**GPU SPACE OPTIMISATION**

*These are some of the possible ways that can help to reduce the shape and size of the model so that the space it takes in GPU is optimized.*

1. **Quantization-**

One way to reduce the size the model takes in GPU is to serialize models into a JSON format with reduced precision. XGBoost’s experimental JSON serialisation allows for quantization, which reduces the precision of floating-point numbers in model parameters.

1. **Batch processing-**

Instead of making predictions for each input individually, batch multiple inputs together to reduce the overhead associated with model prediction function calls. This is particularly effective when handling multiple predictions simultaneously.

1. **Efficient Data Handling-**

Using Streamlit's caching capabilities (@st.cache) to store and reuse computed results (e.g., pre-processed data, model predictions) across sessions can help us avoid redundant computations and speed up app responsiveness.

Another way to optimize the space the model takes in GPU is to deploy it using frameworks that are fast and needs little memory. Following are some other frameworks that can be used-

1. Fast API – It is known for its high performance and low overhead,

making it efficient in terms of CPU and memory usage.

1. Flask- It is a lightweight WSGI web application framework in Python.
2. Django- It is a high-level Python web framework that encourages rapid development and clean, pragmatic design. It's suitable for larger and more complex applications.
3. Serverless Deployments- Serverless computing platforms like AWS Lambda, Google Coud Functions) allow you to deploy functions (such as model prediction functions) without managing the underlying infrastructure.