02_Data_Exploration

April 13, 2022

1 Load required libraries

[3]:
```
import pandas as pd
import boto3
import sagemaker

sess = sagemaker.Session()
bucket = sess.default_bucket()
role = sagemaker.get_execution_role()
region = boto3.Session().region_name
```

2 Download the datasets from private S3 bucket

[4]:
```
!aws s3 cp 's3://ads508-team4-master/result_demo.csv' ./data/
```

```
download: s3://ads508-team4-master/result_demo.csv to data/result_demo.csv
```

[5]:
```
!aws s3 cp 's3://ads508-team4-master/result_psych.csv' ./data/
```

```
download: s3://ads508-team4-master/result_psych.csv to data/result_psych.csv
```

[6]:
```
import csv

df_demo = pd.read_csv(
    "./data/result_demo.csv",
    delimiter=",",
    quoting=csv.QUOTE_NONE,
)
df_demo = df_demo.iloc[:,1:]
df_demo.head(100)
```

[6]:
```
        user_id platform_x      level_1 level_2  level_3  confidence_score  \
0  1.717987e+10    android  Demographics  Income   Medium          1.000000
1  3.435974e+10    android  Demographics  Income   Medium          1.000000
2  1.717987e+10    android  Demographics  Income      Low          1.000000
3  1.717987e+10    android  Demographics     Age  25 - 34          1.000000
4  1.717987e+10    android  Demographics  Gender     Male          0.993641
```

```
..           ...        ...           ...     ...      ...               ...
95  2.576980e+10    android  Demographics  Income   Medium          1.000000
96  8.589935e+10    android  Demographics  Income      Low          1.000000
97  7.730941e+10        web  Demographics  Gender     Male          0.938380
98  7.730941e+10        web  Demographics     Age  18 - 24          1.000000
99  8.589935e+10        iOS  Demographics  Income   Medium          1.000000

   country_code platform_y  asset_id  minutes_viewed showtype  \
0            PH    android     14707              55   Movies
1            PH    android     14707              92   Movies
2            PH    android     14707              76   Movies
3            PH    android     14707              76   Movies
4            PH    android     14707              76   Movies
..          ...        ...       ...             ...      ...
95           PH    android     14734              80   Movies
96           PH    android     14734             112   Movies
97           PH        web     14734              92   Movies
98           PH        web     14734              92   Movies
99           PH        iOS     14734               6   Movies

                   genre  running_minutes source_language  season_id  \
0   Action and Adventure              103         Tagalog        NaN
1   Action and Adventure              103         Tagalog        NaN
2   Action and Adventure              103         Tagalog        NaN
3   Action and Adventure              103         Tagalog        NaN
4   Action and Adventure              103         Tagalog        NaN
..                   ...              ...             ...        ...
95                 Drama              116         Tagalog        NaN
96                 Drama              116         Tagalog        NaN
97                 Drama              116         Tagalog        NaN
98                 Drama              116         Tagalog        NaN
99                 Drama              116         Tagalog        NaN

    series_id  studio_id
0         NaN      448.0
1         NaN      448.0
2         NaN      448.0
3         NaN      448.0
4         NaN      448.0
..        ...        ...
95        NaN      448.0
96        NaN      448.0
97        NaN      448.0
98        NaN      448.0
99        NaN      448.0

[100 rows x 17 columns]
```

[7]:

```python
df_psych = pd.read_csv(
    "./data/result_psych.csv",
    delimiter=",",
    quoting=csv.QUOTE_NONE,
)
df_psych = df_psych.iloc[:,1:]
df_psych.head(100)
```

[7]:

```
        user_id platform_x        level_1        level_2  \
0   8.589935e+10  web-embed  Psychographics  Movies Lovers
1   8.589935e+10  web-embed  Psychographics  Movies Lovers
2   2.576980e+10    android  Psychographics  Movies Lovers
3   2.576980e+10    android  Psychographics      TV Lovers
4   2.576980e+10    android  Psychographics      TV Lovers
..           ...        ...             ...            ...
95  7.700000e+01    android  Psychographics  Movies Lovers
96  7.700000e+01    android  Psychographics  Movies Lovers
97  7.700000e+01    android  Psychographics      TV Lovers
98  6.871948e+10    android  Psychographics      TV Lovers
99  6.871948e+10    android  Psychographics     Travellers

                  level_3  confidence_score country_code  asset_id  \
0       Horror Movies Fans              0.07           ID     10377
1   Indonesian Movies Fans              0.03           ID     10377
2      Romance Movies Fans              0.52           ID     10377
3             Kids TV Fans              0.61           ID     10377
4            Drama TV Fans              0.60           ID     10377
..                    ...               ...          ...       ...
95      Korean Movies Fans              0.54           ID     10377
96     English Movies Fans              0.46           ID     10377
97         English TV Fans              0.40           ID     10377
98         English TV Fans              0.29           ID     10377
99         Local Commuters              0.17           ID     10377

    minutes_viewed showtype   genre  running_minutes source_language  \
0                1   Movies  Horror               87      Indonesian
1                1   Movies  Horror               87      Indonesian
2                3   Movies  Horror               87      Indonesian
3                3   Movies  Horror               87      Indonesian
4                3   Movies  Horror               87      Indonesian
..             ...      ...     ...              ...             ...
95              12   Movies  Horror               87      Indonesian
96              12   Movies  Horror               87      Indonesian
97              12   Movies  Horror               87      Indonesian
98               3   Movies  Horror               87      Indonesian
99               3   Movies  Horror               87      Indonesian
```

```
    season_id  series_id  studio_id  minutes_under_2
0         NaN        NaN      350.0             True
1         NaN        NaN      350.0             True
2         NaN        NaN      350.0            False
3         NaN        NaN      350.0            False
4         NaN        NaN      350.0            False
..        ...        ...        ...              ...
95        NaN        NaN      350.0            False
96        NaN        NaN      350.0            False
97        NaN        NaN      350.0            False
98        NaN        NaN      350.0            False
99        NaN        NaN      350.0            False

[100 rows x 17 columns]
```

3 Beginning Data Exploration

3.1 Checking for duplicated users

[8]:
```
# importing necessary libraries

# run below if seaborn packages dont run as expected
# pip install -U seaborn

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[9]:
```
df_demo['user_id'].value_counts()
```

[9]:
```
8.589935e+09    321
1.717987e+10    216
8.589935e+10    211
1.717987e+10    186
2.576980e+10    186
               ...
6.012954e+10      1
2.576980e+10      1
8.589935e+10      1
4.294967e+10      1
3.435974e+10      1
Name: user_id, Length: 3516, dtype: int64
```

[10]:
```
df_psych['user_id'].value_counts()
```

```
[10]: 1.717987e+10    4123
6.012954e+10    2793
1.060000e+02    2600
8.589935e+10    2560
3.440000e+02    2414
                ...
9.448928e+10       1
9.448928e+10       1
9.448928e+10       1
8.589935e+09       1
9.448928e+10       1
Name: user_id, Length: 5468, dtype: int64
```

Looks like some users have more predictions than others in both demographics and psychographics. A much larger range in psychographics than demographics as well

3.2 Checking for missing values

```
[11]: print("Nulls in demographics: \n\n",df_demo.isnull().sum(),"\n","\nNulls in␣
↪psychographics: \n\n",df_psych.isnull().sum())
```

```
Nulls in demographics:

 user_id                0
platform_x            0
level_1               0
level_2               0
level_3               0
confidence_score      0
country_code          0
platform_y            0
asset_id              0
minutes_viewed        0
showtype              0
genre                 2
running_minutes       0
source_language      90
season_id         18472
series_id         18472
studio_id            40
dtype: int64

Nulls in psychographics:

 user_id                0
platform_x            0
level_1               0
level_2               0
```

```
level_3                 4850
confidence_score           0
country_code               0
asset_id                   0
minutes_viewed             0
showtype                   0
genre                      5
running_minutes            0
source_language          701
season_id             146311
series_id             146311
studio_id                300
minutes_under_2            0
dtype: int64
```

Relatively clean data. Features with a lot of nulls are psychographics.level_3 and season_id and series_id. The former must be a very specific trait that is hard to predict. Assets with missing season_id and series_id must be Movies.

To verify our guess, we'll get a subset of the dataframe.

[12]:
```
df1 = df_psych.loc[df_psych["showtype"]=="Movies"]
df2 = df1[["showtype","season_id","series_id"]]
df2.isna().sum()
```

[12]:
```
showtype          0
season_id    146311
series_id    146311
dtype: int64
```

From this we can confirm that shows with missing season_id and series_id are Movies.

[13]:
```
df_demo.dtypes
```

[13]:
```
user_id             float64
platform_x           object
level_1              object
level_2              object
level_3              object
confidence_score    float64
country_code         object
platform_y           object
asset_id              int64
minutes_viewed        int64
showtype             object
genre                object
running_minutes       int64
source_language      object
season_id           float64
```

```
series_id          float64
studio_id          float64
dtype: object
```

[14]:
```
df_psych.dtypes
```

[14]:
```
user_id             float64
platform_x           object
level_1              object
level_2              object
level_3              object
confidence_score    float64
country_code         object
asset_id              int64
minutes_viewed        int64
showtype             object
genre                object
running_minutes       int64
source_language      object
season_id           float64
series_id           float64
studio_id           float64
minutes_under_2        bool
dtype: object
```

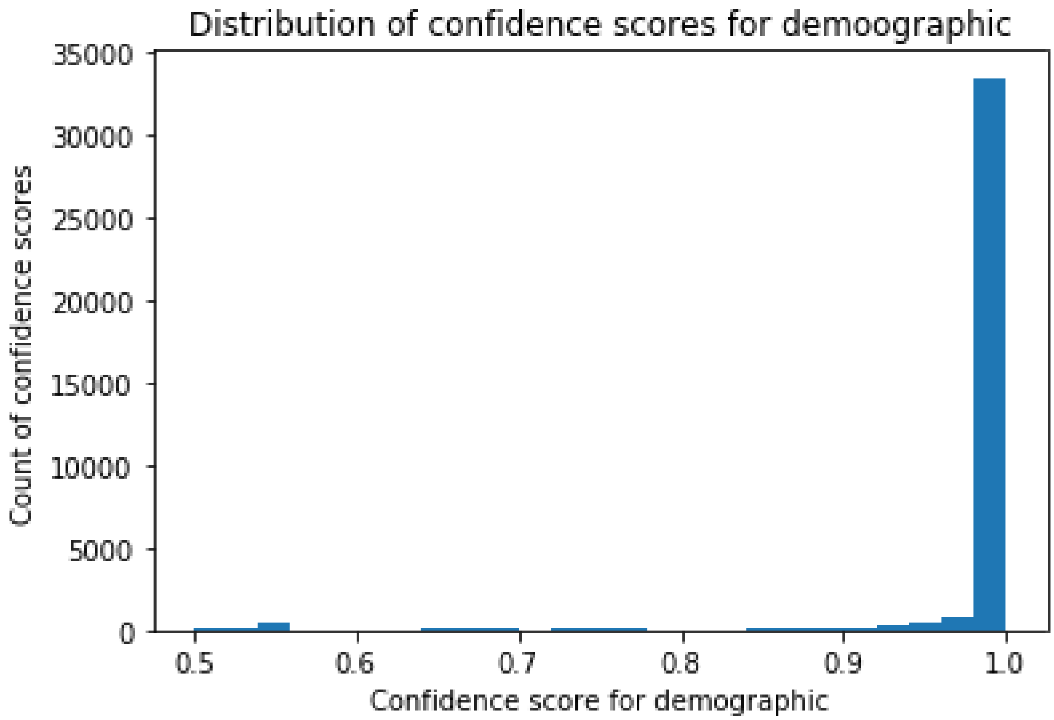
3.3 Statistical Descriptions

[15]:
```
df_demo[['confidence_score','minutes_viewed','running_minutes']].describe()
```

[15]:

	confidence_score	minutes_viewed	running_minutes
count	36849.00000	36849.000000	36849.000000
mean	0.98095	37.295503	76.157399
std	0.07751	57.666188	33.289861
min	0.50000	0.000000	6.000000
25%	1.00000	1.000000	52.000000
50%	1.00000	21.000000	72.000000
75%	1.00000	63.000000	102.000000
max	1.00000	5482.000000	211.000000

[16]:
```
plt.hist(df_demo['confidence_score'], bins = 25)

plt.xlabel("Confidence score for demographic")
plt.ylabel("Count of confidence scores")
plt.title("Distribution of confidence scores for demoographic")
plt.show()
```



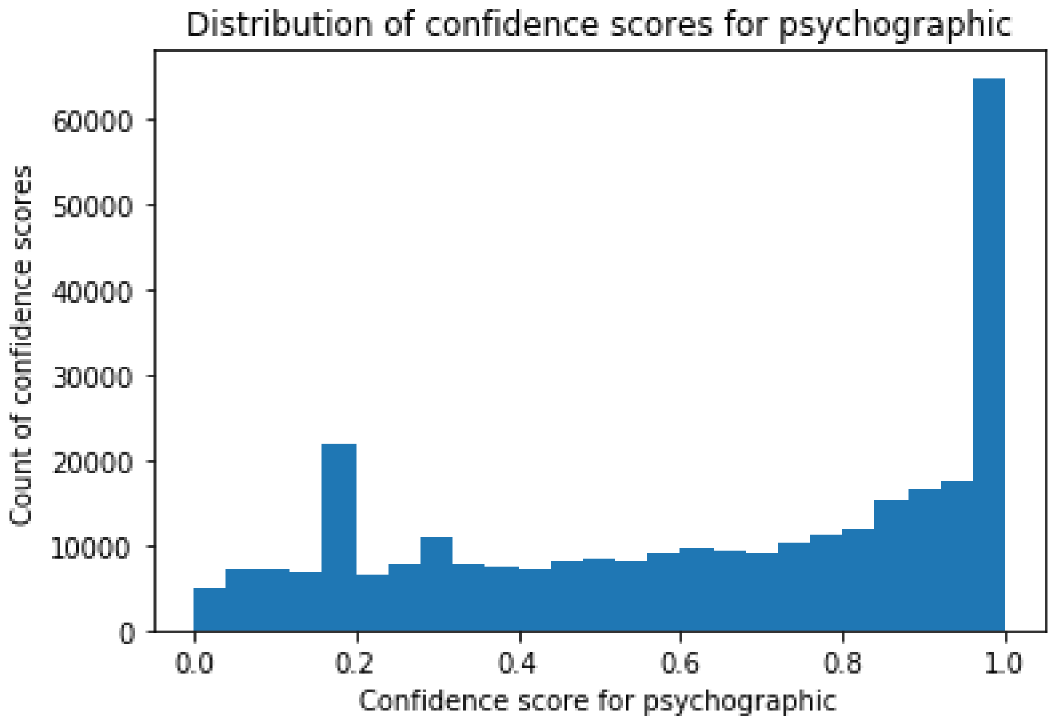
For demographics, there is a very high average for confidence scores, with a heavy left skew distribution as its median is greater than its mean, with its median, and even 25th percentile, at a perfect 1 confidence score. Since our goal is to improve the prediction accuracy for those records with confidence score below 0.5, we are going to focus on the psychographics only for our project.

[17]:
```
df_psych[['confidence_score','minutes_viewed','running_minutes']].describe()
```

[17]:

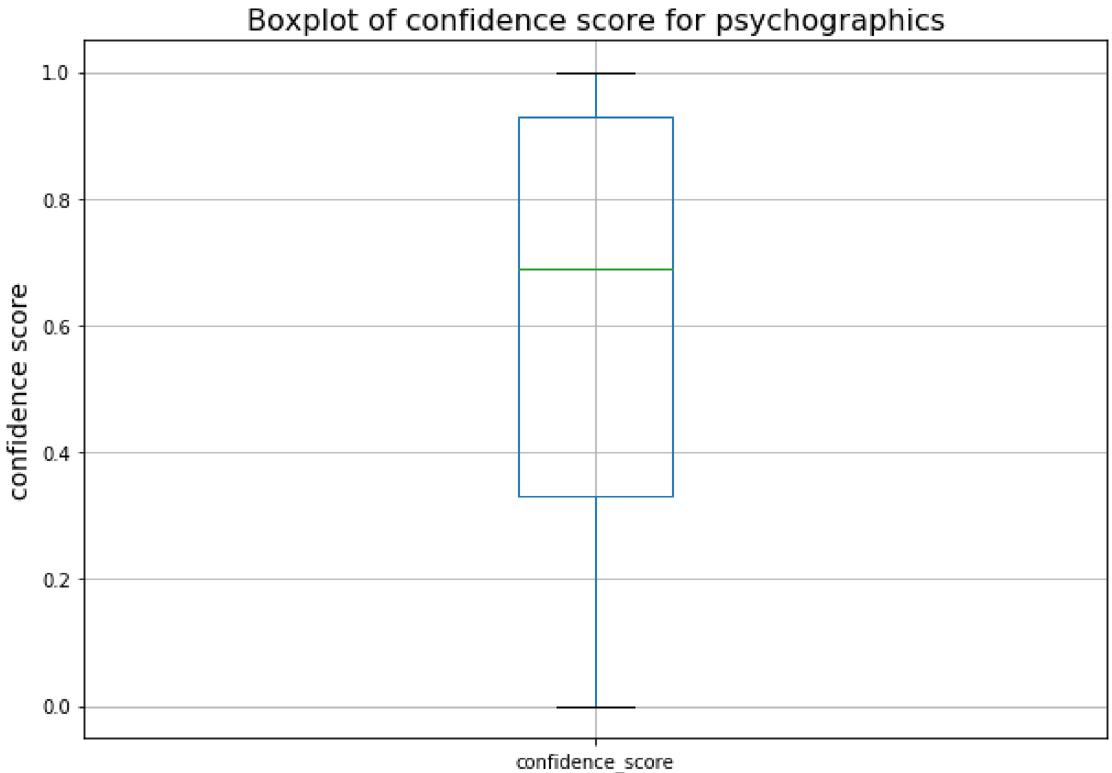
	confidence_score	minutes_viewed	running_minutes
count	303848.000000	303848.000000	303848.000000
mean	0.627092	36.800677	73.081837
std	0.317618	63.448361	34.009141
min	0.000000	0.000000	6.000000
25%	0.330000	2.000000	47.000000
50%	0.690000	19.000000	69.000000
75%	0.930000	62.000000	97.000000
max	1.000000	5482.000000	211.000000

[18]:
```
plt.hist(df_psych['confidence_score'], bins = 25)

plt.xlabel("Confidence score for psychographic")
plt.ylabel("Count of confidence scores")
plt.title("Distribution of confidence scores for psychographic")
plt.show()
```



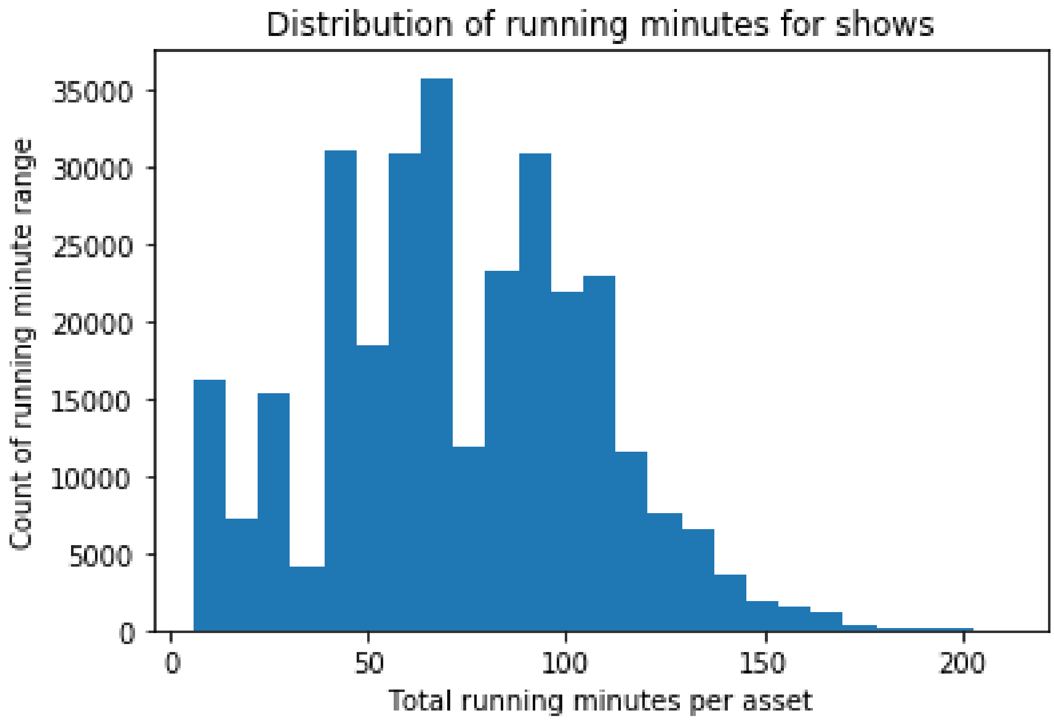
[19]:
```
fig = plt.figure(figsize =(10, 7))
# Creating axes instance
boxplot = df_psych.boxplot(column=['confidence_score'])
plt.title("Boxplot of confidence score for psychographics", fontsize = 16)
plt.ylabel("confidence score", fontsize= 14 )
```

[19]:
```
Text(0, 0.5, 'confidence score')
```



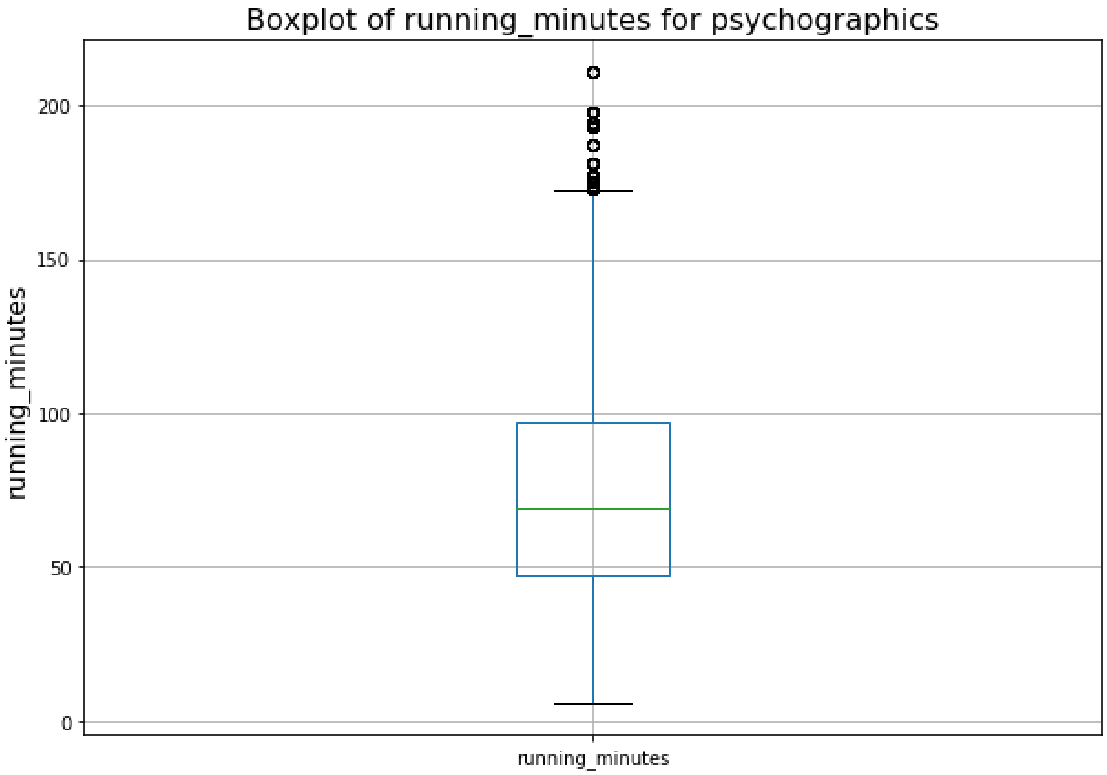
A semi-uniform distribution on the confidence score with a spike of scores around 1. We must exercise caution when working with this data, and we should set some type of cutoff to only retain the best data. Possible data balance is required.

[20]:
```
plt.hist(df_psych['running_minutes'], bins = 25)

plt.xlabel("Total running minutes per asset")
plt.ylabel("Count of running minute range")
plt.title("Distribution of running minutes for shows")
plt.show()
```



[21]:
```
fig = plt.figure(figsize =(10, 7))
# Creating axes instance
boxplot = df_psych.boxplot(column=['running_minutes'])
plt.title("Boxplot of running_minutes for psychographics", fontsize = 16)
plt.ylabel("running_minutes", fontsize= 14 )
```

[21]:
```
Text(0, 0.5, 'running_minutes')
```



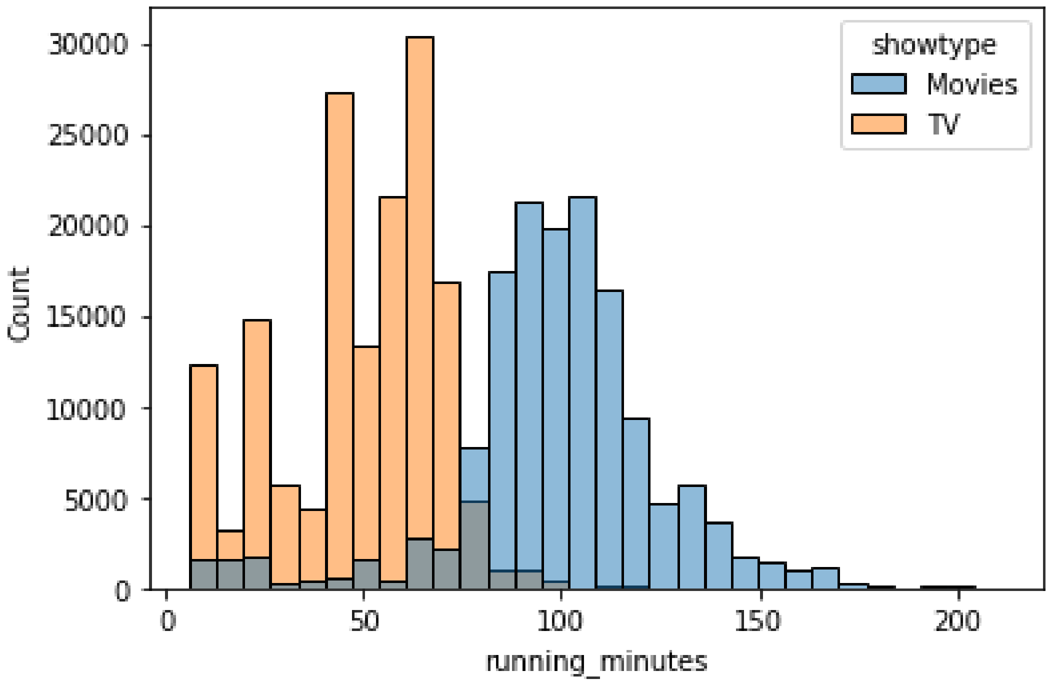
We see that most running minutes fall into 5 to 160 minutes.

[22]:
```
!pip install -U seaborn

import seaborn as sns

sns.histplot(data=df_psych,x='running_minutes', hue = 'showtype', bins = 30)
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
Requirement already satisfied: seaborn in /opt/conda/lib/python3.7/site-packages
(0.11.2)
Requirement already satisfied: matplotlib>=2.2 in /opt/conda/lib/python3.7/site-
packages (from seaborn) (3.1.3)
Requirement already satisfied: pandas>=0.23 in /opt/conda/lib/python3.7/site-
packages (from seaborn) (1.0.1)
Requirement already satisfied: numpy>=1.15 in /opt/conda/lib/python3.7/site-
```

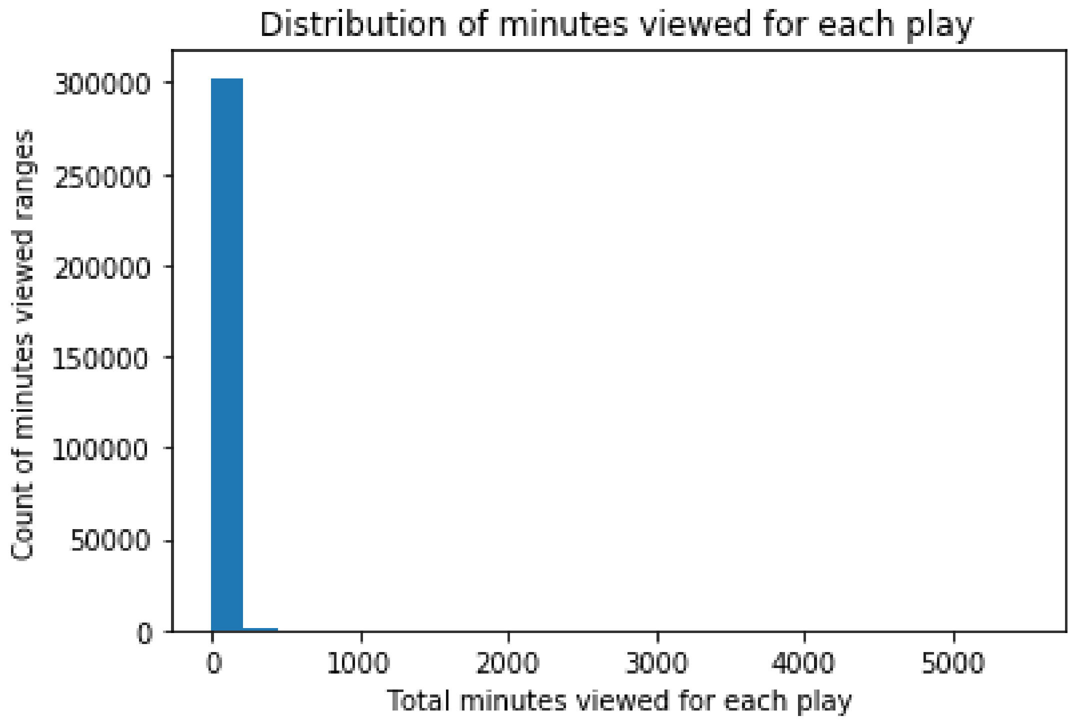
```
packages (from seaborn) (1.20.3)
Requirement already satisfied: scipy>=1.0 in /opt/conda/lib/python3.7/site-
packages (from seaborn) (1.4.1)
Requirement already satisfied: python-dateutil>=2.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (2.8.1)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (2.4.6)
Requirement already satisfied: cycler>=0.10 in /opt/conda/lib/python3.7/site-
packages (from matplotlib>=2.2->seaborn) (0.10.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/conda/lib/python3.7/site-packages (from matplotlib>=2.2->seaborn) (1.1.0)
Requirement already satisfied: pytz>=2017.2 in /opt/conda/lib/python3.7/site-
packages (from pandas>=0.23->seaborn) (2019.3)
Requirement already satisfied: six in /opt/conda/lib/python3.7/site-packages
(from cycler>=0.10->matplotlib>=2.2->seaborn) (1.14.0)
Requirement already satisfied: setuptools in /opt/conda/lib/python3.7/site-
packages (from kiwisolver>=1.0.1->matplotlib>=2.2->seaborn) (59.5.0)
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is
available.
You should consider upgrading via the '/opt/conda/bin/python -m pip install
--upgrade pip' command.
```

[22]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2fae847a50>
```



Predictably, movies tend to be much longer than tv shows. TV shows have two spikes - around 30 and 40 minutes, while movies have a slight right skew with its most around 80 minutes or so.

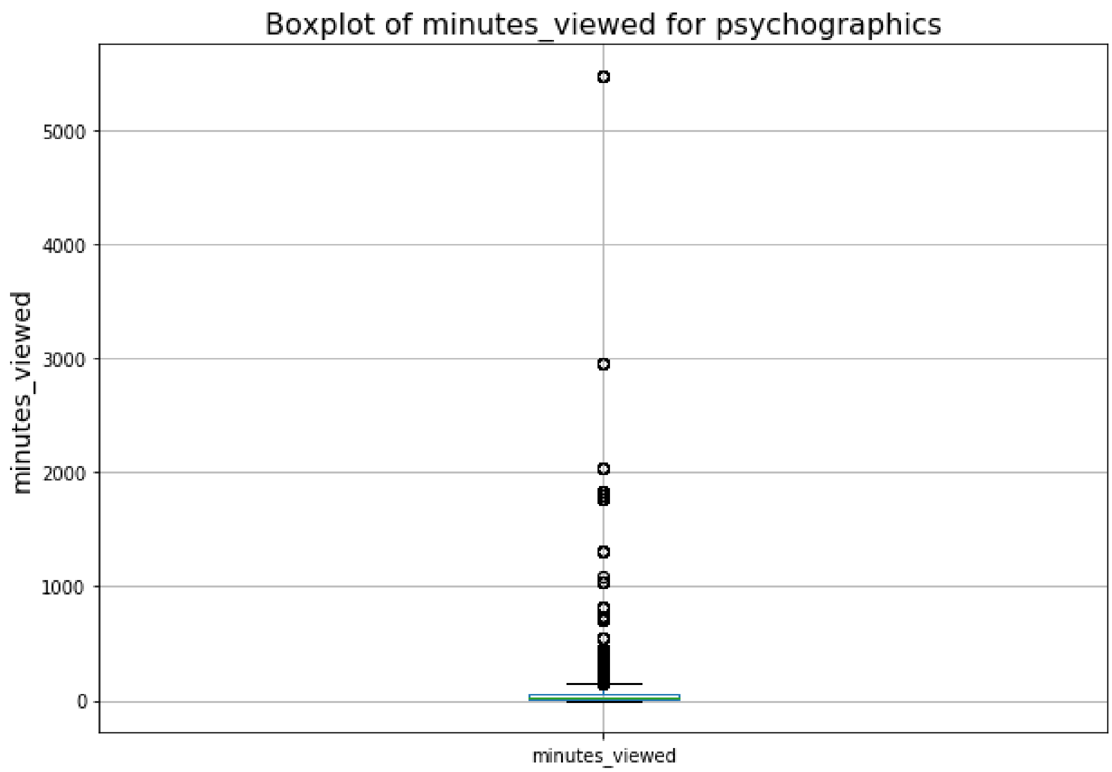
[23]:
```
plt.hist(df_psych['minutes_viewed'], bins = 25)

plt.xlabel("Total minutes viewed for each play")
plt.ylabel("Count of minutes viewed ranges")
plt.title("Distribution of minutes viewed for each play")
plt.show()
```



This distribution has a very large right skew, with its maximum value at 18,078 minutes viewed, per the statistcal summary. Some must be ones people left playing and they weren't watching. These outliers should be eliminated.There are also many minutes viewed near 0, which could be people accidentally hitting play or losing interest after the first few minutes.

[24]:
```
fig = plt.figure(figsize =(10, 7))
# Creating axes instance
boxplot = df_psych.boxplot(column=['minutes_viewed'])
plt.title("Boxplot of minutes_viewed for psychographics", fontsize = 16)
plt.ylabel("minutes_viewed", fontsize= 14 )
```

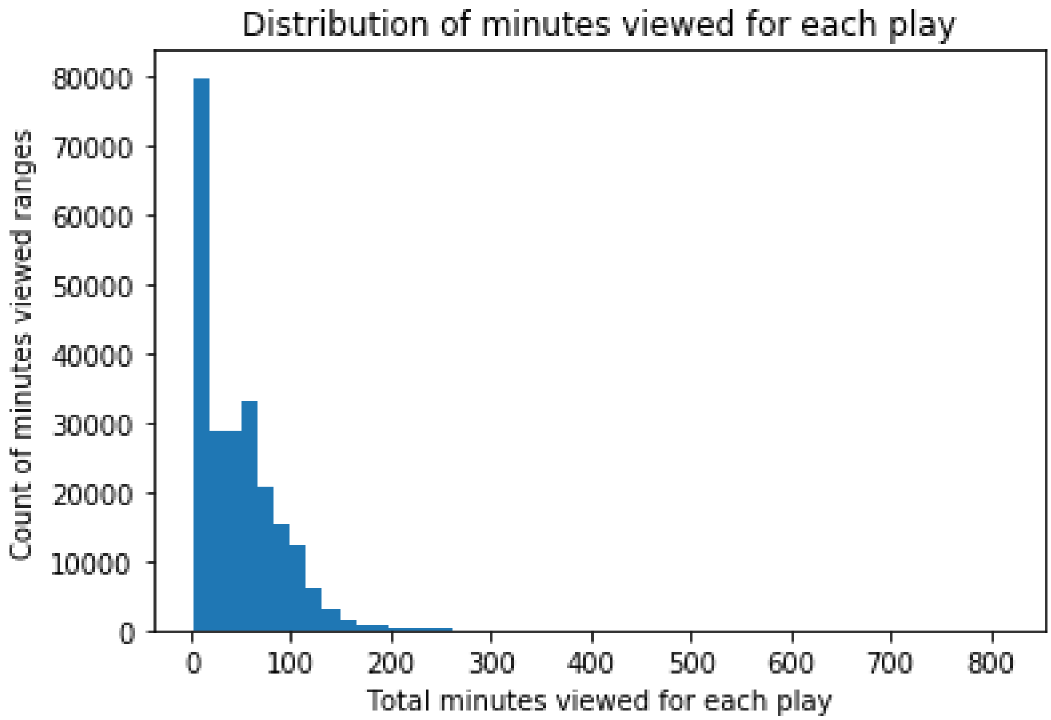
[24]:
```
Text(0, 0.5, 'minutes_viewed')
```



From the boxplot we can see that majority of the viewing time stays within a movie length as < 200 min, this can indicate that more people like to watch movies as opposed to TV shows. There are also some very heavy outliers within minutes_viewed which could represent people hitting play and letting the show or movie continue playing without actually watching.

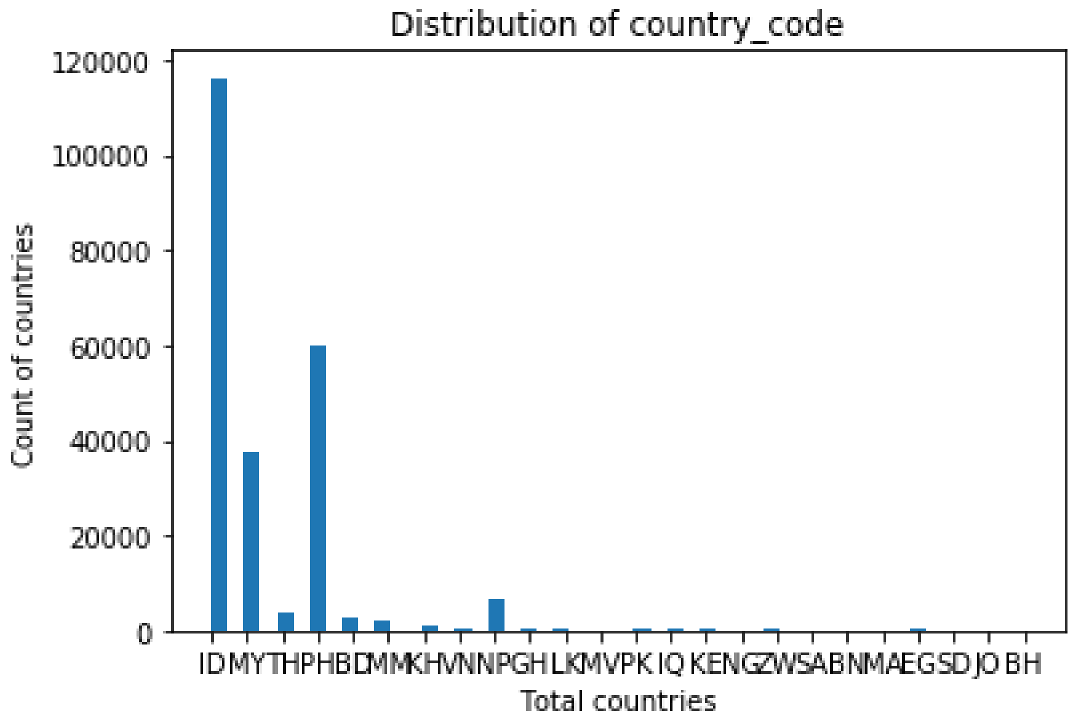
[25]:
```
# Setting some type of filter, can and will be changed to a better measure for␣
↪an outlier.

plays_nol = df_psych.loc[(df_psych['minutes_viewed'] < 850) &␣
↪(df_psych['minutes_viewed'] > 1)]
```

[26]:
```
plt.hist(plays_nol['minutes_viewed'], bins = 50)

plt.xlabel("Total minutes viewed for each play")
plt.ylabel("Count of minutes viewed ranges")
plt.title("Distribution of minutes viewed for each play")
plt.show()
```



This further shows that majority of the viewed times are within a movie length. There are a large amount of minutes viewed close to 0 minutes as well, which could perhaps be some data towards people hitting play and quickly changing their mind.

[27]:
```
plt.hist(plays_nol['country_code'], bins = 50)

plt.xlabel("Total countries")
plt.ylabel("Count of countries")
plt.title("Distribution of country_code")
plt.show()
```

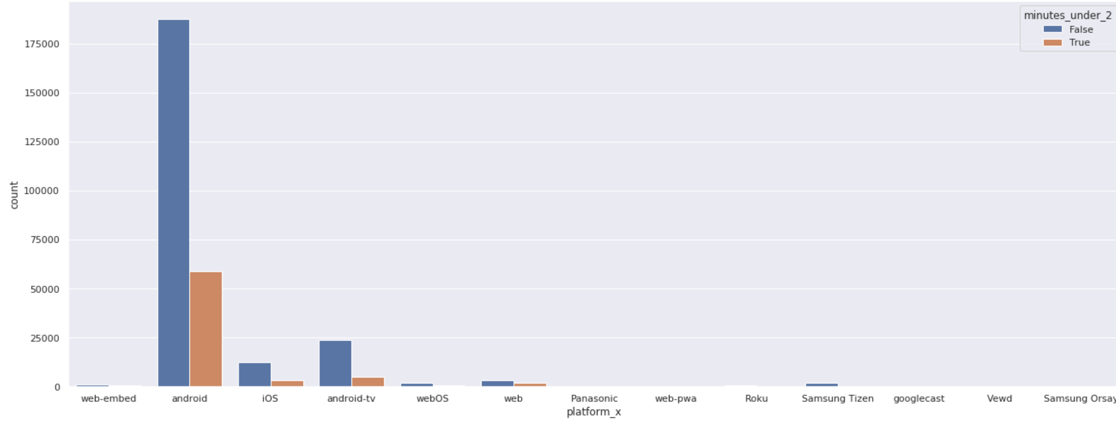


The top countries of our customers are from ID, MY, and PH.

[28]:
```
# Before continuing, lets add a column for where minutes_viewed < 2


df_psych['minutes_under_2'] = df_psych['minutes_viewed'] < 2
```

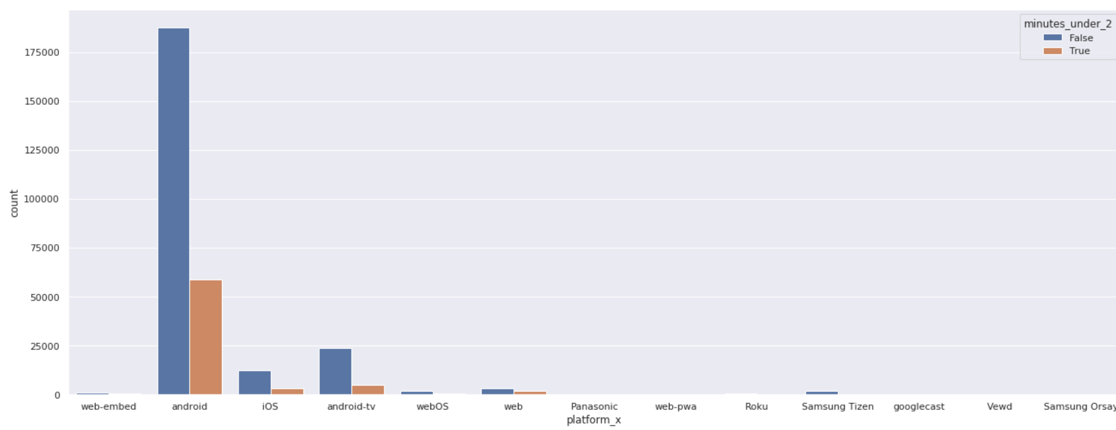
[29]:
```
# what types of devices are videos mostly played on?

sns.set(rc={'figure.figsize':(22,8.27)})

sns.countplot(x='platform_x',hue='minutes_under_2',data = df_psych)
```

[29]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2faec95b50>
```



[30]:
```
# what types of devices are videos mostly played on?

sns.set(rc={'figure.figsize':(22,8.27)})

sns.countplot(x='platform_x',hue='minutes_under_2',data = df_psych)
```

[30]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faedff110>



The majority of views are on an android. There are a relevant amount of records with views under 2 minutes as well, which could be data we don't need. Perhaps, though, that gives a good lens into movies or shows people thought about watching but didn't quite commit to.

[31]:
```
# what we ran to get rid of a redundant column

# df_psych = df_psych.drop(['platform_y'],axis = 1)
# df_psych.head()
```

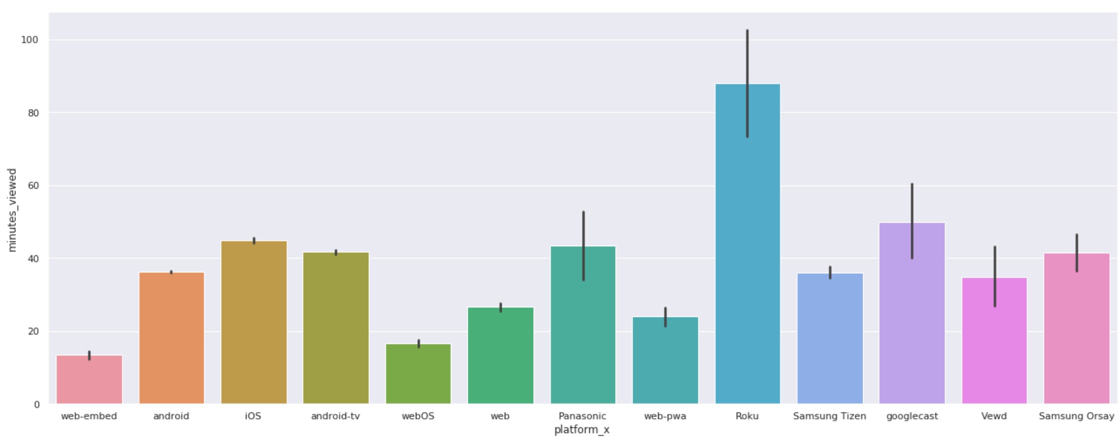
3.4 Upload the new df_psych to S3 bucket

[32]:
```
# what we ran to upload updated df to s3 bucket

# from io import StringIO

# bucket = 'ads508-team4-master'
# csv_buffer = StringIO()

# df_psych.to_csv(csv_buffer)
# s3_resource = boto3.resource('s3')
# s3_resource.Object(bucket, 'df_psych.csv').put(Body=csv_buffer.getvalue())
```

[33]:
```
# Seeing how long users watched on certain platforms

sns.set(rc={'figure.figsize':(22,8.27)})

sns.barplot(df_psych['platform_x'],df_psych['minutes_viewed'])
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

[33]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2faee47a90>
```



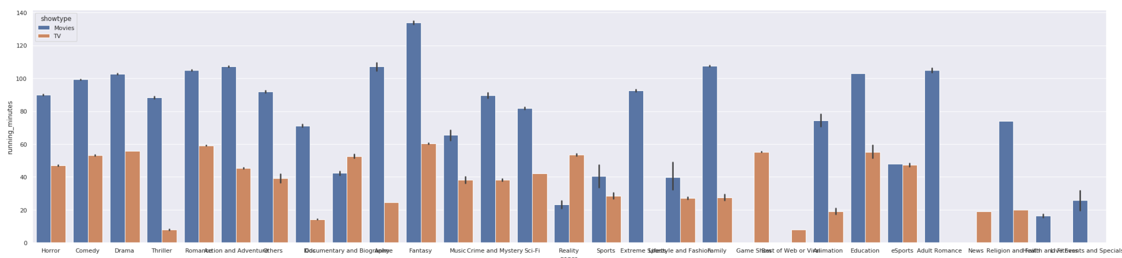
It appears videos on Roku would be viewed much longer than on other platforms even though the user base is in minority. The web-based streaming platforms are generally lower with minutes viewed while mobile devices like android and iOS are in the middle.

[34]:
```
# How long are show/movies? how do they differ? do genres play a role?
```

```
sns.set(rc={'figure.figsize':(36,8)})


sns.barplot(df_psych['genre'],df_psych['running_minutes'], hue =␣
 ↪df_psych['showtype'])
```

```
/opt/conda/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```

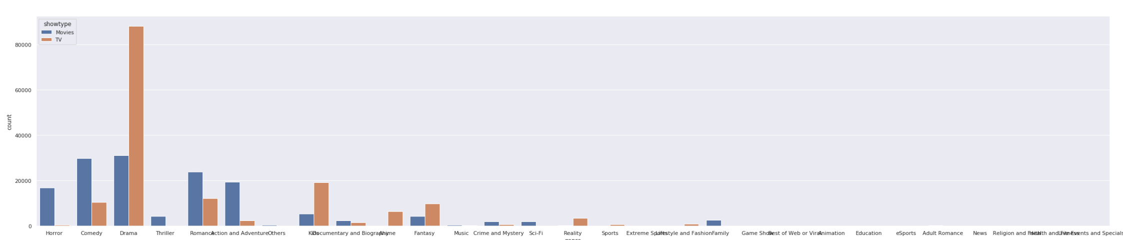
[34]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faef74d90>



Again, movies are definitely longer than tv shows except in a select few genres. genres that tend to have short movies than the rest are music, kids, animation and biographical. Kids and web shows are among the shortest shows. Longest movies are fantasy movies.

[35]:
```
# What genres are represented and how often are each in the dataset? do they␣
 ↪differ depending on show type?

sns.set(rc={'figure.figsize':(40,8)})

sns.countplot(x='genre',hue='showtype',data=df_psych)
```

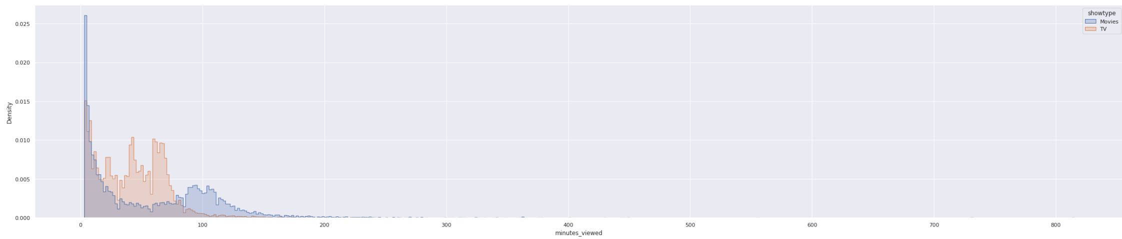
[35]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faf45a0d0>



There are many more tv shows in this data set than movies. The most popular genre for both shows and movies are Dramas. 2nd most popular for shows are kids shows and 2nd most for movies are comedies.

[36]:
```
df_psych_NOL = df_psych.loc[(df_psych['minutes_viewed'] < 850) &␣
↪(df_psych['minutes_viewed'] > 2)]

sns.
↪histplot(df_psych_NOL,x='minutes_viewed',hue='showtype',element='step',stat='density')
```

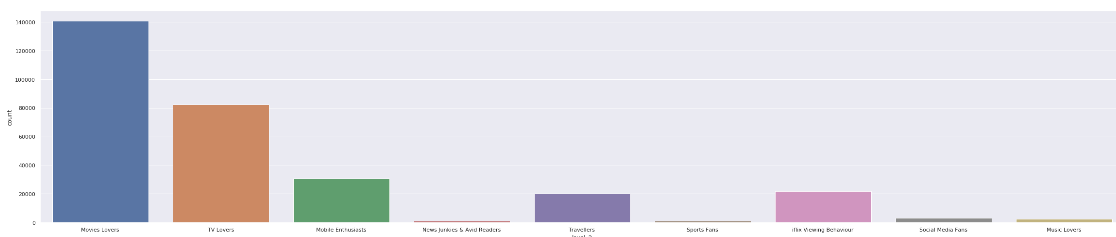
[36]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faefaa790>



Seems like when people watch something between 20 to 80 minutes, it's generally a tv show. If the view time is over 80 minutes, it is more likely a movie.

[37]:
```
# What types of psychographic traits are there?

sns.countplot(x='level_2',data= df_psych)
```

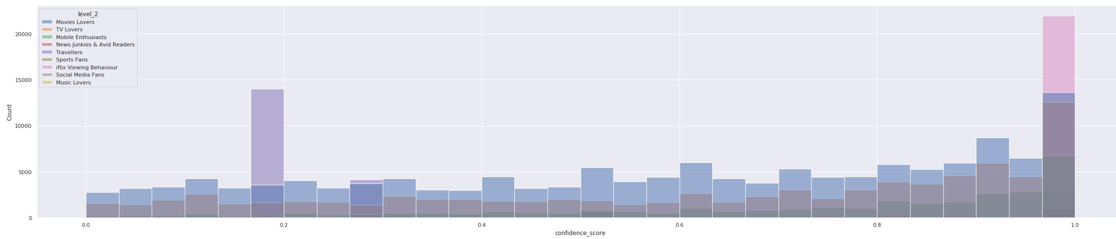
[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faf39f0d0>



From the barchart above we see that there are more Movie Lovers than other type of audience which matches our inital guess.

[38]:
```
sns.histplot(data=df_psych,x='confidence_score', hue='level_2', bins = 30)
```

[38]: <matplotlib.axes._subplots.AxesSubplot at 0x7f2faf25e550>



It looks like there is a roughly uniform distribution of confidence scores for level_2 traits for Movie Lovers, TV Lovers, and Music Lovers. There are roughly 15 thousand records with a potentially perfect confidence score, with a majority of them being due to iflix viewing behavior - a created feature iflix created already. This can induce bias in the data

[39]:
```
# what types of level 3 traits are there for psychographics?
```

[40]:
```
df_psych['level_3'].unique()
```

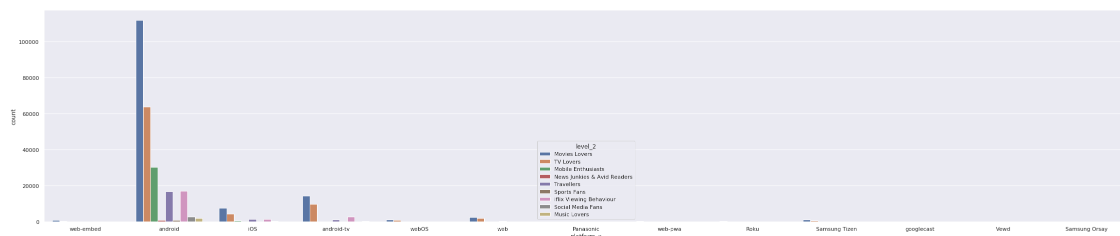
[40]:
```
array(['Horror Movies Fans', 'Indonesian Movies Fans',
       'Romance Movies Fans', 'Kids TV Fans', 'Drama TV Fans',
       'English TV Fans', 'Sports TV Fans', 'Malay TV Fans',
       'English Movies Fans', 'High Data Users',
       'Extreme Sports Movies Fans', 'Reality TV Fans', 'Comedy TV Fans',
       'Action and Adventure TV Fans', 'Comedy Movies Fans', nan,
       'Action and Adventure Movies Fans', 'Kids Movies Fans',
       'Local Commuters', 'Indonesian TV Fans', 'Thriller Movies Fans',
       'Korean TV Fans', 'Korean Movies Fans', 'Sci-Fi Movies Fans',
       'Drama Movies Fans', 'player', 'Malay Movies Fans',
       'Anime TV Fans', 'Fantasy Movies Fans', 'Family Movies Fans',
       '""""""""\\"""""""""\\""""""""""""""""', 'Japanese TV Fans', 'casual',
       'Romance TV Fans', 'Japanese Movies Fans', 'Anime Movies Fans',
       'Chinese Movies Fans', 'Thai Movies Fans', 'Horror TV Fans',
       'Crime and Mystery Movies Fans', 'Hindi Movies Fans',
       'Turkish Movies Fans', 'Tagalog Movies Fans', 'Tagalog TV Fans',
       'Vietnamese Movies Fans', 'French Movies Fans', 'addict',
       'Downloaders', 'Lifestyle and Fashion Movies Fans',
       'Reality Movies Fans', 'International Travellers',
       'Chinese TV Fans', 'Religion and Faith TV Fans',
       'Central Khmer Movies Fans', 'Thai TV Fans', 'Fantasy TV Fans',
       'Spanish; Castilian Movies Fans', 'Music TV Fans',
       'Lifestyle and Fashion TV Fans', 'Game Show TV Fans',
       'Documentary and Biography Movies Fans',
       'Documentary and Biography TV Fans', 'Crime and Mystery TV Fans',
       'Education TV Fans', 'Portuguese Movies Fans',
       'Danish Movies Fans', 'Adult Romance Movies Fans',
       'Thriller TV Fans', 'Animation Movies Fans', 'Bengali Movies Fans',
       'Others TV Fans', 'Health and Fitness Movies Fans',
```

```
'Music Movies Fans', 'Others Movies Fans', 'French TV Fans',
'Burmese Movies Fans', 'Sports Movies Fans', 'Tamil Movies Fans',
'Urdu Movies Fans', 'Religion and Faith Movies Fans',
'Animation TV Fans', 'Nepali Movies Fans', 'Nepali TV Fans',
'Education Movies Fans', 'Italian TV Fans', 'Burmese TV Fans',
'Kanuri TV Fans', 'Italian Movies Fans',
'Best of Web or Viral TV Fans', 'eSports TV Fans', 'Urdu TV Fans',
'Hindi TV Fans', 'Tajik Movies Fans', 'Hungarian Movies Fans',
'Spanish; Castilian TV Fans', 'Bengali TV Fans',
'eSports Movies Fans', 'Arabic Movies Fans', 'Swahili TV Fans',
'Family TV Fans', 'Afrikaans TV Fans', 'Norwegian TV Fans',
'Swedish Movies Fans', 'Arabic TV Fans',
'Live Events and Specials Movies Fans'], dtype=object)
```

There are many different types of level 3 traits that all spawn off of whatever the level 2 category is. It may prove difficult to predict for all of these traits. Perhaps, we can combine some into smaller levels as well.

[41]:
```
sns.countplot(data=df_psych,x='platform_x',hue = 'level_2')
```

[41]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f2fad9cb990>
```



Most of the mobile enthusiasts are android users. The other level 2 traits are roughly similar for all platforms. A bit more travellers with mobile devices.

4 Running Ad-Hoc Data Bias Analysis

[42]:
```
!pip install -q smclarify==0.1
```

```
/opt/conda/lib/python3.7/site-packages/secretstorage/dhcrypto.py:16:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
/opt/conda/lib/python3.7/site-packages/secretstorage/util.py:25:
CryptographyDeprecationWarning: int_from_bytes is deprecated, use int.from_bytes
instead
  from cryptography.utils import int_from_bytes
```

```
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
awscli 1.22.23 requires botocore==1.23.23, but you have botocore 1.23.24 which
is incompatible.
awscli 1.22.23 requires PyYAML<5.5,>=3.10, but you have pyyaml 6.0 which is
incompatible.
awscli 1.22.23 requires rsa<4.8,>=3.1.2, but you have rsa 4.8 which is
incompatible.
WARNING: Running pip as the 'root' user can result in broken permissions
and conflicting behaviour with the system package manager. It is recommended to
use a virtual environment instead: https://pip.pypa.io/warnings/venv
WARNING: You are using pip version 21.3.1; however, version 22.0.4 is
available.
You should consider upgrading via the '/opt/conda/bin/python -m pip install
--upgrade pip' command.
```

[43]:
```
from smclarify.bias import report
```

4.1 Calculate Bias Metrics on imbalanced data

[44]:
```
facet_column = report.FacetColumn(name='platform_x')

label_column = report.LabelColumn(
    name = "level_2",
    data = df_psych["level_2"],
    positive_label_values = ['Movie Lovers','TV Lovers'])
```

4.2 Run Sagemaker Clarify Bias Report

[45]:
```
report.bias_report(
    df=df_psych,
    facet_column=facet_column,
    label_column=label_column,
    stage_type=report.StageType.PRE_TRAINING,
    metrics=["CI","DPL","KL","JS","LP","TVD","KS"]
    )
```

```
[45]: [{'value_or_threshold': 'web-embed',
        'metrics': [{'name': 'CI',
          'description': 'Class Imbalance (CI)',
          'value': 0.9905281588162502},
         {'name': 'DPL',
          'description': 'Difference in Positive Proportions in Labels (DPL)',
          'value': -0.04039952739841901},
         {'name': 'JS',
          'description': 'Jensen-Shannon Divergence (JS)',
          'value': 0.001987138804840232},
         {'name': 'KL',
          'description': 'Kullback-Liebler Divergence (KL)',
          'value': 0.003905058713491401},
         {'name': 'KS',
          'description': 'Kolmogorov-Smirnov Distance (KS)',
          'value': 0.04039952739841901},
         {'name': 'LP',
          'description': 'L-p Norm (LP)',
          'value': 0.05713355956030760 6},
         {'name': 'TVD',
          'description': 'Total Variation Distance (TVD)',
          'value': 0.04039952739841901}]},
       {'value_or_threshold': 'android',
        'metrics': [{'name': 'CI',
          'description': 'Class Imbalance (CI)',
          'value': -0.6213764777125405},
         {'name': 'DPL',
          'description': 'Difference in Positive Proportions in Labels (DPL)',
          'value': 0.06372577570837396},
         {'name': 'JS',
          'description': 'Jensen-Shannon Divergence (JS)',
          'value': 0.0034418366034751095},
         {'name': 'KL',
          'description': 'Kullback-Liebler Divergence (KL)',
          'value': 0.01008689165279715},
         {'name': 'KS',
          'description': 'Kolmogorov-Smirnov Distance (KS)',
          'value': 0.06372577570837396},
         {'name': 'LP',
          'description': 'L-p Norm (LP)',
          'value': 0.09012185627952833},
         {'name': 'TVD',
          'description': 'Total Variation Distance (TVD)',
          'value': 0.06372577570837393}]},
       {'value_or_threshold': 'iOS',
        'metrics': [{'name': 'CI',
          'description': 'Class Imbalance (CI)',
```

```
   'value': 0.8967115136515627},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': -0.002728127052515572},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 8.476573435870475e-06},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 1.8758832900078997e-05},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.002728127052515683},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.003858154277544537},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.0027281270525156276}]},
{'value_or_threshold': 'android-tv',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.8090295147573787},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': -0.07655546950841219},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 0.005854103704981503},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 0.01361901061590369},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.07655546950841219},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.10826578325263646},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.07655546950841219}]},
{'value_or_threshold': 'webOS',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.9829849135093863},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
```

```
    'value': -0.08901926756825718},
   {'name': 'JS',
    'description': 'Jensen-Shannon Divergence (JS)',
    'value': 0.00929789739107772},
   {'name': 'KL',
    'description': 'Kullback-Liebler Divergence (KL)',
    'value': 0.017985168986072206},
   {'name': 'KS',
    'description': 'Kolmogorov-Smirnov Distance (KS)',
    'value': 0.08901926756825718},
   {'name': 'LP', 'description': 'L-p Norm (LP)', 'value': 0.1258922555075487},
   {'name': 'TVD',
    'description': 'Total Variation Distance (TVD)',
    'value': 0.08901926756825718}]},
 {'value_or_threshold': 'web',
  'metrics': [{'name': 'CI',
    'description': 'Class Imbalance (CI)',
    'value': 0.965252363023617},
   {'name': 'DPL',
    'description': 'Difference in Positive Proportions in Labels (DPL)',
    'value': -0.11253716789038248},
   {'name': 'JS',
    'description': 'Jensen-Shannon Divergence (JS)',
    'value': 0.01442398189636883},
   {'name': 'KL',
    'description': 'Kullback-Liebler Divergence (KL)',
    'value': 0.028184746548172476},
   {'name': 'KS',
    'description': 'Kolmogorov-Smirnov Distance (KS)',
    'value': 0.11253716789038248},
   {'name': 'LP', 'description': 'L-p Norm (LP)', 'value': 0.1591515891016369},
   {'name': 'TVD',
    'description': 'Total Variation Distance (TVD)',
    'value': 0.11253716789038248}]},
 {'value_or_threshold': 'Panasonic',
  'metrics': [{'name': 'CI',
    'description': 'Class Imbalance (CI)',
    'value': 0.9997103815065428},
   {'name': 'DPL',
    'description': 'Difference in Positive Proportions in Labels (DPL)',
    'value': -0.13799177938886437},
   {'name': 'JS',
    'description': 'Jensen-Shannon Divergence (JS)',
    'value': 0.022145147679870813},
   {'name': 'KL',
    'description': 'Kullback-Liebler Divergence (KL)',
    'value': 0.04143396645408663},
```

```
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.13799177938886437},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.19514984590772808},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.13799177938886434}]},
{'value_or_threshold': 'web-pwa',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.9971169795424027},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': -0.06459043393969821},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 0.00503428267900028},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 0.009700116789021086},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.06459043393969821},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.09134466767708467},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.06459043393969821}]},
{'value_or_threshold': 'Roku',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.9963336931623706},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': 0.059378770483250576},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 0.004678625590599604},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 0.009932757294777168},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.059378770483250576},
```

```
  {'name': 'LP', 'description': 'L-p Norm (LP)', 'value': 0.0839742625344521},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.05937877048325052}]},
{'value_or_threshold': 'Samsung Tizen',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.9861838814143914},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': -0.01915134748535463},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 0.000451194356501793},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 0.0009025026219929982},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.01915134748535463},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.027084095351508308},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
   'value': 0.019151347485354575}]},
{'value_or_threshold': 'googlecast',
 'metrics': [{'name': 'CI',
   'description': 'Class Imbalance (CI)',
   'value': 0.9995194965904005},
  {'name': 'DPL',
   'description': 'Difference in Positive Proportions in Labels (DPL)',
   'value': 0.05195351191570008},
  {'name': 'JS',
   'description': 'Jensen-Shannon Divergence (JS)',
   'value': 0.003567926747835403},
  {'name': 'KL',
   'description': 'Kullback-Liebler Divergence (KL)',
   'value': 0.007489604893020217},
  {'name': 'KS',
   'description': 'Kolmogorov-Smirnov Distance (KS)',
   'value': 0.05195351191570008},
  {'name': 'LP',
   'description': 'L-p Norm (LP)',
   'value': 0.07347336116409525},
  {'name': 'TVD',
   'description': 'Total Variation Distance (TVD)',
```

```
    'value': 0.05195351191570008}]},
  {'value_or_threshold': 'Vewd',
   'metrics': [{'name': 'CI',
     'description': 'Class Imbalance (CI)',
     'value': 0.9993483583897211},
    {'name': 'DPL',
     'description': 'Difference in Positive Proportions in Labels (DPL)',
     'value': -0.011712986975456974},
    {'name': 'JS',
     'description': 'Jensen-Shannon Divergence (JS)',
     'value': 0.00017192267659395 28},
    {'name': 'KL',
     'description': 'Kullback-Liebler Divergence (KL)',
     'value': 0.00034109163857989205},
    {'name': 'KS',
     'description': 'Kolmogorov-Smirnov Distance (KS)',
     'value': 0.011712986975456974},
    {'name': 'LP',
     'description': 'L-p Norm (LP)',
     'value': 0.01656466503659067},
    {'name': 'TVD',
     'description': 'Total Variation Distance (TVD)',
     'value': 0.011712986975456974}]},
  {'value_or_threshold': 'Samsung Orsay',
   'metrics': [{'name': 'CI',
     'description': 'Class Imbalance (CI)',
     'value': 0.9986572233485164},
    {'name': 'DPL',
     'description': 'Difference in Positive Proportions in Labels (DPL)',
     'value': -0.06716127322475934},
    {'name': 'JS',
     'description': 'Jensen-Shannon Divergence (JS)',
     'value': 0.005442719319812666},
    {'name': 'KL',
     'description': 'Kullback-Liebler Divergence (KL)',
     'value': 0.010457336330327069},
    {'name': 'KS',
     'description': 'Kolmogorov-Smirnov Distance (KS)',
     'value': 0.0671612732247594},
    {'name': 'LP',
     'description': 'L-p Norm (LP)',
     'value': 0.09498038346069972},
    {'name': 'TVD',
     'description': 'Total Variation Distance (TVD)',
     'value': 0.06716127322475937}]}]
```

There is a lot of class imbalance, and this will need to be fixed in the data preparation step to balance the data accordingly.

5 Conclusion

From the data exploration we can conclude that most of our user base enjoys drama movies more and uses android as their main platform; the top 3 countries of our user base is ID, PH, and MY. Other features inclduing level_3 traits, source_language, season_id, series_id and studio_id contain missing values and will be cleaned in the next stage along with balancing the data.