# DERIVING KNOWLEDGE FROM DATA AT SCALE

Lecture 02

Drew Bryant, Ph.D

# Lecture overview

- Review ideas from tidy data
- Practical data exploration
- Intro to tree-based models
- Evaluating and comparing models
- Assignment and Capstone overview

# Tidy data

in practice

# Tidy data

- Let's try it together in a notebook

# Decision trees

How do they grow?

# Decision trees

- Basic tree-building algorithm:
  - Split node to (optimally) decrease entropy
  - Recursively repeat until leaves contain a single example
- Tree-based prediction algorithm:
  - Follow the attribute split decisions until you hit a leaf node
  - The predicted label is the label of the leaf example, or majority vote of the leaf (if >1 example in leaf node)

# Entropy

Computing entropy over a set of examples with k classes

$$H(S) = -\Sigma_{i=1}^{k} p_i log_2 p_i$$

# Information gain

Computing info gain over a set of examples with k classes

$$Gain(S, A) = H(S) - \Sigma_{v \in A}|S_v|/|S| * H(S_v)$$

$$H(S) = -\Sigma_{i=1}^{k} p_i log_2 p_i$$

# Should we play golf?

| | outlook | temperature | humidity | windy | play? |
|---|---|---|---|---|---|
| **0** | sunny | hot | high | False | N |
| **1** | sunny | hot | high | True | N |
| **2** | overcast | hot | high | False | Y |
| **3** | rain | mild | high | False | Y |
| **4** | rain | cool | normal | False | Y |
| **5** | rain | cool | normal | True | N |

# Building the tree

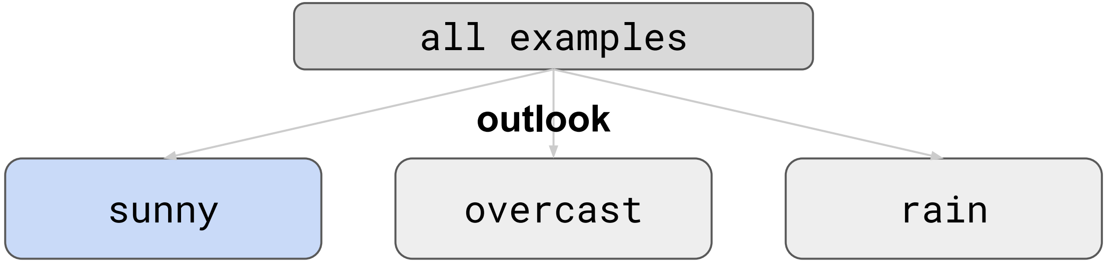
all examples

```
argmax info_gain(examples, attribute)

attributes: outlook, temperature, humidity, windy
```

$$Gain(S, A) = H(S) - \Sigma_{v \in A} |S_v| / |S| * H(S_v)$$

# Building the tree



all examples

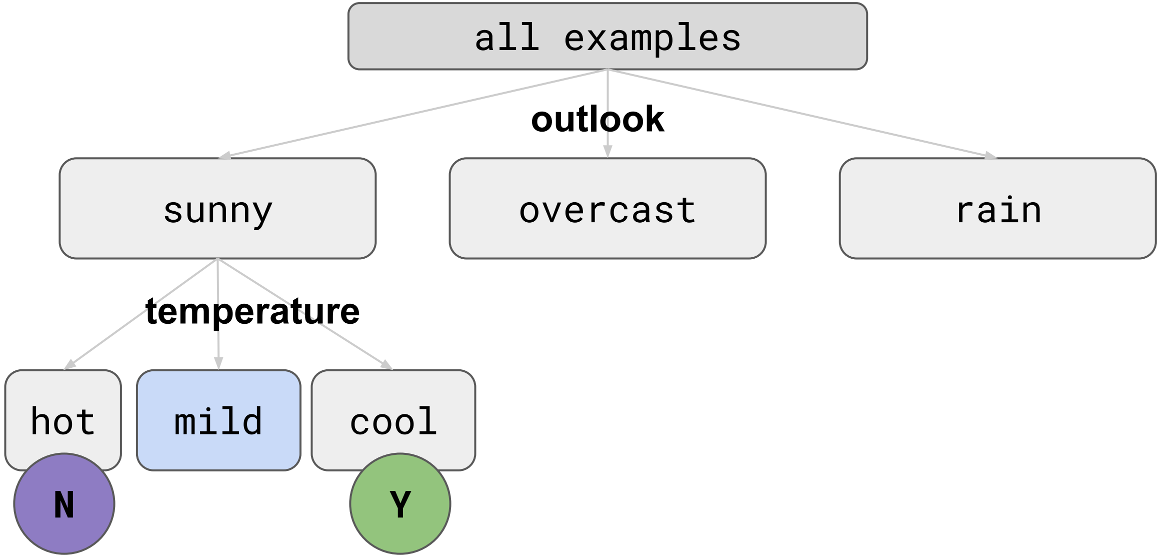
outlook

sunny

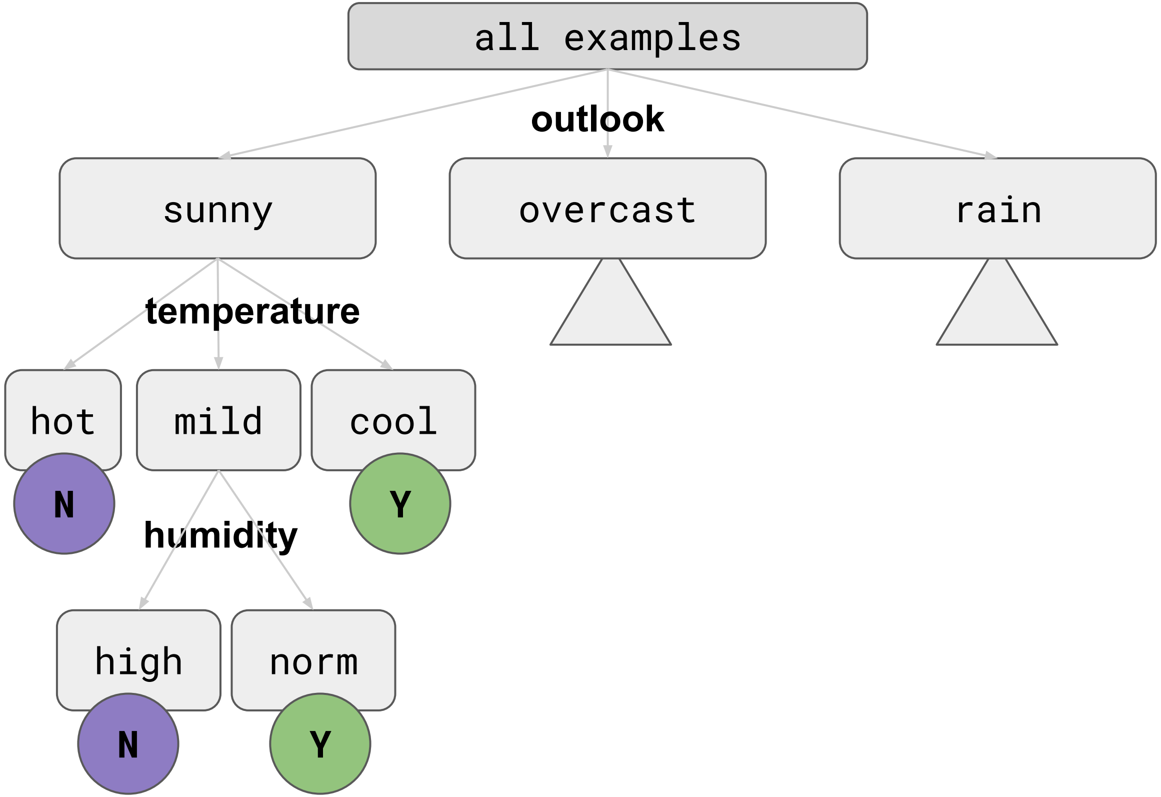
overcast

rain

# Building the tree



# Building the tree



# Let's try it

Jumping over to Jupyter...

# Scaling up

We're going to need a bigger dataset

# Level-wise training

- Select the splits for all nodes at the same level of the tree simultaneously
- Reduces the number of passes over the dataset exponentially: we make one pass for each level, rather than one pass for each node in the tree
- Leads to significant savings in I/O, computation and communication.

# Approximate quantiles

- Single machine implementations typically use sorted unique feature values for continuous features as split candidates for the best split calculation
- Finding sorted unique values is an expensive operation over a distributed dataset
- Spark MLLib uses quantiles for each feature as split candidates: tradeoff for improving decision tree performance without significant loss of accuracy.

# Bin-wise computation

- Best split computation discretizes features into bins
- Those bins are used for computing sufficient statistics for splitting
- Precompute the binned representations of each instance, saving computation on each iteration

# Building models

Theory and practice

# Model building flow

- Define objective
- Access and understand the data
- Pre-process the data
- Feature and/or target construction
- Train/test split
- Feature selection
- Model training
- Model evaluation
- Model assessment and comparison

# Evaluating models

- TPR, FPR, acceptance threshold
- Accuracy, sensitivity, specificity
- Precision/Recall
- AUC (binary class)
- Confusion matrix (multi-class)
- Avoiding leakage

# Comparing tree-based models

- Build N different trees (randomized feature selection vs "optimal" for example)
- Compare the perf of these trees
- Let's try it!

# Deploying models

Productionalize your predictions

# Model deployment

- How to package your model for deployment
- Using your model for batch prediction
- Publishing your model as a prediction API endpoint for online inference
- Next time: work through examples of this in the cloud (AWS, GCP)

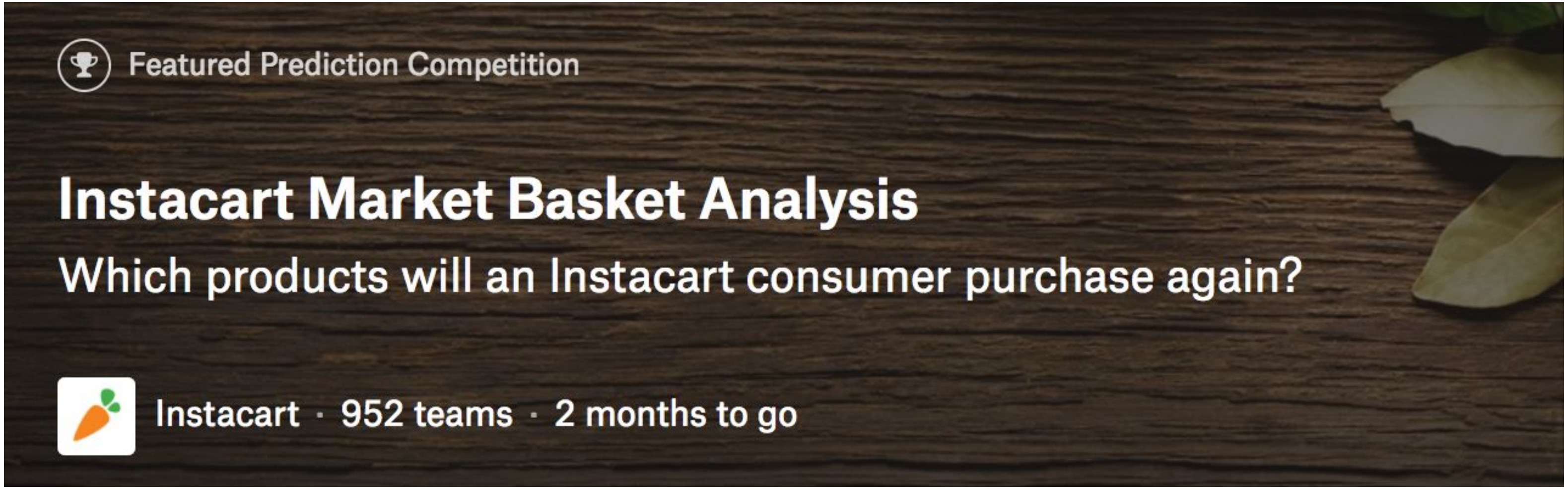
# Model monitoring

- Monitor input (queries) and output (prediction) distributions for skew relative to training set
- Alert and re-train as necessary (or on a schedule)

# Capstone project

Details...

# Capstone project: Instacart



Featured Prediction Competition

**Instacart Market Basket Analysis**

Which products will an Instacart consumer purchase again?

Instacart · 952 teams · 2 months to go

https://www.kaggle.com/c/instacart-market-basket-analysis

# First assignment: wine model

- See files uploaded to resource section
- We'll walk through getting started a bit

# Questions?