Assignment 2

CS-UH-2218: Algorithmic Foundations of Data Science

Assignments are to be submitted in groups of two or three*. Upload the solutions on NYU classes as one PDF file for theoretical assignments and separate source code files for each programming assignment. Submit only one copy per group. Clearly mention the participant names on each file you submit.*

Optional Reading: Section 12.4 of the textbook [FoDS] (`https://www.cs.cornell.edu/jeh/book.pdf`) provides an excellent short review of basic probability theory.

Problem 1 (10 points)**.**
Suppose that we want to pick a unit vector with a random direction in two dimensions. One way to do this is to pick an angle $\theta \in [-\pi, \pi)$ (radians) uniformly at random and consider the vector $(\cos\theta, \sin\theta)$. Another way is to pick $a, b \in [-1, 1]$ independently and uniformly at random and consider the vector $\frac{1}{\sqrt{a^2+b^2}}(a, b)$.[1]

Are these two ways equivalent? In order to test this, generate a large number of vectors using the second method and the plot a distribution of their angles with the positive x axis and see if it is uniform on $[-\pi, \pi)$.

You can divide the range $[-\pi, \pi)$ into small intervals of size $\delta > 0$. For each interval, count the fraction of the vectors whose angles with the positive x axis lie in that interval. Then, you can plot a histogram of these counts. To compute the angle of a vector (a, b) with the positive x axis, you can use the function `math.atan2(b,a)`.

Problem 2 (10 points)**.**
A bigram in a text consists of two consecutive words in the same sentence. For example in the text "Hi, how are you? I am fine.", the bigrams are "Hi how" "how are", "are you", "I am" and "am fine". You may assume that sentences end with either a question mark or a full stop. We will treat words with punctuations in them (e.g. "I'll") as single words. We will also ignore capitalization and treat "How are" and "how are" as the same bigram.

Write a Spark program to compute the top hundred bigrams by frequency in the file: `https://www.gutenberg.org/files/100/100-0.txt`. Download and store the text file in the same directory as your program. To make programming easier, ignore bigrams that go across lines.

Problem 3 (20 points)**.**
Go to the website `https://grouplens.org/datasets/movielens/latest/` . Download the file `ml-latest-small.zip` and unzip it into a folder. Read the corresponding `README.html` file in the above website to understand the format of the files contained in the folder. The first line in each csv file is the header and the rest of lines contain data. Remove the first line

[1] If $a = b = 0$, which happens with probability 0, reject and repeat.

from each of the csv files manually to make programming easier.

Write a Spark program to compute the following:

1. The average number of users a movie is rated by.
2. For each genre, the average rating of all movies in that genre.
3. The names of the top three movies (by average user rating) in each genre.
4. Top ten movie watchers ranked by the number of movies they have rated.
5. Top ten pairs of users ranked by the number of movies both of them has watched.

When considering the ratings for a movie, we only consider the users that have rated the movie. So, for example, the average rating for a movie we add all the ratings of the movie and divide by the number of users that have rated the movie.

All the outputs should be to the console. Your program should assume that the csv files are located in the same directory as your program.

Along with your code give brief explanations for the algorithm used for each of the above tasks.