Mushroom classification

CUNY SPS DATA 621 Spring 2019

Group # 4

May 22, 2019

Contents

Abstract

Technology has changed almost every aspect of human life. Advancements in computing, farming, and statistical/machine learning have equipped us with the tools to immediately access information and accomplish tasks previously unthinkable. These advancements have also become a crutch making knowledge which a few decades prior difficult to master. Mushroom foraging is one of those tasks, for the past 30 years people have made use of a data set with detailing a mushrooms physical attributes and whether or not they are edible has been a popular data set to showcase machine learning techniques which trivialize the process of determining the toxicity of a mushroom for a computer, but are to abstract for an individual to internalize and use without the aid of technology. The goal of this paper is to use machine learning to create a model which can accurately identify edible/poisonous mushrooms that an individual can use to learn how to do it without the aid of technology.

Keywords: Classification, Boruta, Logistic regression, Recursive Feature Elimination method, C5.0

Introduction

Remaining healthy today is much more than annual visits to a primary care physician, reducing stressors, and 30-minute cardio workouts. Studies have shown that our diets are essential and assume the role of the majority in the 80-20 algorithm of health. The adage, "you are what you eat" has caused many to choose lifestyles such as pescatarian to vegan. As an alternative to meat, mushrooms are an acclaimed substitute boasting health promoting properties.

The gathering of mushrooms, known as "foraging" has become relevant of late. Based on Google Trends data from 2004 to the time of writing, one can see how interest has remained constant in certain regions of the country. On the west coast, states like Washington, Oregon, and California while the east coast has data in New York and Massachusetts.[1]

In regards to our project, the goal is to cultivate a model which is "actionable" for a mushroom hunter who is going out foraging. With toxic mushrooms being an outcome to avoid our team hypothesis will be if edible mushrooms can be classified by their features.

Literature review

The Mushroom Classification data set is popular being referenced in over 40 papers. Many of these papers focus on the use of advance clustering, machine learning, or feature selection techniques, such as Naive Bayes classification. These techniques are powerful that with little or no tuning they can achieve near perfect accuracy. One disadvantage of these techniques is that their interpretation can be quite abstract, a computer provided with data can efficiently classify new observations, but can be difficult for a person to internalize and apply themselves.

Our focus is on the papers which focus on feature selection. These techniques can identify the most important features for classifying the response variable as well as interactions between predictors. Among the techniques reviewed; Correlation-Based Filter approaches were computationally efficient and easy to interpret, but are more performant on numeric or ordered data rather than categorical. Other papers address this limitation by working iteratively and applying heuristic methods to improve decision selections. Alternatively probabilistic approaches can reduce reliance on heuristics with more mathematically focused process to find the optimal set of features, at a computational cost. We will make use of alternative feature selection methods not explicitly referenced seen in existing literature.

[1] Google Trends: searches for mushroom foraging in the United States of America

Methodology

The mushroom classification data set is clean and well structured that minimal data preparation is required before analysis can be done. The only explicit requirement is in how to handle the observations for `stalk_root` which are categorized as missing – these were handle through imputation. New features were created based on evaluating cross tabulations and visualizations of the predictors broken out by each response category and then three different techniques were employed to develop an accurate, actionable model. First, Boruta was used for feature selection and the selected features were used in a logistic regression, second was Recursive Feature Selection, and finally the C5.0 classification algorithm.

Experimentation

Data acquisition/preparation

The data come from a Field Guide to North American Mushrooms[2] and were prepared and shared by Jeff Schlimmer with the University of California Irvine Machine Learning Repository[3]. The data contain a total of 8214 observations and 22 attributes all of which are categorical. As provided the data requires little manipulation before analysis can be done.

Missing data

The one exception is the variable `stalk_root` which has a number of observations coded as *missing*. In total 2480 observations are coded as missing which amounts to $\approx 30.53\%$ of the data. These observations seemed to be missing at random and two mechanisms were considered to avoid dropping those observations. One was to treat *missing* as a distinct category and alternatively to re categorize `stalk_root` for those observation using multiple imputation. Neither is ideal, treating them as a distinct category has the potential to obfuscate combinations of attributes of a mushroom that could be used to determine their toxicity and multiple imputation techniques become less performant as the percent of missing observations exceeds 25% of the data. Since the ultimate objective is to create an actionable model that an individual can employ when foraging being able to identify interactions and combinations of attributes of mushrooms was a priority Multiple Imputation Chained Equations [MICE)[4] was employed to recategorize those observations.

Feature engineering

With all attributes in the data being categorical creating a full model results in a high number of predictors. Attempting to include interactions between predictors has the potential to lead to multicolinearity or saturation. To avoid this new features have been created which work to encompass relationships between multiple categories of predictors and response variable as well as interaction between predictors.

The approached used for feature engineering is largely heuristic, if cross-tabulation , *Table 1*, or visualization, *Figure 2*, showed a consistent relationship between the toxicity of a mushroom a feature was created. Similarly if there were no information value from a predictor, such as `veil_type` where all observations were categorized as *partial* it was removed.

[2] The Audubon Society Field Guide to North American Mushrooms (1981). G. H. Lincoff (Pres.), New York: Alfred A. Knopf ISBN: 9780394519920

[3] CUI ML Repository

[4] Multiple Imputation by Chained Equations: What is it and how does it work?

Modeling

The literature review illustrated that a wide variety of methods for modeling and feature selection can be successful in creating a model with near perfect accuracy. With our objective that any model created needs to be actionable so even a novice can identify whether a mushroom is edible or poisonous the decision for modeling was to employ a variety of modeling and feature selection techniques with the constraint that there can be no more than 4 attributes – including all categories within that attribute.

Baseline

Assuming all mushrooms to be poisonous nets a 47.88% accuracy, any model developed needs to surpass this accuracy.

Feature selection

Boruta & logit

Three different methods of feature selection were employed in order to develop models. The first will utilize Boruta[5] and manual forward selection. Boruta is a algorithm built on top of Random Forest for feature selection. The algorithm returns a variable importance metric (VIM) along with an estimation of whether a variable is statistically significant. A weakness of Boruta's algorithm is that the VIM can prioritize binary variables over multi-categorical variable, as such the predictors created based on heuristics have been excluded. *Figure 3* shows the results for Boruta on the original predictors. All of the original predictors were found to be significant with the exception of `veil_type` which as mentioned previously only had one categorization for all observations.

On its own Boruta only serves to assist with feature selection. For modeling features identified by Boruta are applied in a Logistic regression. Figure 3 shows that there is a drop in VIM after the 3 most important predictors. The model shows a marked improvement relative to the baseline with an accuracy of 99.79%.

Although only 4 attributes were utilized due to the high number of categories within those attributes it results in a model with 18 predictors, none of which are statistically significant. This suggests there is room for additional feature engineering. Details of the model can be seen in *Table 4*.

Recursive Feature Elimination

Recursive Feature Elimination (RFE) has two components. The first functions similar to Boruta in that it builds off of a Random Forest to establish VIM for each attribute. It then uses an an iterative processing running backward selection on subsets of the data. This combination makes this process more robust than Boruta when binary variables are present in addition to multicategory variables and reduces the chances of overfitting the training data. Given the model produced by Boruta had no statistically significant predictors and RFE is more robust when dealing with a binary variables dummy variables for each category have been included as predictors. A feature of RFE is the ability to constrain produce an optimal model for differing numbers of attributes, utilizing all attributes and dummy variables created for each category it is possible to achieve 100% accuracy, however 125 predictors is not realistic for an individual to internalize. For a more direct comparison an optimal model was created using 4 attributes which resulted in a similar although marginally less accurate, 99.4%, model. Additional details of the model can be seen in *Table 6*

C5.0

The preceding models have a couple of issues. First, due to the number of predictors neither is something that a novice forager is likely to be remember. Second, which is a much larger issue is that while both of

[5] Feature Selection with the Boruta Package

these model are capable of achieving a perfect accuracy on edible mushrooms, both fall short on poisonous mushrooms. While not all poisonous mushrooms are so toxic that they are deadly the risk associated with a false negative for a poisonous mushroom is much greater than for an edible mushroom. Both of these can be resolved with the C5.0 algorithm.

Despite being built on top of Random Forest neither Boruta or RFE are particularly capable at evaluating interactions between predictors. C5.0 is a decision tree regression algorithm which makes heavy use of sub-trees to evaluate splits and recursion over partitions in the data to evaluate the information gain from these splits. Additionally C5.0 is capable of using these sub-trees to create a set of rules to follow to classify the response variable from the predictors. Running C5.0 on all the predictors produces a model with 100% accuracy and only 6 rules. Additional information about the model can be found in *Table 7*

Evaluation & results

The table below compares the accuracy for all three models across the training and reserved evaluation data. While all three models achieve 100% accuracy on the positive (edible) mushrooms both the Boruta and RFE based models fail to accurately classify the negative (poisonous) mushrooms. While all three models are reasonably accurate given the risk associated with a false negative for a poisonous mushroom C5.0 is the clear winner. Additionally, C5.0 uses the fewest predictors to achieve its accuracy and provides clear rules a forager can follow to classify the mushrooms they encounter.

Model	Boruta Logit	Recursive Feature	C5.0 class
Predictors	18	18	6
Train_Accuracy	0.9978899	0.9940215	1.0000000
Train_Acc_pos.	1	1	1
Train_Acc_neg.	0.9959677	0.9886591	1.0000000
Eval_Accuracy	0.9983586	0.9942552	1.0000000
Eval_Accy_pos.	1	1	1
Eval_Acc_neg.	0.9967949	0.9888712	1.0000000

Rules

Below are a list of the rules provided by the C5.0 model. While the model explicitly defines 6 since rules 5 and 6 are dependent on evaluating 1 and 2 there a forager can safely evaluate toxicity of mushrooms with only 3 rules.

1. Does it smell or smells like something other than anise or almond? It is poisonous.

2. If it has green spores, it is poisonous
3. If no smell, does the surface below the ring is scaly and the color above the ring is brown? It is poisonous.
4. Is it in the leaves and have bruises? It is poisonous
5. Is it in leaves and adheres to rules 1 & 2? It is edible.
6. Does it have bruises adhere to rule 1 & 3? It is edible

Conclusion and future work

The C5.0 model provides the best results with the most actionable model. While this model is actionable it may not be providing a complete idea of what mushrooms are edible. The description of the data explains that any mushroom which was not explicitly classified as edible in the Audubon Society Field Guide were classified as poisonous. Future work could include rerunning the C5.0 classification on the original data from

the field guide treating the unclassified mushrooms as a third distinct category to see how this effects the rules provided.

Appendix

Supplemental materials

Table 1

Comparison of predictor category proportions by response classification

	poisonous (N=3916)	edible (N=4208)	Total (N=8124)	p value
bruises				< 0.001
bruises	624 (15.9%)	2752 (65.4%)	3376 (41.6%)	
no	3292 (84.1%)	1456 (34.6%)	4748 (58.4%)	
cap_color				< 0.001
brown	1020 (26.0%)	1264 (30.0%)	2284 (28.1%)	
buff	120 (3.1%)	48 (1.1%)	168 (2.1%)	
cinnamon	12 (0.3%)	32 (0.8%)	44 (0.5%)	
gray	808 (20.6%)	1032 (24.5%)	1840 (22.6%)	
green	0 (0.0%)	16 (0.4%)	16 (0.2%)	
pink	88 (2.2%)	56 (1.3%)	144 (1.8%)	
purple	0 (0.0%)	16 (0.4%)	16 (0.2%)	
red	876 (22.4%)	624 (14.8%)	1500 (18.5%)	
white	320 (8.2%)	720 (17.1%)	1040 (12.8%)	
yellow	672 (17.2%)	400 (9.5%)	1072 (13.2%)	
cap_shape				< 0.001
bell	48 (1.2%)	404 (9.6%)	452 (5.6%)	
conical	4 (0.1%)	0 (0.0%)	4 (0.0%)	
convex	1708 (43.6%)	1948 (46.3%)	3656 (45.0%)	
flat	1556 (39.7%)	1596 (37.9%)	3152 (38.8%)	
knobbed	600 (15.3%)	228 (5.4%)	828 (10.2%)	
sunken	0 (0.0%)	32 (0.8%)	32 (0.4%)	
cap_surface				< 0.001
fibrous	760 (19.4%)	1560 (37.1%)	2320 (28.6%)	
grooves	4 (0.1%)	0 (0.0%)	4 (0.0%)	
scaly	1740 (44.4%)	1504 (35.7%)	3244 (39.9%)	
smooth	1412 (36.1%)	1144 (27.2%)	2556 (31.5%)	
gill_attachment				< 0.001
attached	18 (0.5%)	192 (4.6%)	210 (2.6%)	
free	3898 (99.5%)	4016 (95.4%)	7914 (97.4%)	
gill_color				< 0.001
black	64 (1.6%)	344 (8.2%)	408 (5.0%)	
brown	112 (2.9%)	936 (22.2%)	1048 (12.9%)	
buff	1728 (44.1%)	0 (0.0%)	1728 (21.3%)	
chocolate	528 (13.5%)	204 (4.8%)	732 (9.0%)	
gray	504 (12.9%)	248 (5.9%)	752 (9.3%)	
green	24 (0.6%)	0 (0.0%)	24 (0.3%)	
orange	0 (0.0%)	64 (1.5%)	64 (0.8%)	
pink	640 (16.3%)	852 (20.2%)	1492 (18.4%)	
purple	48 (1.2%)	444 (10.6%)	492 (6.1%)	
red	0 (0.0%)	96 (2.3%)	96 (1.2%)	

	poisonous (N=3916)	edible (N=4208)	Total (N=8124)	p value
white	246 (6.3%)	956 (22.7%)	1202 (14.8%)	
yellow	22 (0.6%)	64 (1.5%)	86 (1.1%)	
gill_size				< 0.001
broad	1692 (43.2%)	3920 (93.2%)	5612 (69.1%)	
narrow	2224 (56.8%)	288 (6.8%)	2512 (30.9%)	
gill_spacing				< 0.001
close	3804 (97.1%)	3008 (71.5%)	6812 (83.9%)	
crowded	112 (2.9%)	1200 (28.5%)	1312 (16.1%)	
habitat				< 0.001
grasses	740 (18.9%)	1408 (33.5%)	2148 (26.4%)	
leaves	592 (15.1%)	240 (5.7%)	832 (10.2%)	
meadows	36 (0.9%)	256 (6.1%)	292 (3.6%)	
paths	1008 (25.7%)	136 (3.2%)	1144 (14.1%)	
urban	272 (6.9%)	96 (2.3%)	368 (4.5%)	
waste	0 (0.0%)	192 (4.6%)	192 (2.4%)	
woods	1268 (32.4%)	1880 (44.7%)	3148 (38.7%)	
odor				< 0.001
almond	0 (0.0%)	400 (9.5%)	400 (4.9%)	
anise	0 (0.0%)	400 (9.5%)	400 (4.9%)	
creosote	192 (4.9%)	0 (0.0%)	192 (2.4%)	
fishy	576 (14.7%)	0 (0.0%)	576 (7.1%)	
foul	2160 (55.2%)	0 (0.0%)	2160 (26.6%)	
musty	36 (0.9%)	0 (0.0%)	36 (0.4%)	
none	120 (3.1%)	3408 (81.0%)	3528 (43.4%)	
pungent	256 (6.5%)	0 (0.0%)	256 (3.2%)	
spicy	576 (14.7%)	0 (0.0%)	576 (7.1%)	
population				< 0.001
abundant	0 (0.0%)	384 (9.1%)	384 (4.7%)	
clustered	52 (1.3%)	288 (6.8%)	340 (4.2%)	
numerous	0 (0.0%)	400 (9.5%)	400 (4.9%)	
scattered	368 (9.4%)	880 (20.9%)	1248 (15.4%)	
several	2848 (72.7%)	1192 (28.3%)	4040 (49.7%)	
solitary	648 (16.5%)	1064 (25.3%)	1712 (21.1%)	
ring_number				< 0.001
none	36 (0.9%)	0 (0.0%)	36 (0.4%)	
one	3808 (97.2%)	3680 (87.5%)	7488 (92.2%)	
two	72 (1.8%)	528 (12.5%)	600 (7.4%)	
ring_type				< 0.001
evanescent	1768 (45.1%)	1008 (24.0%)	2776 (34.2%)	
flaring	0 (0.0%)	48 (1.1%)	48 (0.6%)	
large	1296 (33.1%)	0 (0.0%)	1296 (16.0%)	
none	36 (0.9%)	0 (0.0%)	36 (0.4%)	
pendant	816 (20.8%)	3152 (74.9%)	3968 (48.8%)	
spore_print_color				< 0.001
black	224 (5.7%)	1648 (39.2%)	1872 (23.0%)	
brown	224 (5.7%)	1744 (41.4%)	1968 (24.2%)	
buff	0 (0.0%)	48 (1.1%)	48 (0.6%)	
chocolate	1584 (40.4%)	48 (1.1%)	1632 (20.1%)	
green	72 (1.8%)	0 (0.0%)	72 (0.9%)	
orange	0 (0.0%)	48 (1.1%)	48 (0.6%)	
purple	0 (0.0%)	48 (1.1%)	48 (0.6%)	
white	1812 (46.3%)	576 (13.7%)	2388 (29.4%)	

	poisonous (N=3916)	edible (N=4208)	Total (N=8124)	p value
yellow	0 (0.0%)	48 (1.1%)	48 (0.6%)	
stalk_color_above_ring				< 0.001
brown	432 (11.0%)	16 (0.4%)	448 (5.5%)	
buff	432 (11.0%)	0 (0.0%)	432 (5.3%)	
cinnamon	36 (0.9%)	0 (0.0%)	36 (0.4%)	
gray	0 (0.0%)	576 (13.7%)	576 (7.1%)	
orange	0 (0.0%)	192 (4.6%)	192 (2.4%)	
pink	1296 (33.1%)	576 (13.7%)	1872 (23.0%)	
red	0 (0.0%)	96 (2.3%)	96 (1.2%)	
white	1712 (43.7%)	2752 (65.4%)	4464 (54.9%)	
yellow	8 (0.2%)	0 (0.0%)	8 (0.1%)	
stalk_color_below_ring				< 0.001
brown	448 (11.4%)	64 (1.5%)	512 (6.3%)	
buff	432 (11.0%)	0 (0.0%)	432 (5.3%)	
cinnamon	36 (0.9%)	0 (0.0%)	36 (0.4%)	
gray	0 (0.0%)	576 (13.7%)	576 (7.1%)	
orange	0 (0.0%)	192 (4.6%)	192 (2.4%)	
pink	1296 (33.1%)	576 (13.7%)	1872 (23.0%)	
red	0 (0.0%)	96 (2.3%)	96 (1.2%)	
white	1680 (42.9%)	2704 (64.3%)	4384 (54.0%)	
yellow	24 (0.6%)	0 (0.0%)	24 (0.3%)	
stalk_root				< 0.001
bulbous	2720 (69.5%)	2496 (59.3%)	5216 (64.2%)	
club	44 (1.1%)	512 (12.2%)	556 (6.8%)	
equal	256 (6.5%)	864 (20.5%)	1120 (13.8%)	
missing	0 (0.0%)	0 (0.0%)	0 (0.0%)	
rooted	896 (22.9%)	336 (8.0%)	1232 (15.2%)	
stalk_shape				< 0.001
enlarging	1900 (48.5%)	1616 (38.4%)	3516 (43.3%)	
tapering	2016 (51.5%)	2592 (61.6%)	4608 (56.7%)	
stalk_surface_above_ring				< 0.001
fibrous	144 (3.7%)	408 (9.7%)	552 (6.8%)	
scaly	8 (0.2%)	16 (0.4%)	24 (0.3%)	
silky	2228 (56.9%)	144 (3.4%)	2372 (29.2%)	
smooth	1536 (39.2%)	3640 (86.5%)	5176 (63.7%)	
stalk_surface_below_ring				< 0.001
fibrous	144 (3.7%)	456 (10.8%)	600 (7.4%)	
scaly	76 (1.9%)	208 (4.9%)	284 (3.5%)	
silky	2160 (55.2%)	144 (3.4%)	2304 (28.4%)	
smooth	1536 (39.2%)	3400 (80.8%)	4936 (60.8%)	
veil_color				< 0.001
brown	0 (0.0%)	96 (2.3%)	96 (1.2%)	
orange	0 (0.0%)	96 (2.3%)	96 (1.2%)	
white	3908 (99.8%)	4016 (95.4%)	7924 (97.5%)	
yellow	8 (0.2%)	0 (0.0%)	8 (0.1%)	
veil_type				1.000
partial	3916 (100.0%)	4208 (100.0%)	8124 (100.0%)	

Figure 2


poisonous
edible
bruises
bruises
no
cap_color
brown
buff
cinnamon
gray
green
pink
purple
red
white
yellow
cap_shape
bell
conical
convex
flat
knobbed
sunken
cap_surface
fibrous
grooves
scaly
smooth
gill_attachment
attached
free
gill_color
black
brown
buff
chocolate
gray
green
orange
pink
purple
red
white
yellow
gill_size
broad
narrow
gill_spacing
close
crowded
habitat
grasses
leaves
meadows
paths
urban
waste
woods
odor
almond
anise
creosote
fishy
foul
musty
none
pungent
spicy
population
abundant
clustered
numerous
scattered
several
solitary
ring_number
none
one
two
ring_type
evanescent
flaring
large
none
pendant
spore_print_color
black
brown
buff
chocolate
green
orange
purple
white
yellow
stalk_color_above_ring
brown
buff
cinnamon
gray
orange
pink
red
white
yellow
stalk_color_below_ring
brown
buff
cinnamon
gray
orange
pink
red
white
yellow
stalk_root
bulbous
club
equal
rooted
stalk_shape
enlarging
tapering
stalk_surface_above_ring
fibrous
scaly
silky
smooth
stalk_surface_below_ring
fibrous
scaly
silky
smooth
veil_color
brown
orange
white
yellow
veil_type
partial

Figure 3

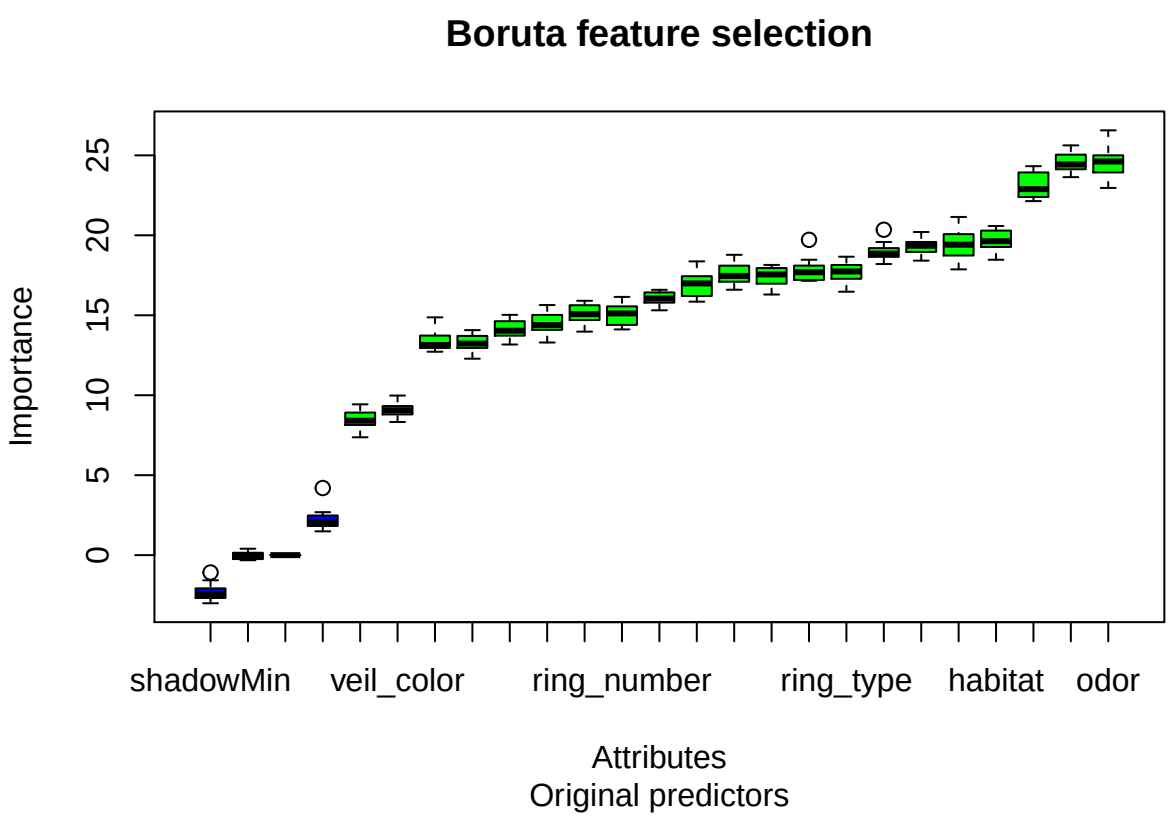

Table 4

Logistic Regression with top 3 predictors identified by Boruta

```
##
## Call:
## glm(formula = target ~ spore_print_color + odor + gill_size +
##     gill_spacing, family = binomial(link = "logit"), data = dfMDLTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6120   0.0000   0.0000   0.0000   0.7981
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 4.503e+01  2.181e+04   0.002    0.998
## spore_print_colorbrown      4.527e-01  1.119e+04   0.000    1.000
## spore_print_colorbuff      -3.971e+01  6.185e+04  -0.001    0.999
## spore_print_colorchocolate  7.880e-01  1.325e+04   0.000    1.000
## spore_print_colorgreen     -9.284e+01  4.989e+04  -0.002    0.999
## spore_print_colororange    -3.971e+01  6.575e+04  -0.001    1.000
## spore_print_colorpurple     3.466e+00  6.632e+04   0.000    1.000
## spore_print_colorwhite     -4.334e+01  8.707e+03  -0.005    0.996
## spore_print_coloryellow    -3.971e+01  5.936e+04  -0.001    0.999
## odoranise                   7.398e-02  2.593e+04   0.000    1.000
## odorcreosote               -4.907e+01  3.317e+04  -0.001    0.999
```

```
## odorfishy                   1.564e+01  2.835e+04   0.001    1.000
## odorfoul                   -7.182e+01  2.476e+04  -0.003    0.998
## odormusty                  -2.827e+01  6.943e+04   0.000    1.000
## odornone                    2.124e+01  2.067e+04   0.001    0.999
## odorpungent                -2.794e+01  3.283e+04  -0.001    0.999
## odorspicy                   1.564e+01  2.867e+04   0.001    1.000
## gill_sizenarrow            -4.390e+01  6.418e+03  -0.007    0.995
## gill_spacingcrowded         2.194e+01  4.598e+03   0.005    0.996
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 7873.640  on 5686  degrees of freedom
## Residual deviance:   51.564  on 5668  degrees of freedom
## AIC: 89.564
##
## Number of Fisher Scoring iterations: 25
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 2711    0
##          1   12 2964
##
##                Accuracy : 0.9979
##                  95% CI : (0.9963, 0.9989)
##     No Information Rate : 0.5212
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.9958
##  Mcnemar's Test P-Value : 0.001496
##
##             Sensitivity : 0.9956
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9960
##              Prevalence : 0.4788
##          Detection Rate : 0.4767
##    Detection Prevalence : 0.4767
##       Balanced Accuracy : 0.9978
##
##        'Positive' Class : 0
##
```

Figure 5

Table 6

RFE model details

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  poisonous edible
```

```
##   poisonous      2689      0
##   edible           34   2964
##
##                Accuracy : 0.994
##                  95% CI : (0.9917, 0.9959)
##     No Information Rate : 0.5212
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.988
##  Mcnemar's Test P-Value : 1.519e-08
##
##             Sensitivity : 0.9875
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 0.9887
##              Prevalence : 0.4788
##          Detection Rate : 0.4728
##    Detection Prevalence : 0.4728
##       Balanced Accuracy : 0.9938
##
##        'Positive' Class : poisonous
##
```

Table 7

C5.0 classification model details

```
##
## Call:
## C50:::C5.0.default(x = x, y = y, rules = TRUE, weights = wts)
##
##
## C5.0 [Release 2.07 GPL Edition]      Thu Jun  6 06:18:01 2019
## -------------------------------
##
## Class specified by attribute `outcome'
##
## Read 5687 cases (200 attributes) from undefined.data
##
## Rules:
##
## Rule 1: (2636, lift 2.1)
##  safeOdor.x <= 0
##  ->  class poisonous  [1.000]
##
## Rule 2: (53, lift 2.1)
##  spore_print_colorgreen > 0
##  ->  class poisonous  [0.982]
##
## Rule 3: (28, lift 2.0)
##  psnSCAR_SSBR.x > 0
##  ->  class poisonous  [0.967]
##
## Rule 4: (6, lift 1.8)
```

```
##  bruisesno <= 0
##  habitatleaves > 0
##  ->  class poisonous  [0.875]
##
## Rule 5: (2800, lift 1.9)
##  habitatleaves <= 0
##  spore_print_colorgreen <= 0
##  safeOdor.x > 0
##  psnSCAR_SSBR.x <= 0
##  ->  class edible  [1.000]
##
## Rule 6: (1015, lift 1.9)
##  bruisesno > 0
##  safeOdor.x > 0
##  psnSCAR_SSBR.x <= 0
##  ->  class edible  [0.999]
##
## Default class: edible
##
##
## Evaluation on training data (5687 cases):
##
##          Rules
##    ----------------
##      No      Errors
##
##       6    0( 0.0%)   <<
##
##
##     (a)   (b)    <-classified as
##    ----  ----
##    2723          (a): class poisonous
##          2964    (b): class edible
##
##
##  Attribute usage:
##
##   98.47% safeOdor.x
##   52.61% psnSCAR_SSBR.x
##   50.17% spore_print_colorgreen
##   49.34% habitatleaves
##   17.95% bruisesno
##
##
## Time: 0.2 secs
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  poisonous edible
##   poisonous      2723      0
##   edible            0   2964
##
##                Accuracy : 1
```

```
##                  95% CI : (0.9994, 1)
##     No Information Rate : 0.5212
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 1
##  Mcnemar's Test P-Value : NA
##
##             Sensitivity : 1.0000
##             Specificity : 1.0000
##          Pos Pred Value : 1.0000
##          Neg Pred Value : 1.0000
##              Prevalence : 0.4788
##          Detection Rate : 0.4788
##    Detection Prevalence : 0.4788
##       Balanced Accuracy : 1.0000
##
##        'Positive' Class : poisonous
##
```

R source code

See included Rmarkdown (rmd) document

Session info

	details
version	R version 3.6.0 (2019-04-26)
system	x86_64, linux-gnu
ui	X11
language	en_US
collate	en_US.UTF-8
tz	America/New_York
date	2019-06-06

package	*	version	date	source
abind		1.4-5	2016-07-21	CRAN (R 3.5.1)
arsenal	*	3.0.0	2019-03-25	CRAN (R 3.5.3)
assertthat		0.2.1	2019-03-21	CRAN (R 3.6.0)
backports		1.1.3	2018-12-14	cran ((**???**))
base	*	3.6.0	2019-05-13	local
Boruta	*	6.0.0	2018-07-17	CRAN (R 3.5.1)
broom		0.5.1	2018-12-05	cran ((**???**))
C50	*	0.1.2	2018-05-22	CRAN (R 3.6.0)
caret	*	6.0-80	2018-05-26	CRAN (R 3.5.1)
class		7.3-15	2019-01-01	CRAN (R 3.5.2)
codetools		0.2-16	2018-12-24	CRAN (R 3.5.2)
coin		1.2-2	2017-11-28	CRAN (R 3.5.2)
colorspace		1.4-1	2019-03-18	CRAN (R 3.6.0)
compiler		3.6.0	2019-05-13	local
crayon		1.3.4	2017-09-16	CRAN (R 3.6.0)

package	*	version	date	source
Cubist		0.2.2	2018-05-21	CRAN (R 3.6.0)
CVST		0.2-2	2018-05-26	CRAN (R 3.5.1)
data.table		1.11.4	2018-05-27	CRAN (R 3.5.1)
datasets	*	3.6.0	2019-05-13	local
ddalpha		1.3.4	2018-06-23	CRAN (R 3.5.1)
DEoptimR		1.0-8	2016-11-19	CRAN (R 3.5.1)
devtools		1.13.6	2018-06-27	CRAN (R 3.5.1)
digest		0.6.19	2019-05-20	CRAN (R 3.6.0)
dimRed		0.1.0	2017-05-04	CRAN (R 3.5.1)
doParallel	*	1.0.14	2018-09-24	CRAN (R 3.6.0)
dplyr	*	0.8.0.1	2019-02-15	cran ((**???**))
DRR		0.0.3	2018-01-06	CRAN (R 3.5.1)
e1071		1.7-0	2018-07-28	CRAN (R 3.5.1)
evaluate		0.14	2019-05-28	CRAN (R 3.6.0)
fastDummies	*	1.3.0	2019-04-22	CRAN (R 3.6.0)
foreach	*	1.4.4	2017-12-12	CRAN (R 3.5.1)
Formula		1.2-3	2018-05-03	CRAN (R 3.5.1)
generics		0.0.2	2018-11-29	cran ((**???**))
geometry		0.3-6	2015-09-09	CRAN (R 3.5.1)
ggfortify	*	0.4.5	2018-05-26	CRAN (R 3.5.1)
ggplot2	*	3.1.1	2019-04-07	CRAN (R 3.6.0)
glue		1.3.1	2019-03-12	CRAN (R 3.6.0)
gower		0.1.2	2017-02-23	CRAN (R 3.5.1)
graphics	*	3.6.0	2019-05-13	local
grDevices	*	3.6.0	2019-05-13	local
grid		3.6.0	2019-05-13	local
gridExtra		2.3	2017-09-09	CRAN (R 3.5.1)
gtable		0.3.0	2019-03-25	CRAN (R 3.6.0)
highr		0.8	2019-03-20	CRAN (R 3.5.3)
htmltools		0.3.6	2017-04-28	CRAN (R 3.5.0)
inum		1.0-0	2017-12-12	CRAN (R 3.5.1)
ipred		0.9-7	2018-08-14	CRAN (R 3.5.1)
iterators	*	1.0.10	2018-07-13	CRAN (R 3.5.1)
jomo		2.6-4	2018-08-30	CRAN (R 3.5.1)
kernlab		0.9-27	2018-08-10	CRAN (R 3.5.1)
knitr		1.23	2019-05-18	CRAN (R 3.6.0)
labeling		0.3	2014-08-23	CRAN (R 3.6.0)
lattice	*	0.20-38	2018-11-04	CRAN (R 3.5.1)
lava		1.6.3	2018-08-10	CRAN (R 3.5.1)
lazyeval		0.2.2	2019-03-15	CRAN (R 3.6.0)
libcoin		1.0-1	2017-12-13	CRAN (R 3.5.1)
lme4		1.1-17	2018-04-03	CRAN (R 3.5.1)
lubridate		1.7.4	2018-04-11	CRAN (R 3.5.1)
magic		1.5-9	2018-09-17	CRAN (R 3.5.1)
magrittr		1.5	2014-11-22	CRAN (R 3.6.0)
MASS		7.3-51.4	2019-04-26	CRAN (R 3.6.0)
Matrix		1.2-17	2019-03-22	CRAN (R 3.5.3)
memoise		1.1.0	2017-04-21	CRAN (R 3.4.1)
methods	*	3.6.0	2019-05-13	local
mice	*	3.3.0	2018-07-27	CRAN (R 3.5.1)
minqa		1.2.4	2014-10-09	CRAN (R 3.5.1)
mitml		0.3-6	2018-07-10	CRAN (R 3.5.1)

package	*	version	date	source
ModelMetrics		1.2.0	2018-08-10	CRAN (R 3.5.1)
modeltools		0.2-22	2018-07-16	CRAN (R 3.5.1)
multcomp		1.4-8	2017-11-08	CRAN (R 3.5.2)
munsell		0.5.0	2018-06-12	CRAN (R 3.6.0)
mvtnorm		1.0-8	2018-05-31	CRAN (R 3.5.1)
nlme		3.1-140	2019-05-12	CRAN (R 3.6.0)
nloptr		1.0.4	2017-08-22	CRAN (R 3.5.1)
nnet		7.3-12	2016-02-02	CRAN (R 3.5.0)
pan		1.6	2018-06-29	CRAN (R 3.5.1)
pander	*	0.6.2	2018-07-08	CRAN (R 3.5.1)
parallel	*	3.6.0	2019-05-13	local
partykit		1.2-2	2018-06-05	CRAN (R 3.5.1)
pillar		1.4.1	2019-05-28	CRAN (R 3.6.0)
pkgconfig		2.0.2	2018-08-16	cran ((**???**))
pls		2.7-0	2018-08-21	CRAN (R 3.5.1)
plyr		1.8.4	2016-06-08	CRAN (R 3.6.0)
pROC		1.13.0	2018-09-24	CRAN (R 3.5.2)
prodlim		2018.04.18	2018-04-18	CRAN (R 3.5.1)
purrr	*	0.3.0	2019-01-27	cran ((**???**))
R6		2.4.0	2019-02-14	CRAN (R 3.6.0)
randomForest	*	4.6-14	2018-03-25	CRAN (R 3.6.0)
ranger	*	0.10.1	2018-06-04	CRAN (R 3.5.1)
Rcpp		1.0.0	2018-11-07	cran ((**???**))
RcppRoll		0.3.0	2018-06-05	CRAN (R 3.5.1)
recipes		0.1.3	2018-06-16	CRAN (R 3.5.1)
reshape2		1.4.3	2017-12-11	CRAN (R 3.6.0)
rlang		0.3.4	2019-04-07	CRAN (R 3.6.0)
rmarkdown		1.11	2018-12-08	CRAN (R 3.5.2)
robustbase		0.93-2	2018-07-27	CRAN (R 3.5.1)
rpart	*	4.1-15	2019-04-12	CRAN (R 3.5.3)
sandwich		2.4-0	2017-07-26	CRAN (R 3.5.1)
scales		1.0.0	2018-08-09	CRAN (R 3.6.0)
sfsmisc		1.1-2	2018-03-05	CRAN (R 3.5.1)
splines		3.6.0	2019-05-13	local
stats	*	3.6.0	2019-05-13	local
stats4		3.6.0	2019-05-13	local
stringi		1.4.3	2019-03-12	CRAN (R 3.6.0)
stringr		1.4.0	2019-02-10	CRAN (R 3.6.0)
survival		2.44-1.1	2019-04-01	CRAN (R 3.5.3)
testthat		2.0.0	2017-12-13	CRAN (R 3.5.1)
TH.data		1.0-9	2018-07-10	CRAN (R 3.5.2)
tibble	*	2.0.1	2019-01-12	cran ((**???**))
tidyr	*	0.8.2	2018-10-28	cran ((**???**))
tidyselect		0.2.5	2018-10-11	cran ((**???**))
timeDate		3043.102	2018-02-21	CRAN (R 3.5.1)
tools		3.6.0	2019-05-13	local
utils	*	3.6.0	2019-05-13	local
withr		2.1.2	2018-03-15	CRAN (R 3.6.0)
xfun		0.3	2018-07-06	CRAN (R 3.5.1)
yaml		2.2.0	2018-07-25	CRAN (R 3.5.1)
zoo		1.8-3	2018-07-16	CRAN (R 3.5.1)

References

Allaire, JJ, Hadley Wickham, Kevin Ushey, and Gary Ritchie. 2017. *Rstudioapi: Safely Access the Rstudio Api*. https://CRAN.R-project.org/package=rstudioapi.

Analytics, Revolution, and Steve Weston. 2018. *Iterators: Provides Iterator Construct for R*. https://CRAN.R-project.org/package=iterators.

Ann, Chotirat, and Dimitrios Gunopulos. n.d. "Scaling up the Naive Bayesian Classifier: Using Decision Trees for Feature Selection." Computer Science Department University of California.

Auguie, Baptiste. 2017. *GridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Bache, Stefan Milton, and Hadley Wickham. 2014. *Magrittr: A Forward-Pipe Operator for R*. https://CRAN.R-project.org/package=magrittr.

Bates, Douglas, and Martin Maechler. 2019. *Matrix: Sparse and Dense Matrix Classes and Methods*. https://CRAN.R-project.org/package=Matrix.

Bates, Douglas, Martin Maechler, Ben Bolker, and Steven Walker. 2018. *Lme4: Linear Mixed-Effects Models Using 'Eigen' and S4*. https://CRAN.R-project.org/package=lme4.

Bates, Douglas, Katharine M. Mullen, John C. Nash, and Ravi Varadhan. 2014. *Minqa: Derivative-Free Optimization Algorithms by Quadratic Approximation*. https://CRAN.R-project.org/package=minqa.

Breiman, Leo, Adele Cutler, Andy Liaw, and Matthew Wiener. 2018. *RandomForest: Breiman and Cutler's Random Forests for Classification and Regression*. https://CRAN.R-project.org/package=randomForest.

Chang, Winston. 2019. *R6: Encapsulated Classes with Reference Semantics*. https://CRAN.R-project.org/package=R6.

Conceicao, Eduardo L. T. 2016. *DEoptimR: Differential Evolution Optimization in Pure R*. https://CRAN.R-project.org/package=DEoptimR.

Corporation, Microsoft, and Steve Weston. 2018. *DoParallel: Foreach Parallel Adaptor for the 'Parallel' Package*. https://CRAN.R-project.org/package=doParallel.

Csárdi, Gábor. 2017. *Crayon: Colored Terminal Output*. https://CRAN.R-project.org/package=crayon.

———. 2018. *Pkgconfig: Private Configuration for 'R' Packages*. https://CRAN.R-project.org/package=pkgconfig.

Daróczi, Gergely, and Roman Tsegelskyi. 2018. *Pander: An R 'Pandoc' Writer*. https://CRAN.R-project.org/package=pander.

Dowle, Matt, and Arun Srinivasan. 2018. *Data.table: Extension of 'Data.frame'*. https://CRAN.R-project.org/package=data.table.

Eddelbuettel, Dirk. 2018. *Digest: Create Compact Hash Digests of R Objects*. https://CRAN.R-project.org/package=digest.

Eddelbuettel, Dirk, Romain Francois, JJ Allaire, Kevin Ushey, Qiang Kou, Nathan Russell, Douglas Bates, and John Chambers. 2018. *Rcpp: Seamless R and C++ Integration*. https://CRAN.R-project.org/package=Rcpp.

Gagolewski, Marek, Bartek Tartanus, and other contributors; IBM, Unicode, Inc., other contributors; Unicode, and Inc. 2019. *Stringi: Character String Processing Facilities*. https://CRAN.R-project.org/package=stringi.

Genz, Alan, Frank Bretz, Tetsuhisa Miwa, Xuefei Mi, and Torsten Hothorn. 2018. *Mvtnorm: Multivariate Normal and T Distributions*. https://CRAN.R-project.org/package=mvtnorm.

Gerds, Thomas A. 2018. *Prodlim: Product-Limit Estimation for Censored Event History Analysis*. https://CRAN.R-project.org/package=prodlim.

Grund, Simon, Alexander Robitzsch, and Oliver Luedtke. 2018. *Mitml: Tools for Multiple Imputation in Multilevel Modeling*. https://CRAN.R-project.org/package=mitml.

Habel, Kai, Raoul Grasman, Robert B. Gramacy, Andreas Stahel, and David C. Sterratt. 2015. *Geometry: Mesh Generation and Surface Tesselation*. https://CRAN.R-project.org/package=geometry.

Hall, Mark A, and Mark A Hall. 1999. "Department of Computer Science Hamilton, Newzealand Correlation-Based Feature Selection for Machine Learning." Doctor of Philosophy at The University of Waikato.

Hall, Mark A, and Lloyd A Smith. 1999. "Feature Selection for Machine Learning: Comparing a Correlation-Based Filter Approach to the Wrapper." In *FLAIRS Conference*, 235.

Hankin, Robin K. S. 2018. *Magic: Create and Investigate Magic Squares*. https://CRAN.R-project.org/package=magic.

Heinzen, Ethan, Jason Sinnwell, Elizabeth Atkinson, Tina Gunderson, and Gregory Dougherty. 2019. *Arsenal: An Arsenal of 'R' Functions for Large-Scale Statistical Summaries*. https://CRAN.R-project.org/package=arsenal.

Henry, Lionel, and Hadley Wickham. 2018. *Tidyselect: Select from a Set of Strings*. https://CRAN.R-project.org/package=tidyselect.

———. 2019a. *Purrr: Functional Programming Tools*. https://CRAN.R-project.org/package=purrr.

———. 2019b. *Rlang: Functions for Base Types and Core R and 'Tidyverse' Features*. https://CRAN.R-project.org/package=rlang.

Hester, Jim. 2019. *Glue: Interpreted String Literals*. https://CRAN.R-project.org/package=glue.

Hester, Jim, Kirill Müller, Kevin Ushey, Hadley Wickham, and Winston Chang. 2018. *Withr: Run Code 'with' Temporarily Modified Global State*. https://CRAN.R-project.org/package=withr.

Holst, Klaus K. 2018. *Lava: Latent Variable Models*. https://CRAN.R-project.org/package=lava.

Horikoshi, Masaaki, and Yuan Tang. 2018. *Ggfortify: Data Visualization Tools for Statistical Analysis Results*. https://CRAN.R-project.org/package=ggfortify.

Hothorn, Torsten. 2017a. *Inum: Interval and Enum-Type Representation of Vectors*. https://CRAN.R-project.org/package=inum.

———. 2017b. *Libcoin: Linear Test Statistics for Permutation Inference*. https://CRAN.R-project.org/package=libcoin.

———. 2018. *TH.data: TH's Data Archive*. https://CRAN.R-project.org/package=TH.data.

Hothorn, Torsten, Frank Bretz, and Peter Westfall. 2017. *Multcomp: Simultaneous Inference in General Parametric Models*. https://CRAN.R-project.org/package=multcomp.

Hothorn, Torsten, Kurt Hornik, Mark A. van de Wiel, Henric Winell, and Achim Zeileis. 2017. *Coin: Conditional Inference Procedures in a Permutation Test Framework*. https://CRAN.R-project.org/package=coin.

Hothorn, Torsten, Friedrich Leisch, and Achim Zeileis. 2018. *Modeltools: Tools and Classes for Statistical Models*. https://CRAN.R-project.org/package=modeltools.

Hothorn, Torsten, and Achim Zeileis. 2018. *Partykit: A Toolkit for Recursive Partytioning*. https://CRAN.R-project.org/package=partykit.

Hunt, Tyler. 2018. *ModelMetrics: Rapid Calculation of Model Metrics*. https://CRAN.R-project.org/package=ModelMetrics.

Ihaka, Ross, Paul Murrell, Kurt Hornik, Jason C. Fisher, Reto Stauffer, Claus O. Wilke, Claire D. McWhite, and Achim Zeileis. 2019. *Colorspace: A Toolbox for Manipulating and Assessing Colors and Palettes*. https://CRAN.R-project.org/package=colorspace.

Jed Wing, Max Kuhn. Contributions from, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, et al. 2018. *Caret: Classification and Regression Training*. https://CRAN.R-project.org/package=caret.

Joseph L. Schafer, Original by. 2018. *Pan: Multiple Imputation for Multivariate Panel or Clustered Data*. https://CRAN.R-project.org/package=pan.

Kaplan, Jacob. 2019. *FastDummies: Fast Creation of Dummy (Binary) Columns and Rows from Categorical Variables*. https://CRAN.R-project.org/package=fastDummies.

Karatzoglou, Alexandros, Alex Smola, and Kurt Hornik. 2018. *Kernlab: Kernel-Based Machine Learning Lab*. https://CRAN.R-project.org/package=kernlab.

Kraemer, Guido. 2017. *DimRed: A Framework for Dimensionality Reduction*. https://CRAN.R-project.org/package=dimRed.

———. 2018. *DRR: Dimensionality Reduction via Regression*. https://CRAN.R-project.org/package=DRR.

Krueger, Tammo, and Mikio Braun. 2018. *CVST: Fast Cross-Validation via Sequential Testing*. https://CRAN.R-project.org/package=CVST.

Kuhn, Max, and Ross Quinlan. 2018a. *C50: C5.0 Decision Trees and Rule-Based Models*. https://CRAN.R-project.org/package=C50.

———. 2018b. *Cubist: Rule- and Instance-Based Regression Modeling*. https://CRAN.R-project.org/package=Cubist.

Kuhn, Max, and Hadley Wickham. 2018. *Recipes: Preprocessing Tools to Create Design Matrices*. https://CRAN.R-project.org/package=recipes.

Kuhn, Max, Hadley Wickham, and Davis Vaughan. 2018. *Generics: Common S3 Generics Not Provided by Base R Methods Related to Model Fitting*. https://CRAN.R-project.org/package=generics.

Kursa, Miron Bartosz, and Witold Remigiusz Rudnicki. 2018. *Boruta: Wrapper Algorithm for All Relevant Feature Selection*. https://CRAN.R-project.org/package=Boruta.

Lang, Michel, and R Core Team. 2018. *Backports: Reimplementations of Functions Introduced Since R-3.0.0*. https://CRAN.R-project.org/package=backports.

Liu, Huan, and Rudy Setiono. 1996. "A Probabilistic Approach to Feature Selection - a Filter Solution." In *ICML*, 319.

———. 1998. "Incremental Feature Selection." *Appl. Intell* 9: 217.

Maechler, Martin. 2018. *Sfsmisc: Utilities from 'Seminar Fuer Statistik' Eth Zurich*. https://CRAN.R-project.org/package=sfsmisc.

Mevik, Bjørn-Helge, Ron Wehrens, and Kristian Hovde Liland. 2018. *Pls: Partial Least Squares and Principal Component Regression*. https://CRAN.R-project.org/package=pls.

Meyer, David, Evgenia Dimitriadou, Kurt Hornik, Andreas Weingessel, and Friedrich Leisch. 2018. *E1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), Tu Wien*. https://CRAN.R-project.org/package=e1071.

Müller, Kirill, and Hadley Wickham. 2018. *Pillar: Coloured Formatting for Columns*. https://CRAN.R-project.org/package=pillar.

———. 2019. *Tibble: Simple Data Frames*. https://CRAN.R-project.org/package=tibble.

Peters, Andrea, and Torsten Hothorn. 2018. *Ipred: Improved Predictors*. https://CRAN.R-project.org/package=ipred.

Pinheiro, José, Douglas Bates, and R-core. 2019. *Nlme: Linear and Nonlinear Mixed Effects Models*. https://CRAN.R-project.org/package=nlme.

Plate, Tony, and Richard Heiberger. 2016. *Abind: Combine Multidimensional Arrays*. https://CRAN.R-project.org/package=abind.

Pokotylo, Oleksii, Pavlo Mozharovskyi, Rainer Dyckerhoff, and Stanislav Nagy. 2018. *Ddalpha: Depth-Based Classification and Calculation of Data Depth*. https://CRAN.R-project.org/package=ddalpha.

Quartagno, Matteo, and James Carpenter. 2018. *Jomo: Multilevel Joint Modelling Multiple Imputation*. https://CRAN.R-project.org/package=jomo.

R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Revolution Analytics, and Steve Weston. n.d. *Foreach: Provides Foreach Looping Construct for R*.

Ripley, Brian. 2016. *Nnet: Feed-Forward Neural Networks and Multinomial Log-Linear Models*. https://CRAN.R-project.org/package=nnet.

———. 2019a. *Class: Functions for Classification*. https://CRAN.R-project.org/package=class.

———. 2019b. *MASS: Support Functions and Datasets for Venables and Ripley's Mass*. https://CRAN.R-project.org/package=MASS.

Robin, Xavier, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. 2018. *PROC: Display and Analyze Roc Curves*. https://CRAN.R-project.org/package=pROC.

Robinson, David, and Alex Hayes. 2018. *Broom: Convert Statistical Analysis Objects into Tidy Tibbles*. https://CRAN.R-project.org/package=broom.

Sarkar, Deepayan. 2018. *Lattice: Trellis Graphics for R*. https://CRAN.R-project.org/package=lattice.

Spinu, Vitalie, Garrett Grolemund, and Hadley Wickham. 2018. *Lubridate: Make Dealing with Dates a Little Easier*. https://CRAN.R-project.org/package=lubridate.

Therneau, Terry, and Beth Atkinson. 2019. *Rpart: Recursive Partitioning and Regression Trees*. https://CRAN.R-project.org/package=rpart.

Therneau, Terry M. 2019. *Survival: Survival Analysis*. https://CRAN.R-project.org/package=survival.

Tierney, Luke. 2018. *Codetools: Code Analysis Tools for R*. https://CRAN.R-project.org/package=codetools.

Todorov, Valentin, Andreas Ruckstuhl, Matias Salibian-Barrera, Tobias Verbeke, Manuel Koller, and Martin Maechler. 2018. *Robustbase: Basic Robust Statistics*. https://CRAN.R-project.org/package=robustbase.

Ushey, Kevin. 2018. *RcppRoll: Efficient Rolling / Windowed Operations*. https://CRAN.R-project.org/package=RcppRoll.

van Buuren, Stef, and Karin Groothuis-Oudshoorn. 2018. *Mice: Multivariate Imputation by Chained Equations*. https://CRAN.R-project.org/package=mice.

van der Loo, Mark. 2017. *Gower: Gower's Distance*. https://CRAN.R-project.org/package=gower.

Wickham, Charlotte. 2018. *Munsell: Utilities for Using Munsell Colours*. https://CRAN.R-project.org/package=munsell.

Wickham, Hadley. 2016. *Plyr: Tools for Splitting, Applying and Combining Data*. https://CRAN.R-project.org/package=plyr.

———. 2017a. *Reshape2: Flexibly Reshape Data: A Reboot of the Reshape Package*. https://CRAN.R-project.org/package=reshape2.

———. 2017b. *Testthat: Unit Testing for R*. https://CRAN.R-project.org/package=testthat.

———. 2018. *Scales: Scale Functions for Visualization*. https://CRAN.R-project.org/package=scales.

———. 2019a. *Assertthat: Easy Pre and Post Assertions*. https://CRAN.R-project.org/package=assertthat.

———. 2019b. *Lazyeval: Lazy (Non-Standard) Evaluation*. https://CRAN.R-project.org/package=lazyeval.

———. 2019c. *Stringr: Simple, Consistent Wrappers for Common String Operations*. https://CRAN.R-project.org/package=stringr.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2019. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2019. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, and Lionel Henry. 2018. *Tidyr: Easily Tidy Data with 'Spread()' and 'Gather()' Functions*. https://CRAN.R-project.org/package=tidyr.

Wickham, Hadley, Jim Hester, and Winston Chang. 2018. *Devtools: Tools to Make Developing R Packages Easier*. https://CRAN.R-project.org/package=devtools.

Wickham, Hadley, Jim Hester, Kirill Müller, and Daniel Cook. 2017. *Memoise: Memoisation of Functions*. https://CRAN.R-project.org/package=memoise.

Wickham, Hadley, and Thomas Lin Pedersen. 2019. *Gtable: Arrange 'Grobs' in Tables*. https://CRAN.R-project.org/package=gtable.

Wright, Marvin N., Stefan Wager, and Philipp Probst. 2018. *Ranger: A Fast Implementation of Random Forests*. https://CRAN.R-project.org/package=ranger.

Wuertz, Diethelm, Tobias Setz, Yohan Chalabi, Martin Maechler, and Joe W. Byers. 2018. *TimeDate: Rmetrics - Chronological and Calendar Objects*. https://CRAN.R-project.org/package=timeDate.

Xie, Yihui. 2018. *Xfun: Miscellaneous Functions by 'Yihui Xie'*. https://CRAN.R-project.org/package=xfun.

———. 2019. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.

Ypma, Jelmer, with contributions by Hans W. Borchers, and Dirk Eddelbuettel. 2017. *Nloptr: R Interface to Nlopt*. https://CRAN.R-project.org/package=nloptr.

Zeileis, Achim, and Yves Croissant. 2018. *Formula: Extended Model Formulas*. https://CRAN.R-project.org/package=Formula.

Zeileis, Achim, Gabor Grothendieck, and Jeffrey A. Ryan. 2018. *Zoo: S3 Infrastructure for Regular and Irregular Time Series (Z's Ordered Observations)*. https://CRAN.R-project.org/package=zoo.

Zeileis, Achim, and Thomas Lumley. 2017. *Sandwich: Robust Covariance Matrix Estimators*. https://CRAN.R-project.org/package=sandwich.