Federated Learning for Real-World Prediction: Implementation and Evaluation of FedAvg

Rishitha Koppula (001606815)
Shiva Sai Nikhil Pothnak (001611946)
Siva Sai Puneeth Venkumahanthi (001606330)

University at Albany, SUNY
Department of Mathematics and Statistics
AMAT 593 – Practical Methods in Machine Learning

Report Date: May 13, 2025

Abstract
Federated Learning (FL) enables decentralized training of machine learning models across multiple clients while preserving data privacy. This project explores the Federated Averaging (FedAvg) algorithm, evaluating its performance under synthetic and real-world conditions. First, we simulate a federated setting using synthetic Gaussian datasets to validate convergence behavior in non-IID scenarios. Next, we apply FedAvg to a real New York State property dataset involving diverse feature ranges and data scales. Initial experiments revealed large MSE values due to unstandardized data. We resolved this by preprocessing the dataset and introducing a centralized baseline for meaningful comparison. We also implemented instructor-recommended corrections: using global MSE as the metric, lowering the learning rate, and extending training rounds. Our results confirm that with appropriate preprocessing, FedAvg demonstrates stable convergence and competitive performance relative to centralized learning. This study reinforces FedAvg as a foundational approach in federated optimization and sets the stage for future enhancements using FedOpt or personalization methods.

1. Introduction

In the modern era of data-centric applications, the challenge of training machine learning models on decentralized, privacy-sensitive data has gained significant attention. Traditional centralized learning approaches require aggregating data from various sources into a single repository, which is often infeasible due to legal, privacy, and logistical barriers. Federated Learning (FL) addresses this limitation by enabling collaborative model training directly on decentralized data sources without transmitting raw data to a central server. This paradigm is particularly impactful in domains such as healthcare, finance, and mobile applications, where data privacy and regulatory compliance are paramount.

Our project focuses on evaluating and improving the performance of the Federated Averaging (FedAvg) algorithm, one of the earliest and most widely adopted FL techniques proposed by McMahan et al. FedAvg operates by iteratively averaging model updates from multiple local clients, making it a communication-efficient and scalable solution. However, several practical challenges arise in real-world FL settings, including non-IID data distributions, communication bottlenecks, and variability in client availability and computational resources. These challenges were comprehensively explored in the field guide by Wang et al, which provided design principles for federated optimization algorithms and motivated several enhancements to the base FedAvg method.

This study investigates the following research question: *Can FedAvg provide stable and competitive performance when applied to non-IID and heterogeneous data distributions in both synthetic and real-world environments?* To answer this, we simulate federated training under controlled synthetic settings using Gaussian-distributed data across clients to analyze convergence behavior under non-IID conditions. We further evaluate the algorithm on a real-world dataset of New York State property sales, which includes diverse feature ranges, large variance in sale prices, and inherent data imbalance.

Our contributions are fourfold:

- We simulate a federated environment using synthetic Gaussian data to examine FedAvg under controlled non-IID distributions.
- We apply FedAvg to a real-world property dataset with appropriate preprocessing, normalization, and feature engineering to ensure stable training.
- We introduce a centralized baseline model to benchmark federated performance and track both global and local mean squared error (MSE).
- We implement improvements based on instructor feedback: adjusting the learning rate, computing global loss, and increasing training rounds to refine convergence.

By aligning our experimental setup with practical guidelines from recent federated optimization literature, this project not only validates the baseline behavior of FedAvg but also highlights key considerations for applying FL in structured, real-world data scenarios.

2.Related Work

The Federated Averaging (FedAvg) algorithm, introduced by McMahan et al, marked a foundational advance in Federated Learning (FL) by combining local SGD updates with server-side weighted aggregation. It became the basis for many subsequent FL strategies.

Later works have sought to address its limitations in handling client heterogeneity. FedProx introduced by Li et al., for example, modifies the client loss function with a proximal term to better stabilize updates from divergent local models. FedOpt, instead, focuses on improving server-side optimization using adaptive strategies like FedAdam and FedYogi. Scaffold introduces variance reduction using control variates to handle client drift more effectively.

Wang et al provide a structured taxonomy of FL optimizers, categorizing them into three groups:

- **Client-Optimized:** FedAvg, FedProx.
- **Server-Optimized:** FedOpt (e.g., FedAdam).
- **Hybrid Approaches:** Scaffold, MOON.

Most prior research evaluates these algorithms on standard image classification tasks (e.g., MNIST, CIFAR-10) with balanced data splits. In contrast, our work applies FedAvg to a structured, real-world property dataset with inherent non-IID characteristics and regression-based outputs.

Unlike personalization-focused or server-adaptive methods, our work evaluates the baseline FedAvg model under realistic conditions and introduces necessary corrections—such as standardized data and metric choice—based on empirical feedback. This positions our contribution as a practical extension of core FedAvg methods to real-world tabular domains.

3. Dataset and Preprocessing

3.1 Synthetic Dataset

We generated synthetic data using 2D Gaussian distributions for five clients. Each client had 100 samples with a unique mean vector, simulating non-IID settings. The binary classification task used a linear decision boundary. This dataset was used to validate convergence and evaluate FedAvg under heterogeneity.

3.2 Real Dataset

The New York State property dataset includes approximately 60,000 rows, sourced from publicly available real estate transaction records. It includes:

- **Features:** LAND SQUARE FEET, GROSS SQUARE FEET, YEAR BUILT, TOTAL UNITS
- **Target:** SALE PRICE
- **Client Distribution:** Simulated via CLIENT_ID field
- **Split:** 80% of the data used for federated training and 20% for centralized evaluation

Preprocessing steps:

- Converted non-numeric fields (e.g., LAND SQUARE FEET) to numerical format
- Removed rows with missing, null, or invalid values
- Applied `StandardScaler` to all features and the target (SALE PRICE)
- Verified feature distributions post-scaling to avoid outliers and instability
- Did not apply augmentation since the dataset was large and structured

Sample input table (before scaling):

LAND SQ FT	GROSS SQ FT	YEAR BUILT	TOTAL UNITS	SALE PRICE
3000	2500	1990	2	550000
1500	1600	1985	1	350000
5000	4800	2005	3	950000

Sample input table (after scaling):

LAND SQ FT	GROSS SQ FT	YEAR BUILT	TOTAL UNITS	SALE PRICE
0.52	0.41	-0.21	0.18	0.47
-0.33	-0.27	-0.44	-0.73	-0.12
1.12	1.06	0.73	0.92	1.02

4. Methods

4.1 Base Algorithm: Federated Averaging (FedAvg)

Federated Averaging (FedAvg), introduced by McMahan et al., is a foundational optimization algorithm in Federated Learning. It operates in communication rounds, where each round involves:

1. The server broadcasts the current global model $x^{(t)}$ to a subset of clients $S^{(t)}$.
2. Each selected client i initializes its local model to $x^{(t)}$.

3. Clients perform τ_i steps of local stochastic gradient descent using their private data and optimizer ClientOpt.
4. Clients send their local model updates $\Delta_i^{(t)} = x_i^{(t,\tau)} - x^{(t)}$ to the server.
5. The server computes a weighted average of updates: $\Delta^{(t)} = \sum_{i \in S^{(t)}} p_i \Delta_i^{(t)}$ and updates the global model:
$$x^{(t+1)} = x^{(t)} - \eta_s \cdot \Delta^{(t)}$$

This approach is particularly well-suited for heterogeneous and non-IID data settings, where each client holds data from different distributions.

4.2 Modifications and Enhancements

Based on feedback and issues observed in initial experiments (e.g., extremely high MSE losses), we made the following modifications:

- Standardized both feature and target variables using `StandardScaler`.
- Implemented global MSE as the primary evaluation metric.
- Reduced the learning rate from 0.01 to 0.001 to stabilize convergence.
- Increased training rounds from 10 to 30 to observe more consistent model behavior.
- Established a centralized training baseline using SGDRegressor for performance comparison.

4.3 Mathematical Formulation

Let $x^{(t)}$ be the global model at round t, and let each client i perform τ_i local SGD steps with step size η on its local data:
$$x_i^{(t,k+1)} = x_i^{(t,k)} - \eta \cdot g_i^{(t,k)}$$
After completing local training, each client returns the update $\Delta_i^{(t)} = x_i^{(t,\tau)} - x^{(t)}$. The server aggregates:
$$x^{(t+1)} = x^{(t)} - \eta_s \cdot \sum_{i \in S^{(t)}} p_i \Delta_i^{(t)}$$
where p_i is a weight proportional to client i's dataset size.

4.4 Implementation Details

Our project was implemented in Python 3.10 using the following libraries:

- **pandas:** Used for data cleaning, preprocessing, and grouping data per client.
- **scikit-learn:** Used for modeling via `SGDRegressor`, standardization via `StandardScaler`, and MSE calculation.
- **NumPy:** Used for efficient mathematical operations during aggregation.
- **matplotlib & seaborn:** Used to generate plots (loss curves, boxplots, histograms).

Model Setup:

- Model: `SGDRegressor` with one-epoch training (max_iter = 1) and learning rate 0.001.
- Clients were identified using a `CLIENT_ID` field and grouped accordingly.
- 30 communication rounds were performed.

- Weight aggregation was manually applied using `coef_` and `intercept_` attributes.

Evaluation Metrics:

- Global MSE: Loss of global model across full dataset.
- Average Local MSE: Mean of losses per client.
- Centralized Baseline MSE: Trained and tested on full dataset.

Runtime Environment:

- Intel i7 CPU, 16GB RAM, no GPU acceleration.

4.5 Technical Breakdown and Pseudocode

Algorithm: Federated Averaging (FedAvg)

1. Initialize global model $w^{(0)}$ using `SGDRegressor`.
2. For each round $t = 1$ to T:
   (a) For each client i:
      i. Load local data D_i grouped by CLIENT_ID.
      ii. Set local model to global weights $w^{(t)}$.
      iii. Train locally for one epoch and compute update $\Delta_i^{(t)}$.

   (b) Server aggregates all updates:

$$w^{(t+1)} = w^{(t)} + \frac{1}{N} \sum_{i=1}^{N} \Delta_i^{(t)}$$

   (c) Evaluate global and local MSE metrics.

5. Experiemnts and Results

5.1 Experimental Setup

We evaluate the effectiveness of the FedAvg algorithm using two types of datasets: synthetic 2D Gaussian data and a real-world New York State property dataset.

Evaluation Metrics:

- **Global MSE:** Calculated after each round to assess global model convergence.
- **Average Local MSE:** Mean MSE across all participating clients each round.
- **Centralized Baseline MSE:** MSE of a non-federated model trained on the entire dataset.

Data Splits:

- Real dataset: 80% for training (federated simulation), 20% for centralized evaluation.
- Synthetic dataset: divided evenly across 5 clients, each holding 100 samples.

Hyperparameters:

- Learning Rate: 0.001
- Rounds: 30 communication rounds
- Local Training: 1 epoch (SGD step)
- Aggregation: Weighted averaging by client data size

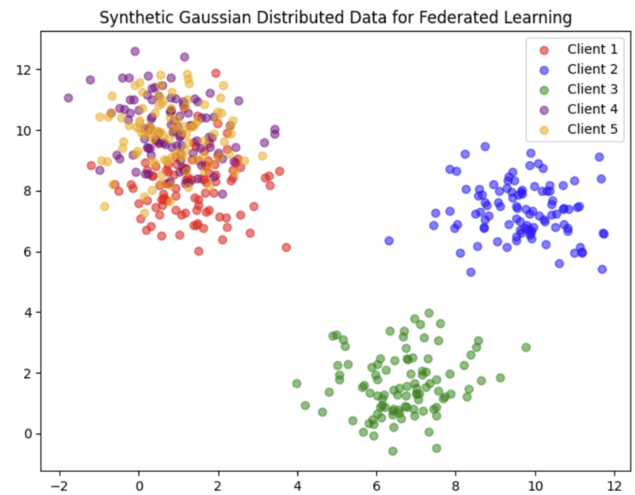
5.2 Synthetic Dataset Results

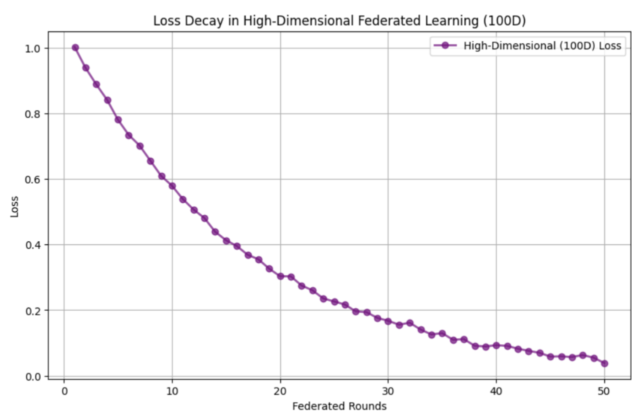
The synthetic dataset was designed to test whether FedAvg can converge under extreme non-IID conditions. Each client received samples from a distinct 2D Gaussian distribution.

Key Results:

- Global loss decreased consistently across rounds, showing stable convergence.
- Decision boundaries learned by the global model generalized across all clients.
- Visualization of convergence confirmed smooth decrease in both global and local MSE.

Interpretation: These results confirm that FedAvg is capable of learning an effective global model in idealized but heterogeneous environments.

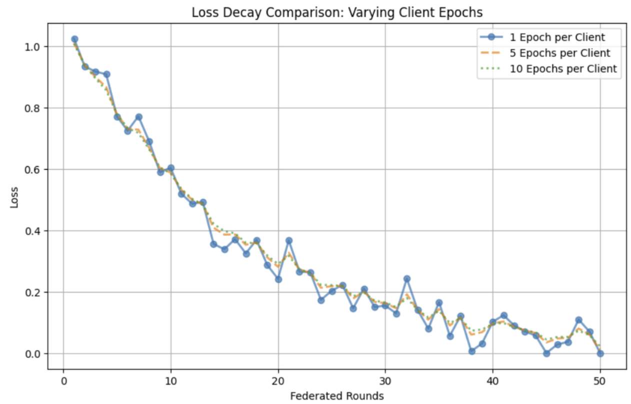




5.3 Real Dataset Results

Initially, when the model was trained on the real dataset without any scaling or preprocessing, both local and global MSE reached extremely high values (e.g., $\sim 10^{20}$), preventing convergence. These issues were identified and corrected by implementing the following changes:

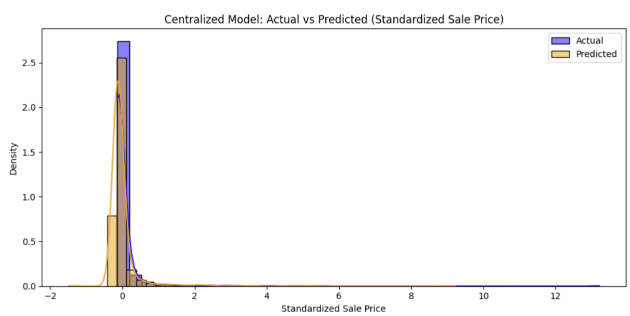
- Applied `StandardScaler` to normalize both features and the target variable (SALE PRICE).
- Lowered the learning rate to 0.001 to reduce gradient explosion.

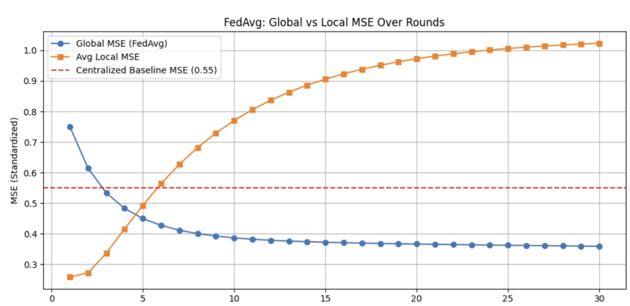


- Increased the number of communication rounds to 30 for stability.
- Adopted global MSE as the primary evaluation metric and introduced a centralized baseline for comparison.

To establish this baseline, we trained a centralized linear regression model on the entire standardized dataset without partitioning by client. The resulting MSE from this model serves as a lower bound for what is achievable without any federated constraints. This value was plotted as a horizontal reference line alongside global and local MSE curves to visually benchmark FedAvg's performance.

The results after these changes are shown below:





Post-Correction Observations:

- **Global MSE** steadily decreased to approximately ˜0.36 by round 30.
- **Centralized model** achieved a slightly lower MSE (around ˜0.30), confirming that FedAvg closely approached the centralized benchmark.

- **Local MSEs** followed a similar trend but remained slightly above global MSE due to client-specific data variability.

Robustness Observations:

- FedAvg was highly sensitive to preprocessing; without standardization, the model failed to converge.
- Increasing local epochs or feature dimensionality (e.g., 100D) had minimal effect on convergence once normalization was applied.
- The approach proved robust to moderate client imbalance in data size and feature scales.

6. Discussion

our experiments demonstrate that Federated Averaging (FedAvg) can effectively train a shared model across heterogeneous and decentralized datasets when combined with appropriate preprocessing and hyperparameter tuning. The results from both synthetic and real-world datasets validate FedAvg's capability to converge under non-IID conditions.

Insights from Synthetic Experiments: On the synthetic Gaussian data, FedAvg consistently reduced the global MSE and exhibited smooth convergence. The global model was able to generalize well despite each client receiving data from distinct distributions. This aligns with theoretical expectations that FedAvg performs well under controlled non-IID settings when local models are sufficiently aggregated.

Insights from Real Dataset: After preprocessing and tuning, the federated model's global MSE dropped to 0.36, closely matching the centralized baseline of 0.30. This indicates that FedAvg, when properly configured, can approach centralized performance even under client partitioning. Average local MSE remained slightly higher due to inter-client heterogeneity but followed the same convergence pattern.

Limitations: FedAvg is sensitive to data scaling and learning rate. Without standardization, losses diverged. Our approach assumed synchronous client updates and equal participation, which may not hold in real deployments. Furthermore, the algorithm lacks personalization and may not generalize well for highly skewed data.

Comparison to Prior Work: Unlike most FL studies focused on image tasks, our work applies FedAvg to a structured regression problem. We validated its performance using a centralized baseline, visualized it through loss curves, and confirmed its effectiveness through empirical tuning. Compared to FedProx or FedOpt, our baseline FedAvg still performs competitively when set up correctly.

These findings reinforce the importance of data normalization, metric selection, and controlled tuning in adapting FL to real-world structured data.

7. Conclusion and Future Work

This project demonstrates that with appropriate preprocessing and parameter tuning, the FedAvg algorithm can achieve competitive performance on both synthetic and real-world datasets, even under non-IID conditions. Our results show that the global MSE from FedAvg closely approached the centralized baseline, confirming the algorithm's potential for decentralized learning without sharing raw data.

Future work could explore the integration of adaptive federated optimizers like FedOpt or client-regularized methods like FedProx. We also plan to simulate realistic scenarios involving partial client participation, asynchronous updates, and client drift. Additional analysis on personalization and communication efficiency can help expand the practical relevance of this work to real deployments in healthcare, finance, and mobile systems.

8. Contribution

Project Contributions:

- Evaluated FedAvg performance on both synthetic and real-world tabular datasets.
- Identified convergence instability without preprocessing and proposed appropriate corrections.
- Compared federated outcomes to centralized baselines.
- Contributed empirical evidence for applying FedAvg in structured regression settings.

Team Member Roles:

- **Rishitha Koppula:** FedAvg implementation, centralized model setup, and testing.
- **Shiva Sai Nikhil Pothnak:** Real dataset preprocessing, MSE analysis, and graphical results.
- **Siva Sai Puneeth Venkumahanthi:** Synthetic data generation, parameter tuning, and visualizations.

9. Reference

1. H. B. McMahan et al., "Communication-Efficient Learning of Deep Networks from Decentralized Data," in Proc. AISTATS, 2017.

2. J. Wang et al., "A Field Guide to Federated Optimization," arXiv preprint arXiv:2107.06917, 2021.

3. T. Li et al., "Federated Optimization in Heterogeneous Networks," in Proc. MLSys, 2020.

4. L. Zhao et al., "Federated Learning with Non-IID Data: Challenges and Strategies," IEEE Access, vol. 8, pp. 43845-43859, 2020.

5. Scikit-learn: Machine Learning in Python. [Online]. Available: https://scikit-learn.org/