HACKATHON

ROUND 2:

Data Analytics and Compression

Release date: 14 July 2023

Due date: 24 July 2023 - 23:59:59 SAST

(Late submissions will not be considered)

LMN Casino wants to do analytics on the results of their roulette table.

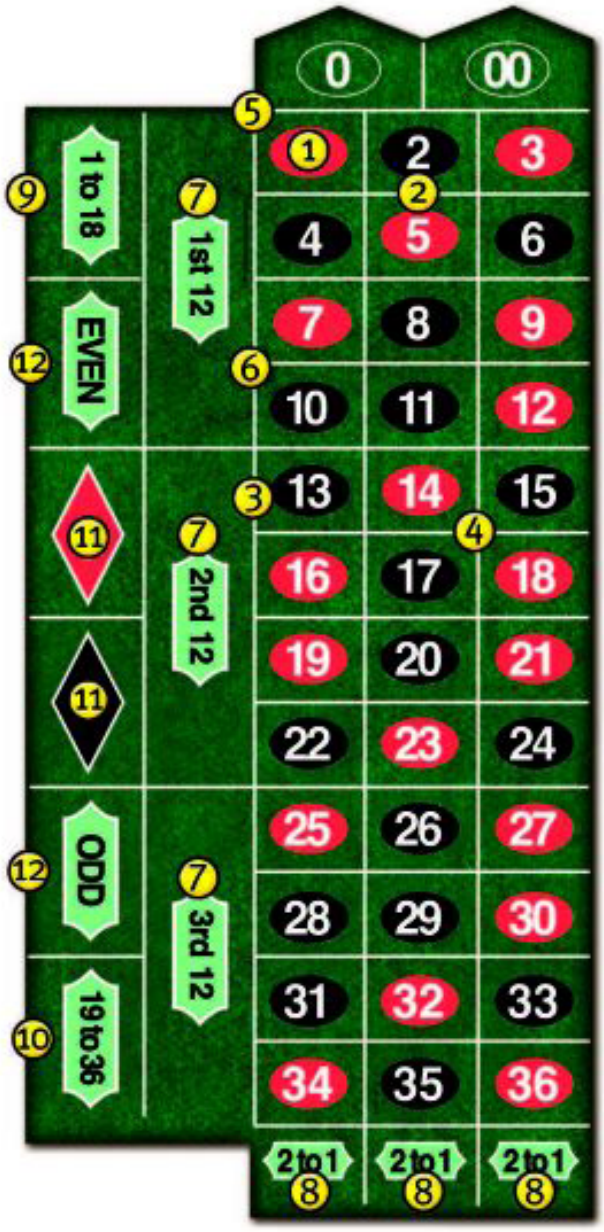
They have asked you to help them generate the analytics and store their data more efficiently.

The casino has provided you with the following information about their roulette table:

How the roulette table works

- The roulette wheel has 38 slots labelled 0, 00 and 1 through 36. The slots 0 and 00 are colored green, 18 of the others are red, and 18 are black.
- The dealer spins the wheel and at the same time rolls a small ball along the wheel in the opposite direction.
- The wheel is carefully balanced so that the ball is equally likely to land in any slot when the wheel slows.
- Bettors can bet on various combinations of numbers and colors, each with different payout amounts.

A breakdown of the various bet types and classifications are shown in the image below:



Key:

1 - 6 not relevant to this project

7
- 1st 12 - numbers 1-12
- 2nd 12 - numbers 13-24
- 3rd 12 - numbers 25-36

8
- 1st Column - numbers 1, 4, 7, ... 31, 34
- 2nd Column - numbers 2, 5, 8, ... 32, 35
- 3rd Column - numbers 3, 6, 9, ... 33, 36

9
- 1-18 - numbers 1-18

10
- 19-36 - numbers 19-36

11
- Red - numbers with red background
- Black - numbers with black background

12
- Even - even numbers (excluding 0 and 00)
- Odd - odd numbers

Their current system is able to track and store the results of each spin of the roulette wheel, with the following information:

- The spin number for the period (1, 2, 3, 4, 5, 6, 7, 8, 9, ..., 999, 1000) - There will always be 1000 spins per period.

- The standard characteristics for the number that won:
  - Red - True or False
  - Black - True or False
  - Even - True or False
  - Odd - True or False
  - 1 to 18 - True or False
  - 19 to 36 - True or False
  - 1st 12 - True or False
  - 2nd 12 - True or False
  - 3rd 12 - True or False
  - 1st Column - True or False
  - 2nd Column - True or False
  - 3rd Column - True or False

Definitions for each of these characteristics can be found in the image on page 2 or online.

- In addition to the standard characteristics above, the casino also stores the following characteristics for the number that won:
  - Prime - True or False
  - Divisible by 4 - True or False - (The casino classifies 0 as divisible by 4, and 00 as not divisible by 4)

Note that the casino does not store the winning number, only the characteristics thereof.

- **Timestamp** - A timestamp for when the spin occurred. (Format = "1970 January 01 00:00:00.000". The time between spins is usually between 60-120 seconds, depending on the number of bettors. The time between spins is not constant).

- **Dealer** - The name of the dealer who spun the wheel. (The name will always be a singular proper noun e.g. "John". Dealers are swapped in denominations of 50 spins).

- **Dealer Tips** - The amount of tips the dealer received from bettors after the spin. (Tips are capped at 500 per bettor per spin. Strictly in denominations of 100).

- **Number of Bettors** - The number of bettors who placed a bet on that spin. (The number of bettors is bound between 1 and 6, and may change between spins).

- **Bets** - The total amount of money bet on that spin. (Bets are bound between 100 and 5000 per bettor per spin. Strictly in denominations of 100).

- **Payouts (incl. Bet)** - The total amount of money paid out to bettors on that spin. (This includes the return of the initial bet. Hence, payouts are bound between 0 and 36 times the total amount bet. Strictly in denominations of 100).

- **Profit** - The total profit (or loss) for the casino on that spin. (Bets - Payouts).

- **Profit per Bettor** - The total profit (or loss) for the casino on that spin divided by the number of bettors. ((Bets - Payouts) / Number of Bettors).

On the Google Drive, you will find a folder for each of the questions below.
Each question folder has a Python Notebook with the name "Question_#.ipynb", where # is the question number. The notebooks act as templates for your solutions, and have demarcated sections for you to write your code in

Link to Google Drive »

The reading and writing of data to files is handled by the template code. If you run the "PERFORMANCE" section of code at the bottom of each template notebook, it will run your logic on the raw data and compare the output to the memo.

You have been supplied with test data for each question, as well as the expected output for the first two questions. Your submissions will be tested on a different set of data, but the format of the data will be the same.

The permitted modules (libraries) for each question are stated at the top of each Python Notebook. Permitted modules will be listed in a file named: "permitted_modules.txt" in the Google Drive folder. If you would like to request access to additional modules, please send your request to 'hackathon@syftanalytics.com'. You may not use any compression libraries for Question 3.

Questions

Question 1 (5)

Given the data above, **write an algorithm** to determine what the winning number was for each spin.

Files named "data_raw_#.txt" containing raw data for 100 spins of the roulette wheel can be found in the "Question_1" folder.

Link to Google Drive »

When you run your algorithm it will generate a file named "winning_numbers_#.txt" containing the winning numbers for each spin.

The expected output can be found in the files named "winning_numbers_memo_#.txt". (These files should not be modified as a part of your solution).

Question 2 (10)

In addition to the raw data, the casino also wants to store the following analytical data for each period:

Number

- Number of spins - The number of spins that occurred in the period
- Number of spins per category - The number of spins that occurred in the period for each of the categories above (e.g. 'Red', 'Black', 'Even', 'Odd', etc.)
- Number of spins per category per dealer - The number of spins that occurred in the period for each of the categories above (e.g. 'Red', 'Black', 'Even', 'Odd', etc.) for each dealer

Percentage

- Percentage of spins per category - The percentage of spins that occurred in the period for each of the categories above (e.g. ‘Red’, ‘Black’, ‘Even’, ‘Odd’, etc.)
- Percentage of spins per category per dealer - The percentage of spins that occurred in the period for each of the categories above (e.g. 'Red', 'Black', 'Even', 'Odd', etc.) for each dealer

Total

- Total number of tips - The total number of tips received by all dealers in the period
- Total tips per dealer - The total number of tips received by each dealer in the period

Average

- Average number of bettors - The average number of bettors per spin in the period
- Average amount bet - The average amount bet per spin in the period
- Average amount bet per bettor - The average amount bet per bettor per spin in the period
- Average amount paid out - The average amount paid out per spin in the period
- Average amount paid out per bettor - The average amount paid out per bettor per spin in the period
- Average profit per spin - The average profit (or loss) per spin in the period
- Average profit per bettor - The average profit (or loss) per bettor per spin in the period

The casino manager is tired of manually calculating all of the analytical data values, as it is time consuming and he sometimes makes mistakes. The manager has asked you to **automate the process of producing the desired data analytics above from the raw data**.

Files named "data_raw_#.txt" containing raw data for 1000 spins of the roulette wheel can be found in the "Question_2" folder.

Link to Google Drive »

When you run your algorithm it will generate a file named "data_analytics_#.txt" which should contain the raw data followed by the analytical data for each period. The expected output can be found in the files named "data_analytics_memo_#.txt". (These files should not be modified as a part of your solution).

Question 3 (35)

The casino manager has fears that storing the data above for each spin will take up too much space, and wants to reduce the amount of data stored for each spin. The manager has asked you to develop algorithms to compress and decompress the data above for each spin.

The manager needs to be able to compress and decompress the data without any loss of information. The manager should be able to compress and/or deompress any of the raw/compressed files in any order on any device using the algorithms you develop. The higher the compression ratio, the happier

Files named "data_raw_####.txt" containing raw data for 1000 spins of the roulette wheel can be found in the "raw_data" folder, inside of the "Question_3" folder.

Link to Google Drive »

When you run the "PERFORMANCE" section of code at the bottom of each template notebook, it will run your logic on the raw data and run your compression and decompression algorithms.
A file named "compressed_data_#.txt" will be created in the "compressed_data" folder, which will contain the compressed output of your compression algorithm.

A file named "decompressed_data_#.txt" will be created in the "decompressed_data" folder, which will contain the decompressed output of your decompression algorithm.

The compression ratio of your compression algorithm will be calculated and displayed in the output of the "PERFORMANCE" section of code, as well as whether your algorithms are lossless or not.

Submission Instructions:

Please upload your Python Notebooks ("Question_#_Surname_Name.ipynb"), to a Google Drive Folder ("Syft Hackathon Round 2 - Surname_Name").

Email the link to your Google Drive Folder to hackathon@syftanalytics.com with the subject line "Syft Hackathon Round 2 Submission - Surname_Name".