**Suggestions**

**1. Feasibility**

* **LSTM + XGBoost hybrid model** is capable of handling **large datasets**, including 100,000+ records.
* **Reason:**
  + LSTM handles sequential/temporal patterns per member (30/60/90-day windows).
  + XGBoost efficiently handles tabular data and scales well with many rows.
* **Typical constraints:** Memory and processing time, especially if sequences are long or features are numerous.

**2. Prototype Considerations**

* **Training:**
  + Training on 100,000+ records may be **slow on a local machine**.
  + For a prototype:
    - Use **subset sampling** (e.g., 10–20% of data) to quickly demonstrate functionality.
    - Optionally, **pretrain on synthetic data**, then fine-tune on smaller real datasets.
* **Prediction:**
  + Once the model is trained, **predicting risk scores for 100k members** is much faster than training.
  + You can batch predictions to avoid memory issues.

**3. Optimization Tips**

1. **Feature Selection:**
   * Reduce input features to only the most relevant to risk prediction.
   * Helps LSTM train faster and reduces memory usage.
2. **Batching & Windowing:**
   * Process sequences in **mini-batches** to avoid GPU/CPU overload.
3. **Use Efficient Libraries:**
   * **TensorFlow/Keras or PyTorch** for LSTM — support GPU acceleration.
   * **XGBoost with DMatrix** — optimized for large datasets.
4. **Incremental Retraining:**
   * Instead of retraining on the entire 100k records every time, use **incremental learning** or **fine-tuning** on new data subsets.

**4. Recommended Approach for Prototype**

* **Training:** Use **smaller sample** (~10k–20k records) for fast experimentation.
* **Testing/Prediction:** Use the full 100k dataset to **show the model can scale** and generate risk tiers.
* **Visualization:** Paginate or filter dashboard views for large datasets to avoid UI slowdown.