

# Deadly Mushroom Classification

Garrett Ransom

October 2018

## Abstract

Eating certain types of mushrooms can be the difference between life and death. In this paper, we evaluate the performance of various classification models that can assist us in distinguishing poisonous mushrooms from edible ones. Our dataset is composed of 8124 examples described by 22 categorical variables and one binary predictor variable.

## 1 Introduction

In this report, we give a brief theoretical introduction to the algorithms that we'll be using for our machine learning task. Then, we give detail into our experiment results, followed by a brief conclusion. The goal of this paper is to find the best algorithm(s) for the task of predicting whether or not a mushroom is poisonous by a set of categorical features.

## 2 Classification Models

### 2.1 Logistic Regression

In the case of Logistic Regression, we use the sigmoid function as a means to map our predictor variable as a probability such that  $y \in \{0, 1\}$ .

$$h_{\theta}(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

where

$$g(z) = \frac{1}{1 + e^{-z}}$$

We maximize the log likelihood function to find the best fit for  $\theta$ .

$$\ell(\theta) = \sum_{i=1}^m y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))$$

We maximize the log likelihood by using gradient descent given by  $\theta := \theta - \alpha \nabla_{\theta} \ell(\theta)$ . By taking the partial derivatives with respect to  $\theta$  for one training example  $(x, y)$ , we get the following equation:

$$\frac{\partial}{\partial \theta_j} \ell(\theta) = (y - h_{\theta}(x)) x_j$$

Therefore, using stochastic gradient descent rule gives us:

$$\theta_j := \theta_j - \alpha (y^{(i)} - h_{\theta}(x^{(i)})) x_j^{(i)}$$

## 2.2 Support Vector Machines

In the case of Support Vector Machines, we create a classifier by using the equation

$$h_{w,b} = g(w^T x + b).$$

Given a training example  $(x^{(i)}, y^{(i)})$ , we can form the idea of a functional margin of  $(w, b)$  with respect to the training example as

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b).$$

Given a training set  $S = \{(x^{(i)}, y^{(i)}); i = 1, \dots, m\}$ , we can define the functional and geometric margin as the smallest margin of across all training examples

$$\gamma = \min_{i=1, \dots, m} \hat{\gamma}^{(i)}.$$

Ultimately, by doing some cool math, we can use the following problem for finding the optimal margin classifier

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{1}$$

$$\text{s.t } y^{(i)}(w^T x + b) \geq 1, i = 1, \dots, m \tag{2}$$

### 2.3 Decision Trees

Suppose we want to create a decision tree for a classification task. If we have  $N$  training examples composed of  $p$  inputs  $(x_i, y_i)$  for  $i, \dots, N$  with  $x_i = (x_{i1}, x_{i2}, \dots, x_{ip})$ , then there exists a set of splitting points that best classify output variables. We can split the data into  $M$  regions  $R_1, R_2, \dots, R_M$  and we can model the response as a constant  $c_m$  in every region:

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m).$$

We can use the Gini Index, Misclassification Error, or Cross-Entropy to find the best splitting points.

$$\text{Gini Index: } \sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$$

### 2.4 Random Forests

Random forests use bootstrapping or bagging during training. The core purpose of random forests is to correct decision trees' natural tendency towards overfitting training data.

---

**Algorithm** *Random Forest Classifier*

---

1. For  $b = 1$  to  $B$ :
  - (a) Draw a bootstrap sample of size  $N$  from  $X, Y$  and call these  $X_b, Y_b$ .
  - (b) Train a classification tree  $f_b$  on  $X_b, Y_b$ .

Let  $\hat{C}_b(x)$  be the classification predication for the  $b$ th random forest tree. The output is the class that wins majority vote.

---

### 2.5 Naive Bayes

The Naive Bayes classifier is a model based on Bayes' theorem with the assumption of independence between features. Bayes' Rule tells us that

$$p(A|B) = \frac{p(A)p(B|A)}{p(B)}.$$

$p(A|B)$  is considered the posterior, while  $p(A)$  is the prior and  $p(B|A)$  is the likelihood and  $p(B)$  is the evidence.

But how does this apply to classification? If we represent a vector  $\mathbf{x} = (x_1, \dots, x_n)$  by  $n$  independent features, we calculate the probability of  $\mathbf{x}$  belonging to a class  $C_k$  as  $p(C_k|x_1, \dots, x_n)$ . Referring back to Bayes' theorem, the conditional probability can be represented as

$$p(C_k|\mathbf{x}) = \frac{p(C_k)p(\mathbf{x}|C_k)}{p(\mathbf{x})}.$$

In practice, we only pay attention to the numerator of this equation. We can further break down the representation into the joint probability model

$$p(C_k, x_1, \dots, x_n)$$

which can be further broken down by using the chain rule of probability

$$p(C_k, x_1, \dots, x_n) = p(x_1|x_2, \dots, x_n, C_k)p(x_2|x_3, \dots, x_n, C_k) \dots p(x_{n-1}|x_n, C_k)p(x_n|C_k)p(C_k).$$

Now, conditional independence comes into play, where we can assume that for each feature  $x_i$ , every other feature  $x_j$  for  $j \neq i$  is conditionally independent relative to  $x_i$ . This means that

$$p(x_i|x_i + 1, \dots, x_n, C_k) = p(x_i|C_k).$$

Therefore, the joint model can be represented as

$$p(C_k|x_1, \dots, x_n) = p(C_k) \prod_{i=1}^n p(x_i|C_k).$$

### 3 Experiment Results and Conclusion

Algorithm	Accuracy Score
Naive Bayes	0.9327
Logistic Regression	1.0
Support Vector Machine	0.997
Decision Tree	1.0
Random Forest	1.0

Here, we must take note that even though all algorithms provide accuracy scores over 90 percent, that our task requires near perfection. Misclassifying a poisonous mushroom is the difference between life and death. Because of this, Naive Bayes actually provides an extremely low score by our criteria. We see that Logistic Regression, Decision Trees, and Random Forests all provide an accuracy score of 100 percent. Therefore, all three of them are quality choices for classifying mushrooms in the future.