Project 4

Music Genre Classification

Submitted by:
Hinda Filali & Andriy Khalik

Foreword

As media consumptions grows, companies nowadays use music classification algorithms to be able to place recommendations to their customers or simply as a product.

Determining music genres is the first step in that direction. Machine Learning techniques have proved to be quite successful in extracting trends and patterns from a large data pool. The same principles are applied in Music Analysis also.

GOALS AND OBJECTIVES

This project is primarily aimed to explore an effective machine learning algorithm to classify and predict music genres.

Objectives:

- ❖ Retrieve and understand data, including features and classes
- ❖ Determine and assess appropriate ML models
- ❖ Analyse model optimization scénarios and a model prediction capacity
- ❖ Deploy the model
- ❖ Discuss model limitations and potential areas of further improvements

AGENDA

1. METHODOLOGY
2. DATA EXPLORATION: PRE-PROCESSING AND ANALYSIS
3. MODEL TESTING
4. MODEL OPTIMIZATION
5. DEPLOYMENT AND PREDICTIONS
6. FINAL THOUGHTS, LIMITATIONS AND FURTHER DEVELOPMENT

1. METHODOLOGY

APPLICATIONS AND LIBRARIES

This is a **ML classification problems.** Three supervised ML models were used to assess the predictive power of the models for the current problem. The models are:

- ❖ Logistic Regression
- ❖ KNN
- ❖ Random Forest

The following libraries and applications were also used in data pre-processing, analysis, and visualization:

- ❖ Scikit-Learn
- ❖ Pandas
- ❖ Matplotlib
- ❖ Flask
- ❖ Heroku
- ❖ Tableau
- ❖ Mongodb
- ❖ seaborn

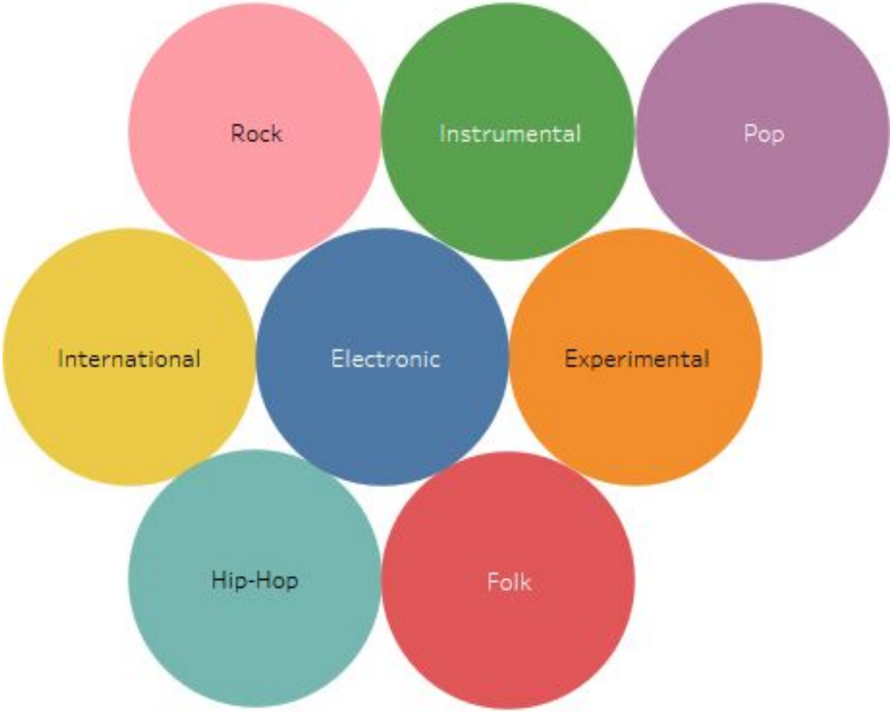
PROCESS

Data retrieval and preprocessing
Data analysis and selection
Model analysis and selection
Models optimization
Deployment and predictions analysis

2. DATA EXPLORATION: PRE-PROCESSING AND ANALYSIS

DATASETS

- The dataset was obtained in a *csv* format and included a list of audio tracks decomposed into core elements: pitch (chroma), prhythm dynamics, etc. Total of 29 elements and elements' derivatives were present in the dataset.
- Each element had two values: *Mean* and *Variance*
- The data contained 8 music genres



DATASET SAMPLE

filename	chroma_stft_mean	chroma_stft_var	rms_mean	rms_var	spectral_centroid_mean	spectral_centroid_var	s
1-000574.mp3	0.381544	0.080175	0.249562	0.001513	1957.407156	39590.736312	
2-000574.mp3	0.436518	0.074070	0.263343	0.001272	1969.829480	75422.537033	
3-000574.mp3	0.484288	0.069135	0.253424	0.001391	2766.351306	309436.169216	
4-000574.mp3	0.425569	0.074970	0.256628	0.001596	2908.207536	157800.863917	
5-000574.mp3	0.468948	0.066816	0.265448	0.001309	2618.340173	129314.679539	

DATA PRE-PROCESSING

Data needed to be cleaned and pre-processed

- ❖ Checking NaN values
- ❖ Removing values which did not affected a predictive power, that is ‘filename’, ‘id’
- ❖ Dropping VAR values for each element to reduce the number of features
- ❖ Checking how balanced the data is
- ❖ Splitting data into **two subsets** (90/10):

  (1) model train and test data (90%), and

  (2) true data to validate predictions after deployment (10%)

DATASETS

Two datasets were obtained and analysed:

1. 30-second audio file with approx 8k records
2. 3-second audio file (a breakup of each 30 sec file into 10 chunks with approx 80k records

We run three ML models on each dataset to test which data could better train the model. The results of this exercise are shown in the table.

Model accuracy scores

	30-sec dataset		3-sec dataset	
	Train	Test	Train	Test
LR	49%	47%	41%	41%
KNN (k=3)	70%	46%	91%	80%
RF	100%	51%	100%	76%

3-second dataset is found to train the model better, therefore this data set will be used in the further analysis.

3. MODEL TESTING

MODEL ANALYSIS

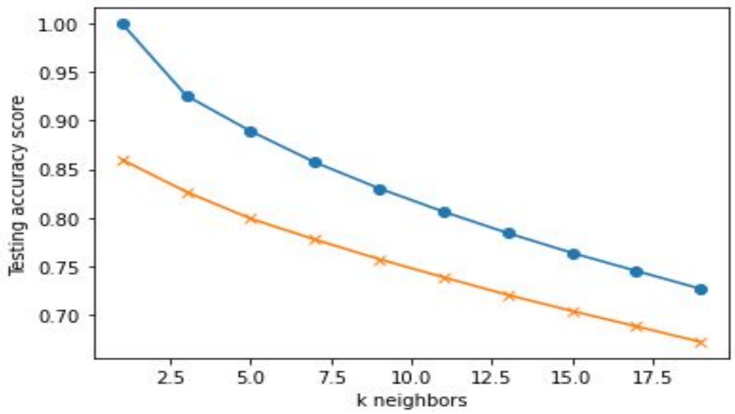
The initial data and model testing suggested that KNN provides better training and testing accuracy.

Model accuracy scores

	3-sec dataset	
	Train	Test
LR	41%	41%
KNN (k=3)	91%	82%
RF	100%	76%

KNN MODEL ANALYSIS

- Split data into 75% training and 25% testing using sklearn.model_selection, train_test_split
- Scaling data using StandardScaler.
- Initializing the model and determining the best K value which is the number 3 and then refit the knn classifier by using this k value.



	precision	recall	f1-score	support
Electronic	0.69	0.84	0.76	2231
Experimental	0.78	0.81	0.79	2258
Folk	0.78	0.81	0.80	2278
Hip-Hop	0.81	0.81	0.81	2234
Instrumental	0.82	0.79	0.80	2286
International	0.81	0.80	0.80	2250
Pop	0.86	0.69	0.76	2247
Rock	0.88	0.84	0.86	2203
accuracy			0.80	17987
macro avg	0.80	0.80	0.80	17987
weighted avg	0.80	0.80	0.80	17987

KNN produces better scores compared to LR and RF models, as indicated in the classification report

RF MODEL ANALYSIS

- Split data into 75% training and 25% testing using sklearn.model_selection, train_test_split
- Scaling data using StandardScaler.

```
clf =  RandomForestClassifier(n_estimators = 100).fit(X_train_scaled, y_train)
y_pred = clf.predict(X_test_scaled)
confusion_matrix(y_test, y_pred)
```

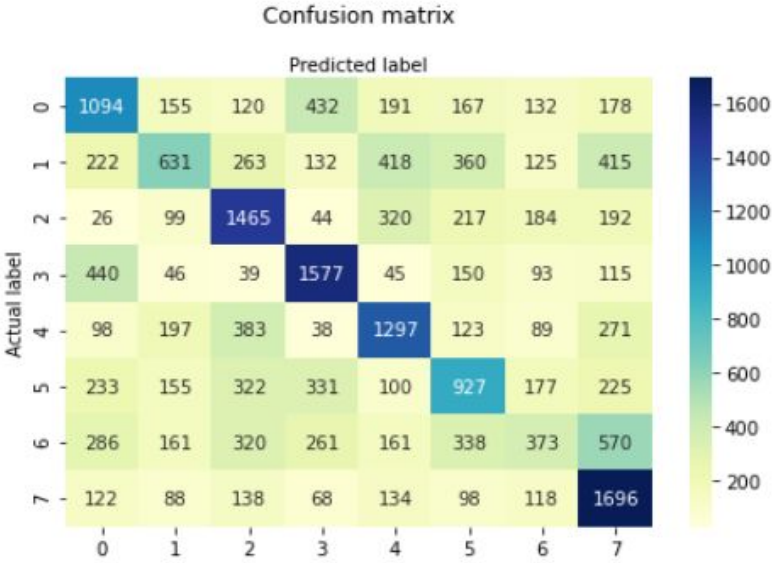
```
array([[650,  45,  24,  56,  49,  24,  30,  22],
       [ 44, 585,  45,  33,  59,  56,  33,  44],
       [  6,  15, 736,   9,  44,  32,  30,  29],
       [ 51,   8,  15, 808,   8,  24,  26,  12],
       [ 20,  27,  42,  10, 686,  25,  22,  36],
       [ 34,  10,  50,  89,   8, 643,  28,  18],
       [ 44,  36,  83,  75,  23,  76, 526,  81],
       [ 21,  24,  42,  28,  24,  30,  39, 643]], dtype=int64)
```

	precision	recall	f1-score	support
Electronic	0.77	0.72	0.75	2220
Experimental	0.83	0.71	0.76	2198
Folk	0.68	0.82	0.74	2207
Hip-Hop	0.74	0.82	0.78	2272
Instrumental	0.78	0.81	0.79	2271
International	0.77	0.76	0.77	2301
Pop	0.76	0.63	0.69	2201
Rock	0.77	0.79	0.78	2317
accuracy			0.76	17987
macro avg	0.76	0.76	0.76	17987
weighted avg	0.76	0.76	0.76	17987

RF training score is 1, suggesting the model is overtrained and may not produce consistent predictions

LR MODEL ANALYSIS

- Split data into 75% training and 25% testing using sklearn.model_selection, train_test_split
- Scaling data using StandardScaler.



	precision	recall	f1-score	support
Electronic	0.39	0.41	0.40	2251
Experimental	0.35	0.21	0.26	2232
Folk	0.44	0.57	0.50	2262
Hip-Hop	0.46	0.52	0.49	2252
Instrumental	0.45	0.47	0.46	2231
International	0.39	0.36	0.37	2292
Pop	0.26	0.12	0.17	2270
Rock	0.43	0.66	0.52	2197
accuracy			0.41	17987
macro avg	0.40	0.41	0.40	17987
weighted avg	0.40	0.41	0.40	17987

LR model produces the lowest accuracy and precision scores among the three models

4. MODEL OPTIMIZATION ANALYSIS

OPTIMIZATION ANALYSIS

We performed the following optimization techniques to increase the predictive power of the models:

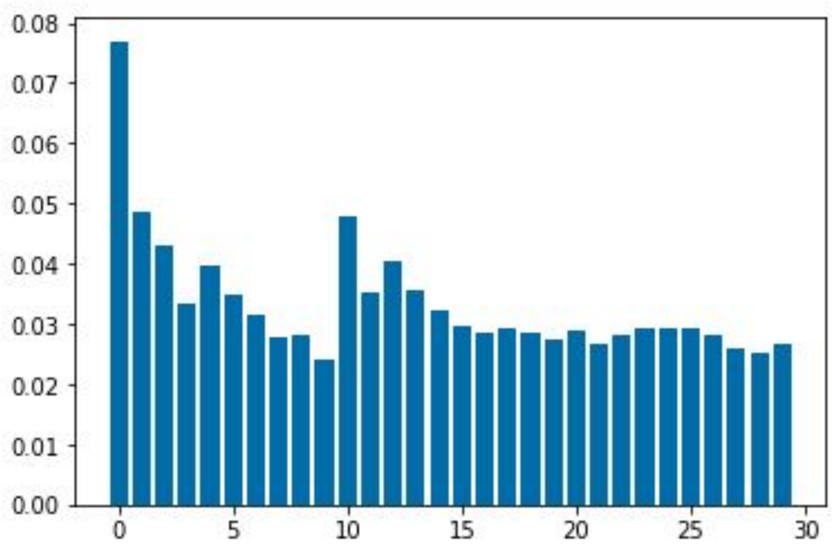
❖ The dataset was optimized using a standard scaler. During the initial testing the models produced better accuracy scores on the scaled data

❖ We attempted to optimize RF model by analysing ‘feature importance’. The quality of the model did not improved after removing ‘unimportant’ features. Moreover, the model suggested that majority of features were somewhat equally important (see chart.

❖ In order to optimize our models further **we divided the label classes into 4 binaries genres**:

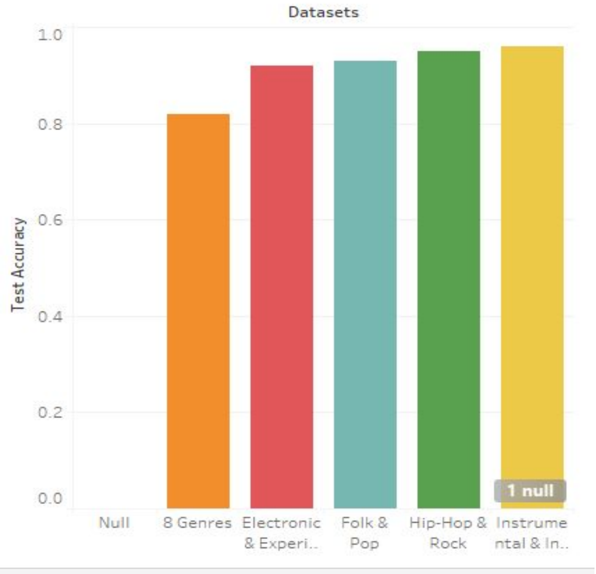
- Pop & Folk
- Hip-hop & Rock
- Instrumental & International
- Electronic & Experimental

Feature importance chart

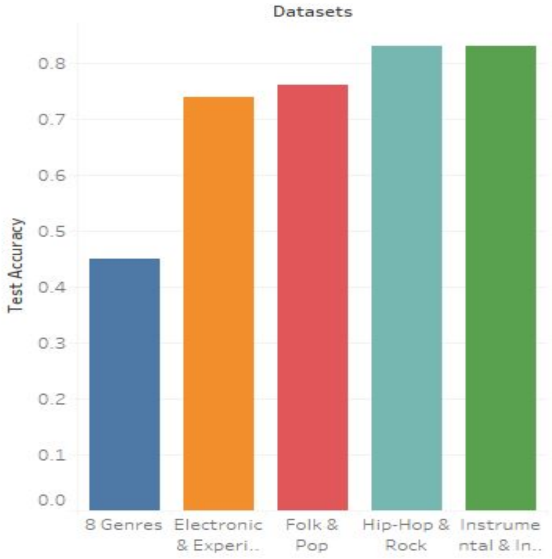


OPTIMIZATION ANALYSIS

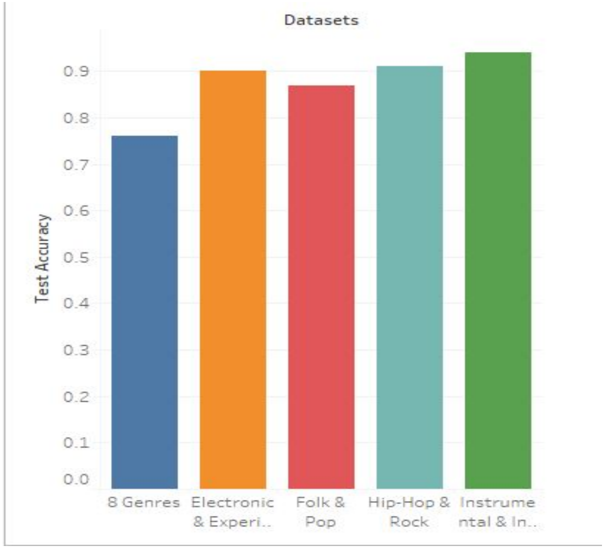
- **K-nearest neighbor (K=3)**



Logistic Regression



Random Forest Classification



Binary model produces better accuracy, with KNN model outperforming LR and RF

5. DEPLOYMENT

DEPLOYMENT

For the deployment purposes we used Knn 8 output model.

We used Flask and Heroku to deploy it.

A simple interface randomly selects five records from a separate dataset and predicts the class.

Song Genre Prediction

Click to Update

KNN model predictions

Prediction Results:

	Prediction	Actual
734	Folk	Hip-Hop
3029	Rock	Rock
2374	Pop	Hip-Hop
983	Instrumental	Rock
4844	International	International

Classification report:

	precision	recall	f1-score	support
Folk	0.000000	0.00	0.000000	0.0
Hip-Hop	0.000000	0.00	0.000000	2.0
Instrumental	0.000000	0.00	0.000000	0.0
International	1.000000	1.00	1.000000	1.0
Pop	0.000000	0.00	0.000000	0.0
Rock	1.000000	0.50	0.666667	2.0
accuracy	0.400000	0.40	0.400000	0.4
macro avg	0.333333	0.25	0.277778	5.0
weighted avg	0.600000	0.40	0.466667	5.0

https://ak-project4.herokuapp.com

6. FINAL THOUGHTS, LIMITATIONS AND FURTHER DEVELOPMENT

FINAL THOUGHTS

We have tried various machine learning algorithms for this project. Our aim was to get maximum accuracy. We found from our research that we can get a maximum accuracy of 96% with Knn model by selecting 2 genre classes but for the web application we decided to use the K-nearest neighbor since it was able to recognise all the genres with a good accuracy of 82%.

LIMITATIONS AND FURTHER RESEARCH

- ❖ Our model is not designed to process audio files and retrieve audio elements/features. It can be further expanded to include this capability.

- ❖ While we achieved 82% score with 8 outputs using Knn, further optimization or models should be explored to increase the performance of the model. Deep learning model could be checked.

- ❖ While we split the dataset into binary classes, we did not test them with real data. Area of further analysis.

Thank you for watching

Questions ?

Some References

1. https://stackabuse.com/random-forest-algorithm-with-python-and-scikit-learn/
2. https://www.freecodecamp.org/news/how-to-deploy-an-application-to-heroku/
3. https://data-flair.training/blogs/python-project-music-genre-classification/#google_vignette
4. https://www.kaggle.com/lakshyagupta/music-genre-prediction
5. https://www.analyticssteps.com/blogs/music-genre-classification-using-machine-learning
6. https://dev.to/hackersandslackers/connect-flask-to-a-database-with-flask-sqlalchemy-5d15#:~:text=%20Connect %20Flask%2[…]0our%20database...%20More%20
7. https://www.youtube.com/watch?v=6wojjJoRxRM
8. Background image link - https://michaeljwhalen.medium.com/the-death-of-the-music-genre-mostly-7039094302f