ISyE 6740 - Spring 2022

Detecting the edibility of Mushrooms in the wild

What is the least number of features to remember to be safe?

Saturday 30th April, 2022

Ayush Subedi (asubedi6@gatech.edu) - Gatech ID: 903743953

Contents

1 Problem Statement **1**
1.1 Introduction . . . 1
1.2 Problem . . . 1
1.3 Questions . . . 2

2 Dataset **3**
2.1 Introduction . . . 3
2.2 Additional Questions . . . 5

3 Methodology **6**

4 Evaluation and Final Results **7**
4.1 Additional Questions . . . 7
4.1.1 Correlation when all features are categorical . . . 7
4.1.2 Outlier Detection when all features are categorical . . . 10
4.1.3 Dimensionality reduction for data visualization when all features are categorical . . . 12
4.2 Classification . . . 16
4.3 Feature Selection . . . 18
4.4 Minimizing False Positives on edibility . . . 20

5 Conclusion **22**

List of Figures

1 Example of mushroom spices with edible and non-edible counterparts [13] . . . 1
2 Number of poisonous and edible samples . . . 3
3 Distribution of samples by edibility and features . . . 4
4 Top 10 positively correlated features with edible and poisonous mushroom respectively . . . 7
5 Cramér's V scores . . . 9
6 Elbow diagram for optimal clusters . . . 11
7 Cluster assignment . . . 11
8 PCA for data visualization . . . 12
9 Contingency table for "odor" and "spore-print-color" . . . 13
10 Correspondence Analysis for "odor" and "spore-print-color" . . . 13
11 MCA for data visualization . . . 14
12 t-SNE for data visualization . . . 15

13 Decision Tree Plot 16
14 Parameters used in Classification models above 17
15 Cramér's V scores 18

1 Problem Statement

1.1 Introduction

Mushroom foraging has garnered a lot of interest as a popular recreational activity for quite some time now [10]. Apart from its recreational indulgence, in several parts of the world, there are people who rely on wild mushroom as a source of food. However, there is a famous adage in the mushroom foraging community, ”Every mushroom is edible … once” [9]. This is because, the consequences of even a minor misjudgement can be lethal. Through generational knowledge transfer and documentation on various species of mushrooms, we have been able to categorize poisonous and non-poisonous mushrooms.

Figure 1: Example of mushroom spices with edible and non-edible counterparts [13]

1.2 Problem

News of hospitalizations and deaths due to poisonous mushroom consumption is commonplace, especially in my home country of Nepal [8]. In the remote areas of the country, foraging the forest to search for mushroom as a source of food is pretty common. There is plenty of research that suggests there is no simple guide for mushroom edibility [14]. For example, there are counterexamples of non-edible mushrooms in several categories of mushrooms; the categories are created using features that are shared among plenty of edible mushrooms. This makes it pretty difficult to make people aware of what properties might constitute separating poisonous mushrooms and edible mushrooms. This is also evident in Figure 1.

It is almost impossible to memorize all the mushrooms of all the species and if its edible. However, is it possible to memorize a few features that might be useful in identifying the edibility? This project focuses on seeking to answer this from a feasibility point of view and if the

answer is yes, identifying the select set of features. Given that a false prediction on edibility is life threatening, the project also seeks answers with the objective of minimizing false positives.

1.3 Questions

1. What is the best model for the detection of the edibility of a mushroom?
2. What is the best model with the least number of features when detecting the edibility of a mushroom?
3. What is the best model with the least number of features that gives the least false positive rate when detecting the edibility of a mushroom?
4. And finally, can this model be used reliably in practise?

2 Dataset

2.1 Introduction

The dataset is available at UCI Machine Learning Repository and the origin of the dataset is the mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms (1981) [5]. This dataset includes descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom. Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one.

After the combination, there only are poisonous and edible mushroom samples. Therefore, this becomes a binary classification problem moving forward.

- There are 8124 samples and 22 features (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).
- **All the features are categorical**. This is particularly interesting because some of the topics covered in class (Principal component analysis, K-Means etc.) needs to be tweaked for this special case. One of the primary reasons for me to select this project was this property of the dataset as well.
- There are 4208 samples of edible mushrooms and 3916 samples of poisonous mushrooms. Class imbalance is not an issue here.

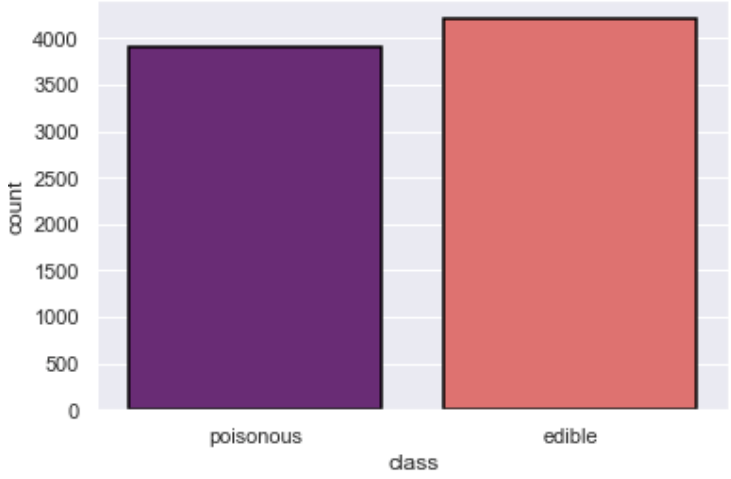

Figure 2: Number of poisonous and edible samples

- One of the features, ”veil-type” only has a single unique value. Therefore, the rest of the analysis will ignore this column.

Feature	Unique values	Values
class	2	poisonous, edible
cap-shape	6	convex, bell, sunken, flat, knobbed, conical
cap-surface	4	smooth, scaly, fibrous, grooves
cap-color	10	brown, yellow, white, gray, red, pink, buff, purple, cinnamon, green
bruises	2	bruises, no
odor	9	pungent, almond, anise, none, foul, creosote, fishy, spicy, musty
gill-attachment	2	free, attached
gill-spacing	2	close, crowded
gill-size	2	narrow, broad
gill-color	12	black, brown, gray, pink, white, chocolate, purple, red, buff, green, yellow, orange
stalk-shape	2	enlarging, tapering
stalk-root	5	equal, club, bulbous, rooted, missing
stalk-surface-above-ring	4	smooth, fibrous, silky, scaly
stalk-surface-below-ring	4	smooth, fibrous, scaly, silky
stalk-color-above-ring	9	white, gray, pink, brown, buff, red, orange, cinnamon, yellow
stalk-color-below-ring	9	white, pink, gray, buff, brown, red, yellow, orange, cinnamon
veil-type	1	partial
veil-color	4	white, brown, orange, yellow
ring-number	3	one, two, none
ring-type	5	pendant, evanescent, large, flaring, none
spore-print-color	9	black, brown, purple, chocolate, white, green, orange, yellow, buff
population	6	scattered, numerous, abundant, several, solitary, clustered
habitat	7	urban, grasses, meadows, woods, paths, waste, leaves

- One of the features, ”veil-type” only has a single unique value. Therefore, the rest of the analysis will ignore this column.

Figure 3: Distribution of samples by edibility and features

- There are 36 samples (0.44%) with missing data in the ”ring-type” feature. All of these

samples represent poisonous mushrooms. The missing data in the ”ring-type” is represented by ”none”.

- There are no duplicate samples.
- There are no two samples where all features are the same but the outcome variable is different.

2.2 Additional Questions

Apart from the aforementioned questions, the report also focuses on answering the following questions.

5. When all features are categorical, does the measure of correlation with the output variable make sense, and what are some methods to accomplish this?
6. When all features are categorical, does outlier detection make sense, and what are some effective outlier detection techniques?
7. When all features are categorical, what are some effective dimensionality reduction techniques?

3 Methodology

To answer all the aforementioned questions, the project explores the Additional Questions discussed in 2.2 first. The main reason for it is that answering these will allow exploration on the patterns and relationships in the data, which is very helpful when answering questions in 1.3.

The project explores the concept of correlation in all nominal categorical features setting. First, **Pearson's correlation**[2], which is popular in numeric features for correlation detection is explored and other techniques that are more relevant for categorical features, such as **Cramér's V**[1] is explored. The results from Cramér's V is also used for feature selection in the classification section (4.2).

The project explores the concept of outlier detection when all features are categorical. Some definitions of outliers are discussed and analysis is performed to find outliers in the poisonous class using **k-modes clustering**[4]. **Elbow method**[3] is used in determining the optimal number of clusters in the class, and analysis on the findings is performed.

The project also explores the concept of dimensionality reduction in all categorical setting. **Principal component analysis**, **Contingency Tables**, **Correspondence analysis**, **Multiple Correspondence analysis**, and **T-distributed Stochastic Neighbor Embedding (t-SNE)** is explored to explore dimensionality reduction. Biplots are visualized for all techniques and pros and cons are discussed. The python package Prince[7] is used for dimensionality reduction and visualization of biplots (for all but t-SNE where Scikit-learn[12] is used).

The project explores classification models using algorithms such as **Decision Tree classifiers, Ada Boost Classifier, Bagging Classifier, Gradient Boosting Classifier, Logistic Regression, Perceptron algorithm** etc. and compares results on validation metrics such as **Accuracy, Precision, Recall, True Positive counts, True Negative counts, False Positive counts and False Negative counts**. The Python package used is Scikit-learn[12].

Since the goal of classification is to find the best model for detection of edibility of mushroom, best model with least number of features and best model with least number of features with least false positive rate, multiple models are built and compared. Exploratory data analysis performed in 2.1 and Additional Questions explored in 2.2 provide significant insight for this section.

4 Evaluation and Final Results

4.1 Additional Questions

4.1.1 Correlation when all features are categorical

Correlation is the statistical measure of the relationship between two or more variables. The (Pearson) correlation coefficient between two random variables X and Y with expected values μ_X and μ_Y and standard deviations σ_X and σ_Y can be represented as:

$$corr(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Since all of the features are categorical in this context, one hot encoding can be used to convert categorical variables into dummy or indicator variables before testing for correlations. It might seem counter intuitive to use one hot encoding here, since the goal of the project is to find minimal features that generate high accuracy. However, generating features from the categorical values to convert them into rules for mushroom consumption might provide easy guide for foragers to follow.

As an example, by using the results of correlation it can be concluded that odor is a very important feature and after tabulating the results, it can be concluded that:

- If the mushroom smells foul, creosote, fishy, musty, pungent or spicy do not eat.
- If the mushroom smells almond, anise it is safe to eat
- If the mushroom has no smell it might still be poisonous, do not take the risk.

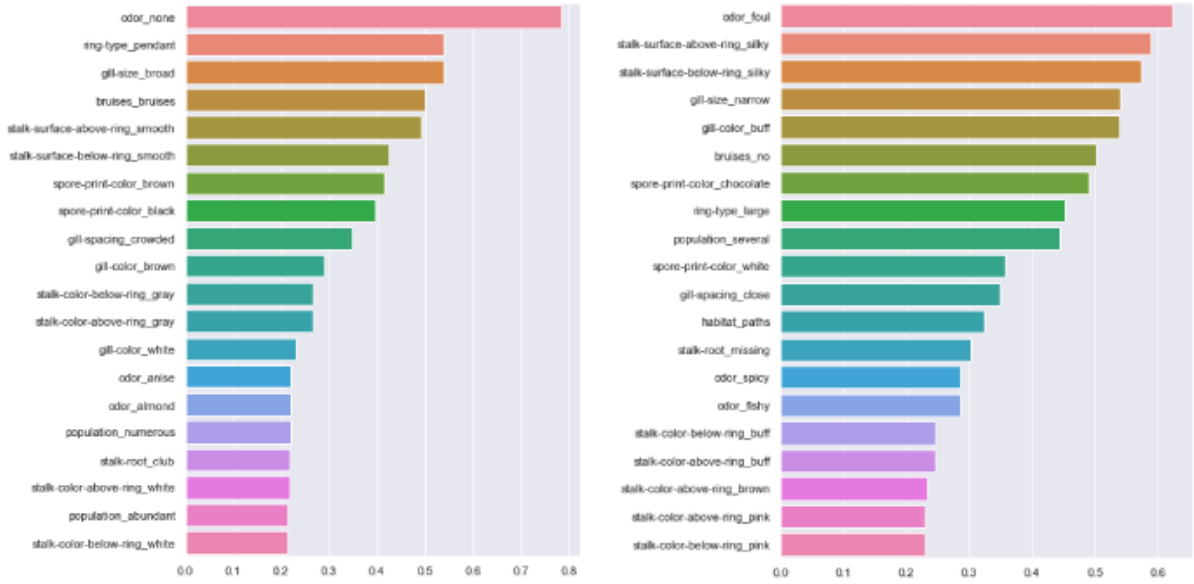

Figure 4: Top 10 positively correlated features with edible and poisonous mushroom respectively

The tabulated counts for odor of edible and poisonous mushroom is listed below:

class	odor	samples count
edible	almond	400
	anise	400
	creosote	0
	fishy	0
	foul	0
	musty	0
	none	3408
	pungent	0
	spicy	0
poisonous	almond	0
	anise	0
	creosote	192
	fishy	576
	foul	2160
	musty	36
	none	120
	pungent	256
	spicy	576

However, for relationships between categorical data, measure of entropy or nominal association can be construed to be correlation.

Cramér's V is a measure of association between two nominal variables. The metric ranges from 0 to 1, 0 indicating no relationship or association between two variables and 1 indicating strong association. It is expected that "odor" has a higher Cramér's V based on the previous result.

$$C = \sqrt{\frac{\frac{X^2}{n}}{min(c-1)(r-1)}}$$

where,

X = Chi-square statistic

c = number of columns

r = number of rows

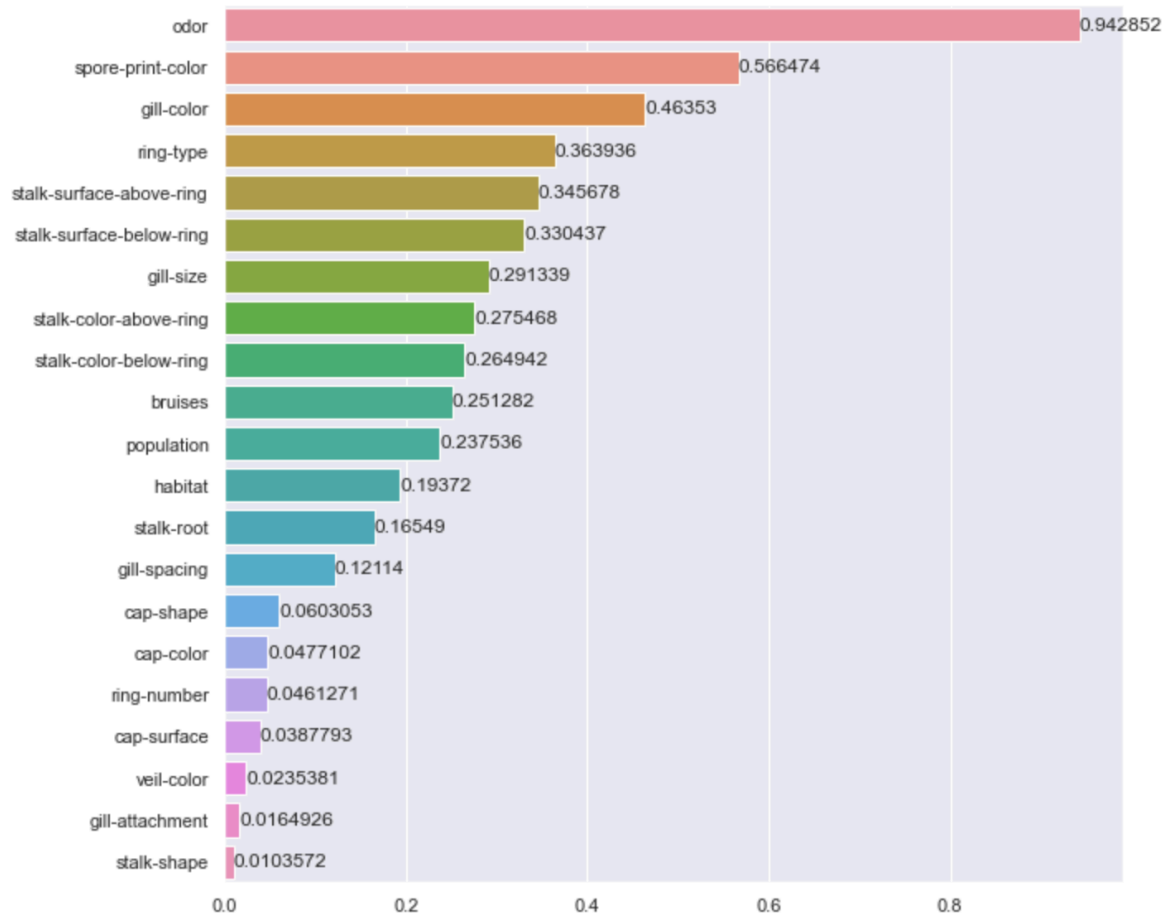

Figure 5: Cramér's V scores

As expected, "Odor" is the most prominent feature. The measure of association of edibility of mushrooms is highest on this feature. In section 4.2, the features with n-highest Cramér's V index is explored to build classification models.

4.1.2 Outlier Detection when all features are categorical

There are several tools that can be used for outlier detection when data is quantitative. Some of these include DBSCAN, Extreme value analysis, Z-Scores etc. However, for a single categorical feature, the concept of outlier detection seems a little obtuse.

Additionally, it has already been established that no two samples with exactly same feature set lead to different results.

A simple way to tackle this problem would be to utilize value counts, ie., if the count of a unique occurrence within a feature is less than or more than a certain set of computed thresholds the value could be treated as an outlier. However, this would not work (and should be avoided) in this case. For example, in the table of counts for odor of edible and poisonous mushroom above, the count of mushrooms with odor ”musty” is only 36. If this is to be treated as an outlier based on count, it has a huge cost because previous results show that all ”musty” odor of mushrooms are poisonous.

The description of the dataset mentions, ”Each species is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended. This latter class was combined with the poisonous one”.

The samples with unknown edibility and not recommended can be considered as outliers within the poisonous class in the dataset. The properties that distinguishes these sub-classes have been combined with the poisonous class. This implies, the prediction might be noisy. Unfortunately, the data dictionary for the dataset does not specify which samples come from unknown edibility and not recommended.

k-modes clustering can be utilized to check for clusters within the poisonous class to detect unknown edibility and not recommended. However, validation is an issue here due to aforementioned reasons. Since all the features are categorical, k-means (which uses distance norms) is not applicable here. k-modes is a variation of the k-means algorithm that uses modes (similarity) to assign clusters [4].

Elbow method can be used in determining the optimal number of clusters in the data set (here: poisonous class of the dataset). Here, based on the elbow diagram the cutoff point for the k-modes cluster is 2, as shown in Figure 6 below.

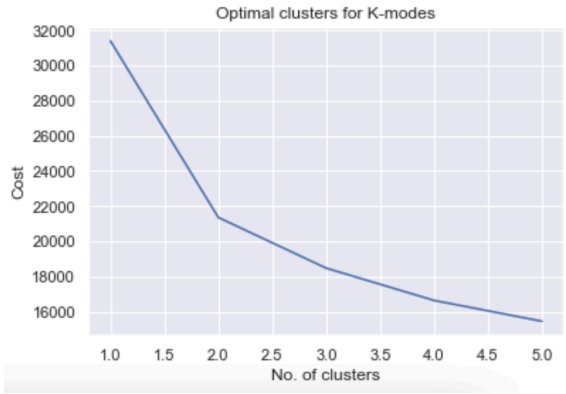

Figure 6: Elbow diagram for optimal clusters

The cluster centers are:

[[’scaly’ ’gray’ ’no’ ’foul’ ’free’ ’close’ ’broad’ ’pink’ ’enlarging’ ’bulbous’ ’silky’ ’silky’ ’brown’ ’brown’ ’white’ ’one’ ’large’ ’chocolate’ ’several’ ’grasses’] [’smooth’ ’brown’ ’no’ ’foul’ ’free’ ’close’ ’narrow’ ’buff’ ’tapering’ ’missing’ ’smooth’ ’smooth’ ’white’ ’white’ ’white’ ’one’ ’evanescent’ ’white’ ’several’ ’woods’]]

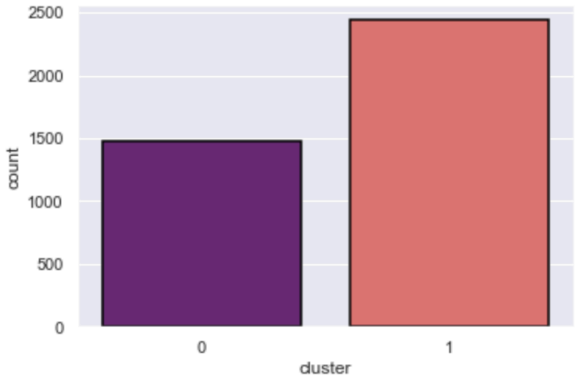

Figure 7: Cluster assignment

Three classes of unknown edibility, not recommended and poisonous, are clustered into two based on the elbow diagram, and the result assigns 1476 to the first cluster (0) and 2440 to the second cluster (1). In the dataset, all of these are categorized under poisonous. The two clusters make sense because the first cluster (0) might represent some samples from not recommended and unknown edibility that might be different from the truly poisonous sample . And, the other cluster (1) might be representative of the truly poisonous class.

However, the unavailability of true labels here does not allow testing this hypothesis, and the only thing that can be truly concluded is, that it is pretty challenging to define and detect outliers in an all categorical setting such as this dataset.

4.1.3 Dimensionality reduction for data visualization when all features are categorical

The visualization of the first two principal components based on **Principal Component Analysis** (PCA) is below. One hot encoding was applied to the categorical data (resulting in 95 features condensing to 2 principal components).

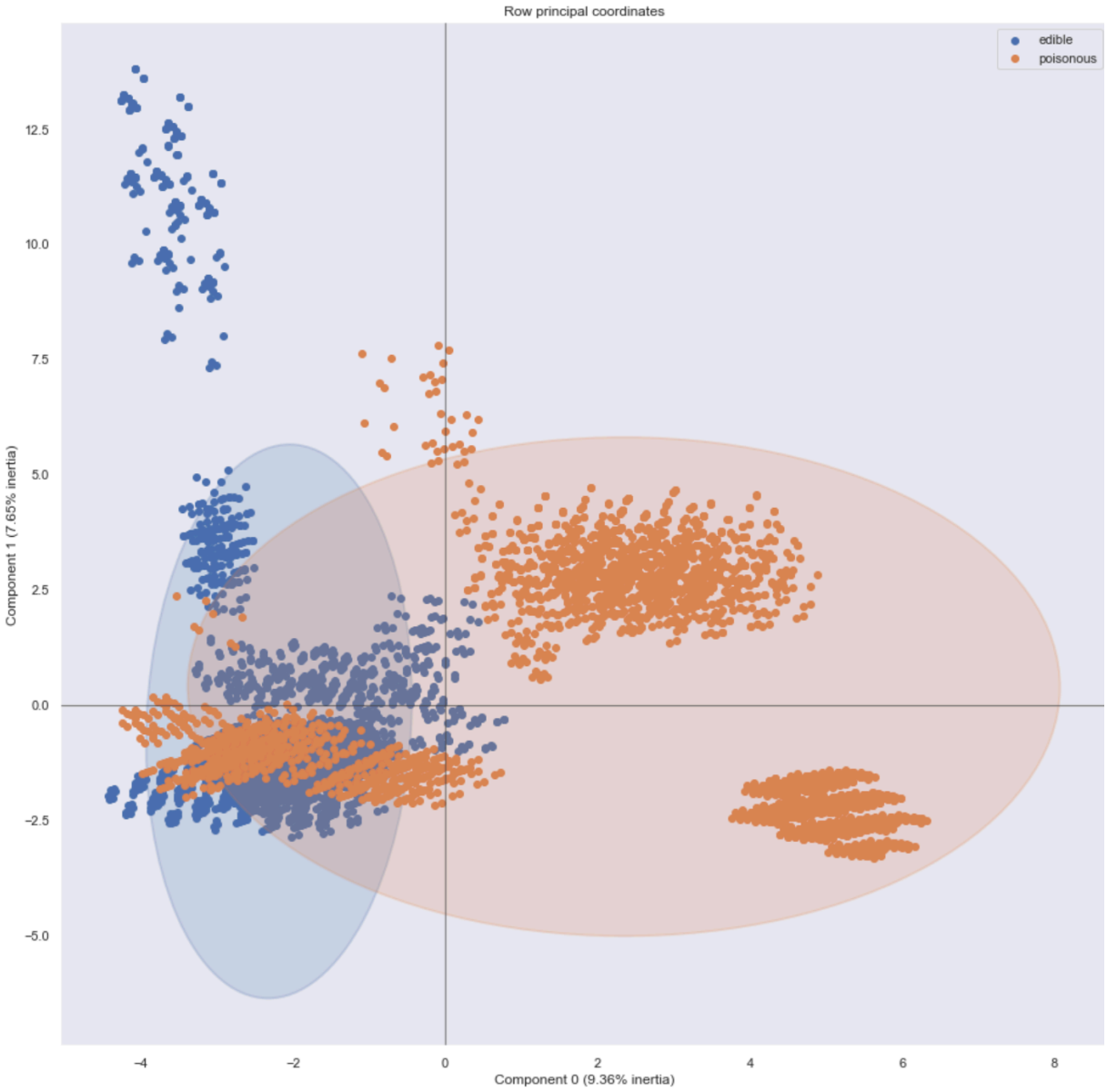

Figure 8: PCA for data visualization

The first principal component (Component 0) contributes to 9.36% of the inertia, and the second principal component (Component 1) contributes to 7.65% of the inertia. Although the clusters of similar mushrooms are distinguishable, there are plenty of overlaps here as well. Since PCA is based on minimization of variance, it does work well with categorical data (at least in this case), even when the data is one-hot encoded. For nominal categorical data, **Multiple Correspondence Analysis** is explored.

Muitiple Correspondence Analysis is an extension on **Correspondence Analysis** which is applicable when analyzing the relationships between two variables using a contingency table. **Contingency table** (or a crosstab) stores the frequency of combinations in the sample. For example, the contingency table and biplot of correspondence analysis for ”odor” and ”spore-print-color” is displayed below:

spore-print-color	black	brown	buff	chocolate	green	orange	purple	white	yellow
odor									
almond	176	200	0	0	0	0	24	0	0
anise	176	200	0	0	0	0	24	0	0
creosote	96	96	0	0	0	0	0	0	0
fishy	0	0	0	0	0	0	0	576	0
foul	0	0	0	1584	0	0	0	576	0
musty	0	0	0	0	0	0	0	36	0
none	1296	1344	48	48	72	48	0	624	48
pungent	128	128	0	0	0	0	0	0	0
spicy	0	0	0	0	0	0	0	576	0

Figure 9: Contingency table for ”odor” and ”spore-print-color”

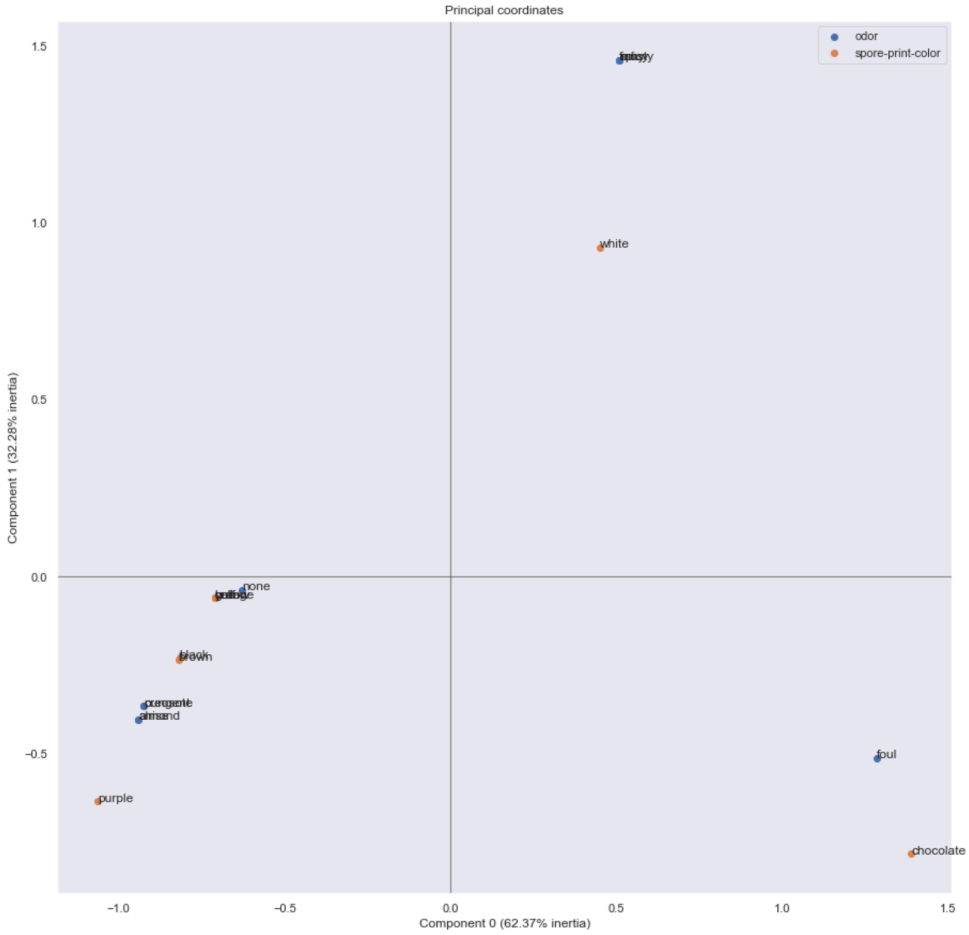

Figure 10: Correspondence Analysis for ”odor” and ”spore-print-color”

In the plot above, strong relationship between ”foul” odor and ”chocolate” spore-print-color is observed. Similarly, ”fishy” odor and ”white” spore-print-color are closely related. Multiple correspondence analysis does the same for multiple categories. MCA is applied by performing CA to the dummy variables. The result is displayed below:

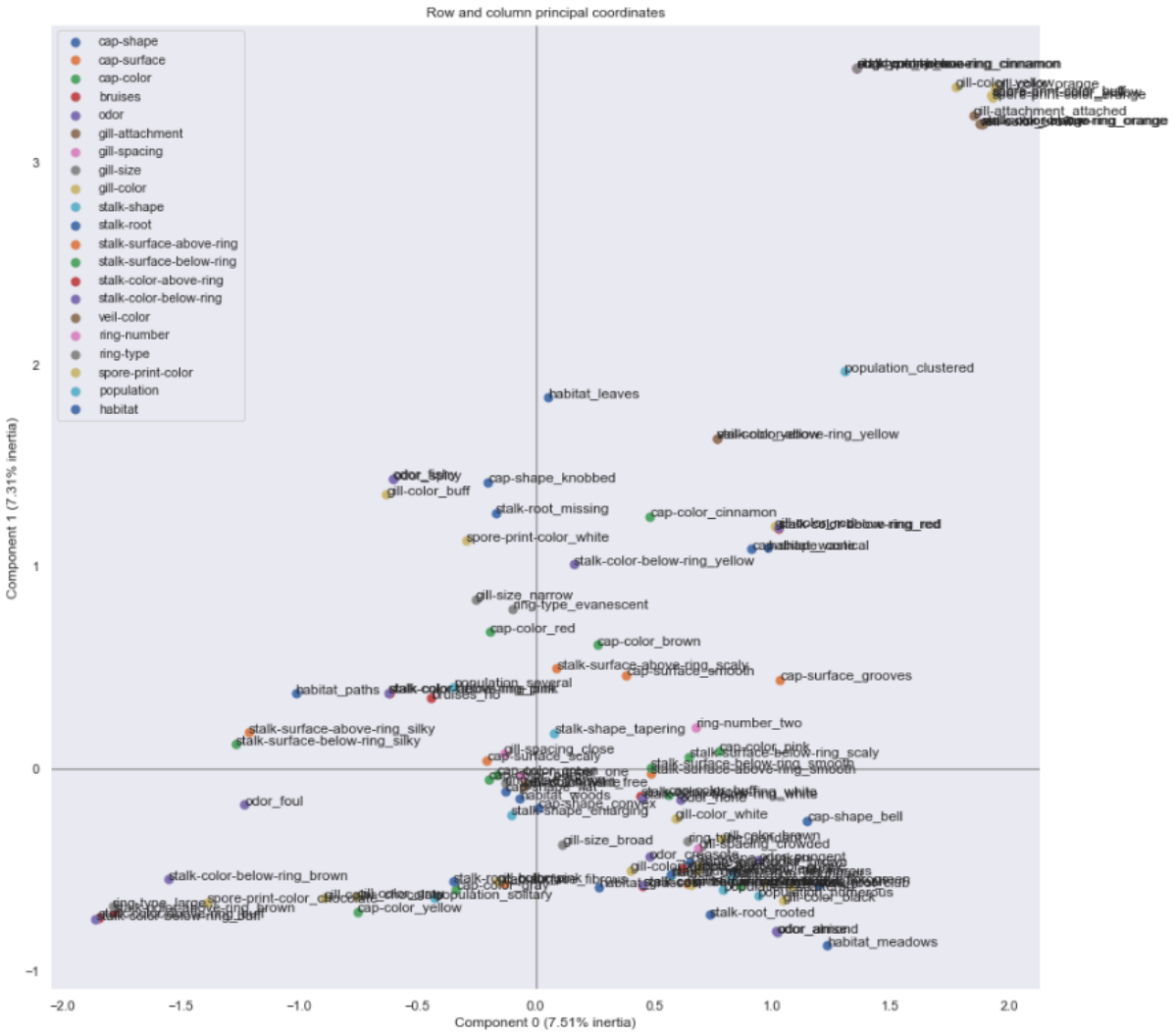

Figure 11: MCA for data visualization

Here the first component (Component 0) represents 7.51% of the inertia, and the second component (Component 1) represents 7.31% of the inertia. Here too, there are several relationships that can be seen. For example, odor ”almond” and habitat ”meadows” at the bottom right of the chart, Odor ”fishy” and gill-color ”buff” in top of the first quadrant etc. Unlike correspondence analysis, this method can be used for multiple categories.

It can also be concluded that several features in the dataset have strong relationship with just another feature.

T-distributed Stochastic Neighbor Embedding (t-SNE) is also applicable here. It is an effective technique for dimensionality reduction, and works well with nominal categorical data. Similar objects are clustered together and dissimilar objects are spread apart. This implies, when

plotted, clusters of poisonous and edible mushrooms should be visible if classification is possible via the dataset.

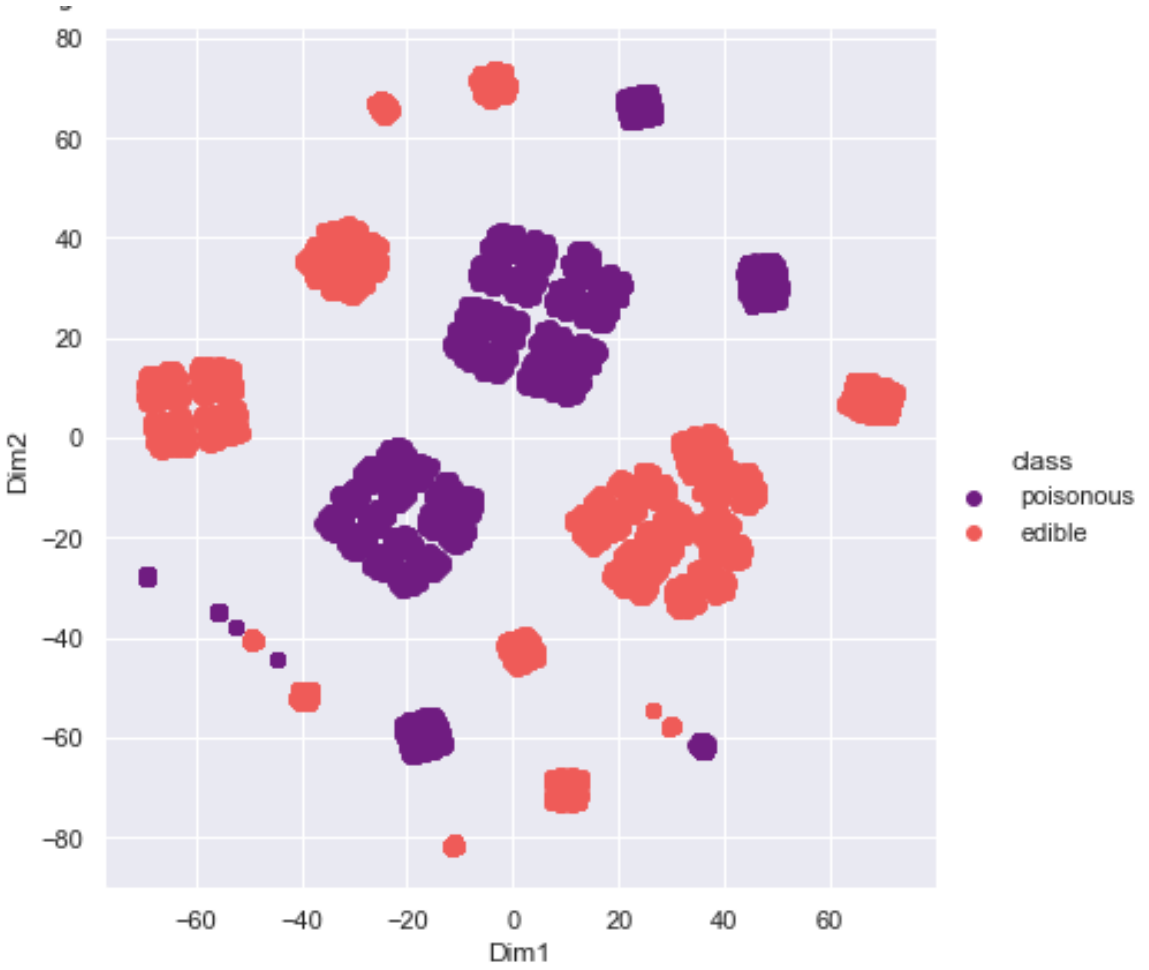

Figure 12: t-SNE for data visualization

t-SNE uses perplexity, which is related to the number of neighbors used in manifold learning algorithms [6]. The plot above was created with a perplexity value of 30. Perplexity is defined as $2^{H(p)}$ where, $H(p)$ is the entropy. Therefore,

$$P = 2^{-\Sigma_x p(x) log_2 p(x)}$$

The result from t-SNE show clear clusters of poisonous and edible mushrooms. Although there are plenty of clusters, the distinction is clear.

For dimensionality reduction in nominal categorical setting, Multiple Correspondence Analysis and T-distributed Stochastic Neighbor Embedding were more relevant compared to Principal Component Analysis.

4.2 Classification

The previous sections provided plenty of insight on patterns in the data. These are helpful in identifying relationships, and important features when classifying poisonous and edible mushrooms. However, based on question 1 of the project, since the goal is to find the best model, for the next section, all features are used to classify mushrooms.

To begin with, since the goal here are ”rules” for edibility (or identification of poisonous mushrooms), Decision Tree is explored.

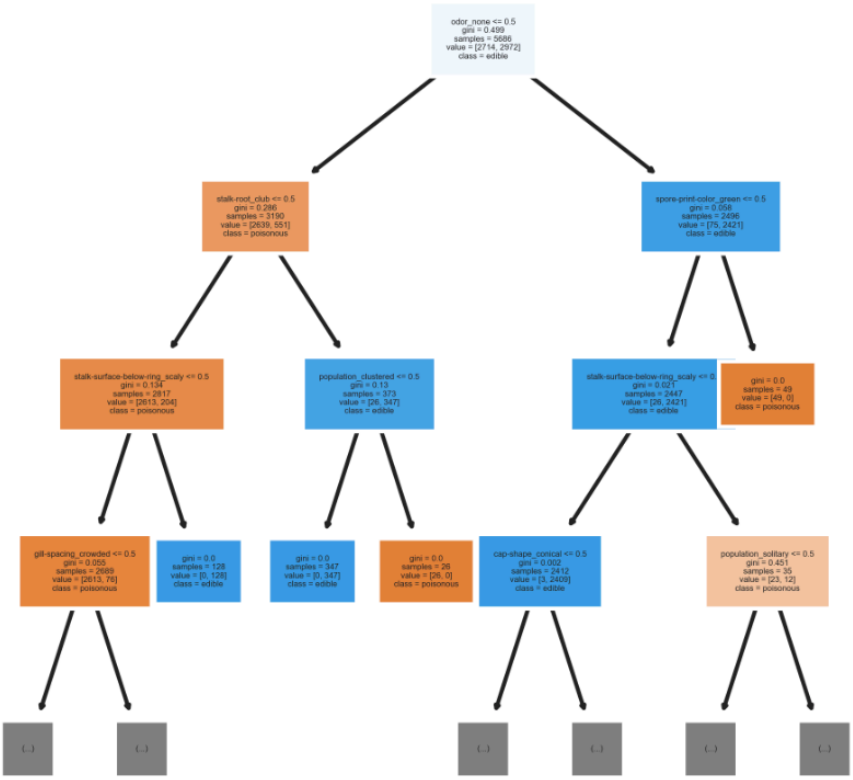

Figure 13: Decision Tree Plot

Using 70% data for training and 30% for testing, and 3 as the maximum depth, the accuracy of the Decision Tree model is 0.988. Similarly, at maximum depth of 7, the model produces an accuracy of 1. The decision tree is visualized above. Additionally, several classification algorithms (such as AdaBoost, Gaussian Naive Bayes, Logistic Regression) from scikit-learn implementation were used to classify the same split of dataset explained above and there results are listed on the table below. The primary validation metrics that are also tabulated are Accuracy, Precision, Recall, True Positive count, False Negative count, False Positive count and True Negative count. The positive class that is being predicted here is if the mushroom is edible.

	algorithm	accuracy	TP	FP	FN	TN	precision	recall
1	AdaBoostClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
2	BaggingClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
3	DecisionTreeClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
4	ExtraTreeClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
5	GaussianNB()	0.993847	1200	2	13	1223	0.998336	0.989283
6	GradientBoostingClassifier()	0.999180	1200	2	0	1236	0.998336	1.000000
7	KNeighborsClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
8	LogisticRegression()	0.997949	1197	5	0	1236	0.995840	1.000000
9	Perceptron()	1.000000	1202	0	0	1236	1.000000	1.000000
10	RandomForestClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
11	SVC(probability=True)	0.998359	1198	4	0	1236	0.996672	1.000000

The hyper-parameters used in the algorithm above is listed below. Here, the defaults of scikit-learn worked well in all cases.

algorithm	parameters
AdaBoostClassifier()	{'algorithm': 'SAMME.R', 'base_estimator': None, 'learning_rate': 1.0, 'n_estimators': 50, 'random_state': None}
BaggingClassifier()	{'base_estimator': None, 'bootstrap': True, 'bootstrap_features': False, 'max_features': 1.0, 'max_samples': 1.0, 'n_estimators': 10, 'n_jobs': None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
DecisionTreeClassifier()	{'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'random_state': None, 'splitter': 'best'}
ExtraTreeClassifier()	{'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'random_state': None, 'splitter': 'random'}
GaussianNB()	{'priors': None, 'var_smoothing': 1e-09}
GradientBoostingClassifier()	{'ccp_alpha': 0.0, 'criterion': 'friedman_mse', 'init': None, 'learning_rate': 0.1, 'loss': 'deviance', 'max_depth': 3, 'max_features': None, 'max_leaf_nodes': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 100, 'n_iter_no_change': None, 'random_state': None, 'subsample': 1.0, 'tol': 0.0001, 'validation_fraction': 0.1, 'verbose': 0, 'warm_start': False}
KNeighborsClassifier()	{'algorithm': 'auto', 'leaf_size': 30, 'metric': 'minkowski', 'metric_params': None, 'n_jobs': None, 'n_neighbors': 5, 'p': 2, 'weights': 'uniform'}
LogisticRegression()	{'C': 1.0, 'class_weight': None, 'dual': False, 'fit_intercept': True, 'intercept_scaling': 1, 'l1_ratio': None, 'max_iter': 100, 'multi_class': 'auto', 'n_jobs': None, 'penalty': 'l2', 'random_state': None, 'solver': 'lbfgs', 'tol': 0.0001, 'verbose': 0, 'warm_start': False}
Perceptron()	{'alpha': 0.0001, 'class_weight': None, 'early_stopping': False, 'eta0': 1.0, 'fit_intercept': True, 'l1_ratio': 0.15, 'max_iter': 1000, 'n_iter_no_change': 5, 'n_jobs': None, 'penalty': None, 'random_state': 0, 'shuffle': True, 'tol': 0.001, 'validation_fraction': 0.1, 'verbose': 0, 'warm_start': False}
RandomForestClassifier()	{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto', 'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, 'n_estimators': 100, 'n_jobs': None, 'oob_score': False, 'random_state': None, 'verbose': 0, 'warm_start': False}
SVC(probability=True)	{'C': 1.0, 'break_ties': False, 'cache_size': 200, 'class_weight': None, 'coef0': 0.0, 'decision_function_shape': 'ovr', 'degree': 3, 'gamma': 'scale', 'kernel': 'rbf', 'max_iter': -1, 'probability': True, 'random_state': None, 'shrinking': True, 'tol': 0.001, 'verbose': False}

Figure 14: Parameters used in Classification models above

4.3 Feature Selection

Although there are different ways of feature selection, the project relies on Cramér's V which was explored in section 4.1.1 of the project. The result from Cramér's V was:

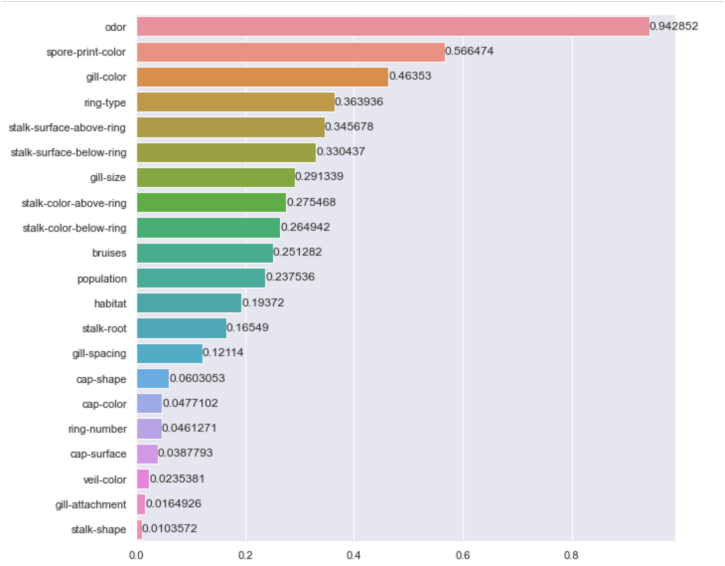

Figure 15: Cramér's V scores

If only the first two features (odor and spore-print-color) are used to build the models, listed below are the results. It is interesting that all the results have the same accuracy (of 0.99). Additionally, poisonous mushroom identified as edible goes against the objective of the project. Also, since one hot encoding is applied, the model is not really only build using two models. The parameters used in all these models are also the scikit-learn defaults as shown in Figure 14.

	algorithm	**accuracy**	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**
1	AdaBoostClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
2	BaggingClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
3	DecisionTreeClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
4	ExtraTreeClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
5	GaussianNB()	0.990976	1180	22	0	1236	0.981697	1.0
6	GradientBoostingClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
7	KNeighborsClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
8	LogisticRegression()	0.990976	1180	22	0	1236	0.981697	1.0
9	Perceptron()	0.990976	1180	22	0	1236	0.981697	1.0
10	RandomForestClassifier()	0.990976	1180	22	0	1236	0.981697	1.0
11	SVC(probability=True)	0.990976	1180	22	0	1236	0.981697	1.0

Just for the purposes of illustration, if the last two features from the Cramér's V scores (gill-attachment and stalk-shape) is used to create a classification model, here are the scores.

The accuracy scores of approximately 50 percent is explained by the number of classes that we are trying to predict and these features do not provide us any significant information (compared to odor and spore-print color that was explored above.

	algorithm	**accuracy**	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**
1	AdaBoostClassifier()	0.568499	574	628	424	812	0.477537	0.575150
2	BaggingClassifier()	0.568499	574	628	424	812	0.477537	0.575150
3	DecisionTreeClassifier()	0.568499	574	628	424	812	0.477537	0.575150
4	ExtraTreeClassifier()	0.568499	574	628	424	812	0.477537	0.575150
5	GaussianNB()	0.515587	1197	5	1176	60	0.995840	0.504425
6	GradientBoostingClassifier()	0.568499	574	628	424	812	0.477537	0.575150
7	KNeighborsClassifier()	0.568499	574	628	424	812	0.477537	0.575150
8	LogisticRegression()	0.568499	574	628	424	812	0.477537	0.575150
9	Perceptron()	0.568499	574	628	424	812	0.477537	0.575150
10	RandomForestClassifier()	0.568499	574	628	424	812	0.477537	0.575150
11	SVC(probability=True)	0.568499	574	628	424	812	0.477537	0.575150

4.4 Minimizing False Positives on edibility

The False Positive here refer to poisonous mushroom identified as edible, and False Negatives refer to edible mushroom identified as poisonous. The cost of False Positive is very high. Unfortunately, all the well performing models explored above with fewer features (section 4.3) seem to support False Negatives instead of False Positives. In other words, we want our model to have higher precision compared to recall.

This section explores if adding few more features based on the Cramer's V score will allow this. Here is the score if 'odor', 'spore-print-color', and 'gill-color' is used to create the model.

	algorithm	**accuracy**	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**
1	AdaBoostClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
2	BaggingClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
3	DecisionTreeClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
4	ExtraTreeClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
5	GaussianNB()	0.990156	1180	22	2	1234	0.981697	0.998308
6	GradientBoostingClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
7	KNeighborsClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
8	LogisticRegression()	0.990976	1180	22	0	1236	0.981697	1.000000
9	Perceptron()	0.991386	1181	21	0	1236	0.982529	1.000000
10	RandomForestClassifier()	0.991386	1181	21	0	1236	0.982529	1.000000
11	SVC(probability=True)	0.991386	1181	21	0	1236	0.982529	1.000000

While this is marginally better, adding one more feature produces the following result. The features used now are 'odor', 'spore-print-color', 'gill-color', 'ring-type'.

	algorithm	**accuracy**	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**
1	AdaBoostClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
2	BaggingClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
3	DecisionTreeClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
4	ExtraTreeClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
5	GaussianNB()	0.990976	1180	22	0	1236	0.981697	1.0
6	GradientBoostingClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
7	KNeighborsClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
8	LogisticRegression()	0.990976	1180	22	0	1236	0.981697	1.0
9	Perceptron()	0.991386	1181	21	0	1236	0.982529	1.0
10	RandomForestClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
11	SVC(probability=True)	0.991386	1181	21	0	1236	0.982529	1.0

This did not improve the results as much. Adding additional feature produces the following result. The features used now are 'odor', 'spore-print-color', 'gill-color', 'ring-type', 'stalk-surface-above-ring'.

	algorithm	accuracy	TP	FP	FN	TN	precision	recall
1	AdaBoostClassifier()	0.991386	1181	21	0	1236	0.982529	1.0
2	BaggingClassifier()	0.997949	1197	5	0	1236	0.995840	1.0
3	DecisionTreeClassifier()	0.997949	1197	5	0	1236	0.995840	1.0
4	ExtraTreeClassifier()	0.997949	1197	5	0	1236	0.995840	1.0
5	GaussianNB()	0.990976	1180	22	0	1236	0.981697	1.0
6	GradientBoostingClassifier()	0.992207	1183	19	0	1236	0.984193	1.0
7	KNeighborsClassifier()	0.997129	1195	7	0	1236	0.994176	1.0
8	LogisticRegression()	0.991386	1181	21	0	1236	0.982529	1.0
9	Perceptron()	0.997949	1197	5	0	1236	0.995840	1.0
10	RandomForestClassifier()	0.997949	1197	5	0	1236	0.995840	1.0
11	SVC(probability=True)	0.997949	1197	5	0	1236	0.995840	1.0

This combination of features significantly dropped the True Positives. After using 'odor', 'spore-print-color', 'gill-color', 'ring-type', 'stalk-surface-above-ring', 'stalk-surface-below-ring', 'gill-size', finally the True Positive count dropped to 0 and the accuracy is 100%.

	algorithm	accuracy	TP	FP	FN	TN	precision	recall
1	AdaBoostClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
2	BaggingClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
3	DecisionTreeClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
4	ExtraTreeClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
5	GaussianNB()	0.990976	1180	22	0	1236	0.981697	1.000000
6	GradientBoostingClassifier()	0.997949	1197	5	0	1236	0.995840	1.000000
7	KNeighborsClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
8	LogisticRegression()	0.997949	1197	5	0	1236	0.995840	1.000000
9	Perceptron()	0.998359	1202	0	4	1232	1.000000	0.996683
10	RandomForestClassifier()	1.000000	1202	0	0	1236	1.000000	1.000000
11	SVC(probability=True)	0.997949	1197	5	0	1236	0.995840	1.000000

5 Conclusion

In conclusion, using 7 features, out of the original 22, it is possible to create a model that provides 100% accuracy. The 7 features were recognized using Cramér's V. The features are 'odor', 'spore-print-color', 'gill-color', 'ring-type', 'stalk-surface-above-ring', 'stalk-surface-below-ring', 'gill-size'.

It is easier to train foragers to focus on these 7 features instead of the 22 features. This project looked into finding the best model for the detection of edibility of mushroom, best model with the least number of features, and the best model with the least number of false positives. Additionally, from a methodology point of view, this project investigated the meaning of correlation in all categorical setting, the idea of outliers in all categorical setting and the idea of dimensionality reduction in all categorical setting. The project was successful in answering all the questions.

However, at present, there are more than 10000 identified species of mushrooms[11]. The dataset used here only represents 23 species of gilled mushrooms in the Agaricus and Lepiota genera. Therefore, the findings should not be used in the real world. Some of the conclusions such as "If the mushroom smells almond, anise it is safe to eat", might not be true for other species of mushrooms that were not explored here.

References

[1] H. Akoglu. User's guide to correlation coefficients. *Turkish journal of emergency medicine*, 18(3):91–93, 2018.

[2] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[3] P. Bholowalia and A. Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[4] N. J. de Vos. kmodes categorical clustering library. `https://github.com/nicodv/kmodes`, 2015–2021.

[5] D. Dua and C. Graff. UCI machine learning repository, 2017.

[6] R. Frigg and C. Werndl. Entropy-a guide for the perplexed. 2011.

[7] M. Halford. Maxhalford/prince: Python factor analysis library (pca, ca, mca, mfa, famd).

[8] P. T. o. India. 6 die after consuming wild mushroom in nepal, Jun 2019.

[9] C. D. G. Kaufman. Every mushroom is edible - but some only once, Dec 2016.

[10] B. Keough. Here's what you'll need to start foraging mushrooms, Jul 2020.

[11] M. Mushroom, Jimfirjf, and Jenny. Different types of mushrooms and their uses, Feb 2022.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[13] A. Ramírez-Terrazo, E. Adriana Montoya, R. Garibay-Orijel, J. Caballero-Nieto, A. Kong-Luz, and C. Méndez-Espinoza. Breaking the paradigms of residual categories and neglectable importance of non-used resources: The "vital" traditional knowledge of non-edible mushrooms and their substantive cultural significance - journal of ethnobiology and ethnomedicine, Apr 2021.

[14] I. Svanberg and H. Lindh. Mushroom hunting and consumption in twenty-first century post-industrial sweden - journal of ethnobiology and ethnomedicine, Aug 2019.