Project_01

June 22, 2021

1 Yandex_Music

Based on the data from Yandex.Music it's required to compare the behavior of the users in two cities.

Survey target — to test three hypotheses: 1. Users activity is depend on the day of the week. Users activity in Moscow and Saint-Petersburg is different.

2. On monday morning in Moscow several types of music genres is popular, but in saint-Petersburg is other genres. On friday evening is the same differences in users behavior.
3. Users in Moscow and Saint-Petersburg prefer different music genres. Most popular genre in Moscow - pop, in Saint-Petersburg - rap.

1.1 Data overview

[1]:
```
# Pandas library import
import pandas as pd
```

[2]:
```
# import of data
df = pd.read_csv('yandex_music_project.csv',index_col=[0] )
```

[3]:
```
# print o first 10 rows of df
df.head(10)
```

[3]:
```
     userID                     Track            artist   genre  \
0  FFB692EC          Kamigata To Boots  The Mass Missile    rock
1  55204538  Delayed Because of Accident  Andreas Rönnberg    rock
2    20EC38          Funiculì funiculà       Mario Lanza     pop
3  A3DD03C9        Dragons in the Sunset        Fire + Ice    folk
4  E2DC1FAE               Soul People        Space Echo   dance
5  842029A1                               IMPERVTOR  rusrap
6  4CB90AA5                      True      Roman Messer   dance
7  F03E1C1F          Feeling This Way   Polina Griffith   dance
8  8FA1D3BE                                 NaN  ruspop
9  E772D5C0                 Pessimist               NaN   dance

               City      time        Day
0  Saint-Petersburg  20:28:33  Wednesday
```

```
1             Moscow  14:07:09     Friday
2  Saint-Petersburg  20:58:07  Wednesday
3  Saint-Petersburg  08:37:09     Monday
4            Moscow  08:34:34     Monday
5  Saint-Petersburg  13:09:41     Friday
6            Moscow  13:00:07  Wednesday
7            Moscow  20:47:49  Wednesday
8            Moscow  09:17:40     Friday
9  Saint-Petersburg  21:20:49  Wednesday
```

[4]:
```
# print of overall information of df
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 65079 entries, 0 to 65078
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0     userID  65079 non-null  object
 1   Track     63848 non-null  object
 2   artist    57876 non-null  object
 3   genre     63881 non-null  object
 4     City    65079 non-null  object
 5   time      65079 non-null  object
 6   Day       65079 non-null  object
dtypes: object(7)
memory usage: 4.0+ MB
```

Conclusions

In every row of df is information on the music track. Some columns describe the track itself: name, compsitor, genre. Other columns provide the information on the user: city, day and time of the listening.

Preiminary we can assume that it's enough data to check the hypotheses, hovewer df has nulls in data and has some colomn names in "bad" style.

1.2 Data Preparation

1.2.1 Column names style

[5]:
```
# print of df columns names
df.columns
```

[5]:
```
Index(['  userID', 'Track', 'artist', 'genre', '  City  ', 'time', 'Day'],
dtype='object')
```

[6]:
```
# rename of the bad columns names
df = df.rename(columns={'  userID':'user_id','Track':'track','  City  ':
↪'city','Day':'day'})
```

[7]:
```
# chechk of the results
df.columns
```

[7]:
```
Index(['user_id', 'track', 'artist', 'genre', 'city', 'time', 'day'],
dtype='object')
```

1.2.2 Nulls processing

[8]:
```
# Calculation of nulls
df.isna().sum()
```

[8]:
```
user_id        0
track       1231
artist      7203
genre       1198
city           0
time           0
day            0
dtype: int64
```

[9]:
```
# selection of the columns with nulls and replace the nulls with 'unknown'

columns_to_replace = ['track','artist','genre']

for i in columns_to_replace:
    df[i] = df[i].fillna('unknown')
```

[10]:
```
# Chechk of the result
df.isna().sum()
```

[10]:
```
user_id    0
track      0
artist     0
genre      0
city       0
time       0
day        0
dtype: int64
```

1.2.3 Duplicates processing

[11]:
```
# calculation of obvious duplicates
df.duplicated().sum()
```

[11]:
```
3826
```

[12]:
```
# deletion of obvious duplicates (with index resetting)
df = df.drop_duplicates().reset_index(drop=True)
```

[13]:
```python
# check of the results
df.duplicated().sum()
```

[13]: 0

[14]:
```python
# chech of unique genres
df['genre'].sort_values().unique()
```

[14]:
```
array(['acid', 'acoustic', 'action', 'adult', 'africa', 'afrikaans',
       'alternative', 'alternativepunk', 'ambient', 'americana',
       'animated', 'anime', 'arabesk', 'arabic', 'arena',
       'argentinetango', 'art', 'audiobook', 'author', 'avantgarde',
       'axé', 'baile', 'balkan', 'beats', 'bigroom', 'black', 'bluegrass',
       'blues', 'bollywood', 'bossa', 'brazilian', 'breakbeat', 'breaks',
       'broadway', 'cantautori', 'cantopop', 'canzone', 'caribbean',
       'caucasian', 'celtic', 'chamber', 'chanson', 'children', 'chill',
       'chinese', 'choral', 'christian', 'christmas', 'classical',
       'classicmetal', 'club', 'colombian', 'comedy', 'conjazz',
       'contemporary', 'country', 'cuban', 'dance', 'dancehall',
       'dancepop', 'dark', 'death', 'deep', 'deutschrock', 'deutschspr',
       'dirty', 'disco', 'dnb', 'documentary', 'downbeat', 'downtempo',
       'drum', 'dub', 'dubstep', 'eastern', 'easy', 'electronic',
       'electropop', 'emo', 'entehno', 'epicmetal', 'estrada', 'ethnic',
       'eurofolk', 'european', 'experimental', 'extrememetal', 'fado',
       'fairytail', 'film', 'fitness', 'flamenco', 'folk', 'folklore',
       'folkmetal', 'folkrock', 'folktronica', 'forró', 'frankreich',
       'französisch', 'french', 'funk', 'future', 'gangsta', 'garage',
       'german', 'ghazal', 'gitarre', 'glitch', 'gospel', 'gothic',
       'grime', 'grunge', 'gypsy', 'handsup', "hard'n'heavy", 'hardcore',
       'hardstyle', 'hardtechno', 'hip', 'hip-hop', 'hiphop',
       'historisch', 'holiday', 'hop', 'horror', 'house', 'hymn', 'idm',
       'independent', 'indian', 'indie', 'indipop', 'industrial',
       'inspirational', 'instrumental', 'international', 'irish', 'jam',
       'japanese', 'jazz', 'jewish', 'jpop', 'jungle', 'k-pop',
       'karadeniz', 'karaoke', 'kayokyoku', 'korean', 'laiko', 'latin',
       'latino', 'leftfield', 'local', 'lounge', 'loungeelectronic',
       'lovers', 'malaysian', 'mandopop', 'marschmusik', 'meditative',
       'mediterranean', 'melodic', 'metal', 'metalcore', 'mexican',
       'middle', 'minimal', 'miscellaneous', 'modern', 'mood', 'mpb',
       'muslim', 'native', 'neoklassik', 'neue', 'new', 'newage',
       'newwave', 'nu', 'nujazz', 'numetal', 'oceania', 'old', 'opera',
       'orchestral', 'other', 'piano', 'podcasts', 'pop', 'popdance',
       'popelectronic', 'popeurodance', 'poprussian', 'post',
       'posthardcore', 'postrock', 'power', 'progmetal', 'progressive',
       'psychedelic', 'punjabi', 'punk', 'quebecois', 'ragga', 'ram',
       'rancheras', 'rap', 'rave', 'reggae', 'reggaeton', 'regional',
       'relax', 'religious', 'retro', 'rhythm', 'rnb', 'rnr', 'rock',
```

```
       'rockabilly', 'rockalternative', 'rockindie', 'rockother',
       'romance', 'roots', 'ruspop', 'rusrap', 'rusrock', 'russian',
       'salsa', 'samba', 'scenic', 'schlager', 'self', 'sertanejo',
       'shanson', 'shoegazing', 'showtunes', 'singer', 'ska', 'skarock',
       'slow', 'smooth', 'soft', 'soul', 'soulful', 'sound', 'soundtrack',
       'southern', 'specialty', 'speech', 'spiritual', 'sport',
       'stonerrock', 'surf', 'swing', 'synthpop', 'synthrock',
       'sängerportrait', 'tango', 'tanzorchester', 'taraftar', 'tatar',
       'tech', 'techno', 'teen', 'thrash', 'top', 'traditional',
       'tradjazz', 'trance', 'tribal', 'trip', 'triphop', 'tropical',
       'türk', 'türkçe', 'ukrrock', 'unknown', 'urban', 'uzbek',
       'variété', 'vi', 'videogame', 'vocal', 'western', 'world',
       'worldbeat', 'ïîî', '       '], dtype=object)
```

[15]:
```
# function for replacing of duplicated genres
def replace_wrong_genres (wrong_genres, correct_genre):
    hip_hop_list = wrong_genres
    for i in hip_hop_list:
        df['genre'] = df['genre'].replace(i, correct_genre)
    return df
```

[16]:
```
# deletion of implicit duplicates
wrong_genres_list = ['hip', 'hop', 'hip-hop']
df = replace_wrong_genres (wrong_genres_list, 'hiphop')
```

[17]:
```
# chechk of the result
df_genre = df['genre']
df_genre = df_genre.sort_values()
df_genre.unique()
```

[17]:
```
array(['acid', 'acoustic', 'action', 'adult', 'africa', 'afrikaans',
       'alternative', 'alternativepunk', 'ambient', 'americana',
       'animated', 'anime', 'arabesk', 'arabic', 'arena',
       'argentinetango', 'art', 'audiobook', 'author', 'avantgarde',
       'axé', 'baile', 'balkan', 'beats', 'bigroom', 'black', 'bluegrass',
       'blues', 'bollywood', 'bossa', 'brazilian', 'breakbeat', 'breaks',
       'broadway', 'cantautori', 'cantopop', 'canzone', 'caribbean',
       'caucasian', 'celtic', 'chamber', 'chanson', 'children', 'chill',
       'chinese', 'choral', 'christian', 'christmas', 'classical',
       'classicmetal', 'club', 'colombian', 'comedy', 'conjazz',
       'contemporary', 'country', 'cuban', 'dance', 'dancehall',
       'dancepop', 'dark', 'death', 'deep', 'deutschrock', 'deutschspr',
       'dirty', 'disco', 'dnb', 'documentary', 'downbeat', 'downtempo',
       'drum', 'dub', 'dubstep', 'eastern', 'easy', 'electronic',
       'electropop', 'emo', 'entehno', 'epicmetal', 'estrada', 'ethnic',
       'eurofolk', 'european', 'experimental', 'extrememetal', 'fado',
       'fairytail', 'film', 'fitness', 'flamenco', 'folk', 'folklore',
```

```
'folkmetal', 'folkrock', 'folktronica', 'forró', 'frankreich',
'französisch', 'french', 'funk', 'future', 'gangsta', 'garage',
'german', 'ghazal', 'gitarre', 'glitch', 'gospel', 'gothic',
'grime', 'grunge', 'gypsy', 'handsup', "hard'n'heavy", 'hardcore',
'hardstyle', 'hardtechno', 'hiphop', 'historisch', 'holiday',
'horror', 'house', 'hymn', 'idm', 'independent', 'indian', 'indie',
'indipop', 'industrial', 'inspirational', 'instrumental',
'international', 'irish', 'jam', 'japanese', 'jazz', 'jewish',
'jpop', 'jungle', 'k-pop', 'karadeniz', 'karaoke', 'kayokyoku',
'korean', 'laiko', 'latin', 'latino', 'leftfield', 'local',
'lounge', 'loungeelectronic', 'lovers', 'malaysian', 'mandopop',
'marschmusik', 'meditative', 'mediterranean', 'melodic', 'metal',
'metalcore', 'mexican', 'middle', 'minimal', 'miscellaneous',
'modern', 'mood', 'mpb', 'muslim', 'native', 'neoklassik', 'neue',
'new', 'newage', 'newwave', 'nu', 'nujazz', 'numetal', 'oceania',
'old', 'opera', 'orchestral', 'other', 'piano', 'podcasts', 'pop',
'popdance', 'popelectronic', 'popeurodance', 'poprussian', 'post',
'posthardcore', 'postrock', 'power', 'progmetal', 'progressive',
'psychedelic', 'punjabi', 'punk', 'quebecois', 'ragga', 'ram',
'rancheras', 'rap', 'rave', 'reggae', 'reggaeton', 'regional',
'relax', 'religious', 'retro', 'rhythm', 'rnb', 'rnr', 'rock',
'rockabilly', 'rockalternative', 'rockindie', 'rockother',
'romance', 'roots', 'ruspop', 'rusrap', 'rusrock', 'russian',
'salsa', 'samba', 'scenic', 'schlager', 'self', 'sertanejo',
'shanson', 'shoegazing', 'showtunes', 'singer', 'ska', 'skarock',
'slow', 'smooth', 'soft', 'soul', 'soulful', 'sound', 'soundtrack',
'southern', 'specialty', 'speech', 'spiritual', 'sport',
'stonerrock', 'surf', 'swing', 'synthpop', 'synthrock',
'sängerportrait', 'tango', 'tanzorchester', 'taraftar', 'tatar',
'tech', 'techno', 'teen', 'thrash', 'top', 'traditional',
'tradjazz', 'trance', 'tribal', 'trip', 'triphop', 'tropical',
'türk', 'türkçe', 'ukrrock', 'unknown', 'urban', 'uzbek',
'variété', 'vi', 'videogame', 'vocal', 'western', 'world',
'worldbeat', 'ïîï', '      '], dtype=object)
```

Conclusions

Data preparation has revealed 3 issues in data: - bad colomns names style; - nulls in data; - duplicated data - obvious and implicit

The columns names were corrected and duplicted deleted. The missing genres were replaced on "unknown".

After the completion of the data preparation we can start the hypothesis testing.

1.3 Hypothesis testing

1.3.1 Compare the behavior of the users in two cities

Users activity is depend on the day of the week. Users activity in Moscow and Saint-Petersburg is different.

[18]:
```
# calculation of playback in every city
df.groupby('city')['track'].count()
```

[18]:
```
city
Moscow              42741
Saint-Petersburg    18512
Name: track, dtype: int64
```

[19]:
```
# calculation of playback in every day
df.groupby('day')['track'].count()
```

[19]:
```
day
Friday       21840
Monday       21354
Wednesday    18059
Name: track, dtype: int64
```

[20]:
```
# creation of function number_tracks()

def number_tracks (city,day):
    track_list=df[df['day']==day]
    track_list=track_list[track_list['city']==city]
    track_list_count = track_list['user_id'].count()
    return track_list_count
```

[21]:
```
# quantity of playback in Moscow on Monday

moscow_monday = number_tracks('Moscow','Monday')
moscow_monday
```

[21]:
```
15740
```

[22]:
```
# quantity of playback in Saint-Petersburg on Monday

spb_monday = number_tracks('Saint-Petersburg','Monday')
spb_monday
```

[22]:
```
5614
```

[23]:
```
# quantity of playback in Moscow on Wednesday
moscow_wendsday = number_tracks('Moscow','Wednesday')
moscow_wendsday
```

[23]: 11056

[24]:
```
# quantity of playback in Saint-Petersburg on Wednesday
spb_wendsday = number_tracks('Saint-Petersburg','Wednesday')
spb_wendsday
```

[24]: 7003

[25]:
```
# quantity of playback in Moscow on Friday
moscow_friday = number_tracks('Moscow','Friday')
moscow_friday
```

[25]: 15945

[26]:
```
# quantity of playback in Saint-Petersburg on Friday
spb_friday = number_tracks('Saint-Petersburg','Friday')
spb_friday
```

[26]: 5895

[27]:
```
# print of results
pd.DataFrame(data=[
    ['Moscow', moscow_monday, moscow_wendsday, moscow_friday],
    ['Saint-Petersburg', spb_monday, spb_wendsday, spb_friday]],
    columns=['city', 'monday', 'wednesday', 'friday'])
```

[27]:

	city	monday	wednesday	friday
0	Moscow	15740	11056	15945
1	Saint-Petersburg	5614	7003	5895

Conclusion on Hypothesis one

The data shows the differences in users behavior: - The highest quantity of playbacks in Moscom is on Monday and Friday, but of the Wendsday it's lower. - In the Saint-Petersburg is opposite situation - the highest quantity is on Wendsday. On Monday and Friday the quantity of playbacks is lower than on Wendsday and almost the same.

- , .
- , , . .

Hypothesis is correct.

1.3.2 Music in the begining and end of the week

On monday morning in Moscow several types of music genres is popular, but in Saint-Petersburg is other genres. On friday evening is the same differences in users behavior.

[28]:
```
# getting the moscow_general from df, where 'city' column equal to 'Moscow'
moscow_general = df[df['city']=='Moscow']
```

[29]:
```
# getting the moscow_general from df, where 'city' column equal to␣
↪'Saint-Petersburg'
spb_general = df[df['city']=='Saint-Petersburg']
```

[30]:
```
# genre_weekday() function
def genre_weekday (table, day, time1, time2):
    genre_df = table[table['day']==day]
    genre_df = genre_df[genre_df['time']>=time1]
    genre_df = genre_df[genre_df['time']<=time2]
    genre_df_count = genre_df.groupby('genre')['track'].count()
    genre_df_sorted = genre_df_count.sort_values(ascending = False)
    return genre_df_sorted.head(10)
```

[31]:
```
# selection of the top genres in Moscow on Monday morning
genre_weekday(moscow_general, 'Monday', '07:00', '11:00')
```

[31]:
```
genre
pop            781
dance          549
electronic     480
rock           474
hiphop         286
ruspop         186
world          181
rusrap         175
alternative    164
unknown        161
Name: track, dtype: int64
```

[32]:
```
# selection of the top genres in Saint-Petersburg on Monday morning
genre_weekday(spb_general, 'Monday', '07:00', '11:00')
```

[32]:
```
genre
pop            218
dance          182
rock           162
electronic     147
hiphop          80
ruspop          64
alternative     58
rusrap          55
jazz            44
classical       40
Name: track, dtype: int64
```

[33]:
```
# selection of the top genres in Moscow on Friday evening
genre_weekday(moscow_general, 'Friday', '17:00', '23:00')
```

[33]:
```
genre
pop            713
rock           517
dance          495
electronic     482
hiphop         273
world          208
ruspop         170
alternative    163
classical      163
rusrap         142
Name: track, dtype: int64
```

[34]:
```
# selection of the top genres in Saint-Petersburg on Friday evening
genre_weekday(spb_general, 'Friday', '17:00', '23:00')
```

[34]:
```
genre
pop            256
electronic     216
rock           216
dance          210
hiphop          97
alternative     63
jazz            61
classical       60
rusrap          59
world           54
Name: track, dtype: int64
```

Hypothesis 2 conclusions

Comparing top 10 genres of Monday morning we can cocnclude the following: 1. In Moscow and Saint-Petersburg users are listening the similar music. The only difference is that in Moscow top 10 Monday morning rating was icnluded genre “world”, but in Saint-Petersburg “jazz” and “classic”.

2. In Moscow the were too many playbacks of tracks w/o genres that value “unkown” were placed on the 10th position of most popular genres. That means that data with nulls value has the sufficient portion in the data and puts in the danger the certainty of the survey.

On the Friday evenening the result of analysis is similar.Some of the genres goes up, another goes dwn, bot overall top 10 genres stay the same.

The second hepothesis was confirmed only partially: * Users are listeting the similar music in the beginning and the end of the week. * Differnece in genres priority in Moscow and Saint-Petersburg is slighlty different. In Moscow users prefer russian pop music, in Saint-Petersburg jazz

Hovewer the nulls in data is not allowed us to confirm the result on 100%. The quantity of null genres in Moscow is so high that the rationg of top 10 genres could be different if that data were not lost.

1.3.3 Genres preferences in Moscow and Saint-Petersburg

Users in Moscow and Saint-Petersburg prefer different music genres. Most popular genre in Moscow - pop, in Saint-Petersburg - rap.

[35]:
```
# Moscow_general df group by row 'genre'
moscow_genres = moscow_general.groupby('genre')['user_id'].count()
moscow_genres = moscow_genres.sort_values(ascending = False)
```

[36]:
```
# print of first 10 rows
moscow_genres.head(10)
```

[36]:
```
genre
pop            5892
dance          4435
rock           3965
electronic     3786
hiphop         2096
classical      1616
world          1432
alternative    1379
ruspop         1372
rusrap         1161
Name: user_id, dtype: int64
```

[37]:
```
# Saint-Peterpurg_general df group by row 'genre'
spb_genres = spb_general.groupby('genre')['user_id'].count()
spb_genres = spb_genres.sort_values(ascending = False)
```

[38]:
```
# print of first 10 rows
spb_genres.head(10)
```

[38]:
```
genre
pop            2431
dance          1932
rock           1879
electronic     1736
hiphop          960
alternative     649
classical       646
rusrap          564
ruspop          538
world           515
Name: user_id, dtype: int64
```

Third Hypothesis conclusions

Hypothesis is correct only partially: * Pop Music - the most popoular genre in Moscow, as it was origionally stated in hypothesis. Moreover the top 10 genres includes also russion pop music. *

Against the expectations rap is not the most popular genre in Saint-Petersburg, but it has similar popuarity in both cities.

1.4 Survey Conclusions

1. Day of the week has difference infuelnce on the users activity in Moscow and Saint-Petersburg.

First hypothesis is correct.

2. Music preferences has a slighly changes during the day of the week in Moscor or Saint-Petersburg. A small differences is appeared only in the beginning of the week, on mondays:

- in Moscow users prefer "world" genre
- in Saint-Petersbirg - jazz or classic

Therefore second hypothesis was correct only partially. The result of hyputhesis testing could be different if the data has full information on all genres (less nulls).

3. Users preferences in Moscow and Saint-Petersburg have more common rather than differenet. Against the expectations the genre preferences in both cities are very similar.

Third hypothesis is not correct. The difference in music preferences in the both cities is minor and could not be spotted on the most part of users.