to overfitting. Using the loss function while it is fitted and trained on the data it adopts the more appropriate values for missing data. Also, there is an extra randomisation parameter to avoid correlation problems. Finally, an internal cross-validation is attached to the procedure.

2.2.4 LightGBM

In 2016, a team within Microsoft published a paper with their answer to XGBoost and all gradient algorithms. Their proposed algorithm is called *Light Gradient Boosting Machine* or LightGBM. The team claimed that the

---

[24] https://tqchen.com/

[25] https://medium.com/hackernoon/gradient-boosting-and-xgboost-90862daa6c77

[26] https://en.wikipedia.org/wiki/Lasso_(statistics)

method is “highly efficient and scalable” even “*when the feature dimension is high and data size is large*”. Just like XGBoost it is a machine learning Gradient Boosting algorithm, that operates using decision trees. It is a fast algorithm with low memory consumption that can handle larger data sets. According to its creators, it can achieve results 20 times faster than the other gradient algorithms on a variety of public data sets.

The algorithm is very fast on large data sets. This is achieved through an important fact that the team of Ke et al. 2017 noticed. They realised there is a correlation between the gradient of each data point to training errors. A data instance with a small error in training had also small gradients. This way they assumed that this instance is trained enough but these couldn’t be fully removed from training because it would hurt the distribution of the entirety of the data. Their proposed solution was a new technique called *Gradient-based One-Side Sampling* (GOSS) that would keep the data points with a big gradient and perform a random sampling on the previous-mentioned small gradients. To further secure the data distribution the small gradients are multiplied by a constant multiplier (equation 6), where a represent a random sampling ratio of the large gradient data instances and b is the corresponding one for the small gradients. The ration of the two must be chosen carefully to avoid overfitting.

$$Constant = \frac{1-a}{b} \tag{6}$$

Another feature of the algorithm is its ability to handle not only large data sets from a data point perspective but from a data features as well. This is achieved through a new technique they proposed on the same paper the algorithm was introduced and GOSS as well. This method to reduce features is called *Exclusive Feature Bundling* (EFB). This technique depends on two assumptions. The first one is that usually high dimensional data are also very sparse and the second one is that most of the time some features cannot be zero at the same time. Those features as the name suggests are “bundled” together, thus, there is instant dimensionality reduction.

The algorithm in general focuses on achieving higher accuracy rates. This is achieved using a different method than XGBoost when splitting the nodes on the decision trees. While XGBoost and other gradient algorithms use a level wise approach while building the trees, LightGBM uses a depth first approach (see 2.2.1). This way, the loss can be reduced significantly and the accuracy can augment in comparison to level-wise growth algorithms. The accuracy is proportional to the number of leaves. Although the above make the algorithm effective in many situations, at the same time, they make him susceptible to overfitting issues, especially on smaller data sets. The framework of LightGBM offers parameters to tune to avoid such issues.

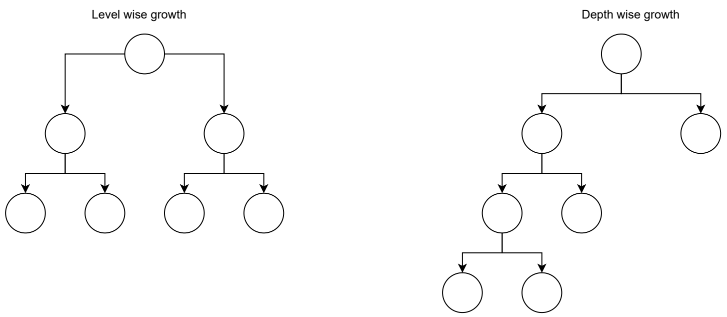


Figure 5: Comparison of tree node growth, level-wise (left) and depth-wise (right) (Graph created using: https://app.diagrams.net/)

2.2.5 Regression

Regression is a statistical method that is used in many scientific and not only fields, in order to understand the relationship between a codependent and one or more dependant variables. By performing regression analysis on a dependant variable one can identify what factors really contribute to value of the aforementioned. It is method with a plethora of uses in many cases and it has been adapted to be included in many frameworks. Regression analysis can be done for both liner and non-linear relationships between variables. The fathers of regression are considered Adrien-Marie Legendre 1805 who was a French mathematician and Johann Carl Friedrich Gauss (1809) a German mathematician and physicist. The two scientists used the method of least squares initially for astrology purposes. A string of publications from others developed regression in its today forms.

Regression is commonly used in machine learning problems when a continuous variable needs to be predicted or on binary classification problems (example: logistic regression). Many of the more modern techniques have been adapted to be able to handle regression problems such as Random Forests, Neural networks, Support vector Machines. This has made regression today very popular amongst managers and companies. It can be used as a technique to understand every phenomenon they like, predict what will impact sales etc and it has evolved to their “go-to”[27] method. In this document it was used to predict the “valence” (see chapter 3) of each song.

The first regression method used in this document was Random Forest Regression. This technique works mostly like it’s classifier counterpart (see 2.2.2). Again several trees are involved and they work as an ensemble to produce an outcome. While each decision tree in Random Forest Classifiers (RFC) produces an outcome for the classification and then the one that most trees agree is the final outcome, in Random Forest Regressors, the final outcome (a number) is actually the mean of all trees. Each tree/model makes a prediction on the dependant variable and the mean is presented as a result. This process is referred to as “aggregation[28]”. Again, the same features as RFC appear. The method offers an inner feature selection and they are selected from a random bunch of features to avoid overfitting. Finally the method is widely available for different frameworks and programming languages with many parameters for hyper-tuning.

The second regression method utilized was one implemented using a Neural Network (see 2.2.1). The NN are mostly used for classification problems but actually it is possible for them to be used in regression analysis as well. The general structure of a Neural network used for regression shares similarity with the MLP networks. The network again begins with an activation function and it contains as many hidden layers as the user finds appropriate. On the last layer, the “exit” of the network there is one neuron and no function. Using a sigmoid function would made the network a binary classifier, a softmax a multi-class classifier etc. The last layer is simply linear to predict the continuous variable. These type of networks have usually as an optimizer a mean -squared error function but Adam is used as well. The loss function of the network is the mean absolute error (MAE), the absolute difference between the predictions of the network and the real values.

---

[27] https://hbr.org/2015/11/a-refresher-on-regression-analysis

[28] https://www.geeksforgeeks.org/random-forest-regression-in-python/

3 Literature Review

Emotion recognition in music is a field in the scientific world that has gotten more attention in recent year primarily by companies that build music libraries are seeing the necessity to have an emotional classifier in order to group the songs. They are after all, the means to keep their customers interesting in their service. People tend to hear music that suits their emotional state so available labels offered by companies is part of a business strategy meant to engage the customer in the service for as long as possible. This task is very challenging and in recent years many methodologies have been suggested by various scientists. The methods range from statistical algorithms to machine learning techniques. Not only the methods of prediction and classification vary in the field but the initial data set used. From audio data, features of the songs to lyrical data all try to capture how to model the emotions.

Valence

Valence as previously mentioned, has been reinterpreted from the "Echo Nest" algorithm as the measure of happiness and sadness in a song. They are not the first to associate music and valence. In 1990, Robert E. Thayer in his book *"The Biopsychology of Mood and Arousal"* (Thayer 1990) suggested a model to describe the emotion in music. According to him two variables are the driving forces in the emotion residing in music, energetic arousal and tense arousal. Valence was identified by him as various combinations of energetic and tense arousal (Eerola and Vuoskoski 2011). This approach can be visualized in his emotion plane (see figure 3) and while it is a continuous multi-dimension approach it also justifies the perception of emotion as discrete and distinct categories (Nguyen et al. 2017).

Thayer's approach was not the only one trying to model emotions according to valence. Two and three dimensional models have been proposed as well and even predated his. One of the earliest examples of emotional models has been a three dimensional one that pairs emotions with their complete opposite (Wundt 1896). There are doubts that three dimensional models are accurate although they can be helpful in the exploration of two dimensional solutions (Eerola and Vuoskoski 2011). The two dimensional approach has been proposed by Russell 1980 and he argued that all emotions can be described as a specific combination of arousal and valence. Actually many refer to Thayer's emotional plane as a derivation (Tan, Villarino, and Maderazo 2019) or an adaptation (Hızlısoy, Yildirim, and Tüfekci 2020) of Russell's model. The valence-arousal approach overflows the field as many have used it as a technique for their initial emotional detection models. Despite the wide acceptance of Thayer's emotional plane by the scientific world, few have viewed the emotion plane from a continuous perspective mostly physiologists (Y.-H. Yang and H. H. Chen 2011). Most papers use the plane in order to discretize the emotions.

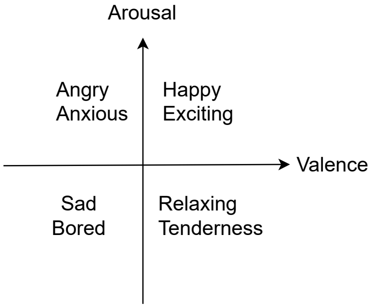


Figure 6: Thayer's musical Arousal-Valence Emotion Plane (Graph created using: https://app.diagrams.net/)

Neural Networks

Neural Networks are a very popular technique in the field with different types of networks and architectures or even combinations of them. Liu et al. 2018 created a Convolutional Neural Network (CNN) model that got as input raw digital sound transformed to two dimensional input using a Fourier Transformation. In this way their input was a spectogram for each song. They tested their Arousal-Valence model in 1000 songs. To enrich their data set, they divided their 45 second clips to 5 second audio clips. The CNN was using a 5x5 filter, the "Adam" optimizer and a 0.0001 learning rate.To further validate their model, a comparison was made with a traditional Support Vector Machines algorithm. The CNN spectogram model highest accuracy was 78% which was significantly higher than the 40.3% of the SVM model.

One of the most recent attributions to the music emotional recognition field is the one from Hızlısoy, Yildirim, and Tüfekci 2020 with their paper published in October 30 2020. They used a combination of neural networks architectures, a convolutional long-short term memory neural network

(CRSTM) to predict the emotion of Turkish traditional songs. They employed, as annotators, Turkish music university students to rate the valence and arousal of the data set. This data set is a the same time a database of emotionally annotated Turkish traditional songs made available by the researchers. Their deep network had four convolutional layers, three max pooling layers, trained for a hundred epochs. The songs were to be classified in three emotional discrete categories. They reached their best accuracy percentage when conducting feature selection to reduce correlation between variables. Their model was compared with other more conventional techniques such as Random Forest classifier, SVM and K nearest neighbours and it outperformed all in accuracy after the feature selection.

Schmidt and Kim 2012, used Deep Belief Networks and Restricted Boltzman Machines (RMB) for emotional recognition in audio data. They claimed that these techniques can help researchers in better understanding the relationship between the emotion and the music entities. The labels of their data were modeled in the image of the two dimensional Arousal-Valence (A-V) model. They highlight their view of emotion as a space, differentiating from previews researchers that view it as linear combination of Arousal and Valence. This space can can model the relationships of audio data and variables over time as well. To achieve this, a Conditional random field (CRF), a graphical model, that is *"trained to predict conditional probability"* was used. The CRFs were used, alongside a logistic regression, to gain the conditional probabilities of a song transitioning from an emotional class to another. The feature selection and learning was achieved with RBMs greedily trained. They believed that their algorithm could become more accurate in prediction using high-dimensional data and more powerful regression algorithms. In their future work section it is also mentioned that the CRFs are the key to better understand the emotional space distribution.

Support Vector Machines

Other machine learning techniques such as Support Vector Machines (SVM) have been proposed by researchers. The team of T. Li and Ogihara 2003, divided 499 songs in 13 emotion categories. With the help of a subject those songs got one or more of the 13 labels and from his rating emerged six subgroups. Then, they created multiple binary classifiers using SVM models. They tested their classifiers first on the thirteen original categories then on the six subgroups. The results of the first experiment were not very successful and it was attributed to many songs potentially belonging to more than one category. That statement was re-enforced when their overall accuracy improved when the classifiers were used on the six categories. In conclusion, an expansion of the data set was suggested and more carefully data labeling for better confidence in the labels.

h. Yang, Lin, Cheng, et al. 2008, proposed the use of the method for emotional detection using a multi-modal approach with lyrical and audio data. They used natural language processing techniques to process the lyrical data and they opted for Thayer's emotional plane as well. They divided their songs in four categories, happy, angry, sad and relaxing. Their approach was implemented using 1240 Chinese songs as the data set. The higher accuracy of their models was 74% after adding textural features to their data. They concluded that lyrics are important in emotional recognition. For this part he used Thayer's emotional plane and divided the songs in four distinct categories.

A system that combined standard and melodic features of songs for Music emotional Recognition was brought to the academic field by Renato Panda, Rocha, and R. P. Paiva 2015. They utilized a data set created by their team and used in previous publications as well (R. Panda and R. Paiva 2012) composed of 903 audio clips and containing 5 emotional clusters. Each cluster was not limited to one emotional state but many. For reference cluster 1 was described by the researchers as "passionate, rousing, confident, boisterous, rowdy". The "standard" features of the data included pitch, harmony, loudness and timbre among others. The "melodic" features were pitch, duration and vibrato features but their statistical measures such as the mean, standard deviation, skewness and kurtosis. Many supervised machine learning methods were tried but the one that provide the most prominent results were the Support vector Machines with 67.8% accuracy. Overall the combination of the features categories proved to produce better results than the part were the feature categories were used individually.

A perspective based on the emotional recognition based on the genre of the song was proposed by Koutras 2017. Using a thousand 30 second audio samples from songs as a data set and their characteristics such as timbre etc, he suggested a system that firstly recognises the genre of the song before emotionally categorizing it. He chose four different music genres and used four different support vector machines each one adapted according to the genre. Both the genre recognition and emotional recognition algorithms provided fruitful results. The latter on had achieved a total accuracy of 77.57% and in the individual category 80.24% was the highest accuracy achieved and it was for the “happy” songs of the spectrum. After this, he repeated the procedure but with a feature selection method and the total accuracy rose to 85.15% and 88.75% in the “calm” category. He suggested that genre classification could be expanded with more categories in future researches so the emotional recognition ca be expanded as well.

Tan, Villarino, and Maderazo 2019, used Russell’s two dimensional arousal-valence model and support vector machines in order to predict in which of the four planes of Thayer’s emotional plane the each song belongs to. They also used a Naive-Bayes classifier in order to compare the performance of the two methodologies. They used both lyrical and audio data which had lyrical and audio annotations respectively. They trained two support vector machine and Naive-Bayes models one to predict the plane using valence and one using arousal. In their findings was that lyrical data were very accurate for valence predictions (Naive-Bayes model) and audio data for arousal prediction. Also the SVM model had better precision on the *arousal* data than the *valence*.

Other approaches

Most of the research done on emotional music recognition is based on Thayer’s emotional plane and the discrete categories of emotions. A continuous approach was proposed by h. Yang, Lin, Su, et al. 2008. They argue that this way the vague and many times not understandable results of traditional classification techniques could be by-passed. According to this perspective all spots in the emotional plane represent an emotional state but is also ignores the possibility that valence and arousal could be possibly correlated. To materialize their ideas, they used a regression model to predict the value of valence and arousal from audio samples. They reduced the features of the data set using principal component analysis. The found, on accuracy level, that support vector regression was outperforming linear regression and Adaboost. The performance of the chosen model was measured through the R2 statistic which was 60% for arousal and 30% for valence.

The need for subjectivity in music recognition systems was drawn to attention by h. Yang, Su, et al. 2007. In their paper about individuality and its importance in music recognition systems, they argue that these systems should become more personalised in order to be more successful for the companies using them. According to them, they were the first researchers to measure the impact of variables such as sex, cultural background, knowledge in music in a mensurable way. Using sixty English songs as a data set, annotated by university students they divided their research in two parts, the group approach and the user approach. In the group approach the trained one Support Vector Regression model for each of the characteristics of the students (they called them groups) such as gender, whether they play an instrument etc, creating dichotomous variables out of them, to classify emotions. The second approach was based on the annotation of each student for the particular song. The results showed a slight improving in accuracy and the need for further research of the topic was addressed.

An alternative technique in the field was proposed by Yu et al. 2015. They used the dimensional Valence-Arousal model in order to create a weighted graph that would predict the emotion of certain words in English and Chinese. The paper did not contain any data related to music but their model could possibly be implemented in lyrical data, that is why it is included in the literature review. They used two lexicons with Valence-Arousal ratings for each word. They compared their proposed weighted graph with Pagerank [29], a Kernel model and linear regression. Their results were in favor of their graph model and Pagerank who both outperformed the other methods [30]. Another finding was that Arousal was more difficult to predict than valence. This comes in agreement with Tan, Villarino, and Maderazo 2019's paper that also concluded that valence is easier to predict with lyrical data.

A combination of the categorical and the two-dimensional approach was proposed by Nguyen et al. 2017. Their goal was detecting more emotions from Thayer's model via classification techniques implemented in the WEKA software[31]. They used three hundred audio files that were containing samples from different music songs. The songs were labeled by annotators as to which emotion they conveyed. They created two models, one for arousal and one for valence and after a feature selection for each model, three different supervised classification techniques were tried. The best results were from Random Forest Classifiers reaching a 70% on arousal and a 57% on Valence. The other two methods were a classification tree and a logistic regression. Their final classification was for six classes of emotions. It is recognised

---

[29] https://neo4j.com/docs/graphdatascience/current/algorithms/pagerank/

[30] https://www.overleaf.com/project/5fba8675294c2e4b0d87d068

[31] https://www.cs.waikato.ac.nz/ml/weka/

that some accuracy scores were low and the subjectivity in music emotional recognition is emphasized as one of the reasons.

4 The Data

4.1 Presenting the data

The present research is a machine learning application for analyzing data from the API Spotify and using them in order to achieve two goals. Firstly predicting the emotion that a song conveys to the hearer and secondly predicting the "valence" variable that is spotify's way of identifying a song's emotion. In order to achieve the aforementioned goals a data set was created by choosing more than 2000 songs from "official" Spotify playlists branded with emotions. There are more than four billion playlists available in the application.[32] Each playlist is advertised by Spotify by a title that presupposes the listener. An example from each emotion identified by the playlists are presented in table 1. According to spotify you can "choose what you want to listen to, or let Spotify surprise you"[33].

The data selection does not have any other particular themes. There are from various artists, languages, year's or music genres. The only selection criteria was that the song was in a Spotify official playlist categorized in one of the four emotions that were selected as the classes for this research. The four emotions are also one emotion from each of the four categories of Thayer's arousal-valence emotion plane (see figure 3). The official playlist on Spotify is recognized by a small logo of the application on the cover image of the playlist on the top left (see figure 7) . If the logo is not there then the playlist is created and made public by a user. Those playlist were deemed unreliable. Spotify categorizes the songs based on the valence variable[34] so the categorization is probably based on some kind of algorithm results based on the songs technical structure. The user's playlist selection criteria are not objective.

The four emotions that were decided to be utilized in the data are Happy, Sad, Angry and Calm. This decision was made because other characterizations from playlists could not be considered clear emotions and were combined with other categories. For example "joy" was combined with "happy", "melancholy" was combined with "sad" and "relaxed" with "calm". If those categories were to be separated the separation between categories could be

---

[32] https://newsroom.spotify.com/company-info/

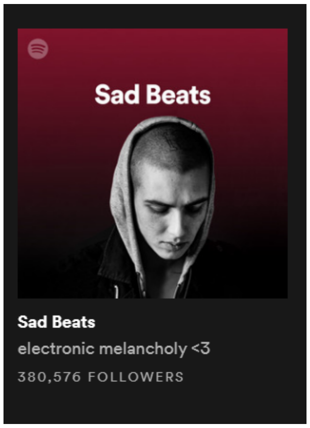
[33] https://www.spotify.com/us/about-us/contact/

[34] https://developer.spotify.com/documentation/web-api/reference/endpoint-get-audio-features

Playlist Name	Supposed Emotion
“Chill Vibes”	Calm
“Sad indies”	Sad
“Angry at the world”	Angry
“Happy Diathesi”	Happy

Table 1: Spotify’s playlist names and supposed emotions

blurry. The final song catalogue was consisted of 2011 songs in total. It is widely accepted that balanced data sets provide better results in machine learning so it was crucial that all the categories were equally represented in the data. The length of each playlist was not a consistent variable and it could range from 50 song to over 200 song playlists. That is why the amount of playlists for each category was not a concern. The goal was an equal amount of data for each emotion.



Sad Beats

electronic melancholy <3

380,576 FOLLOWERS

Figure 7: Example of official Spotify playlist (Sourse: Spotify App)

The entirety of the data were collected through the official Web Spotify API for Developers. The accessibility of the API was easy because a Spotify Premium Account was available, so the only necessity was a simply log in. To actually obtain the data, Python programming language was used because there is a simple library (spotipy), that contains all the necessary commands for connecting to the API and scrapping the necessary data for each song.

The *Spotify Web Api*, returns JSON metadata about music artists, albums, and tracks, directly from the Spotify Data Catalogue. For each song

Spotify provides two categories of information. The first is called "Audio Features" and the second is called "Audio Analysis". For all the 2011 songs both types of data were taken. In order to get the emotion for each song, official Spotify playlists were selected that were already categorized by the app. For example one playlist called "Mood Booster" is filled with happy songs to boost your mood. In that case the songs of that playlist were categorized as *Happy* at the moment it was downloaded. In order to be able to download the information fast the URI of each playlist (a resource identifier that you can enter, for example, in the Spotify Desktop client's search box to locate an artist, album, or track) was used.

4.1.1 Audio Features

As previously mentioned *Spotify Web Api* categorizes the information about each song into two categories, the *Features* and the *Analysis.* The *features* section is consisted of audio information about each song. The first variable is the unique id of each song (was used later to get the Audio Analysis) or as Spotify calls it the "Spotify ID for the track". Then, some generic info about each song such as title, artist name and the names of other artists featured in the song. Moreover, for each song 16 variables (Table 2) are provided about each song. Those variables offer a plethora of information about the song overall in comparison to Audio Analysis (see Audio Analysis section). At first all the variables were download in their original form in order to decide which ones would help the desired goals of the analysis. All the variables except the emotion, the title and the artists that are involved in the song are numerical variables. There were no categorical data in the Audio Features section. For each variable a description is provided.

The variables can be divided in three categories. The first category are the float variables. Their value is a float number that is within some boundaries (example from 0 to 1, valence) and the value depends on structural characteristics of the song. The second category are the variables that are integers. They provide clear information about the song like its duration. The third and final category are the "number" variables. Those variables simply inform the user of the number of the "audio analysis" characteristics of the song. One of those are the number of the Segments (see Audio Analysis). These variables depend heavily on the duration of the songs. In the "floats" category are nine out of the sixteen variables.

Variable Name
Danceability
Energy
Key
Loudness
Mode
Speechiness
Acousticness
Instrumentalness
Liveness
Valence
Duration ms
Time signature
Num bars
Num Sections
Num Segments

Table 2: Spotify's audio features for each song

Danceability: This variable has its values ranging from 0.0 to 1.0. The value given for each song is according to how one can dance to this song. How "danceable" is it. The more "danceable" a song is the higher is the value. Songs with 1.0 are the most suitable songs for dancing. This value is based on a number of elements of the music of the song such as tempo, rhythm stability, beat strength, and overall regularity.

Energy: Same as *Danceability*, this variable ranges from 0.0 to 1.0 and it represents the "energy" of the song meaning its intensity and activity. Of course this is an objective perception but according to Spotify's Api manual "energetic tracks feel fast, loud, and noisy" and "*perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy*". An example provided by the website is that metal songs have more energy than a classical slow piece.

Acousticness: Again a 0.0 to 1.0 scale but this variable is actually a confidence measure. The higher the value, the highest the confidence that the song is acoustic.

Liveness: This variable is trying to pinpoint if the song is from a live performance (audience present or not) or a studio recording. Directly for Spotify's Api manual : "Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live".

Speechiness: This variable "detects the presence of spoken words in a track". Spotify does not contain exclusively songs. Its catalogue contains also podcasts, talk shows and audio books. Those tracks of course have thesis "speechness" values closer to 1.0, as they are spoken words - dialogues. According to the official website when a track has a 0.66 value or higher then it is probably only words in its entirety. When a a track has a value lower than the aforementioned by higher than 0.33 then it contains music and speech (sections or layered). An example from this category is rap songs. Values below 0.33 most likely represent music and other non-speech-like tracks.

Instrumentalness: The type of the variable as provided is a float . Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.

Valence: Valence is the most important of the variables as according to spotify, it is used to label a song's emotion. The range of Valence a song can have is from 0.0, to 1.0. The higher the valence the more happy and joyous is the song. Of course the reverse is true as well. The lowest the valence the more sad, melancholic the song is.

Tempo: Tempo is the speed at which a piece of music is played[35]. This "speed" is counted by *beats per minute* (BPM) . This variable offers the beats per minute of each song.
Loudness: This is the last of the "float" variables and its value is measured in decibels (DB). The range of the loudness of a song is usually from -60 to 0

---

[35] https://www.masterclass.com/articles/music-101-what-is-tempo-how-is-tempo-used-in-musicwhat-is-tempo

decibels. The Spotify variable provides the song's average loudness (decibels) and "*is the quality of a sound that is the primary psychological correlate of physical strength (amplitude)*". It has also been described as "the subjective judgment of the intensity of a sound" (Jehan 2005).

The next category of variables are the "integer" and they occupy 4 out of the 16 available variables.

Key : This variable is the key the track is in. In music theory key is *a group of pitches, or notes, that form the harmonic foundation of a piece of music.*[36]. The integers provided for each song correspond to pitches using standard Pitch Class notation. (Eg. 0=C, 1= ♯ \ D♭ , 2- D and so on).

Mode: Mode *indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived.* Major is represented by 1 and minor is 0. Even though this is an integer variable it is also categorical one.

Duration ms : The duration of the track in milliseconds.

Time signature: The duration of the track in milliseconds. An estimated overall time signature of a track. The time signature (meter) is a *"notational convention to specify how many beats are in each bar (or measure".*

Num bars : Number of bars the songs contains.

Num sections : Number of Sections the songs contains.

Num segments : Number of Segments the songs contains.

The last three variables are the third category of variables and is explained thoroughly in the Audio Analysis section.

---

[36] https://hellomusictheory.com/learn/keys/

4.1.2 Audio Analysis

The Audio Analysis section of Spotify gives data about the songs' structure. A song is a musical entity and that is what data are given to the user. The "feature" variable give an overall idea about each song, but the audio analysis give low-level analysis. This means that the data concerning the song could be about every second of the song, or every 15 seconds etc. Many of the aforementioned variables are repeated but the are multiple values for each song. Some confidence variables are included as well because the data collection has happened through audio samples from the spotify library, in this case the selected playlists. This confidence variables indicate how reliable the variables are. A warning is present that variables with low confidence values could not be corresponding to the truth.

In the audio features section, each song is a line in the data set and for each feature is given a value, this is not the case in the Audio analysis section of spotify. The songs as musical entities and can be divided based on rhythm in five ways. These are called Sections, Bars, Beats, Segments and Tatums. These divisions are contained into each other in a hierarchy. As one move up the hierarchy the division of the sounds is to bigger sections and at the top is the whole song (see figure 8). The Audio analysis sections provides for each song five different whole data sets containing information about all the beats, bars etc of the song. There is a correlation between the duration of the song and the number of sections the song is divided. Each section, bar, beat, segment, tatum has a some properties including rhythmic information about the song.

Sections: According to Bye 1993, *"in music, a section is a complete, but not independent, musical ide"*. As the "largest rhythmic division of the song", sections are an accumulation of bars, beats etc and because of their size they have large variations in rhythm. A whole chorus of a song could be a sections or a guitar solo and that causes variety in each section. Each section of a song is defined by twelve variables (see Table 3). The entirety of the song is represented as a data frame, where each row is a section and each column is the value of a variable for that particular section. Out of those twelve variables *Loudness*, *Mode*, *Tempo*, Time signature and *Key* have already been explained (see audio features). Their respective "confidence" variables as previously mentioned are how reliable their values are. The *start* variable indicates the second of the song that the section begins, the *duration* how

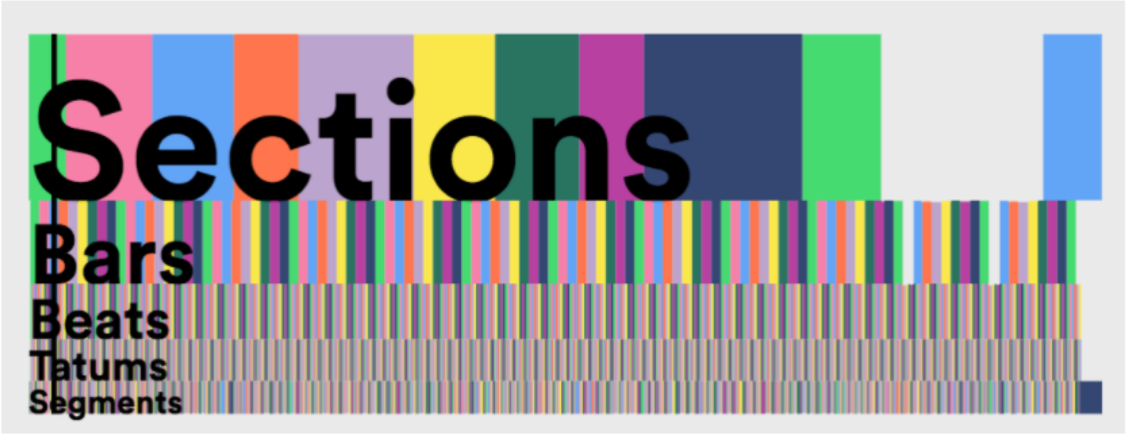


Figure 8: A song divided into Sections, Bars, Beats, Tatums and Segments (Sourse: Spotify Api Website)

long it lasts (in seconds) and the *confidence* how accurate do they think the two previous mentioned values are. For each song the number of sections is different so as a result there are different length data frames for each song. This will be handled later in the processing phase of the data.

Start	Duration
Confidence	Loudness
Tempo	Tempo Confidence
Key	Key Confidence
Mode	Mode confidence
Time_signature	Time_signature_confidence

Table 3: Variables for each Section

Bars: A bar (or measure) is *a segment of time corresponding to a specific number of beats.* The point of bars in music is to provide reference points and it makes it easier to identify locations inside the music[37]. There are three variables provided for the bars of each song, *start*, *duration* and *confidence.* Just like the sections, the *start* variable indicates when the beats starts in the song (in seconds), the *duration* is its duration in seconds and the *confidence*

---

[37] http://www.howmusicworks.org/504/Meter-and-Rhythm/BarMeasure-Divisions

provides the reliability of the data.

Beats: In music, the beat is *the basic unit of time, the pulse (regularly repeating event), of the mensural notation level* (Berry 1976/1986). It is also described as the rhythm someone's toes are tapped to the ground while listening to a song or the tick of a metronome. Again there are three variables offered for each beat of each song start, duration and confidence. Their explanation is the same as their counterparts in *bars*.

Tatums: Tatums are "*the smallest time interval between successive notes in a rhythmic phrase*"(Bilmes 1993). Tatums are smaller than beats (see Figure 8) and is the smallest pulse that is distinct to the human ear. Just like Beats and Bars there are the same three variables offered: start, duration and confidence

Segments: According to Spotify segments are *a set of sound entities (typically under a second) each relatively uniform in timbre and harmony.* It is a piece of time that has a consistent sound unlike sections that are characterized by variability. There are nine variables for the segments of the songs: *start*, *duration*, *confidence*, *loudness_start*, *loudness_max_time*, *loudness_max*, *loudness_end*, *pitches* and *timbre*. The first three variables are exactly the same as beats, bars and tatums. The loudness_start is when the loudness (see audio features) starts how "loud it is" (measured in decibels) . The loudness_max_time, describes for how long the loudness persists for (in seconds). The loudness_max is measured in decibels and is the maximum value of loudness of the segment. The and loudness_end is how "loud" is the "loudness" when it ends.

The pitches and timbre variables are actually two vectors containing the information instead of a value for each segment. For all the segments there is a pitch vector with twelve numbers. These numbers are ranging from 0 to 1 and they correspond to the dominance of each of twelve the pitch classes in the segment (see Table 4). These values are predicted. t Because this could resolve in values accumulating near zero (pure tones) or one (noisy sounds), the values of the vector are normalized. The last variable is *timbre.* Timbre is the sound quality, or tone quality, of a note played on a particular musical instrument[38]. The same notes can produce very different sounds depending

---

[38] https://www.masterclass.com/articles/guide-to-timbre-in-music

Pitch Name	Tonal counterparts
0	C (B ♯ , etc.)
1	C ♯ , D♭
2	D (C𝄪, etc.)
3	D ♯ , E♭
4	E (F, etc.)
5	F (E♯ , etc.)
6	F♯ , G♭
7	G (F𝄪, etc.)
8	G♯ , A♭
9	A (G𝄪, etc.)
10	A♯ , B♭
11	B (C♭, etc.)

Table 4: The 12 Pitch Classes
Source:
https://viva.pressbooks.pub/openmusictheory/chapter/pitch-and-pitch-class

the instrument. That is caused by *timbre*. It also contributes to the *energy* variable in the audio features section. As previously mentioned timbre is represented by a twelve dimension vector, one for each Segment. The first value in every vector is the average loudness (see audio features), the second value is brightness (happier songs have higher values).

4.2 Processing the data

All the processing of the data was achieved through the usage of Python programming language. The implementation of the code was done on Google Collab because of the plethora of available libraries without installation and the processing power of the software.

4.2.1 Audio Features

As previously mentioned, the audio features of spotify provide 16 different variables that provide information about every song. Along with these the final data set that was downloaded through spotify API also provided the ID of the song (a unique number spotify provides), the title of the song, the first artist and all the other artist that may be featured in the track. In addition to those, there is also the "emotion" variable that was added as explained. That brought the data set to twenty-one variables. Of course not all of the were used in the end to create the models.

The data set did not need much processing. There were no missing values and the variables that should be float were float and the ones that should be integers were integers. Some insight about the data were drawn from a correlation plot (see figure 9). The variables that seem to be the most negative correlated (around -0.75) are *accousticness-energy* and *accousticness-loudness*. There also appears to be some negative correlation between other variables: *num_segments - accousticness*, *danceability - accousticness*, *danceability - instrumentalness*.

On the positive correlation side, an unsurprising correlation appeared which is the one between the varibale that represents the duration of the song (duration_ms) and the three layer ones: *numsections, num_bars and num_degments*. As previously mentioned those three features of each song are in reality a "*division*" of the song in parts depending from which perspective one is looking at the song from. It is only logical that the duration of the song would have a relationship with the number of Sections, segments and bars. A longer songs means more of the aforementioned. Other pair of positively correlated values are *energy* and *loudness*, the *numsections, num_bars and num_degments* with each other. An approxiate 0.75 correlation appeared between *valence* and *dancaebility*, *tempo* and *num_bars*. The *to be predicted* variance seemed slightly correlate with the *energy*, *loudness*, mode and slightly negative with *key*, *liveness* and *tempo*. The decision was made to remove from the data set any variables with a correlation of 0.80 and above. Those variables were 'loudness' and 'acousticness'.

Valence

Valence is value that defines the emotion of the song on Spotify. The higher the valence the "happier" the song is and the opposite. The lower it is the "sadder" the song is. The lowest value that appears in the data is 0.0273 and it is an instrumental theme from a movie. The theme is called "*Main Titles - Why Are We Here*"[39] and is written by *Jeff Russo* and is

---

[39] https://www.youtube.com/watch?v=CRNa1Qpfk_c&ab_channel=JeffRusso-Topic

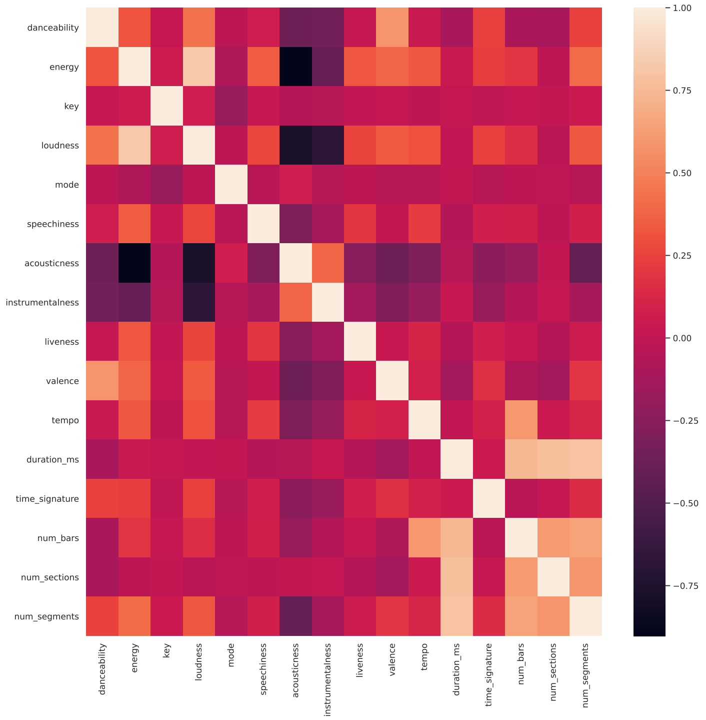


Figure 9: Correlation plot of the Variables

categorized in the "Calm" spectrum. Another look in the data and it is "visible" that it possible is a theme and not a song as it's *intrumentalness* value is 0.865 and *danceability* 0.0629. On the other hand, the highest value that appears is 0.972 and it is a song called "*Jaspers Keys*"[40] by *Doorly* and it is a song without lyrics, a dancing uplifting beat. It is categorized in the "Happy" bin, its danceability is 0.801 and its energy is 0.764. It's intsumentalness is also high (0.839) which is to be expected as the song does not contain any lyrics.

The *Happy* category has the highest concentration of values closer to 1.0 than any other category (see Figure 10). The mean and median of the category were fairly close 0.607 and 0.628 respectively. Also, the range (R=0.81)

---

[40] https://www.youtube.com/watch?v=Ks4aLxr1FxI&ab_channel=Doorly-Topic

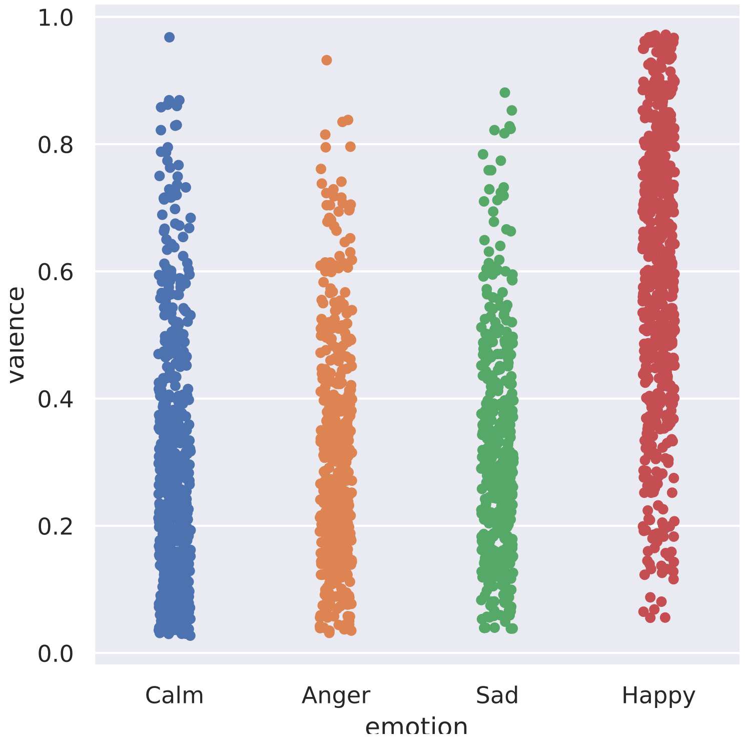


Figure 10: Each song's valence and emotion

was the lowest between the categories. The higher value of the category was the previous-mentioned song. Although "happy" songs are considered to have high value (this seems to be true in most cases), the category has it's own "outliers" with values in all the spectrum (see Figure 10) and the lowest one being 0.055. It is a song called "Brighter Days"[41] by two artists *Bingo Players& Oomloud*. This is a "regular" songs, that contains lyrics and a very uplifting music. It is weird that it's valence is so low and there is a possibility that some of the other variables have a value that brings valence down.

The *Calm* category is the one that appears to have it's data points most spread in the scale. This can be guessed by looking figure 4, and confirmed by the fact that it is the category where the mean (0.274) and median (0.232) have the largest difference and the range (R=0.941) is also the highest be-

---

[41] https://www.youtube.com/watch?v=vDlVka1ujeU&ab_channel=Spinnin%27Records

tween the other emotions. The lowest value that appears in this category is the lowest of the data set, the instrumental piece. On the other hand, the highest rated valence was 0.968 and it is a song, not an instrumental piece, called *Desire* [42] by an artist called Butch. It is not a song that would be characterised as “calm”, but it is plausible that it is in this category because it could be considered a relaxed/chilling “vibes” song. This song is a clear example of a data point that could be controversial about the “calm” label and could also confuse an algorithm and cause misclassification. At the same time it could be a data point valuable for the regression models in order to predict valence.

The *Angry* category is the second most spread category in the data. It’s range is 0.901 and the difference between the mean (0.304) and median (0.271) is the second largest after “*calm*”. The highest valence that resides within the “Angry” spectrum is 0.932 and it is a song called *Back of Your Head*[43] by the band Screeching Weasel. It is a punk-rock song, that it could be considered as angry by the formula that calculates valence, because it is “loud” and it has electronic instruments. On the other side of the “angry” spectrum is a song with a valence value at 0.0316. It is a song called “THE DROP”[44] by a DJ named Gammer. It is a beat with no lyrics and it could be considered angry because it is a loud, fast pacing with many intense “moments”.

The final emotion, “sadness” seems to be, unsurprisingly, the exact opposite of “happiness”. Most of the values are concentrated at the lower end of the spectrum, closer to 0.0 and the “sadness” range is 0.023, the second lowest after the “happy” range. The same is true about the difference between the mean and the median with their values being 0.313 and 0.29 respectively. Two songs share the highest valence (0.881) and they are “*Come A Little Bit Closer*” [45] by the band Jay & The Americans and “Fox On The Run”[46] by the band Sweet. Both are songs that with a first hearing would not be considered sad. The song with the lowest valence (0.0384) is a song called *Berlin*[47] by an artist called dRY X and it is definetely a song that would be

---

[42] https://www.youtube.com/watch?v=acJgpX4jsGs&ab_channel=MsRunningBack

[43] https://www.youtube.com/watch?v=_eIRot2tmEk&ab_channel=ScreechingWeasel-Topic

[44] https://www.youtube.com/watch?v=VG8hI_5R2tw&ab_channel=MonstercatUncaged

[45] https://www.youtube.com/watch?v=ZuWkVqum6a8&ab_channel=top401965

[46] https://www.youtube.com/watch?v=VP2umy6TdEU&ab_channel=SweetVEVO

[47] https://www.youtube.com/watch?v=cFXN20bpWtY&ab_channel=RYX

considered sad.

The interpretation and perception of an emotion is a question that can cause disagreements and ambiguity (Schimdt and Kim, 2011). Identifying the emotion emotion using valence does not seem to be clear in all cases either. Some songs could be in the playlists by mistake, or by design or by the "algorithm" combining what the listeners like to hear together or that the "classification" is based on other variables as well. Despite all that the happy songs have the highest valence and mean out of all categories and the sad ones the lowest valence and mean (see Table 5). Maybe the divisions of the songs to calm/angry and other categories is subjective and valence works better for clear happy and sad songs. Or maybe some songs have some values in their variables that end in lower/higher valence that the "emotion" that the songs conveys to the listener.

Emotion	Mean Valence
Happy	0.607
Sad	0.274
Angry	0.304
Calm	0.313

Table 5: Mean *valence* comparison between emotions

When compared with other variables, valence does not always seem to be an important factor in the separation of the categories. This is not always the case as when plotted against some variables the categories seem to be visible. Of course there are intertwined data but a clustering algorithm could possible separate the categories with some success (see figure 11). The categories seem to cluster better plotted against energy and loudness.

4.2.2 Audio Analysis

As previously mentioned, the Spotify Audio Analysis section, offers data for the structure of all the song from it's catalogue. Those data are divided in the five categories : Sections, Segments, Beats, Bars and Tatums. The process for processing all those data was the same. All the songs have five different excel files, one per musical feature. There was a need to create one data set for each feature (not five) in order to create an input data set for a model. With this aim, at first all the data for each features were imported into

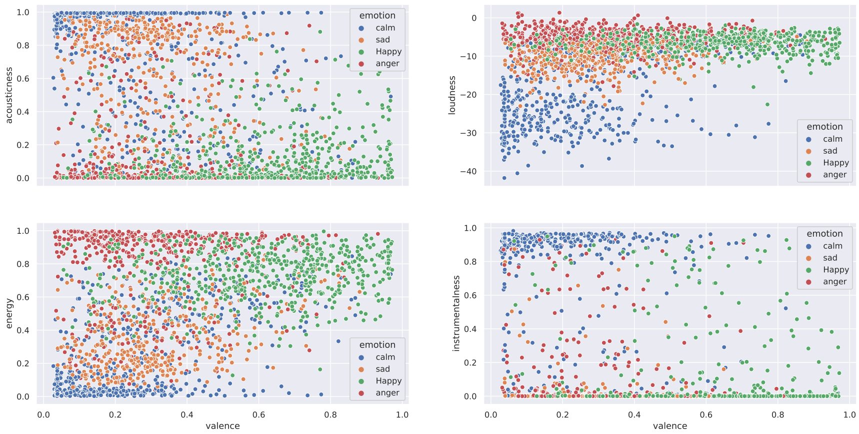


Figure 11: Valence plotted against 4 different variables and the emotions

a 3D data frame, one for each feature. The first dimension were the variables for each feature, the second was the features and the third the songs. In each 3D data frame, because of the variety in the songs' duration, there were different length in the dimensions. This was an issue that was addressed by adding the value zero everywhere that was necessary in order to make the features dimension even for each feature. For example if the largest song had fifteen Sections then all the songs appeared to have an equal amount but the subtraction of their real number of Sections to fifteen was filled with zeros. Because the emotion was not available in those data sets, the Spotify ID (which is unique) was used in order to match the songs emotion from the "Audio Features" data set.

5 Implementation-Results

In this document an effort was made to predict the emotion from the spotify playlists using the different data (Audio Features and Audio Analysis) the company provides and to predict the “valence” variable using the same data set. In order to achieve this some Classification and Regression methods were used, respectively, from Random Forests to Neural Networks.

5.1 Methodology

Using the Audio Features data both emotion classification and valence prediction was attempted. For the first part, the emotion classification , using the audio features data set four methods were implemented. Those were an MLP neural network, a random forest, LightGBM and XGboost. All data were scaled in order for the machine learning algorithms to have better results. In order to have a general idea for the models’ performances a “dummy classifier” was implemented. This means that this “model”/classifier will classify the songs based on a simple rule without any training. The dummy classifiers used here were a *most frequent* one and a *uniform* one. The first classifies the data always based on the predominant class of the data and the second one predicts uniformly at random. Those classifiers are useful because they create a baseline to compare the models. If the models have worse performances than them, then there are no good and need to be rebuilt. The first classifier had an mean accuracy 29.75% and the second one a 24.79%. This offers all the models above 30% a chance to be reviewed and all below that level to be discarded.

The first model used for the emotion classification was an MLP neural network. The *keras* library[48] was used to build and implement the model. The neural network was a sequential model with a dense layer of 34 neurons in the input. The variables used were 14 as loudness and acousticness were not included as they were deemed as highly correlated variables. On the input layer a ReLU activation function was added as well (see page 9-10). On dense hidden layer was added to the network, one with 16 neurons. Then as an output layer was a dense layer with 4 neurons, representing the four emotional classes of the problem. The architecture of the network had in total 1076 trainable parameters. As a loss function, sparse categorical crossentropy was

---

[48] https://keras.io/

chosen and Adam as an optimizer. The meter that drove the network's train process was the accuracy of the classification. Furthermore, all classes had almost an even amount of data points and even though it was not necessary, some class weights were added to the model to ensure even training and validation for all categories of emotions.

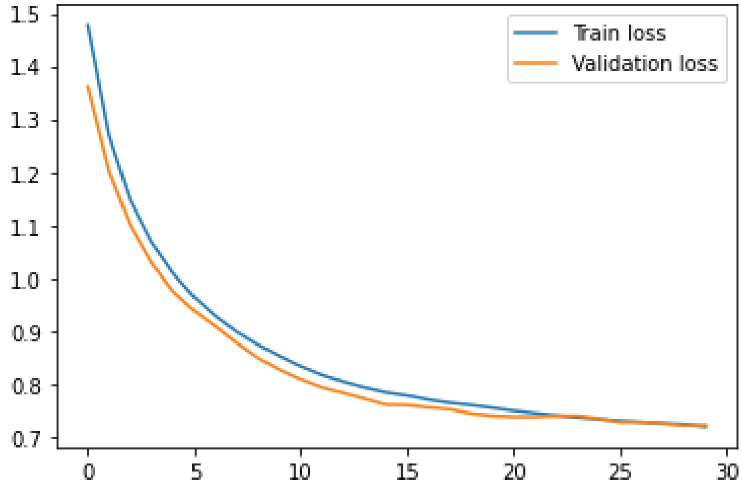


Figure 12: Loss curve of MLP model for classification

Not all data were used for the training and validation process of the network. A portion of the data was kept out of the sight of the model during the training process to be used as a test data set of unseen data to further evaluate the model. The model was trained for 30 epochs with 150 data points each time (batch size). The loss curve (see figure 12) showed a good learning rate before train and validation curves meet after 22 epochs. On the accuracy curve (see figure 13) there seems to be no significant gap between the train and validation accuracy curves. If the gap were significant this would be an indication of an overfitted model.

Another technique used for classification was XGboost machine learning algorithm. The variables 'loudness' and 'acousticness' were removed from this implementation as well as their correlation was over 0.80 and all the other variables included in the model and scaled for better results. There was a 65-35 stratified[49] split on the data for train and validation of the model. Some of the parameter setting of the model were a learning rate of 0.1, a max

---

[49] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedShuffleSplit.html

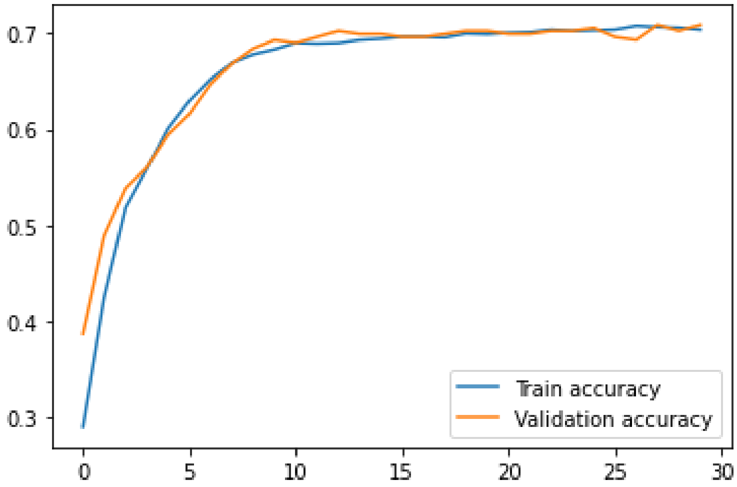


Figure 13: Accuracy curve of MLP model for classification

tree depth for base learners of 3 and 100 gradient boosted trees. Finally the algorithm was chosen to run using one parallel thread and a 10 fold cross validation was conducted for evaluation purposes.

From the gradient boosting algorithms LightGBM was also implemented on the data as well. The same train-test split was used on the data just like XGBoost and the two aforementioned variables were removed as well. No class weights were assigned to the data, the learning rate was set to 0.1 again. As far as the classifiers were concerned, 100 gradient boosting trees were employed and the maximum number of tree leaves for the base learners was set to 31 with the max depth set to limitless.

Two Random Forest models were built for the classification. Because the Random Forests showed prominent behaviour on the data a *Grid Search* was utilized to boost the accuracy of the model. It tries all different combinations of hyper parameters to find what fits best. From that procedure a third model was built from the results. The criterion for ranking the feature importance was set to be Gini in all situations as it was deemed to be fit for the circumstances. Some of the parameters changed and investigated through grid search were the number of estimators, the maximum depth of each tree, the number of features considered for optimal splits, the minimum samples for each leaf node, the minimum number of samples required to split an internal node and the bootstrap parameter. The last one defines whether the whole sample will be used in the building process or bootstrap samples[50]. All values are presented in table 6.

Parameters	Model 1	Model 2	Grid Search Model	Range in Grid Search
Bootstrap	True	True	False	True of False
Max depth	None	70	10	10, 31, 52, 73, 94, 115, 136, 157, 178, 200, None
Max features	auto	auto	sqrt	auto or sqrt (sqrt of the number of features)
Min leaf samples	1	4	4	1, 2, 4, 5
Min split samples	2	10	2	2, 5, 10, 15
Number of Estimators	500	1000	1166	500, 666, 833, 1000, 1166, 1333, 1500, 1666, 1833, 2000

Table 6: Random Forest Model parameters & Grid Search range

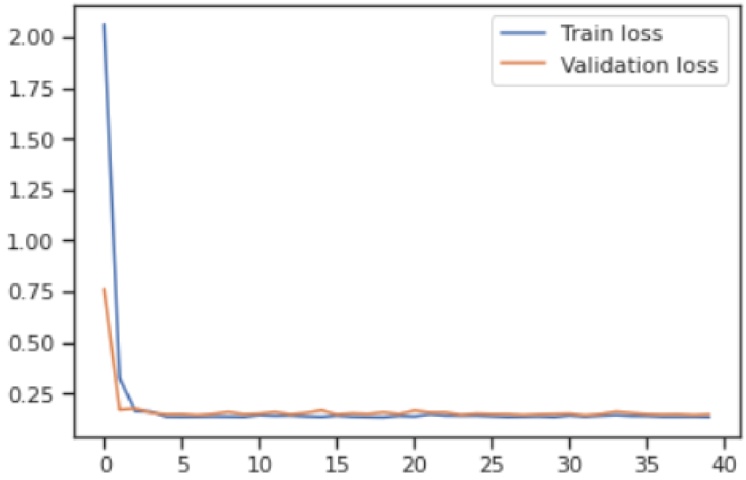


Figure 14: Loss curve of MLP model for regression

For the valence prediction part of this document, two regression techniques were utilised. A random Forest Regressor and a neural network. For the MLP regressor neural network the keras library was used just like the classification NN. The data were scaled again, and loudness and accousticness were not included in the data. This time valence was not in the training

---

[50] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

data set either as it was the dependant variable of the procedure. The initial data were split as 70% train and 30% test and validation. The 30% has split in half, the first part as testing data during the training process and the other half as a validation data set to evaluate the model. On the input layer a ReLU activation function was added again (see page 9-10) with 64 neurons and 13 variables. Then another hidden dense layer with 64 neurons was included as well. The final layer has one neuron only as it has no classes to classify data only a valence value is predicted for every song. The learning rate was set to 0.1 and the optimizer was Adam. The meter that drove the network's train process was the mean absolute error. It was chosen because it is less sensitive to outliers.he architecture of the network had in total 5121 trainable parameters. In the training process, 40 epochs happened and the batch size was 100 songs. Finally, the loss curve (see figure 14) showed a higher learning rate before train and validation curves meet after 5 epochs.

The other method utilized for regression was the Random Forest Regressor. The same split (70% split, 30% test data)vwas used for this method just like the MLP neural network. For the prediction process 1000 decision tress estimators were chosen with a max depth set to "none" (no limit). The bootstrap parameter was set to "bootstrap" and the criterion was the mean squared error.

Using the Audio features data, emotional classification of the songs was attempted. The songs and emotional annotation were the same as the previous data set . For each song were 5 data sets one for each layer of it's musical structure (Segments, Sections, Bars, Beats and Tatums). All the data of each layer were gathered into a 3 dimensional data set to train the models. All the data sets were scaled according to the first song of the data. Furthermore, all data sets were split into train validation and an unseen split for performance evaluation. Just like the first data set, a dummy classifier was applied in order to have a baseline for the results. The dummy classifier used here was a *most frequent* one and it's accuracy was 28%. Because the songs are the same in all data sets it is expected that the classifier will have almost the same results in all data sets with differences only in the decimals.

The first data set was the "Segments" layers of the songs. A stratified split of an initial 60-40 train-validation split was done and then an additional 20% was kept unseen by the model. Even though the data were almost equally distributed amongst the classes, class weights were also applied. The model chosen for the classification was a Recurrent Neural Network. The model was a sequential and the first layer was a LSTM layer with 64 neurons. A

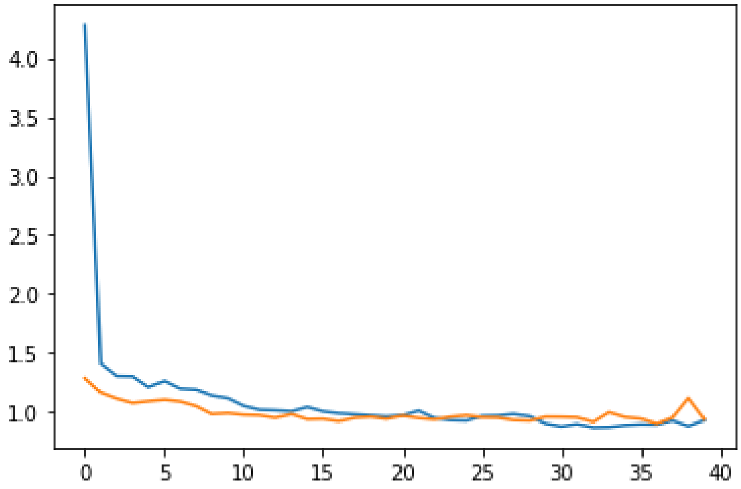


Figure 15: Loss curve of RNN model for Segments data set

flatten layer followed, which is common to be after a 3 dimensional input layer. Then a dense layer with 32 neurons and a ReLu activation function and a dropout layer of 30% . A dropout layer is one that ignores a percentage of the neurons in order to avoid overfitting problems. The exiting layer had 4 neurons same as the classes and a softmax activation function.

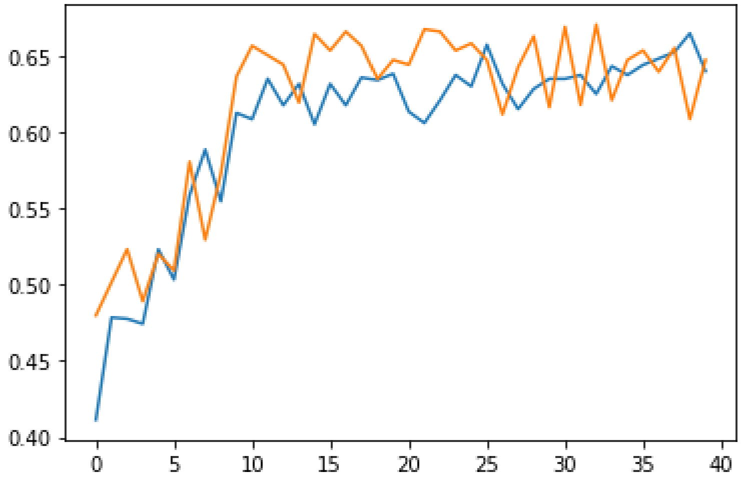


Figure 16: Accuracy curve of RNN model for Segments data set

The model was compiled with a categorical crossentropy loss function and an adam optimizer. The metric for evaluation during the training was the validation accuracy. The batch size of the training process was 120 songs and the training was set to 20 epochs. The loss curve (figure 15) shows a high learning rate and slight overfitting after epoch 28. On the accuracy training curve (figure 16) no huge distance between the graphs appear but there are on the same spectrum.

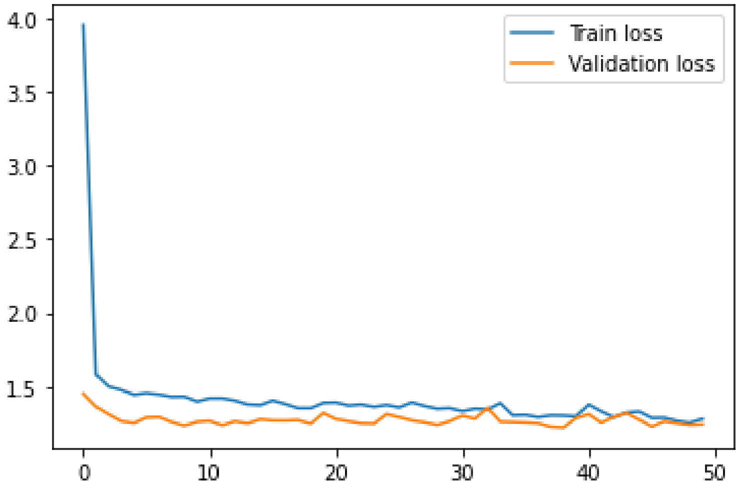


Figure 17: Loss curve of RNN model for Tatums data set

The next musical layer data set to be used were the “Tatums” data set. The same train-test-validation split was applied to the data as the “Section” data set and class weights were applied. A Recurrent neural network was applied to these data with the same structure as the Segments model. An input LSTM layer with 64 neurons, a flatten layer, a dense layer of 32 neurons and a ReLu activation function, a dropout layer (30%) and the output layer with 4 neurons and a softmax function. The model was compiled with a categorical crossentropy loss function and an Adam optimizer. The batch size of the training input was 150 songs and the model was training for 50 epochs. On the loss curve (figure 17) there seems to be a high learning rate and the two curves meet at almost 50 epochs and no overfitting issues can be detected through the graph. On the other hand the accuracy graph (figure 18) seems turbulent but the two lines follow the same path after 40 epochs.

In the third model, the beats model, again the same train-test-validation

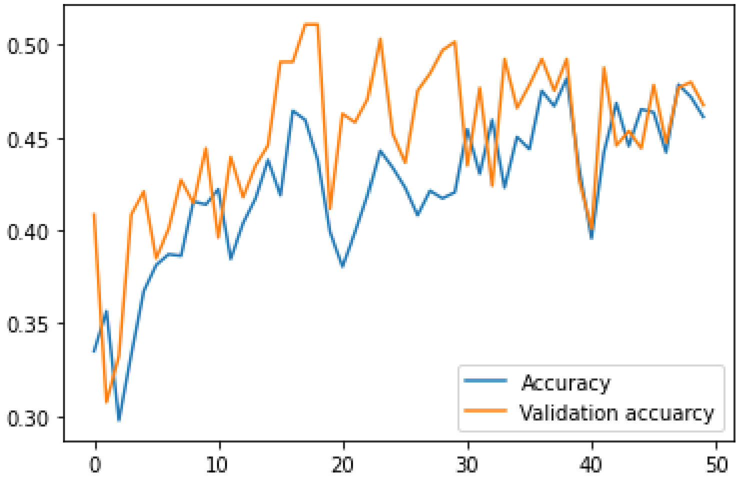


Figure 18: Accuracy curve of RNN model for Tatums data set

was used. The model used, was again an LSTM recurrent neural network model with a different architecture than the predecessors. The model was built as a sequential model, with a LSTM input layer containing 32 neurons followed by the flatten layer. Two dense hidden layers came next with ReLu activation functions, the first contained 16 neurons and the second 8. The next layer, with a softmax function, was the output layer with the four neurons representing the four classes. No dropout layers were included as it seems that they downgraded the performance. The model was compiled with a categorical crossentropy loss function and an Adam optimizer just like the previous models but the learning rate was set to 0.01 as higher or lower values resulted in excessively overfitted models.

When training the model for 30 epochs, 100 songs were set to be the batch size and class weights were also applied. On the loss curve (figure 19) there seems to be a high learning rate and the two curves meet at 30 epochs. It seems that there are not overfitting issues but it is unclear what would happen if the model trained for more epochs. On the accuracy graph (figure 20), many ups and downs appear but at the end the validation accuracy curve (orange line) has an upward trend. There is a gap between training and validation accuracy before the lines meet at 26 epochs before separating again.

The next layer was the *Bars* data set. Again the same train-test-validation

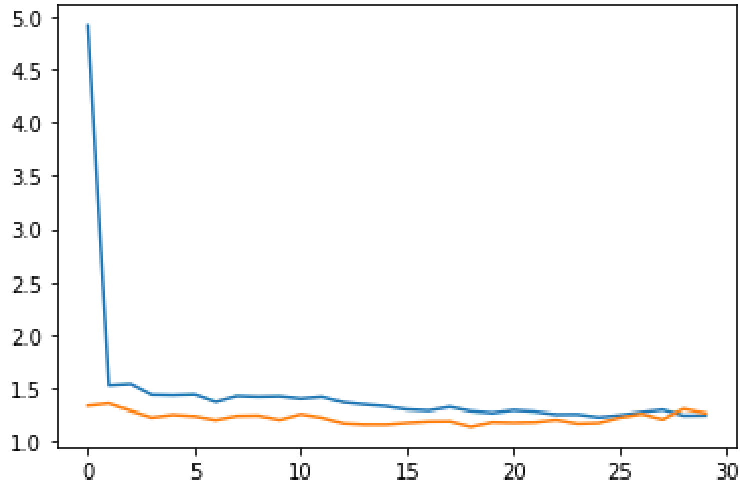


Figure 19: Loss curve of RNN model for Beats data set

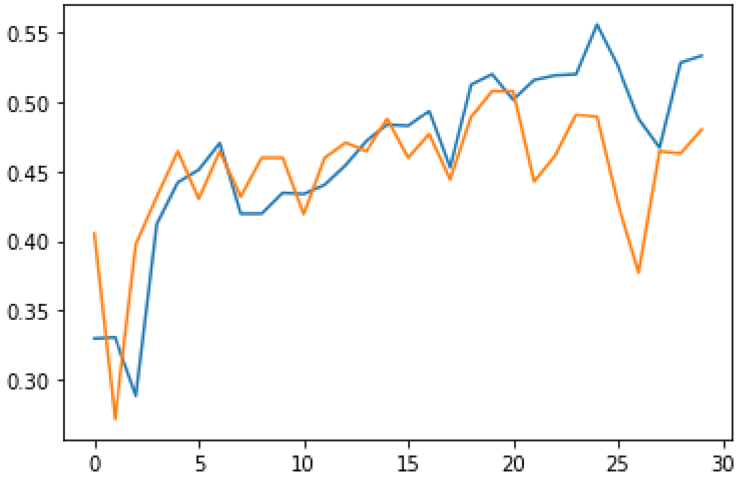


Figure 20: Accuracy curve of RNN model for Beats data set

was used. The model, was an LSTM RNN with an input layer with 32 neurons. Then the flatten layer was applied like the rest of the models, followed by a 32 neuron dense layer with a ReLu activation function. Two hidden dense layers followed with 32 neurons each. Finally the output layer with 4 neurons and the softmax activation function. No dropout layer was added to this model. The model was compiled with a categorical crossentropy loss

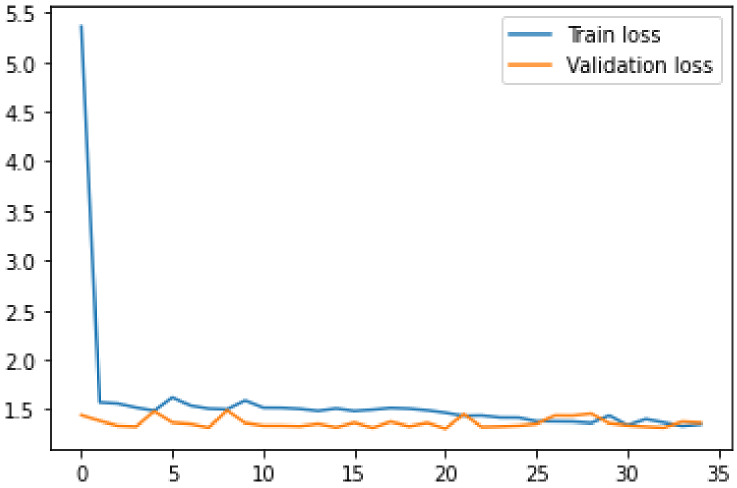


Figure 21: Loss curve of RNN model for Bars data set

function and an Adam optimizer just like the previous models but the learning rate was set to 0.01.

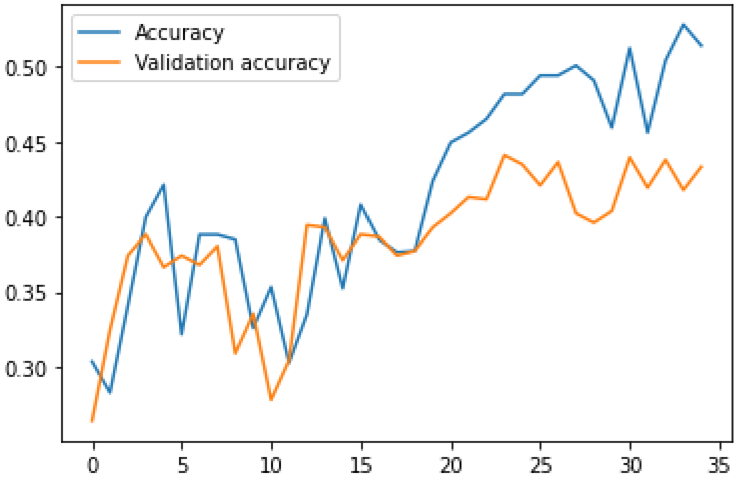


Figure 22: Accuracy curve of RNN model for Bars data set

The model was trained for 35 epochs and 100 songs were used as a batch size. Just like the rest of the models, class weights were applied to the data. The loss curve (figure) shows a high learning rate and a good result as the train loss and validation loss meet before 35 epochs. On the accuracy graph (figurefig: bars accuracy curve) there is a gap between the two lines after 15 epochs. The validation accuracy curve is at 35 epochs at an upward trend.

The last model, the Section data set did not produce any promising models. Different neural network architectures were tried but the data seemed to not be a good fit. The accuracy and loss curves always indicated extremely underfit models.

5.2 Results

5.2.1 Audio Features Emotional Classification

The *Audio Features* data set was used to classify the Spotify-labeled emotions of the songs. In table 7 the mean classification accuracy results of the 8 models, including the dummy classifier, are presented. The best accuracy results were produced by the Random Forest Grid search model with a 75.33% but that was not the case on the accuracy of positive predictions (precision) (Mohajon n.d.) per emotion (table 8). In non of the individual classes the model provided the highest precision score of the respective class. It had the second highest precision on *happy* songs (86%) , on *angry* songs (81%) and *calm* songs (68%). All models had relatively high scores with the biggest difference between them being 4.34%. The MLP neural network model had the lowest accuracy score (70.99%) , excluding the dummy classifier.

Model	Total Accuracy (%)
Dummy Classifier	29.75%
MLP Neural Network	70.99%
LightGBM	72.91%
XGBoost	73.15 %
Random Forest (Grid Search Model)	75.33%
Random Forest Model 1	74.97%
Random Forest Model 2	74.97%

Table 7: Mean classification accuracy per model

The results when examining the precision of the models on emotion classes were fluctuating. The highest score for *happy* songs was achieved by the random forest model 2 (RFM 2)with the 1000 estimators (87%), followed by the Grid search model (86%) and the random forest model 1 (RFM 1) with 500 estimators (85%). In total the happy songs were most successfully classified by the models with the lowest precision score being 73% by the MLP neural network. The highest scores for *angry* songs was achieved by LightGBM (74%) followed by the RFM 1 (83%). The lowest score was achieved by the MLP NN (73%). For the *calm* songs the results were significantly lower than the first two categories but consistent through the different models. Half of the models had a precision score of 65% , two models had a 63% and one a 64%. The biggest fluctuations appeared on the classification of the *sad* songs. In total contrast to the other categories the highest precision score was achieved by the MLP NN (77%) followed by the Grid Search model (68%) and the lowest by LightGBM (65%).

Method	Happy	Angry	Calm	Sad
Dummy Classifier	26%	27%	21%	24%
MLP Neural Network	73%	73%	63%	77%
LightGBM	80%	84%	65%	59%
XGBoost	82%	79%	65%	62%
Random Forest (Grid Search Model)	86%	81%	63%	68%
Random Forest Model 1	85%	83%	65%	62%
Random Forest Model 2	87%	81%	64%	64%

Table 8: Total Classification accuracy of positive predictions (precision) per model and per emotion

To further investigate the performance of the models on the classification of emotions the f1 score was calculated per model and per emotional category (table 9). This metric is usually used to compare classification models and it ranges between 0 and 1. It is considered a weighted average of other two metrics which are the previous mentioned *precision* and *recall* which is "*the fraction of positives that were correctly identified*". The higher the f1-score the better for the ability of each model to correctly predict the label of an instance. The higher scores were achieved by the *angry* songs, with the random forest models having the best f1-scores and LightGBM following (0.83). The results were promising on the *happy* category as well (form 0.73 to 0.78). The lowest F1-scores were seen in the *sad* songs as all models had results ranging from 0.60 to 0.66. Finally, the *calm* songs had scores between 0.68 and 0.71.

To validate that the results of the model were not due to a specific train-split of the data, a 10 fold cross validation was performed in all models except the MLP NN. This was the case because the model was validated on unseen

Method	Happy	Angry	Calm	Sad
MLP Neural Network	0.73	0.76	0.71	0.64
LightGBM	0.77	0.83	0.68	0.60
XGBoost	0.75	0.82	0.69	0.64
Random Forest (Grid Search Model)	0.78	0.85	0.70	0.66
Random Forest Model 1	0.78	0.85	0.70	0.63
Random Forest Model 2	0.78	0.84	0.71	0.64

Table 9: F1-score per model and per emotion

data as previously mentioned. All the other models were subjected to a 10-fold cross validation. This method splits the data on k (on this document 10) parts and each time it uses a different part as a test data set and the rest of the data as a train. This way, the case of the model randomly having high accuracy scores is eliminated. The total accuracy for the models (table 7) could also be interpreted as a mean accuracy of all accuracy scores in each iteration of the cross-validation. On table 10 the best and worst accuracy scores of the procedure are presented in percentage for each model. The highest accuracy was achieved by a fold of RFM 1 (81.81%) and the lowest by a fold of XGBoost (65.06%). It is worth mentioning that the RF Grid search model who had the highest mean accuracy, had also the highest lowest accuracy out of all models, thus explaining the high average.

Method	Highest accuracy score	Lowest accuracy score
LightGBM	79.08%	67.97%
XGBoost	81.71%	65.06%
Random Forest (Grid Search Model)	80.39%	72.08%
Random Forest Model 1	81.82%	70.59%
Random Forest Model 2	80.39%	70.59%

Table 10: Highest and Lowest accuracy scores of the 10-fold cross validation per model

5.2.2 Audio Features Valence Prediction

Once the songs were classified by their emotion, the valence prediction procedure followed. As previously mentioned a Neural network regressor was used at first in order to predict valence. After training for 40 epochs, the loss of the model on evaluation was 0.1506. A 50% of the test data were kept unseen from the model to be tested for validation. On figure 14 the predicted data can be seen (blue) in comparison to the real values of valence (red). It is noticed that while valence ranges from 0 to 1, the model has predicted 3 negative values (-0.009633, -0.081935, -0.009003) and some over 1 (ex 1.12). The model was further evaluated with the calculations of two metrics: the root-square-mean-error (rmse) and the mean squared error (mse). The first one was 0.196 and the latter was 0.038. The rmse error shows the differences between the real data and the prediction. The mse offers the average of the set of errors that appear between data and prediction and it is considered that the lowest the value the better prediction the model has made.

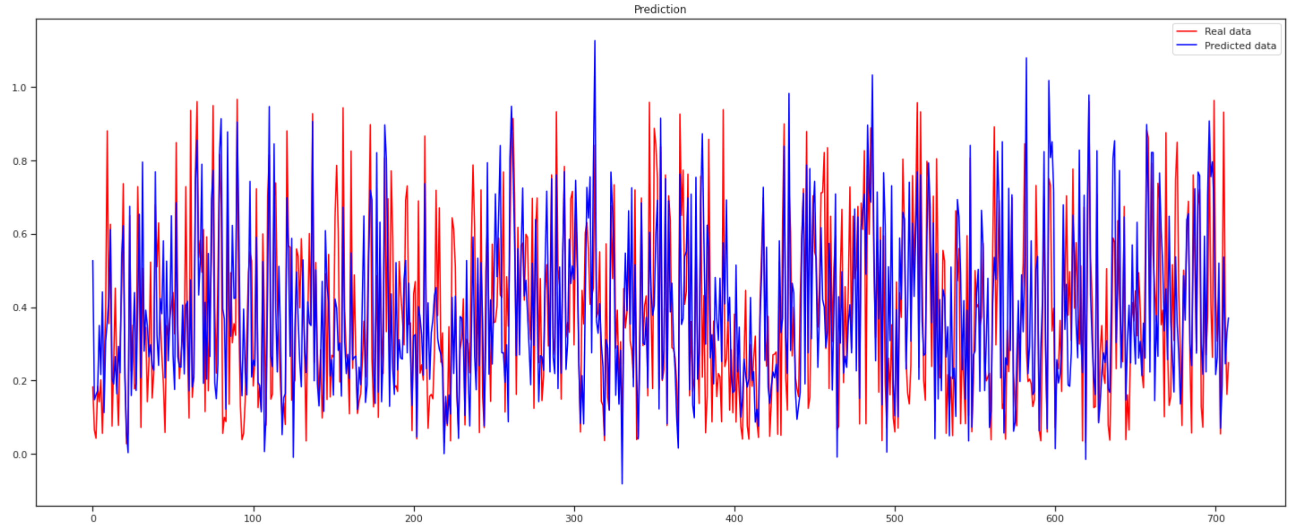


Figure 23: Prediction of valence compared to their real counterparts (NN model)

The same metrics were used to evaluate the performance of the Random Forest regressor (except loss). On figure 15 the same comparison of predicted data and real values is presented. It is visible that the model has predicted relatively lower values as the predictions do not cross neither the upper nor the lower limits of the valence thresholds. The argument that the model outperformed the NN can be made using the rmse and mse values that were calculated. Both were lower than the previous presented numbers. The rmse was 0.158 and mse was 0.025. The data seemed to be a better fit for the second model rather the first.

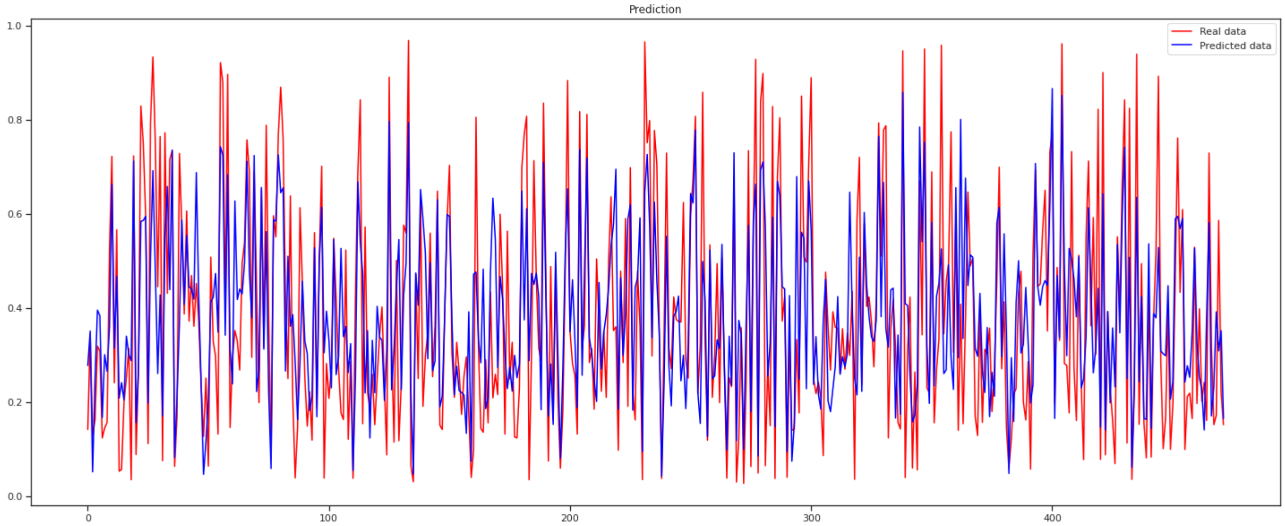


Figure 24: Prediction of valence compared to their real counterparts (Random Forest model)

5.2.3 Audio Analysis Emotional Classification

The Audio Analysis data sets were used for emotional classification of the songs. The analysis was performed per structural musical layer. As previously mentioned only the Sections data set did not produce any models worth mentioning. This left the other four layers and the dummy classifier for comparison. In table the test classification accuracy results of the 4 models, including the dummy classifier, are presented. The best accuracy results were produced by the Segments model with 67.081% accuracy on the data set. Even though it is lower than the audio features data set it is almost double the results of the dummy classifier/ The second best accuracy was achieved by the Tatums data set (57.764%). The Bars and Beats data sets were very close in their results as their difference in test accuracy was less than 2%.

Data set	Test accuracy
Dummy Classifier	28.78%
Segments	67.081%
Tatums	57.764 %
Beats	50.932 %
Bars	49.068 %

Table 11: Test accuracy of the Audio Analysis models

In table 12 the precision per layer data set and per emotion is presented. The results are widely different than the audio features data sets as the highest score was achieved by the Bars model on the happy songs and it was 85%. On the other hand the lowest score was achieved by the Bars model again on the sad songs, being only 6%. On all emotion except happy the model that achieved the highest precision score was the Segments model. The highest score on angry songs was 84%, on calm songs 51% and on sad songs 76%. In general the lowest scores were on the calm songs and the higher on happy songs.

Method	Happy	Angry	Calm	Sad
Dummy Classifier	66%	38%	32%	9%
Segments	74%	84%	51%	76%
Tatums	83%	51%	40%	54%
Beats	72%	76%	31%	21%
Bars	85%	51%	42%	6%

Table 12: Total Classification accuracy of positive predictions (precision) per layer and per emotion

All models had high and lows scores depending on the scale of their general performance. The segments model had 3 scores over 74% but a 51% on calm songs. The tatums model was the second best on happy songs but the other categories were on 54% or less. The beats model had high scores on happy and angry songs but almost half on the other two categories. Finally the bars model had the highest result of the table and the lower as well.

In this part of the research the f1 scores was calculated as well with the same structure as before (13). The results were different from the audio features ones. This time the angry songs did not have the best performance in all cases, only on the Segments model (0.79). Excluding this, the happy songs had the best scores in the other three models. The segments model has the best in all categories, all above 0.64. The sad songs on the beats and the bars model had the most disappointing results with a 0.25 and 017 respectively. The sad songs result were on par with the audio features results. Finally, the calm songs had a relatively good score on the Segments model (0.67) and the tatums one (0.55) and lower on Beats (0.42) and Bars (0.50) models.

Method	Happy	Angry	Calm	Sad
Segments	0.74	0.79	0.67	0.64
Tatums	0.67	0.51	0.55	0.52
Beats	0.66	0.56	0.42	0.25
Bars	0.65	0.51	0.50	0.17

Table 13: F1-score per layer and per emotion

6 Conclusions

Spotify is a company for artists, producers and music lovers with great opportunities for analytics practicum and different tool building. It is possible for one to use the data it is providing for many different applications not only for machine learning causes but for musical composition cases. The data available do not only benefit the curious, the developers and data enthusiasts but the company it self. Using those data produced by the everyday use of the platform and the ones by the songs, the company is able to better understand the market, keep the clients interesting in the content the know and love and introducing them to new all the time. This also benefits the artist that promote their work. The situation in general can be considered beneficial for all parties involved. The creation of the different playlist based on the emotion and ones mood is a complex one. It involves not only info from the song it self but how it is paired by the users with other songs as well.

In this document, an effort was made to mimic the creation of the playlists using machine learning models and different data and also predicting the value of valence which is considered a central part of the emotional identity of each song. For the first part of the analysis the most promising results were produced by the Random Forest model in the Audio features data set after a Grid Search application. The results indicate that it can be considered the most effective method for the aforementioned data set. On the emotion level of the analysis the best results were achieved by the happy and angry songs and not the calm and sad ones. This can be considered a step for feature research, to use only those categories as opposite side of the spectrum or the happy songs as in a binary problem in which a song is either happy or it is not. It is believed that the calm and sad categories even the angry songs were the more ambiguous ones because even individuals would disagree in cases. One can see an angry song as a sad one and vice versa.

The other data set used to predict the emotions of the songs did not produce as high results as the previous one. That does not mean that it did not produce promising results in categories and in different layers. One by seeing the results can say that the best results were by the Segments model and that the emotion of the song is hidden in this lowest layer of the music structure of the song. It appears that the deepest on dives into the layers of a song can define the emotion conveyed better. The next best results were produced by the second deepest layer, the tatums. The other two following layers the Beats and Bars produce semi - promising results and by reaching the "surface" of the song the Sections, the emotion could not be found at all.

On the valence prediction side of the research, regression methods were utilised. The best results were once more produced by the Random forest models . The method seemed to fit the data very well and predicted valence values within the boundaries of the value itself. The neural network model did have some predictions outside the boundaries but the result were satisfying. It is believed that there are more things to be explored concerning this variable. At first it could be possible predicted using the Audio Analysis data as well or even different regression methods. Further more there is possibility to find or even compute the "arousal" in the data provided by Spotify and create a two-dimensional prediction for the variable.

The data provided are a rich source for data mining and analytics opportunities. For future expansion of this document it is proposed to implement the models on more playlists and songs to expand possibilities. Also it could be possible to ignore the labeled emotion and by either clustering methodologies or by creating an "emotional plane" like Thayer's from the available data to classify the songs in emotional categories. Finally as previously mentioned the Audio features could be potentially be used for valence prediction and other prediction projects.

References

Legendre, A.M. (1805). *Nouvelles méthodes pour la détermination des orbites des comètes*. Firmin Didot, Paris.

Wundt, Wilhelm Max (1896). *Grundrisse der psycologie [Outlines of psycology]*. Leipzig, W. Engelmann; New York, G.E. Stechert.

McCulloch, Warren S. and Walter Pitts (1943). "A logical calculus of the ideas immanent in nervous activity". In: *Bulletin of Mathematical Biophysics* 5, pp. 115–133. DOI: `https://doi.org/10.1007/BF02478259`.

Rosenblatt, Frank (1958). "A probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65(6), pp. 386–408. DOI: `https://doi.org/10.1037/h0042519`.

Russell, James (Dec. 1980). "A Circumplex Model of Affect". In: *Journal of Personality and Social Psychology* 39, pp. 1161–1178. DOI: `10.1037/h0077714`.

Thayer, Robert E. (1990). *The Biopsychology of Mood and Arousal*. Oxford University Press USA.

Bilmes, Je A. (1993). "Techniques to foster drum machine expressivity". In: *In Proc. Int. Comp. Music Conf. (ICMC*, pp. 276–283.

Bye, Dean L. (1993). *Mel Bay Presents Student's Musical Dictionary*. ISBN: 0-87166-313-9.

Breiman, Leo (2001). "Random Forests". In: *Machine Learning* 45, pp. 5–32. DOI: `https://doi.org/10.1023/A:1010933404324`.

Li, Tao and Mitsunori Ogihara (Nov. 2003). "Detecting Emotion in Music". In: *Proc. ISMIR 2003; 4th Int. Symp. Music Information Retrieval* 2003.

Jehan, Tristan (2005). "Creating Music by Listening". PhD thesis. Massachusetts Institute of Technology. URL: `https://web.media.mit.edu/~tristan/phd/`.

Yang, yi-hsuan, Ya-Fan Su, et al. (Jan. 2007). "Music emotion recognition: The role of individuality". In: *Proceedings of the ACM International Multimedia Conference and Exhibition*, pp. 13–22. DOI: `10.1145/1290128.1290132`.

Yang, yi-hsuan, Yu-Ching Lin, Heng-Tze Cheng, et al. (Dec. 2008). "Toward Multi-modal Music Emotion Classification". In: pp. 70–79. ISBN: 978-3-540-89795-8. DOI: `10.1007/978-3-540-89796-5_8`.

Yang, yi-hsuan, Yu-Ching Lin, Ya-Fan Su, et al. (Mar. 2008). "A Regression Approach to Music Emotion Recognition". In: *Audio, Speech, and Lan-*

guage Processing, IEEE Transactions on 16, pp. 448–457. DOI: 10.1109/TASL.2007.911513.

Eerola, Tuomas and Jonna Vuoskoski (Jan. 2011). “A comparison of the discrete and dimensional models of emotion in music”. In: *Psychology of Music*. DOI: 10.1177/0305735610362821.

Yang, Yi-Hsuan and Homer H. Chen (2011). *Music Emotion Recognition*. 1st. USA: CRC Press, Inc. ISBN: 9781439850466.

Panda, R. and R. Paiva (2012). “MUSIC EMOTION CLASSIFICATION: DATASET ACQUISITION AND COMPARATIVE ANALYSIS TEMPLATES FOR DAFX-08, FINLAND, FRANCE”. In:

Schmidt, Erik M. and Youngmoo E. Kim (2012). “Modeling and Predicting Emotion in Music”. In:

Panda, Renato, Bruno Rocha, and Rui Pedro Paiva (Apr. 2015). “Music Emotion Recognition with Standard and Melodic Audio Features”. In: *Applied Artificial Intelligence* 29, pp. 313–334. DOI: 10.1080/08839514.2015.1016389.

Yu, Liang-Chih et al. (Jan. 2015). “Predicting Valence-Arousal Ratings of Words Using a Weighted Graph Method”. In: pp. 788–793. DOI: 10.3115/v1/P15-2129.

Chollet, Francois (2017). *Deep learning with python*. City of publication New York, NY. ISBN: 9781617294433.

Ke, Guolin et al. (Dec. 2017). “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems 30 (NIP 2017)*.

Koutras, Athanasios (Sept. 2017). “Song Emotion Recognition Using Music Genre Information”. In: pp. 669–679. ISBN: 978-3-319-66428-6. DOI: 10.1007/978-3-319-66429-3_67.

Nguyen, Van et al. (June 2017). “A New Recognition Method for Visualizing Music Emotion”. In: *International Journal of Electrical and Computer Engineering* 7, pp. 1246–1254. DOI: 10.11591/ijece.v7i3.pp1246-1254.

Liu, Tong et al. (May 2018). “Audio-based deep music emotion recognition”. In: vol. 1967, p. 040021. DOI: 10.1063/1.5039095.

Tan, K, M Villarino, and Christian Maderazo (Mar. 2019). “Automatic music mood recognition using Russell’s twodimensional valence-arousal space from audio and lyrical data as classified using SVM and Naïve Bayes”. In: *IOP Conference Series: Materials Science and Engineering* 482, p. 012019. DOI: 10.1088/1757-899X/482/1/012019.

Hızlısoy, Serhat, Serdar Yildirim, and Zekeriya Tüfekci (Nov. 2020). “Music emotion recognition using convolutional long short term memory deep neural networks”. In: *Engineering Science and Technology an International Journal* 24, pp. 760–767. DOI: `10.1016/j.jestch.2020.10.009`.

Misra, Siddharth and Hao Li (2020). “Chapter 9 - Noninvasive fracture characterization based on the classification of sonic wave travel times”. In: *Machine Learning for Subsurface Characterization*. Ed. by Siddharth Misra, Hao Li, and Jiabo He. Gulf Professional Publishing, pp. 243–287. ISBN: 978-0-12-817736-5. DOI: `https://doi.org/10.1016/B978-0-12-817736-5.00009-0`. URL: `https://www.sciencedirect.com/science/article/pii/B9780128177365000090`.

Binieli, Moshe (n.d.). *Machine learning: an introduction to mean squared error and regression lines*. URL: `https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/`.

Breiman, Leo and Adele Cutler (n.d.). *Random Forests*. URL: `https://www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm`.

Brownlee, Jason (n.d.[a]). *A Gentle Introduction to XGBoost for Applied Machine Learning*. URL: `https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/`.

— (n.d.[b]). *Loss and Loss Functions for Training Deep Learning Neural Networks*. URL: `https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/`.

Bushaev, Vitaly (n.d.). *Adam — latest trends in deep learning optimization*. URL: `https://towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c`.

Chakure, Afroz (n.d.). *Random Forest Regression: Along with its implementation in Python*. URL: `https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f`.

Chauhan, Nagesh Singh (n.d.). *Optimization Algorithms in Neural Networks*. URL: `https://www.kdnuggets.com/2020/12/optimization-algorithms-neural-networks.html`.

Chen, Tianqi (n.d.). *Tianqi Chen*. URL: `https://tqchen.com/`.

Donges, Niklas (n.d.). *A COMPLETE GUIDE TO THE RANDOM FOREST ALGORITHM*. URL: `https://builtin.com/data-science/random-forest-algorithm`.

Education, IBM Cloud (n.d.). *Recurrent Neural Networks*. URL: `https://www.ibm.com/cloud/learn/recurrent-neural-networks`.

Goyal, Kechit (n.d.). *6 Types of Activation Function in Neural Networks You Need to Know*. URL: `https://www.upgrad.com/blog/types-of-activation-function-in-neural-networks/`.

Grover, Prince (n.d.). *5 Regression Loss Functions All Machine Learners Should Know*. URL: `https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0`.

Khandelwal, Pranjal (n.d.). *Which algorithm takes the crown: Light GBM vs XGBOOST?* URL: `https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/`.

Mandot, Pushkar (n.d.). *What is LightGBM, How to implement it? How to fine tune the parameters?* URL: `https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc`.

Mohajon, Joydwip (n.d.). *Confusion Matrix for Your Multi-Class Machine Learning Model*. URL: `https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826`.

Morde, Vishal (n.d.). *XGBoost Algorithm: Long May She Reign!* URL: `https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d`.

Muthukrishnan (n.d.). *Understanding the Classification report through sklearn*. URL: `https://muthu.co/understanding-the-classification-report-in-sklearn/`.

Srinivasa, Aishwarya V (n.d.). *Stochastic Gradient Descent — Clearly Explained !!* URL: `https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31`.

SYNCED, BY (n.d.). *Tree Boosting With XGBoost – Why Does XGBoost Win "Every" Machine Learning Competition?* URL: `https://syncedreview.com/2017/10/22/tree-boosting-with-xgboost-why-does-xgboost-win-every-machine-learning-competition/`.

T, Sam (n.d.). *Entropy: How Decision Trees Make Decisions*. URL: `https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8`.

Berry, Wallace (1976/1986). "Structural Functions in Music". In: Dover Publications, p. 349.