Statistical Machine Learning for Foodborne Disease Source Attribution

Amos Okutse Zexuan Yu Rophence Ojiambo

23 October, 2022

METHODOLOGY

In this section, we present a summary of the potential methods for consideration into our analysis and their implementation including the model building process, model validation including tuning parameters, model evaluation and selection including the considered performance metrics. We also highlight the potential software and justify choices associated with the implemented methods including the specific packages considered for our analysis.

Objective:

To develop a throughput statistical machine learning classification model for *Listeria monocytogene* pathogen food source attribution.

Statistical methods

This section presents a summary of the potential methods for inclusion into our analysis based on the scope of the project which is foodborne disease source attribution modeling using *Listeria monocytogene* pathogens.

Naïve Bayes The Naïve Bayes algorithm is a supervised machine learning technique for classification that uses a Bayesian-based approach under the *naïve* assumption that features in a data set are independent of each of each other. The methodology proceeds by using conditional probability and the principles of the Bayes theorem to classify an input based on features present in a dataset. The Bayes theorem which forms a fundamental building block of the algorithm follows as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

where A is the proposition, B is the evidence, $P(A|B)$ denotes the posterior probability, $P(B|A)$ denotes the likelihood, $P(A)$ is the prior probability of event A and $P(B)$ denotes the prior probability of the evidence, B [1].

In order for inferences to be made about the an unknown observation, the method proceeds by first building a distribution for the unknown observation unconditioned on the observed data, that is, the prior predictive distribution. Given the data have been observed, the prior predictive distribution is then updated to yield the posterior predictive distribution. The outcome, Y is often dependent on multiple features, p which can be denoted as $\mathbf{X} = x_1, x_2, \cdots, x_p$ and one or more output classes, $\mathbf{C} = C_1, C_2, \cdots, C_k$. The goal of the algorithm

is to determine the conditional probability of an event with features, $\mathbf{X}$ belonging to a class, C_k. Based on the Bayes theorem, these can be written as:

$$P(C_k|\mathbf{X}) = \frac{P(C_k)P(\mathbf{X}|C_k)}{P(\mathbf{X})} \tag{2}$$

The Naive Bayes algorithm allots a class label $\hat{y} = C_k$ for some k and is defined as:

$$\hat{y} = \underset{k\in\{1,\cdots,k\}}{\arg\max}\, P(C_k)\prod_{i=1}^{p} P(x_i|C_k) \tag{3}$$

The suitability of this algorithm for our particular analyses followed from our interest in perform multi-class and potentially, multi-label classification. The algorithm is easy to implement, is relatively fast to train and in classifying unknown individuals, and has been shown to perform relatively well with minimal to no hyper-parameter tuning when compared with other algorithms. Additionally, the algorithm uses evidence to update the estimation of the posterior and is robust [2].

Random forest Random forests are an ensemble of multiple decision trees which employ bagging to reduce over-fitting and thus enhance the generalizability of the resulting model. Multiple trees are fitted on the training data using randomly selected predictors to ensure that the resulting trees are uncorrelated. In the training of each tree, some samples are not used in the training process and are used in the estimation of the model performance by averaging. Each of the trees building up the random forest makes its prediction/classification to results in multiple predictions which are then averaged to inform the final result. For an in-depth analysis of the method, see Breiman [3].

Bayesian additive regression trees [BART] BART is an ensemble machine learning methodology that allows flexible modeling of interactions in non-linear regressions and classification problems in the Bayesian context. The method involves sums of classification or regression trees and is able to flexibly capture non-linearities. The estimation of the unknown function, f, involves recursive partitioning of the variable space using a fully Bayesian probability modeling approach [4]. Every tree in this process is a weak learner that can only explain a portion of the variability in the outcome. The trees form decision trees which are highly interpretable and flexible but are prone to overfitting. Overfitting, in BART, is controlled using a regularization prior that forces every tree in the model to explain only a small amount of the relationships between the outcome and the predictor variables. The model can be summarized as:

$$\begin{aligned} E[Y|X] &= f(\mathbf{X}) + \epsilon \\ &= T_1^M(\mathbf{X}) + \cdots + T_m^M(\mathbf{X}) + \epsilon \,\text{where}\, \epsilon = N(0, \sigma^2\mathbf{I}_n) \end{aligned} \tag{4}$$

where Y denotes the vector of outcomes, $\mathbf{X}$ denotes the $n \times p$ design matrix where p denotes the number of predictors in the model and ϵ denotes some noise. m is the number of unique trees, T denotes tree structure, and M denotes the terminal node or leaf parameters. BART models consist of three prior components including tree structure, leaf parameter, and error variance. Chipman et al. [4] presents an in-depth discussion of this methodology.

Model building

Data curation and preprocessing

The data considered for analysis had a very high percentage of missing data which rendered the use of data imputation methods unreasonable. The inclusion of potentially predictive features in the model resulted in

the exclusion of all possible observations under a complete case analysis. As such, the analysis that followed was limited to the variables we deemed useful in linking a particular *Listeria monocytogene* strain to a food source. The features which we deemed relevant for this analysis included the food source, the minimum single nucleotide polymorphism (SNP) distance to another isolate of the same isolation type for example, the minimum SNP distance from one clinical isolate to another clinical isolate (`Min.same`), the minimum SNP distance to another isolate of a different isolation type (`Min.diff`), the microbial strain name used to distinguish a genetically distinct lineage separated from another strain by one or two mutations (`Strain`), the isolate, and the location (state) from which the particular isolate was collected, all of which were informed by literature [5], [5], [6].

The food source variable employed was recategorized into 38 unique levels, however, for simplicity and to avoid potential empty levels characterized by food source categories with a limited number of samples, we restricted our analysis to the top 8 food categories which had $n \geq 150$ samples, that is, water (n = 1343 samples), dairy (n = 935 samples), fish (n = 244 samples), beef (n = 227 samples), pork (n = 214 samples), avocado (n = 209 samples), chicken (n = 168 samples), and beans (n = 150 samples). Even though the environment as a food source had a substantial number of samples linked to it, we chose to exclude this level considering the breadth of the term. We also excluded unknown food types from the outcome considering their uninformative nature. The resulting data set had n = 2143 *Listeria monocytogene* isolates which we use in all subsequent statistical modeling processes. We split into the training and testing sets with ¾ of the observations used in the training and the remaining ¼ of the observations used in the model validation. The split criteria considered the probable imbalance in the outcome food source by stratifying the way in which this split was performed using the food source variable. Table 1 highlights the proportions of the food sources in the train and test samples, respectively. All models are trained on the same dataset using the same features to reduce the potential effect of varied model building strategies on the resulting model performance estimates and thus also enhance comparability of the performance metrics. In the following section, we describe the specific implementation strategies considered for each of the implemented algorithms.

Implementation

(1) Naïve Bayes algorithm

The Naïve Bayes algorithm will be implemented using the `naive_bayes()` function which uses Bayes' theorem to calculate the probabilities of belonging to a given class. The implementation of this method used workflows derived from `tidymodels` package [7]. The `naive_bayes()` function takes in four arguments including the outcome mode (e.g., unknown, regression, or classification), the engine specifying what computational engine would be utilized in the model fitting. The function also requires the specification of two hyperparameters for tuning, that is, smoothness a non-negative number for controlling the smoothness of the boundaries with smaller values resulting in flexible models and larger values resulting in less adaptable class boundaries and a value for the Laplace correction to allow smoothing low-frequency counts [8]. The implementation of the Naïve Bayes algorithm using `tidymodels` and the `naivebayes` engine which also requires the extension package `discrim` [9]. The value for the Laplace smoothing is set to $\alpha = 1$ a value which has been shown to be appropriate in handling zero probability incidences in data.

(2) Random forest

The random forest algorithm will be implemented using the `rand_forest()` function which defines a model for the creation of a substantial number of trees that are independent of each other. The final classification follows from averaging/bagging or majority voting. The model will use the `ranger` engine, a workflow, similarly derived from the `tidymodels` package [7]. The `rand_forest()` function takes in five arguments including the outcome mode of the prediction, the engine used in model fitting, and tuning parameters `mtry`, `trees`, and `min_n` corresponding to the number of predictors on which to split the creation of a tree, the number of trees in the ensemble, and the minimum number of data points in a node for an additional split to be performed at this node. Hyperparameter tuning proceeded using a space-filling design with 25 candidate models using the `tune_grid()` function and the `show_best()` metric set to the AUC. For this process, the

training data is split further using the `validation_split()` function with 20% of the samples used in the validation. The `ranger` [10] package is used for parallel processing to allow computational efficiency using 1000 trees.

(3) Bayesian additive regression trees [BART]

The Bayesian

Model evaluation

All statistical machine learning models implemented in this project will be evaluated based on their classification accuracy as well as the area under the receiver operating characteristic curve (AUC). The accuracy in a classification problem context refers to the proportion of correctly classified instances and is a useful metric in determining how a given model discounts or recognizes a condition [11]. The choice of the accuracy as a performance measure is informed by our intention to compare the performance of different algorithms and the fact that this measure allows for model comparisons since it is more related to the data as well as the objective function. Even so, we note the limitation associated with the accuracy as an evaluation metric, particularly in contexts characterized by multi-class classification problems because of the lack of transparency about how the accuracy breaks down within each potential outcome category [11]. Given the limitation associated with the accuracy as a measure of model performance, we also considered Cohen's kappa [12], a measure similar to the accuracy but which is normalized by the accuracy that would be expected by a model that assigns units to classes with an equal probability, a chance-based model, and is thus relatively suitable for situations in which classes are likely to show disproportionate frequency distributions.

On the other hand, the ability of the models to discriminate between the different food sources will be expressed using the AUC which has been shown to be a suitable measure when summarizing the diagnostic accuracy of a test [11], [13]. The AUC is bounded between 0 and 1 with 1 corresponding to a perfectly accurate model and 0 to a perfectly inaccurate model. An AUC value of 0.5 suggests a model with no discriminatory ability whereas values within the range of 0.7 to 0.8 are considered acceptable and values >0.8 considered excellent. For all models, we test the null hypothesis that the model's discriminatory ability is not significantly different from 0.5 [13]. We also examine the gain associated with using each model in the classification as opposed to random guessing [14]. Our analyses do not consider, explicitly, performance metrics such as sensitivity, specificity, negative and positive predictive values which, despite being useful metrics relative to the accuracy, for example, are decision threshold-dependent and thus do not provide a unique account of diagnostic performance [11]. The computation of the performance metrics were implemented using the `accuracy()` , `kap()`, `roc_auc()` functions for accuracy and Cohen's kappa, respectively in package `yardstick` [15], [16] implemented in the `tidymodels` suite of model building tools. Model performance metrics are presented using tables and graphics using `kable()`, `gain_curve()`, and `roc_curve()` respectively.

Model selection

Model selection was based on the performance metrics discussed in to determine a suitable model with a combination of the highest accuracy as well as ability to discriminate between the different food sources to facilitate a relatively accurate linkage of the varied *Listeria monocytogene* pathogenic isolates to a potential food source and thus facilitate contact tracing and outcome investigation rapidly. The best model based on the metrics above will be trained on the full set of features to allow the model to learn the heterogeneity in the full data set and thus enhance robustness. This approach has been employed elsewhere [5], [17] and shown to yield substantially throughput results for predictive models.

Statistical software, code, and data availability

All statistical analyses were performed using the open-source R statistical programming environment, version 4.2.1 [18]. All package and model building extension have been highlighted in the text where relevant and are all available as open source distributions. All data and code employed in all analysis as part of this project are available online on GitHub at https://github.com/okutse/foodborne_diseases

PRELIMINARY RESULTS

This section highlights the preliminary results based on the implementation of the Naïve Bayes algorithm. Other models which will be included here are random forest and Bayesian Additive Regression Trees (BART).

Table 1 highlights the proportion of samples under each selected food category in the training and testing sets of the data. We notice that water, dairy, and beef have slightly higher proportions in the train and test data sets, respectively. Stratified sampling allows the data to be split with particular consideration of the proportion in each food category.

Table 1: Proportion in training and test datasets by food source

Food source	Train set sample size	Train set proportion	Test set sample size	Test set proportion
avocado	81	0.050	28	0.052
beans	74	0.046	23	0.043
beef	117	0.073	29	0.054
chicken	61	0.038	24	0.045
dairy	368	0.229	131	0.244
fish	113	0.070	35	0.065
pork	96	0.060	24	0.045
water	696	0.433	243	0.453

Model performance

Table 2 presents the preliminary results based on the implementation of the Naïve Bayes algorithm. Figure 1 and Figure 2 on the other hand present the results based on the analysis of the gain associated with the use of this model relative to random guessing. All results suggest that the model has a reasonable accuracy as well as discriminatory ability.

Table 2: Performance metrics based on the implementation of the Naive Bayes algorithm for *Listeria monocytogene* pathogen food source attribution

.metric	.estimator	.estimate
accuracy	multiclass	0.723
kap	multiclass	0.594
mn_log_loss	multiclass	0.888
roc_auc	hand_till	0.823

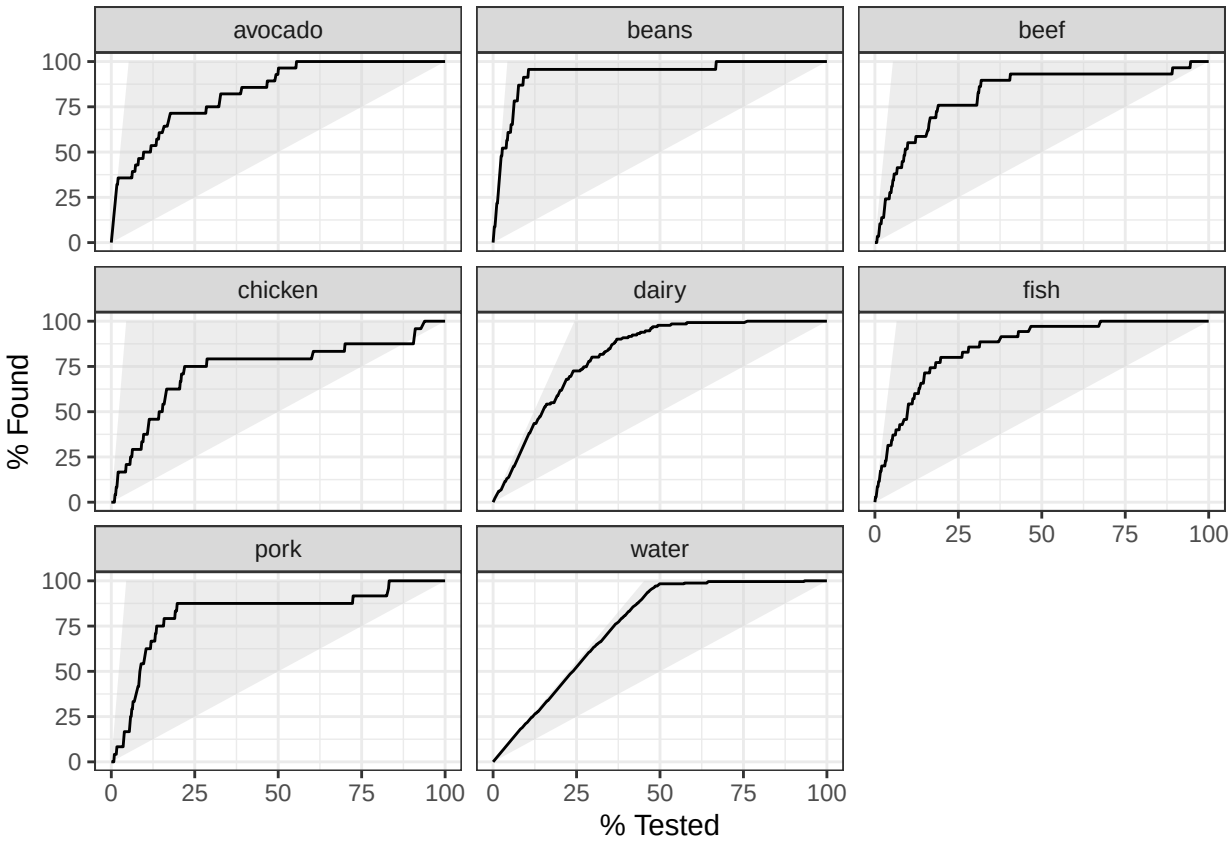

Figure 1: Gain curve based on the implemented naive Bayesian model

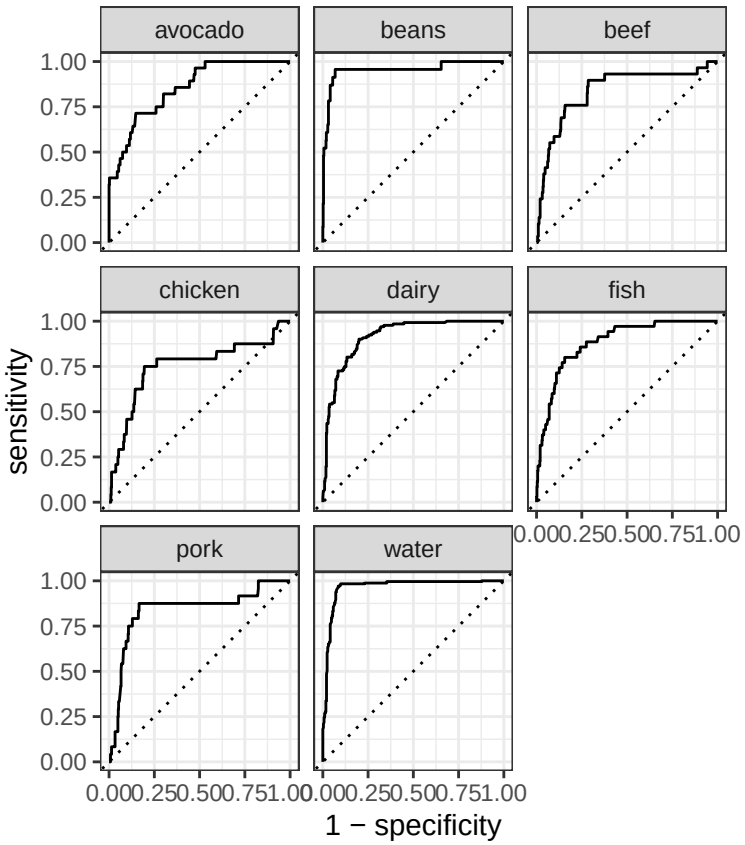

Figure 2: ROC curve based on the implemented naive Bayesian model

References

[1] Gelman A, Carlin JB, Stern HS, et al. *Bayesian data analysis*. Chapman; Hall/CRC, 1995.

[2] Chen H, Hu S, Hua R, et al. Improved naive bayes classification algorithm for traffic risk management. *Eurasip Journal on Advances in Signal Processing*; 2021. Epub ahead of print 2021. DOI: 10.1186/s13634-021-00742-6.

[3] Breiman L. Random forests. *Machine learning* 2001; 45: 5–32.

[4] Chipman HA, George EI, McCulloch RE. BART: Bayesian additive regression trees. *The Annals of Applied Statistics* 2010; 4: 266–298.

[5] Tanui CK, Karanth S, Njage PM, et al. Machine learning-based predictive modeling to identify genotypic traits associated with salmonella enterica disease endpoints in isolates from ground chicken. *LWT* 2022; 154: 112701.

[6] Filipello V, Mughini-Gras L, Gallina S, et al. Attribution of listeria monocytogenes human infections to food and animal sources in northern italy. *Food microbiology* 2020; 89: 103433.

[7] Kuhn M, Wickham H. *Tidymodels: A collection of packages for modeling and machine learning using tidyverse principles.*, https://www.tidymodels.org (2020).

[8] Kuhn M, Johnson K. *Applied predictive modeling*. 2013. Epub ahead of print 2013. DOI: 10.1007/978-1-4614-6849-3.

[9] Hvitfeldt E, Kuhn M. *Discrim: Model wrappers for discriminant analysis*, https://CRAN.R-project.org/package=discrim (2022).

[10] Wright MN, Ziegler A. ranger: A fast implementation of random forests for high dimensional data in C++ and R. *Journal of Statistical Software* 2017; 77: 1–17.

[11] Metz CE. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*; 8. Epub ahead of print 1978. DOI: 10.1016/S0001-2998(78)80014-2.

[12] Cohen J. A coefficient of agreement for nominal scales. Educational and psychological measurement. *Educational and Psychological Measurement*; 20.

[13] Mandrekar JN. Receiver operating characteristic curve in diagnostic test assessment. *Journal of Thoracic Oncology*; 5. Epub ahead of print 2010. DOI: 10.1097/JTO.0b013e3181ec173d.

[14] Engelmann B, Hayden E, Tasche D. Measuring the discriminative power of rating systems. *SSRN Electronic Journal*. Epub ahead of print 2021. DOI: 10.2139/ssrn.2793951.

[15] Kuhn M, Vaughan D. *Yardstick: Tidy characterizations of model performance*, https://CRAN.R-project.org/package=yardstick (2022).

[16] Hand DJ, Till RJ. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*; 45. Epub ahead of print 2001. DOI: 10.1023/A:1010920819831.

[17] Munck N, Njage PMK, Leekitcharoenphon P, et al. Application of whole-genome sequences and machine learning in source attribution of salmonella typhimurium. *Risk Analysis*; 40. Epub ahead of print 2020. DOI: 10.1111/risa.13510.

[18] R Core Team. *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, https://www.R-project.org/ (2022).