Mushroom Classification Using Logistic Regression, Linear Regression, and Naïve Bayes

Nicolas DeMilio
Department of Computer Science
SUNY New Paltz
Email: demilion1@newpaltz.edu

***Abstract*—Mycotourism, also known as Mushroom foraging, is the practice of collecting wild mushrooms for consumption or recreational use. However, this hobby comes with some risk from the uncertainty when distinguishing the difference between poisonous and edible mushrooms. This is where data driven classification comes into play; With the use of the UCI Mushroom Dataset, which contains roughly eight thousand instance of mushrooms while recording 22 categorical features, one can make use of machine learning algorithms such as Naive Bayes, Linear Regression, and Logistic Regression to identify which mushrooms are edible and poisonous. The process to implement these algorithms encompass data cleaning, categorical feature hot-encoding, dataset partitioning, and model training. TODO: ATTACH RESUTLS**

I. Introduction

Mycotourism, a popular activity all around the world in countries like Spain, Poland, the Netherlands, and many more. Originally used for survival, mycotourism has developed into a popular hobby by many recreationally. Keeping in mind the overhead or risk when participating, one can become extremely ill with just one misidentification.

By capitalizing existing machine learning algorithms such as Naive Bayes, Linear Regression, and Logistic Regression one can aid in a mushroom edibility classification. Features such as cap color, odor, and habitat to apply the various supervised machine learning algorithms for binary classification. Additionally, one can compare each algorithm to one another to determine effectiveness on the dataset.

The problem being investigated can be formally defined as: Given a dataset consisting of categorical mushroom attributes with a tag for edibility, is it possible to train a machine learning model to identify whether or not a mushroom is edible? This problem consists of a binary classification that is poisonous or edible based on given categorical attributes. To accomplish this we use the UCI Mushroom dataset that is presented as a csv. Clean the data by removing incomplete entries and hot-encoding columns accordingly. Doing this minimalizes error and increases the model's reliablilty because it ensures a complete and high quality dataset. Then each model was trained on a partitioned portion of the dataset and evaluated on the other.

II. Motivation

The motivation for this investigation stems from public health and data science perspectives. Just alone in the USA there are over 7,000 cases of poisoning from mushrooms a year. The majority of these incidences are due to misidentification of mushrooms. The silver lining is that most these incidents aren't fatal but there are a couple couple lives taken among the statistics. Mushroom foraging has typically been a hobby typically passed through social learning, person to person. Given the state of the data available and machine learning algorithms, it is assumed that machine learning cannot replace the knowledge held by hobbyists and experts. Therefore it would be better to supplement the identification of edibility. From the data science perspective this type of categorical data is ideal for a machine learning model. Therefore its a great opportunity to compare different models doing the same task to emphasize strengths and weaknesses between one another.

III. Related Work

The UCI Mushroom dataset is a well known benchmark for binary classification tasks. There are 77 papers known to be have cited this dataset at this time (Oct 2025). The earlier papers were known to use this dataset for data mining research whereas whithin the last 10 years the focus really turned to machine learning research. Most of these works are focused on the technical advancement of solving classification problems. A paper from May 2019 by Taiping He and others, called **High-Performance Support Vector Machines and Its Applications** was focused on scaling the support vector machine classification technique via cloud computation disrtibution technique along with minimizing intercommunications between machines.

The tools resulting from these algorithms are a fantastic supplement for practicing safe foraging practices. For example a tool like this would help prevent cross contamination between mushrooms in the sense of putting a poisonous one in with the edible mushrooms, although they should be kept separated by type until consumption.

Going outside the scope of the dataset, there are studies done via computer vision and deep learning in attempt to capture some of the cultural or ecological aspects. Aspects such as the environment its in as well as species. There are many european cultures that are deeply intertwined with mycology. There was a social study conducted in Poland in March of 2024 by Mikotaj Jalinik called **Mushroom Picking as a Special Form of Recreation and Tourism in Woodland Areas**. Of which focused on the benefits of mushrooms in

health, recreation, and tourism. In poland mycology has proven to play a large cultural role in recreation and a tourism asset. However, it is still found that the knowledge of health benefits of mushrooms seems to be a mystery to most.

IV. Methods

A. Dataset

The Mushroom dataset contains 8124 entries of mushroom examples while covering 22 features for each example. We obtained this dataset from Kaggle.com.Each mushroom is labeled as either edible (e) or poisonous (p). Features include attributes such as cap shape, surface, color, odor, gill size, and habitat. Each feature has categorical values consisting of letters $ie.\ mushroom\ color\ =\ red\ (r)$, shows how a color can be represented by a letter. The dataset was made in 1981 by the University of California, Irving, and spans 23 different classes of mushrooms. It's worth mentioning that the dataset states that there is no difinitive rule for declaring mushrooms to be poisonous, ie. "leaflets three, let it be" for poison ivy/oak.

B. Data Preprocessing

For preprocessing we started by cleaning the dataset via removing imcomplete entries. In other words we remove examples that contain any features missing. This allows us to avoid any bias coming from the model not being able to identify trends among features. This step also generally increases the quality of the data, but also decreases the amount of entries by roughly 30%. Then we fixed the naming convention to follow snake case, `multi_word_var`, from the starting kebab case, `multi-word-var`. Then all that was left was to shuffle the dataset with a seed for reproducability and separate features from outputs. The reason we separate features and outputs is to give the model a clear prediction target.

C. Algorithms Implemented

Each algorithm implementation on the data set followed the same method with some minor differences. First is to partition the newly cleaned dataset into 2 partitions, training and testing, we start with and 80/20 split respectively. From there generically fit the line using the base paramters provided, this gives a preview to the accuracy. Then since the dataset is small, cross-validation is a great resource to really validate our data. We use a k-fold cross validation algorithm with 5 folds. Its standard to use folds in multiples of 5. The three algorithms we train with are the following:

- **Logistic Regression:** Used for binary classification. Regularization was applied to prevent overfitting. We use hyperparamters, which in this context are learning settings for the model, such as ensuring even spacing between a feature's categorical range, prevent coefficients from going to zero, and ensure we use stochastic gradient descent. Stochastic gradient descent uses one example at a time to reduce variances in dataset, its means is using a memory of past gradients to reduce noise, it takes longer but is effective in this case due to our small dataset.
- **Linear Regression:** Used as a baseline to demonstrate the faults of regression algorithms for categorical outputs. We didn't do anything special to these hyparameters except using a scaler serves the purpose of ensuring each feature's range is proportionate for each category encapsulated.
- **Naïve Bayes:** Implemented using a categorical distribution (GaussianNB). Suitable for this dataset since features are discrete and can be assumed to be independant of one another given the current state of information on mushroom identification. For the hyperparameters we implement "variable smoothing" to ensure coefficients do not sky rocket to 1 or plummet to 0. If this were to happen we'd obtain severely underfitted skewed results.

D. Evaluation Metrics

Model performance was evaluated using multiple metrics and visualizations to provide comprehensive assessment. Each model was evaluated on accuracy, precision, recall, F1-score, Mean Squared Error (MSE), and Area Under the ROC Curve (AUC). Confusion matrices visualize classification performance by showing true positives, true negatives, false positives, and false negatives for each model.

The ROC curve shows the true positive (TPR) rate for reach false positive rate (FPR), of which you can take the integral of the TPR with respect of FPR to get the area under curve. This allows to see how the model variably ranks positive scores to negative. Its a way of seeing different threshholds, in otherwords how does one trade off with the other. Which is why the top left of corner of the graph is the sweet spot for the curve, it shows if we can have a high true rate with a low negative rate. However, this method of evaluation ignores model calibration and absolute score values. Therefore we must use other forms of calibrated metrics for a complete idea.

The confusion matrix for Logisic Regression above summarizes where the model's flaws would be, in the context of actual edibility of the mushroom and the model's prediction for each test sample respectively.

The feature plot is used to get an inside peek to understand how the model is using each feature in relation to it's output. We get an inside peek on each feature using coefficients as the metric, it basically provides the weight of each feature.

V. Results

TODO: IMPLIMENT precision and recall and F-measure

A. Logisitic Regression

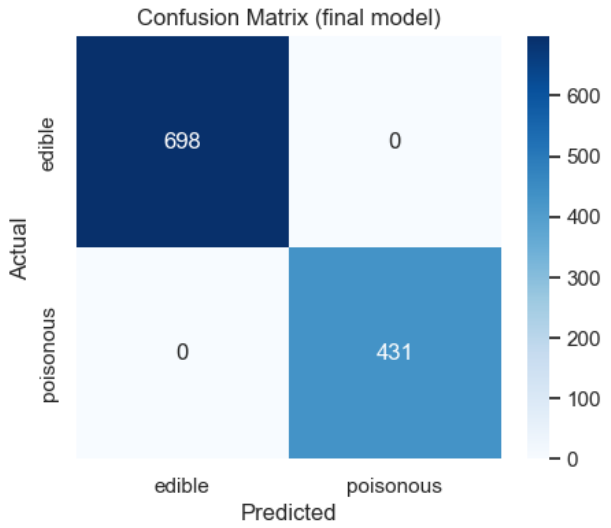

Fig. 1: Confusion Matrix for Logistic Regression.

Since True Positive and True Negative values completely made up the testing output, it can be assumed that the model is vary capable of class separation (the model has a clear sense which features are influential over output). However, the results being literally perfect tell us that we should be checking for data leakage or duplicated rows which could be the reason this is we are getting these results. Otherwise the features are just simple to predict.

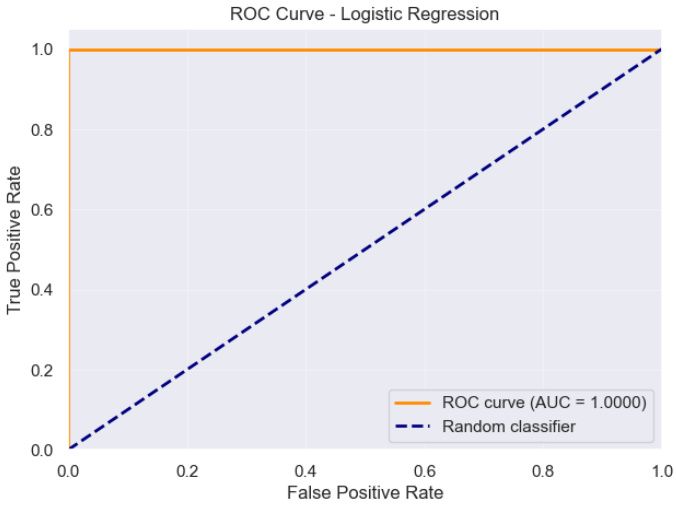

Fig. 2: ROC curve with AUC score for Logistic Regression.

The area under curve is a 1 meaning that we can have a perfect model with no false positives with maximum true positives. Given that this seems too good to be true we must check the other metrics.

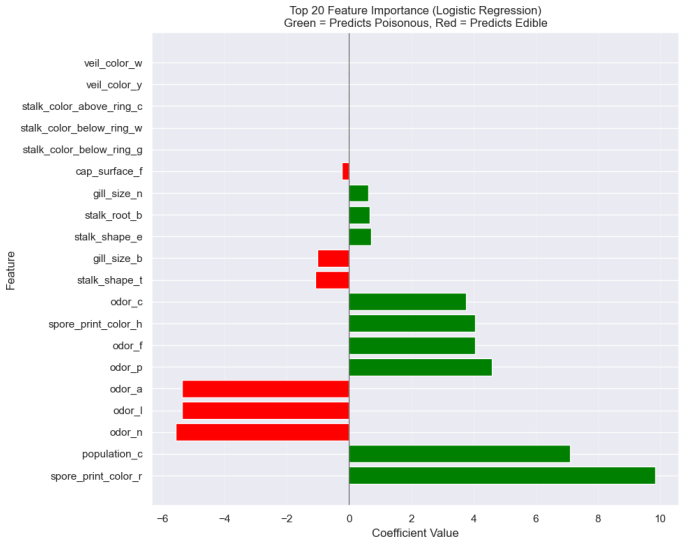

Fig. 3: Log feature weights

All this chart is showing is that the categories odor, spore print, population, gill size, and stalk root have is what are mainly considered when predicting edibility. You can notice this by taking account of how many instances of each feature are present. Additionally, we can see that odor is the main contributor for declaring a mushroom poisonous. This is deduced from the majority of the red values being associated with the various odor categories. Whereas edible mushroom predictions take more into acc

B. Linear Regression

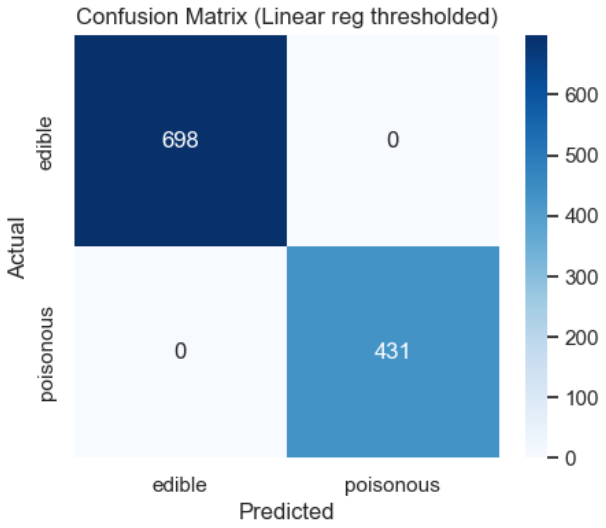

Fig. 4: Confusion Matrix for Linear Regression.

The confusion Matrix for the linear regression baseline is supposed to show where the regression to threshold method fails for categorical classification. However, in this case it seems to score just as well as logistic regression. This can

be due to data leakage when training, or the simplicity of the dataset.

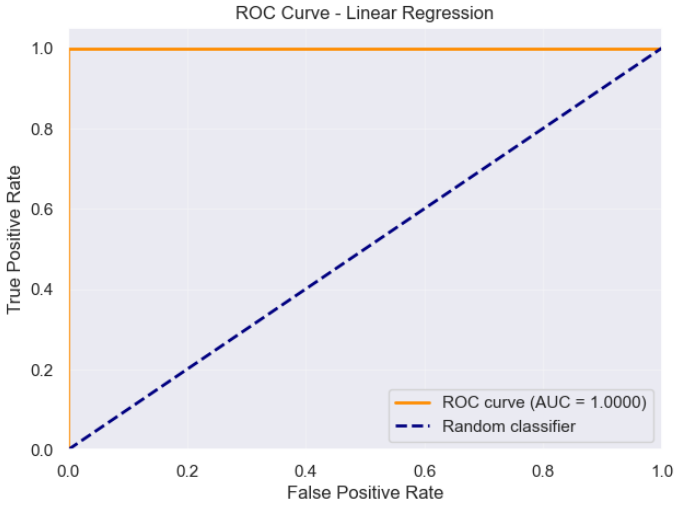

Fig. 5: ROC curve with AUC score for Linear Regression.

Given that the confusion matrix was a perfect output, it is a bit redundant to provide the ROC curve. However, just to cover all points we see that it also scored perfectly indicating to look at the other calibrated metrics.

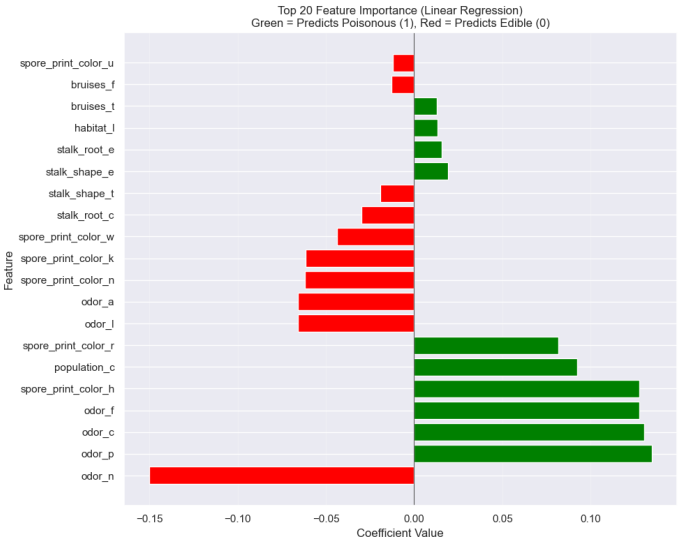

Fig. 6: Linear model feature coefficients (weights).

As seen in the figure above we notice that the model scored the features very similarly with a slight change in weights. This aids in consistency about the features of the dataset being simple to predict. There is a commonality with scoreing features like odor, spore print color, and stalk roots having heavier weights for prediction.

C. Naïve Bayes

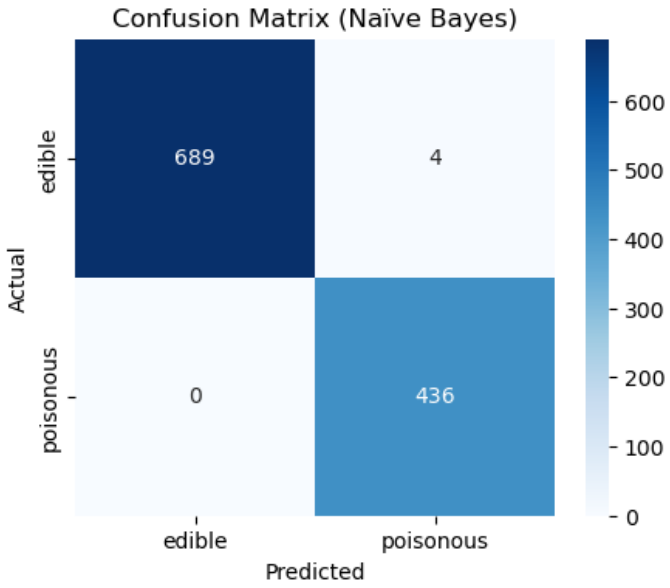

Fig. 7: Confusion Matrix for Naïve Bayes feature

The confusion matrix for the Naïve Bayes model was the only one that had any false negatives. This shows that this model doesn't have perfect accuracy, but pretty darn close. This matix highlights that the model is more prone to give false negatives which isn't the worst. We'd rather have false negatives over false positives in the implementation due to the use case. We don't want a user to trust the machine full heartedly and eat a poisonous mushroom just becuase it was predected to be edible. For the safety of the user it is better to have false negatives for the idea that a mushroom could be poisonous so why take the chance eating itwhen you don't have to do so.

D. Naïve Bayes

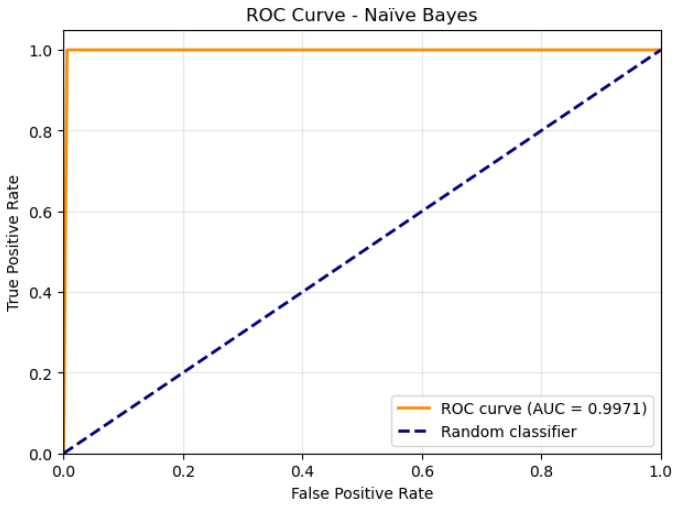

Fig. 8: ROC curve with AUC score Naïve Bayes feature

The area unde the ROC curve is 0.997, this shows that the model is borderline perfect. Although the graph doesn't quite capture it the there is a small space to above the line on the left side which is encapsulating the error. Even though we see a very accurate model, this doesn't mean that the ignoration of the calibrated metrics won't impact this finding.

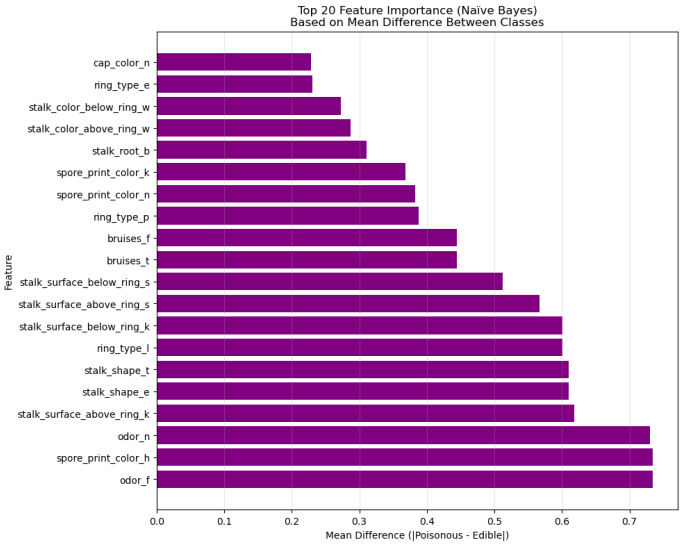

Fig. 9: Naïve Bayes feature importances (class-conditional differences).

The figure above shows the features taken into account independently as per the Naïve Bayes assumption of features, therefore we use feature conditional statistics to show the feature weights. This is done via the mean difference between poisonous and edible for each feature. Diving into the data shown, we see that the same features are weighted heavier similar to the other two machine learning models implemented. The features weighted heaviest are odor, spore print color, and stalk shape. This commonality really drives home that these features have the most to do with a mushrooms edibility.

VI. Conclusion

TODO: Implement

References

[1] G. Alexander, "Responsible Mushroom Hunting," Earth911, Mar. 14, 2023. [Online]. Available: https://earth911.com/home-garden/responsible-mushroom-hunting/

[2] M. Jalinik, T. Pawłowicz, P. Borowik, and T. Oszako, "Mushroom Picking as a Special Form of Recreation and Tourism in Woodland Areas—A Case Study of Poland," Forests, vol. 15, no. 3, p. 573, 2024, doi:10.3390/f15030573.

[3] I. Svanberg and H. Lindh, "Mushroom Hunting and Consumption in Twenty-First Century Post-Industrial Sweden," Journal of Ethnobiology and Ethnomedicine, vol. 15, no. 1, 2019, doi:10.1186/s13002-019-0318-z.

[4] T. He, T. Wang, R. Abbey, and J. Griffin, "High-Performance Support Vector Machines and Its Applications," arXiv preprint, 2019. [Online]. Available: https://www.semanticscholar.org/paper/High-Performance-Support-Vector-Machines-and-Its-He-Wang/72200eb638423bb0810c40eb804633299bb184b9

[5] UCI Machine Learning Repository, "Mushroom Data Set." [Online]. Available: https://archive.ics.uci.edu/dataset/73/mushroom

[6] UCIML, "Mushroom Classification," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/uciml/mushroom-classification

[7] UCI Machine Learning Repository, 1981. [Online]. Available: https://doi.org/10.24432/C5959T

[8] Becoming Better, "Classification with Machine Learning (Python) — Mushroom Dataset," Medium. [Online]. Available: https://medium.com/becoming-for-better/classification-with-machine-learning-python-mushroom-dataset-790a275610df

[9] R Project, "Mushroom — arulesCBA Dataset Documentation." [Online]. Available: https://search.r-project.org/CRAN/refmans/arulesCBA/html/Mushroom.html

[10] UCI Machine Learning Repository, "Datasets related to ecology." [Online]. Available: https://archive.ics.uci.edu/datasets?Keywords=ecology

[11] Scikit-learn contributors, "scikit-learn: Machine Learning in Python." [Online]. Available: https://scikit-learn.org/stable/index.html

[12] Scikit-learn contributors, "Nested versus non-nested cross-validation." [Online]. Available: https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html