Preliminary Report - Edibility of Mushroom Species

Ishaan Saxena isaxena@purdue.edu
Nikita Rajaneesh nrajanee@purdue.edu
Swaraj Bhaduri sbhadur@purdue.edu
Utkarsh Jain jain192@purdue.edu

November 16, 2018

1 Introduction to the Problem

1.1 Definition of the Problem

Given a dataset $\mathcal{D}$ with $n = 8124$ samples where each sample represents a mushroom with features being the observations about the characterestics of the mushrooms such as odor, color, etc., we aim to test and compare various supervised learning models for the problem of classifying each sample into either poisonous or edible. Further, we will optimize the Hyperparameters of the model which initially performs the best on the dataset.[1]

1.2 Data Description

We are given $\mathcal{D}$ with $n = 8124$ samples wherein each sample has the following 22 features (excluding the class label).

1. cap-shape
2. cap-surface
3. cap-color
4. bruises
5. odor
6. gill-attachment
7. gill-color
8. stalk-root
9. stalk-surface-above-ring
10. stalk-surface-below-ring
11. stalk-color-above-ring
12. stalk-color-below-ring
13. veil-type
14. veil-color
15. ring-type
16. spore-print-color
17. habitat
18. gill-spacing
19. gill-size
20. stalk-shape
21. ring-number
22. population

These features have been further enumerated in Appendix A.

1.3 Encoding the Data

Note that all the features in our dataset are categorical variables. As a result, to proceed with evaluation of model performance, we must first encode these variables into numerical/binary values.

We need to deal with two kinds of categorical variables when encoding the features into numerical data. These are ordinal categorical variables and nominal categorical variables.[2] We will use different techniques to

[1] This dataset can be found at https://www.kaggle.com/uciml/mushroom-classification.
[2] Information on which features are which kind of categorical variables can be found in Appendix A.

encode both of these kinds of categorical variables as they inherently represent different kinds of categorical data.

To encode nominal categorical variables, we will use one-hot binary features. For instance, if a feature f from a feature set $\mathcal{F}$ has k different categorical values, we can create k different binary features for each feature f of this kind. This is done because the values of each such feature do not hold any ordinal information, in that there should not be different weights for having a specific value of a specific feature.

The ordinal categorical variables, on the other hand, have been encoded as numerical labels, as these informations contain valueable information about the 'scale' of a certain feature. For instance, if a feature f from a feature set $\mathcal{F}$ has k different features, it would be changed into numeric values $i \in 0, 1, ..., k - 1$.

The following code segment is used to encode the data.[3]

```
def encode(df):
    # Encode Ordinal Variables
    ordinal_columns = ['gill-spacing', 'gill-size',
            'stalk-shape', 'ring-number', 'population', 'class']
    columns = ordinal_columns[:]

    for column in columns:
        df[column] = df[column].astype('category')

        columns = df.select_dtypes(['category']).columns
        df[columns] = df[columns].apply(lambda x: x.cat.codes)

    # Encoding Nominal Variables
    columns = ordinal_columns[:]

    for column in df:
        if column not in columns:
            dummies = pd.get_dummies(df.pop(column))
            column_names = [column + "_" + x for x in dummies.columns]
            dummies.columns = column_names
            df = df.join(dummies)

    return df
```

[3]This can be found in the data.py file.

Appendix A: Data Description

Classes: edible=e, poisonous=p (y-values)

Size of dataset (before encoding): $(n = 8124, d = 22)$
Size of dataset (after encoding): $(n = 8124, d = 107)$
Attribute Information and Encoding:

1. Nominal Categorical Variables:
   These variables will be encoded as binary one-hot features. As a result, each feature in this category would be replace by the k features in the encoded dataset if the feature has k possible values. These featues include:

   i. **cap-shape**: bell=b, conical=c, convex=x, flat=f, knobbed=k, sunken=s
   ii. **cap-surface**: fibrous=f, grooves=g, scaly=y, smooth=s
   iii. **cap-color**: brown=n, buff=b, cinnamon=c, gray=g, green=r, pink=p, purple=u, red=e, white=w, yellow=y
   iv. **bruises**: bruises=t, no=f
   v. **odor**: almond=a, anise=l, creosote=c, fishy=y, foul=f, musty=m, none=n, pungent=p, spicy=s
   vi. **gill-attachment**: attached=a, descending=d, free=f, notched=n
   vii. **gill-color**: black=k, brown=n, buff=b, chocolate=h, gray=g, green=r, orange=o, pink=p, purple=u, red=e, white=w, yellow=y
   viii. **stalk-root**: bulbous=b, club=c, cup=u, equal=e, rhizomorphs=z, rooted=r, missing=?
   ix. **stalk-surface-above-ring**: fibrous=f, scaly=y, silky=k, smooth=s
   x. **stalk-surface-below-ring**: fibrous=f, scaly=y, silky=k, smooth=s
   xi. **stalk-color-above-ring**: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
   xii. **stalk-color-below-ring**: brown=n, buff=b, cinnamon=c, gray=g, orange=o, pink=p, red=e, white=w, yellow=y
   xiii. **veil-type**: partial=p, universal=u
   xiv. **veil-color**: brown=n, orange=o, white=w, yellow=y
   xv. **ring-type**: cobwebby=c, evanescent=e, flaring=f, large=l, none=n, pendant=p, sheathing=s, zone=z
   xvi. **spore-print-color**: black=k, brown=n, buff=b, chocolate=h, green=r, orange=o, purple=u, white=w, yellow=y
   xvii. **habitat**: grasses=g, leaves=l, meadows=m, paths=p, urban=u, waste=w, woods=d

2. Ordinal Categorical Variables:
   These variables will be encoded in place by encoding labels, as the data here has ordinal meaning to it. These variables include:

   i. **gill-spacing**: close=c→0, crowded=w→1, distant=d→2
   ii. **gill-size**: broad=b→0, narrow=n→1
   iii. **stalk-shape**: enlarging=e→0, tapering=t→1
   iv. **ring-number**: none=n→0, one=o→1, two=t→2
   v. **population**: abundant=a→0, clustered=c→1, numerous=n→2, scattered=s→3, several=v→4, solitary=y→5