SOUND AND MEDIA TECHNOLOGY PROJECT REPORT

Shpat Cheliku
Utrecht University
s.cheliku@students.uu.nl
6549128

Emir Zeylan
Utrecht University
a.e.zeylan@students.uu.nl
6036791

1. INTRODUCTION

1.1 Research topic

Our research project is about music classification based on feature extraction and musical patterns. More precisely, the goal is to be able to use Computing Science and Artificial Intelligence techniques to be able to differentiate between real songs and computer generated songs.

Our research was inspired by the deep learning model named *Folk-RNN* [1] that is used to model and generate folk music. *Folk-RNN* is trained on over 23.000 traditional folk tunes retrieved from *The Session*, an online community of folk enthusiasts contributing transcriptions of tunes in ABC format.

1.2 Research questions

Our task is to compare real tunes retrieved from *The Session* and computer generated tunes from the *Folk-RNN* model.

- Use features to discover differences and commonalities between original songs from *The Session* and the computer-generated songs with *Folk-RNN* and use those features to do classification.
- Which features can tell us about the "artificialness" of the generated tunes?

2. METHOD

2.1 Base network model

The *Folk-RNN* model is build of 3 LSTM layers with 512 units. A softmax output layer is given over the vocabulary conditioned on the one-hot encoded input. Each model is trained by backpropagation with one-hot encoded vectors, a mini-batch approach of 50, and with drop out of 0.5.

2.2 Dataset

Our dataset consisted of 400 songs, 200 of which were real songs and 200 were generated songs. The real songs were downloaded from *The Session*, an online community dedicated to traditional Irish music. The rest of the songs were generated using *Folk-RNN*, more precisely from the website *https://folkrnn.org/*, a website that lets us generate music using *Folk-RNN*.

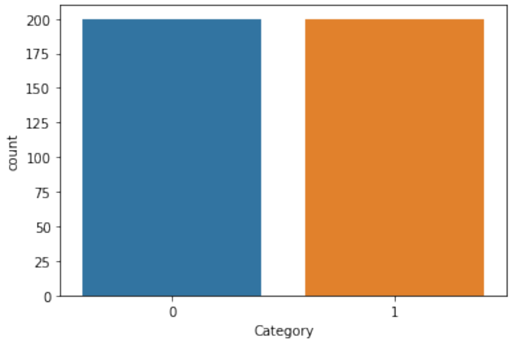


Figure 1. Dataset of MIDI files

2.3 jSymbolic

In order to extract features from both real and generated songs we used an MIR tool called *jSymbolic*.

jSymbolic is a software application intended for conducting research in the fields of music information retrieval (MIR), musicology and music theory. Its primary purpose is to extract statistical information from musical data stored symbolically in file formats such as MIDI or MEI. This statistical information is formulated as feature values, which may be fed directly into automatic classification systems, may be used to query large musical datasets, or may be used by musicologists and music theorists for conducting empirical musical research.

In total there were 1495 features extracted using *jSymbolic* for each song. Later in preprocessing steps the four dynamic features were removed as they deemed unnecessary for this particular analysis.

2.4 Jupyter Notebook / Google Colaboratory

Our exploratory data analysis was performed using Python/Jupyter Notebook in a Google Colaboratory environment.

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical sim-

Attribution: Shpat Cheliku, Emir Zeylan. “Sound and Media Technology Project Report”, 20th International Society for Music Information Retrieval Conference, Delft, The Netherlands, 2019.

ulation, statistical modeling, data visualization, machine learning, and much more.

Google Colaboratory is a Google research project created to help disseminate machine learning education and research. It's a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud.

The specifications of Google Colaboratory:

- GPU: 1xTesla K80, compute 3.7, having 2496 CUDA cores, 12GB GDDR5 VRAM
- CPU: 1xsingle core hyper threaded Xeon Processors @2.3Ghz i.e(1 core, 2 threads)
- RAM: 12.6 GB Available
- Disk: 320 GB Available

2.5 Libraries

Several Python libraries were used to perform the data analysis:

- Pandas
- NumPy
- Matplotlib
- SciPy
- Scikit-learn

2.6 Preprocessing

After feature extraction using jSymbolic we ended up with 1495 features for each individual song. The preprocessing steps we took included:

- Remove features with $StandardDeviation = 0$
- Remove dynamic features
- Scale the data using MinMax Scaler

The reason for dropping features with **Standard Deviation = 0** was that all songs had the exact same value for those particular features and that makes them irrelevant for with respect to our research questions.

The dynamic features were also irrelevant for this project because they did not contain information that helps differentiate real songs from generated ones.

One of the reasons we chose MinMaxScaler is that it preserves the shape of the original distribution and it does not meaningfully change the information embedded in the original data. The default range for the feature returned by MinMaxScaler is 0 to 1. The other reason is that Machine Learning models tend to perform better with scaled data as input.

After all the preprocessing steps we ended up with a total of 679 features.

2.7 Feature importance

We performed several feature importance techniques in order to get as much insight as possible from different perspectives and with the hopes of more techniques confirming the results of one another.

The techniques we performed for feature selection were:

- Extra Trees Classifier for Feature Selection
- Feature selection based on Pearson Correlation
- Univariate Feature Selection

2.7.1 Extra Trees Classifier

Extra Trees Classifier is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a "forest" to output its classification result.

To perform feature selection using Extra Trees Classifier, during the construction of the forest, for each feature, the normalized total reduction in the mathematical criteria used in the decision of feature of split is computed. This value is called the Gini Importance of the feature. To perform feature selection, each feature is ordered in descending order according to the Gini Importance of each feature and the user selects the top k features according to our choice. We selected the Top 15 features this feature selection technique recommended.

2.7.2 Pearson Correlation Feature Selection

Another technique to perform feature selection is by filtering and using only the subset of relevant features. The filtering in this case is performed using a correlation matrix and this is most commonly performed using Pearson correlation.

First we plot the Pearson correlation heat map and see the correlation of independent variables with the output variable Category (Real or Generated). The correlation coefficient has values between -1 to 1:

- A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
- A value closer to 1 implies stronger positive correlation
- A value closer to -1 implies stronger negative correlation

We will only select features which have correlation of above 0.5 (taking absolute value) with the output variable.

2.7.3 Univariate Feature Selection

Univariate feature selection works by selecting the best features based on univariate statistical tests. It works by comparing each feature to the target variable (Category), to see whether there is any statistically significant relationship between them. It is also called analysis of variance (ANOVA). When it analyzes the relationship between one feature and the target variable, it ignores the other features.

That is why it is called ‘univariate’. Each feature has its test score.

We use the chi-squared statistical test on the scaled data set containing no negative values. Finally, all the test scores are compared, and the Top 10 features with the highest scores will be selected.

2.8 Detecting the common features

The following thought process was established for detecting commonalities between real and generated songs:

- Again use the whole data set of 400 songs (real+generated)
- Remove all features with $StandardDeviation = 0$ because they are irrelevant to both finding commonalities or differences.
- Select a subset of features with very small standard deviation between 0.1 and 0.2.
- The resulting very small subset of features represents commonalities between real and generated songs.

2.9 Artificialness of the generated tunes

In order to determine features that are indicators of song "artificialness" we performed the following steps:

- We calculated the means of all relevant features for both Real and Generated tunes individually.
- Then we calculated the differences between the means of the real songs dataset and the generated songs dataset. We did this for all relevant features.
- difference(feature) = generated(mean) - real(mean)
- We took the features with difference > 0.5 as features indicating artificialness of tunes.
- Reason: Folk-RNN tune generator focused more on those features than a real person does when creating real tunes.
- We didn’t take the absolute values of the differences because that would also have included features with big enough mean difference but that indicate the "realness" of the real songs.

3. RESULTS

3.1 Feature importance

3.1.1 *Extra Trees Classifier Feature Selection*

Top 15 features according to Extra Trees Classifier technique:

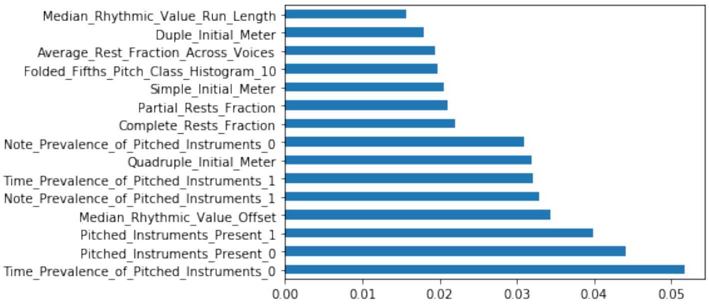


Figure 2. Top 15 Features - Extra Trees Classifier

Top 3 Features:

- **Time_Prevalence_of_Pitched_Instruments_0.** A feature vector indicating the fraction of time during which (pitched) notes are being sounded by each of the 128 General MIDI Instrument patches (0 is Acoustic Piano, 40 is Violin, etc.). Has one entry for each of these instruments, and the value of each is set to the total time in seconds in a piece during which at least one note is being sounded with the corresponding MIDI patch, divided by the total length of the piece in seconds.
- **Pitched_Instruments_Present_0.** A feature vector indicating which pitched instruments are present. Has one entry for each of the 128 General MIDI Instrument patches (0 is Acoustic Piano, 40 is Violin, etc.). Each value is set to 1 if at least one note is played using the corresponding patch, or to 0 if that patch is never used.
- **Pitched_Instruments_Present_1.** A feature vector indicating which pitched instruments are present. Has one entry for each of the 128 General MIDI Instrument patches (0 is Acoustic Piano, 40 is Violin, etc.). Each value is set to 1 if at least one note is played using the corresponding patch, or to 0 if that patch is never used.

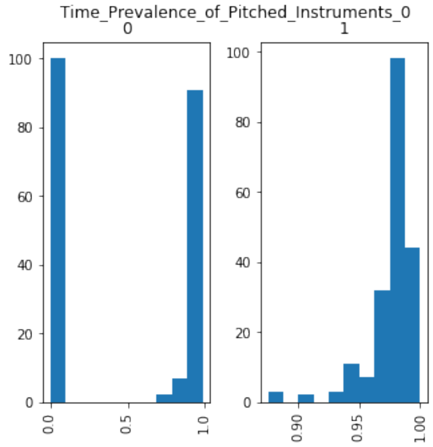
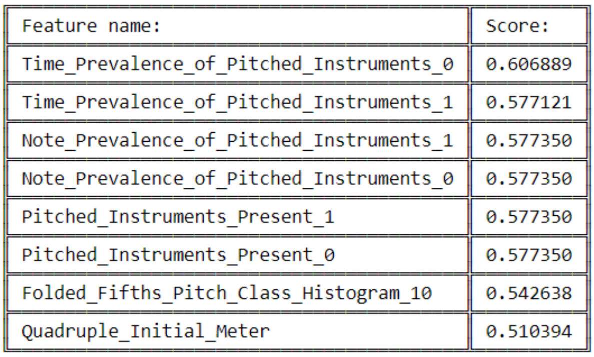


Figure 3. Top 1 Feature - Extra Trees Classifier

3.1.2 Pearson Correlation Feature Selection

Top 8 features with correlation coefficient > 0.5 with target variable Category:



Feature name:	Score:
Time_Prevalence_of_Pitched_Instruments_0	0.606889
Time_Prevalence_of_Pitched_Instruments_1	0.577121
Note_Prevalence_of_Pitched_Instruments_1	0.577350
Note_Prevalence_of_Pitched_Instruments_0	0.577350
Pitched_Instruments_Present_1	0.577350
Pitched_Instruments_Present_0	0.577350
Folded_Fifths_Pitch_Class_Histogram_10	0.542638
Quadruple_Initial_Meter	0.510394

Figure 4. Top 8 Features - Pearson Correlation

Top 3 features according to Pearson Correlation technique:

- **Time_Prevalence_of_Pitched_Instruments_0**
- **Time_Prevalence_of_Pitched_Instruments_1**
- **Note_Prevalence_of_Pitched_Instruments_1.** A feature vector indicating the fraction of (pitched) notes played with each of the 128 General MIDI Instrument patches (0 is Acoustic Piano, 40 is Violin, etc.). Has one entry for each of these instruments, and the value of each is set to the number of Note Ons played with the corresponding MIDI patch, divided by the total number of Note Ons in the piece.

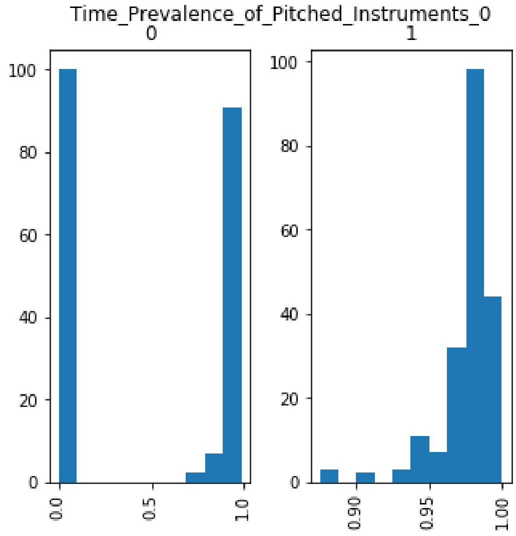
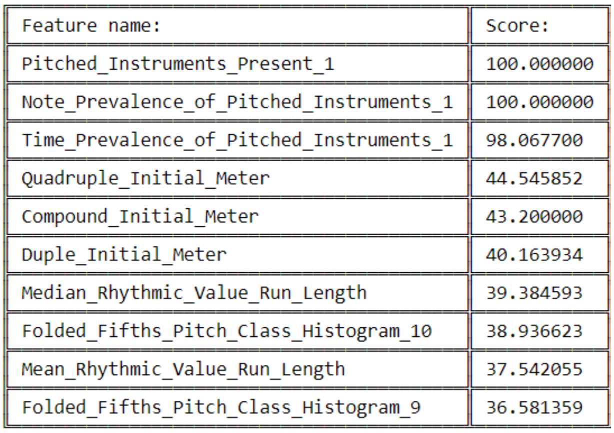


Figure 5. Top 1 Feature - Pearson Correlation

3.1.3 Univariate Feature Selection

Top 10 features according to Univariate Feature Selection technique for the scaled data:



Feature name:	Score:
Pitched_Instruments_Present_1	100.000000
Note_Prevalence_of_Pitched_Instruments_1	100.000000
Time_Prevalence_of_Pitched_Instruments_1	98.067700
Quadruple_Initial_Meter	44.545852
Compound_Initial_Meter	43.200000
Duple_Initial_Meter	40.163934
Median_Rhythmic_Value_Run_Length	39.384593
Folded_Fifths_Pitch_Class_Histogram_10	38.936623
Mean_Rhythmic_Value_Run_Length	37.542055
Folded_Fifths_Pitch_Class_Histogram_9	36.581359

Figure 6. Top 10 Feature - Univariate Feature Selection

Top 3 Features according to Univariate Feature Selection technique:

- **Pitched_Instruments_Present_1**
- **Note_Prevalence_of_Pitched_Instruments_1**
- **Time_Prevalence_of_Pitched_Instruments_1**

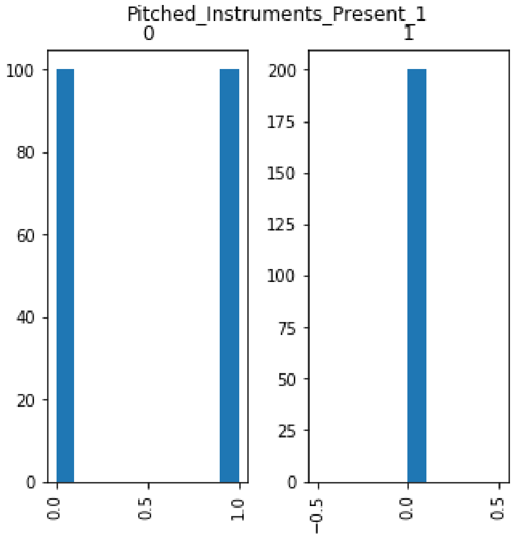


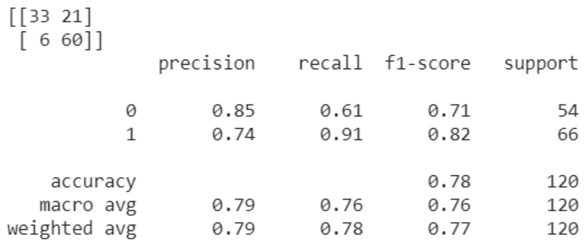
Figure 7. Top 1 Feature - Univariate Feature Selection

3.2 Classification

3.2.1 Classification specifications

- Classifier: Support Vector Machine
- Train set: 280 songs (70%)
- Test set: 120 songs (30%)

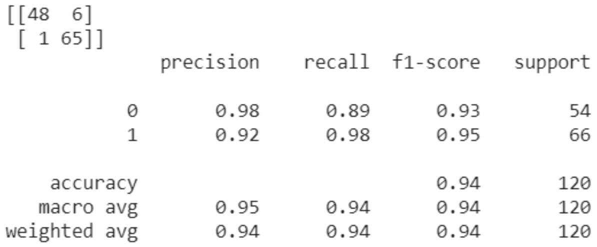
3.2.2 Classification using all 679 features



```
[[33 21]
 [ 6 60]]
              precision    recall  f1-score   support

           0       0.85      0.61      0.71        54
           1       0.74      0.91      0.82        66

    accuracy                           0.78       120
   macro avg       0.79      0.76      0.76       120
weighted avg       0.79      0.78      0.77       120
```

Figure 8. Classification - All features

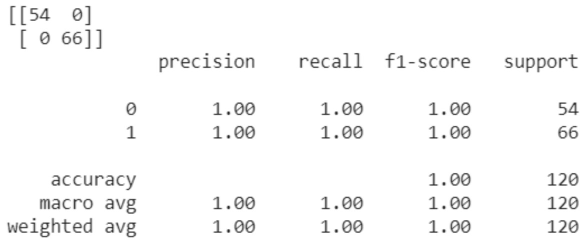
3.2.3 Classification using Top 15 features of Extra Trees Classifier



```
[[48  6]
 [ 1 65]]
              precision    recall  f1-score   support

           0       0.98      0.89      0.93        54
           1       0.92      0.98      0.95        66

    accuracy                           0.94       120
   macro avg       0.95      0.94      0.94       120
weighted avg       0.94      0.94      0.94       120
```

Figure 9. Classification - Extra Trees Classifier

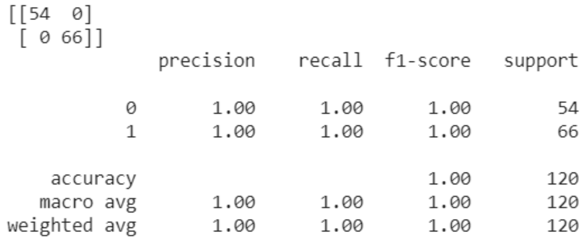
3.2.4 Classification using Top 8 features of Pearson Correlation



```
[[54  0]
 [ 0 66]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        54
           1       1.00      1.00      1.00        66

    accuracy                           1.00       120
   macro avg       1.00      1.00      1.00       120
weighted avg       1.00      1.00      1.00       120
```

Figure 10. Classification - Pearson Correlation

3.2.5 Classification using Top 10 features of Univariate Feature Selection

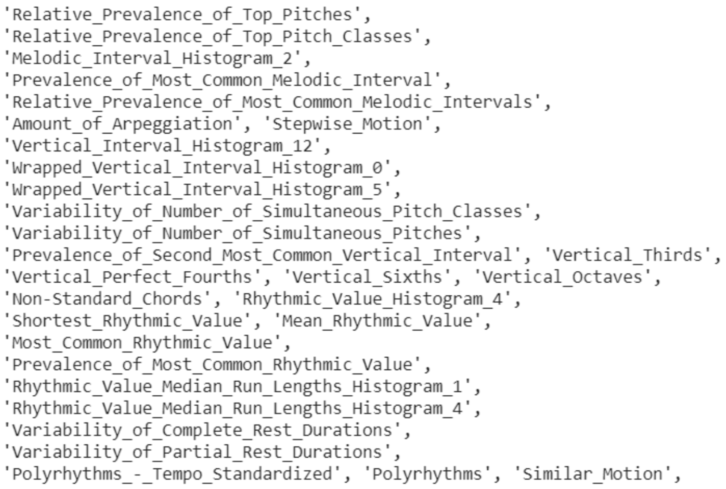


```
[[54  0]
 [ 0 66]]
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        54
           1       1.00      1.00      1.00        66

    accuracy                           1.00       120
   macro avg       1.00      1.00      1.00       120
weighted avg       1.00      1.00      1.00       120
```

Figure 11. Classification - Univariate Feature Selection

3.3 Common features

The 30 features that are indicators of commonalities between real and generated songs are listed below:



```
'Relative_Prevalence_of_Top_Pitches',
'Relative_Prevalence_of_Top_Pitch_Classes',
'Melodic_Interval_Histogram_2',
'Prevalence_of_Most_Common_Melodic_Interval',
'Relative_Prevalence_of_Most_Common_Melodic_Intervals',
'Amount_of_Arpeggiation', 'Stepwise_Motion',
'Vertical_Interval_Histogram_12',
'Wrapped_Vertical_Interval_Histogram_0',
'Wrapped_Vertical_Interval_Histogram_5',
'Variability_of_Number_of_Simultaneous_Pitch_Classes',
'Variability_of_Number_of_Simultaneous_Pitches',
'Prevalence_of_Second_Most_Common_Vertical_Interval', 'Vertical_Thirds',
'Vertical_Perfect_Fourths', 'Vertical_Sixths', 'Vertical_Octaves',
'Non-Standard_Chords', 'Rhythmic_Value_Histogram_4',
'Shortest_Rhythmic_Value', 'Mean_Rhythmic_Value',
'Most_Common_Rhythmic_Value',
'Prevalence_of_Most_Common_Rhythmic_Value',
'Rhythmic_Value_Median_Run_Lengths_Histogram_1',
'Rhythmic_Value_Median_Run_Lengths_Histogram_4',
'Variability_of_Complete_Rest_Durations',
'Variability_of_Partial_Rest_Durations',
'Polyrhythms_-_Tempo_Standardized', 'Polyrhythms', 'Similar_Motion',
```

Figure 12. 30 features indicating commonalities

Some examples of the common features:

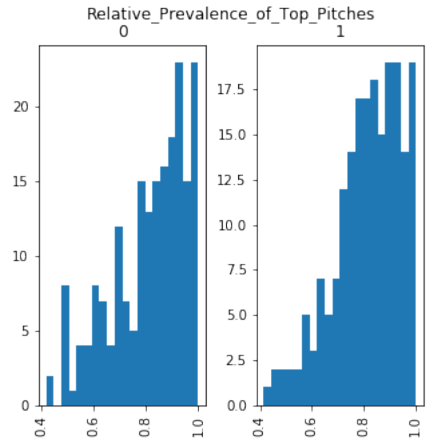


Figure 13. Common feature example

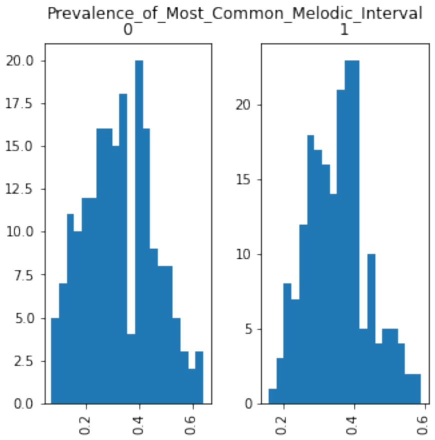
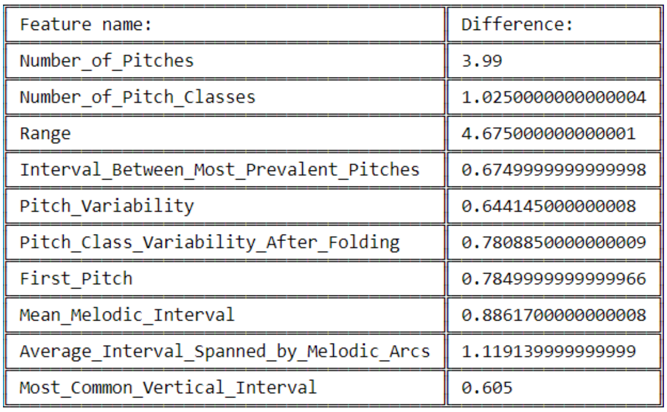


Figure 14. Common feature example

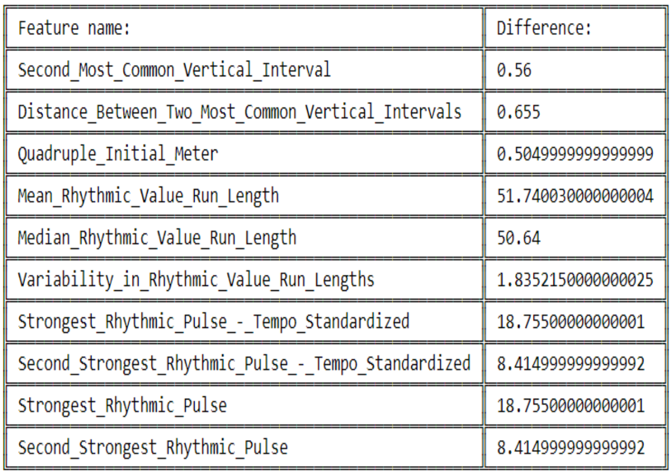
3.4 Artificialness indicators

The 20 features indicating the "artificialness" of the generated songs are shown below:



Feature name:	Difference:
Number_of_Pitches	3.99
Number_of_Pitch_Classes	1.0250000000000004
Range	4.675000000000001
Interval_Between_Most_Prevalent_Pitches	0.6749999999999998
Pitch_Variability	0.644145000000008
Pitch_Class_Variability_After_Folding	0.7808850000000009
First_Pitch	0.7849999999999966
Mean_Melodic_Interval	0.8861700000000008
Average_Interval_Spanned_by_Melodic_Arcs	1.119139999999999
Most_Common_Vertical_Interval	0.605

Figure 15. Artificialness indicators



Feature name:	Difference:
Second_Most_Common_Vertical_Interval	0.56
Distance_Between_Two_Most_Common_Vertical_Intervals	0.655
Quadruple_Initial_Meter	0.5049999999999999
Mean_Rhythmic_Value_Run_Length	51.740030000000004
Median_Rhythmic_Value_Run_Length	50.64
Variability_in_Rhythmic_Value_Run_Lengths	1.8352150000000025
Strongest_Rhythmic_Pulse_-_Tempo_Standardized	18.75500000000001
Second_Strongest_Rhythmic_Pulse_-_Tempo_Standardized	8.414999999999992
Strongest_Rhythmic_Pulse	18.75500000000001
Second_Strongest_Rhythmic_Pulse	8.414999999999992

Figure 16. Artificialness indicators

4. FUTURE WORK

- Team up with a more musically inclined individual to gather more insights and better understand individual features.
- Analyze the artificialness of the features in a more detailed way and from other points of view.

5. CONCLUSIONS

5.1 Features indicating differences

After performing several techniques for feature selection such as Extra Trees Classifier, Pearson Correlation and Univariate Feature Selection, they all agreed on the features that help discriminate the most between real and generated songs.

Those features were:

- **Time_Prevalence_of_Pitched_Instruments_0.**
- **Time_Prevalence_of_Pitched_Instruments_1.**
- **Note_Prevalence_of_Pitched_Instruments_1.**
- **Pitched_Instruments_Present_1**
- **Pitched_Instruments_Present_0**

When it comes to the classification results the accuracy improved considerably when feature selection was performed and overfitting was reduced.

- Using all 679 features: **Accuracy = 78%**
- Using Top 15 Extra Trees Classifier features: **Accuracy = 94%**
- Using Top 8 Pearson Correlation features: **Accuracy = 100%**
- Using Top 10 Univariate Feature Selection features: **Accuracy = 100%**

5.2 Features indicating commonalities

Out of all 1495 features extracted from jSymbolic only **30** of those could be considered as common between real and generated songs and those features are shown in **Figure 12**.

5.3 Features indicating artificialness

Our analysis returned *20* features that indicate artificialness of the generated songs. Mos of them have to do with **Pitch** features and **Rhythm** features.

6. REFERENCES

[1] Bob L. Sturm, João Felipe Santos, and Iryna Korshunova. Folk music style modelling by recurrent neural networks with long short term memory units. 2015.