Assignment 2

due October 21

CSMC 352: Machine Learning
Prof. Rose Sloan
Nathan Cho
nc5123@bard.edu

1. Perceptron

Implementation

The following modifications were made to ***problem1.py*** based on the provided python source code.

Added: function ***train_perceptron*** — function returning weight and bias from given training set.
Added: function ***test_perceptron*** — function testing given data set, returns performance metrics.
Modified: function ***evaluate_perceptron*** — based on provided function ***compute_perceptron_error***, reports additional metrics including accuracy, precision, and recall score.
Modified: function ***rank_features*** — based on original function, modified to return the feature label along with weights.

Results

Separately training the algorithm 1,000 times resulted in the following number of iterations per training.

> Training took more than 1 iteration: 100%
> Training took more than 2 iterations: 52.8%
> Training took more than 3 iterations: 5.5%
> Training took more than 4 iterations: 0.2%

As shown above, the validation accuracy reached maximum after 2 iterations in most cases. Additionally, in most (79.6%) cases, accuracy/precision/recall scores were all 1.0. Occasionally, precision and recall scores dropped to 0.66. The reason for accuracy score being very close to 1.0 could be because data points/features are easily linearly separable in the given data set. The following is the average scores from 1,000 separate training results:

> Average test **accuracy score**: 0.99937
> Average test **precision score**: 0.95108
> Average test **recall score**: 0.95087

The three most indicative features of poisonous mushroom were ***spore-print-color_green*** ($w_{avg} = 13.01$), ***odor_creosote*** ($w_{avg} = 10.90$), ***odor_foul*** ($w_{avg} = 10.48$).

The three most indicative features of edible mushroom were ***odor_none*** ($w_{avg} = -11.97$), ***gill-size_broad*** ($w_{avg} = -10.84$), ***odor_anise*** ($w_{avg} = -10.22$).

Full results are available at ***problem1.result.txt***.

2. Logistic Regression

Implementation

The following modifications were made to ***problem2.py*** based on the provided python source code.

Added: function ***train_logistic_regression*** — function returning weight and bias from given training set.
Added: function ***test_logistic_regression*** — function testing given data set, returns performance metrics.
Modified: function ***evaluate_logistic_regression*** — based on provided function ***compute_logistic_regression_error***, reports additional metrics including accuracy, precision, and recall score.
Modified: function ***rank_features*** — based on original function, modified to return the feature label along with weights.

Results

Separately training the algorithm 1,000 times resulted in the following number of iterations per training.

Training took more than 1 iteration: 100%
Training took more than 2 iterations: 39.2%
Training took more than 3 iterations: 12.7%
Training took more than 4 iterations: 2.4%
Training took more than 5 iterations: 0.2%

As shown above, the validation accuracy reached maximum after 3 iterations in most cases. Additionally, accuracy score was > 0.9 in 44.1% of cases, and > 0.8 in all cases. The following is the average scores from 1,000 separate training results:

Average test **accuracy score**: 0.89809
Average test **precision score**: 0.91772
Average test **recall score**: 0.89441

The three most indicative features of poisonous mushroom were ***odor_foul*** ($w_{avg} = 0.0215$), ***stalk-surface-above-ring_silky*** ($w_{avg} = 0.0207$), ***stalk-surface-below-ring_silky*** ($w_{avg} = 0.0200$).

The three most indicative features of edible mushroom were ***odor_none*** ($w_{avg} = -0.0322$), ***ring-type_pendant*** ($w_{avg} = -0.0224$), ***gill-size_broad*** ($w_{avg} = -0.0215$).

Full results are available at ***problem2.result.txt***.

3. Analysis

Model evaluation

In this problem of classifying whether a mushroom is poisonous or not, misidentifying a poisonous mushroom as edible will be fatal if someone uses this classification results. (Imagine a survival app which helps you identify what mushroom you can eat.) On the other hand, misidentifying an edible mushroom as poisonous won't be dangerous as the previous case.

Therefore in this problem, we can say that the **recall** score on classifying poisonous mushrooms correctly is the most important metric. This denotes the proportion of actual positives identified correctly from true positives (*actually* poisonous and *classified* as poisonous) and false negatives (*actually* poisonous but *classified* as edible).

Classifier evaluation

As shown in parts 1 and 2, both classifiers identified most indicative features such as ***odor_foul*** for poisonous, and features such as ***odor_none*** and ***gill-size_broad*** for edible mushrooms. In performance, both showed similar runtime performance but perceptron had better accuracy/precision/recall scores and reached near-optimal decisions faster. The reason for this seems to be because the features are linearly separable, giving advantage to the perceptron model.

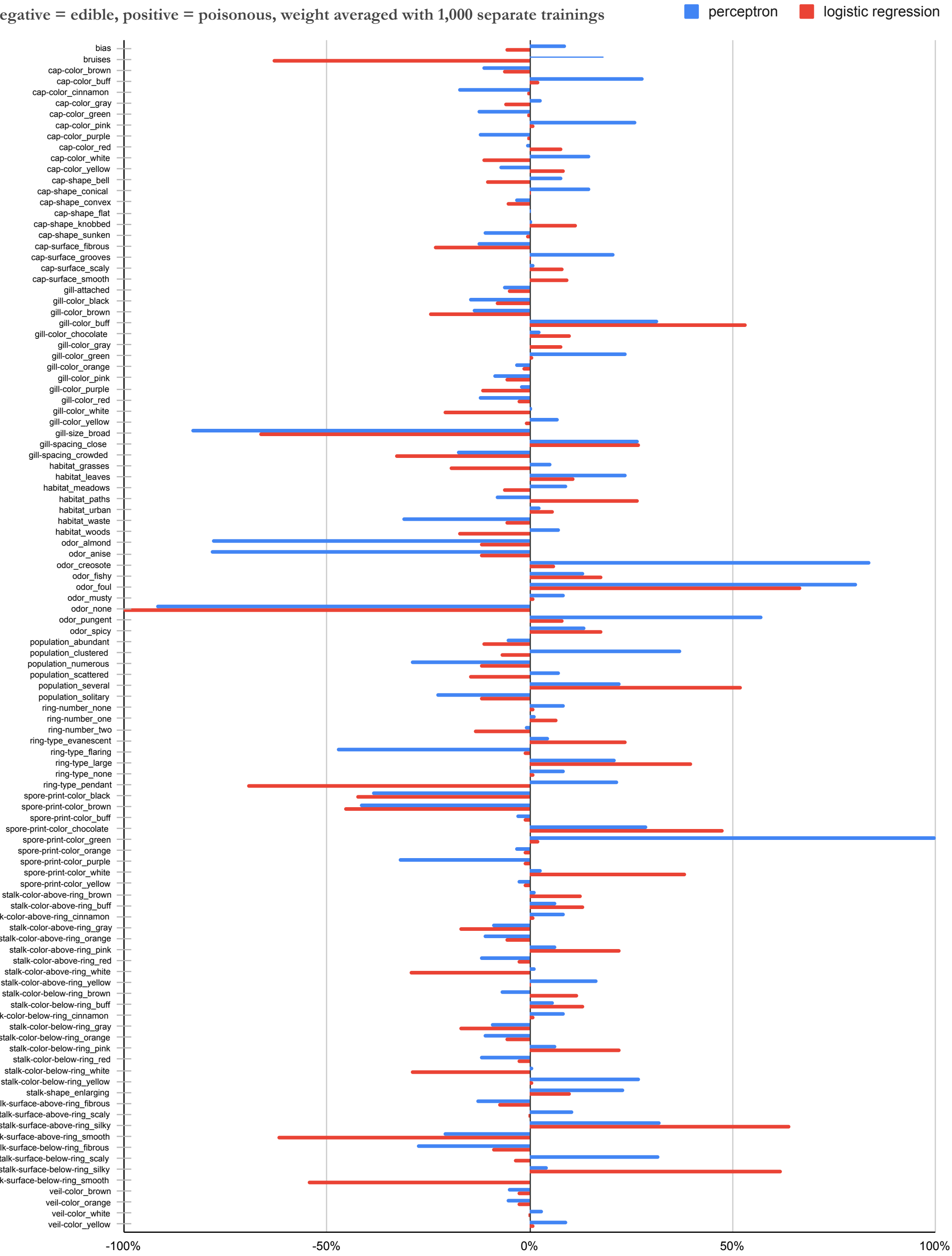
Additionally, perceptron had 3-4 strong indicative features ($weight_{scaled} > 75\%$, see page 5) for both poisonous and edible mushrooms compared to 1-2 strong indicative features in logistic regression. These strong features weighing more than others would've led perceptron to classify with more accuracy than logistic regression.

Therefore for this problem, perceptron is better in all performance metrics.

Feature analysis with weights for each classifier are available below from *page 4* to *page 6*.

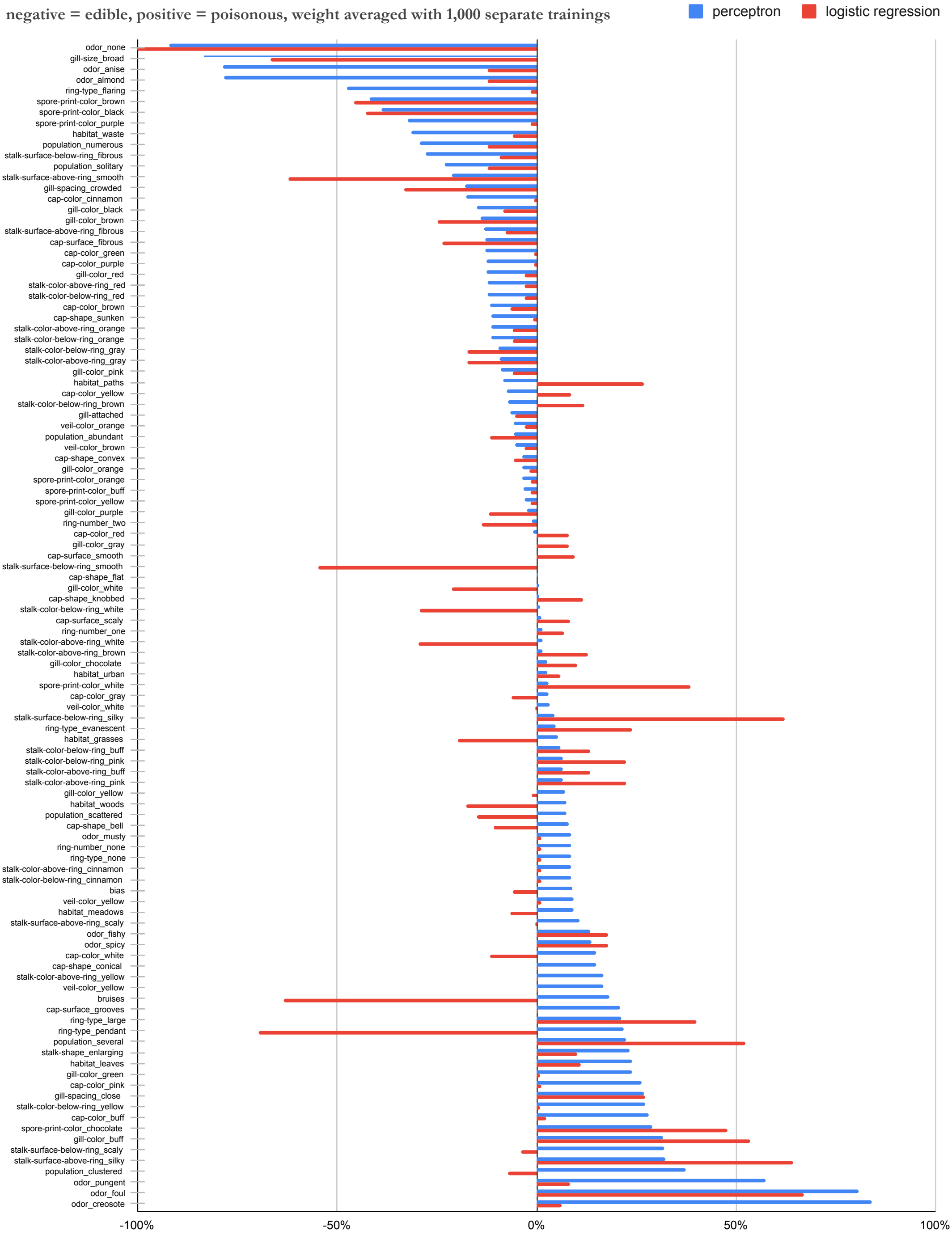
Mushroom Feature Analysis with Weights

negative = edible, positive = poisonous, weight averaged with 1,000 separate trainings



Mushroom Feature Analysis with Weights

negative = edible, positive = poisonous, weight averaged with 1,000 separate trainings



Mushroom Feature Analysis with Weights

negative = edible, positive = poisonous, weight averaged with 1,000 separate trainings