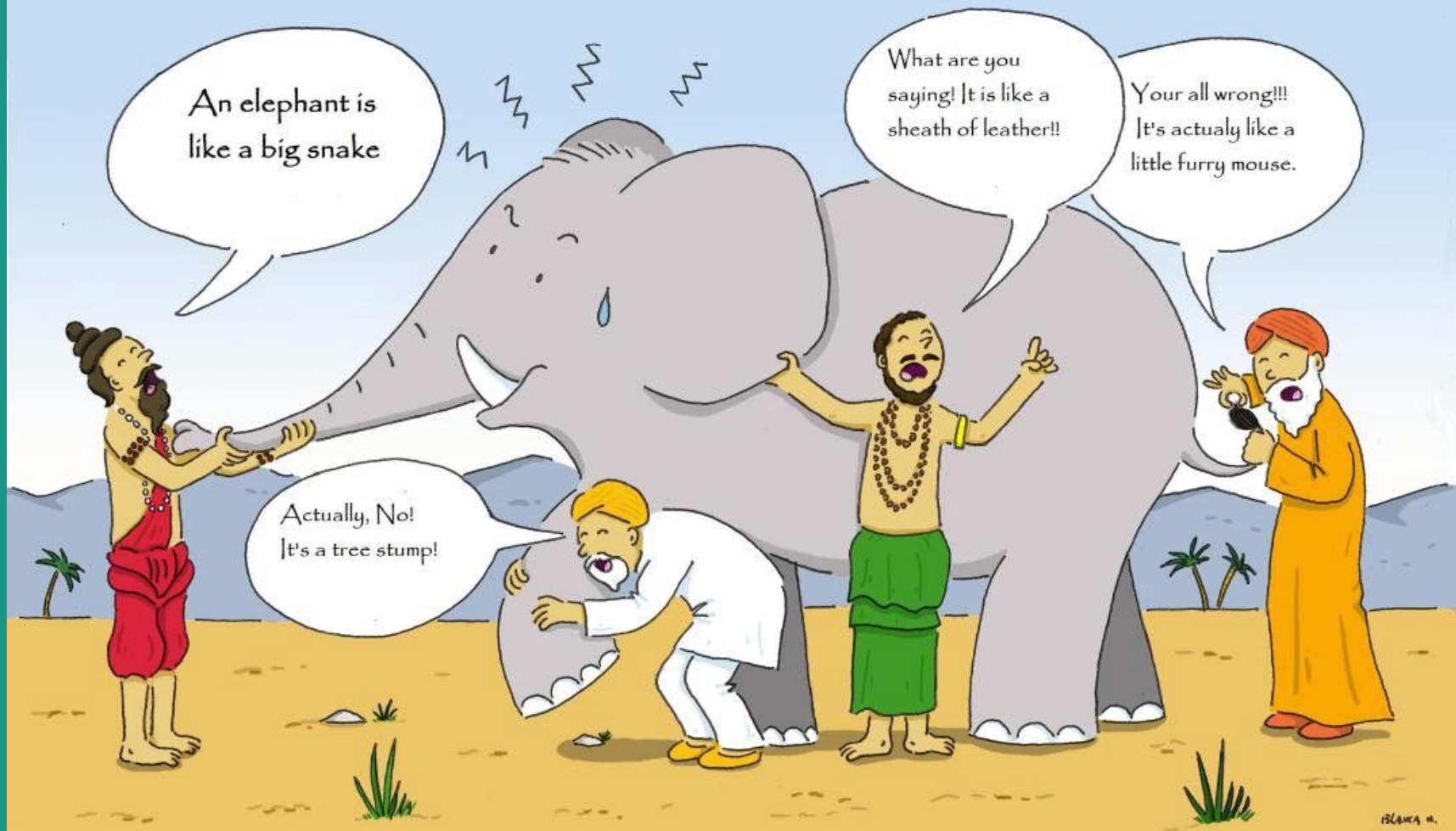




Ensemble Learning

Ayush Thada(16BCE1333)

ayush.thada2016@vitstudent.ac.in





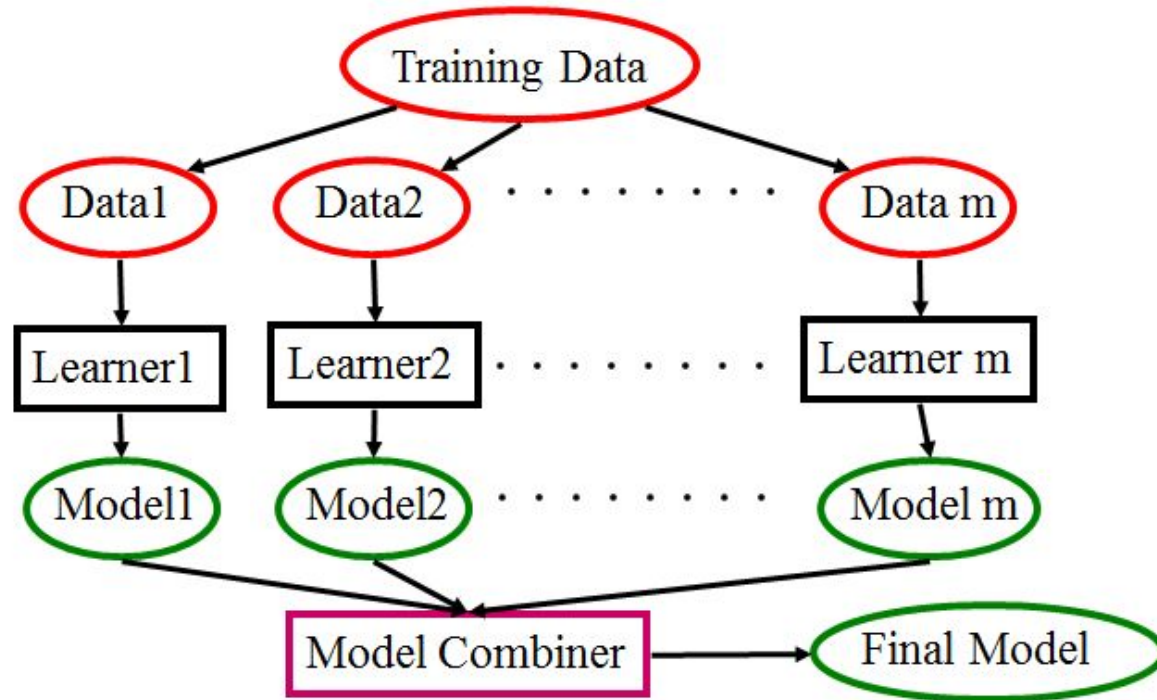
Topics for Discussion

- Introduction
- Value of Ensembles
- Intuition
- Classification
 - Bagging
 - Boosting
 - Adaptive Boosting
 - Gradient Boosting
 - Stacking
- Other Ensembling Techniques
- Advantages
- Disadvantages
- References

Introduction

- So far – learning methods that learn a single hypothesis, chosen from a hypothesis space that is used to make predictions.
- Ensemble learning à select a collection (ensemble) of hypotheses and combine their predictions.
- Example 1 - generate 100 different decision trees from the same or different training set and have them vote on the best classification for a new example.
- Key motivation: reduce the error rate. Hope is that it will become much more unlikely that the ensemble of will misclassify an example.

- Learn multiple alternative definitions of a concept using different training data or different learning algorithms.
- Combine decisions of multiple definitions, e.g. using weighted voting.



Value of Ensembles

- “No Free Lunch” Theorem
 - No single algorithm wins all the time!
- When combining multiple independent and diverse decisions each of which is at least more accurate than random guessing, random errors cancel each other out, correct decisions are reinforced.
- Examples: Human ensembles are demonstrably better
 - How many jelly beans in the jar?: Individual estimates vs. group average.
 - Who Wants to be a Millionaire: Audience vote.


















































Reality							
1							
2							
3							
4							
5							
Combine							

Fig: Weather Forecasting

—

Intuition

Majority vote

- Suppose we have 5 completely independent classifiers...
 - If accuracy is 70% for each
 - $(.7^5) + 5(.7^4)(.3) + 10(.7^3)(.3^2)$
 - **83.7% majority vote accuracy**
 - 101 such classifiers
 - **99.9% majority vote accuracy**

[Note] Binomial Distribution: The probability of observing x heads in a sample of n independent coin tosses, where in each toss the probability of heads is p , is

$$P(X = x|p, n) = \frac{n!}{x!(n-x)!} p^x (1 - p)^{n-x}$$

- Another way of thinking about ensemble learning:

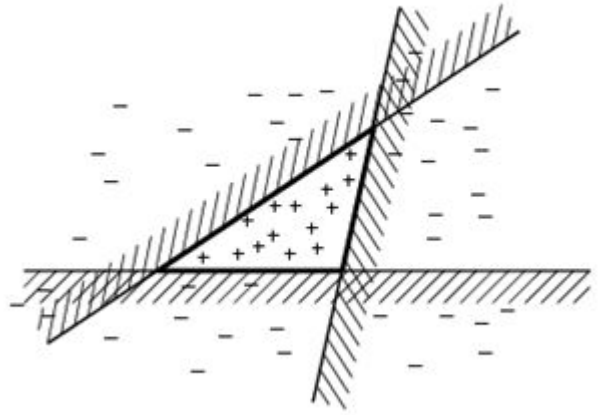
It's a way of enlarging the hypothesis space, i.e., the ensemble itself is a hypothesis and the new hypothesis space is the set of all possible ensembles constructible from hypotheses of the original space.

- Increasing power of ensemble learning:

Three linear threshold hypothesis

Ensemble classifies as positive any example

Classified positively by all three. The resulting triangular region hypothesis is not expressible in the original hypothesis space.



Different learners can have

- Different learning algorithms
- Algorithms with different choice for parameters
- Data set with different features
- Data set = different subsets or we can say different sampling strategy.

Classification

- Homogeneous Ensemble

- Use a single, arbitrary learning algorithm but manipulate training data to make it learn multiple models.

- $\text{Data1}^1 \neq \text{Data2}^1 \neq \dots \neq \text{Data } m$

- $\text{Learner } 1 = \text{Learner } 2 = \dots = \text{Learner } m$

- Examples:

- Bagging: Resample training data

- Boosting: Reweight training data

- Heterogeneous Ensemble

- Use multiple learning algorithm with similar data.
- Use multiple learning algorithm with manipulated data.
- Examples:
 - Voting
 - Bucket of Models

[Note] In WEKA, these are called meta-learners, they take a learning algorithm as an argument (base learner) and create a new learning algorithm.

- Different ensemble methods are used for different purposes. We can use **Bias-Variance tradeoff** as a criterion to decide which method to use:
 - Ensemble method to reduce Bias:
 - Bagging
 - Random Forests
 - Ensemble methods to reduce variance:
 - Functional Gradient Boosting
 - Boosting
 - Ensemble Selection

Bagging

- Create ensembles by “bootstrap aggregation”, i.e., repeatedly randomly resampling the training data (Breiman, 1996).
- Bootstrap: draw N items from X with replacement.
- Bagging
 - Train M learners on M bootstrap samples
 - Combine outputs by voting (e.g., majority vote)
 - Decreases error by decreasing the variance in the results due to unstable learners, algorithms (like decision trees and neural networks) whose output can change dramatically when the training data is slightly changed.

- Algorithm

- Given a standard training set D of size n
 - For $i = 1 \dots M$
 - -Draw a sample of size $n^* < n$ from D uniformly and with replacement
 - -Learn classifier C_i
 - Final classifier is a vote of $C_1 \dots C_M$
- Increases classifier stability/reduces variance.

Boosting

- **Strong Learner**

- Take labeled data for training
- Produce a classifier which can be **arbitrarily accurate.**
- Strong learners are very difficult to construct

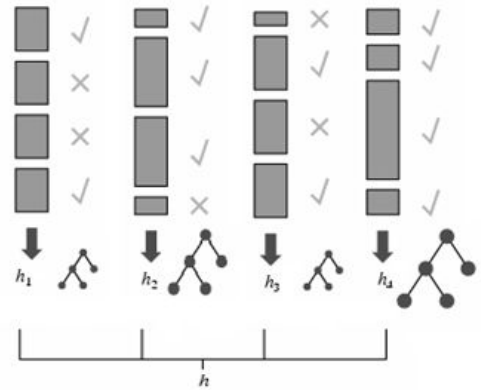
- **Weak Learner**

- Take labeled data for training
- Produce a classifier which is **more accurate than random guessing**
- They only need to generate a hypothesis with a training accuracy greater than 0.5, i.e., < 50% error over any distribution.
- Constructing weaker Learners is relatively easy.

- Originally developed by computational learning theorists to guarantee performance improvements on fitting training data for a weak learner that only needs to generate a hypothesis with a training accuracy greater than 0.5 (Schapire, 1990).
- Revised to be a practical algorithm, AdaBoost, for building ensembles that empirically improves generalization performance (Freund & Shapire, 1996).
- Key Insights
 - Instead of sampling (as in bagging) re-weight examples!
 - Examples are given weights. At each iteration, a new hypothesis is learned (weak learner) and the examples are reweighted to focus the system on examples that the most recently learned classifier got wrong.
 - Final classification based on weighted vote of weak classifiers

Adaptive Boosting

- Each rectangle corresponds to an example, with weight proportional to its height.
- Crosses correspond to misclassified examples.
- Size of decision tree indicates the weight of that hypothesis in the final ensemble.



Construction of a Weak Classifier

- Using Different Data Distribution
 - Start with uniform weighting
 - During each step of learning
 - Increase weights of the examples which are not correctly learned by the weak learner
 - Decrease weights of the examples which are correctly learned by the weak learner
- Idea
 - Focus on difficult examples which are not correctly classified in the previous steps.

Combining Weak Classifier

- Weighted Voting
 - Construct strong classifier by weighted voting of the weak classifiers.
- Idea
 - Better weak classifier gets a larger weight.
 - Iteratively add weak classifiers.
 - Increase accuracy of the combined classifier through minimization of a cost function.

Algorithm

- $C = 0$ /* counter */
- $M = m$ /* number of hypotheses to generate */
- Set same weight for all the examples (typically each example has weight = 1)
- While ($C < M$)
 - Increase counter C by 1.
 - Generate hypothesis h_C .
 - Increase the weight of the misclassified examples in hypothesis h_C
- Weighted majority combination of all M hypotheses (weights according to how well it performed on the training set).

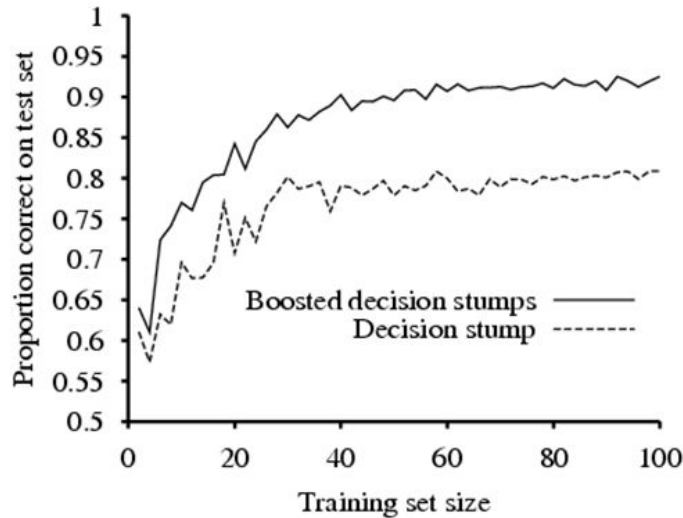
[Note]: Many variants depending on how to set the weights and how to combine the hypotheses.

ADABOOST quite popular!!!!

Performance of AdaBoost

- Learner = Hypothesis = Classifier
- Weak Learner: $< 50\%$ error over any distribution
- M number of hypothesis in the ensemble.
- If the input learning is a Weak Learner, then ADABOOST will return a hypothesis that classifies the training data perfectly for a large enough M, boosting the accuracy of the original learning algorithm on the training data.
- Strong Classifier: thresholded linear combination of weak learner outputs.

[Note]: Boosting approximates Bayesian Learning, which can be shown to be an optimal learning algorithm.



Training error reaches zero for $M=20$ (as predicted by the theorem), and remains zero as more stumps are added to the ensemble.

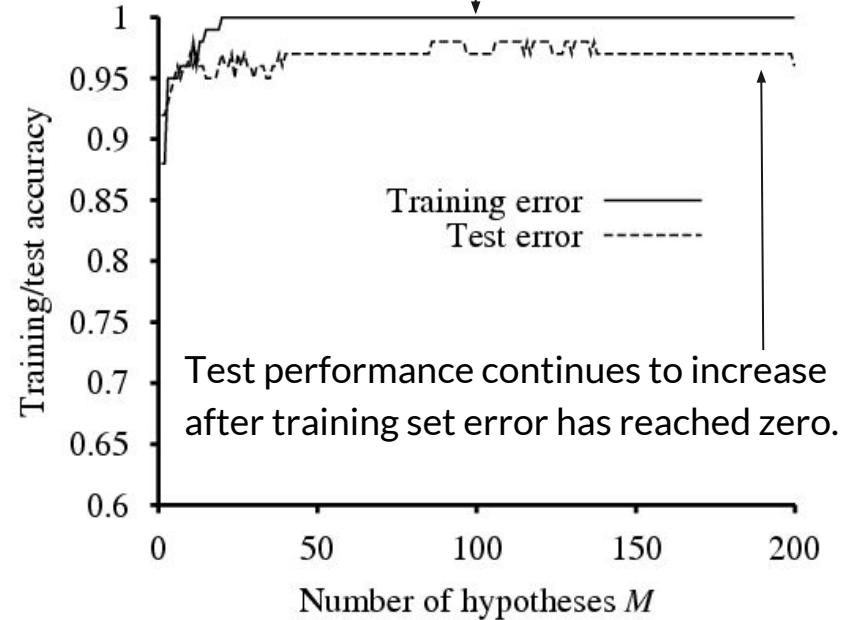


Fig: Analysis of Restaurant Dataset



Gradient Boosting

- Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees.
- The objective of any supervised learning algorithm is to **define a loss function and minimize it.**
- Go through this demo

http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

Algorithm

1. Initialize $f_0(x) = \arg \min_{\gamma} \sum_{i=1}^N L(y_i, \gamma)$.

2. For $m = 1$ to M :

a. For $i = 1, 2, \dots, N$ compute

$$r_{im} = - \left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)} \right]_{f=f_{m-1}}$$

b. Fit a regression tree to the targets r_{im} giving terminal regions

$$R_{jm}, j = 1, 2, \dots, J_m.$$

c. For $j = 1, 2, \dots, J_m$ compute

$$\gamma_{jm} = \arg \min_{\gamma} \sum_{x_i \in R_{jm}} L(y_i, f_{m-1}(x_i) + \gamma).$$

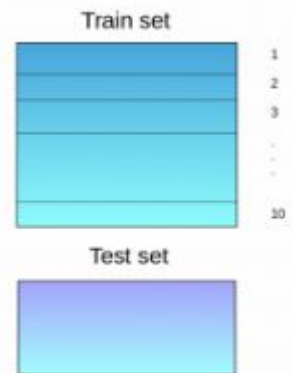
d. Update $f_m(x) = f_{m-1}(x) + \sum_{j=1}^{J_m} \gamma_{jm} I(x \in R_{jm})$.

3. Output $\hat{f}(x) = f_M(x)$.

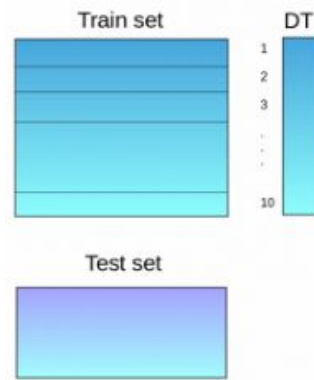
Stacking

- Stacking is an ensemble learning technique that uses predictions from multiple models (for example decision tree, knn or svm) to build a new model. This model is used for making predictions on the test set. Below is a step-wise explanation for a simple stacked ensemble:

Step1: The train set is split into 10 parts.

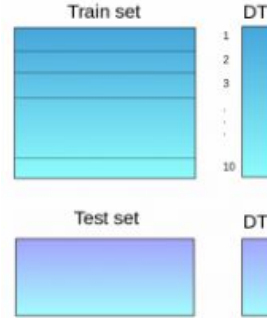


Step2: A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.

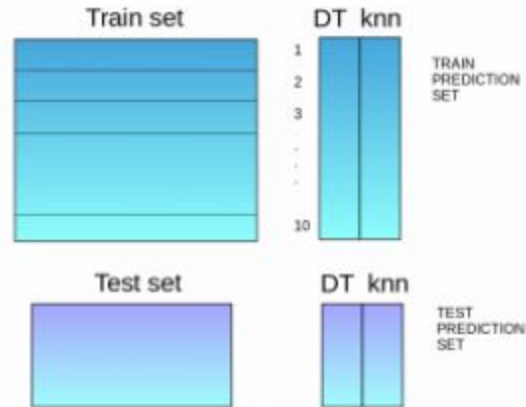


Step3: The base model (in this case, decision tree) is then fitted on the whole train dataset.

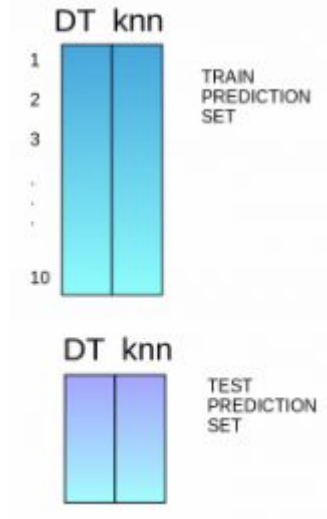
Step4: Using this model, predictions are made on the test set.



Step5: The base model (in this case, decision tree) is then fitted on the whole train dataset.



Step6: The predictions from the train set are used as features to build a new model.



Step7: This model is used to make final predictions on the test prediction set.

Other Ensemble Techniques

—

- There are several other ensemble techniques are there which are not discussed in slides.
 - Bayes optimal classifier
 - Bayesian parameter averaging
 - Bayesian model combination
 - Bucket of models
 - Voting

[Note]: *These topics are not the part of the course. Interested student can ask me to explain these during my free hours.*

Advantages

- Intuitively, ensembles allow the different needs of a difficult problem to be handled by hypotheses suited to those particular needs.
- Mathematically, ensembles provide an extra degree of freedom in the classical bias/variance tradeoff, allowing solutions that would be difficult (if not impossible) to reach with only a single hypothesis.
- They're unlikely to overfit.
- These models are more stable as compared to independent models.
- The aggregate opinion of a multiple models is less noisy than other models. In finance, we called it “Diversification” a mixed portfolio of many stocks will be much less variable than just one of the stocks alone.

Disadvantages

- The model that is closest to the true data generating process will always be best and will beat most ensemble methods. So if the data come from a linear process, linear models will be much superior to ensemble models.
- Ensemble models suffer from lack of interpretability. Sometimes we need predictions and explanations of the predictions. Variable importance analysis can help with insights, but if the ensemble is more accurate than a linear additive model, the ensemble is probably exploiting some non-linear and interaction effects that the variable importance analysis can't completely account for.
- Ensemble methods are usually computationally expensive. Therefore, they add learning time and memory constraints to the problem.

References

- https://en.wikipedia.org/wiki/Ensemble_learning
- <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-for-ensemble-models/>
- http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html
- <http://www.cs.cornell.edu/courses/cs4700/2008fa/PPT/CS4700-EL.ppt>
- <https://www.youtube.com/watch?v=UHBmv7qCey4>
- <https://www.youtube.com/watch?v=P76Gy2eg46A&list=PLehuLRPyt1Hy-4ObWBK4Ab0xk97s6imfC&index=16>
- <https://web.stanford.edu/~hastie/TALKS/boost.pdf>
- Tom Mitchell's Machine Learning Book