<u>Assignment – 3</u>

Name: Pola Gnana Shekar
Roll No: 21CS10052

Project Report: Predictive Modeling with Neural Networks and Cross-Validation

Introduction:

This project involves the development of predictive models using neural networks to predict housing prices. We will evaluate the model's performance using both non-cross-validated and cross-validated approaches.

1. Data Loading and Preprocessing:

We start by loading a dataset of housing prices using Pandas. The dataset is examined, and it contains information about various features related to housing. We split the data into training and testing sets, standardize the input features, and standardize the target values to make the training process more stable.

2. Neural Network Implementation:

A custom neural network class is implemented. The network consists of an input layer, a hidden layer with a specified number of neurons, and an output layer. Key methods include:

forward: Performs forward propagation to compute network predictions.

backward: Implements backpropagation to update weights and biases during training.

train: Conducts training, iterating through epochs until a stopping criterion is met.

predict: Generates predictions based on the trained model.

Three different neural networks are created, each with a different number of hidden layer neurons and learning rates. These models are then trained on the standardized training data.

3. Model Evaluation:

We calculate accuracy, regression metrics (Mean Absolute Error, Mean Squared Error, and R-squared), and a custom accuracy function (accuracy within a threshold) for the test dataset.

4. Results:

Below are the accuracy scores for each model without cross-validation:

Accuracy for case (a): 0.0

Accuracy for case (b): 0.8509

Accuracy for case (c): 0.8553

5. Cross-Validation:

To assess the models' generalization performance, we implement 5-fold and 10-fold cross-validation. This involves partitioning the data into multiple subsets, training and evaluating the models on different subsets iteratively. For cross-validation, we use the same neural network architectures as before.

6. Cross-Validation Results:

Here are the results for cross-validation:

<u>5-Fold Cross-Validation:</u>

For Hidden Layer Size: 3, Learning Rate: 0.01

Average MAE: 350,815,018.0658

Average MSE: 146,910,446,668,433,344.0000

Average R2 Score: -5,493,553.4593

Average Accuracy: 0.0

For Hidden Layer Size: 4, Learning Rate: 0.001

Average MAE: 53,809.4370

Average MSE: 4,978,237,164.7639

Average R2 Score: 0.8025

Average Accuracy: 0.8346

For Hidden Layer Size: 5, Learning Rate: 0.0001

Average MAE: 65,271.0249

Average MSE: 7,312,254,794.2248

Average R2 Score: 0.7128

Average Accuracy: 0.8426

10-Fold Cross-Validation:

For Hidden Layer Size: 3, Learning Rate: 0.01

Average MAE: 326,823,740.9056

Average MSE: 111,349,741,700,364,192.0000

Average R2 Score: -4,478,238.8882

Average Accuracy: 0.0

For Hidden Layer Size: 4, Learning Rate: 0.001

Average MAE: 51,582.4774

Average MSE: 4,607,021,405.2516

Average R2 Score: 0.8158

Average Accuracy: 0.8323

For Hidden Layer Size: 5, Learning Rate: 0.0001

Average MAE: 62,144.5582

Average MSE: 6,767,841,556.7417

Average R2 Score: 0.7332

Average Accuracy: 0.8420

Conclusion:

The project implemented neural networks to predict housing prices with different architectures and training parameters.

The models were evaluated based on accuracy and regression metrics.

The accuracy varied between models, with case (c) demonstrating the best results without cross-validation.

Cross-validation revealed that case (b) consistently performed well across multiple folds, with high accuracy and good regression metrics.

The case (a) model exhibited diverging loss during training, indicating a need for parameter adjustment.

The case (c) model provided promising results, demonstrating robustness and consistency across cross-validation folds.

In conclusion, cross-validation is a critical step to evaluate a model's generalization performance. While the neural network models can provide good results on certain datasets, careful tuning of architecture and training parameters is crucial to achieve the best performance.