**Understanding of Federated Learning:**

**Machine Learning Model and Dataset:**

For this project, we have used a Linear Regression machine learning model to predict housing values based on the California Housing Dataset. The dataset is made up of 20640 total data samples and consists of 8 features including the median income, housing median age, average rooms, average bedrooms, population, average occupancy, and geographical factors like latitude and longitude. There is also one target variable, the median house value. The features included in the dataset are common potential predictors of house prices and thus enable our model to reliably predict the median house prices based on the physical properties and location of houses. Due to the presence of a single target variable, and our objective being a regression problem, the Linear Regression model is suitable choice for predicting the house value based on the given 8 regression factors. This suggests that there would be a linear correlation between the target variable and at least one regression factor. Since our scenario has 5 clients, the dataset is split up across each client, with each client having their own unique testing and training data on which they train their local model.

**Federated Learning Algorithm:**

The Federated Learning algorithm implemented in our project is the FedAvg (Federated Averaging) algorithm. The FedAvg algorithm is a core approach in Federated Learning which allows for collaborative model training across multiple clients while keeping each client’s data private. Our system consists of 1 server and 5 clients where each client has access to its own private dataset which is a unique subset of the California Housing Dataset. In the FedAvg algorithm, each client independently trains a local model on its data, and then sends the model updates, rather than the actual data, to the server. The server then aggregates the updates received from the clients to improve the current global model, and then distributes the new global model back to the clients so they can continue local training and updating. By decentralising data, it maintains data security and privacy, and reduces the storage strain on the server as the data is stored on clients instead which in turn improves scalability by simply adding on clients to the system.

**Implementation:**

Text