Python Implementation of Logistic Regression for the Classification of Mushrooms

Dylan Scott and Jacob Zahn

Texas Tech University

April 2019

Introduction

The Python program described in this report performs logistic regression to classify mushrooms as either poisonous or edible given a set of features. Training from data provided by UCI's "Mushroom Classification" dataset, the system successfully and consistently classified poisonous and edible mushrooms with an accuracy of >93 percent given 5,000 epochs.

A. Testing Data and Model Validation

When loading the dataset from 'mushrooms.csv', the data is randomly permuted and split into training and testing sets. 80 percent of the data went to the training set, while the remaining 20 percent was used for determining testing error to validate our model. Testing error was calculated by averaging the total number of mismatches between the predicted and actual values across the testing set.

B. Dataset

The data used for this project came from UCI's "Mushroom Classification" dataset. The data is "descriptions of hypothetical samples corresponding to 23 species of gilled mushrooms in the Agaricus and Lepiota Family Mushroom drawn from The Audubon Society Field Guide to North American Mushrooms (1981)." This dataset has 22 total feature vectors consisting of discrete qualitative observations for each sample such as 'cap-shape', 'bruises', 'gill-size', and 'veil-color.' The data included 8,124 total samples. Five examples with all included features are shown in table I below.

Table I: Table of Example Data with Features

class	cap-shape	cap-surface	cap-color	bruises	odor	gill-attachn	gill-spacing	gill-size	gill-color	stalk-shape	stalk-root	stalk-surfac
p	x	s	n	t	p	f	c	n	k	e	e	s
e	x	s	y	t	a	f	c	b	k	e	c	s
e	b	s	w	t	l	f	c	b	n	e	c	s
p	x	y	w	t	p	f	c	n	n	e	e	s
e	x	s	g	f	n	f	w	b	k	t	e	s

stalk-color-	stalk-color-	veil-type	veil-color	ring-numbe	ring-type	spore-print	population	habitat	class
w	w	p	w	o	p	k	s	u	p
w	w	p	w	o	p	n	n	g	e
w	w	p	w	o	p	n	n	m	e
w	w	p	w	o	p	k	s	u	p
w	w	p	w	o	e	n	a	g	e

Each feature had between two and seven total possible char values for each sample. Thus, an algorithm was implemented during the program's data fetching function that converted the char values in each feature into an integer value between zero and seven so logistic regression could be performed properly.

C. Logistic Regression Implementation and Performance

To classify each sample as either 'p', poisonous or 'e', edible, logistic regression was implemented via the NumPy library in python. It was able to achieve a runtime of 3.44 seconds for 5,000 epochs, yielding a test error of .067. Higher epoch counts did improve accuracy at the cost of linear time. Counts of 9,000 or more were consistently able to achieve test errors of .05 or less, eventually reaching a limit of around .03. Figure 1 below depicts the relationship between test error and number of epochs.

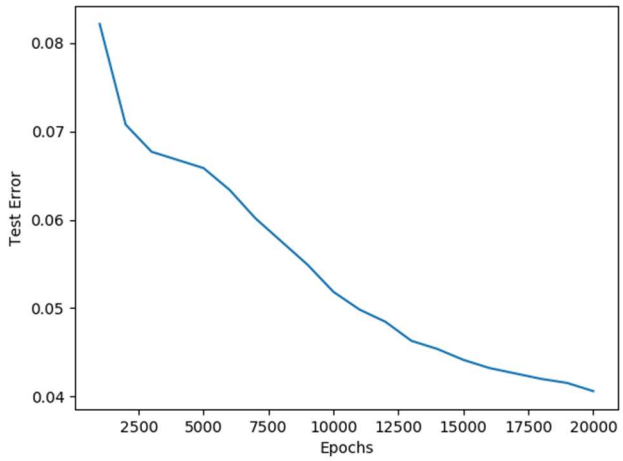


Figure 1: Plot of Epochs vs Test Error

While most of the original testing used a standard learning rate of .01, further investigation into the optimal learning rate revealed that this is not optimal. Figure 2 depicts the relationship between testing error and learning rate. Testing error significantly improves as the learning rate increases from .001 to .02, eventually plateauing when the rate reaches around .023.

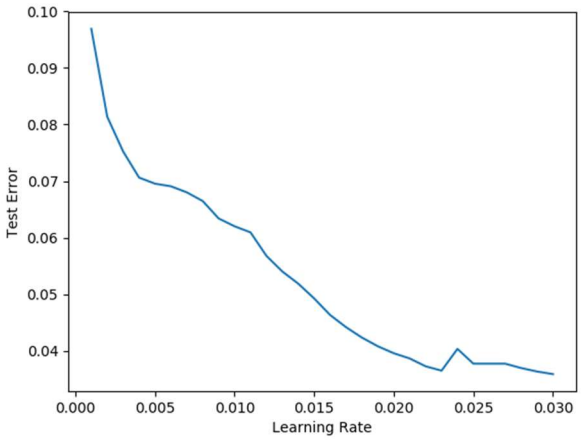
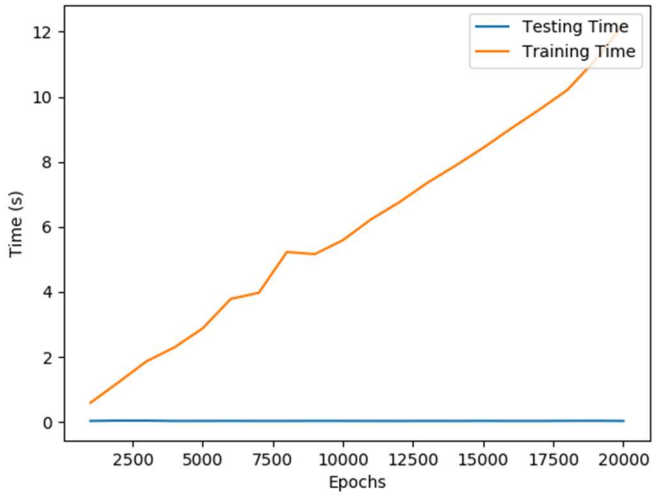


Figure 2: Plot of Learning Rate vs Testing Error

D. Model Training and Testing Time

For a standard run of the model with 5,000 epochs and a learning rate of .01, training time is 3.41 seconds, and testing time is .03 seconds. Training time scales linearly with the number of epochs at a rate of about .5 seconds per additional 1,000 epochs, while testing time remains constant. Figure 2 below shows the relationship between training/testing times and the number of epochs.

Figure 2: Plot of Epochs vs Training and Testing Time



References

Public Domain (1987) Mushroom Classification [Table]. *UCI Machine Learning repository*, *Kaggle.* https://www.kaggle.com/uciml/mushroom-classification