Case Study Taylor Swift vs Biggie Small

May 11, 2020

0.1 Setup

Import necessary modules:

[1]:
```
%matplotlib inline

import matplotlib.pyplot as plt
import seaborn

import librosa
import librosa.display

from IPython.display import Audio

import requests
import os

import sklearn
import numpy as np

import pandas as pd
```

[2]:
```
t1, t1_sr = librosa.load("audio/taylor.mp3",duration=120)
t2, t2_sr = librosa.load("audio/taylor2.mp3",duration=120)
t3, t3_sr = librosa.load("audio/taylor3.mp3",duration=120)
b1, b1_sr = librosa.load("audio/big.mp3",duration=120)
b2, b2_sr = librosa.load("audio/big2.mp3",duration=120)
b3, b3_sr = librosa.load("audio/big3.mp3",duration=120)
```

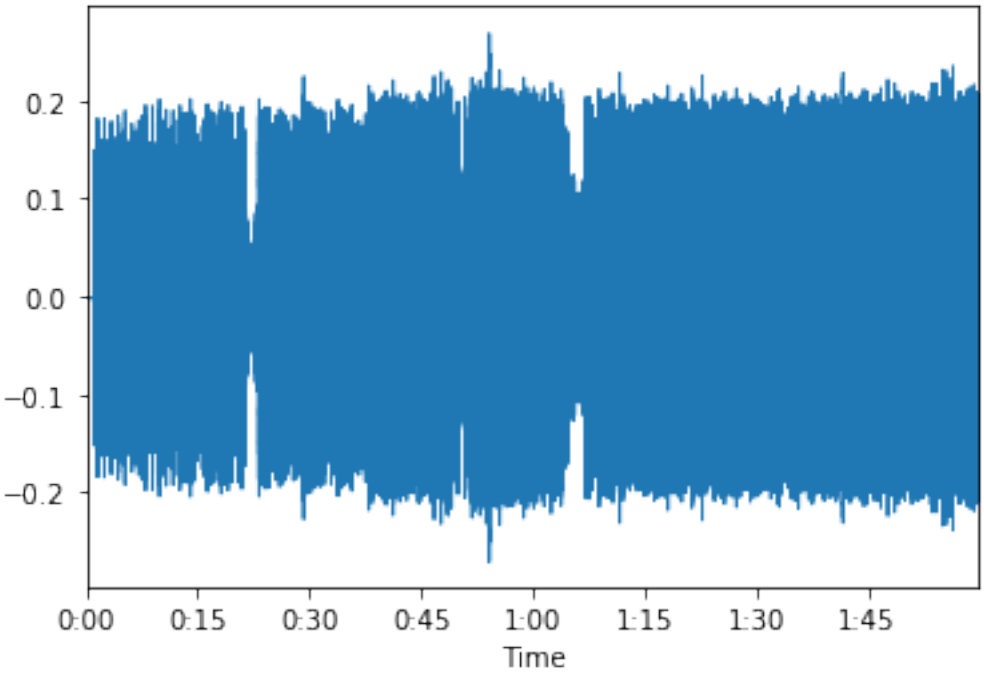
```
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
```

```
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
```

1 Taylor Swift three sound samples: WavePlots

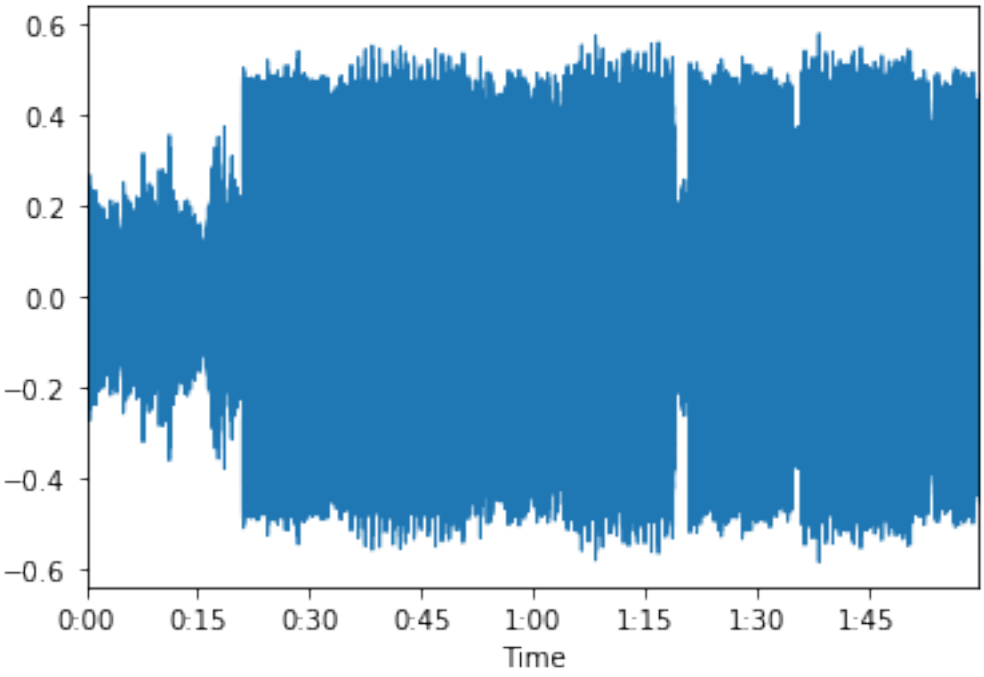
[3]:
```
librosa.display.waveplot(t1,sr=t1_sr)
```

[3]:
```
<matplotlib.collections.PolyCollection at 0x7fb51687a358>
```



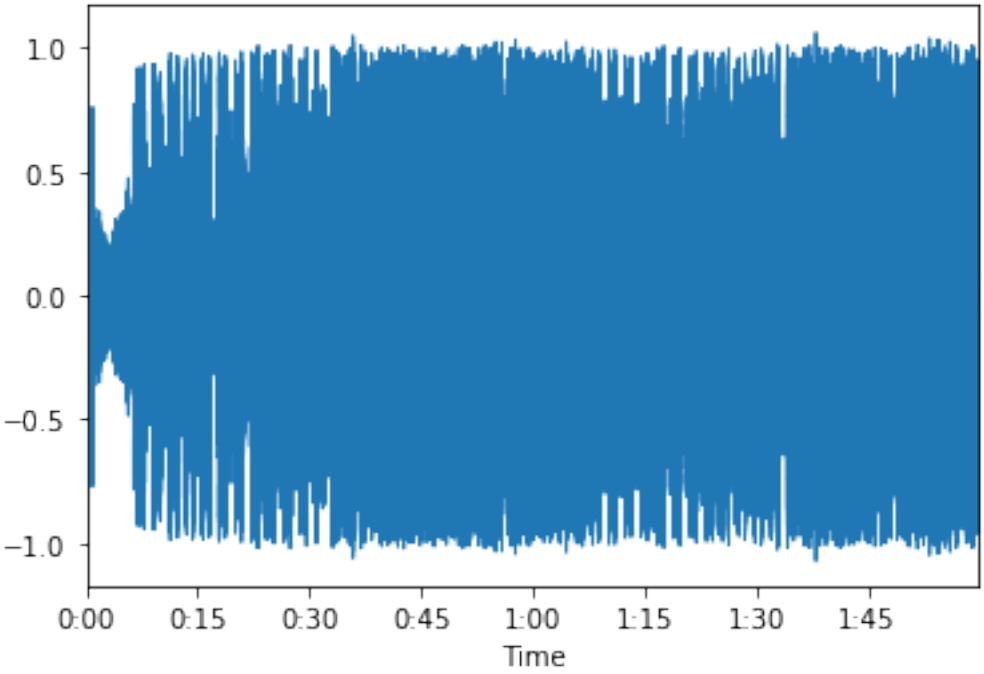
[4]:
```
librosa.display.waveplot(t2,sr=t2_sr)
```

[4]:
```
<matplotlib.collections.PolyCollection at 0x7fb516805cc0>
```



[5]:
```
librosa.display.waveplot(t3,sr=t3_sr)
```

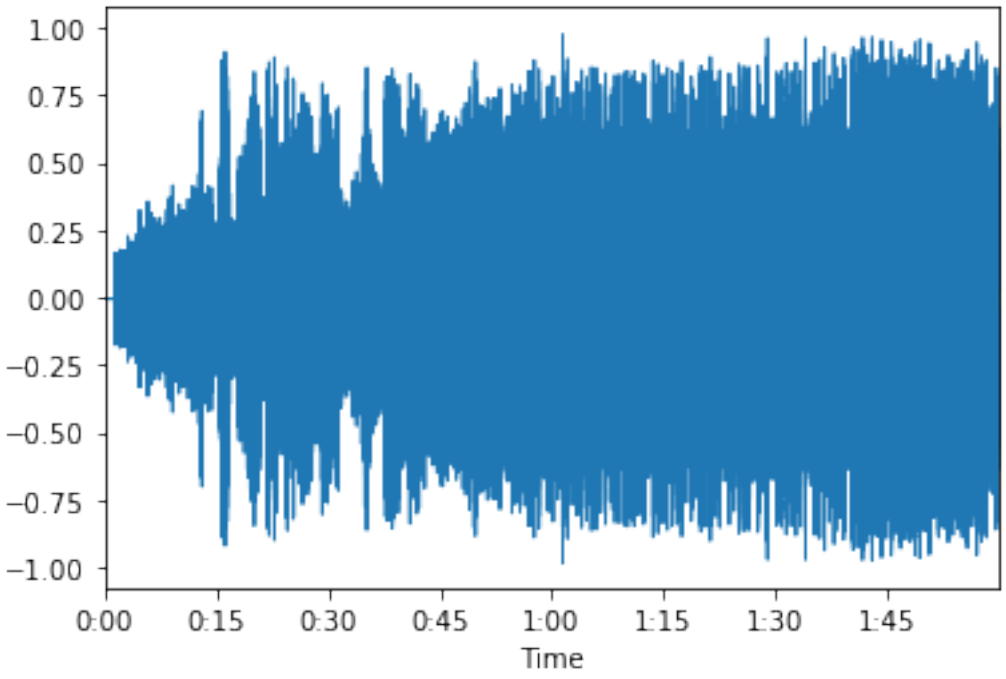
[5]: <matplotlib.collections.PolyCollection at 0x7fb51676cdd8>



2 Biggie Small three sound samples

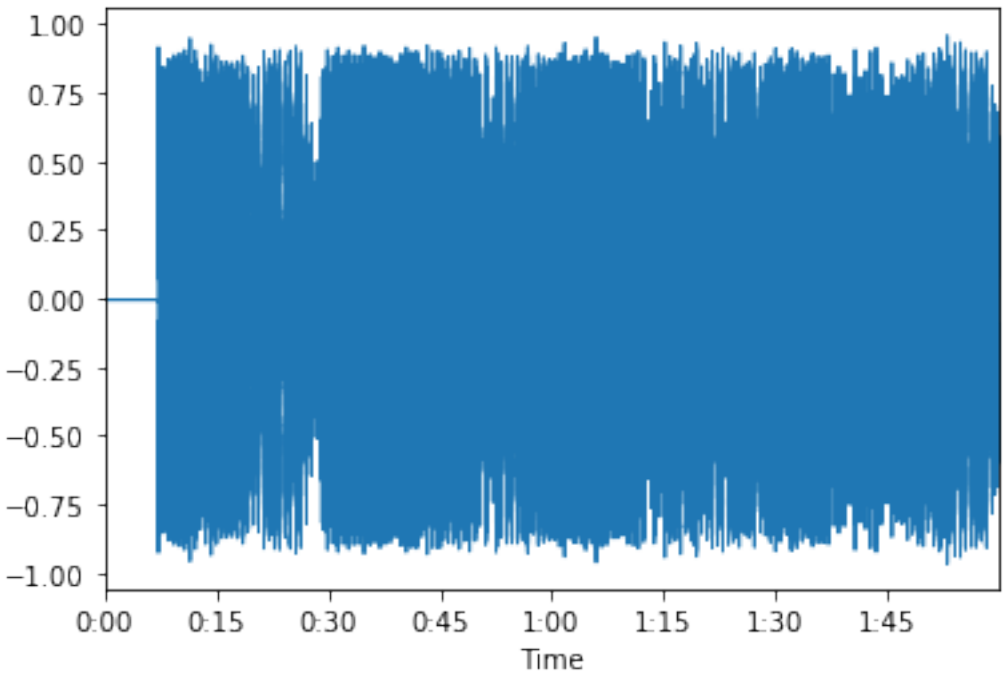
[6]:
```
librosa.display.waveplot(b1,sr=b1_sr)
```

[6]:
```
<matplotlib.collections.PolyCollection at 0x7fb51674af98>
```



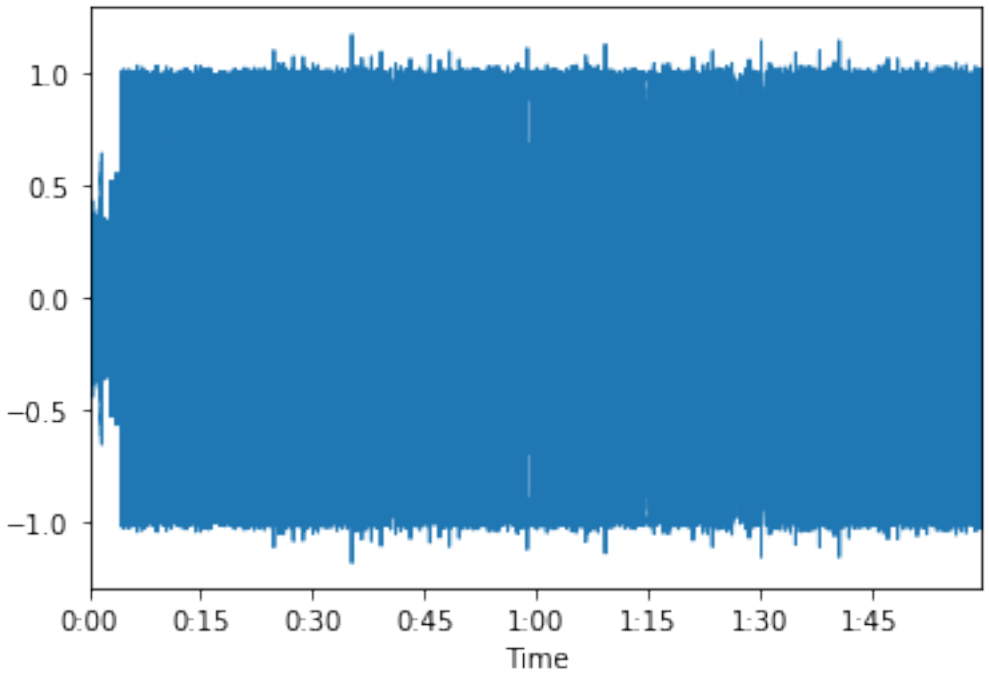
[7]:
```
librosa.display.waveplot(b2,sr=b2_sr)
```

[7]:
```
<matplotlib.collections.PolyCollection at 0x7fb5166b8668>
```



[9]:
```
librosa.display.waveplot(b3,sr=b3_sr)
```

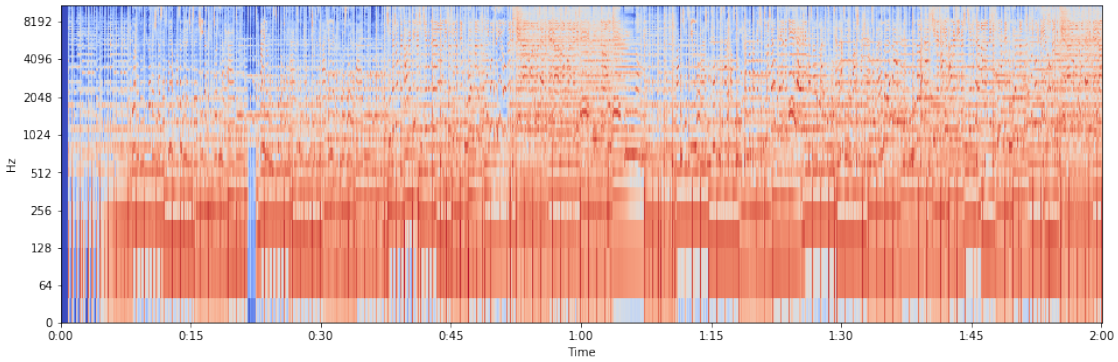
[9]:
```
<matplotlib.collections.PolyCollection at 0x7fb5165b64e0>
```

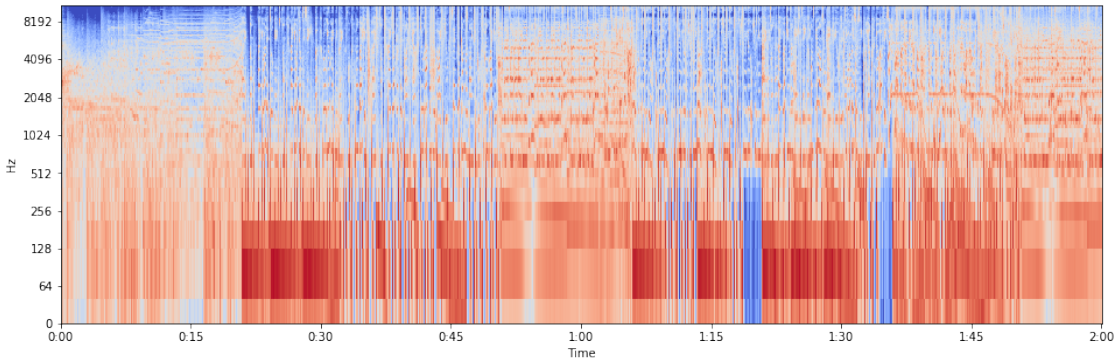


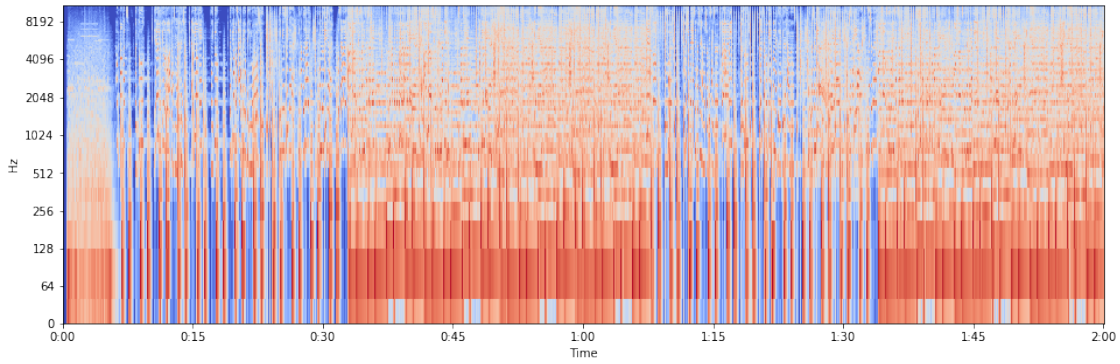
[10]:
```
def draw(y,sr):
    # calculate
    melspe = librosa.feature.melspectrogram(y=y, sr=sr)
    logspect = librosa.power_to_db(abs(melspe))

    # display
    plt.figure(figsize=(16, 5))
    librosa.display.specshow(logspect, sr=sr, x_axis='time',y_axis='log')
    plt.show()
```

3 Taylor Swift three sound samples: logspect

[13]:
```
draw(t1,t1_sr)
draw(t2,t2_sr)
draw(t3,t3_sr)
```

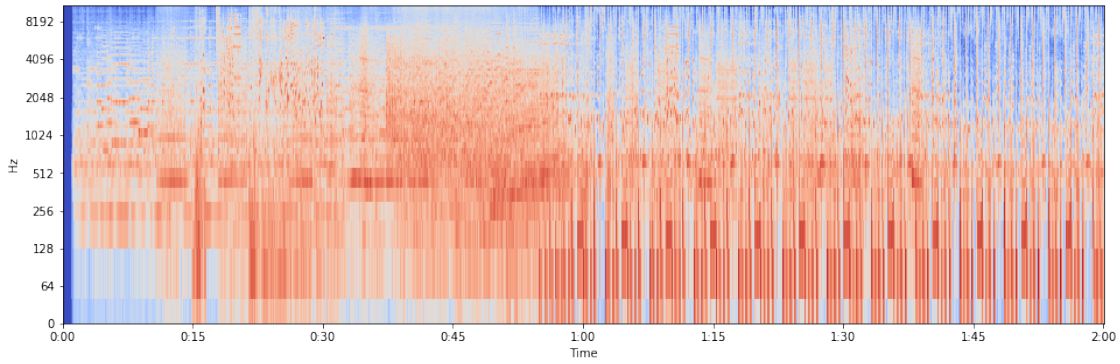


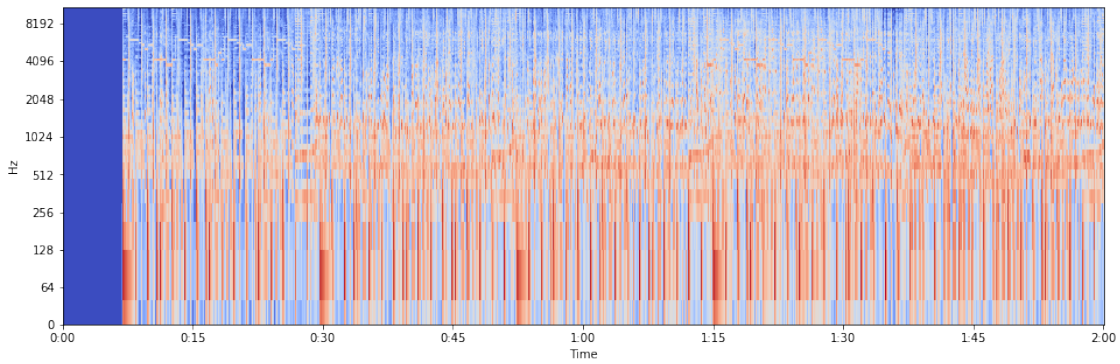


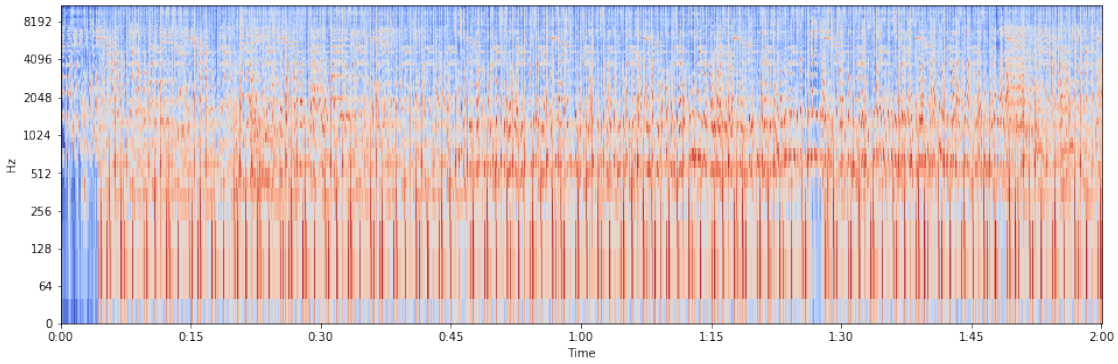


4 Biggie Small three sound samples: logspect

[14]:
```
draw(b1,b1_sr)
draw(b2,b2_sr)
draw(b3,b3_sr)
```







[36]:
```
def model(y,sr,y2,sr2):
    mfcc = librosa.feature.mfcc(y, sr=sr, n_mfcc= 12)
    mfcc = mfcc.T
    scaler = sklearn.preprocessing.StandardScaler()
    scaler.fit(mfcc)
    mfcc_scaled = scaler.transform(mfcc)
    mfcc2 = librosa.feature.mfcc(y2, sr=sr2, n_mfcc= 12)
    mfcc2 = mfcc2.T
    scaler.fit(mfcc2)
    mfcc2_scaled = scaler.transform(mfcc2)
    features = np.vstack((mfcc_scaled,mfcc2_scaled))
    labels = np.concatenate((np.zeros(len(mfcc_scaled)), np.
↪ones(len(mfcc2_scaled))))
    model = sklearn.svm.SVC()
    model.fit(features, labels)
    return model
```

[29]:
```
def result(y,sr,y2,sr2,model):
    scaler = sklearn.preprocessing.StandardScaler()
    mfcc1 = librosa.feature.mfcc(y, sr=sr, n_mfcc= 12).T
    mfcc2 = librosa.feature.mfcc(y2, sr=sr2, n_mfcc= 12).T
    scaler.fit(mfcc1)
    mfcc1_scaled = scaler.transform(mfcc1)
    scaler.fit(mfcc2)
    mfcc2_scaled = scaler.transform(mfcc2)
    test_features = np.vstack((mfcc1_scaled, mfcc2_scaled))
    test_labels = np.concatenate((np.zeros(len(mfcc1_scaled)), np.
↪ones(len(mfcc2_scaled))))
    predicted_labels = model.predict(test_features)
    return model.score(test_features,test_labels)
```

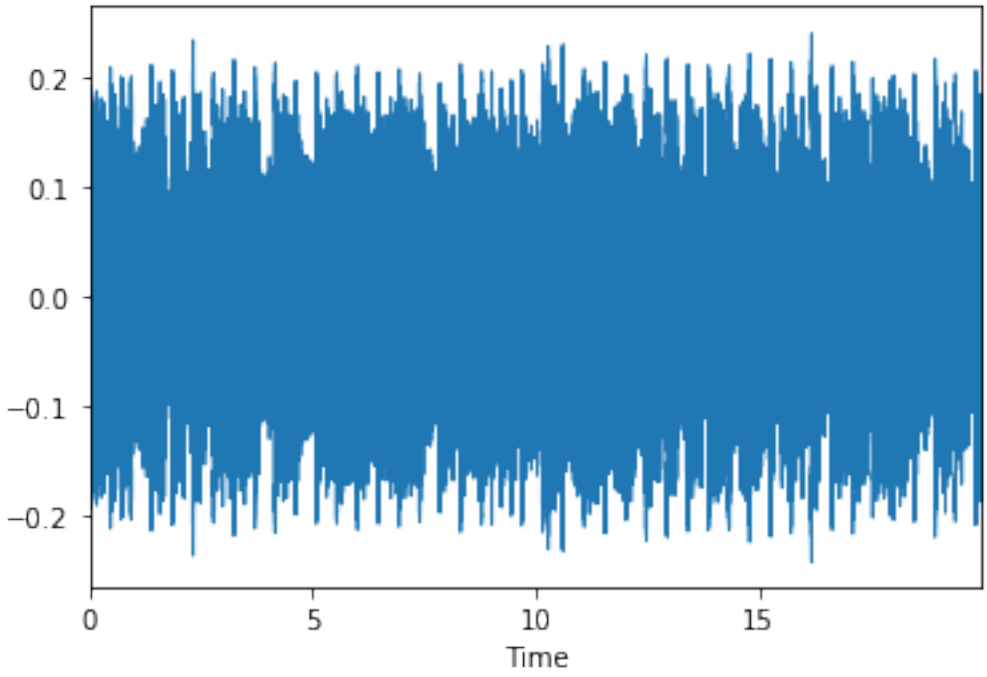
[17]:
```
t1t, t1t_sr = librosa.load("audio/taylor.mp3",duration=20,offset=120)
t2t, t2t_sr = librosa.load("audio/taylor2.mp3",duration=20,offset=120)
t3t, t3t_sr = librosa.load("audio/taylor3.mp3",duration=20,offset=120)
b1t, b1t_sr = librosa.load("audio/big.mp3",duration=20,offset=120)
b2t, b2t_sr = librosa.load("audio/big2.mp3",duration=20,offset=120)
b3t, b3t_sr = librosa.load("audio/big3.mp3",duration=20,offset=120)
```

```
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
/home/yizong/.local/lib/python3.7/site-packages/librosa/core/audio.py:161:
UserWarning: PySoundFile failed. Trying audioread instead.
  warnings.warn('PySoundFile failed. Trying audioread instead.')
```

5 Taylor Swift test files

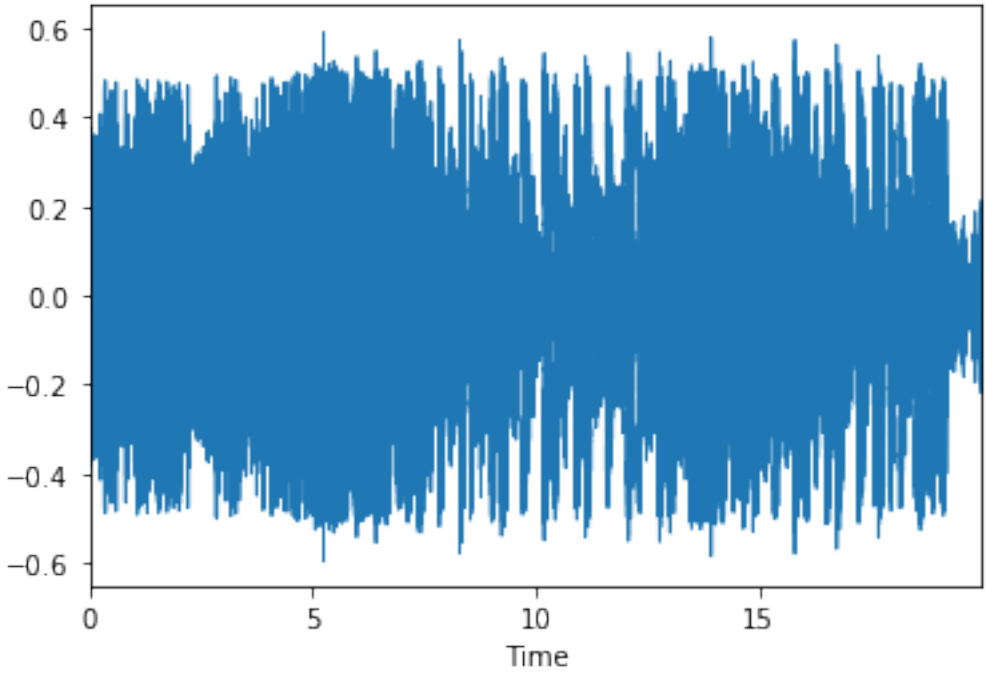
[18]:
```
librosa.display.waveplot(t1t,sr=t1t_sr)
```

[18]:
```
<matplotlib.collections.PolyCollection at 0x7fb51653cba8>
```



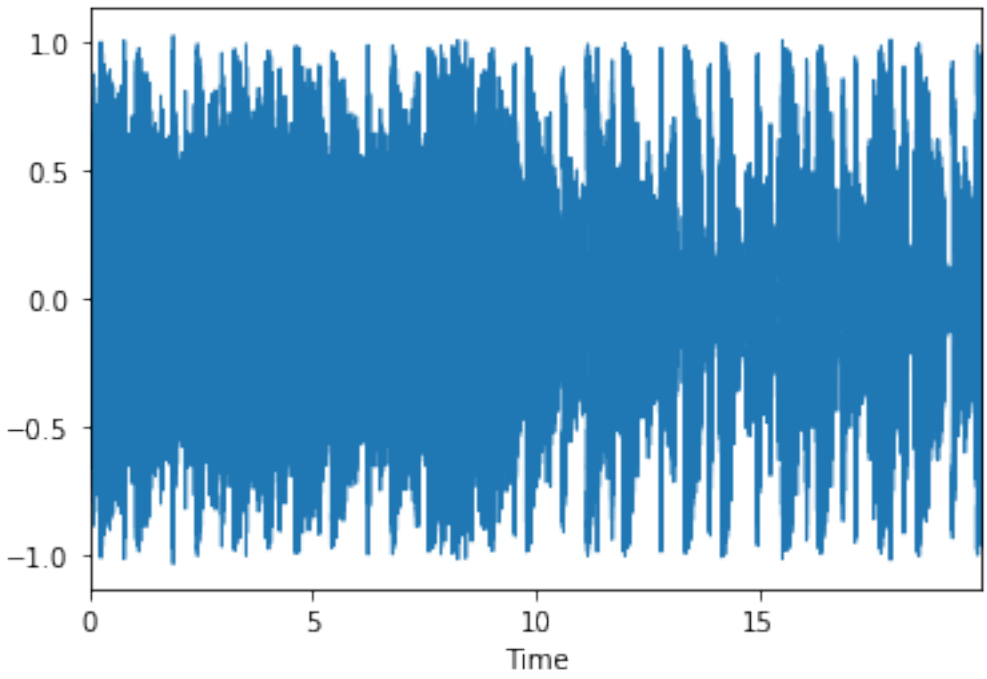
[19]:
```
librosa.display.waveplot(t2t,sr=t2t_sr)
```

[19]:
```
<matplotlib.collections.PolyCollection at 0x7fb516909dd8>
```



[20]:
```
librosa.display.waveplot(t3t,sr=t3t_sr)
```

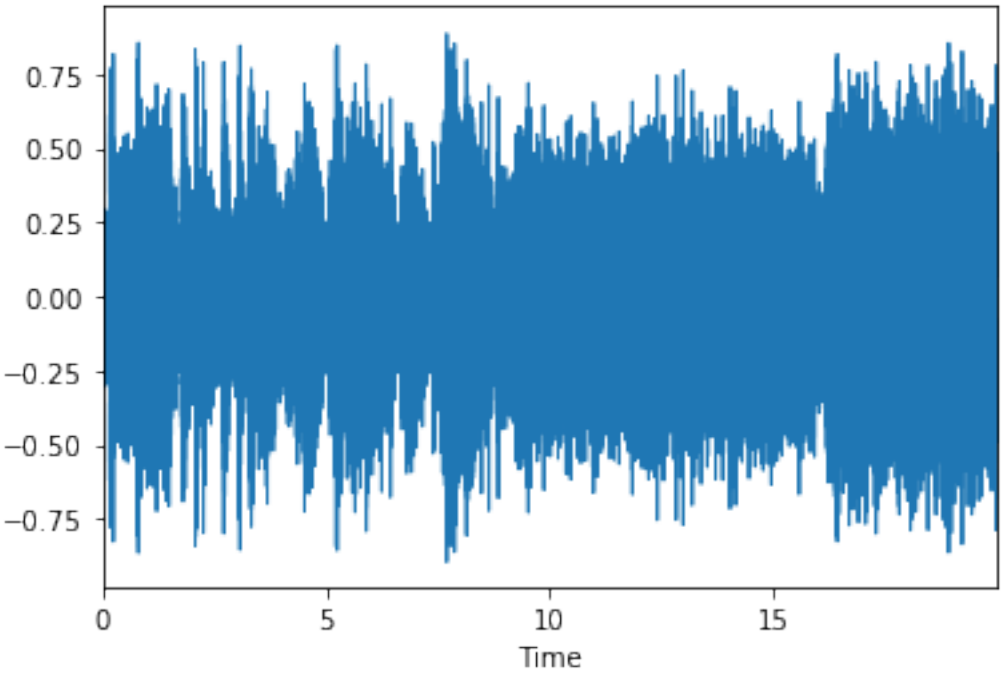
[20]:
```
<matplotlib.collections.PolyCollection at 0x7fb516c0c128>
```



6 Biggie Small test files

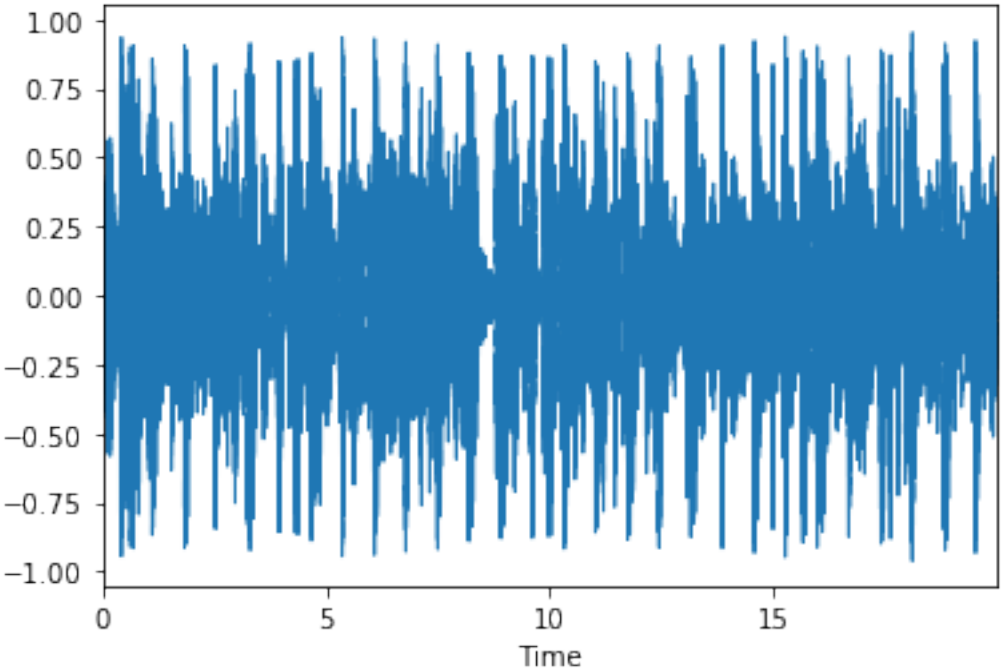
[23]:
```
librosa.display.waveplot(b1t,sr=b1t_sr)
```

[23]:
```
<matplotlib.collections.PolyCollection at 0x7fb51698f1d0>
```



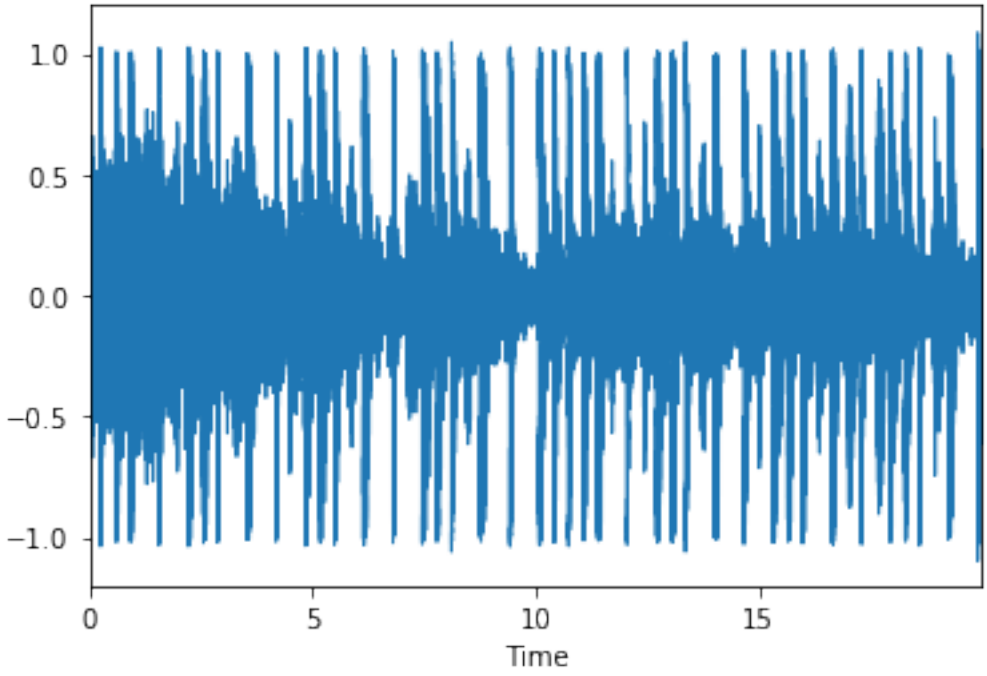
[24]:
```
librosa.display.waveplot(b2t,sr=b2t_sr)
```

[24]:
```
<matplotlib.collections.PolyCollection at 0x7fb5163a1828>
```



[25]:
```
librosa.display.waveplot(b3t,sr=b3t_sr)
```

[25]:
```
<matplotlib.collections.PolyCollection at 0x7fb516ba1940>
```



7 Getting Results

[32]:
```
taylor_train, taylor_test = [(t1,t1_sr),(t2,t2_sr),(t3,t3_sr)],␣
↪[(t1t,t1t_sr),(t2t,t2t_sr),(t3t,t3t_sr)]
biggie_train, biggie_test = [(b1,b1_sr),(b2,b2_sr),(b3,b3_sr)],␣
↪[(b1t,b1t_sr),(b2t,b2t_sr),(b3t,b3t_sr)]
```

8 Compare to oneself: Taylor Swift

[42]:
```
accuracy_t2t = []
for i in range(len(taylor_train)):
    for j in range(i+1,len(taylor_train)):
        accuracy_t2t.
↪append(result(*taylor_test[i],*taylor_test[j],model(*taylor_train[i],*taylor_train[j])))
accuracy_t2t
```

[42]:
```
[0.744199535962877, 0.8010440835266821, 0.7900232018561485]
```

[43]:
```
np.mean(accuracy_t2t)
```

[43]:
```
0.7784222737819025
```

9 Compare to oneself: Biggie Small

[41]:
```
accuracy_b2b = []
for i in range(len(biggie_train)):
    for j in range(i+1,len(biggie_train)):
        accuracy_b2b.
 ↪append(result(*biggie_test[i],*biggie_test[j],model(*biggie_train[i],*biggie_train[j])))
accuracy_b2b
```

[41]:
```
[0.7198375870069605, 0.759860788863109, 0.8080046403712297]
```

[44]:
```
np.mean(accuracy_b2b)
```

[44]:
```
0.7625676720804332
```

10 Compare to Each other

[45]:
```
accuracy_b2t = []
for i in range(len(biggie_train)):
    for j in range(len(taylor_train)):
        accuracy_b2t.
 ↪append(result(*biggie_test[i],*taylor_test[j],model(*biggie_train[i],*taylor_train[j])))
accuracy_b2t
```

[45]:
```
[0.6403712296983759,
 0.7209976798143851,
 0.8288863109048724,
 0.7529002320185615,
 0.7911832946635731,
 0.8451276102088167,
 0.7140371229698376,
 0.8155452436194895,
 0.8938515081206496]
```

[46]:
```
np.mean(accuracy_b2t)
```

[46]:
```
0.7781000257798402
```

11 Discussion of Results

For comparison within Taylor's music, we didn't find extreme similarity in between, which means Taylor makes some variation in her composition styles. However, since the classifer didn't show high score, we can infer that there is a underlying similarity inbetween.

For the same reason, we can say Biggie's music is also homogeneous. If we assume these two figures are representative of Pop and Hip Hop, we might lean towards the idea that there are not much difference in the extent of styles variation between Pop and Hip Hop. It would also be interesting to see new generation Hip Hop artists since Biggie was representative of old style of Hip Hop.

For comparison between two artists, we found that for every song of Biggie, the classifier shows distinction between it and Taylor's new Hit "The Man". This is very interesting since it implies both the difference between Pop and Hip Hop and the difference between different generations.

[ ]: