Challenge 1:

We have historical prices and data for a product in the csv file :
https://drive.google.com/drive/folders/11MXgRzCHXJfzDqQXUFmOcHlCxu59PDCs?usp=sharing

The explanation of the columns are as follows

Report Date: The date for which this particular datapoint belongs to. This can be handy if you want to include timeseries
Product Price: The price at which the product was sold.
Organic Conversion Percentage: The conversion rate for the product at this date and price
Ad Conversion Percentage: the conversion rate for the ads of this product at this date and price. Usually it should correlate to Organic Conversion Percentage
Total Profit: The total profit made on that particular day for this price point.
Total Sales: The total sales we made on that particular day for this price point.
Predicted Sales: this column is for future dates where we predict how much the product will sell if we have the same price point. The forecasting model will be better if we include this in RL simulation

Challenge:
Write a reinforcement learning module that can

1) predict what should be the ideal price of the product tomorrow.
2) Push the product price higher and provide higher rewards for higher prices at the same time maximizing sales
3) Ensure the model does not get stuck in historical median value. Eg if historically the best price was $16, the RL has to keep pushing the price above $16 by rewarding higher prices as well as higher total sales.

The historical data contains product price and the corresponding conversion rate, ads conversion rate, total profit and total sales.
The two major columns are product price and total sales. The rest of the columns are optional and helps to fine tune the model with better reward and punishment model
The aim of the RL module is to forecast the price that will maximize the sales. The conversion rate and the ads conversion rate can be used as rewards as maximizing the conversion rate will maximize the sales as well. The conversion rates are optional
You can use the predicted sales for tomorrow to predict the forecasted price better.

One way to create the RL would be to have a simulation model where given a price and date the model can use Timeseries forecasting to return the total sales.
Or the RL can just use historical data for simulating the state space.

You have to cater to both exploration (finding new price points that not in history) vs exploitation (finding the best historical price point)
The default tendency for exploration would be to increase the product price to test new price points.

Reward:
The reward for this RL will maximize the Sales, Organic Conversion Percentage as well as Ad Conversion Percentage.
Higher the conversion or sales from median values, higher will be the reward.

Punishment: Any lowering of sales from predicted sales will be punished.

Eg
Product Price: $14, Ideal Price is around $16.5

Input Price: $14, Output Price $14.8
Input Price: $15, Output Price $14.7
Input Price: $15.7, Output Price $16.2
Input Price: $16.2, Output Price $16.6
Input Price: $16.6, Output Price $16.5
Input Price: $16.5, Output Price $16.6

Challenge 2:

Find Matching Jobs

Employer Profiles:
https://docs.google.com/document/d/1ENqNtgBGfeJNrKCx1swQ78msMuqaNVEQfdGv6PwmQds/edit?usp=sharing

Candidate Profiles:

https://docs.google.com/document/d/19vu4dES9WWKqsE4xDzjCYTgkkJdT_wGsaADWmXTlK-A/edit?usp=sharing

Challenge:
There are 5 employers looking for hiring people. They have set specifications in natural language as well as other accessory requirements like location, remote work, salary etc.
Use these fields for ranking candidates: Job Description, Work Location, Salary, Education, Work Experience.
Using the job description and other accessories, rank the Candidate for each job depending on how well their profile matches the job description as well as their accessory requirements like candidate salary, willingness to move etc.
A candidate could possibly not have indicated some of the requirements eg Work Location or willingness to move, current location etc. Handle this accordingly.

Output:
Each employer, followed by all the candidates ranked according to best match.