

Goals

1. Collect information about users (if possible, or artists if usernames are inaccessible due to privacy settings) on last.fm API. Looking for information regarding:
 - a. For Users: Top artists and/or genres listened to, time spent listening, etc.
 - b. For Artists: Genre, popularity count?
2. Run each name through Genderize API to determine the gender of each user or Artist
3. Create two visualizations:
 - a. Polar area diagram separated by genre, visualizing the percentage of artists who are female/male.
 - b. Sunburst diagram separated by gender (male, female, none) visualizing the percentage of the total collected artists or users who were determined to identify as a specific gender by the API with their aggregate listeners.

Goals Achieved

1. Collected chart-topping artists from last.fm with the following information: name, genre. Stored in a database with an artist_id number for each artist for future linking.
2. Collected gender information based on names from last.fm data using Genderize including: artist_id, gender prediction, probability of correct prediction. Stored in the database, using two linked tables: one for assigning each gender a gender_id number and one for linking the artist information to gender/probability using a linked artist_id number and gender_id.
3. Created three visualizations:
 - a. Bar Graph of number of artists for each gender (Male, Female, None)
 - b. Radar Plot of number of artists of each gender, organized by genre.
 - c. Scatterplot of gender prediction probabilities of correctness, by gender (M, F)

Problems Faced

1. Collecting user data: At first, it seemed like public usernames might be available from last.fm, but upon further research we discovered that the only way to gather user information was to input a username and gather listening history individually, and there was no way to gather a list of usernames on the site. So, we switched to artist names instead, knowing that there would be a greater margin of error due to the fact that some artist names are not typically associated with gender (ex. Coldplay, The Strokes) and some artists would be reported as having no gender. We planned to report these numbers, but use the data regardless, because it also included probability of correctness in prediction.
2. Collecting genre information about artists: When gathering artists from the popularity chart, we realized that genre information was not included in the data. So, we decided to pick a limited list of genres ourselves and instead gather the top 25 artists per genre, rather than the top 100 artists total. This way, we could still analyze the gender data by genre.

3. Genderize API Limit: It was rather difficult to debug the functions that used Genderize to gather Gender predictions and insert them into the database, because we had a limit of 1000 names per day. It ended up taking a few days to debug some simple errors because of that, but luckily we were working ahead of schedule and it didn't cause too much worry!
4. 25 item limit adding to database: Like many students, we struggled with this requirement. We originally called the api 6 times, once for each genre of music to pull the top 25 from each, and then added the whole list to the database. To adjust for the requirement, we changed the format of the program so you must run it 6 times total, and edit the index of the genre being added each time you run it from 0 to 6 to collect a full range of genres.

Calculations File: resultsdemo.txt

Number of artists of each reported gender (M, F, None)

```
{"male": 107, "female": 33, "none": 10}
```

Collection of probabilities of each male artist prediction

```
[0.88, 0.92, 0.79, 0.99, 0.75, 0.97, 0.92, 0.86, 1.0, 0.95, 0.87,
0.98, 0.97, 0.94, 0.98, 1.0, 0.86, 0.8, 0.99, 0.88, 0.78, 0.98, 0.88,
0.98, 0.99, 0.95, 0.8, 1.0, 0.75, 0.75, 0.98, 0.93, 0.98, 0.99, 0.86,
0.95, 0.89, 0.86, 0.68, 0.92, 0.94, 0.99, 0.93, 0.86, 0.74, 0.95,
0.88, 0.86, 0.99, 0.9, 0.97, 0.98, 0.56, 0.99, 0.9, 0.67, 0.97, 0.96,
0.99, 0.99, 0.94, 0.94, 0.82, 0.99, 0.98, 0.99, 0.99, 0.99, 0.99,
0.99, 0.99, 0.98, 0.99, 0.99, 0.95, 0.98, 0.94, 0.95, 0.97, 0.99,
0.99, 0.99, 0.99, 0.99, 0.99, 0.86, 0.97, 0.86, 0.86, 0.83, 0.97,
0.61, 0.96, 0.65, 0.91, 0.86, 0.97, 1.0, 0.77, 0.86, 0.86, 1.0, 0.78,
0.76, 0.76, 0.86, 0.92]
```

Collection of probabilities of each female artist prediction

```
[0.71, 0.52, 0.74, 0.97, 0.97, 0.97, 0.97, 0.98, 0.94, 0.98, 0.92,
0.67, 0.96, 0.98, 0.77, 0.98, 0.95, 0.78, 0.96, 0.98, 0.77, 0.97,
0.98, 0.97, 0.99, 0.98, 0.93, 0.52, 0.98, 0.86, 0.98, 0.95, 0.59]
```

Number of male artists of each genre

```
{"rock": 18, "pop": 7, "folk": 22, "rnb": 16, "singer-songwriter":
22, "indie": 22}
```

Number of female artists of each genre

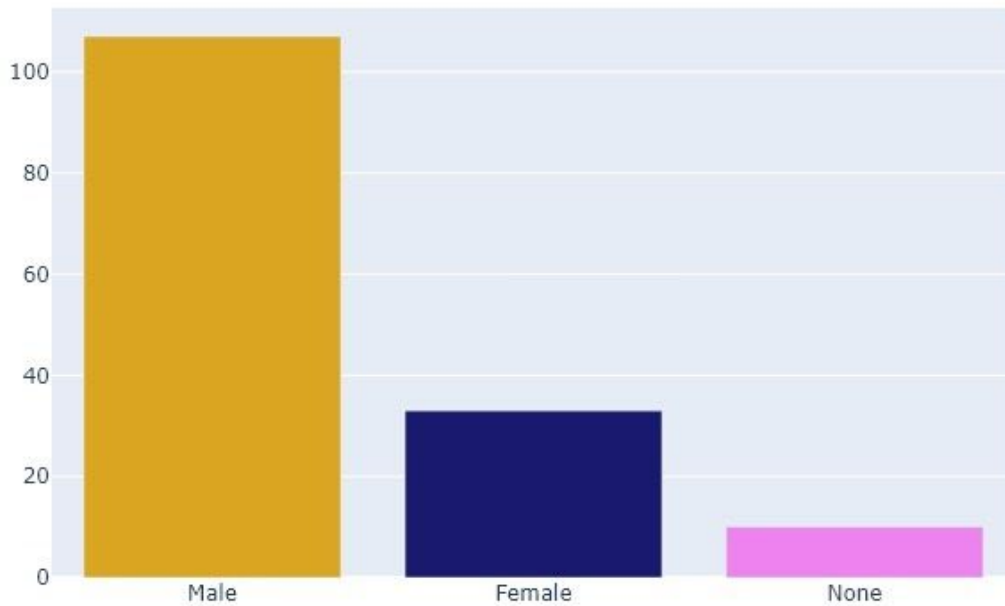
```
{"rock": 3, "pop": 16, "folk": 3, "rnb": 5, "singer-songwriter": 3,
"indie": 3}
```

Number of nonbinary artists of each genre

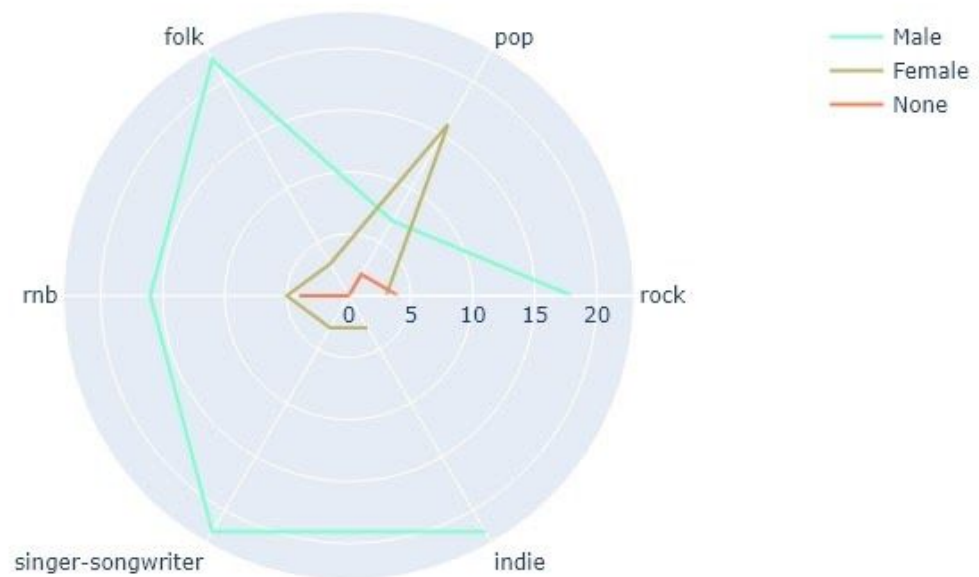
```
{"rock": 4, "pop": 2, "rnb": 4}
```

Visualizations

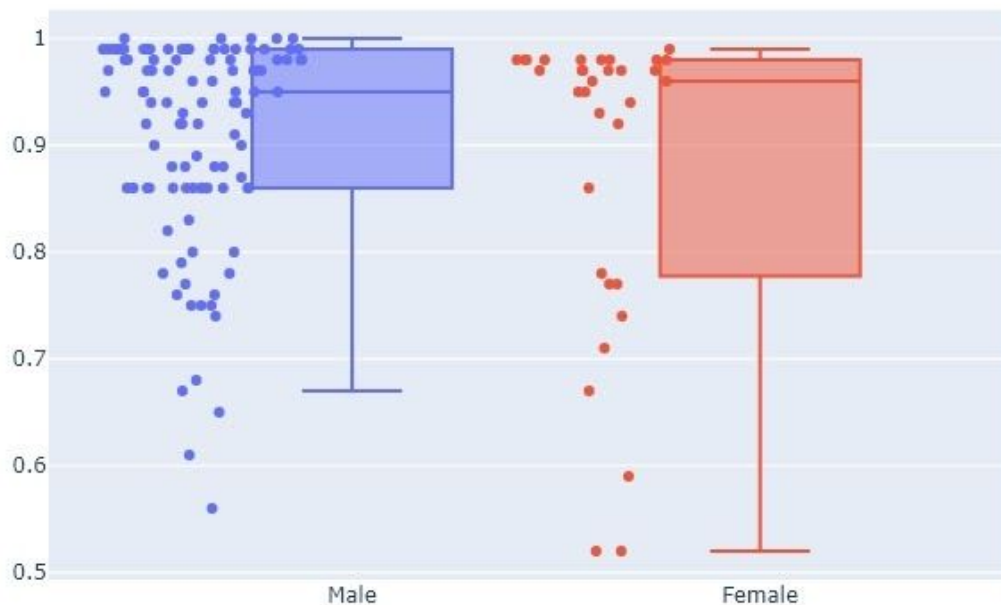
Number of Artists on Top Artist Charts (all genres) by Gender



Number of Top Artists on Top Charts for Genres by Gender



Probability of Genderize API by Gender



Code Instructions

1. If you already have a database on your computer titled 'lastfm.db', delete it. If you have any of the following files, delete them as well: 'results.txt', 'radar_plot.jpeg', 'scatter_plot.jpeg', 'barChart.jpeg.'
2. Install Genderize, Plotly, and Plotly - Orca on your Anaconda terminal:
 - a. `pip install Genderize`
 - b. `pip install plotly`
 - c. `conda install -c plotly plotly-orca`
3. Unzip the folder.
4. Run the first file, 'LastFM_API.py.' six times. Each time, you must change the index of this line: `genre = genres[0]`. This line is noted with a comment in the file. Run this program with indexes 0, 1, 2, 3, 4, and 5. This creates and fills the database with information from the called APIs 25 items at a time, running once for each genre in the list of genres being used, called `genres`.
5. Run the second file, 'calculations.py.' This will gather data from 'lastfm.db' and write it to a file, then construct the visualizations using plotly.
6. View the 3 visualizations. They are jpeg files stored in the folder.
7. You can also view the results file, titled 'resultsdemo.txt.'

Documentation

LastFM_API.py

```
13 """ Requires: db_name (string)
14 Modifies: cur, conn
15 Effects: connects to SQLite database and returns connection and cursor"""
16 > def connectToDatabase(db_name):...
21
22 """ Requires: cur, conn (database connection), genres (list)
23 Modifies: data_dict
24 Effects: Gathers number of artists in database from each genre
25 | | | in the genre list and returns a dictionary with the genres as keys, and counts as values"""
26 > def gatherArtistsPerGenre(cur, conn, genres):...
33
34 """ Requires: cur, conn (database connection)
35 Modifies: data_dict
36 Effects: Gathers number of artists in database organized by gender (F, M, None)
37 | | | in the genre list and returns a dictionary with the genders as keys, and counts as values"""
38 > def gatherArtistsPerGender(cur, conn):...
50
51 """ Requires: cur, conn (database connection)
52 Modifies: male_probs, fem_probs
53 Effects: Gathers the probability of each gender prediction for every artist that was
54 | | | determined to be male or female, as predictions of none have no probability returned. """
55 > def gatherProbabilities(cur, conn):...
69
70 """ Requires: cur, conn (database connection), gender (int), genre (list)
71 Modifies: data_dict
72 Effects: Gathers the number of artists from gender passed, organized by genre. """
73 > def gatherGenderbyGenre(cur, conn, gender, genres):...
79
80 """ Requires: gender_dict
81 Modifies: nothing
82 Effects: Generates a bar graph displaying the number of artists of each gender (Male, Female, None) """
83 > def makeBarChart(gender_dict):...
85
86 """ Requires: ??????
87 Modifies: nothing
88 Effects: Generates a radar plot displaying the number of artists of each gender per genre. """
89 > def makeRadarPlot():...
121
122
123 """ Requires: male_probs, fem_probs
124 Modifies: nothing
125 Effects: Generates a scatter plot displaying the probability of correctness of each gender predictiton,
126 | | | organized by gender(Male, Female, None) with the average probability noted. """
127 > def makeScatterPlot():...
129
130 """ Requires: nothing
131 Modifies: gender_dict, male_probs, fem_probs, fem_count, male_count, none_count
132 Effects: calls connectToDatabase(), gatherArtistsPerGender(), gatherProbabilities(), gatherGenderbyGenre() 3 times.
133 | | | writes results to resultsdemo.txt.
134 | | | calls makeRadarPlot(), makeBarChart(), makeScatterPlot(). """
135 > def main():...
```

calculations.py

```
14
15 """ Requires: string
16 Modifies: split_string
17 Effects: returns first word separated by whitespace in string"""
18 > def get_first_name(string):...
22
23 """ Requires: genre (string)
24 Modifies: artist_list, full_json
25 Effects: returns json object of all artists, gathered by genre from last.fm API, as well as a list of artist names from the same API calls."""
26 > def getArtistsbyGenre(genre):...
45
46
47 """ Requires: artist_list (list of strings)
48 Modifies: first_name_list, gender_data
49 Effects: returns json object of all artists gathered from last.fm API with predicted gender and probability for each artist.
50 | calls get_first_name() to get first names of artists."""
51 > def genderize(artist_list):...
67
68 """ Requires: db_name (string)
69 Modifies: cur, conn
70 Effects: creates SQLite database and returns connection and cursor"""
71 > def setUpDatabase(db_name):...
76
77 """ Requires: cur, conn (database)
78 Modifies: SQLite database
79 Effects: creates Genders table in database and adds rows for each gender category from Genderize."""
80 > def setUpGenderTable(cur, conn):...
91
92 """ Requires: data (json obj), cur, conn (database)
93 Modifies: SQLite database
94 Effects: creates Artists table in database and adds rows for each artist collected from last.fm API."""
95 > def setUpArtistTable(data, cur, conn, i):...
105
106 """ Requires: data (json obj), cur, conn (database)
107 Modifies: SQLite database
108 Effects: creates ArtistGender table in database and adds rows for each artist, their respective predicted gender, and probability of correctness."""
109 > def setUpArtistGenderTable(data, cur, conn):...
119
120 """ Requires: nothing
121 Modifies: genre, name_list, json_data, gender_data, cur, conn, i, lastfm.db
122 Effects: calls getArtistsbyGenre(), genderize(), setUpDatabase(), setUpGenderTable(), setUpArtistTable(), setUpArtistGenderTable()"""
123 > def main():...
```

Resources

Date	Issue Description	Resource Location	Result
11/15	Learning about last.fm user data gathering	https://www.last.fm/api/show/user.getInfo	Determined constraints of initial project plan.
11/15	Learning about Genderize	https://genderize.io/	Determined constraints of initial project plan.
11/29	Looking for easy way to use Genderize for large batches of names	https://pypi.org/project/Genderize/	Use this library to send lists of names into the API rather than appending to a

			link manually.
11/29	Unable to gather user data without authentication	https://www.last.fm/api/show/chart.getTopArtists	Changed plan for data gathering to collecting Artist info.
12/1	TypeError: 'NoneType' object is not subscriptable	https://careerkarma.com/blog/python-typeerror-nonetype-object-is-not-subscriptable/	Fixed bug allowing database additions to be made.
12/1	Repeat data being stored in database	https://ask.xiaolee.net/questions/1030186	Removed repeated artists from database
12/6	Forgot how to write to files.	https://stackoverflow.com/questions/36965507/writing-a-dictionary-to-a-text-file	Reviewed how to write dictionaries to files for calculations report
12/7	Learning how to make radar chart with plotly	https://plotly.com/python/radar-chart/ https://plotly.com/python/polar-chart/	Learned how to create radar chart and about the underlying polar chart documentation
12/8	Learning how to make bar chart with plotly	https://plotly.com/python/bar-charts/#customizing-individual-bar-colors	Learned how to create bar chart and change colors of individual bars for clarity
12/8	Learned how to make scatter plot with plotly	https://plotly.com/python/line-and-scatter/#simple-scatter-plot	Learned how to make scatter plot
12/8	Picking colors that were distinct from each other for different categories of data	https://learnui.design/tools/data-color-picker.html#palette	Used tool to pick distinct colors for charts
12/8	Couldn't figure out why scatter plot was not plotting repeating data points when x values were categorical	https://community.plotly.com/t/best-way-to-create-a-categorical-scatter-plot/11875/2	Created box plots with scatter plot on side showing data points