FINAL REPORT

**Abstract**

Many mushroom species make delicious snacks, while some can be deadly and indistinguishable to the untrained eye. Nearly 40 people die from mushroom ingestion per year in the United States. While identifying a given species can be difficult, many mushroom traits are easy to spot, and in this project, we built various models to predict edibility with these traits as predictors. We maximized various metrics on test data to reduce risk of harm and tuned the parameters of these models. Demonstrated by their performance on the test data, the predictive ability of the models discussed will be beneficial to our stakeholders.

**Explanation of the problem statement, stakeholders & data sources.**

Mushrooms have served as a food source for humans throughout time. However, these natural sources of energy have also led to the death of countless individuals as a few species contain toxic elements that can lead to death in a few instances. We decided to make a change and leverage current technologies and data science techniques to save countless lives.

The problem statement we seek to answer is: How can individuals predict whether mushrooms are poisonous or safe to eat to save their life? To tackle the prediction problem, we utilized various classification models. The ultimate objective is to create a model that accurately determines if a mushroom is poisonous or not, with an emphasis on minimizing results where poisonous mushrooms appear safe to eat (minimizing the false-negative rate).

Our dataset was from Kaggle (https://www.kaggle.com/datasets/uciml/mushroom-classification) and included classification variables solely. There were 23 variables that ranged from the physical characteristics of a mushroom to the habitat in which it could be found. It should be noted that a few variables would require scientific procedures and equipment to identify them correctly. For example, gill color can’t be determined by sight. Other variables may need individuals to touch them to understand the true property, putting them at risk if the mushroom is poisonous. There are very few properties that can clearly be identified strictly from vision, such as cap color and shape. The variable of interest was the class of the mushroom, which was either poisonous or edible.

We identified four stakeholders: naturalists, hikers, mycologists, and chefs. Naturalists try to educate those around them and will be able to help those seeking information on mushrooms. They will be able to enhance their own knowledge of mushrooms by using the model to test their own intuition. NAturalists can educate hikers that use wild mushrooms as a food source and chefs that try to experiment with fungi. Hikers may be wandering through the woods and want to avoid specific paths and not touch certain mushrooms as well. On the other hand, chefs can select safe mushrooms when trying to expand their menus. Lastly, mycologists will be able to isolate characteristics that make mushrooms poisonous to narrow their research approach. The model will save them time from experimenting with specific traits that have minimal effects on the poisonous attribute of mushrooms. Ultimately, the success of our model will save the lives of individuals around the world.

**Identify the metric(s) to be optimized for the model and explain why those metric(s) were chosen**

As previously mentioned, the negative consequences of consuming poisonous fungi range from mild illness to death. We chose the performance metrics to optimize, keeping in mind the severity of any false negative, with poison being considered positive. Recall includes the false negative rate, so we chose to optimize recall. By looking at recall, we could adjust the parameters and predictors of our models to minimize the false negatives on the training data. Simplicity is another, albeit subjective, metric we wanted to optimize. A complicated linear model or decision tree, utilizing a large number of predictors, would have little to no usability for the average hiker or forager. The ideal model for this problem is simple enough to quickly classify mushrooms in the field, while achieving a near 0% false negative rate. We found that each of these metrics could be satisfied individually, but the challenge was to find an intersection.

Our initial un-tuned decision tree, with a max depth of three, produced a false negative rate of 0.11 on the training data. This tree considered all possible predictors, and a FNR of 0.11 means this model categorized ~11% of poisonous mushrooms in the dataset as edible. Despite the convenience of such a simple decision tree, this model is nowhere near usable accuracy. After tuning the decision tree for max\_depth, max\_features, and max\_leaf\_nodes, and adjusting the cutoff to 0.4, we reached an optimized recall of 0.95. However, the accuracy simultaneously dipped to 0.59.

Understanding the limitations of any model simple enough to be used in the field, we optimized a single decision tree for recall, which could be used by hikers or foragers to determine potentially edible species. These mushrooms could be collected and identified later, but false negatives will be frequent.

The other models we created, while less practical for a hiker or forager, achieved accuracy >0.99 and recall of 1.0, as mentioned in the next section. Because of the trade-off between recall and simplicity/usability, we built multiple models, each tailored to multiple or individual stakeholders.

**Developing the model(s)**

In developing our classification models, the first step was to create a baseline model to allow comparison with the subsequent results of more complex, finely tuned models. In creating a baseline classification model, our first intuition was to create a simple logistic regression. In attempting to create a logistic regression, predicting class (0 or 1 – poisonous or non-poisonous), we used dummy variables of all predictors within the cleaned dataset (odor and veil\_type removed). Using sm.logit(), the standard statsmodels process for running a logistic regression, we encountered convergence errors due to the presence of a singular matrix. After further investigation, we concluded that this error was likely due to the presence of extremely collinear predictor variables. Despite the error, we still set out to create a logistic regression base model. Through exploring various routes of troubleshooting and debugging, we decided to utilize the sklearn library, referencing code from a previous participant of the Kaggle mushroom competition (see appendix for source). The logistic regression model formulated through sklearn had an accuracy >99% and a recall of 100% (see Table 1 for full results). Clearly, even a simple logistic regression base model could be very successful, and thus our model-building priority shifted.

| Statistic | Value (%) |
| --- | --- |
| Accuracy  FNR  Recall | 99.9  0.0  100.0 |

Table 1. Logistic Regression model statistics.

With the success of the logistic regression base model, our focus shifted towards recreating a similarly successful model using an ensemble method. Using an ensemble model containing Random Forest, Decision Tree, and XGBoost classifiers, we aimed to tune specifically to minimize the false negative rate (maximize recall). Each of these individual models used all the cleaned predictors available (odor and veil\_type excluded), although the tuning processes differed slightly. For the Random Forest classifier, we tuned for max\_features to optimize the tradeoff between the speed and complexity of the model, ensuring that sufficient predictors were considered to maximize performance. We also set n\_estimators to a large number – 500 – to ensure that the bagging process would consider a high number of trees, thus improving the performance of the model. In tuning the Decision Tree classifier, we focused on three parameters that all emphasized model interpretability: max\_leaf\_nodes, max\_features, and max\_depth. For the XGBoost classifier, we tuned the standard set of parameters that we had practiced during lecture: n\_estimators, max\_depth, learning\_rate, gamma, reg\_lambda, and scale\_pos\_weight. After tuning each of the three models individually, a stacking classifier was chosen to use the strength of each individual base estimator. The results of the stacking ensemble can be seen in Table 2.

| Statistic | Value (%) |
| --- | --- |
| Accuracy  FNR  Recall | 99.3  0.38  99.8 |

Table 2. Ensemble model statistics.

While the stacking ensemble model performs extremely well along our desired metrics (low FNR and high recall), it is nearly impossible to interpret. With the knowledge that we can produce a nearly “perfect” (yet more complex) model, we shifted our focus, striving for a balance of model performance and interpretability. Thus, we created the “Rule of Five” model.

After researching images of many different types of mushrooms in their natural habitats, our group extracted ***five*** *traits that are identifiable to the human eye within* ***five*** *seconds*, hence the “Rule of ***Five***” name. The selected traits were cap\_shape, cap\_surface, cap\_color, habitat, and stalk\_shape. We intentionally avoided any traits related to the gills and/or veils, as these physical characteristics are located underneath the mushroom cap, in most cases unable to be observed by the human eye within five seconds.

Once these traits were chosen, we set out to make a highly-interpretable model – one that could potentially be referenced by a stakeholder. We determined that a Decision Tree classifier was the most effective way to do this, as we hoped that the tree visualization would assist a potential stakeholder in discerning a poisonous from a non-poisonous mushroom. In tuning the “Rule of Five” decision tree, we selected the same parameters as the previous, more complex decision tree: max\_depth, max\_leaf\_nodes, and max\_features – all selected to ensure the model is interpretable and not overly-complex. Max\_depth, for example, was limited from range 1 to 4, as we did not want the tree to extend greater than 4 levels. After using a 5-fold cross validation search grid to tune the parameters, we obtained the decision tree depicted in Figure 1, with the model statistics in Table 3.

Figure 1: “Rule of Five” Decision Tree

| Statistic | Value (%) |
| --- | --- |
| Accuracy  FNR  Recall | 60.5  10.9  89.1 |

Table 3. “Rule of Five” Decision Tree model statistics.

As seen in Table 3, the FNR is quite high. With the use of this model, a poisonous mushroom will be predicted as non-poisonous approximately 3 times out of 10. When considering the severity of this mislabeling, it becomes clear that a FNR of 30% is extremely undesirable. To further optimize the model, decreasing FNR and increasing recall, we observed the precision recall tradeoff curve, as seen in Figure 2.

Figure 2. Precision-Recall Curve for “Rule of Five” model.

Keeping in mind that the previous FNR of ~30% was highly undesirable given the severity of the false negative scenario, we determined that accepting a lower overall model accuracy in the pursuit of an extremely low FNR was worthwhile. By adjusting the cutoff value from 0.5 to 0.4, we noticed a large improvement in the FNR, decreasing from 10.9% to 4.2%. The full model statistics can be seen in Table 4.

| Statistic | Value (%) |
| --- | --- |
| Accuracy  FNR  Recall | 60.5  10.9  89.1 |

Table 4. “Rule of Five” model statistics, cutoff of 0.2.

In sum, we sacrificed overall model accuracy and elite model FNR and recall for model interpretability. Upon reaching an interpretable and visually representable model, we further sacrificed model accuracy in return for low FNR (high recall). This model-building process was an exercise in qualitative reasoning, as interpretability and stakeholder usability were prioritized over the pure statistical success of the models we developed. For further detail regarding model development, refer to the Appendix.

**Conclusions**

As shown in our model development, all changes throughout the process were made with the stakeholders in mind. With our advanced model, it is possible to ensure with almost complete certainty that a mushroom is safe to eat if all data points are collected. Because of our optimized false negative rate this may mean edible mushrooms will be classified as poisonous, but this resignation is worthwhile to be confident that no poisonous mushrooms are deemed edible.

Acknowledging that it may not be practical for our stakeholders to gather all of the information that is necessary for our advanced model, our “Rule of Five” model aims to bridge the gap between accuracy and interpretability. From this model, our stakeholders can still be fairly confident that a mushroom is edible, while only needing to identify five of its basic characteristics. Further generalizing, a couple baseline “rules” can be seen from our model that stakeholders would do well to remember. First, to avoid non-grass habitats and flat capped mushrooms. Conversely, mushrooms in grass habitats with a tapering stalk shape are generally safe to eat.

With all of these options and suggestions available, our stakeholders should feel much more comfortable eating wild mushrooms. Depending on their situation, they may choose to spend more time gathering information for a more accurate result, or trust their first impressions based on our “Rule of Five”. Lastly, we recognize that our data does not contain every species of mushroom, so regardless of our model accuracy stakeholders should use caution and prioritize their safety first.