CS M148 Project 3 Report

Wonhee Lee 905284414

1. Introduction

The objective of this report is to build a model that can accurately classify whether a mushroom is poisonous or not based on certain physical characteristics. The dataset used for this project contained information on over 50,000 mushrooms and included features such as cap diameter, stem height, stem width, stalk-to-cap ratio, and stem volume. Three machine learning algorithms were implemented: Decision Tree classification, Support Vector Classification (SVC), and Random Forest classification. PCA was tested for dimensionality reduction, but it did not improve the data so much. The models were evaluated using accuracy, precision, recall, and F1-score metrics. The best performing model was the Decision Tree classification model, with an accuracy of 0.65 and an F1-score of 0.64. However, all models showed promising results, demonstrating that the physical characteristics of a mushroom can be used to predict whether it is poisonous or not with reasonable accuracy.

2. Methodology

(a) Data Loading, Splitting, Exploration and Visualization

In the data exploration process, the focus was on identifying the relevant features that could be used for classification purposes. The original dataset contained many null-values and features with low variance, which could negatively impact the model's performance. To further understand the data, bar graphs were used to visualize the categorical features, while histograms were used for the numerical features. The histograms helped to identify the distribution of the numerical features, and the bar graphs helped to identify the frequency of the categorical features. Overall, the data exploration process helped to identify the relevant features for classification and understand their distribution, which could guide the selection of appropriate machine learning algorithms.

(b) Data Pre-Processing

First, I dropped all the features that had high null-values and low variance. Then filled the missing values of the numerical features with their respective mean values. Next, augmented two features, namely the stalk-to-cap ratio and stem volume, and finally, scaled all the numerical features using standard scaling.

I preprocessed the data in this way to handle the missing values and eliminate the categorical features, which are more difficult to work with in certain machine learning algorithms. Dropping low variance features may have also been done to simplify the model and improve performance.

Augmenting numerical features, such as the stalk-to-cap ratio and stem volume, may have been an attempt to provide additional information to the model that could potentially improve accuracy. Overall, the preprocessing steps were aimed at improving the quality and usefulness of the data for machine learning purposes.

(c) Data Augmentation

I augmented two features: "stalk-to-cap-ratio" and "stem-volume".

"Stalk-to-cap-ratio" is the ratio of the stem height to the cap diameter. This feature captures the ratio of the size of the stem to the size of the cap, which could be a useful indicator in determining whether a mushroom is edible or poisonous. For example, some poisonous mushrooms may have very tall and slender stems in proportion to their caps.

"Stem-volume" is the volume of the stem, calculated using the formula for the volume of a cylinder (pi*(r^2)*h). This feature captures the overall size of the stem, which could also be a useful indicator in determining whether a mushroom is edible or poisonous. For example, some poisonous mushrooms may have very large and thick stems, while others may have very small and thin stems.

By augmenting these features, I was able to provide the models with additional information that could potentially improve their accuracy in classifying whether a mushroom is edible or poisonous.

(d) Statistical Hypothesis Testing

As per the output of the logistic regression model, all the five variables (cap-diameter, stem-height, stem-width, stalk-to-cap-ratio, stem-volume) have a statistically significant relationship with the target variable (class) at a significance level of 0.05.

From the coefficients of the logistic regression model, we can see that stem-volume has the highest positive relationship with the target variable, followed by cap-diameter, stem-height, and stalk-to-cap ratio. Stem-width has a negative relationship with the target variable.

This suggests that these variables are important in predicting the class of mushrooms in the dataset, and could potentially be used as features in a machine learning model for classification.

Overall, this suggests that the data has some predictive power in determining the class of mushrooms based on their physical characteristics, and could be useful for further analysis and modeling.

(e) Models of your choice (2 distinct models)
For the non-ensemble methods, I implemented a Decision Tree and Support Vector Machine.

Decision Tree is a simple and interpretable model that works well for problems with categorical data and is not sensitive to outliers. It is also easy to visualize the decision-making process of a decision tree model, which can be helpful for understanding which features are most important in the classification task.

SVM is a powerful and versatile model that works well with both linear and non-linear data. It is also able to handle high-dimensional data, which is useful in this case since there are several numerical features. SVM is often used in classification tasks where there is a clear boundary between classes.

(f) Ensemble Method (1 ensemble method)
I chose to use Random Forest classification.

Random Forest is an ensemble learning method that creates multiple decision trees and combines their predictions to improve accuracy and reduce overfitting. It works well with high-dimensional data and can handle missing values and noisy data. Random Forest is often used in classification tasks where there are a large number of features, as is the case here.

(g) Hyper-parameter Tuning
Grid search is a widely used technique to find the optimal hyperparameters for the model. By using grid search, we can explore different combinations of hyperparameters, evaluate the model's performance on a validation set, and select the best combination that gives the highest performance. This can lead to better generalization of the model and improve its accuracy on the test data.

For the Decision Trees, the hyperparameters tuned are the maximum depth of the tree, minimum number of samples required to split an internal node, minimum number of samples required to be at a leaf node, and the maximum number of features to be considered for the best split. The best parameters were found to be a maximum depth of 3, no maximum features, a minimum sample leaf of 1, and a minimum sample split of 2. The best accuracy achieved with these parameters was 0.566.

For SVC, the hyperparameters tuned are the regularization parameter (C), kernel function ('linear' or 'rbf'), and gamma coefficient. The best parameters were found to be a regularization parameter C of 0.1, using the 'rbf' kernel, and 'auto' gamma. The best score achieved with these parameters was 0.547.

For the Random Forest, the hyperparameters tuned are the number of trees in the forest, the maximum depth of the tree, and the maximum number of features to be considered for the best split. The best parameters were found to be a maximum depth of 5, using 'sqrt' as the maximum number of features, and 200 estimators. The best score achieved with these parameters was 0.545.

3. Results

Based on the standard evaluation metrics of accuracy, precision, recall, and F1-score, the decision tree model had the highest accuracy and precision, while the random forest model had the highest recall and F1-score. The support vector machine (SVM) model had the lowest performance across all metrics. It is important to note that the performance of each model was relatively low, with accuracy scores ranging from 0.55 to 0.65.

The cross-validation strategy used in this analysis was a standard 5-fold cross-validation. This strategy involves splitting the data into 5 equal parts, using 4 parts for training and 1 part for testing, and repeating this process 5 times with a different part of the data held out for testing each time. This approach is commonly used in machine learning as it provides a good balance between training and testing the model on different subsets of the data. It also helps to mitigate the effects of overfitting and ensures that the model is generalizable to new data. Overall, this was an appropriate cross-validation strategy for the problem at hand, given the relatively small size of the dataset.

4. Conclusion

Based on the evaluation metrics and cross-validation results, it seems like the Decision Tree model has the best performance among the three models. It has the highest accuracy, precision, recall, and F1 score on the test set, and it also has the highest cross-validation accuracy score. Additionally, the Decision Tree model is simple and easy to interpret, which is beneficial when making decisions in a real-world scenario. Therefore, I would choose the Decision Tree model for my adventure through Mushroomia.

There are several limitations to this project that need to be acknowledged. Firstly, the dataset used in this project has some limitations. While it is a fairly comprehensive dataset on mushrooms, it is possible that some species of mushrooms may be misidentified, leading to

inaccurate labels. Additionally, the dataset only includes information on mushrooms found in North America, so the models may not generalize well to mushrooms found in other parts of the world.

Secondly, the preprocessing steps taken in this project were limited to dropping categorical features and imputing missing values. While these steps helped to clean the data and prepare it for modeling, there may be additional preprocessing steps that could improve the performance of the models.

Thirdly, the models used in this project are limited in their ability to handle non-linear relationships between features and the target variable. This means that the models may not be able to capture complex interactions between features and accurately predict the target variable.

Finally, the evaluation of the models was based on standard evaluation metrics such as accuracy, precision, recall, and F1-score. While these metrics provide a useful summary of model performance, they do not capture the full range of trade-offs between different types of errors. For example, in the context of mushroom classification, false negatives (i.e. classifying an edible mushroom as poisonous) may be more concerning than false positives (i.e. classifying a poisonous mushroom as edible). Therefore, it would be important to consider the specific context and consequences of different types of errors when evaluating model performance.