

# Federated Learning With Flower

Progetto Finale per il corso di Machine Learning 2023/2024, Traccia B1

Giulio Appetito

University of Roma Tor Vergata

Rome, Italy

giulio.appetito@alumni.uniroma2.eu

0321669

Anastasia Brinati

University of Roma Tor Vergata

Rome, Italy

anastasia.brinati@alumni.uniroma2.eu

0321654

**Sommario**—Il presente documento intende esplorare e studiare sperimentalmente il Federated Learning in una varietà di scenari, tramite il framework Flower, al fine di confrontare le performance nelle varie configurazioni, per poi comparare tali performance con quelle ottenute dal modello centralizzato in ognuno dei task considerati.

## I. INTRODUZIONE

La seguente relazione ha lo scopo di illustrare uno studio sperimentale sul funzionamento del Federated Learning in una varietà di scenari diversi, utilizzando il framework per FL Flower. Flower è un framework open source per il federated learning che si integra con diversi backend, tra cui TensorFlow e PyTorch. Consente l'addestramento distribuito di modelli di machine learning in un ambiente federato, lavorando con i dati distribuiti sugli edge devices, senza la necessità di centralizzarli, in modo da garantirne la sicurezza e la privacy. Permette di simulare vari scenari di FL, eseguendo un numero anche elevato di client su una singola macchina, per eseguire esperimenti a larga scala e con dispositivi eterogenei.

Al fine di analizzare il comportamento di Flower in scenari differenti, si è scelto di affrontare la simulazione di tre differenti task (con rispettivi dataset):

- **Task I** Un task di regressione;
- **Task II** Un task di classificazione binaria (di dati tabulari);
- **Task III** Un task di classificazione multiclasse (di immagini).

Per ognuno di questi abbiamo considerato varie configurazioni, variando il numero di clients e la strategia di aggregazione; lo studio è stato rivolto al confronto, dunque, tra le varie configurazioni dell'ambiente di simulazione in termini di varie metriche dipendeneiti dal task, ed il confronto delle prestazioni tra il modello federato e la versione centralizzata del modello.

## II. PREPROCESSING

Idealmente, il dataset viene partizionato e distribuito ai vari clients di modo che si occupino del preprocessing in locale, prima di effettuare l'addestramento. Nonostante vari tentativi, su tutti i dataset considerati, anche nei casi

più favorevoli alla riuscita si ricadeva in problemi di serializzabilità/serializzazione, per cui i client non riuscivano ad eseguire le elaborazioni richieste. Una delle sfide principali è sicuramente dover lavorare con le strutture dati FederatedDataset e \_PrefetchDataset (\_MapDataset e \_TensorSliceDataset), fondamentali nell'ambiente simulato, che non hanno permesso di interagire e manipolare i training set. Stesso problema emerge affrontando lo scaling dei dati, anche quando si introduce un layer di normalizzazione nel modello stesso.

## III. MODELLAZIONE

In tutti e tre i task per la modellazione è stata utilizzata la classe `tf.keras.models.Model`, la quale raggruppa i livelli in un oggetto con funzionalità di addestramento e inferenza. L'utilizzo di questa specifica classe per la definizione del modello è dovuta ad una problematica introdotta dal framework Flower: come riportato nella documentazione, questo framework è stato testato solo su un numero limitato di datasets, i quali consistevano di una unica feature; pertanto, l'utilizzo di un dataset con più di una feature è risultato particolarmente ostico per via del fatto che il modello federato si aspettava in input una singola feature.

Al fine di risolvere questo problema si è deciso di progettare il modello introducendo livelli composti da una singola unità che gestissero individualmente le varie features, utilizzando la classe `tf.keras.layers.Input`, la quale consente di definire un tensore Keras. Questi livelli vengono poi concatenati in un unico livello, che fornisce l'input completo al modello vero e proprio. Sebbene questa struttura sia artificiosa e poco riusabile a livello di codice, è stata la soluzione più efficace per risolvere le varie problematiche sorte per via del framework.

Nonostante la struttura del modello federato differisca dalla struttura del modello centralizzato per via di questi livelli di input aggiuntivi, nella pratica questo non impatta in alcun modo l'addestramento e l'inferenza del modello dal momento che questi livelli non sono parametrici.

#### IV. TUNING

La validazione viene svolta sul modello centralizzato, poichè non potendo affrontare gli step di preprocessing in locale ai client, il dataset viene manipolato ed elaborato prima di essere distribuito; proprio per questa ragione, abbiamo ritenuto opportuno trovare gli iperparametri ideali per il modello quando si hanno più dati a disposizione.

Per affrontare il tuning degli iperparametri del modello è stata utilizzata la libreria Keras Tuner, messa a disposizione dall'ambiente TensorFlow, la quale aiuta nella selezione del set ottimale di iperparametri per la rete neurale. Nello specifico, è stato adottato il sintonizzatore HyperBand, il quale implementa una allocazione adattiva delle risorse e l'early stopping, al fine di convergere in modo rapido su un modello prestante: il sintonizzatore addestra un gran numero di modelli per alcune epoche, ma poi porta solo la metà dei modelli con le migliori prestazioni al round successivo.

Per il tuning degli iperparametri è stata dunque definita la griglia dei valori da testare per quanto riguarda il numero di livelli della rete neurale, le funzioni di attivazione da utilizzare nei vari livelli, e il numero di unità in ogni livello. Per tutti i tre task studiati, il tuning degli iperparametri è stato svolto in maniera centralizzata avendo l'intero dataset a disposizione, dal momento che risulterebbe particolarmente complesso trovare una configurazione ottima degli iperparametri che sia consistente e adatta a tutti gli scenari di simulazione che sono stati ideati per il nostro lavoro. La scelta di mantenere la stessa configurazione degli iperparametri per il modello centralizzato e quello distribuito ha lo scopo di confrontare le prestazioni tra i due differenti approcci di addestramento.

#### V. ESPERIMENTI

Sono state eseguite run di simulazione per varie configurazioni, variando il numero di client (5,10,100) e il tipo di strategia adottata (FedAvg, QFedAvg e FedProx). I valori per il numero di clients sono stati, in parte, dettati dalla dimensione dei dataset a disposizione: su alcuni dataset, un numero leggermente superiore ai 300 clients comportava infatti problemi di partizionamento dei dataset da parte del framework. Successivamente, abbiamo limitato il range del numero di clients a 100, per limitare le tempistiche delle simulazioni; inoltre, poichè le differenze nelle prestazioni tra 10 e 100 clients sono risultate minime, abbiamo optato per scegliere 5 come ultimo valore per il numero di clients. Il ciclo FL guida il processo di apprendimento federato, delegando le decisioni alla Strategia (Strategy): quest'ultima configura i round di FL, invia le configurazioni ai client, riceve gli aggiornamenti ed aggrega i risultati; il lato client si occupa di reagire alle istruzioni del server. Per quanto riguarda le *Strategy* considerate, abbiamo lavorato con tre strategie predefinite:

- **FedAvg**, una strategia che combina lo Stochastic Gradient Descent (SGD) locale su ogni client con il server, che esegue la media dei modelli [5];

- **FedProx**, una strategia che è una generalizzazione e riparametrizzazione di FedAvg in un contesto di eterogeneità tra i sistemi: FedAvg non consente ai clients di svolgere quantità variabili di lavoro in base ai loro vincoli di sistema sottostanti. Ad esempio, si sceglie di rigettare i client che falliscono nel calcolare  $E$  epoche in una certa finestra temporale. FedProx risolve questi problemi consentendo di avere quantità di lavoro variabili sui vari device, e introduce un sottoproblema locale, in cui si aggiunge un termine di prossimità per limitare efficacemente l'impatto degli aggiornamenti locali variabili [5];
- **QFedAvg**, una strategia che prova a risolvere il problema della allocazione fair delle risorse alle risorse di apprendimento (i client). L'idea su cui si basa questa strategia per raggiungere la fairness è quella di ripesare la funzione obiettivo del problema assegnando pesi maggiori ai device con performance peggiori, in modo tale che la distribuzione delle accuracy della rete sia quanto più uniforme possibile [5].

Il numero di round di simulazione è stato fissato a 10, un valore che ha consentito di mostrare proprietà di convergenza senza rendere eccessivamente prolungata la simulazione, dovendo eseguirne un gran numero.

Le tabelle in fondo alla relazione riportano il valore delle metriche sul test set raggiunto per ogni round, dato uno specifico numero di clients utilizzato per l'addestramento federato, suddivise per strategia adottata; sono stati prodotti anche i relativi grafici presenti nei notebook per aiutare a visualizzare i confronti e la convergenza dell'algoritmo al variare delle impostazioni della simulazione.

Al fine di confrontare i risultati delle simulazioni, l'idea è stata quella di progettare il confronto nel seguente modo:

- 1) per ogni metrica considerata, si fissa il numero di clients, e si studia la convergenza dei modelli federati e del modello centralizzato in funzione delle epoche/round;
- 2) per ogni metrica considerata, si fissa la strategia adottata, e si studia la convergenza delle varie configurazioni dei modelli in termini di numero di clients e del modello centralizzato in funzione delle epoche/round.

#### VI. TASK 1:

##### A. Dataset

Il primo task selezionato per lo studio è un task di classificazione binaria di dati tabulari; il dataset scelto è *Customer Churn Dataset* [1], ed è stato ottenuto dalla piattaforma *Kaggle*. Il dataset consiste in informazioni su 64.374 clienti, con caratteristiche come età, genere, permanenza, frequenza di utilizzo, supporto, pagamenti, abbonamenti e spesa totale; questo viene utilizzato per prevedere se i clienti lasceranno il servizio (churn).

##### B. Modello

Il modello scelto per questo task di classificazione binaria è una rete neurale Feedforward. A seguito del tuning degli

iperparametri, il modello selezionato per questo task è il seguente:

Layer	Numero di unità	Funzione di attivazione
Layer 1	32	ReLU (Rectified Linear Unit)
Layer 2	96	PReLU (Parametric ReLU)
Layer 3	128	tanh (Tangente iperbolica)
Layer 4	1	sigmoid

### C. Metriche

Per valutare le prestazioni del modello, sono state utilizzate quattro delle metriche più comuni per i compiti di classificazione binaria, fornite dalla libreria *keras.metrics*:

- Accuracy;
- Precision;
- Recall;
- F1 score.

La scelta di diverse metriche ci ha permesso di valutare le prestazioni del modello centralizzato e del modello federato da diverse prospettive. Per quanto riguarda la loss function, è stata selezionata la *binary cross-entropy*, poiché il task di classificazione è di natura binaria.

### D. Esperimenti

- **Confronto 1.** I seguenti grafici mostrano, per ogni metrica considerata, la convergenza degli algoritmi, fissando di volta in volta la strategia adottata e variando il numero di clients nella simulazione.

Il fattore comune che si nota nei grafici di tutte le metriche è che l'aumento nel numero di clients comporta un rallentamento nella convergenza dell'algoritmo, seppur minima dato il range del numero di clients considerati. Tuttavia, nella maggioranza dei casi la differenza nelle metriche calcolate nel modello federato e in quello centralizzato tende a diminuire molto velocemente con l'avanzare dei round.

Per quanto riguarda il confronto con il modello centralizzato - che come ricordiamo, ha la stessa configurazione di iperparametri del modello federato - si nota come per la totalità delle metriche, il modello centralizzato offre una convergenza migliore (anche se spesso in misura minima) rispetto alle varie configurazioni di modello federato. Andando nello specifico delle varie metriche,

- **Accuracy:** le tre strategie hanno una convergenza molto simile tra loro, con le strategie FedAvg e QFedAvg che tendono a comportarsi leggermente meglio rispetto alla strategia FedProx, in termini anche di differenza dal modello centralizzato;
- **Precision:** le tre strategie hanno una convergenza quasi identica tra loro, ed inoltre mostrano un andamento quasi sovrapposto a quello del modello centralizzato, nonostante il numero variabile di clients;
- **Recall:** anche per questa metrica l'andamento sembra essere molto simile a quello del modello centralizzato per tutte e tre le strategie; si nota ancora

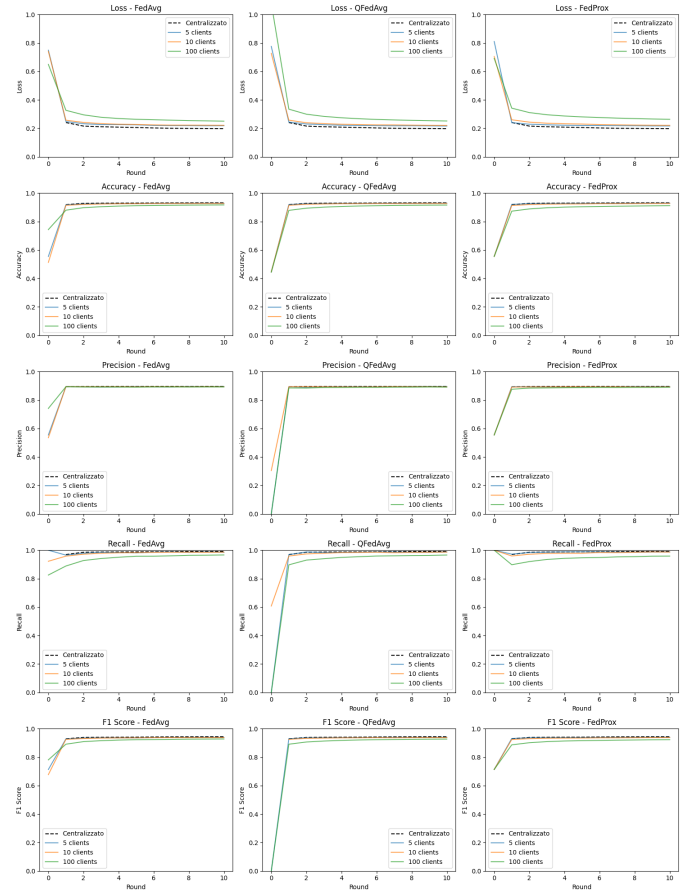


Figura 1. Task 1 - Confronto 1

una volta una differenza significativa nei valori di Recall della simulazione con 100 clients, il che dimostra un peggioramento - anche se minimo - delle performance all'aumentare del numero di clients;

- **F1 Score:** questa metrica mostra una convergenza regolare per tutte e tre le strategie, che tende a scostarsi in minima misura dai risultati del modello centralizzato. Quello che si nota è che tra le tre strategie, QFedAvg sembra leggermente quella più robusta all'aumento del numero di client.

Per quanto riguarda la loss, la convergenza ha circa lo stesso andamento delle metriche sopra descritte: all'aumentare del numero di clients, i valori della loss function risultano più alti, ed anche la convergenza risulta più lenta. Come atteso, i valori di loss dei modelli federati risultano in ogni caso peggiori rispetto a quelli ottenuti dal modello centralizzato.

In generale, i valori migliori raggiunti per ciascuna metrica tendono a peggiorare all'aumentare del numero di clients.

- **Confronto 2.** I seguenti grafici mostrano, per ogni metrica, la convergenza al variare della strategia, fissato il

numero di clients. Questo secondo confronto consente di

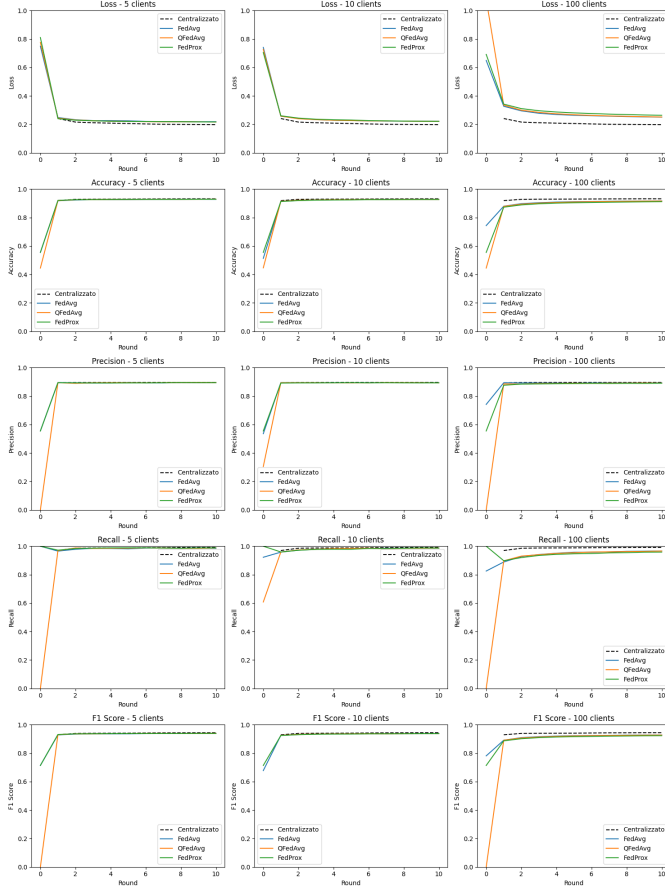


Figura 2. Task 1 - Confronto 2

notare come, a parità di numero di client, si comportano le varie strategie in rapporto al modello centralizzato. Un fatto che si nota immediatamente è che le configurazioni con 5 e 10 client consentono di avere valori molto prossimi a quelli ottenuti dal modello centralizzato; al contrario, le configurazioni con 100 clients restituiscono valori delle metriche che in molti casi si discostano di una quantità maggiore dai valori del modello centralizzato (molto evidente nella loss function, ad esempio), indipendentemente dal tipo di strategia adottata. Dunque, emerge che le strategie differiscono maggiormente nei primi rounds, mentre successivamente tendono ad avere prestazioni più o meno uniformi.

La seguenti tabelle riportano i risultati per le metriche di accuracy, precision, recall ed F1 score, nel caso del modello centralizzato e nel caso del modello federato, al variare del numero di clients e strategia adottata.

## VII. TASK 2:

### A. Dataset

In secondo luogo abbiamo selezionato un task di regressione, lavorando con il dataset *SGEMM GPU kernel performance* [2], che raccoglie il running time del prodotto matrice-matrice ( $A*B = C$ ) di taglia 2048 x 2048, usando un SGEMM GPU kernel parametrizzabile, con 1327104 possibili combinazioni, di cui solo 241600 feasible (due to various kernel constraints). Per ciascuna combinazione testata, vengono performate 4 run ed i loro risultati sono riportati nelle ultime 4 colonne ('RunX'). Tutti i tempi sono riportati in millisecondi.

Poichè il dataset consiste per la maggior parte di feature intere e le restanti categoriche sono binarie, la manipolazione dei dati è stata limitata a due step:

- aggregazione dei quattro risultati in un'unica feature 'avg\_runs';
- shuffle;
- normalizzazione.

### B. Modello

Il modello adottato è keras.Sequential con 3 livelli densi, il primo di 128 unità con input\_shape (14,), il secondo con 128 unità, ed infine l'output layer con 1 unità; tutti i layers adottano come funzione di attivazione 'sigmoid'.

Abbiamo adottato 'MAE' come loss function poichè di facile comprensione e interpretazione: rappresenta la magnitudine media degli errori commessi dal modello; è la differenza assoluta media fra le predizioni e gli actual values. Oltre all'interpretabilità, le caratteristiche per cui selezionarla sono la robustezza agli outliers, che nel caso del nostro dataset non è una problematica evidente, ma contribuisce comunque alla stabilità del modello, e pesare egualmente tutti gli errori a prescindere dalla loro magnitudine.

### C. Metriche

In ogni caso, è fondamentale cogliere le caratteristiche del problema e gli obiettivi in analisi, con l'osservazione di più metriche. Abbiamo deciso di adottare alcune fra le metriche più diffuse per la regressione che Keras mette a disposizione:

- MSE;
- MAPE;
- MSLE;
- R2\_Score.

Mean Squared Error (MSE) è più sensibile agli outliers, ma come previsto non aggiunge dettagli alle osservazioni che si possono fare guardando al mae; discorso equivalente vale per Mean Squared Logarithmic Error (MSLE) solitamente utilizzato nel caso di dati con un grande range di valori, e gli errori relativi sono più importanti di quelli assoluti.

**MAPE** [3], indica in percentuale la deviazione media dei valori predetti rispetto agli actual values, ed è la metrica in cui abbiamo notato più oscillazioni. Infine **R2**, il coefficiente di determinazione, che misura il legame tra la variabilità dei dati e la correttezza del modello statistico utilizzato. Utilizzato per mostrare la frazione della varianza non spiegata dal modello.

### D. Esperimenti

- **Confronto 1.** I seguenti grafici mostrano la convergenza per ogni metrica considerata, fissata la strategia e variando il numero di clients.

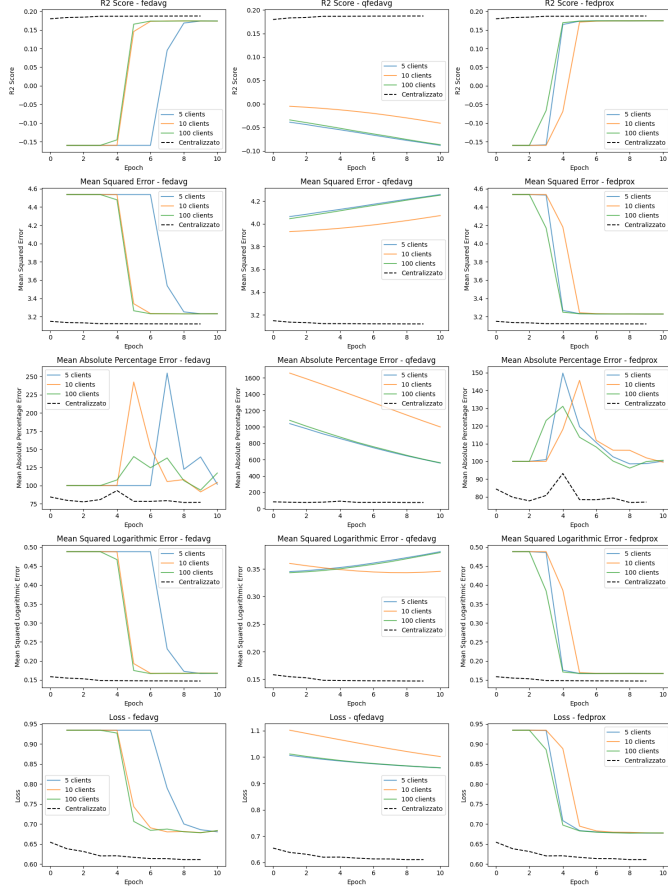


Figura 3. Task 2 - Confronto 1

- **R2 Score:** FedAvg e FedProx si comportano in modo molto simile, anche se la prima converge più lentamente nel caso con 5 clients; FedQAvg ha un andamento completamente diverso.
- **mse:** anche per il mse il comportamento è lo stesso appena descritto, ed addirittura risulta incrementare con la strategia FedQAvg.
- **MAPE:** metrica che in generale ha denotato maggiore varianza nelle simulazioni, sembra denotare leggere variazioni anche nel caso centralizzato, che ovviamente sono accentuati nel caso federato; FedAvg e FedProx continuano a comportarsi come per le metriche sopracitate, anche se stavolta FedAvg si avvicina maggiormente al modello centralizzato.

- **msle:** discorso analogo alle prime due metriche considerate.
- **Loss:** anche per la loss notiamo lo stesso andamento visto in precedenza.

Risalta immediatamente dai grafici come la strategy FedQAvg presenti convergenze completamente differenti, su ogni metrica, rispetto alla altre due presentate. In generale vediamo come FedProx è più stabile e mantiene prestazioni molto simili anche al variare del numero di clients; invece, vediamo invece come FedAvg converge più lentamente nel caso di 5 clients. Come risultante anche dagli altri task e dataset considerati, gli esperimenti confermano che il caso centralizzato converge molto più velocemente rispetto al modello federato.

- **Confronto 2.** I seguenti grafici mostrano la convergenza per ogni metrica, fissato il numero di clients e variando la strategia adottata.

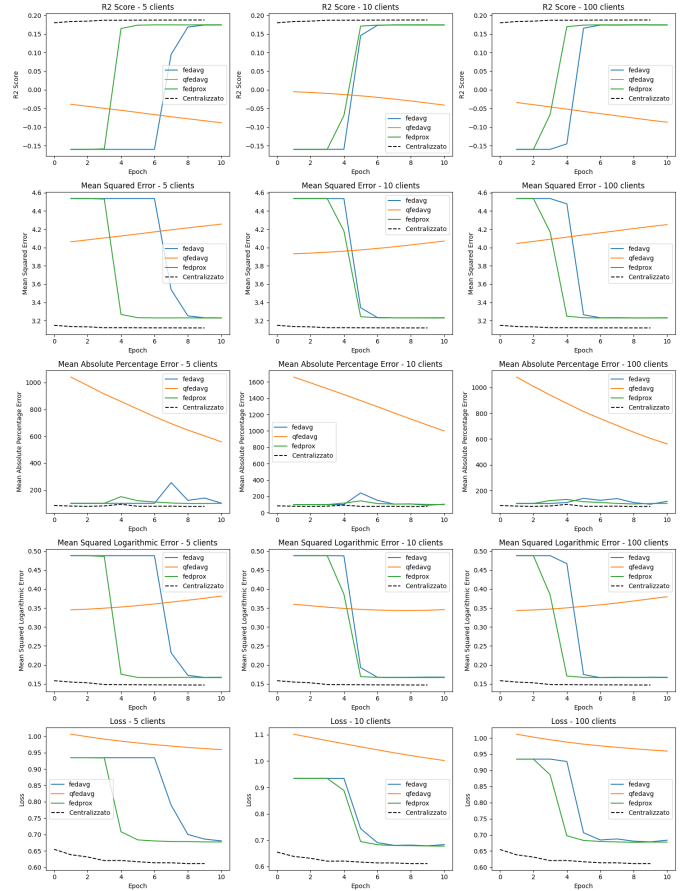


Figura 4. Task 2 - Confronto 2

- **R2 Score:** FedAvg e FedProx si comportano in modo molto simile, evidenziando maggiore distacco nel caso con 5 clients.
- **mse:** anche per il mse il comportamento è lo stesso appena descritto.
- **MAPE:** risulta molto evidente in questo caso come la strategia FedQAvg allontani il modello federato

dalle prestazioni di quello centralizzato.

- **msle**: discorso analogo alle prime due metriche considerate.
- **Loss**: anche per la loss notiamo lo stesso andamento visto in precedenza.

Il confronto fissando il numero di clients conferma FedProx come strategia preferibile in quanto permette nella maggior parte dei casi raggiungere la convergenza anche un paio di epoche prima delle altre. Notiamo anche come nel caso centrale di 10 clients le strategie FedAvg e FedProx si comportino pressochè allo stesso modo. In tutti i casi vediamo come solamente la strategia FedQAvg presenta prestazioni notevolmente peggiori, mentre le altre si avvicinano sempre al caso centralizzato anche se più o meno lentamente.

### VIII. TASK 3:

#### A. Dataset

Per quanto riguarda il terzo ed ultimo task, abbiamo optato per un task di classificazione multiclasse di immagini. In particolare, abbiamo selezionato il dataset MNIST, una vasta base di dati di cifre scritte a mano che è comunemente impiegata come insieme di addestramento in vari sistemi per l'elaborazione delle immagini. La scelta di tale dataset è stata in parte dettata dalla comodità, in quanto come riportato in precedenza nella relazione, il framework Flower, di base, si aspetta in input un dataset composto da una sola feature (in questo caso, l'immagine di ogni istanza), e tale dataset è tra i pochi inclusi nativamente nell'ambiente di sviluppo Flower.

#### B. Modello

Per quanto riguarda il modello scelto per questo task di classificazione multiclasse è stata una Rete Neurale Convolutionale, il cui tuning degli iperparametri è stato eseguito in maniera analoga ai task precedenti. In seguito al tuning, il modello risultante è stato il seguente:

Livello	Unità	Numero Filtri	Attivazione
Conv2D	-	1 (Kernel: 5)	ReLU
MaxPooling2D	-	-	-
Conv2D	-	48 (Kernel: 5)	ReLU
MaxPooling2D	-	-	-
Flatten	-	-	-
Dense	64	-	PReLU
Dense	80	-	ReLU
Dense	32	-	PReLU
Dense	10	-	Softmax

La loss function considerata è *(sparse) categorical crossentropy* [4], adottata nei proprio per i dataset con due o più classi di etichette. *Sparse* poichè prevede di ricevere le etichette sotto forma di interi.

#### C. Metriche

Le metriche considerate sono le seguenti:

- **(sparse) categorical accuracy**: questa metrica calcola la accuracy del modello nella classificazione delle etichette

target sparse, ovvero quando le etichette sono rappresentate come interi anziché come codifica one-hot. Ad esempio, se ci sono 3 classi e l'etichetta per un certo campione è 2, allora questa etichetta è rappresentata come un singolo intero anziché un vettore  $[0, 0, 1]$ . La "sparse categorical accuracy" confronta le predizioni del modello con queste etichette sparse. Un'etichetta è considerata predetta correttamente se l'indice della classe con la probabilità più alta coincide con l'indice dell'etichetta target. La metrica calcola quindi la percentuale di campioni correttamente classificati rispetto al totale dei campioni.

- **(sparse) top k categorical accuracy**: Questa metrica è simile alla "sparse categorical accuracy", ma invece di considerare solo la classe con la probabilità più alta, tiene conto delle prime k classi con le probabilità più alte. Se l'etichetta target è tra queste prime k classi, viene considerata una previsione corretta. Questa metrica è utile quando è di interesse sapere se il modello ha predetto correttamente l'etichetta target tra le prime k probabilità più alte, anche se non necessariamente come la predizione più probabile.

#### D. Esperimenti

- **Confronto 1**. I seguenti grafici mostrano, per ogni metrica considerata, la convergenza degli algoritmi, fissando di volta in volta la strategia adottata e variando il numero di clients nella simulazione.

Il pattern osservato ricalca quello individuato negli altri due task: in linea generale, l'aumento nel numero di clients nella simulazione comporta un degrado delle prestazioni.

- **Categorical Accuracy**: per quanto riguarda questa metrica, le strategie FedAvg e QFedAvg confermano il pattern osservato, eccezion fatta per FedAvg in cui nei primi round la configurazione con 100 clients raggiunge valori lievemente migliori rispetto a quella con 10 clients. Di nuovo, la situazione è diversa per FedProx, in quanto l'aumento nel numero di clients porta a valori migliori della metrica, soprattutto nei primi round.
- **Top k categorical accuracy**: un comportamento quasi analogo si ha per questa metrica, sebbene le differenze tra le varie configurazioni siano meno evidenti.
- **Loss**: per quanto riguarda la loss function, la situazione risulta meno delineata nel caso di FedAvg, in cui non si evidenzia un trend particolare tra le varie configurazioni. QFedAvg conferma il degrado nelle prestazioni all'aumento del numero di clients, mentre FedProx di nuovo mostra un comportamento opposto alle altre due strategie, evidenziando una convergenza migliore, soprattutto nei primi round nel caso di più clients.

Per tutte le metriche considerate, indipendentemente dal numero di clients, il modello centralizzato mostra valori nettamente migliori al modello federato. Per di più,

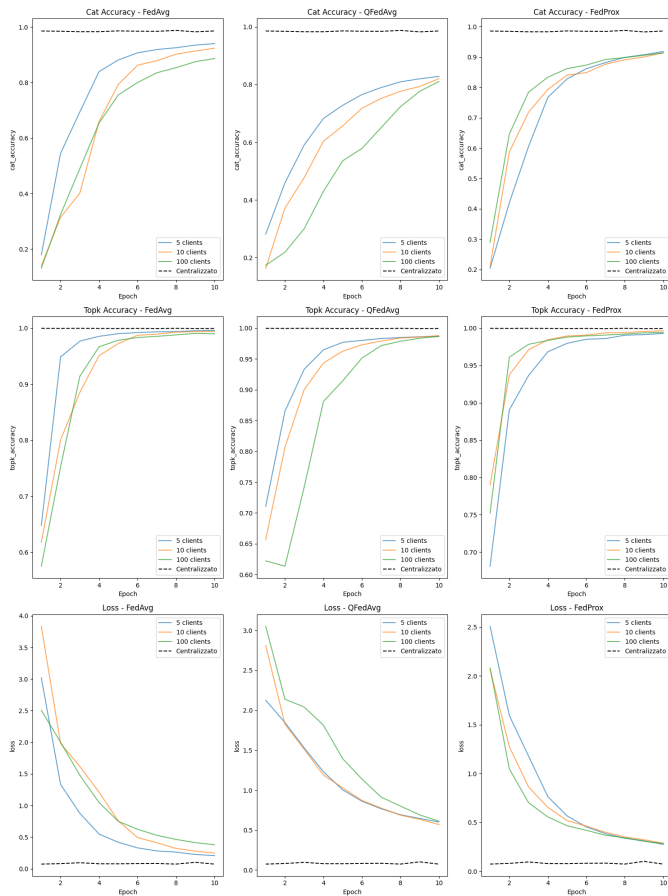


Figura 5. Task 3 - Confronto 1

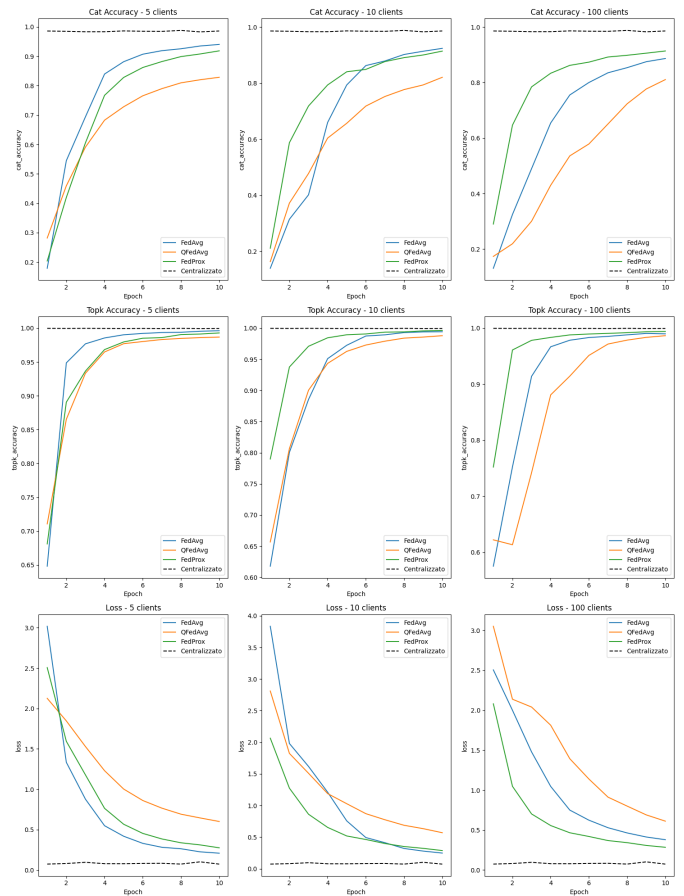


Figura 6. Task 3 - Confronto 2

non si nota una particolare convergenza per il modello centralizzato, bensì quest'ultimo si assesta sin dalle prime epoche su un range di valori molto ristretto.

- **Confronto 2.** I seguenti grafici mostrano ogni metrica, al variare della strategia, fissato di volta in volta il numero di clients.

- **Categorical accuracy:** per questa metrica si presenta una situazione analoga alla precedente, in cui la strategia QFedAvg si dimostra essere la peggiore in termini di convergenza. All'aumentare del numero di clients, la strategia FedProx sembra prevalere su FedAvg, dimostrandosi più robusta.
- **Top k categorical accuracy:** per questa metrica, indipendentemente dal numero di clients considerato, tutte e tre le strategie si dimostrano essere abbastanza robuste all'aumentare del numero di clients, eccetto un rallentamento nella convergenza per i primi round.
- **Loss:** per quanto riguarda la loss function, ancora una volta per tutte e tre le configurazioni di clients, la strategia QFedAvg si dimostra essere la peggiore tra le tre testate. D'altra parte, le strategie FedAvg e FedProx mostrano un andamento abbastanza simile, anche se all'aumentare del numero di clients la differenza tra le due si rende più evidente, mostrando

come FedProx risulti più robusta.



## IX. TABELLE RISULTATI

Di seguito sono riportati in forma tabulare i risultati ottenuti dalle simulazioni nei vari task.

Tabella I

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO CENTRALIZZATO

Epoch	Loss	Accuracy	Precision	Recall	F1 Score
1	0.240955	0.918867	0.893496	0.969436	0.929555
2	0.216137	0.92796	0.895164	0.985695	0.938858
3	0.21159	0.929132	0.895696	0.987344	0.939948
4	0.208675	0.929447	0.895846	0.987779	0.940416
5	0.206116	0.929787	0.896032	0.988213	0.94091
6	0.203009	0.9304	0.896012	0.989489	0.941997
7	0.200668	0.930966	0.896072	0.990558	0.943037
8	0.19951	0.931222	0.895956	0.991243	0.943344
9	0.198446	0.931643	0.896326	0.991583	0.943882
10	0.197927	0.931578	0.896235	0.991578	0.943862

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.726965	44.6497	90	0.304927	0.607795
1	0.257616	91.4105	89.3963	95.8986	92.5334
2	0.23965	92.1508	89.4216	97.3769	93.2299
3	0.233383	92.3715	89.3946	97.8655	93.4384
4	0.228958	92.5467	89.3254	98.3202	93.6072
5	0.226568	92.5971	89.3062	98.4504	93.6556
6	0.224671	92.6842	89.3214	98.6073	93.735
7	0.224338	92.6268	89.5056	98.2328	93.6664
8	0.222689	92.7129	89.4577	98.4754	93.7502
9	0.221333	92.7298	89.435	98.5413	93.7676
10	0.220905	92.7139	89.3152	98.6769	93.7629

Tabella II

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO  
STRATEGIA QFEDAVG E 10 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.740793	0.513252	0.535719	0.922288	0.677757
1	0.257336	0.914234	0.894733	0.958202	0.925381
2	0.241842	0.9203	0.893992	0.971611	0.931187
3	0.233606	0.923517	0.893642	0.978673	0.934226
4	0.229075	0.92511	0.893843	0.981651	0.935691
5	0.227176	0.925635	0.893577	0.983095	0.936201
6	0.224487	0.926872	0.894123	0.984861	0.937301
7	0.222802	0.927129	0.893977	0.985592	0.937551
8	0.22273	0.926931	0.893649	0.985645	0.937396
9	0.222207	0.927129	0.894334	0.985092	0.937522
10	0.221532	0.927248	0.893806	0.986073	0.937675

Tabella III

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON  
STRATEGIA FEDAVG E 10 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.704966	0.555007	0.555007	1	0.713832
1	0.260447	0.91267	0.891501	0.959414	0.924212
2	0.243967	0.919271	0.892649	0.971362	0.930343
3	0.236263	0.922339	0.892718	0.97755	0.93321
4	0.23259	0.923398	0.893416	0.978744	0.934136
5	0.230488	0.923962	0.8944	0.97853	0.934575
6	0.226525	0.925605	0.893036	0.983791	0.936219
7	0.224931	0.925813	0.893735	0.983238	0.936353
8	0.223162	0.926466	0.893072	0.985503	0.937014
9	0.222039	0.927208	0.892707	0.987535	0.93773
10	0.22141	0.927159	0.893181	0.986769	0.937645

Tabella IV

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON  
STRATEGIA FEDPROX E 10 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.691752	0.554997	0.555002	0.999982	0.713824
1	0.342165	0.872894	0.876175	0.897876	0.886893
2	0.311003	0.888779	0.884275	0.920006	0.901786
3	0.295854	0.896835	0.885619	0.93486	0.909574
4	0.287018	0.901457	0.886943	0.942599	0.913924
5	0.280672	0.904079	0.887858	0.946754	0.91636
6	0.276312	0.90597	0.889383	0.948555	0.918016
7	0.271998	0.907642	0.888593	0.953084	0.91971
8	0.268997	0.909107	0.889603	0.954707	0.921006
9	0.266026	0.910572	0.889487	0.957881	0.922418
10	0.26384	0.911631	0.890599	0.958523	0.923313

Tabella V

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON  
STRATEGIA FEDPROX E 100 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	1.07961	0.444993	0	0	0
1	0.335088	0.878447	0.885422	0.897074	0.89121
2	0.29963	0.894004	0.884646	0.93033	0.906913
3	0.284216	0.901308	0.888453	0.940227	0.913607
4	0.274498	0.906059	0.888932	0.949357	0.918151
5	0.26788	0.909275	0.890127	0.954332	0.921112
6	0.262752	0.911334	0.889847	0.958951	0.923107
7	0.259357	0.912858	0.891056	0.960413	0.924436
8	0.256327	0.914026	0.891294	0.962481	0.925521
9	0.254125	0.915154	0.892339	0.963355	0.926488
10	0.251894	0.915886	0.891489	0.96603	0.927264

Tabella VI

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON  
STRATEGIA QFEDAVG E 100 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.64946	0.743572	0.741681	0.825478	0.781339
1	0.327015	0.879773	0.893894	0.888889	0.891384
2	0.295169	0.897557	0.892205	0.927477	0.909499
3	0.278108	0.904317	0.89215	0.941404	0.916116
4	0.269274	0.908682	0.891988	0.95057	0.920348
5	0.26394	0.911423	0.891093	0.957417	0.923065
6	0.26076	0.91266	0.892671	0.957792	0.924085
7	0.257591	0.913907	0.8927	0.960306	0.92527
8	0.254548	0.915322	0.892097	0.964033	0.926671
9	0.252757	0.916104	0.892678	0.964835	0.927355
10	0.250679	0.916807	0.892493	0.966529	0.928037

Tabella VII

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON  
STRATEGIA FEDAVG E 100 CLIENTS



Round	Loss	Accuracy (%)	Precision	Recall	F1_score
0	0.77594	0.444993	0	0	0
1	0.24427	0.919024	0.894182	0.968741	0.929969
2	0.230825	0.9251	0.88964	0.987553	0.936043
3	0.225774	0.9261	0.893594	0.984023	0.936631
4	0.22208	0.927248	0.894291	0.985396	0.937635
5	0.220977	0.927456	0.894416	0.985645	0.937817
6	0.219482	0.927931	0.893404	0.988035	0.938339
7	0.218376	0.928307	0.894524	0.987232	0.938595
8	0.218701	0.927802	0.895512	0.984825	0.938047
9	0.21799	0.92802	0.895896	0.984736	0.938217
10	0.216447	0.928822	0.894868	0.987803	0.939042

Tabella VIII

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA QFedAVG E 5 CLIENTS

round	r2_prox	mse_prox	mape_prox	msle_prox	loss_prox
1	-0.159773	4.536204	99.999832	0.487908	0.93444
2	-0.159771	4.536196	100.002647	0.487904	0.93444
3	-0.158326	4.530542	100.980331	0.485681	0.933808
4	0.164455	3.268056	149.804108	0.175392	0.708424
5	0.173493	3.232704	119.574417	0.166720	0.683495
6	0.174212	3.229892	110.613647	0.166439	0.680148
7	0.174251	3.229741	102.588463	0.166694	0.678648
8	0.174063	3.230473	98.588257	0.167074	0.678383
9	0.174425	3.229059	98.732056	0.166520	0.677230
10	0.174464	3.228906	100.152863	0.166480	0.677253

Tabella XII

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDPROX E 5 CLIENTS

Round	Loss	Accuracy (%)	Precision	Recall	F1_score
0	0.749308	0.555007	0.555007	1	0.713832
1	0.248058	0.917866	0.894954	0.965317	0.928805
2	0.23259	0.923359	0.893691	0.978281	0.934075
3	0.226495	0.925833	0.892251	0.98536	0.936497
4	0.226579	0.925526	0.892781	0.983987	0.936168
5	0.225163	0.925932	0.895152	0.981508	0.936343
6	0.220737	0.927278	0.893303	0.98684	0.937745
7	0.219963	0.927664	0.892673	0.988516	0.938153
8	0.21973	0.927753	0.895439	0.984825	0.938007
9	0.218632	0.928178	0.893916	0.987821	0.938526
10	0.219088	0.927684	0.895248	0.98495	0.937959

Tabella IX

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDAVG E 5 CLIENTS

round	r2_qavg	mse_qavg	mape_qavg	msle_qavg	loss_qavg
1	-0.038929	4.063546	1040.778809	0.345072	1.006306
2	-0.044094	4.083748	978.655701	0.346872	0.998679
3	-0.049749	4.105868	915.088745	0.349496	0.991300
4	-0.055025	4.126504	858.891418	0.352478	0.985156
5	-0.060640	4.148464	802.461853	0.356186	0.979595
6	-0.066540	4.171540	746.391968	0.360625	0.974560
7	-0.072246	4.193858	694.489624	0.365406	0.970081
8	-0.077753	4.215397	646.320923	0.370449	0.966047
9	-0.082950	4.235724	602.545837	0.375580	0.962494
10	-0.088348	4.256840	558.664307	0.381279	0.959051

Tabella XIII

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDQAVG E 5 CLIENTS

Round	Loss	Accuracy	Precision	Recall	F1_score
0	0.810259	0.555007	0.555007	1	0.713832
1	0.240329	0.920637	0.894038	0.972236	0.931498
2	0.229183	0.925368	0.892931	0.983452	0.936008
3	0.225238	0.92613	0.892126	0.986145	0.936782
4	0.221867	0.92709	0.892042	0.988231	0.937676
5	0.22051	0.927654	0.892507	0.98873	0.938158
6	0.220283	0.927792	0.892759	0.988659	0.938265
7	0.219512	0.927624	0.893782	0.986876	0.938025
8	0.218889	0.928	0.895238	0.98561	0.938253
9	0.217379	0.928535	0.89464	0.987535	0.938795
10	0.216915	0.928733	0.894559	0.988053	0.938984

Tabella X

TASK1: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDPROX E 5 CLIENTS

round	r2_qavg	mse_qavg	mape_qavg	msle_qavg	loss_qavg
1	-0.159768	4.536181	100.008499	0.487897	0.93444
2	-0.159773	4.536201	100.000008	0.487907	0.93444
3	-0.159770	4.536192	100.004105	0.487902	0.93444
4	-0.159465	4.534999	100.231705	0.487408	0.934316
5	0.145597	3.341815	242.549881	0.192786	0.743602
6	0.172998	3.234639	152.504593	0.167351	0.690544
7	0.173969	3.230843	105.542000	0.166795	0.680298
8	0.173977	3.230810	108.258484	0.167205	0.681332
9	0.173879	3.231196	91.323975	0.167707	0.678554
10	0.173640	3.232130	104.274834	0.167719	0.683049

Tabella XIV

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDAVG E 10 CLIENTS

round	r2_avg	mse_avg	mape_avg	msle_avg	loss_avg
1	-0.159772	4.536199	100.001656	0.487906	0.93444
2	-0.159774	4.536206	99.999771	0.48791	0.93444
3	-0.159774	4.536205	99.999786	0.48791	0.93444
4	-0.159772	4.5362	100.000877	0.487907	0.93444
5	-0.159772	4.536198	100.001709	0.487906	0.93444
6	-0.159754	4.536129	100.024345	0.487872	0.934437
7	0.095176	3.539023	254.785141	0.231574	0.789448
8	0.16845	3.252428	122.435867	0.17246	0.700098
9	0.173803	3.231492	139.457962	0.166702	0.685768
10	0.173894	3.231137	101.962616	0.16725	0.680446

Tabella XI

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDAVG E 5 CLIENTS

round	r2_prox	mse_prox	mape_prox	msle_prox	loss_prox
1	-0.159773	4.536201	100.000259	0.487907	0.93444
2	-0.159773	4.536201	100.000587	0.487907	0.93444
3	-0.159766	4.536175	100.009506	0.487894	0.934439
4	-0.068974	4.181060	118.042313	0.386204	0.88839
5	0.170903	3.242836	145.703079	0.169003	0.694669
6	0.173511	3.232634	111.831459	0.167015	0.682743
7	0.174152	3.230126	106.305733	0.166629	0.679604
8	0.174308	3.229515	106.238319	0.166608	0.679228
9	0.174417	3.229089	101.954605	0.166529	0.677882
10	0.174371	3.229271	99.536942	0.166657	0.677419

Tabella XV

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDPROX E 10 CLIENTS

round	r2_qavg	mse_qavg	mape_qavg	msle_qavg	loss_qavg	Epoch	Val Loss	Val Acc	Val Cat Acc	Val Top-K Cat Acc
1	-0.005093	3.931203	1658.599976	0.359964	1.101884	0	0.047182	0.985417	0.985417	0.999583
2	-0.007134	3.939188	1588.970215	0.355774	1.089777	1	0.051051	0.984000	0.984000	0.999583
3	-0.009641	3.948995	1517.632446	0.352034	1.077690	2	0.040717	0.988750	0.988750	0.999833
4	-0.012628	3.960676	1445.581665	0.348873	1.065847	3	0.041703	0.987917	0.987917	0.999667
5	-0.016163	3.974501	1371.580688	0.346307	1.054089	4	0.046178	0.987083	0.987083	0.999667
6	-0.020172	3.990185	1297.480347	0.344467	1.042741	5	0.050389	0.987167	0.987167	0.999833
7	-0.024852	4.008488	1220.603516	0.343398	1.031461	6	0.037293	0.990167	0.990167	0.999833
8	-0.029889	4.028190	1145.836914	0.343219	1.020971	7	0.044297	0.988083	0.988083	0.999833
9	-0.035364	4.049602	1071.925171	0.343951	1.011085	8	0.051666	0.989417	0.989417	0.999833
10	-0.041095	4.072019	1000.736694	0.345560	1.002041	9	0.075779	0.984917	0.984917	0.999500

Tabella XVI

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDQAVG E 10 CLIENTS

Tabella XX

TABELLA DELLE METRICHE PER OGNI EPOCA

round	r2_avg	mse_avg	mape_avg	msle_avg	loss_avg
1	-0.159755	4.536134	100.026421	0.487875	0.93444
2	-0.159772	4.536198	100.001831	0.487905	0.93444
3	-0.159762	4.536160	100.014900	0.487887	0.934439
4	-0.145003	4.478434	107.778694	0.466935	0.927268
5	0.165347	3.264564	139.972153	0.174531	0.706536
6	0.173709	3.231861	124.542587	0.166392	0.684164
7	0.173527	3.232573	137.992813	0.167373	0.687349
8	0.174131	3.230210	106.839127	0.166867	0.680561
9	0.173989	3.230766	93.856712	0.167519	0.678331
10	0.173803	3.231493	117.283958	0.167247	0.683554

Tabella XVII

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDAVG E 100 CLIENTS

Tabella XXI  
TASK3: RISULTATI DELLE METRICHE PER IL MODELLO CENTRALIZZATO

Epoch	cat_accuracy	topk_accuracy	loss
1	0.1308	0.5753	2.505817
2	0.3246	0.7525	2.000379
3	0.4898	0.9141	1.478582
4	0.6543	0.9668	1.046845
5	0.7552	0.9785	0.750211
6	0.8001	0.9833	0.623913
7	0.8352	0.9854	0.529405
8	0.8536	0.9881	0.464159
9	0.8750	0.9906	0.413245
10	0.8866	0.9899	0.379287

round	r2_prox	mse_prox	mape_prox	msle_prox	loss_prox
1	-0.159760	4.536153	100.019691	0.487883	0.934439
2	-0.159761	4.536155	100.016273	0.487885	0.934438
3	-0.066254	4.170422	123.066887	0.384461	0.885895
4	0.169462	3.248470	131.076019	0.170648	0.697338
5	0.173644	3.232113	113.603127	0.166872	0.682718
6	0.174282	3.229618	108.092812	0.166477	0.679581
7	0.174300	3.229549	100.192329	0.166766	0.678160
8	0.174375	3.229254	96.198891	0.166825	0.677046
9	0.174396	3.229172	99.913826	0.166705	0.677745
10	0.174443	3.228990	100.616287	0.166603	0.677719

Tabella XVIII

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDPROX E 100 CLIENTS

Tabella XXII  
TASK3: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGY FEDPROX E 100 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.2901	0.7522	2.081158
2	0.6465	0.9611	1.047874
3	0.7844	0.9783	0.701940
4	0.8335	0.9835	0.556559
5	0.8619	0.9880	0.466229
6	0.8735	0.9896	0.419731
7	0.8920	0.9907	0.368518
8	0.8982	0.9920	0.342408
9	0.9058	0.9936	0.307516
10	0.9137	0.9942	0.284420

round	r2_qavg	mse_qavg	mape_qavg	msle_qavg	loss_qavg
1	-0.034129	4.044774	1080.976440	0.343173	1.011565
2	-0.039911	4.067389	1009.415894	0.344734	1.002501
3	-0.045783	4.090356	942.036987	0.347087	0.994473
4	-0.051844	4.114060	877.082764	0.350228	0.987184
5	-0.058080	4.138453	814.192810	0.354128	0.980678
6	-0.063909	4.161250	758.587280	0.358335	0.975573
7	-0.069688	4.183853	706.047241	0.363008	0.970987
8	-0.075778	4.207674	653.270447	0.368445	0.966562
9	-0.081590	4.230405	604.900879	0.374091	0.962633
10	-0.086870	4.251060	562.371094	0.379593	0.959283

Tabella XIX

TASK2: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGIA FEDQAVG E 100 CLIENTS

Tabella XXIII  
TASK3: RISULTATI DELLE METRICHE PER IL MODELLO FEDERATO CON STRATEGY QFEDAVG E 100 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.1736	0.6223	3.052144
2	0.2191	0.6136	2.138689
3	0.3004	0.7422	2.040695
4	0.4299	0.8810	1.813220
5	0.5358	0.9144	1.393641
6	0.5787	0.9514	1.139368
7	0.6509	0.9717	0.912194
8	0.7234	0.9787	0.800335
9	0.7767	0.9836	0.690224
10	0.8106	0.9866	0.611684

Tabella XXIV

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY FEDAVG E 100 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.1396	0.6178	3.834348
2	0.3144	0.8005	1.981152
3	0.4015	0.8854	1.618643
4	0.6595	0.9510	1.211388
5	0.7933	0.9725	0.757223
6	0.8624	0.9874	0.493698
7	0.8788	0.9893	0.412210
8	0.9020	0.9929	0.320931
9	0.9135	0.9940	0.279641
10	0.9238	0.9942	0.248362

Tabella XXV

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY FEDPROX E 10 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.2111	0.7899	2.065002
2	0.5874	0.9376	1.274748
3	0.7182	0.9709	0.862452
4	0.7933	0.9845	0.654375
5	0.8403	0.9894	0.518748
6	0.8488	0.9907	0.462882
7	0.8767	0.9936	0.398931
8	0.8907	0.9940	0.352000
9	0.9002	0.9956	0.322833
10	0.9140	0.9964	0.288012

Tabella XXVI

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY QFEDAVG E 10 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.1633	0.6567	2.810671
2	0.3720	0.8066	1.824364
3	0.4776	0.9003	1.510225
4	0.6038	0.9434	1.189084
5	0.6566	0.9627	1.028795
6	0.7180	0.9729	0.871393
7	0.7522	0.9791	0.775111
8	0.7768	0.9840	0.687708
9	0.7934	0.9857	0.634439
10	0.8206	0.9879	0.569984

Tabella XXVII

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY FEDAVG E 5 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.1791	0.6482	3.018399
2	0.5450	0.9487	1.334101
3	0.6940	0.9768	0.877807
4	0.8395	0.9855	0.549155
5	0.8814	0.9902	0.418776
6	0.9069	0.9922	0.330590
7	0.9189	0.9935	0.281983
8	0.9254	0.9939	0.263710
9	0.9348	0.9954	0.225585
10	0.9402	0.9962	0.209047

Tabella XXVIII

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY FEDPROX E 5 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.2039	0.6811	2.506892
2	0.4199	0.8906	1.593910
3	0.6056	0.9365	1.178540
4	0.7671	0.9683	0.764252
5	0.8279	0.9796	0.567366
6	0.8617	0.9851	0.453980
7	0.8819	0.9860	0.385182
8	0.8988	0.9905	0.337637
9	0.9077	0.9913	0.311508
10	0.9183	0.9929	0.275550

Tabella XXIX

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
FEDERATO CON STRATEGY QFEDAVG E 5 CLIENTS

Epoch	cat_accuracy	topk_accuracy	loss
1	0.2820	0.7107	2.126181
2	0.4595	0.8653	1.846541
3	0.5911	0.9333	1.529425
4	0.6826	0.9648	1.230552
5	0.7282	0.9770	1.003095
6	0.7655	0.9802	0.861222
7	0.7898	0.9832	0.766520
8	0.8096	0.9848	0.691888
9	0.8200	0.9860	0.645418
10	0.8285	0.9868	0.601402

Tabella XXX

TASK3: RISULTATI DELLE METRICHE PER IL MODELLO  
CENTRALIZZATO

Epoch	cat_accuracy	topk_accuracy	loss
1	0.074014	0.985750	0.999667
2	0.080995	0.984750	0.999500
3	0.095862	0.983250	0.999667
4	0.078904	0.983333	0.999500
5	0.078506	0.986083	0.999333
6	0.082024	0.984917	0.999500
7	0.083500	0.984583	0.999250
8	0.074037	0.987667	0.999667
9	0.101887	0.982833	0.999250
10	0.073561	0.985833	0.999417

## X. CONCLUSIONI

Durante il lavoro, abbiamo riscontrato diverse difficoltà nell'orientarci all'utilizzo del framework, dovendo spesso risolvere i problemi implementativi in maniera ad-hoc. La comodità principale è stata poter simulare ogni esperimento interamente su Google Colab, avendo fortunatamente a disposizione le librerie necessarie e potenza di calcolo (GPU) per rendere le simulazioni più veloci. Le potenzialità di Flower risiedono nella customizzazione di client e server, tuttavia si è vincolati alle strutture dati utilizzate ed ai pochi esempi documentati, i quali vertono su un insieme ristretto di dataset molto simili.

Gli esperimenti svolti ci hanno consentito di confrontare tra loro alcune delle strategie di aggregazione implementate dal framework stesso, confrontare le prestazioni dei modelli federati con le prestazioni ottenute da modelli centralizzati, comprendere meglio il funzionamento del FL in generale, affrontare alcune problematiche che incorrono in questi scenari ed eventuali soluzioni.

Come ulteriori sviluppi futuri, sarebbe interessante utilizzare Flower oltre l'ambito di simulazione, al fine studiare il comportamento del FL in uno scenario reale, dove si avrebbe effettivamente a che fare con problemi quali eterogeneità dei dispositivi client, problemi di rete e gestione dei dati provenienti da sorgenti diverse.

## RIFERIMENTI BIBLIOGRAFICI

- [1] <https://www.kaggle.com/datasets/muhammadshahidazeem/customer-churn-dataset>
- [2] <https://archive.ics.uci.edu/dataset/440/sgemm+gpu+kernel+performance>
- [3] <https://www.aporia.com/learn/a-comprehensive-guide-to-mean-absolute-percentage-error-mape/>
- [4] [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy)
- [5] <https://github.com/itslastonenikhil/federated-learning>