PROJECT 02: PHÂN TÍCH DỮ LIỆU

- Lê Minh Hữu - 19120525
- Ninh Duy Huy - 19120533
- Cao Thanh Khiết - 19120544
- Nguyễn Tuấn Khoa - 19120547
- Trần Tuấn Kiệt - 19120557

Phân công công việc

- Lê Minh Hữu: tiền xử lí dữ liệu, đặt câu hỏi 5
- Nguyễn Tuấn Khoa: đặt câu hỏi 1, 2
- Cao Thanh Khiết: đặt câu hỏi 3, review code
- Ninh Duy Huy: đặt câu hỏi 6
- Trần Tuấn Kiệt: đặt câu hỏi 7, 8

Cấu trúc thư mục

- data folder: chứa file dữ liệu lấy từ API soundcloud
- cleaned_data folder: chứa file dữ liệu sau khi 1 vài bước tiền xử lí cơ bản (xóa các hàng hoặc cột khi mất nhiều giá trị hơn ngưỡng cho trước)
- source.ipynb: file source code

TIỀN XỬ LÝ DỮ LIỆU

In [1]:
```python
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

In [2]:
```python
playlists_df = pd.read_csv("data/playlists.csv", sep=",")
tracks_df = pd.read_csv("data/tracks.csv", sep=",")
users_df = pd.read_csv("data/users.csv", sep=",")

# configure matplotlib
plt.rcParams["figure.figsize"] = (15, 10)
font = {'family': 'serif',
        'color':  'blue',
        }
plt.rcParams['font.size'] = 20
```

In [3]:
```python
### Thông tin về số dòng và cột của dữ liệu ban đầu
print(playlists_df.shape)
print(tracks_df.shape)
print(users_df.shape)
```

```
(608, 3)
(6979, 45)
(422, 33)
```

In [4]:
```python
# delete some row with missing values more than the given ratio
def preprocess_data(old_df, ratio):
    '''
    Hàm này sẽ tiền xử lí dữ liệu, loại bỏ một vài dòng hoặc cột khi mất mát dữ liệu quá tỉ lệ cho trước
    '''
    attributes = list(old_df.columns)
    row = len(old_df.index)
    col = len(attributes)

    a = []
    arr=[]
    for i in range(row):
        for j in range(col):
            a.append(old_df[attributes[j]][i])
        arr.append(a)
        a = []

    result_arr = []
    temp_a = []
    for i in range(row):
        for j in range(col):
            temp_a.append(arr[i][j])
        result_arr.append(temp_a)
        temp_a = []
# xóa dòng
    for value in arr:
        count_missing = 0
        for i in value:
            if i == 'None' or i!=i :
                count_missing +=1
        if (count_missing / col) > ratio:
            result_arr.remove(value)
# xóa cột
    new_row = len(result_arr)
    temp_arr = []
    temp_a = []
    for i in range(new_row):
        for j in range(col):
            temp_a.append(result_arr[i][j])
        temp_arr.append(temp_a)
        temp_a = []

    count_del_col = 0 #biến đếm các cột đã bị xóa
    for i in range(col):
        count_missing = 0
        for j in range(new_row):
            if temp_arr[j][i] != temp_arr[j][i] or temp_arr[j][i] == 'None':
                count_missing += 1
        if (count_missing /new_row) > ratio:
            del attributes[i-count_del_col]
            for r in result_arr:
                del r[i-count_del_col]
            count_del_col += 1

    # tạo một DataFrame mới để lưu kết quả
    new_data = {}
    new_row = len(result_arr)
    for i in range(len(attributes)):
        new_data[attributes[i]] = [result_arr[j][i] for j in range(new_row)]

    new_df = pd.DataFrame(new_data, columns = attributes)
    return new_df
```

In [5]:
```python
'''
Tiền xử lí dữ liệu
'''
playlists_new = preprocess_data(playlists_df,0.5)
tracks_new = preprocess_data(tracks_df,0.5)
users_new = preprocess_data(users_df,0.5)
```

In [6]:
```python
# ghi dataframe ra file mới
playlists_new.to_csv('cleaned_data/playlists_new.csv', index=False)
tracks_new.to_csv('cleaned_data/tracks_new.csv', index=False)
users_new.to_csv('cleaned_data/users_new.csv', index=False)
```

In [7]:
```python
'''
File dữ liệu sau khi tiền xử lí cơ bản
'''
print(playlists_new.shape)
print(tracks_new.shape)
print(users_new.shape)
```

```
(608, 3)
(3485, 38)
(216, 32)
```

Câu 1: top 100 các tracks được download nhiều nhất vẽ biểu đồ histogram để thể hiện các tracks có số lượng download ở khoảng bao nhiêu chiếm đa số

- Trước khi trả lời câu hỏi này ta sẽ xác định một số ý như sau
  - Các số liệu đã được xác định
  - Ứng dụng câu hỏi là tìm ra các tracks được ưa chuộng nhất
  - Câu hỏi có thể trả lời với dữ liệu có sẵn

In [8]:
```python
My_tracks = tracks_new
```

Ta cần tiền xử lý một vài dữ liệu bất hợp lý

- Bỏ đi những dòng có dữ liệu bất hợp lý đó là những dòng có downloadable là False nhưng dòng download_count vẫn có giá trị khác 0.0
- Loại bỏ đi những dòng có giá trị là 0.0 hay NaN trong cột download_count
- Loại bỏ đi những dòng trùng nhau và giữ lại dòng xuất hiện đầu tiên

In [9]:
```python
My_tracks = My_tracks[My_tracks['downloadable'] == True]
My_tracks = My_tracks[My_tracks['download_count'].isna() == False]
My_tracks = My_tracks[My_tracks['download_count'] != 0.0]
```

Tiếp theo ta sẽ sắp xếp lại theo thứ tự giảm dần rồi chọn 100 dòng đầu tiên

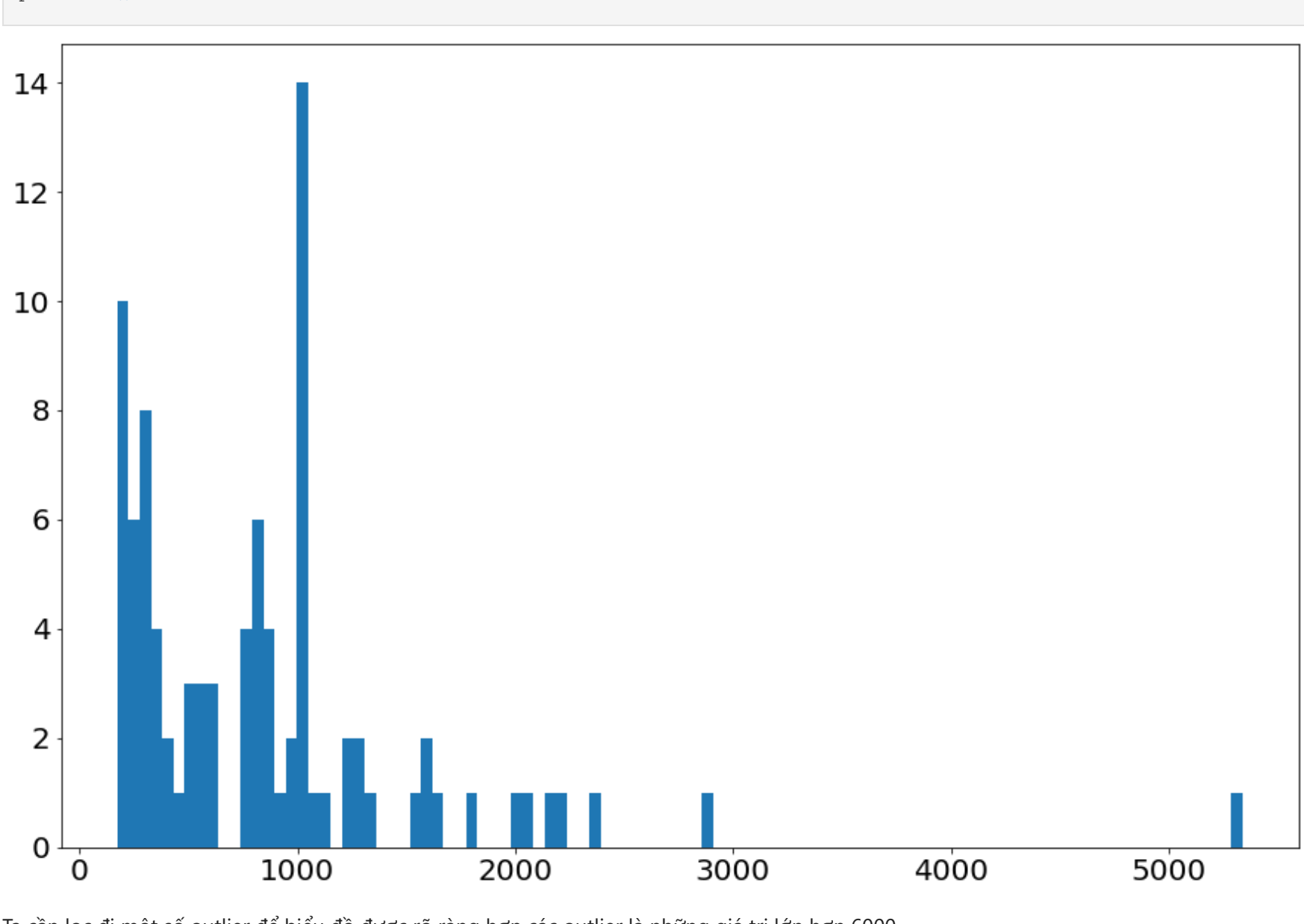
In [10]:
```python
My_df = My_tracks[['download_count', 'title']]
My_df = My_df.drop_duplicates(subset=['download_count', 'title'], keep='first')
My_df = My_df.sort_values(by='download_count', ascending=False)
My_df = My_df.head(100)
My_df
```

Out[10]:

	download_count	title
715	94327.0	Better Days
2585	63685.0	ODESZA - My Friends Never Die
860	45030.0	The Process
844	28523.0	Good Morning
771	19074.0	Summertime Love
...	...	...
815	212.0	Pieza 7: Jueces
720	198.0	The Way - Instrumental Music Oud / موسيقى عود
855	189.0	Pieza 40: Mateo
813	176.0	Pieza 23: Isaías
810	173.0	Pieza 17: Nehemías

100 rows × 2 columns

In [11]:
```python
'''
Ta sẽ loại bỏ đi 1 vài giá trị outliers, các giá trị này quá lệch so với các giá trị bình thường dẫn đến việc v
'''
My_df = My_df[My_df['download_count'] <= 6000]
plt.hist(My_df['download_count'], bins=100)
plt.show()
```



Ta cần lọc đi một số outlier để biểu đồ được rõ ràng hơn các outlier là những giá trị lớn hơn 6000

- Nhận xét:
  - Histogram thuộc dạng phân bố lệch trái
  - Số lượng download các tracks dao động trong khoảng 0 đến 1000 chiếm đa số

Câu 2: top 100 users có followers cao nhất và vẽ biểu đồ histogram để thể hiện các users có số lượng followers ở khoảng bao nhiêu chiếm đa số

- Trước khi trả lời câu hỏi này ta sẽ xác định một số ý như sau
  - Các số liệu đã được xác định
  - Ứng dụng câu hỏi là tìm ra users nổi tiếng và có nhiều sản phẩm âm nhạc được ưa thích
  - Câu hỏi có thể trả lời với dữ liệu có sẵn

In [12]:
```python
My_users = users_new
```

Ta cần tiền xử lý một số dữ liệu bất hợp lý

- Loại bỏ các dòng trùng nhau và giữ lại dòng xuất hiện đầu tiên

In [13]:
```python
My_users = My_users[['followers_count', 'username']]
My_users = My_users.drop_duplicates(subset=['followers_count', 'username'], keep='first')
```

Tiếp theo ta sẽ sắp xếp lại theo thứ tự giảm dần rồi chọn 100 dòng đầu tiên

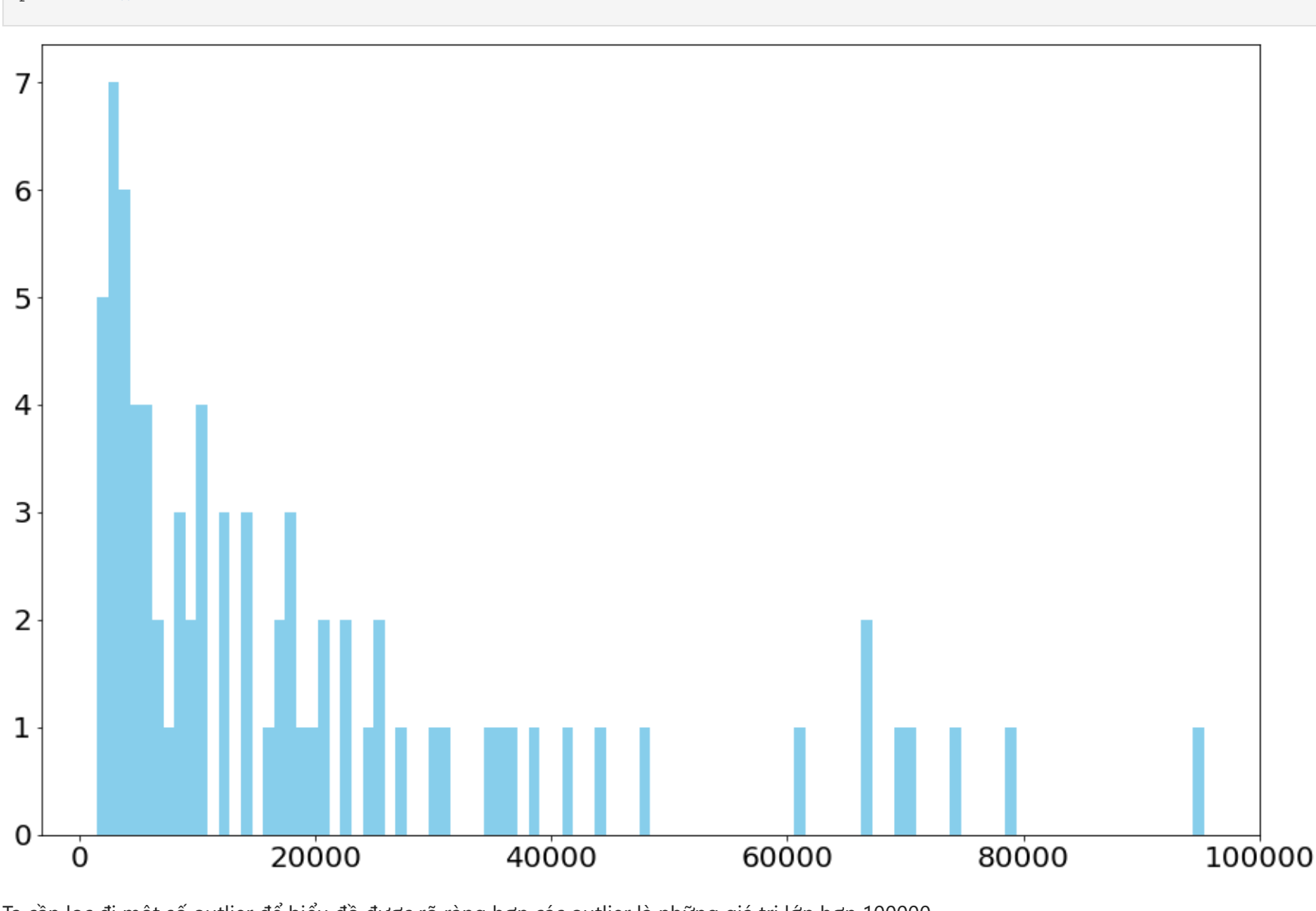
In [14]:
```python
My_users = My_users.sort_values(by='followers_count', ascending=False)
My_users = My_users.head(100)
My_users
```

Out[14]:

	followers_count	username
113	6498535.0	Skrillex
110	3649072.0	Seven Lions
205	3467276.0	Damian Marley
189	2785568.0	dubmatix
190	2768063.0	VP RECORDS
...	...	...
215	2349.0	Black Roots
120	2114.0	Tracey Chattaway
48	2011.0	Copeland Network
132	1814.0	Francesco Berta
183	1480.0	BOST & BIM

100 rows × 2 columns

In [15]:
```python
My_users = My_users[My_users['followers_count'] <= 100000]
plt.hist(My_users['followers_count'], bins=100, color = "skyblue")
plt.show()
```



Ta cần lọc đi một số outlier để biểu đồ được rõ ràng hơn các outlier là những giá trị lớn hơn 100000

- Nhận xét:
  - Histogram thuộc dạng phân bố lệch trái
  - Số lượng followers các users dao động trong khoảng 0 đến 20000 chiếm đa số

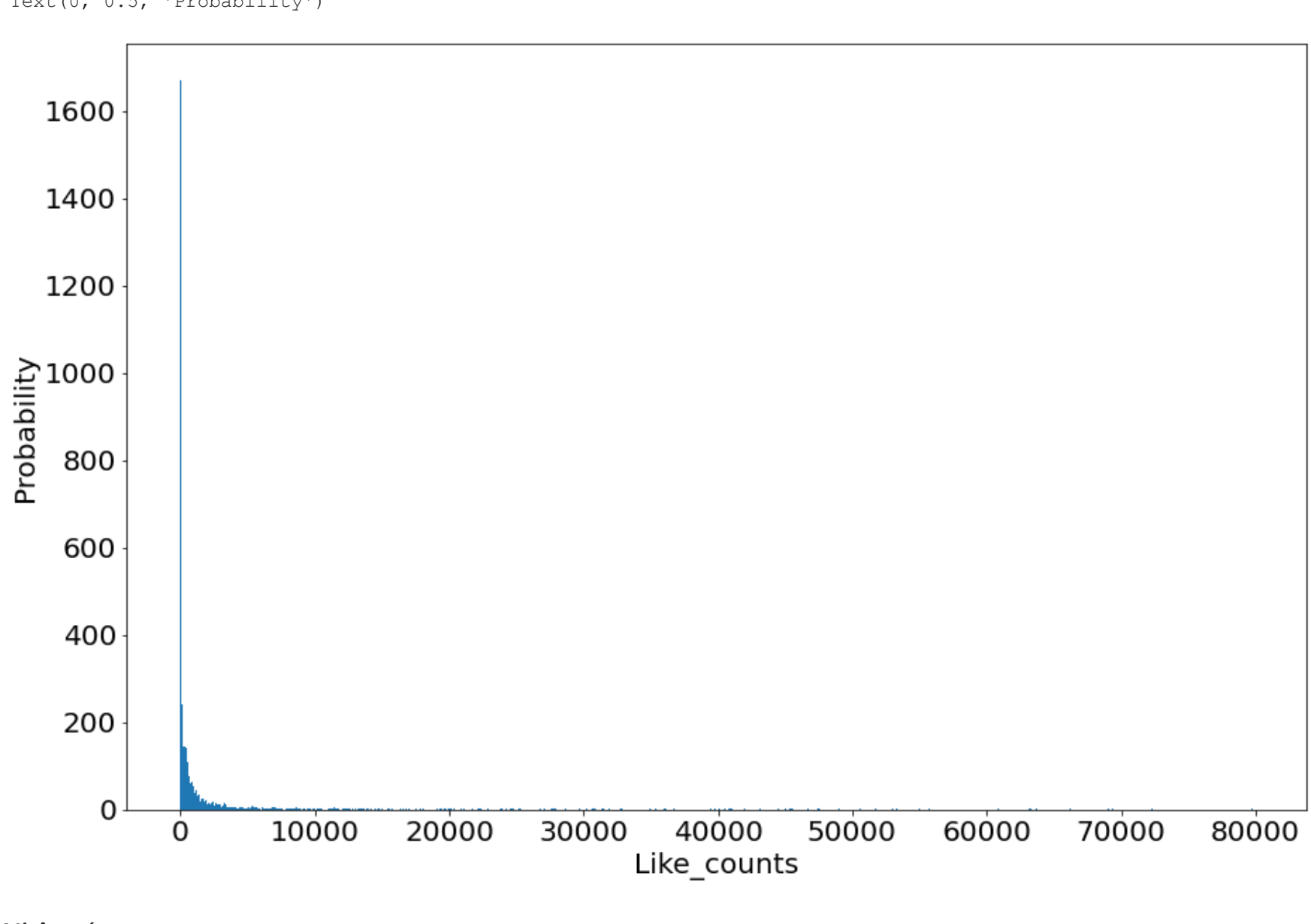
Câu 3: Vẽ biểu đồ histogram để xem lượt like_counts của mỗi track sẽ phân bố như thế nào ?

In [16]:
```python
his_df = tracks_new['likes_count']
his_df = tracks_new[tracks_new['likes_count'].notnull()]
his_df = his_df[his_df['likes_count'] <= 80000] ### Ta sẽ loại bỏ đi các outliers để có thể dễ trực quan hơn.
x = his_df['likes_count'].to_list()
x = np.asarray(x)
```

In [17]:
```python
# Calculate bins by Freedman-Diaconis method and plot the chart
q25, q75 = np.percentile(x, [25, 75])
bin_width = 2 * (q75 - q25) * len(x) ** (-1/3)
bins = round((x.max() - x.min()) / bin_width)
print("Freedman-Diaconis number of bins:", bins)

# plot distribution
plt.hist(x, bins=bins)
plt.xlabel('Like_counts')
plt.ylabel('Probability')
```

```
Freedman-Diaconis number of bins: 700
```

Out[17]:
```
Text(0, 0.5, 'Probability')
```



Nhận xét

- Đây là biểu đồ thể hiện sự phân bố của số lượt like của track
- Ta thấy đây không phải phân bố chuẩn mà nó bị lệch về phía bên trái rất nhiều (nhiều nhất nằm ở 0 lượt like)
- Điều này chứng tỏ có rất nhiều track được đăng tải lên nhưng không tiếp cận đến được người dùng nhiều, hoặc track đó không hay nên rất ít lượt like. Dựa vào đó ta sẽ cần cải thiện khả năng gợi ý track đến người dùng hơn để có thể tiếp cận bài hát đó. Hoặc ta sẽ phải phát triển bộ nhận dạng những track rác hoặc không hay để loại bỏ giúp Soundcloud phát triển hơn

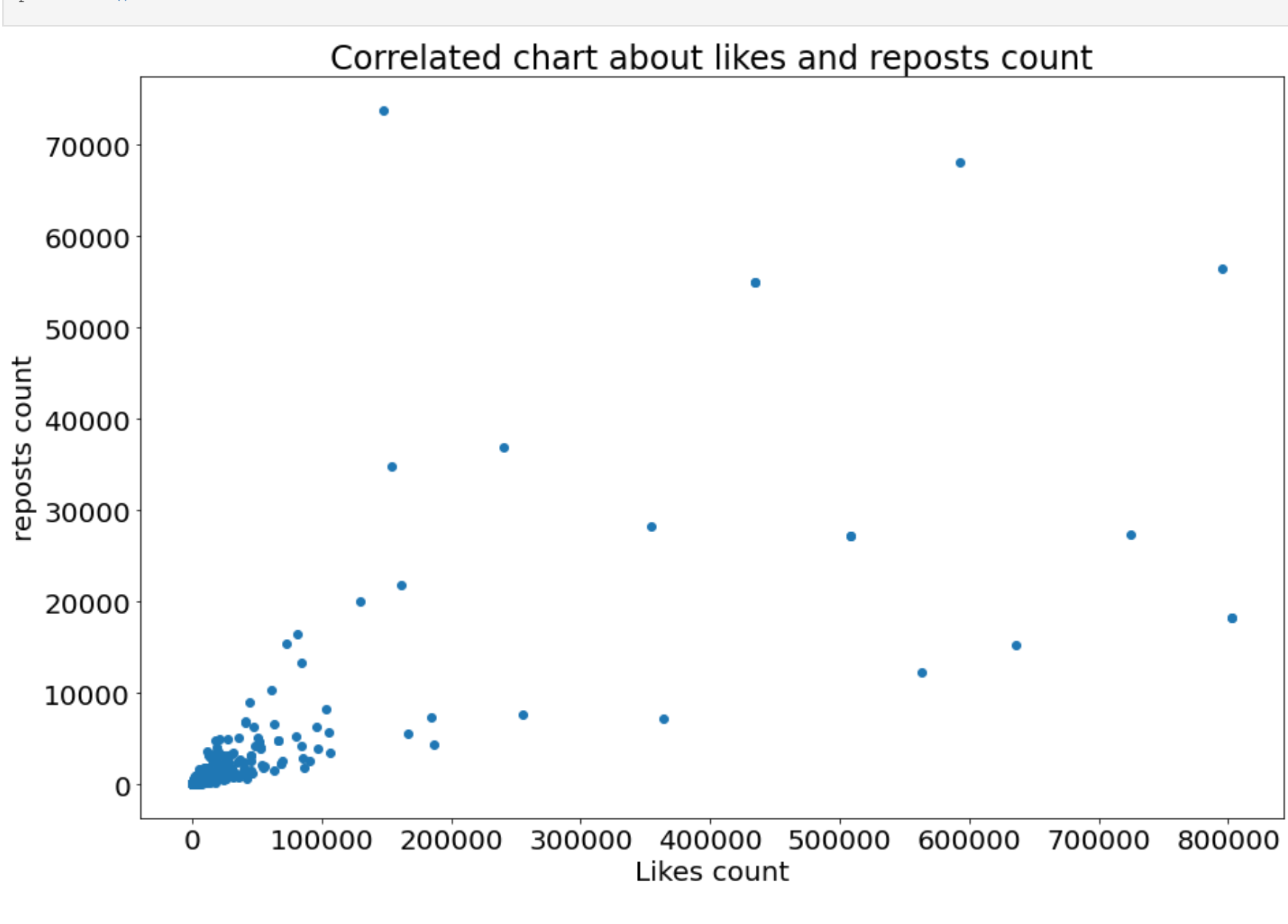
Câu hỏi 4: Sự tương quan giữa số lượt like và số lượt repost

Xét dataset **tracks** "**(3485, 38)**" chứa thông tin của từng track với số lượng sample là 3485. Ta sẽ xem xét 2 thuộc tính là:

- **likes_count**: số lượng người thích của track đó
- **reposts_count**: số lượng người đăng lại track đó Ta sẽ xem sự tương quan giữa số lượt thích **likes_count** với số lượng đăng lại **reposts_count** bằng Scatter plot để xem nếu một track có số lượt thích cao thì khả năng sẽ được repost lại cũng sẽ cao

In [18]:
```python
cor_data = tracks_new[['likes_count', 'reposts_count']]
```

In [19]:
```python
plt.scatter(cor_data['likes_count'], cor_data['reposts_count'], marker='o');

plt.xlabel('Likes count')
plt.ylabel('reposts count')
plt.title("Correlated chart about likes and reposts count")
plt.show()
```



Nhận xét:

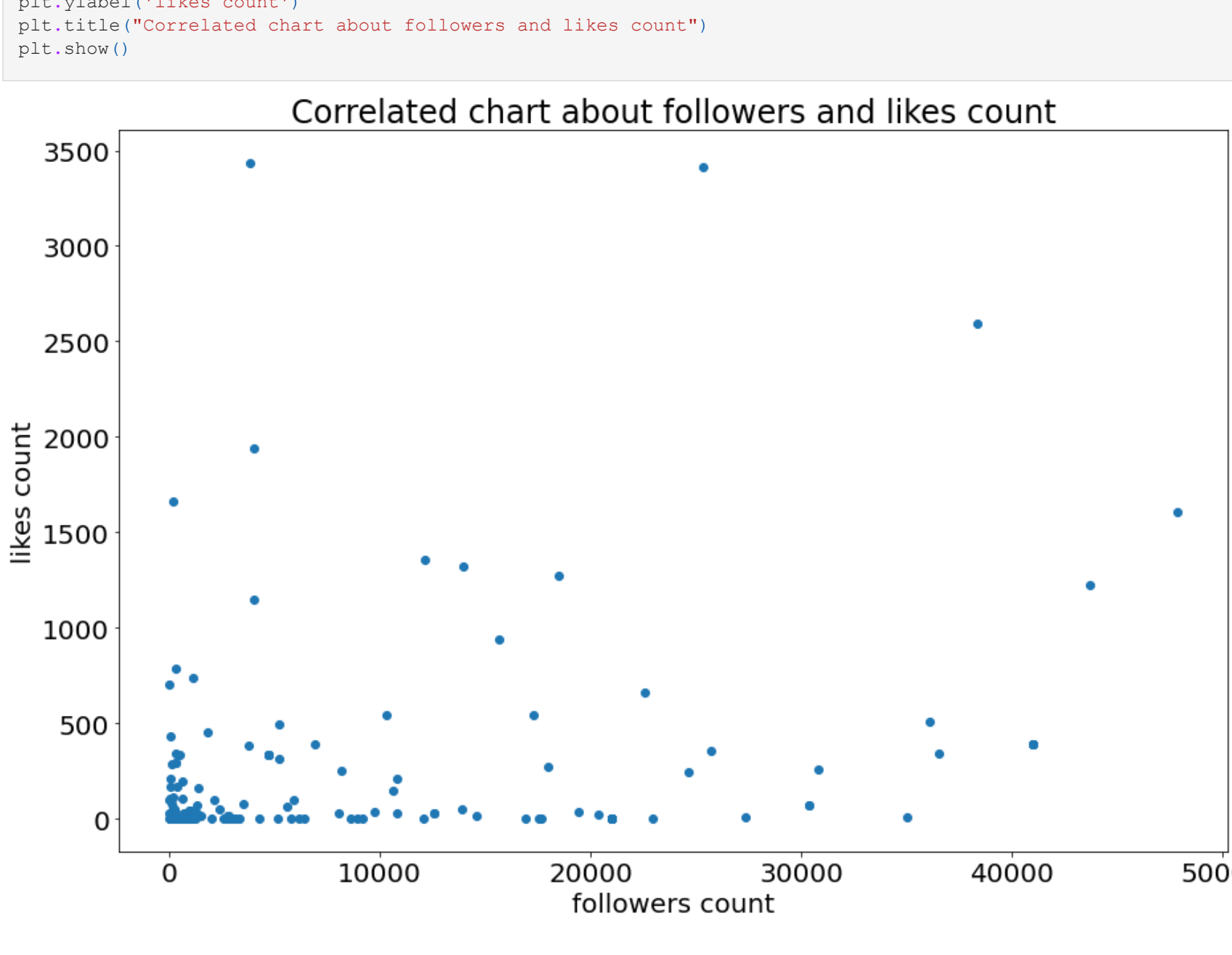
- Với phương pháp này, ta có thể nhận thấy các điểm outlier và ta có thể loại trừ chúng
- ta sẽ xem xét kĩ trong khoảng từ 0-1000, bởi vì đây là nơi các điểm tập trung lớn nhất. Ta thấy các điểm có xu hướng theo một đường thẳng xéo đi lên, vì vậy 2 biến **like_count** và **repost_count** có sự tương quan với nhau theo hướng dương (positive correlation), nghĩa là nếu lượt like_count tăng, thì track đó sẽ được yêu thích nhiều hơn dẫn đến lượt repost sẽ cao hơn

Câu 5: Đánh giá mối tương quan giữa followers và likes_count

- Trước khi trả lời câu hỏi này ta sẽ xác định một số ý như sau
  - Các số liệu đã được xác định.
  - Ứng dụng câu hỏi là xem xét mức độ ảnh hưởng của **số lượng theo dõi** so với **tổng số lượt thích**.
  - Câu hỏi có thể trả lời với dữ liệu có sẵn.
  - Dự đoán tổng số lượt thích sẽ tỉ lệ thuận với tổng số người theo dõi của *user* đó

In [20]:
```python
data_fl_likes = users_new[['likes_count', 'followers_count']]
# loại bỏ các ngoại lệ
data_fl_likes = data_fl_likes[data_fl_likes['followers_count'] < 50000]
```

In [21]:
```python
plt.scatter(data_fl_likes['followers_count'], data_fl_likes['likes_count'], marker='o');

plt.xlabel('followers count')
plt.ylabel('likes count')
plt.title("Correlated chart about followers and likes count")
plt.show()
```



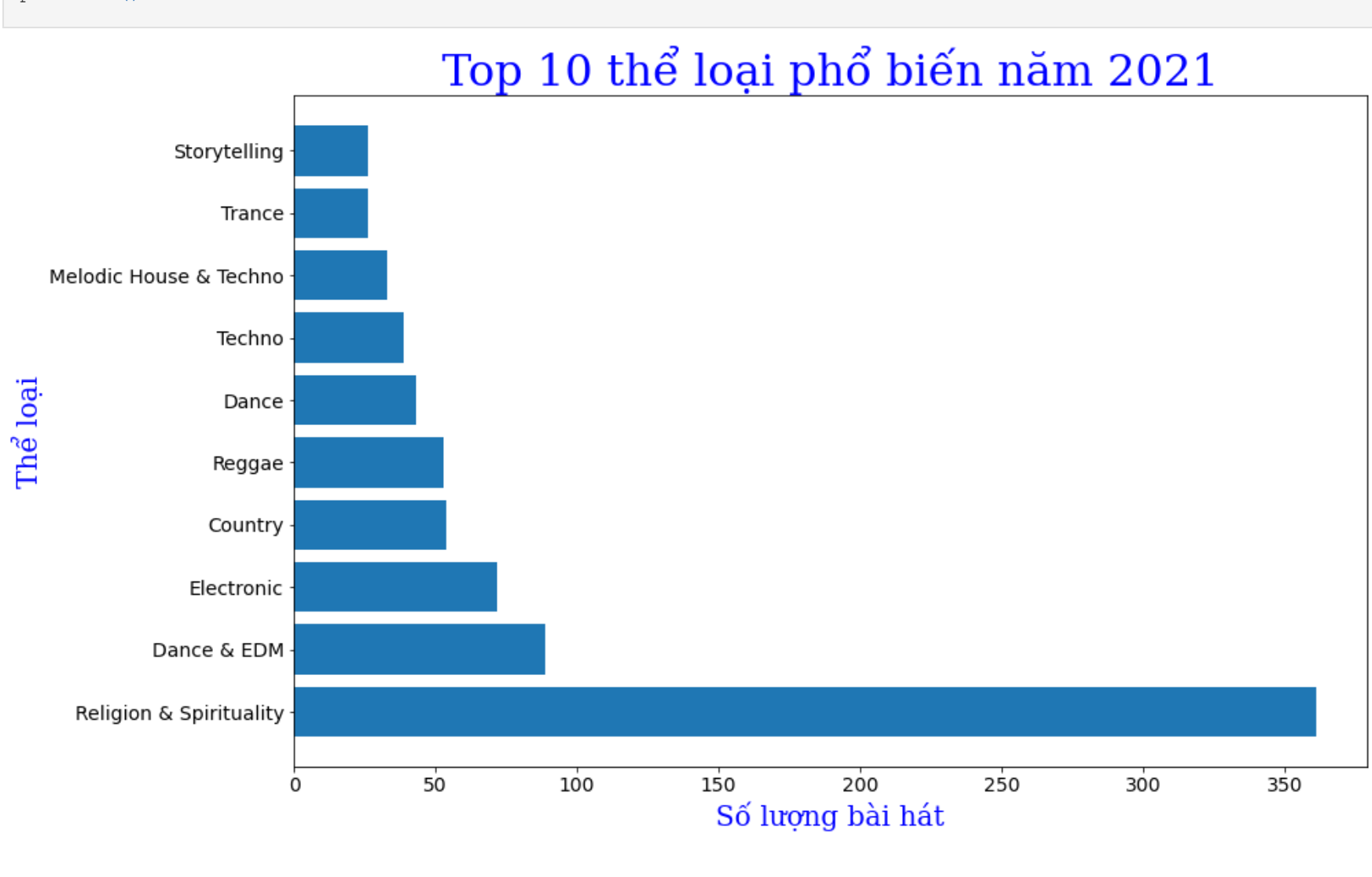
Nhận xét qua biểu đồ:

- Lượng theo dõi không ảnh hưởng quá nhiều đến tổng số lượt thích của *user* đó.
- Đa phần lượt thích chỉ trong khoảng 0 - 500, và nó đúng với phần lớn dữ liệu.
- Số lượng lượt thích gần bằng 0 rất nhiều cho dù số lượng theo dõi của user đó khá lớn (khoảng 10000 - 35000), điều đó có thể giải thích được người dùng chỉ like khi bài hát đó thực sự gây ấn tượng với họ.
- Đối với những user có lượng theo dõi ít (ít hơn 10000), tuy nhiên số lượt thích lại rất cao. Đây là những user mới nhưng đã gây được ấn tượng mạnh đối với người dùng với những bản nhạc đi vào lòng người. Đây là những user cực kì tiềm năng và sẽ thành công trong tương lại nếu tiếp tục cho ra những bản nhạc chất lượng.
- Đối với những user có lượng theo dõi lớn nhưng chưa có được số lượt thích cao, điều này có thể giải thích là những người này đã thành lập tài khoản đã lâu và ra nhiều bài chưa gây ấn tượng mạnh nhưng người dùng vẫn mong chờ họ có thể cho ra các ca khúc tốt hơn nên đã theo dõi, hoặc có thể họ dùng chiêu trò mua lượng theo dõi để tăng độ uy tín cho người dùng.

Câu 6: Top 10 thể loại nhạc phổ biến năm 2021

In [22]:
```python
df = pd.read_csv('cleaned_data/tracks_new.csv')
# lấy 2 thuộc tính là 'genre và created_at' và bỏ qua dòng có giá trị null.
dt_df = df[['genre', 'created_at']].dropna()

# đổi kiểu dữ liệu thuộc tính 'created_at' thành kiểu datetime
dt_df['created_at'] = pd.to_datetime(dt_df['created_at'])

# lấy top 10 thể loại phổ biến nhất năm 2021.
dt_df['year'] = dt_df['created_at'].apply(lambda x: x.year)
top_10_popular = dt_df.query('year == 2021')['genre'].value_counts()[:10]
```

In [23]:
```python
# visualization
fig, ax = plt.subplots(figsize=(14, 9))

ax.barh(top_10_popular[:10].index, top_10_popular[:10])

ax.set_title('Top 10 thể loại phổ biến năm 2021',fontdict=font, fontsize=32)
ax.set_xlabel('Số lượng bài hát', fontdict=font, fontsize=20)
ax.set_ylabel('Thể loại', fontdict=font, fontsize=20)
ax.tick_params(labelsize=14)
# fig.autofmt_xdate()

plt.show()
```



- số liệu trước khi bắt đầu cột thuộc tính "genre": tên thể loại và "created_at": ngày tạo.
- ngữ cảnh biết được xu hướng âm nhạc trong năm 2021, cho thể cho ra những sản phẩm nằm trong top 10 thể loại phổ biến, có thể giúp cho bài hát được nhiều người biết tới.
- mức độ đánh giá cho câu trả lời không cao (dữ liệu có thêm thuộc tính lược view thì kết quả sẽ tốt hơn).

- nhìn vào biểu đồ ta có thể tháy thể loại "Religion & Spirituality" là thể loại có nhiều bài hát nhất năm 2021 ta có thể đánh giá nó là thể loại đang được hướng tới từ những người nghệ sĩ.

Câu 7: Top 10 thể loại nhạc phổ biến dựa trên likes_count và playback_count

Trong file "tracks_new.csv":

- Tìm tổng số lượt likes_count, playback_count của các 'genre' và lưu vào dataFrame
- Sắp xếp dataFrame này theo thứ tự tăng dần likes_count
- Vẽ 2 biểu đồ thể hiện
- Mục đích: tìm được thể loại có tổng số lượt likes_count cao nhất và playback_count cao nhất

In [24]:
```python
tracks_df = pd.read_csv('cleaned_data/tracks_new.csv', index_col='Unnamed: 0')
users_df = pd.read_csv('cleaned_data/users_new.csv', index_col='Unnamed: 0')
```

In [25]:
```python
genres = tracks_df['genre'].unique()
#create genre df:
genre_df = pd.DataFrame(columns = ['sum_likes_count', 'sum_playback_count'], index = genres)

for g in genres:
    genre_df.loc[g,'sum_likes_count']=tracks_df[tracks_df['genre']==g]['likes_count'].sum()
    genre_df.loc[g,'sum_playback_count']=tracks_df[tracks_df['genre']==g]['playback_count'].sum()

genre_df.sort_values(by='sum_likes_count', ascending=False, inplace=True)
```

Xem kết quả của 10 dòng đầu

In [26]:
```python
genre_df.head(10)
```

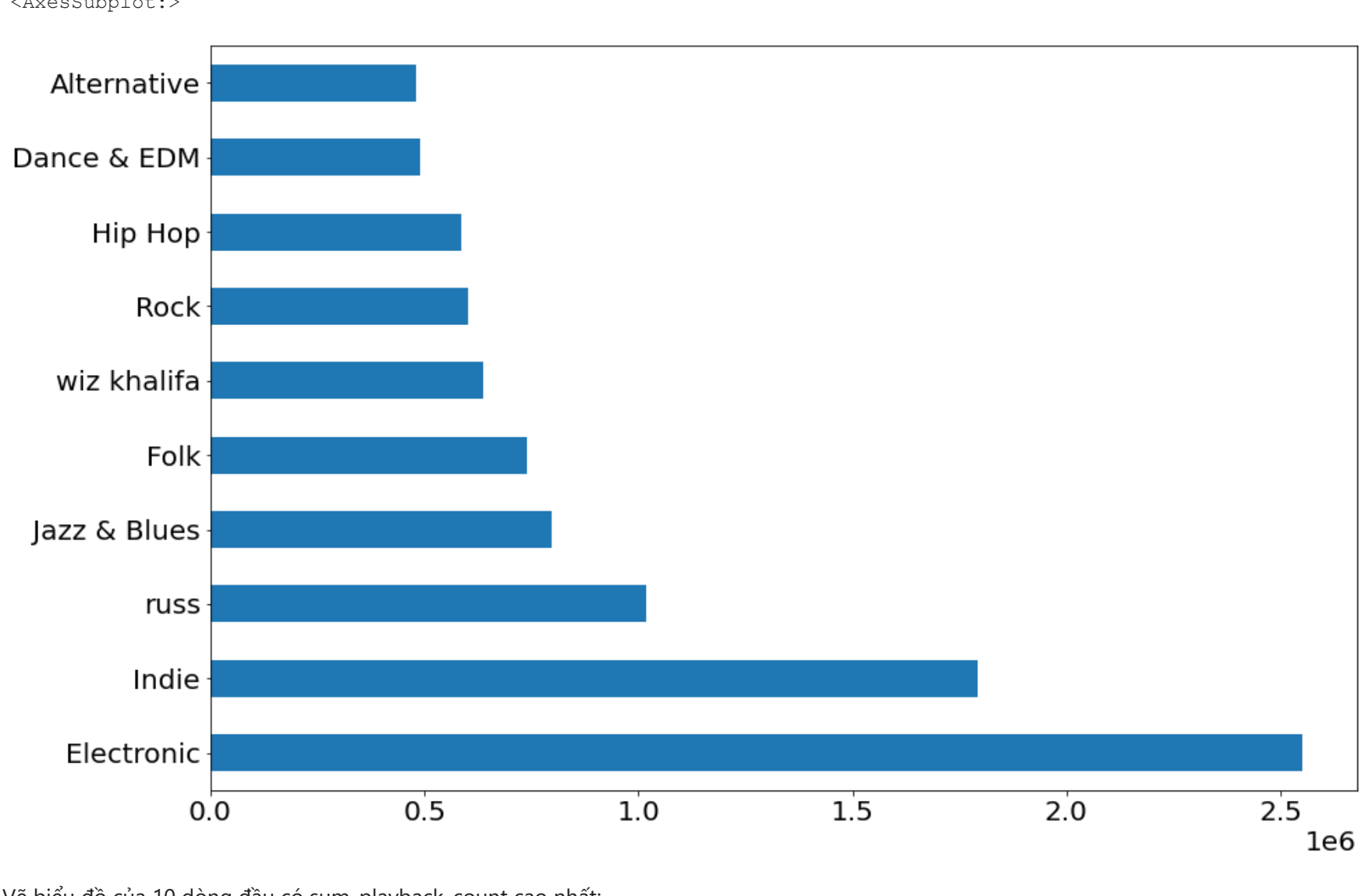
Out[26]:

	sum_likes_count	sum_playback_count
Electronic	2550977.0	165270409.0
Indie	1791551.0	140774857.0
russ	1016850.0	70141840.0
Jazz & Blues	795824.0	47363084.0
Folk	739395.0	35819052.0
wiz khalifa	635977.0	41964962.0
Rock	600719.0	52365698.0
Hip Hop	585424.0	35667370.0
Dance & EDM	490538.0	24802890.0
Alternative	481036.0	31872256.0

Vẽ biểu đồ của 10 dòng đầu có sum_likes_count cao nhất:

In [27]:
```python
genre_df['sum_likes_count'].head(10).plot.barh()
```

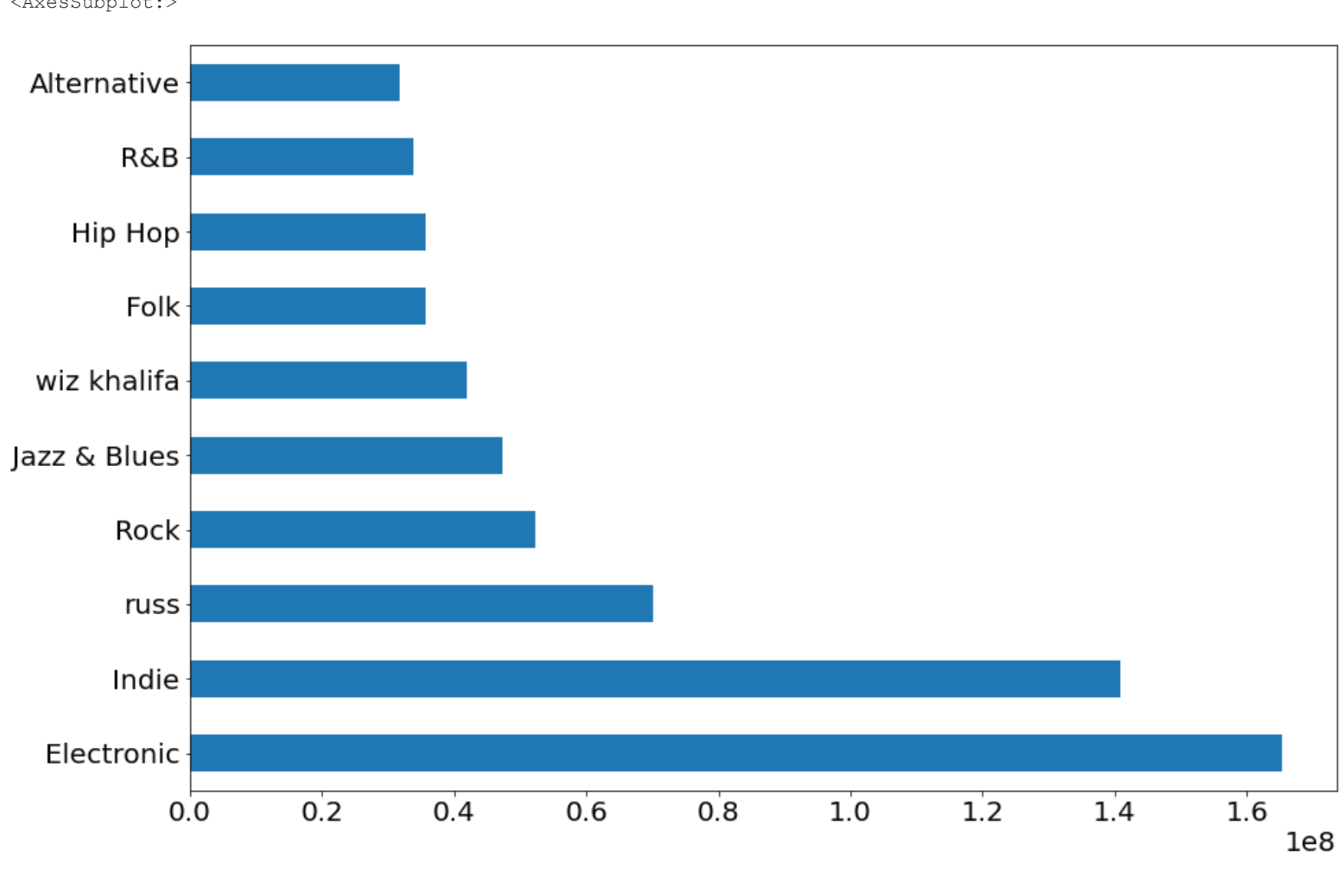
Out[27]:
```
<AxesSubplot:>
```



Vẽ biểu đồ của 10 dòng đầu có sum_playback_count cao nhất:

In [28]:
```python
genre_df.sort_values(by='sum_playback_count',ascending=False)['sum_playback_count'].head(10).plot.barh()
```

Out[28]:
```
<AxesSubplot:>
```



Câu 8: Users đến từ đâu là nhiều nhất

Trong "users_new.csv":

- Cột "country_code" thể hiện user này đến từ nước nào (VD: US: Hoa Kì, VN: Việt Nam)
- Tạo ra một DataFrame để lưu lại các thông tin sau: Index = "country_code", 1 cột number_of_users lưu số lượng user.
- Không xử lí ô trống
- Tìm hiểu users trong file này đến từ nước nào nhiều nhất, sắp xếp theo thứ tự giảm dần

In [29]:
```python
#country_code viết tắt là cc
cc = users_df[users_df['country_code'].notna()]['country_code'].unique()
#create df
cc_df = pd.DataFrame(columns=['number_of_users'],index=cc)

for c in cc:
    cc_df.loc[c,'number_of_users'] = (users_df['country_code']==c).sum()

cc_df.sort_values(by='number_of_users',ascending=False,inplace=True)
```

Xem kết quả:

In [30]:
```python
cc_df
```

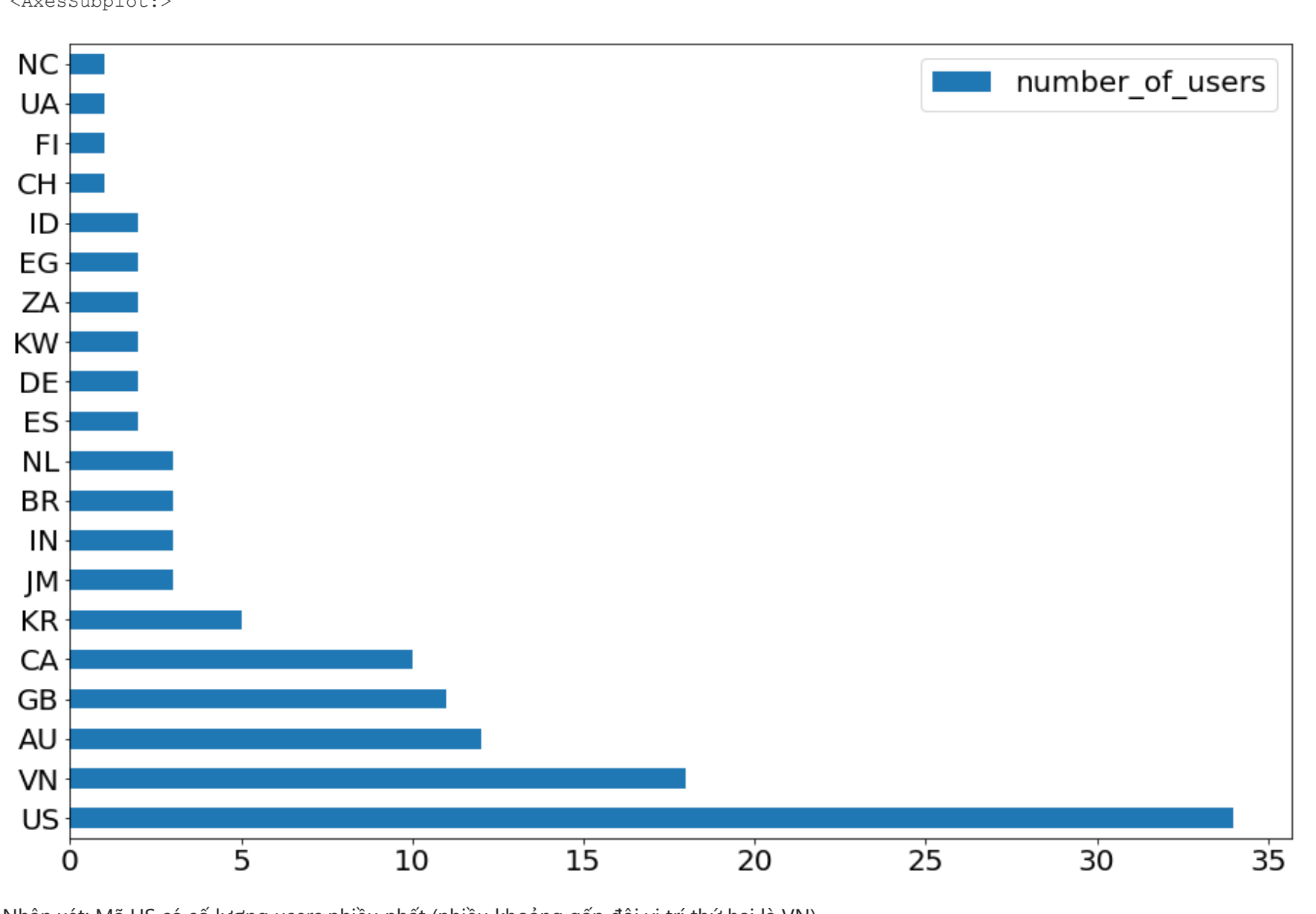
Out[30]:

	number_of_users
US	34
VN	18
AU	12
GB	11
CA	10
KR	5
JM	3
IN	3
BR	3
NL	3
ES	2
DE	2
KW	2
ZA	2
EG	2
ID	2
CH	1
FI	1
UA	1
NC	1
GR	1
JP	1
HU	1
PL	1
FR	1
DK	1
PH	1
SE	1
PT	1
CZ	1
DO	1

Vẽ biểu đồ của 20 nước có users nhiều nhất

In [31]:
```python
cc_df.head(20).plot.barh()
```

Out[31]: <AxesSubplot:>



Nhận xét: Mã US có số lượng users nhiều nhất (nhiều khoảng gấp đôi vị trí thứ hai là VN)