APPLICATION ASSESSMENT

This assessment is designed to assess your Data Science skills in various areas to solve a practical problem through a case study. It is intended to take approximately **3 hours** to complete.

Case Study Description

You are assigned to work as a Data Scientist to collaborate on a business-critical R&D project. The project is concerned with modelling fraudulent behaviour so that such behaviour can be detected. The aims of this project are to prevent frauds, and to prevent frictions to legitimate customers. Preventing fraud is about correctly identifying cases of fraud, and preventing frictions is about not identifying good users as fraudsters. The business has established a target for fraud prevention, where your colleagues in the Marketing team have extracted information from various clients of a 70-90% fraud recall rate. Meanwhile, the client in question is also extremely keen to reduce frictions to a value as close to 0 as possible. A good solution is able to perform both tasks effectively and to find an appropriate trade-off between both.

Data

A client has provided cases of confirmed fraud and as well as normal traffic made by users to their website. Each case is a "request" that was made to their website. The following files are included:

requests.csv

This dataset contains one row per request. The following fields are included:

- **count_feat_0 – count_feat_17**: Count-related features computed in our feature store. Might include counts, differences, or ratios between counts. Examples include the number of times the user has successfully logged-in the past 7 days.
- **anomaly_feat_0 – anomaly_feat_3:** These indicate if our system has detected several types of anomalies in the request.
- **interaction_feat_0 – interaction_feat_3:** These describe user interaction with the interface during the request
- **accounted:** Account ID of the user (anonymized)
- **is_attack**: Indicates if the request was fraudulent or not
- **timestamp:** Time stamp of the request represented as an integer (for anonymization purposes). The ordering of the requests has been preserved.

device_info.csv

This dataset contains device information. The following fields are included:

- **accounted:** Account ID of the user (anonymized)
- **device_info_1 – device_info_2**: Information about the user's device.

Task

You are tasked to build a Machine Learning model that can solve for the R&D project described above. Your approach should at least (but not limited to) address the following aspects:

1. Are there transformation / data-cleaning steps required?
2. What are the key driving features?
3. How is the model selected?
4. How is model performance evaluated?
5. How is the trade-off between preventing frauds and reducing frictions considered?

Deliverables

1. Please provide the code and analysis you used to address the task. You may choose to submit a single Jupyter notebook.

2. Please write a 1-page report summarizing your approach and results. The report should be clear, well-structured including justifications for your choices and any potential limitations or challenges you foresee. Your "audience" would be a selection of non-technical Product Managers, Data Scientists and Machine Learning Engineers.

3. **(Bonus 1)** Your coworkers are impressed with your report and would like to have this model deployed in production in a real-time fashion, so the client can receive the prediction and make decision about **the request in real time.** What other considerations and/or code changes would you make prior to deploying your model to a production environment? Below are some considerations (but not limited to) you might address in your answer:
    - In what ways is/isn't this code ready for production deployment?
    - What approach and technology would you use for deployment?
    - When deployed, how do we know whether the solution is working effectively? What metrics might we want to track to confirm this?
    - How would we update and maintain the model?

4. **(Bonus 2)** You found out from the client that the fraud dataset is incomplete, meaning that some of the normal traffic in fact includes fraudulent requests (but not reflected in the **is_attack** target column). However, the client is unable to identify which requests exactly are fraudulent. How would you modify your solution to address this additional information?
    - How might this impact the results you presented previously?
    - Will you still be able to use the dataset?
    - What machine learning approaches can help deal with the incomplete fraudulent information?