Predicting Edibility of Mushrooms

Cheng Tian Cui, Peidong He, Joosun Hwang

November 21, 2023

1 Abstract

This study explores the application of machine learning algorithms to classify mushrooms as edible or poisonous using a dataset with numerical and categorical variables. After comparative analysis, Random Forest, K-NN, and SVM were chosen for their low false positive rates and accuracy. Hyperparameter tuning identified the KNN algorithm with K=1 as the most accurate, due to its absence of false positives high classification accuracy. Challenges such as the ineffective feature selection via mutual information and limited re-evaluation methods were encountered. Improvements and changes in methodology and technique for future studies will be discussed.

2 The Dataset

The dataset used in this project to train our machine learning models (MLM) was the *Secondary Mushroom Dataset*[1] procured from the UCI Machine Learning Repository. The dataset contains 61068 hypothetical mushroom species based on 173 real world species. Each entry is labeled as either edible or poisonous with any ambiguously edible mushrooms being labeled as poisonous. It should be emphasized that the mushrooms present in the dataset are randomly generated from attributes of species found in the dataset *Mushroom*[2] also found on the UCI Machine Learning Repository. As such, it is possible for entries in the dataset used to be non-present in the real world.

3 Exploratory Data Analysis

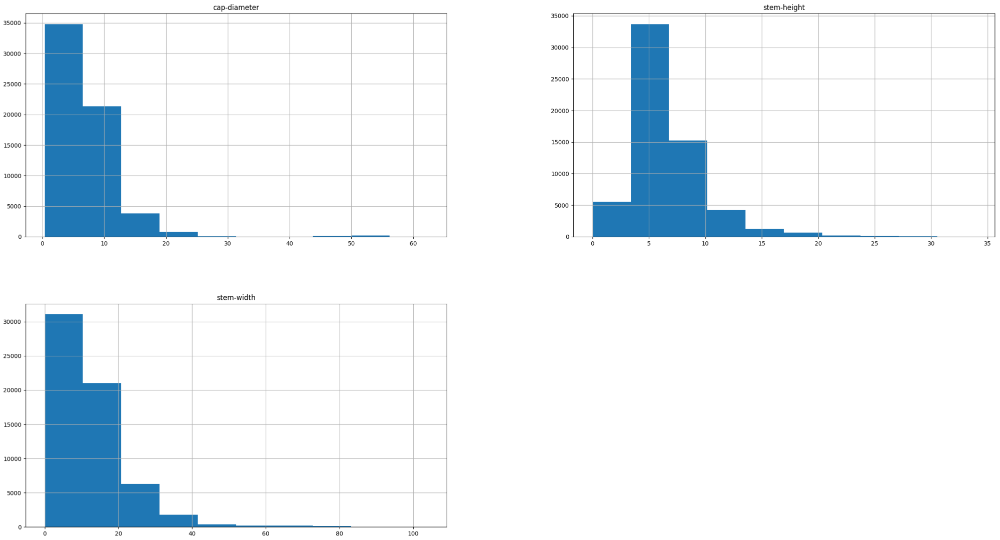


Figure 1: Histogram of numerical values

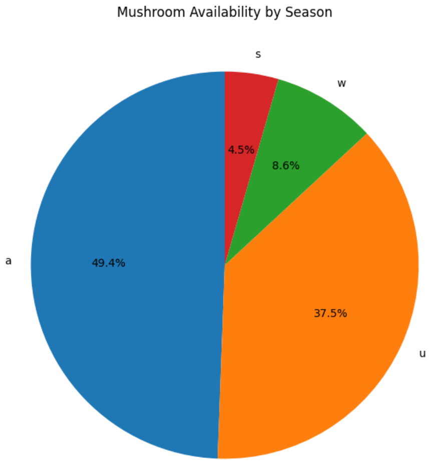


Figure 2: Proportion of entries by season they appear in

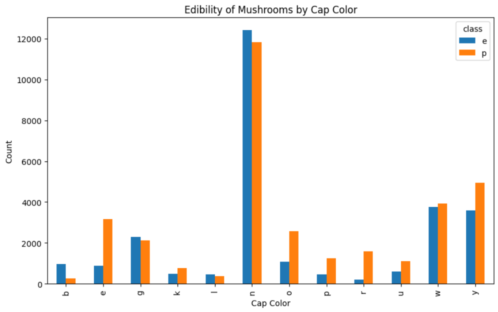
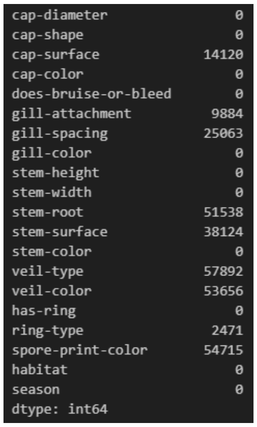


Figure 3: Count of edible and poisonous entries of each cap color

4 Data Cleaning and Pre-processing

The biggest issue encountered during this phase was how sparsely populated some features were.



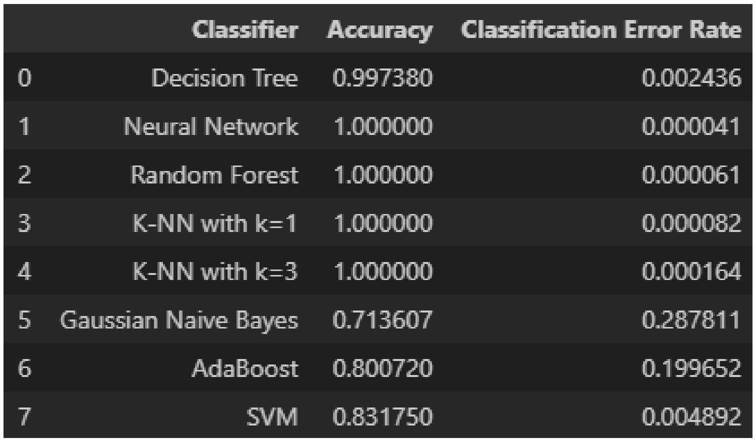
```
cap-diameter              0
cap-shape                 0
cap-surface           14120
cap-color                 0
does-bruise-or-bleed      0
gill-attachment        9884
gill-spacing          25063
gill-color                0
stem-height               0
stem-width                0
stem-root             51538
stem-surface          38124
stem-color                0
veil-type             57892
veil-color            53656
has-ring                  0
ring-type              2471
spore-print-color     54715
habitat                   0
season                    0
dtype: int64
```

Figure 4: Tally of missing values in features

Due to the sample size relative to the population for features like `veil-type`, `veil-color`, and `spore-print-color`, we could not reasonably use imputation to fill missing values. Thus for any feature that was filled to less than 50% pf the population, we decided to omit when training the models. Of the remaining features, numerical missing values were filled with the mean while categorical missing values were filled with the most frequent non-null value.

5 Training, Evaluation, Comparison of Models

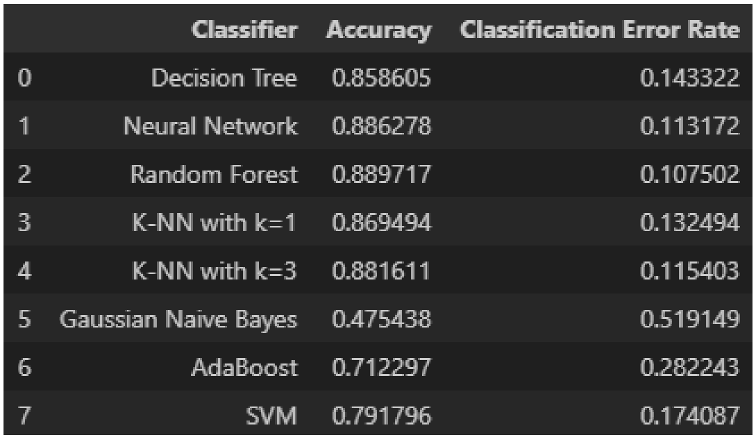
The dataset was split into 60% for training, 20% for testing, and 20% for validating the models. The algorithms we experimented with were Decision Tree, Neural Network, Random Forest, K-Nearest Neighbors, Gaussian Naive Bayes, AdaBoost, and Support Vector Machine classifications. Our initial models had extremely high accuracy, which made us skeptical. So we considered the possibility that the models are overfitted.



	Classifier	Accuracy	Classification Error Rate
0	Decision Tree	0.997380	0.002436
1	Neural Network	1.000000	0.000041
2	Random Forest	1.000000	0.000061
3	K-NN with k=1	1.000000	0.000082
4	K-NN with k=3	1.000000	0.000164
5	Gaussian Naive Bayes	0.713607	0.287811
6	AdaBoost	0.800720	0.199652
7	SVM	0.831750	0.004892

Figure 5: Accuracy of initial models

We then used the mutual information score to select the 5 most important features and drop the rest to obtain a reduced dataset. We then trained the models on this reduced dataset in order to compare with our initial models.



	Classifier	Accuracy	Classification Error Rate
0	Decision Tree	0.858605	0.143322
1	Neural Network	0.886278	0.113172
2	Random Forest	0.889717	0.107502
3	K-NN with k=1	0.869494	0.132494
4	K-NN with k=3	0.881611	0.115403
5	Gaussian Naive Bayes	0.475438	0.519149
6	AdaBoost	0.712297	0.282243
7	SVM	0.791796	0.174087

Figure 6: Accuracy of models trained on trimmed data

The accuracy of the models decreased when the dataset was trimmed so we concluded that the unmodified dataset was the most appropriate for training.

To evaluate which model was best, we followed the following train of thought. Since we are predicting the edibility of mushrooms, it would be the most dangerous if a poisonous mushroom was classified as edible. Thus we rated the models based on the false positive rate for edibility, then by accuracy.

6 Best Algorithm

We selected the K-NN with k=1 model as the best overall. It had the least false positives at 0 and high overall classification accuracy.

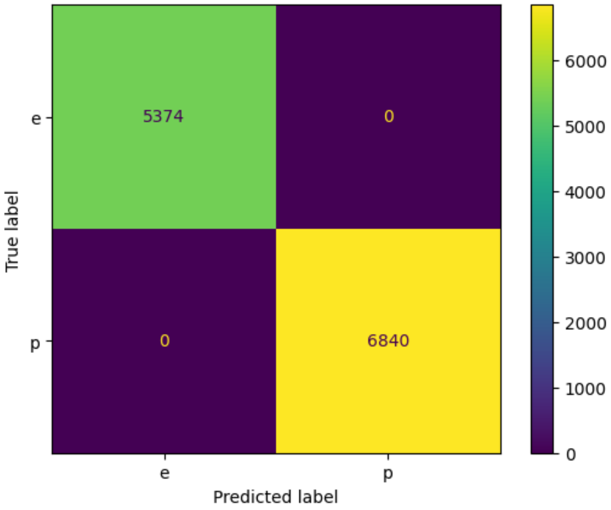


Figure 7: Confusion matrix of K-NN model with k=1

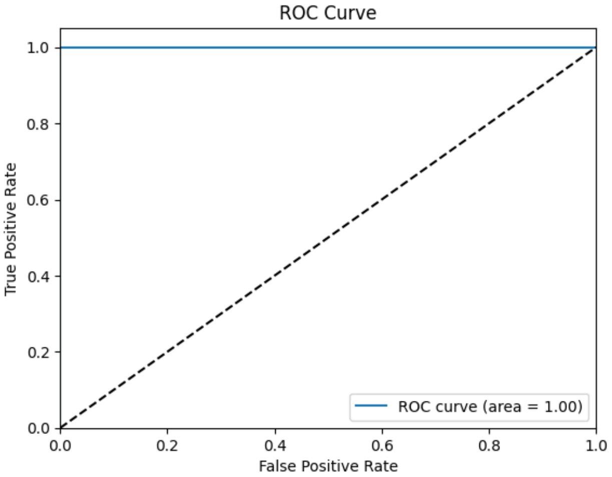


Figure 8: ROC curve of K-NN model with k=1

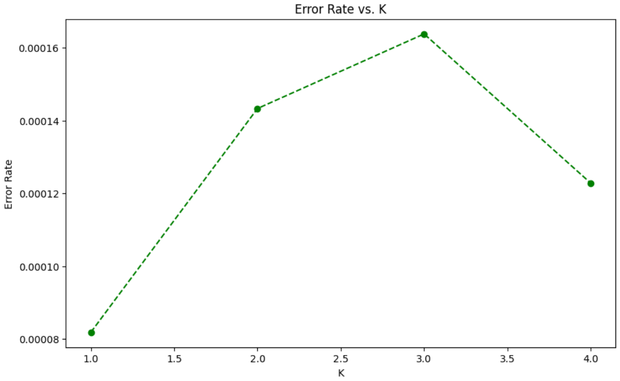


Figure 9: Classification error rate versus value of k

7 Challenges

One notable issue was the possible loss of critical information as a result of removing features with high frequencies of missing values, which could have mistakenly removed key predictive signals from our models. Furthermore, our attempts at feature selection using mutual information scores did not produce the desired gains; in fact, they reduced model quality and raised error rates, indicating that this strategy was unsuitable for the intricate relationships in our data. Time constraints restricted our ability to iterate and tune the models, which is critical for getting optimal performance.

8 Improvements and Suggestions

The mutual information score method of feature selection greatly reduced model accuracy, contrary to expectations. This suggests that it may not be an effective method of feature selection for this dataset. Additionally, it raises concerns that the sparsely populated features discarded during pre-processing could have also been important features. To enhance future performance, we suggest the utilization of more sophisticated methodologies, such as Extremely Randomized Tree classifiers, for feature selection. This approach would help mitigate the risk of discarding valuable and informative features. Furthermore, there is a justified need to investigate Neural Networks because of their increasing prevalence and proven efficacy in attaining notable precision and minimal rates of classification errors.

During this study, the Neural Network models were only given a cursory examination. However, due to increasing prevalence and proven efficiency, it may be of interest to explore more deeply in its application to this dataset.

9 Appendix 1: Sourced Code

- `https://eclass.yorku.ca/mod/resource/view.php?id=2609179`

10 Appendix 2: Links

1. **Dataset**: `https://archive.ics.uci.edu/dataset/848/secondary+mushroom+dataset`
2. **Precursor Dataset**: `https://archive.ics.uci.edu/dataset/73/mushroom`

3. **Python Notebook**: `link/to/python/notebook`
4. **Video Presentation**: `https://youtu.be/xEf7R0F7Fa4?si=g6GXcr-vkdRV35B6`