DML Assignment – 1 (Part-1)

Group Number 15. (Members):

1. **Aishwary Shukla (2023AA05448)**
2. **Hemant Kumar Parakh (2023AA05741)**
3. **Sushil Kumar (2023AA05849)**
4. **TANU MADAAN (2023AA05511)**

Part 1: Theory and Concepts (10 Points)

1. Define vertical partitioning and horizontal partitioning in the context of AI models.

VERTICAL PARTITIONING:

Vertical partitioning involves splitting a dataset or model into multiple smaller parts based on features (columns). Each partition focuses on a subset of features that are relevant to specific aspects of the problem. In AI models, vertical partitioning can be used to separate different types of features and process them independently before combining their outputs to make final predictions.

Example: In a wireless network latency prediction problem, the dataset consists of network-related and user-related features:

a) **Model A:** Processes network-related features such as Signal Strength and Network Traffic.

b) **Model B:** Processes user-related features such as User Count and Device Type.

c) The outputs of both models are then combined to predict latency using methods like Model Averaging, Feature Ensemble, Ensemble Averaging.

HORIZONTAL PARTITIONING:

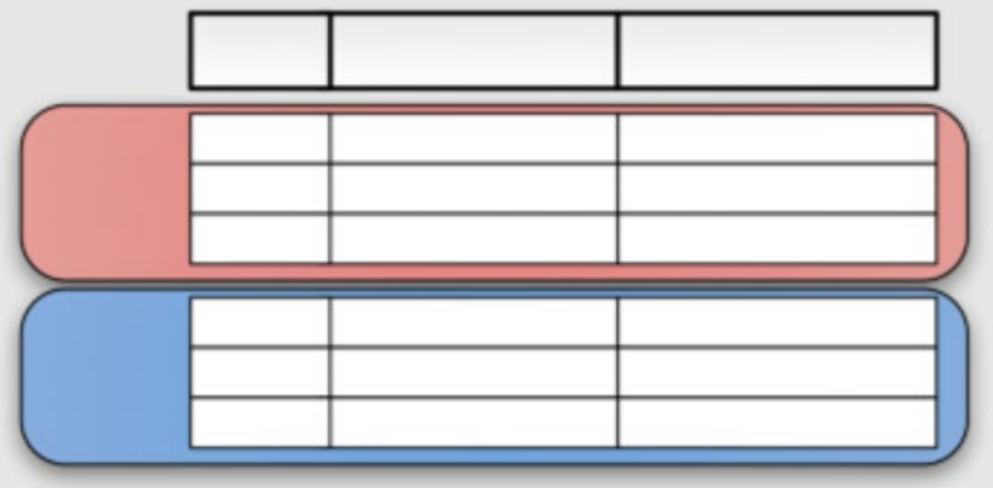
Horizontal partitioning divides the dataset into subsets based on rows, typically using a specific criterion (e.g., geographic location, device type, or network conditions). Each partition contains a portion of the dataset that shares a common characteristic.

Example: For network latency prediction, we can split the dataset into:

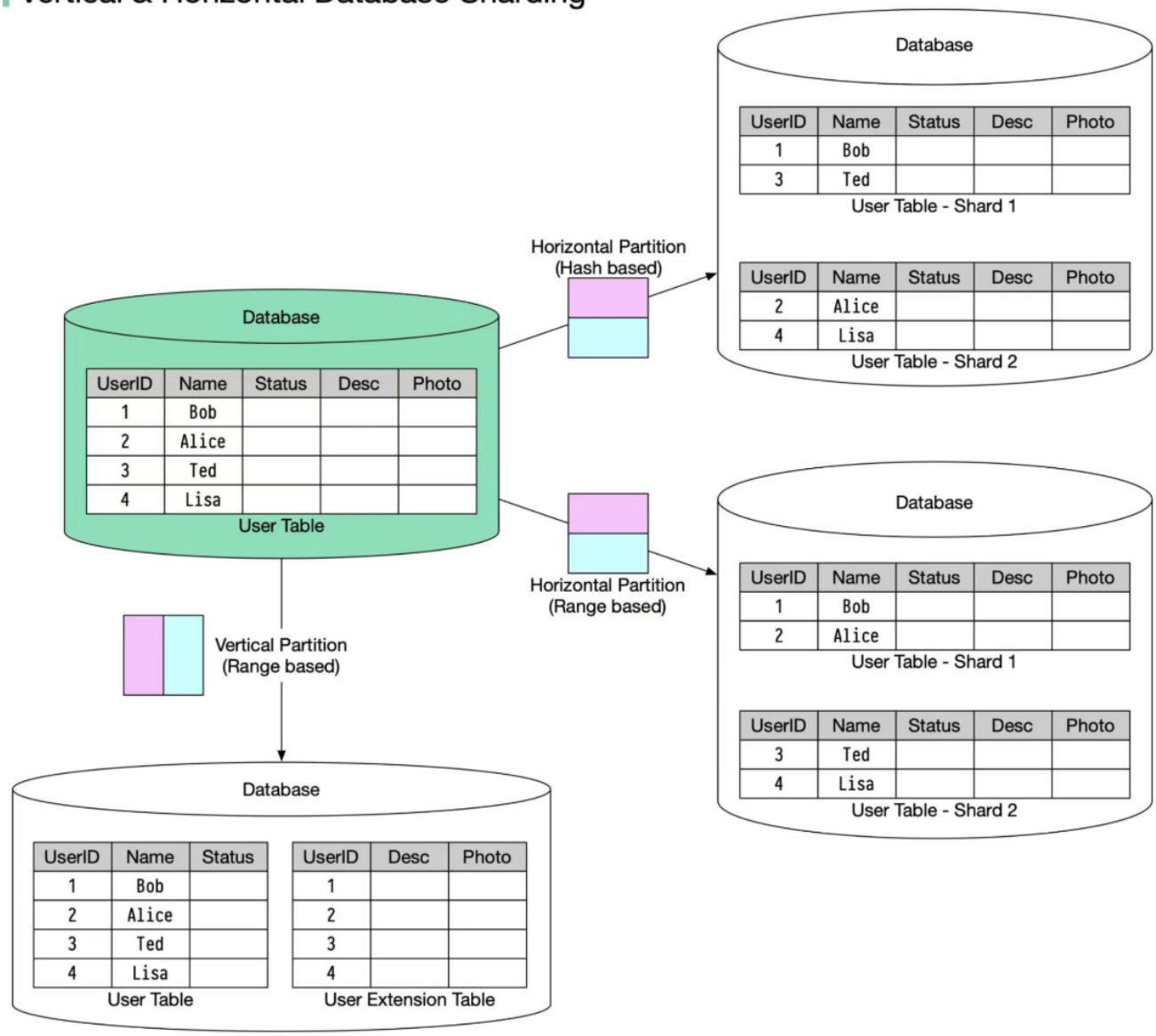
a) **Subset 1:** Urban towers.

b) **Subset 2:** Rural towers.

c) Separate models are trained for each subset, which can result in better predictions tailored to each group. Class balancing to be performed using Stratified sampling.



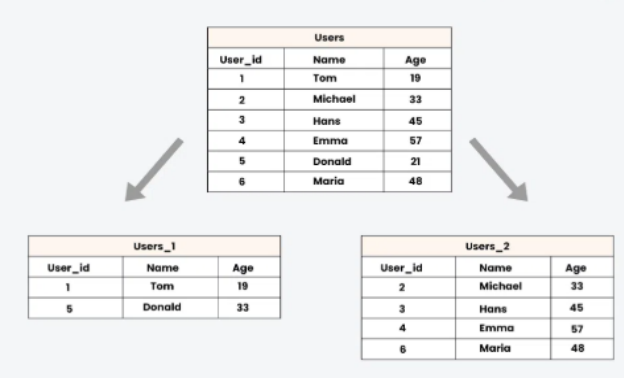
Vertical & Horizontal Database Sharding



2. Compare the advantages and disadvantages of these partitioning methods, focusing on computational efficiency, scalability, and real-world application in wireless networks (you can take any dataset).

Model and Data Partitioning are essential techniques in Distributed Machine Learning to handle large-scale datasets and complex models efficiently. Model partitioning splits a model across multiple devices to overcome memory limitations, while data partitioning divides data among workers to enable parallel processing, improving training speed and scalability.

HORIZONTAL PARTITIONING



Users

User_id	Name	Age
1	Tom	19
2	Michael	33
3	Hans	45
4	Emma	57
5	Donald	21
6	Maria	48

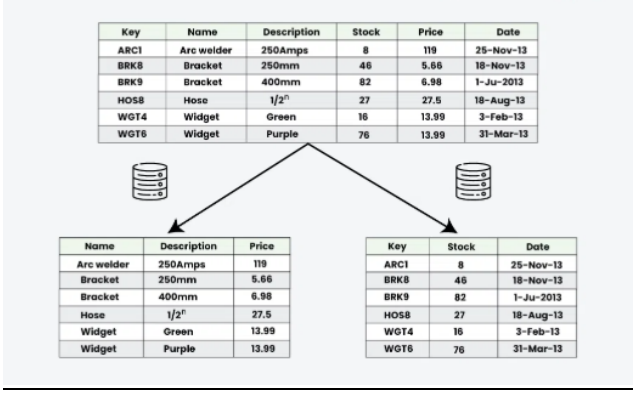
Users_1

User_id	Name	Age
1	Tom	19
5	Donald	33

Users_2

User_id	Name	Age
2	Michael	33
3	Hans	45
4	Emma	57
6	Maria	48

VERTICAL PARTITIONING



Key	Name	Description	Stock	Price	Date
ARC1	Arc welder	250Amps	8	119	25-Nov-13
BRK8	Bracket	250mm	46	5.66	18-Nov-13
BRK9	Bracket	400mm	82	6.98	1-Ju-2013
HOS8	Hose	1/2"	27	27.5	18-Aug-13
WGT4	Widget	Green	16	13.99	3-Feb-13
WGT6	Widget	Purple	76	13.99	31-Mar-13

Name	Description	Price
Arc welder	250Amps	119
Bracket	250mm	5.66
Bracket	400mm	6.98
Hose	1/2"	27.5
Widget	Green	13.99
Widget	Purple	13.99

Key	Stock	Date
ARC1	8	25-Nov-13
BRK8	46	18-Nov-13
BRK9	82	1-Ju-2013
HOS8	27	18-Aug-13
WGT4	16	3-Feb-13
WGT6	76	31-Mar-13

COMPARISON TABLE

Aspect	Horizontal Partitioning	Vertical Partitioning
Computational Efficiency	- Enables parallel processing by distributing rows across multiple nodes.	- Improves query efficiency when only a subset of columns is needed.
	- Reduces the size of data processed per query, improving query performance for row-based operations.	- Reduces storage I/O by limiting data retrieval to necessary attributes.
	- Can slow down operations requiring joins across partitions.	- Joins across partitions can be expensive and may require additional processing.

Scalability	- Highly scalable as data can be distributed across multiple servers.	- Less scalable when the number of columns grows.
	- Supports distributed processing and load balancing.	- Schema modifications (adding new columns) may require reorganization of partitions.
	- Efficient for handling increasing data volume.	- Efficient when only specific columns are frequently accessed.
Real-World Applications in Wireless Networks	- Used in large-scale network monitoring systems where each partition contains data for specific time periods or regions.	- Useful in wireless network optimization where different network parameters (signal strength, traffic, latency) are accessed independently.
	- Helps in optimizing query performance by storing recent network activity separately from historical data.	- Helps in efficient data retrieval when only specific attributes are needed for AI model predictions.
	- Ensures fault tolerance, as failure of one partition does not impact others.	- Reduces data transfer costs by retrieving only relevant network features for processing.

USAGE

- **Use Horizontal Partitioning** when handling large volumes of wireless network data that need to be distributed across multiple servers for better load balancing and parallel processing.

- **Use Vertical Partitioning** when optimizing AI models that require specific network parameters (e.g., signal strength, latency) without retrieving unnecessary attributes, reducing I/O costs and improving query performance.