Biostats625_Final_Report

Team: - Yulin Shao - Tong Liu - Yichen Zhao - Yana Xu

2024-12-17

Introduction

Diabetes is a widespread health condition, impacting millions of people around the globe. Early detection can significantly improve patient outcomes and reduce medical costs over time. Our group set out to build strong and efficient machine learning models to predict diabetes using demographic and health-related information. Initially, we tested our pipeline on a smaller Heart Disease dataset (roughly 900 entries), but its modest size didn't fully expose the computational and memory challenges that accompany much larger datasets. Consequently, we transitioned to the **CDC Diabetes Health Indicators dataset**, comprising over 250,000 records. Handling so many observations led us to use parallel computing, sparse data structures, and other optimizations to keep training times feasible while aiming for high predictive accuracy.

Methods

K-Nearest Neighbors

We initially picked K-Nearest Neighbors (KNN) for its simplicity, but it quickly became obvious that naive distance computations across hundreds of thousands of rows took a lot of time and memory. We decided to reduce dimensionality by applying PCA for numeric features and MCA for binary ones, dropping from 20 to 9 dimensions. Then, we rewrote the distance function in C++ and used partial sorting (nth_element) so we only sorted the top k neighbors instead of sorting all distances. We also parallelized the test set predictions through parLapply by processing 1,000 test samples per chunk. These changes brought runtime down from 85 seconds to 32 seconds (a ~62% speedup). KNN ended up at about **72% accuracy** once we optimized it heavily.

Logistic Regression

Logistic Regression is more interpretable, mapping features to a probability of having diabetes:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=1}^{p} \beta_i X_i)}} \tag{1}$$

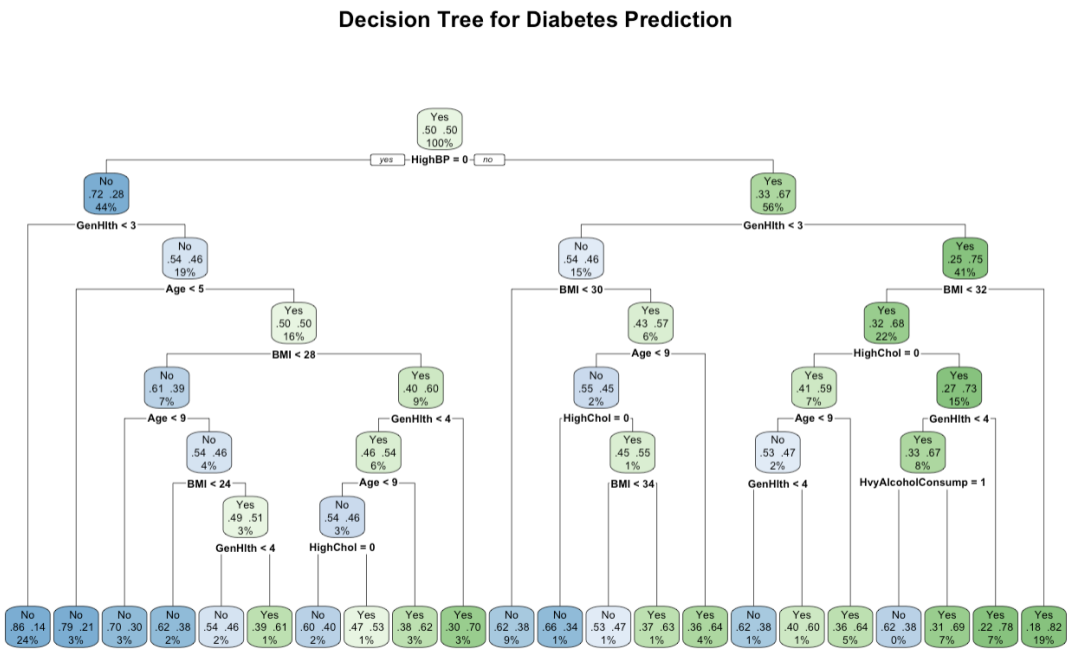
Because the dataset was massive, we applied elastic net regularization:

$$\min_{\beta_0, \beta} \left[-\sum_{i=1}^{n} \log(P(y_i|x_i)) + \lambda(\alpha||\beta||_1 + \frac{1-\alpha}{2}||\beta||_2^2) \right] \tag{2}$$

Cross-validation of this model was still expensive. We addressed it by parallelizing the folds (doParallel), greatly reducing training time. In the end, Logistic Regression scored ~**70%** accuracy. Although less accurate than ensembles, it was relatively straightforward to interpret and faster than KNN once parallelization was introduced.

Decision Tree

A single Decision Tree offered a simpler approach. Trees split the data by greedily selecting features that maximize information gain at each node. Although naive trees can underperform compared to ensembles, they are easy to train and interpret—key for many real-world health applications. We did a grid search over complexity parameter (cp) from 0.0005 to 0.02, speeding up 5-fold cross-validation with parallel processing. Our final model had **73.76%** accuracy, AUC ~0.7929, sensitivity 0.7636, and specificity 0.7116. While short of the accuracy seen with ensembles, this single-tree approach remained computationally light and transparent.



Random Forest

Random Forest merges multiple bootstrapped trees into a robust ensemble. Initially, we tried the basic randomForest package, but it performed slowly for 250k rows. Switching to ranger made a big difference: we harnessed multi-threading, cutting training time from 362 seconds to ~3.42 seconds. The final accuracy reached **74.81%** (AUC ~0.8263). This demonstrates that carefully chosen libraries and parallelization can make ensemble methods viable even for big data.

XGBoost

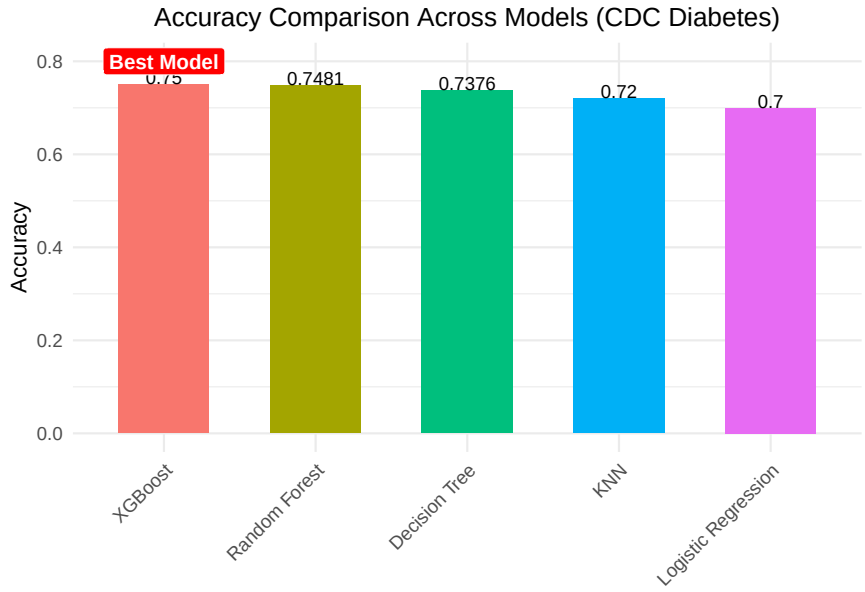
XGBoost is a boosted tree framework that includes parallel tree-building and strong regularization to help control overfitting:

$$\mathcal{L} = \sum_{i=1}^{n} l(y_i, \hat{y}i) + \sum k = 1^K \Omega(f_k) \tag{3}$$

We compared dense vs. sparse matrix representations, as well as sequential vs. parallel implementations. Sparse matrices skip zero values—handy if you have a lot of one-hot features—improving memory usage and training time. We tuned the learning $0.01, 0.1, 0.3$ and used early stopping (10 rounds) to prevent overfitting. Ultimately, XGBoost reached **75%** accuracy and ~0.82 AUC, comparable to Random Forest.

Results

Below is some R code comparing the final accuracies of our models in a single plot:



The chart shows XGBoost and Random Forest at the top with around 75% accuracy (AUC near 0.82), a single Decision Tree at ~73.76%, KNN at 72%, and Logistic Regression at 70%. While decision trees are the simplest approach and KNN required significant optimization, the ensemble methods maintained both high accuracy and efficient training via parallelization.

Conclusion

All of these models, from single Decision Trees to complex ensembles, are designed with the hope of early diabetes detection. In public health contexts, an accurate yet efficient model could help identify high-risk individuals sooner, potentially lowering complications and long-term costs. Our primary objective was to balance predictive accuracy with computational feasibility, ensuring the approach could handle 250k+ rows. Each technique—KNN, Logistic Regression, Decision Tree, Random Forest, and XGBoost—used specialized parallel or memory-saving strategies (e.g., chunk-based predictions, sparse matrices, multi-threading) to cope with the dataset's size.

Our project demonstrated that advanced optimization strategies—like parallel cross-validation, sparse representations, chunk-based processing, and partial sorting—are crucial for handling large-scale diabetes data efficiently. Even classic models (like KNN or Logistic Regression) can be sped up drastically if we use the right libraries and parallelization. Random Forest and XGBoost performed the best, each reaching ~75% accuracy with AUC around 0.82, and training in a fraction of the time needed by naive implementations. Meanwhile, the single Decision Tree maintained decent interpretability but now shows improved accuracy (~73.76%) compared to our initial estimate. Future healthcare applications may want to systematically benchmark these methods under different parallel and sparse settings, balancing speed, memory usage, and accuracy to facilitate early diabetes detection on massive datasets.

References

- Rios Burrows, N., Hora, I., Geiss, L. S., Gregg, E. W., & Albright, A. (2017). Incidence of End-Stage Renal Disease Attributed to Diabetes Among Persons with Diagnosed Diabetes—United States and

Puerto Rico, 2000–2014. Morbidity and Mortality Weekly Report, 66(43), 1165–1170.
- Detrano, R., Jánosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S., & Froelicher, V. (1989). International application of a new probability algorithm for the diagnosis of coronary artery disease. American Journal of Cardiology.