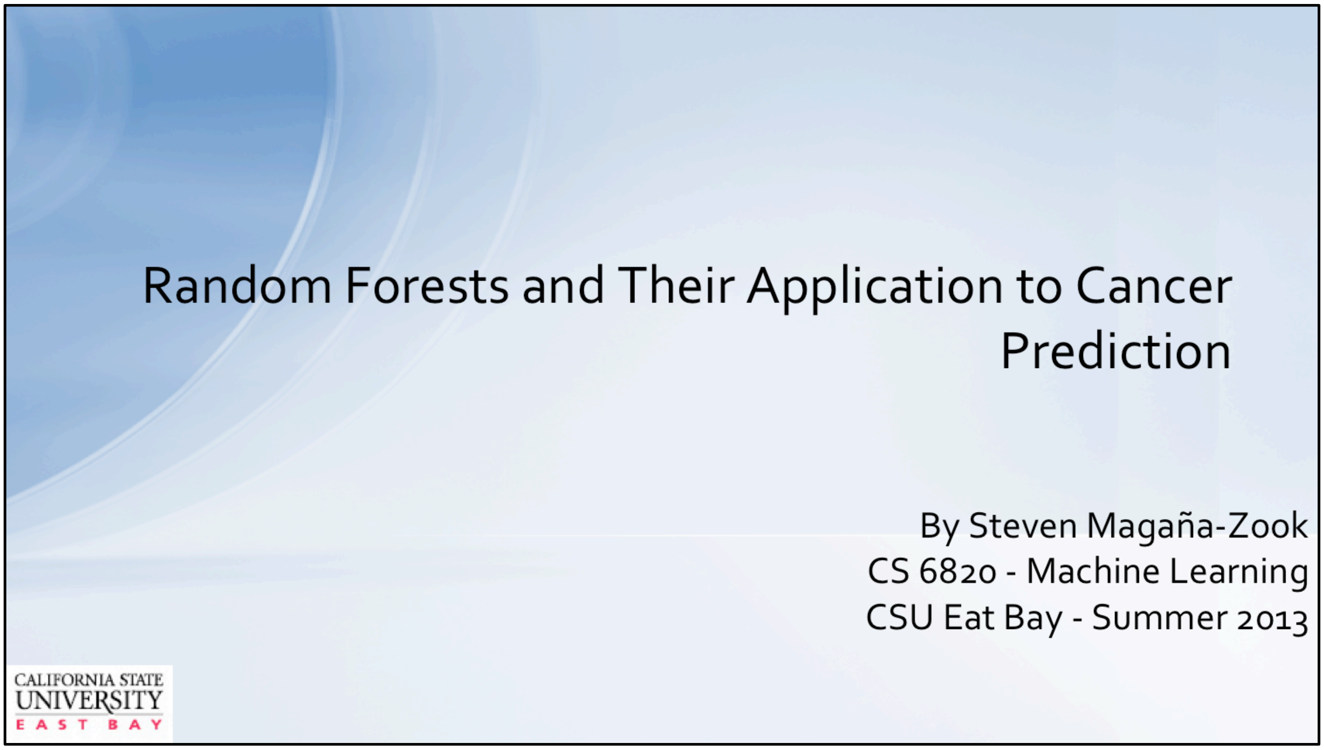
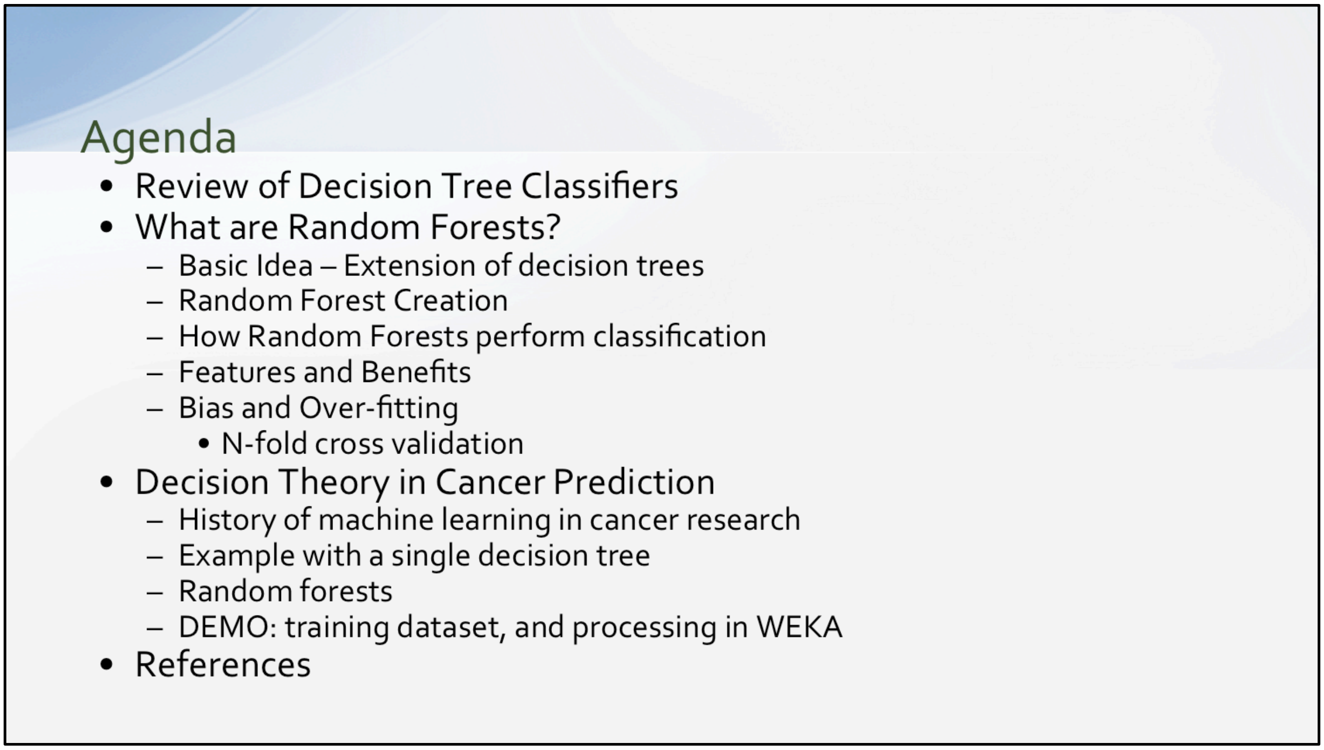


Random Forests and Their Application to Cancer Prediction

By Steven Magaña-Zook
CS 6820 - Machine Learning
CSU Eat Bay - Summer 2013

CALIFORNIA STATE
UNIVERSITY
EAST BAY

Based on previous similar work
Introduced in a paper by UC Berkeley professor Leo Breiman



Agenda

- Review of Decision Tree Classifiers
- What are Random Forests?
  - Basic Idea – Extension of decision trees
  - Random Forest Creation
  - How Random Forests perform classification
  - Features and Benefits
  - Bias and Over-fitting
    - N-fold cross validation
- Decision Theory in Cancer Prediction
  - History of machine learning in cancer research
  - Example with a single decision tree
  - Random forests
  - DEMO: training dataset, and processing in WEKA
- References

Review

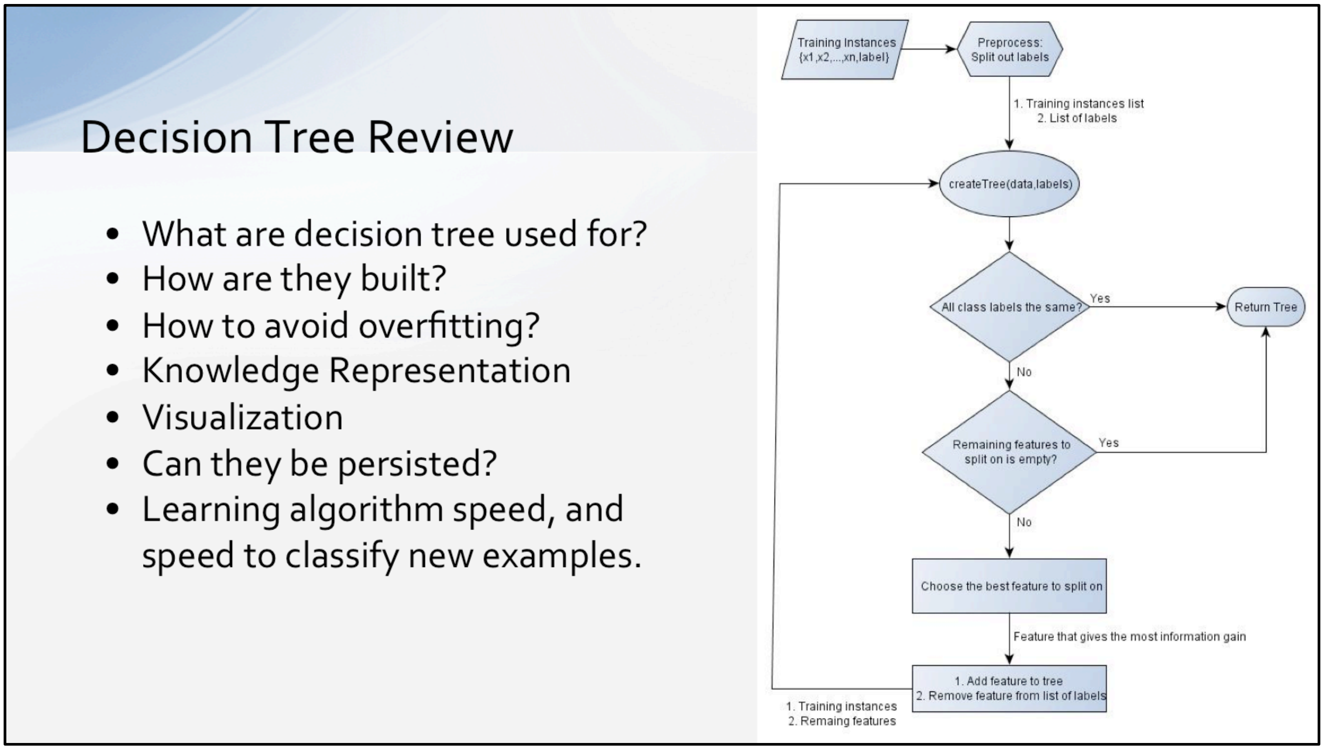
Random Forests use many decision trees, lets review what DTs are

Random Forests

Basic idea – a high level overview
Creation – how exactly you create them
classification – get answer to new example
features/benefits: Why would you want to use this ML over other algorithms
Bias/Overfitting/cross validation: How can we give the algoirthm the best chance of generalizing, and how do we prove it.
Vs others: We will look at some qualities of ML algorithms and compare RF to others. Cancer:
First we look at how a single decision tree handles cancer classification – we will se it overfits
Doctors do this by hand
We will see how random forests help classify genetic training instances with 1000s of features
Demo: I will demonstrate WEKA processing a breast cancer data set

References:

Will list some really good reading for random forest trees



Decision Tree Review

- What are decision tree used for?
- How are they built?
- How to avoid overfitting?
- Knowledge Representation
- Visualization
- Can they be persisted?
- Learning algorithm speed, and speed to classify new examples.

Training Instances {x1,x2,...,xn,label}

Preprocess: Split out labels

1. Training instances list
2. List of labels

createTree(data,labels)

All class labels the same?

Yes

Return Tree

No

Remaining features to split on is empty?

Yes

No

Choose the best feature to split on

Feature that gives the most information gain

1. Add feature to tree
2. Remove feature from list of labels

1. Training instances
2. Remaing features

Uses:

for both classification and regression problems.

Built:

Tree building algorithm recursively creates nodes by determining which feature will best classify the training instances.

Overfitting:

Generated tree may need to be “pruned” to avoid over-fitting the training data.

Removes branches/leaves that do not add much information

Knowledge Representation / visualization:

Resulting tree can be visualized for domain expert verification

Can be stored many ways (classic linked tree, in a heap, sub-dictionaries)

Persistence:

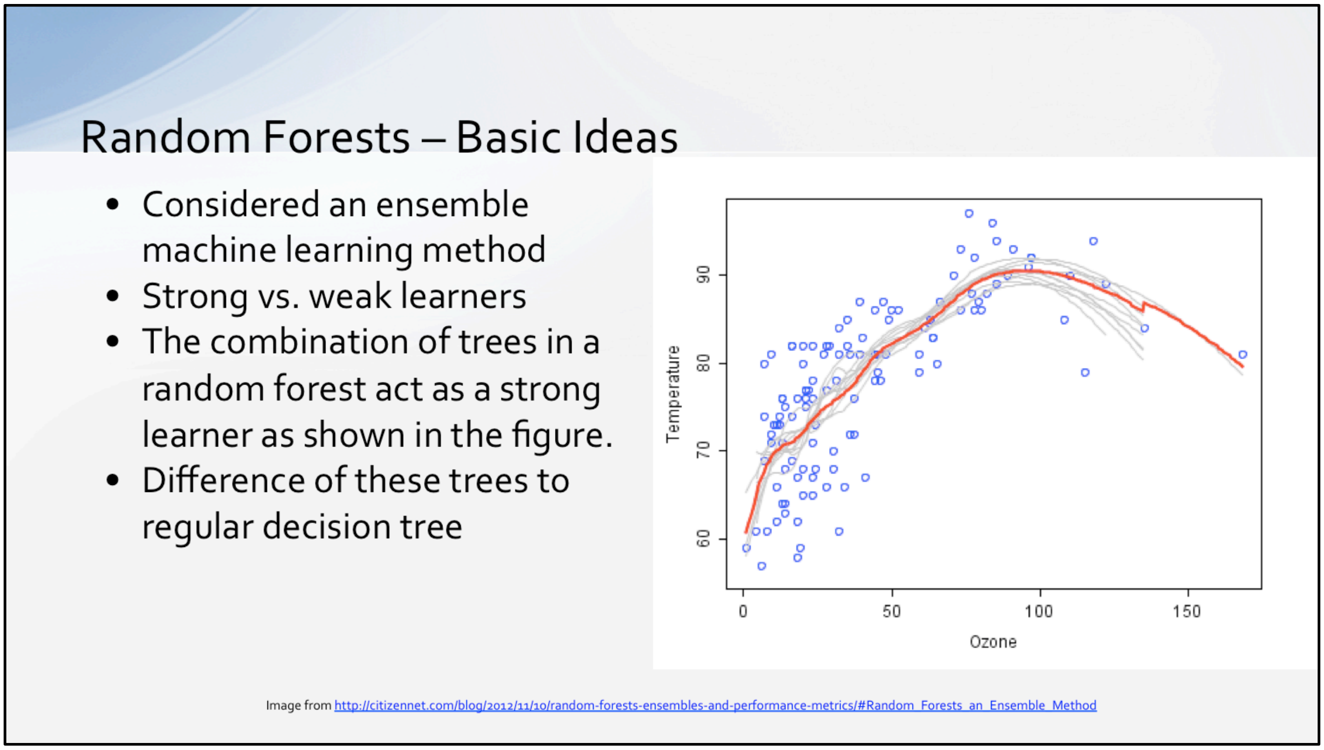
Tree can be persisted for future classification tasks without needing training data (unlike knn)

Speed:

Decision trees are faster to train than other algorithms like ANNs (show this later), are equally fast to classify new examples with.

CLICK

Discuss algorithm



Random Forests – Basic Ideas

- Considered an ensemble machine learning method
- Strong vs. weak learners
- The combination of trees in a random forest act as a strong learner as shown in the figure.
- Difference of these trees to regular decision tree

Image from http://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/#Random_Forests_an_Ensemble_Method

Considered an ensemble ML method:

Decision is made by a number of different classifiers

N decision trees are generated, each tree votes on the correct classification, class with the most votes wins.

Strong vs. weak learners:

Weak learners approximate a good classifier but are not as good as they can be.

Each tree is considered a "weak learner":

FIGURE: the gray lines are weak learners

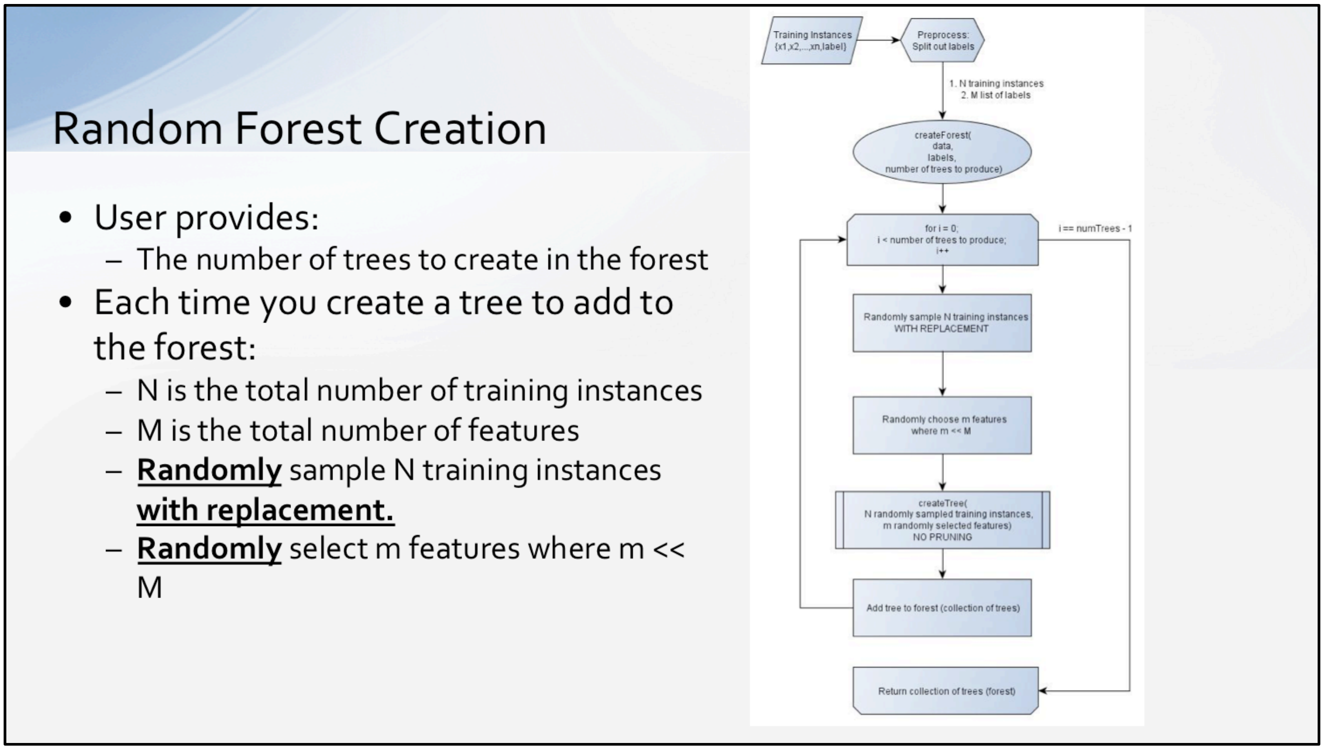
They don't match the function exactly

If they all voted we would get a better (stronger) fit to the model

The combination of trees in a random forest act as a strong learner as shown in the figure.

Difference:

No pruning of generated decision trees



Random Forest Creation

- User provides:
  - The number of trees to create in the forest
- Each time you create a tree to add to the forest:
  - N is the total number of training instances
  - M is the total number of features
  - **Randomly** sample N training instances **with replacement.**
  - **Randomly** select m features where m << M

Training Instances {x1,x2,...,xn,label}
Preprocess: Split out labels
1. N training instances
2. M list of labels
createForest(data, labels, number of trees to produce)
for i = 0; i < number of trees to produce; i++
i == numTrees - 1
Randomly sample N training instances WITH REPLACEMENT
Randomly choose m features where m << M
createTree(N randomly sampled training instances, m randomly selected features) NO PRUNING
Add tree to forest (collection of trees)
Return collection of trees (forest)

User has to determine the number of trees to create.

There is no max, we will talk about over fitting and the error rate of the forest soon

The algorithm keeps some variables:

N is the number of training instances you provide

We will always send N instances to create a tree

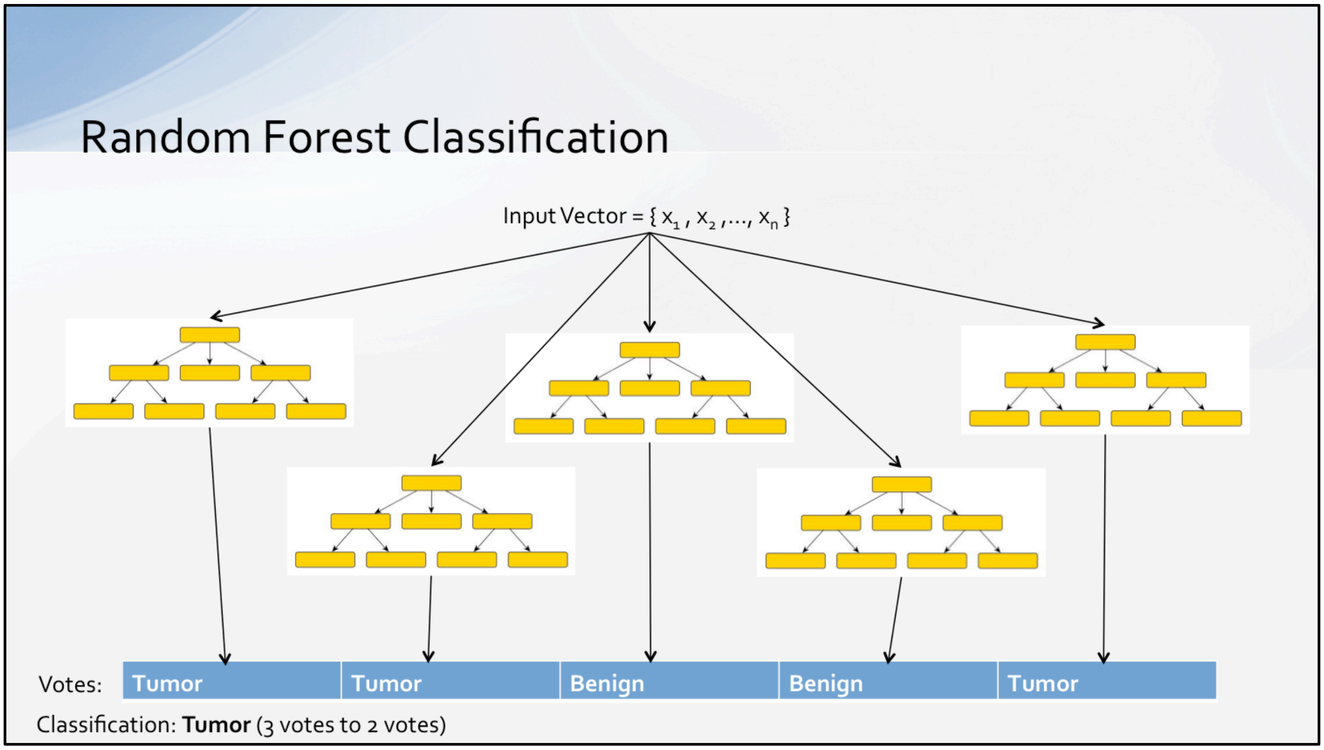
The N instances will be randomly sampled with replacement from all instances

M is the total number of features

to build a tree, we randomly select m features to use (m << M)

We then build a tree based on the N randomly sampled training instances and m randomly chosen features

See why it is a weak learner?



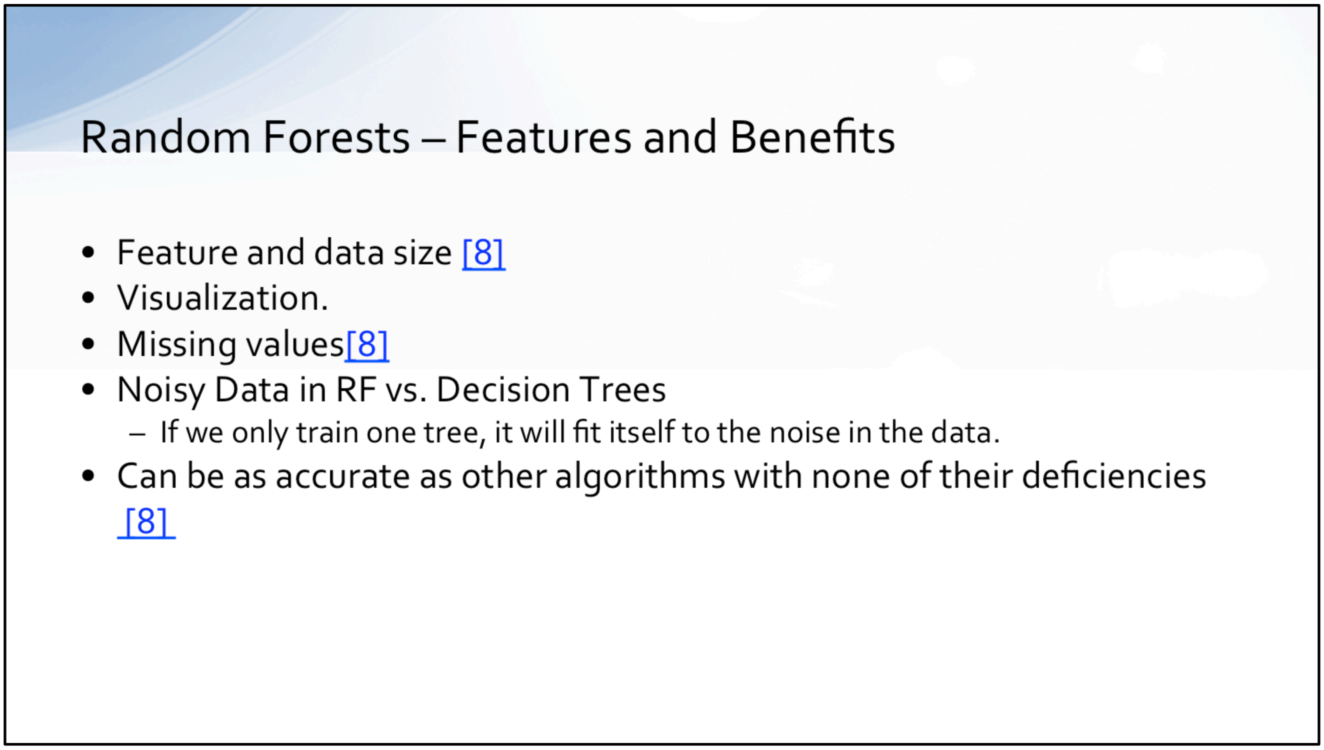
Random Forest Classification

Input Vector = { x_1 , x_2 ,…, x_n }

Votes:	**Tumor**	**Tumor**	**Benign**	**Benign**	**Tumor**

Classification: **Tumor** (3 votes to 2 votes)

Imagine this on a thousand trees



Random Forests – Features and Benefits

- Feature and data size [8]
- Visualization.
- Missing values[8]
- Noisy Data in RF vs. Decision Trees
  - If we only train one tree, it will fit itself to the noise in the data.
- Can be as accurate as other algorithms with none of their deficiencies [8]

Feature size:

Able to quickly learn a large number (1000+) of features over large datasets [8]

Visualization:

Like Decision Trees, Random Forests can be visualized to show how they classify training instances.

Missing values:

Able to handle large amount of missing values in training instances [8]

Creator experimented with even removing 60% of values and it worked

Noise: WHAT IS NOISE

Resilient to noise in training instances

If we only train one tree, it will fit itself to the noise in the data.

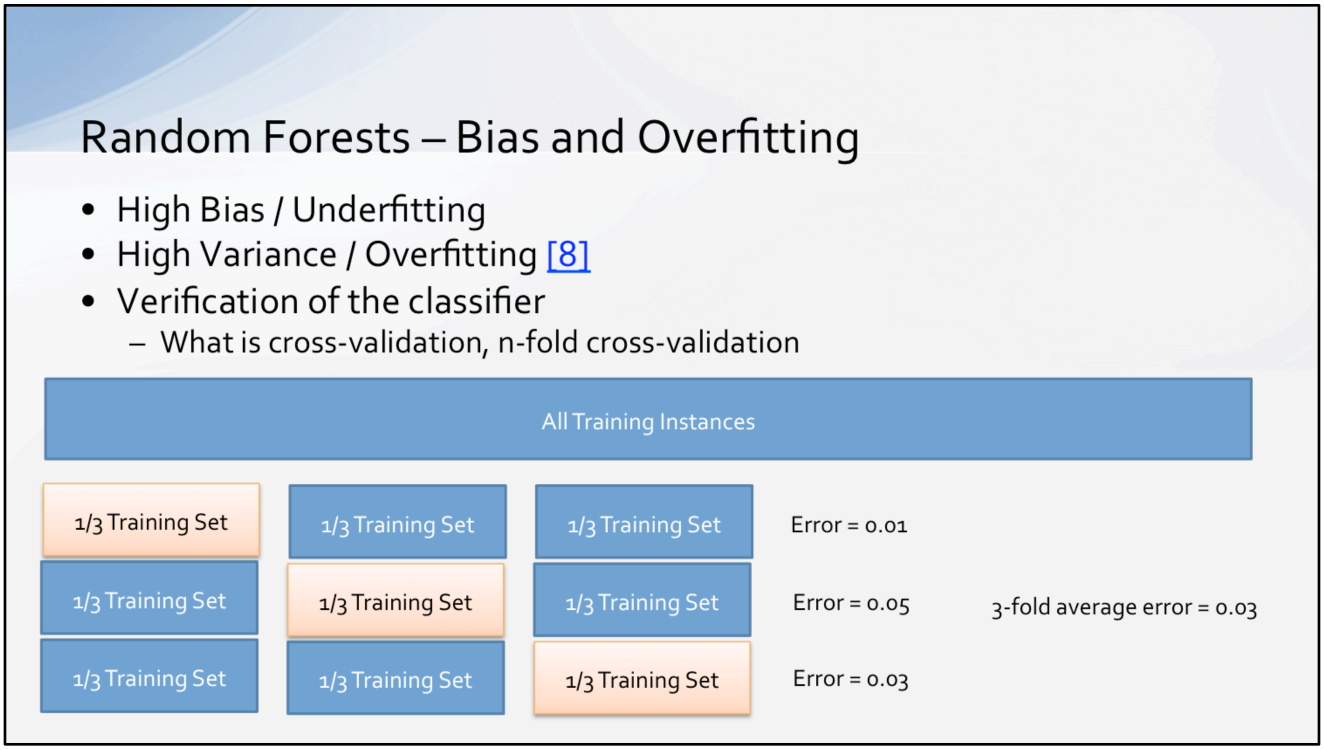
Accuracy/ Deficiencies:

As accurate as neural networks without NN's "black box" knowledge representation.

Can perform non-linear regression while also consuming nominal values.

Future classifications do not require the original dataset in memory (KNN)

Learning algorithm opens up to parallel operations (each tree can be separately trained)



Random Forests – Bias and Overfitting

- High Bias / Underfitting
- High Variance / Overfitting [8]
- Verification of the classifier
  - What is cross-validation, n-fold cross-validation

All Training Instances

1/3 Training Set	1/3 Training Set	1/3 Training Set	Error = 0.01	
1/3 Training Set	1/3 Training Set	1/3 Training Set	Error = 0.05	3-fold average error = 0.03
1/3 Training Set	1/3 Training Set	1/3 Training Set	Error = 0.03	

Bias:

The generated model does not accurately predict the training data (i.e. linear regression on quadratic training examples)

These trees are considered weak learners because they do not fit the data exactly (they show some bias).

Variance / over fitting:

The generated model (decision tree/ random forest) very accurately predicts the training data, but does not generalize well when used to classify new data.

The error rate of the forest is only dependant on how randomly the trees were created (the more random/ uncorrelated the better) and how strong the individual trees fit the data

Cross Validation (3 fold cv shown):

Purpose of cross validation: to estimate how accurately a predictive model will perform in practice / general on new examples

Gold squares used as test/validation set

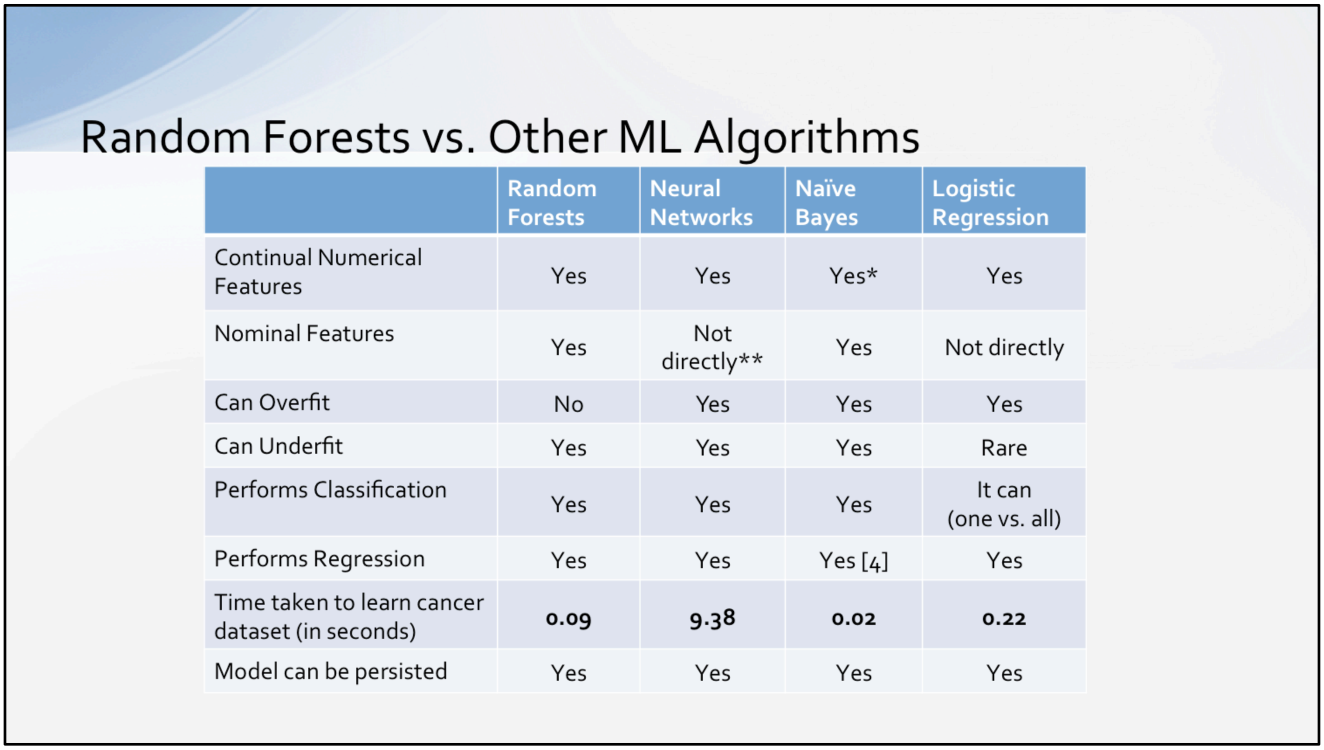
In *k*-fold cross-validation,

the original sample is **randomly** partitioned into *k* equal size subsamples.

Of the *k* subsamples, a single subsample is retained as the validation data for testing the model, and the remaining *k* − 1 subsamples are used as training data.

The cross-validation process is then repeated *k* times (the *folds*), with each of the *k* subsamples used exactly once as the validation data.

The *k* results from the folds then can be averaged (or otherwise combined) to produce a single estimation.



Random Forests vs. Other ML Algorithms

	Random Forests	Neural Networks	Naïve Bayes	Logistic Regression
Continual Numerical Features	Yes	Yes	Yes*	Yes
Nominal Features	Yes	Not directly**	Yes	Not directly
Can Overfit	No	Yes	Yes	Yes
Can Underfit	Yes	Yes	Yes	Rare
Performs Classification	Yes	Yes	Yes	It can (one vs. all)
Performs Regression	Yes	Yes	Yes [4]	Yes
Time taken to learn cancer dataset (in seconds)	**0.09**	**9.38**	**0.02**	**0.22**
Model can be persisted	Yes	Yes	Yes	Yes

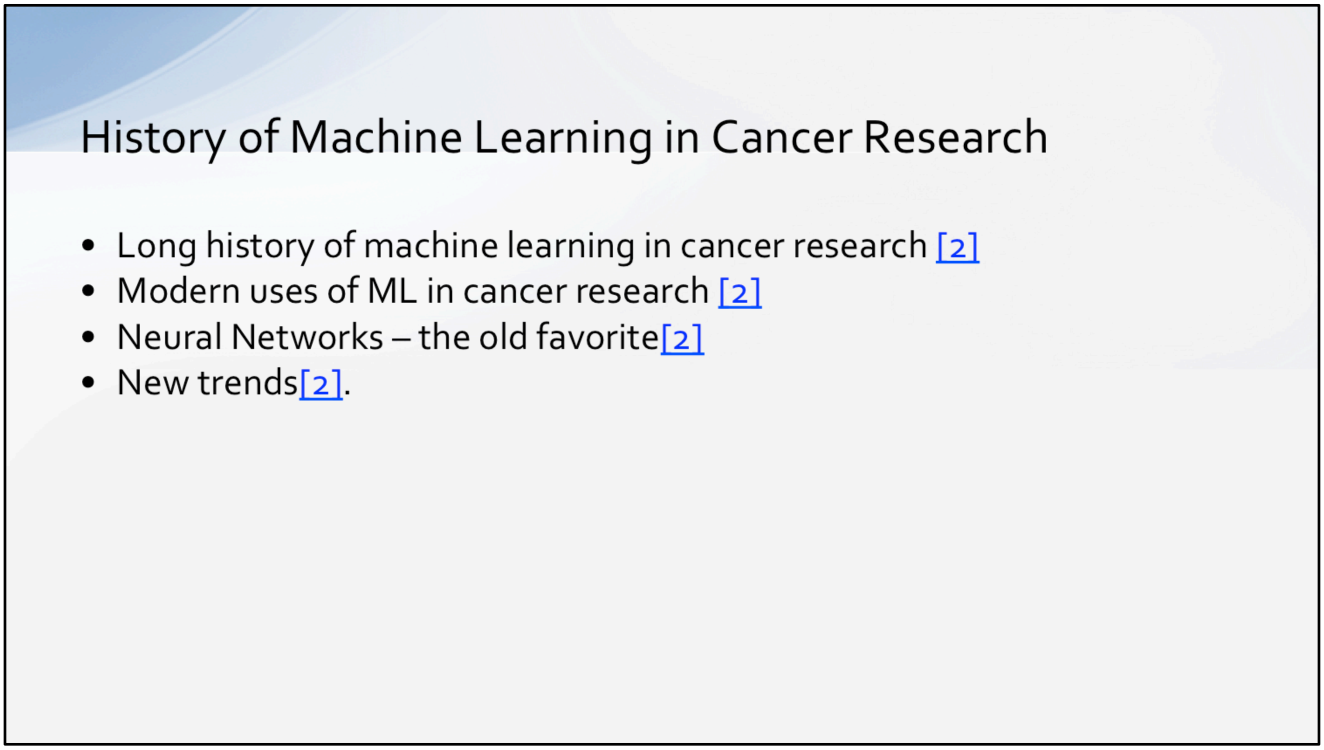
Cancer dataset is 568 instances with 32 features

* Continuous variables are handled in naïve Bayes by computing the mean and variance and plugging them into the normal distribution or "binning" them.
** Nominal values are coded (binary vector)

Neural networks show bias when there are not enough "hidden units to represent the required mappings" Page 11 out of 12 here: http://www.cs.bham.ac.uk/~jxb/NN/l9.pdf

Over fit and bias on algorithms can be due to poor training sets

Logistic regression can have bias if implemented poorly without the bias weight/ parameter



History of Machine Learning in Cancer Research

- Long history of machine learning in cancer research [2]
- Modern uses of ML in cancer research [2]
- Neural Networks – the old favorite[2]
- New trends[2].

History

20+ years 1985 paper by Simes RJ used statistical decision theory to predict the best treatment plan for ovarian cancer patients based on quality-of-life choices [1]

1500+ papers on machine learning and cancer in PubMed database [2]

Modern uses:

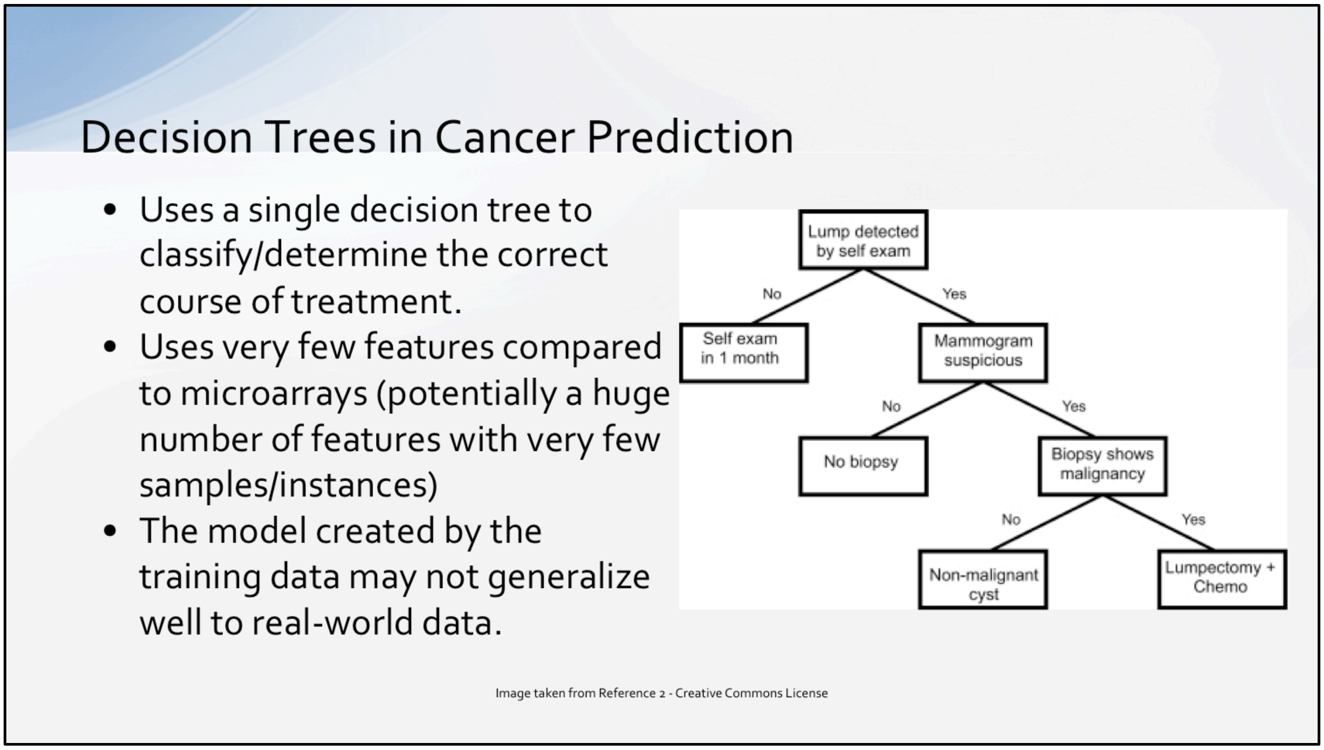
Identification and classification of tumors
DNA analysis (microarrays)
A patient's likely response to treatments

Neural Nets

70% of papers favor Neural Networks as the primary ML algorithm

New Trends

away from using NNs and also towards cancer prediction instead of ex post facto diagnosis

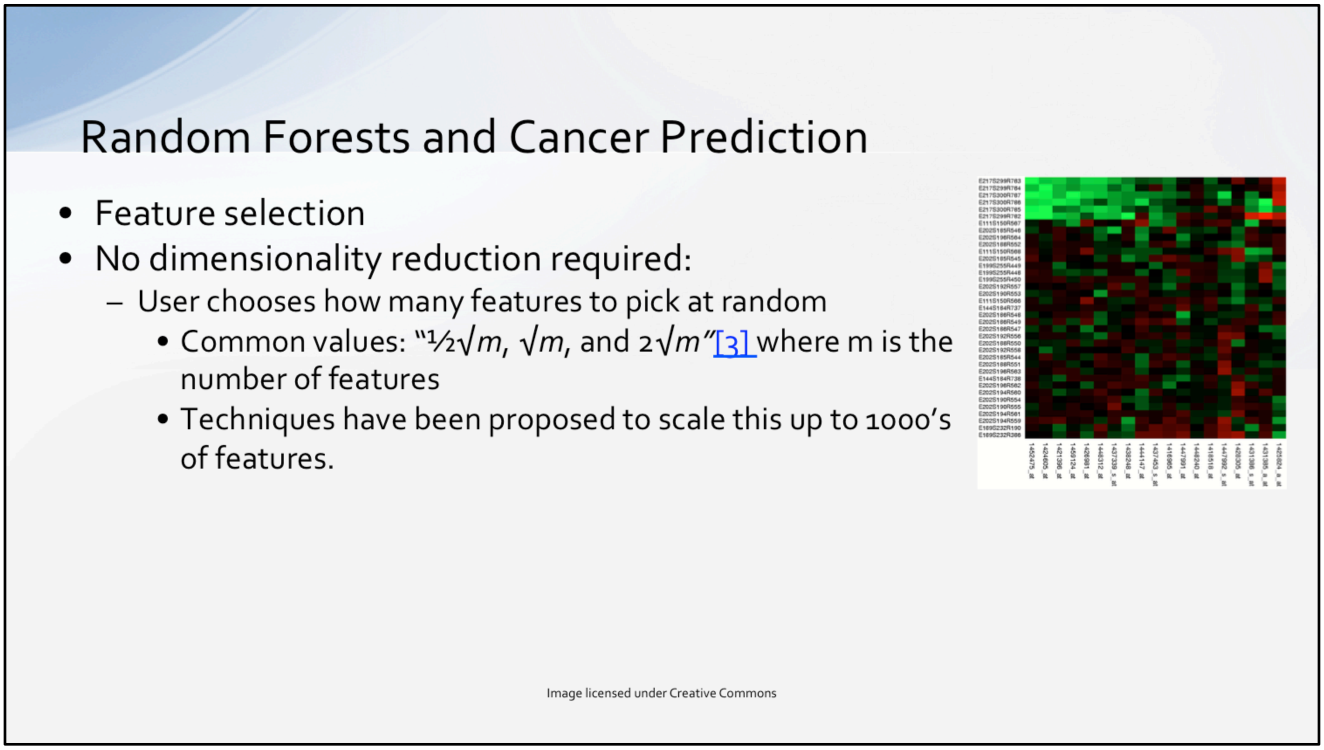


Decision Trees in Cancer Prediction

- Uses a single decision tree to classify/determine the correct course of treatment.
- Uses very few features compared to microarrays (potentially a huge number of features with very few samples/instances)
- The model created by the training data may not generalize well to real-world data.

Image taken from Reference 2 - Creative Commons License

Talk about dimensionality problem #Features >> #instances
Talk about how microarrays provide numeric representations about the expression of genes from a sample (tumor / normal cell)



Random Forests and Cancer Prediction

- Feature selection
- No dimensionality reduction required:
  - User chooses how many features to pick at random
    - Common values: "½√*m*, √*m*, and 2√*m*"[3] where m is the number of features
    - Techniques have been proposed to scale this up to 1000's of features.

Image licensed under Creative Commons

Feature Selection:

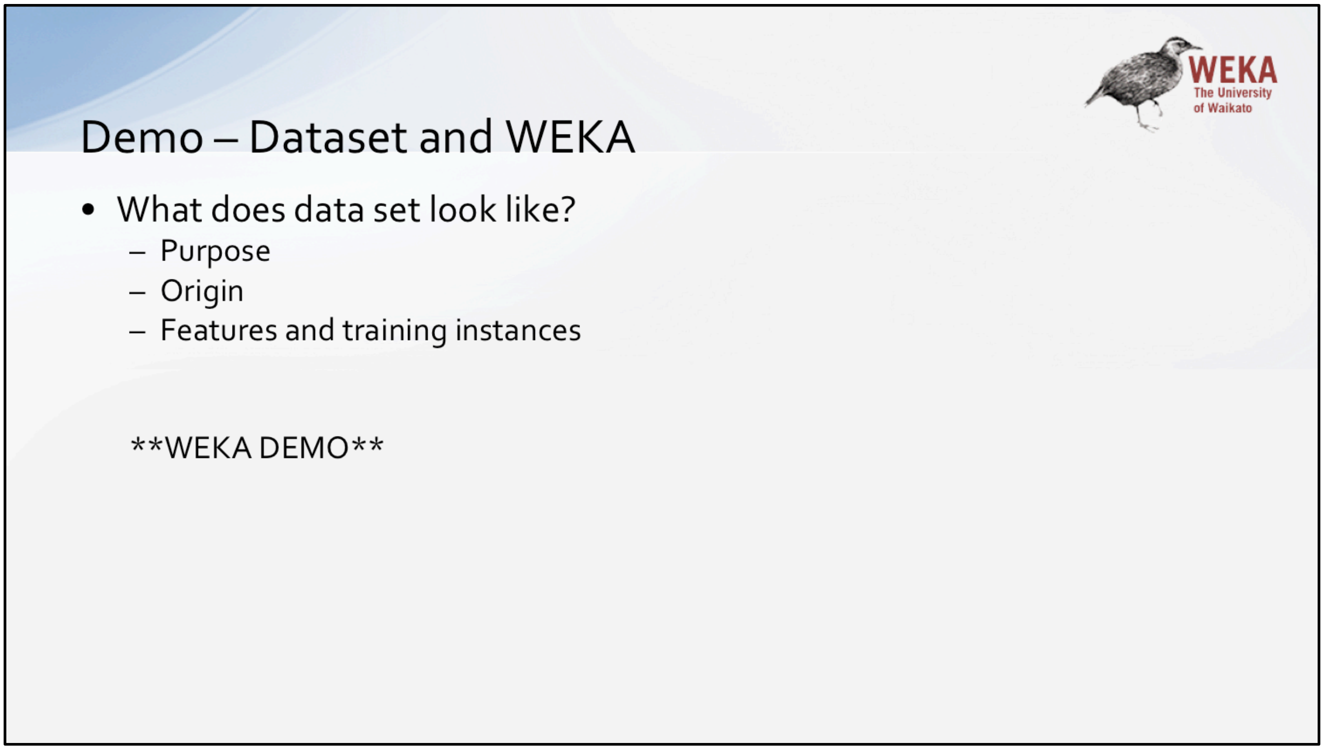
When cancer researchers have so many genes to look at, random forests provide them a way of finding the relevant ones.

They can then throw away the ones with no predictive power.

What if there is still a ton of features?

1000s of features:

Paper: Technique involves assigning weights to features to prevent the effects of correlated features [5]



Demo – Dataset and WEKA

- What does data set look like?
  - Purpose
  - Origin
  - Features and training instances

WEKA DEMO

WEKA
The University
of Waikato

Purpose:

The purpose of the dataset is to try to figure out if a tumor is malignant or benign.

Origin:

The training data comes from UC Irvine's Machine Learning Repository (free datasets)
http://archive.ics.uci.edu/ml/datasets.html

Features:

Dataset contains 32 attributes and 569 training instances
Attributes include: patient id, 30 continuous features, and a class label

What is WEKA?

Features

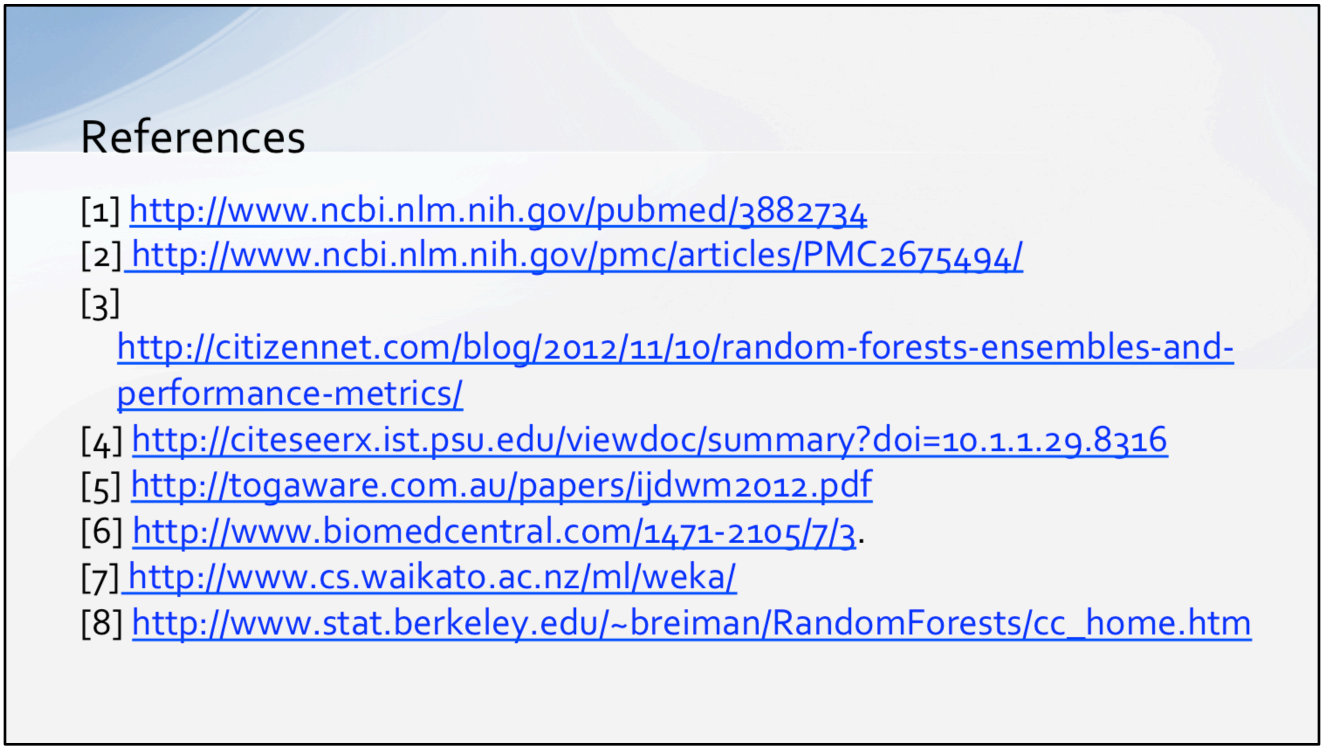
"contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization"[7]

Cost/ license

free, open source

Why use it

Very useful tool for exploring how different machine learning algorithms perform on a dataset!



References

[1] http://www.ncbi.nlm.nih.gov/pubmed/3882734

[2] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2675494/

[3] http://citizennet.com/blog/2012/11/10/random-forests-ensembles-and-performance-metrics/

[4] http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.8316

[5] http://togaware.com.au/papers/ijdwm2012.pdf

[6] http://www.biomedcentral.com/1471-2105/7/3.

[7] http://www.cs.waikato.ac.nz/ml/weka/

[8] http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm