In [1]:
```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from surprise import Dataset, Reader
from surprise import accuracy
from surprise.model_selection import train_test_split
from surprise.prediction_algorithms import knns
from surprise.similarities import cosine,pearson
from surprise import accuracy
```

In [2]:
```python
#Caricamento dataset
song_df_1 = pd.read_csv('triplets_file.csv')
song_df_1.columns = ['user_id', 'song_id', 'listen_count']

song_df_2 =  pd.read_csv('song_data.csv')
song_df_2.drop_duplicates(['song_id'], inplace=True)

#Unione dataset
df_songs = pd.merge(song_df_1, song_df_2, on="song_id", how="left")
```

In [3]:
```python
# Numero canzoni ascoltate da ogni utente
user_counts = df_songs.groupby('user_id')['song_id'].count()

# Utenti che hanno ascoltato almeno 30 canzoni
user_ten_id = user_counts[user_counts > 30].index.to_list()
```

In [4]:
```python
# Numero di utenti che hanno ascoltato ogni canzone
song_counts = df_songs.groupby('song_id')['user_id'].count()

# Canzoni ascoltate da almeno 500 utenti
song_ten_id = song_counts[song_counts > 500].index.to_list()
```

In [5]:
```python
df_songs_ridotto = df_songs[(df_songs['user_id'].isin(user_ten_id)) & (df_songs['song_id'].isin(song_ten_id))].reset_index(drop=True)
df_songs_ridotto
```

Out[5]:

	user_id	song_id	listen_count	title	release	artist_name	year
0	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBXHDL12A81C204C0	1	Stronger	Graduation	Kanye West	2007
1	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOBYHAJ12A6701BF1D	1	Constellations	In Between Dreams	Jack Johnson	2005
2	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODACBL12A8C13C273	1	Learn To Fly	There Is Nothing Left To Lose	Foo Fighters	1999
3	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SODXRTY12AB0180F3B	1	Paper Gangsta	The Fame Monster	Lady GaGa	2008
4	b80344d063b5ccb3212f76538f3d9e43d87dca9e	SOFRQTD12A81C233C0	1	Sehr kosmisch	Musik von Harmonia	Harmonia	0
...	...	...	...	...	...	...	...
336425	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOHZPIK12A58A7CCAE	4	Te Amo	Rated R	Rihanna	2009
336426	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOIDDNJ12AC9071B90	1	That Should Be Me	My Worlds	Justin Bieber	2010
336427	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOIOESO12A6D4F621D	4	Unwell (Album Version)	More Than You Think You Are	matchbox twenty	2003
336428	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJKQSF12A6D4F5EE9	3	What I've Done (Album Version)	What I've Done	Linkin Park	2007
336429	d8bfd4ec88f0f3773a9e022e3c1a0f1d3b7b6a92	SOJUXGA12AC961885C	1	Up	My Worlds	Justin Bieber	2010

336430 rows × 7 columns

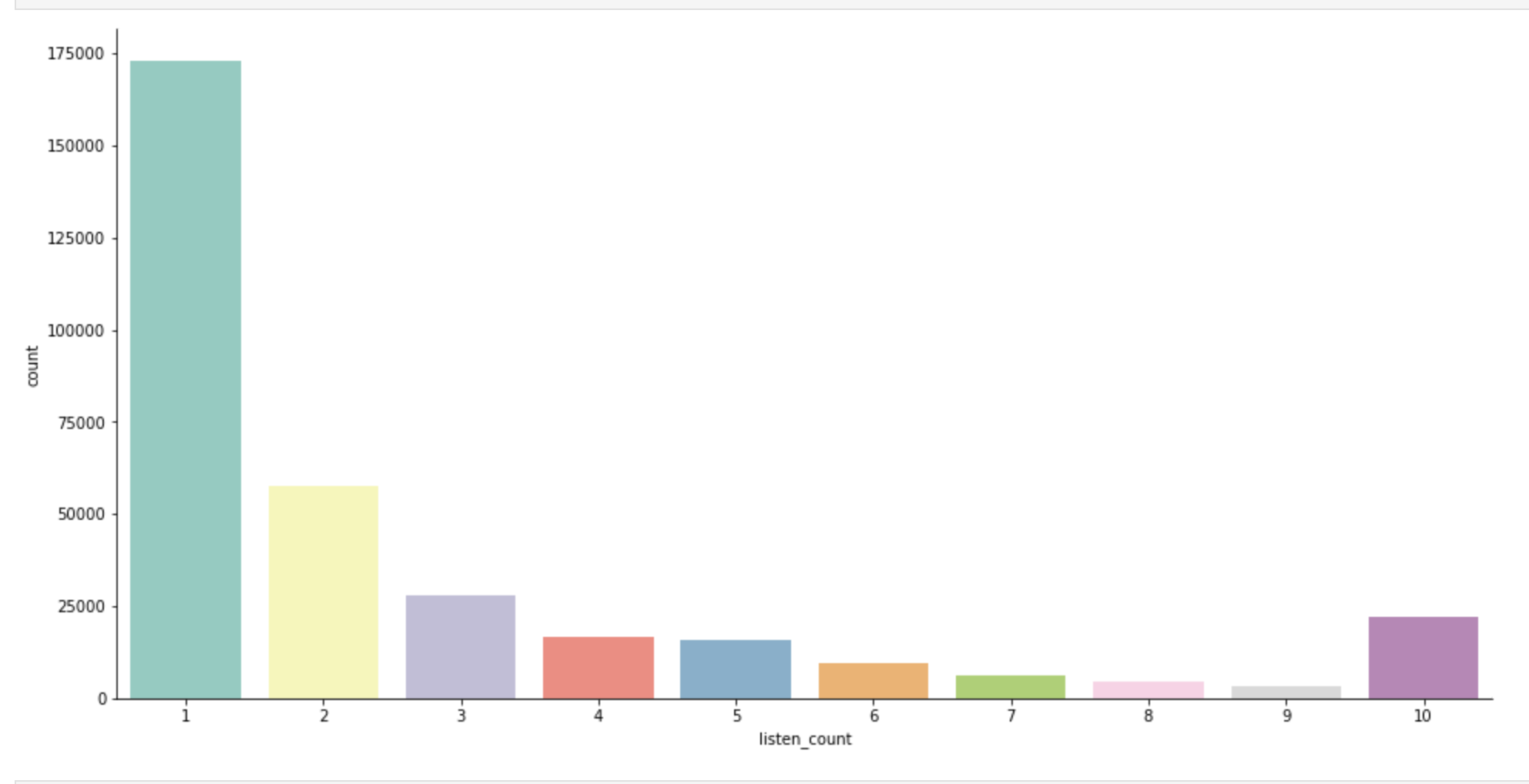
In [52]:
```python
# Creazione scala valutazione

scala = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 2214]

df_songs_ridotto ['listen_count'] = pd.cut(df_songs_ridotto ['listen_count'], bins=scala, labels=[1, 2, 3, 4, 5, 6, 7, 8, 9, 10])

listen_counts = pd.DataFrame(df_songs_ridotto .groupby('listen_count').size(), columns=['count']).reset_index(drop=False)

plt.figure(figsize=(16, 8))
sns.barplot(x='listen_count', y='count', palette='Set3', data=listen_counts)
plt.gca().spines['top'].set_visible(False)
plt.gca().spines['right'].set_visible(False)
plt.show();
```



In [7]:
```python
reader = Reader(rating_scale=(1, 10))
data = Dataset.load_from_df(df_songs_ridotto [['user_id', 'title', 'listen_count']], reader)
```

In [8]:
```python
train_set, test_set = train_test_split(data, test_size=.25)
```

In [ ]:
```python
# Funzione get top n predictions di Surprise
from collections import defaultdict
def get_top_n(predictions, n=10):

    # First map the predictions to each user.
    top_n = defaultdict(list)
    for uid, iid, r_ui, est, _ in predictions:
        top_n[uid].append((iid, est, r_ui))

    # Then sort the predictions for each user and retrieve the k highest ones.
    for uid, user_ratings in top_n.items():
        # print(user_ratings)
        user_ratings = sorted(user_ratings,key=lambda x: x[1], reverse=True)
        top_n[uid] = user_ratings[:n]

    return top_n
```

KNN BASIC

In [9]:
```python
# Definizione indice similarità
sim_cos = {'name':'cosine', 'user_based':False}
```

In [10]:
```python
# Addestramento del modello
Kbasic = knns.KNNBasic(sim_options=sim_cos)
Kbasic.fit(train_set)
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
```

Out[10]:
```
<surprise.prediction_algorithms.knns.KNNBasic at 0x278b7b79910>
```

In [11]:
```python
# Test del modello
predictions = Kbasic.test(test_set)
```

In [12]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions)
mae = accuracy.mae(predictions)
```

```
RMSE: 2.3775
MAE:  1.6536
```

In [14]:
```python
# Generazione top n previsioni
top_pred_knn = get_top_n(predictions,10)
```

In [15]:
```python
top_pred_knn_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_knn.items() for pair in row],columns=["userId" ,"title","rat_pred"])
top_pred_knn_df
```

Out[15]:

	userId	title	rat_pred
0	ec92f6866e661261652f902a0193c05c7dea197b	Hochmah (Interlude)	2.362388
1	ec92f6866e661261652f902a0193c05c7dea197b	Forgive Me	2.319035
2	ec92f6866e661261652f902a0193c05c7dea197b	You Belong With Me	2.201268
3	f0bab3b7337264c5bd3d482e69294f5d896dcf69	Mia	2.167763
4	f0bab3b7337264c5bd3d482e69294f5d896dcf69	The District Sleeps Alone Tonight (Album)	1.984358
...	...	...	...
73785	9bb4d212405e4d9cf45a35ee31fc5e114f62c51e	The Funeral (Album Version)	1.239380
73786	f36fcce478855992603edd192dc7aa1e012cfe9e	Una Confusion	2.509336
73787	75bcbe7587cd0ef9609e097ae2175bedefc76cae	Burden In My Hand	1.450015
73788	7d1cdbf9efdef869fa2054e42cc48624354606eb	Revelry	1.943061
73789	530dccd19f1f4fd61072e33840a679cd4ba3d59a	Jamaica Roots II(Agora E Sempre)	3.008972

73790 rows × 3 columns

In [16]:
```python
top_pred_knn_df.loc[top_pred_knn_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[16]:

	title	rat_pred
11402	Life In Technicolor ii	5.237305
11403	Clocks	5.139316
11404	Don't Panic	5.063960
11405	Fix You	4.984168
11406	Lost!	4.881390
11407	Every Little Thing She Does Is Magic	4.880238
11408	The Ballad of Michael Valentine	4.860616
11409	I Kissed A Girl	4.785901
11410	Trouble	4.754914
11411	The District Sleeps Alone Tonight (Album)	4.717154

KNN MEANS MODEL

In [17]:
```python
# Addestramento del modello
KNNMeans = knns.KNNWithMeans(sim_options=sim_cos)
KNNMeans.fit(train_set)
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
```

Out[17]:
```
<surprise.prediction_algorithms.knns.KNNWithMeans at 0x278d555fdf0>
```

In [18]:
```python
# Test del modello
predictions_mean = KNNMeans.test(test_set)
```

In [19]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions_mean)
mae = accuracy.mae(predictions_mean)
```

```
RMSE: 2.3138
MAE:  1.6075
```

In [20]:
```python
# Generazione top n previsioni
top_pred_mean = get_top_n(predictions_mean,10)
```

In [21]:
```python
top_pred_mean['b048f21afd5e7467f187bf9f9d413e97c32313a9']
```

Out[21]:
```
[('Clocks', 5.204273056107248, 10.0),
 ('Life In Technicolor ii', 5.172153169602957, 4.0),
 ('Lost!', 5.078141236192131, 8.0),
 ('Fix You', 5.009313135000344, 7.0),
 ('Elephant Gun', 4.961543938828061, 1.0),
 ('I Kissed A Girl', 4.944117178332749, 3.0),
 ("Don't Panic", 4.914951116323143, 10.0),
 ('Trouble', 4.767175079463666, 1.0),
 ('Every Little Thing She Does Is Magic', 4.662350235045847, 5.0),
 ('The District Sleeps Alone Tonight (Album)', 4.363397796585078, 1.0)]
```

In [22]:
```python
top_pred_mean_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_mean.items() for pair in row],columns=["userId" ,"title","rat_pred"])

top_pred_mean_df.loc[top_pred_mean_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[22]:

	title	rat_pred
11402	Clocks	5.204273
11403	Life In Technicolor ii	5.172153
11404	Lost!	5.078141
11405	Fix You	5.009313
11406	Elephant Gun	4.961544
11407	I Kissed A Girl	4.944117
11408	Don't Panic	4.914951
11409	Trouble	4.767175
11410	Every Little Thing She Does Is Magic	4.662350
11411	The District Sleeps Alone Tonight (Album)	4.363398

KNN Z SCORE MODEL

In [23]:
```python
# Addestramento del modello
KNN_Z = knns.KNNWithZScore(sim_options=sim_cos)
KNN_Z.fit(train_set)
```

```
Computing the cosine similarity matrix...
Done computing similarity matrix.
```

Out[23]:
```
<surprise.prediction_algorithms.knns.KNNWithZScore at 0x278da8ee970>
```

In [24]:
```python
# Test del modello
predictions_Z = KNN_Z.test(test_set)
```

In [25]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions_Z)
mae = accuracy.mae(predictions_Z)
```

```
RMSE: 2.3251
MAE:  1.5997
```

In [26]:
```python
# Generazione top n previsioni
top_pred_Z = get_top_n(predictions_Z,10)
```

In [27]:
```python
top_pred_Z_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_Z.items() for pair in row],columns=["userId" ,"title","rat_pred"])

top_pred_Z_df.loc[top_pred_Z_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[27]:

	title	rat_pred
11402	Elephant Gun	5.571085
11403	Lost!	5.440697
11404	Clocks	5.354612
11405	I Kissed A Girl	5.281291
11406	Life In Technicolor ii	5.156917
11407	Fix You	5.015369
11408	Don't Panic	4.916655
11409	Trouble	4.850568
11410	Every Little Thing She Does Is Magic	4.462821
11411	We Will Become Silhouettes (Album)	4.061760

USER BASED

BASIC KNN

In [28]:
```python
# Definizione indice similarità
sim_pearson = {'name':'pearson', 'user_based':True}
```

In [29]:
```python
# Addestramento del modello
KNN_basic_u = knns.KNNBasic(sim_options=sim_pearson)
KNN_basic_u.fit(train_set)
```

```
Computing the pearson similarity matrix...
Done computing similarity matrix.
```

Out[29]:
```
<surprise.prediction_algorithms.knns.KNNBasic at 0x278b2c5d3a0>
```

In [30]:
```python
# Test del modello
predictions_basic_u = KNN_basic_u.test(test_set)
```

In [31]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions_basic_u)
mae = accuracy.mae(predictions_basic_u)
```

```
RMSE: 2.5794
MAE:  1.9333
```

In [32]:
```python
# Generazione top n previsioni
top_pred_basic_u = get_top_n(predictions_basic_u,10)
```

In [33]:
```python
top_pred_basic_u_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_basic_u.items() for pair in row],columns=["userId" ,"title","rat_pred"])

top_pred_basic_u_df.loc[top_pred_basic_u_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[33]:

	title	rat_pred
11402	Life In Technicolor ii	2.874059
11403	Lost!	2.869134
11404	Fix You	2.625000
11405	Clocks	2.500000
11406	I Kissed A Girl	2.485766
11407	Elephant Gun	2.447237
11408	Don't Panic	2.213656
11409	Trouble	2.206178
11410	Every Little Thing She Does Is Magic	1.801824
11411	The Ballad of Michael Valentine	1.695871

WITH MEANS

In [34]:
```python
# Addestramento del modello
KNNMeans_u = knns.KNNWithMeans(sim_options=sim_pearson)
KNNMeans_u.fit(train_set)
```

```
Computing the pearson similarity matrix...
Done computing similarity matrix.
```

Out[34]:
```
<surprise.prediction_algorithms.knns.KNNWithMeans at 0x278df6799d0>
```

In [35]:
```python
# Test del modello
predictions_mean_u = KNNMeans_u.test(test_set)
```

In [36]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions_mean_u)
mae = accuracy.mae(predictions_mean_u)
```

```
RMSE: 2.3704
MAE:  1.6308
```

In [37]:
```python
# Generazione top n previsioni
top_pred_mean_u = get_top_n(predictions_mean_u,10)
```

In [38]:
```python
top_pred_mean_u_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_mean_u.items() for pair in row],columns=["userId" ,"title","rat_pred"])

top_pred_mean_u_df.loc[ top_pred_mean_u_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[38]:

	title	rat_pred
11402	Lost!	4.962950
11403	Fix You	4.747983
11404	Life In Technicolor ii	4.747067
11405	Elephant Gun	4.707179
11406	Don't Panic	4.561623
11407	I Kissed A Girl	4.486354
11408	Clocks	4.446981
11409	We Will Become Silhouettes (Album)	4.400861
11410	Trouble	4.376889
11411	The Ballad of Michael Valentine	4.280665

Z SCORE KNN

In [39]:
```python
# Addestramento del modello
KNN_Z_u = knns.KNNWithZScore(sim_options=sim_pearson)
KNN_Z_u.fit(train_set)
```

```
Computing the pearson similarity matrix...
Done computing similarity matrix.
```

Out[39]:
```
<surprise.prediction_algorithms.knns.KNNWithZScore at 0x278e1aff2b0>
```

In [40]:
```python
# Test del modello
predictions_Z_u = KNN_Z_u.test(test_set)
```

In [41]:
```python
# Metriche di valutazione
rmse = accuracy.rmse(predictions_Z_u)
mae = accuracy.mae(predictions_Z_u)
```

```
RMSE: 2.3645
MAE:  1.6052
```

In [42]:
```python
# Generazione top n previsioni
top_pred_Z_u = get_top_n(predictions_Z_u,10)
```

In [43]:
```python
top_pred_Z_u_df = pd.DataFrame([(id, pair[0],pair[1]) for id, row in top_pred_Z_u.items() for pair in row],columns=["userId" ,"title","rat_pred"])

top_pred_Z_u_df.loc[top_pred_Z_u_df['userId'] == 'b048f21afd5e7467f187bf9f9d413e97c32313a9', ('title', 'rat_pred')]
```

Out[43]:

	title	rat_pred
11402	Lost!	5.377371
11403	Fix You	4.691215
11404	Life In Technicolor ii	4.562340
11405	Clocks	4.415978
11406	I Kissed A Girl	4.310783
11407	Don't Panic	4.260575
11408	Elephant Gun	4.042171
11409	The Ballad of Michael Valentine	3.865661
11410	We Will Become Silhouettes (Album)	3.865249
11411	Every Little Thing She Does Is Magic	3.760643

TABELLA CONFRONTO

In [50]:
```python
accuracy_tab = {'RMSE': [2.3775, 2.3138,  2.3251, 2.5794, 2.3704, 2.3645],
        'MAE': [ 1.6536, 1.6075,  1.5997, 1.9333, 1.6308, 1.6052]
        }

accuracy_tab_df = pd.DataFrame(accuracy_tab, index=['Basic_I','WithMeans_I','Zscore_I','Basic_U','WithMeans_U','Zscore_U'])
display(accuracy_tab_df)
```

	RMSE	MAE
Basic_I	2.3775	1.6536
WithMeans_I	2.3138	1.6075
Zscore_I	2.3251	1.5997
Basic_U	2.5794	1.9333
WithMeans_U	2.3704	1.6308
Zscore_U	2.3645	1.6052

In [ ]: