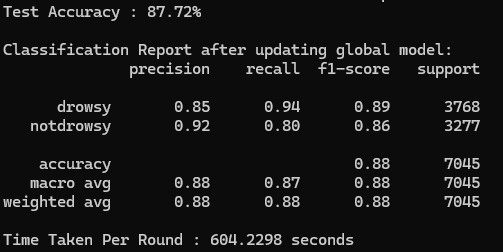
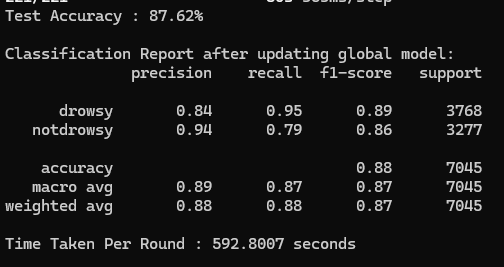
Server Round 1



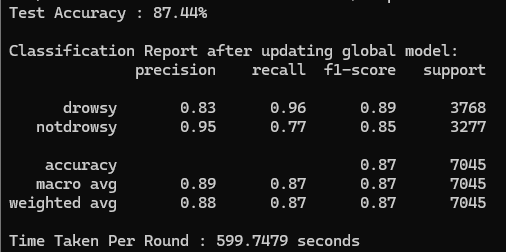
Round 2



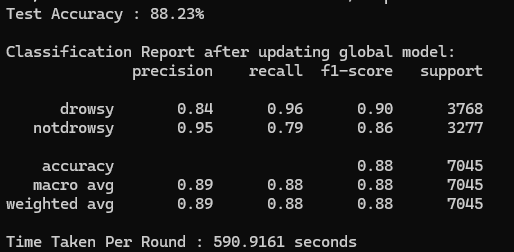
Round 3



Round 4

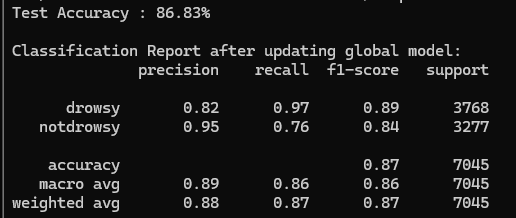


Round 5

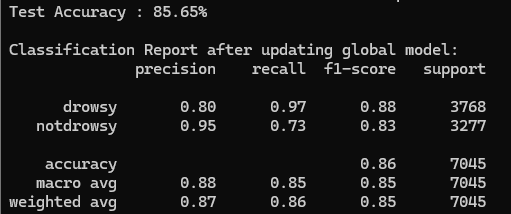


Client1

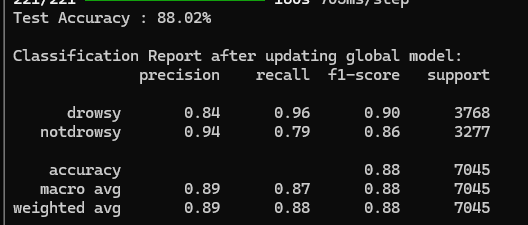
Round 1



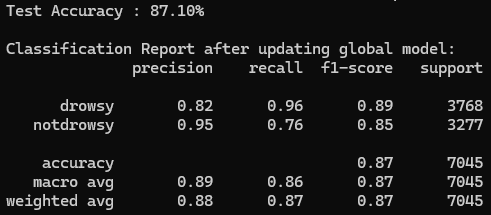
Round2



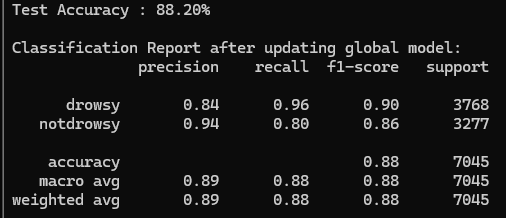
Round 3



Round 4

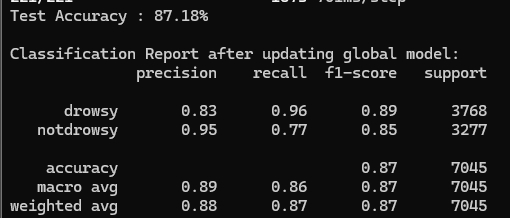


Round 5

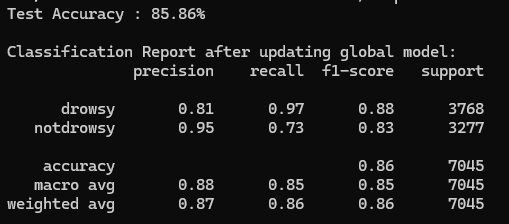


Client2

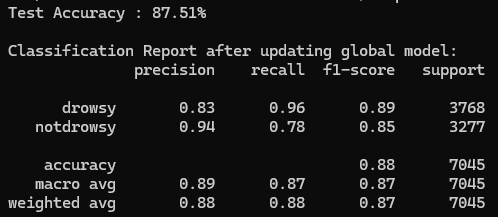
Round 1



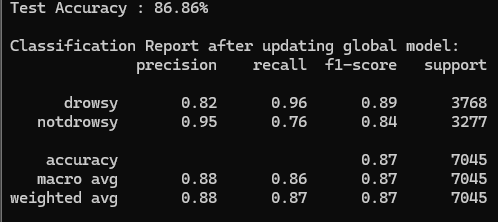
Round 2



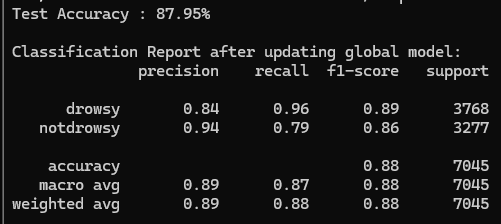
Round 3



Round 4



Round 5



| **Feature / Step** | **CDA\_V1 (Your Screenshot)** | **CDA\_Improved\_KMeans (My Code)** | **Comments** |
| --- | --- | --- | --- |
| **Client Clustering** | ✅ Yes (via cluster\_clients) | ✅ Yes (real KMeans using flattened weights) | ✅ Both cluster, mine explicitly uses sklearn.KMeans |
| **Handling Empty Clusters** | ✅ Skips empty ones | ✅ Same | Well-handled in both |
| **Intra-cluster Aggregation** | ✅ Yes (mean per layer) | ✅ Yes | Same logic |
| **Inter-cluster Aggregation** | ❌ Not explicitly handled (just adds clusters together) | ✅ Uses **divergence-aware fusion** if 2 clusters | ⚠️ This is the key **missing piece** in your code |
| **Cosine Similarity Fusion** | ❌ Not used | ✅ Yes | FedCDA paper emphasizes divergence-based blending |
| **Cluster Size Weighting** | ❌ No | 🔁 Optional | Could improve fairness in unbalanced scenarios |
| **Temporal Smoothing** | ✅ Yes (adaptive lambda) | ✅ Yes | Both correctly implement FedCDA's smoothing |
| **Code Modularity** | ⚠️ Moderate | ✅ High (more structured & future-ready) | My version is more plug-and-play |
| **Scalability** | ✅ Yes (multi-client support) | ✅ Yes | Both scale well beyond 2 clients |

