Paul Kelly - DATS 6103 - Data Mining - Final Project Report - 6/23/22

Proposal

For this project, I wanted to explore data related to natural phenomena. I've always been fascinated by plant life and fungi, and the dangers that exist to humans in that realm, so the question occurred to me - might I explore something to do with naturally occurring toxins and how to predict whether something in nature might be toxic based on its observable features. With that framing in mind, I began to look for publicly available restriction-free data and located the Mushroom Classification data set, upon which this project is based (https://www.kaggle.com/datasets/uciml/mushroom-classification?select=mushrooms.csv). The data set contains 23 variables and 8,124 observations; not large by any means but certainly large enough for predictive modeling in a classroom setting. I decided to use a Python 3 Anacondas environment and code within PyCharm, as required by this class; major libraries utilized would be pandas, sklearn, matplotlib, and seaborn; algorithms explored would be Naïve Bayes, Decision Tree, and Random Forest; and performance would be measured via accuracy scoring (higher is better).

Introduction and Overview

This project aims to answer the question "what features indicate whether a mushroom is poisonous or edible" by applying Naïve Bayes, Decision Tree, and Random Forest analyses on the Mushroom Classification data set. Once data was selected, the following steps were completed: data load-in, exploratory data analysis, data cleaning, data preprocessing, test-train split, model training, and finally model performance evaluation. This report will discuss each step of this classification problem in detail.

Description of the Data Set

The Mushroom Classification data set (henceforth referred to as "mushroom.csv") was located on Kaggle, but was created by and originates from the UC Irvine Machine Learning repository. It "includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981). It contains 23 variables and 8,124 observations.

Its variables are: class, cap shape, cap surface, cap color, bruises, odor, gill attachment, gill spacing, gill size, gill color, stalk shape, stalk root, stalk surface above ring, stalk surface below ring, stalk color above ring, stalk color below ring, veil type, veil color, ring number, ring type, spore print color, population, and habitat. Each value in the data set contains a single character that stands in for a more detailed description (for example, in the "class" variable, "e" represents "edible" and "p" represent poisonous), except for "ring number," which contains a numeric value of either 0, 1, or 2.

Exploratory Data Analysis

Although I would not decide to use "class" as my target variable until later in the project, it is important to note that before cleaning and preprocessing, "class" was an almost perfectly even split between "e" and "p" (around 4,000 values apiece), meaning that the data set was well balanced (see Figure 1).

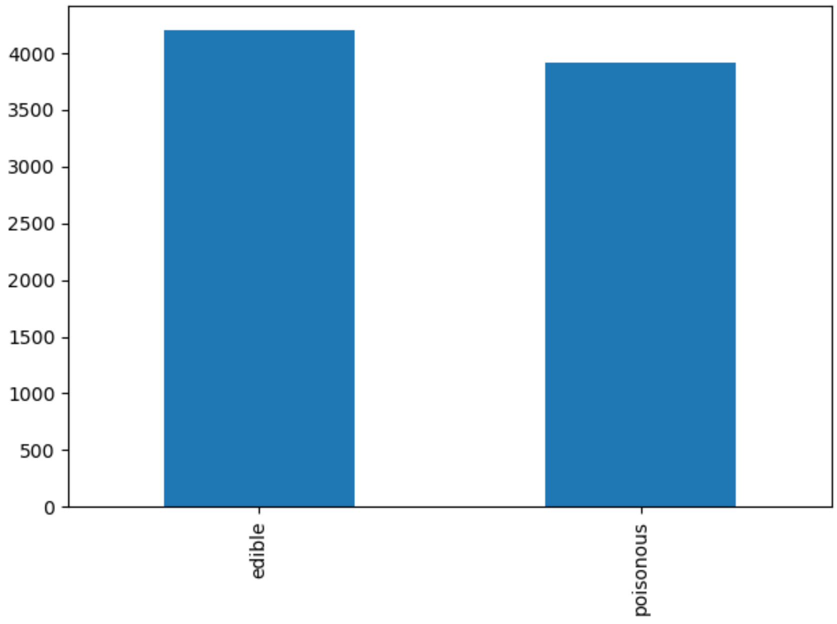

Figure 1, bar chart showing edible vs. poisonous

With this in mind, I then visualized multiple variables at once in order to explore potential relationships between them and differences between classes. When examining class against population (the patterns in which mushrooms grow), it becomes evident that edibles grow in numerous and abundant patterns, while poisonous do not (see Figure 2).

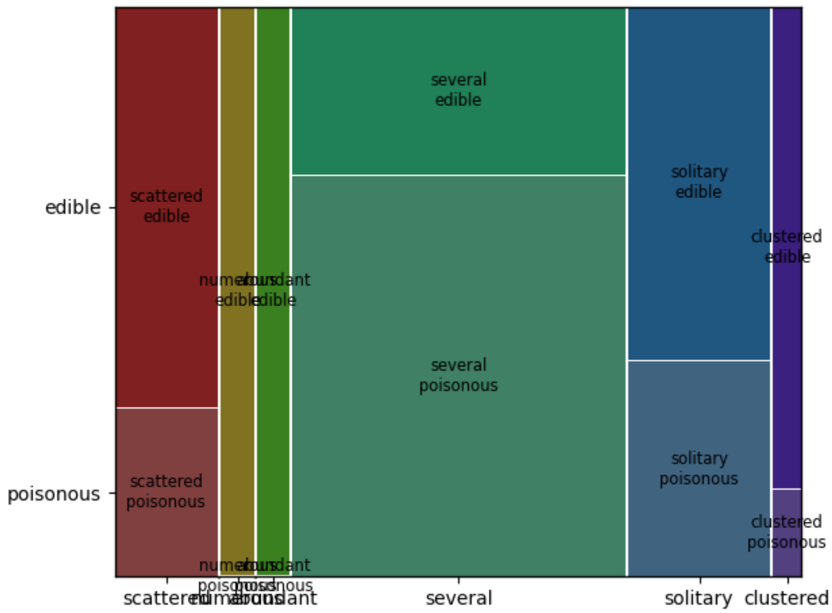

Figure 2, mosaic chart showing class vs. population

When examining class against habitat (where a certain variety of mushroom tends to grow), we see it more likely to find poisonous varieties on paths than edibles, and less likely to find poisonous varieties in wooded areas versus edible varieties. Apparently, poisonous mushrooms also never grow in “waste” (see Figure 3).

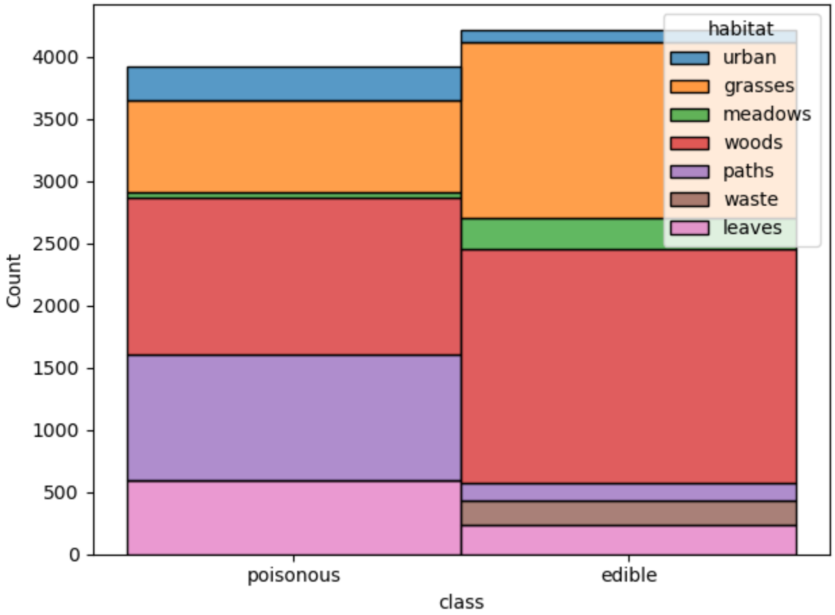

Figure 3, catplot showing class vs. habitat

When examining class against odor, we see that edibles are much more likely to be odorless or smell “pleasant,” like almond or anise, while poisonous are likely to be foul or fishy (see Figure 4).

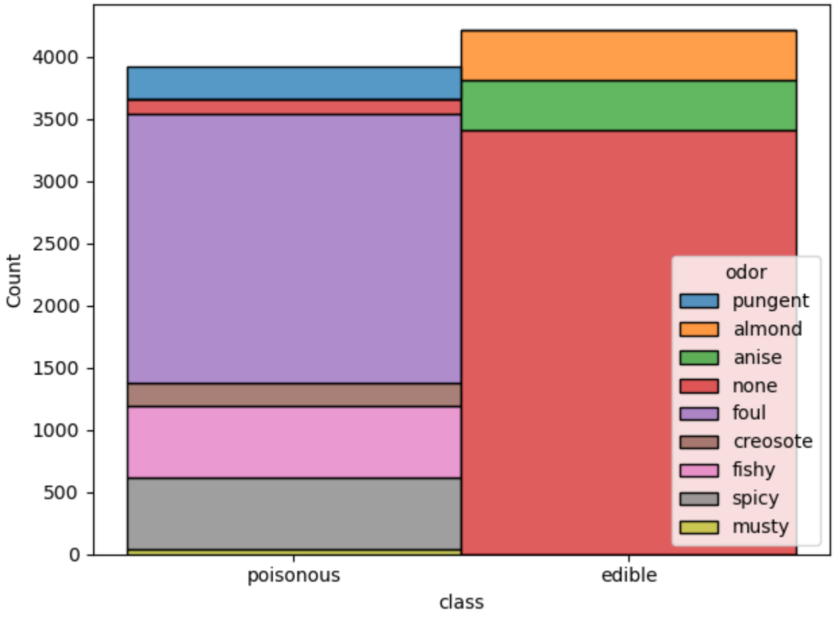

Figure 4, catplot showing class vs. odor

When examining class against gill size, we see that edibles edible mushrooms more often have broad gills, whereas poisonous ones are more often narrow (see Figure 5).

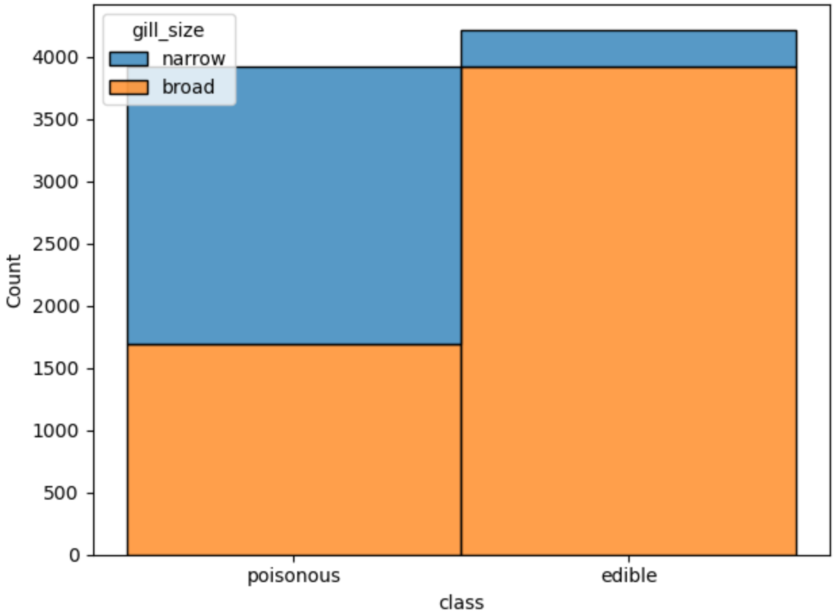

Figure 5, catplot showing class vs. gill size

When examining stalk surface below ring against class, we see that edibles are likely to have smooth surfaces, while poisonous mushrooms are likely to be silky (see Figure 6).

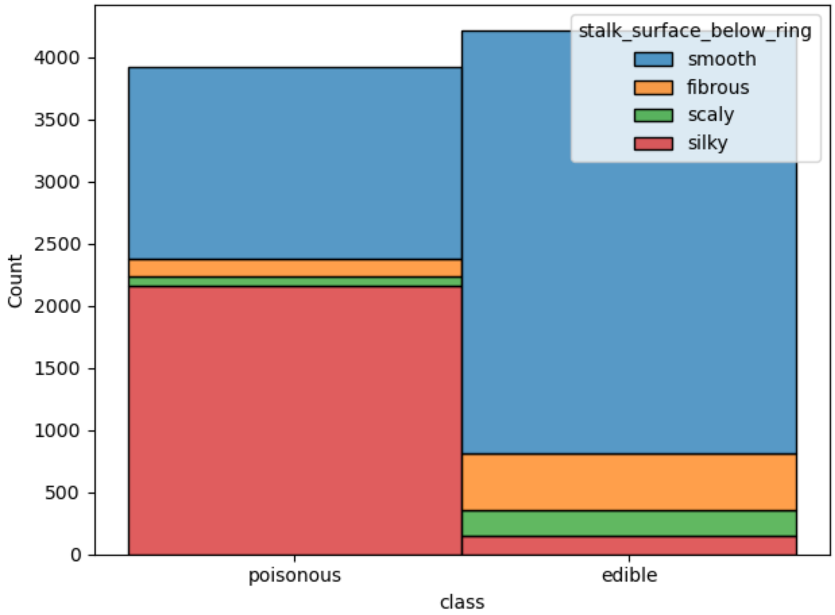

Figure 6, catplot showing class vs. stalk surface below ring

When examining spore print color against class, learn that if a mushroom has a chocolate colored spore print, it’s likely poisonous; if its spore print is black, it’s likely to be edible (see Figure 7).

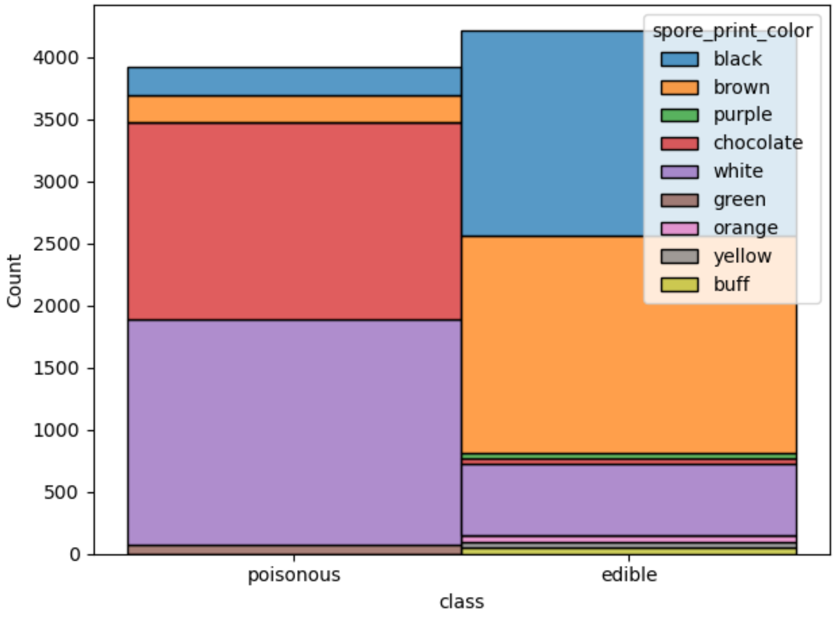

Figure 7, catplot showing class vs. spore print color

Data Cleaning and Preprocessing

Certain problems with the data became evident during EDA. Firstly, although most of the variable names utilized underscores to separate words, some used hyphens, which can complicate manipulating the data in Python whenever a variable with hyphens needs to be referred to outside of inverted commas. To that end, code was re-read in without headers, and new headers free of this irregularity were assigned.

Next, I checked each variable for unique values and determined that several of them included question marks (or '?') in place of unknown values. I replaced those values with 'NaN' and then dropped all NAs from the data set, reducing observations from 8124 to 5644, leaving around 69% of the data intact.

As previously mentioned, each variable save one is composed of a single character representing fuller descriptive information. The Python environment interpreted those values as strings, which would be problematic for modeling later. All datatypes were therefore changed to category, resulting in an almost entirely categorical, nominal data set. Then, variable values were changed from single letter representations to full words to ensure that plots and graphs would be more easily interpretable (again, an example here is "e" becoming "edible" and "p" becoming "poisonous"). This is more memory intensive to be sure, but is also, in my opinion, worth it to create more engaging visualizations. Finally, to ensure the modeling algorithms would accept the data, all categorical values were transformed into numeric values. In retrospect, this was perhaps not the best course of action considering the existence of LabelEncoder, but at the time I was under the impression that label encoding was only to be applied to our target variable, which was done.

Algorithms Used

The algorithms I selected for this project are NaÏve Bayes, Decision Tree, and Random Forest. NaÏve Bayes is a simple classification technique that relies on conditional probability and predicts the most probable class given a set of inputs. Decision Trees are a hierarchical technique in which a series of decisions are made based on some kind of metric. Random Forest is an ensemble method, and involves bagging and boosting of multiple classification trees for more robust results. Key to the use of all three in this study, however, is the ability of

Naïve Bayes, Decision Tree, and Random Forest to handle categorical data, of which our data set is comprised entirely. I measured relative success of these models via their accuracy scores and confusion matrices produced.

Experimental Setup

Once data was cleaned and preprocessed, I separated target variable "class" from the rest of the data. The target variable was then run through LabelEncoder, recoded the remaining variables as numeric, then split the data into training and testing groups with a test size of 30% and training size of 70%. In each case, an algorithm was called and run (either GaussianNB, DecisionTreeClassifier (both gini and entropy), and RandomForestClassifier) using X_train and Y_train, and predictions generated by running that model on X_test. Accuracy scores and confusion matrices were output once models were complete to determine model quality.

Results

NaÏve Bayes completed with 75.9% accuracy and 95.9 ROC_AUC scores. The confusion matrix identifies correctly 1034 poisonous (true positives) and 253 edible (true negatives) observations. It incorrectly categorizes 13 poisonous mushrooms as edible (false positives) and 394 edible mushrooms as poisonous (false negatives) (see Figure 8).

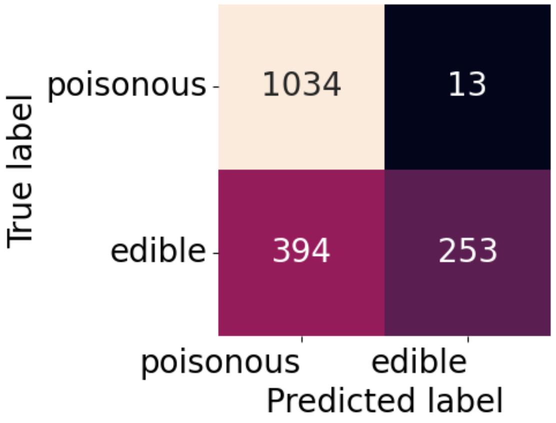

Figure 8, Naïve Bayes confusion matrix

Decision Tree completed with 97.5% (gini) and 97.3% (entropy) accuracy scores; the gini confusion matrix identifies correctly 1040 poisonous (true positives) and 613 edible (true negatives) observations. It incorrectly categorizes 7 poisonous mushrooms as edible (false positives) and 24 edible mushrooms as poisonous (false negatives) (see Figure 9).

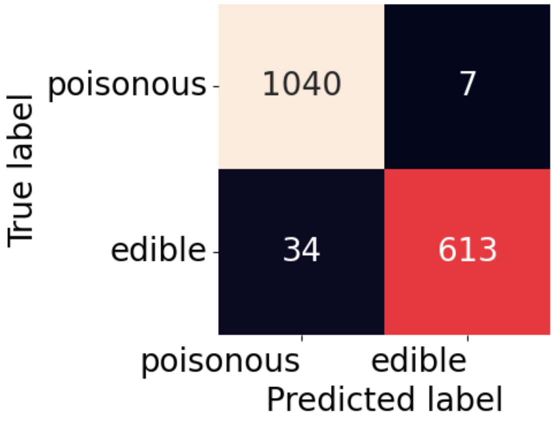

Figure 9, DT Gini confusion matrix

The entropy confusion matrix identifies correctly 1002 poisonous (true positives) and 647 edible (true negatives) observations. It incorrectly categorizes 45 poisonous mushrooms as edible (false positives) and 0 edible mushrooms as poisonous (false negatives) (see Figure 10).

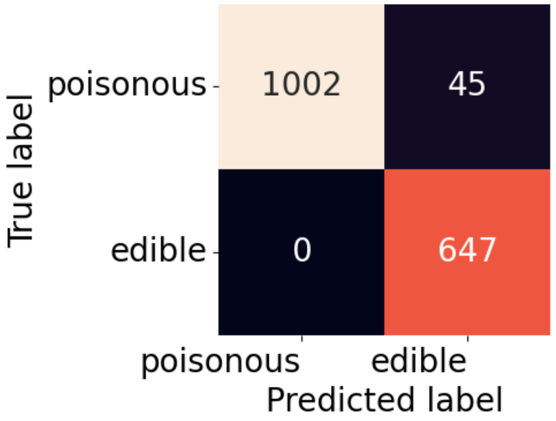

Figure 10, DT Entropy confusion matrix

Random Forest completed with 100% accuracy and 100% ROC_AUC scores. Its confusion matrix identifies correctly 1002 poisonous (true positives) and 647 edible (true negatives) observations. It incorrectly categorizes 0 poisonous mushrooms as edible (false positives) and 0 edible mushrooms as poisonous (false negatives) (see Figure 11).

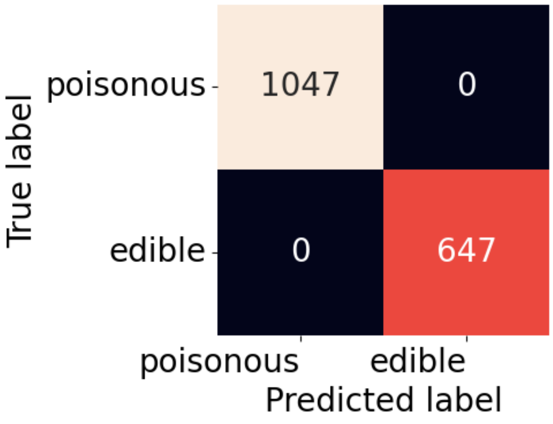

Figure 11, Random Forest confusion matrix

Finally, according to Random Forest feature importance ranking, most important features in predicting mushroom class are odor, spore print color, stalk shape, stalk surface below ring, and gill size (see Figure 12).

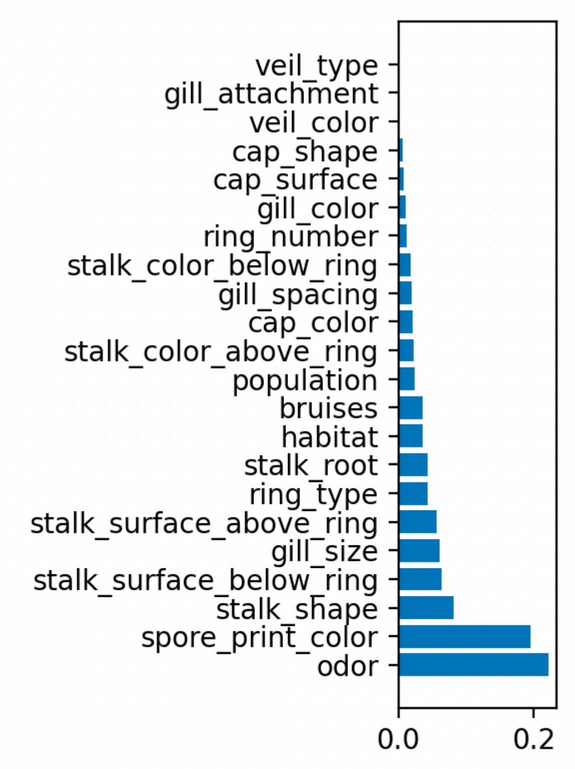

Figure 12, Random Forest importance of features

Summary and Conclusions

In summary, of the three unboosted models, Decision Tree (Gini) performed the best at 97.5% accuracy. By using boosting methods, however, Random Forest achieves 100% accuracy, and seems to be the best performing model. Additionally, Random Forest provides insights into importance of features for further modeling consideration. As previously mentioned, the highlighted features are odor, spore print color, stalk shape, stalk surface below ring, and gill size, which not only recalls general impressions from the Exploratory Data Analysis, but answers our initial research question. That said, 100% accuracy may be too good to be true, and potentially indicates overfitting of the model or another unseen bug in project code.

Lessons learned include python coding techniques for re-parameterizing categorical variables; implementing an interactive GUI for demonstrating and adjusting models in real time; and to never eat wild fungi without first determining whether its odor is pleasant or foul. Ideas for future improvement may include utilizing gridsearchcv to select only the best features for testing in additional models. It would also be extremely interesting to incorporate images of each mushroom type into further studies.

Works Cited

EDA, cleaning/preprocessing, and modeling code:

https://github.com/amir-jafari/Data-Mining

GUI code:

https://github.com/amir-jafari/Data-Mining/blob/master/Demo/PyQt5/Demo/Main.py

Mosaic chart:

https://stackoverflow.com/questions/31029560/plotting-categorical-data-with-pandas-and-matplotlib

Catplots:

https://datascience.stackexchange.com/questions/89692/plot-two-categorical-variables