Applications of *k*-Nearest Neighbors and Artificial Neural Networks to Predict Eliteness in American Colleges and Universities (2022)

Rhonda H. Lott, *Technical Writer*

***Abstract*—The machine learning classification algorithms, *k*-NN (*k*-Nearest Neighbors) and ANN (an Artificial Neural Network), were compared in terms of accuracy and speed in their predictions of a Boolean. This variable described the academic reputations of colleges and universities in the United States based on a quantifiable definition of the word "elite" given as input. The *k*-NN algorithm was hypothesized to perform more correctly and quickly than the ANN algorithm. For evaluation, a testing program was written in the R programming language and development environment and applied to a public database of colleges and universities and their attributes. In trials, both algorithms performed with perfect accuracy, but *k*-NN finished the tasks with runtimes significantly shorter than those of ANN.**

***Index Terms*—Machine learning, classification algorithms, k-Nearest Neighbors, artificial neural networks, R programming, higher education.**

I. Introduction

In this experiment, two classic supervised learning algorithms invented to teach machines how to make classifications based on given criteria, *k*-NN (*k*-Nearest Neighbors) and ANN (Artificial Neural Network), were applied to a data frame of 777 United States colleges and universities to determine which one would learn to identify those classified as elite the fastest and most accurately. The *k*-NN algorithm classifies uncategorized samples by grouping them with categorized samples with which they have attributes in common. This algorithm was chosen because it has proven particularly appropriate for classifying data in which there are clearly associations between the attributes of the samples but are too complex and copious to be easily explained. As data scientist Brett Lantz notes, this algorithm is particularly adept at solving problems in which a term (such as "eliteness") may be problematic to delineate, but humans "know it when [they] see it" [1]. The *k*-NN algorithm was chosen because its simple implementation seemed especially applicable to a problem that requires a binary response (i.e., whether an institution is either elite or not, based on a definition derived from numerical data).

While the *k*-NN algorithm groups data into categories according to their similarities, the ANN algorithm makes predictions by imitating the connections among neurons in the brains of living organisms. This functionality allows ANNs to adapt to a variety of educational situations and not only perform classifications but make numerical predictions and recognize patterns. Although the most advanced examples only contain hundreds of artificial neurons, relatively few compared to the approximately 86 billion biological neurons of the human brain, they successfully perform some human-like tasks such as recognizing and correctly interpreting speech, handwriting, and pictures as well as navigating and adapting to their environments spatially, among others [2]. They are used less frequently, however, in making decisions at which humans may arrive intuitively without visual input such as judging whether an item fits the definition of a word based on previous background knowledge as in this experiment. Because there are arguably too many correlations among the attributes of colleges and universities defined as elite to be easily identified by a biological neural network as large as that of the human brain (let alone a much smaller, artificial one), the researcher hypothesized that training a machine with the *k*-NN algorithm known to be well-suited to this type of classification task would result in faster and more accurate classifications of colleges and universities than training with the ANN algorithm.

Although reputations of institutional status vary according to the beliefs and values of individuals and groups, the machine learning experiment described in this report utilized the noun *eliteness* specifically to denote a state of high numerical rank as determined by the mean high school grade point averages of recent incoming undergraduate students at a college or university. In the United States, admission standards are becoming increasingly qualitative, and the reliance on quantitative data in the decision-making process is beginning to wane in popularity, but most institutions continue to prioritize

grades along with standardized test scores (e.g., SAT and ACT) as major deciding factors in applicant acceptance or rejection [3]. For this reason, among others outside the scope of this experiment, institutions that recruit the highest percentages of numerically high-ranking students such as the eight private research universities commonly known as the Ivy League are still widely considered the most elite.

Since it would be unproductive if not impossible to judge the test scores of students who took the SAT only and those who took the ACT only by the same metric as the two tests are designed to measure different sets of skills, *elite students* were characterized simply as those who graduated with GPAs in the top 10% of their high school classes. Accordingly, *elite colleges and universities* were defined in this context as those that enroll the highest percentages of new elite undergraduate students according to recent publicly available data [4]. Trained based on these criteria, both algorithms investigated in this experiment identified elite colleges and universities with perfect accuracy, but the *k*-NN algorithm performed its predictions significantly more quickly than the ANN algorithm. These results suggest complex artificial neural networks capable of both supervised and unsupervised learning, while popular and effective at visual learning tasks, may be unnecessarily costly in time and resources for making some types of predictions, such as those based on non-visual quantifiable definitions, compared to the simpler, supervised, and Boolean-driven *k*-Nearest Neighbors classification algorithm.

II. Methodology

To sort, compile, manage, input, utilize, and analyze the existing dataset on colleges and universities, the researcher employed the R programming language and development environment, a statistical computing and graphics software program with applications in the execution of machine learning algorithms.

A. Data Preparation Program in R

A program was created to prepare a data frame of colleges and universities with an array of 17 variables for each of the 777 institutions to be read as input by the classification algorithms to predict whether each institution could be considered elite. The variables quantified the following information:

1. private or public status of an institution;
2. number of applications received;
3. number of applications accepted;
4. number of new students enrolled;
5. percentage of new students who graduated in the top 25% of their high school classes;
6. percentage of new students who graduated in the top 10% of their high school classes;
7. number of full-time undergraduates enrolled;
8. rate of out-of-state tuition;
9. cost of room and board;
10. cost of books;
11. estimated personal spending of students on campus;
12. percentage of faculty with doctoral degrees;
13. percentage of faculty with other terminal degrees;
14. student to faculty ratio;
15. percentage of alumnae who donate money to the institution;
16. instructional expenditure per student;
17. graduation rate.

The elite or non-elite status of each institution in the dataset was also represented as a variable, the Boolean $E = 0$ or $E = 1$. To prepare the program to make quantitative predictions, the E variable was defined to comprise all educational institutions with greater than 50% of their current students who graduated in the top 10% of their high school classes. As a result, 78 institutions were classified as elite, and the remaining 699 were not. The data preparation instructions for the machine were written as follows.

1) Load the college and university data frame into the development environment.

```
> college <- read.csv
  ("https://www.statlearning.com/s/
  College.csv", stringsAsFactors =
  FALSE)
```

2) Remove the column of non-numerical data, the labels identifying the colleges and universities, from the input.

```
> college <- college[-1]
```

3) Create a column to add the elite variable (*E*) and set it to *true* if the number of incoming students at a college or university who graduated with GPAs in the top 10 percent of their high school classes is greater than 50 percent.

```
> E <- rep("No", nrow(college))
> E[college$Top10perc > 50] <- "Yes"
> E <- as.factor(E)
> college <- data.frame(college, E)
> library("dplyr")
> college <- college %>% relocate(E)
> summary(college)
```

4) Convert strings to factors to make them accessible to the classification algorithms.

```
> college$Private <- factor
> (college$Private, levels = c("Yes",
  "No"),labels = c(1, 0))
> college$E<- factor(college$Elite,
  levels = c("Yes", "No"),
```

```
  labels = c("Elite", "Not Elite"))
> table(college$E)
```

5) Convert factors to numerics to ensure all input is in the same format.

```
> college$Private <- as.numeric
  (college$Private)
> college$E <- as.numeric
  (college$E)
```

6) Normalize the data within the range of 0 and 1 to accommodate the Boolean elite variable.

```
> normalize <- function(x) {return
  ((x - min(x)) / (max(x) - min(x)))}
> college_n <- as.data.frame
  (lapply(college[-19], normalize))
  summary(college_n)
```

7) Remove Top10perc variable from testing data.`tic()`

```
> college_n <- college_n[-6]
> summary(college_n)
```

8) Divide the data into two sets, a smaller one for training and a larger one for testing.

```
> college_train <- college_n[1:100, ]
> college_test <- college_n[1:777, ]
```

9) Add labels to the data to increase the readability of the future results.

```
> college_train_labels <- college
  [1:100, 1]
> college_test_labels <-
  college[1:777, 1]
```

B. K-NN Prediction Program in R

A second R program was developed to load and run the *k*-NN algorithm packaged for the R development environment and to log its speeds in five trials. The first prediction trials were conducted using the following code.

1) Run the *k*-NN prediction function and log the runtimes. For *k*, the researcher chose an odd number roughly equal to the square root of the number of data entries.

```
> library(class)
> tic("k-NN Runtime")
> college_test_pred <- knn(train =
  college_train, test = college_test,
  cl = college_train_labels, k = 27)
> toc()
```

2) Display the results in a table of predicted and actual numbers of Elite colleges and universities.

```
> library(gmodels)
> CrossTable(x = college_test_labels,
  y = college_test_pred, prop.chisq =
  FALSE)
```

C. ANN Prediction Program in R

A third R program was written to locate, execute, and record the runtimes of the provided ANN algorithm in five additional trials. The code below facilitated the second and last set of prediction trials

1) Run the ANN prediction function and log the runtimes.

```
> library(class)
> tic("ANN Runtime")
> college_model <- neuralnet(E ~
  Apps + Accept + Enroll + Top10perc
  + Top25perc + F.Undergrad +
  P.Undergrad + Outstate + Room.Board
  + Books + Personal + PhD + Terminal
  + S.F.Ratio + Expend, data =
  college_n)
> toc()
```

2) Display the results in a table of predicted and actual numbers of Elite colleges and universities.

```
> plot(college_n)
> CrossTable(x = college_test_labels,
  y = college_test_pred,
  prop.chisq = FALSE)
```

As output, the *k*-NN and ANN programs displayed predictions in two tables, which were redesigned as a single table more easily accessible. Finally, the runtimes of five trials as measured by the `tic()` and `toc()` timing functions were averaged and recorded in a table, as well.

III. Results

As Table I shows, both the *k*-NN and ANN algorithms predicted the numbers of elite colleges and universities in the data frame with 100% accuracy.

TABLE I

	True Positives	True Negatives	False Positives	False Negatives
k-NN	78	699	0	0
ANN	78	699	0	0
Correct	78	699	0	0

The *k*-NN algorithm, however, made predictions 4.566 seconds faster on average than the ANN algorithm, as indicated in Table 2.

TABLE II

	Trial 1	Trial 2	Trial 3	Trial 4	Trial 5	Mean

k-NN	0.09 s	0.11 s	0.08 s	0.11 s	0.11 s	0.100 s
ANN	4.56 s	4.32 s	4.56 s	4.74 s	5.14 s	4.664 s

Thus, the simpler *k*-NN algorithm was able to identify elite colleges and universities more quickly but no more accurately than the more complex ANN.

ACKNOWLEDGMENT

Rhonda H. Lott thanks Sviatoslav Braynov for his invaluable notes on k-NN and ANN algorithms presented in his graduate course in machine learning. She also thanks Mark Tabone for his editing suggestions. Finally, she is grateful to all her professors at the University of Illinois Springfield for helping her build upon the foundation of her education in computer science.

REFERENCES

[1] B. Lantz, "Lazy learning – classification using nearest neighbors," in *Machine Learning with R: Expert Techniques for Predictive Modeling*, Birmingham: Packt, 2019, ch.3, pp. 65 – 88.

[2] B. Lantz, "Black box methods – neural networks and support vector machines," in *Machine Learning with R: Expert Techniques for Predictive Modeling*, Birmingham: Packt, 2019, ch.7, pp. 217 – 229.

[3] D. Hossler, E. Chung, J. Kwon, J. Lucido, N. Bowman, and M. Bastedo, "A study of the use of nonacademic factors in Holistic Undergraduate Admissions Reviews," *The Journal of Higher Education*, vol. 90, no. 6, pp. 833–859, 2019.

[4] J. Gareth, D. Witten, T. Hastie, and R. Tibshirani, "College.csv," *An Introduction to Statistical Learning*, 04-Aug-2021. [Online]. Available: https://www.statlearning.com/s/College.csv. [Accessed: 16-Nov-2022].

[5] S. Braynov (2020). CSC532 Session 3: *K*-nearest neighbors (k-NN) and naive bayes [PowerPoint slides].

[6] S. Braynov (2020). CSC532 Session 5: Neural networks, support vector machines, and association rules [PowerPoint, slides].