Song genre classification based on song attributes

Jovan Đorđić SW46-2018
Jelena Miletić SW45-2018

1. Motivation

Modern music streaming services offer users personalized recommendations based on the user's preferences in regards to various attributes of the song. In order for the song recommendation algorithm to give good predictions, each song has a myriad of tags attached to it. Inspired by the amount and specificity of these tags attached to songs on the music streaming platform Spotify, we decided to explore the correlation between various song attributes and more traditional music genres.

2. Research questions

Our project's main aim was to create a ML model that would be able to predict a song's genre based on its other attributes/tags, such as:

1. The song title
2. Release year
3. BPM (beats per minute)
4. Energy
5. Danceability
6. Loudness
7. Liveness
8. Valence
9. Duration
10. Acousticness
11. Speechiness
12. Popularity

These tags were attached to every song in the dataset we used as the basis for our project. In addition to them, each song contained an "Artist" tag, but we decided not to include it in the group of tags used for prediction due to the strong correlation that exists between artists and genres.
The dataset we used is publicly available on *kaggle* and is called *Spotify - All Time Top 2000s Mega Dataset*. It contains the top 2000 songs available on Spotify of all time, in CSV format.

3. Related work

Before implementing our own algorithm, we explored related work. In particular, we took a look at some implementations done using the same dataset we used. Compared to the approach we took, other solutions usually relied on the Naive Bayes algorithm. In addition, they often pre-processed data in a different way compared to our solution. More specifically, steps like choosing the columns utilized as the basis for the predictions & the grouping of similar genres together were often done with different priorities in mind.

4. Methodology

Looking at the data at hand, we decided to first visualize some of it. We looked at the distribution of each attribute, and noticed that some of the columns were skewed (figure 1.). The columns which were skewed were “liveness”, “accousticness” and “speechiness”. In order to improve our prediction they were removed from the dataset.

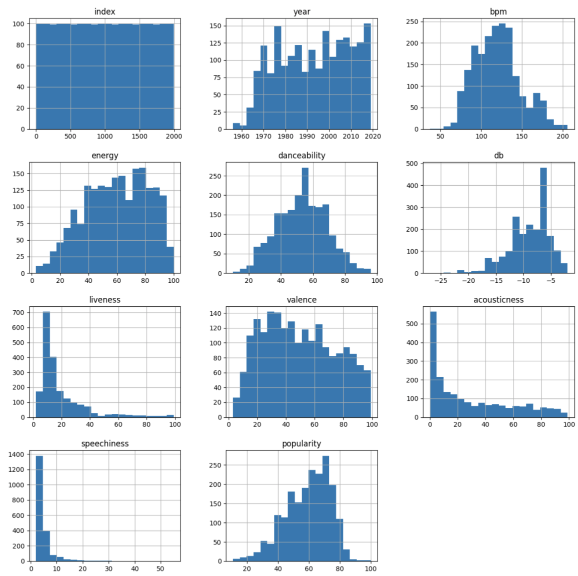

Figure 1. Dataset features distribution

After removing these columns, we decided to remove three other columns which we didn’t have a need for in the dataset. These were “Index”, “Artist” and “Title”. We removed the “Artist” attribute because the same artist usually has songs which fit in the same genre.
To begin with there were 158 different genres in the dataset, many of which could fit under the same genre. For example “british rock” and “garage rock” could be put under the same genre “rock”. After we did this, we were left with 68 genres, of which we took the top 20 with the most data. Taking the ones with the most data did not improve our classification result, so we decided on working with all 68 genres.
When picking which classification method we were going to use we decided on using ensemble methods because we decided that combining a few of them is better than just using one. The three classification methods we used were SVM, Logistic Regression and Random Forest classifier.
For SVM and Logistic Regression we used OneVersusRest heuristics because we were working with multi-label classification problems and these two previously stated methods were only capable of working with binary problems. The OneVersusRest heuristic splits multi-label classification problem into one binary problem per class. Scaling of the data was also carried out before putting data into the SVM classifier. Another thing which was done was PCA, or dimensionality reduction of the data.

5. Discussion

Hyperparameters for SVM were optimized using a grid search method, but at the end the only parameter that made any significant improvements to the overall accuracy was regularization parameter C . The value of the parameter that was optimal was “0.01” according to the grid search method. Random forest classifier and Logistic Regression were optimized empirically by us during our testing of the classification.

Since the dataset wasn't that big, we decided on splitting the data 70/30 on the training and test data. We decided on using accuracy score as the metric since the data was quite balanced. Accuracy measures how many positive or negative observations were made.

Accuracy scores for Ensemble methods on a 70/30 split dataset were between 0.41 and 0.48.

6. References

https://www.kaggle.com/datasets/iamsumat/spotify-top-2000s-mega-dataset
https://machinelearningmastery.com/why-use-ensemble-learning/
https://machinelearningmastery.com/one-vs-rest-and-one-vs-one-for-multi-class-classification/
https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html
https://scikit-learn.org/stable/modules/generated/sklearn.multiclass.OneVsRestClassifier.html?highlight=one%20vs%20rest#sklearn.multiclass.OneVsRestClassifier
https://neptune.ai/blog/f1-score-accuracy-roc-auc-pr-auc