HitHub — A Hit List Predictor

M. C. Demir B. Geyik Y. Keten

Abstract

Given the competition in the music industry, which is one of the largest industries in the world, money, time and effort are enormous and success is paramount. Producing hits that everyone will love is often the primary goal. There are features that ensure whether a song is a hit or not. Examples of these features are danceability, energy, loudness, mode, acousticness, instrumentalness and tempo. With the help of these features, we can understand why people love music. We made predictions with Deep Learning and KNN, SVM, Logistic Regression, and Naive Bayes algorithms using 13 features in the data set of Spotify Web API and music data from different year intervals, and we obtained certain results.

1. Introduction

The music industry is one of the largest industries in the world. Every year companies and musicians make billions of dollars in total from this industry. For this reason, there is a high level of competition in the industry. Every company, every musician strives to be more successful than others and to gain a foothold in the industry. To do this, they spend time and effort and take risks and try to come up with hits. The taste of music varies from person to person. For example, some people like slower songs, while some people like slower songs. For this reason, it is not easy to get a song to be a hit. Musicians try various ways to minimize risks and to put music that people will love to listen to. There is no compensation for the time they spend for this. In addition, a lot of time and effort is spent and a song produced with the expectation that it will be a hit causes a great loss of prestige and disappointment. Here, we are taking approaches to eliminate this problem. People may or may not like a piece of music. We can put forward the reasons for this in a concrete way. Each song has features such as danceability, energy, loudness, mode, acousticness, instrumentalness, tempo, etc. These characteristics are the factors that determine whether people will like a music or not. Using these factors, we can analyze which features have effects on the hit songs and decide whether a song is a hit or not with the results of these analyzes. Using advanced machine learning and deep learning algorithms, we tried to determine in advance whether a song can be a hit when it is released. We used the dataset and features provided by the Spotify Web API. While making predictions, we also worked with basic machine learning algorithms such as KNN, SVM, Logistic Regression, and Naive Bayes as well as Deep Learning.

2. Related Work

There are lots of studies that predict success before a song is released. Most of them used classical machine learning algorithms such as K-Nearest Neighbor, Naive Bayes, etc. The studies that use internal factors have low accuracy than other studies that use external factors like an advertisement. (Salganik et al., 2006)

In 2016, a study conducted by Pham et al., at Stanford University, used different machine learning algorithms using both audio features and metadata to predict the popularity of songs. They used a subset of The Million Song Dataset for classification. The results of the research which includes Neural Networks and some machine learning algorithms had similar accuracy and more accurate than the previous studies with values ranging from 0.70 to 0.85. (Pham, 2015)

Some studies used Spotify API to get audio features and tried to predict the popularity of a song. For example, Minna and Philippa used different machine learning algorithms for prediction and the accuracy of their model is similar to the accuracy of our current model. (Reiman, 2018)

3. The Approach

3.1. Dataset and Features

In the first stage of our project, a dataset uploaded to Kaggle is used. We have taken the data between 2010-2019 for now. This dataset is a combination of features extracted from music by Spotify. This approach is chosen as the features are already there without further process. The features provided by Spotify are acousticness, danceability, duration, energy, instrumentalness, key, liveness, loudness, mode, speechiness, tempo, time signature, valence. Below you can see the descriptions of these features.

– Acousticness is the value that measures whether a track is acoustic. This value is minimum 0.0, maximum 1.0. The closer the value is to 1.0, the more likely the track is to be

an acoustic track.
– Danceability is the feature that describes how suitable a track is for dancing, how much that tranch arouses the desire to dance in the listener. It contains a combination of different musical elements. Examples of these elements are beat strength, rhythm stability, tempo and general regularity. The danceability value also ranges from 0.0 to 1.0, and the closer the value is to 1.0, the more danceable the track is. Closer to 0.0, the fitness to dance decreases.
– Duration is the length of the track's duration, in milliseconds.
– Energy indicates how fast, loud and noisy the track is. It specifies measures of the activity and intensity of this track and sets out the appropriate assessments. The energy value ranges between 0.0 and 1.0, and as the value approaches 1.0, the track can be classified as a track with higher energy. As it gets closer to 0.0, its energy decreases.
– Instrumentalness is the feature that shows the intensity and effect of the sounds coming from the musical instruments in the track. The Instrumentalness value ranges between 0.0 and 1.0. The closer the value to 1.0, the louder the song's instrumental sounds, and the closer the value to 1.0 the more likely it is that the track consists of only instrument sounds. Values of 0.5 and above usually represent instrumental tracks.
– Key is the key that the track is in and mapping is done using Pitch Class notation.
– Liveness indicates whether the track is a recording from a live performance or not. It does this by detecting outside sounds in the recording. The liveness value ranges between 0.0 and 1.0. The closer the value is to 1.0, the more likely the track is to be a live performance recording, and above 0.8 provides a strong probability of that.
– Loudness is the measure of the loudness and power of the sound. Indicates the quality of the sound. Loudness values usually range from -60 decibels to 0 decibels. – Mode indicates the modality of the track, ie major or minor. If the mode value is 1, the modality of the track is major, if the value is 0, the modality is minor. It also specifies the scale type and the melodic content derived from it.
– Speechiness is a feature that shows how long words and sentences are in a track and how much these words are in the foreground compared to the music. It does this by detecting the words in the track. The Speechiness value ranges between 0.0 and 1.0. The closer the value is to 0, the more prominent speech and words are over music and rhythm. By approximate comparison, tracks with values above 0.66 are generally purely speech tracks such as poetry, audiobooks, podcasts. Tracks with values between 0.33 and 0.66 are generally tracks like rap music where words and speech are prominent, but have a music and rhythm behind them. Tracks with a speechiness value below 0.33 are usually normal music tracks. The closer to 0, the closer the tracks get to the nonverbal tracks.
– Tempo is the feature that indicates the number of beats per minute of the track in BPM. The higher the number of hits, the higher the tempo value.
– Time signature is a feature that shows how many beats are in each bar in a certain time interval.
– Valence is the feature that shows whether the effect and feelings of the track are positive feelings or negative feelings. Valence ranges between 0.0 and 1.0. As the value approaches 1.0, positive feelings such as happiness and joy become prominent on the track, while as it approaches 0.0, the effects of negative and depressive feelings such as sadness and anger increase. In the dataset, we have also a feature "Target", which indicates that whether the music is hit. '1' means that the song has been featured in Billboards Hot-100 list at least once, so the song is hit. '0' indicates that the track has not been featured in Billboards Hot-100 list, so the song is not hit. (Ansari, 2020) (Spotify, 2021)

3.2. Algorithms

To predict whether the song is going to be successful, the following algorithms are used: Logistic Regression, Gaussian Naive Bayes, Support Vector Machine, KNN (n=5); and our custom Artificial Neural Network (ANN), with an input layer contains 13 neurons (which are features), with a hidden layer contains 16 neurons with Swish activation function and an output layer contains 1 neuron with Sigmoid activation function. In addition, the ANN model is set to be trained with batch gradient descent with 100 epochs with a learning rate of 0.1.

3.2.1. Naive Bayes

Naive Bayes Classifier is a classification algorithm under the title of Supervised Learning. It is a simplified version of Bayes' Theorem by providing the independence condition between the properties. Naive Bayes Classifier processes the data and calculates probability and obtains rates for each case separately. It performs classification with these probabilities. Since the features are considered independent, it performs better than models such as Logistic Regression, but the lack of independence in natural conditions is a disadvantage for Naive Bayes. It is easy to apply and works fast. So it can be used in real-time systems. It is also widely used in recommendation systems, category classifications, and text classifications.

3.2.2. Support Vector Machine

Support Vector Machine is an algorithm under the title of Supervised Learning. It is often used in classification problems. The algorithm aims to separate the data belonging to the two classes most appropriately and correctly. While doing this tries to provide the maximum distance between

the points and the dividing line while separating the data on a plane. This distance is called margin and maximum margin provides the best classification.

If there is data in the margin part, that margin is called soft margin. If there is no data in the margin part, it is called hard margin. The size of the margin is controlled by the 'C' value and as 'C' increases the margin decreases. 'C' needs to be reduced in case of overfitting.

Some datasets cannot be classified correctly in 2 dimensions. We use Kernel Trick to perform operations and calculations in 3 dimensions. Kernel Trick is divided into Polynomial Kernel and Gaussian RBF Kernel.

3.2.3. K-Nearest Neighbor

K-Nearest Neighbor is also an algorithm under the title of Supervised Learning, like the algorithms above, and is generally used in classification problems. It is a non-Parametric and lazy learning algorithm.

The K-Nearest Neighbor algorithm uses the values of the previous data to measure the distance of the newly added data to all values and classifies according to the class of the nearest "k" data elements. There are different distance functions such as Euclidean Distance, Manhattan Distance, and Minkowski Distance when calculating these distances.

The K-Nearest Neighbor algorithm is often used because it is a familiar and simple and understandable algorithm. But when calculating distances, since distances are calculated for all data one by one, many results are kept and take up a lot of memory.

3.2.4. Artificial Neural Network

The artificial neural network is a learning algorithm created by utilizing the structure and learning mechanism of the human brain. By imitating the learning mechanism of the brain and the behavior of neurons, they provide the function of learning and discovery without any assistance. It is the mathematical modeling of the brain's learning function.

There is unsupervised learning in artificial neural networks. They can produce consequences for unseen outputs. They can work in parallel and process real-time information. They also have fault tolerance. The basic operation is to find the weight parameter (w) and bias value (b) from which we can get the best score from the model. The input value in a cell and the weight value in the cell are multiplied and transmitted to the next cell, and weighted addition is performed with the values from different cells. Afterward, it is added with the bias value and transferred to the activation function and then to the output. There are various activation functions such as Sigmoid and ReLU.

Artificial neural networks consist of 3 layers: input layer, intermediate or hidden layers, and output layer. After the information is transmitted to the input layer, it is transmitted to the hidden layers. The information processed in the intermediate layers is transmitted to the output layer and converted to output. Artificial neural networks are divided into 2 according to the number of hidden layers: Single Layer Neural Network and Multilayer Neural Network. Artificial neural networks have many uses. Examples are classification, prediction, data processing, data filtering, and diagnostics.

4. Experimental Results

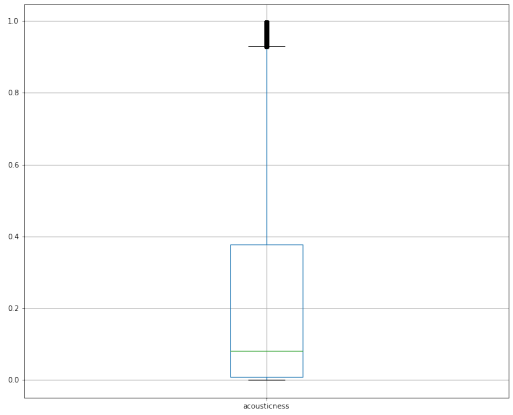

Figure 1. Acousticness

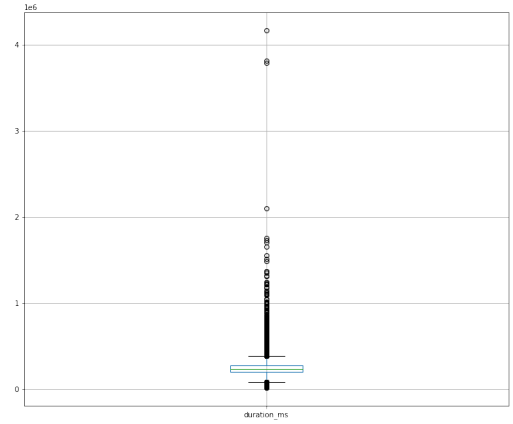

Figure 2. Duration (ms)

First of all, we found it necessary to scale the data we have in some way because we have no way of knowing that all the data we have are in similar intervals. For example, the ”danceability” feature might be positioned between 1 and 10, while the ”mode” might be between 0-1.

To overcome this problem, we thought we should use some scaling methods and we came up with three options. MinMaxScaler, StandardScaler, and RobustScaler, which are found in the library ”scikit-learn”. MinMaxScaler finds a value by subtracting the smallest data from the largest data of that data set and divides each data by this value and pulls the dataset between 0-1. StandardScaler, on the other hand, makes the data set homogeneous by setting the mean value to 0 and the standard deviation to 1, thus eliminating the importance of the gradient direction when trying to reach the global minimum in algorithms such as gradient descent. While contours without scaling appear squished, contours become uniform with StandardScaler. RobustScaler performs a scaling using the quartiles and median. The advantage of RobustScaler is that unlike MinMaxScaler and StandardScaler, it is not affected by outlier values. Therefore, we have to check if the features in our dataset have outlier values.

As you can see with Figure 1 and 2, features have a lot of outlier values. In order not to fill the report, we will not include all the charts, but as we can see, 7 of the 13 features have at least one outlier value. In this case, the best option is to scale the data with RobustScaler.

Next, we need to look at the relationships of features with some methods and which features we should eliminate, which will make our model work correctly.

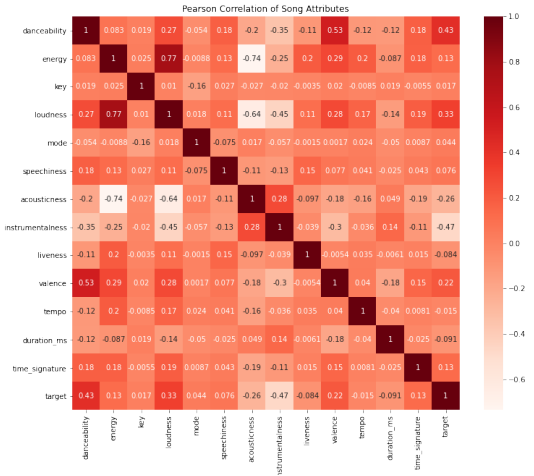

Figure 3. Pearson Correlation of Song Attributes

In Figure 3, the correlation between features can be seen. The three most positive correlated feature pair are ”energy-loudness = 0.77”, ”valence-danceability = 0.53” and ”valence-energy = 0.29”. The three most negative correlated feature pair are ”acousticness-energy = -0.74”, ”acousticness-loudness = -0.65” and ”instrumentalness-loudness = -0.53” Although the correlations between features are here, the most important pair of the feature set is ”target” versus all, as we want to eliminate unrelated features from the dataset. Thus, we choose some features that are close to zero, so the features selected are not going to be even negatively correlated with ”target”. As you can see, the three most ”unrelated” features for the ”target” are:

- key
- mode
- tempo

Moreover, the unrelatedness is stronger than others.

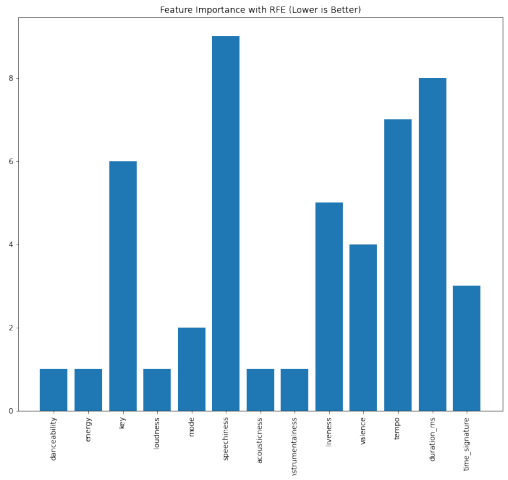

Figure 4. Feature Importance with RFE (Lower is Better)

Recursive Feature Elimination (RFE) is a feature selection algorithm and it is an algorithm for finding the most relevant features. It filters and scores features through a machine learning algorithm. As a result, the features with the lowest score are the best features.

Here we tried to use the Recursive Feature Elimination algorithm with Logistic Regression. As seen in 4, the most unnecessary features are as follows:

- speechiness
- duration_ms

- tempo
- key
- liveness

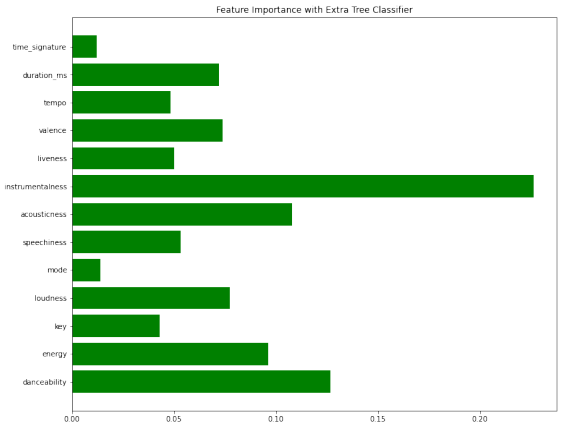

Figure 5. Feature Importance with Extra Tree Classifier

Extra Tree Classifier is a classification algorithm very similar to Random Forest algorithm. In short, it makes feature selection with multiple Decision Trees. As a result, the features with the highest score are the best features.

As seen in 5, the most unnecessary features are as follows:

- time_signature
- mode
- key
- tempo
- liveness
- speechiness

As a result of these processes, we decided to remove mode, key and tempo features from our feature list. The reason is that most of these feature elimination methods are strongly observing that these features are unnecessary.

Data is divided by 80% training and 20% development set for machine learning approaches, with 10% development and 10% predict for artificial neural network approach, as we don't actually use development set in machine learning models as we used in our artificial neural network model. We have tried to change various parameters such as scaling, the neighbor parameter for KNN, and something like these. As a result for feature elimination techniques, the dataset used on the project contains 14232 training examples with 10 features and 3558 development examples (divided by half for Artificial Neural Network).

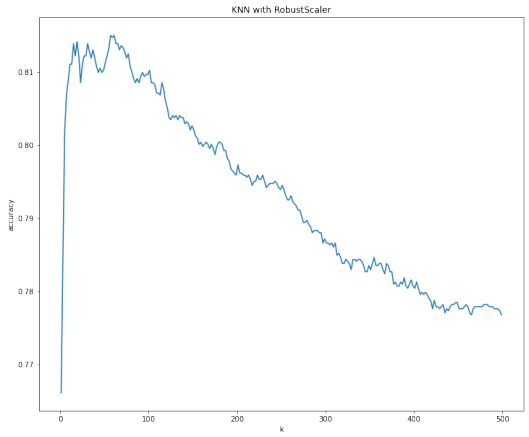

Figure 6. KNN, graph

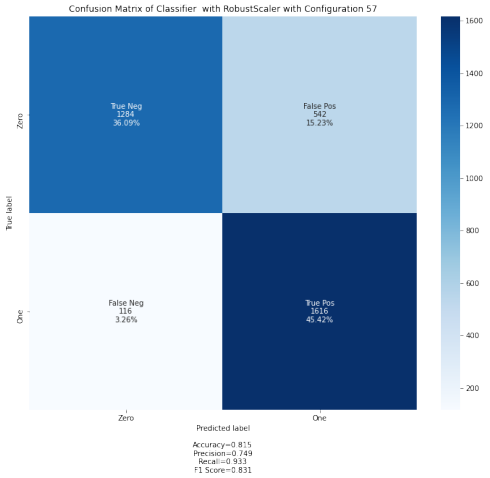

Figure 7. KNN, confusion matrix of best

In Figure 6 and 7,k values from 1 to 500 were tested for the K-Nearest Neighbor (KNN) algorithm and k=57 was found to be the parameter that gave the most accurate result. As can be seen from Figure 6, as the k value increased, accuracy decreased significantly after a while. The best accuracy value for KNN found was 81.5%.

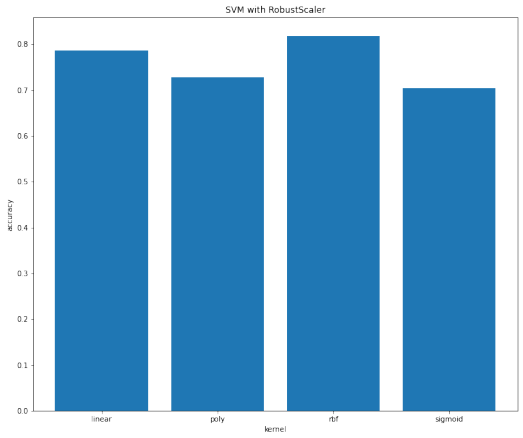

Figure 8. SVM, graph

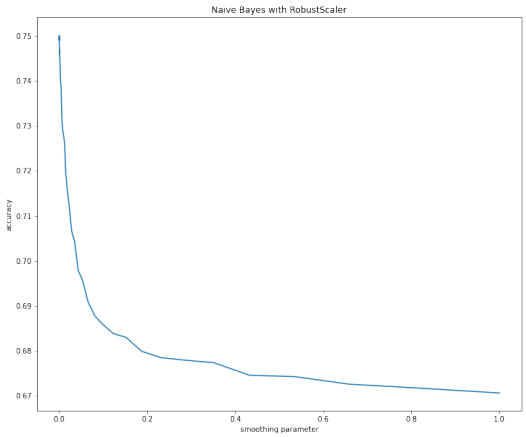

Figure 10. Gaussian Naive Bayes, graph

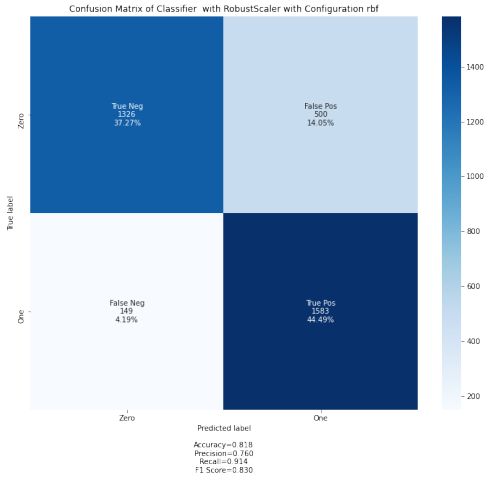

Figure 9. SVM, confusion matrix of best

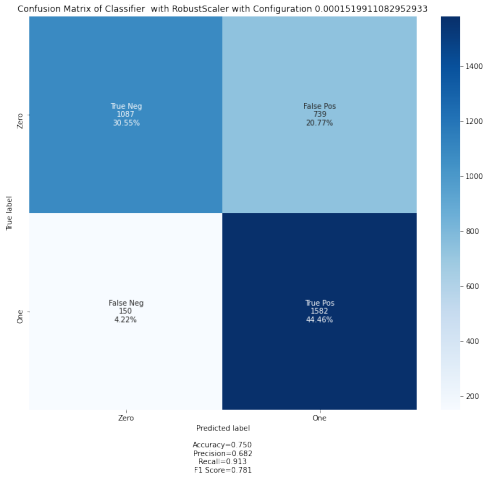

Figure 11. Gaussian Naive Bayes, confusion matrix of best

As can be seen with in Figure 8 and 9 , 4 different kernels have been tried for SVM, which are ”linear”, ”poly”, ”rbf” and ”sigmoid”, it seems that we got the best result with the ”rbf” kernel at 81.8%.

In Figure10 and 11, it can be seen that more than one smoothing parameter has been tried for Gaussian Naive Bayes. Again, as our smoothing parameter increases, the predictive ability of our model decreases significantly. The most accurate value for the Smoothing parameter was 0.000151 and the accuracy of our model was observed as 75%.

To talk about our Artificial Neural Network (ANN) model, 48 different combinations were tried by creating more than one hyperparameter change and combination. These are as follows:

- For activation functions ”lisht” and ”swish” are used.
- For optimizer ”adam” is used.
- For optimizer ”adam” is used.
- For layer sizes [16], [128], [128, 128] are used.
- Different batch sizes and early stop callbacks are used.

The best is following:

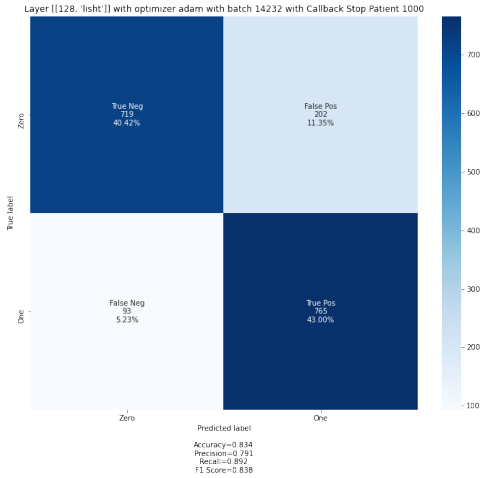

Figure 12. ANN, confusion matrix of best

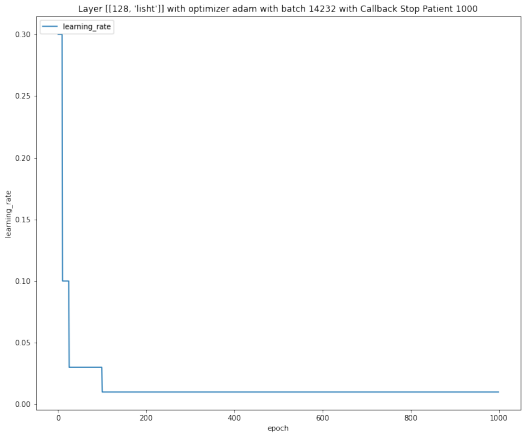

Figure 13. ANN, learning rate of best

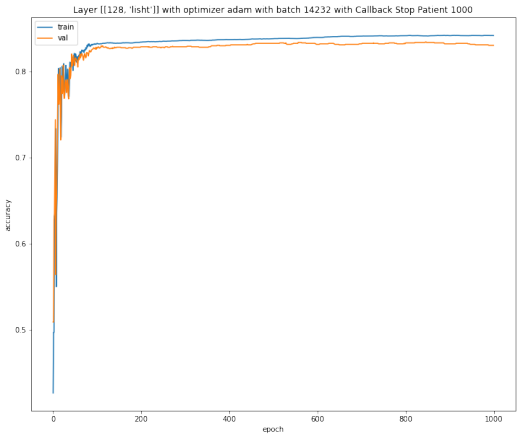

Figure 14. ANN, loss of best

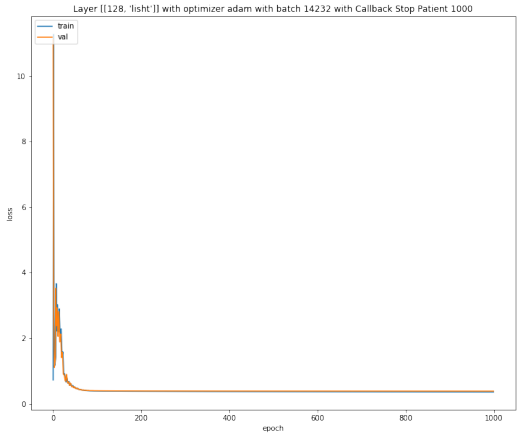

Figure 15. ANN, accuracy of best

5. Conclusions

We aimed to predict whether a song can be a hit when it is released. The accuracy of our model is similar to previous studies which use classical machine learning algorithms. We tested most of the machine learning algorithms, and also Neural Networks. The best accuracy that we got in this study is nearly 84 percent. We may improve our model with Siamese Networks. However, this accuracy is still enough for prediction in the industry. As a result, machine learning algorithms can solve our problem with enough accuracy.

References

Ansari, F. The spotify hit predictor dataset (1960-2019), 2020. URL https://www.kaggle.com/theoverman/the-spotify-hit-predictor-dataset.

Pham, J. Q. Predicting song popularity. 2015. URL https://api.semanticscholar.org/CorpusID:15802233.

Reiman, M., . P. Predicting hit songs with machine learning. 2018. URL http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-229705.

Salganik, M. J., Dodds, P. S., and Watts, D. J. Experimental study of inequality and unpredictability in an artificial cultural market. *Science*, 311(5762):854–856, 2006. ISSN 0036-8075. doi: 10.1126/science.1121066. URL https://science.sciencemag.org/content/311/5762/854.

Spotify. Web API reference, 2021. URL https://developer.spotify.com/documentation/web-api/reference/#endpoint-get-audio-features.