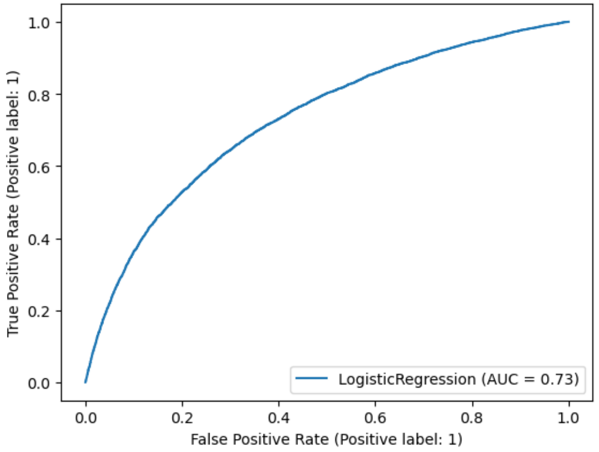
Classification Trees and Ensemble Methods

DUKE
FUQUA
SCHOOL OF BUSINESS

ROC Curves vs. Precision-Recall Curves

- ROC curve for banking data logistic regression model (module 1):

sensitivity

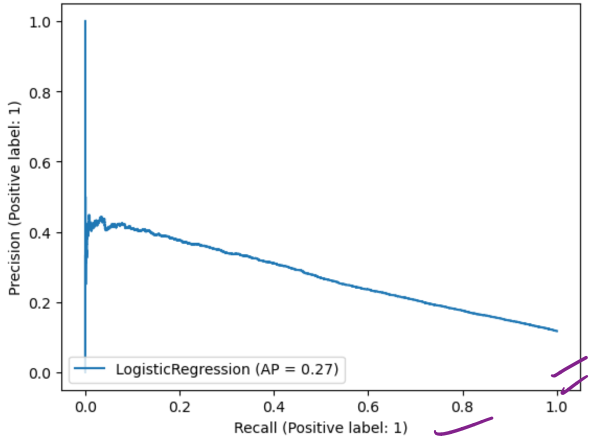


1-specifity

- Question: Can we see the precision on an ROC curve?
- Answer:
  1. No.
  2. However, you could calculate it if you knew base rates.
  3. Alternatively, use `sklearn.metrics.PrecisionRecallDisplay.from_estimator(estimator, X, Y)`

Results from banking data logistic regression model:

$\frac{TP}{TP+FP}$



= sensitivity

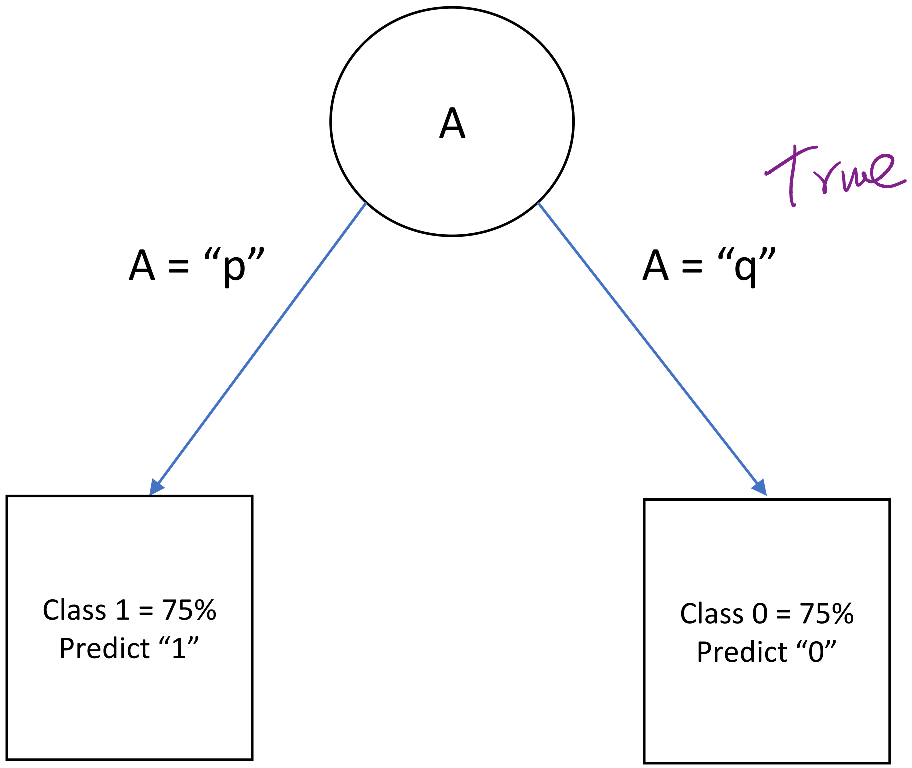
Classification trees: simple example

- Consider our targeted marketing example:
  - Customers are in two classes: class 1 (“purchase”) or class 0 (“no purchase”)
  - Every customer has two features A and B that we will use for prediction
    - Feature A can take two values: “p” or “q”
    - Feature B can take two values: “r” or “s”

- The true population of all customers (we don’t know this!)
  - Overall, 50% of customers are class 1, and 50% of customers are class 0
  - Of the class 1 customers, 75% have “p” for feature A
  - Of the class 0 customers, 75% have “q” for feature A
  - Among all customers, 50% have “r” for feature B and 50% have “s” for feature B

An optimal classification tree

(normalized)



		Predicted 0	Predicted 1
True	0	$.5 \times .75 = .375$	$.5 \times .25 = .125$
	1	$.125$	$.375$

- Accuracy?

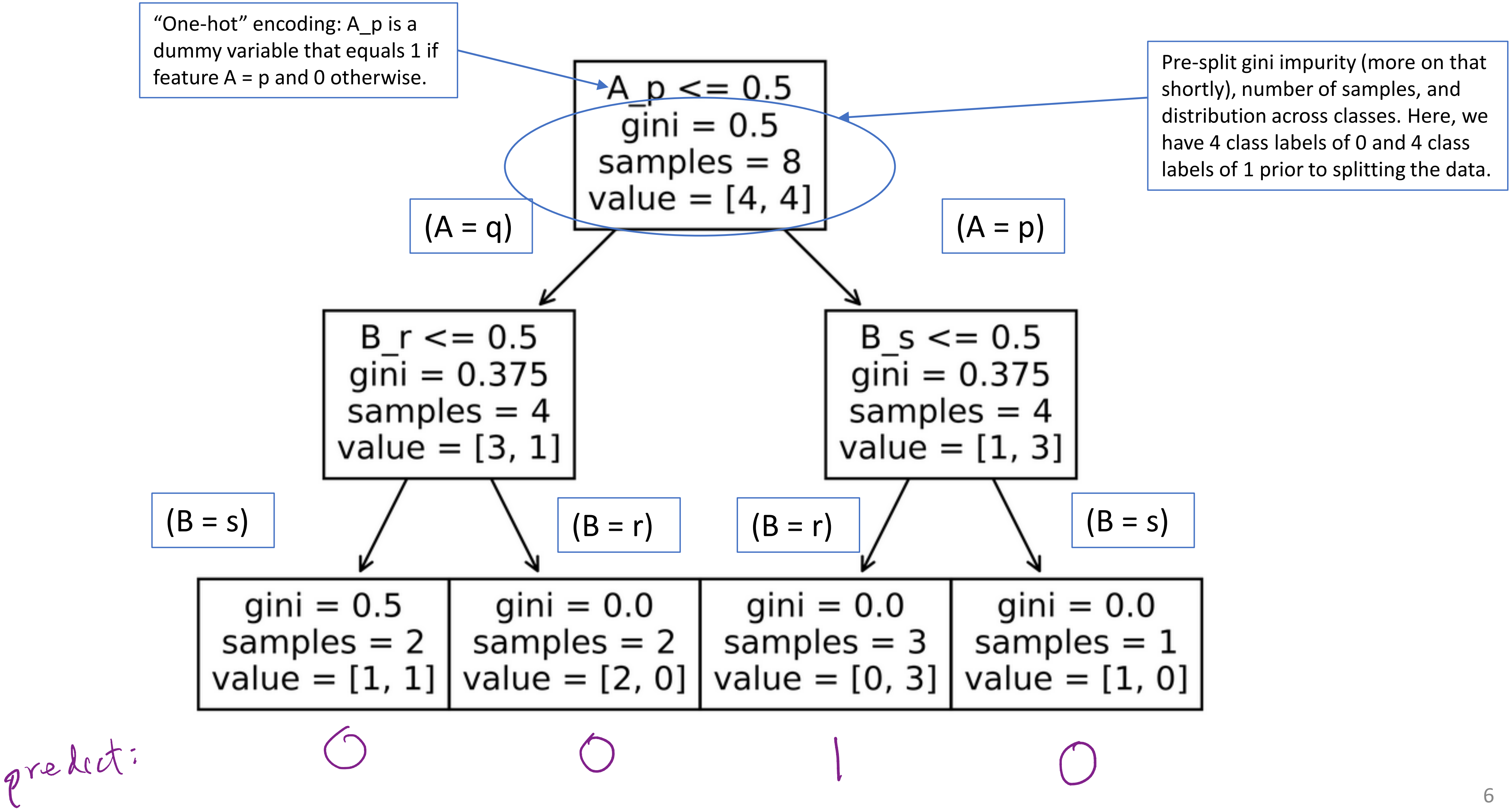
$$\text{Accuracy} = .375 + .375 = .75$$

Fitting a classification tree to training data

- What would a tree fitted to this data look like?

Row	Feature A	Feature B	Class
1	p	r	1
2	p	r	1
3	p	r	1
4	q	s	1
5	p	s	0
6	q	r	0
7	q	s	0
8	q	r	0

Classification tree on the training data (from sklearn)



What is the "Gini impurity?"

- Imagine randomly sampling an instance from a section of our training data without seeing the instance's class label (0 or 1).
- If we randomly classify the instance according to the fraction of this section of the data that is class 1, then the Gini impurity = our probability of making a mistake.
- Example:

Row	Feature A	Feature B	Class
1	p	r	1
2	p	r	1
3	p	r	1
4	q	s	1
5	p	s	0
6	q	r	0
7	q	s	0
8	q	r	0

Consider all the training data. Here, half (0.5) of the instances are of class 1.

Thus, if we consider a random instance (row) from our training data and classify it as a "1" with probability 0.5, our probability of getting the label correct is:

Prob(Correct) = Prob(Class=0)*Prob(Predict=0) +
Prob(Class=1)*Prob(Predict=1)
= 0.5*0.5+0.5*0.5
= 0.5.

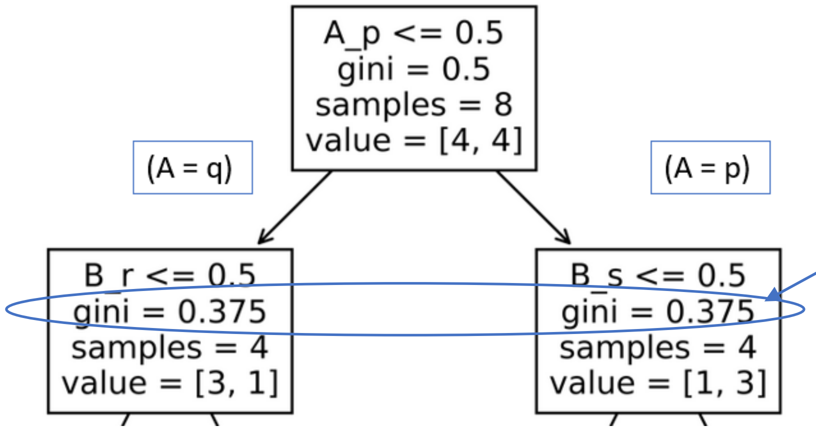
Thus, Prob(Wrong) = Gini = 1-0.5 = 0.5.

(p = fraction of 1's in data)

Gini Impurity $= 2 \cdot p \cdot (1-p)$

How the classification tree algorithm works

- Starting from the "root" node, the algorithm considers splitting the data across each available feature
- For each available feature, the algorithm calculates the average Gini impurity associated with splitting on that feature
- The algorithm chooses the feature with the smallest Gini impurity
- Process repeats...
- First split in above example is across feature A:



After splitting on feature A, the average Gini impurity equals:

(4/8) * 0.375 + (4/8) * 0.375 = 0.375.

What would have the average Gini impurity been if we split on feature B instead?

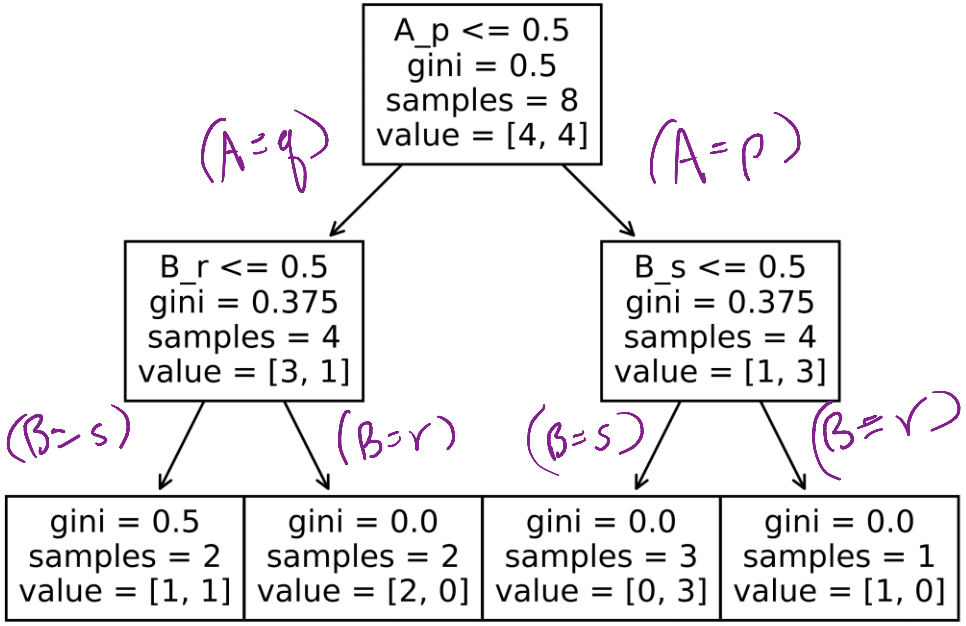
r: $\frac{5}{8}$ [2, 3]

s: $\frac{3}{8}$ [2, 1]

Avg. Gini Impurity $= \frac{7}{15} \approx .467$

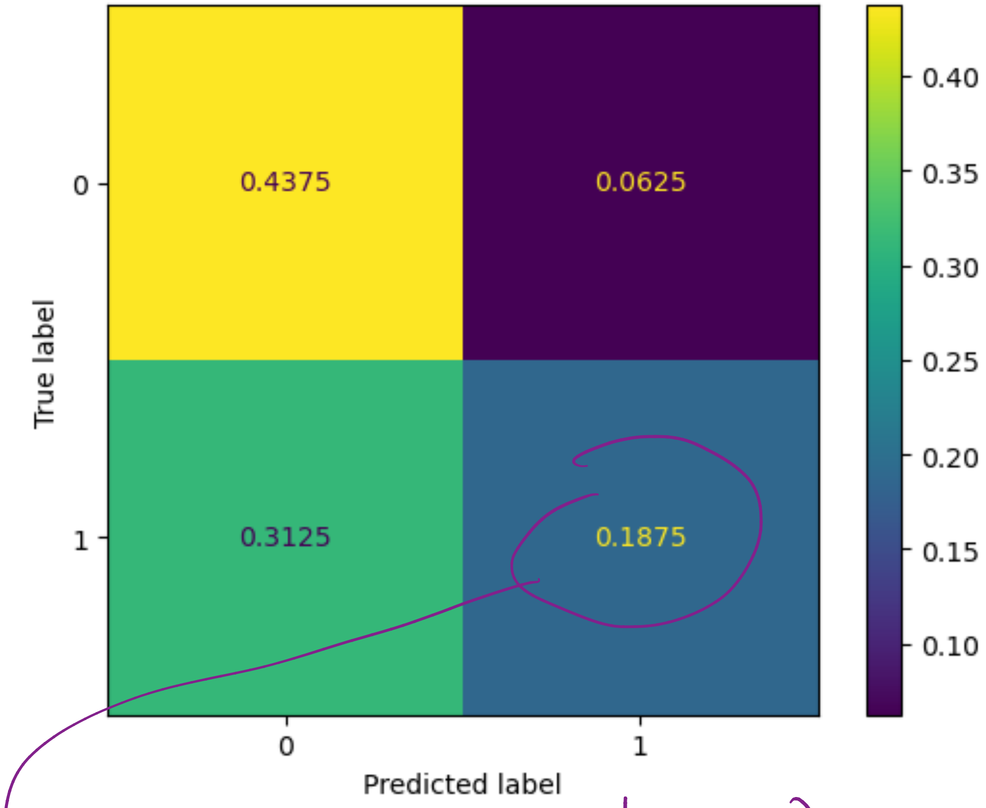
Performance of the fitted tree on test data

Training Data



Predict 0 0 1 0

- Accuracy = 62.5%

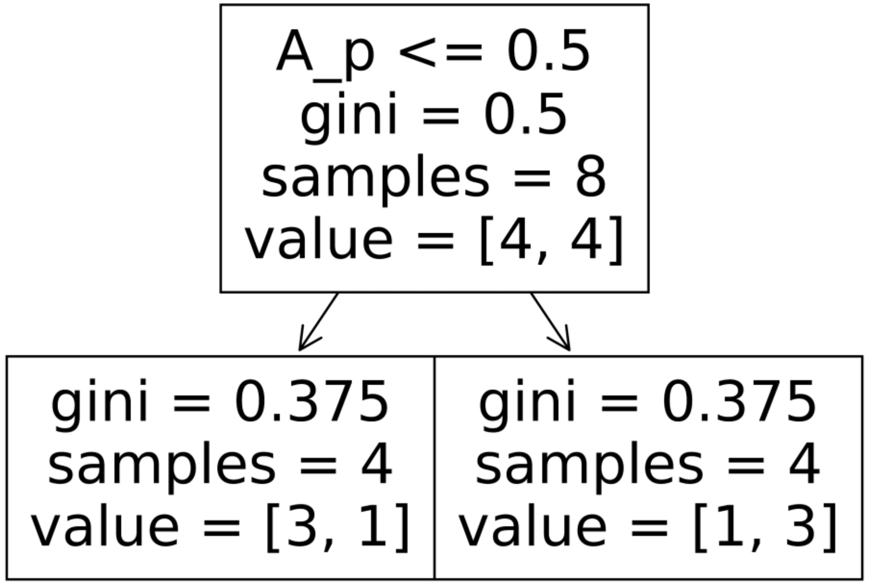


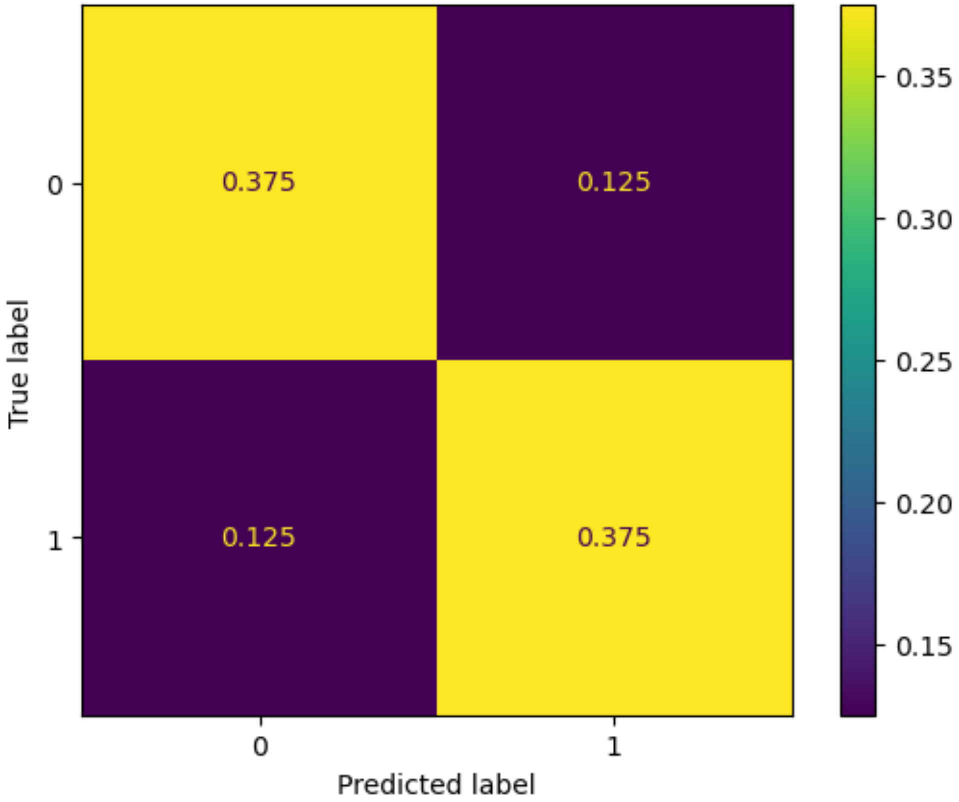
P(Class 1) × P(Predict 1 | Class 1)

= .5 × .75 × .5

= .1875

Fitted tree with the restriction that max depth = 1





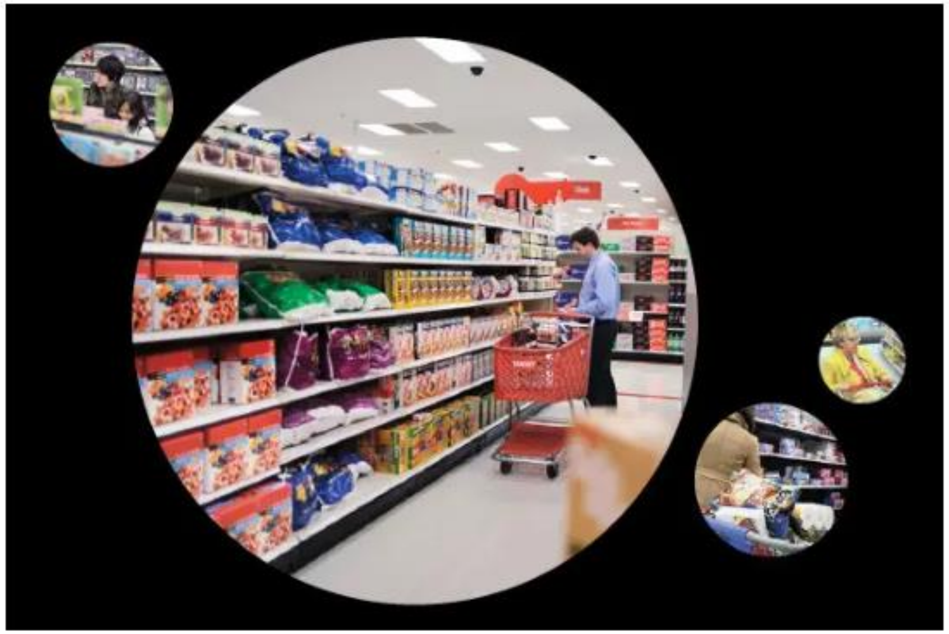
- Accuracy = 75%

Retail analytics: predicting pregnant customers

A dystopian future???

How Companies Learn Your Secrets

Charles Duhigg



Credit...Antonio Bolfo/Reportage for The New York Times

Source: https://www.nytimes.com/2012/02/19/magazine/shopping-habits.html

An example training data set

Data available at:
https://media.wiley.com/product_ancillary/6X/11186614/DOWNLOAD/ch07.zip

See *Data Smart: Using Data Science to Transform Information into Insight* by John W. Foreman

- Top of the training data (csv file):

Male	Female	Home	Apt	Pregnancy Tes	Birth Control	Feminine Hyg	Folic Acid	Prenatal Vitan	Prenatal Yoga	Body Pillow	Ginger Ale	Sea Bands	Stopped buyir	Cigarettes	Smoking Cess:	Stopped buyir	Wine	Maternity Clot	PREGNANT
1	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
1	0	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	1
0	1	1	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
1	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1
0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1
1	0	0	1	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1
1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

- 1000 rows and 20 columns; also a test data file of same size

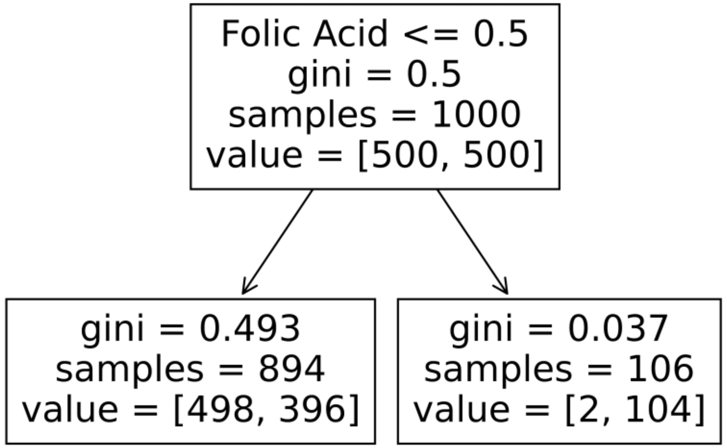
- Means in training data and test data:

```
pregnancy_training_data.mean(axis=0)

Male                      0.401
Female                    0.495
Home                      0.488
Apt                       0.420
Pregnancy Test            0.075
Birth Control             0.140
Feminine Hygiene          0.141
Folic Acid                0.106
Prenatal Vitamins         0.128
Prenatal Yoga             0.018
Body Pillow               0.018
Ginger Ale                0.069
Sea Bands                 0.030
Stopped buying ciggies    0.092
Cigarettes                0.097
Smoking Cessation         0.060
Stopped buying wine       0.130
Wine                      0.123
Maternity Clothes         0.131
PREGNANT                  0.500
dtype: float64
```

```
pregnancy_test_data.mean(axis=0)

Male                      0.424
Female                    0.251
Home                      0.469
Apt                       0.226
Pregnancy Test            0.011
Birth Control             0.216
Feminine Hygiene          0.209
Folic Acid                0.020
Prenatal Vitamins         0.043
Prenatal Yoga             0.005
Body Pillow               0.008
Ginger Ale                0.032
Sea Bands                 0.013
Stopped buying ciggies    0.050
Cigarettes                0.148
Smoking Cessation         0.009
Stopped buying wine       0.080
Wine                      0.202
Maternity Clothes         0.052
PREGNANT                  0.060
dtype: float64
```

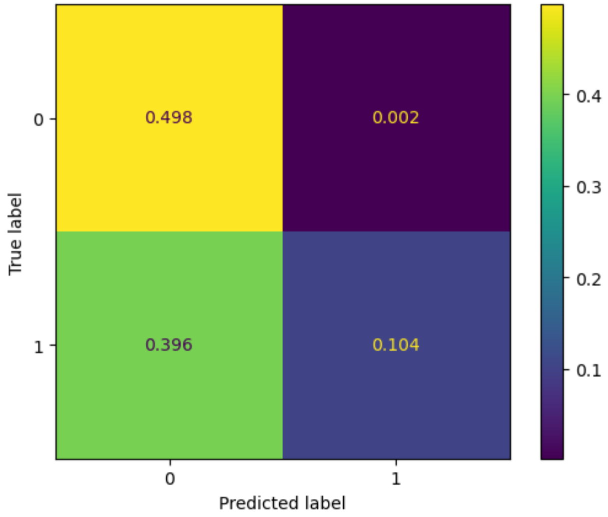
Classification tree with max depth = 1

- A "classification stump:"

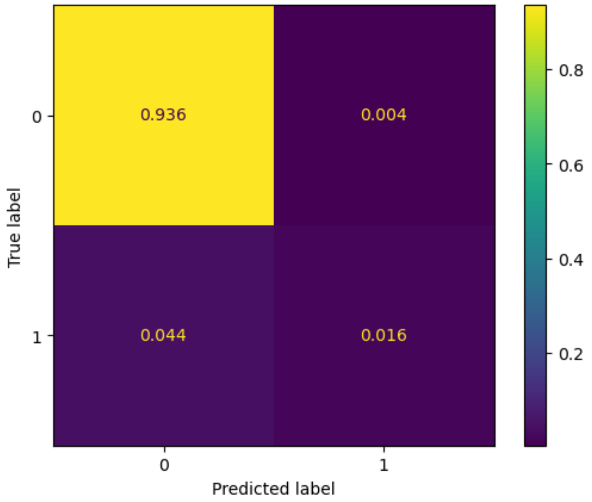


- Confusion matrices:

Training data (accuracy 60.2%):

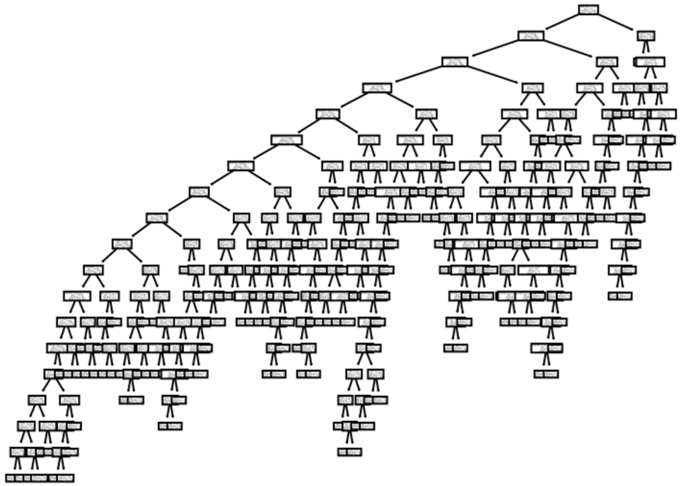


Test (accuracy 95.2%):

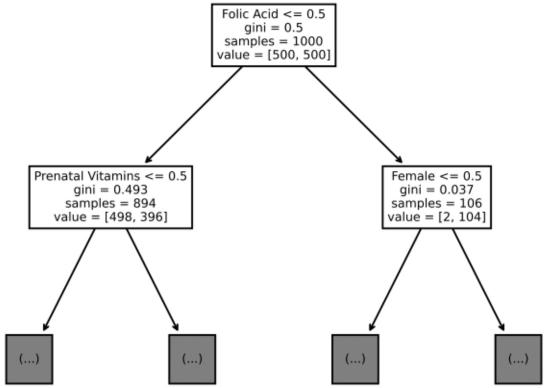


Classification tree with no constraints

- Tree of depth 18!

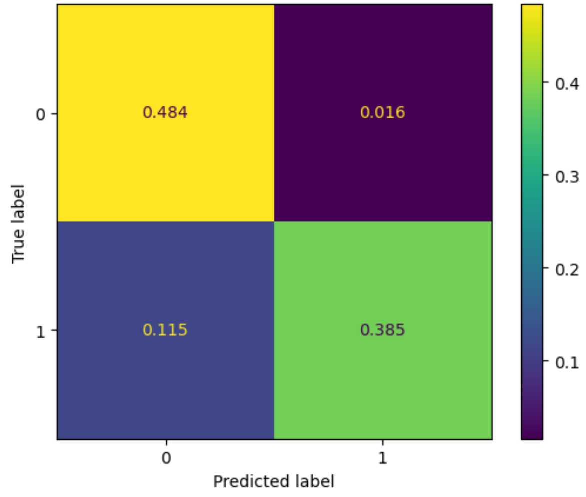


Top of tree:

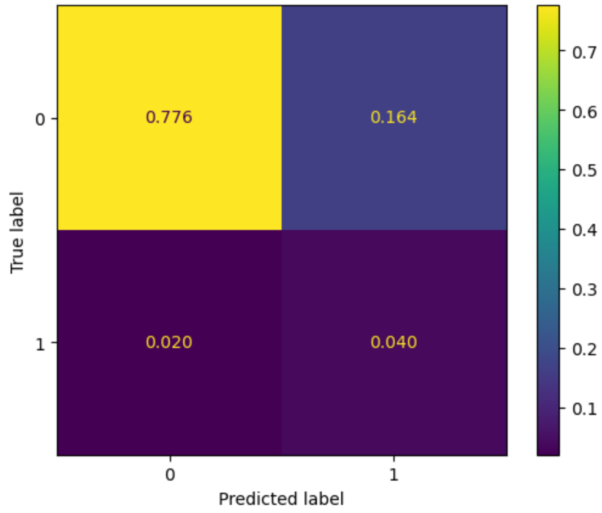


- Confusion matrices:

Training data (accuracy 86.9%):



Test (accuracy 81.6%):



Incorporating an objective ("value function" in HW4)

- The marketing team has conducted research and concludes the following values associated with a classification model are appropriate:

Outcome	Value (scaled $)
True negative	0
False negative	-5
True positive	+10
False positive	-1

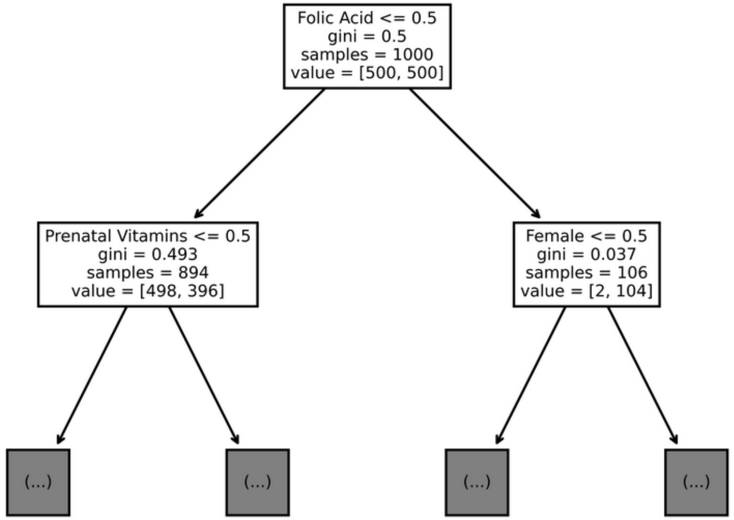
True \ Predicted	0	1
0	TN	FP
1	FN	TP

	0	-1
	-5	10

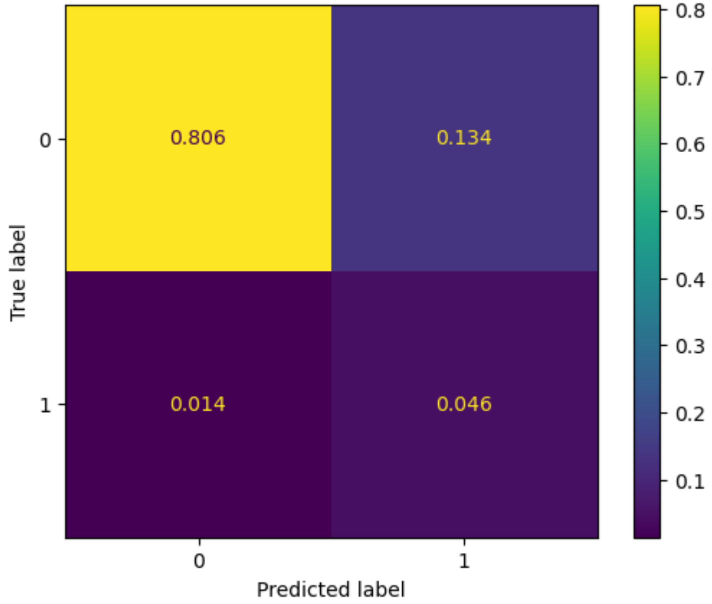
- How can we find an "optimal" tree for this objective?

Optimized tree using GridSearchCV on max_depth

- Results with 10-fold cross-validation
  - Tree of depth 6 is optimal



- Confusion matrix (test data):



Performance on the objective ($):

Model	Expected profit ($)
Optimized tree, training data	2.85
Optimized tree, test data	0.26
Dummy classifier, test data	-0.30
Tree of depth 18, test data	0.14
Tree of depth 1, test data	-0.06

Ensemble methods

- Ensemble methods involve combining multiple individual predictive models into a single model; often used with tree-based models, but applies more broadly
- The ensemble model often outperforms its constituent models
- Used widely and gained fame in the [Netflix Prize Competitions]
- Two main forms of ensemble models: aggregation and boosting

If you ask 100 people to run a 100-meter race, the average time will not be better than the time of the fastest runners. It will be worse - a mediocre time. But ask 100 people to answer a question or solve a problem, and the average number will often be as least as good as the answer of the smartest member. With most things, the average is mediocrity. With decision making, it's often excellence.

Diversity and independence are important because the best collective decisions are the product of disagreement and contest, not consensus or compromise.

— James Surowiecki, *The Wisdom of Crowds*

“Bagged” models are a widely used ensemble model

For training:

1. Randomly sample rows of the training data (typically with replacement) to generate “new” training data that is similar to the original training data
2. Build a model (often a classification tree) on the new data set
3. Repeat

For classification:

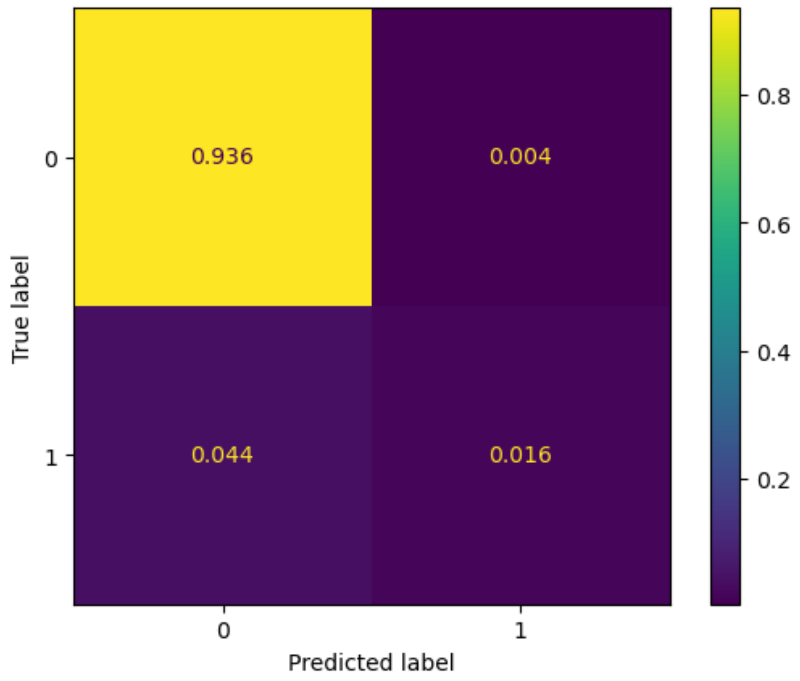
- Use the majority vote of all the individual models

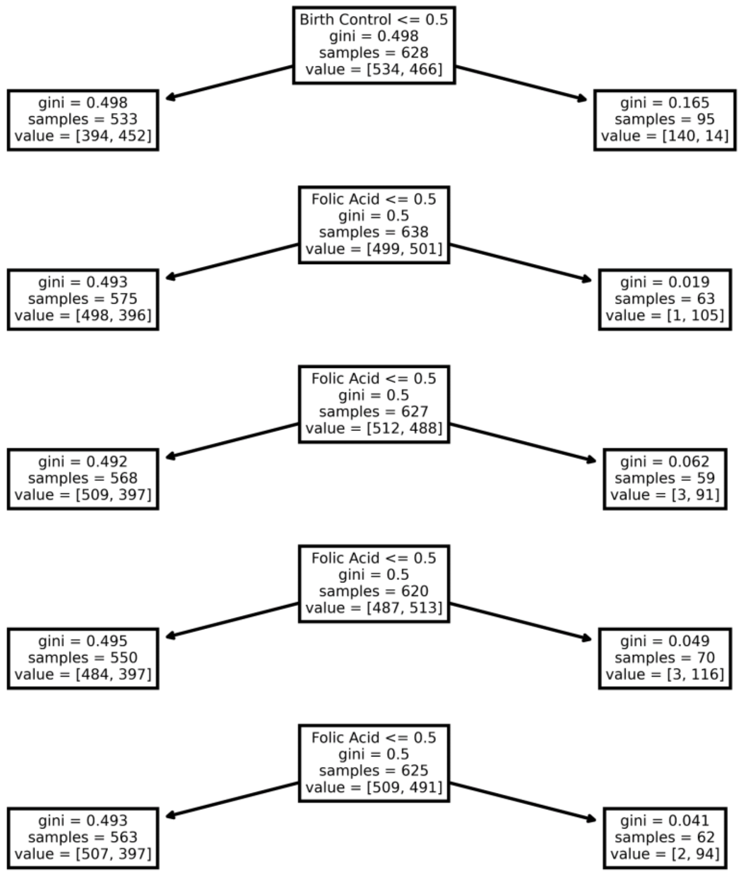
- Advantages:
  - Conceptually and computationally simple
  - *Many* tuning parameters available (how to sample from data, how deep/complex each tree is, how many features to use, …)
- Disadvantages:
  - Randomly sampling rows of data alone may not encourage diversity across individual models

A bagged model on the pregnancy training data

- A bag of 5 classification stumps

Confusion matrix on test data (95.2% accuracy):



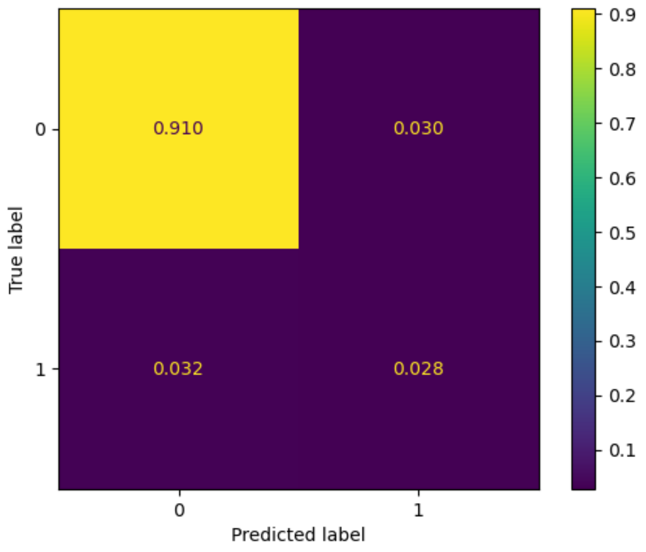


Model	Expected profit ($)
Bagged model	-0.06

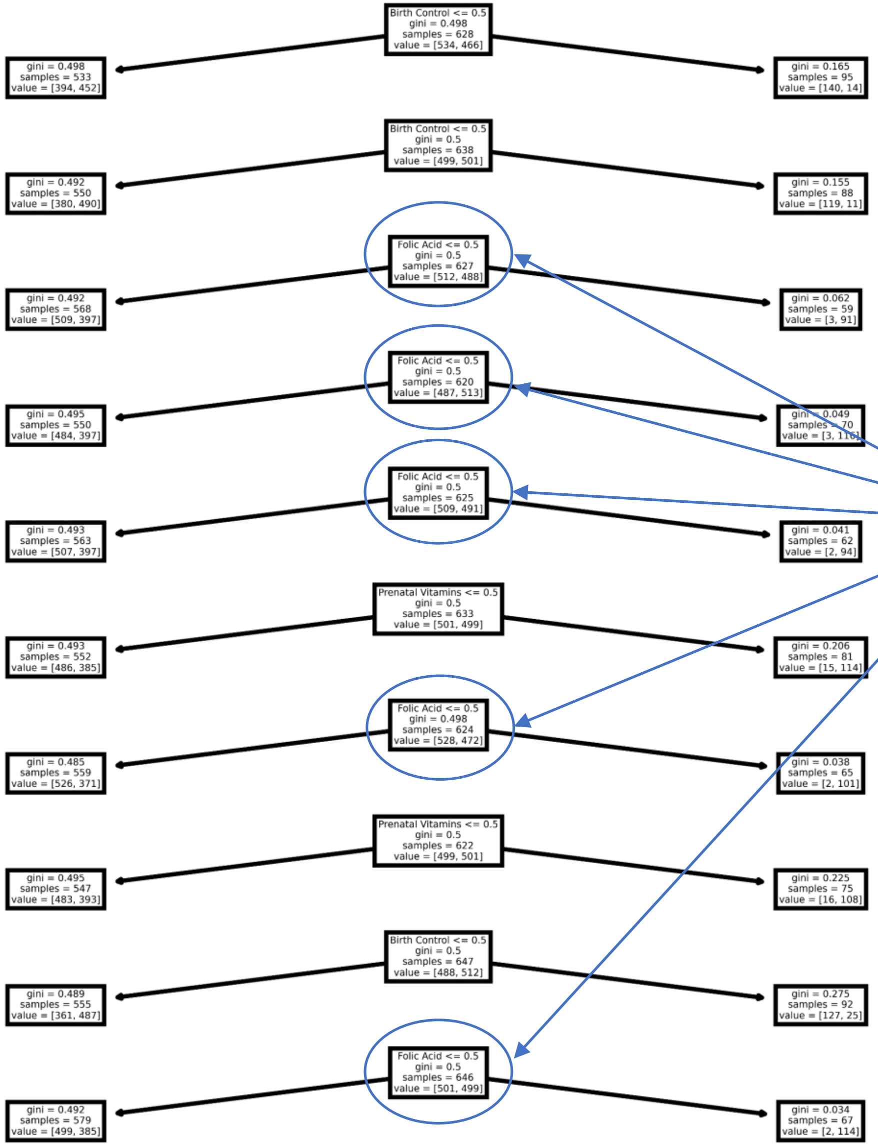
Random forests = bagging with randomly selected features

- A random forest of 10 stumps

Confusion matrix on test data (93.8% accuracy)



Model	Expected profit ($)
Random forest	0.09



Note: Each model chooses max_features randomly for splitting (default = square root of number of features)

Folic acid

Boosting is an *iterative* ensemble method

- Uses a weighted combination of a fixed collection of “weak learners;” often this is a classification stump for each feature (e.g., “Folic Acid,” “Female,” “Wine” etc.)
- Every sample in the training data has a “weight,” updated over iterations (starts at 1/N)

Adaptive Boosting Algorithm (AdaBoost)

weight on L_k

- At each iteration k:
    1. Pick the weak learner (call this L_k) with the lowest weighted error;
    2. Calculate the quantity α_k which equals $\frac{1}{2}\log\frac{1-Weighted\ Error}{Weighted\ Error}$
    3. Update training data weights: weights multiplied up for samples in which L_k makes mistakes, and down otherwise

This choice for α_k minimizes an particular exponential loss function at the current iteration. Other loss functions possible (“gradient boosting” or “XGboost” can accommodate these).

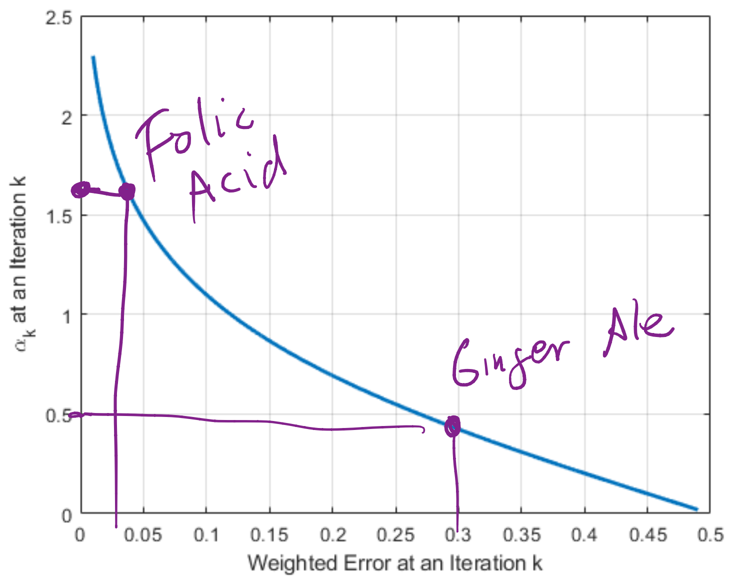
- Repeat for a number of iterations K (or until weighted error is high enough)

- Boosting classification rule:

If $\alpha_1 \cdot Prediction(L_1) + \cdots + \alpha_K \cdot Prediction(L_K)$ is positive, predict “1” otherwise predict “0”

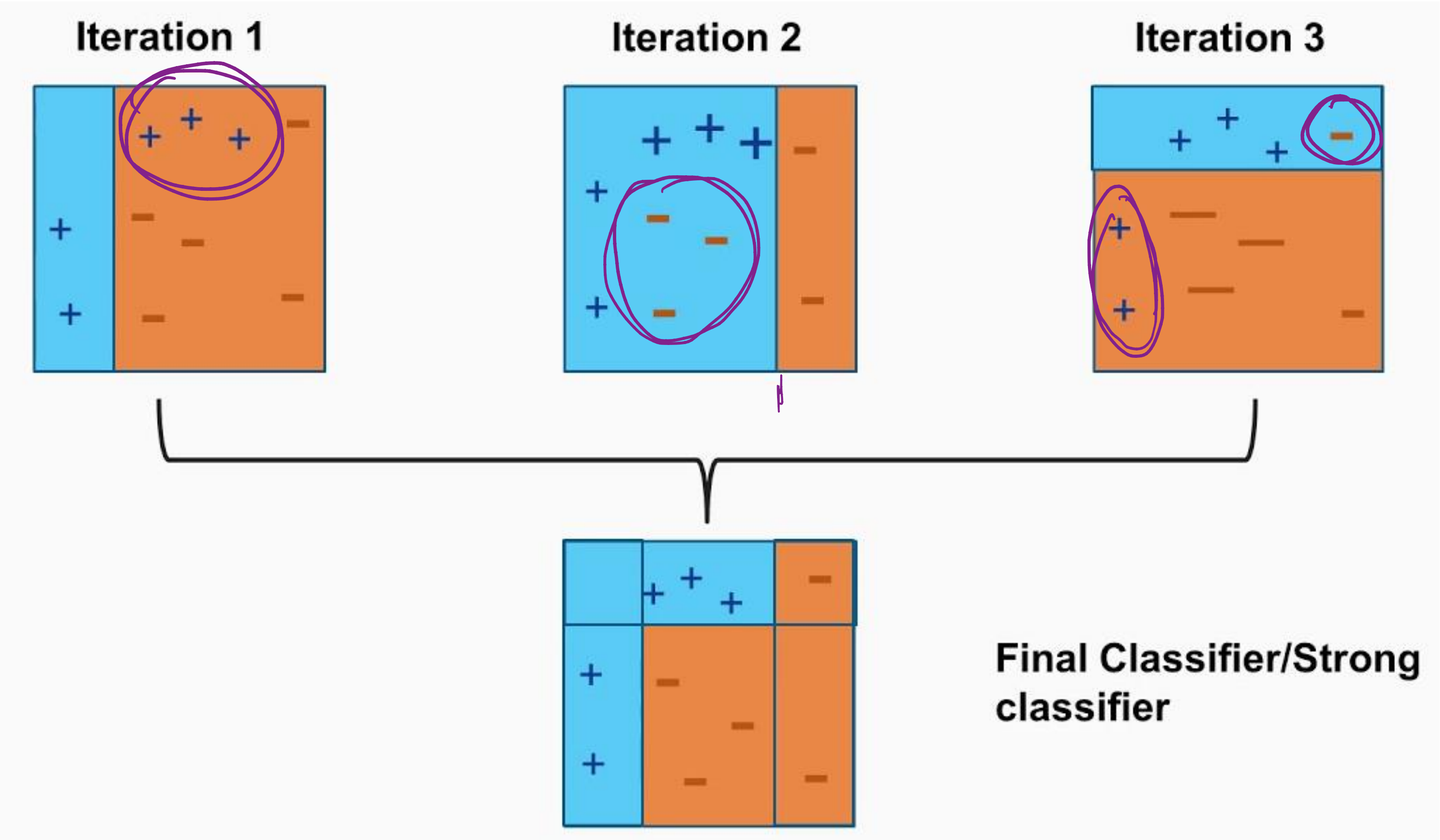
Progression of the AdaBoost algorithm

- Weighted error and α_k are inversely related:



- At early iterations:
  - The winning weak learner will tend to have low weighted error and high α_k (and hence high influence in final classifier)
- At later iterations:
  - The training data is highly weighted towards "tough" points to classify/outliers, and the winning weak learner has higher weighted error and low α_k (and hence low influence in final classifier)
- The "learning rate" is a tunable parameter – scales α_k down by a constant factor at every iteration to dampen the effect of outliers

A illustration of boosting from module 4



GridSearchCV over AdaBoosted models on pregnancy data

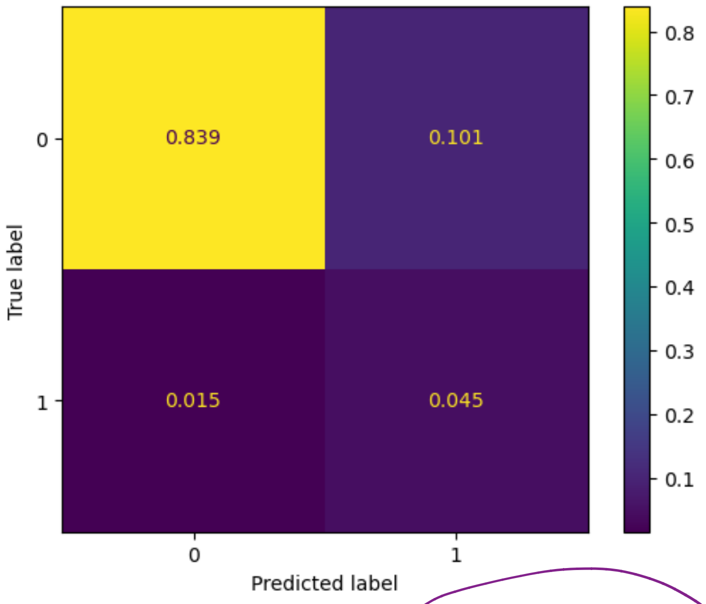
- Grid search on n_estimators (iterations) and learning_rate with the "value function" as the scorer
- Optimal boosted model combines 18 features:

α_3

Iteration	Winning Feature
1	Folic Acid
2	Birth Control
3	Prenatal Vitamins
4	Maternity Clothes
5	Feminine Hygiene
6	Wine
7	Pregnancy Test
8	Stopped buying wine
9	Ginger Ale
10	Stopped buying ciggies
11	Cigarettes
12	Smoking Cessation
13	Prenatal Vitamins
14	Birth Control
15	Prenatal Yoga
16	Feminine Hygiene
17	Maternity Clothes
18	Male

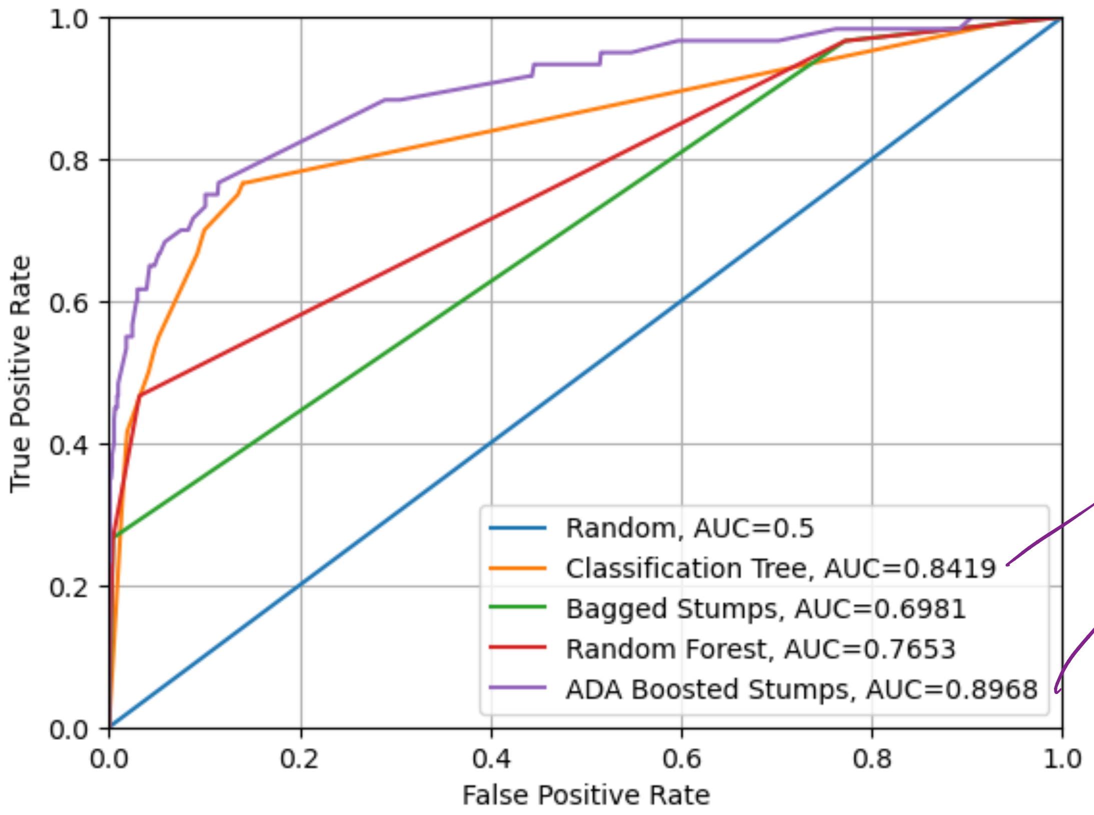
α_{13}

Confusion matrix on test data (88.4% accuracy)



Model	Expected profit ($)
AdaBoost with grid search	0.27

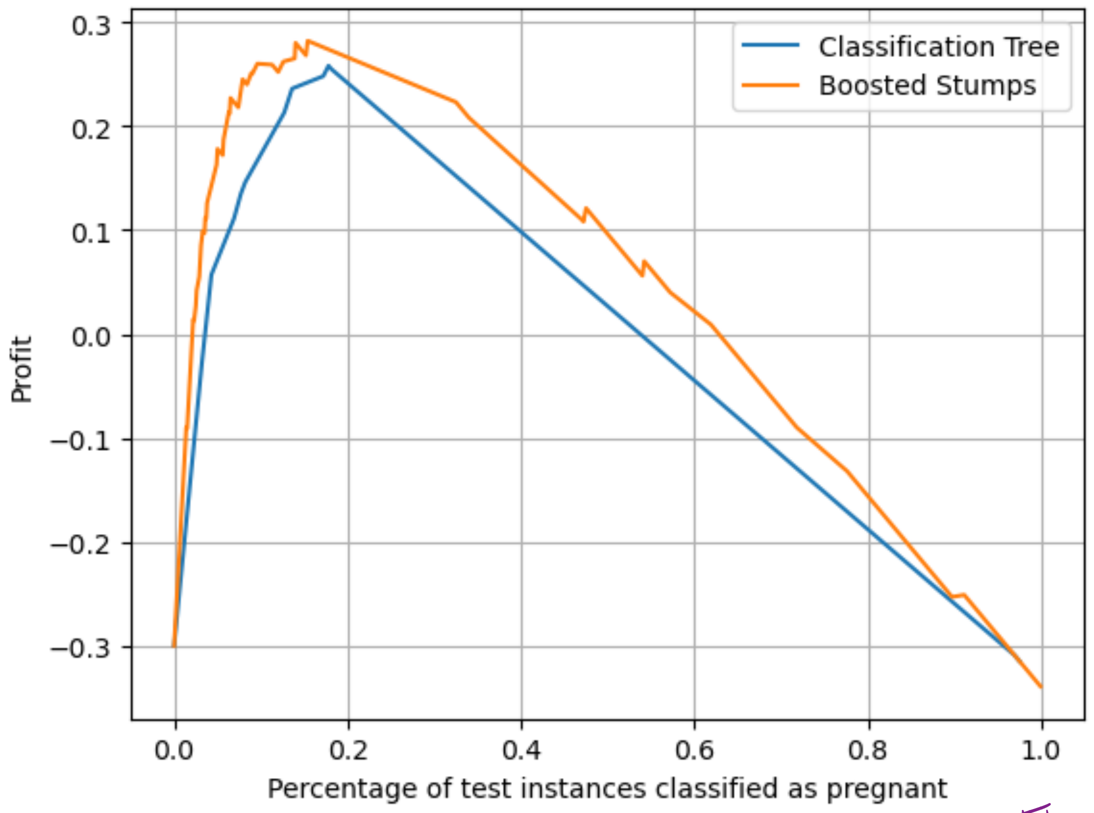
Comparing ROC curves of the models



optimized

Profit curves provide another useful visualization

- Here we vary the threshold for classifying a customer as pregnant from high to low:



Thresh. High

Thresh. Low

Summary

- Classification tree models are widely used!
  - Provide a nonlinear classification rule based on progressively dividing data across features
  - Also can be used in regression (predict the average at each leaf)
  - Potential hazard: prone to overfitting!

- Ensemble methods involve combining models – also widely used!
  - Bagging: randomly select from training data and aggregate resulting models
  - Random forest: bagging with randomly selected features
  - Boosting: using weighted combinations of “weak learners”

Ensemble models are used widely!

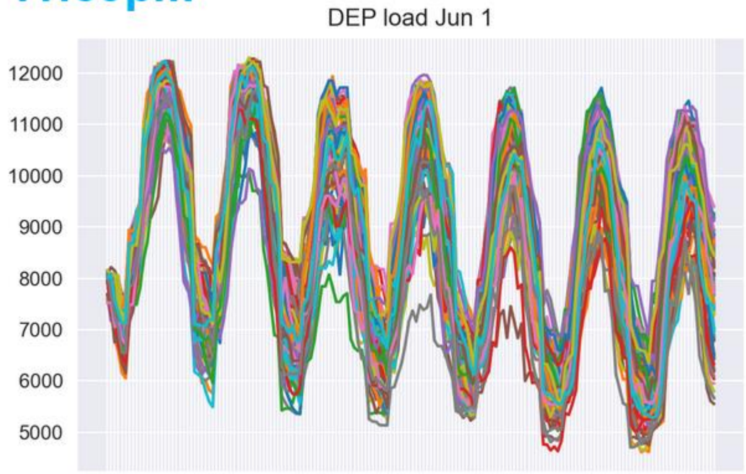
- Predictive modeling for electricity demand and solar (PV) production for Duke Energy as part of the GRACE project (A Grid that's Risk-Aware for Clean Electricity):

Ensembles of weather data ensembles for load and PV production

Method	Value
Train- Test size	70% - 30%
Cross Validation	Yes – 5 fold

Model	Parameters	R-Squared	MAE	MAPE
Linear Regression		0.42	1557.71	12.82
Random Forest		0.94	475.19	3.97
Gradient Boosting	'subsample': 0.8 'n_estimators': 4000 'min_samples_split': 100 'min_samples_leaf': 4 'max_features': 5 'max_depth': 200 'learning_rate': 0.005	**0.94**	**465.96**	**3.95**
Multilayer Perceptron	'Hidden Layers': 1 'Neurons in HL': 1152	0.93	504.48	4.24

500 traces of load from June 1, 2019 12:00am – June 7, 2019 11:59pm



Looking ahead to next time (Class 5)

- Homework 4 due at 11:59pm on Monday
  - Main goals: practice your understanding of classification trees and ensemble methods
  - TA support available over the weekend!
    - Parallel sessions available on Monday: see Canvas for details

- Class 5:
  - Unsupervised learning
    - Clustering
    - Dimensionality reduction

- Final exam (April 6-April 15):
  - Multiple choice, ~2 hours, conceptual only, flexible window, open book, individual work only

A joke for the weekend…