**Data Scientist Hiring Case Study**

**Note**: Candidates are advised not to use AI tools to generate code. Our AI team can distinguish between AI-generated and human-written code. If an AI tool (e.g., ChatGPT) is used, a disclaimer must be provided. Case studies will be evaluated accordingly, and failure to disclose AI-generated code may result in disqualification of the candidature.

**Challenge:**

Please work through the problems and answer the questions. Be prepared to share your thoughts supported by your desired instruments (presentation, whiteboard sketches, etc.) We will ask some additional questions based on your presentation.

**Timeline:**

* 72 hours preparation: case study sent in advance
* 15-minute presentation
* 45-minute Q&A

**Language**

* English/Python

**Problem**

**Hands-on mini projects (Can provide Colab notebook or a .ipynb file)**

1. Create a DL training pipeline to train a LSTM/GRU (any one) and one Transformers network to identify Sentiments in a sentence. You are free to choose any open source dataset (like IMDB reviews or Amazon product ratings or any other). Report your accuracy. Create an inference pipeline which can accept a user text and provide score to each sentiment on it.
2. Create a DL training pipeline to train a ViT/ResNet model (feel free to choose any) on any of the open source datasets like MNIST, COCO, ImageNet or any other dataset of your choice. Ofcourse the model training features will depend on the dataset that you will be choosing. Report the results. Also create a simple inference pipeline so that a manually downloaded data point from each class of the trained model can be used to infer the class of the image. For example, if you are training on MNIST data, download 10 image of numbers 0-9 (or maybe just write it on a piece of paper) and feed to the model and report the results.
3. Create a DL training pipeline to train a simple transcription (Audio to Text) model. You are free to choose any easy dataset from huggingface or any source of your choice. Report the results. Create an inference pipeline to provide transcription on any audio/podcast/anything else downloaded from internet

**Instructions:**

Network should be created from scratch. A simple network would suffice which can achieve an accuracy of 0.5 or more. Please feel free to assume any details required to do this. We will be judging on

1. Coding Practices- Clean Code and Reusable code
2. Efficient Network building using any library of your choice; torch/tf/any other
3. Intuitive Training loop and Inference pipeline
4. Result report should talk about change in loss and discuss any other observations