STAT6309 Final Project: Predicting the Edibility of Mushrooms

Delaney Helgeson

Introduction

One of the most important skills mushroom foraging enthusiasts need to acquire is the ability to distinguish between poisonous and edible mushrooms. This data exploration details the process of building a Penalized Logistic Regression model in order to predict the probability that a mushroom is poisonous. Clustering methods will also be explored. The data set is the Secondary Mushroom Dataset from the UCI machine learning repository[1]. The data set contains 61,069 observations and 21 attributes. There are 17 categorical attributes and 3 quantitative attributes.

Mushroom attributes are primarily grouped into four categories: the cap, gills, stem, and veil. The cap refers to the round or flat head-like structure at the top of the fungus; the gills refer to the spore producing surface on the underside of the cap; the stem refers to the structure that connects the mushroom to the ground; lastly, the veil refers to the thin membrane that connects the cap to the stem in order to cover the gills of an immature mushroom. For each of these four categories, there are variables related to the size, color, and shape. Note that not all mushrooms contain every structure. Other attributes include whether or not the mushroom bleeds or bruises, meaning whether the mushroom changes color or oozes fluid when cut open; the ring of the mushroom, meaning whether the mushroom produces 'fairy rings' (an offshoot structure that surrounds the parent mushroom); spore-print-color, meaning the color of the spores of a mushroom; habitat, meaning where the mushroom grows; and season, the season in which the mushroom can grow successfully.

The levels of each categorical variable are coded as letters that represent abbreviations for naturally observed characteristics. For example, the cap shape of a mushroom may be bell-shaped, conical, convex, flat, sunken, spherical, or other. These characteristics are coded in the dataset as 'b', 'c', 'x', 'f', 's', 'p', or 'o', respectively. See the data dictionary for a more detailed description of the data.

Preprocessing

In order to prepare the data for model building, it must be pre-processed. There were nine variables which contained missing values, including variables describing: cap surface, gill attachment, gill

[1] https://archive.ics.uci.edu/ml/datasets/Secondary+Mushroom+Dataset

spacing, stem root, stem surface, veil type, veil color, ring type, and spore print color. Variables which contained more than 85% of missing values were dropped from the dataset entirely; these variables were: veil type, veil color, and spore print color. For the remaining features, missing values were imputed using nearest neighbor imputation.

Nearest neighbor imputation was accomplished by first encoding all of the categorical levels numerically. Numbers were assigned in alphabetical order by abbreviation. This step was necessary in order to use the *sklearn* implementation of the k-NN algorithm, which is not compatible with non-numeric data types. Only the naturally numeric columns with non-missing values were used in the calculation of the Euclidean distance in order to identify the nearest neighbor; these columns were cap diameter, stem height, and stem width. Note that it would be inappropriate to use the other categorical columns in the calculation of the Euclidean distance, since the levels are not ordinal, nor truly integers. The nearest neighbor was then used as a donor for the missing value of the appropriate feature. Using the mean or median of more than one nearest neighbor as a donor for a categorical attribute would be inappropriate as well, since an average or median of integer-encoded categorical variables is meaningless. Furthermore, the mode of more than two nearest neighbors was not an option for the donor value in the *sklearn* implementation. Therefore, it is best to use only the single nearest neighbor as a donor, in order not to distort the meaning of the categorical variables.

Following imputation, the levels of each categorical variable were manually converted back into strings. The full word provided by the dictionary was utilized rather than the original abbreviation to allow for easier reference and interpretability. Next, the remaining thirteen variables were dummy encoded. During dummy encoding, categorical columns with *k* levels were replaced with *k-1* binary indicator columns representing all levels but the reference level. The reference level for each variable was the level that came first alphabetically, by default. Dummy encoding was used rather than one-hot encoding in order to mitigate the effect of the dimensionality increase. Since logistic regression was chosen as the model-building framework, the intercept accounted for the contributions of the reference levels.

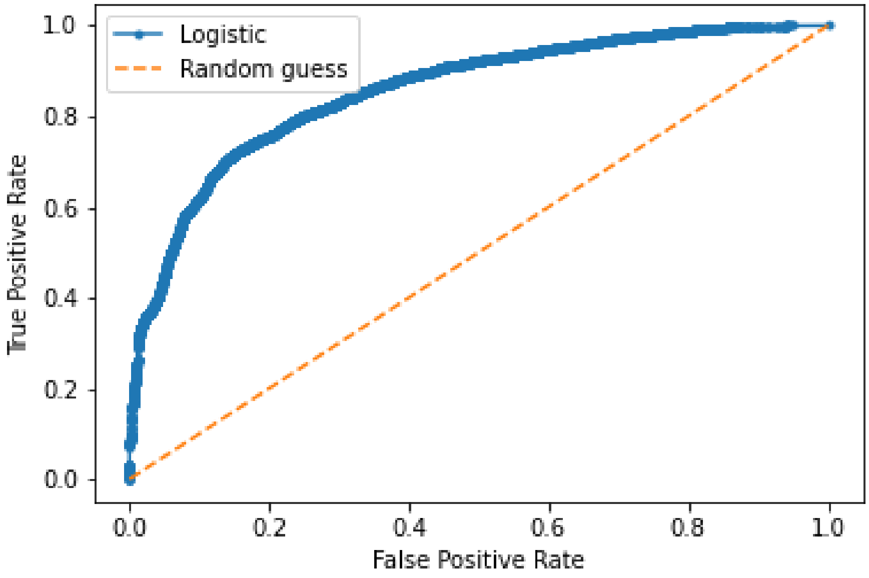
Logistic Regression

Ten-fold cross-validation was used to identify the optimal penalty parameter for Penalized (Ridge) Logistic Regression. Recall was the metric used to identify the optimal hyperparameter. Recall was chosen over accuracy as the tuning metric because in context, it would be more important to identify truly poisonous mushrooms and minimize the occurrence of poisonous mushrooms being predicted as edible. Ten different penalties were assessed ranging from 0.0001 to 1000. The optimal penalty was 2.793 with a recall of 0.7492. However, the recall values of the other penalties were not less than 0.02 of the recall of the optimal hyperparameter.

The no information rate of the test data was 55.941%, which implies that 55.941% of the observations in the test data were classified as poisonous, and the remaining observations were classified as edible. Any model with an accuracy higher than the no information rate would indicate that the model performed better than predicting the most commonly occurring class for each test observation. The accuracy of the final model (the model with fit on all data with the optimal penalty) was 0.75385, meaning that 75.385% of the observations in the test set were correctly classified. The precision was 0.80345, meaning that of all mushrooms that were identified as poisonous, 80.345% truly were. The recall was 0.72322, meaning that 72.322% of mushrooms that were truly poisonous were correctly identified as poisonous. Lastly, the F-value was 0.76123.

The ROC curve is pictured in figure 1. Based on the moderately strong concave downward curvature shown in the plot, the model performs considerably better than random chance. The area under the curve is 0.858

Figure 1: ROC Curve: Penalized Logistic Regression (Ridge)

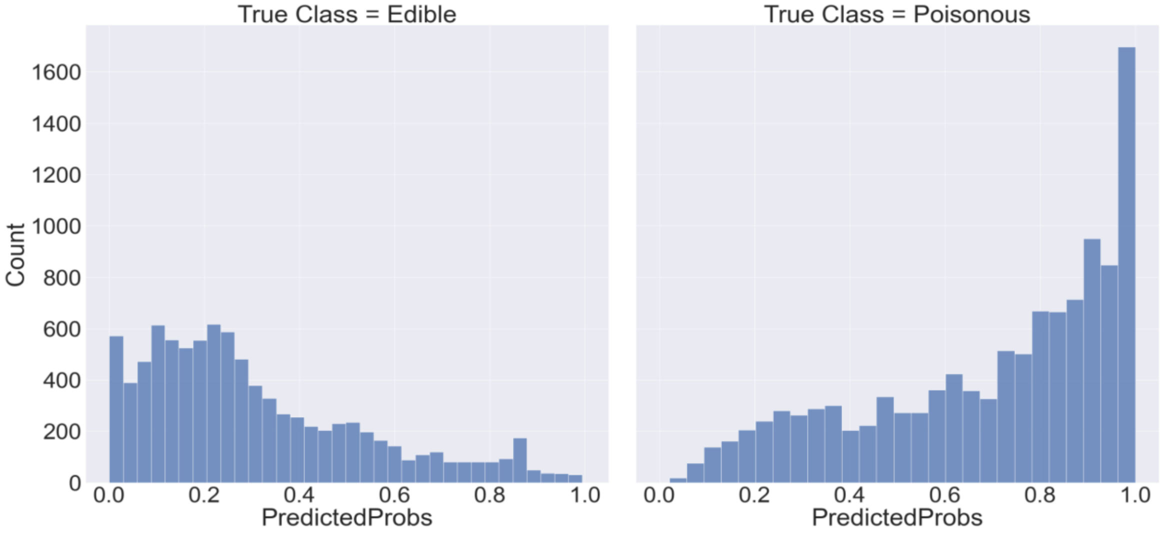


The confusion matrix of the model is illustrated below in Table 1. As reflected in the accuracy stated above, the model performed moderately in detecting true cases.

Table 1		
	Predicted Poisonous	Predicted Edible
Actually Poisonous	7272	1779
Actually Edible	2783	8523

Figure 2 illustrates the predicted probabilities for the test data faceted by the true outcome. The model does a fairly good job at identifying edible and poisonous mushrooms, but tends to do better at identifying edible mushrooms, as indicated by the fatter left tail in the ‘Poisonous’ distribution. In an ideal graph, a bathtub shaped curve would be expected across the two facets. For truly edible mushrooms a low probability of poisonousness would be expected; vice versa, for truly poisonous mushrooms a high probability of poisonousness would be expected. This phenomenon is present in figure 2.

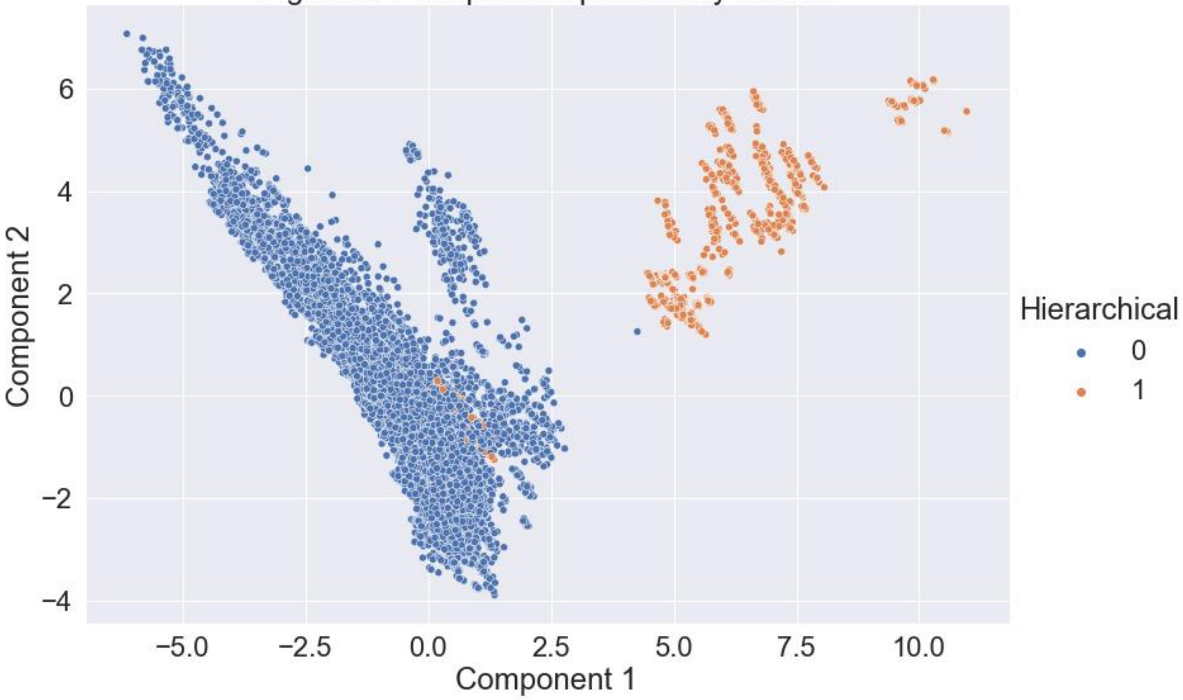
Figure 2: Predicted Probabilities Faceted by True Outcome



Clustering

Hierarchical clustering was used on the predictor columns to generate two clusters. Hierarchical clustering was chosen because of its effectiveness on datasets with many categorical variables. The data were scaled prior to clustering. For the purpose of visualization, principal component analysis was performed on the predictor variables to reduce the data down to two columns. In figure 3 below, the principal components are plotted and colored by cluster. There is one cluster that contains the majority of the points on the left half of the graph, and another cluster that contains the remaining points on the right half of the graph.

Figure 3: Principal Components by Cluster



As shown in figure 4, when color coded by true class, there is no clear distinction between the two outcomes. In essence, poisonous and edible mushrooms do not appear to be associated with one cluster or another. The lack of correlation between the clusters and the true class is reflected in table 2 as well.

Figure 4: Principal Components by True Class

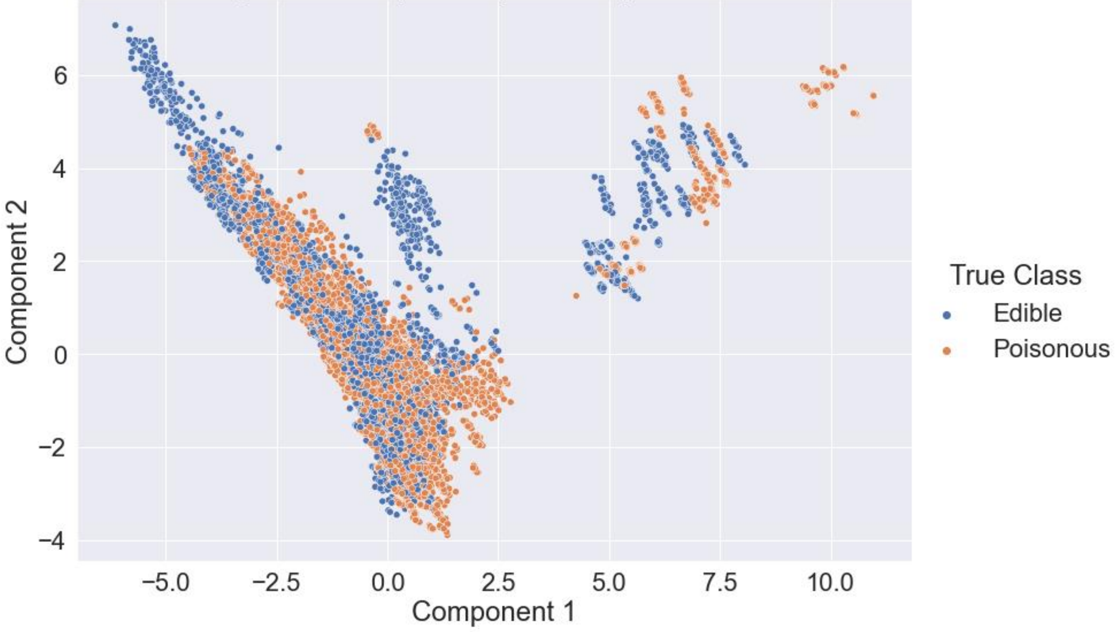


Table 2: Observation Counts by Clustering Assignment and True Class

	True Poisonous	TrueEdible
Cluster 0	10400	8580
Cluster 1	906	471

Conclusion

Overall, the logistic regression performed moderately in predicting the class outcome, and the clustering method effectively clustered the observations into two distinct groups, but the clusters were not strongly correlated with the edibility of the mushrooms. With a modest recall of 72.322%, it might not be wise to use this model in practice to predict the probability of a poisonous mushroom for the purpose of consumption. Future exploration might include experimenting with cut point adjustments for dichotomizing a predicted probability into the class outcome. Furthermore, it might be worthwhile to consult with a subject matter expert to better understand the features which are strongly correlated with the edibility of wild mushrooms, and consider these aspects more heavily during model building.