Exploring Genre Detection

Presented by:
Arad Ben Menashe: 207083353
Eli Greenblatt: 316302934

Slides 16–18 were added after Presentation – (*)

The Problem:

Given a song, classify it’s genre.

Datasets

1. GTZAN Dataset: Music Genre Classification
- Song Length: 30 seconds
- Number of genres: 10
- File Type: WAV

2. Emotify: Emotion Classification in Songs
- Song Length: 60 seconds
- Number of Genres: 5
- File Type: MP3

Challenges Ahead

- Different file types and audio lengths
- Corrupted Data
- Usage of large WAV files for ML
- Combine data from different datasets
- Use ML libraries on our new data

WAV Features Extractions Methods

Methods used

- MFCC - Mel-Frequency Cepstral Coefficients
- Chroma Features

Methods not used

- Spectogram

```
file,genre,mfcc_0,mfcc_1,mfcc_2,mfcc_3,mfcc_4,mfcc_5,mfcc_6,mfcc_7,mfcc_8,mfcc_9,mfcc_10,mfcc_11,mfcc_12
16.wav,pop,-238.8751678466797,127.27025604248047,-37.97389221191406,51.89480972290039,-13.782609939575195,
17.wav,pop,-247.178634643554 7,184.70950317382812,37.14341354370117,25.353908538818 36,3.418922185897827,25.
15.wav,pop,-188.47991943359375,175.90480041503906,-9.14249324798584,36.22831344604492,-1.502336410522 46,2
29.wav,pop,-223.921768188476 56,187.35585021972656,-25.039817810058594,28.438974380493164,2.540235757827759
100.wav,pop,-172.490921020507 8,166.0281982421875,-36.30755615234375,54.785789489746094,7.260797023773193,1
28.wav,pop,-231.122772216797,153.77931213378906,-17.64138412475586,24.799285888671875,0.279492527246 4752,
14.wav,pop,-231.86044311523438,170.10203552246094,21.578001022338867,32.51295852661133,4.515195846557617,3
38.wav,pop,-164.681396484375,173.43675231933594,-18.966567993164062,43.24700164794922,-13.787580490112305,
10.wav,pop,-200.66616821289062,186.0851593017578,10.936894416809082,6.681351661682129,-2.294553279876709,1
11.wav,pop,-217.62081909179688,123.14049530029297,-16.90761947631836,44.94382095336914,-0.865926921367645 3
39.wav,pop,-245.10174560546875,92.23126220703125,14.564506530761719,62.01643753051758,-1.265625,31.4864349
13.wav,pop,-67.70549774169922,158.69857788085938,-38.999305725097656,46.975704193115234,-21.535657882690 43
12.wav,pop,-131.30885314941406,175.67430114746094,-18.154264450073242,34.842769622802734,-5.956207275390 62
61.wav,pop,-201.24797058105 47,160.71444702148438,-19.36237335205078,16.297029495239258,-1.447929263114929 2
75.wav,pop,-109.52352142333984,143.96873474121094,-23.366195678710938,44.0577507019043,-2.352391242980957,
49.wav,pop,-97.08464050292969,113.88195037841797,-16.087966918945312,63.044334411621094,-14.155562400817 87
48.wav,pop,-187.17840576171875,179.35305786132812,-9.070626258850098,26.782920837402344,-10.108542442321 77
74.wav,pop,-161.5270538330078,194.51112365722656,-10.375207901000977,33.64959716796875,7.19343900680542,24
60.wav,pop,-262.2650146484375,127.01448059082031,26.948165893554688,30.852624893188477,-27.781049728393555
76.wav,pop,-202.28515625,169.3562774658203,1.0927232503890991,38.323482513427734,-4.044477939605713,27.986
62.wav,pop,-69.69837951660156,140.46401977539062,-35.15774917602539,56.11129379272461,-13.17696475982666,3
89.wav,pop,-161.9852294921875,148.39500427246094,-21.091291427612305,64.70287322998047,-9.243388175964355,
88.wav,pop,-286.9643249511719,160.53822326660156,7.539116859436035,38.24270248413086,9.084381103515625,15
```

Col 2: genre

file,genre,chroma_0,chroma_1,chroma_2,chroma_3,chroma_4,chroma_5,chroma_6,chroma_7,chroma_8,chroma_9,ch
16.wav,pop,0.4283204972743988,0.4497471153736145,0.5473080277442932,0.5878274440765381,0.5223448872566
17.wav,pop,0.5471160411834717,0.3818798363208771,0.3686733543872833,0.3901504278182983 4,0.5275225043296
15.wav,pop,0.6062553513110352,0.5214902758598328,0.5146049857139587,0.5388562083244324,0.5325037240982 0
29.wav,pop,0.3514741063117981,0.3121398985385895,0.4498779773712158,0.3878830671310425,0.3101525008678 4
100.wav,pop,0.5576605200767517,0.4074510037899017 3,0.3481497168540954 6,0.2935646772384643 6,0.3473254740
28.wav,pop,0.5078974962234497,0.3343603312969208,0.3732349574565887 5,0.4237965941429138,0.3678251802921
14.wav,pop,0.3795134723186493,0.2749434411525726 3,0.3409235775470733 6,0.5181711912155151,0.6072219610 21
38.wav,pop,0.3121578097343445,0.4176201820373535,0.4505765736103058,0.4300203621387481 7,0.5863553285598
10.wav,pop,0.3361562788486480 7,0.4516386687755584 7,0.3733663856983185,0.3711501657962799,0.3379685878 75
11.wav,pop,0.5018292665481567,0.4613409042358398 4,0.5561655163764954,0.6144249439239502,0.6678922176361
39.wav,pop,0.5990266203880310,0.5641305446624756,0.6041485071182251,0.4979655146598816,0.4726255238056 18
13.wav,pop,0.5927106142044067,0.4474815428256988 5,0.4351474046707153 3,0.5110740065574646,0.6339769363 40
12.wav,pop,0.4134972989559173 6,0.4643614292144775 4,0.3848580718040466 3,0.4605232179164886 5,0.3821253776
61.wav,pop,0.2829972505569458,0.2785330116748810 7,0.3801622688770294,0.2637479901313782,0.2726676166057 58
75.wav,pop,0.4055005311965942 4,0.5750549435615540 7,0.4734390079975128,0.4180013835430145,0.4346026480197 9
49.wav,pop,0.4802807867527008,0.5090691447257996,0.5939783453941345,0.6884092688560486,0.6954329013824 4
48.wav,pop,0.4354296028614044,0.6172732114791870 7,0.4234341382980346 7,0.3910520076751709,0.4423178136348 7
74.wav,pop,0.3515998721122741 7,0.3517164289951324 5,0.4368494153022766,0.3753995001316070 6,0.3588384985 9
60.wav,pop,0.5035510659217834,0.3608613610267639,0.4538823962211609,0.4616377651691437,0.3445466458797 4
76.wav,pop,0.4080460965633392 3,0.3859205543994903 6,0.4909915328025818,0.5005362629890442,0.4822545647 62
62.wav,pop,0.5079818367958069,0.5953387618064880 7,0.5028584599494934,0.4832869768142700 7,0.5862894654273987,
89.wav,pop,0.2882965207099914 6,0.4149371683597564 7,0.4188010692596435 5,0.3972326219081878 7,0.4470335245

ML Models

- KNN - Classifies based on the nearby data points
- NN - Neureal Network using 5 layes of 128 nodes.
- Decision Trees - Make decisions by following a tree-like structure.
- Random Forests - Multiple decision trees, voting together.
- SVM - Finds the best hyperplane to separate classes.

Decision Tree

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-ditection/src/ml_algorithms/decision_tree.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
{'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.23
{'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.26
{'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.27
{'max_depth': 7, 'criterion': 'gini', 'min_samples_split': 10, 'min_samples_leaf': 4}: 0.27
{'max_depth': 10, 'criterion': 'entropy', 'min_samples_split': 10, 'min_samples_leaf': 2}: 0.26
File: features/mfcc/features_29032024_1930.csv
{'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.31
{'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.36
{'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.42
{'max_depth': 7, 'criterion': 'gini', 'min_samples_split': 10, 'min_samples_leaf': 4}: 0.38
{'max_depth': 10, 'criterion': 'entropy', 'min_samples_split': 10, 'min_samples_leaf': 2}: 0.41
```

Parameters

- max_depth - Maximum depth of the tree
- criterion - Measure of quality
- min_samples_split - Minimum samples required to split a node
- min_samples_leaf - Minimum samples required for a leaf node

Random Forest

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-ditection/src/ml_algorithms/random_forrest.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.28
{'n_estimators': 150, 'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.30
{'n_estimators': 100, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.37
File: features/mfcc/features_29032024_1930.csv
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.36
{'n_estimators': 150, 'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.46
{'n_estimators': 100, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.57
```

Parameters

- n_estimators - Number of trees in the forest
- max_depth - Maximum depth of the trees
- criterion - Measure of quality
- min_samples_split - Minimum samples required to split a node
- min_samples_leaf - Minimum samples required for a leaf node

SVM

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-ditection/src/ml_algorithms/svm.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
C=1.0, kernel=linear: 0.30
C=1.0, kernel=rbf: 0.34
C=1.0, kernel=poly: 0.29
C=1.0, kernel=sigmoid: 0.20
File: features/mfcc/features_29032024_1930.csv
C=1.0, kernel=linear: 0.50
C=1.0, kernel=rbf: 0.59
C=1.0, kernel=poly: 0.49
C=1.0, kernel=sigmoid: 0.30
```

Parameters

- C - Controls trade off between low training error and low model complexity
- kernel - Specifies the kernel function used in the SVM algorithm

KNN

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-ditection/src/ml_algorithms/knn.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
k=3: 0.32
k=5: 0.33
k=7: 0.33
k=9: 0.34
k=13: 0.32
k=17: 0.32
File: features/mfcc/features_29032024_1930.csv
k=3: 0.53
k=5: 0.54
k=7: 0.55
k=9: 0.56
k=13: 0.56
k=17: 0.54
```

Parameters

- k - Number of closest neighbors we sample

NN

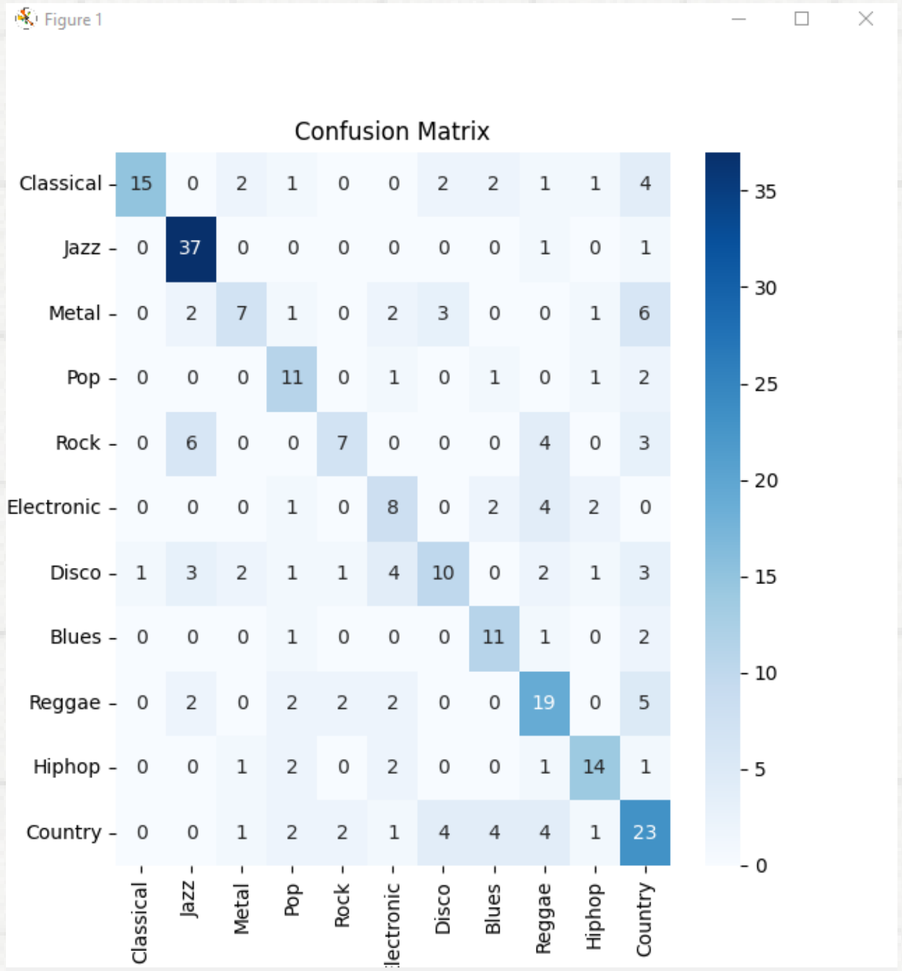
```
Epoch [250/2000], Loss: 2.2051994800567627
Epoch [500/2000], Loss: 1.8653644323349
Epoch [750/2000], Loss: 1.9205589294433594
Epoch [1000/2000], Loss: 1.8609071969985962
Epoch [1250/2000], Loss: 1.7027665376663208
Epoch [1500/2000], Loss: 1.827384352684021
Epoch [1750/2000], Loss: 1.702250599861145
Epoch [2000/2000], Loss: 1.826667308807373
Neural Network
File: features/chroma/features_29032024_1938.csv
Accuracy: 35.0%
```

```
Epoch [250/2000], Loss: 2.4140083789825444
Epoch [500/2000], Loss: 2.349492311477661
Epoch [750/2000], Loss: 2.349492311477661
Epoch [1000/2000], Loss: 2.3817503452301025
Epoch [1250/2000], Loss: 2.3817503452301025
Epoch [1500/2000], Loss: 2.4140083789825444
Epoch [1750/2000], Loss: 2.4140083789825444
Epoch [2000/2000], Loss: 2.4462664127349854
Neural Network
File: features/mfcc/features_29032024_1930.csv
Accuracy: 11.428571428571429%
```

Parameters

- Epoch - Times we run the NN
- Learning Rate - Size of the step

SVM with the MFCC features: 59% ACCURACY



Other Exploration Possibilities

- Combine the 2 features tables
- Spectrogram
- Adaboost

Other Exploration Possibilities - Continued

- Combined the 2 features tables - We implemented after the presentation, gaining better results in every model up to 23% increase in each one. And our best accuracy before - 59% increased to - 61%.

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-detection/src/ml_algorithms/svm.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv
Evaluating file: features/combined_features.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
C=1.0, kernel=linear: 0.30
C=1.0, kernel=rbf: 0.34
C=1.0, kernel=poly: 0.29
C=1.0, kernel=sigmoid: 0.20
File: features/mfcc/features_29032024_1930.csv
C=1.0, kernel=linear: 0.50
C=1.0, kernel=rbf: 0.59
C=1.0, kernel=poly: 0.49
C=1.0, kernel=sigmoid: 0.30
File: features/combined_features.csv
C=1.0, kernel=linear: 0.56
C=1.0, kernel=rbf: 0.61
C=1.0, kernel=poly: 0.50
C=1.0, kernel=sigmoid: 0.42
```

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-detection/src/ml_algorithms/random_forrest.py
Evaluating file: features/chroma/features_29032024_1938.csv
Evaluating file: features/mfcc/features_29032024_1930.csv
Evaluating file: features/combined_features.csv

Overall Results:
File: features/chroma/features_29032024_1938.csv
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.28
{'n_estimators': 150, 'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.31
{'n_estimators': 100, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.36
{'n_estimators': 150, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.36
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.27
{'n_estimators': 150, 'max_depth': 7, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.35
File: features/mfcc/features_29032024_1930.csv
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.36
{'n_estimators': 150, 'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.45
{'n_estimators': 100, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.57
{'n_estimators': 150, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.57
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.37
{'n_estimators': 150, 'max_depth': 7, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.54
File: features/combined_features.csv
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'gini', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.38
{'n_estimators': 150, 'max_depth': 5, 'criterion': 'gini', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.47
{'n_estimators': 100, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.59
{'n_estimators': 150, 'max_depth': None, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.58
{'n_estimators': 100, 'max_depth': 3, 'criterion': 'entropy', 'min_samples_split': 2, 'min_samples_leaf': 1}: 0.38
{'n_estimators': 150, 'max_depth': 7, 'criterion': 'entropy', 'min_samples_split': 4, 'min_samples_leaf': 2}: 0.55
```

```
❯ /opt/homebrew/bin/python3.11 /Users/aradbm/Desktop/machine-learning/genre-detection/src/ml_algorithms/nn.py
Epoch [250/2500], Loss: 1.8411993980407715
Epoch [500/2500], Loss: 1.732395052909851
Epoch [750/2500], Loss: 1.670743703842163
Epoch [1000/2500], Loss: 1.6700602769851685
Epoch [1250/2500], Loss: 1.6369163990020752
Epoch [1500/2500], Loss: 1.7365854978561401
Epoch [1750/2500], Loss: 1.6055430173873901
Epoch [2000/2500], Loss: 1.6398082971572876
Epoch [2250/2500], Loss: 1.7037113904953003
Epoch [2500/2500], Loss: 1.6698817014694214
Neural Network
File: features/combined_features.csv
Accuracy: 53.214285714285715%
```

Nureal Network Improved from 35/11% to 53%!

Thank you for listening!

github.com/aradbm/genre-detection