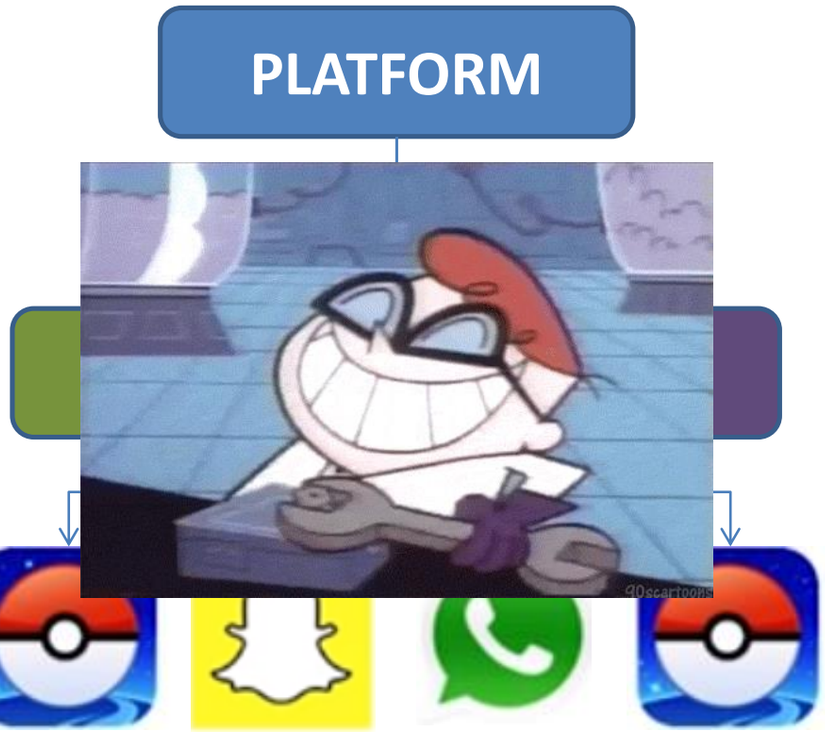
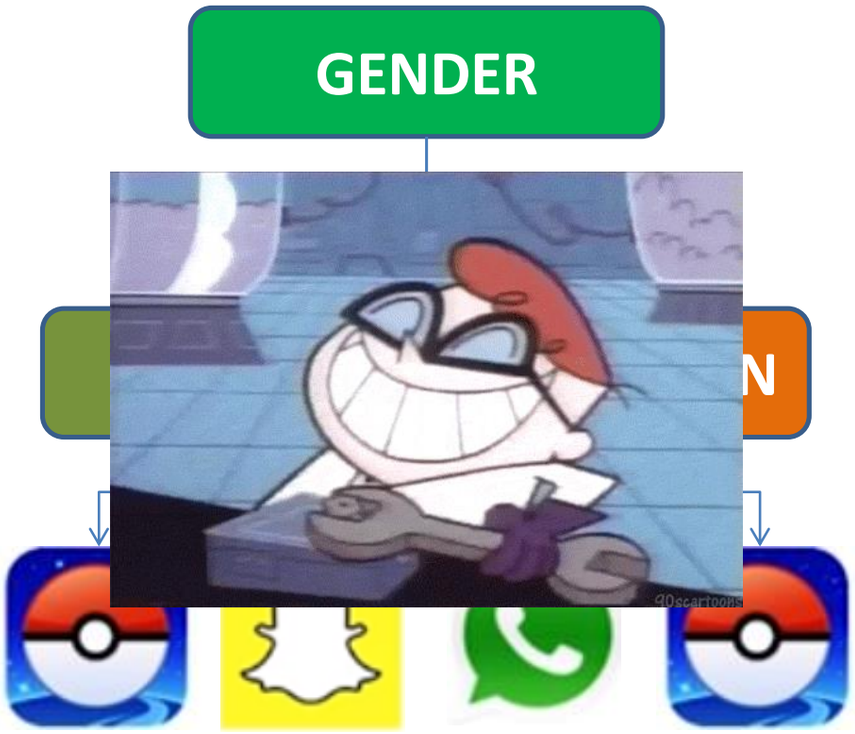
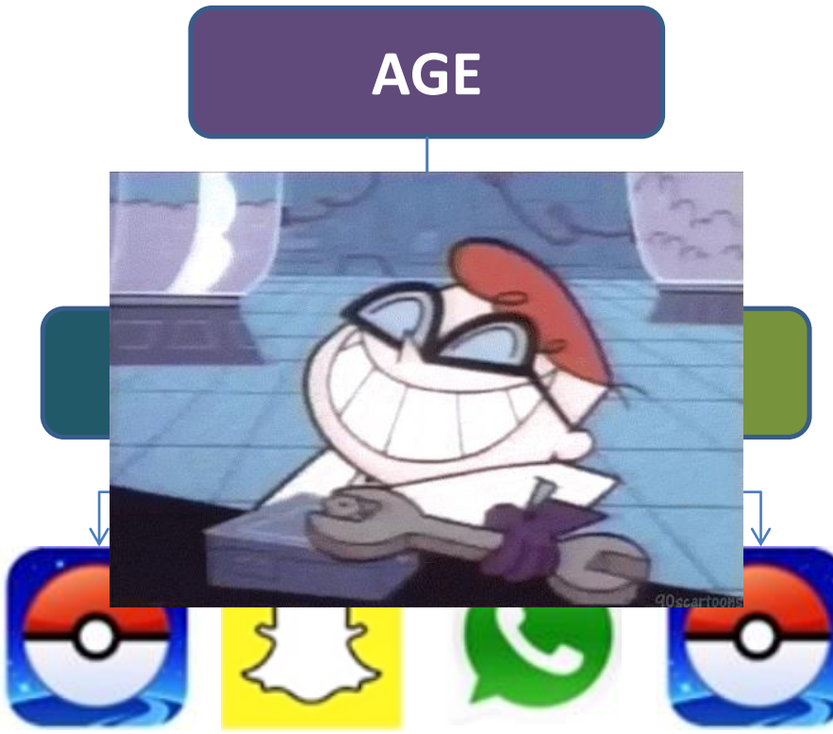
Random Forest

In Random Forest, we grow multiple trees as opposed to a single tree in CART model d. To classify a new object based on attributes, each tree gives a classification and we say the tree “votes” for that class. The forest chooses the classification having the most votes (over all the trees in the forest) and in case of regression, it takes the average of outputs by different trees.

Gender	Age	Location	Platform	Job	Hobby	App
F	15	US	iOS	School	Videogames	
F	25	France	Android	Work	Tennis	
M	32	Chile	iOS	Temp	Tennis	
F	40	China	iOS	Retired	Chess	
M	12	US	Android	School	Tennis	
M	14	Australia	Android	School	Videogames	



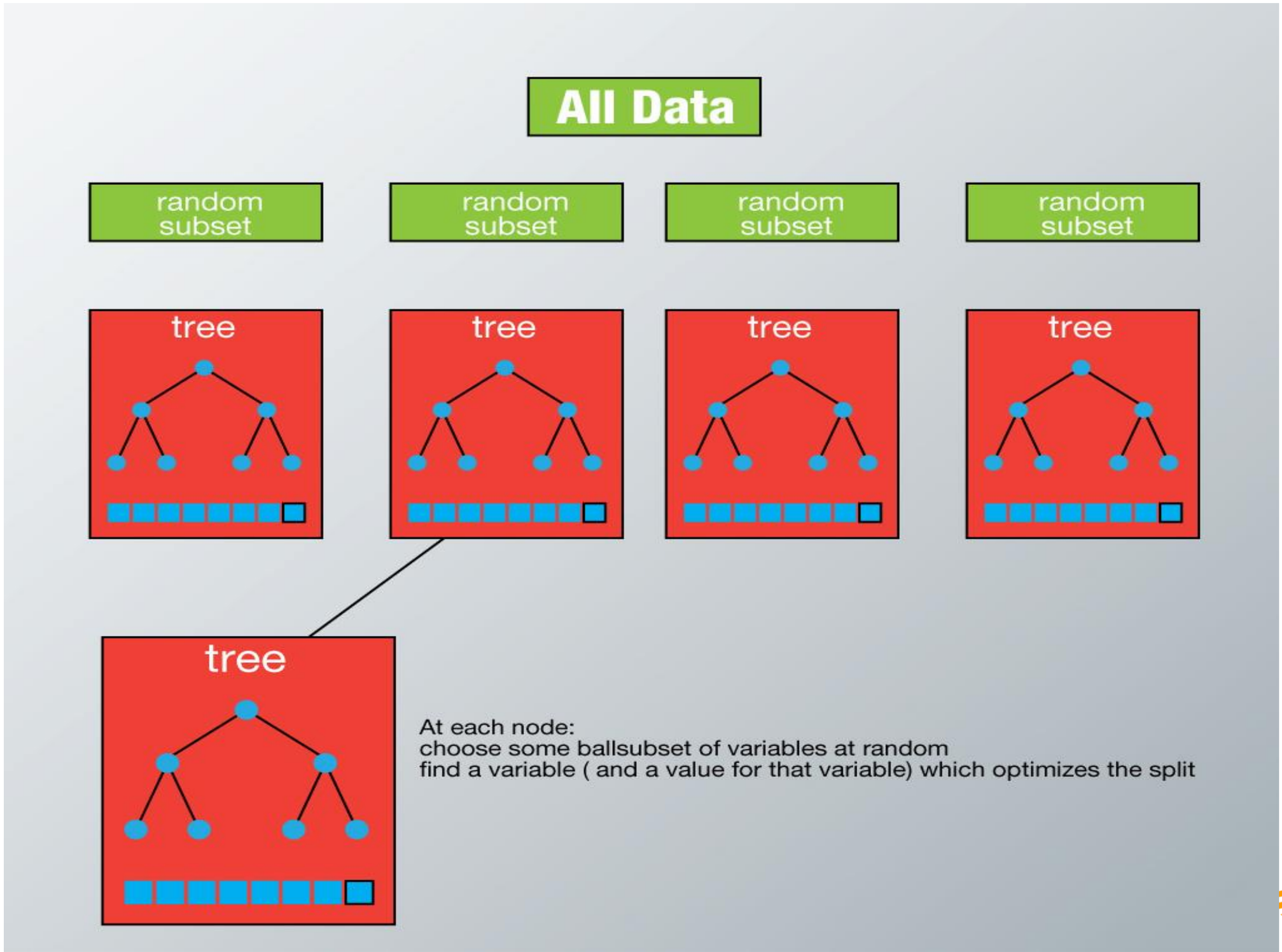




Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

How Random Forest Works?



Random Forest Hyperparameters

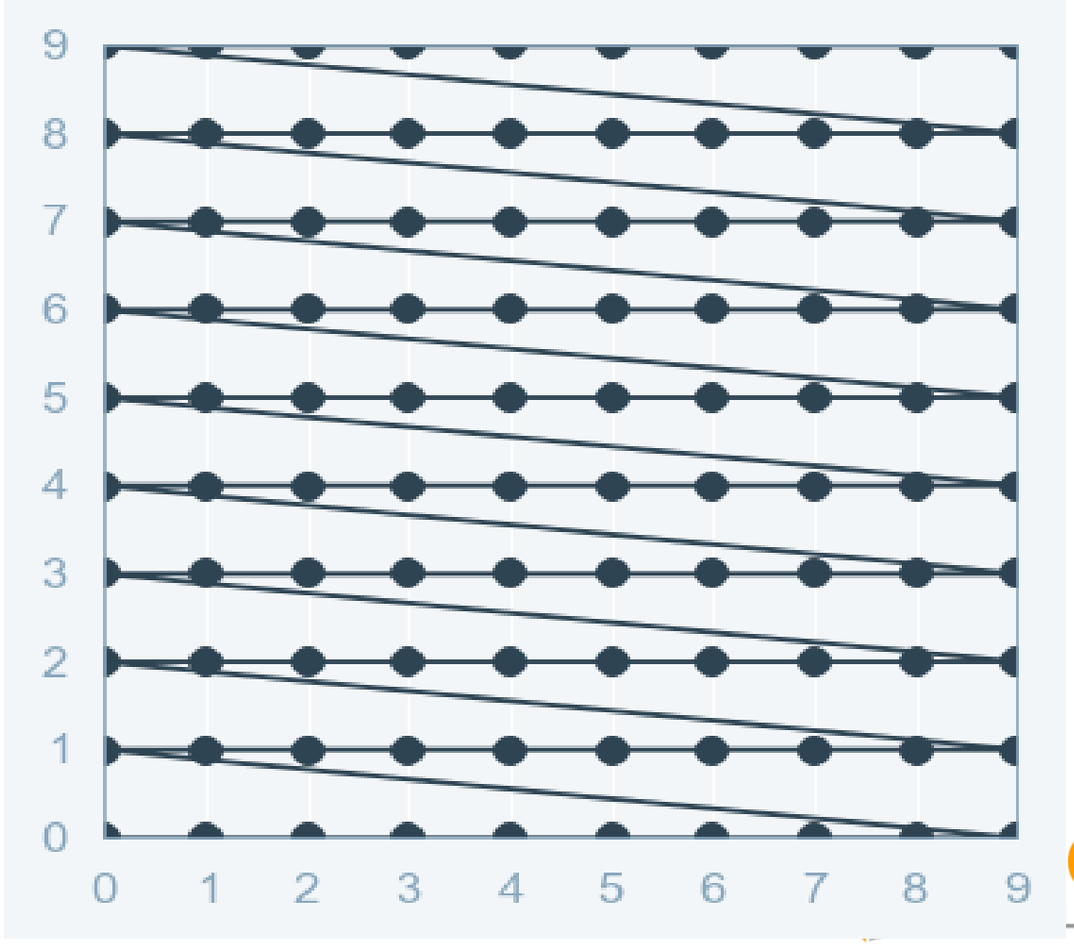
There are 4 hyperparameters required for a Random Forest classifier;

- The number of trees in the forest (n_estimators).
- The number of features to consider at each split. By default: square root of total number of features (max_features).
- The maximum depth of a tree i.e. number of nodes (max_depth).
- The minimum number of samples required to be at a leaf node / bottom of a tree (min_samples_leaf).

Tuning by GridSearchCV

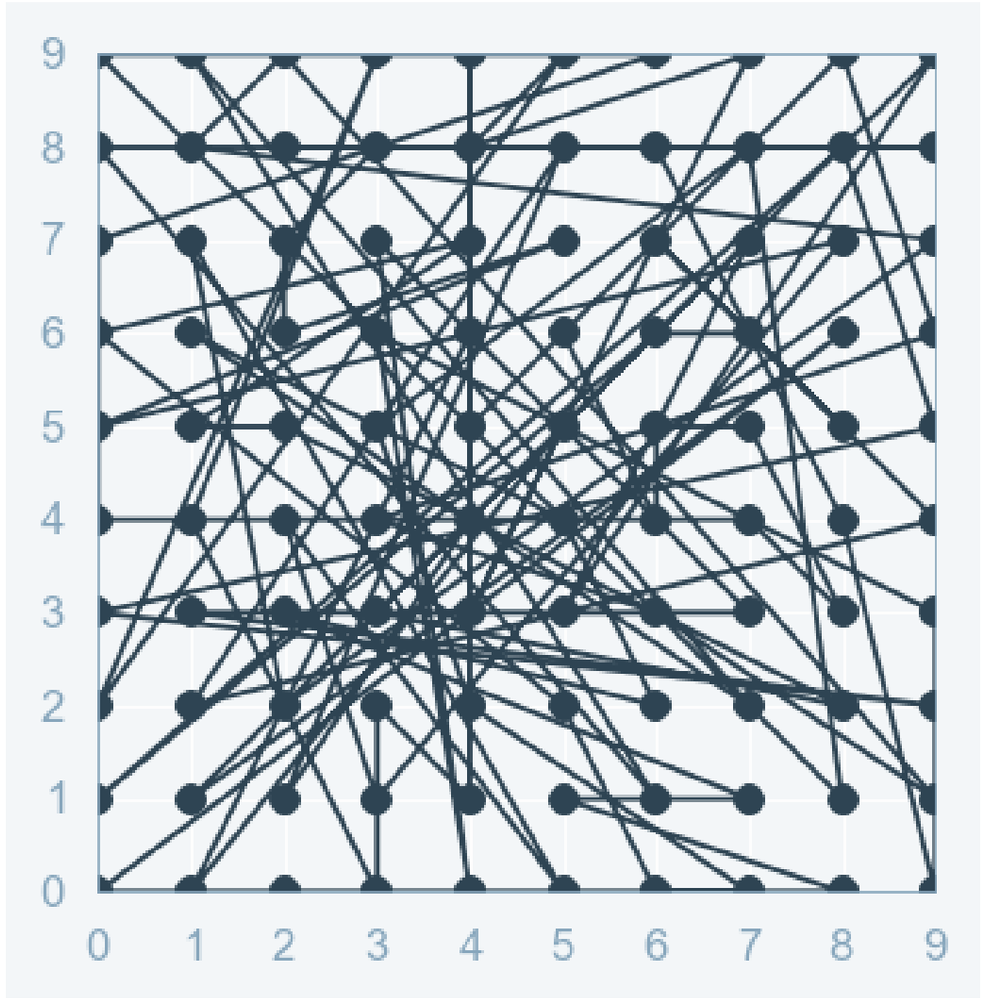
- We try every combination of a preset list of values of the hyper-parameters and evaluate the model for each combination.
- Pattern followed here is similar to the grid, where all the values are placed in the form of a matrix
- Once all the combinations are evaluated, the model with the set of parameters which give the top accuracy is considered to be the best.

max_depth \ min_samples_leaf	1	2	4	6	8
1	0.814	0.798	0.823	0.678	0.745
5	0.669	0.668	0.767	0.667	0.667
10	0.812	0.812	0.852	0.812	0.812
15	0.706	0.746	0.796	0.756	0.746
20	0.814	0.823	0.618	0.813	0.813



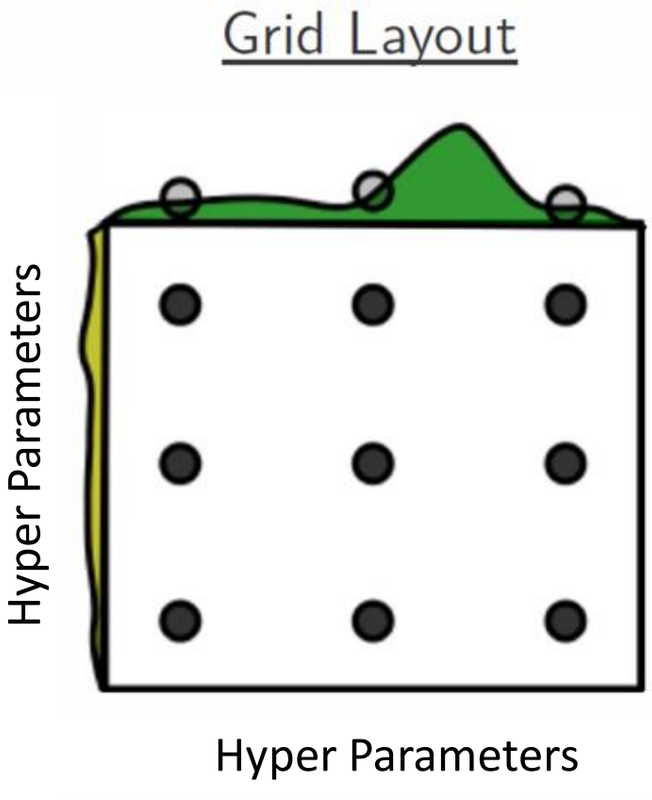
Tuning by RandomSearchCV

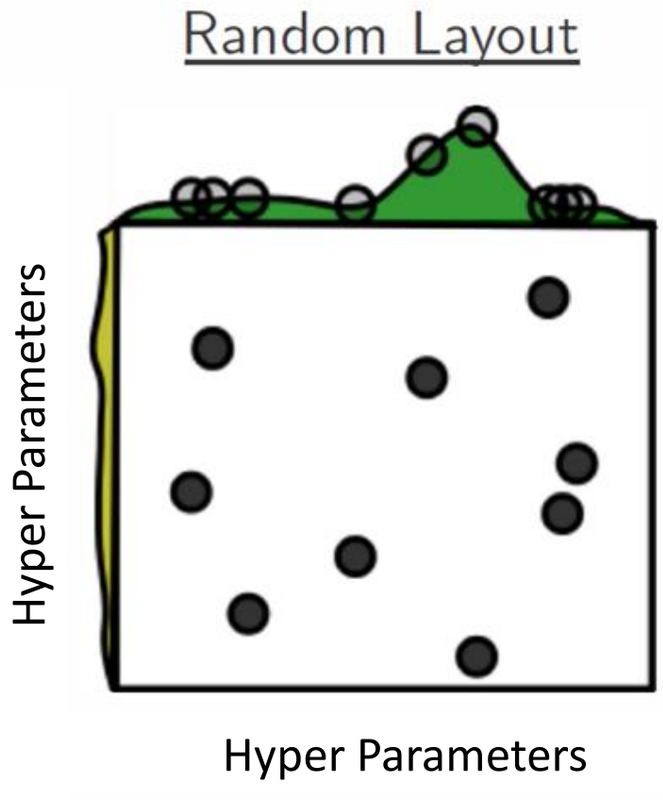
- Random search is a technique where random combinations of the hyper parameters are used to find the best solution for the built model
- It tries random combinations of a range of values
- To optimize with random search, the function is evaluated at some number of random configurations in the parameter space



GridSearch vs RandomSearch

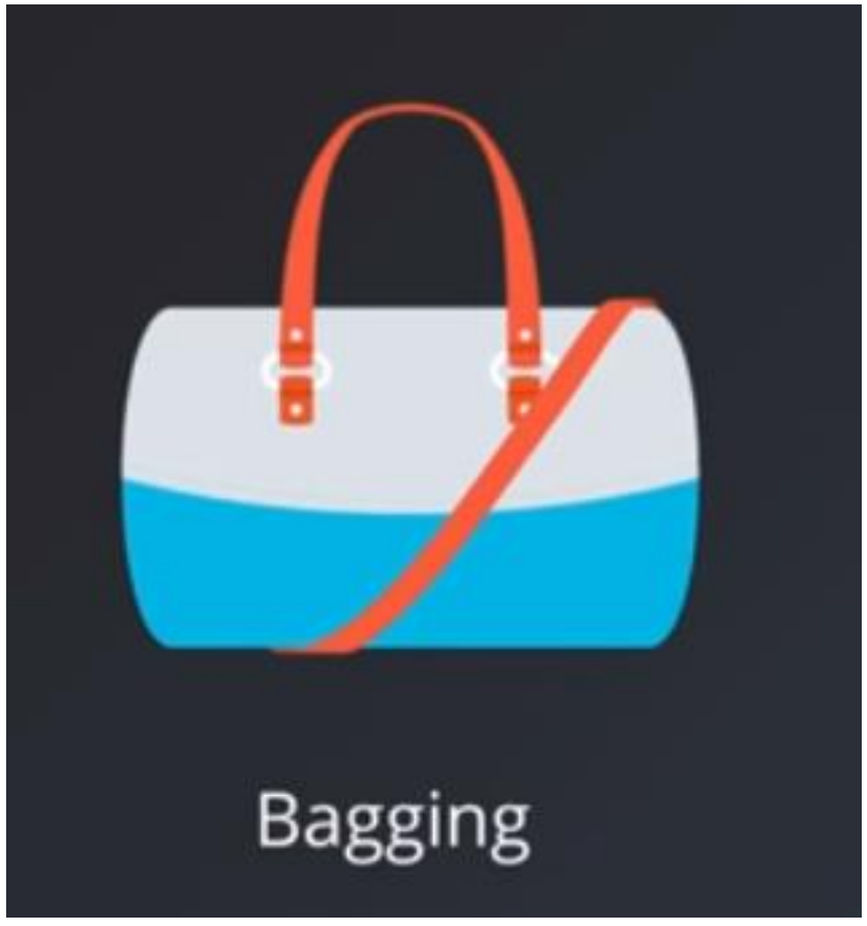
- One of the major drawbacks of grid search is that when it comes to dimensionality, it suffers when the number of hyper parameters grows exponentially.
- The chances of finding the optimal parameter are comparatively higher in random search because of the random search pattern where the model might end up being trained on the optimized parameters without any aliasing number of random configurations in the parameter space





Let’s Practice

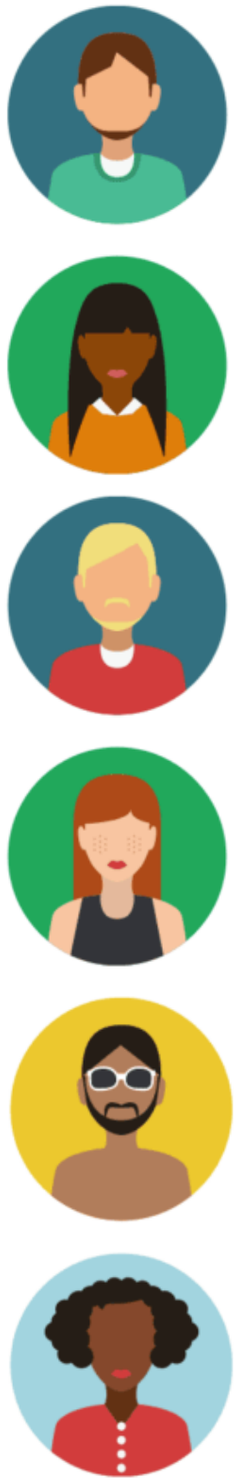
ENSEMBLE METHODS

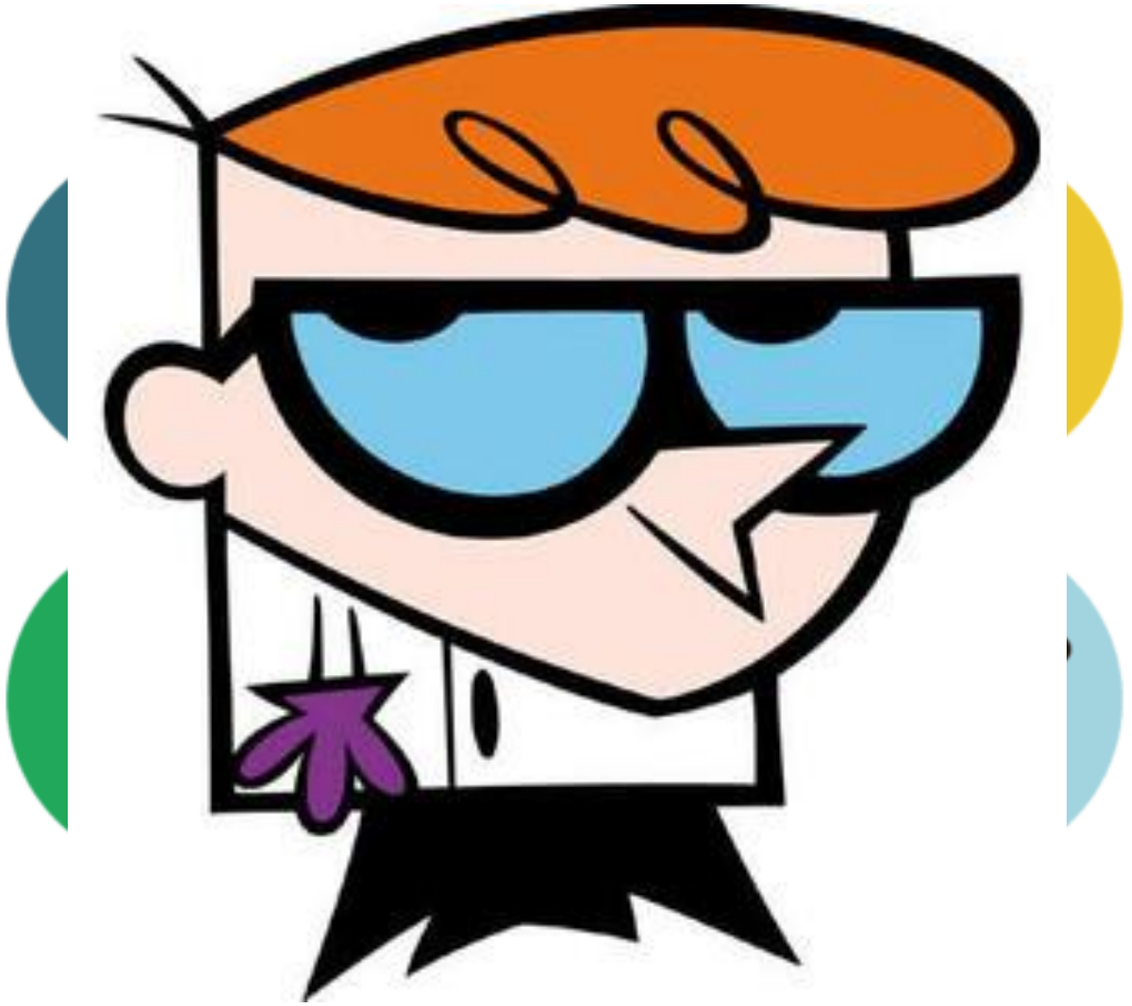


BOOTSRAP AGGREGATION

DICE ANALYTICS

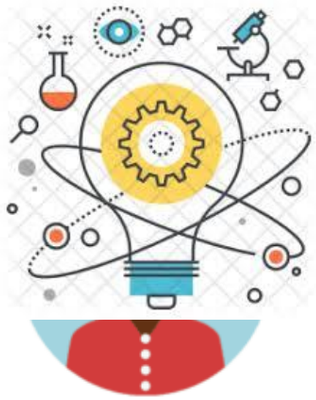
BAGGING





Source: Udacity Machine Learning Nano-Degree

BOOSTING



Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

ENSEMBLE METHODS

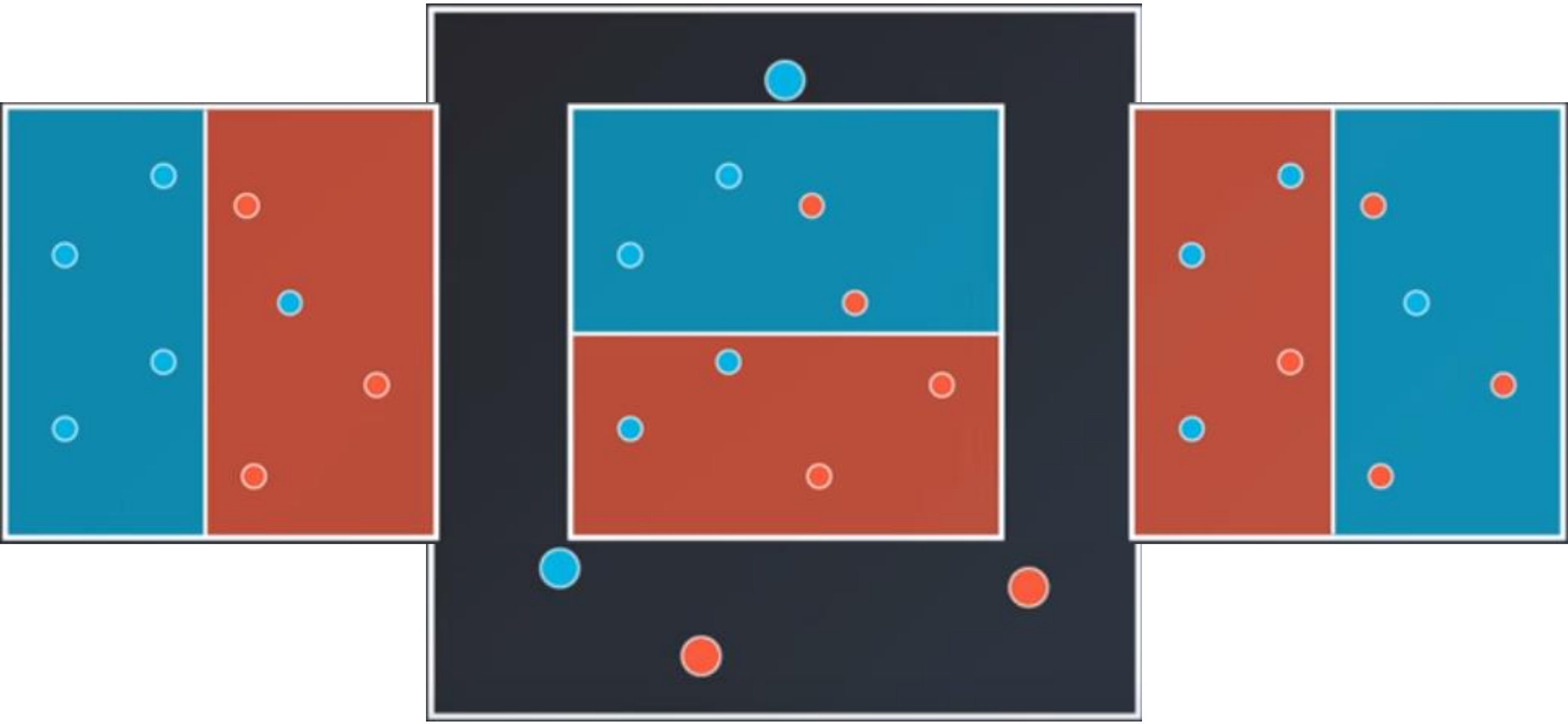
WEAK LEARNERS

STRONG LEARNER

Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

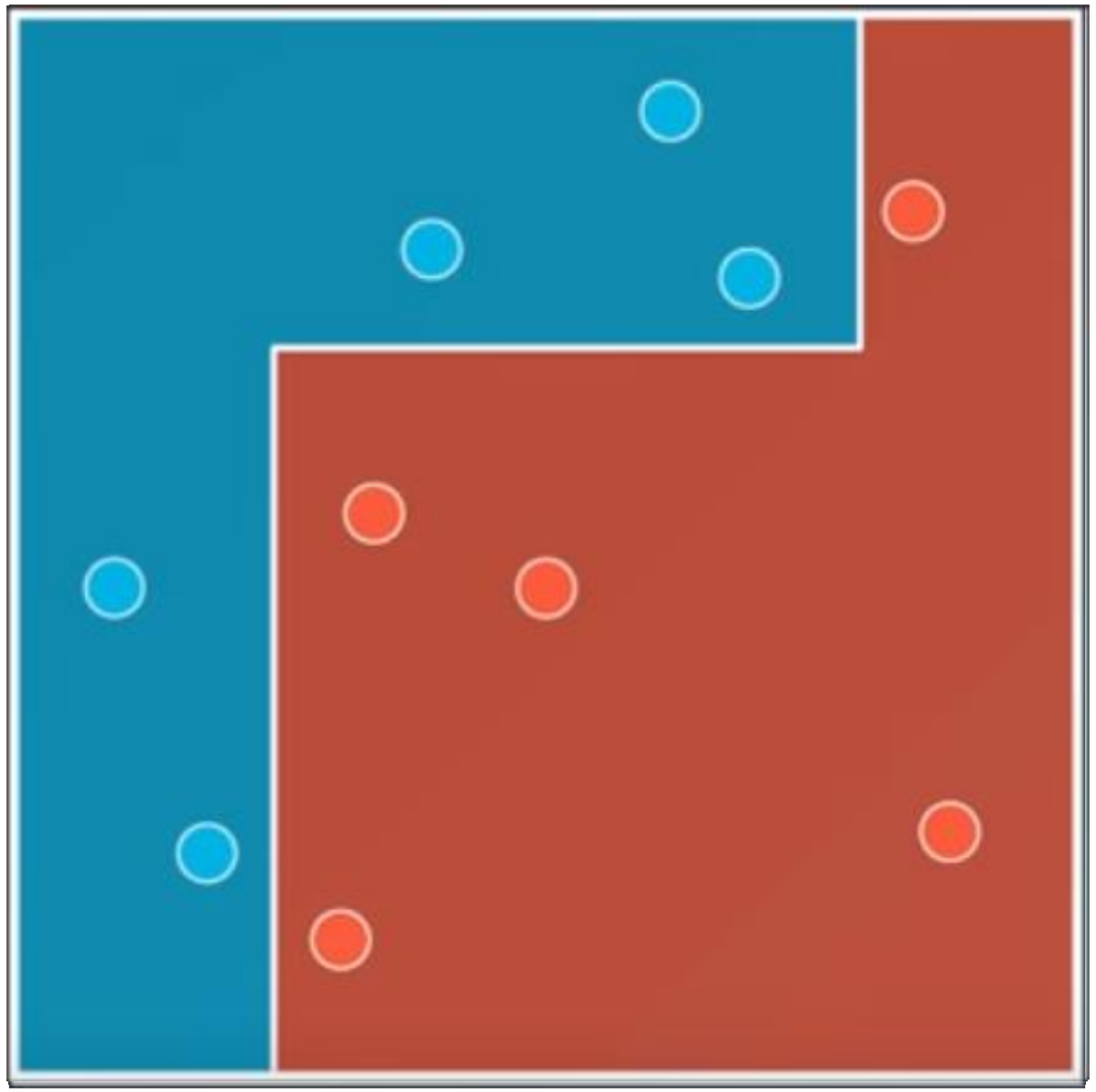
Bagging Explained



Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

Bagging Explained



Source: Udacity Machine Learning Nano-Degree

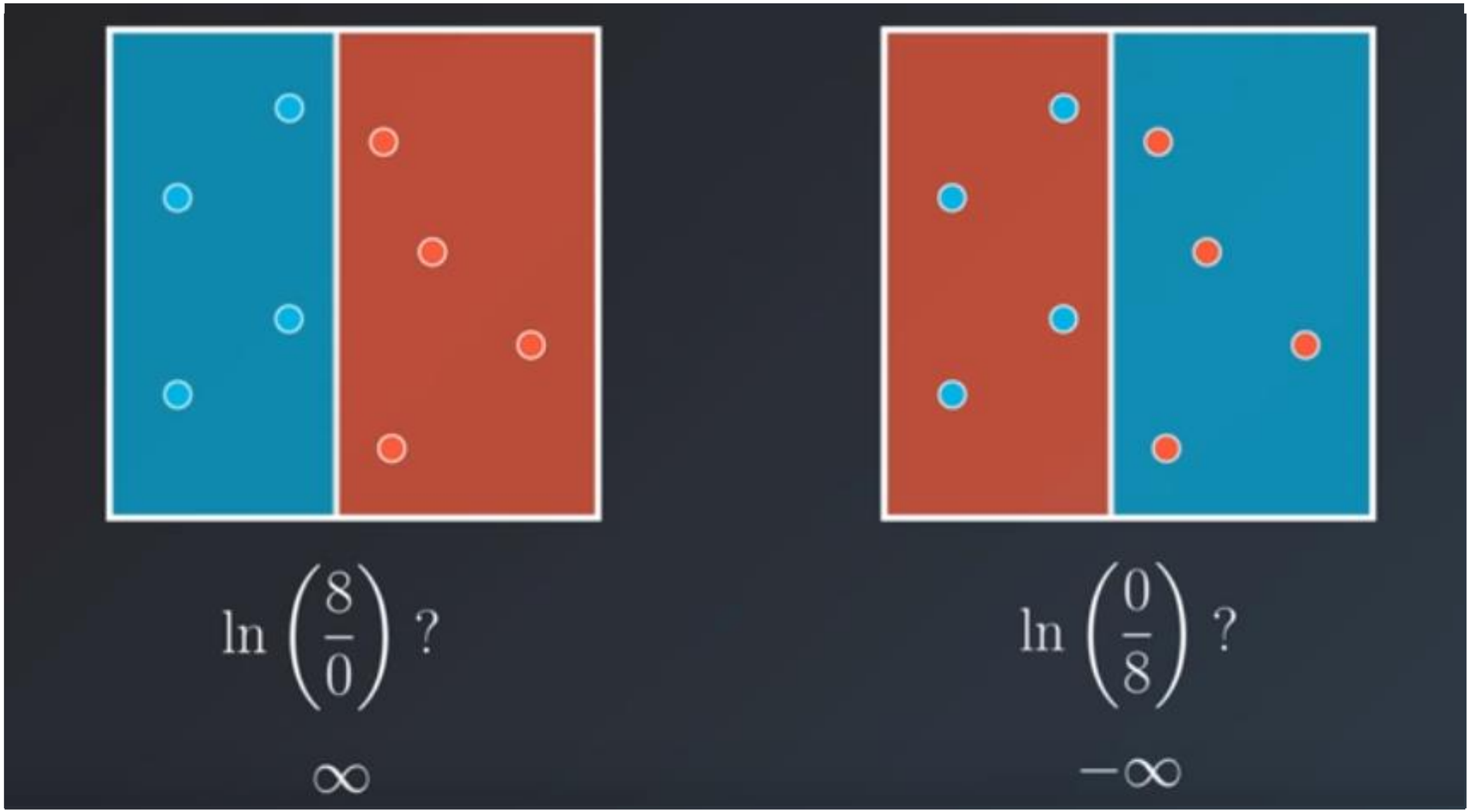
DICE ANALYTICS

Boosting Explained (Adaboost)

Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

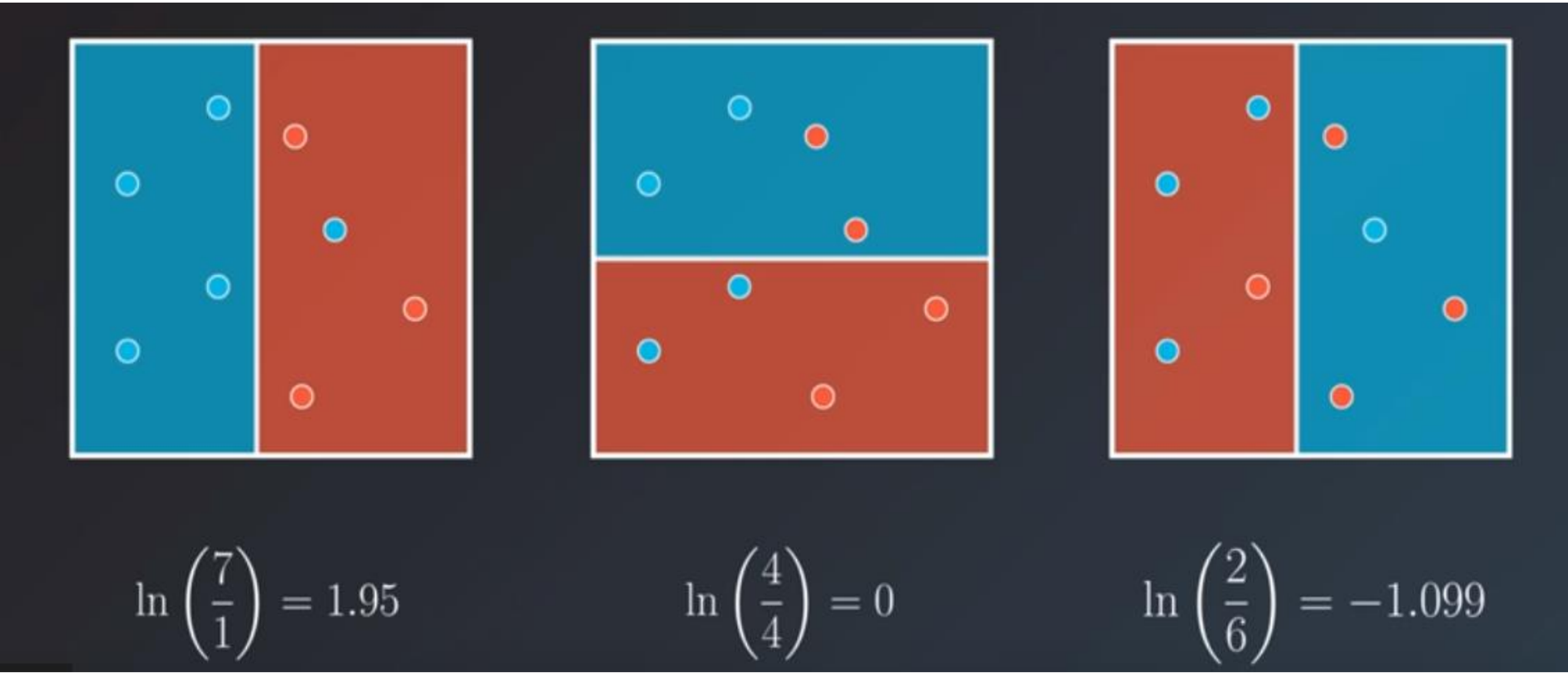
Boosting Explained (Adaboost)



Source: Udacity Machine Learning Nano-Degree

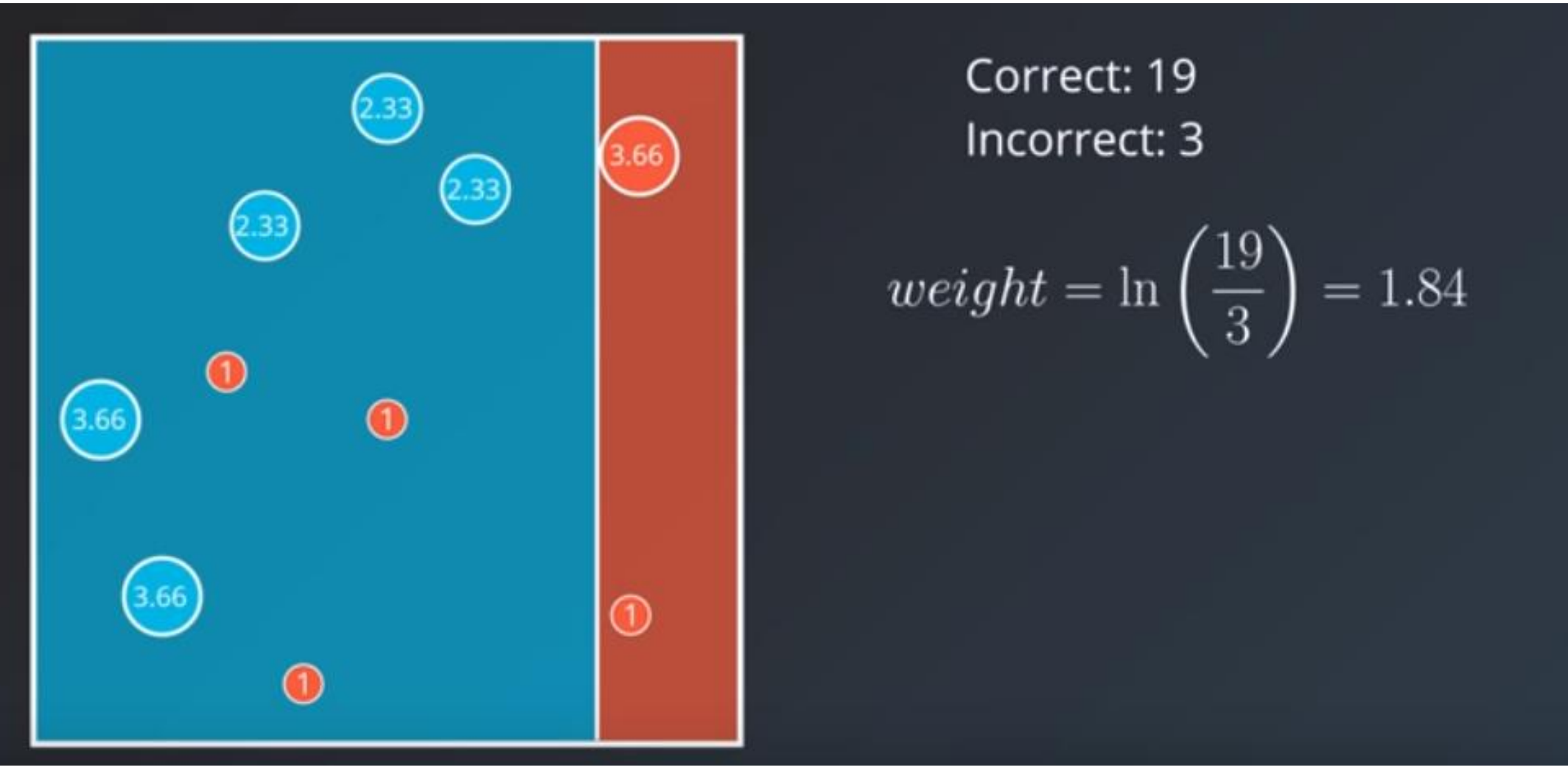
DICE ANALYTICS

Boosting Explained (Adaboost)



Source: Udacity Machine Learning Nano-Degree

Boosting Explained (Adaboost)



Source: Udacity Machine Learning Nano-Degree

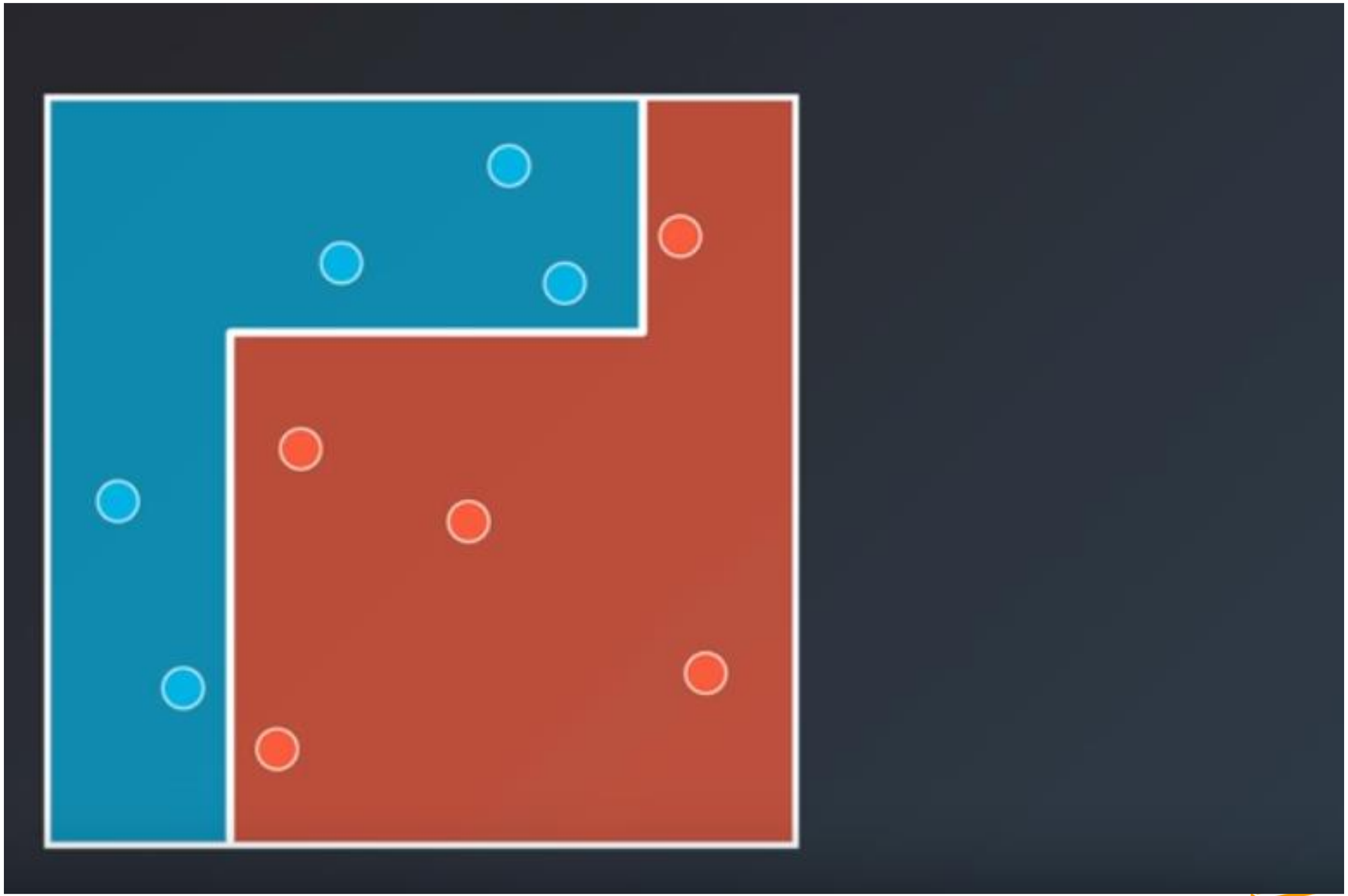
DICE ANALYTICS

Boosting Explained (Adaboost)

Source: Udacity Machine Learning Nano-Degree

DICE ANALYTICS

Boosting Explained (Adaboost)



Source: Udacity Machine Learning Nano-Degree