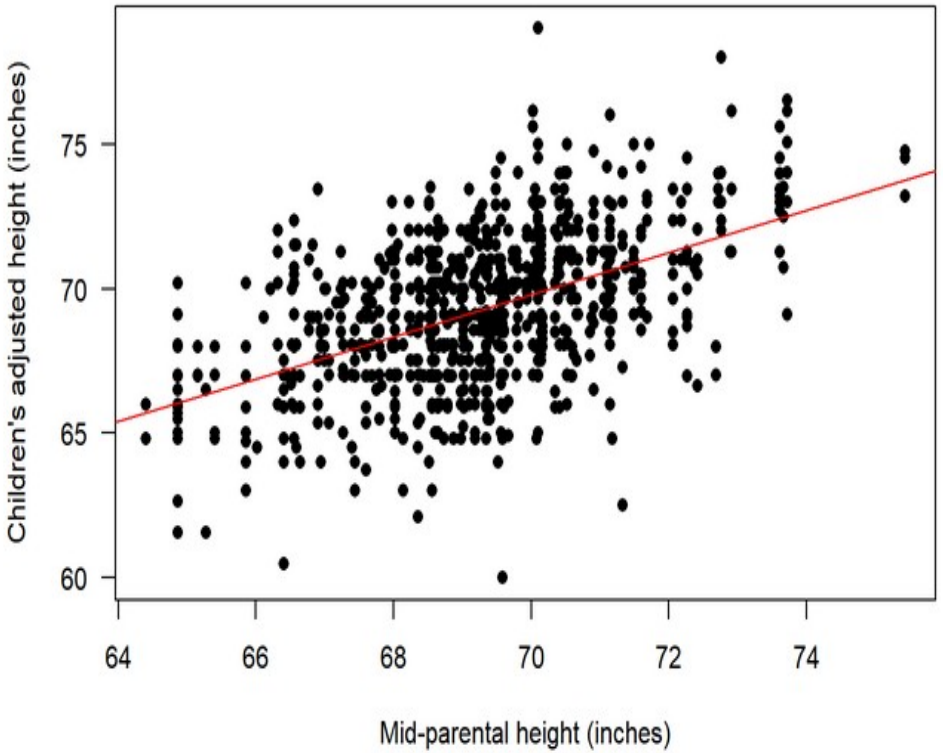
Random Forest and Gradient Boosting

4. Mai Python for Data Analytics @ Redi

What to learn today

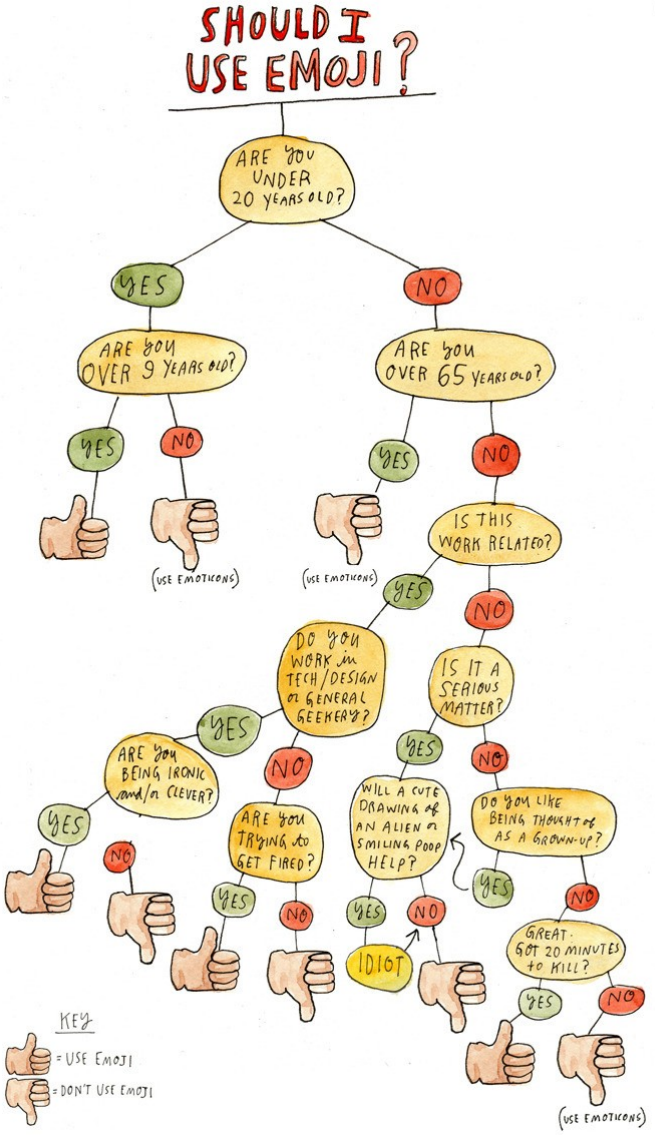
- Classification and Regression
- Decision Tree
- Decision Criteria
- Ensemble Algorithm
  - Bagging and Boosting

Review:
Regression and Classification



Picture from Thinkstock; Jay Wennington/ Unsplash

Decision Tree



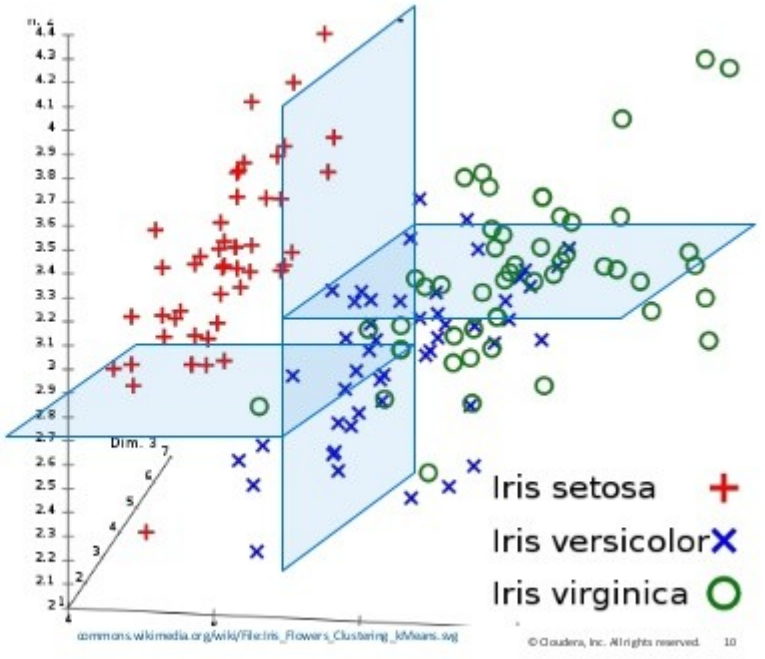
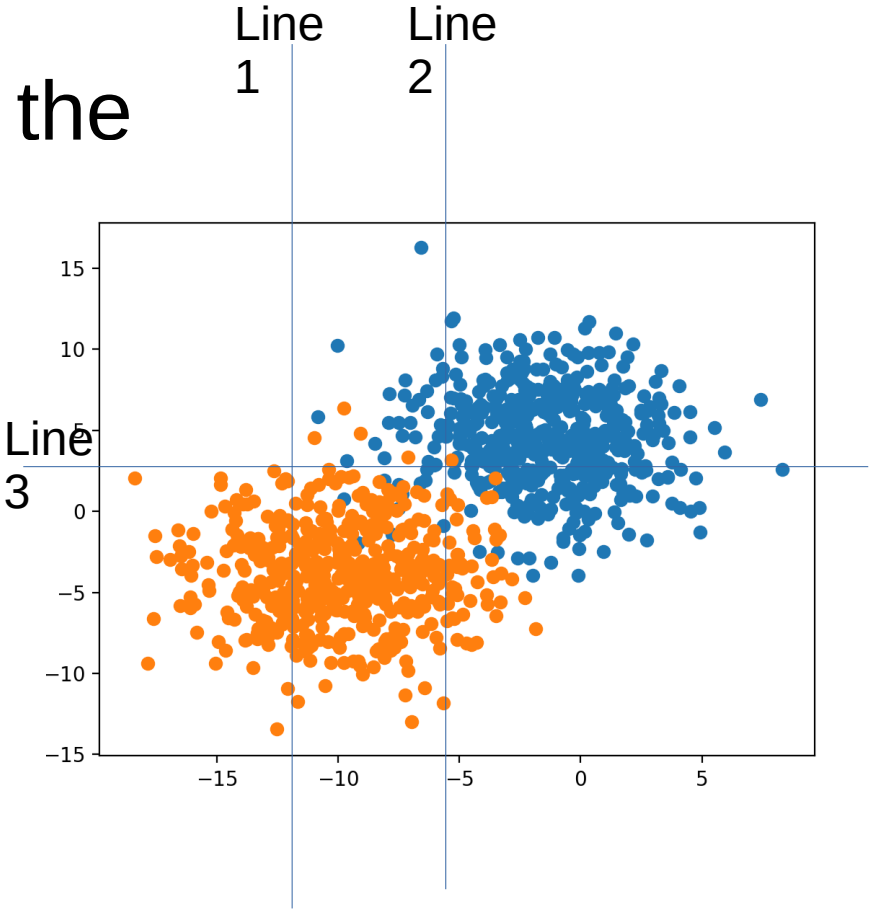


Illustration by Wendy Macnaughton

Decision Criteria

- how can we automatically find the decision step?
  - Max Information Gain
    - Min Entropy after split
    - = least randomness after the split
  - Min Gini Index
    - = most uneven after the split



Ensemble: wisdom of crowd

- Can we build a better model by combining the models?
- Bagging and Random Forest
- Boosting and Gradient Boosting

Wisdom of Crowd

- 5 models and each model has 60% accuracy
- The accuracy of the majority vote: (5 C 3)* 0.6^3 * 0.4^2 + (5 C 4) * 0.6^4* 0.4 + (5 C 5) * 0.6^5 = 68.2%
  - n C m is n!/(m! * (n-m)!)
- The accuracy for 99 models: 97.8%

Bagging = Bootstrap Aggregating

- Randomly pick n samples with a replacement
  = bootstrap
- Build n models using each sample
- Average or vote the outputs of each model
  = aggregation

- Reduce Overfitting

Random Forest

- Bagging of Decision Trees

class sklearn.ensemble.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, ccp_alpha=0.0, max_samples=None)

- n_estimators : the number of decision trees, max_depth : the maximum depth of one decision tree

Boosting

- Compute an initialized model with equal weights to data points
- Compute residuals (how far each data point from the model) and compute the model to learn the residuals
- Repeat

- Works better with more complex data

Gradient Boosting

- Boosting of Decision Trees

- class sklearn.ensemble.GradientBoostingClassifier(loss='deviance', learning_rate=0.1, n_estimators=100, subsample=1.0, criterion='friedman_mse', min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_depth=3, min_impurity_decrease=0.0, min_impurity_split=None, init=None, random_state=None, max_features=None, verbose=0, max_leaf_nodes=None, warm_start=False, presort='deprecated', validation_fraction=0.1, n_iter_no_change=None, tol=0.0001, ccp_alpha=0.0)