FLATIRON SCHOOL

Hybridizing Learners with Ensemble Algorithms.

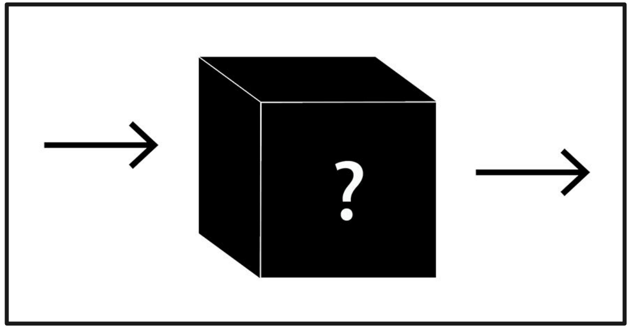
DS

hosted by KASH

CRITICAL QUESTION

How can we combine the power of many basic classifiers in order to perform optimal signal learning with improved generalizability and robustness?

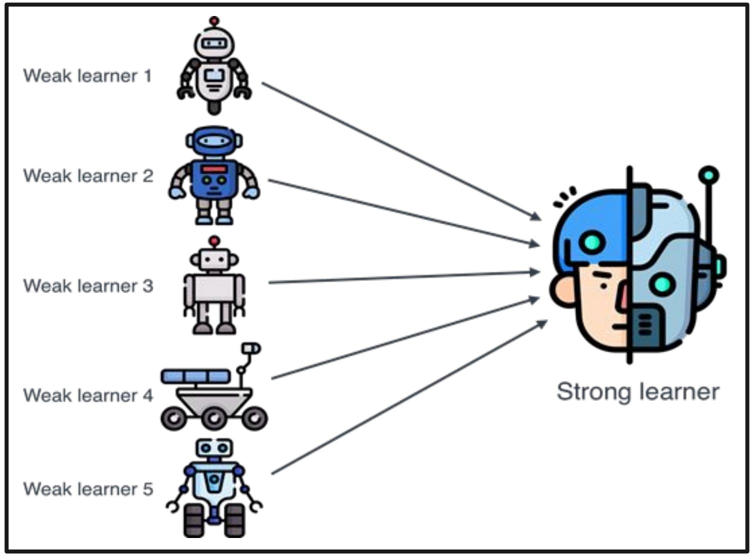
HISTORICAL MACHINE LEARNING: SUPER SOLDIER MODELS



Most of the algorithms we've interfaced with throughout this course have been heavily-engineered **single learners**: individual algorithms that ideally have enough **in-built mathematical and technical complexity** to handle processing large quantities of data.

However, this isn't always the case – indeed, the **growing sums of data** combined with **computing limitations** have given rise to entirely new methods of conducting machine learning… ***why use one model when we can use many?***

MODERN MACHINE LEARNING: LEGIONS OF MODELS



As machine learning has advanced, new ensembling methods have developed in order to combat the difficulties of data demand and computational complexity: **ensemble methods** combine multitudes of **“weak learners”** together in some summative or aggregative manner in order to **reduce the impact of noise** and **retain generative signal to improve overall accuracy**.

CRITICAL QUESTION

How can we utilize bagging, boosting, and stacking algorithms to predict new data?

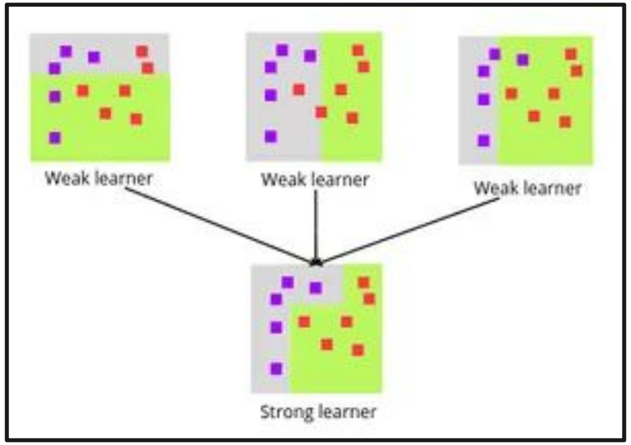
WRITTEN IDEATION

RANK PREVIOUSLY COVERED CLASSIFIERS BASED ON ALGORITHMIC COMPLEXITY.

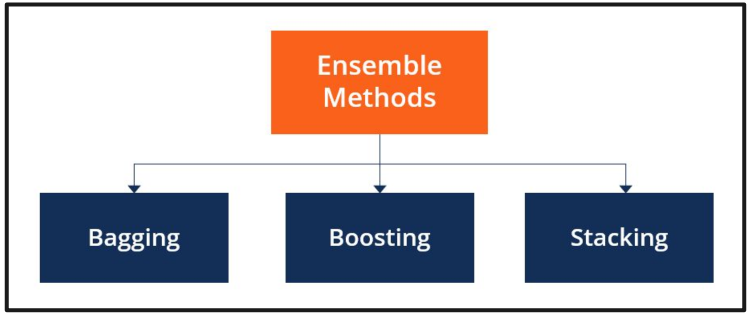
UNITED WE STAND, DIVIDED WE… KINDA SORTA STAND

Much of the inspiration behind ensemble models have to do with combating the **bias-variance tradeoff** in machine learning: the notion that **model complexity must be balanced** in a way such that **underfitting to high biases** and **overfitting to high variances** can both be avoided.

Ensembles attempt to solve this problem by constructing **weak learners** (that are vulnerable to high bias/variance) and combining many of them to create a **strong learner** with optimal performance and reduced noise impact.



THE MAJOR PILLARS OF ENSEMBLE MODELING

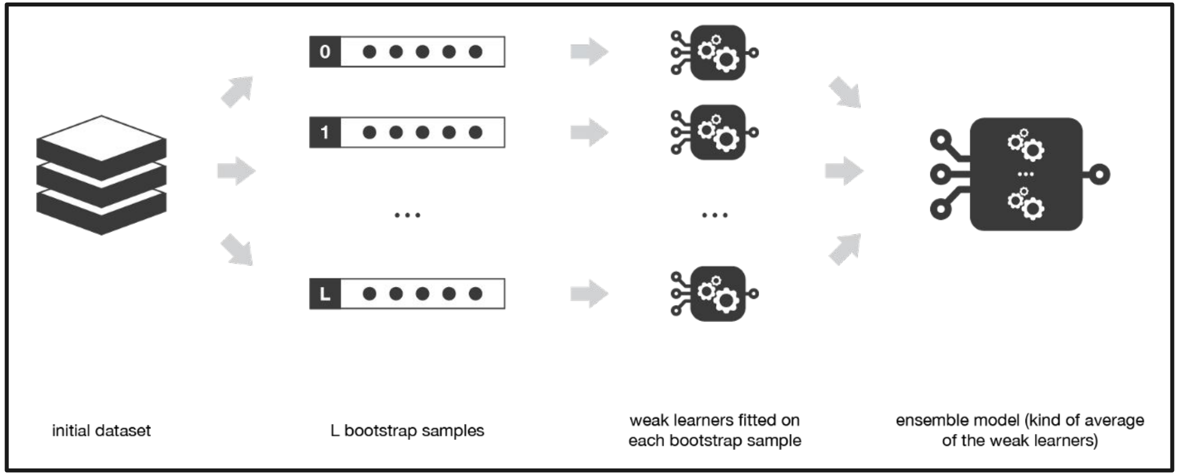


The actual ways that ensemble models function in terms of **converging results from multiple weak learners into a strong learner** depends greatly on the intricacies of how the models are setup and mathematically function.

The three major archetypes of ensemble model design are known as **bagging**, **boosting**, and **stacking** algorithms; they differ primarily based on the **objective and design of the converged strong learner**.

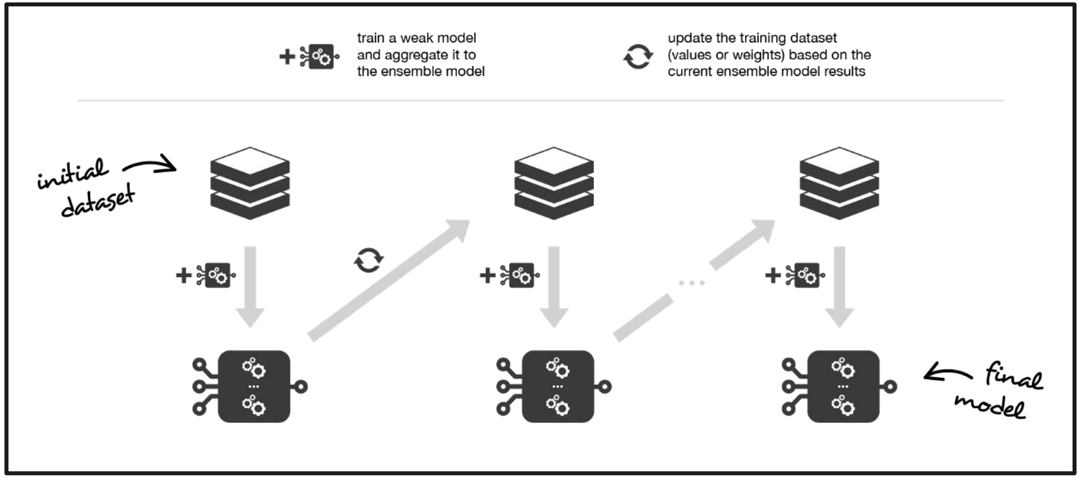
ENSEMBLE MODELING: BAGGING

Bagging (Bootstrap Aggregation) aims to **decrease the overall variance** of our model's predictions by constructing **"approximately independent" models and "averaging" their predictions** in order to obtain a single model with a lower variance that doesn't have to be trained on exponentially more data.


0
1
...
L
...
initial dataset
L bootstrap samples
weak learners fitted on each bootstrap sample
ensemble model (kind of average of the weak learners)

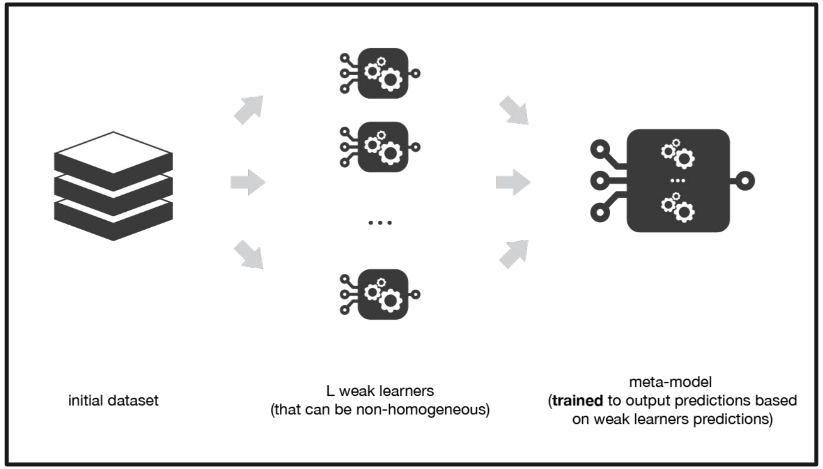
ENSEMBLE MODELING: BOOSTING

Boosting works similarly to produce a family of weak learners that are aggregated to attain a strong learner – however, boosting aims to **reduce bias** by **sequentially fitting weak learners** to data segments with the goal of **systematically improving fitness across particularly hard-to-learn data**.


train a weak model and aggregate it to the ensemble model
update the training dataset (values or weights) based on the current ensemble model results
initial dataset
...
final model

ENSEMBLE MODELING: STACKING

Stacking is significantly unique to the former two methods: they leverage **different parallelized learning algorithms (heterogeneous weak learners)** that are then combined in *"stacks"* and processed by **non-deterministic meta-models** in order to perform final predictions.


initial dataset
L weak learners
(that can be non-homogeneous)
meta-model
(trained to output predictions based
on weak learners predictions)

WRITTEN IDEATION

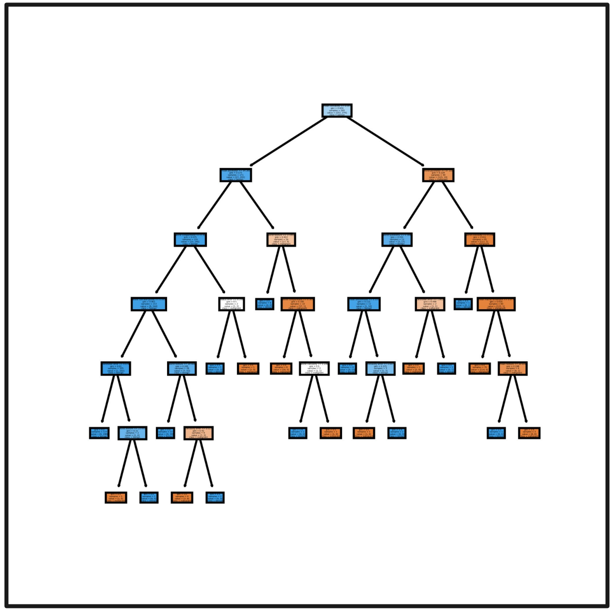
BRAINSTORM **WHICH TYPE OF ENSEMBLE MODEL MAY BE OPTIMAL** FOR A PARTICULAR **MACHINE LEARNING TASK.**

THE VAST OCEAN OF ENSEMBLING ALGORITHMS

Because of the many different ways that ensemble methods can be constructed based on their **three unique schools of thought** as well as the nuance of **leveraging virtually any model as a weak learner**, ensembling algorithms can compose an incredibly vast subfield of machine learning.

Luckily, most of the field has recognized **major archetypes of tree-based ensembles** that can be treated as their own independent algorithms to work with.

BAGGING ENSEMBLES: BAGGED DECISION TREES



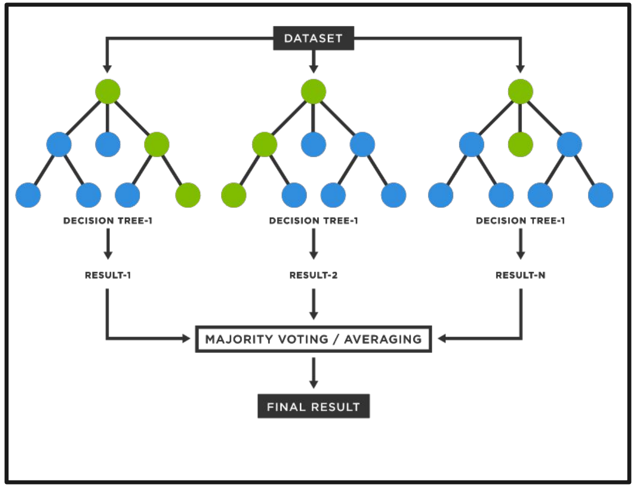
Bagged Decision Trees are a simple application of the concept of bagging: **multiple shallow decision trees** are created that are then trained across **bootstrapped samples** of the data.

The output is then aggregated from all the shallow trees to construct **one larger, stronger tree that then predicts the output**.

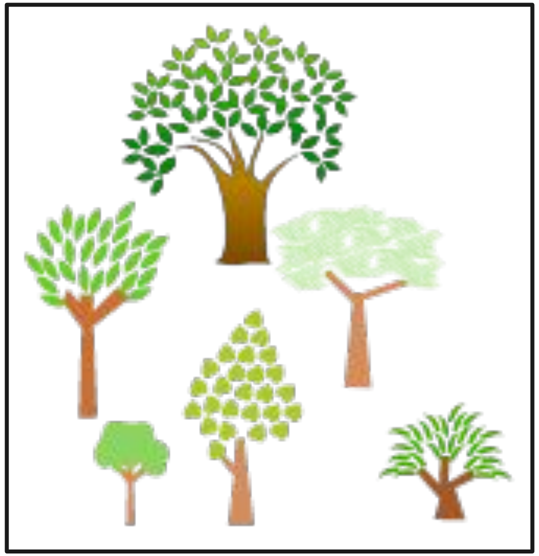
BAGGING ENSEMBLES: RANDOM FORESTS

Random Forests are some of the most popular ensemble models in all of machine learning due to their simplicity and effectiveness for learning complex patterns.

They're remarkably similar to bagged decision trees: however, the major advantageous difference is that they introduce **randomness in how each shallow learner creates decision splits** (instead of searching for the most important feature, **it searches for the best feature among a random selection of features**).



BAGGING ENSEMBLES: EXTRA TREES

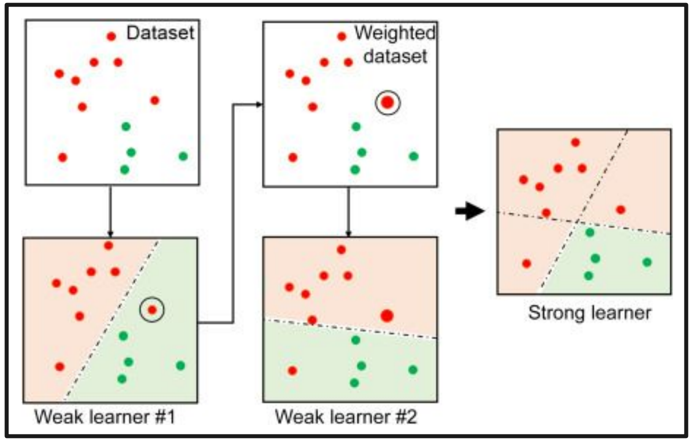


The **Extremely Randomized Trees (Extra Trees)** algorithm is quite similar to random forests as well – however, the major difference in design paradigm is that unlike how random forests fit each shallow learner to a bootstrapped subsection of the dataset, **Extra Trees creates a very high number of very shallow trees that are each fitted across the entire training dataset**.

BOOSTING ENSEMBLES: ADABOOST

The **Adaptive Boosting (AdaBoost)** algorithm is one of the most original and groundbreaking boosting classifiers of its generation.

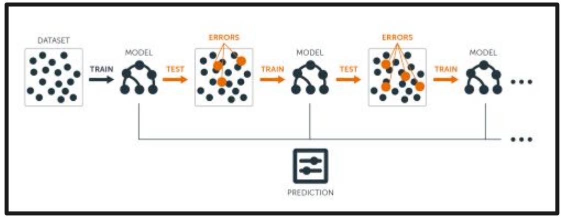
AdaBoost produces very shallow decision trees (called **“decision stumps”**) that are trained in sequence: over time, **AdaBoost places more “training weight” on difficult-to-classify data points** so that smarter stumps can adapt and learn the overall data better.



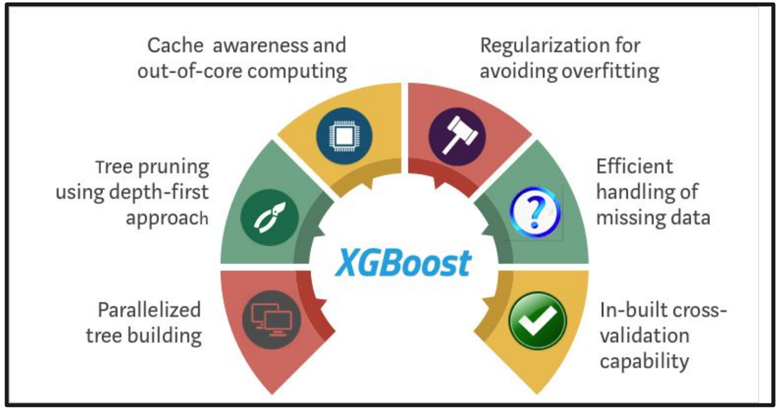
BOOSTING ENSEMBLES: STOCHASTIC GRADIENT BOOSTING

Over time, newer generations of boosting algorithms were designed to generalize over data through the inclusion of a **"loss function" for error minimization** – in this way, rather than trying to maximize a largely computer-uninterpretable accuracy metric, **boosters would minimize an overall error score over time**.

The **Stochastic Gradient Boosting** algorithm is one such prime example that leverages random sampling of the training data across each step alongside cost minimization to reduce error and maximize accuracy.



BOOSTING ENSEMBLES: XGBOOST

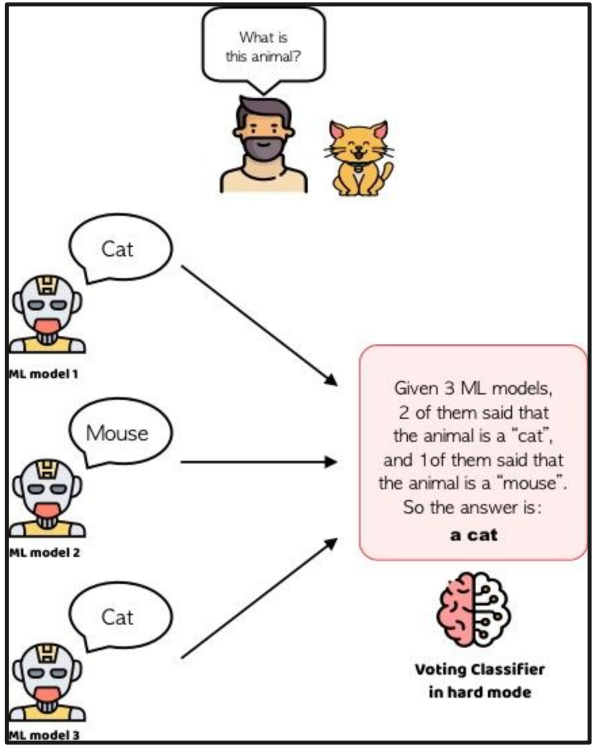

Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient handling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

XGBoost (or "eXtreme Gradient Boosting") is a **powerful extension of generalized gradient boosting algorithms** that incorporates a plethora of performance improvement techniques in order to optimize robustness across new data, including **parallel processing**, **tree-pruning**, **implicit null value imputation**, and a technique called "**regularization**" designed to **mitigate overfitting/bias**.

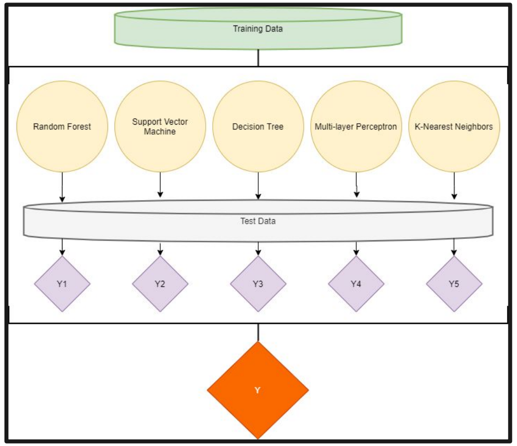
STACKING ENSEMBLES: VOTING

The essence of stacking models is to make best use of the variation of different stacked algorithms in order to provide best signal to the meta-learner down the line.

Voting ensembles involve a relatively simple method, where output predictions from the weak learners are evaluated by a **voting classifier** that chooses a winning prediction based on **"the most votes"** (a.k.a. **which class shows up more often or is more likely based on predictions of the weak learners**).


What is this animal?
Cat
ML model 1
Mouse
ML model 2
Cat
ML model 3
Given 3 ML models, 2 of them said that the animal is a "cat", and 1 of them said that the animal is a "mouse". So the answer is: a cat
Voting Classifier in hard mode

STACKING ENSEMBLES: WEIGHTED AVERAGING



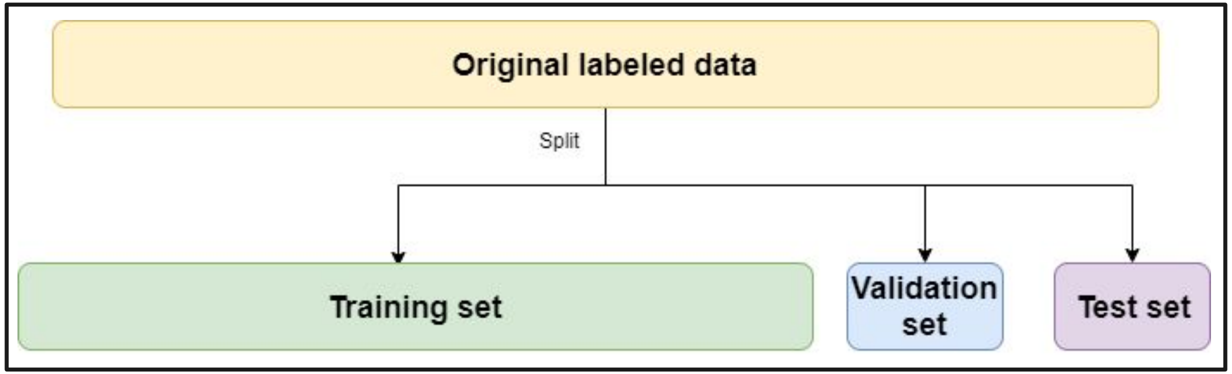
Weighted Averaging Ensembles are a slight deviation from the generalized stacking algorithm in that it assumes that **some models in the ensemble are "more skilled" than others** and **give them more contribution when making predictions**.

This is usually notated in the form of **weights** assigned to each model – these weights change over training and dictate how *"loud/soft"* the predictions from those models should be "heard" by the meta-learner.

STACKING ENSEMBLES: BLENDING

Blending is a technique derived from generalized stacking: the major difference is that rather than relying on cross-validation to segment data iteratively into two major segments (training, testing), **blending implements a "one-holdout set" which represents a small portion of the training data used for testing** in addition to true testing data.

This holdout set is often referred to as the **"validation set"**.



RECOMMENDED RESOURCE

BONUS CHALLENGE:

M.L.M. Tutorial on Bagging Ensembles

RECOMMENDED RESOURCE

BONUS CHALLENGE:

M.L.M. Tutorial on Bagging Ensembles with the Random Forest Algorithm

RECOMMENDED RESOURCE

BONUS CHALLENGE:

M.L.M. Tutorial on Stacking Ensembles

RECOMMENDED RESOURCE

BONUS CHALLENGE:

M.L.M. Tutorial on Boosting Ensembles with the AdaBoost Algorithm

RECOMMENDED RESOURCE

BONUS CHALLENGE:

M.L.M. Tutorial on Boosting Ensembles with the XGBoost Algorithm

THANK YOU
FOR YOUR TIME!