barcelona gse
graduate school of economics

14D010 - TextMining for Social Sciences

Final Project Report

Author:	Felix Gutmann
Student number:	125604
Program:	M.S. Data Science
E-Mail:	felix.gutmann@barcelonagse.eu

I Table of Contents

II List of Figures

III List of Tables

1 Introduction

Musical work has both a acoustic and lyrical component. This paper aims to evaluate the application possibilities of text data on identifying successful songs. A natural way of measuring success is by considering monetary success. However, precise information on sales data are not immediately available. A more modern indicator of success would be to consider streaming data (e.g. Spotify or YouTube). Gathering those might be feasible, but the time expense of this process might disproportional to the scope of the project.

Based on revenues some singles (or LPs) are awarded with golden or even platinum records. An advantage of those awards is that they a more time independent, because the awarding criteria got adjusted constantly over the years. Based on that the objective is to find out if the lyrics can be processed and used in such a way that they can be used to predict awards for a given song using several unsupervised learning algorithms. The underlying data for this projects consists of lyrics of the yearly final chart entries of the Billboard Hot-100 single chars and related awards for each song (see in more detail later on). The time frame considered in this project is from 2000 to 2015.

The report has the following structure. In section two the data gathering process is outlined. Additionally, this section involves an outline of the data pre-processing and a summary of the resulting final data set.

Section three is dedicated to the analysis. First there is a small introduction of the subsequently applied classifiers. This is especially dedicated to random forest classifier. Data in a text mining context are usually high dimensional. Hence, this section also involves a short discussion how features are selected to find a adequate model. To boost predictive power some additional feature extracted from the lyrics are also outlined. This section closes with a summary of prediction results. Finally the papers closes by summarizing results and some critical remarks.

2 Data; Collection, processing and summary

This section is dedicated to the data collection process. This involves also a short outline of related problems.

The data for this project come from three different websites. [Wikipedia, 2016] provides a yearly overview of the yearly Billboard Hot 100 closing single charts. It is listing the most successful singles of a given year (based revenues). Information on chart position, artist and song title got scrapped from this source.These information are used to get the remaining data.
For the lyrics [Songlyrics, 2016] serves as an appropriate data source. Signed up users can comment and enter lyrics. It is just one of plenty options, but the site has a nice and clean structure, which makes it fairly easy to scrape. Some lyrics failed to scrape, mostly due to artist spelling. No uniform pattern could be identified to automatize those exceptions, so they were ignored (more details are provided in the data summary later on).
Finally, The awards come from [RIAA, 2016]. They award golden, or platinum record as a benchmark for success. There a certain criteria, which have to be fulfilled to be awarded. Superficially spoken records have to generate a certain amount of revenue.[1]
The categories are ordinal, meaning when single is awarded with platinum it also got awarded with a gold record. In the following a song solely belongs to the highest achieved award. Furthermore, a single can get multiple platinum records. For example the latest Rihanna song "Work" has already three platinum awards. Those cases are also not considered in this project. A single can only belong to either one of the three categories; *"no award"*, *"gold"* or *"platinum"*. Finally table 1 summarizes the three data sources and related data

Website	**Content**
Wikipedia.org	Yearly charts artists and song titles
songlyrics.com	Lyrics for each scraped song
Riaa.com	Information about award

Table 1: Overview of data sources

For 15 years we should find 1500 songs. However, the sample got reduced due to

[1] There is a more detailed system how much for example digital units or records contribute to revenue or additional extras in collectors editions. A more detailed breakdown can be found on [RIAA, 2016].

several reasons. First, 33 singles and 83 lyrics could not be scrapped, mostly due to special artist spelling. A unique pattern could not be identified and hence a subsequent manual scrape was not performed.

Furthermore, one can observe that some songs are in Spanish. Those got removed from the data, because the would have to be processed individually. With four songs the number of removed songs was quite moderate.[2]

The remaining text got processed in the following standardized way. Non alpha numeric characters got filtered from the text. In the next step stop words got removed.[3]After browsing the resulting lyrics, further adjustments were necessary. For example a lot of lyrics contain small words like "oh" or "ah" with a high frequency. Obviously, they are not contributing to the analysis at all. Hence, all words with length less than three got also removed. Finally, the tokens for each document got stemmed with a porter stemmer.[4]. Based on this the term document matrix got computed.

Table 2 gives a summary of the final data set. We can observe that the class of platinum records is oversampled while gold records are under sampled. This makes sense since platinum records also include gold awards.

Indicator	**Value**	**Ratio**
Number of observations	1,380	-
Number of distinct artists	521	-
Number of unique stemmed terms in corpus	9.585	-
Number of distinct stemmed terms per document	86.55	-
Single records with no awards	499	36.16
Singles records with gold awards	208	15.07
Singles records with platinum awards	673	48.77

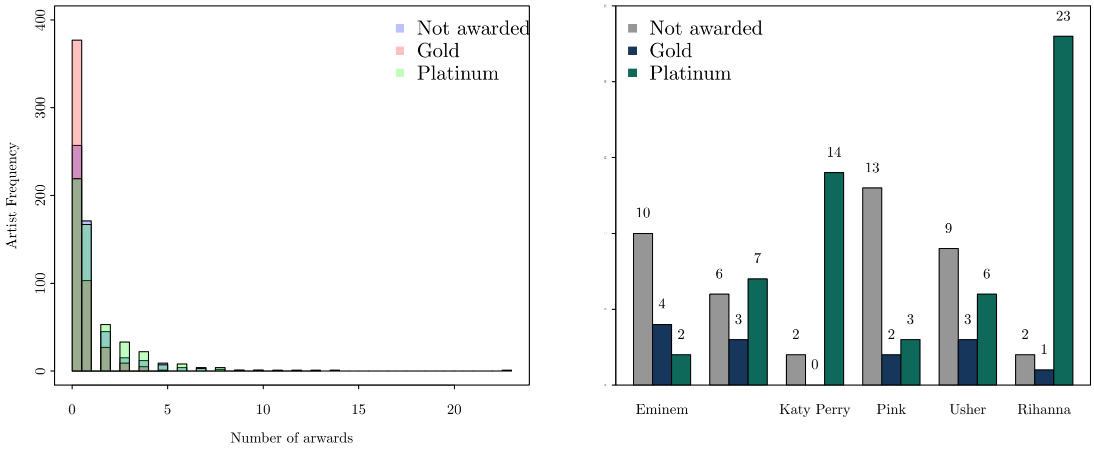
Table 2: Data Summary - Yearly Billboard Hot 100 (Cleaned, 2000 - 2015)

[2] The language got detect with langdetect library. A simple approach using Spanish stop words turned out to be not reliable enough

[3] The stop word list comes from the natural language toolkit library (nltk)

[4] This also come from nltk library

Figure 1 depicts the artist - award distribution. Sug-figure 1 (a) shows the number of awards and corresponding frequency for each of the three categories in the data set. We can observe that all three follow roughly a power law distribution. Sub figure (b) illustrates top 5 five artist in the tail based on the most frequent appearance. We see that those artist are rather really high or non awarded.



(a) Artist arward distribution

(b) Tail values for awards with artists

Figure 1: Award structure

3 Implementation and Analysis

This section contains the analysis for this project. Besides presenting results this involves a discussion of one additional supervised algorithms, feature engineering and selection. The scikit library provides a wide range of classifier and feature handling algorithms and is solely applied in the analysis..

3.1 Applied classifiers

In this analysis four different supervised learning algorithms were applied.

- Mutlinomial Naive Bayes
- K-NN
- Support Vector Classifier
- Random Forest

Since random forest are pretty successful, but not discussed in the lectures we might introduce them quickly. Several reasons might justify that choice. It is fairly easy to tune. Furthermore, it is assign importance to features and thus is expected performs quite well on big data sets.
The classifier is basically an extension of a tree classifier and got introduced by [Breiman, 2001].[5] A tree classifier partitions the feature space in different cells and assigns a label to each cell. New points falling in one cell are getting labeled with the class of the cell. Figure 2 illustrates a tree classification for the iris data set.

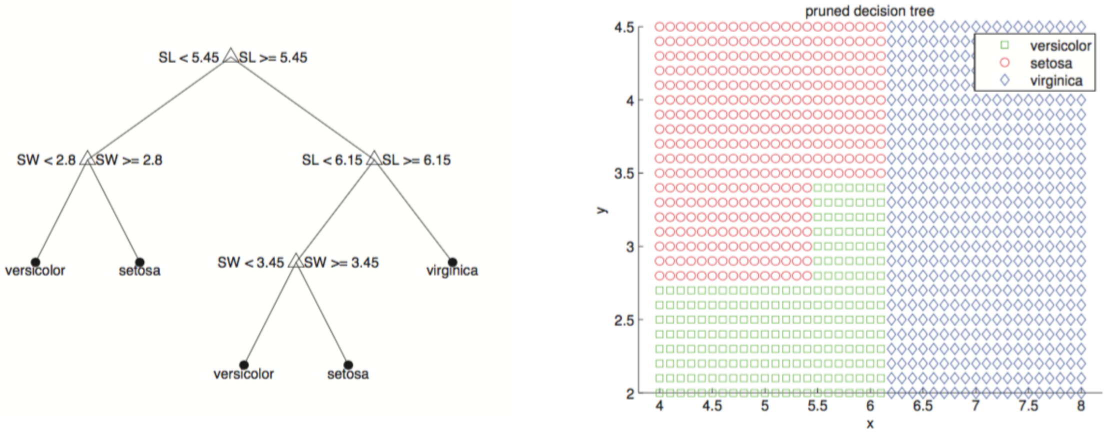


Figure 2: Example tree classifier on iris data set (source [Murphy, 2012, page 550])

The tree classifier has several problems. By dividing the feature space over and over

[5] This paper contains also a more formal discussion of the method. In the following we focus on the intuition

again the classifier has the problem of over-fitting the training data [Wasserman, 2004, page 360 et. seqq.]. Furthermore, they said to be unstable with respect to small changes in the input data. Hence in a cross validation the results might vary a lot and the estimator is said to have a high a variance. The variance is decreased by averaging over many tree-predictions on different subsets of the data. However, the cure for the variance has the downturn that predictions are correlated. So additionally each tree classifier is also trained on different subsets of the features to prevent correlation. By ensembling the individual trees the final model is obtained. This procedure characterize the idea of a random forest classifier [Murphy, 2012, page 550 et. seqq.].

3.2 Feature engineering and feature selection

The data set got enriched with some additional variables. A simple way is to add counts of unique terms and title length.[6] Additionally six additional features are created using an LDA model. The challenge is to choose the number of topics. The size of data sets allows to perform an optimization approach. Therefore, the model was run from one to twenty topics and the one with the highest likelihood got picked (which turned out to be six). Finally a simple sentiment analysis was performed. The AFINN-methods matches words from a list with them in the text and and assigns a sentiment count.[7]. Those features defines the full model for the analysis. Obviously the column space of the data set can be quite enormous. To increase performance on might apply some feature reduction techniques. [Divya and Kumar, 2015] gives a broad overview on several possible techniques. Among others, the paper proposes the chi square statistic to measure association with a feature from the document term matrix and the category [ibidem, page 16]. It is implemented in the scikit library and therefore picked as the applied methods. This will be used to study subsequently the prediction results by reducing the column space sequentially (more details on that are outlined in the next section). The results of the classification are discussed next.

[6] It can be discussed if that adding any more necessary information to the model. It is more a heuristic and got observed in many different sources.

[7] The python package description can be found here

3.3 Analysis results

All following displayed results are computed with 5-fold cross validation. It is convenient to define base line classifiers to test relative performance of applied methods. Within this analysis two such base line classifier are considered. The first benchmark are draws from a uniform distribution. The second one is weighted modification. For each step in the cross validation the relative frequency of the labels in the temporary training set are computed. Based on those empirical probabilities of the training labels, predicted labels got sampled from a discrete distribution. Finally the third base line classifier is the 1-NN classifier. The average accuracy of those methods are defined as the benchmark for the subsequent applied classifiers. The following table **??** gives an overview of the results for the benchmark predictions.

Method	**Cross validation score**
Uniform Sampling	0.335
Weighted Sampling	0.365
1 - NN	0.372

Table 3: Benchmark Classifier (5-folds)

The following table 4 shows the classification results for each classifier. The results are computed as follows. In each round the top *k* features of term document are computed using the chi-square statistic. Then the additional features like LDA and sentiment values etc. are added to that data frame.[8] For each *k* the average accuracy was computed for each classifier. This procedure was executed for k varying from 100 to 9585 terms in steps of 500. From the table we can observe that all classifier are able to beat the benchmarks at some point. The random forest performed the best with an average accuracy of 0.471 on 4000 terms. Furthermore, the feature reduction increased results slightly.

[8] In the analysis Multinomial Naive Bayes was applied, so it was only performed using the features from the term document matrix.

	100	...	1500	2000	2500	3000	3500	4000	4500	5000	...	8500	...
MN - Naive Bayes [0]	0.377	...	0.394	0.406	0.406	0.417	0.41	0.411	0.417	0.421	...	**0.434**	...
K-NN [1]	0.368	...	0.387	0.389	0.389	0.389	0.387	0.393	**0.394**	0.393	...	0.389	...
SVM [2]	0.39	...	**0.442**	0.427	0.427	0.425	0.428	0.426	0.424	0.429	...	0.432	...
Random Forest [3]	0.394	...	0.461	0.463	0.453	0.459	0.453	**0.471**	0.454	0.452	...	0.455	...

[0]

Table 4: 5-fold cross validation Accuracy scores for different sizes of feature matrix

We saw that the random forest performed slightly better than the other models. We have a more closer look on those results. Using the setting obtained from the feature extraction study the following table shows the confusion matrix of a random forest classifier with 4000 terms plus additional features from LDA etc. Training on random subset of 80 % and testing on remaining set achieved a accuracy of 0.6. From the confusion matrix we can observe that class three gets picked up quite well. There are problems of especially in picking up class 1, but also in 2. Given the class imbalance this is somehow expected.

	None	Gold	Platinum
None	26	0	52
Gold	10	3	33
Platinum	14	1	136

Table 5: Confusion matrix for random forest prediction on 4000 features

3.4 Conclusion and review of results

The objective of this project was to study the prediction performance using solely text related features of song lyrics. The Corpus consists of the songs of the final yearly Billboard Hot 100 single charts from 2000 to 2015. The idea was to use gold and platinum awards as a time independent proxy of success. Additional features like LDA topics got computed from the data. Chi-Square score is used to reduce the feature space.

We saw that all classifiers we able to beat the baselines at some point. Random forest achieved the highest results among the used classifiers. However, there is some room for critical remarks on the results. Due to the class imbalance a lot of mistakes were made predicting gold records and un-awarded records. Hence, results are not very reliable.

One might considering balancing classes. This can be quite an extensive procedure and so it remains to future work. Furthermore, it is questionable if using the yearly final chart entries are the best data choice, since they are all quite successful. Looking backwards it might be more interesting to scrape all songs from weekly charts to have a more realistic data set, where awards are more an exotic phenomena. Then the class balancing problem becomes even more important. However with some more effort the text data seemed to be at least a good extension for classification.

4 List of Literature

[Breiman, 2001] Breiman, L. (2001). Random Forests. *Machine learning*, 45.1:5–32.

[Divya and Kumar, 2015] Divya, P. and Kumar, G. S. N. (2015). Study on Feature Selection Methods for Text Mining. 2(1):11–19.

[Murphy, 2012] Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

[RIAA, 2016] RIAA (2016). Certification criteia. http://www.riaa.com.

[Songlyrics, 2016] Songlyrics (2016). Songlyrics.com. http://www.songlyrics.com.

[Wasserman, 2004] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*, volume 1542.

[Wikipedia, 2016] Wikipedia (2016). Billboard year end hot 100 singles of 2015.