Leaflets three, let it be?

-- Mushroom edibility classification

Wanli Zhang
Martin Guyer
Shangjia Dong

Introduction

- **Task:**

  Classify mushrooms as poisonous or edible based on their physical attributes

- **Method**
  - Binary classification problem
  - Naive Bayes

Introduction

- **Data set:**
  - Response: Binary (poisonous/edible)
  - Attribute: Categorical
  - Instances: 8124
  - Attributes: 22
    - Missing value in one attribute (2480)
    - Not missing completely at random
    - Treat missing values as another category

Methodology

Naïve Bayes classifier

Bayesian classifiers assign the most likely class to a given instance described by its feature vector.

Aim to computing the probability:

$$p(C|F_1, \dots F_n)$$

Where C denotes a class variable with some number of classes.

Methodology

Based on the Naïve Bayes assumption

$$p(C|F_1, \dots F_n) = \frac{1}{Z} p(C) \prod_{i=1}^{n} p(F_i|C)$$

Where

$Z = p(F_1, \dots F_n)$

$p(C)$: Calculated or estimated by relative frequencies.

$p(F_i|C)$: Bernoulli and Multinomial distribution are adopted for feature probability distribution.

Methodology

Classification Rule

In Binary classification, a threshold is utilized:

- ➢ If the prediction probability falls above the threshold, the instance is labeled positive (poisonous, in our case)
- ➢ If not, negative (edible).

Multi-class:

- ➢ Assign instance to most probable class

Methodology

Validation

K-fold cross validation

- Data set is split into six equally-sized folds
- Apply Naïve Bayes once for each run
- Each time:
  - Five folds as training set
  - Remaining one as test set

Methodology

- Receiver Operating Characteristics (ROC)

  Select an optimal threshold to map instances to predicted classes

- Apparent Error Rate (APER)

  Fraction of misclassified sample observations

- False Negative (FN) Rate

  The probability of classifying a poisonous (positive) mushroom as being edible (negative)

Methodology

R packages

- **Naïve Bayes:**

  library(e1071)

  Function: naiveBayes()

  predict()

- **ROC plot:**

  library(ROCR)

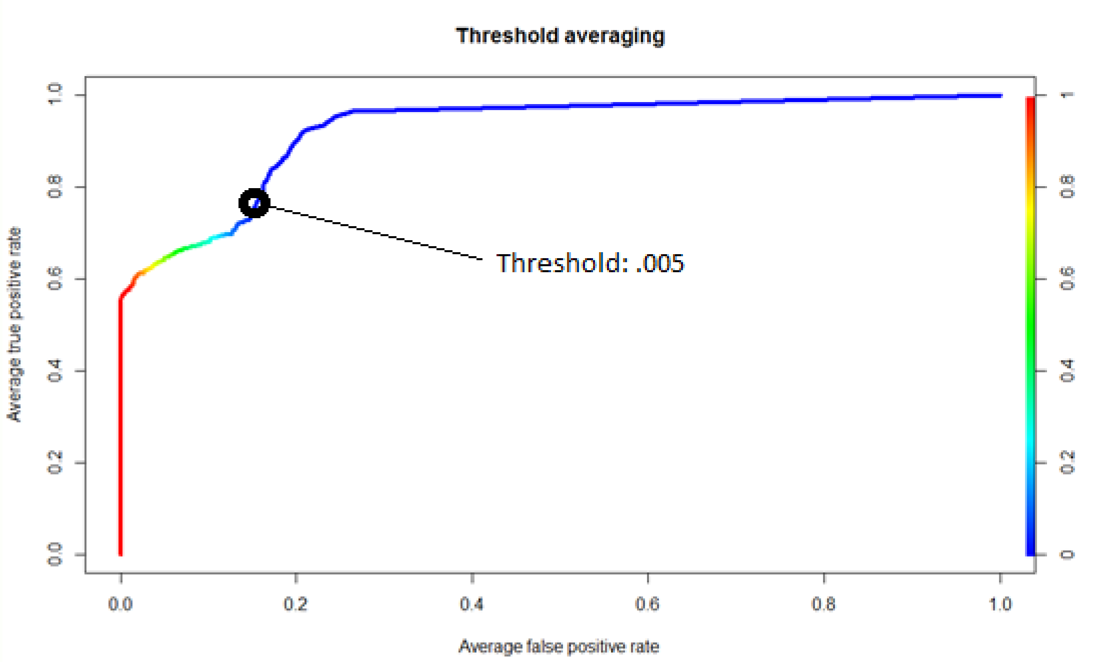
  Function: pred()

  performance()

Oregon State UNIVERSITY

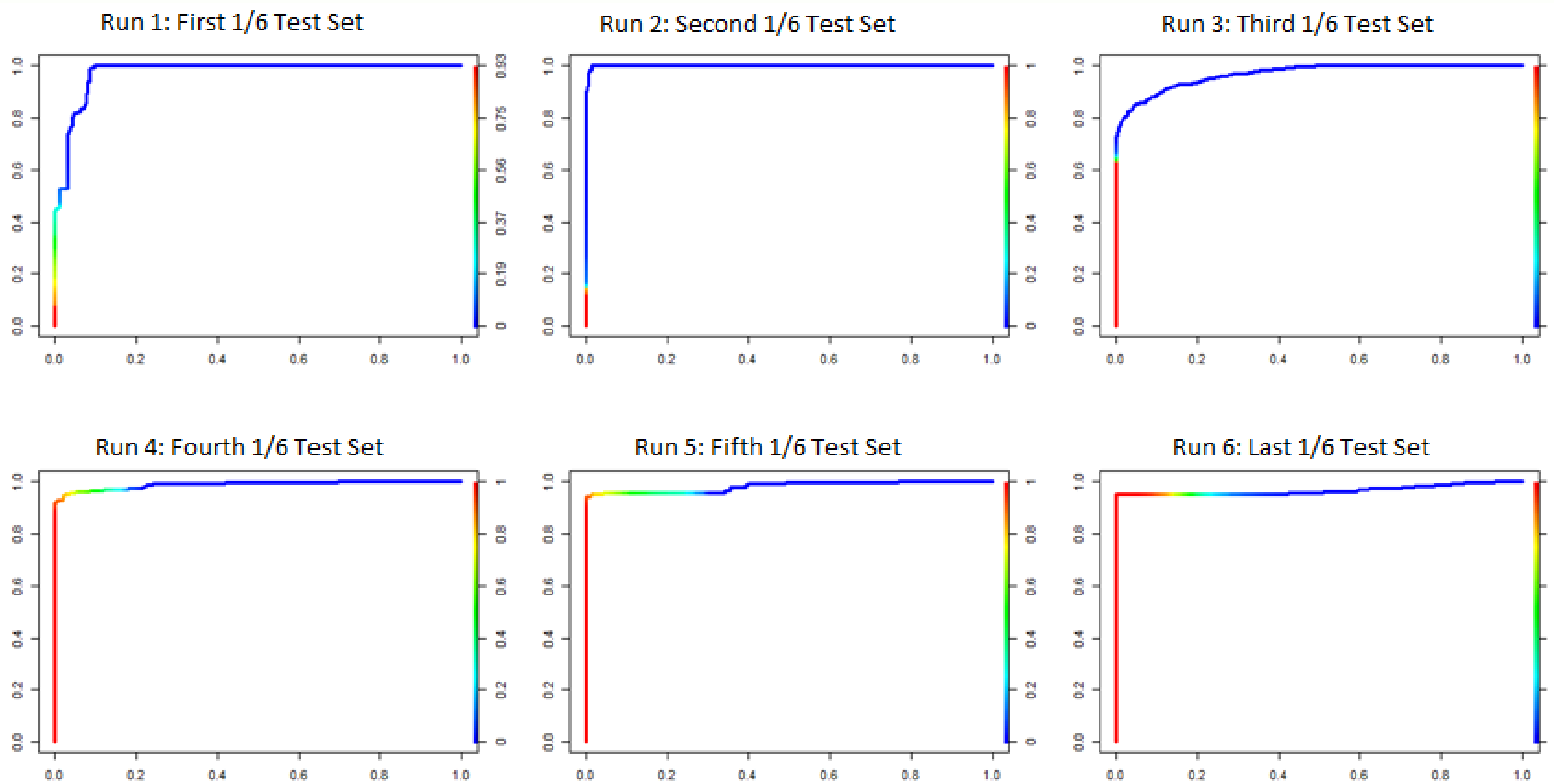
Findings

ROC graph depicts relative tradeoffs between benefits (TP) and costs (FP). The more top-left a point lies, the better the corresponding classifier performance.



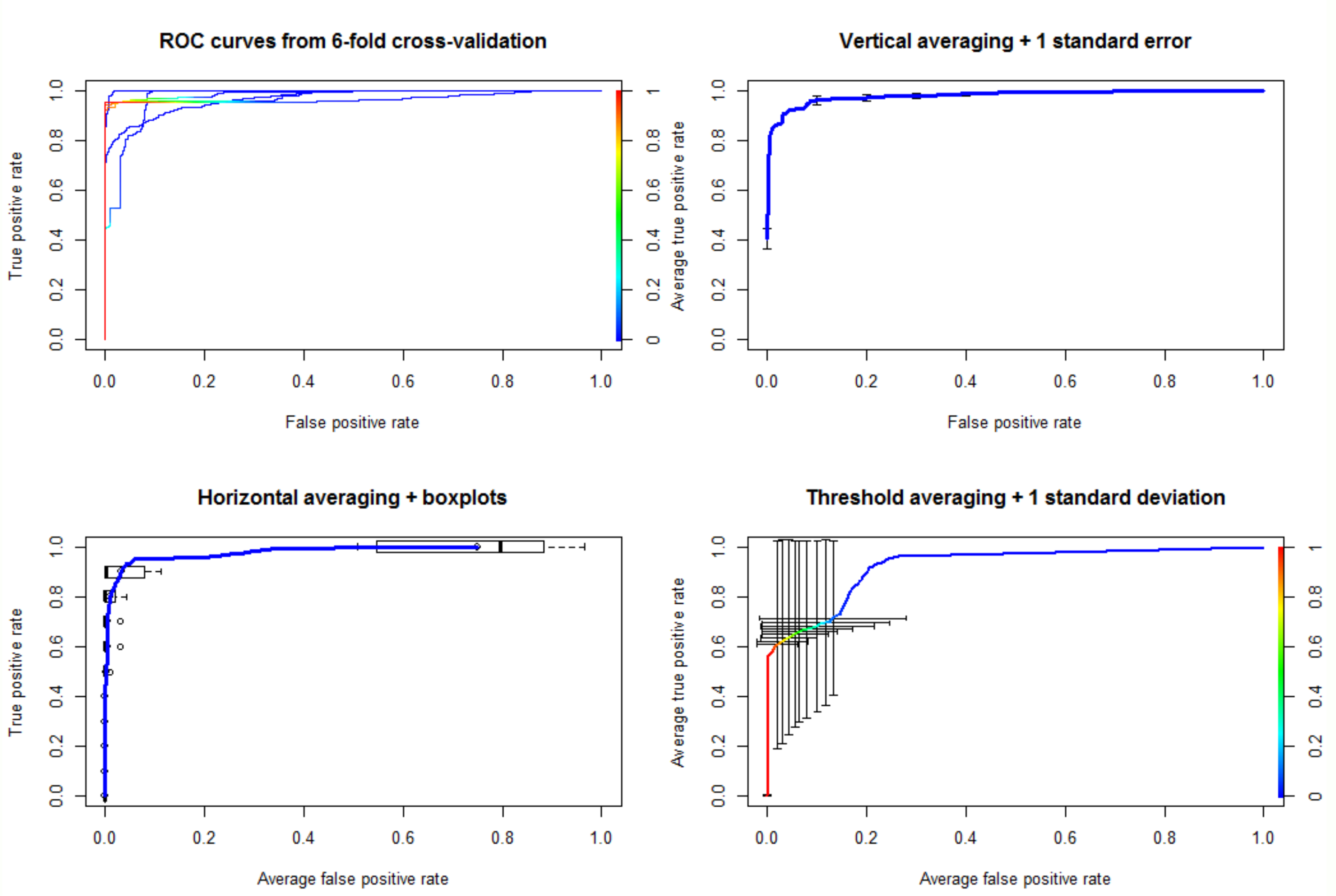
Oregon State UNIVERSITY

Findings



Oregon State UNIVERSITY

Findings



ROC curves from 6-fold cross-validation
True positive rate
False positive rate
Average true positive rate
Vertical averaging + 1 standard error
False positive rate
Horizontal averaging + boxplots
True positive rate
Average false positive rate
Threshold averaging + 1 standard deviation
Average true positive rate
Average false positive rate

Findings

Finding optimal threshold by averaging over all runs:

Threshold	False Negative Rate	Apparent Error Rate
0.5	0.70588235	0.07976366
0.1	0.48366013	0.06351551
0.05	0.45751634	0.07828656
0.01	0.1633987	0.0760709
0.005	0.03921569	0.07976366
0.001	0.0000000	0.1329394

Oregon State UNIVERSITY

Findings

APER & FNR for each run:

Run	APER	FNR
1	0.0798	0.0392
2	0.0214	0.2302
3	0.0805	0.2013
4	0.0517	0.0234
5	0.0805	0.0428
6	0.2349	0.0385
Avg	0.0798	0.0392

Threshold = 0.005

Findings

With our choice of threshold (0.005):

- Naïve Bayes misclassifies 7.98% of mushrooms
- Poisonous mushrooms identified as being edible 3.92% of time

Discussion

Assumption

- Independence between features given class labels
- Unrealistic in practice
- Naïve Bayes gives good results even if assumption is not met
- Reason behind robustness is an open question

Discussion

Scaling to Big data

- Run time of 6-fold CV: **6.13s**
- Expect run time to increase with size of test set
- 1/6 training and 5/6 testing : **42.14s**
- Expect run time to increase with # of features
  - Assume multinomial distribution
  - Each new feature with *k* levels means an extra *k-1* parameters to estimate

Oregon State UNIVERSITY

Discussion

Obstacles

Which classification method should we use?

- Logistic regression: did not work, because iterative algorithm didn't converge

- K-nearest neighbors: did not work, because *knn* does not take distance matrix as input, instead, it calculates a Euclidean distance matrix automatically
  - Side: R doesn't like a distance matrix of dim 8124 X 8124

Oregon State UNIVERSITY


ANY
QUESTIONS
?