



MUSIC Classification

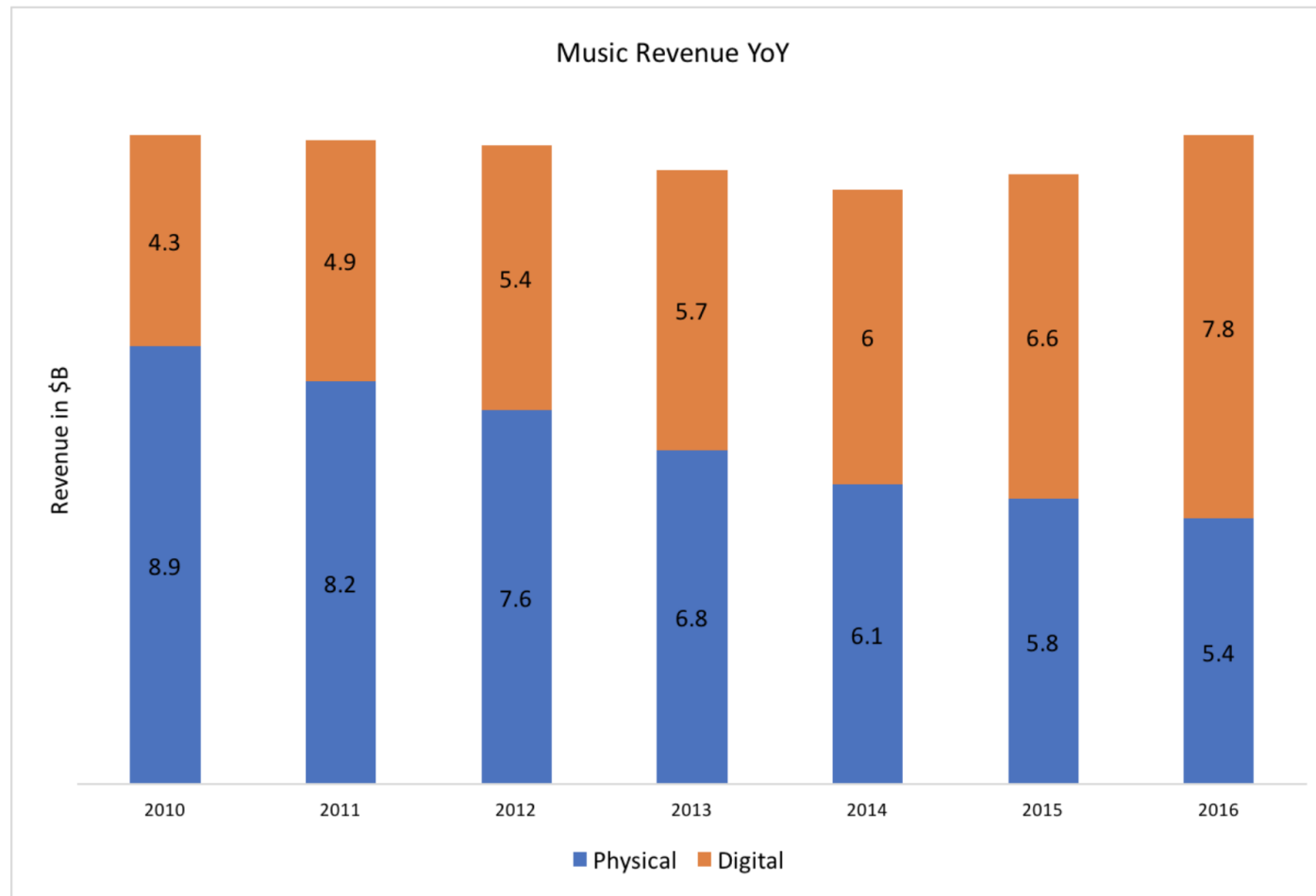




Scale and Growth of Digital Music Industry

- YouTube has ~1.5B monthly active users and a revenue of 40B
- Digital Music revenue(subscription+ streaming+ Download) has grown 10% YoY
- Physical Music revenue has dropped 10% YoY







Music Consumption has become predominantly Digital

- Digital Music content accounts for >50% of music revenues
- For the first time, we have the Data to understand user behaviour
- Understanding user behaviour is a key to growth of music platforms





Music Recommendation

1. Collaborative Filtering of User behaviour

- Agnostic of Music content
- Accurate with large datasets

2. Understanding and modeling using Music content

- Understand the human perception of music
- Build recommendation engine on top of that
- **Music classification is the key**



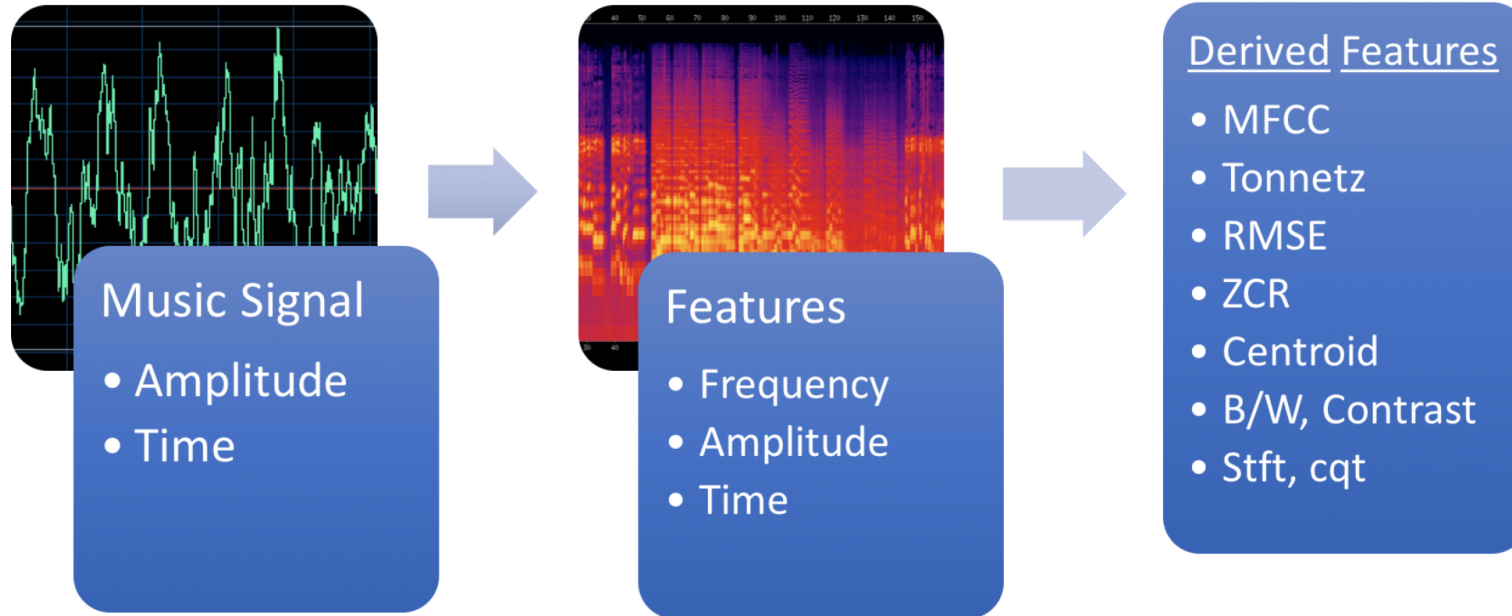


Goal: Music Classification





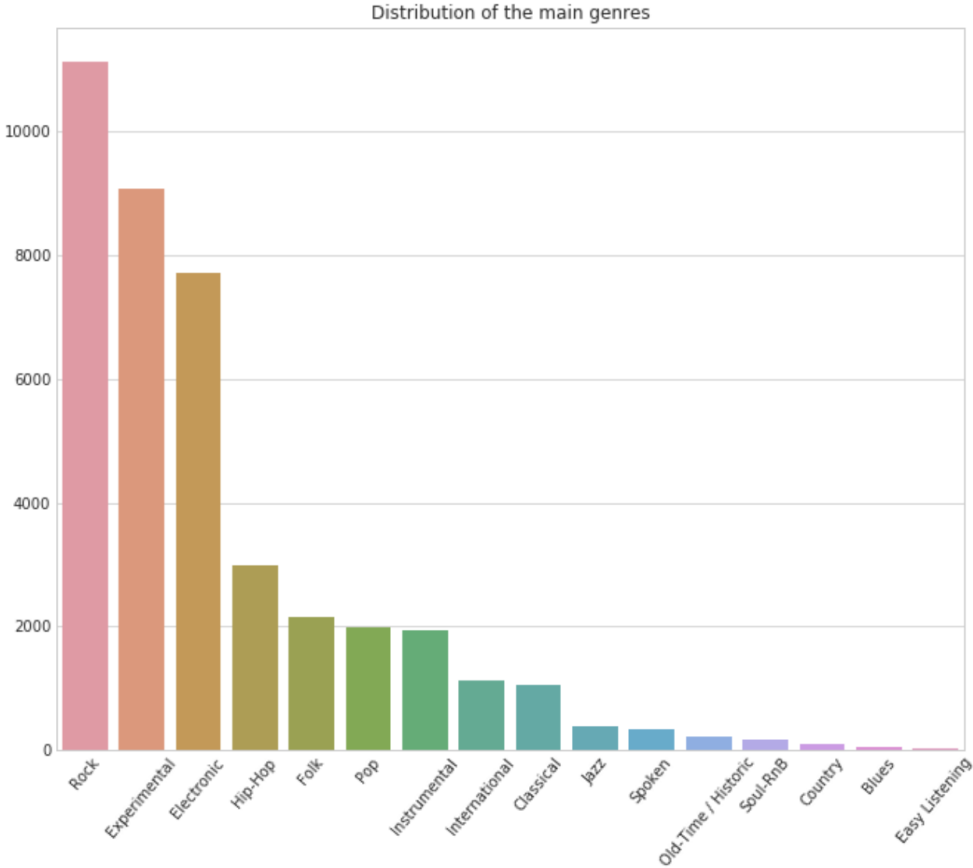
Features



- 7 sub_features -> Mean, Median, Min, Max, Std.Dev, Skew and Kurtosis

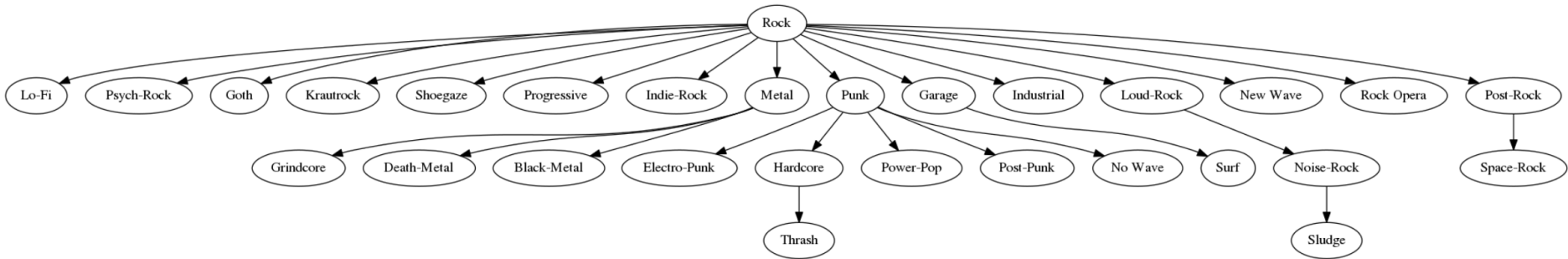


Dataset: 91213 songs, 163 genres , 15 main genres, 522 features





Genre and Sub-Genre





Methodology

We have used five approaches to predict Genres:

- 1.1 KNN
- 1.2 Random Forest
- 1.3 Gradient Boosting
- 1.4 Neural Network
- 2.0 Multilabel Classification





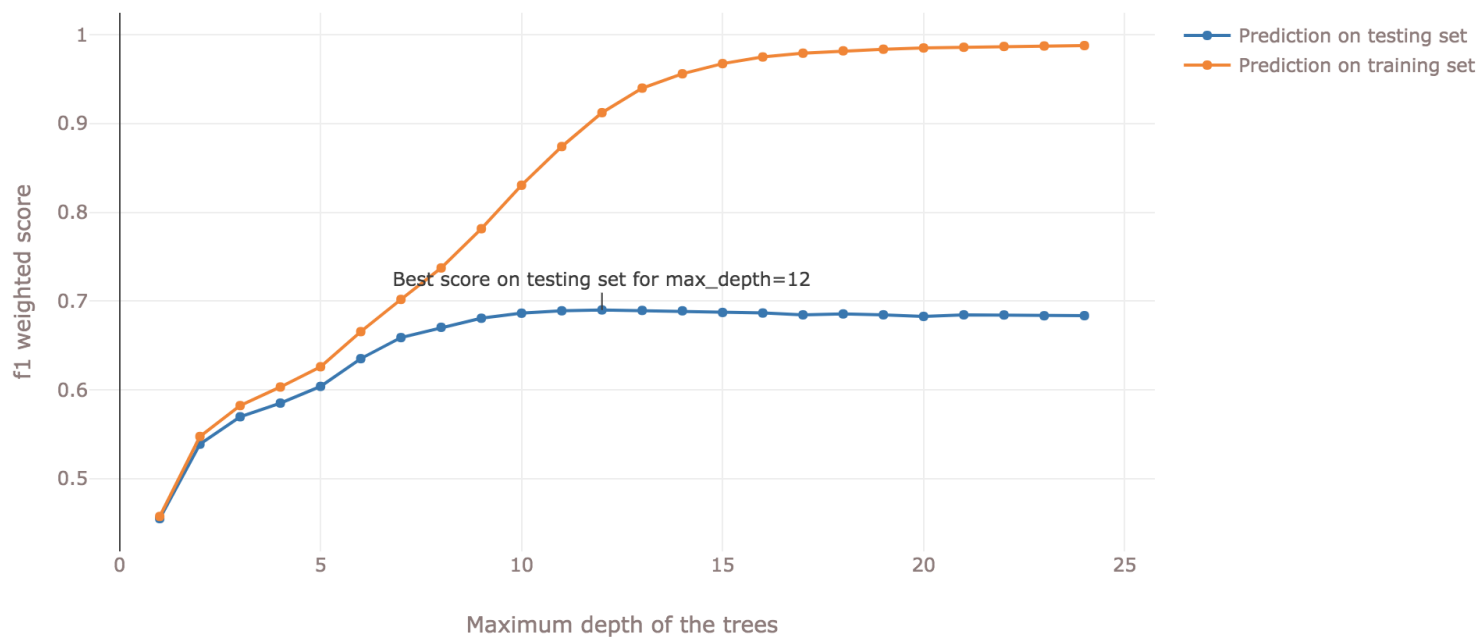
KNN, Random Forest, Gradient Boost





```
In [7]: fig = Figure(data=data, layout=layout)
        plotly.offline.iplot(fig)
```

Score of our model as a function of the maximum depth

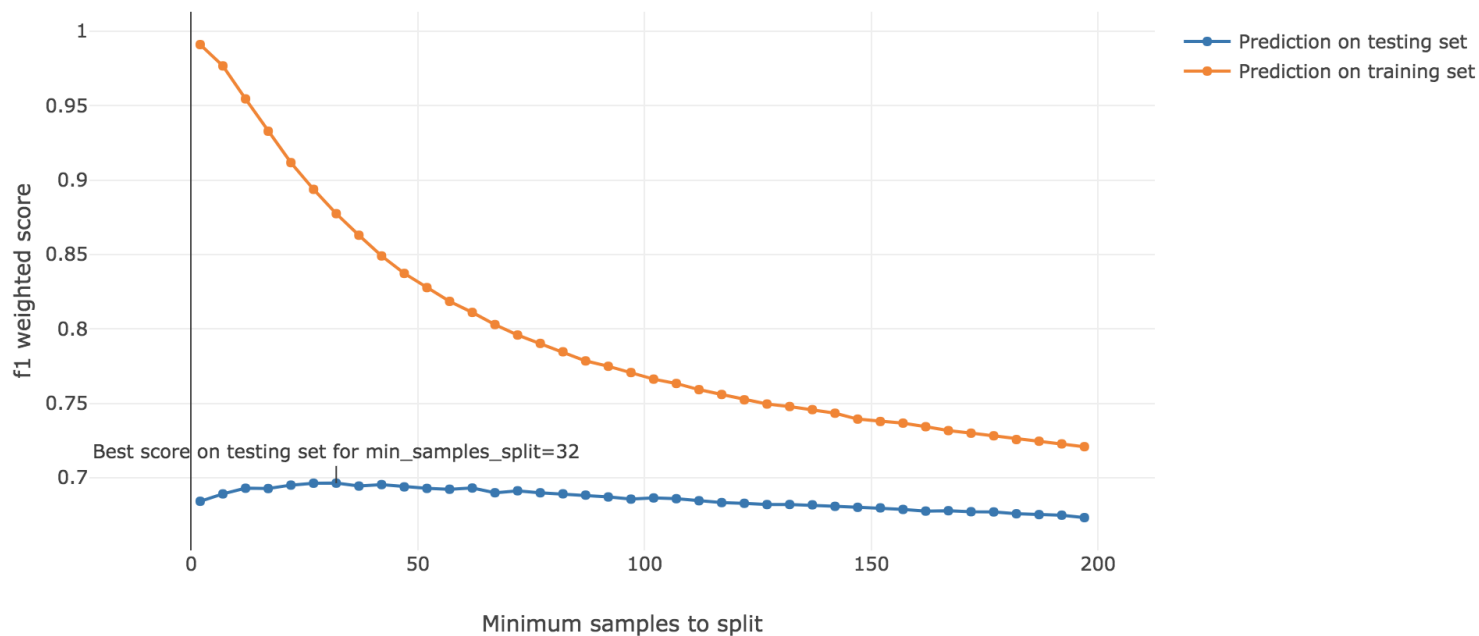


[Export to plot.ly »](#)



```
In [5]: fig = Figure(data=data, layout=layout)
        plotly.offline.iplot(fig)
```

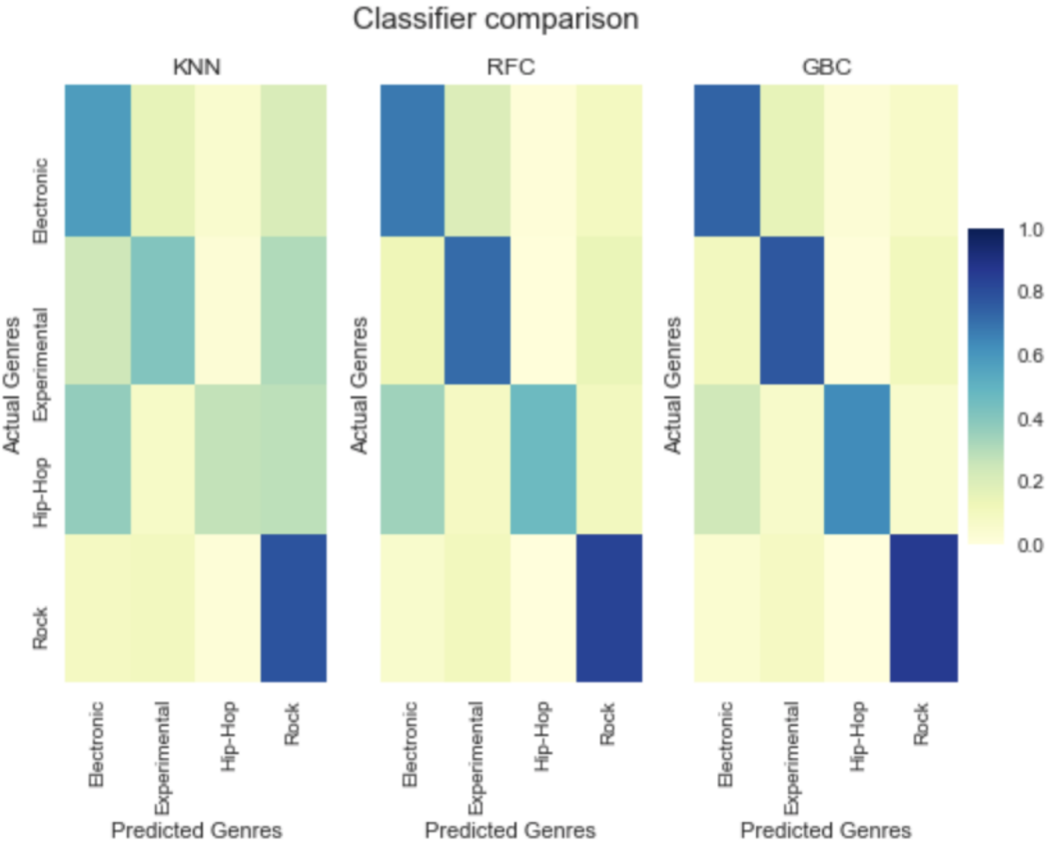
Score of our model as a function of minimum samples to split



[Export to plot.ly »](#)



Results: KNN, Random Forest, Gradient Boost



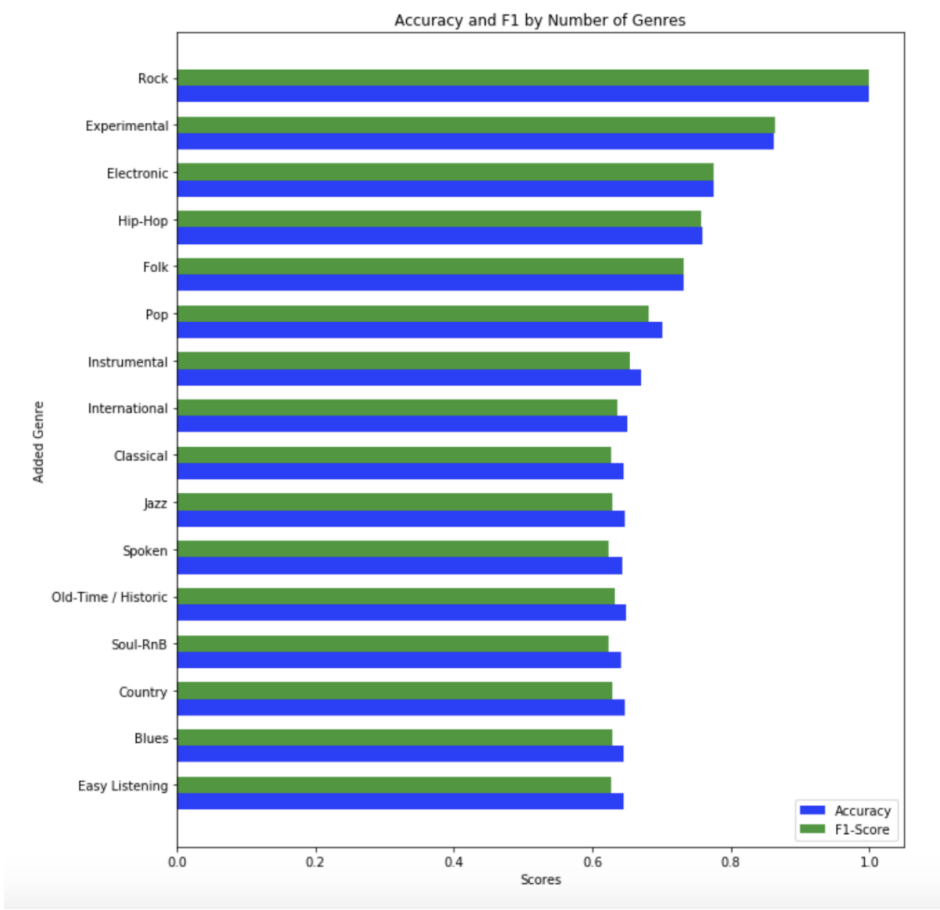


Neural Networks





Accuracy





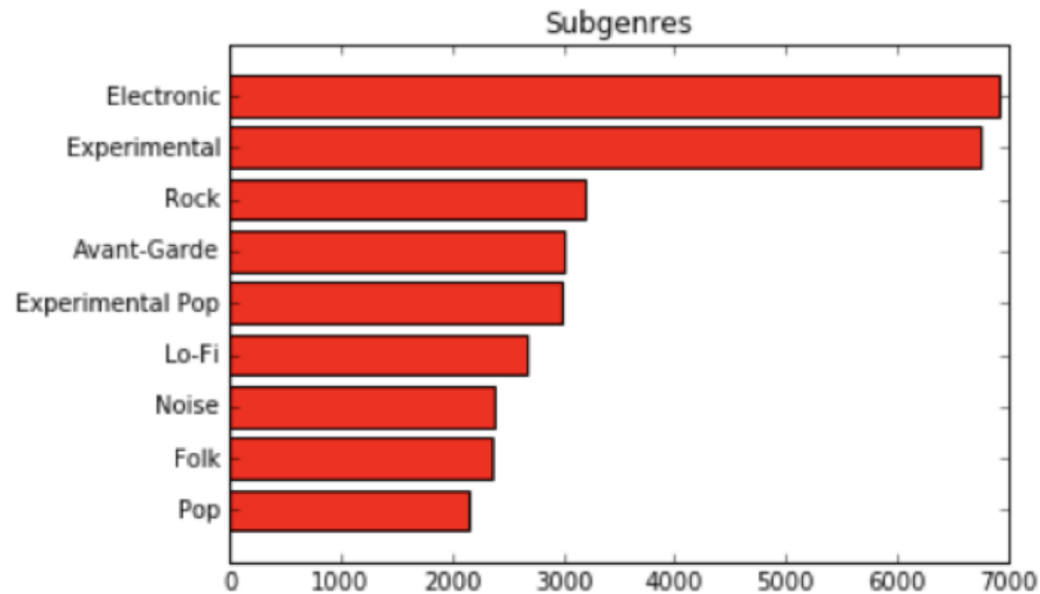
Multilabel classification





Multilabel classification

- One vs Rest classification
- Chain classification





Chain Classification

The order in which the y's are predicted impacts overall accuracy of the prediction. We tried three different approaches:

- sorting the y's by increasing number of combinations they appear in
- sorting the y's by decreasing number of combinations they appear in
- randomly sorting the y's

There is no significant difference in accuracy between these three options on our data!



Results

	F1_score
KNN_OneVsRest	0.46
KNN_Chain	0.46
RF_OneVsRest	0.31
RF_Chain	0.33
NN_OneVsRest	0.41
NN_Chain	0.44



Results: RFC vs NN

	Labels	RFC_recall	NN_recall
0	Avant-Garde	0.051102	0.314629
1	Electronic	0.428692	0.570884
2	Experimental	0.341001	0.487596
3	Experimental Pop	0.041000	0.320000
4	Folk	0.148489	0.445466
5	Lo-Fi	0.210349	0.431946
6	Noise	0.132147	0.324015
7	Pop	0.064103	0.326211
8	Rock	0.208571	0.514286

	Labels	RFC_precision	NN_precision
0	Avant-Garde	0.392308	0.288338
1	Electronic	0.709384	0.595848
2	Experimental	0.540386	0.484319
3	Experimental Pop	0.414141	0.279720
4	Folk	0.624309	0.405018
5	Lo-Fi	0.834821	0.381710
6	Noise	0.611765	0.331169
7	Pop	0.608108	0.302111
8	Rock	0.725166	0.507042



Conclusion

- Used extracted features from .mp3 files for genre classification
- MFCC and spectral_contrast are top features
- Trained and tested various models
- Achieved ~80% accuracy for 4-Genre classification
- Achieved ~65% accuracy for 16-Genre classification
- Achieved ~45% accuracy for multilabel classification with 9 Genres



Way Forward

- Improve the tuning of the presented algorithms
- Extract features from lyrics
- Sentiment analysis of songs
- Extract features of actual MP3 files in order to run the models on our favourite songs!
- Diversify metric choice and choose an efficient model for each metric

