DePaul University
College of Computing and Digital Media

Casey Bennett, PhD

Sept.25, 2019

This Week

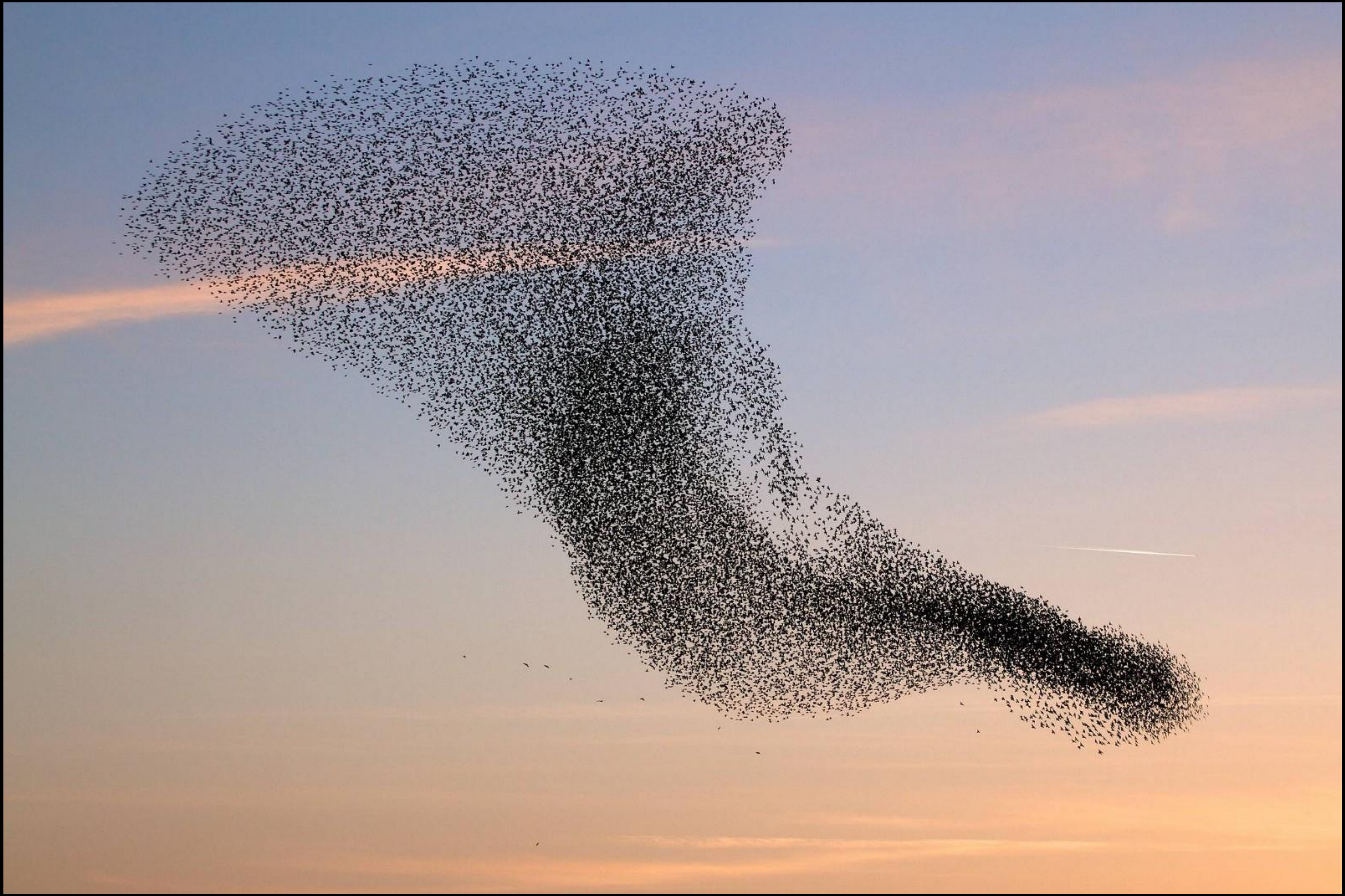
1) HW1 is due *today*
2) HW2 releases that same night

3) **Week of Oct.16 Class will be cancelled**
   - Slides will be posted for the week
   - I will include stuff on PCA and Feature Selection the week before and after
   - There will still be a paper to read, and online discussion that week

Random Forest and Bagging

https://pollev.com/caseybennett801

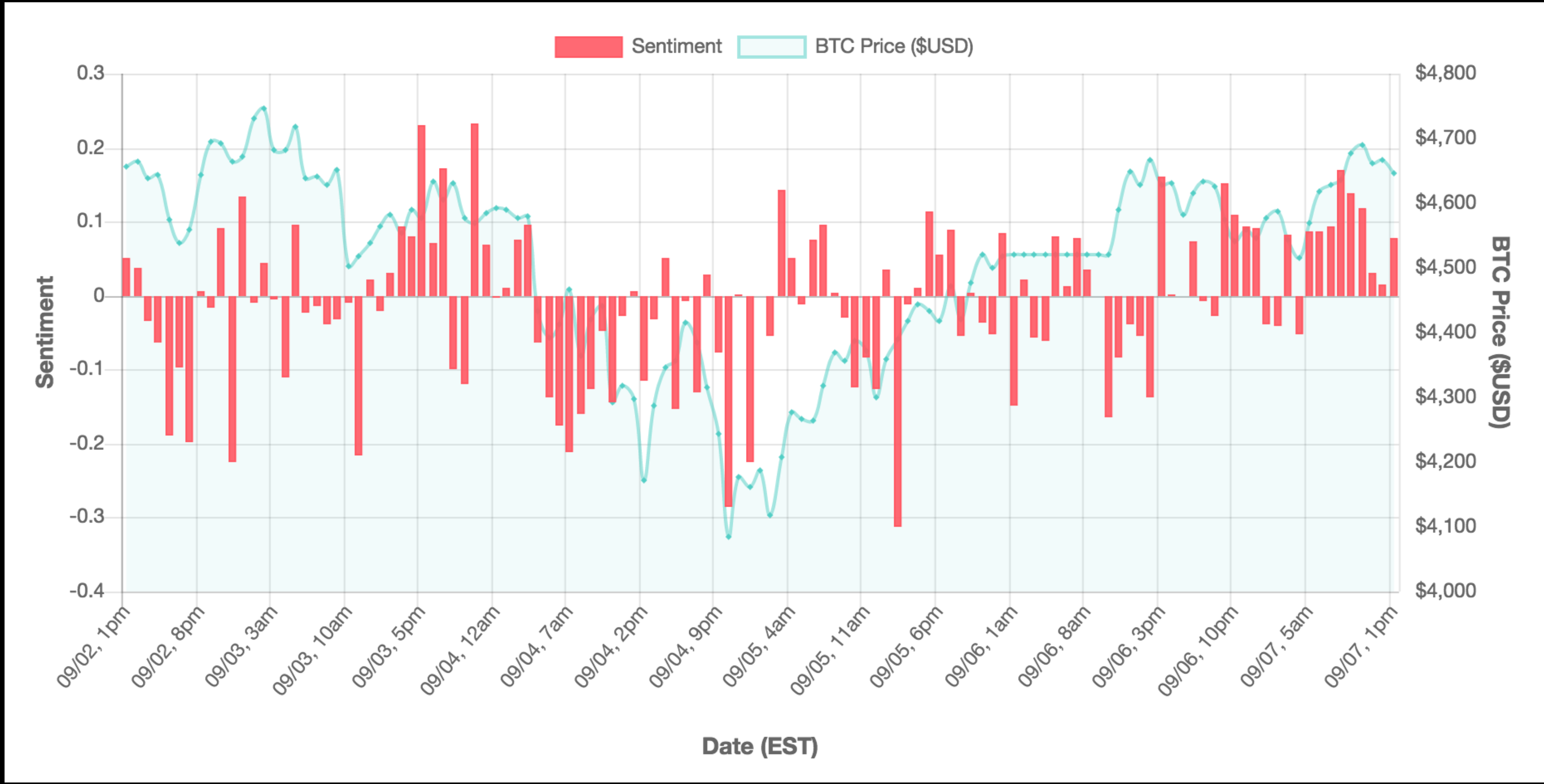
or text "caseybennett801" to 37607



Fish Schooling Video

So how do a bunch of *dumb* individuals (e.g. fish) exhibit such *intelligent* collective behavior (e.g. schooling to evade predators)?

Collective Intelligence



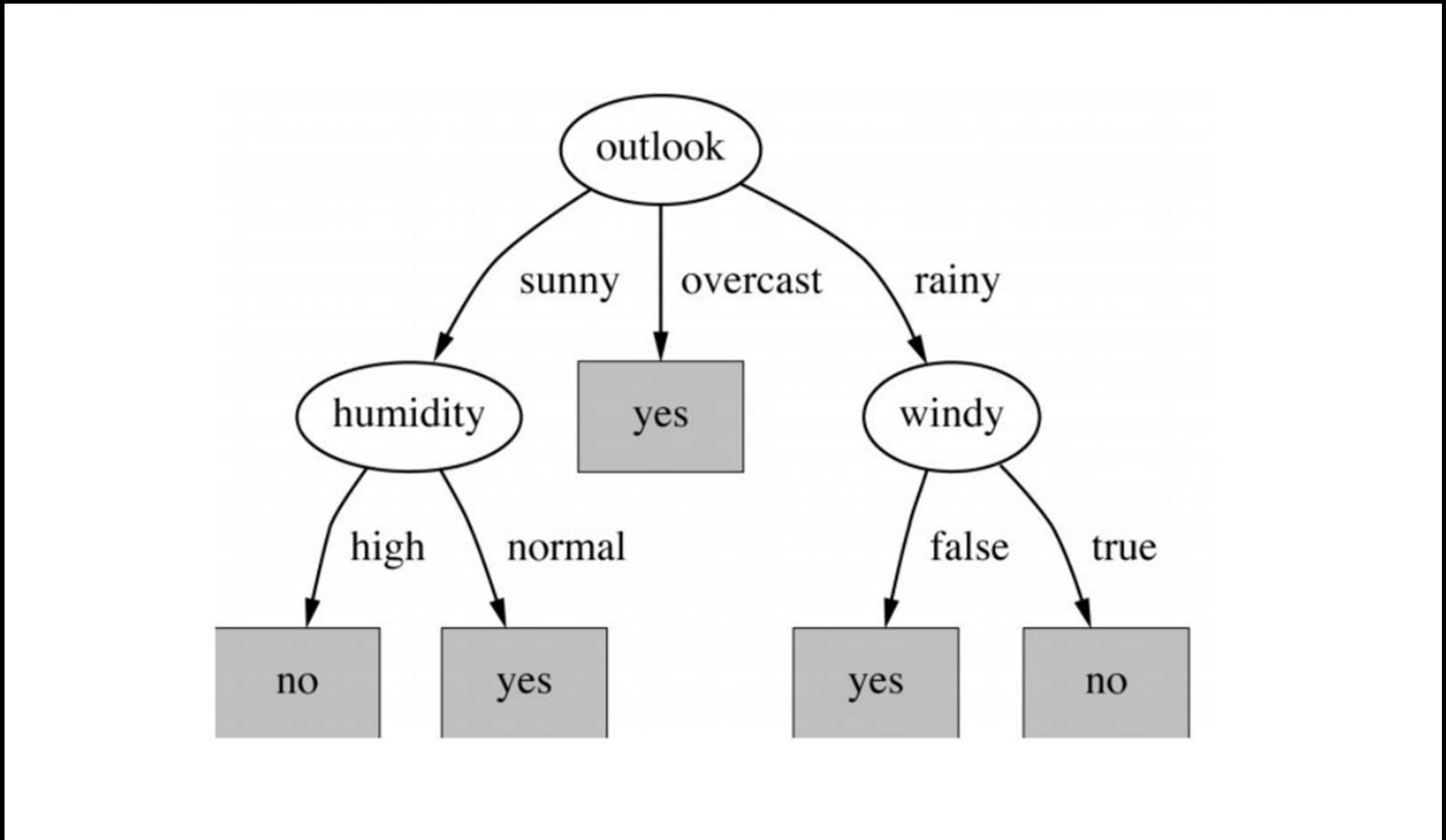
Sentiment
BTC Price ($USD)
0.3
0.2
0.1
0
-0.1
-0.2
-0.3
-0.4
$4,800
$4,700
$4,600
$4,500
$4,400
$4,300
$4,200
$4,100
$4,000
BTC Price ($USD)
09/02, 1pm
09/02, 8pm
09/03, 3am
09/03, 10am
09/03, 5pm
09/04, 12am
09/04, 7am
09/04, 2pm
09/04, 9pm
09/05, 4am
09/05, 11am
09/05, 6pm
09/06, 1am
09/06, 8am
09/06, 3pm
09/06, 10pm
09/07, 5am
09/07, 1pm
Date (EST)

Everyone who votes is an idiot, but markets are made up of the same people. Even if they are horribly misinformed, markets as a whole make smart decisions … how?

- Markets can make smart decisions, even if the individuals within aren’t so bright, or lack info
- Ensemble Learning (e.g. Voting)

Random Forests and Bagging

- Bagging = Bootstrap Aggregating



Decision Trees

- **Advantages**
  - ➢ Take a dataset, create a “tree”, split at each branch based on some metric (e.g. information gain, gini index)
  - ➢ Decision Trees are easy to explain to lay-people (e.g. your boss), and provide a digestible visual representation of their output

Decision Trees

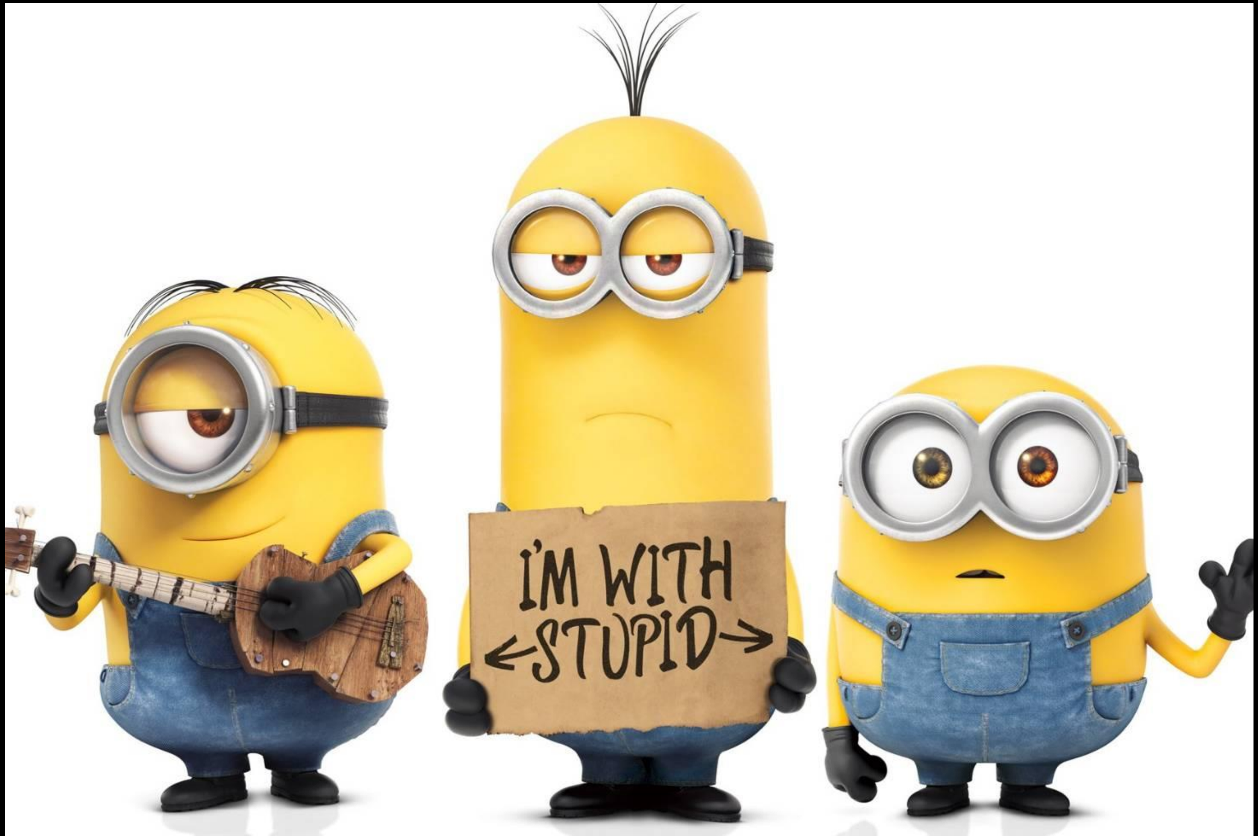
- **Disadvantages**
  - The fundamental problem is that an individual decision tree is very sensitive to the dataset being used
  - If you take different slices of the data, or a different dataset, you get a different answer
  - But if you build multiple trees on the same dataset, you just get a bunch of duplicate trees
  - In short, they don't *generalize* well

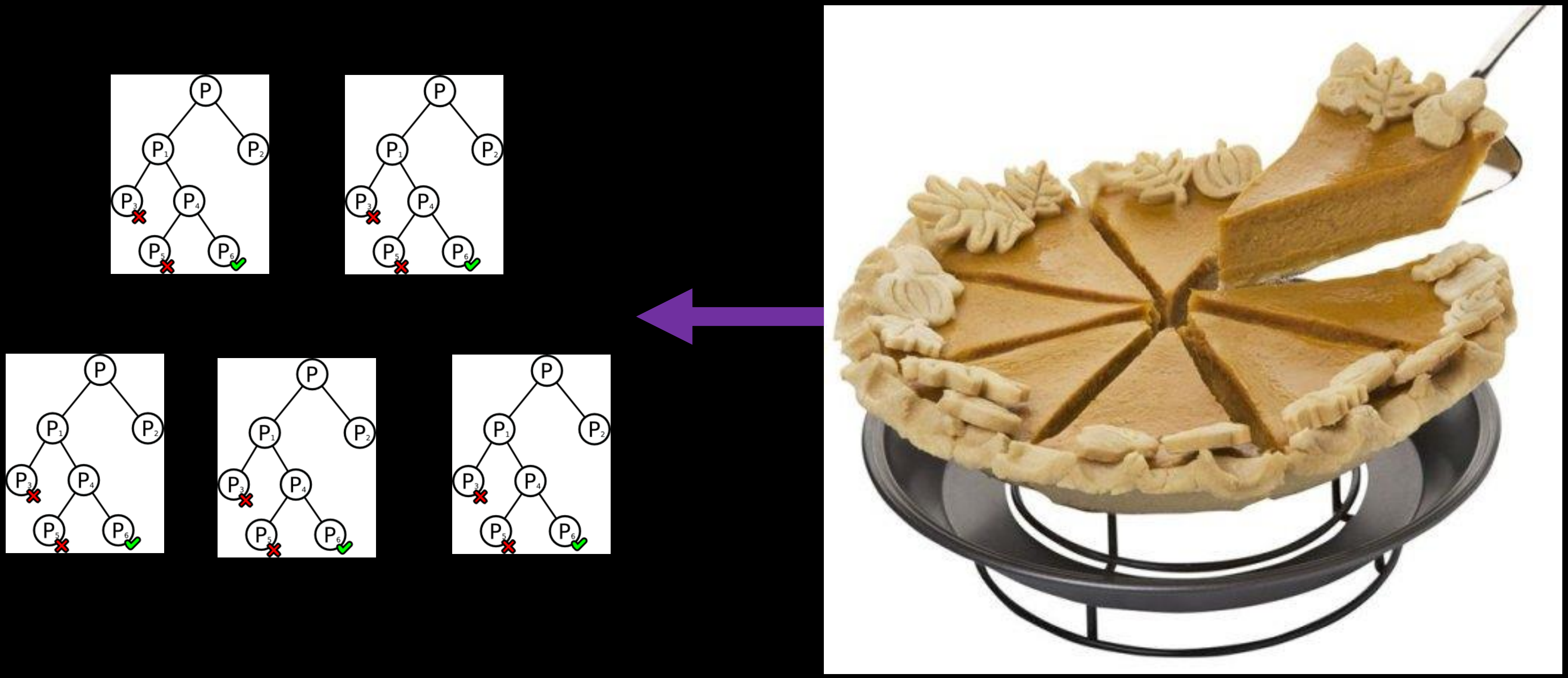
Intuition

- A simple approach to try to deal with this is *pruning*, to reduce overfitting
- A **better approach** though would be to build an ensemble of trees, but how do we do that on a single dataset that without just building a bunch of highly correlated trees?

How can we build an ensemble of trees on a single dataset, without just building a bunch of the same exact tree?

Different kinds of idiots working together





- We can accomplish this by giving different individuals different parts of the information
- e.g. different subsets of data or variables
- Random Forests (bagging)

Intuition

- **Bagging**
  - Build multiple trees using random subsets of *data*
  - Bootstrapping = sampling with replacement
  - Average prediction across ensemble

- **Random Forest**
  - Choose a random subset of *features* at each split point of each tree
  - Builds on bagging
  - Resulting trees are even less correlated than bagging
  - Number of trees to build (more trees “might” improve results)

Intuition

- **Decision Trees**
  - The fundamental problem is that an individual decision tree is very sensitive to the dataset being used
- **Bagging**
  - Build multiple trees using random subsets of *data*
- **Random Forest**
  - Choose a random subset of *features* at each split point of each tree

A couple side notes

- Bagging can be done with any sort of classifier
  - In practice, this is often done *after the fact*, e.g. when people are trying to boost performance of their existing classifier model from 80% to 85%

- In Python Scikit, bagging is also how you can implement random subspaces
  - For random subspaces, each classifier (e.g. tree) is fed a random subset of features, rather than at each split

- In Scikit, two major options for how you choose each branch split:
  - **Gini Index** – measure of "unequal" distribution in some variable (e.g. income), https://en.wikipedia.org/wiki/Gini_coefficient
  - **Entropy** – measure of information, order and "chaos", https://en.wikipedia.org/wiki/Entropy_(information_theory)

Random Forest Pseudocode

1) Load Data
2) Split data into cross-validation folds (or test/training split)
3) Select random subset of data (bagging) for new tree
4) Select random subset of features for first split (typically sqrt[k] or k/3)
5) Evaluate features from subset using metric (e.g. info gain, gini index, sum of squared errors)
6) Pick best feature and create split
7) Recurse down the tree, repeating steps #4-6, until minimum leaf size
8) Go back to step #3 and start with new tree, until maximum tree number
9) Combine tree predictions (e.g. mean, voting)
10) Calculate performance (Accuracy, AUC, RMSE, etc.)

Code Implementation

- Raw Python Code
- Scikit method
- Spark method

Code Implementation

#SciKit Random Forest
```
RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2,
                       min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features='auto’,
                       max_leaf_nodes=None, min_impurity_decrease=0.0, min_impurity_split=None,
                       bootstrap=True, oob_score=False, n_jobs=None, random_state=None,
                       verbose=0, warm_start=False, class_weight=None)
```

#Spark Random Forest
```
RandomForestClassifier(labelCol="idxLabel", featuresCol="idxFeatures", numTrees=100, impurity=‘gini’,
                       maxDepth=5, minInstancesPerNode=1, featureSubsetStrategy="auto")
```

- There are also *Regressor* versions for RandomForest in both Scikit and Spark, for when you have a continuous numerical target variable you are trying to predict

Code Implementation

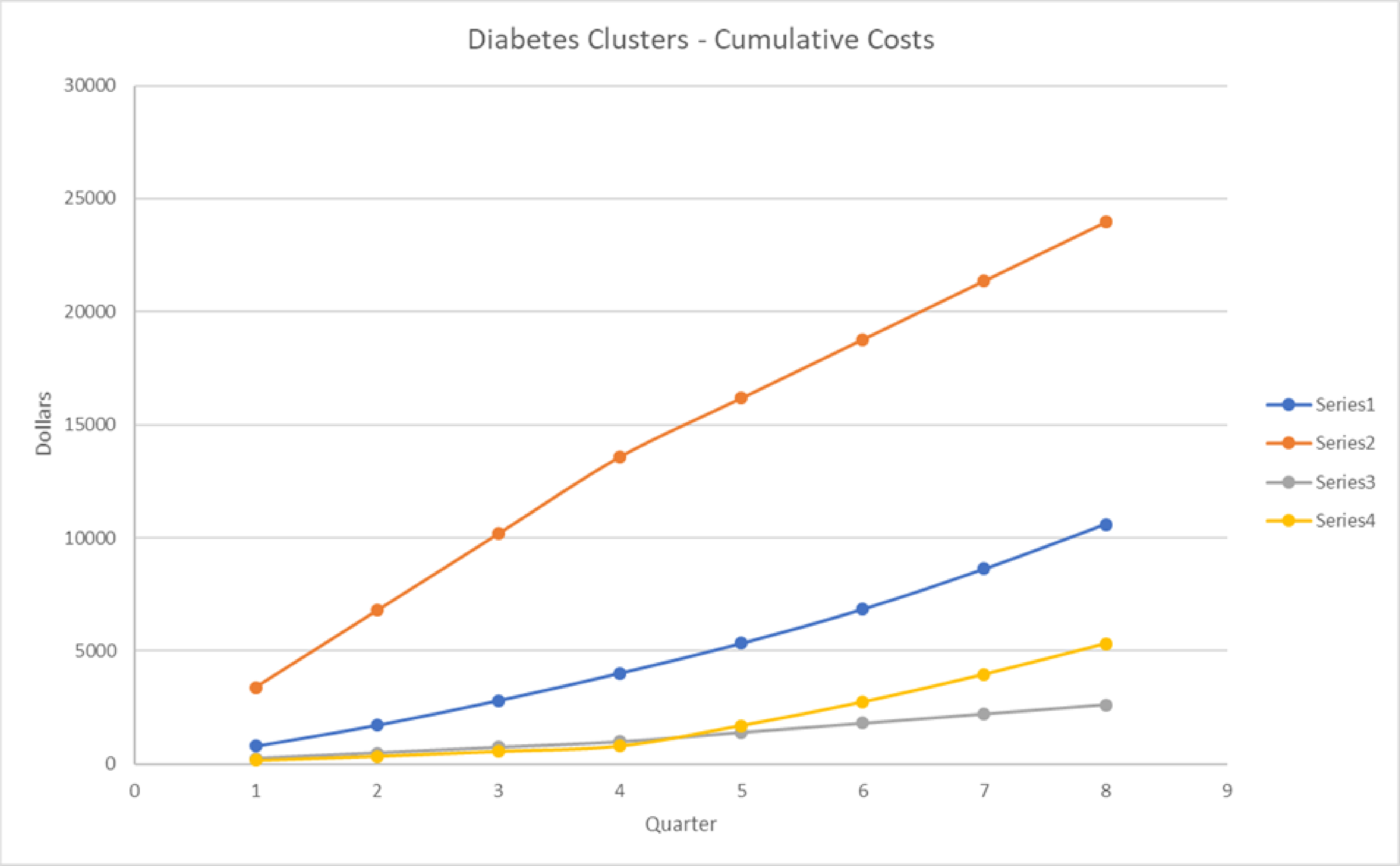
- Pay careful attention to a few parameters:
  - Number of Trees (aka estimators)
  - Feature Subset Strategy (aka max_features)
  - Minimum samples for a split to occur
  - Max depth of each tree

Real World Example

- Evaluated a large state-wide population in the U.S. of over 300,000 unique patients spanning 3 years from 2014-2016 using random forests
- Payor claims data and social determinants of health data
- Can we detect meaningful clusters of trajectories for *diabetes progression*, in order to create cost-effective screening programs

	Diabetes Progression Models		
Prediction	**Non PredPos %**	**PredPos %**	**Total Acc**
Pre-Diabetes (2014) to Full Diabetes (2015)	30.5%	72.9%	71.6%
Diabetes to Complications (2015)	19.9%	87.0%	83.5%

Real World Example



- Orange Group – High utilizers, high incidence renal complications
- Gray Group – Low Utilizers, with few complications except CV
- Blue Group – Falling in between Orange/Gray
- Yellow Group – "newer" cases with fewer complications, fewer mental health issues, earlier med stage

Orange and Blue groups were TWICE as likely to have mental health comorbidity

ML Stages

Setup Environment, Import Stuff

1) **Load Data**
   - *Read File, Parse header and row data*
2) **Preprocess**
   - *Normalize, Discretize, Impute, etc.*
3) **Feature Selection**
   - *Select subset of relevant features*
4) **Train Model**
   - *Fit some model(s) to the dataset*
5) **Evaluate Performance**
   - Did it work?

Effects of Preprocessing

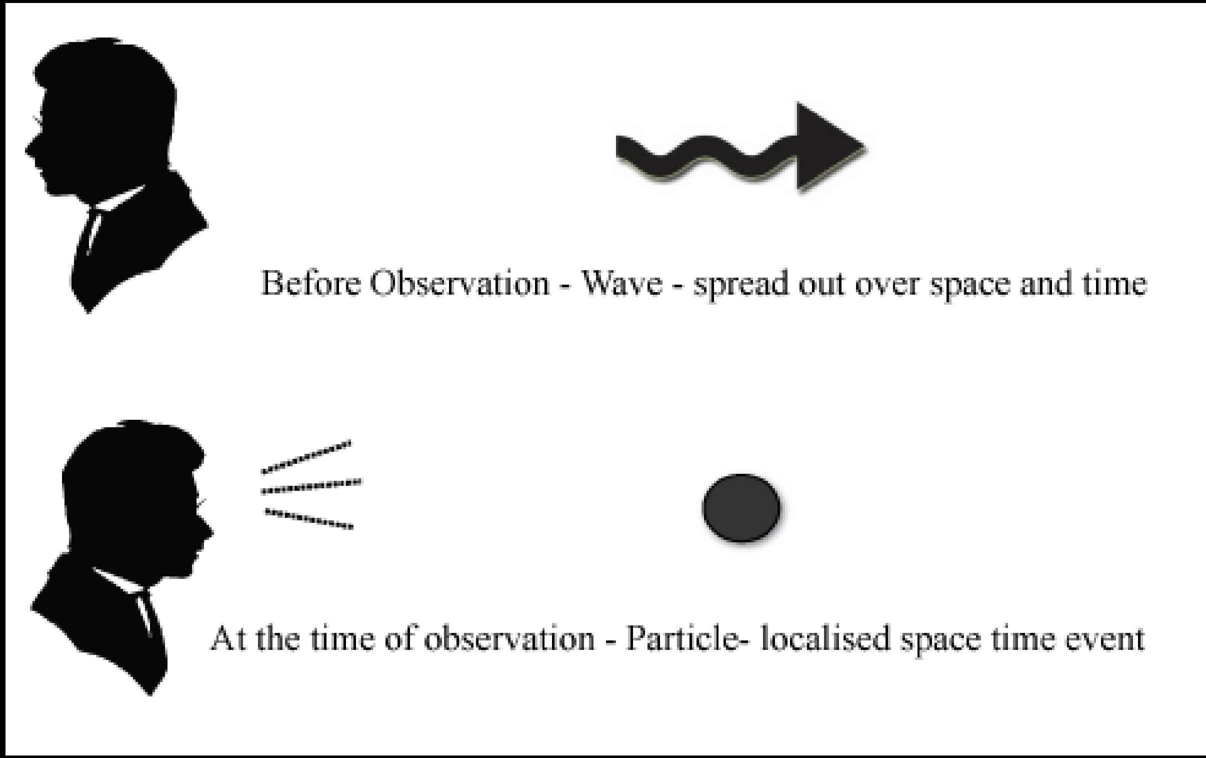
Target

Features

1) Class Rebalancing (undersampling, SMOTE, etc.)
2) Normalization
3) Discretization (i.e. binning)
4) Imputation
5) Outlier removal (i.e. winsorize)
6) etc. etc.

What is the *fundamental* problem when we do pre-processing on the data?

Observer Effect



observer effect

Does the act of observation alter or change the phenomena being observed?

refers to changes that the act of observation make on the phenomenon observed; often the result of instruments that, by necessity, alter the state of what they measure in some manner; the effect can be observed in the domain of physics

Effects of Preprocessing

Target

Features

1) Class Rebalancing (undersampling, SMOTE, etc.)
2) Normalization
3) Discretization (i.e. binning)
4) Imputation
5) Outlier removal (i.e. winsorize)
6) etc. etc.

For next week

1) Homework #2 released (all about random forests and bagging)
2) First paper review will be due next week (Oct.4)
   - 2-3 pages
   - Follow directions on submission