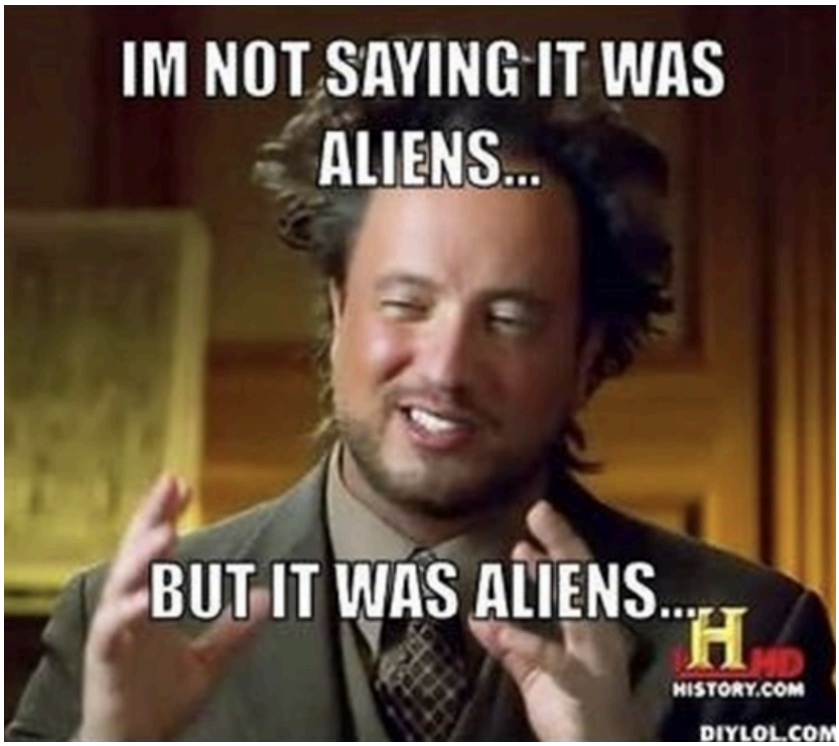
LDSSA Hackathon #6 - DS in the real world

Schedule

Hour	***Activity***
08:00	Arrival, Student setup
08:30	Hackathon Prompt, Team Assignment
09:00	**Starting hacking!**
12:15	**Trial round - test your servers**
12:45	Lunch (no need to stop hacking)
13:45	**First evaluation**
16:15	**Final evaluation**
16:00	Start Working on Presentation
17:00	**Deliver presentation!**
17:30	Team Presentations
18:20	Instructor’s Presentation
18:30	Winners Announced
18:45	Closing Remarks

Overview

The United Galactic Enforcement Division needs assistance to resolve some critical issues. They enforce a protocol across all planetary sectors that mandates officers to conduct inspections only when there is significant justification. However, there have been reports in the galactic media suggesting that certain alien species are subjected to these inspections more frequently than others. They require us to develop and maintain an API endpoint for authorizing these inspections, which will be integrated into a system used by officers across galaxies to ensure uniformity in decision-making. Your task is to establish this system.



Further details are as follows:

- They demand high confidence that when the system advises an inspection, that decision is accurate.
- They seek to eliminate discrimination based on species and reproduction, and will impose penalties for any bias demonstrated by companies.

They need you to create a service with two endpoints:

- Predict: where they will send requests for new recommendations.
- Update: where they will send actual outcomes determining if an inspection was warranted.

The enforcement officers will utilize these endpoints for multiple testing phases and will select the top 10 teams based on a combination of F1-score and adherence to fairness standards. These teams will then be invited to present their services and models.

Assemble your team and prepare for an interstellar challenge. Good luck!

Objective

In this hackathon, you will engage with a binary classification challenge set within an interstellar context. As outlined earlier, this challenge involves predicting, based on various attributes, whether an entity should or should not be inspected. Certain attributes are considered sensitive — such as species type and reproduction — and must be handled carefully to prevent discrimination against minority species.

You will receive an initial dataset containing examples of entities that were inspected and the results of those inspections. **Your task is to develop a model using this dataset**, adhering to the specific demands of the Galactic Enforcement Division:

1. Aim to maximize the accuracy of predictions for the class “**Inspection Outcome.**”

2. Ensure that your model does not exhibit discrepancies in prediction accuracy based on species type or reproduction, with a maximum allowable difference in accuracy of no more than 0.15.

The dataset includes the following fields:

- Age
- Date of inspection
- Inspection type
- Enforcement station
- Reproduction
- Galactic coordinates (X, Y)
- Governing law
- Object of inspection
- Officer-defined species category
- Inspection outcome
- Outcome relevance to the object of inspection
- Whether part of a standard enforcement protocol
- Whether the inspection involved more than just outerwear
- Entity's self-defined species category

The primary field you must use for outcomes is:

- Inspection Outcome

As this model is to be integrated with standard enforcement operations, the field equivalent to "Whether part of a standard enforcement protocol" will not be provided, as it is always assumed true. You are required to set up a server that hosts this model, with two specific endpoints, described below:

Endpoint	Expected payload (input)	Expected response (output)

POST /predict	{
	"observation_id": "8b2de40d-d98b-4cb5-aa49-f471gbja8b",
	"Type": "Entity inspection",
	"Date": "3919-08-16 14:37:00+00:00",
	"Part of a standard enforcement protocol": true,
	"Galactic X": 3434.23,
	"Galactic Y": 2321.12,
	"Reproduction": "Sexual",
	"Age range": "Young",
	"Self-defined species category": "Terran - Northern",
	"Officer-defined species category": "Terran",
	"Governing law": "Intergalactic Substance Regulation 3919",
	"Object of inspection": "Controlled substances",
	"Inspection involving more than just outerwear": false,
	"Enforcement station": "Dyson Sphere F76-JK"
	}
		{
		"observation_id": "8b2de40d-d98b-4cb5",
		"label": true
		}

POST /update	{
	"observation_id": "8b2de40d-d98b-4cb5",
	"label": true
	}
		{
		"observation_id": "8b2de40d-d98b-4cb5",
		"label": true
		}

If your application receives a request that doesn't conform to the specified formats, it should either return an appropriate error message or disregard the request. Importantly, your system must be robust and fault-tolerant, meaning **it shouldn't crash due to an anomalous request!**

A preliminary evaluation phase will take place, during which several test cases will be sent to verify if your endpoints function correctly. By this stage, you should have already provided your endpoint details so they can be accessed for queries. Soon after, **the first round of evaluations will begin**, where your server will be queried for predictions. The outcomes will be calculated based on the initial metric and displayed on a leaderboard, allowing you to gauge your performance against other competing teams.

Following the predictions, a round of updates will be communicated back to your update endpoint, which you should capture and use as feedback to refine your model in preparation for the ultimate round of predictions. If you happen to fail in storing the update data, it will be forwarded to you in a CSV format, but be aware that there will be penalties for not maintaining your server adequately. After retraining and reassessing

your model, you will enter the final evaluation stage. **During this period, your server will face a series of prediction requests, culminating in a final score**.

To summarize:

- Develop a model to address the outlined challenge.
- Implement your model and set up a `predict` endpoint.
- Establish an `update` endpoint to receive true values.
- Retrain and redeploy your model for the final evaluation.

Evaluation criteria for your model

You will be assessed based on several metrics:

Galactic Enforcement Metric (GEM) - Metric used by the enforcement division to select the top 10 teams: F1-score (Galactic_Enforcement_Metric).

Fairness Penalty - Difference in percentage points from the client's fairness standard (see below). For example, if the largest discrepancy between reproduction (`Reproduction`) or species groups (`Self-defined species category`) is 0.20 and the client's requirement is 0.15, you'll be penalized by 0.05 (0.20-0.15). If you meet the requirement, fairness_penalty=0.

Fairness Requirement provided by the client: "The model should not exhibit a metric variance exceeding 0.15 between different reproduction or species groups."

*Hacking_score = GEM - 0.1 * Fairness_penalty*

Server Reliability

Your predictions will only be processed if your server is stable and responds to all requests. While there is no explicit metric for reliability, your overall performance could be affected by it.

Presentation

Following the competition, you will present your work to the Galactic Council. Although you may utilize foundational elements from previous hackathons, there is an additional component for this presentation.

We expect you to highlight the key aspects of your service and the client's demands, crafting a concise presentation to demonstrate the value of your work. Choose the elements you discuss wisely. The final ranking will be determined by the following criteria:

$$FinalRank = 0.5 * Hacking\_rank + 0.5 * Presentation\_rank$$

Where:

- ***Hacking_rank*** is the rank of your team in the leaderboard, considering the scores given by **Hacking_score**
- ***Presentation_rank*** is the rank of your team in the presentation evaluation

Recommendations

❗ Establish a simple baseline for the task and make sure you assess not only the leaderboard metric but all the client requirements and aspects
❗ Setup your service and make sure it is functioning correctly (both endpoints!)
❗ Take advantage of the trial run to verify the correct functioning

- You should check your app logs to make sure there were no errors
- You can use the dummy score on the leaderboard to make sure your scores were delivered correctly.
- Remember: these will be dummy examples, so the scores themselves do not necessarily reflect the quality of the model

❗ In case you submitted some requests to tests your app and you see issues with the database (eg. errors saving new requests) reset the database by removing the service and de-linking it to your app (we still haven't figured out a better way to do this in Railway). **Do this only before the first official evaluation moment.**
❗ Distribute work by team members: Maybe one is responsible for improving the robustness of the server, while others attempt to increase the performance of your model
❗ Remember: More data doesn't always mean better results.
❗ Remember: More complex models don't always yield better results.
❗ The presentation can take a **maximum of 4 minutes**. This is a hard limit! We'll literally silence you and move on to the next group after the 4 minutes have passed.

❗ The team can decide who is presenting. There are no rules here, you can go with one person presenting everything or having everyone presenting a part

Hackathon Rules

- The selection of the teams is **random**.
- Instructors will be available to help at any time. The instructors will **not** help your team solve the challenge but they will help your team to be on track and answer technical questions that your team might have.
- **No more submissions and questions** to the instructors shall be done after the end of the challenge.
- Your team will have to prepare a presentation to share your findings with everyone. This presentation will be considered in the overall evaluation of your team, so don't consider it less important than the ML model and service!
- There will be only 2 moments of evaluation on this hackathon and they will happen at scheduled times! So make sure you have everything set up by then. Failing to do so will result in the minimum score.

Feel free to ask any questions!