INFO251 – Applied Machine Learning

Lab 8

Announcements

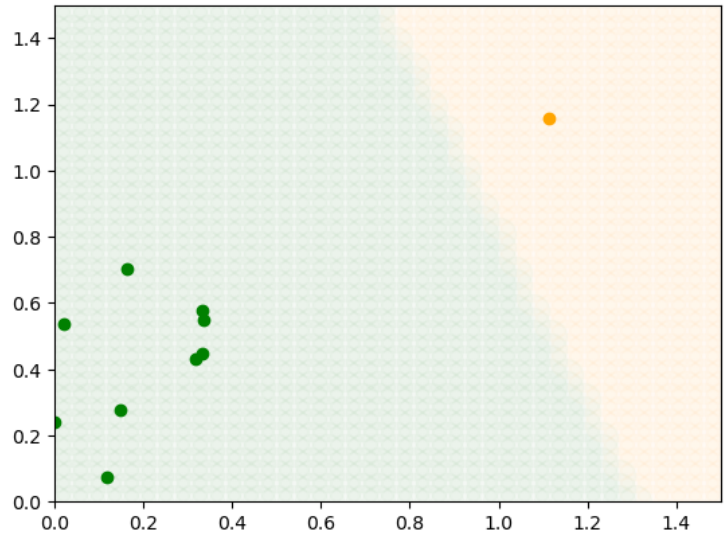
- PS3 grades posted

Today’s Topics

- Classification Measures of Accuracy
- Decision Trees
- Random Forests

Performance / Evaluation Metrics

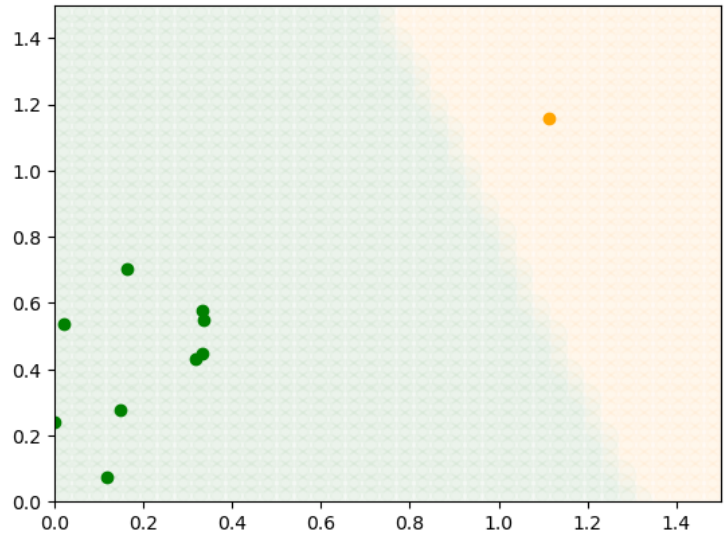
- Assume that green is the “positive” class here
- Accuracy = TP + TN / (TP + FP + FN + TN)
- TPR = TP / (TP + FN)
- FPR = FP / (FP + TN)
- Precision = TP / (TP + FP)



		Predicted	
		Green	**Orange**
Actual	**Green**	TP	FN
	Orange	FP	TN

Performance / Evaluation Metrics

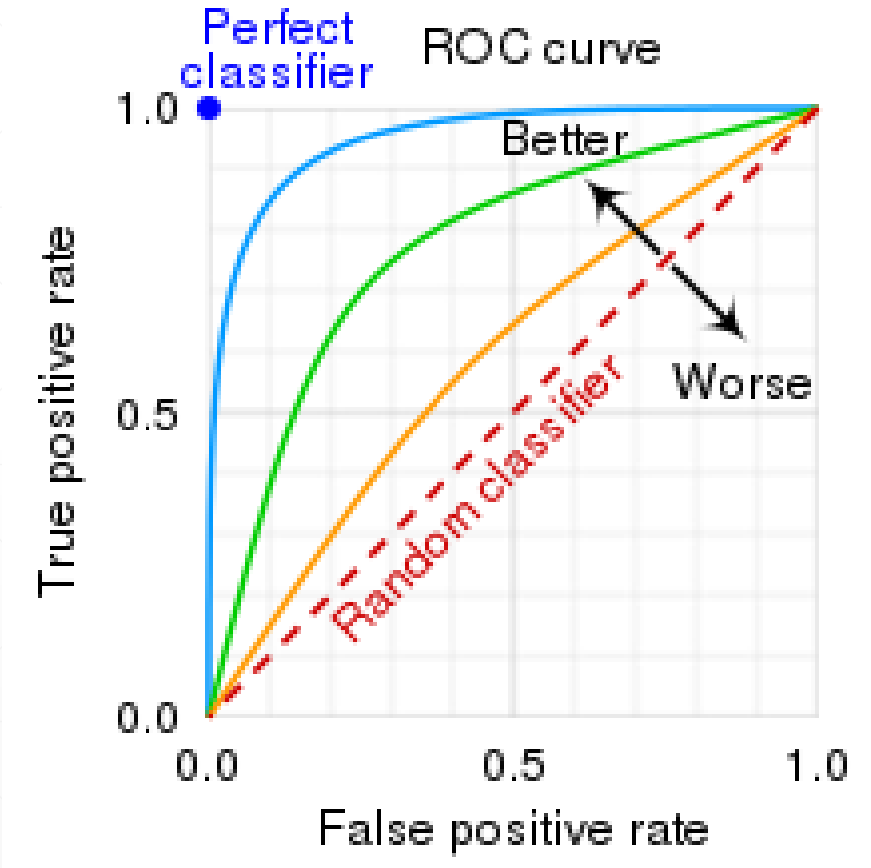
- TPR = P[ŷ(x) = Green | y = Green]
- FPR = P[ŷ(x) = Green | y = Orange]
- Precision = P[y = Green | ŷ(x) = Green]



		Predicted	
		Green	**Orange**
Actual	**Green**	TP	FN
	Orange	FP	TN

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



Trees / Forests

- Pick evaluation metrics
- Determine hyperparameters to tune
- Assess Feature Importance

Example: 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))
```

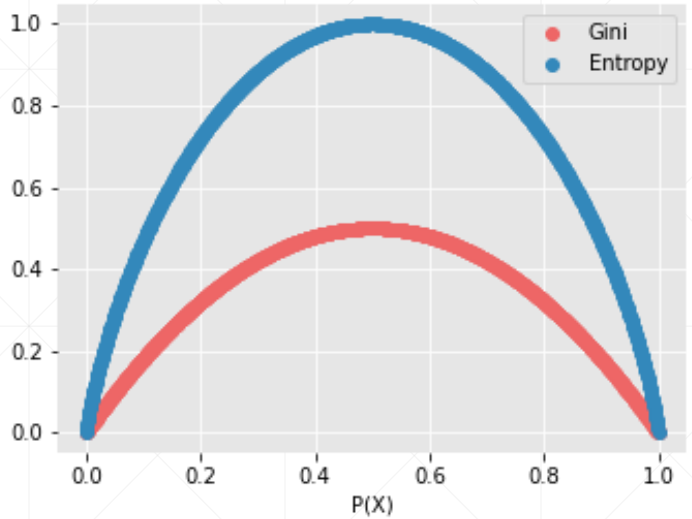
Example: Classification Decision Tree Algorithm

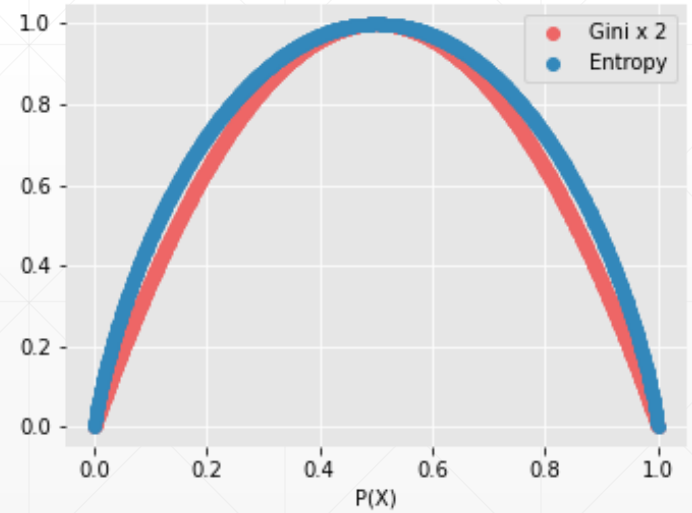
```
def GrowTree(S):
    if y == 0 for all (x, y in S):
        return leaf(0)
    elif y == 1 for all (x, y in S):
        return leaf(1)
    else:
        choose attribute x_j
        s0 = [(x, y) in S if x_j == 0]
        s1 = [(x, y) in S if xj == 1]
        return node(x, GrowTree(s0), GrowTree(s1))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

What are the hyperparameters?

Decision Tree Splitting Criteria

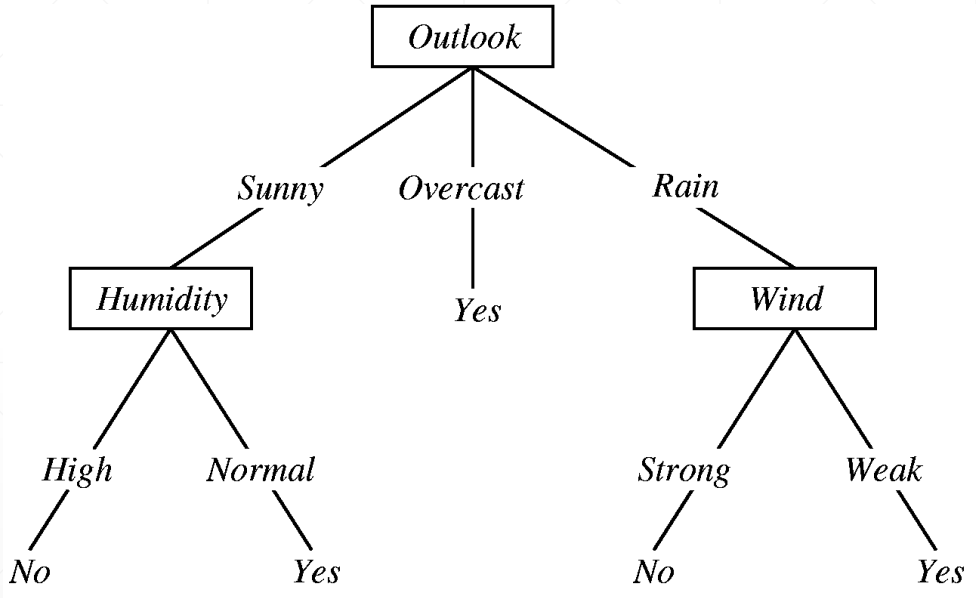
- Classification
  - **Entropy**: $-\sum_{c=0}^{C} p_c \log_2 p_c$
  - **Gini Impurity: 1 -** $\sum_{c=0}^{C} {p_c}^2$
- Regression
  - **Sum of Squared Errors:** $\sum_{i=1}^{N} y_i - \bar{\bar{y}}$





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)



Example: 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
  - What are the hyperparameters?
- Feature Importances
  - Mean feature importance across all trees (can also take standard deviation)
  - Feature permutation