Machine Learning Tasks

(Note: All answers would be checked for plagiarism, thus solve the tasks diligently)

Task 1.1

Minion Mashup

Everyone's favorite supervillain (or so he thinks!), Gru, is throwing a victory bash for his loyal minions after their, *ahem*, "semi-successful" moon heist. But with four talented DJs – Kevin, Stuart, Bob, & Edith – each with their own unique musical tastes, Gru needs your help to create the perfect party playlist!

There's a twist, though! Only two DJs can work together to keep the dance floor pumping. To ensure maximum musical mayhem & party satisfaction, there needs to be at least a 30% overlap in the artists they both like.

Here's the challenge:

- We have four **DJs**: Kevin, Stuart, Bob & Edith.
- Each of the DJs has a distinct **set of artists** they like, given as 4 sets:

```
Kevin = {"Halsey", "Taylor Swift", "Mitski" , "Joji",
         "Shawn Mendes", "Sabrina Carpenter" , "Nicky Minaj",
         "Conan Gray", "One Direction", "Justin Bieber"}

Stuart = {"Kendrick Lamar", "Steve Lacy", "Tyler the Creator",
          "Joji", "TheWeeknd", "Coldplay", "Kanye West",
          "Travis Scott", "Frank Ocean", "Brent Faiyaz"}

Bob = {"Conan Gray", "Joji", "Dove Cameron", "Mitski",
       "Arctic Monkeys", "Steve Lacy", "Kendrick Lamar",
       "Isabel LaRosa", "Shawn Mendes", "Coldplay", "Lauv"}

Edith = {"Metallica", "Billie Eilish" , "TheWeeknd" , "Mitski",
         "NF", "Conan Gray", "Kendrick Lamar", "Nicky Minaj",
         "Kanye West", "Coldplay"}
```

Determine all possible DJ pairs that share at least 30% of their preferred artists. Rank these pairs based on the highest percentage of shared artists, providing Gru with the most musically compatible options for the ultimate minion bash!

Remember:

- Only two DJs can work together.
- There must be a **minimum 30%** overlap in the artists they both like.

    - This overlap must be checked **individually**, i.e. the number of common artists for the pair divided by the number of favorite artists of each DJ in the pair should be at least 30%.
- Sort the results based on the DJ pair whose overall overlap is the highest.

Some in-built Python functions recommended:

- combinations(): from itertools
- len()
- zip()
- intersection()
- sort() (**Hint:** look up lambda functions for the key parameter!)

Good luck, & may the music be ever in your favor (and Gru's)!

Task 1.2

Batcave Sorting Protocol

The Riddler Strikes Again! This time, he's scrambled the Batcave's inventory database. Alfred needs your help to decipher it. The database is stored as a series of tuples, each containing:

- product_name(String): The name of the gadget (e.g., Batarang, Grappling Hook)
- quantity(Integer): The number of gadgets currently in stock
- in_stock(Boolean):The flag is True if the gadget is in stock, False if out-of-stock

Here's the scrambled list of gadgets:

```
gadgets = [
    ("Explosive Batarangs", 3, True),
    ("Batarangs", 5, True),
    ("Smoke Pellets", 0, False),
    ("Tear Gas Grenades", 2, True),
    ("Night Vision Goggles", 1, True),
    ("Batclaw", 0, False),
    ("Grapple Gun", 3, True),
    ("Batsignal", 0, False),
    ("Utility Belt", 1, True),
    ("Batmobile Tires", 4, True)
]
```

The Batcomputer needs to sort this data efficiently to ensure Batman has the right tools for the job.

The order is:

- **Top Priority:** In-Stock Items: Gadgets currently available for immediate use should be listed first. Out-of-stock items come next.
- **Within Stocked Status:** For in-stock items, prioritize Quantity. Gadgets with the highest quantity should be listed first, ensuring Batman has enough of the essentials.
- **Breaking Ties:** If multiple items have the same quantity, sort them alphabetically by Gadget Name (A-Z) for easy reference.

You must return the gadgets list, sorted in the right order.

Hint: Use sort() or sorted() functions, and see how key and reverse parameters work.

Can you help the Batcomputer decipher this mess and sort the inventory data? Gotham needs you!

Task 1.3

Balance is Key

When you get selected, ReQuest will be your first event as a co-committee member. Assuming you do ;), We need you to write three functions to help us with the calculation of funds in our adorable admin's account (don't you think she deserves a bit of rest?).

The functions should be:

1. total_spending(request_spending, account_id: str, category: str) should return the total amount spent by the given account, in the given category.
2. account_balance(request_spending, account_id: str) should return the balance of the given account at the end of the event.
3. money_owed(request_spending, account_id: str) should return the amount of money owed to the given account at the end of the event.

You are provided with a dictionary request_spending, containing transaction information for different committee member accounts. Each member is identified by an account_id & has two key-value pairs:

- balance: The current account balance (float).

- transactions: A list of dictionaries representing transactions, where each dictionary has:
    - amount: The transaction amount (float, positive for deposits, negative for withdrawals).
    - category: The transaction category (string).

```
request_spending = {
 "Mahek": {
    "balance": 3000.00,
    "transactions": [
      {"amount": -9000.00, "category": "Creatives"},
      {"amount": 7000.00, "category": "Sponsor"},
      {"amount": -2000.00, "category": "Prize-Money"}
    ]
 },
"Arham": {
    "balance": 5000.00,
    "transactions": [
      {"amount": 8000.00, "category": "Stalls"},
      {"amount": 7500.00, "category": "Seminars"}
    ]
  },
"Unnati": {
    "balance": 3500.00,
    "transactions": [
      {"amount": -5000.00, "category": "Attraction"},
      {"amount": 2500.00, "category": "Promo"},
      {"amount": -900.00, "category": "Lighting"},
      {"amount": -3000.00, "category": "Games"}
    ]
  },
"Gaurang": {
    "balance": 2000.00,
    "transactions": [
      {"amount": -1500.00, "category": "Website"},
      {"amount": -1000.00, "category": "C2C"},
      {"amount": -500.00, "category": "Posters"}
    ]
}
```

Do this well, so our admin has more time to save money for the ~~after party~~ *ahem* next event 👀

Task 1.4

Avatar: Bending the wait time

In the heart of Ba Sing Se, a bustling metropolis where benders & non-benders alike coexist, there stands a humble tea shop, the Jasmine Dragon. The tea shop is renowned for its unique blends & the skill of its tea master, Iroh. Customers from all walks of life flock to the shop, eager to savor Iroh’s creations.

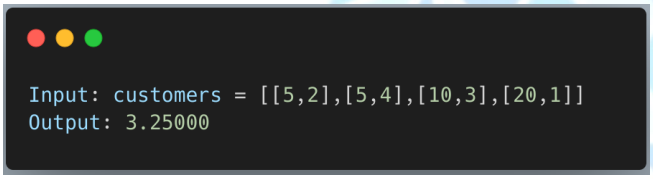
After the war against Ozai, Aang needs help lifting the spirits of Earth Kingdom citizens. As the newest Air Acolyte, you have been asked to help optimize the tea-making process to ensure the smoothest possible experience for the customers.

You are given a scroll detailing the arrival_time of customers, one by one, & the time_required to prepare each tea order.

The customers arrive in a steady stream, each placing their order with you. Since Iroh is the only tea master, he prepares one order at a time, whenever there is a customer waiting for their order.

Your goal is to calculate the average waiting time for all customers. This will help Iroh understand the efficiency of his tea-making process & make necessary adjustments to improve customer satisfaction.

Sample input & output:



1. The first customer arrives at time 5, Iroh takes his order and starts preparing it immediately at time 5, and finishes at time 7, so the waiting time of the first customer is 7 - 5 = 2.
2. The second customer arrives at time 5, Iroh takes his order and starts preparing it at time 7, and finishes at time 11, so the waiting time of the second customer is 11 - 5 = 6.
3. The third customer arrives at time 10, Iroh takes his order and starts preparing it at time 11, and finishes at time 14, so the waiting time of the third customer is 14 - 10 = 4.
4. The fourth customer arrives at time 20, Iroh takes his order and starts preparing it immediately at time 20, and finishes at time 21, so the waiting time of the fourth customer is 21 - 20 = 1.

So the average waiting time = (2 + 6 + 4 + 1) / 4 = 3.25.

Hint: Iterate on the customers, maintaining the time Iroh will finish the previous orders.

Remember, even the smallest improvement in waiting time can bring great joy to the people of Ba Sing Se & honor the spirit of harmony, so keep Iroh informed!

Task Instructions:

- DO TASK 1 IN PYTHON!
- Try to write modular code, divided into meaningful functions.

- Take inputs as parameters for your functions
    - You may take inputs from the user but do run the sample inputs provided and make sure we can see your output.
    - Writing your own test cases is a good idea too!
- Use as many in-built python functions as you need!!

For any doubts related to this task, contact:
Dhvani Thakkar: +91 7738009991
Divyam Jain: +91 9619377044

Task 2

Click the link below and finish task 2 :)
https://www.kaggle.com/code/taranshah/task2-ipynb/

Task 3

For this task, create a word doc or pdf

Task 3.1:

This question is **compulsory**:
Let's say you are given a large amount of textual data- messages, emails, books, etc. Before performing any operations on this data, it is necessary to clean and preprocess the data (removing unnecessary words or symbols, etc.). Explain how you would go about preprocessing. What different steps would be followed? Why are they necessary?

Task 3.2:

At least 1 out of 3:
Imagine using a random prompt like "a cat riding a bicycle on Mars" and seeing an AI generate an image that matches your description perfectly. This is made possible by using advanced models like DALL-E, which use various machine learning techniques, including the diffusion processes. For this task, explain what basic diffusion is, how it works, and why it is used in generating such impressive output.

OR

Have you ever wondered how streaming platforms like Netflix work and how they recommend movies or shows based on your current watch? How does a bank decide which customers get loans and which do not? This all is done using Unsupervised learning. Machine Learning is internally subdivided into different parts- one of them is Unsupervised learning.
The technique used for these kinds of problems is known as Clustering. So, for this task, explain what clustering is and describe any two types of clustering.

OR

You must have heard of or used ChatGPT at some point in the last year, maybe even before. It's like Janet from the Good Place, it always has the answers to your question. Unlike Janet, this may not be 100% correct but does get the job done. But it wasn't always like this. Earlier stages of ChatGPT used to give bogus answers all the time, but over time it learnt how to give appropriate answers, relevant to the context, being as accurate as it possibly can. This is done through a method called Reinforcement Learning. For this task, you must understand and explain the working of reinforcement learning. Additionally, list some other examples and explain how they work.

For any doubts related to this task, contact:
Saumya Mehta: +91 9167370585
Taran Shah: +91 9619223242

Task 4 (Bonus Task)

Click the link below and finish task 4 :)
https://www.kaggle.com/code/atharvmendhe/synapse-ml-interview-task4

Upload your tasks on GitHub in a readable format, like an .ipynb file or a format where we can see your outputs and code. See you in the interviews :))

Contact Us

In case you have any doubts regarding the tasks or the ML interviews, contact any of us. We also encourage you to Google any terms you don't understand. Stack-Overflow is your messiah!!

(Beware: Be honest and clear about your ML Knowledge; interview mai toh pata chal hi jayega ;))

- Dhvani Thakkar: +91 7738009991
- Divyam Jain: +91 9619377044
- Hirali Sangani: +91 91675 74134
- Atharv Mendhe: +91 84549 93809
- Saumya Mehta: +91 9167370585
- Taran Shah: +91 9619223242