Modeling Complex DDA Purchasing and Use Patterns with Machine Learning

Kevin W. Walker
The University of Alabama, USA

Zhehan Jiang
Peking University, China

Introduction

This paper describes a project aimed at exploring the practical application of machine learning in the library environment by way of demand-driven acquisitions (DDA). This research uses DDA program purchasing data to build predictive models of future DDA purchasing by two separate methods. A model utilizing the *Adaptive Boosting* (AdaBoost) algorithm is compared against a model utilizing a more traditional logistic regression method, to determine what benefits, if any, machine learning offers toward predicting DDA purchasing patterns. This paper also explores, if only superficially, additional questions surrounding applied machine learning in the academic library setting and its meaning in relation to established library assessment practice.

Literature Review

DDA provides libraries with an opportunity to build collections in a different way. Instead of purchasing high quality materials *just-in-case* a user need may present itself, libraries can instead focus on building *just-in-time* collections that meet users' demonstrable needs at the point of need.[1] "As DDA purchases are triggered by use, the collections that result are implicitly associated with higher levels of initial use than traditionally acquired collections."[2] Several studies bear out this assertion, as well as show increased levels of continued use for DDA materials, when compared with more traditional, librarian-mediated selections.[3]

As a data-driven purchasing model, DDA benefits from the real-time resource usage reporting that are only possible with networked information systems as a foundation. These systems produce a wealth of data that not only inform trigger activity, but can be further analyzed by libraries toward a deeper understanding of where DDA is performing as expected and where change is necessary to improve program performance. Basic descriptive and inferential analyses can be used to determine if purchasing and use patterns align, signaling that purchasing levels are appropriate.[4] However, engaging in this level of analysis on an ongoing basis is likely to prove an unsustainable challenge for most librarians and this is where machine learning may provide a solution.

Machine Learning and Libraries

Within the library context, the use of data science techniques, such as *data harvesting* and *machine learning*, shows promise as a means of developing sustainable, ongoing assessment programs that support more effective organizational decision-making through predictive modeling.[5] In its simplest terms, machine learning involves using computers (i.e., *machines*) to identify patterns within large amounts of data.[6] As the name suggests, machine learning implies that, over time as experience accrues (i.e., a larger sample of data analyzed), the computer will become better at performing analytical tasks.[7]

Generally speaking, what makes machine learning possible are *learning algorithms*, which facilitate one of two main learning model approaches—*supervised* and *unsupervised* (or *non-supervised*) learning.[8] These algorithms are detailed, step-by-step instructions that allow a computer to solve a particular type of learning problem.[9] In the case of supervised learning, the computer is provided guidance, in the form of classified data, to help solve the problem at hand.[10] Typically, such learning problems are related to either *regression* or *classification* tasks.[11] In the case of unsupervised learning, the computer is fed unlabeled data that it must then separate into different classes based on similarities and dissimilarities.[12] Solutions to these types of problems involve either a *cluster-based approach* or *principal components analysis*.[13]

Scholars are illustrating the broad applicability of data science techniques within the academic environment. Mitchell (2006) discusses the use of web crawlers and other data harvesting tools to help build finding aids for high quality internet resources that support local research needs.[14] Such an application also highlights the potential for these technologies to automate and improve classifications of information resources, which can improve resource discoverability for library users. Renaud et al. (2015) discuss how data mining techniques can be used to harvest data from various, unconnected university systems both in and outside the library to better understand library use patterns.[15] Solis et al. (2018) describe their use of machine learning algorithms to predict university student desertion.[16] Litsey and Mauldin (2018) discuss similar approaches used to help improve library services.[17]

While there are few if any examples of machine learning used within the context of DDA, traditional methods have been leveraged toward predictive modeling within this context. A recent study by Kohn (2018) applies logistic regression toward an examination of factors believed to affect e-book usage. In that study, facets of *Library of Congress Classification* (*LC class*), *publisher type*, and *e-book platform* are shown to be statistically significant explanatory variables.[18] However, it is important to note that the overall explanatory power of Kohn's model is very low, as indicated by a McFadden's Pseudo-R^2 of 0.0396. In practical terms, this means that Kohn's model can only predict around 4% of e-book usage variability, which is far too low to support effective evidence-based decision-making.

Methodology

The data used in this study were generated by DDA activity spanning a 24-month period and cover a total of 52,628 titles, 9,767 of which had been triggered/purchased. These titles represent works produced by 306 different publishers, across 57 unique publications years, and feature a total retail value of nearly $5.3 million USD. The average amount of time these titles have been available to users is 532 days (527 days for untriggered titles and 556 days for triggered titles).

While Kohn (2018) focused on predicting e-book usage levels, this study focuses instead on predicting whether or not a title will be triggered for purchase. This decision is based on several factors. First, the work of Kohn (2018) has already illustrated that readily available DDA data are not well-suited to predicting usage levels, as evidenced by the extremely low McFadden's pseudo-R^2 values associated with that study's various predictive models. What's more, the AdaBoost learning algorithm used here is designed to solve classification problems, which in the case of DDA is represented by the dichotomous title status of either *triggered* or *untriggered*.

The final data set includes the following data for each e-book: *publisher*, *LC class*, *price*, and *trigger status*. *Publisher* is defined as the contract publisher who produced the e-book title, which takes the form of a categorical (i.e., nominal) text string value. *LC class* defines the subject matter of the e-book title and is also a categorical text string value. Importantly, *LC class* designation was limited to the top-level classification (i.e., the single, primary letter of the classification) to avoid *overfitting*[19] within the model. *Price* is a monetary value (i.e., ordinal numerical value) defined as the publisher's listed price for a specified e-book title. *Trigger status* defines whether or not an e-book has been triggered for purchase via one of the various purchasing triggers[20] established within the DDA platform. This variable takes the form of a simple binary value where one (1) means a title is triggered and zero (0) means untriggered.

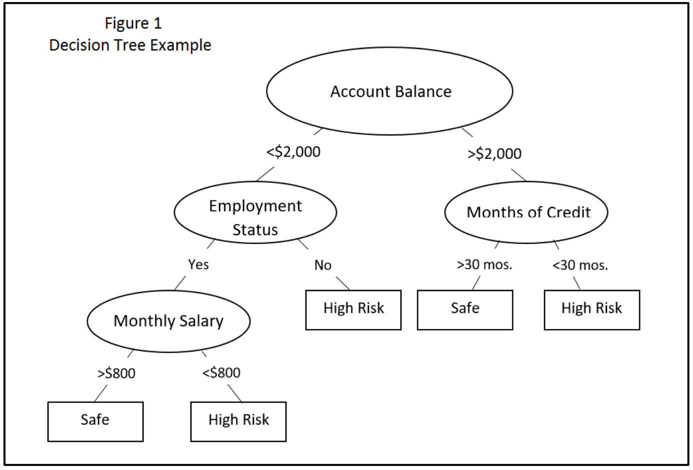
Machine Learning Algorithm

As suggested by the data set parameters, the analytical objective of this study involves a classification problem for which a supervised learning approach is most appropriate. A *classifier* is a rule that governs the division of individual population elements into two or more groups. In this case, the target outcome (i.e., dependent variable) for the classifier is represented by the binary trigger status of either *triggered* or *untriggered*. Further, variations in the sub-population features (i.e., explanatory variables), represented by each e-book's *publisher*, *publication year*, *LC class*, and *price*, help to define the classification rule (i.e., whether an e-book purchase is triggered).

Traditionally, due to high levels of interpretability, inferential capacity, and fast computational speed, logistic regression has been the method of choice, across a variety of disciplines, for solving binary classification problems. However, as the work of Kohn (2018) illustrates, traditional regression-based approaches can fall short in their ability to successfully predict outcomes. For this research study, decision trees are the fundamental unit of construction for a DDA purchasing prediction model. A *decision tree* is a classifier that uses a set of binary rules applied to calculate a target value; essentially decisions trees operate within the context of "if this than that" conditions to yield a result.

Unlike logistic regression, decision trees require no statistical assumptions. To illustrate, Figure 1 illustrates a decision tree process for evaluating bankruptcy risk, where the set of determining factors includes *account balance*, *employment status*, *credit longevity (months)*, and *monthly salary*. In this model, the set of prediction outcomes (i.e., classes) are labeled as *high risk* and *safe*. It should be noted that Figure 1 illustrates a fairly simple decision tree, as denoted by the limited number of decision and leaf nodes.[21]

Figure 1: Decision Tree Example



Theoretically, depending on the complexity of the training data used, a highly specific and equally complex set of decision rules will define the resulting tree. However, highly specific rules that provide for 100% predictive success in relation to the underlying training data used, will most likely have much lower predictive success within the larger test data set, or any other data not included in the sample-based training data set. An occurrence such as this exemplifies the concept of model overfitting, where decision rules have been allowed to become so specific to the training data that the underlying trend or phenomenon driving outcomes is obscured by statistical *noise*[22] that is unique to those training data. To help control for overfitting, one can limit the number of branches and leaves via *pruning*[23] or use an advanced version of decision trees known as *random forests*,[24] which is an ensemble method using a collection of decision trees whose results are aggregated to derive a single best-fit classifier.[25]

An *ensemble decision algorithm* (or ensemble method) is an algorithm that either uses a combination of multiple base learning algorithms, several analytical iterations of the same algorithm, or both, to derive a single predictive model more accurate than would have been possible using a single method or analytical iteration.[26] The *random forest* algorithm is an ensemble method that combines elements of statistical *bootstrapping*, *bootstrap aggregation* (bagging), and *decision tree learning*. As previously mentioned, this approach benefits from a single best-fit decision model derived from the outcomes of multiple decision trees, each of which is built upon a random selection of base features (i.e., decision factors, or independent variables) derived from sampled training data. Studies have indicated that, under a variety of circumstances, random forests are more robust and accurate than logistic regression and traditional decision tree methods, especially when the size of the test data is moderate to large.[27]

Boosting is an algorithm-based, ensemble classification technique similar to random forests. It builds a more accurate classifier (i.e., predictive model) through step-wise optimization, which involves a sequence of multiple analytical iterations—where with each successive iteration's analytical focus is more heavily weighted toward *weak-learners* (i.e., classifiers that have poor predictive capacity). In essence, the algorithm is deriving a best-fit predictive model from the aggregated results of many attempts. By adjusting the aggregate model's fit so that predictive capacity is continually improved in relation to weak-learners, this algorithm is able to construct more accurate classifiers with each successive round of analysis.[28]

Several variants of the boosting algorithm have been developed in recent years, such as XGBoost,[29] CatBoost,[30] and AdaBoost.[31] For the purposes of this study, the AdaBoost algorithm (*AdaBoost.M1* via the *Adabag*[32] R package) was selected, which has proven to be faster and more accurate in a variety of situations.[33] In addition, compared with other boost algorithm variants, AdaBoost requires fewer algorithmic parameter tunings,[34] which are integral to the effective deployment of XGBoost and CatBoost. Additional technical detail regarding the AdaBoost algorithm can be found in Freund and Schapire (1997).[35]

DDA Trigger Analysis

All analyses were performed within the R software environment.[36] The logistic regression analysis was executed via the built-in general linear model (glm) R function, while the AdaBoost portion of the study was powered by the *adabag*[37] R package. Alternatively, one might choose the *fastAdaboost*[38] package for performing AdaBoost algorithm. The fastAdaboost package was programmed using C++, a low-level programming language that operates more efficiently than the R language. However, on this occasion the adabag package was chosen due to the ease with which that package can output the importance weightings of those factors shaping the resulting classifier. The base classifier adopted for the AdaBoost model is *random forests*. For those wishing to replicate these methods, code used to initiate both the logistic regression and AdaBoost decision models can be found in *Appendix A* and *Appendix B*, respectively.

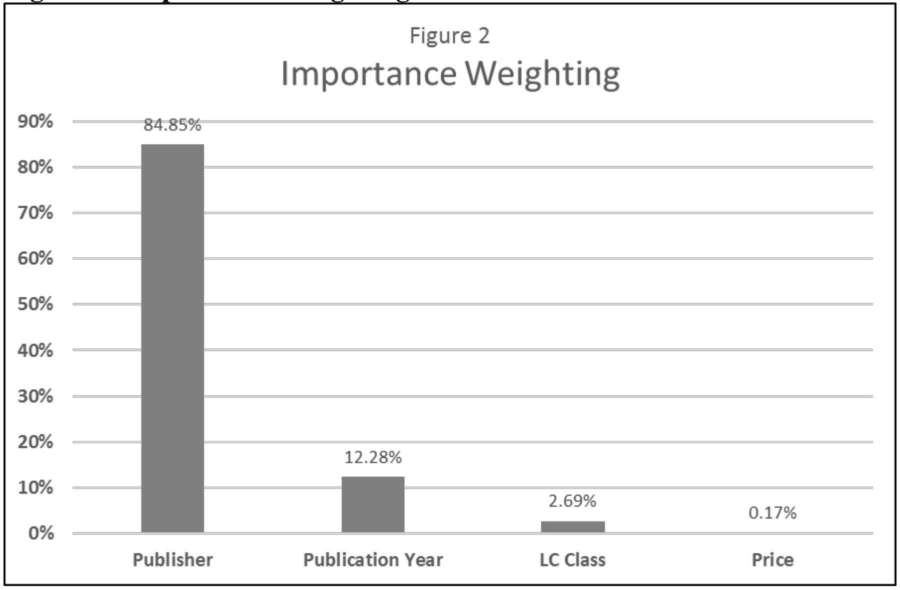
To begin, all required R packages are downloaded and activated within the RStudio Server[39] workspace. Original DDA purchasing data in the form of a Microsoft Excel file is imported into RStudio Server and named as a data frame object. Then, both a training and a testing data set are created using a randomly selected subset[40] of those original data. The training data is then fed through the AdaBoost algorithm. The resulting model is named *dda.adaboost*. Analytical parameters for the boosting algorithm are set to optimize this process. In this case, the number of analytical iterations is set to twenty (20)[41] and decision tree depth is limited to three levels.[42] The time required for model training will vary based on: the size of the training data set, the complexity of the model (i.e., selected analytical parameters), and the processing/memory specifications of the computer system being used. In the case of this study, analysis via normal desktop computer would take several days. For this reason, the researchers chose to use RStudio Server, which utilized eight dedicated processing cores and 32GB of RAM on a local server. This server-based analysis was completed in less than five minutes.

The process of feeding data through each decision model, such that model parameters can be estimated, is called *training*. With the training process complete, testing data were fed through the newly-trained predictive model. Thresholds for title-level classifications are, by default, set to 0.5. Therefore, if the estimated probability for a title being triggered is higher than 50%, the predicted outcome is labeled as *triggered* and, otherwise, as *untriggered*. To evaluate the model's predictive capacity, predictions are compared with the known trigger outcomes found within the data set. More specifically, when the predicted outcome is incorrect (i.e., the prediction is *triggered* and the actual result is *not triggered*) the accuracy count is marked as zero (0) and, conversely, one (1) when the opposite occurs. In a case where 20 correct predictions are made for 80 items, the prediction accuracy is 20/80 (0.25 or 25% accurate).

Findings and Discussion

The prediction accuracy of the AdaBoost model is 0.824, while the logistic regression model features a McFadden's Pseudo-R^2 of 0.104. This means that only around 10% of the DDA variability is explained by the selected predictors within the logistic regression model, while the AdaBoost model can predict outcomes correctly in 82% of cases. This coincides with findings in previous research showing logistic regression to be less reliable.

Figure 2: Importance Weighting



The importance of each predictor in the AdaBoost model is analyzed to better understand the relative contribution of each factor to the overall prediction outcome. As shown in Figure 2, the importance weightings for *publisher*, *publication year*, *LC class*, and *price* are 84.85, 12.28, 2.69, and 0.17, respectively. Within the context of these data, and this particular model, *publisher* and *publication year* are the two most important predictors. While one would not typically expect *price* to be a determining factor in a DDA title being triggered for purchase, the fact that *LC class* is weighted so low within the AdaBoost model is surprising.

Although the relative importance of model features can be determined, as just illustrated, it is important to note that AdaBoost models are not as straightforwardly interpretable as regression model coefficients (i.e., *x* change of variable(s) will lead to *y* changes in the outcome). Nevertheless, the trained AdaBoost model can be used to predict the likelihood of new titles being triggered for purchase. Table 1 illustrates a series of prediction outcomes generated by feeding a small number of testing data through the trained AdaBoost model. As one can see, the first and the third items feature a lower estimated probability of being triggered (with a 16% and 9% probability, respectively), while the second and fourth items feature relatively high probabilities of being triggered for purchase (with a 94% and 77% probability, respectively).

Table 1

Trigger Estimation Probabilities

Publisher	Pub Year	LC	Price	Trigger *p*
John Wiley & Sons, Inc	2014	G	$217.00	0.16
John Wiley & Sons, Inc	2016	L	$65.00	0.94
Hachette Book Group	2016	T	$20.00	0.09
CRC Press (CAM)	2012	T	$175.00	0.77

Importantly, this example illustrates where those predictors of lower importance will have the greatest impact. While the first two records are produced by the same publisher in the same year, their trigger probabilities differ greatly. This is due to the fact that in cases where high-impact predictors are equal, the lower-impact predictors (in this case, *LC class* and the *price*) will provide greater influence over the final decision estimate. In essence, ties between high-impact predictors are broken by the differences seen in the lower-level predictors.

Conclusion

With an overall predictive capacity of 0.824, compared with the McFadden pseudo R^2 of 0.104 for the logistic regression model, the AdaBoost model shows great promise as an actionable predictive model. Libraries can deploy the machine learning approach described herein to assist in optimizing their commercial DDA program, or as the basis of an in-house DDA alternative. This type of in-house DDA would involve establishing pseudo-triggers based on minimal levels of content usage, as defined within the context of usage for e-book titles already owned by the library. As with the commercial DDA solution, these data would be used to train an AdaBoost model that predicts a dichotomous pseudo-trigger within a list of e-book titles not yet purchased. This type of analysis could be conducted one or two times per year, with a result the authors predict would closely align with vendor-based DDA program outcomes. Importantly, such an in-house program could reduce e-book costs by 30–50%.

Practical deployment aside, there remain other important questions surrounding the use of machine learning in library assessment. For example, consider that, unlike regression-based models that support explanatory generalizations to a broader population, machine learning models are typically quite opaque with regard to the causal factors underlying their predictions. This raises an important question:

Considering this lack of explanatory power, can the deployment of machine learning be considered assessment? As library scholars have noted, assessment is more than the tools libraries use to evaluate quality and value within their services, collections, and facilities. It is also about conveying the story of that quality or value to library stakeholders. It is about fostering an organizational culture of reflective practice and thoughtful, evidence-based planning.

Within the context of this research project, it is difficult to claim that machine learning is anything more than a tool of assessment. It does not provide a rich tapestry of feedback that leads to deeper understanding. It does not promote a culture of reflective practice. Nevertheless, for what machine learning lacks in the way of warm and fuzzy assessment goodness, it makes up with sharp and penetrating effectiveness as an engine of effective evidence-based decision making. Of course, as with any tool, especially those with a sharp edge, it must be wielded in a thoughtful and purposeful manner.

There are several lines of investigation that, moving forward, could successfully build upon the findings of this study. Outside of the aforementioned questions surrounding an ongoing, sustainable deployment of this technology, another line of inquiry would seek to determine if AdaBoost could be used to predict usage patterns for tangible print materials. Considering the important differences that distinguish the discovery/use of e-books from tangible print books, one might expect model parameters that differ from those witnessed in this study. Nevertheless, there is little reason to doubt the feasibility of such an approach.

Kevin W. Walker, PhD
Associate Professor
The University of Alabama
309F Amelia Gayle Gorgas Library
Box 870266
Tuscaloosa, AL 35487
Phone 205-348-1357
kwwalker@ua.edu

Zhehan Jiang, PhD
Assistant Professor
Peking University
Beijing, China
Institute of Medical Education
jiangzhehan@bjmu.edu.cn

Works Cited

[1] M. Gilbertson, E.c. McKee, and L. Salisbury, “Just in Case or Just in Time? Outcomes of a 15-Month Patron-Driven Acquisition of e-Books at the University of Arkansas Libraries,” *Library Collections, Acquisition and Technical Services* 38, no. 1–2 (January 2014): 10–20, https://doi.org/10.1080/14649055.2014.924072; Rick Lugg, “Collecting for the Moment: Patron-Driven Acquisitions as a Disruptive Technology,” in *Patron-Driven Acquisitions: History and Best Practices*, Current Topics in Library and Information Practice (Berlin; Boston: De Gruyter Saur, 2011); Jennifer Perdue and James A. Van Fleet, “Borrow or Buy? Cost-Effective Delivery of Monographs,” *Journal of Interlibrary Loan, Document Delivery & Information Supply* 9, no.4 (April 1999): 19.

[2] Kevin W. Walker and Michael A. Arthur, “Judging the Need for and Value of DDA in an Academic Research Library Setting,” *The Journal of Academic Librarianship* 44, no. 5 (September 2018): 650–62, https://doi.org/10.1016/j.acalib.2018.07.011.

[3] Kay Downey et al., “A Comparative Study of Print Book and DDA Ebook Acquisition and Use,” *Technical Services Quarterly* 31, no. 2 (April 2014): 139; Gilbertson, McKee, and Salisbury, “Just in Case or Just in Time?”; D. H. Longley, “Demand Driven Acquisition of E-Books in a Small Online Academic Library: Growing Pains and Assessing Gains,” *Journal of Library and Information Services in Distance Learning* 10, no. 3–4 (January 2016): 320–31, https://doi.org/10.1080/1533290X.2016.1221616.

[4] Downey et al., “A Comparative Study of Print Book and DDA Ebook Acquisition and Use.”

[5] Kwok Tai Chui et al., “Predicting At-Risk University Students in a Virtual Learning Environment via a Machine Learning Algorithm,” *Computers in Human Behavior*, June 27, 2018, https://doi.org/10.1016/j.chb.2018.06.032; Ryan Litsey and Weston Mauldin, “Knowing What the Patron Wants: Using Predictive Analytics to Transform Library Decision Making,” *The Journal of Academic Librarianship* 44, no. 1 (January 2018): 140–44, https://doi.org/10.1016/j.acalib.2017.09.004; Steve Mitchell, “Machine Assistance in Collection Building: New Tools, Research, Issues, and Reflections,” *Information Technology & Libraries* 25, no. 4 (December 2006): 190–216.

[6] V. Kishore Ayyadevara, “Basics of Machine Learning,” in *Pro Machine Learning Algorithms: A Hands-On Approach to Implementing Algorithms in Python and R*, ed. V Kishore Ayyadevara (Berkeley, CA: Apress, 2018), 1–15, https://doi.org/10.1007/978-1-4842-3564-5_1; Rodrigo Fernandes de Mello and Moacir Antonelli Ponti, “A Brief Review on Machine Learning,” in *Machine Learning: A Practical Approach on the Statistical Learning Theory*, ed. Rodrigo Fernandes de Mello and Moacir Antonelli Ponti (Cham: Springer International Publishing, 2018), 1–74, https://doi.org/10.1007/978-3-319-94989-5_1.

[7] Shankar Bellam, “Robotics vs Machine Learning vs Artificial Intelligence: Identifying the Right Tools for the Right Problems,” *Credit & Financial Management Review* 24, no. 2 (2018): 1–10; Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[8] Ayyadevara, “Basics of Machine Learning”; Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[9] Bellam, “Robotics vs Machine Learning vs Artificial Intelligence.”

[10] Ayyadevara, “Basics of Machine Learning”; Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[11] Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[12] Ayyadevara, “Basics of Machine Learning”; Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[13] Ayyadevara, “Basics of Machine Learning”; Fernandes de Mello and Antonelli Ponti, “A Brief Review on Machine Learning.”

[14] Mitchell, “Machine Assistance in Collection Building.”

[15] John Renaud et al., "Mining Library and University Data to Understand Library Use Patterns," *Electronic Library* 33, no. 3 (May 2015): 355.

[16] Martin Solis, "Perspectives to Predict Dropout in University Students with Machine Learning," *2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI), Bioinspired Intelligence (IWOBI), 2018 IEEE International Work Conference*, 2018, https://doi.org/10.1109/IWOBI.2018.8464191.

[17] Litsey and Mauldin, "Knowing What the Patron Wants."

[18] Karen Kohn, "Using Logistic Regression to Examine Multiple Factors Related to E-Book Use," *Library Resources & Technical Services* 62, no. 2 (April 2018): 54–65.

[19] Overfitting occurs when a predictive model is negatively influenced by unique aspects of those data on which it is constructed. Specifically, the model is so closely fitted to a particular set of data that it cannot reliably predict patterns outside of those data.

[20] Triggers associated with the DDA program under study adhere to the standard 10-10-1-1-1 model. This model dictates that a purchase is triggered when an e-book title, within the course of a single session for a single library user, experiences 10 minutes of viewing, 10 page views, 1 download, 1 copy, or 1 print.

[21] Each branch of the decision tree is a decision node, while leaf nodes are represented by the final decisions at the end of each branch.

[22] Statistical *noise* is variability within a set of data that cannot be explained by any single phenomenon. That is to say, such variations are the result of random chance.

[23] Pruning involves placing limitations on the total number of decision branch levels allowed within a given decision tree or set of trees. Jonathan Cheung-Wai Chan and Desiré Paelinckx, "Evaluation of Random Forest and Adaboost Tree-Based Ensemble Classification and Spectral Band Selection for Ecotope Mapping Using Airborne Hyperspectral Imagery," *Remote Sensing of Environment* 112, no. 6 (June 16, 2008): 2999–3011, https://doi.org/10.1016/j.rse.2008.02.011.

[24] *Random forest* is an ensemble learning algorithm, meaning it utilizes multiple learning algorithms within a single context to yield more accurate results than would have been the case with a single learning approach. In this case, the random forest method combines elements of statistical *bootstrapping*, *bootstrap aggregation* (bagging), and *decision tree learning*.

[25] Trevor Hastie, Robert Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Second edition, corrected 12th printing, Springer Series in Statistics (New York: Springer, 2017).

[26] Chan and Paelinckx, "Evaluation of Random Forest and Adaboost Tree-Based Ensemble Classification and Spectral Band Selection for Ecotope Mapping Using Airborne Hyperspectral Imagery"; Steven W. Knox, *Machine Learning: A Concise Introduction*, Wiley Series in Probability and Statistics (Hoboken, New Jersey: Wiley, 2018).

[27] David Muchlinski et al., "Comparing Random Forest with Logistic Regression for Predicting Class-Imbalanced Civil War Onset Data," *Political Analysis* 24, no. 1 (ed 2016): 87–103, https://doi.org/10.1093/pan/mpv024.

[28] Andreas Mayr et al., "The Evolution of Boosting Algorithms—From Machine Learning to Statistical Modelling," 2014, https://doi.org/10.3414/ME13-01-0122.

[29] Tianqi Chen and Carlos Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining—KDD '16*, 2016, 785–94, https://doi.org/10.1145/2939672.2939785.

[30] Liudmila Prokhorenkova et al., "CatBoost: Unbiased Boosting with Categorical Features," 2017.

[31] Yoav Freund and Robert E Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences* 55, no. 1 (August 1, 1997): 119–39, https://doi.org/10.1006/jcss.1997.1504.

[32] It is worth noting that there are very few R packages that support boosting. Further the Adabag package is widely used and been proven effective in the research setting. Further, it is arguably the best

boosting package for deployment by the layman. Other boosting packages require special runtime environments (e.g., Python, Linux, etc.) that many users may not be comfortable or competent in deploying.

[33] Mayr et al., “The Evolution of Boosting Algorithms—From Machine Learning to Statistical Modelling.”

[34] The AdaBoost algorithm is simpler to use—requiring fewer adjustments to the algorithm’s underlying analytical parameters (those parameters controlling the overall analysis).

[35] Freund and Schaphire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.”

[36] Additional information about the R programming language and environment can be found on the web at: https://www.r-project.org.

[37] See: Alfaro et al. (2013).

[38] See: Chatterjee (2016).

[39] RStudio Server, as opposed to RStudio, allows one to take advantage of faster, server-based analyses made possible through multiple-core processing.

[40] As per an often used standard, the training data set uses an 80% random sample of the original data, while the test data set uses a 20% random sample.

[41] For example, twenty (20) different models, or decision trees, will be created, optimized, and aggregated.

[42] For example, trees will not be allowed to grow more than three (3) decision nodes deep.

Appendix A

R Code for Logistic Regression Model

```
#Create data frame for a logistic regression model that uses "trigger" as the dependent variable and all
other variables within the DDA dataset as independent variables.

dda.logistic <- glm(trigger~., data = dda.data, family = binomial)

#Show regression model output.

summary(dda.logistic)
```

Appendix B

R Code for AdaBoost Model

```
#Load the appropriate R packages

library(“adabag”)

library(“caret”)

library(“rpart”)

library(“readxl”)

#Import excel or CSV data file with all data and assign it to a named data frame (e.g., "dda.data")

dda.data <- "filename.csv"

#Inspect the structure of the newly created data frame.

str(dda.data)

#Update and data types that are not correct. In this case, several variables had to be changed to "factor"
type variables.

dda.data$trigger <- as.factor(dda.data$trigger)

dda.data$publisher <- as.factor(dda.data$publisher)

dda.data$pub.year <- as.factor(dda.data$pub.year)

dda.data$lc1let <- as.factor(dda.data$lc1let)

#Create indicator that marks two subsets of data - one for training and one for testing - using a random
80/20 sampling method.

set.seed(1234)

ind <- sample(2,nrow(dda.data), replace = T, prob = c(0.8, 0.2))
```

#Define data frames for both training and testing data sets - identified by their previously defined sample indicator of '1' or '2'.

```
train <- dda.data[ind==1,]
```

```
test <- dda.data[ind==2,]
```

#Feed the training data into the AdaBoost algorithm to create a new fitted model type (of the boosting class) named “dda.adaboost.” Note, the learner type is identified as 'Breiman', which is the random forests learner.

```
dda.adaboost <- boosting(trigger ~ ., data = train, boos = TRUE, mfinal = 20, control =
rpart.control(maxdepth=3), coeflearn = 'Breiman')
```

```
dda.adaboost
```

```
barplot(dda.adaboost$imp[order(dda.adaboost$imp,decreasing=TRUE)], ylim = c(0, 100),
main="Variables relative importance", col = "lightblue")
```

#Run the test data through the dda.adaboost model. Show testing model output, including prediction error.

```
dda.predboost <- predict.boosting (dda.adaboost, newdata = test, coeflear = 'Breiman')
```

```
dda.predboost$class
```

```
dda.predboost$prob
```

```
dda.predboost$confusion
```

```
dda.predboost$error
```