Build a Fully Homomorphic Encryption (FHE) Regression Model for Estimating Housing Prices

1. Introduction

This project addresses the need for privacy-preserving machine learning through the implementation of a deep learning regression model integrated with Fully Homomorphic Encryption (FHE). The model is trained on housing price data using TensorFlow and later adapted for encrypted inference using TenSEAL, an FHE library that supports CKKS encryption. This approach allows users to receive predictions on encrypted inputs, preserving data confidentiality during the entire inference process.

2. Dataset

The dataset was sourced from the FHERMA Challenge and includes two files: X_train.csv and y_train.csv. X_train.csv includes 13 numerical features such as average number of rooms, population, and income. y_train.csv contains the target variable, which is the median house price. The data was split into training and validation sets using an 80-20 ratio. StandardScaler was applied to input features and MinMaxScaler on target values for efficient model learning.

3. Overview of Workflow

The complete process of building the FHE-secure housing price regression system is outlined as follows:

Step 1: Load the dataset and perform preprocessing using scaling techniques.
Step 2: Build and compile a regression model using TensorFlow.
Step 3: Train the model on scaled data and validate its performance.
Step 4: Evaluate the model using MAE, MSE, and R^2 metrics.
Step 5: Integrate the trained model into an encrypted pipeline using TenSEAL.
Step 6: Encrypt sample input, perform inference on the encrypted data, and decrypt the result.
Step 7: Interpret and report results ensuring end-to-end privacy of the user data.

4. Implementation

Step 1: Dataset Loading and Preprocessing

The dataset is loaded using pandas. Input features (X) are normalized using StandardScaler to zero mean and unit variance. Target values (y) are scaled to [0, 1] using MinMaxScaler to stabilize training. The data is then split into training and validation sets using an 80-20 ratio, ensuring representative samples across both sets.

Step 2: Model Construction and Compilation

A TensorFlow Sequential model is constructed with three hidden layers (256, 128, and 64 neurons), each using ReLU activation. BatchNormalization and Dropout layers are included to enhance model generalization and prevent overfitting. The model is compiled with Adam optimizer (learning rate = 0.0005), Mean Squared Error (MSE) loss, and Mean Absolute Error (MAE) metric.

Step 3: Model Training and Validation

The model is trained on the training dataset for 100 epochs with a batch size of 32. Training time and validation loss are monitored per epoch. Early stopping or model checkpoints can be optionally used to preserve the best weights.

Step 4: Evaluation

After training, the model is evaluated on the validation dataset using MSE, MAE, and R^2 score. These metrics quantify error and accuracy. The model achieved Validation MSE: 0.0116, MAE: 0.0705, and R^2 Score: 0.7916. Training took approximately 211.90 seconds.

Step 5: FHE Integration via TenSEAL

The trained model is integrated with TenSEAL, which supports CKKS encryption for approximate real number operations. The encryption context is set with poly_modulus_degree = 8192 and coeff_mod_bit_sizes = [60, 40, 40, 60]. Encrypted vectors are passed into the model for prediction. The server performs inference without decrypting data.

Step 6: Encrypted Inference and Decryption

A test input vector is encrypted using TenSEAL’s ckks_vector. It is fed into the model for encrypted inference. The output is also encrypted and must be decrypted by the client to obtain usable predictions. This ensures that both input and output remain private and secure throughout the process.

Step 7: Interpretation of Encrypted Output

The decrypted prediction is rescaled back to its original value range. The final result is comparable to plaintext inference. This confirms the correctness and practicality of FHE-based secure machine learning inference.

5. Results and Evaluation

The model was evaluated using standard regression metrics. It performed well with the following metrics:

• Training Time: 211.90 seconds

• Validation MSE: 0.0116

• Validation MAE: 0.0705

• R^2 Score: 0.7916

Encrypted inference time was under 0.1 seconds per sample. The decrypted predictions matched expected results, validating the accuracy and correctness of the secure pipeline.

6. Conclusion

This project successfully demonstrates the integration of deep learning and Fully Homomorphic Encryption for secure housing price prediction. The implementation of encrypted inference using TenSEAL and the CKKS scheme ensures that user data remains private throughout the prediction workflow. This approach is applicable to many real-world ML tasks requiring strict privacy guarantees.