Hello,

The goal of this exercise is to help us understand your approach towards and experience with building ML solutions. There are several problems in the FinTech domain and we've chosen churn prediction as a case-study for you. You will find all the necessary information below. In case of any doubts, please write back to the concerned HR or Hiring manager and give 24 hours for response.

Problem statement:

Customer churn prediction is crucial in the FinTech domain. Your task is to build a robust pipeline addressing data extraction, model development, and system design. Below are the details.

Data:

1. Data Source: Telco Customer Churn Dataset
2. Time Feature: Add a new timestamp feature at a daily or hourly level frequency depending on how you plan to use time related features.

Hypothesis Building:

State a concise hypothesis connecting features to potential influences on customer churn. If you are evaluating more than one hypothesis then build separate pipelines for them and showcase your understanding of pipelines by re-using components.

Standard EDA:

1. Covariance and Correlation Matrix: Display matrices.
2. Data Quality Check: Handle missing values, outliers, etc.

Feature Engineering and Reduction:

- Max 6 raw/derived features, use suitable techniques.

Model Evaluation Metrics:

- Decide on metrics, connect to business KPIs.

Model Development:

- Model Creation:
    - Develop two models for comparison (e.g., Logistic Regression, Random Forest).
    - Implement hyperparameter tuning for one model.
- Airflow/Kubeflow Integration:
    - Create an Airflow pipeline for data processing and model training.
    - Use Kubeflow for managing the ML workflow on a simulated cluster.
- Container Deployment:
    - Containerize the model using Docker.
    - Deploy the containerized model on a local cluster with minikube and kind.

Model Deployment Plan and Architecture Design:

- Create a working solution and share recorded video or screenshots
- Highlight components for model serving, monitoring, logging, and iteration/updation.

Success Metrics:

- Model accuracy on the test dataset > 70%.

Bonus Points:

- Packaging:
    - Include a README for installation and execution of the end-to-end pipeline.

- CI/CD:
    - Implement CI using Github Actions.
    - Implement CD if you are deploy on cloud with ECR/EC2/EKS

- Post Model-Serving Stages:
    - Present a strategic roll-out and A/B/N testing plan. And demo at least one.
    - Explain how would you handle drift detection with your pipelines

- Documentation Skills:
    - Showcase the documentation's value to the organization.

- Version Control
    - For code
    - For artifacts (model, data, pipelines)

Deliverables (In a Zip File):

- Report (PDF):
    - Pipeline description and design choices.
    - Model performance evaluation.
    - Scaling up the pipeline discussion.
    - Future work discussion.

- Source Code:
    - Working code
- Video/Screenshots
    - Video or screenshots of a working solution. Preferably use a video and pitch yourself in 5 mins explaining the solution and working components

Timeline:

- One week for submission. Contact HR/HM for extensions.

Note:

While we are not expecting everything to be covered, ensure the solution is sufficiently complete for evaluation on coding standards, ML, documentation, and system design.