Data Size	Hidden Layers	Training Error	Validation Error	Execution Time (s)
1000	1×4	0.0400	0.0300	1.19
1000	2×4	0.0400	0.1000	1.14
10000	1×4	0.0030	0.0040	3.19
10000	2×4	0.0021	0.0070	3.33
100000	1×4	0.2419	0.2425	9.10
100000	2×4	0.2419	0.2425	9.49

2. Based on the results, which model do you consider as superior, among the deep learning models fit?

Based on the results of the deep learning models across three dataset sizes (1,000; 10,000; and 100,000 rows), the configuration with 1 hidden layer and 4 nodes consistently outperformed the 2-layer architecture in terms of validation error and overall efficiency. For the smallest dataset (1,000 rows), the 1-layer model achieved a lower validation error (0.0300) compared to the 2-layer model (0.1000), suggesting better generalization despite similar training errors. In the medium dataset (10,000 rows), the 1-layer model again performed better with a lower validation error (0.0040 vs. 0.0070) and slightly shorter training time. For the largest dataset (100,000 rows), both models exhibited high training and validation errors (~0.24), indicating underfitting or limited capacity to model the data. Overall, the 1-layer model with 4 nodes is considered superior due to its better generalization performance and comparable or better training efficiency across all dataset sizes.

3. Next, report the results (for the particular numbers of observations) from applying xgboost (week 11 – provide the relevant results here in a table). Comparing the results from XGBoost and deep learning models fit, which model would you say is superior to others? What is the basis for your judgment?

XGBoost is the superior model, particularly the Python implementation using scikit-learn. It consistently achieved higher predictive accuracy across all dataset sizes and demonstrated excellent scalability with minimal computational cost. While deep learning models performed comparably on small datasets, they struggled significantly on larger datasets, especially with high validation error and longer training time.

What is the basis of judgment?

The judgment is based on three key criteria:

1. Predictive Performance: XGBoost achieved higher accuracy (e.g., over 98% for large datasets) with low validation error, while deep learning models showed degraded performance, particularly at 100,000 rows (validation error of 0.2425).
2. Execution Time: XGBoost models trained in under a second even for millions of rows, whereas deep learning models required several seconds and still underperformed.
3. Scalability and Stability: XGBoost maintained consistent performance across data sizes and implementations (Python, R, caret), while deep learning models struggled with underfitting and were less reliable on larger data.

These factors combined demonstrate that XGBoost offers the best balance of accuracy, efficiency, and robustness for this task.

Data Size	Method	Testing Accuracy	Validation Error	Time to Fit (sec)
100	Python (scikit-learn)	0.8500	0.1500	0.14
1,000	Python (scikit-learn)	0.9450	0.0550	0.10
10,000	Python (scikit-learn)	0.9755	0.0245	0.28
100,000	Python (scikit-learn)	0.9826	0.0174	0.34
1,000,000	Python (scikit-learn)	0.9824	0.0176	0.38
10,000,000	Python (scikit-learn)	0.9823	0.0177	0.36
100	R (xgboost)	0.9000	0.1000	0.00
1,000	R (xgboost)	0.9450	0.0550	0.02
10,000	R (xgboost)	0.9720	0.0280	0.18
100,000	R (xgboost)	0.9848	0.0152	1.51
1,000,000	R (xgboost)	0.9888	0.0112	14.44
10,000,000	R (xgboost)	0.9896	0.0104	145.46
100	R (caret + xgboost)	0.9000	0.1000	0.14
1,000	R (caret + xgboost)	0.9250	0.0750	0.11
10,000	R (caret + xgboost)	0.9550	0.0450	0.42
100,000	R (caret + xgboost)	0.9677	0.0323	3.37
1,000,000	R (caret + xgboost)	0.9663	0.0337	34.01
10,000,000	R (caret + xgboost)	0.9671	0.0329	372.75