Part 1: Short Answer Questions (30 points)

1. **Problem Definition (6 points)**

Hello! I would definitely be able to formulate a hypothetical AI problem, enumerate its goals and stakeholders, and suggest a success KPI. Problem Statement: AI-Powered Personalized Learning Path Generator has defined the hypothetical AI problem as such: create an AI system that will automatically generate individualized learning pathways to K-12 students depending on their own performance, learning styles, and cognitive load capacity. The following goals in this AI system are: Maximize Student Proficiency: To go up by at least 15 percent, on average, in the subject proficiency scores of a student using the personalized path, as compared with students on a standardized path, within a single academic semester. Optimize Learning Efficiency: to decrease by 20 percent, on average, the total time a student spends studying on remedial or already mastered material, by dynamically skipping or speeding up learning content, to reach the desired outcome of increasing scores on average. Enh Stakeholders two principal stakeholders of this project are: Students (The Users): Students are the direct beneficiaries of the personalized and optimized learning process, which, hopefully, will result in better grades, less frustration, and confidence to trust their own abilities to solve specific problems.Teachers/Educators: The teachers/educators are the direct users of the system to receive detailed student progress information and effective and automatically adjusted instruction materials, which will free them to concentrate on high-touch interactions and critical problem-solving. Key Performance Indicator (KPI)My proposed Key Performance Indicator that will be used to measure the success of the AI system is: Average Time-to-Mastery (TTM) Reduction: This KPI is a measure of the average percentage percentage change in total time (measured in minutes/hours on the platform) required to reach a pre-determined level of what could be termed success or mastery (e.g., 90 percent on a unit assessment) relative to a control group taking the standard, non-AI path. Success is defined as a 10 percent It means that the AI is effectively streamlining the path by eliminating redundant material and concentrating on the most effective learning flow.

2. Data Collection & Preprocessing (8 points)

Preprocessing and Collection of Data.

Data Sources

Two major sources of data that I would use are:

Learning Management System (LMS) Interaction Logs: It is the most important source of data, which includes behavioral and performance information.

Information Collected: Student evaluation scores (quizzes, tests, homework), time spent on a specific content item (video, article, practice problem), the number of attempts before passing, order of content viewing, and clickstream (e.g., watching the same video again).

Student Demographic/Psychometric Records: It is information that furnishes the personal context of the individual.

Information Collected: Academic history of student (past grades, standardized tests), self-reported learning style (ex, visual, auditory, kinesthetic) based on initial intake survey, and possibly cognitive load proxy (e.g., time of day he/she studies, length of learning session).

Potential Data Bias

One of the major possible biases of the gathered data is Sampling and Engagement Bias.

Justification: The information would mainly be gathered through the already enrolled and actively using students of the digital learning site.

Highly successful and motivated students can utilize the platform on a regular basis and record valuable data that is rich and constructive.

Students with struggling, disengaged, or poor internet/device will produce little, partial, or negative data (i.e., lots of missing activity logs, low scores).

Impact: When a model is trained on a preponderance of the data of high engagement or particular demographic groups, the personalized paths will not work well with the marginalized or low-engagement students; in fact, it might only strengthen the existing educational disparities instead of alleviating them.

[?] Preprocessing Steps

To train the model, I would consider the following three major preprocessing steps required to prepare raw data:

Feature Engineering and Imputation of Missing Data:

Purpose: Establish meaningful features and deal with missingness.

Action: In case of performance data, missing scores of an assessment (e.g., student missed a quiz) would be filled with a conservative estimate (assuming the situation), such as the lowest current score of the student or the average of the class. I would also program new attributes such as Pace Index (score/time spent) and Attempt-to-Mastery Count of each topic.

Normalization/ Temporal and Categorical Encoding:

Purpose: Transform mixed types of data into a machine learning algorithm format.

Action: Categorical data that include Learning Style (Visual, Auditory) and Subject Area would be coded by One-Hot Encoding. Scaling of numerical characteristics, such as Time Spent and Assessment Score, would be done through Min-Max normalization (scaling the values to a range of 0-1) to avoid those features that had larger values dominating the model training process.

Attending to Time/Sequential Data:

Purpose: To maintain the order and time sequence of the data, which is essential in the learning paths.

Action: The series of interactions of content and performance would be organized as learning sequences for the students. This can include sequence padding, truncation to equal length in order to fit sequence models (such as RNNs or Transformers), or the sequence itself can be transformed into state-based features (e.g., current performance delta of the last two topics).

3. Model Development (8 points)

That's the core of the project! In the case of the AI-Powered Personalized Learning Path Generator, I would select a particular category of Neural Network and define the development process.

Model Development

1[?] Model Choice and Justification.

My model of choice is a Recurrent Neural Network (RNN) that includes Long Short-Term Memory (LSTM) units.

Alternative: Recurrent Neural Network based on LSTM.

Justification: The sequence-to-sequence or sequence-based prediction problem is inherent within the generation of a personalized learning path problem.

Sequential Data: Student learning is a time series; the result of the current action of a student (i.e., spending 10 minutes on Topic A) is directly related to his/her preparation for the next topic (Topic B). LSTMs are good at learning these long-term sequential dependencies in sequential data, which feed-forward networks are unable to.

Prediction Task: The model is expected to forecast the best following content item or the likelihood of mastery according to the whole history sequence of content intake, time intake, and the mastery marks. LSTMs are ideal for the modeling of this student learning path.

2[?] Data Splitting Strategy

The splitting approach has to take into consideration the type of data (student learning sequences). I would apply a Group-Based Stratified Split so that the information of one specific student does not reflect on more than one set.

Training Set (70%): It is a set that is used to learn to train the LSTM model so that it can learn the complex patterns and dependencies between various learning actions and outcomes.

Validation Set (15%): This one is applied in the process of model training to optimize the hyperparameters (as provided below) and avoid overfitting. Each time the model completes an epoch, I would observe the performance (e.g., loss on validation) of the model on this set.

Test Set (15%): This is the set that is stored apart and is not touched till the last model is chosen. It gives a fair assessment of the generalization capability of the final model before deployment.

Crucial Grouping: Grouping would be done based on the Student ID level. This implies that all sequential data of Student A are in the training set, and all the data of Student B are in the test set, and so on. This stops data leakage, in which the model may be trained to know who one of the students is and which specific pattern they perform, and artificially inflate performance measures.

3[?] Hyperparameters for Tuning

Two important hyperparameters of the LSTM model that I would concentrate on would be:

Learning Rate:

Why Tune: The learning rate is the increment with which the optimization algorithm (such as Adam or SGD) changes the weights of the model in response to the error calculated.

Impact: In the case of a high learning rate, the model may skip over to the optimal solution (minimum loss). Long training time and the model will be trapped at an inadequate local minimum. By tuning it, it is guaranteed to have an efficient and effective convergence.

Number of LSTMs Layers and Hidden Units (Complexity of the Model):

Why Tune: This is the capacity of the neural network. The model is determined by the number of layers and the size of the hidden units (the number of neurons in each layer) of the pattern the model can learn.

* Impact: A small network will fail to identify the non-linear effects in student learning behavior (underfitting). An overfitting network will not only memorize the training data (overfitting), but it will do well on the training set, but badly on new students (validation/test sets). This gives the generalization sweet spot, which is found by tuning.

4. Evaluation & Deployment (8 points)

Evaluation: The two metrics that I would use to measure the performance of the LSTM model are: Prediction Accuracy (Next Step Recommendation):Metric: Top-N Accuracy (e.g., Top-3 Accuracy). This is the percentage of times the model can include the best next content item (the next item the student would want to study to ensure they master the content efficiently) in its top N items.Relevance: This is a direct metric of the utility of the model. When the AI suggests incorrect content, it will be a waste of time of the student, and it will not be a well-optimized path indeed. Root Mean Square Error (RMSE) (of Mastery Prediction):Metric: RMSE is the average size of the distance between what the model predicts the student to score on and what the student actually scores on.Relevance: The fundamental role of the system is to make sure the student is at a level of learning to move to the next topic. RMSE provides a real, understandable evaluation of the capability of the model in predicting student preparedness. The smaller the RMSE, the closer the forecast of the model to the actual level of proficiency of a student, enabling better pacing and path choices. Monitoring Concept drift Concept drift is a phenomenon in which the statistical characteristics of the target variable (the thing we are trying to predict, e.g., the best next action, or the level of student mastery) change over time in ways not reflected by the original training data. Relevance to this Problem: Changes in the school curriculum, new standards, or changes in student learning behavior (going to higher dependence on some outside resource) may cause the previously existing relationship between features (time spent, past scores) and the target (future mastery) to become invalid. Monitoring Post-Deployment I would establish, in case the Top-3 Accuracy decreases to a pre-determined limit (e.g., below $80\%) over a certain duration, this would indicate possible drift and issue an alert. Input Data Distribution (Confirmation): When the performance decreases, I would compare the distribution of the important input features (e.g., average time-per-session, average assessment score, frequency of content skipping) of the newest data to the distribution of the original training data. The difference between input patterns is statistically significant (e.g., confirmed by the Kolmogorov-Smirnov test), which means that the input patterns have altered and the model needs to be retrained or recalibrated 3[?]. Technical Deployment Challenge: The key technical obstacles to the deployment of this LSTM-based recommender are Low-Latency Inference and Scalability. The Challenge: Generating a personalized path would require the model to make real-time predictions each time a student finishes a piece of content or pauses an activity. As LSTMs are complex, sequential models, it may be computationally expensive and slow to calculate the best next step to serve millions of students at the same time, which is undesirable since slow response time does not favor user experience. Impact: A high latency service (high latency) implies that it may take seconds to serve millions of students with the next recommended activity, which is unfavorable since high latency does not complement user experience. Mitigation Strategy: I would use an optimized serving implementation (such as TensorFlow Serving or ONNX Runtime) and hardware acceleration (GP Importantly, I would use Micro-batching and Pre-computation- predictions could be made overnight when dealing with less personalized or more static sections of the path, but the real-time model is only used to make the immediate, highly personalized transitions.