Lab #4: Introducing Classification

Everything Data
CompSci 216 Spring 2019
Announcements (Tue Feb 12)

• **HW 1 & 2 Graded**
  – refer to solutions in repo

• **Project team formation**
  – due in two weeks (Tuesday 2/26)
  – 5 is the ideal team size; talk to us if you need special arrangement
Format of this lab

• Introduction to classification
• Lab #4
  – Team challenge: extra credits!
• Discussion of Lab #4 (~5 minutes)
Introducing Lab #4

**Classification** problem example: Given the set of movies a user rated, and the user’s occupation, predict the user’s gender

![Table showing features and outcomes]

**Accuracy** = (# test records classified correctly) / (# test records)
Where is test data?

What if no test data is specified, or we don’t know the right answers?

• We can still evaluate our classifier by splitting the data given to us

Rookie mistake: 
train and test using the same (whole) dataset
Lucky splits, unlucky splits

• What if a particular split gets lucky or unlucky?
• Should we tweak the heck out of our classification algorithm just for this split?

☞ Answer: cross-validation, a smart way to make best use of available data
$r$-fold cross-validation

- Randomly divide data into $r$ groups (say 10)
- Hold out each group for testing; train on the remaining $r - 1$ groups
  - $r$ train-test runs and $r$ accuracy measurements
  - A better picture of performance
Three little classifiers

- `classifyA.py`: a “mystery” classifier
  - Read the code to see what it does
- `classifyB.py`: Naïve Bayes Classifier
  - Along the same line as Homework #4, 3(C)
- `classifyC.py`: $k$-Nearest-Neighbor Classifier
  - Given $x$, choose the $k$ training data points closest to $x$; predict the majority class

http://www.weirdspace.dk/Disney/ThreeLittlePigs.htm
More on the $k$NN classifier

$k = 1$

Source: Daniel B. Neil’s slides for 90-866 at CMU
How do we determine nearest?

• Euclidean distance?
  – Two attributes $x$ and $y$:
    \[ D = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}. \]
  – Three attributes $x$, $y$, and $z$:
    \[ D = \sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2} \]
  – and so on, but beware of the curse of dimensionality
Team work

1. Train-Test Runs and the Mystery of $A$
   (A) Which classifier seems to work best?
   (B) What exactly does $A$ do?

2. Tweaking $k$NN
   (A) How does $k$ affect accuracies on training vs. test data? Is big or small $k$ better for this problem?
   (B) How does $k = 500$ compare with $A$?
Team challenge

*The Evil SQL Splitters:* find a train-test split such that the classifiers are great on training data but horrible on test

*Redemption of Naïve Bayes:* find a train-test split such that $B$ beats $A$ and $C$ hands-down

- Extra credit worth 5% of a homework if $4 \times$ and $B$ has $\geq 60\%$ accuracy; must get checked off in class