**Partitioning Techniques in AI Models for Wireless Network Optimization**

**Objective**

The goal of this assignment is to explore, implement, and analyze **vertical and horizontal partitioning** techniques for AI models, applying them to a real-world problem of **network latency prediction** in wireless communication systems. You will use a dataset representing tower and user attributes to study how partitioning impacts performance, scalability, and deployment.

Dataset: <partitioning_dataset.xlsx>

**Part 1: Conceptual Understanding (10 Points)**

**Instructions:**

1. **Define the following concepts in the context of AI model design**:
   * **Vertical Partitioning**: Splitting features (input variables) among sub-models.
   * **Horizontal Partitioning**: Splitting the dataset into subsets based on sample characteristics.
2. **Compare the two partitioning approaches** across the following dimensions:
   * **Computational Efficiency**
   * **Scalability**
   * **Deployment Considerations in Wireless Networks**
3. **Support your discussion with examples**, ideally referencing wireless network datasets.

**Part 2: Practical Implementation (60 Points)**

You are provided with a pre-trained model designed to **predict network latency**, along with a dataset containing the following attributes:

* Tower ID
* Signal Strength (dBm)
* Network Traffic (MB)
* Latency (ms)
* User Count
* Device Type

**A. Vertical Partitioning (30 Points)**

**Tasks:**

* Divide the input features into two logical groups:
  + **Model A**: Features related to network infrastructure (e.g., Signal Strength, Network Traffic).
  + **Model B**: Features related to user behavior (e.g., User Count, Device Type).
* Build two separate sub-models and **fuse their outputs** to predict **Latency**.
* **Compare performance** with the original monolithic model using metrics such as:
  + Mean Absolute Error (MAE)
  + Root Mean Squared Error (RMSE)
  + R² Score

**Deliverables:**

* Python script or Jupyter Notebook implementing vertical partitioning.
* Performance evaluation table and brief interpretation.

**B. Horizontal Partitioning (30 Points)**

**Tasks:**

* Categorize your dataset into two geographical groups:
  + **Subset 1**: Urban cell towers
  + **Subset 2**: Rural cell towers  
    *(If location data is unavailable, simulate this by manually tagging towers.)*
* Train a separate model for each subset.
* Compare the performance of these **specialized models** with a **global model** trained on the full dataset.

**Deliverables:**

* Python script or Jupyter Notebook implementing horizontal partitioning.
* Performance comparison and summary of findings.

**Part 3: Strategic Analysis and Recommendations (30 Points)**

**Tasks:**

* Based on your results in Parts 1 and 2, provide **real-world recommendations**:
  + In what scenarios is **vertical partitioning** more beneficial?
  + When is **horizontal partitioning** preferable?
* Imagine you are presenting this to a telecom company (e.g., **Qualcomm**). In a report, explain:
  + How these techniques could enhance **5G network optimization**, edge AI, or mobile infrastructure.
  + Potential **deployment challenges** (e.g., data integration, latency trade-offs).
  + Suggested **solutions and design strategies**.

**Submission Guidelines**

| **Component** | **Format** | **Requirements** |
| --- | --- | --- |
| **Part 1** | Word | Theory section |
| **Part 2** | .ipynb | Python Notebooks with code and output |
| **Part 3** | PDF | Theory section |

**Evaluation Criteria**

| **Criteria** | **Weight** |
| --- | --- |
| Conceptual Clarity | 20% |
| Code Quality and Implementation | 40% |
| Analysis and Interpretation | 30% |
| Professional Presentation | 10% |