Title: Averaging and boosting methods in ensemble-based classifiers for text readability

Author: Ruslan Korniichuk, Mariusz Boryczka

Citation style: Korniichuk Ruslan, Boryczka Mariusz. (2021). Averaging and boosting methods in ensemble-based classifiers for text readability. "Procedia Computer Science" (Vol. 192 (2021), s. 3677-3685), DOI:10.1016/j.procs.2021.09.141

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 192 (2021) 3677–3685

Procedia Computer Science

www.elsevier.com/locate/procedia

25th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

Averaging and Boosting Methods in Ensemble-Based Classifiers for Text Readability

Ruslan Korniichuk*, Mariusz Boryczka

University of Silesia in Katowice, Institute of Computer Science, Będzińska 39, 41-200 Sosnowiec, Poland

Abstract

The purpose of this paper is to investigate whether it is possible to predict text readability with ensemble-based classifiers. In this article, the authors calculated and analyzed the readability indices. In the next stage, they defined additional features for each text and determined the relationships between readability and features. Among the various tasks of machine learning, they chose the classification problem. The authors calculated and compared the accuracy of different machine learning models. After building the models, they proceeded to the Random decision forests model interpretation step using the SHAP method. The authors show that machine learning models based on only three features are capable of predicting text readability. Long sentences and a low percentage of stop words can cause low readability. The machine learning model shown in this paper allows to classify texts according to readability with a model accuracy of 0.9.

Keywords: Ensemble Methods; Averaging Methods; Boosting Methods; Classification; Explainable Prediction; Readability Indices

1. Introduction

Readability makes some texts easier to read and understand than others. Readability is often confused with legibility, which refers to the visual clarity of individual symbols.

* Corresponding author. Tel.: +48 791 065 877.
E-mail address: ruslan.korniichuk@gmail.com

10.1016/j.procs.2021.09.141

The concept of readability is related to the ease of reading and comprehension of written texts. When assessing readability, several factors should be taken into account, such as the average sentence length, the number of difficult words in the text, and the grammatical complexity of the language used [18].

This paper aimed to investigate whether it is possible to predict text readability with ensemble-based classifiers using averaging and boosting methods. The data was obtained from Webhose Ltd. (https://webhose.io)—the leading data provider turning unstructured web content into machine-readable data.

The remaining part of the paper is organized as follows. Section 2 provides a brief description of the problem behind the paper. Section 3 presents a description of the selected methods for assessing text readability and the algorithms used to build and interpret the machine learning model. Section 4 describes the research conducted, the results obtained, and their interpretation. The paper concludes with a summary, included in Section 5.

2. Problem description and related works

The subject of our analysis is the texts examined in terms of their readability. Readability refers to how easy it is to read and understand a text, depending on its specific unique characteristics. The readability index, in turn, is a measure related to the difficulty of text perception by the reader. It can be calculated based on various attributes: word/sentence length, number of multi-character/polysyllabic/difficult words, etc.

However, in this paper, we do not deal directly with the analysis of text readability. Our work is the study of the possibility of predicting the level of text readability using ensemble-based classifiers. Therefore, we do not describe here the methods of calculation of text readability themselves, but instead, ensemble methods.

The purpose of ensemble methods is to combine the predictions of several base estimators to improve predictive performance over a single estimator.

In general, these methods vary in the way they construct various classifiers and combine their predictions. The first step of constructing a group of classifiers can be differentiated according to the dependencies among classifiers. The independent approach trains classifiers randomly, for example, Bagging [2] and Random decision forests [3]. The dependent approach constructs a new classifier while taking advantage of knowledge obtained during the construction of past classifiers, such as AdaBoost [13] and Gradient tree boosting [14].

In the second step of combining the classifiers' predictions, majority voting is one intuitive method to choose the dominant decision [2-3, 23]. As majority voting cannot guarantee that the voting result will be better than the best individual one, the weighting method is introduced, which assigns competent classifiers higher weights, such as performance weighting [11, 13-14, 24], Naive Bayes weighting [9], and entropy weighting [19].

The main two families of ensemble methods are usually distinguished: averaging methods, boosting methods. In averaging methods, the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimators because its variance is reduced.

In boosting methods, base estimators are built sequentially, and one tries to reduce the bias of the combined estimator. The motivation of the approach is to combine several weak models to produce a powerful ensemble.

3. Methodology

3.1. Readability indices

The readability indices used in this study can be divided into several groups. The first group is based on word length counted in syllables and sentence length. Longer words and longer sentences are more complex for reading and comprehension. The group includes the Flesch [12], Flesch-Kincaid [10], Fog [1], and Strain [21] readability indices.

The second group is based on word length counted in characters and sentence length. These include the Automated Readability Index (ARI) [20], Coleman-Liau [4], and Rix [5] readability indices.

The next readability index used in this paper, the New Dale-Chall [7-8], is unique. This index is based on the sentence length and the number of difficult words. Initially, this readability index was based on a list of words that every statistical American student is required to understand before their senior year. Words that are not on this list are considered difficult to read and understand.

The last readability index, the Bormuth [22], combines the approaches of the New Dale-Chall readability index and the second group of indices.

3.2. Machine learning algorithms used in experiments

The decision tree is a non-parametric supervised learning method. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

The AdaBoost [13] is an ensemble meta-algorithm that may be used in conjunction with many other types of learning algorithms to improve their performance. Subsequent weak learners are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost is a particular case of the boosting methods family. An implementation of the AdaBoost for decision tree algorithm was used.

The *k*-nearest neighbors (*k*-NN) algorithm [6] is a neighbors-based classification—a type of instance-based learning or non-generalizing learning. It does not attempt to construct a general internal model but simply stores instances of the training data. Classification is computed from a simple majority vote of the nearest neighbors of each point.

Bagging [2] is an ensemble meta-algorithm designed to improve the accuracy of non-meta machine learning algorithms. While it is typically used for decision tree methods, it can be used with any type of method. Bagging is a particular case of the averaging methods family. The applied implementation is based on the Bagging for *k*-NN.

Gradient tree boosting ensemble meta-algorithm is a generalization of boosting to arbitrary differentiable loss functions. It is an accurate and effective off-the-shelf procedure that can be used for classification. Gradient tree boosting is a particular case of the boosting methods family.

Random decision forests [22] are a machine learning method that involves constructing multiple decision trees during learning time and generating a class that is the dominant class of each tree. Random decision forests are a particular case of the averaging methods family.

3.3. Interpretation of machine learning models

SHAP is an approach to explain individual predictions [15-16]. It assigns each feature an importance value for a particular prediction. SHAP is based on the theoretically optimal Shapley values. The Shapley value, in turn, is a solution concept in cooperative game theory. To each game, it assigns a unique distribution of a total surplus generated by the coalition of all participants.

4. Experiments

The flow of experiments was as follows. We started with the preparation of the text data. We added 9 readability indices with their interpretations: Flesch, Flesch-Kincaid, Fog, Strain, Automated Readability Index (ARI), Coleman-Liau, Rix, New Dale-Chall, and Bormuth.

Next, we created the *readability_class* column describing the target classes. We assigned the data to class 0, class 1, or class 2 based on the median value of interpretations of readability indices.

Class 0 represents texts that are confusing to read and understand. Class 1 means standard texts, while class 2 represents easy-to-read texts.

We specified an additional 11 features (listed in Table 1) at the next stage and determined the dependencies between the features and the target variable *readability_class*. Before the stage of building machine learning models, we solved the problem of strongly correlated features and decreased the number of features from 11 to 3 features: average number of characters per sentence (*acs*), percentage of stop words (*psw*), and percentage of marketing words (*pmw*).

4.1. Data preparation

The initial data set [25] contained 499,610 English news articles originated in the US from the top 1,000 (based on the ranking provided by Alexa) news sites. The data crawled during November 2016. We cleaned up text data before readability indices calculation.

4.2. Analysis of additional features

Table 1 includes the complete list of added features with the Pearson correlation coefficient between the readability and the feature. Long sentences and long words can cause low readability. A high percentage of difficult words (according to the New Dale-Chall readability index) can cause low readability. On the other hand, a low percentage of marketing and stop words can cause low readability.

We defined the level of correlation between the readability and the average number of characters per sentence (*acs*) as high. Also, we determined the level of correlation between the readability and the percentage of stop words (*psw*) as high. Finally, we defined the level of correlation between the readability and percentage of marketing words (*pmw*) as low.

Since we had features with low and high significance of the correlation between readability and feature, we decided to move on to building machine learning models.

4.3. Building machine learning models

We obtained 173,421 examples during the machine learning model building stage. That is because we decided to include 57,807 examples from each readability class (0/1/2) to achieve a perfectly balanced set.

There are 3 main parameters for the Decision tree model tuning to be described: *min_samples_split*, *min_samples_leaf*, *max_features*. In the final version of the model, nodes are expanded until all leaves are pure or until all leaves contain less than 2 samples (*min_samples_split*). The *min_samples_leaf* parameter is equal to 1. The parameter describes the minimum number of samples required to be at a leaf node. The *max_features* parameter is equal to 3 and represents the number of features to consider when looking for the best split.

An implementation of the AdaBoost for decision tree algorithm was used as the next model. The *n_estimators* parameter is equal to 5. The *n_estimators* is the maximum number of estimators at which boosting is terminated.

The *n_neighbors* parameter of the *k*-NN model is equal to 19. The *n_neighbors* parameter describes the number of neighbors to use by default to find the nearest neighbors of a point.

Table 1. The complete list of added features.

Feature	Pearson correlation coefficient
the average number of syllables per sentence	−0.649444
the average number of syllables per word	−0.620834
the average number of characters per sentence (*acs*)	−0.615322
percentage of difficult words (according to the New Dale-Chall readability index)	−0.588468
the average number of words per sentence	−0.585447
percentage of polysyllabic words	−0.562748
percentage of multi-character words	−0.537126
percentage of echomimetic (onomatopoeic) words	−0.090742
percentage of unique words	0.028991
percentage of marketing words (*pmw*)	0.130881
percentage of stop words (*psw*)	0.526561

An implementation of the Bagging for *k*-NN algorithm was used as the next model. The *n_neighbors* parameter is also equal to 19. The *n_estimators* parameter is equal to 43. The *n_estimators* is the number of base estimators in the ensemble.

The *n_estimators* parameter of the Gradient tree boosting model is equal to 150. The *n_estimators* parameter represents the number of boosting stages to perform.

The final model used in this paper, the Random decision forests, is the model with the best classification accuracy. The *n_estimators* parameter is equal to 118. The *n_estimators* is the number of trees in the forest.

Table 2 shows the complete list of built machine learning models along with their accuracy and F_1 score for each readability class.

4.4. Interpretation of the Random decision forests model

To analyze the results, we used the KernelExplainer of the SHAP package. The mean absolute value of the SHAP values can show how much each feature contributed to predicting the value of the target variable.

Fig. 1 represents the feature importance graph. The graph lists the most significant features in descending order of importance. The top features contribute the most to the model. The lower a feature is, the weaker it is—it has less predictive power. Thus, the two strongest features in our model are *acs* and *psw*.

Table 2. The quality of the machine learning models.

Model name	Ensemble method family	Accuracy	F_1 score (for classes 0/1/2)
Decision tree	n/a	0.864051	0.92/0.80/0.88
AdaBoost for decision tree	boosting methods	0.880254	0.94/0.80/0.89
k-NN	n/a	0.892959	0.94/0.83/0.90
Bagging for *k*-NN	averaging methods	0.893402	0.94/0.83/0.90
Gradient tree boosting	boosting methods	0.897015	0.94/0.84/0.91
Random decision forests	averaging methods	0.897880	0.94/0.84/0.91

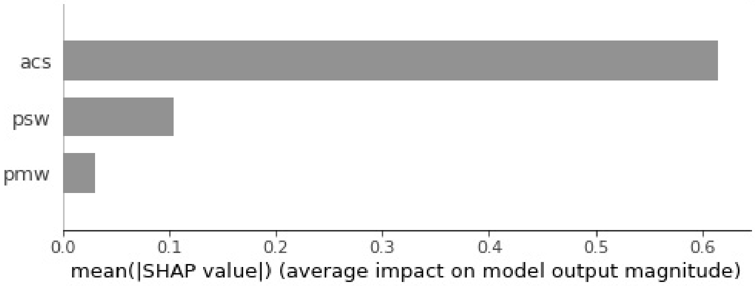

Fig. 1. Global feature importance.

Now let us look at the so-called dependency graphs, which show whether and what kind of dependency exists between the target and the object. Fig. 2 shows that the relationship between the aim and the object exists. There is a negative correlation between readability and the average number of characters per sentence (*acs*). Also, there is a positive correlation between the readability and the features *psw* and *pmw*. However, Fig. 2 shows a low correlation between the readability and percentage of marketing words (*pmw*).

We also checked differently the influence of the *acs*, *psw*, and *pmw* features on the output of the model. Fig. 3–5 show so-called force plots. Force plots visualize the given SHAP values with an additive force layout [17]. We can observe there when the value of a feature has a positive and when a negative effect on the value of the target variable *readability_class*.

Fig. 3 shows that the average number of characters per sentence (*acs*) negatively affects text readability. Long sentences can cause low readability. The readability class changes with an average sentence length of about 150 characters.

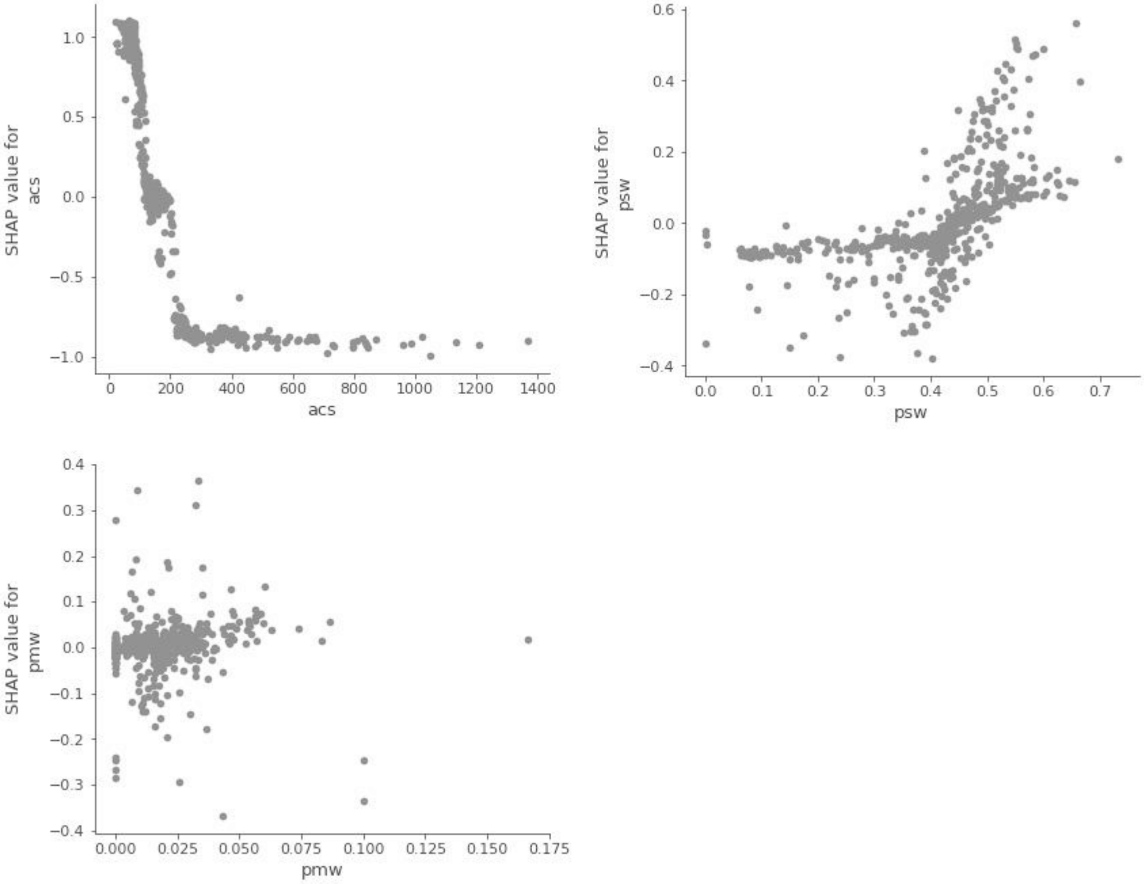

Fig. 2. Dependence between features (*acs*, *psw*, *pmw*) and the target variable *readability_class*.

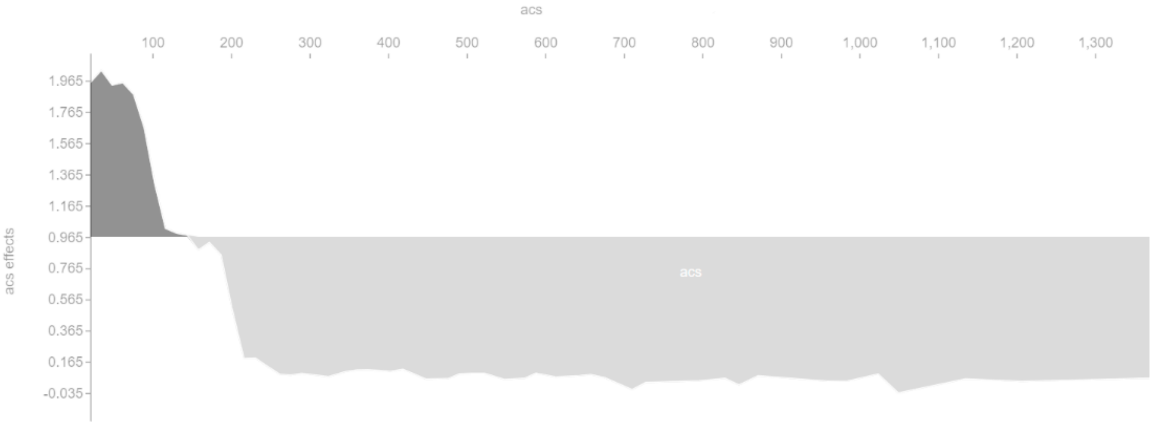

Fig. 3. Positive and negative influence of the *acs* feature on the target variable *readability_class*.

Fig. 4 shows that the percentage of stop words (*psw*) positively affects text readability. A low percentage of stop words can result in low readability. The readability class changes with a percentage of stop words of about 44%.

Fig. 5 shows a low level of dependency between the readability and percentage of marketing words (*pmw*). The low percentage of marketing words can cause low readability.

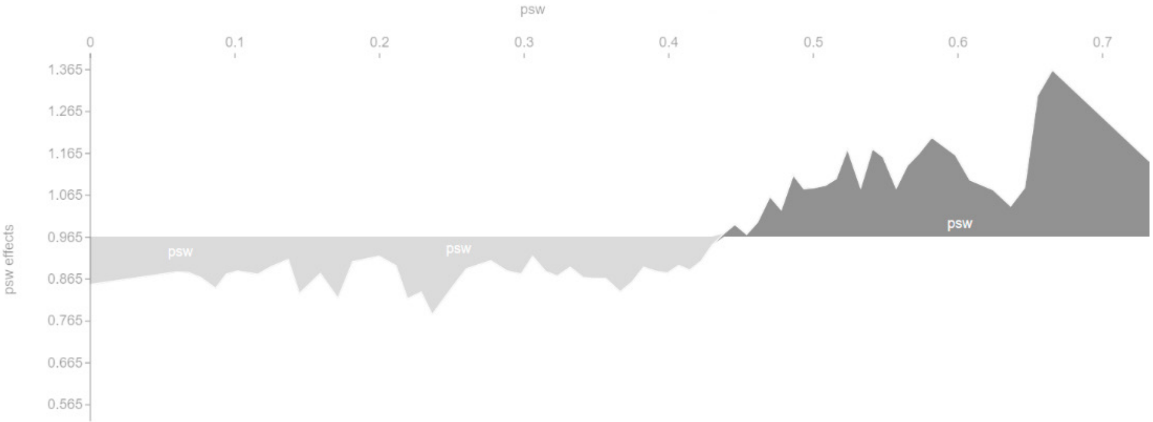

Fig. 4. Positive and negative influence of the *psw* feature on the target variable *readability_class*.

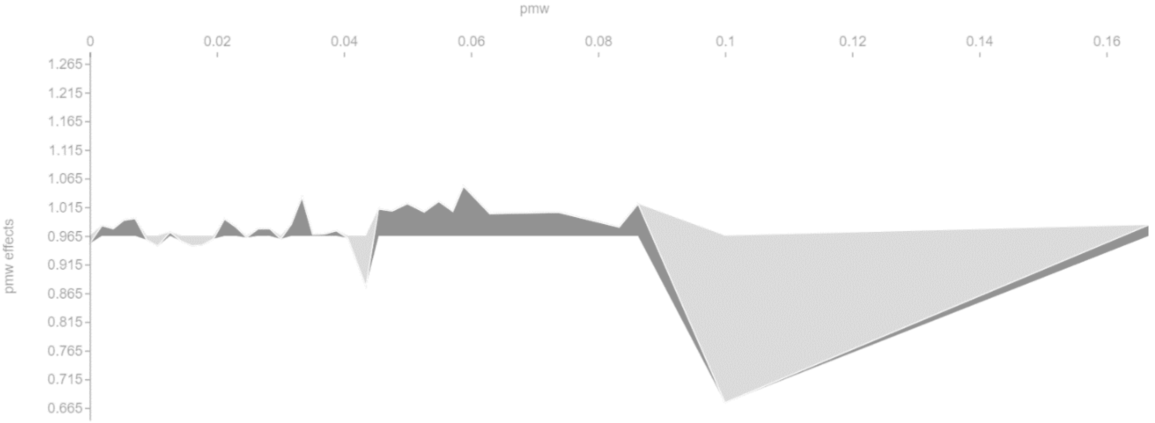

Fig. 5. Positive and negative influence of the *pmw* feature on the target variable *readability_class*.

5. Conclusions

In this article, we show, using a data set [25] containing English news articles originated in the US from the top 1,000 news sites, that averaging and boosting methods in ensemble-based classifiers can predict text readability. The Random decision forests model shown in this paper allows to classify texts according to readability with a model accuracy of 0.9.

Readability classification can be achieved based on just three features: average number of characters per sentence (*acs*), percentage of stop words (*psw*), and percentage of marketing words (*pmw*). The two strongest features in our model are *acs* and *psw*. Long sentences and a low percentage of stop words can cause low readability. On the other hand, we defined the level of correlation between the readability and percentage of marketing words (*pmw*) as low.

We also showed that both averaging and boosting method can improve model accuracy. Ensemble-based classifiers are more useful for weak (e.g., decision trees) rather than strong non-meta machine learning algorithms.

The next important step will be to analyze a broader list of features that can affect text readability. They are, for example, text type, average paragraph length, percentage of passive voice constructions, percentage of transition words, sentiment.

References

[1] Bogert, J. (1985) "In Defense of the Fog Index." *The Bulletin of the Association for Business Communication* **48** (**2**): 9–12. doi:10.1177/108056998504800203

[2] Breiman, L. (1996) "Bagging Predictors." *The Journal of Machine Learning Research* **24** (**2**): 123–140.

[3] Breiman, L. (2001) "Random Forests." *The Journal of Machine Learning Research* **45** (**1**): 5–32.

[4] Coleman, M., and Liau, T. L. (1975) "A Computer Readability Formula Designed for Machine Scoring." *Journal of Applied Psychology* **60** (**2**): 283–284. doi:10.1037/h0076540

[5] Courtis, J. K. (1987) "Fry, Smog, Lix and Rix: Insinuations About Corporate Business Communications." *Journal of Business Communications* **24** (**2**): 19–27. doi:10.1177/002194368702400202

[6] Cover, T. M., and Hart, P. E. (1967) "Nearest Neighbor Pattern Classification." *IEEE Transactions on Information Theory* **13** (**1**): 21–27. doi:10.1109/TIT.1967.1053964

[7] Dale, E., and Chall, J. S. (1948a) "A Formula for Predicting Readability." *Educational Research Bulletin* **27** (**1**): 11–20, 28.

[8] Dale, E., and Chall, J. S. (1948b) "A Formula for Predicting Readability: Instructions." *Educational Research Bulletin* **27** (**2**): 37–54.

[9] Domingos, P., and Pazzani, M. (1997) "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss." *The Journal of Machine Learning Research* **29** (**2**–**3**): 103–130.

[10] DuBay, W. H. (2004) *The Principles of Readability*, Impact Information.

[11] Eibl, G., and Pfeiffer, K.–P. (2005) "Multiclass Boosting for Weak Classifiers." *The Journal of Machine Learning Research* **6**: 189–210.

[12] Flesch, R. (1948) "A New Readability Yardstick." *Journal of Applied Psychology* **32** (**3**): 221–233. doi:10.1037/h0057532

[13] Freund, Y., and Schapire, R. E. (1996) "Experiments with a New Boosting Algorithm," in *Machine Learning: Proceedings of the Thirteenth International Conference* (pp. 148–156), Morgan Kaufmann Publishers.

[14] Friedman, J. H. (2002) "Stochastic Gradient Boosting." *Computational Statistics & Data Analysis* **38** (**4**): 367–378.

[15] Lundberg, S. M., and Su-In, L. (2017) "A Unified Approach to Interpreting Model Predictions," in I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds) *Advances in Neural Information Processing Systems* 30 (pp. 4765–4774), Curran Associates.

[16] Lundberg, S. M., Erion, G., Hugh, C., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., and Su-In, L. (2020) "From Local Explanations to Global Understanding with Explainable AI for Trees." *Nature Machine Intelligence* **2**: 56–67. doi:10.1038/s42256-019-0138-9

[17] Lundberg, S. M (2018) *SHAP latest documenation* [online]. Available at https://shap.readthedocs.io/en/latest/generated/shap.plots.force.html.

[18] Richards, J. C., and Schmidt R. (2010) *Longman Dictionary of Language Teaching and Applied Linguistics* (4th ed.), Pearson Education.

[19] Shen, C., and Li, H. (2010) "On the Dual Formulation of Boosting Algorithms." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **32** (**12**): 2216–2231.

[20] Smith, E. A., and Senter, R. J. (1967) "Automated Readability Index," technical report AMRL-TR-66-220, Aerospace Medical Research Laboratories. Available at https://apps.dtic.mil/sti/pdfs/AD0667273.pdf.

[21] Solomon, W. (2017) *A Quantitative Analysis of Media Language*, Lambert Academic Publishing.

[22] Tin Kam, H. (1995) "Random Decision Forests." *Proceedings of 3rd International Conference on Document Analysis and Recognition* **1**: 278–282. doi:10.1109/ICDAR.1995.598994

[23] Tin Kam, H. (1998) "The Random Subspace Method for Constructing Decision Forests." *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20** (**8**): 832–844.

[24] Wang, H., Fan, W., Yu, P. S., and Han, J. (2003) "Mining Concept-Drifting Data Streams using Ensemble Classifiers," in *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)* (pp. 226–235), Association for Computing Machinery (ACM). doi:10.1145/956750.956778

[25] Webhose (2016) "English news articles." *Free Datasets for Machine Learning and Data Mining* [online]. Available at https://webhose.io/datasets/.