Predicting Mushroom Edibility using Logistic Regression and Decision Tree

Sean Dixit
Department of Computer Science
Indiana University
Bloomington, US
sedixit@iu.edu

Luke Harris
Department of Computer Science
Indiana University
Bloomington, US
johaharr@iu.edu

Emma Wilson
Department of Computer Science
Indiana University
Bloomington, US
emrowils@iu.edu

Ethan Kawamara
Department of Computer Science
Indiana University
Bloomington, US
emugire@iu.edu

***Abstract*— Whether a mushroom is poisonous or not is associated with some traits of said mushroom more than others. Using logistic regression and decision trees, we can predict whether a mushroom is poisonous or not given its physical attributes with a 95%+ accuracy. Understanding the relationship between attributes of mushrooms and edibility of mushrooms can serve campers, explorers and any other individuals interacting with mushrooms to predict whether mushrooms they encounter can be safely eaten.**

Keywords—logistic, regression, decision, trees, edibility, Pearson, correlation, testing, training

I. Background

Various types of wild mushrooms grow in forests and grasslands, and it is common for the local population to consume them. It is estimated that fewer than 1% of the world's mushroom species are known to science [1], of which about 3% are known to be poisonous [2]. Given the sheer number of undiscovered mushrooms, it should come to no surprise that the common populace like campers and explorers, rather than scientists, tend to come across these mushrooms [2]. Throughout the US and other countries in the world, consumption of wild and potentially virulent mushrooms is common. A study published in the journal *Mycologia*, took the compiled U.S. mushroom exposures as reported by the National Poison Data System (NPDS) from 1999 to 2016 and found that 133,700 cases of mushroom exposure, mostly by ingestion, were reported. Approximately 704 of these exposures resulted in major harm [2]. As common an issue this is, very little work or investigation has been made to develop a means of helping the common population identify whether or not a mushroom is poisonous. Many plants and other flora that present a risk such as poison oak or poison ivy are exempt from this with clear rules like "leaves of three, let them be", however the information surrounding identification of edible mushrooms remains misleading at times with assumptions that are not consistent with the reality. If simple, clear, and consistent rules were available for identifying poisonous mushrooms, they could save hundreds from harm.

II. Dataset

The dataset for this project comes from the UC: Irvine machine learning repository, which is a collection of databases, domain theories, and data generators that are used by the machine learning community for empirical analysis of machine learning algorithms. The repository currently maintains up to 622 data sets available to the general public. This data set includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota family. The dataset contains 8,214 mushroom samples and each mushroom is identified as either edible or poisonous, which are the target attribute values. The dataset is fairly balanced, with 52% of the mushroom instances belonging to the edible class, and the rest belonging to the poisonous class. The data dictionary available on the UCI website describes the 22 categorical attributes of the mushroom samples, including characteristics such as cap shape, cap color, odor, gill size and color, stalk shape, and habitat. In the raw dataset, each attribute value is a letter that is a short form representation (done to compress the size of the dataset by the author). Our goal is to draw a prediction using regression models based on those attributes that are the most correlated with toxicity and edibility of the complete set of mushrooms.

III. Methods and measures used

This project primarily uses the R coding software for data pre-processing, ranking the features and creating the models. R's wide variety of functions and functionalities involving data cleaning, easy data manipulation, and creating a decision tree and logistic regression model made it the obvious choice. Additionally, we used the following R libraries: ggplot2 (for plotting), rpart (to create decision trees), dplyr (for dataframe manipulation functions like mutate), purr (specifically for the map function that applies a function to each element of a vector), caret (to create a confusion matrix from a decision tree), reshape2 (to display heatmaps), and ROCR (for plotting ROC curves). We first pre-processed the data, and then we used Pearson correlation to select the most relevant features (according to the correlation metric), and

then we trained, tested and compared our logistic regression and decision tree models.

IV. Results and Interpretations

A. Pre-processing Data

In order to feed the data into our machine learning models, we had to first fetch, label and clean the data. We used Excel to convert the raw .data file into .csv so we could read it using R. In R, we added column names to the table generated by reading the data and, to increase interpretability, replaced attribute value letters with the words that they are meant to represent. The dataset contained 2,480 instances that had missing attribute values, all of which were for stalk_root. We considered two options: ignore the instances with a missing attribute value, or ignore the stalk_root attribute. In ignoring the instances with missing values, our dataset became unbalanced, with 62% edible mushroom instances and 38% poisonous instances. On the other hand, in ignoring the stalk-root attribute, our dataset remained balanced, with 52% edible mushroom instances and 48% poisonous instances. So, we chose to remove the stalk_root attribute for analysis. Furthermore, the veil_type attribute had only one attribute value. We chose to ignore this attribute for our analysis since it will not make any difference in our model results.

B. Pearson Correlation for Feature Selection

In order to find the attributes that best allow for predicting edibility of mushrooms, we found the attributes that are most correlated, or associated, with the target attribute. In order to do so, we found the Pearson Correlation Coefficient between each attribute to get a correlation matrix. The coefficients between two variables can be between -1 and 1, and the further away they are from 0, the stronger the linear relationship between the two. Figure 1 shows the heatmap we get from the correlation matrix, with lighter blocks representing a high positive correlation (and darker blocks representing a high negative correlation).

Out of the first column of the correlation matrix where we see the coefficients between the target attribute and the other attributes, we created a bar plot that we then sorted to see which attributes (excluding the target attribute) are most correlated with the target attribute, as shown in Figure 1.

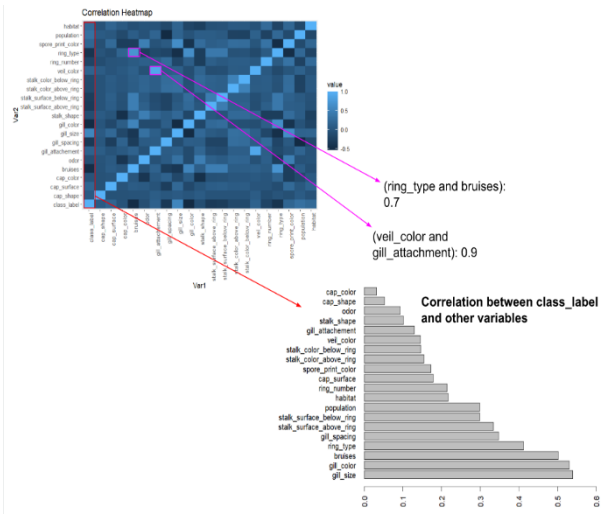

Fig. 1. Correlation Heatmap and barplot ranking each attribute based on it's association with the target attribute (class_label)

Additionally, to perform logistic regression, we also handled *multicollinear* attributes, that are, two or more attributes (excluding the target attribute) that are highly correlated with one another. We do so as multiple logistic regression assumes absence of multicollinearity, and the presence of multicollinear attributes can worsen the performance of our model. We saw that two pairs of attributes: ring_type and bruises, and veil_color and gill_attachment, had correlation coefficients of greater than 0.7. We removed the attribute less correlated with the target attribute in each pair, which turned out to be ring_type and gill_attachment. So we can make a fair comparison between our Decision Tree and logistic regression model, we decided to use the same attributes for both. Since the rpart function in R only used about 10 attributes when we ran it with all of the attributes, we limited the number of attributes to use in analysis to the top 10 features from the barplot (excluding ring_type and gill_attachment). As it turns out, gill_size is most correlated with the target variable, followed by gill_color, with gill_spacing coming in at the 4th spot. This tells us that the gills can tell us whether a mushroom is poisonous or not. Note that correlation doesn't quite quantify how well variables predict the target variable, but rather gives us an idea of how associated they are with the target variable. Regression would be more suitable to quantify exactly how well each variable predicts the target variable.

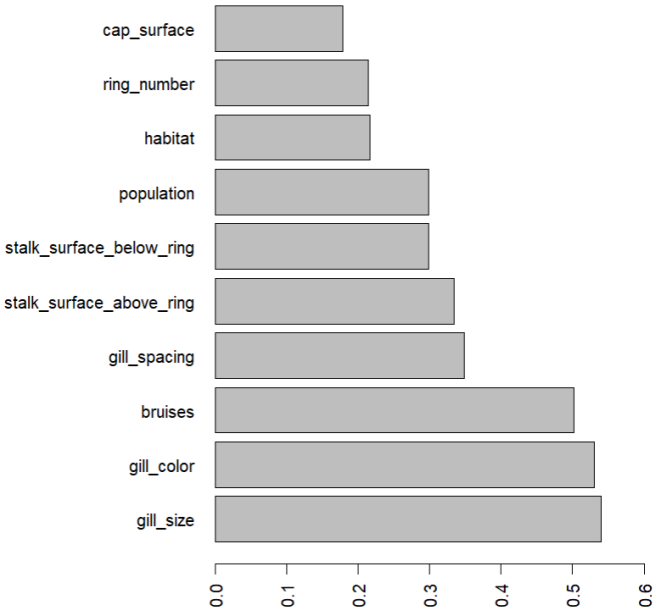

Fig. 2. Top ten attributes ranked by correlation with target attribute.

C. *Logistic Regression and Decision Tree*

Before creating our models, we partitioned the data using the holdout method by partitioning 60% of the data for training and the rest for testing, ensuring the split does not cause any unbalanced partition.

Since linear regression is suitable for when the predicted value is continuous, we instead used logistic regression, which is widely used for predicting categorical binary target attributes. We first mapped the categorical attribute values of the training and testing set to numerical values. We then used R's glm built-in function, which handled our categorical attributes that have more than two attribute values by creating dummy variables.

We then tested the logistic regression model in predicting whether mushrooms are poisonous or not. As Figure 3 shows, we get 66 Type I errors (False Positives), which represent the mushrooms predicted as edible that are actually poisonous: the most harmful predictions. The accuracy of the model comes out to be ~95%.

Predicted \ Actual	1	2
1	1638	66
2	88	1458

Fig. 3. Confusion matrix of the predictions of the target variable values of the testing set made by the logistic regression model and the actual target variable values of the testing set. The blue square outlines Type I errors. 1 represents "edible", while 2 represents "poisonous".

To create our decision tree, we used R's rpart library. We also used cross validation using the printcp function to find the best alpha to prune our tree, which happened to be zero as it gave us the least errors committed on the test partition (without an error rate of 0 on the train partition that would otherwise imply overfitting).

```
Root node error: 2369/4873 = 0.48615

n= 4873

           CP nsplit rel error    xerror       xstd
1  0.60574082      0 1.0000000 1.0000000 0.01472775
2  0.10299705      1 0.3942592 0.3942592 0.01159854
3  0.08020262      2 0.2912621 0.2912621 0.01027319
4  0.03503588      3 0.2110595 0.2110595 0.00894152
5  0.02912621      4 0.1760236 0.1903757 0.00853954
6  0.02026171      5 0.1468974 0.1468974 0.00758815
7  0.01730688      6 0.1266357 0.1283242 0.00712663
8  0.01477417      7 0.1093288 0.1148164 0.00676468
9  0.01308569      8 0.0945547 0.1051076 0.00648852
10 0.00633179      9 0.0814690 0.0814690 0.00574696
11 0.00548755     11 0.0688054 0.0688054 0.00529835
12 0.00450260     12 0.0633179 0.0650063 0.00515492
13 0.00401013     15 0.0498100 0.0590967 0.00492231
14 0.00379907     17 0.0417898 0.0481216 0.00445397
15 0.00295483     19 0.0341916 0.0350359 0.00381280
16 0.00232165     20 0.0312368 0.0308147 0.00357947
17 0.00211060     22 0.0265935 0.0223723 0.00305632
18 0.00147742     29 0.0084424 0.0126636 0.00230491
19 0.00126636     31 0.0054875 0.0075981 0.00178759
20 0.00098494     32 0.0042212 0.0067539 0.00168570
21 0.00000000     35 0.0012664 0.0012664 0.00073091
```

Fig. 4. Results of printcp. Red outline indicates the cp we chose for pruning.

On running our decision tree model to predict on the testing set, we got 3 Type I errors and 0 Type II errors as shown in Figure 5, along with an accuracy of 99.91%. However, we were uncomfortable with having any Type I errors in our predictions. So, we added a penalty matrix as an input to the rpart function that penalizes Type I errors 8 times more than Type II errors. Running the decision tree with this penalty matrix gave us 0 Type I errors and 9 Type II errors, thereby successfully minimizing the number of Type I errors. The accuracy of this model, however, is 99.72%; this is lower than the accuracy without the penalty matrix.

W/OUT penalty matrix

Predicted \ Actual	edible	poisonous
edible	1543	3
poisonous	0	1704

W/ penalty matrix

Predicted \ Actual	edible	poisonous
edible	1546	0
poisonous	9	1695

Fig. 5. Confusion matrices of the predictions of the target variable values of the testing set made by the decision tree models and the actual target variable values of the testing set. The blue squares outline Type I errors.

D. *Comparing the models*

Comparing the models, it is clear that the decision tree model outperforms the logistic regression model in predicting whether a mushroom is edible or poisonous, given the high accuracy and low number of Type I errors of the decision tree

model. The ROC curves also reinforce this conclusion, as the area under the curve of the decision tree ROC curve is higher than that of the logistic regression ROC curve, as shown in Figure 6.

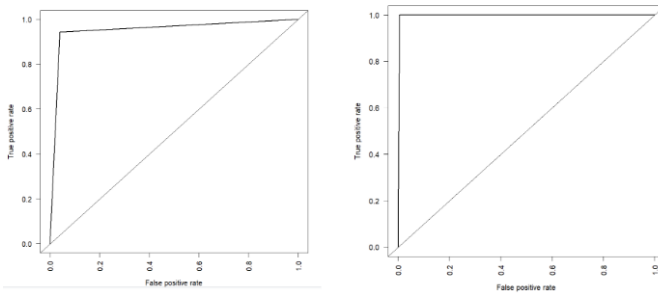

Fig. 6. On the left: ROC curve for the logistic regression model. On the right: ROC curve for the decision tree model with a penalty matrix.

When comparing the decision tree model with the penalty matrix and without the penalty matrix, however, it is not as simple as just comparing the accuracy of the models. Since Type II errors entail predicting edible mushrooms to be poisonous, and since the goal of our models are to ultimately prevent harm to someone depending on them, we don't mind extra Type II errors that decrease the accuracy in exchange for a lower chance of Type I errors, which is why we prefer the decision tree with the penalty matrix over the tree without the penalty matrix.

V. Challenges and key takeaways

Perhaps the biggest challenge we had was working with nominal categorical variables, as this made it difficult to interpret and display the results of logistic regression. This subsequently also made it difficult to find the best way to rank features based on how well each predicted our target variable (edibility).

Our comparison of the decision tree models serve as a great example of when a particular type of error is more important to minimize than the other. Since the purpose of our models was to aid campers and in general the common populace in determining whether they can consume a mushroom they found in the wild with the expectation of not being poisoned, we minimized Type I errors. To create effective models, one must take into account the application of the model and the purpose of it.

Contributions of the authors

The group members added the following contributions to the regression analysis of mushroom edibility. Sean cleaned the dataset, wrote the R code for logistic regression and decision tree, wrote some of the final report and formatted it and created some of the presentation slides. Emma found the dataset, established the objective of the research, performed background research, created the data visualizations in R for the slideshow presentation, wrote part of the presentation and wrote part of the final report.

References

[1] Rae Solomon, "With so many undiscovered mushrooms, citizen scientists find new species all the time," Sept. 2022. Accessed: Dec. 14, 2022. [Online]. Available: https://www.npr.org/2022/09/22/1124590354/with-so-many-undiscovered-mushrooms-citizen-scientists-find-new-species-all-the-.

[2] S. H. Eren, Y. Demirel, S. Ugurlu, I. Korkmaz, C. Aktas, F. M. K. Guven, "Mushroom poisoning: retrospective analysis of 294 cases," May 2010. Accessed: Dec 14, 2022. [Online]. doi: 10.1590/S1807-59322010000500006. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2882543/.