Federated Learning Algorithm without Synthetic Data (FL_LR):

1. *Data Distribution:*
   Each client splits their data into 70% training data and 30% testing data.

2. *Parameter Initialization:*
   The server generates the initial model parameters (IMP) to all be 0.

3. *Stopping Criteria Initialization:*
   (a) If the absolute difference between two values of the logistic loss function from two consecutive, distinct iterations is less than or equal to $\epsilon = 0.01$
   OR
   (b) The maximum number of iterations (e.g., $I = 1000$) has been reached, whose counter is initially set to a value of 0.

4. Local Training: Server sends the model to the clients for training.
5. Start timer.
   If either stopping criterion is not met, then:
   (a) Client trains their local data using logistic regression, gradient descent, and the parameters sent by the server until criterion (a) is met.
   (b) Client's model parameters are then updated accordingly.
   (c) Client sends the model's parameters to the global server.
   (d) The server uses Federated Averaging to create the global model parameters using the clients' local model parameters.
   (e) The server sends the global model parameters to the clients.
   (f) Each client uses these global model parameters for testing on their testing set and then calculates the value of the logistic loss function.
   (g) Each client sends the calculated value of the logistic loss function to the server.
   (h) The server aggregates all of the clients' values of the logistic loss function by taking the average.
   (i) The server checks for stopping criterion (a).
   (j) Iteration counter is incremented by 1.
6. Stop timer, record final number of iterations, and calculate total time taken.

Federated Learning Algorithm with Synthetic Data (FL_SD_LR):

1. *Data Distribution:*
   (a) Each client splits their data into 70% training data and 30% testing data.
   (b) Each client creates a model to fit their training data.
   (c) Start timer.
   (d) Synthetic data from the model is then generated.

2. *Parameter Initialization:*
   (a) Each client sends the SD to the global server.
   (b) The global server creates the initial model parameters from the SD sent by the clients.

3. *Stopping Criteria Initialization:*
   (a) If the absolute difference between two values of the logistic loss function from two consecutive, distinct iterations is less than or equal to $\epsilon = 0.01$

   OR

   (b) The maximum number of iterations (e.g., $I = 800$) has been reached, whose counter is initially set to a value of 0.

4. Local Training: Server sends the model created in 2(b) to the clients for training.
   If either stopping criterion is not met, then:
   (a) Client trains their real, non-SD local data using logistic regression, gradient descent, and the parameters sent by the server until criterion (a) is met.
   (b) Client's model parameters are then updated accordingly.
   (c) Client sends the model's parameters to the global server.
   (d) The server uses Federated Averaging to create the global model parameters using the clients' local model parameters.
   (e) The server sends the global model parameters to the clients.
   (f) Each client uses these global model parameters for testing on their testing set and then calculates the value of the logistic loss function.
   (g) Each client sends the calculated value of the logistic loss function to the server.
   (h) The server aggregates all of the clients' values of the logistic loss function by taking the average.
   (i) The server checks for stopping criterion (a).
   (j) Iteration counter is incremented by 1.
5. Stop timer, record final number of iterations, and calculate total time taken.

6. Steps 1(c) to 5 are repeated 19 more times (a final total of 20 times).

Time Measurement:

There are two possible ways to compute time taken for algorithm completion.

1. There exists a built-in timer function in python3 that will help with calculating the time the algorithm takes to be completed for model convergence.

   Here are the steps:
   i. A timer is started as soon as the first iteration begins; that is, when the client first trains their data.[1]
   ii. The timer is then stopped once model convergence occurs; that is, either when the server finds that criterion (a) or (b) is true.
   iii. The difference between the time when stopped and started is the total time the algorithm has taken to terminate.

2. Alternatively, we can also measure time taken algorithm completion by the final number of iterations when model convergence occurs.

Final Model Performance Calculations:

1. Total Number of Iterations:

   The final total number of iterations for model convergence is calculated as previously outlined. It is one insight of final model performance because it emphasizes how many times a model had to be trained and tested for it to reach optimal parameters. So, given two models (one with a higher number of total iterations than the other), one can rightfully hypothesize the algorithm used for the model that took fewer iterations for convergence performs, on average, better and more efficiently than the other model.

2. Total (Computation) Time:

   The total (computation) time for model convergence is calculated as outlined above. It is one insight of final model performance because it helps assess both efficiency and scalability of the algorithm. So, given two models (one that took less time for model convergences than the other), once can rightfully hypothesize the algorithm used for the model that took less computational time is likely to be more efficient and highly scalable than the other model.

[1] For FL_SD_LR, the timer starts when the process of synthetic data generation starts.

Final Model Comparison

1. Logistic Loss Function (Binary Cross Entropy):

The logistic loss can provide insights into the relative performance of two models on the same dataset when used as a comparative measure. By comparing the logistic loss values between two models, you can assess which model performs better in terms of minimizing prediction error and aligning predicted probabilities with true binary labels.

When comparing two models on the same dataset, lower logistic loss values indicate better performance. A model with lower logistic loss is closer to making accurate predictions and has a better fit to the data. It suggests that the model's predicted probabilities align more closely with the true binary labels.

By comparing the logistic loss values, you can make relative judgments about the performance of the models. For example, if Model A has a lower logistic loss than Model B on the same dataset, it suggests that Model A is performing better in terms of prediction accuracy and goodness of fit.

However, it is important to note that logistic loss alone may not provide a complete picture of model performance, and it should be considered alongside another evaluation metric, such as Logarithmic Likelihood.

The following is its formula:

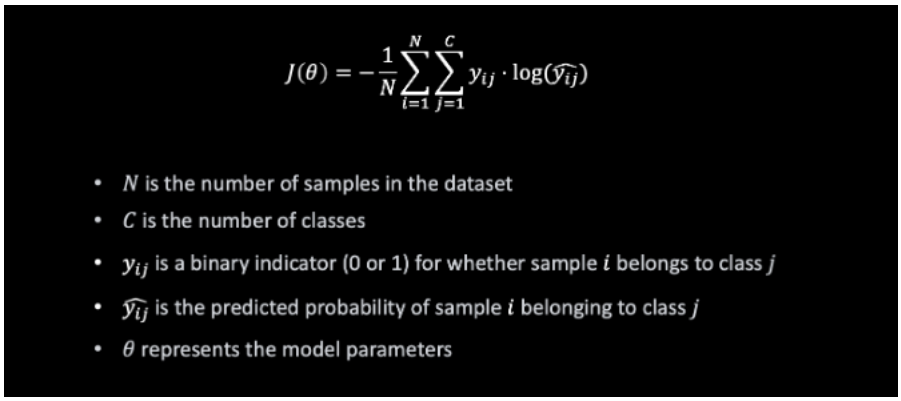
$$L(y, \hat{y}) = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)$$

- y is the actual binary label (0 or 1).
- $\hat{y}$ is the predicted probability of the positive class (i.e., the probability that the label is 1)
- N is the number of samples in the data set.

2. Logistic Loss Function (Categorical Cross Entropy):

As opposed to Binary Cross Entropy, Categorical Cross Entropy is used when there exist more than 2 predicted classes in the dataset. Nonetheless, it retains the same properties and insights to model prediction errors.

The following is its formula:



3. Logarithmic Likelihood:

Though there is a lot of resemblance in the formulas between Logistic Loss and Logarithmic Likelihood, they differ in purpose. Logarithmic Likelihood is best used to compare different models (while every other variable is fixed). Logarithmic Likelihood between two models gives an insight to how well these models fit, given the data. The range of the Logarithmic Likelihood function in this context is from negative infinity to 0, where the model is the best fit when its Logarithmic Likelihood is 0 and its worst as it approaches negative infinity.

Since Logistic Loss and Logarithmic Likelihood share the exact same input requirements, it was the clear answer to choose Logarithmic Likelihood as another metric to measure similarities and differences between final models of FL_LR and FL_SD_LR.

The logarithmic Likelihood was measured using only the final model from FL_LR and FL_SD_LR for each of the six datasets.

Here is the formula for Logarithmic Likelihood for Binary-Cross Entropy and Categorical-Cross Entropy, respectively:

$$LLH(\theta) = \sum_{i=1}^{N} y_i log(\widehat{y_i}) + (1 - y_i) log(1 - \widehat{y_i})$$

$$LLH(\theta) = \sum_{i=1}^{N}\sum_{j=1}^{C} y_{ij} log(\widehat{y_{ij}})$$

Further Information

1. Elastic-Net Regularization:

- Regularization is a technique used to prevent overfitting, which occurs when a model becomes too complex and starts fitting the noise in the training data rather than the underlying patterns.
- In the context of the log loss function, regularization is applied to the model's parameters to prevent overfitting.
- By adding a regularization term to the log loss function, the model is penalized for having large or complex coefficients, which can help to prevent overfitting.
- The strength of the regularization term is controlled by a hyperparameter, which is tuned during model training using techniques such as cross-validation.
- Elastic-Net Regularization provides the following benefits: feature selection, handles correlated features well, effective for both small and large datasets, better performance than other regularization methods when features are correlated.

The following is the term added at the end of both Logistic Loss Functions (for both FL_SD and FL_SD_LR) for Elastic-Net Regularization:

$$\lambda_1 \sum_{j=1}^{p} |w_j| + \lambda_2 \sum_{j=1}^{p} |w_j|^2$$

- w_j is the logistic model's parameters
- p is the number of features
- λ is a hyperparameter that controls the strenght of the regularization

2. Client Information

i. Adult Income Dataset

- Domain of Clients: Continents
- Total Number of Clients: 5
- Total Number of Features per Client: 14
- Total Number of Instances: 48,842
- Response Variable: If an individual from a given continent earns <=$50k or >$50k US Dollars.
- Variable Type: Binary
- Dataset Link: http://archive.ics.uci.edu/ml/datasets/adult

ii. Automobile Dataset

- Domain of Clients: Automobile Companies
- Total Number of Clients: 22
- Total Number of Features per Client: 26
- Total Number of Instances: 205
- Response Variable: The price of the car, which can be one of 11 categories in range of $5,118 – $45,400 US Dollars.
- Variable Type: Categorical
- Dataset Link: https://archive.ics.uci.edu/ml/datasets/Automobile

iii. Diabetes Dataset

- Domain of Clients: Hospitals
- Total Number of Clients: 439
- Total Number of Features per Client: 49
- Total Number of Instances: 101,767
- Response Variable: If a patient in a given hospital has (non-zero) <30%, >=30%, or 0% chance of readmission.
- Variable Type: Binary
- Dataset Link: NA[2]

iv. Heart Disease Dataset

- Domain of Clients: Hospitals

[2] Dataset was provided via email with no URL link.

- Total Number of Clients: 4
- Total Number of Features per Client: 75
- Total Number of Instances: 920
- Response Variable: If a patient from a given hospital has low likelihood of presence of heart disease (0-2) or low likelihood of presence of heart disease (3-4)
- Variable Type: Categorical turned binary.
- Dataset Link: https://archive.ics.uci.edu/ml/datasets/heart+Disease

v. Student Performance Dataset

- Domain of Clients: Portuguese Schools
- Total Number of Clients: 2
- Total Number of Features per Client: 30
- Total Number of Instances: 1,044
- Response Variable: If a student passed with a score of >=50% or failed <50%.
- Variable Type: Binary
- Dataset Link: https://archive-beta.ics.uci.edu/dataset/320/student+performance

vi. University Quality of Life Dataset

- Domain of Clients: US States
- Total Number of Clients: 38
- Total Number of Features per Client: 17
- Total Number of Instances: 285
- Response Variable: If a university from a given state has a bad quality of life (0-2) or good quality of life (3-5).
- Variable Type: Categorical turned binary.
- Dataset Link: https://archive.ics.uci.edu/ml/datasets/University

Interpretation of Results:[3]

1. Number of Iterations:

   We can say that the difference in the total number of iterations taken between FL_LR and FL_SD_LR is statistically significant, at a 95% confidence rate. So, we reject the Null Hypothesis.

2. Total (Computation) Time:

   We can say that the difference in the total (computation) time taken between FL_LR and FL_SD_LR is statistically significant, at a 95% confidence rate. So, we reject the Null Hypothesis.

3. Logistic Loss Function Values:

   We can say that the difference in the logistic loss values between FL_LR and FL_SD_LR is statistically not significant, at a 95% confidence rate. So, we fail to reject the Null Hypothesis.

4. Logarithmic Likelihood:

   We can say that the difference in the logarithmic likelihood values between FL_LR and FL_SD_LR is statistically not significant, at a 95% confidence rate. So, we fail to reject the Null Hypothesis.

[3] With the given data and their types, their assumptions did not meet the criteria for accepted statistical tests. Therefore, it is enough to show significance and insignificance of the differences in values by seeing whether values of FL_LR lie within the confidence interval of values of FL_SD_LR, which were attained at a 95% confidence level.