Bagging / Random Forest

An Introduction to ensemble models

Danny Lumian | Michael Dyer | Elliot Cohen | Ivan Corneillet | Taryn Heilman | Erich Wellinger

Objectives

galvanize

- Review decision trees
- What is an ensemble model?
  - Why is it useful?
  - How does it deal with the bias-variance trade-off?
- What is bagging?
  - What are pros and cons of bagging?
- What are random forests?
  - How are they different from bagging?
  - What are the tunable hyperparameters?
- Individual Assignment
  - Implement bagging / random forests
- Pair assignment
  - Predict churn with random forests

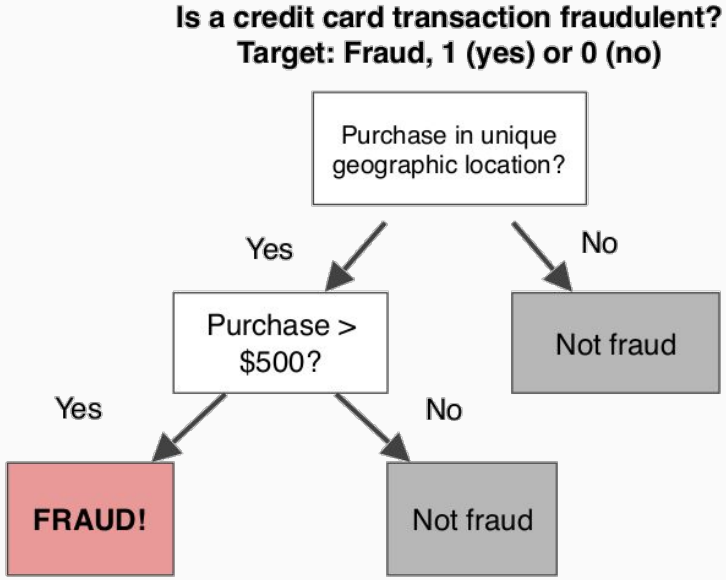
Decision Trees

galvanize

- Describe metrics used to describe the impurity in a node of categorical variables (there should be 2)
  - Shannon Entropy & Gini Index
- What about for a continuous variable?
  - RSS
- Describe how a decision tree "decides" where to make its next split.
- Are decision trees deterministic?
  - Yes
- Decision trees are non-parametric, but they have some hyper parameters that we can tune - name and describe them!
  - Max Depth, Min Samples, Max Leaf Nodes

Decision Trees Review

- Decision Trees are deterministic:
  - Once they are trained, they will arrive at the same conclusion every time given the same X values.
- Decision Trees are "nonparametric":
  - We don't make any assumptions (mean, variance, etc.) about our data when we create a tree.



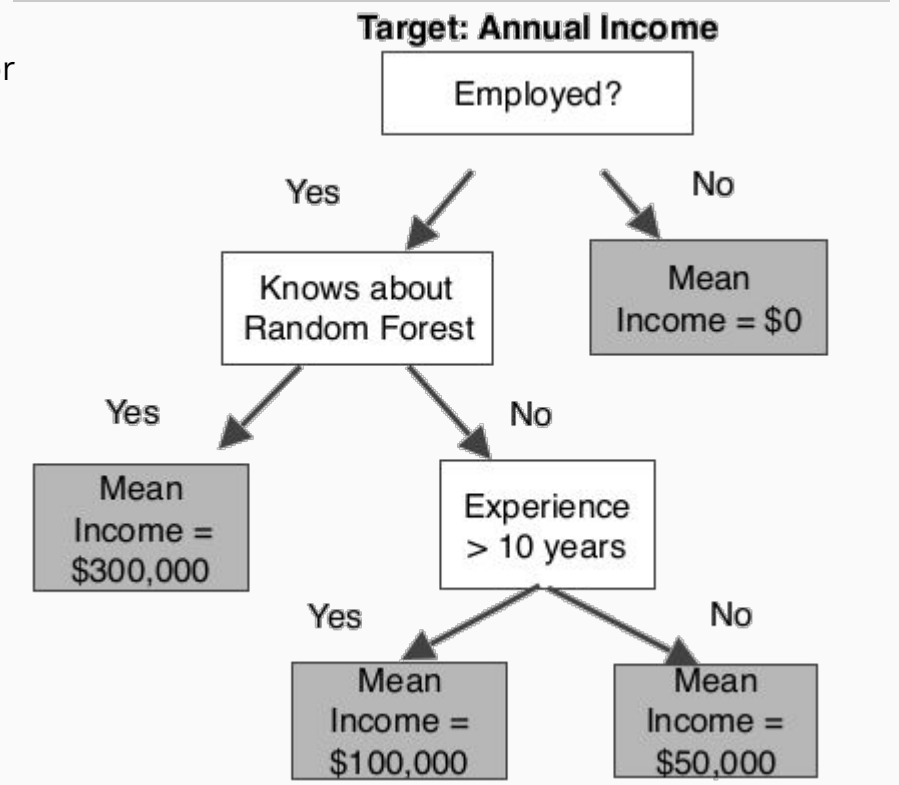
Decision Trees: Regression

Each split tries to minimize the Total Squared Error

$$\sum_{j=1}^{J} \sum_{i \in R_j} (y_i - \hat{y}_{R_J})^2$$

Where J is the number of child nodes and R is each child node

The model output is typically the mean within each leaf node



Pros and Cons of Decision Trees

galvanize

Pros

- Handle non-linear relationships well
- Highly interpretable (unless we have a ton of features)
- Useful for classification & regression
- No feature scaling required

Cons

- Expensive to train
- **Often poor predictors (high variance)**

Ensemble Models: Wisdom of the Crowd

galvanize

Suppose you are trying to guess how many Skittles are in a jar.

- Looks like I have a water bottle here!
  - Guess how many are in the jar and submit your answer:
  - https://goo.gl/forms/7Q1fw2tPB4ZYRkWF3
- Do you think your guess was better than average?
- Let's see: Results

Ensemble Model

galvanize

Combines multiple weak learners to form a strong learner

- Basic Idea: Wisdom of the Crowd
  - Run multiple models on the data and aggregate the predictive results to produce an overall prediction that is better than any of the individual models could do on their own
  - The overall prediction is the **average prediction** for a regressor or
  - **Plurality choice** for a classifier (or the average of the percentages of each class)

Ensemble for decision trees?

galvanize

- What is a common pitfall of decision trees?
  - Overfitting to the data
  - Let's try using a large number of trees to reduce variance
- What if we just build a bunch of decision trees and average the results?
  - Do you foresee any issues?

Ensemble for decision trees?

galvanize

- Decision trees are deterministic (given the same data, they will return the same splits)
  - An ensemble model won't work here!
- How can we solve this?
  - Train each learner on a different subset of the data
  - But how do we do this when we only have one set of data to work with?
    - Bootstrapping!

Bootstrapping Review

galvanize

- What is a bootstrap sample?
  - Given n data points we select a sample of n points with replacement
- What have we used it for?
  - Return better estimates of sample statistics (95% confidence interval for example)

How to bootstrap

galvanize

- Method
  - Start with your dataset of size n
  - Sample from your dataset with replacement to create one bootstrap sample of size n
  - Repeat B times
  - Each bootstrap sample can then be used as a separate dataset for estimation or model fitting
- Advantages
  - Requires no theoretical calculations
  - Available regardless of how complicated the estimator might be

Bias and Variance Review

galvanize

- Bias:
  - Error from failure to match the training set
  - Caused by choosing a model that is too simple

- Variance:
  - Error due to random noise specific to our training set
  - Decreases as we get more data

Bias Variance Trade-off for Bushy Trees

galvanize

- What's great about bushy decision trees?
  - Low bias
- What's not so great about them?
  - High variance
- Bias
  - Trees can be low bias by being highly complex
- Variance
  - "Given a set of n independent observations Z_1 ,...,Z_n , each with variance σ^2 , the variance of the mean Z-bar of the observations is given by σ^2 / n. In other words, averaging a set of observations reduces variance."
    -- Introduction to Statistical Learning

$$\text{Variance} = \frac{\sigma^2}{B}$$

Where B is the number of independent estimators

Bagging

galvanize

Bagging* , or **B**ootstrap **ag**gregation, is a general-purpose procedure **for reducing the variance** of a statistical learning method

- Bootstrapped trees provide low-bias, high variance predictors
- Trees are generally grown deep and are not pruned, hence each individual tree has high variance, but low bias
- Averaged predictors are still low-bias
- Averaged estimators are lower variance than single predictors
- The number of trees B is not a critical parameter with bagging; using a very large B won't lead to overfitting
- In practice we want to use a large enough B such that our test error has settled down

*Bagging is a general ensemble procedure often used with trees

Bagging

galvanize

- This seems a little like cheating right?
- What did we assume about our estimators?
  - Independence
  - Bootstrapped samples are not independent

$$Var\left(\hat{f}_{bag}(x_0)\right) = \rho\sigma^2 + \frac{(1-\rho)\sigma^2}{B}$$

- ρ = rho, the correlation coefficient for the different samples
- As rho goes towards one, the effect of multiple estimators on decreasing variance weakens
- Only the uncorrelated parts of the sample decrease variance

Bagging Review

galvanize

- What is an ensemble method?
- Why do we need to bootstrap?
- The general idea of bagging is to start with ___ bias, ___ variance trees and aggregate across multiple models to decrease _____
- Why are bootstrapped samples correlated?

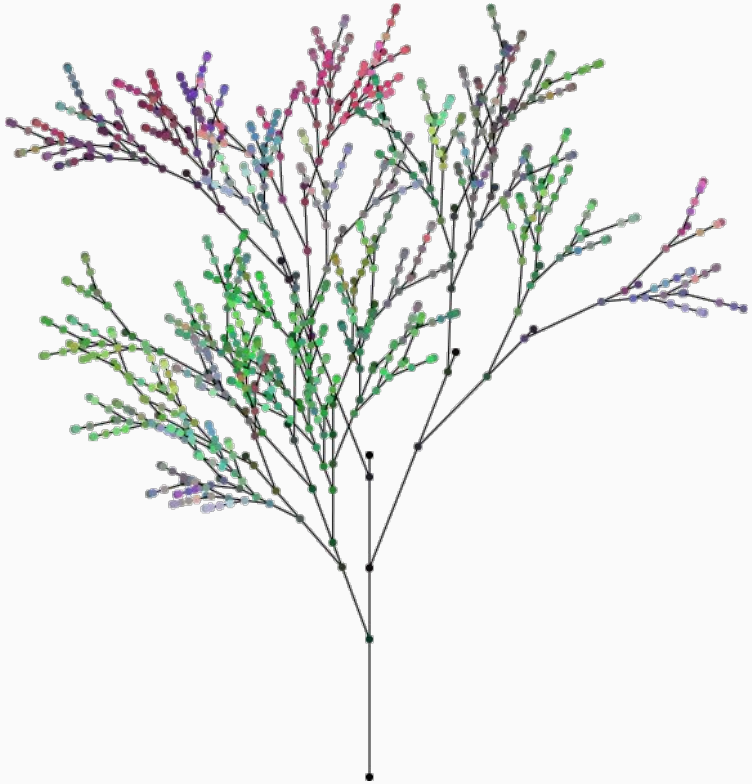
Decorrelating Trees

galvanize

- Bagged trees are correlated because:
  - Bootstrap samples are about the same (approximately ⅔)
    - Can't do much about this
  - Influential features tend to be the same
    - We **can** fix this

Random Forests

Everyone's Favorite ML algorithm



galvanize

http://luc.devroye.org/lucsforest2008/

Random Forests

galvanize

- Ensemble method similar to bagging
- Trees are trained on bootstrapped samples
- But, at each split we only use a random subset of features (subset sampling)
  - Only use some features to make splits
  - This prevents trees from always making the same splits
  - Subset of features is resampled at every split
  - Smaller subset of features = less correlation between trees

Tuning Random Forests

galvanize

- Tuning
  - m = number of features included at each split
  - p = total number of features
  - m = $\sqrt{p}$ (sklearn default) or m = p/3 features
- Performance
  - Random forests are robust to overfitting and other issues associated with decision trees
  - Generally use a large number of bushy trees
  - Often has state-of-the-art performance without much tuning

Random Forest Parameters

galvanize

- Total number of trees (n_estimators)
- Number of features to consider at each split (max_features)
- Individual decision tree parameters
  - E.g., tree depth (max_depth), pruning (min_samples_split, min_samples_leaf)

How many trees should we use?

- Variance decreases with more trees
  - But with diminishing returns
- Runtime scales linearly with the number of trees
- More is still better, but start small first to reduce computation time

Random Forest Pros and Cons

galvanize

- Pros
  - Often give near state-of-the-art performance
  - Good out-of-the-box performance
  - No feature scaling needed
  - Models nonlinear relationships
- Cons
  - Can be expensive to train (though can be done in parallel)
  - Models can be quite large (the pickled version of a several hundred tree model can easily be several GBs)
  - Not interpretable (although techniques such as predicted value plots can help)

Random Forests Review

galvanize

- How is random forest an improvement over bagging?
- As the subset of features (m) available at each split increases, tree correlation _____
- What are two pros of Random Forests?
- Two cons?

Afternoon

Random Forest in Sklearn

galvanize

class sklearn.ensemble.**RandomForestClassifier**(*n_estimators='warn'*, *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*)[source]

```
from sklearn.ensemble import RandomForestClassifier # or RandomForestRegressor

rfc = RandomForestClassifier(n_estimators = 100, n_jobs = -1)
rfc.fit(X_train)

y_pred = rfc.predict(X_test)
```

Out of Bag Score (OOB Score)

galvanize

OOB Score is a quick and dirty "replacement" for cross validation

- We already have data that each tree has not seen yet -- each bootstrapped sample only includes about ⅔ of the data.
- We can feed the data that wasn't used in a tree as a **test set** for that tree.
- We can then aggregate the *accuracy score** for each of our points (each test data point tested on ~⅓ of our trees)

*The downside is that oob_score in sklearn only computes accuracy or R^2, so if we want precision, recall, or other metrics we will still need to cross validate

```
rf = RandomForestClassifier(n_estimators=100, oob_score=True)
rf.fit(X_train, y_train)
print(rf.oob_score_)
```

Interpreting Tree Models

galvanize

Recall, one of the main strengths of Decision Trees is interpretability.

However, when we aggregate our trees with simple Bagging or Random Forests, it's not so easy…

- We can no longer simply rank our features in the order in which they were split on
- We can look at ***Feature Importances*** *(Note: not as interpretable as coefficients for a linear regression)*

Feature Importances

galvanize

- Measure the total amount the information gain increases due to splits over a given feature

  rf.feature_importances_ (where rf is your fit RandomForestClassifier / RandomForestRegressor) does this

Measuring Feature importance

galvanize

Let's get some intuition for how we calculate feature importances…

1. For each feature m_j, we calculate the decrease in our impurity criterion (MSE, Gini, etc.) for the node(s) that split on m_j
2. We then weight it by how many points passed through the nodes that split on m_j
3. And finally, we average the calculations for steps 1 and 2 across our entire forest

Leave One Out Feature Importance

galvanize

Two more methods...

- When tree B_i is grown, score it with OOB, then remove that feature and score it again to measure the change in your validation metric(s)
  - This is called **Leave One Out Feature Importances**

- Iterate through features dropping m_j out and plotting feature importances -- help with "multicollinearity"

What does feature importance not tell us?

- It is difficult to learn the effect size and direction of a feature
- For most real world problems, features don't have a constant effect size across all X-values
    - Effect direction can even reverse at different levels of X.

Partial Dependence Plots

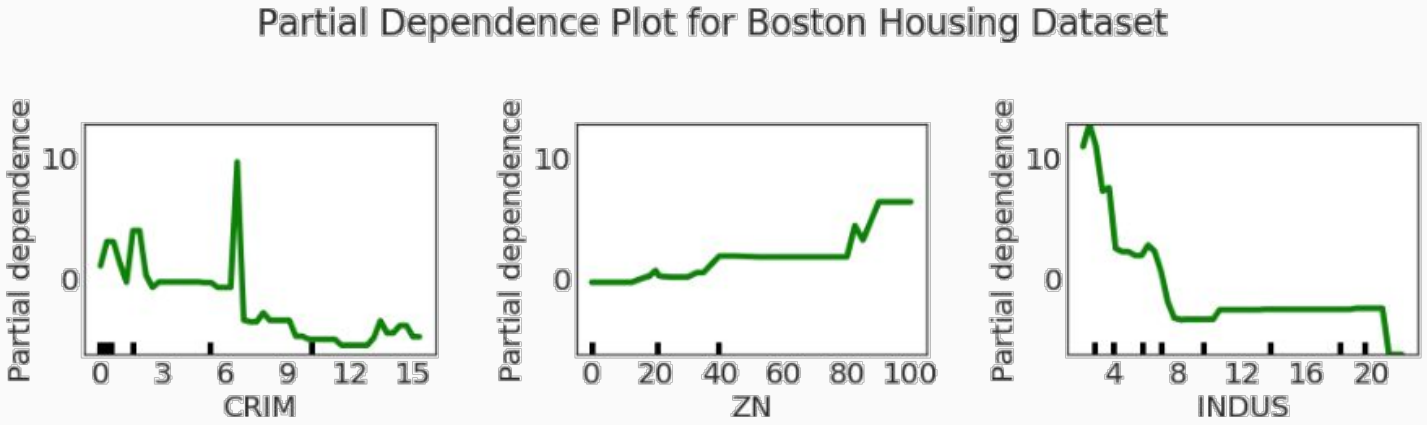
galvanize

How can we view how a single variable affects the prediction?

- Permute values for each feature column and compare predictive "success" of the feature at each value
- Compare one (or two, if you use a 3D plot) features and how predictive they are at different values
- How to interpret y - axis:
  - "The partial dependence curve at a certain feature value represents the average prediction when we force all data points to take on that feature value."[1]

1. https://christophm.github.io/interpretable-ml-book/pdp.html

Partial Dependence Plots



Partial Dependence Plots

galvanize

They can be used for all kinds of models, not just trees!

To the notebook!

galvanize

- Jupyter-notebook comparing decision trees with random forests
- Examples of feature importances and partial dependence plots
- Example usage of sklearn implementations

Predicting churn with sklearn

Random Forests

galvanize