In [3]:

```
import pandas as pd
import datetime
from datetime import datetime
import regex as re
import numpy as np
import os
```

In [4]:

```
small_files_path = r'random_songs_split_after_crawling'
```

In [5]:

```
# pattern of date and views as taken from youtube:
pattern = re.compile("^([A-Z][a-z]{1}[A-z])[ ](0?[1-9]|[12][0-9]|3[01])[ ]\d{4}$")
```

In [6]:

```
# concats youtube and google trends crawling data from many small CSVs.
def union_csvs(path):
    results_df = pd.DataFrame()
    for filename in os.listdir(path):
        if filename.endswith(".csv") and not filename.startswith('combinedfile'):
            file_df = pd.read_csv(path+'\\' +filename)
            results_df = results_df.append(file_df)
    return results_df.drop_duplicates(subset='track_id')

# normalizes
def normalize(df):
#     df['duration_ms_stand'] = np.where(df['duration_ms'] >= 10 * 60 * 1000, 10 * 60 *
1000, df['duration_ms'])
    df = df[df['duration_ms'] <= 10 * 60 * 1000]
    df['duration_norm'] = (df['duration_ms']-df['duration_ms'].min())/(df['duration_ms'
].max()-df['duration_ms'].min())

    df['loudness_stand'] = np.where(df['loudness'] <= -40, -40 , df['loudness'])
    df['loudness_norm'] = (df['loudness_stand']-df['loudness_stand'].min())/(df['loudne
ss_stand'].max()-df['loudness_stand'].min())

    df['tempo_norm'] = (df['tempo']-df['tempo'].min())/(df['tempo'].max()-df['tempo'].m
in())
    return df

# calculates the number of days passed since the clip was uploaded to youtube
def days(strdate):
    strdate = strdate.lstrip()
    date = datetime.strptime(strdate, '%b %d %Y')
    date_oct = datetime.strptime('Oct 1 2018', '%b %d %Y')

    if date_oct <= date:
        return -1

    return (date.today() - date).days

# extracts the number of views and number of days since file was uploaded
def extract_dates_and_views(data):
    print(str(len(data)) + ' before filtering')

    data = data[data['date_and_views'] != "(None, None)"]
    print(str(len(data)) + ' after removing rows when views and dates are both None')
    data['str_dv'] = data['date_and_views'].astype(str)
    data['str_dv'] = data.str_dv.apply(lambda x: x.replace('(','').replace(')','',).rep
lace(',','-',1).split('-'))
    data['views'] = data.str_dv.apply(lambda x: x[0].replace("'", ''))
    data = data[data['views'].apply(lambda x: x.isdigit())]
    print(str(len(data)) + ' after removing rows when views is not a digit')
    data['views'] = data.views.apply(lambda x: int(x))
    data['upload_date'] = data.str_dv.apply(lambda x: x[1].replace(',','').replace('on'
,'').replace("'",'').lstrip())
    data = data[data['upload_date'] != "None"]
    print(str(len(data)) + ' after removing rows where date is None')
    data = data[data.upload_date.apply(lambda s: pattern.match(s) != None)]
    print(str(len(data)) + ' after removing rows where pattern does not match')
    data['days_since_upload'] = data.upload_date.apply(lambda x: days(x))
    data = data[data['days_since_upload'] >0]
    print(str(len(data)) + ' after removing rows where song is too new')
    return data.drop('str_dv',axis=1)
```

```
def add_popularity_measures(data):
    data['days_views_ratio'] = data['views']/ data['days_since_upload']
    data['days_views_ratio'] = np.where(data['days_views_ratio'] >= 10000, 10000, data[
'days_views_ratio'])
    data['youtube_popularity'] = np.ceil(((data['days_views_ratio'] - (data['days_views
_ratio'].min())) / \
                                (data['days_views_ratio'].max() - data['days_views_ratio']
.min()))*100)
    data['spotify_popularity'] = np.where(data['popularity'] == 0, 1, data['popularity'
])
    data['youtube_spotify_popularity_ratio'] = data['youtube_popularity']/ data['spotif
y_popularity']
    return data
```

csvs were split to small files with 100 rows each in order to perform the crawling. the first thing that we do is concat all of them:

In [7]:

```
raw_data = union_csvs(small_files_path).drop(labels = ['Unnamed: 0','Unnamed: 0.1'],axi
s =1 )
raw_data.sample(3)
```

Out[7]:

	artist_name	track_id	track_name	acousticness	danceability	dura
25	Tray Loop	5UGZizzQ5jwuucU36aXrF9	Fuck You	0.168	0.633	
10	DragonFlex	1Y1jVQKIoLuKoWgUckIVMI	Flags	0.995	0.438	
82	Adrian Von Ziegler	2xKqGMDMf0XeO6Amivi3CF	Till Valhöll	0.529	0.363	

3 rows × 271 columns

some coloumns need to be normalized between 0 to 1, and songs longer than 10 minutes are removed (mostly classical music or podcasts):

In [8]:

```
normalized = normalize(raw_data)
normalized.head(3)
```

```
c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy

c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:16: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
  app.launch_new_instance()
c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:17: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:19: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
```

Out[8]:

	artist_name	track_id	track_name	acousticness	danceability	duration_
0	YG	2RM4jf1Xa9zPgMGRDiht8O	Big Bank feat. 2 Chainz, Big Sean, Nicki Minaj	0.00582	0.743	238
1	YG	1tHDG53xJNGsItRA3vfVgs	BAND DRUM (feat. A$AP Rocky)	0.02440	0.846	214
2	R3HAB	6Wosx2euFPMT14UXiWudMy	Radio Silence	0.02500	0.603	138

3 rows × 275 columns

youtube upload dates and number of views are extrcted, and filtering is done for invalid values:

In [9]:

```
filtered = extract_dates_and_views(normalized)
filtered.head(3)
```

```
42138 before filtering
40742 after removing rows when views and dates are both None

c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:39: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:40: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy
c:\users\netta\appdata\local\programs\python\python35\lib\site-packages\ip
ykernel_launcher.py:41: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: http://pandas.pydata.org/pandas-doc
s/stable/indexing.html#indexing-view-versus-copy

40452 after removing rows when views is not a digit
40452 after removing rows where date is None
40278 after removing rows where pattern does not match
37234 after removing rows where song is too new
```

Out[9]:

	artist_name	track_id	track_name	acousticness	danceability	duration_
0	YG	2RM4jf1Xa9zPgMGRDiht8O	Big Bank feat. 2 Chainz, Big Sean, Nicki Minaj	0.00582	0.743	238
1	YG	1tHDG53xJNGsItRA3vfVgs	BAND DRUM (feat. A$AP Rocky)	0.02440	0.846	214
2	R3HAB	6Wosx2euFPMT14UXiWudMy	Radio Silence	0.02500	0.603	138

3 rows × 278 columns

popularity measures are added according to the ratio of views and date since upload. the long tail is cut and the ration is normalized:

In [10]:

```
with_popularity = add_popularity_measures(filtered)
with_popularity.head(3)
```

Out[10]:

	artist_name	track_id	track_name	acousticness	danceability	duration_
0	YG	2RM4jf1Xa9zPgMGRDiht8O	Big Bank feat. 2 Chainz, Big Sean, Nicki Minaj	0.00582	0.743	238
1	YG	1tHDG53xJNGsItRA3vfVgs	BAND DRUM (feat. A$AP Rocky)	0.02440	0.846	214
2	R3HAB	6Wosx2euFPMT14UXiWudMy	Radio Silence	0.02500	0.603	138

3 rows × 282 columns

redundant columns are removed and csv is exported:

In [11]:

```
with_popularity.artist_popularity = with_popularity.artist_popularity/100
with_popularity.to_csv("random_songs_processed_dataset.csv",encoding='utf-8',index = Fa
lse)
```