Binary Classification of Mushroom

Zequn Chen[1] and Yichen Zhong[2]

University of Michigan, Ann Arbor, MI

1 Introduction

1.1 Problem Address

This Secondary Mushroom Dataset contains 173 species with 353 mushrooms in each species. Each mushroom is identified as definitely edible, definitely poisonous, or of unknown edibility and not recommended(this class was combined with poisonous class). For each mushroom, we have 20 feature variables(such as cap-diameter, cap-shape). Our task is to build an accurate binary classifier that can tell whether a mushroom is edible or poisonous based on given features.

1.2 Motivation

When living in the wild, we often encounter different species of mushrooms. Edible mushrooms are rich in nutrients, but poisonous mushrooms can be deadly. It is feasible to distinguish the toxicity of mushrooms with the naked eye, but in order to reduce the possibility of people eating poisonous mushrooms by mistake, we intend to make a high-accuracy classifier to accurately judge whether mushrooms are edible according to their appearance. In other words, we try to figure out which feature that a edible or a poisonous mushroom represent the best.

1.3 Application

As mentioned before, our project serves as a classifier for edible mushroom based on their features. So the most direct application is that it can be used as a reliable tool to detect whether a mushroom is edible, which helps the people live and travel in the wild a lot.

Further from that, we hope the result of our research can help scientist, especially biologist who need to identify species. The result in this paper should provide them with some, if not at all, useful information about the poisonous of mushroom regardless the mushroom species exists. So when the biologist encounters a new species, it is easy to identify its poison based on our classfier.

2 Methods

2.1 Technical Approach

Based on the data we described in the above section, we need to predict the toxic of a mushroom given its features. We can load our data directly from a csv file downloaded from UCI machine learning dataset. Then we should do some pre-processing. Firstly, we should create dummy variable to convert categorical variable into numerical features in order to analysis easily using Pandas. Secondly, we need to standardize our data to fit the input structure of PCA using the StandardScaler package from sklearn. After doing the data pre-processing, we do our dimensional reduction. We will try both PCA and t-SNE, and choose a method with better performance. After that, we can spilt our concentrated reduced data into training and test set, and train our data with several machine learning method like Naive Bayes, Logistics Regression, Support Vector Machine, Random Forest, and KNN. Finally, we compare the prediction result with the test dataset and choose a model that have the highest accuracy among those models. Here is a pipeline to show the flowchart of our work.

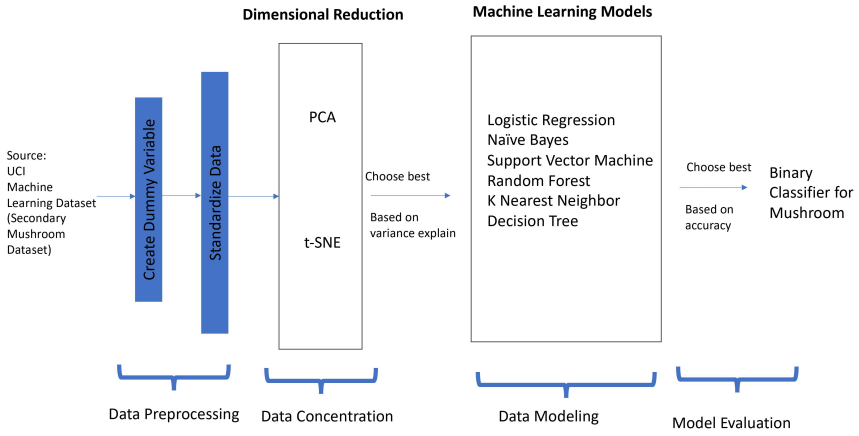

Fig. 1: Model Pipeline

2.2 Data Pre-processing

Since we have 22 features and every feature is categorical, we should use the get_dummies function from Pandas package to convert those categorical features into numerical features. Besides, we should standardize our data related to the highest term among all features before applying PCA techniques. We could use StandardScaler package from sklearn to achive this. What's more, in order to visualize our result better, we use the hot-encoding way to our binary classification variable, and let p represents poisonous and e represents edible.

2.3 Dimensional Reduction

The first important step of our model is to do dimension reduction. We will use two method, one is the linear algorithm PCA, the other is the nonlinear t-SNE. We will evaluate the efficient of these two models and choose the best for dimension reduction.

Principal Component Analysis(PCA) We first try the traditional way, the principal component analysis(PCA) The principal components are new variables that are constructed as linear combinations or mixtures of the initial variables, and most of the information within the initial variables is compressed into the first few components. Since we've already done the standardize data part in preprocessing, we can directly apply PCA through sklearn.decomposition. There are two ways to apply PCA.

t-distributed stochastic neighbor embedding(t-SNE) t-SNE is a state-of-art model in current industry. Compared to the linear dimension reduction method PCA, t-SNE is nonlinear. This means the components of t-SNE do not contain meaning. In stead, t-SNE attempts to identify the structure of data by moving similar and dissimilar points away from each other and do not use labels for classifying. A significant characteristic of t-SNE is that it can compress dimension into no more than 5 dimension, but it requires a lot of computation power to execute this algorithm.

2.4 Machine Learning Models

After we do the dimensional reduction, we try to use several machine learning models to build our binary classifier. After that, we compare the accuracy of each classifier and choose the best to be our final high-accurate classifier.

Naive Bayes Model Naive Bayes Classifier is one of the most suitable methods for classification. The key idea for this model is the Bayes Theorem, which use the prior and likelihood to estimate and predict the posterior probability. One important assumption is the predictors and features are independent, that's why the model is called 'Naive' Bayes. In this project, we will use the BernoulliNB() function in the package sklearn.naive_bayes.

Logistics Regression Model Logistic Regression is a good model to build the classifier and class probability estimation. It takes a linear combination of features and applies to them a nonlinear sigmoidal function. In the baseline of logistic regression, the output feature is binary, but it can be easily extended into multiple classes by using the sklearn package.

Support Vector Machine Support Vector Machine (SVM) is an effective method in high-dimensional classification problems. Its idea is to find a N-dimensional (N is the number of features) hyperplane that distinctly classifies the data and maximize the distance between data points of each class. We build our SVM model using sklearn package. Similar to previous, we set the max iteration number to 10000 to ensure convergence.

Random Forest Random Forest consists of a large number of individual decision tree that work as an ensemble. Each individual decision tree spits out a class prediction and the class with the most votes becomes our final prediction. It is an effective classification method since a large number of uncorrelated trees as a whole will outperform any individual tree model.We also tune the max_depth down to 3 to avoid overfitting with data.

3 Results

3.1 Evaluation Method

We take several methods to evaluate our result. For the dimension reduction part, we use the information explained/captured given specific reduced features to find the better model. Then we do the train_test _split part, with 70% for training and 30% for testing. We also use cross-validation method since the data amount is not too large and we should make use of every data. After we fit our compressed data with five model, we compute the accuracy of each method, and select the method with best accuracy score to be our final classifier.

3.2 Result Analysis

For the dimension reduction part, we have 102 origin features. When we use the PCA, the top 5 components could only explain 17.6% of information, which is a small percent. If we set the information explained threshold to 90%, we need to contain at least 73 out of 102 features. Based on these data, we can conclude that the PCA does not do a good job here. For the t-SNE part, this method could create a relatively less overlapping visualization. We also tune the hyper parameter perplexity to 70 and n_iteration to 1000 to get the best fit, as shown in the figure 2. Hence we will choose t-SNE to do our dimension reduction.

For the machine learning model part, the accuracy data is in the table below. From the table1, we can know that the random forest has the highest accuracy, so we decide to choose random forest to be our classifier.

4 Conclusion

4.1 Main Contribution

In this project, we have build a high-accurate binary classifier for mushroom. With the help of t-SNE, an efficient dimensionality reduction method, effective

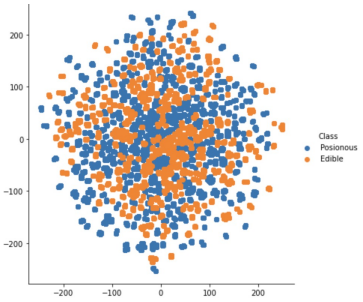

Fig. 2: t-SNE Fit Result

Method	Accuracy
Naive Bayes	0.82
Logistics Regression	0.87
Support Vector Machine	0.90
Random Forest	0.93
K Nearest Neighbor	0.89

Table 1: ML Model Accuracy

data preprocessing and a variety of machine learning methods, we successfully used t-SNE dimensionality reduction and random forest method to construct a classification with an accuracy rate of 93% device.

4.2 What Did and Did Not Work

Our advantage is that we can compress many kinds of features into two dimensions, and we can build a high-accuracy classifier based on this. But our weakness is that because t-SNE is non-linear, the two highly compressed features cannot be well interpreted. Therefore, we can't actually know the relationship between the feature and the toxicity, we can only input the feature into the classifier to judge.

References

1. Dennis Wagner, Dr. G. Hattab, "Mushroom data creation, curation, and simulation to support classification tasks", Scientific Reports, Apr. 2021, doi: 10.1038/s41598-021-87602-3.
2. Vu, Duy-Hien. "Privacy-preserving Naive Bayes classification in semi-fully distributed data model." Computers Security 115 (2022): 102630.
3. Meyners, Michael, and Anne Hasted. "Reply to Bi and Kuesten: ANOVA outperforms logistic regression for the analysis of CATA data." Food Quality and Preference 95 (2022): 104339.
4. Kaul, Ajay, and Sneha Raina. "Support vector machine versus convolutional neural network for hyperspectral image classification: A systematic review." Concurrency and Computation: Practice and Experience (2022): e6945.