Autor: Antonio Ortiz

# Intent of the application

The purpose of this application is to classify a variety of mushrooms into several categories, depending on whether they are visually pleasing or not, pleasantly smelling or not, or toxic or not, and use this information in order to create a stew with the mushrooms classified as visually pleasing, nice smelling, and not toxic.

However, these categories were defined in a completely subjective manner, as visually pleasant and nice smell are both subjective. Therefore, these categories and the resulting mushrooms are only valid for the people involved in the determination of the classification.

# Functional description

This application will use the Mushroom database, available through the UC Irvine Machine Learning Repository site, to perform basic exploration of the data, observe the distribution of the data, as well as the distribution in relation to the poisonous mushrooms.

Afterwards, several classification techniques will be used on the data to be able to classify mushrooms in the 8 main categories. Four will be supervised classification techniques, and two fill be unsupervised.

# Dataset to be used

Mushroom dataset: This dataset contains 8124 observations of different mushrooms, each with 22 attributes, and one resulting label.

The attributes include information on the visual characteristics of the mushrooms, their smell, and, using this information, whether they are poisonous or not could be predicted.

As the label for poisonous had already been predicted with a high accuracy, it was directly used as one more category in the dataset.

The dataset can be found here: https://archive.ics.uci.edu/dataset/73/mushroom

# Mathematical background

## KNN (K-Nearest Neighbors)

It is a non-parametric supervised learning method, used for both classification and regression.

In the case of this application, it will be used for classification.

The only argument that the algorithm requires is the number of clusters, and it is optional.

In the case of this application, two attempts were made: one where only four classes were considered (those that actually had members) and another attempt in which the full categories (8) were taken into consideration. The results are the same, as the later 4 categories did not have a single element in them.

## Decision Tree

Non-parametric, supervised learning algorithm. It can be used for both regression and classification, but for the purposes of this application, it will be used for classification.

It uses a tree-like model of decisions in order to get to the result.

It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

## Multiclass Logistic Regression

Also known as multinomial logistic regression, it is a classification technique that generalizes the logistic regression algorithm into multiclass problems.

It works by predicts the probabilities that a given element belongs to a class. The class with the highest probability is the class to which the given element belongs.

One of the conditions for this technique is that the dependent variable (label) is nominal, which means that the categories cannot be ordered in any meaningful, other than arbitrarily decided.

It also requires that each observation belongs to only one category.

## Random Forest

Ensemble learning method (it uses multiple instances of other learning algorithms) used for classification, regression, and others. In the case of this application, it will be used for classification.

It uses multiple instances of decision trees, and the output of the random forest is the solution to which most decision trees arrived to. By using the output of multiple decision trees, it maximizes the changes that the prediction is correct.

## DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

It is a density-based, non-parametric data clustering algorithm.

It is a not-supervised algorithm, so it does not require the expected labels as an input.

It uses the concept of closest neighbors to group together closely-packed points, marking as outliers those points that are in sparsely-populated areas.

## K-means

Vector quantization method used to partition n observations into k clusters. Each observation belongs to the cluster with the closest mean, and each cluster updates its own mean each time it “absorbs” a new element.

It is an unsupervised technique.

# Use case

This application can be used by anyone looking to prepare a mushroom stew, or just any dish related to mushrooms. The answers presented in this specific iteration of the application are subjective, according to the tastes of the person creating the application but can be easily modified to accommodate the tastes of the users.

# Variables

* poisonous
* 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
* habitat

# Labels

The expected labels for the classification algorithms are the categories to which each observation of the mushrooms belongs to. These are the categories:

* VPPSNT: Visually pleasing, pleasantly smelling, and not toxic.
* VPPST: Visually pleasing, pleasantly smelling, and toxic.
* VPNPSNT: Visually pleasing, not pleasantly smelling, and not toxic.
* VPNPST: Visually pleasing, not pleasantly smelling, and toxic.
* NVPPSNT: Not visually pleasing, pleasantly smelling, and not toxic.
* NVPPST: Not visually pleasing, pleasantly smelling, and toxic.
* NVPNPSNT: Not visually pleasing, not pleasantly smelling, and not toxic.
* NVPNPST: Not visually pleasing, not pleasantly smelling, and toxic.

# Data import

In this application, there is no input needed from the user.

## Proposed Libraries

Seaborn: Data visualization library based on matplotlib. Source: https://seaborn.pydata.org/. Version: 0.11.2

Sklearn: Machine learning tool for predictive data analysis. Supports both supervised and unsupervised learning. Version: 1.0.2

Source -> https://scikit-learn.org/stable/getting\_started.html

Numpy: Used for vectorization and indexing for scientific computing. Version: 1.25.1 Source -> https://github.com/numpy/numpy

Pandas: Data analysis and data manipulation library. Version: 2.0.3 Source -> https://pandas.pydata.org/getting\_started.html

Matplotlib: Library for creating static, animated, and interactive visualizations in Python. Source:3.7.2 https://matplotlib.org/stable/users/release\_notes.html.

# Plots

Gráfico, Gráfico de barras

Descripción generada automáticamente

Figure 1 Bar plot for the distribution of the cap shape in reference to the poison

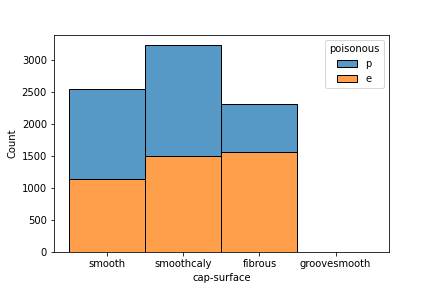


Figure 2 Bar plot for the distribution of the cap surface in reference to the poison

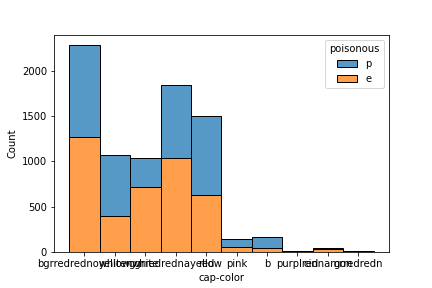


Figure 3 Bar plot for the distribution of the cap color in reference to the poison

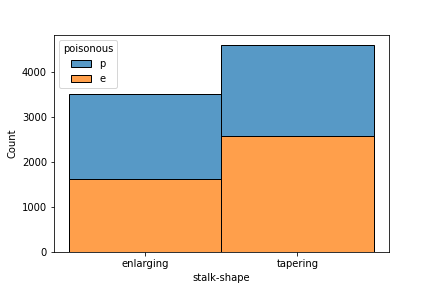


Figure 4 Bar plot for the distribution of the stalk shape in reference to the poison

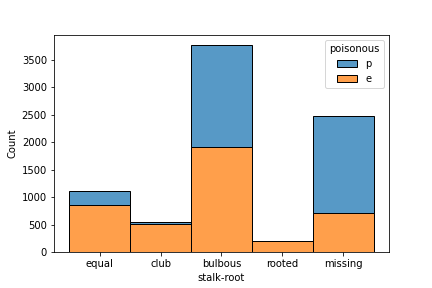


Figure 5 Bar plot for the distribution of the cap root in reference to the poison

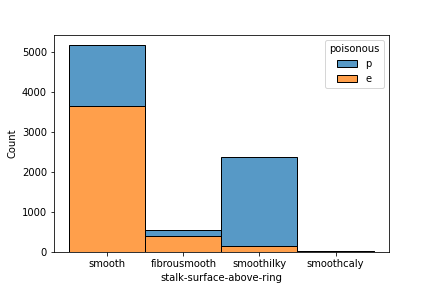


Figure 6 Bar plot for the distribution of the cap surface in reference to the poison

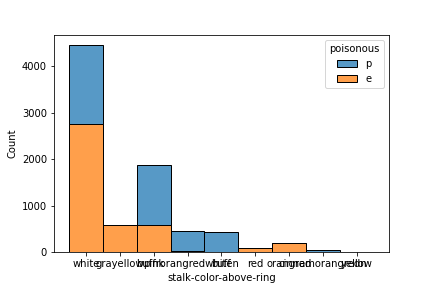


Figure 7 Bar plot for the distribution of the cap color in reference to the poison

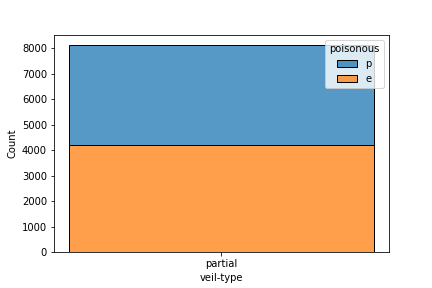


Figure 8 Bar plot for the distribution of the veil type in reference to the poison

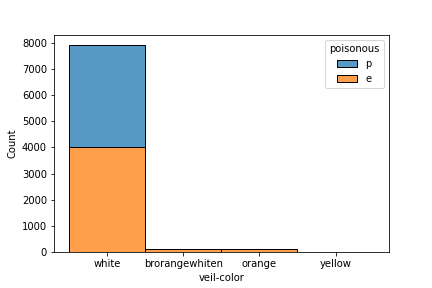


Figure 9 Bar plot for the distribution of the veil color in reference to the poison

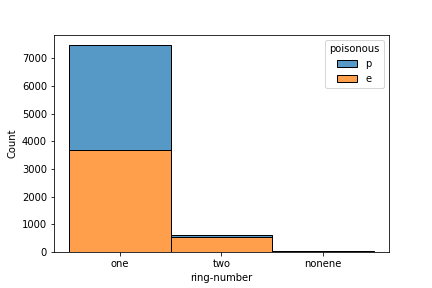


Figure 10 Bar plot for the distribution of the ring number in reference to the poison

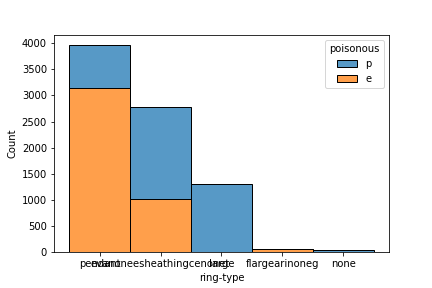


Figure 11 Bar plot for the distribution of the ring type in reference to the poison

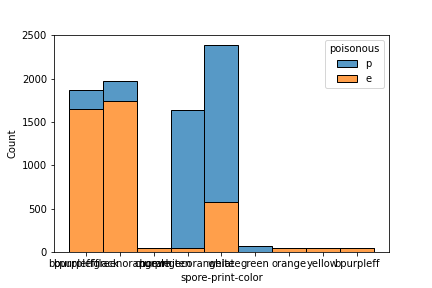


Figure 12 Bar plot for the distribution of the spore print color in reference to the poison

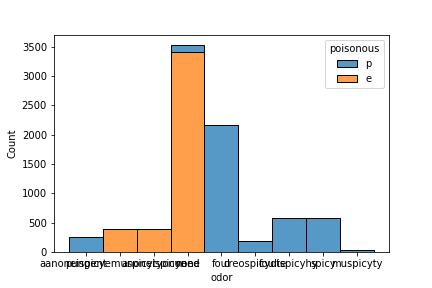


Figure 13 Bar plot for the distribution of the odor in reference to the poison

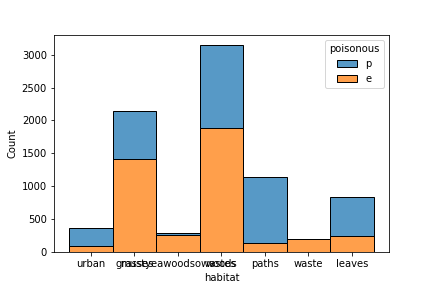


Figure 14 Bar plot for the distribution of the habitat in reference to the poison

# Observations

In order to use the classification algorithms, the data had to be transformed, as the variables and labels were categorical.

The data was hot encoded.

With this transformed data, the algorithms were trained.

## KNN

After the data was hot encoded, the model could be trained.

The first time, it was trained with the assumption that there were only 4 categories, as only 4 categories had elements. With this, every mushroom was correctly classified.

As there were a total of 8 categories, the algorithm was trained again, but this time, including the remaining 4 categories.

However, as these categories did not include a single element, the results were the same, and every mushroom was correctly classified.

## Decision Tree

The criterion for the training of the decision tree was “Entropy”.

Another parameter was the maximum depth for the tree. Two values were given: 3 and 10.

The first time it was trained, it was given a maximum depth of 3. The results were acceptable, but approximately 200 mushrooms were incorrectly classified.

Then, the algorithm was trained again, with a maximum depth of 10. The results improved, as every mushroom was correctly classified.

## Logistic Regression

This algorithm does not accept multiple labels as the output, but it accepts multiple classes for a single label, so the process for one hot encoding for the labels were reversed.

With this data, the algorithm was trained with a maximum iterations parameter of 1000.

The results were satisfactory, as every mushroom was correctly classified.

## Random Forest

The same procedure was followed as the one in Decision Tree, and the results were the same.

With this algorithm, it was possible to see the most relevant categories and values for the predictions.

The most relevant categories were:

* Poisonous: True
* Odor: F
* Odor: N
* Gill size: N
* Odor: S
* Spore print color: H
* Stalk surface ring: k
* Odor: Y
* Ring type: L
* Ring type: P

## DBSCAN

The parameters used for the training in this algorithm were:

* EPS: Maximum distance between two samples for them to be considered the same cluster. It was chosen to be 2.9.

This value returned exactly 9 categories, which were very close to 8, and the values for the categories were similar to the ones observed with the manual filtering. However, the actual predictions were too different to the expected values, as over 3000 mushrooms were incorrectly classified.

## K-means

For this algorithm, the only parameter used was the number of clusters, which was set to 4, just as the KNN algorithm.

However, the results were unsatisfactory, as every cluster was similarly distributed, which is not what the expected result is.

# Conclusion

The conclusions for the results of the execution of this application are subjected to completely subjective observations of the data, such as which categories are visually pleasing in combination, and which smells are pleasantly smelling. The only relatively objective category is toxicity, as this is a category predicted by a previous model, and the toxicity of a mushroom is an objective measurement. However, the grade of toxicity, although objective, can be taken as pleasant or not, depending on how toxic it is, the effect of its toxin, as well as the grade.

Therefore, the results of the application can easily change if the categories are changed in relevance by another user of this application that decides which mushrooms are visually pleasing, and which smells are pleasantly smelling. This means that any results discussed here can only be applied to the exact categories previously chosen.

In the case of these subjective measurements, it can be observed that most mushrooms were classified as not visually pleasing. Only 192 out of 8124 mushrooms were considered visually pleasing.

Regarding the smell, 5480 mushrooms were classified as pleasantly smelling.

Finally, for toxicity, 4208 mushrooms were considered safe to eat.

With these three categories, another 8 categories were obtained, combinations of them.

However, only 4 of these new categories have elements in them. Therefore, the classification algorithms could only predict these 4 categories.

The results for the supervised algorithms were the same: Every mushroom was correctly predicted and correctly classified.

Unfortunately, the results for the unsupervised algorithms were incorrect.

With these observations, it can be concluded that supervised algorithms provide better results for data that is classified in a completely subjective manner, and the labels can be manually filtered.

The unsupervised algorithms did not provide the expected answers, as the clusters they may have created could be correctly clustered, but in an objective manner, which is not relevant for the purposes of this application.