1 Background and objectives for the Group Assignment

In this assignment, we had the opportunity to use newly acquired abilities, within both the Applied Data Science and Machine Learning module, the Network Analysis & NLP module and the Deep Learning module, to investigate a given problem statement. Through the M3 course, we have applied deep learning techniques for business analytics, and acquired proficiency in new methods autonomously. Based on this, we will try to carry out an analysis which contains elements of data manipulation, exploration and deep learning.

2 Problem statement

Use Deep Learning techniques to investigate whether we can make a deep learning (Neural Network) model predict the sentiment of a song given the lyrics of this song. In addition, we would like to investigate whether it is also possible to predict this using a Multiclass model given these lyrics.

3 Data acquisition

Given the problem statement we choose and obtain datasets which we consider interesting and appropriate for this analysis. Through Kaggle we were able to find a relevant dataset. The dataset is separated into two parts, and combined these contains 13 columnsfeatures, which describes different characteristics for selected songs eg. genre and lyrics.

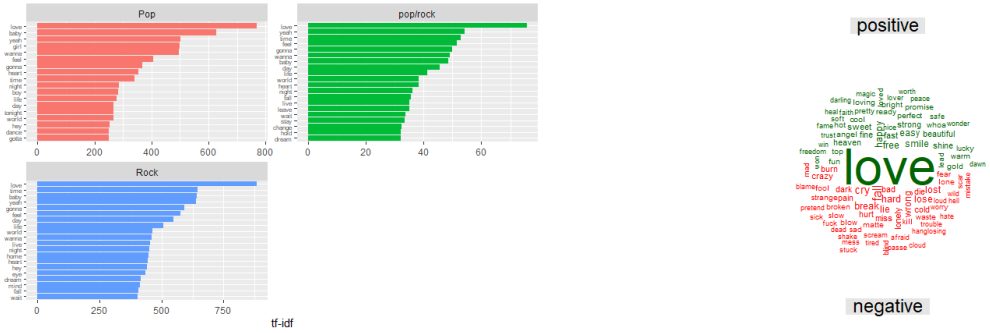
Based on these, we describe the variables that are used the most in relation to our problem statement. This includes three character strings. The first variable **name** assign each song with the respective artist and name of the song. The second variable **text** contains a text that refers to the lyrics of each song. The third variable **genre** describes which genre or several genres a given song belongs to. In order to make the analysis more applicable only the three most common genres (**Pop**, **Rock** and **Pop/Rock**) are used. In addition, we have used an external dataset from Spotify, which contains the column **release_year** of these songs.

4 Results

4.1 EDA sentiment

In the EDA session two subjects has been vizualised, first the sentiment based on words within the genres of Pop, Rock and Pop/Rock. Secondly also the sentiment based on songs written by our favorit artists has been analyzed.

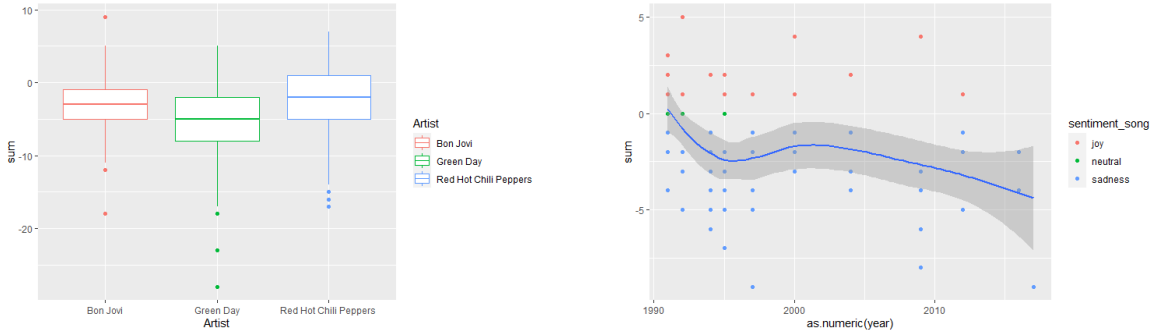
At first we take a look at the 20 most used words within the 3 genres in figure 1. Before the words are plotted, more common words as "the", "us"and "in"are removed using stopwords. Also stemming has been used. It can be seen that "Love"is the most used word within all three genres and in general they have many words in common. Now the sentiment of these words will be shown, useing a wordcloud as in figure 1. Here the top 50 words is seen within the pop/rock genre, where green words are positive, while red words are negative.



(a) Top 20 Words score **(b)** wordcloud pop rock names

Figur 1: Sentiment genre

The second part of the analisys is regarding the sentiment used by the artists Bon Jovi, Green Day and Red Hot Chili Peppers. where figure 2 shows a boxplot for the three artists based on the sentiment of there songs. It can be seen that RHCP on average has the most positive songs. Yes still all of the artists has on average more negative songs.



(a) Sentiment songs by artists score

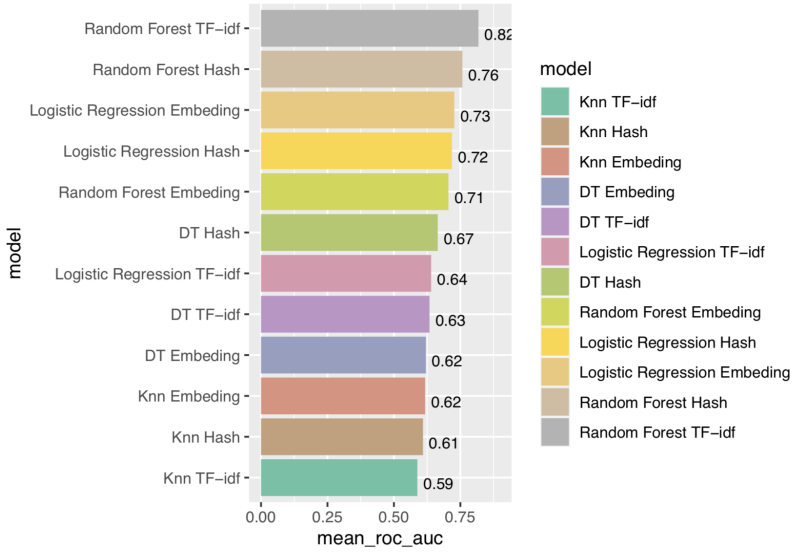
(b) Sentiment development of Green Day names

Figur 2: Sentiment songs

Lastly the sentiment used in Green Day songs will be vizualised over time, looking at figure 2 the development of Green Days songs are plotted over time, based on the sentiments "joy"and "sadness". It can bee seen that over time, Green Day's songs has become more sad compared to joyfull. :(

4.2 Supervised machine learning

The goal for the supervised machine learning model is first to be able to predict the sentiment of the song (positive or negative) based on its lyrics (binary model prediction). After that we also make a multiclass model prediction, predicting which sentiment a song belongs to between: "trust", "sadness", "joy", "fear". These two models are used as non-neural baseline models to be able to compare with neural networks which were made later on. In this stakeholder only the

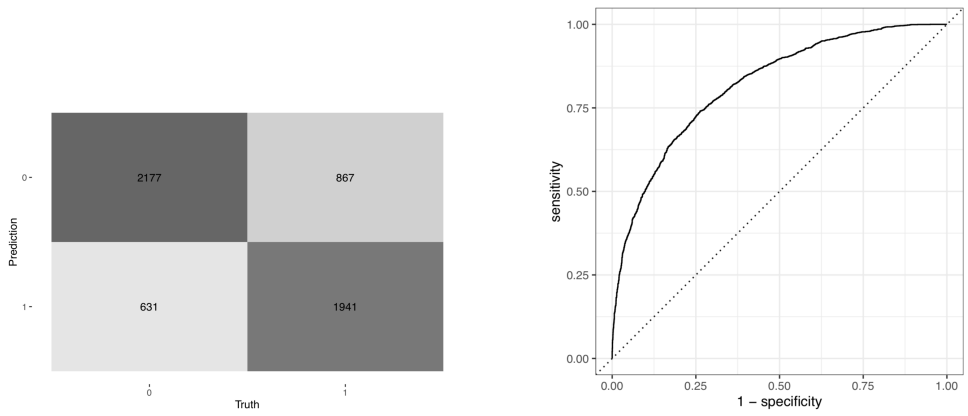


Figur 3: Model comparison

binary model will be commented on.

First the dataset is split into training and test data, then the training data is downsampled to get an equal representation of each sentiment. The data is preprocesed using 3 different methods: tf-idf, word-embedding and text-hash. Next four models are defined including a Logistic model, K-Nearest neighboor model, a Random forest model and lastly a decision tree model. After hypertuning, the four models are fitted using resampled data, and different measures are collected for each of the 12 models. They are plotted for comparison in figure 3 where the models are ranked by area under the curve.

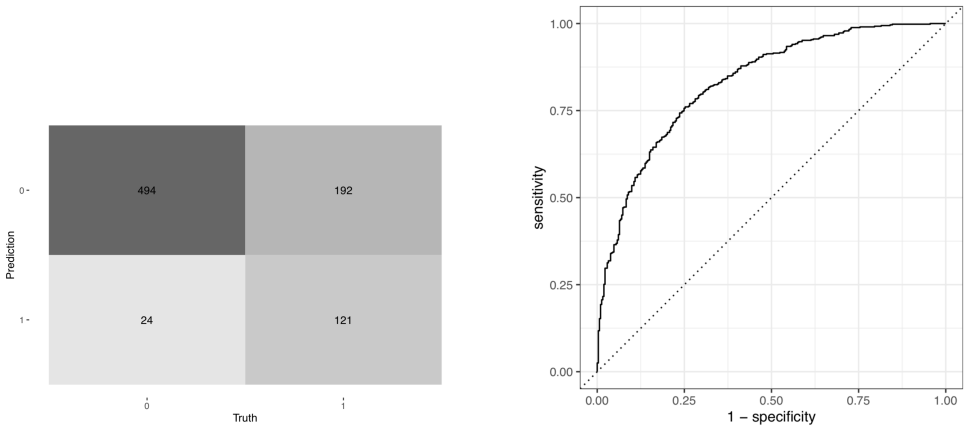
After looking at other measures the Random forest model TF-IDF and random forest hash embedding model performs the best in the binary model prediction, so the confusion matrix and roc-curve is made for these models shown in figure 4. The more observations in the diagonal the better for the confusion matrix, which means more true predictions. The binary classification model seems to do fine in predicting the sentiments, with the best model having an ROC-AUC value of 0.82, showing the False-positive fraction agains the True-positive fraction. The accurucy is 0.731 on the training data and 0.741 for the testing data. These results will be used to compare with the neural networks.



(a) Confusion matrix score **(b)** Area under the curve

Figur 4: Model Comparison plot

Now the two models are used on the test data, if the models are not over fitted we should get almost the same results.



(a) Confusion matrix on test data score

(b) Roc-auc on test data

Figur 5: Model Comparison test data plot

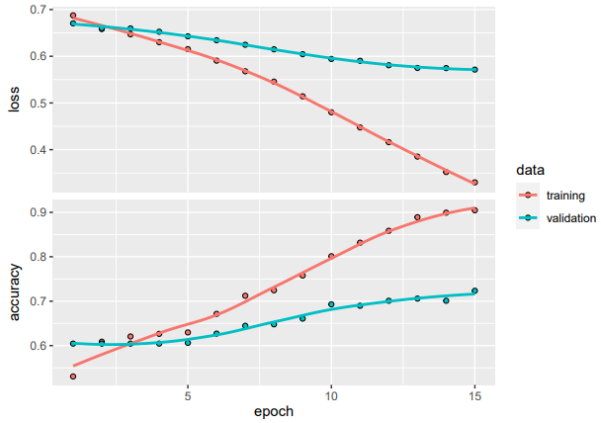
The testing data is not downsampled like the training data. Still looking at figure 5 the model is great at predicting joy and does an okay job at predicting trust, fear and sadness.

5 Neural Network

In addition to the above results, both binary and multiclass Neural networks have been created to make a comparative analysis of the ML models results. In this stakeholder only the binary model will be commented on.

The analysis includes a ANN (artificial neural network), RNN (recurrent neural network) model and a LSTM model. Here ANN has been run through two dense "relu"layers, where the last dense layer is the output layer, which is of unit 1 and is a "sigmoid"layer (returns a value between 0 and 1) which is the most optimal for a binary model. Then the model is compiled with a optimzation function and loss function and maximized for accuracy. The baseline model is run 10 times with a batch size of 256, and from the plot of this we observe that the training set does a lot better than the validation set, which is an indicator for the model being over-fitted. As a result the baseline model is then tuned where we introduce some dropout layers and reduce the weights of each layer in order to minimize the number of parameters in the model to prevent the overfitting. The tuned baseline model is run 10 times with a batch size of 512. The tuned baseline model seems to do better as we observe that the loss function both of the validation data and the

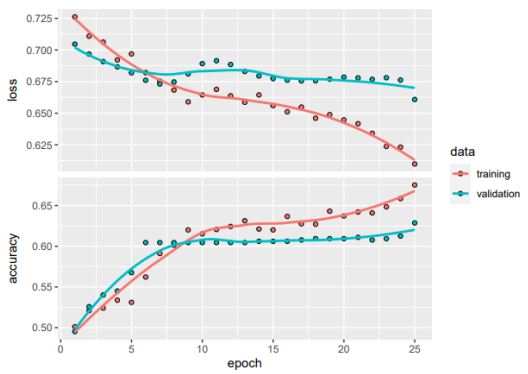
training data continues to go down after every epoch. The accuracy also does a better job with accuracy on our traning data on 90% and validation accurucy on 73%, however on the test data it is 63% so still not optimal.



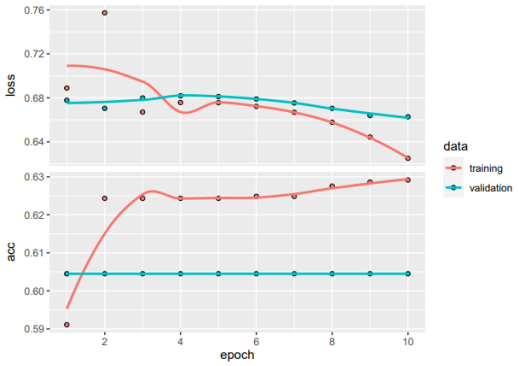
Figur 6: Validation and traning loss and accuracy of tuned ANN

The RNN model and LSTM with paded data take the word-order (sequence) into account. The RNN model has been run through a layer_embedding to compress our initial one-hotencoding vector of length 5000 to a lower dimensionality of 32. The next layer is a layer_simple_rnn, and the last dense layer is the output layer for the binary sentiment prediction. Again the model is compiled using a basic setup for binary prediction. The RNN too seems over-fitted, and therefore tuned. This time we again try to reduce the number of parameters, make another RNN layer and add a drop_out layer for the input. The tuned RNN model seems to do better, however we observe some fluctuations after a couple of epochs. The accuracy is better than the non-tuned RNN model, however an accuracy of 55% on the test_data is still not optimal and worse than on the ANN model. The same goes for the traning accuracy which after 10 epochs only hit 67%.

The LSTM model has initially also been runned through a layer_embedding and then through a special lstm layer, which is able to use dropout layers for both the state weights and the input weights. Again we use a base compile settings to compile the model. Both the traning and validation accuracy is steady around 60 to 63%, and running on our test data yields a accuracy of 63%, which is the best of any neural network model, but worse than the best machine learning model.



(a) Validation and traning loss and accuracy of tuned RNN



(b) Validation and traning loss and accuracy of tuned LSTM

Figur 7: RNN and LSTM output