Music Genre Classification

Omer Diab & Chi Ung

Objectives

- Developing a Machine Learning Models to classify Music into genres based on various different features,.
- Reaching a good accuracy so that the model classifies new music into its genre correctly.
- Finding the best model for genre classification

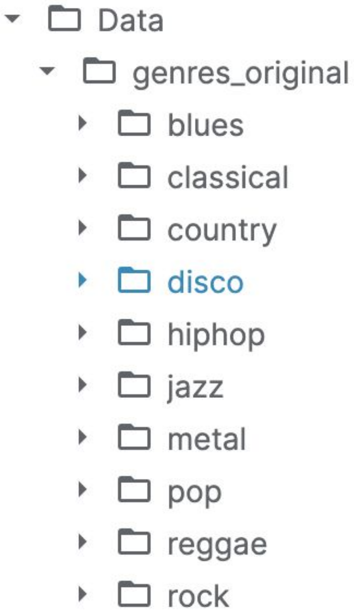
Goal

- To build a machine learning models which classifies music into its respective genre

- To compare the accuracies between these models and draw the necessary conclusions.

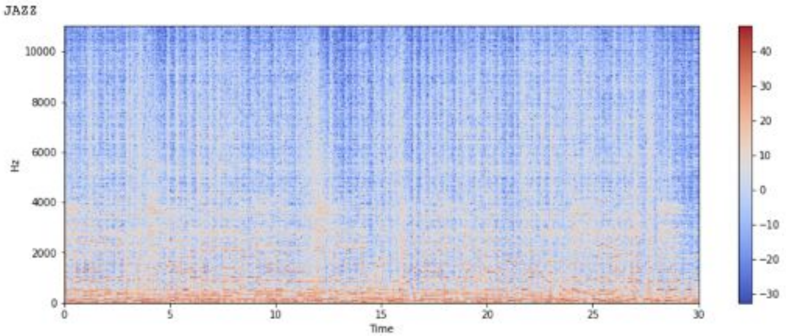
GTZAN Dataset

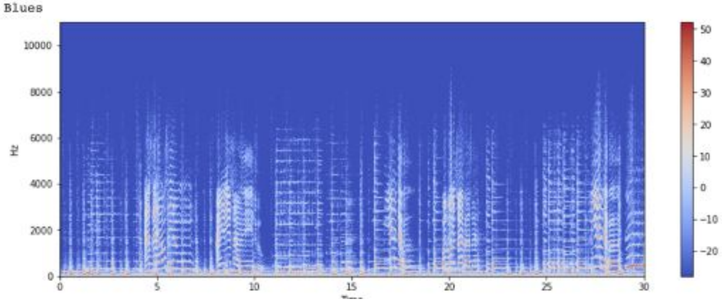
- Collection of 10 genres with 100 audio files each, and all of them have a length exactly 30 seconds
- Most used data in public for music genre recognition, classifier or different purpose related to audio files
- Collected between 2000-2001 from variety of sources by Marsyas



Features Extraction

- Librosa Python library
- Extracted Features
  - Spectral Centroid
  - Spectral Rolloff
  - Mel-Frequency Cepstral Coefficients (MFCC)
  - Chroma Frequencies





filename	chroma_stft	rmse	spectral_centroid	spectral_bandwidth	rolloff	zero_crossing_rate	tempo
0.wav	0.278484461674225 0	0.07697049528360370	1198.6076653608000	1573.308974392400	2478.376680261950 0	0.05198759191176470 0	83.3543346774194 0
1.wav	0.26932002161441300	0.11907171458005900	1361.045467327890	1567.8045957296200	2739.6251005284900	0.0691239206414474 0	92.28515625
2.wav	0.3990254820962200	0.127310633659363 0	2155.6549226375500	2372.40360414164 0	5012.01969300261 0	0.0871653837316176 0	129.19921875
3.wav	0.363602838496103	0.17557303607463800	1552.4819582458900	1747.1659849613100	3040.51494775542 0	0.0763007479924536 0	161.4990234375
4.wav	0.2332303656876700 0	0.197817638516426 0	1247.24481504879 0	1908.0527219040400	2620.5924869690900	0.0369037828947368 0	151.9990808823530 0
5.wav	0.337922168530768 0	0.12909629607544 00	2258.53841866570 0	2176.03118928977 0	4755.42957790127 0	0.1127650021768580 0	161.4990234375
6.wav	0.3766867027528340 0	0.1410178095102310 0	1239.3372282343400	1659.4664703825000	2517.6181096410600	0.05090785543246900	103.359375
7.wav	0.3962583163207620 0	0.235238000750542 0	2061.1507350613700	2085.1594475651900	4221.14947528662 0	0.1133968955921100	112.3471467391300 0
8.wav	0.408875618175811 0	0.2432173937559130 0	2206.7712464152600	2191.4735056963400	4657.38850407556 0	0.1115257806453170 0	99.3840144230769 0
9.wav	0.33645438248493200	0.11224512755870800	2013.3824371007600	2310.30551561594 0	4849.4956757631 1	0.07534497145897830	151.9990808823530 0

flux	contrast	flatness	mfcc1	mfcc2	mfcc3	mfcc4
1.435081956982080	21.972193081321500	0.00045489592594094600	-284.81950384065600	108.78562772693600	9.13195612642055 0	51.25902998309400
1.519149619770120	22.135934626217100	0.000767664809245616 0	-207.20808000096700	132.79917547173500	-15.438985574580700	60.9867270103288
1.3077578525283800	20.802436569429800	0.00646488601341844 0	-109.165355109461 0	100.62150013173600	-8.61472085506729 0	47.3584750420803 0
1.6382578632284800	22.197265163058100	0.0026277729775756600	-90.7543937912098 0	140.45990690263600	-29.109965287934800	31.689014398052000
1.4618357651403700	27.371254391953700	0.00044875507592223600	-200.22073177846700	116.34518082707800	18.060785149574200	25.288819404536400
1.3769762144535100	22.474323585465500	0.00575657887384296 0	-95.4244227767093	101.36865217103400	-20.68249652542900 0	48.6554756317883 0
1.2392396756030300	22.898596571831900	0.0011128768092021300	-206.278431438099 0	126.62746798711600	10.585204843575800	43.2231678634159 0
1.46262563896239 0	21.791968250547200	0.00729308929294348 0	-38.9659407655062 0	112.0398426976800 0	-31.817035125846600	38.2408351673679 0
1.52397832819660 0	21.405812093115900	0.008233388885855680	-29.010990408564400	104.53291407548900	-30.974207325223 20	38.156392118947200
1.0723682114087000	22.397010299398500	0.0031964604277163700	-149.95170135848400	93.6294798173271 0	6.34345438379006 0	71.5119457322069 0

mfcc15	mfcc16	mfcc17	mfcc18	mfcc19	mfcc20	label
3.32345474405318 0	3.2589197585578700	-4.55110598894488 0	0.49384497352911100	5.93706562598184 0	3.23154428197825 0	blues
-5.1889237507533200	-9.52745528651867 0	-9.24439404898140 0	-2.8482737510322900	-1.4187068621065600	-5.93260695967391 0	blues
-15.442803770843800	1.5387503660590100	-6.73247438416931 0	1.41777406986448 0	-3.9617504697981300	3.2874602683449200	blues
-9.21835889703751 0	2.4558052297775800	-7.726901333051800	-1.8157238511277200	-3.43343427176565 0	-2.22682144103072	blues
-11.959013510454900	-12.61797674724140	-14.187781208837 00	-8.20447551470853	-15.024769441842800	-10.322150110988700	blues
-9.80944501458191 0	6.91175557515353 0	-11.303163834042900	8.940704316986400	-2.959638893321990	-0.9754280541750500	blues
-7.15746381943972 0	1.1711138995440900	-6.22197459678454	0.600702590027873 0	-1.6050949507100900	-0.528773854695392 0	blues
-5.03127693080652 0	7.20098200991324 0	-6.75496913708767 0	2.66361161792598 0	-4.38042979122566 0	0.4140547073570290	blues
-10.294857593805200	6.96784540929861 0	-10.256099593498600	0.7050139543501770	-6.00072201650268 0	1.3489551857679500	blues
-12.40164114461870	9.62459961398473 0	-5.01418484370993 0	-3.7041455184211100	2.3425105218444400	4.16107727080114 0	blues

Dataset Standardization

- Feature scaling the dataset by applying StandarScaler(), provided from sklearn
- Encode the target (genres) with value between 0 and 9
- Placing the data and label in the same scaled space

```
scaler = StandardScaler()
X = scaler.fit_transform(np.array(data.iloc[:, :-1], dtype = float))

genre_list = data.iloc[:, -1]
encoder = LabelEncoder()
y = encoder.fit_transform(genre_list)
```

Classification

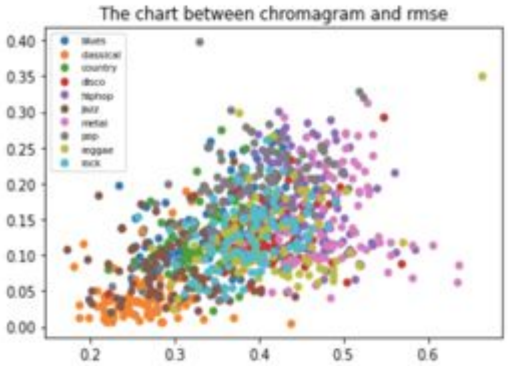
- K-Nearest Neighbors

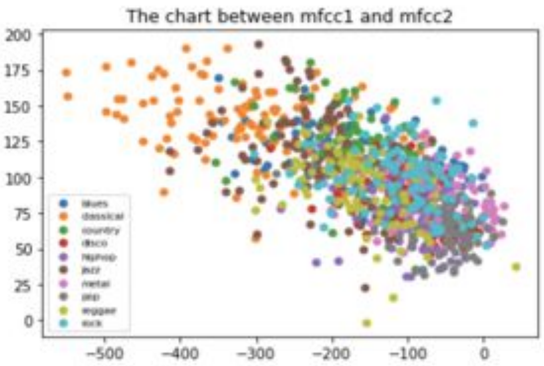
- Support Vector Machine (SVM) includes 3 kernels
  - Linear
  - RBF
  - Polynomial
- Logistic Regression

Analysis of MFCCs and Chromagram

- MFCCs perform better than Chromagram
- Chromagram closely relates 12 pitch classes -> common and used across all genres
- MFCCS models the characteristic of human voices and related to tones and musical instrument
- Tones and instrument does related to music genres

Algorithms	Chromagram	MFCC
Logistic Regression	20%	47%
Linear_SVC	19%	41%
RBF_SVC	25%	49%
Poly_SVC	18%	45%
KNN (neighbors = 3)	23%	53%

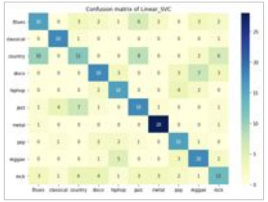
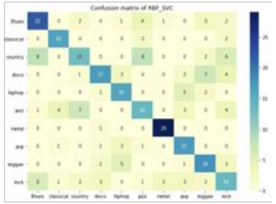
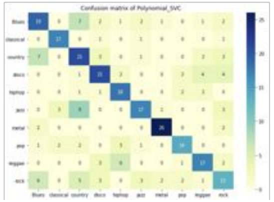
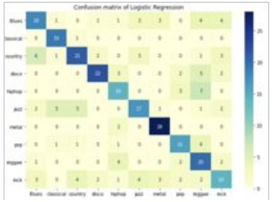


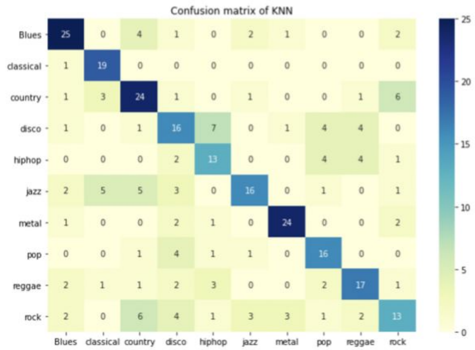


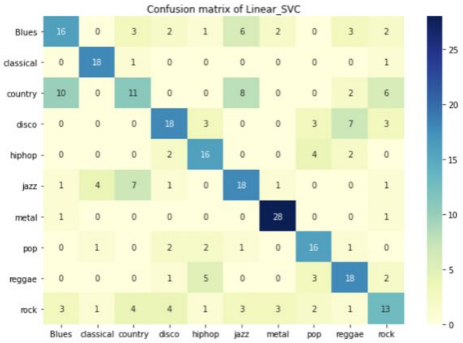
Analysis of The Dataset

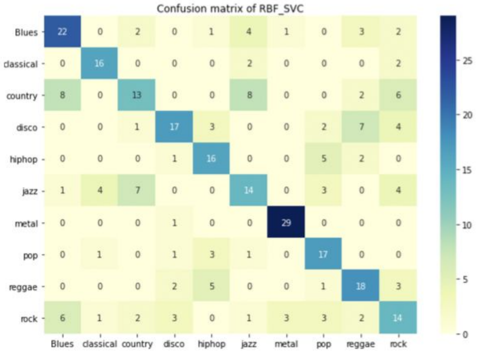
- Polynomial_SVC has best approach and Linear_SVC has least approach
- The confusion matrix graphs show:
  - Predicted Blue music genre as Country genre
  - Predicted Country genre as Jazz genre
  - Predicted Reggae genre as disco genre

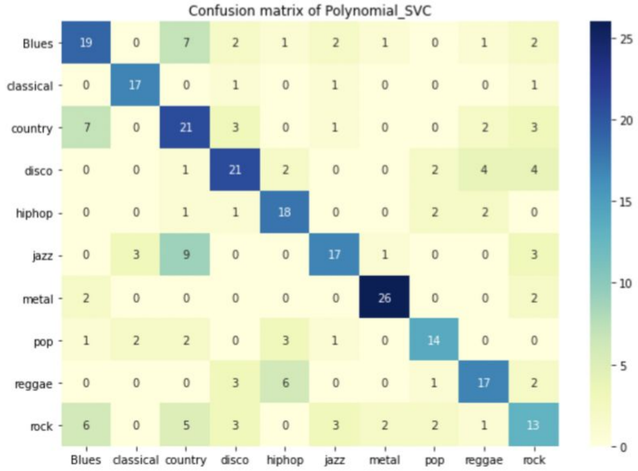
Models / Algorithms	Training Accuracy	Test Accuracy
Logistic Regression	76%	63%
Linear_SVC	70%	57%
RBF_SVC	79%	59%
Polynomial_SVC	83%	61%
KNN (best k value is 5)	78%	61%

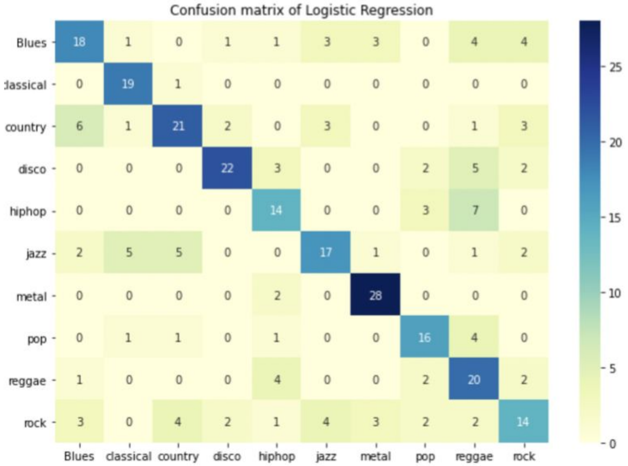







Confusion matrix of KNN
Blues classical country disco hiphop jazz metal pop reggae rock


Confusion matrix of Linear_SVC
Blues classical country disco hiphop jazz metal pop reggae rock


Confusion matrix of RBF_SVC
Blues classical country disco hiphop jazz metal pop reggae rock


Confusion matrix of Polynomial_SVC
Blues classical country disco hiphop jazz metal pop reggae rock


Confusion matrix of Logistic Regression
Blues classical country disco hiphop jazz metal pop reggae rock

Conclusion

- We Created our own dataset using GTZEN dataset of music files , and LIBROSA python library to extract Music Features.
- We used different approaches for the classification of music genres. These approaches:
  - K-Nearest Neighbor
  - Support Vector Machine (SVM)
  - Logistic Regression
- the best approach was Polynomial_SVC kernel in SVC algorithm with accuracy of 83% in training accuracy and 61% in testing accuracy.

Future Work

- Enlarge the Dataset by extracting features of more music
- enhance our classification models to get a better accuracy
- implement different options such as Music Emotion Recognition

Questions