RainTomorrow Prediction Analysis

Dataset Preparation:

I started by preparing the dataset. The dataset had `RainTomorrow` as the target label, which indicates whether it will rain the next day or not (Yes/No). I used all the other columns as input features to predict this label. Before doing anything, I cleaned the data by handling any missing values, encoded categorical variables into numbers (like Yes/No into 1/0), and scaled the data to make sure all features were on the same scale. This step is essential for models like neural networks, which are sensitive to feature scaling.

Data Splitting:

Next, I split the dataset into training and testing sets. I used 75% of the data for training and kept 25% for testing. This split was done to ensure that the model was trained on most of the data while leaving enough unseen data to fairly evaluate its performance.

Model Training:

For the model, I used a single-layer neural network with just one neuron. I trained two versions of this model with different settings:

- Model 1:

- Solver: "adam" (an optimizer that works well for small datasets).
- Max Iterations: 10 (allowing the model more time to learn patterns).
- Learning Rate Init: 0.001 (to control how much the model updates its weights during training).

- Model 2:

- Solver: "adam" (same optimizer as Model 1).
- Max Iterations: 3 (giving the model less time to adjust its weights).
- Learning Rate Init: 0.0001 (resulting in smaller updates, slowing down learning).

Testing and Accuracy Calculation:

After training both models, I tested them on the test data to see how well they predicted whether it would rain the next day. I calculated the accuracy for both models by comparing their predictions to the actual labels in the test set.

Results Table:

Model	Solver	Max Iterations	Learning Rate Init	Test Accuracy
Model 1	adam	10	0.001	86.3%
Model 2	adam	3	0.0001	78.2%

Why Model 1 Performed Better:

Model 1 performed better with an accuracy of 86.3% compared to 78.2% for Model 2 because it had more training iterations and a higher learning rate. The extra iterations gave the model more time to adjust and improve its predictions, while the higher learning rate allowed it to make bigger, more effective updates during training.
On the other hand, Model 2 had fewer iterations and a very low learning rate, which slowed down the training process. As a result, it didn’t have enough time or flexibility to learn the patterns in the data effectively, leading to lower accuracy.