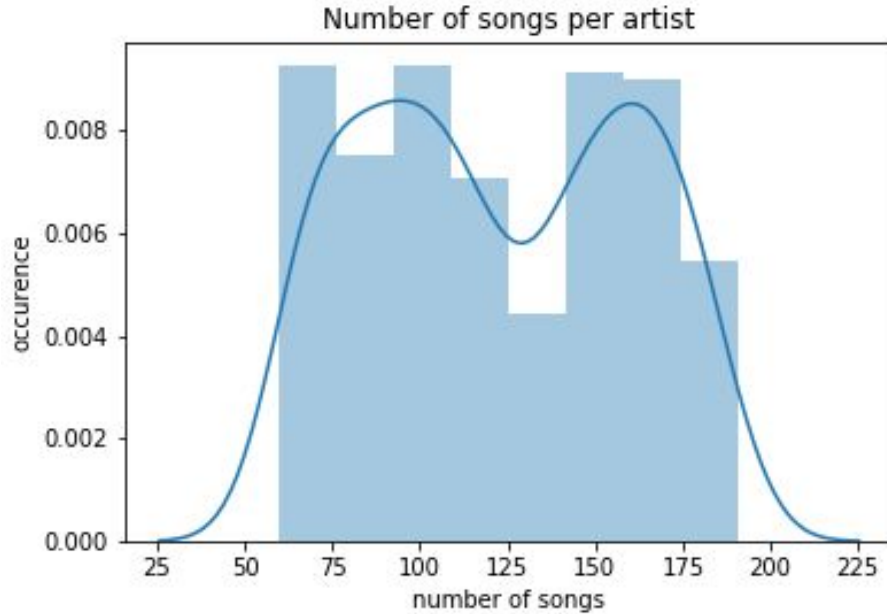


# Clustering

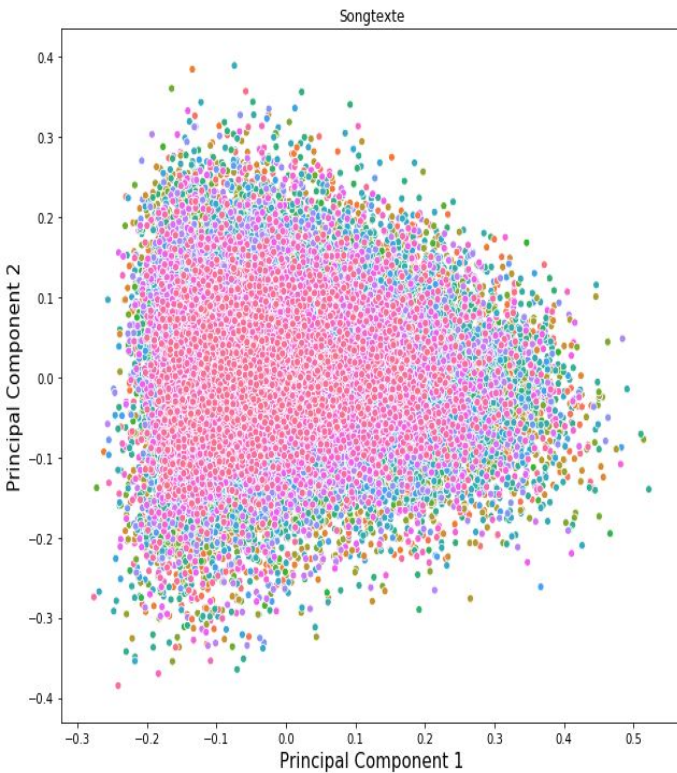
Ist das Genre abhängig von Lyrics?

# Erster Datensatz

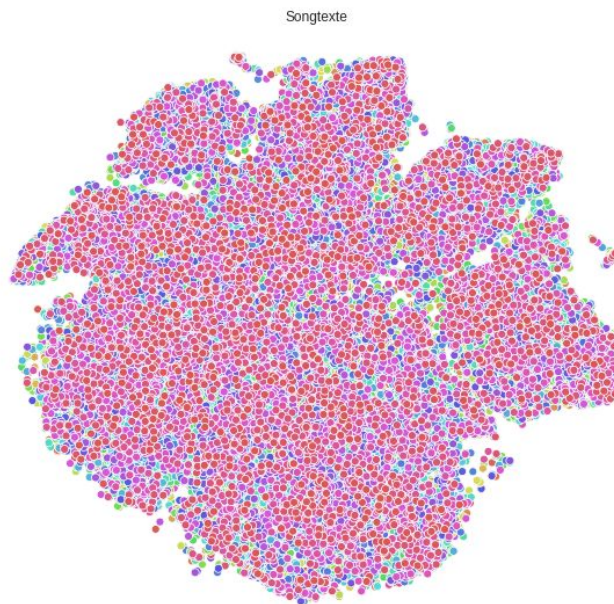


- 643 Bands
- 57 650 Songs

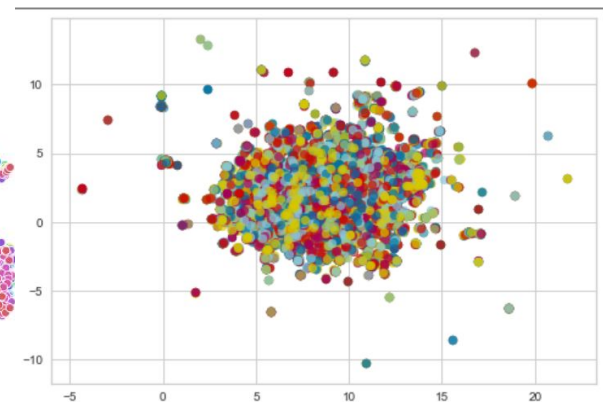
# PCA



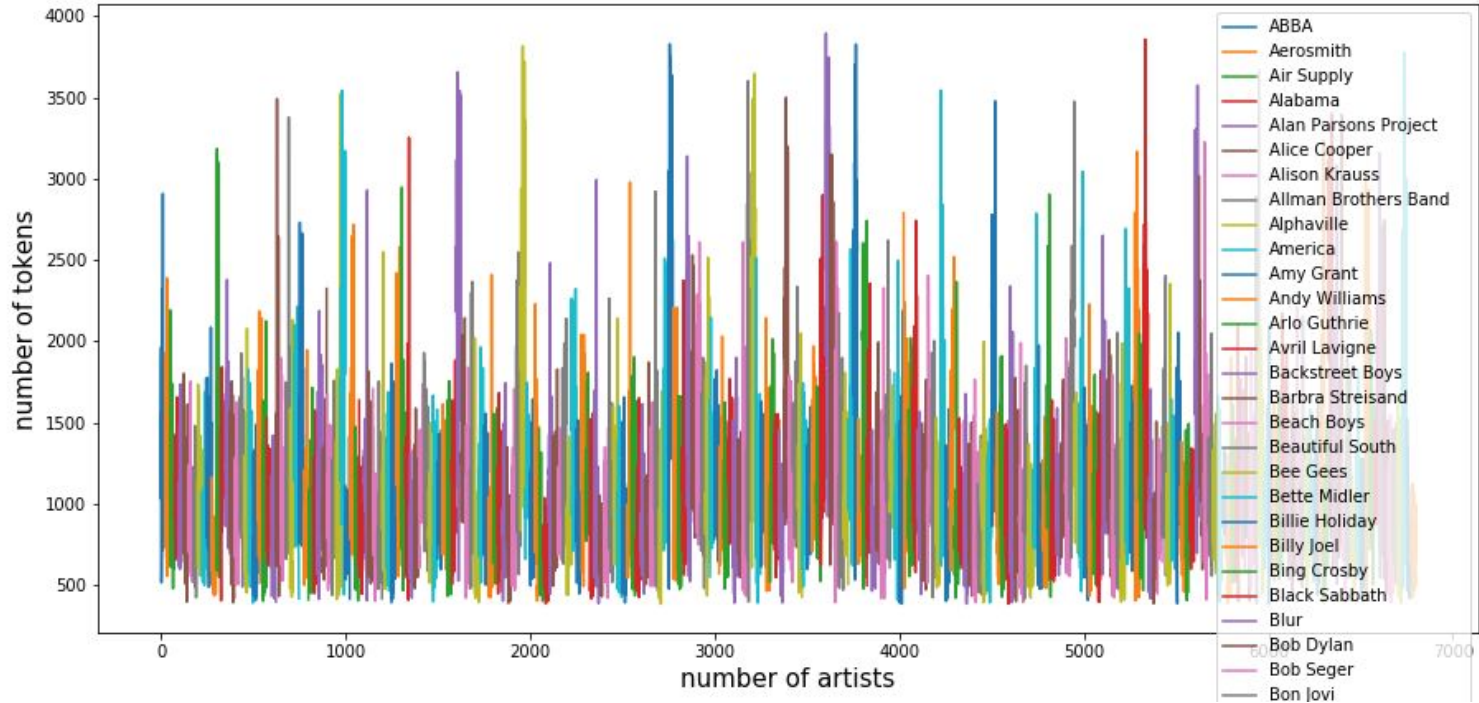
# TSNE



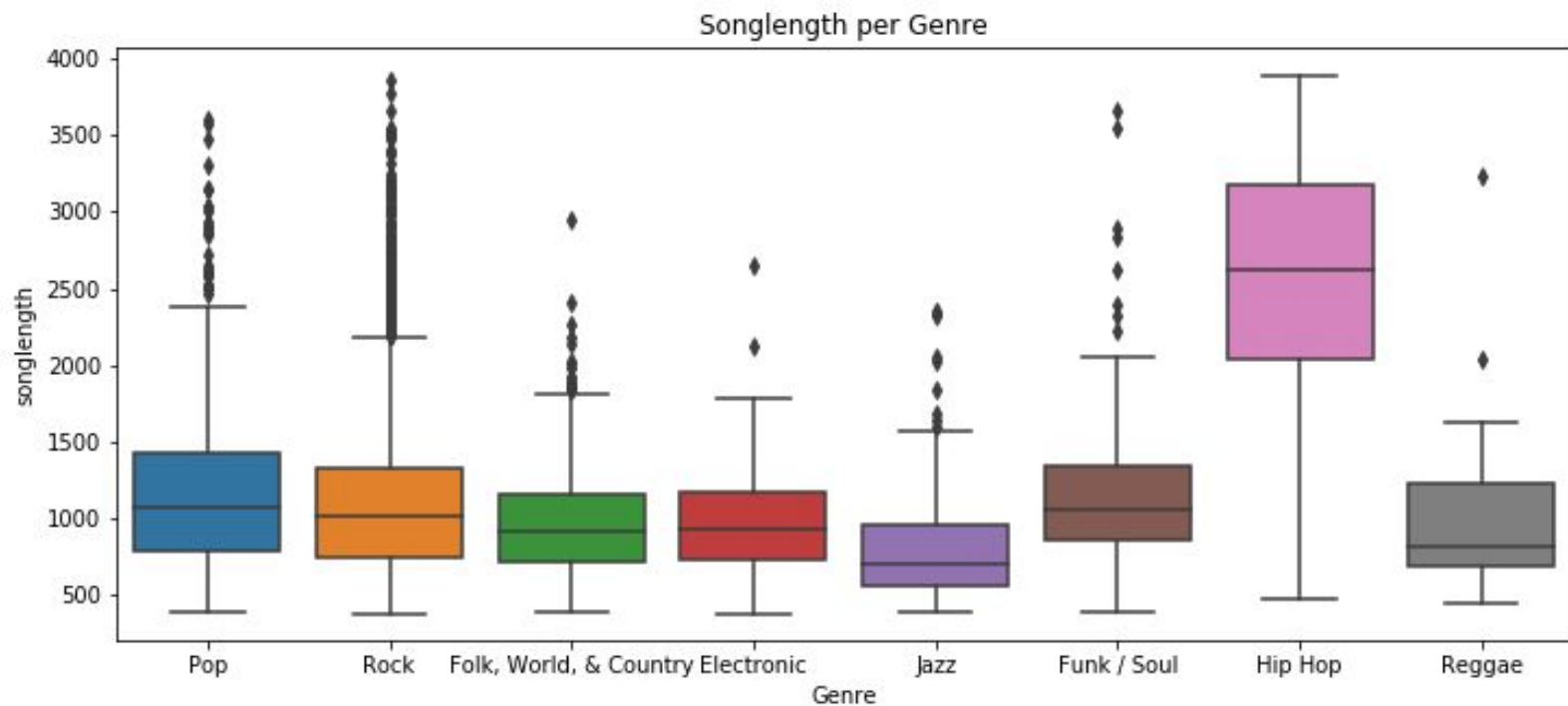
# UMAP



# Datensatz anpassen

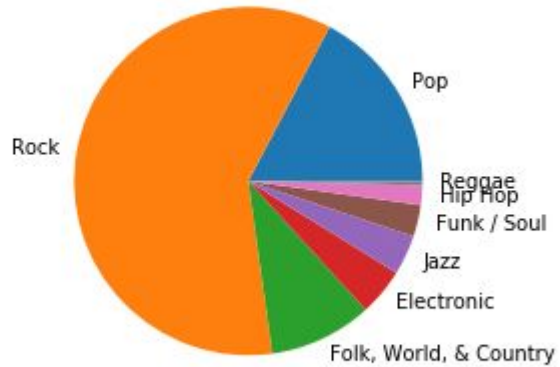


# Genres

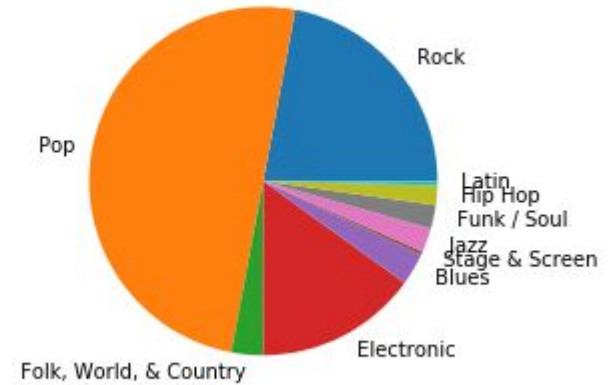


# Genres Verteilung

Genre1

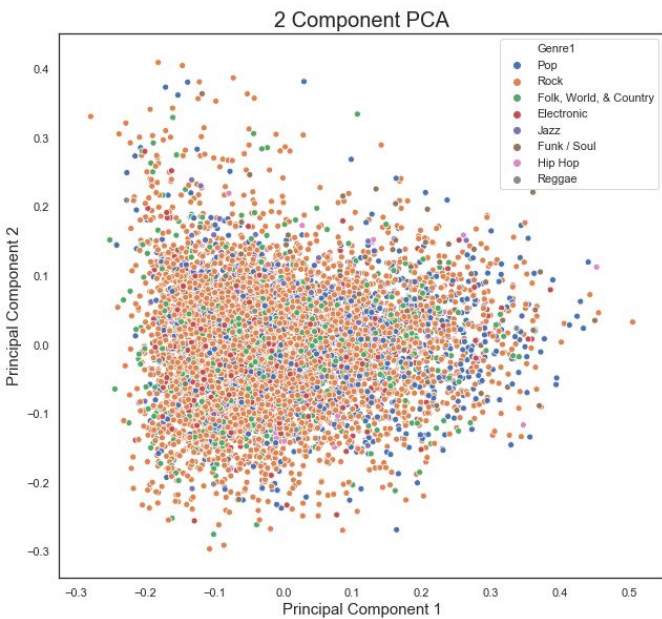


Genre2

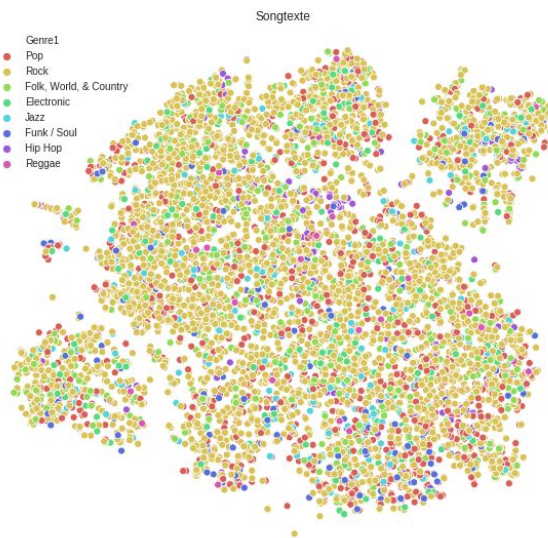


# Lyrics

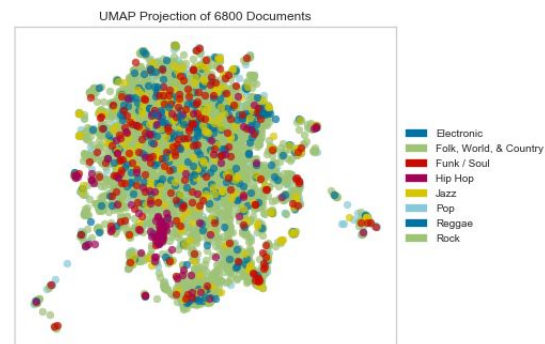
PCA



t-SNE



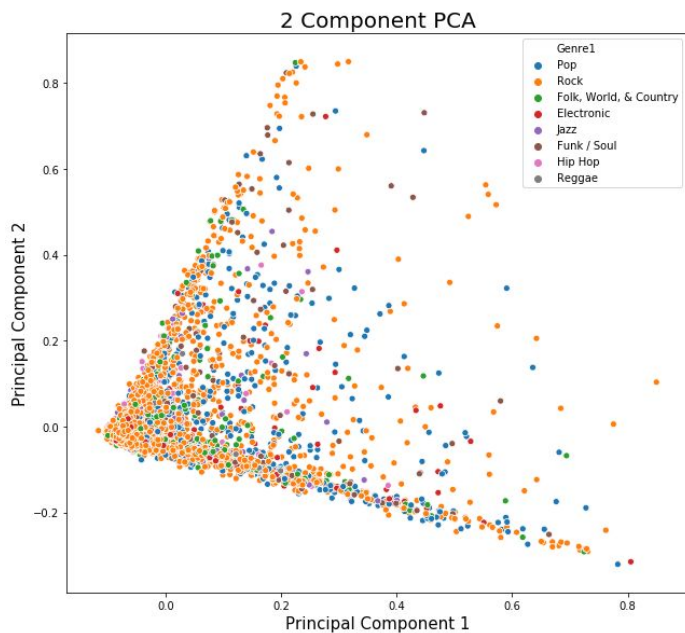
UMAP



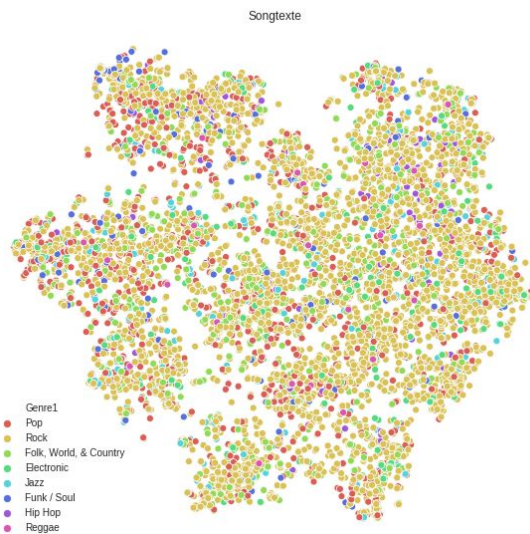


# POS

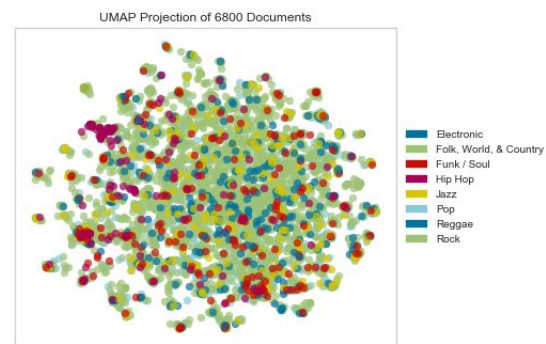
PCA



t-SNE

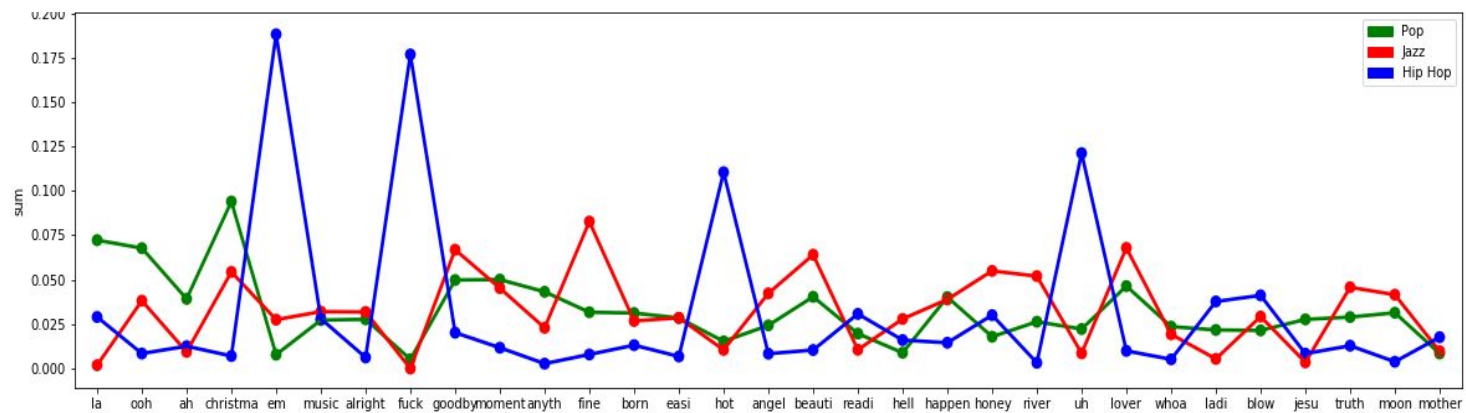


UMAP

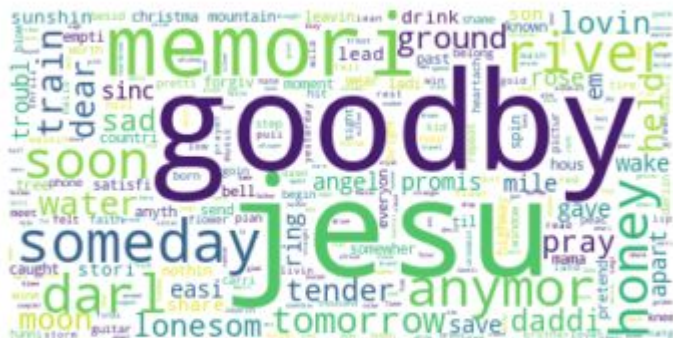




# 30 häufigste Wörter nach Genre

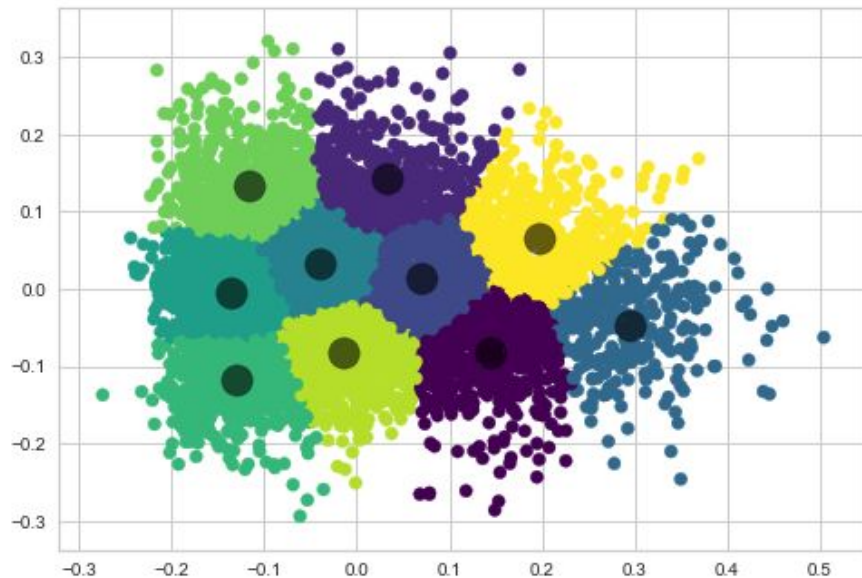


# Wordclouds Country - Hip Hop

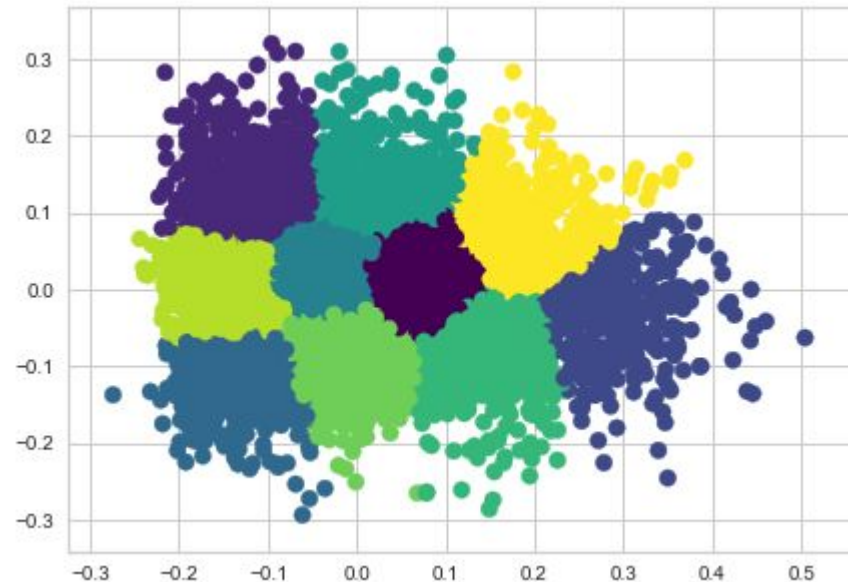


# K-Means

PCA -- eigene Implementierung



PCA -- SCIKIT-LEARN

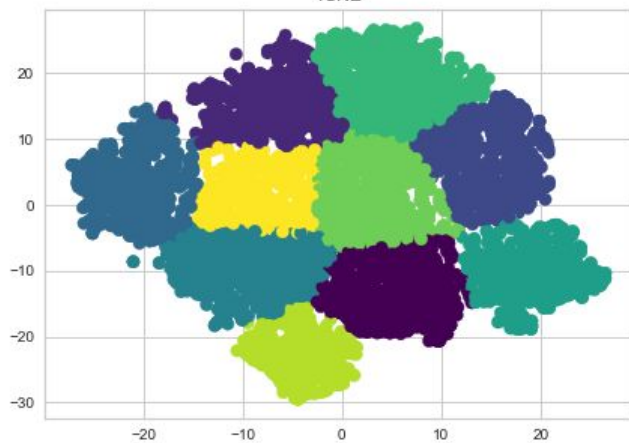


# K-Means Lyrics

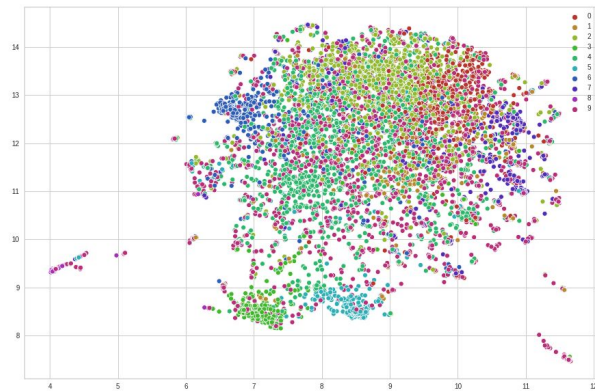
PCA



t-SNE

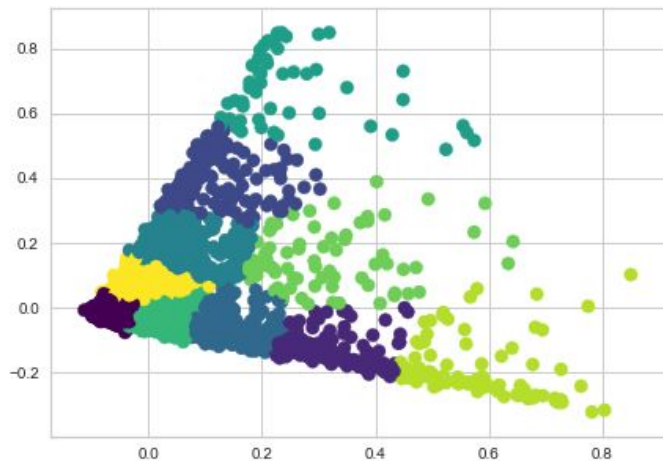


UMAP

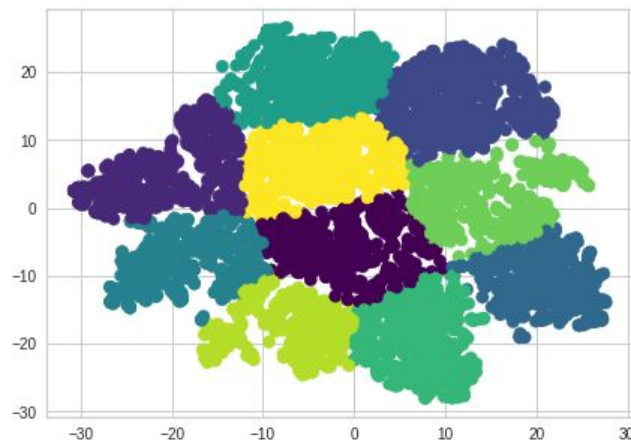


# K-Means POS

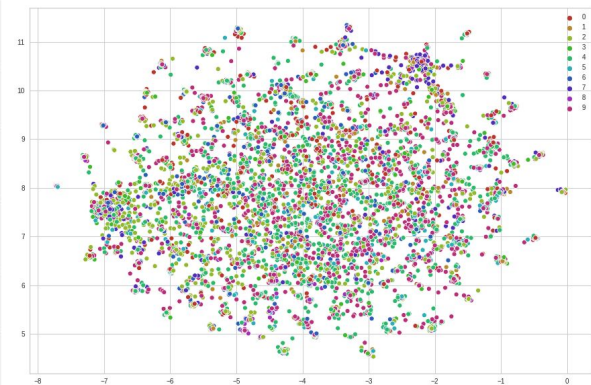
PCA



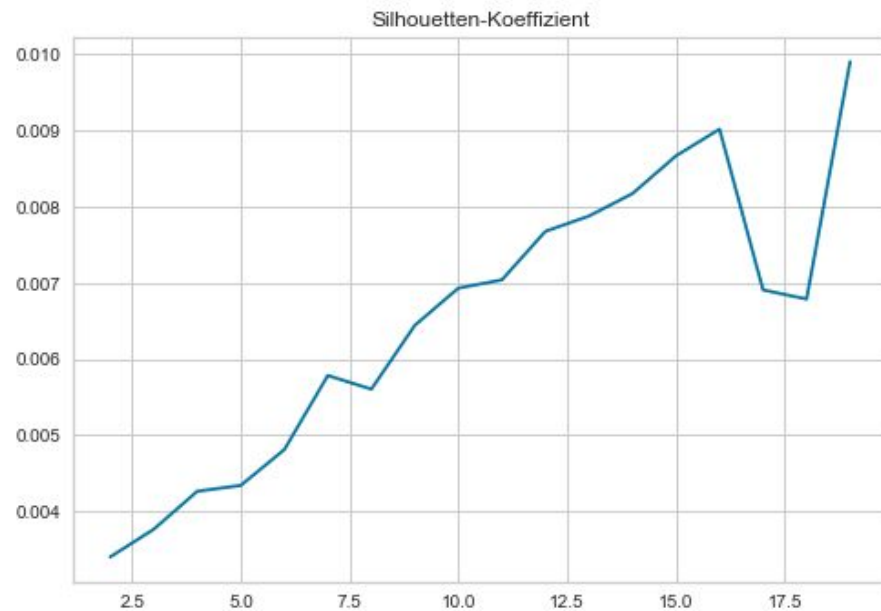
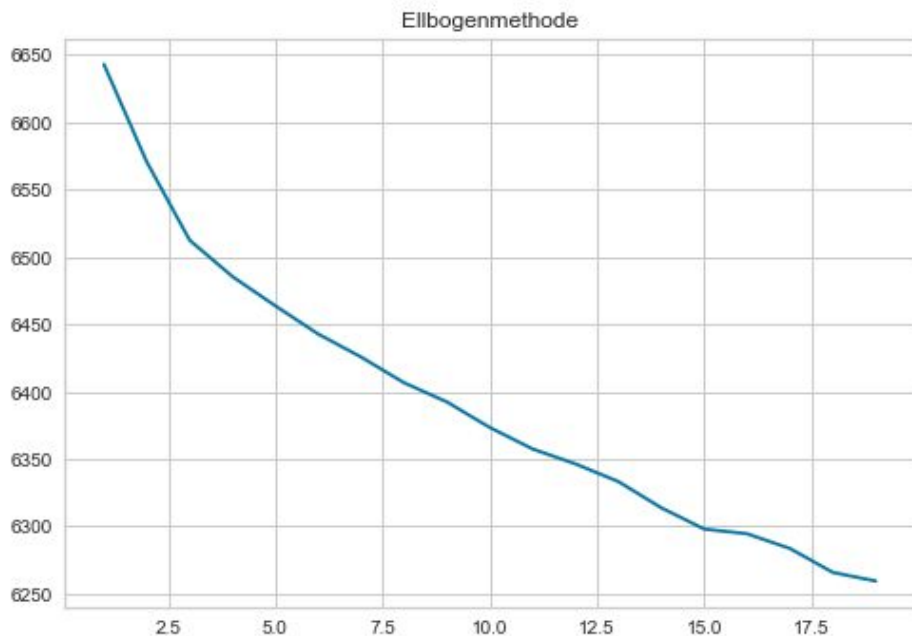
t-SNE



UMAP



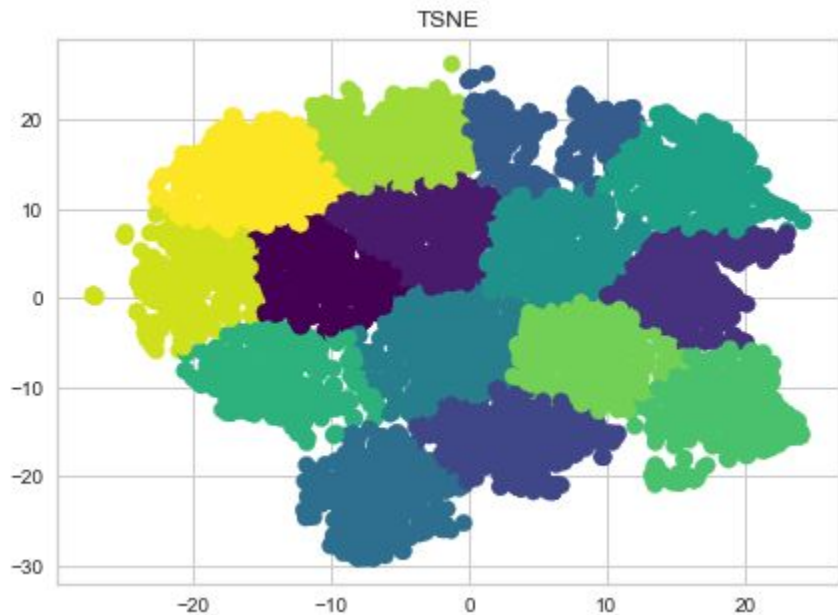
# Anzahl der Cluster



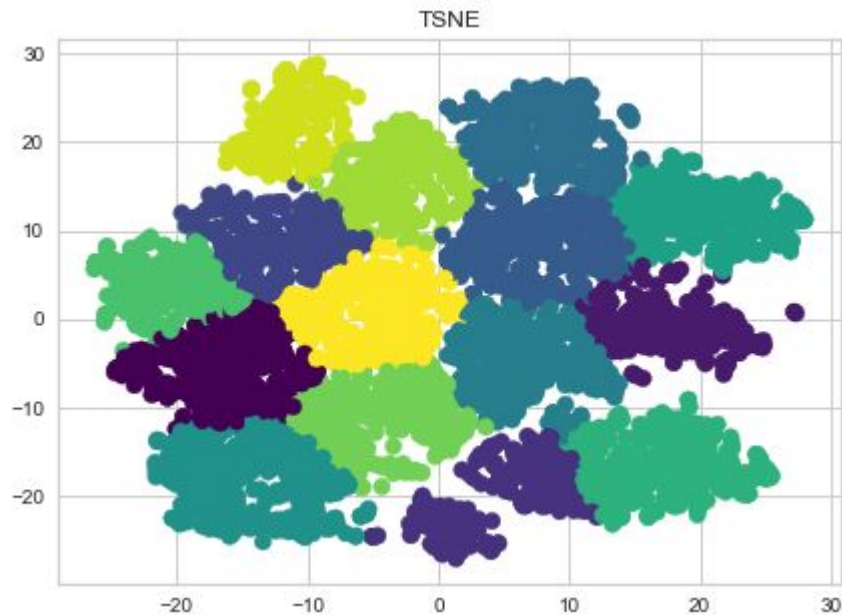


# t-SNE mit $k = 15$

Lyrics



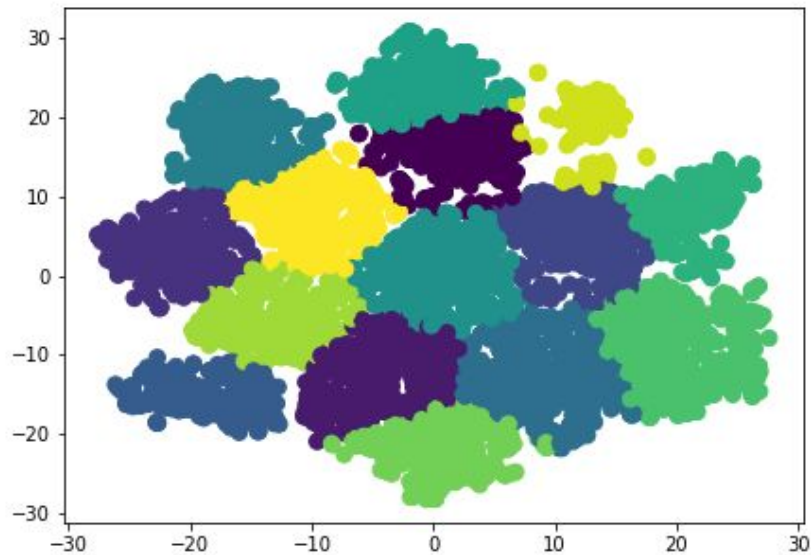
POS



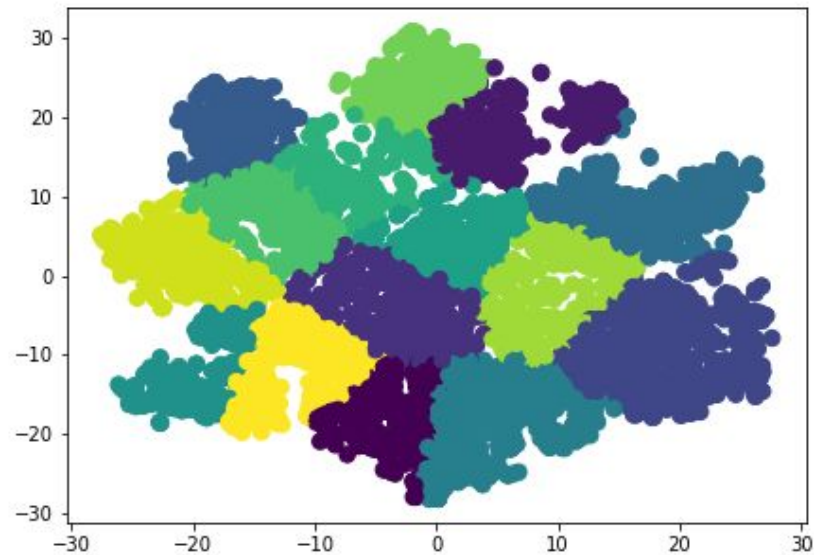


# Mini-Batch mit t-SNE und POS

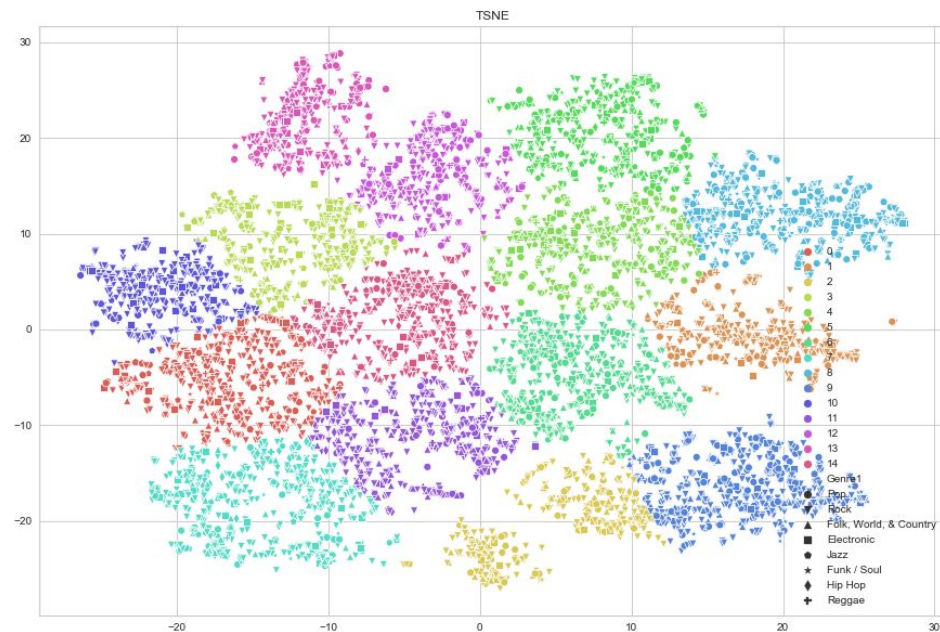
K-Means



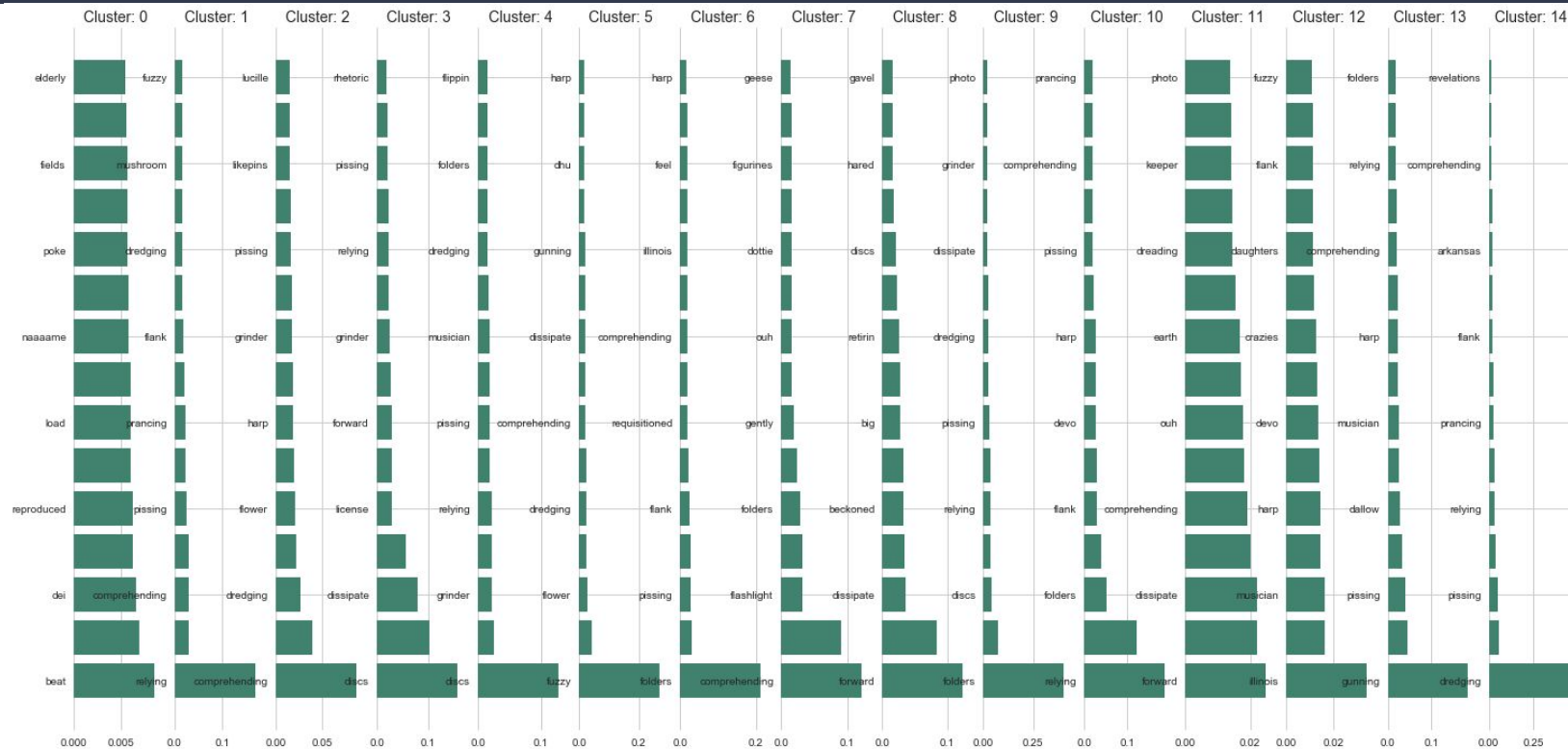
Mini-Batch



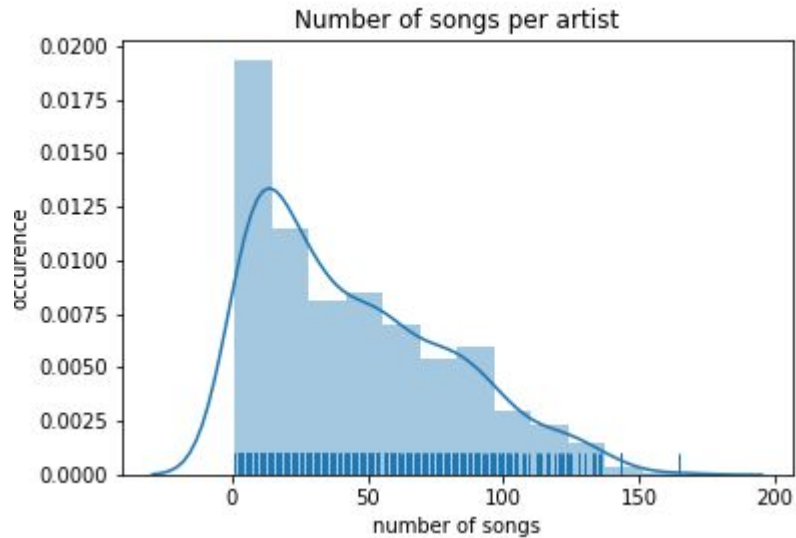
# t-SNE mit Scikit-Learn und Genres



# Features in Cluster

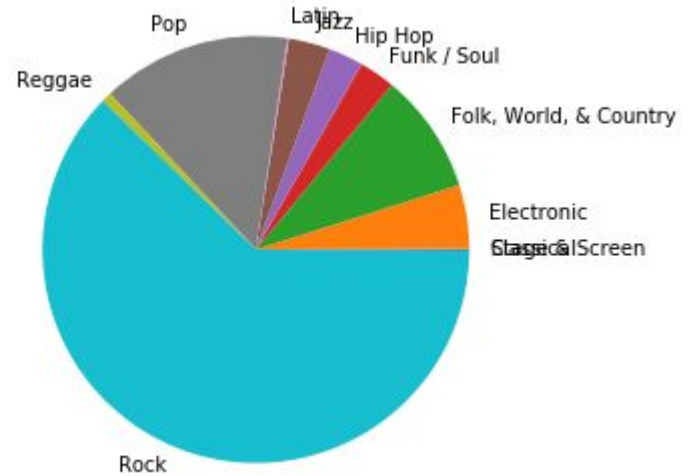
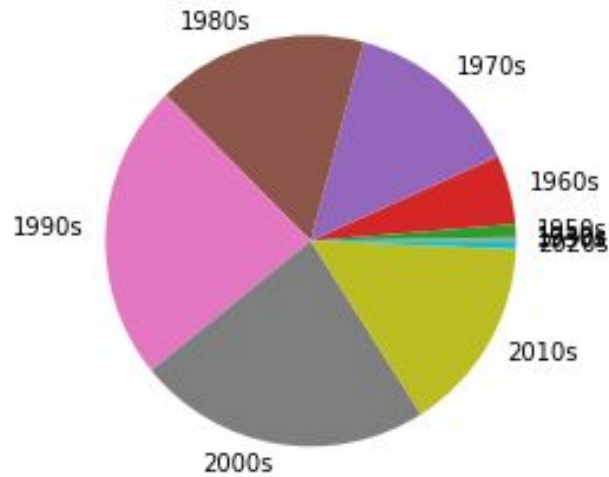


# neuer Datensatz



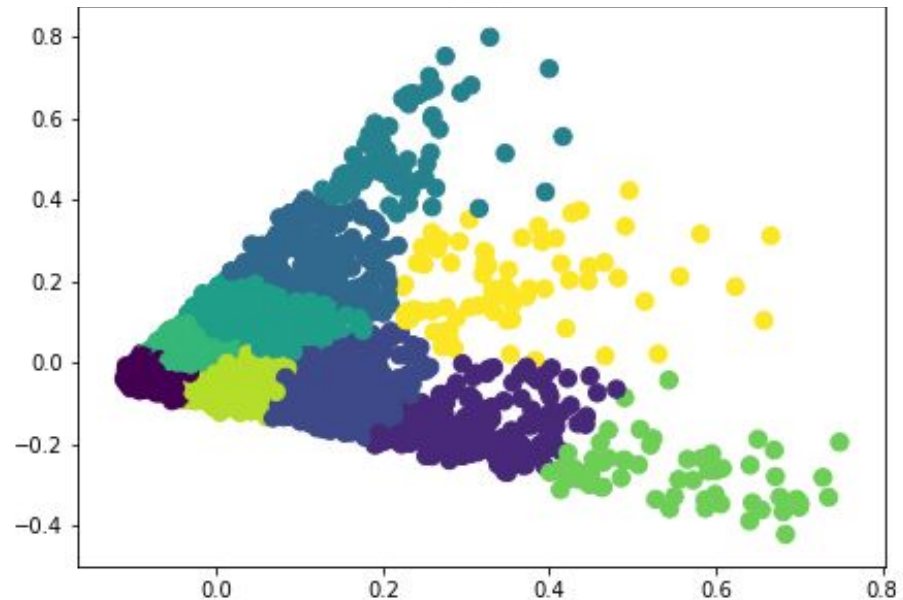
- 541 Bands
- 24.443 Songs

# neuer Datensatz

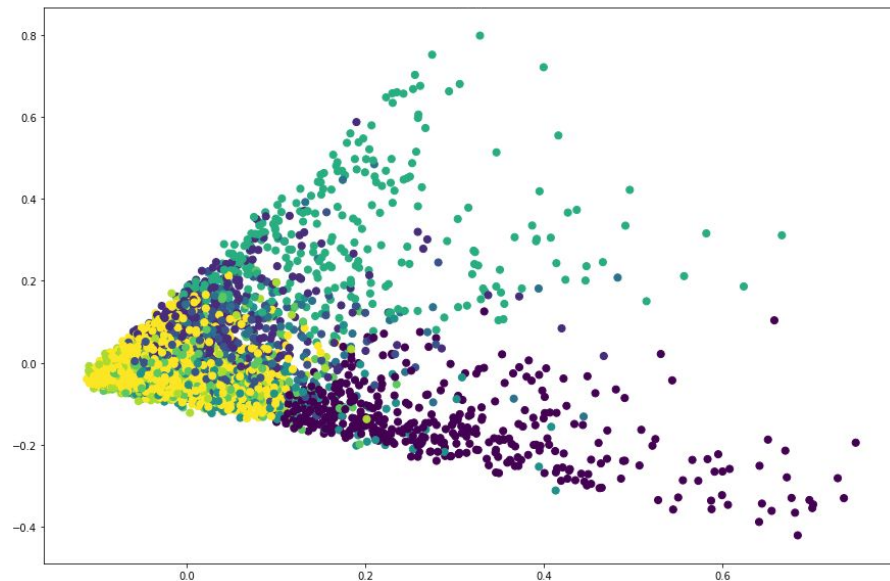


# K-Means mit PCA

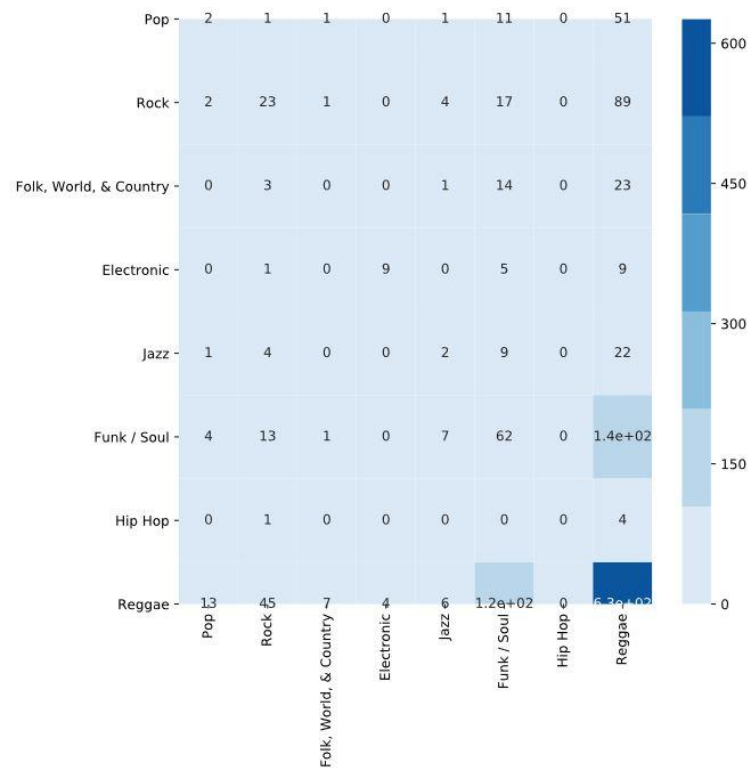
PCA vor KMeans



PCA nach KMeans

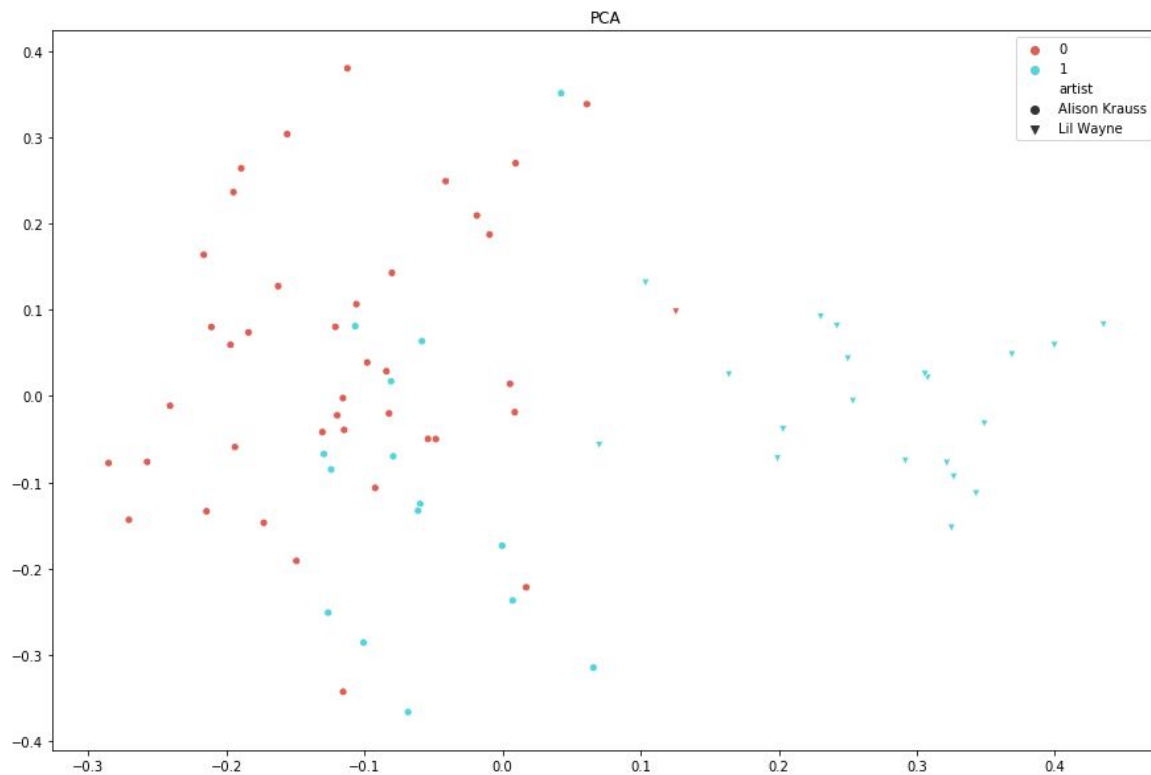


# SVM



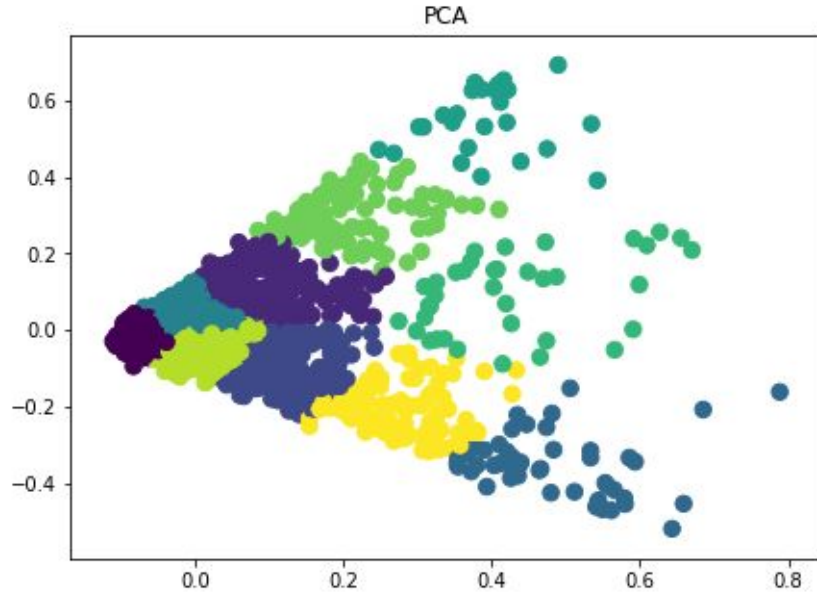


# Künstler

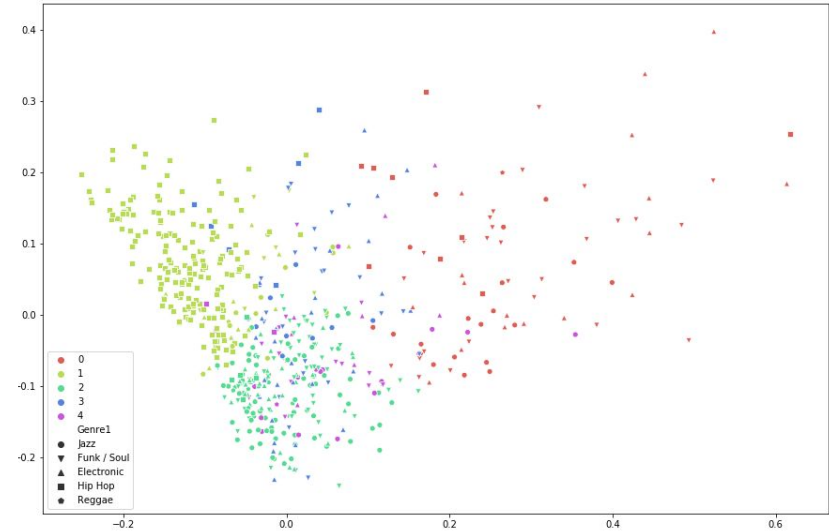


# Genres

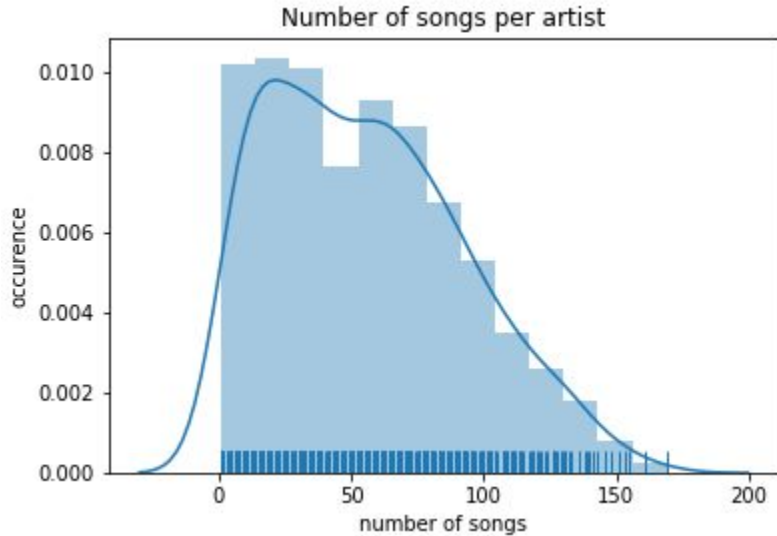
Rock



alle Genres außer Pop und Rock



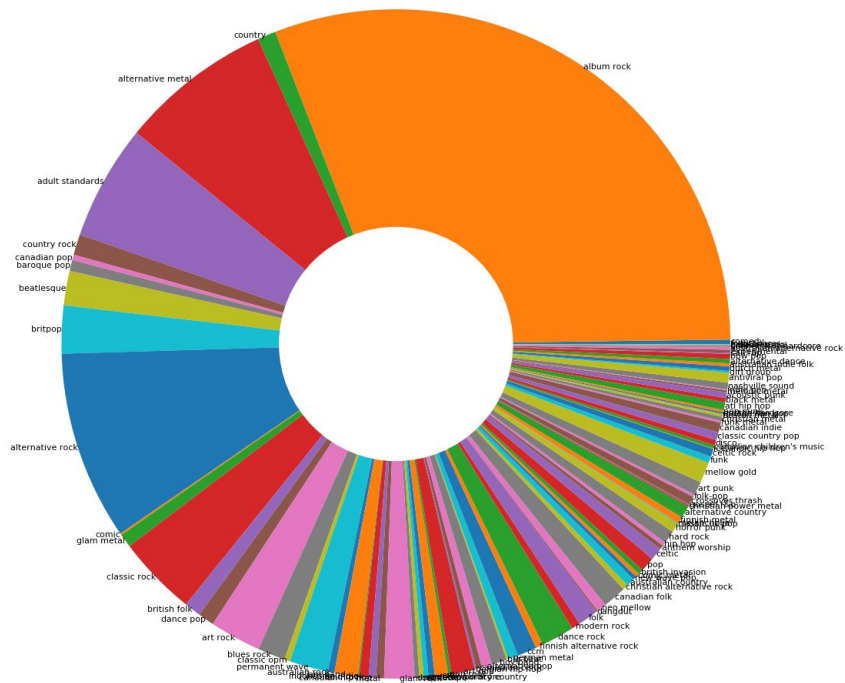
# neuer Datensatz



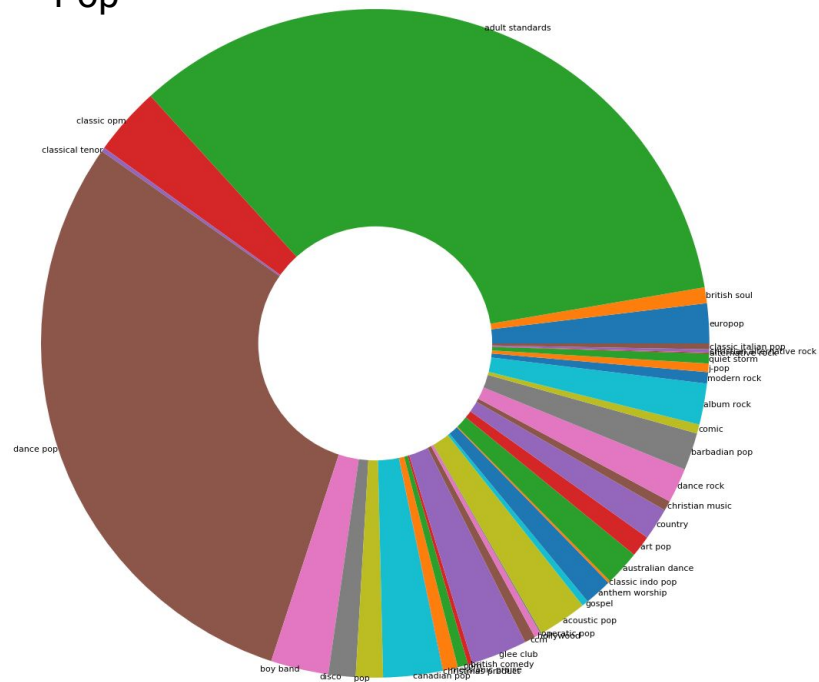
- 597 Bands
- 33.440 Songs

# Subgenre

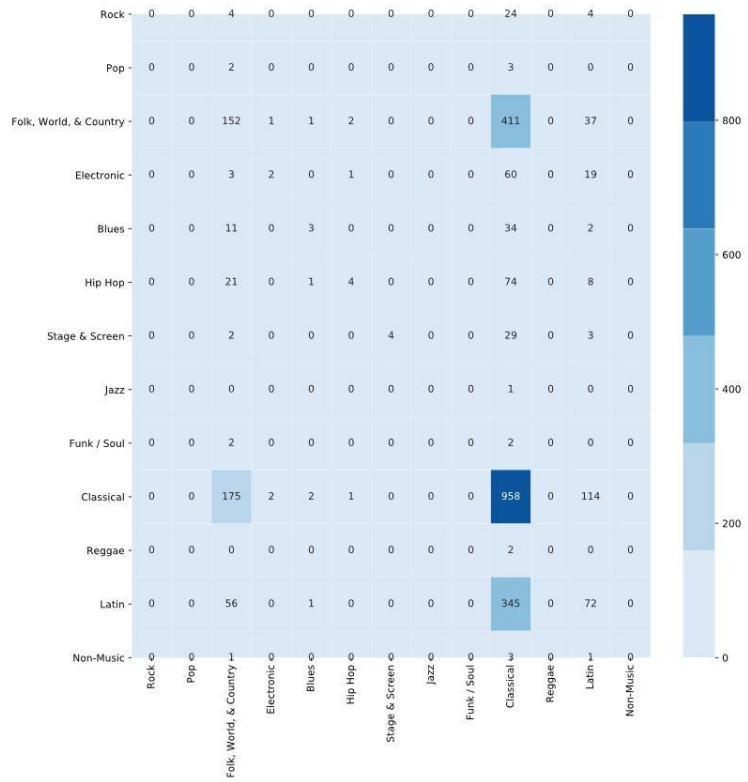
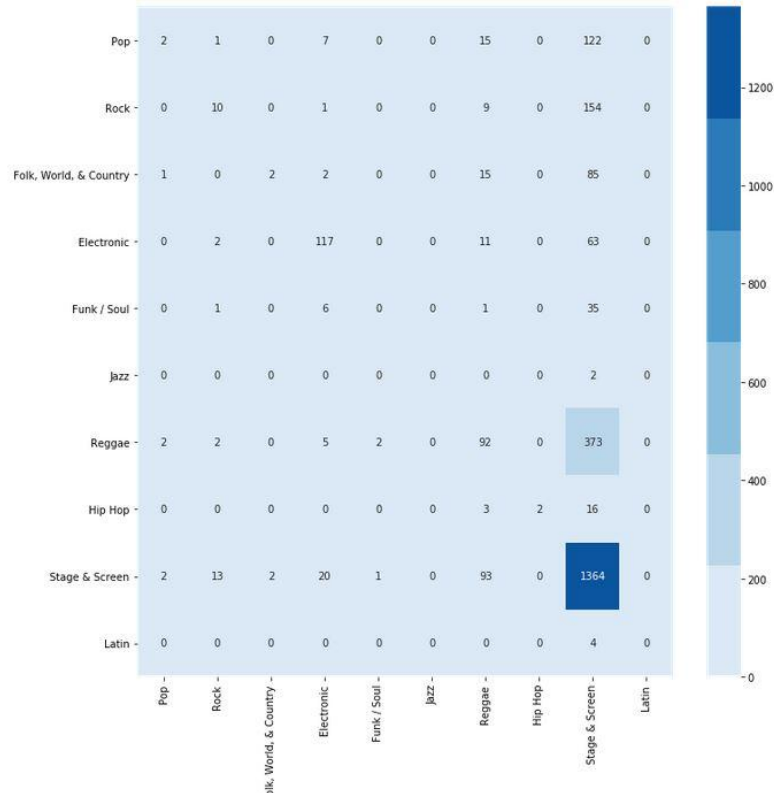
# Rock



Pop

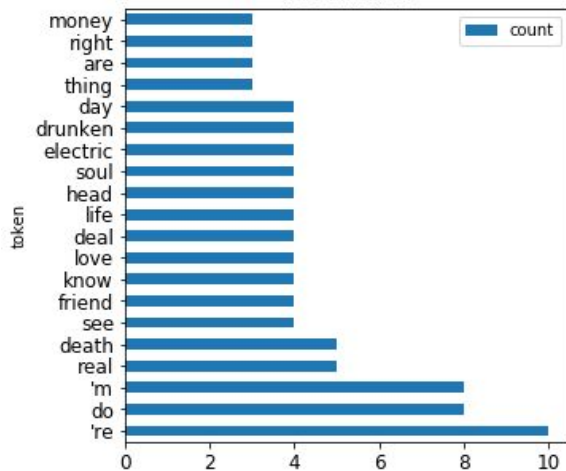


# SVM

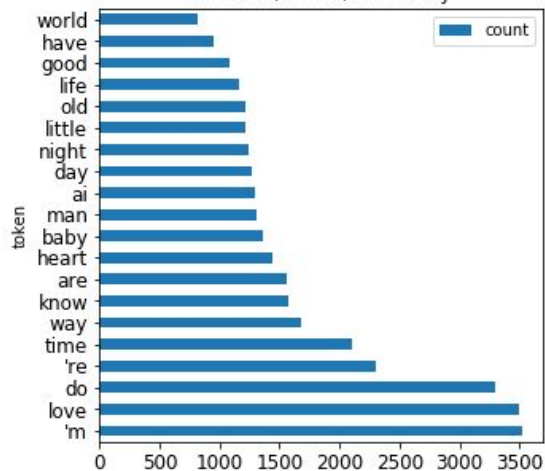


# Nach SVM -- most common words

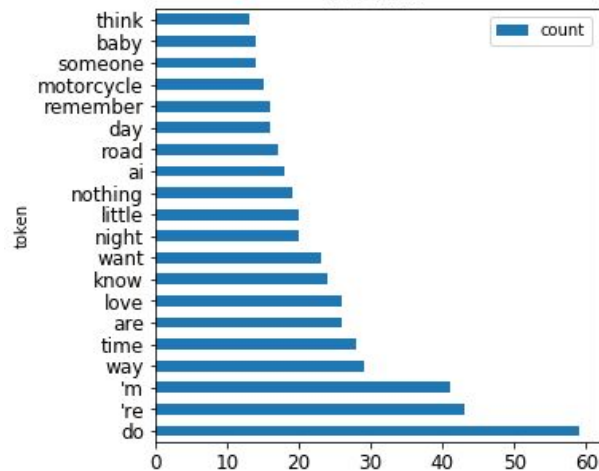
POS Classical



POS Folk, World, & Country

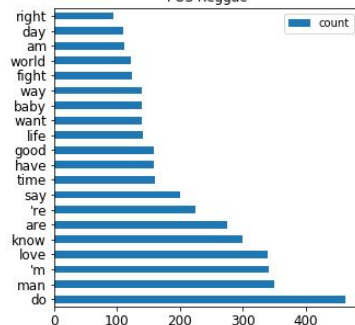


POS Latin

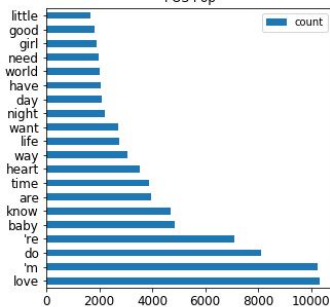


# Genres – most common words

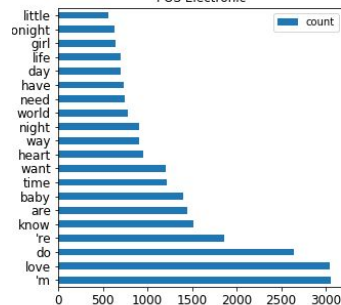
POS Reggae



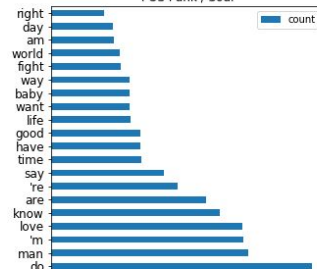
POS Pop



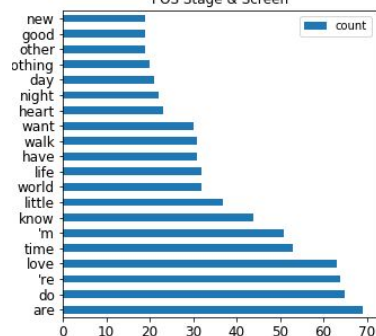
POS Electronic



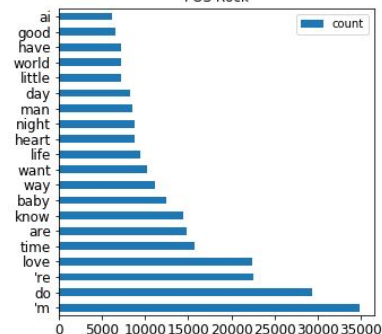
POS Funk / Soul



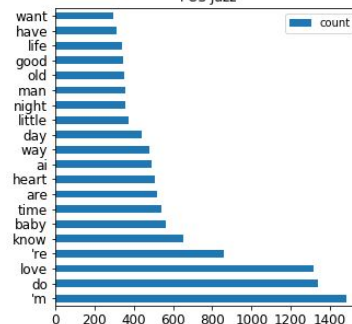
POS Stage & Screen



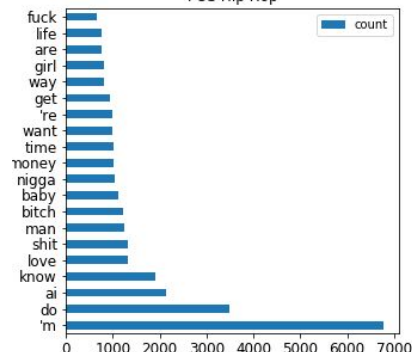
POS Rock



POS Jazz

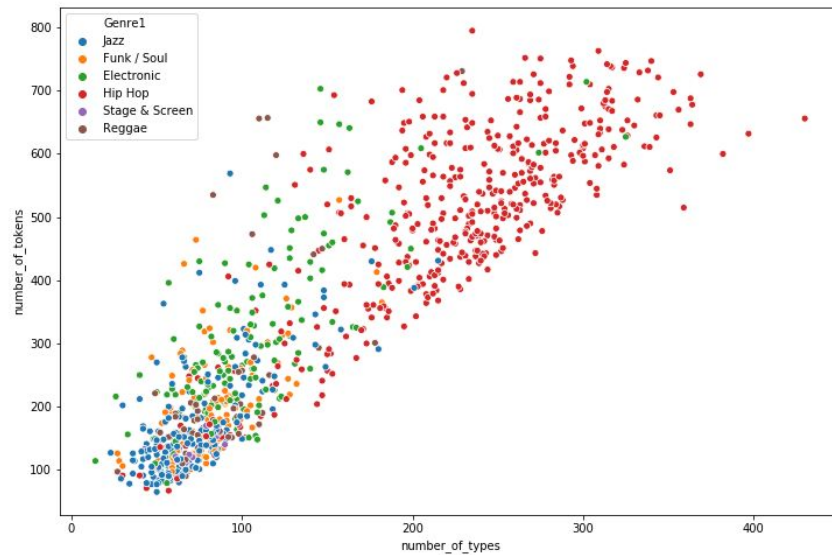
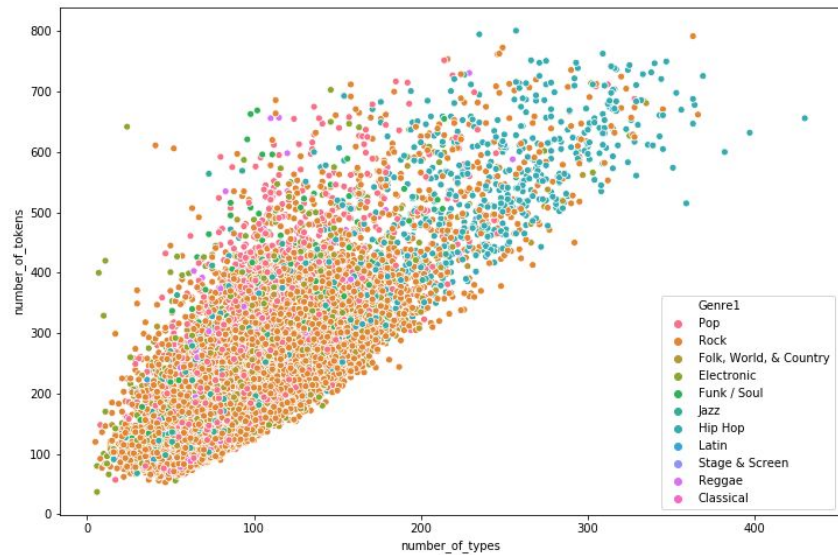


POS Hip Hop



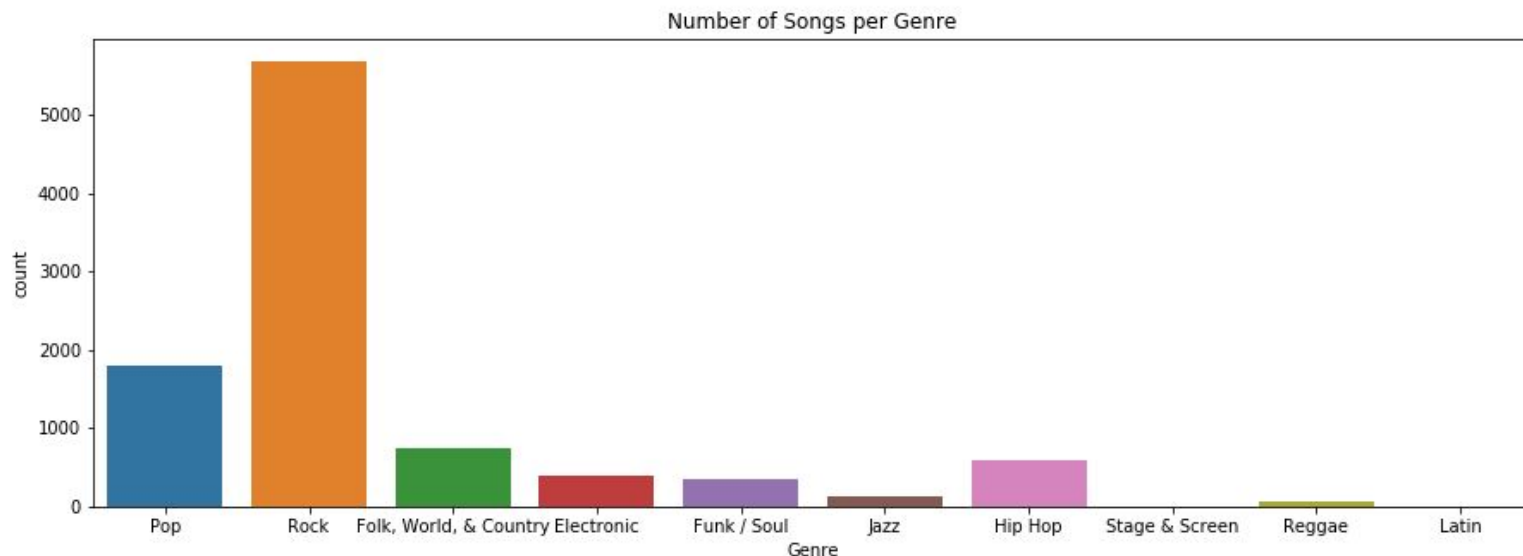


# type-token ratio



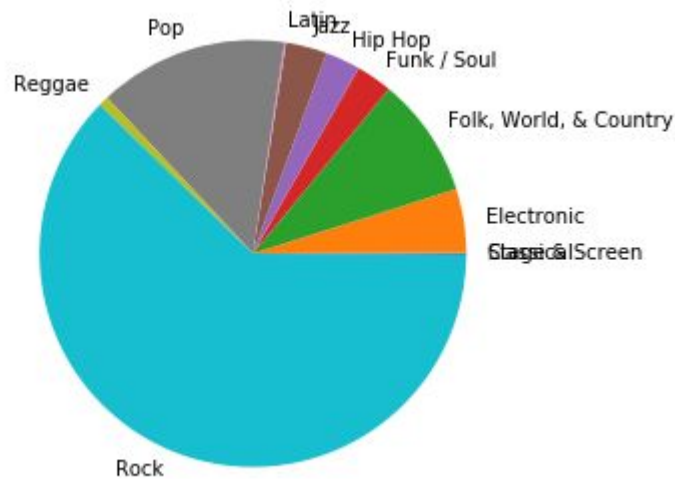
# verkleinerter Datensatz

nur Texte über der Durchschnittslänge: hier ca. 222 Tokens

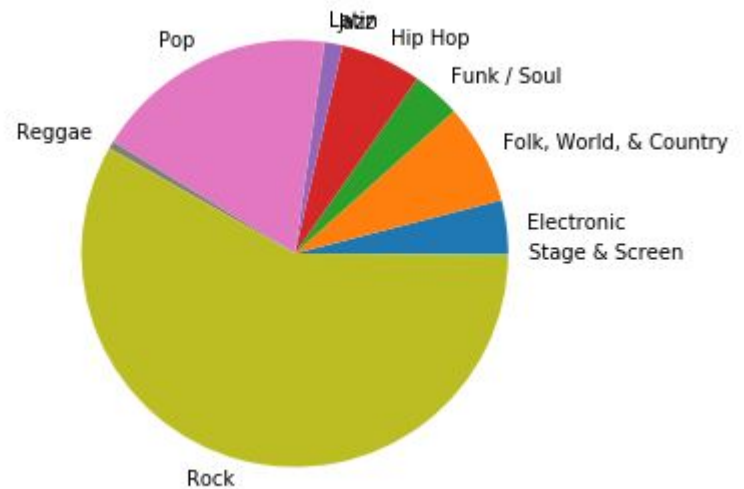


# Verteilung der Genres

vor Verkleinerung

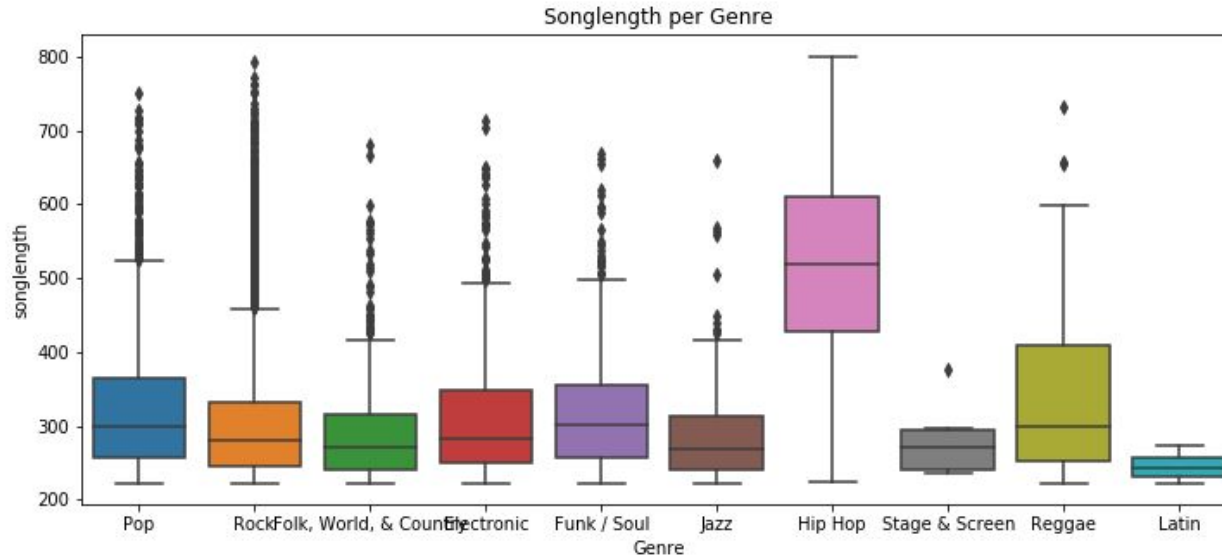


nach Verkleinerung



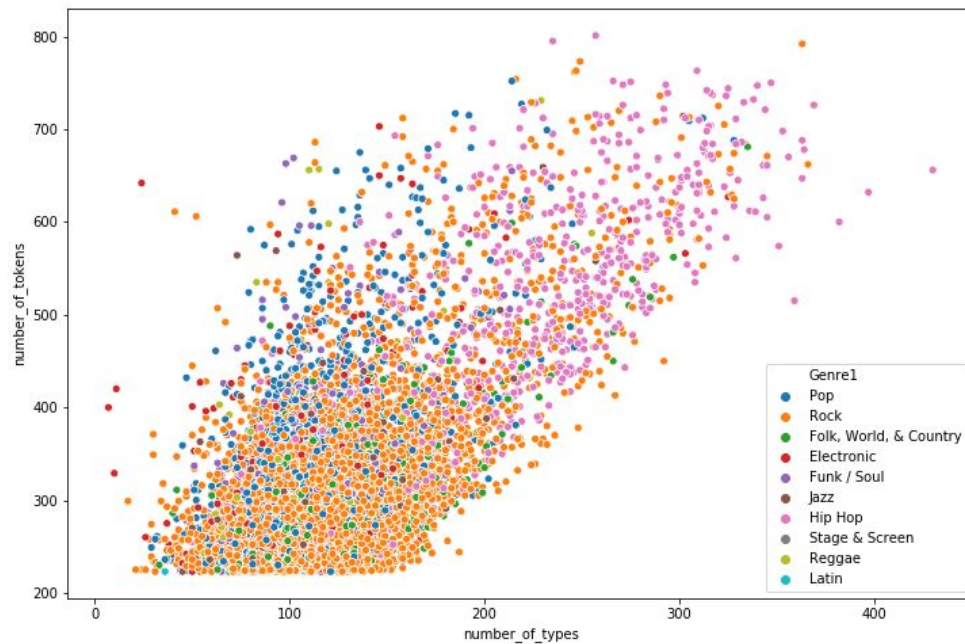
# Textlängen pro Genre

Textlänge von HipHop fällt noch stärker auf als vor Verkleinerung des Datensatzes

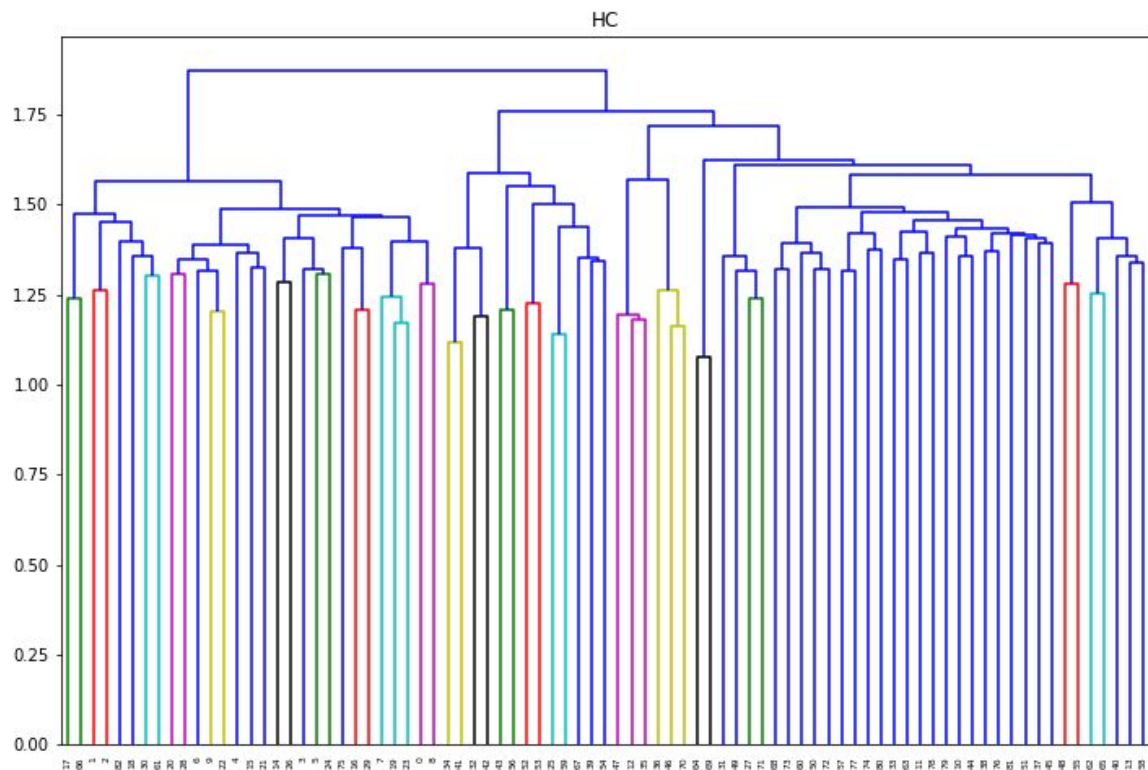


# type-token-ratio

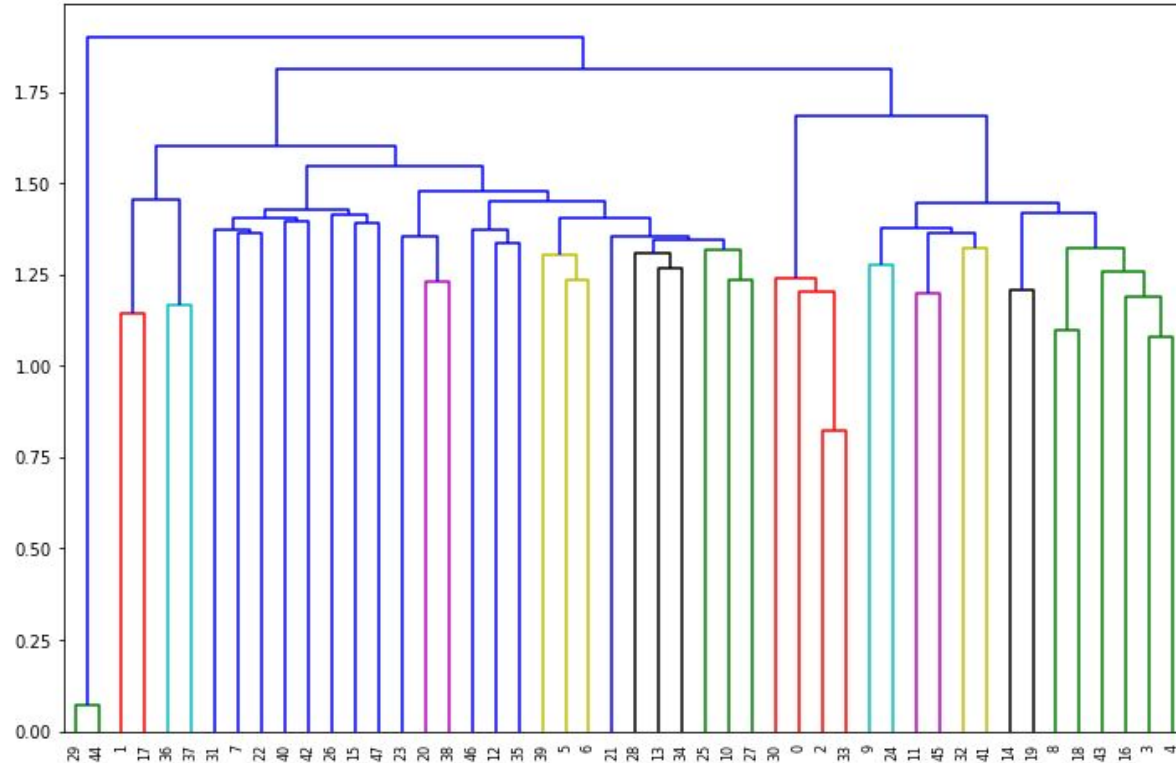
Datensatz: nur  
überdurchschnittlich  
lange Texte



# Hierarchical Clustering – Eminem & ABBA

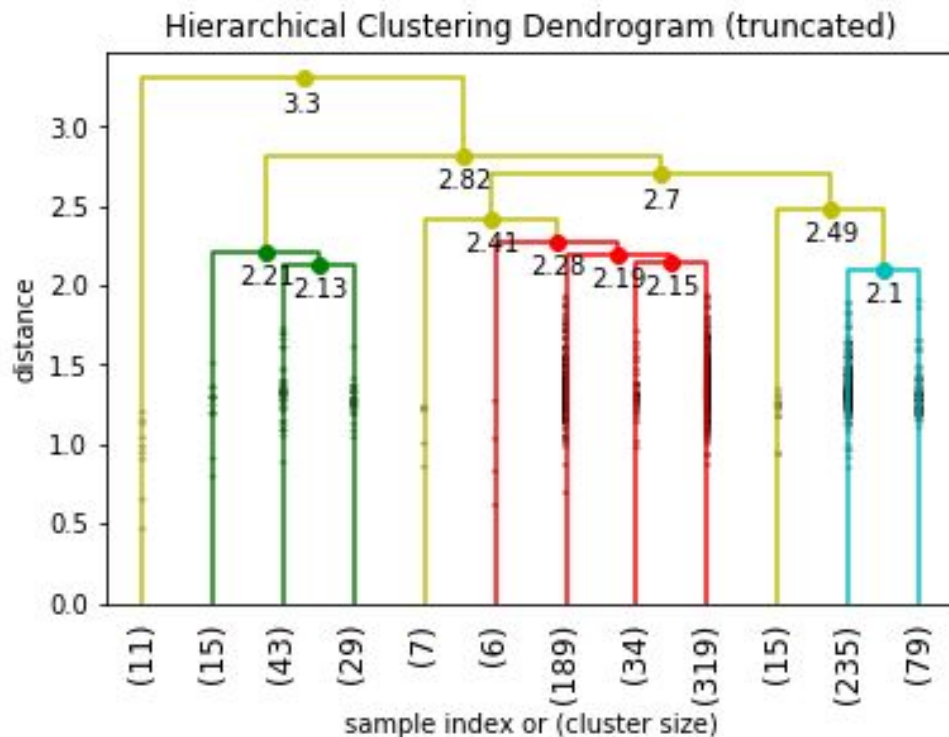


# Hierarchical Clustering – 1980: Electronic & Reggae

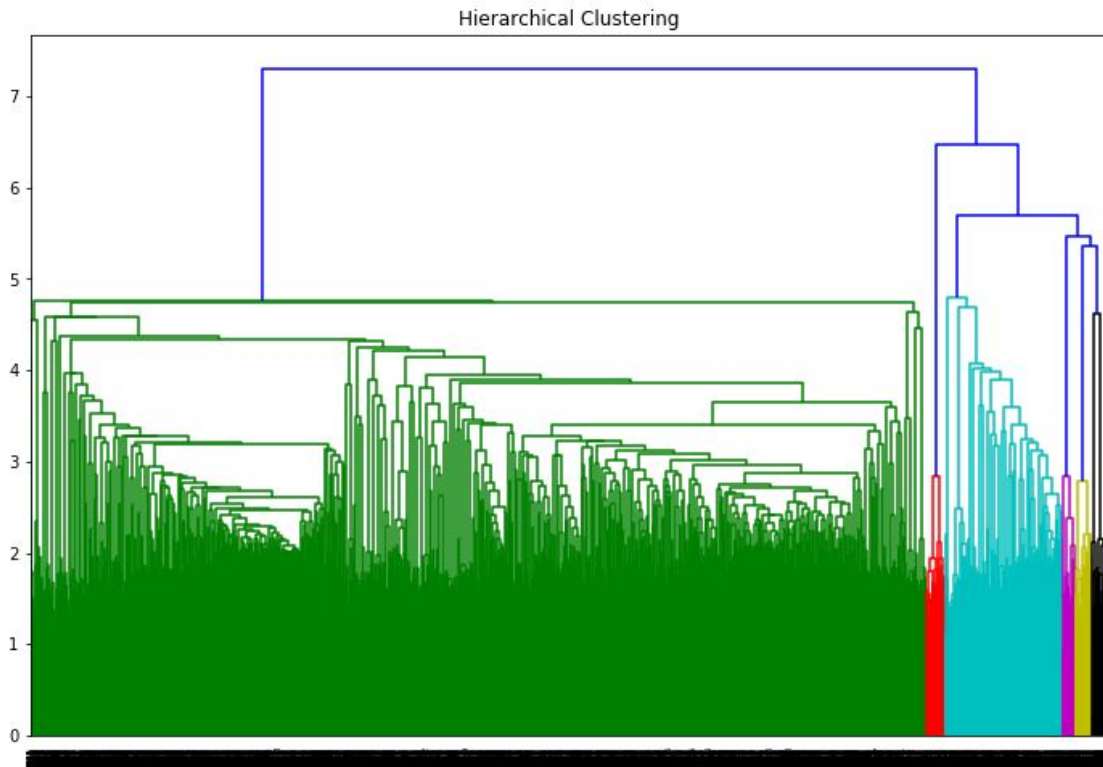




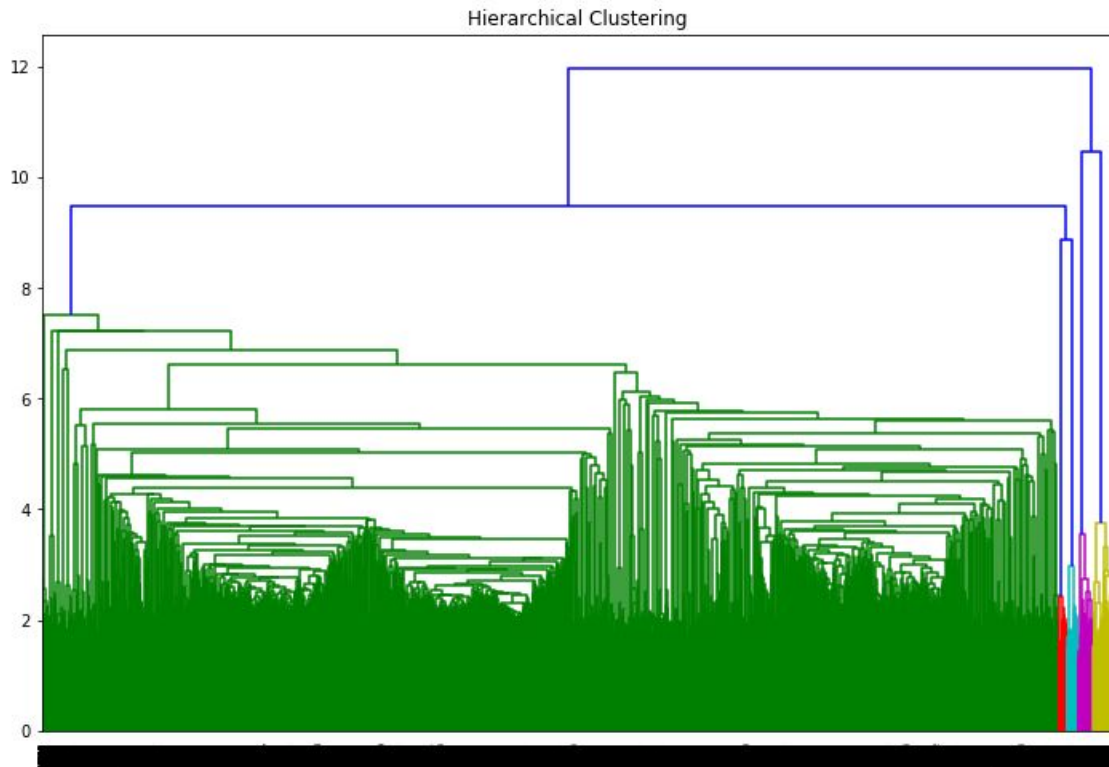
# Hierarchical Clustering – Latin & HipHop



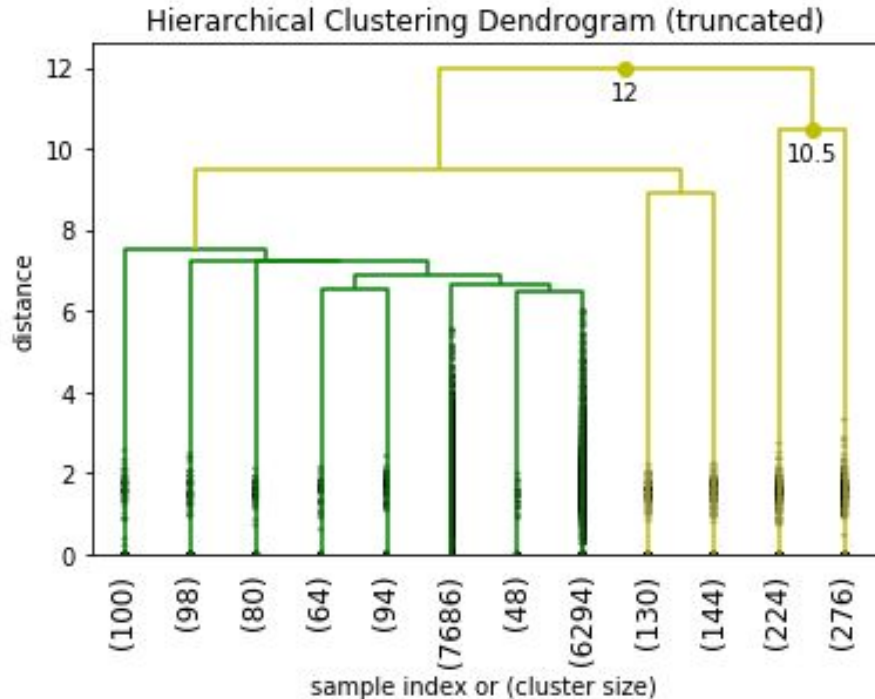
# Hierarchical Clustering – Pop (POS)



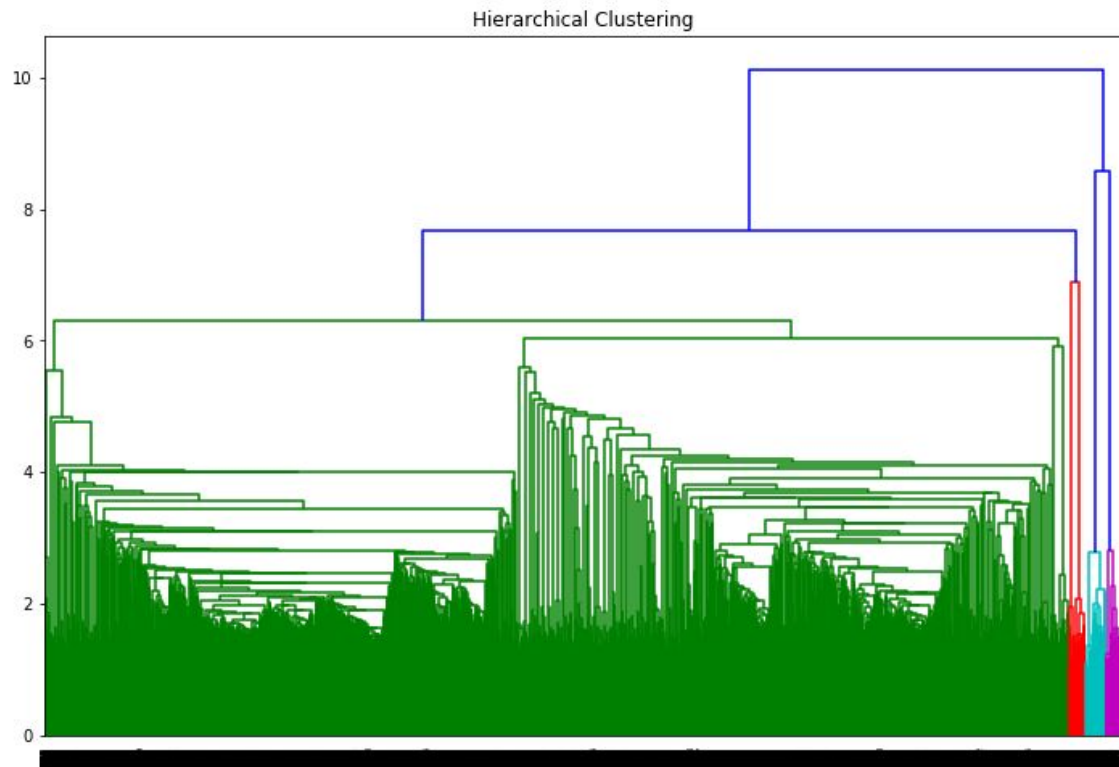
# Hierarchical Clustering – Rock (POS)



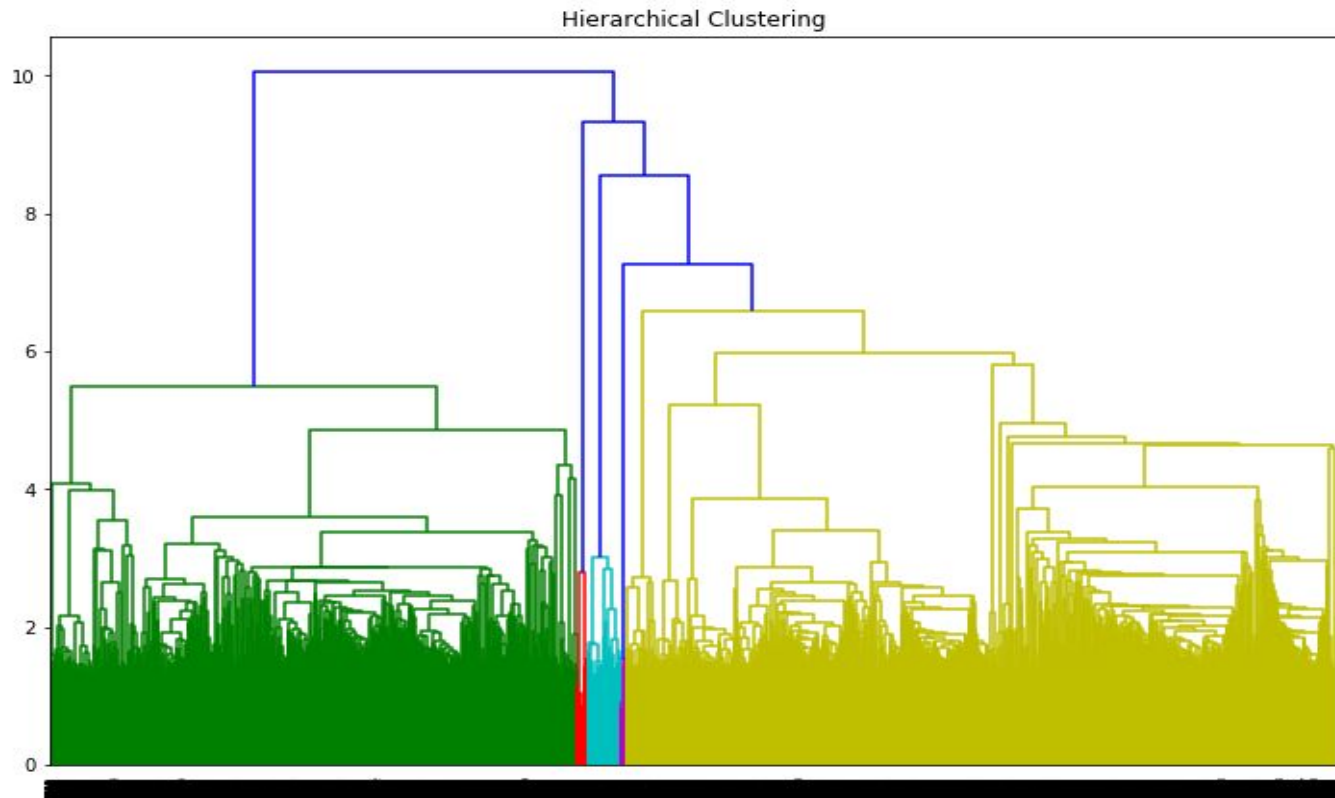
# Hierarchical Clustering – Rock (POS)



# Hierarchical Clustering – Rock & Pop (POS)



# Hierarchical Clustering – Rock & Pop (Text)



# Code – HC

```
In [139]: Z = linkage(tf_idf_array, 'ward')
          Z[:20]
```

```
Out[139]: array([[4.28000000e+02, 5.79000000e+02, 4.66649503e-01, 2.00000000e+00],
 [7.78000000e+02, 7.79000000e+02, 6.15168343e-01, 2.00000000e+00],
 [2.39000000e+02, 9.82000000e+02, 6.50724643e-01, 3.00000000e+00],
 [4.09000000e+02, 5.80000000e+02, 6.94752655e-01, 2.00000000e+00],
 [7.14000000e+02, 7.19000000e+02, 7.95616619e-01, 2.00000000e+00],
 [7.75000000e+02, 8.66000000e+02, 8.29488222e-01, 2.00000000e+00],
 [3.24000000e+02, 6.71000000e+02, 8.50526308e-01, 2.00000000e+00],
 [6.82000000e+02, 6.87000000e+02, 8.56495952e-01, 2.00000000e+00],
 [2.90000000e+02, 5.78000000e+02, 8.69025799e-01, 2.00000000e+00],
 [3.41000000e+02, 4.02000000e+02, 8.70202274e-01, 2.00000000e+00],
 [1.59000000e+02, 9.77000000e+02, 8.85721802e-01, 2.00000000e+00],
 [6.91000000e+02, 9.84000000e+02, 9.03952944e-01, 4.00000000e+00],
 [7.30000000e+02, 9.86000000e+02, 9.09266209e-01, 3.00000000e+00],
 [8.91000000e+02, 9.61000000e+02, 9.15191540e-01, 2.00000000e+00],
 [1.10000000e+02, 7.99000000e+02, 9.25897696e-01, 2.00000000e+00],
 [8.18000000e+02, 8.54000000e+02, 9.31039325e-01, 2.00000000e+00],
 [1.26000000e+02, 9.57000000e+02, 9.38272178e-01, 2.00000000e+00],
 [7.69000000e+02, 9.93000000e+02, 9.45615989e-01, 5.00000000e+00],
 [2.40000000e+01, 9.23000000e+02, 9.45918300e-01, 2.00000000e+00],
 [1.39000000e+02, 1.66000000e+02, 9.60723958e-01, 2.00000000e+00]])
```

```
In [141]: model = AgglomerativeClustering(n_clusters=None, distance_threshold=0).fit(tf_idf_array)
          link_matrix = linkage_matrix(tf_idf_array.shape[0], model.children_, model.distances_)
          link_matrix[:20]
```

```
Out[141]: array([[4.28000000e+02, 5.79000000e+02, 4.66649503e-01, 2.00000000e+00],
 [7.78000000e+02, 7.79000000e+02, 6.15168343e-01, 2.00000000e+00],
 [2.39000000e+02, 9.82000000e+02, 6.50724643e-01, 3.00000000e+00],
 [4.09000000e+02, 5.80000000e+02, 6.94752655e-01, 2.00000000e+00],
 [7.14000000e+02, 7.19000000e+02, 7.95616619e-01, 2.00000000e+00],
 [7.75000000e+02, 8.66000000e+02, 8.29488222e-01, 2.00000000e+00],
 [3.24000000e+02, 6.71000000e+02, 8.50526308e-01, 2.00000000e+00],
 [6.82000000e+02, 6.87000000e+02, 8.56495952e-01, 2.00000000e+00],
 [2.90000000e+02, 5.78000000e+02, 8.69025799e-01, 2.00000000e+00],
 [3.41000000e+02, 4.02000000e+02, 8.70202274e-01, 2.00000000e+00],
 [1.59000000e+02, 9.77000000e+02, 8.85721802e-01, 2.00000000e+00],
 [6.91000000e+02, 9.84000000e+02, 9.03952944e-01, 4.00000000e+00],
 [7.30000000e+02, 9.86000000e+02, 9.09266209e-01, 3.00000000e+00],
 [8.91000000e+02, 9.61000000e+02, 9.15191540e-01, 2.00000000e+00],
 [1.10000000e+02, 7.99000000e+02, 9.25897696e-01, 2.00000000e+00],
 [8.18000000e+02, 8.54000000e+02, 9.31039325e-01, 2.00000000e+00],
 [1.26000000e+02, 9.57000000e+02, 9.38272178e-01, 2.00000000e+00],
 [7.69000000e+02, 9.93000000e+02, 9.45615989e-01, 5.00000000e+00],
 [2.40000000e+01, 9.23000000e+02, 9.45918300e-01, 2.00000000e+00],
 [1.39000000e+02, 1.66000000e+02, 9.60723958e-01, 2.00000000e+00]])
```

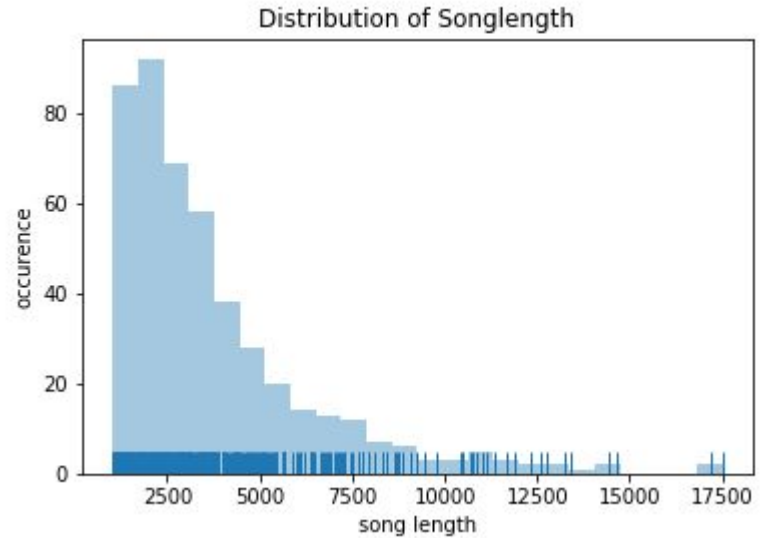
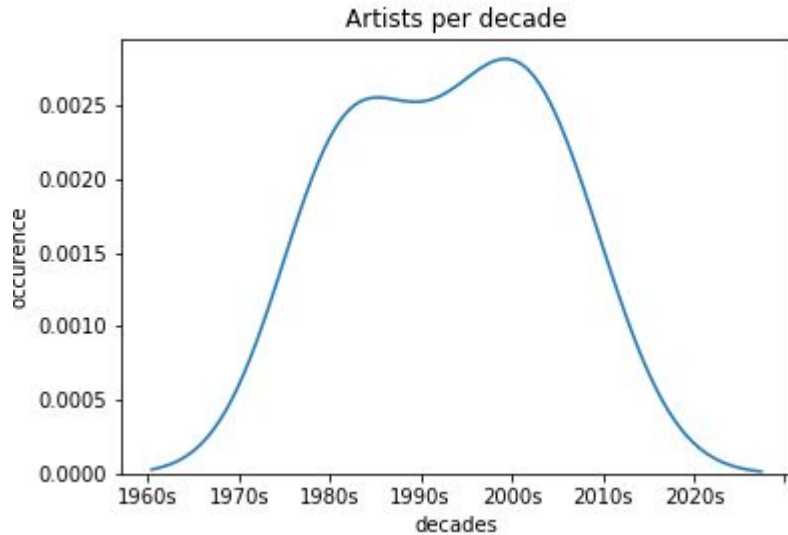
# Überlegungen

- Genre und Subgenre neu sortieren
- word embedding
- SVM Features hinzufügen

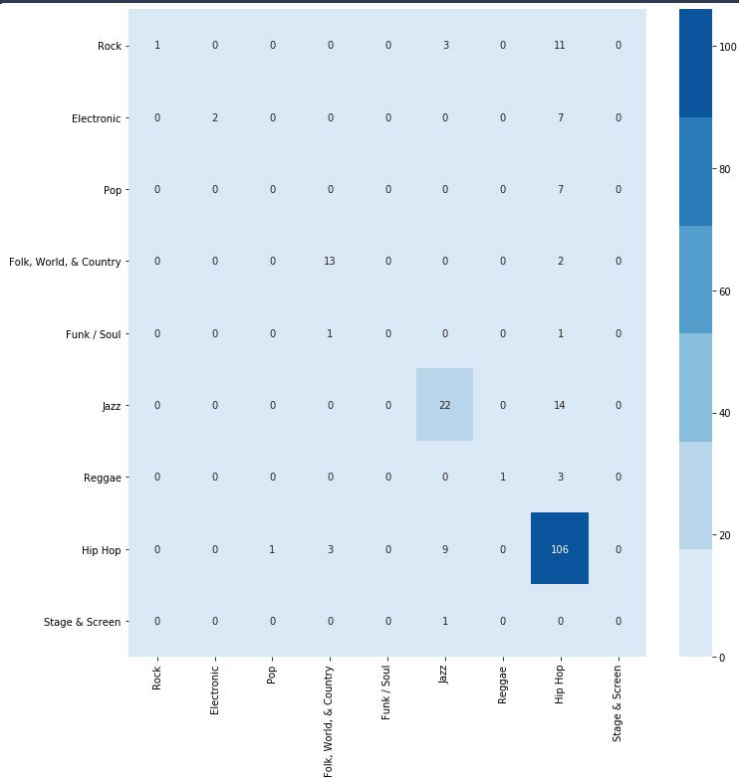


# neuer Datensatz

## - 466 Bands

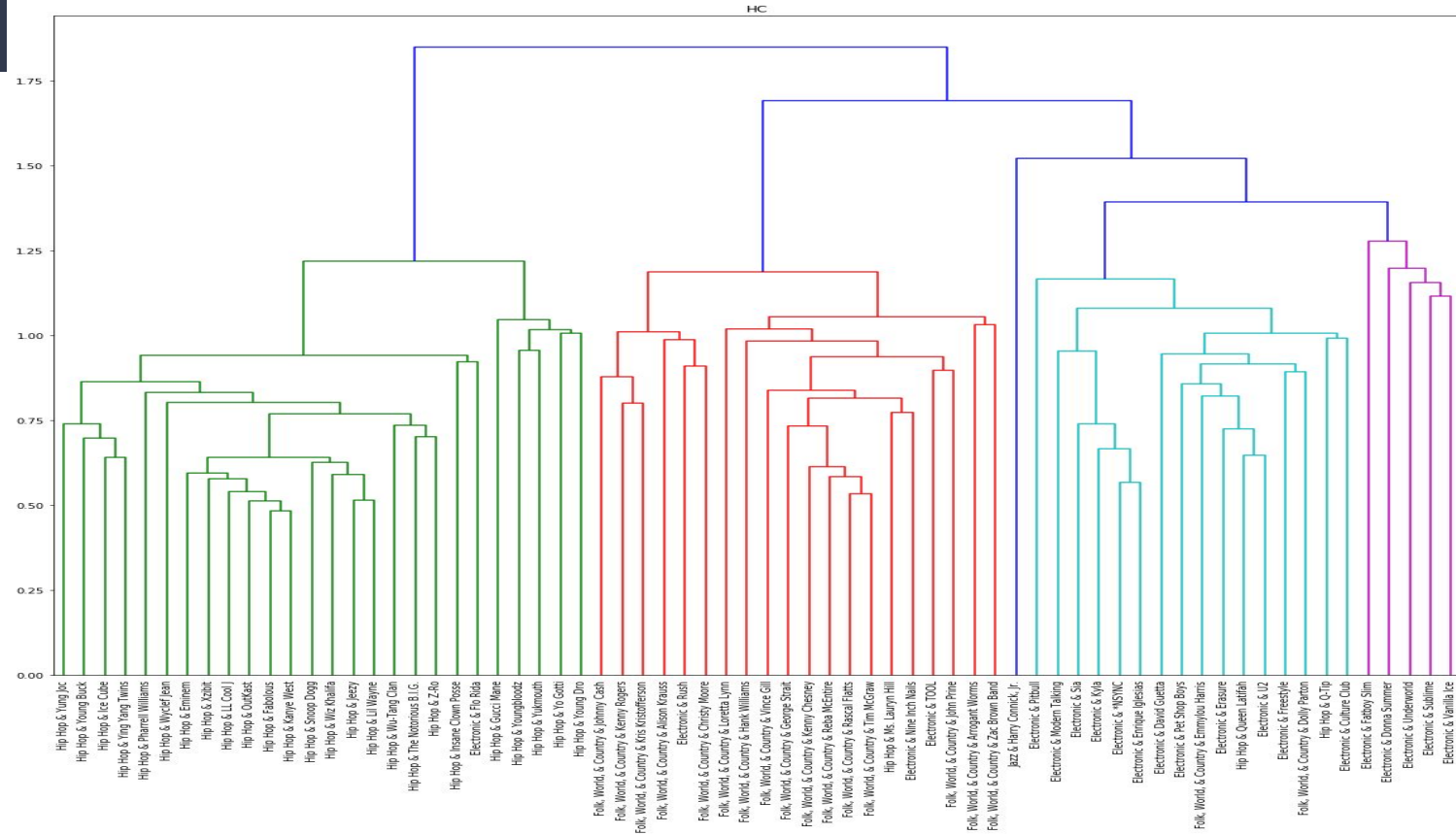


# SVM



	precision	recall	f1-score	support
Rock	1.00	0.07	0.12	15
Electronic	1.00	0.22	0.36	9
Pop	0.00	0.00	0.00	7
Folk, World, & Country	0.76	0.87	0.81	15
Funk / Soul	0.00	0.00	0.00	2
Jazz	0.63	0.61	0.62	36
Reggae	1.00	0.25	0.40	4
Hip Hop	0.70	0.89	0.79	119
Stage & Screen	0.00	0.00	0.00	1
accuracy			0.70	208
macro avg	0.57	0.32	0.35	208
weighted avg	0.70	0.70	0.65	208

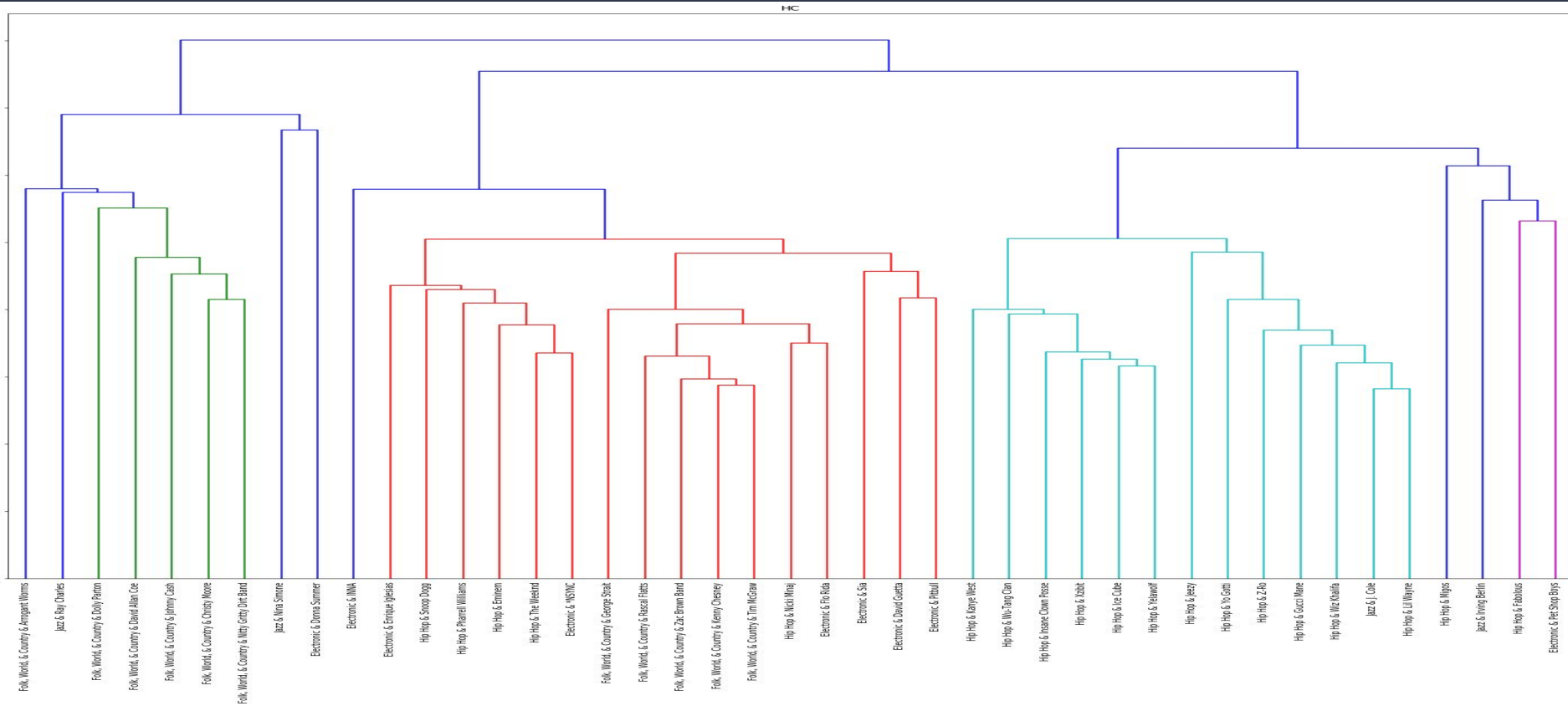
# HC - 2000



# Begründung für Ausreißer

- Flo Rida (Hip Hop) mit Electronic in Hip Hop : wird auch dem Genre EDM zugeordnet
- Nine Inch Nails (Electronic) mit Electronic in Country: für Country Award nominiert
- Tool (Electronic) in Country: laut Gruppenmitglied Adam Jones viele Einflüsse aus Country
- Queen Latifah ( Hip Hop) mit Hip Hop in Electronic: Album "Nature of sista" house music und electronic

# HC - 2010



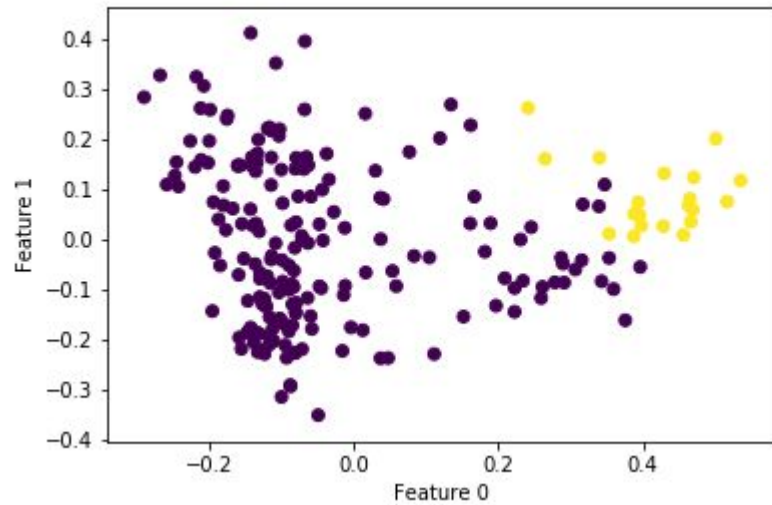
# Begründung für Ausreißer

- Enrique Iglesias mit Electronic in Hip Hop: Zusammenarbeit mit Hip Hop- Künstlern/ Hip Hop Reggaeton / Song 2008 Hip Hop und R&B
- Nicki Minaj mit Hip Hop in Electronic: Lieder werden teilweise dem Electro-House Genre zugeordnet und Zusammenarbeit mit David Guetta und Flo Rida (2010)
- J. Cole (Rapper) mit Jazz in Hip Hop: 4. Album hoher Jazz Einfluss
- Pet Shop Boys (Electro Pop) in Hip Hop: 2002-2005 verschieden Genre, unter anderem Hip Hop

# Ausreißer nach Wahrscheinlichkeit

artist	genre	decade	0	1	2	3
Nine Inch Nails	Electronic	2000s	0.0	2.706984e-239	1.000000e+00	0.0
George Strait	Folk, World, & Country	2000s	0.0	8.050388e-149	1.000000e+00	0.0
Enrique Iglesias	Electronic	2000s	0.0	1.000000e+00	2.388637e-125	0.0

# DBscan





Viktoria Ermisch: 1986364

Timo Günther: 2033581

Julia Jäger: 2124649

Teresa Kaiser: 2353056