Final Capstone Project

Analyze song lyrics, determine its sentiment and recommend songs.

By Karen McGee

Problem Statement

- Perform text analysis of song lyrics to classify differences between songs and sentiment analysis.
  - Based on the lyrics can the model determine the type of mood the song?
  - Based on the mood of the song can the model recommend other songs?

Research Questions:

- Which genre have the highest positive and/or negative sentiment analysis?
- What is the word mean distribution for a given sentiment label and genre?
- What are the top 10 words that could summarize the document topic

Solution statement

- Acquire and explore dataset
- Preprocess the dataset and associate sentiment analysis
- Generate features utilizing the process data
- Analyze the word topics based on features
- Generate a song recommender algorithm
- Model the data and evaluate the performance

Evaluation of Sentiment analyzer and Models

- Review samples of songs identified as positive or negative and determine if the song is labeled correctly.
- Use confusion matrix, classification report and accuracy score to evaluate the performance of each model.

Analysis – example text file

Example of text file that will be used in our models.

	index	song	year	artist	genre	lyrics	clean_lyrics	lemmatized	lemmatized_features	sentiment score
0	0	ego-remix	2009	beyonce-knowles	Pop	Oh baby, how you doing? \nYou know I'm gonna cu...	Oh baby, how you doing? You know I'm gonna cut...	Oh baby , how you do ? You know I 'm gon na cu...	Oh baby how you doing You know Im gonna cut ri...	{'neg': 0.077, 'neu': 0.7, 'pos': 0.223, 'comp...
1	1	then-tell-me	2009	beyonce-knowles	Pop	playin' everything so easy,\nit's like you see...	playin' everything so easy, it's like you seem...	playin ' everything so easy , it 's like you s...	playin everything so easy its like you seem so...	{'neg': 0.075, 'neu': 0.783, 'pos': 0.142, 'co...
2	2	honesty	2009	beyonce-knowles	Pop	If you search\nFor tenderness\nIt isn't hard t...	If you search For tenderness It isn't hard to ...	If you search For tenderness It be n't hard to...	If you search For tenderness It isnt hard to f...	{'neg': 0.09, 'neu': 0.685, 'pos': 0.225, 'com...
3	3	you-are-my-rock	2009	beyonce-knowles	Pop	Oh oh oh I, oh oh oh I\n[Verse 1:]\nIf I wrote...	Oh oh oh I, oh oh oh I [Verse 1:] If I wrote a...	Oh oh oh I , oh oh oh I [Verse 1 :] If I wri...	Oh oh oh I oh oh oh I If I wrote a book about ...	{'neg': 0.017, 'neu': 0.728, 'pos': 0.255, 'co...
4	4	black-culture	2009	beyonce-knowles	Pop	Party the people, the people the	Party the people, the people the party it's	Party the people , the people the	Party the people the people the party its popp...	{'neg': 0.038, 'neu': 0.888, 'pos':

Analysis – Sentiment Analysis text file cleaned, lemmatized, and tokenized

"Oh baby , how you do ? You know I 'm gon na cut right to the chase Some women be make but me
, myself I like to think that I be create for a special purpose You know , what 's more specia
l than you ? You feel me It 's on baby , let 's get lose You do n't need to call into work 'ca
use you 're the boss For real , want you to show me how you feel I consider myself lucky , tha
t 's a big deal Why ? Well , you get the key to my heart But you ai n't gon na need it , I 'd
rather you open up my body And show me secrets , you do n't know be inside No need for me to l
ie It 's too big , it 's too wide It 's too strong , it wo n't fit It 's too much , it 's too
tough He talk like this 'cause he can back it up He get a big ego , such a huge ego I love his
big ego , it 's too much He walk like this 'cause he can back it up Usually I 'm humble , righ
t now I do n't choose You can leave with me or you could have the blue Some call it arrogant ,
I call it confident You decide when you find on what I 'm work with Damn I know I 'm kill you
with them legs Better yet them thighs Matter a fact it 's my smile or maybe my eye Boy you a s
ite to see , kind of something like me It 's too big , it 's too wide It 's too strong , it wo
n't fit It 's too much , it 's too tough I talk like this 'cause I can back it up I get a big
ego , such a huge ego But he love my big ego , it 's too much I walk like this 'cause I can ba
ck it up I , I walk like this 'cause I can back it up I , I talk like this 'cause I can back i
t up I , I can back it up , I can back it up I walk like this 'cause I can back it up It 's to
o big , it 's too wide It 's too strong , it wo n't fit It 's too much , it 's too tough He ta
lk like this 'cause he can back it up He get a big ego , such a huge ego , such a huge ego I l
ove his big ego , it 's too much He walk like this 'cause he can back it up Ego so big , you m
ust admit I get every reason to feel like I 'm that bitch Ego so strong , if you ai n't know I
do n't need no beat , I can sing it with piano"

Analysis – Text file cleaned, lemmatized, tokenized, and used for models

'Oh baby how you doing You know Im gonna cut right to the chase Some women were made but me my self I like to think that I was created for a special purpose You know whats more special than you You feel me Its on baby lets get lost You dont need to call into work cause youre the boss For real want you to show me how you feel I consider myself lucky thats a big deal Why Well yo u got the key to my heart But you aint gonna need it Id rather you open up my body And show me secrets you didnt know was inside No need for me to lie Its too big its too wide Its too stron g it wont fit Its too much its too tough He talk like this cause he can back it up He got a bi g ego such a huge ego I love his big ego its too much He walk like this cause he can back it u p Usually Im humble right now I dont choose You can leave with me or you could have the blues Some call it arrogant I call it confident You decide when you find on what Im working with Dam n I know Im killing you with them legs Better yet them thighs Matter a fact its my smile or ma ybe my eyes Boy you a site to see kind of something like me Its too big its too wide Its too s trong it wont fit Its too much its too tough I talk like this cause I can back it up I got a b ig ego such a huge ego But he love my big ego its too much I walk like this cause I can back i t up I I walk like this cause I can back it up I I talk like this cause I can back it up I I c an back it up I can back it up I walk like this cause I can back it up Its too big its too wid e Its too strong it wont fit Its too much its too tough He talk like this cause he can back it up He got a big ego such a huge ego such a huge ego I love his big ego its too much He walk li ke this cause he can back it up Ego so big you must admit I got every reason to feel like Im t hat bitch Ego so strong if you aint know I dont need no beat I can sing it with piano'

Sentiment Analysis - Vader

VADER produces four sentiment metrics from these word ratings based on lexicons of sentiment-related words.

sentiment_label_code	sentiment_label	song	artist	sentiment score	compound	neg	neu	pos
1	Positive	ego-remix	beyonce-knowles	{'neg': 0.077, 'neu': 0.7, 'pos': 0.223, 'comp...	0.9978	0.077	0.700	0.223
1	Positive	then-tell-me	beyonce-knowles	{'neg': 0.075, 'neu': 0.783, 'pos': 0.142, 'co...	0.9561	0.075	0.783	0.142
1	Positive	honesty	beyonce-knowles	{'neg': 0.09, 'neu': 0.685, 'pos': 0.225, 'com...	0.9819	0.090	0.685	0.225
1	Positive	you-are-my-rock	beyonce-knowles	{'neg': 0.017, 'neu': 0.728, 'pos': 0.255, 'co...	0.9993	0.017	0.728	0.255
1	Positive	black-culture	beyonce-knowles	{'neg': 0.038, 'neu': 0.888, 'pos': 0.074, 'co...	0.8659	0.038	0.888	0.074
-1	Negative	all-i-could-do-was-cry	beyonce-knowles	{'neg': 0.187, 'neu': 0.716, 'pos': 0.097, 'co...	-0.9153	0.187	0.716	0.097
1	Positive	once-in-a-lifetime	beyonce-knowles	{'neg': 0.015, 'neu': 0.657, 'pos': 0.328, 'co...	0.9990	0.015	0.657	0.328

Sentiment Analysis – Distribution of Vader compound score

Higher frequency of positive words than negative.

Sentiment Analysis by genre

Rock – has the highest positive and negative sentiment analysis
Pop – has the second highest positive sentiment analysis
Mental – has the second highest negative sentiment followed by Hip-Hop

Positive - Sentiment Analysis by artist “The Dells”

The songs stay in my corner and oh what a night are two highly rated positive songs.

Negative - Sentiment Analysis by artist “2 Pac”

The songs Homeboyz and High-Speed are two highly rated negative songs.

Sentiment Analysis by artist and song total

Relationship of the sentiment analysis between two random select artist Beyonce and 112 and there total count of songs.

Sentiment Analysis – Review by word total

- Added a new variable to capture the correlation between the total word count and sentiment analysis.

Sentiment Analysis – Average distribution of words by genre

Hip Hop has the highest frequency of words for both positive and negative sentiment analysis.

Analysis – Importance of words based on Frequency

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance.

Beyonc
Ay
Oh
Im
waiting
know
say
Nobody
likes
played
heard
people
gotta
baby
Love
Look
Hey
isnt
choir
London
Beyonce
Growl
Youre
Welcome
Party
Youve
spitting
name
book
never
night
another
day
tornado
Hello
everything
friend
first
Feel
left
Hurricane
hard
Church
Injili
waitin
tenderness
world
hurts
search
Remember
alone
read
better
wrote
bells
popp
seem
quite
got
spend
face
honey
thin
cut
playin
easy
popping
bad
losing
gonna
ringing
log
distance
clearly
couple
finally
Took
deal
voice
magazines
whats
hes
yes
stars
sight
aroun
Wa
Let
hear
Lord
woman
thought
goo
awake
give
really
re
Set
Pull
always
sit
st
let
anywh
fills
mind
verse
wide
Sorrow
see
intro
Intro
Call
Fall
Will
Hell
Thy
Distance
Calibrate
Lay
Saturn
Pull
Long
Short
Walking
Waking

Analysis – Feature generation

Parameters used for TFIDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
vectorizer = TfidfVectorizer(max_df=0.75, # drop words that occur in more than 3/4 of the se
                             min_df=2, # only use words that appear at least twice
                             stop_words=stopwords,
                             lowercase=True,
                             use_idf=True, # use inverse document frequencies in our weighti
                             norm=u'l2', # Apply a correction factor so that longer sentence
                             smooth_idf=True # Adds 1 to all document frequencies, as if an
                             #ngram_range=(1,3)
                            )
```

Features produced by TFIDF

	aaaaaalright	aaaaah	aaaaaha	aaaaahh	aaaaahhhhh	aaaaall	aaaah	aaaahaahahahaa	aaaahh	aaaahha	aaaahhh	aaa
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

5 rows × 169984 columns

Analyze – Word topics by components

Word topic provides more information that is helpful with displaying words that express the main topics of the songs.

Main topics contain the following top 10 words: love, like, want, don't, youre, yeah, baby, heart, never, go.

Topic 0:

	LSA	LDA	NNMF
0	love 3181.4	love 25.08	like 42.55
0	dont 2960.1	dont 20.47	dont 14.74
0	like 2437.34	like 17.01	love 14.18
0	youre 1991.37	time 14.06	youre 9.74
0	time 1905.59	youre 14.01	never 9.11
0	never 1883.79	one 13.56	want 9.07
0	one 1831.8	never 13.33	feel 8.63
0	see 1796.22	see 13.19	see 8.59
0	go 1756.34	want 12.76	cause 8.57
0	want 1722.39	go 12.7	go 8.52

Topic 1:

	LSA	LDA	NNMF
1	love 1910.69	na 76.58	love 219.89
1	baby 247.58	love 24.45	dont 43.04
1	heart 166.26	dont 21.11	baby 38.36
1	true 80.69	like 16.87	like 30.22
1	darling 52.88	youre 14.61	want 29.89
1	sweet 48.33	ja 14.55	heart 29.71
1	forever 46.49	go 13.85	never 29.65
1	loving 45.22	time 13.55	youre 29.02
1	loves 43.05	want 13.46	one 27.11
1	arms 38.48	never 13.42	ill 26.07

Topic 11:

	LSA	LDA	NNMF
11	ill 442.96	amor 176.82	ill 71.54
11	yeah 340.22	si 176.23	love 23.45
11	life 125.48	quiero 143.55	dont 18.77
11	feel 120.24	vida 103.43	never 16.36
11	youre 119.88	ms 96.23	youre 13.74
11	heart 111.13	ser 89.09	go 13.65
11	need 97.0	siempre 78.31	like 13.03
11	dont 96.37	solo 77.35	time 12.89
11	nigga 92.5	mas 74.29	one 12.61
11	give 91.54	voy 72.43	want 12.54

Topic 12:

	LSA	LDA	NNMF
12	wanna 386.47	je 145.81	wanna 44.16
12	like 345.88	et 121.81	dont 14.39
12	ill 229.05	pas 75.39	love 9.21
12	feel 223.89	qui 67.72	baby 8.49
12	girl 190.35	dans 65.41	like 8.05
12	shes 161.16	cest 56.62	go 7.09
12	tonight 150.12	pour 51.29	want 6.51
12	gonna 145.25	ne 47.29	make 5.77
12	night 107.47	mon 46.71	youre 5.7
12	make 97.33	tout 44.18	yeah 5.69

Topic 48:

	LSA	LDA	NNMF
48	hold 154.9	love 23.14	ooh 63.08
48	day 131.73	dont 21.28	love 12.47
48	nothing 129.64	like 17.57	baby 11.46
48	god 124.76	youre 14.49	dont 8.39
48	make 108.96	one 14.2	yeah 8.38
48	us 106.32	time 13.8	like 6.86
48	lord 93.65	go 13.54	want 6.13
48	right 87.13	never 13.26	youre 6.1
48	would 85.09	want 13.25	ah 5.35
48	little 82.74	see 12.94	girl 5.08

Topic 49:

	LSA	LDA	NNMF
49	little 216.92	fr 60.79	right 56.27
49	think 103.12	mehr 46.31	dont 20.85
49	life 97.97	immer 43.84	love 20.7
49	find 94.6	leben 40.52	like 15.8
49	fall 79.7	liebe 36.53	youre 14.66
49	run 75.4	schon 34.92	wrong 14.6
49	instrumental 74.01	nie 34.38	time 14.41
49	je 72.45	nacht 32.4	baby 13.86
49	night 71.61	mal 30.73	want 13.39
49	gone 70.12	zeit 28.47	one 12.41

Kmeans – Unsupervised cluster

Kmeans is used to see how effective it can be in separating the data into two different sentiments analysis.

```
km = KMeans(n_clusters=20, init='k-means++', n_jobs = -1, max_iter=100, n_init=1).fit(X_trai
y_pred = km.predict(X_train_tfidf)
y_test_pred = km.predict(X_test_tfidf)
labels = km.labels_
```

```
 Comparing training k-means clusters against sentiment label codes:
col_0                     0      1     2   3    4   5      6     7   8     9  \
sentiment_label_code
-1                      480  38749   759  20   61  13  23813   689  14   237
1                      1081  45531  3819  50  930  29  39224   638  11   827
Total                  1561  84280  4578  70  991  42  63037  1327  25  1064

col_0                     10   11   12  13  14  15     16  17  18  19   Total
sentiment_label_code
-1                     12103   81  195   0   0   8    987   1   0   0   78210
1                      14818  448  746   1   2  20  10793   1   1   1  118971
Total                  26921  529  941   1   2  28  11780   2   1   1  197181

 Comparing testing k-means clusters against sentiment label codes:
col_0                     0      1     2   3    4   5      6    7   8    9  \
sentiment_label_code
-1                      181  12660   229   5   17   9   7869  221  13   85
1                       342  15554  1254  16  265   6  13011  212   2  262
Total                   523  28214  1483  21  282  15  20880  433  15  347

col_0                     10   11   12  15    16  17  19  Total
sentiment_label_code
-1                      4018   32   73   5   340   2   0  25759
1                       4940  168  226   3  3706   0   2  39969
Total                   8958  200  299   8  4046   2   2  65728

 The adjusted rand score is: -0.005733621125138868
```

Song recommender algorithm – Positive sentiment

Based on the mood of a given song my algorithm should provide a recommended list of ten songs.

```
get_recommendations2('honesty','beyonce-knowles')
```

```
The name of the song provided: honesty
The name of the artist provided: beyonce-knowles
The genre of the artist provided: ['Pop']
The year of the song provided: [2009]
The song is considered positive with a score of: [0.9819]
```

	artist	song	genre	year	compound	sentiment_label
222476	american-idol	in-the-still-of-the-night	Pop	2012	0.9819	Positive
222665	american-idol	the-circle-of-life	Pop	2011	0.9819	Positive
113054	christopher-wilde	what-you-mean-to-me	Pop	2010	0.9819	Positive
205450	carly-rae-jepsen	turn-me-up	Pop	2012	0.9819	Positive
62373	emma-acs	very-own-human	Pop	2013	0.9819	Positive
57383	george-michael	kissing-a-fool	Pop	2011	0.9819	Positive
189138	fergie	you-already-know	Pop	2016	0.9819	Positive
82662	gloria-estefan	how-long-has-this-been-going-on	Pop	2013	0.9819	Positive
59095	alexa-borden	hope-in-the-shambles	Pop	2010	0.9819	Positive
225273	ashley-jana	spin-the-bottle	Pop	2012	0.9819	Positive

Song recommender algorithm – Negative sentiment

Based on the mood of a given song my algorithm should provide a recommended list of ten songs.

```
get_recommendations2('high-speed','2pac-outlawz')
```

```
The name of the song provided: high-speed
The name of the artist provided: 2pac-outlawz
The genre of the artist provided: ['Hip-Hop']
The year of the song provided: [2008]
The song is considered negative with a score of: [-0.9969]
```

	artist	song	genre	year	compound	sentiment_label
85021	chinx-drugz	paranoid-remix	Hip-Hop	2014	-0.9969	Negative
156392	2pac-outlawz	high-speed	Hip-Hop	2008	-0.9969	Negative
36468	ace-hood	brothers-keeper	Hip-Hop	2014	-0.9969	Negative
125662	dj-premier-bumpy-knuckles	d-lah	Hip-Hop	2012	-0.9969	Negative
90498	dmx	i-m-back	Hip-Hop	2012	-0.9969	Negative
225528	cam-meekins	cut-me-off	Hip-Hop	2012	-0.9969	Negative
227547	eminem	thats-all-she-wrote	Hip-Hop	2010	-0.9969	Negative
99317	cupcakke	image	Hip-Hop	2016	-0.9969	Negative
125661	dj-premier-bumpy-knuckles	more-levels	Hip-Hop	2012	-0.9969	Negative
6213	50-cent	get-low	Hip-Hop	2015	-0.9969	Negative

Generate Model and evaluate performance – Decision Tree

```
param_grid = {'max_depth': [10, 20, 30],
              'max_features': ['auto', 'sqrt']}

scores = ['percision','recall','f1']

for score in scores:
    print("# Tuning hyper-parameters for %s" % score)
    print()

dtc = tree.DecisionTreeClassifier()
grid_clf_dtc = GridSearchCV(dtc, param_grid, cv=3, scoring='%s_macro' % score)
```

```
Best parameter choice for logistic model: {'max_depth': 30, 'max_features': 'sqrt'}
Training score for the best parameter: 0.544550255918021

 Decison Tree confusion matrix
 [[ 5396 20363]
 [ 2612 37357]]

 Decison Tree classification report
               precision    recall  f1-score   support

           -1       0.67      0.21      0.32     25759
            1       0.65      0.93      0.76     39969

   micro avg       0.65      0.65      0.65     65728
   macro avg       0.66      0.57      0.54     65728
weighted avg       0.66      0.65      0.59     65728

Decison Tree accuracy score: 0.6504533836416748
```

Generate Model and evaluate performance – Naïve Bayes

```
param_grid = {'alpha': [0.001,0.01,0.1,1,3]}

scores = ['precision','recall','f1']
for score in scores:
    print("# Tuning hyper-parameters for %s" %  score)
    print()


nb = MultinomialNB()
grid_clf_nb = GridSearchCV(nb, param_grid, cv=3, scoring='%s_macro' % score)
```

```
Best parameter choice for logistic model: {'alpha': 0.01}
Training score for the best parameter: 0.6938168131333016

 Naive Bayes cufusion matrix
 [[12345 13414]
 [ 4097 35872]]

 Naive Bayes classification report
                precision    recall  f1-score   support

           -1       0.75      0.48      0.59     25759
            1       0.73      0.90      0.80     39969

    micro avg       0.73      0.73      0.73     65728
    macro avg       0.74      0.69      0.69     65728
 weighted avg       0.74      0.73      0.72     65728

Naive Bayes accuracy score: 0.7335838607594937
```

Generate Model and evaluate performance – Support vector

```
param_grid = {'C': [6,7,8,9,10,11,12]}

scores = ['precision','recall','f1']
for score in scores:
    print("# Tuning hyper-parameters for %s" %  score)
    print()


svc = LinearSVC()
grid_clf_svm = GridSearchCV(svc, param_grid, cv=3, scoring='%s_macro' % score)
```

```
Best parameter choice for logistic model: {'C': 6}
Training score for the best parameter: 0.806716518713244 1

 Support vector cufusion matrix
 [[19445  6314]
 [ 5143 34826]]

 Support vector classification report
               precision    recall  f1-score   support

           -1       0.79      0.75      0.77     25759
            1       0.85      0.87      0.86     39969

    micro avg       0.83      0.83      0.83     65728
    macro avg       0.82      0.81      0.82     65728
 weighted avg       0.82      0.83      0.82     65728

 Support vector accuracy score: 0.825690725413826 7
```

Generate Model and evaluate performance – Logistic Regression

```
param_grid = {'C': [6,7,8,9,10,11,12],
              'penalty': ['l1','l2']
              }

scores = ['precision','recall','f1']
for score in scores:
    print("# Tuning hyper-parameters for %s" %  score)
    print()

lr = LogisticRegression(n_jobs=1)
grid_clf_lr = GridSearchCV(lr, param_grid, cv=3, scoring='%s_macro' % score)
```

```
Best parameter choice for logistic model: {'C': 6, 'penalty': 'l2'}
Training score for the best parameter: 0.8352472971422913

 Logistic cufusion matrix
 [[20064  5695]
 [ 4049 35920]]

 Logistic classification report
               precision    recall  f1-score   support

           -1       0.83      0.78      0.80     25759
            1       0.86      0.90      0.88     39969

    micro avg       0.85      0.85      0.85     65728
    macro avg       0.85      0.84      0.84     65728
 weighted avg       0.85      0.85      0.85     65728

Logistic accuracy score: 0.8517526777020448
```

Generate Model and evaluate performance – Stochastic Gradient Decent

```
param_grid = {'loss': ["hinge", "log", "perceptron"],
              'alpha': [0.001,0.0001,0.00001],
               'max_iter': [5,7,9],
               'tol': [None]}

scores = ['precision','recall','f1']
for score in scores:
    print("# Tuning hyper-parameters for %s" %  score)
    print()

sgdc = SGDClassifier()
grid_clf_sgdc = GridSearchCV(sgdc, param_grid, cv=3, scoring='%s_macro' % score)
```

```
Best parameter choice for SGDC model: {'alpha': 1e-05, 'loss': 'hinge', 'max_iter': 9, 'tol':
None}
Training score for the best parameter: 0.8356525824757436

 SGDC cufusion matrix
 [[19876  5883]
 [ 3847 36122]]

 SGDC classification report
               precision    recall  f1-score   support

           -1       0.84      0.77      0.80     25759
            1       0.86      0.90      0.88     39969

    micro avg       0.85      0.85      0.85     65728
    macro avg       0.85      0.84      0.84     65728
 weighted avg       0.85      0.85      0.85     65728

SGDC accuracy score: 0.851965676728335
```

Sample song testing

To validate my model would continue to predict the sentiment analysis of any song, I randomly download two songs from the internet that was not part of my training or testing source data.

	index	song	year	artist	genre	lyrics	clean_lyrics	lemmatized_sample	sample_sentiment_label
0	0	It Never Rains in Southern California	1990	tony-toni-tone	R&B	I haven't seen your face in a year \rI can't w...	I havent seen your face in a year I cant wait ...	I havent see your face in a year I cant wait t...	1
1	1	Gimme the Loot	1994	The-Notorious-BIG	Hip-Hop	Yeah, mothafuckas better know\rI'm a bad bad b...	Yeah mothafuckas better know Im a bad bad bad ...	Yeah mothafuckas better know Im a bad bad bad ...	-1

Sample execution results

Executed the sample set in my highest performing model SGDC and the model was able to successfully predict the sentiment analysis.

```
SGDC sample cufusion matrix
[[1 0]
[0 1]]

SGDC sample classification report
              precision    recall  f1-score   support

          -1       1.00      1.00      1.00         1
           1       1.00      1.00      1.00         1

   micro avg       1.00      1.00      1.00         2
   macro avg       1.00      1.00      1.00         2
weighted avg       1.00      1.00      1.00         2

SGDC sample accuracy score: 1.0
```

Summary of results:

- Overall the model was able to successfully analyze song lyrics and predict the sentiment analysis of those lyrics as either a positive or negative song. Utilizing Vader sentiment analysis the model was able to utilize the numerical score of a song based on the lyrics and offer a list of recommended songs based on the sentiment analysis of a given song.

- The model also performed two types of unsupervised techniques (Topic modeling using LSA, LDA and NNMF and Kmeans). Topic modeling cluster the data by terms to reflect relevant topics. Kmeans cluster the data by sentiment analysis but adjusted random index score of -0.005 shows it's not performing well.

- Five models were used to predict the sentiment analysis of a song. Decision Tree: 65% accuracy , Naïve Bayes 73% accuracy, Support Vector accuracy 82%, Logistic regression accuracy 85.17% and SDGC performed just slightly better with 85.19% accuracy.

	Model	Training score	Accuracy
0	Decision Tree	0.544550	0.650453
1	Navie Bayes	0.693817	0.733584
2	Support Vector	0.806717	0.825691
3	Logistic Regression	0.835247	0.851753
4	Stochastic Gradient Decent	0.835653	0.851966