INFO251 – Applied Machine Learning

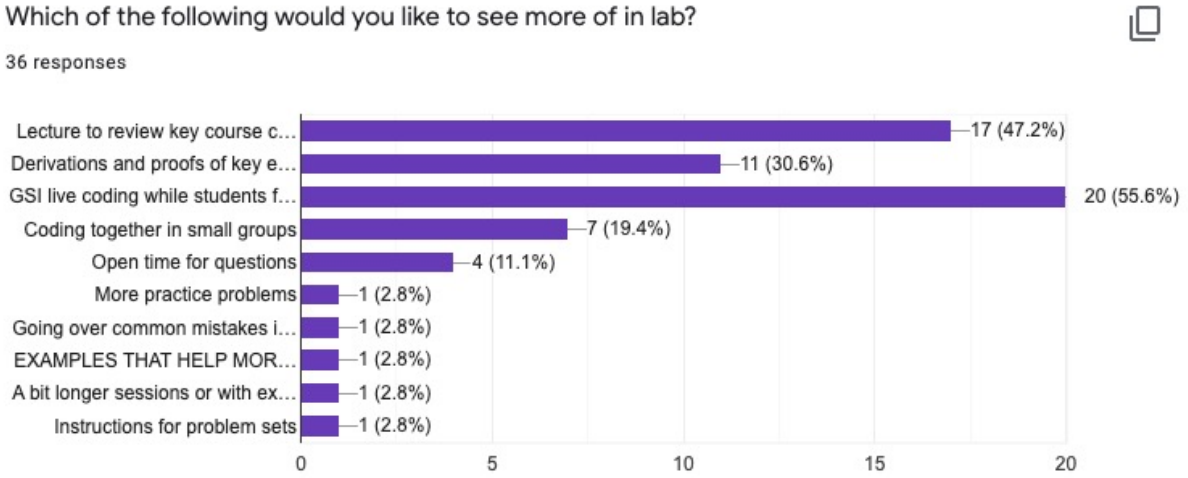
Lab 7

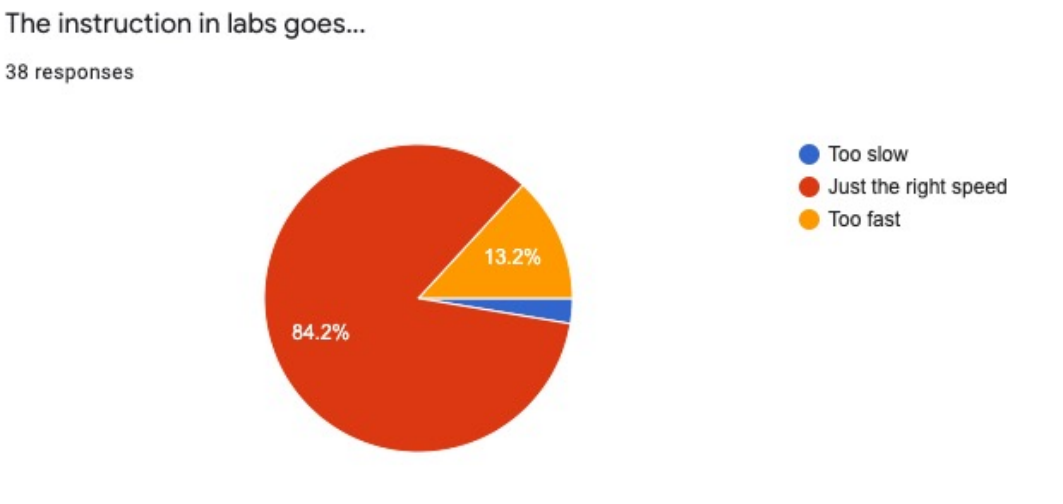
Emily Aiken

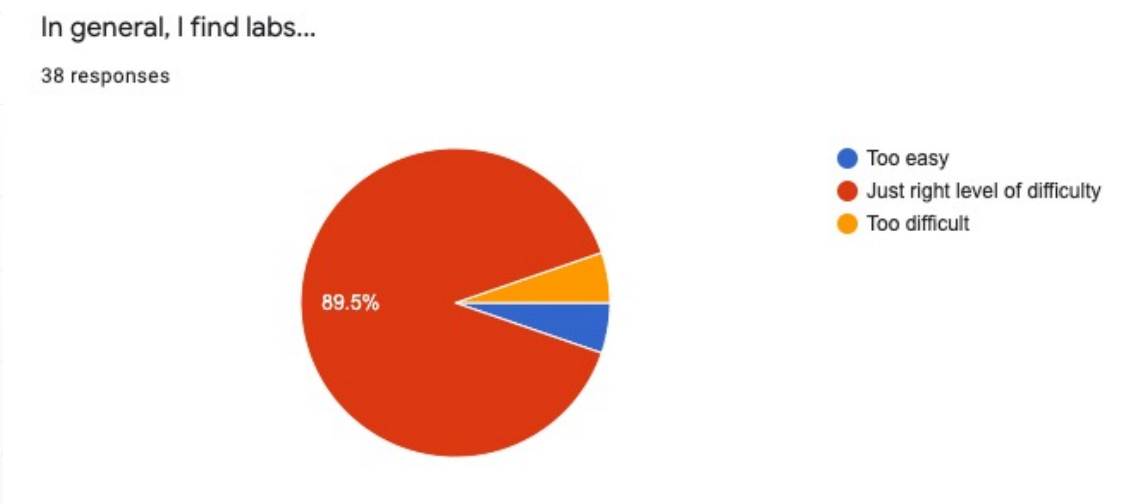
Announcements

- PS3 grades posted
- PS4 due Monday March 14
- PS5 released next week
- Problem set submission reminders
  - Restart kernel and run all cells
  - Submit Jupyter notebook (.ipynb) and PDF

Feedback


Which of the following would you like to see more of in lab?
36 responses
Lecture to review key course c...
17 (47.2%)
Derivations and proofs of key e...
11 (30.6%)
GSI live coding while students f...
20 (55.6%)
Coding together in small groups
7 (19.4%)
Open time for questions
4 (11.1%)
More practice problems
1 (2.8%)
Going over common mistakes i...
1 (2.8%)
EXAMPLES THAT HELP MOR...
1 (2.8%)
A bit longer sessions or with ex...
1 (2.8%)
Instructions for problem sets
1 (2.8%)
0
5
10
15
20


The instruction in labs goes...
38 responses
13.2%
84.2%
Too slow
Just the right speed
Too fast

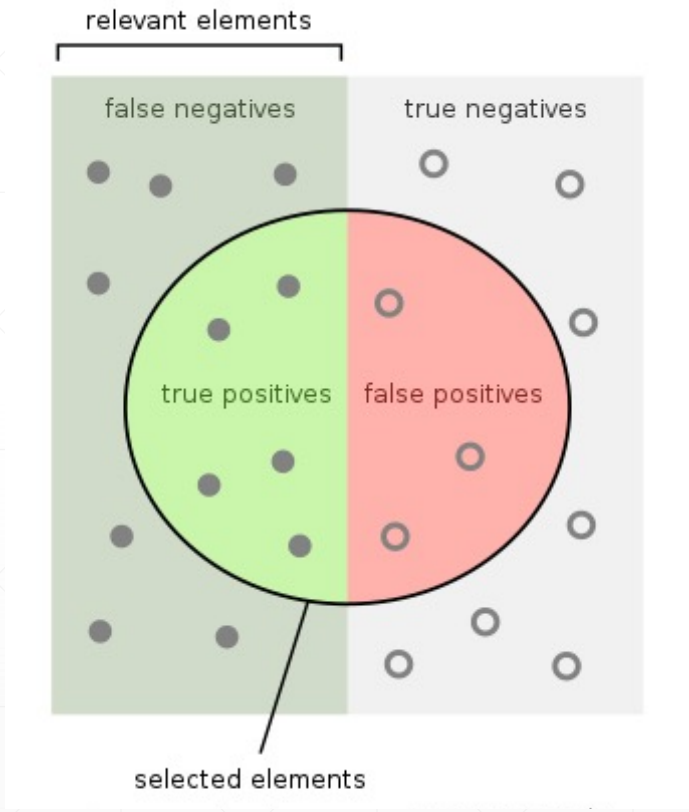

In general, I find labs...
38 responses
89.5%
Too easy
Just right level of difficulty
Too difficult

Today's Topics

- Classification Measures of Accuracy
- Decision Trees
- Random Forests

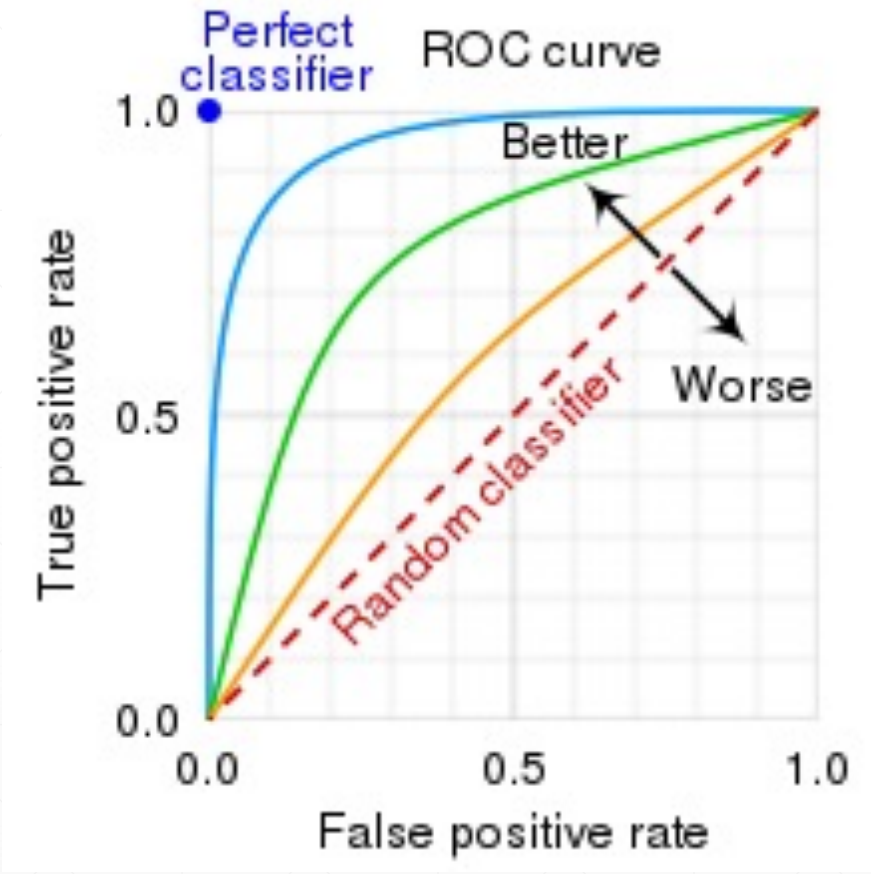
Classification Measures of Accuracy

- *Accuracy* = (TP + TN)/(TP + TN + FP + FN)
- *True Positive Rate (TPR)* = *Sensitivity* = *Recall* = TP/(TP + FN)
- *False Positive Rate (FPR)* = FP/(TN + FP)
- *True Negative Rate (TNR)* = *Specificity* = TN/(TN + FP) = 1 - FPR
- *False Negative Rate (FNR)* = FN/(TP + FN) = 1 – TPR
- *Precision* = *Positive Predictive Value* = TP/(TP + FP)
- *F1 score* = (2 x Precision x Recall) / (Precision + Recall)



ROC Curves

- Test alternative classification thresholds, record trade-off between TPR and FPR
- “Optimal” point on ROC curve: Closest to top-left corner?
- Other option for “quota” problems: Set “acceptance rate” to the rate of positive observations in the training set
- **Exercise**: Prove that calibrating the acceptance rate balances precision and recall

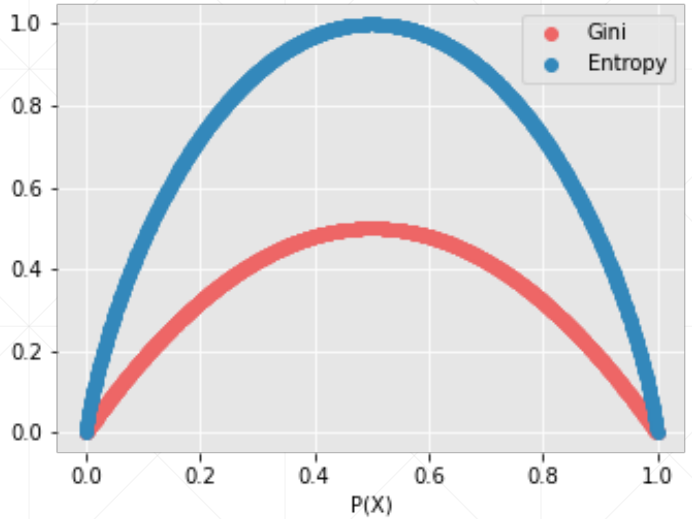


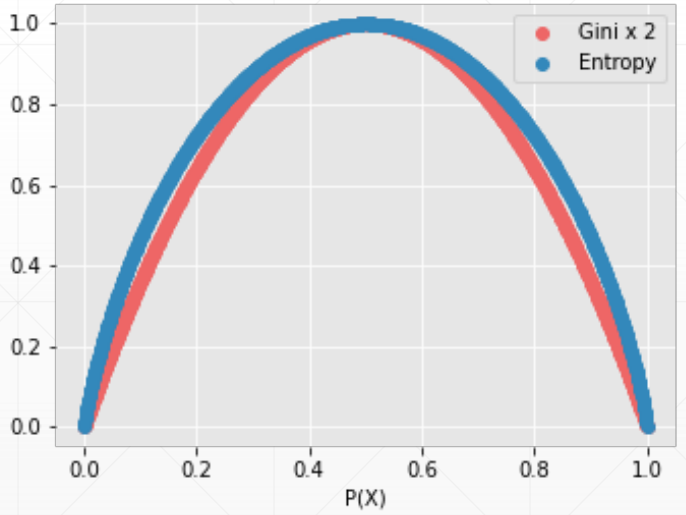
Classification Decision Tree Algorithm

```
def GrowTree(S):
        if y == 0 for all (x, y in S):
                return leaf(0)
        elif y == 1 for all (x, y in S):
                return leaf(1)
        else:
                choose attribute x_j
                s0 = [(x, y) in S if x_j  == 0]
                s1 = [(x, y) in S if xj == 1]
                return node(x, GrowTree(s0), GrowTree(s1))
```

Decision Tree Splitting Criteria

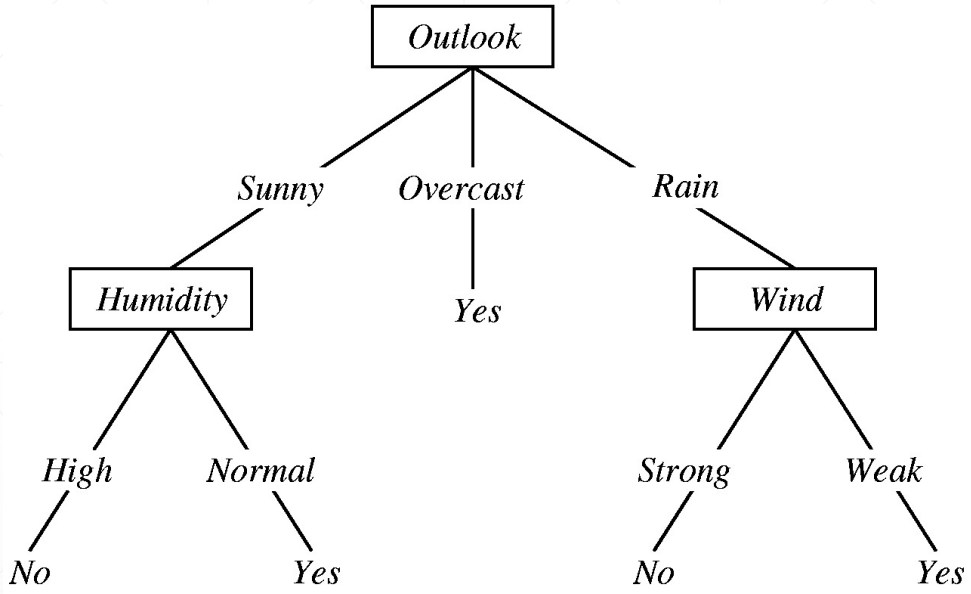
- Classification
  - **Entropy**: $-\sum_{c=0}^{C} p_c \log_2 p_c$
  - **Gini Impurity: 1 -** $\sum_{c=0}^{C} {p_c}^2$
- Regression
  - **Sum of Squared Errors:** $\sum_{i=1}^{N} y_i - \bar{\bar{y}}$





Decision Tree Interpretability

- Tree Diagram
- Feature Importances
  - **Either**: Number of times the feature was split on
  - **Either**: Feature permutation
  - **Classification**: Weighted mean reduction in impurity (across all splits)
  - **Regression**: Weighted mean reduction in MSE (across all splits)



Random Forests

- **Bagging** = **B**ootstrapp **agg**regating
  - Build an **ensemble** of models based on random subsets of the data (sampled with replacement)
  - Model predictions vote (classification) or are averaged (regression) for the ensemble prediction
- **Random Forests:** Bootstrap aggregating with decision trees, plus select random subsets of features (with replacement) for each tree
- Feature Importances
  - Mean feature importance across all trees (can also take standard deviation)
  - Feature permutation