AI Genre Classification by Lyrics

Ariel Amar Hanan Benzion Elad Gershon Ruth Taboada

Abstract— **Text mining is often associated with the process of natural language processing (NLP) methods to formulate a general conclusion about a body of text. In this work we applied this concept in attempt to build models that classify a song's genre based only on its lyrics. Such models must be able to analyze the words that compose a song's lyrics and categorize the song as one of four genres: Hip-Hop, Pop, Rock and Metal. The analyzing of the different features is compared with different machine learning classifiers: neural network, decision trees, random forest and multinomial naive base. It also discusses the significance of data collection and pre-processing in order to ensure valid results particularly when dealing with string values.**

I. INTRODUCTION

Autonomic Music classification is an important and well researched task in the growing research field of music information retrieval (MIR). Music genre classification, especially by lyrics alone is still a challenging task in the MIR field. Genre classification using lyrics presents itself as a natural language processing (NLP) problem. In NLP the aim is to assign meaning and labels to text; here this equates to a genre classification of the lyrical text. Traditional approaches in text classification have utilized n-gram models and algorithms such as Support Vector Machines (SVMs), k-Nearest Neighbour (k-NN), and Naive Bayes (NB). In recent years the use of deep learning and Artificial Intelligence methods (like neural networks) has produced superior results and represent an exciting breakthrough in NLP. In this project we used data of more then 60,000 different songs from the genres: Hip-Hop, Metal, Pop and Rock. We extracted the information from a song's lyrics and identify features that help music genre classification. Our analysis of lyrics relies mainly on natural language processing (NLP) techniques Based on data mining. In our work we also compared different classifiers, including neural networks, decision trees, random forest and multinomial naive Bayes. We will analyze in this paper the conclusions of using the different features for genre songs classification, and the efficiency of the different classifiers for the task.

II. METHODS AND MATERIAL

A. Raw data

We used a Kaggle's dataset which had more than 240,000 different songs with a lot of different genres. After cleaning the data (deleting songs with no lyrics, deleting songs in other languages, etc) we created a new dataset by reducing the dataset's genres into four: Pop, Hip-hop, Rock, and Metal, and by choosing 10,000 random songs from each genre to work with. For the test set, we took randomly from the dataset 100 songs from each genre.

The Hebrew University of Jeruslem

B. Preparation of the data

Using string tools and regexes we were able filter and adjust the words so we could create a reliable bag of words for each song. At last we created the bag-of-words which is translation of each vector of each song so we could retranslate it faster. This all process take time so to short it up we saved the bag-of-words into a pkl file, so we could load it up and reuse the same bag-of-words.

C. Heuristics

We have developed few heuristics to help us classify songs to genres:

- **Bag of words** - Lastly using once more NLP tools to create bag of words which is simplifying representation of all the words in the song as vector to help us know seek it in the vectorizer we built.
- **Length of the song** - Average length of songs including return of choruses. As we see rock and metal are more long then hip-hop and pop.
- **Frequency of words** - Average frequency of the words. As we see pop usually reuse and resay a lot of words again.
- **Ratio of verbs, nouns and adjectives** - Using NLP (Natural language processing) tools we could know which option is each word, that told us hip-hop use more adjectives and pop more verbs
- **Number of unique words** - How many unique words are in each song is great addition to frequency of words and told us the ratio of
- **Positive and negative words** - Using NLP tools we could know if the word has a positive or negative or neutral connotations. Negative songs were mostly hip-hop and then metal and positive songs were more likely to be pop.

D. AI Classifiers

We compared and analyze our work using the following classifiers:

- **Decision Tree:** We built a decision tree using the library sklearn. We implemented a decision tree with Comparison between Gini Impurity and Information Gain criterion and with pruning. Decision trees are popular methods of building classification models because of their ease of interpretation and clarity, but may not always be the best option depending on the type of data and attribute set. In this work, the dataset contains

50,000 attributes, so a decision tree generated would be so large it would be impractical to try and scan it for suspicious nodes or meaningless splits, though its results may look plausible. Nonetheless, it is interesting to use this method regardless and examine how its results differ from those of other methods that try to account for these weaknesses.

- **Random Forest:** We ran the Random Forests algorithm using the Random Forest from the sklearn library. This method generates a number of decision trees at random and uses them in conjunction with each other when testing the model. Using this model typically generates a higher accuracy than a single decision tree because it allows for more specific splits, though may lead to a higher chance of over-fitting data because of the higher number of splits. The initial run of the Random Forest algorithm was done generating 10 random trees, then 100 random trees and even 1000 random trees.In the next section a comparison of the results from these runs is given.
- **Multinomial Naive-Bayes:** We also used the Naive Bayes algorithm which is perhaps the most popular classification model used in text mining applications. It is fundamentally grounded in Bayes’ Theorem, as follows:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where $P(A|B)$ is the probability of an event A, given the event B. This theorem provides an excellent relationship between prior and posterior probabilities, and therefore makes for an effective technique in classification. This theorem can be extended in the following way. Suppose a vector $x = (x_1, ..., x_n)$ represents n attributes, and some event C is to be predicted. According to Bayes’ Theorem:

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}$$

This probability is most likely a more manageable value to compute. The Naive Bayes classification model uses this approach to classify instances. The term ’Naive’ reflects that the algorithm assumes all attributes are statistically independent. In the example above, this implies that all x_i) in the vector $(x_1, ..., x_n)$ are uncorrelated with each other.

- **Neural network (Multilayer Perceptron):** We used the sklearn Multi-layer Perceptron classifier. Using the ’SGD’ solver gave poor results where using the ’ADAM’ solver that is known for working pretty well on relatively large datasets (with thousands of training samples or more) in terms of both training time and validation score. We compared networks with different amount of neurons in each hidden layer - 2, 8 and 100. While 2 neurons resulted in underfitting and 100 neurons overfitting, 8 neurons gave us the best results. alternating the number of layers in the network didn’t affect the results. In our work the activation function ’relu’ gave poor results compare to ’tanh’ activation function.

E. Related existing projects

Automatic classification of music is an important and well-researched task in Music Information Retrieval (MIR) and there are several works about classifying songs with the lyrics information. In one such work[1], the use of lyrics is provided in addition to the traditional methods to classify songs based on categories of moods (X. Hu, J. Downie, A. Ehmann). This work exhibits similar pre-processing techniques, including the removal of non-lyric text, but makes use of the lyrics in different ways. For example, it explains that function words (called stop-words in this work) actually exhibit predictive power in terms of text style analysis, and are used as an independent feature set, though yield worse results than other methods used. In another work[2], lyrics are used to find underlying emotional meanings in songs (D. Yang, W. Lee). This work uses 23 emotion categories such as power/strength/dominance vs. weak, active vs. passive, understatement vs. exaggeration, etc, in which songs are grouped into. Therefore, the approach used to generate a feature set is an adaptation of Part-of-Speech; rather than grouping words based on their grammatical function, they are grouped based on a pre-defined, arguably subjective, emotional function. Nonetheless, this work achieved an accuracy of 67% using an ensemble method in Weka. Another work[3] (Anthony Canicatti) using decision tree, random forest and similar data got accuracy between 30% to 52%.

III. RESULTS

First, for model selection we used the cross validation method and saw that MLP classifier and Random Forest were the best among all classifiers with approximately 65% on the validation set. Decision tree was the worst with a 53.2% average success rate on the validation set.

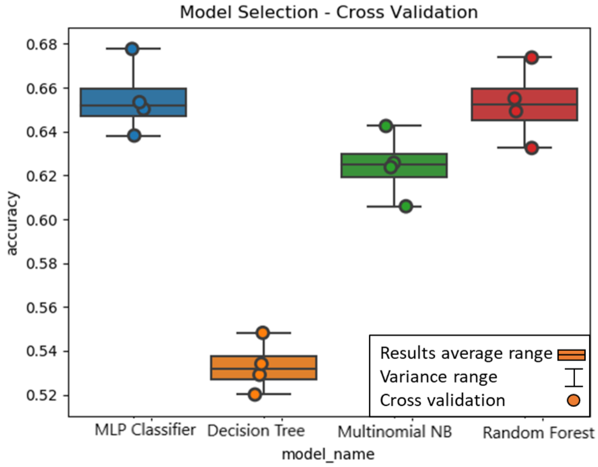
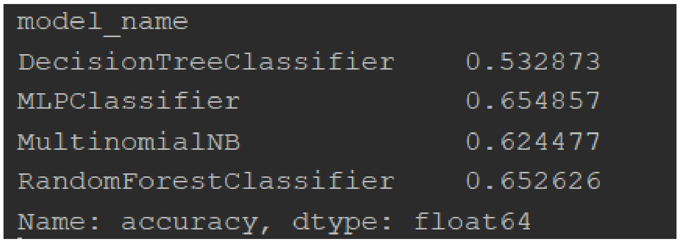


Fig. 1: Cross validation results for all classifiers.

In order to test each model, we chose the best classifier of each model from the cross validation stage and applied it on the test set. The following graph compares the classifiers’ accuracy on the test set:



```
model_name
DecisionTreeClassifier     0.532873
MLPClassifier              0.654857
MultinomialNB              0.624477
RandomForestClassifier     0.652626
Name: accuracy, dtype: float64
```

Fig. 2: Cross validation average results.

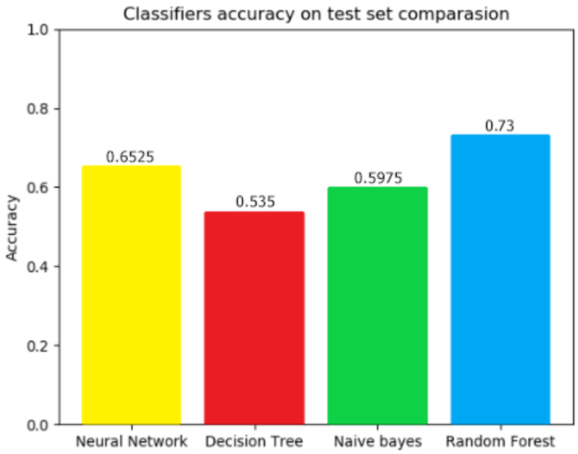


Fig. 3: Classifiers results on the test set.

In order to qualify each classifier, a main parameter we have taken into account, is the fitting percentage that is calculated by taking the difference between the best classifier's accuracy over the validation set and the accuracy over the test set. We can see the results in the following table:

Classifier	Validation's best	Test	Fitting
Neural Network	67.3%	65.25%	1.55%
Decision Tree	58.51%	53.5%	5.01%
Naive Bayes	63.48%	59.75%	3.73%
Random forest	70.18%	73%	-2.82%

Fig. 4: Classifiers' accuracy and fitting percentage

As mentioned above, the test set is made of 25% of each genre. We wanted to check if the classifiers preserves the distribution of the test set in their predictions. When a classifier predicts a specific genre more than others it means that the classifier overfits to the features common in that genre. In the pie charts below we can see how the predictions of all classifiers distribute:

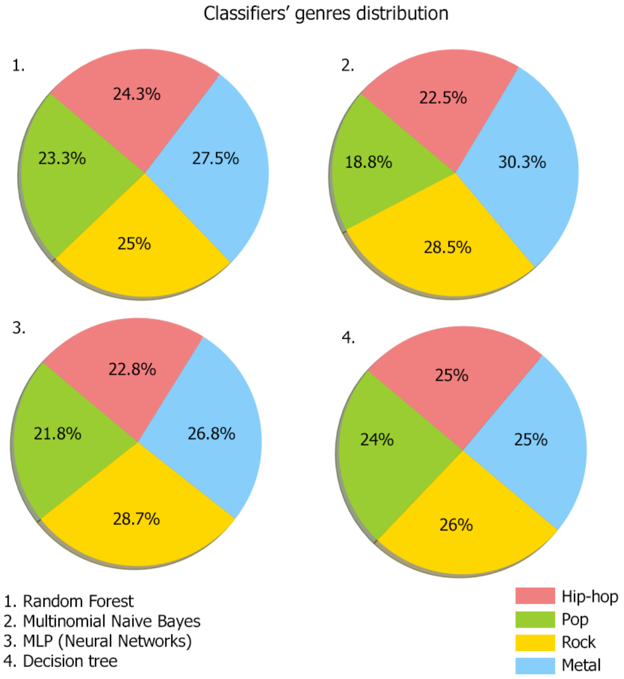


Fig. 5: Genres distribution of each classifier.

Random forest classifier has the option to choose the number of decision trees to use in the forest. We compared the accuracy on the test set of random forest with 10, 100, 1000 decision trees in order to check how the amount of trees affects the accurecy results.

Number of trees	Validation's best	Test set
10	65%	63%
100	69.2%	69.2%
1000	69.9%	65%

Fig. 6: Accuracy As function of number of trees in random forest.

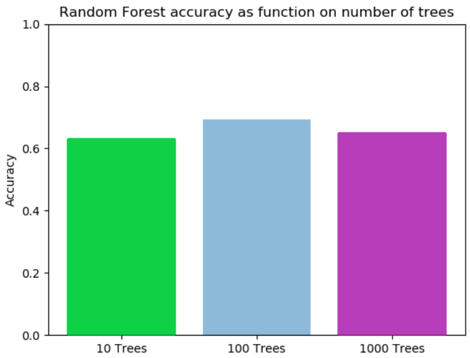


Fig. 7: Accuracy As function of number of trees in random forest.

IV. CONCLUSIONS

We compared the classifiers' accuracy on the test set and we got that the Random forest's classifier was the best one among all classifiers. We expected the Random Forest classifier to be successful given that it chooses the best features from many decision trees decreasing the overfitting. However we were surprised to see it outperforms all the other classifiers. As a text classification problem, based on previous courses and online articles, we expected Naive Bayes to be the most successful classifier since it is a popular method for text categorization. The MLP (Neural Network) classifier was also among the top ones, which was not surprising since Neural Networks are known by their ability to find the best features in data to learn from. At last, we got that the Decision tree performed the worst among all. We got an accuracy rate of approximately 53% but it is not as disappointing as it seems. A classifier which makes random predictions has an expected accuracy of 25% so that shows us that the decision tree did learn from the data but as learned in class, using a decision tree on a very large data, leads to overfitting.

Fig. 4 shows the fitting percentage of each classifier. We can see that Neural Network, Decision Tree and Naive Bayes overfitted the data, and Random forest underfitted the data. As we suspected the model which overfitted the most was decision tree. The most reliable model is Neural Network because it had the lowest fitting percentage. Surprisingly, random forest underfitted the data even though we thought it would overfit the data due to the fact it is based on decision trees.t

The pie charts we represented in the results section show that most of the classifiers preserve the distribution of the test set, but one classifier - multinomial naive bayes, gave interesting results. We saw the the naive bayes classifier predicted Rock and Metal more than the other genres. Naive bayes found features that strongly represent Rock and Metal, we can conclude from that, that there might be a representative features for each genre.

As seen in Fig. 6 and Fig. 7, we compared the accuracy of different number of decision trees in the random forest. We can see that taking more trees improves the validation's accuracy, however taking too many trees decreases the test set accuracy due to overfitting.

V. FUTURE WORK

- We would like to classify more than 4 genres.
- Adding more heuristics that uses NLP methods to classify the text better.
- We would want to add more machine learning algorithms such as Support Vector Machines (SVMs), k-Nearest Neighbour (k-NN), k- means.

REFERENCES

[1] Xiao Hu, J. Stephen Downie, and Andreas F. Ehmann Lyric Text Mining in Music Mood Classification. American music, 2009.

[2] Dan Yang and Won-Sook Lee Music Emotion Identification from Lyrics. Multimedia, 2009. ISM '09. 11th IEEE International Symposium on, San Diego, CA, 2009.

[3] Anthony Canicatti, see Song Genre Classification via Lyric Text Mining. Computer and Information Science Dept., Fordham University, Bronx, NY, USA.