DATA 1202-DATA ANALYSIS TOOLS ANALYTICS

Assignment #4

Submitted by: Group #7
Aysegul Yalcinkaya - 100841003
Sayeed S Ahmed - 100853349
Oluseye Ibitoye - 100873496
Shedeva Campbell - 100867998
Ifeoluwa Owolabi - 100888430

In this assignment we defined 3 functions to make our code more modular. We also passed the number of records, that we want to get, as a parameter to the function to make the function reusable for different number of records.
For the first question, we got top 1000 records from the dataframe after reading youtube.csv file. Then we calculated the channeltype distribution both for all records and top 1000 records. At the end we plot the values in a bar chart.
For the second question, we saved the top 1000 rows into topresults.csv file. Then we imported this csv into youtube table in data1202 schema.
In the Test section of the assignment we displayed the number of rows and columns of youtube.csv file
In addition, we displayed the number of rows and columns in youtube table and the first 5 rows of table data

Import libraries

In [1]:
```
import pandas as pd
from sqlalchemy import create_engine
import pymysql
import matplotlib.pyplot as plt
```

Function that returns the Top N rows

top_n function takes 2 arguments, dataframe and an integer value. This function gets the first N number of rows from the given dataframe (df) and returns the Top N rows.

In [2]:
```
def top_n(df,n):
    df_top_n=df.iloc[:n,:]
    return df_top_n
```

Function that calculates the distribution of channel type

channeltype_distribution function takes 1 arguments, dataframe and returns the channeltype distribution.

In [3]:
```
# Function to get Top N rows from dataframe
# and calculate the distribution of channel type
def channeltype_distribution(df_top_n):
    distribution=df_top_n["channeltype"].value_counts()
    return distribution
```

Function that load records into a csv file and database

load_top_n function takes 1 arguments, dataframe. This function loads the records into "topresults.csv" file with headers. In the next step it loads the data into "youtube" table in "data1202" schema. If "youtube" table already exists, it is replaced.

In [4]:
```
def load_top_n(df_top_n):

    # write top x data into csv file
    df_top_n.to_csv("topresults.csv",index=False)

    # load Top N rows into database
    try:
        engine=create_engine('mysql+pymysql://root:@localhost/data1202')
        df_top_n.to_sql("youtube",engine,if_exists='replace',index=False)
    except:
        print("Could not load data into database")
```

Read data from youtube.csv file

In [5]:
```
df=pd.read_csv("youtube_dataset.csv")
```

In [6]:
```
df.head()
```

Out[6]:

	web-scraper-order	web-scraper-start-url	userID	user
0	1553043067-5148	https://socialblade.com/youtube/top/5000/mosts...	PewDiePie	https://socialblade.com/youtube/c/pe
1	1553043063-5147	https://socialblade.com/youtube/top/5000/mosts...	T-Series	https://socialblade.com/youtube/c/tseri
2	1553043059-5146	https://socialblade.com/youtube/top/5000/mosts...	Gaming	https://socialblade.com/youtube/channel/UC
3	1553043055-5145	https://socialblade.com/youtube/top/5000/mosts...	YouTube Movies	https://socialblade.com/youtube/channel/U
4	1553043051-5144	https://socialblade.com/youtube/top/5000/mosts...	Sports	https://socialblade.com/youtube/channel/U

In [7]:
```
df.shape
```

Out[7]:
```
(3944, 20)
```

In [8]:
```
distribution_all=channeltype_distribution(df)
```

Call top_n function with parameters df and 1000. (Top 1000 rows will be returned)

In [9]:
```
df_top_1000=top_n(df,1000)
```

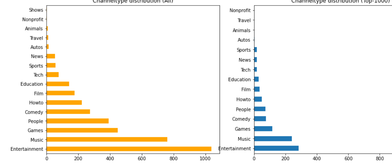
Call channeltype_distribution function with parameter df_top_1000. (Distribution of top 1000 rows will be returned)

In [10]:
```
distribution=channeltype_distribution(df_top_1000)
distribution
```

Out[10]:
```
Entertainment    284
Music            240
Games            115
Comedy            76
People            72
Howto             49
Film              36
Education         30
Tech              19
News              17
Sports            17
Autos              3
Animals            2
Travel             1
Nonprofit          1
Name: channeltype, dtype: int64
```

Plot top 1000 rows channeltype distribution

In [11]:
```
fig, axes = plt.subplots(nrows=1,ncols=2,figsize=(15, 6))
distribution_all.plot(ax= axes[0], kind="barh",color="orange",
                      title="Channeltype distribution (All)")
axes[1].set_xlim(0,1100)
distribution.plot(ax = axes[1], kind="barh",
                  title="Channeltype distribution (Top-1000)")
plt.show()
```



Call load_top_X function with parameters df and 1000. (Top 1000 rows will be saved into csv file and imported into database)

In [ ]:
```
load_top_n(df_top_1000)
```

TESTS

This section shows that top 1000 records are saved into topresults.csv and importes into database youtube table.

Get content of topresults.csv in order to see if it is loaded

In [ ]:
```
topresults=pd.read_csv("topresults.csv")
topresults.shape
```

In [ ]:
```
topresults.head()
```

Connect to database and select data from youtube table

In [ ]:
```
try:
    engine=create_engine('mysql+pymysql://root:@localhost/data1202')
    conn=engine.connect()
    df=pd.read_sql("select * from youtube",conn)
    print("Number of records and columns in youtube table ",df.shape)
    print(df.head())
except:
    print("Could not get data from database")
```

Assignment Log

Aysegul Yalcinkaya - 100841003	Coding
Shedeva Campbell - 100867998	Coding
Oluseye Ibitoye - 100873496	Reporting & Comments
Sayeed S Ahmed - 100853349	Testing
Ifeoluwa Owolabi - 100888430	Testing