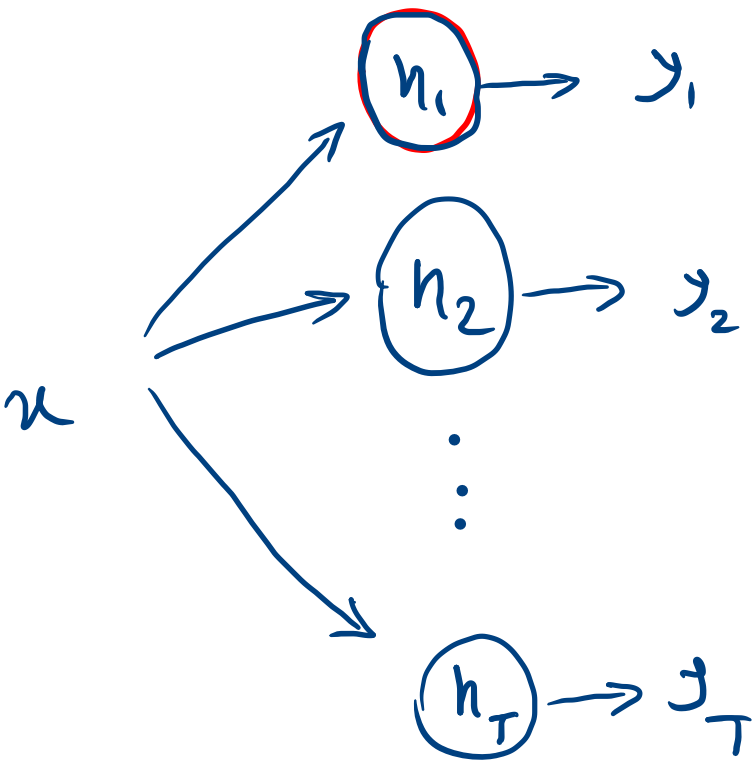
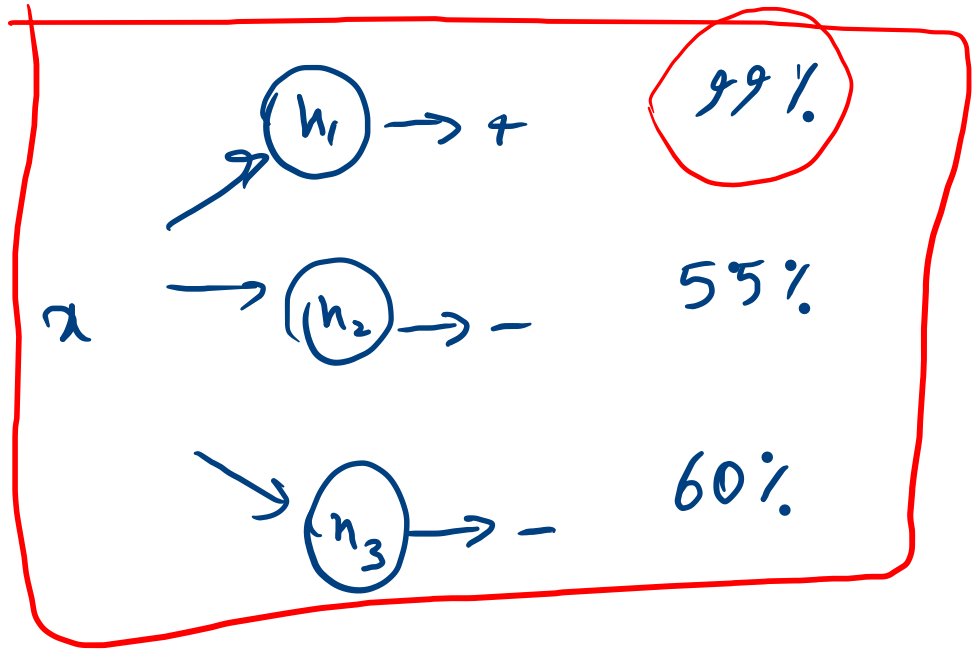


$$= \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$



$h_1(x) \in \{+1, -1\}$

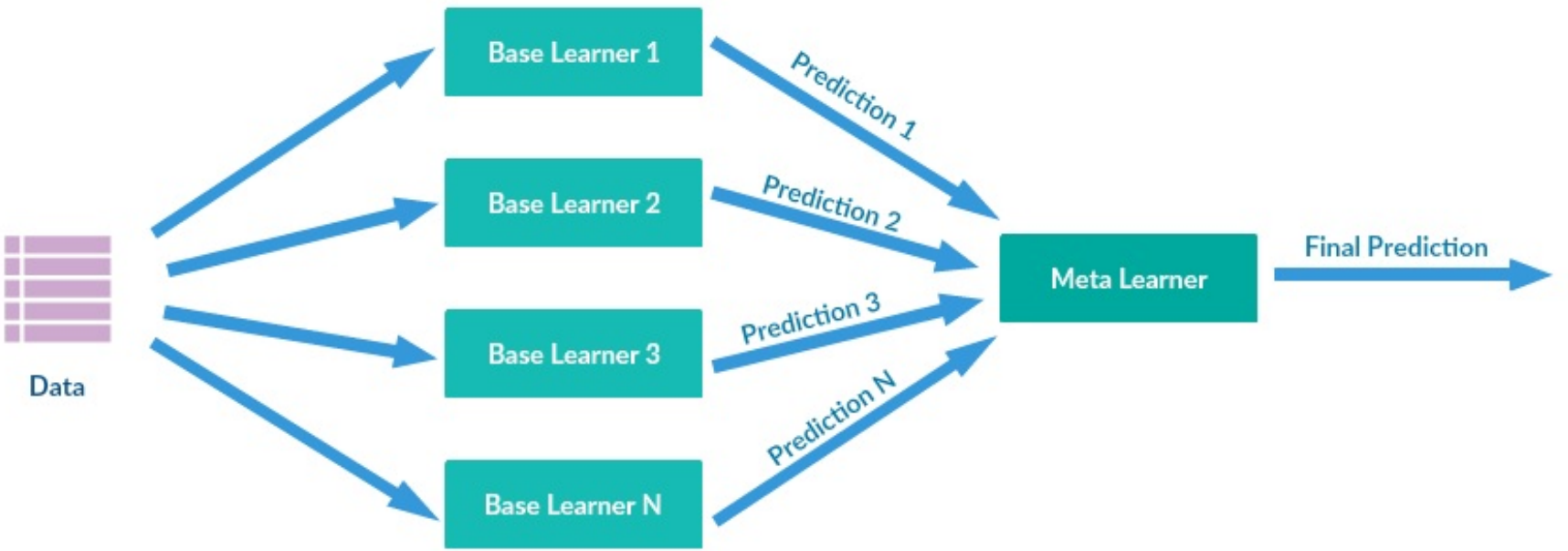
$\vdots$

$h_T(x) \in \{+1, -1\}$

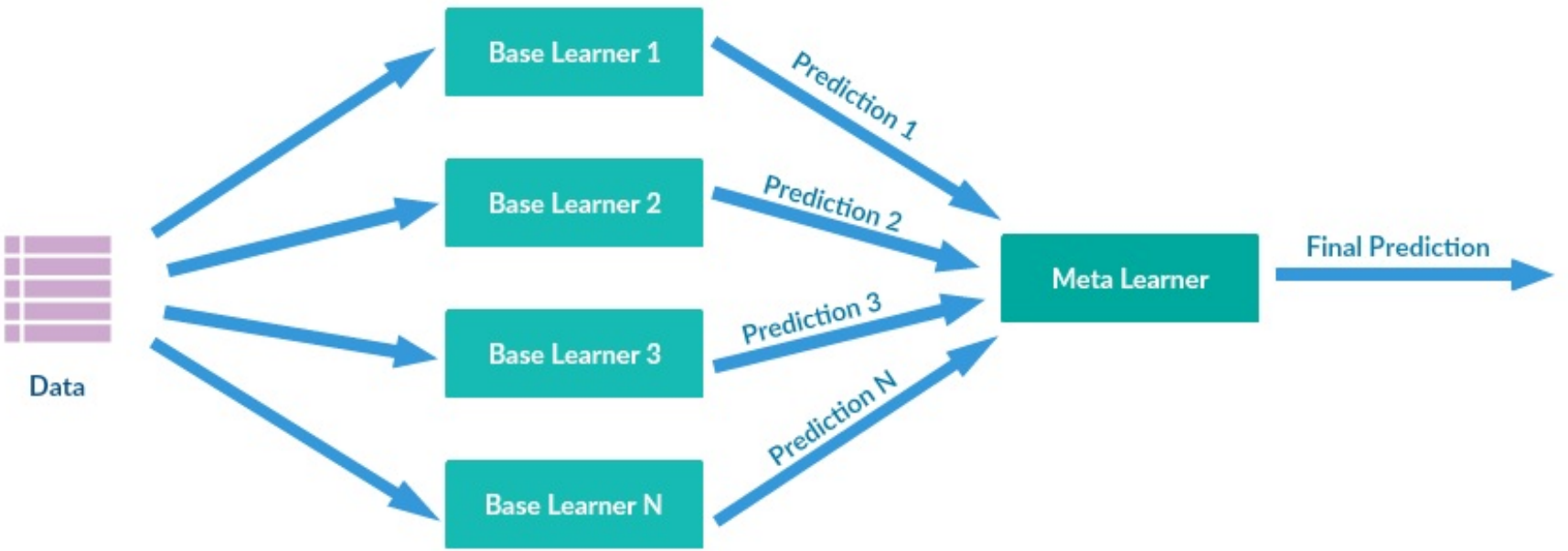
Ensemble Learning

estimator

weak Learner



Ensemble Learning



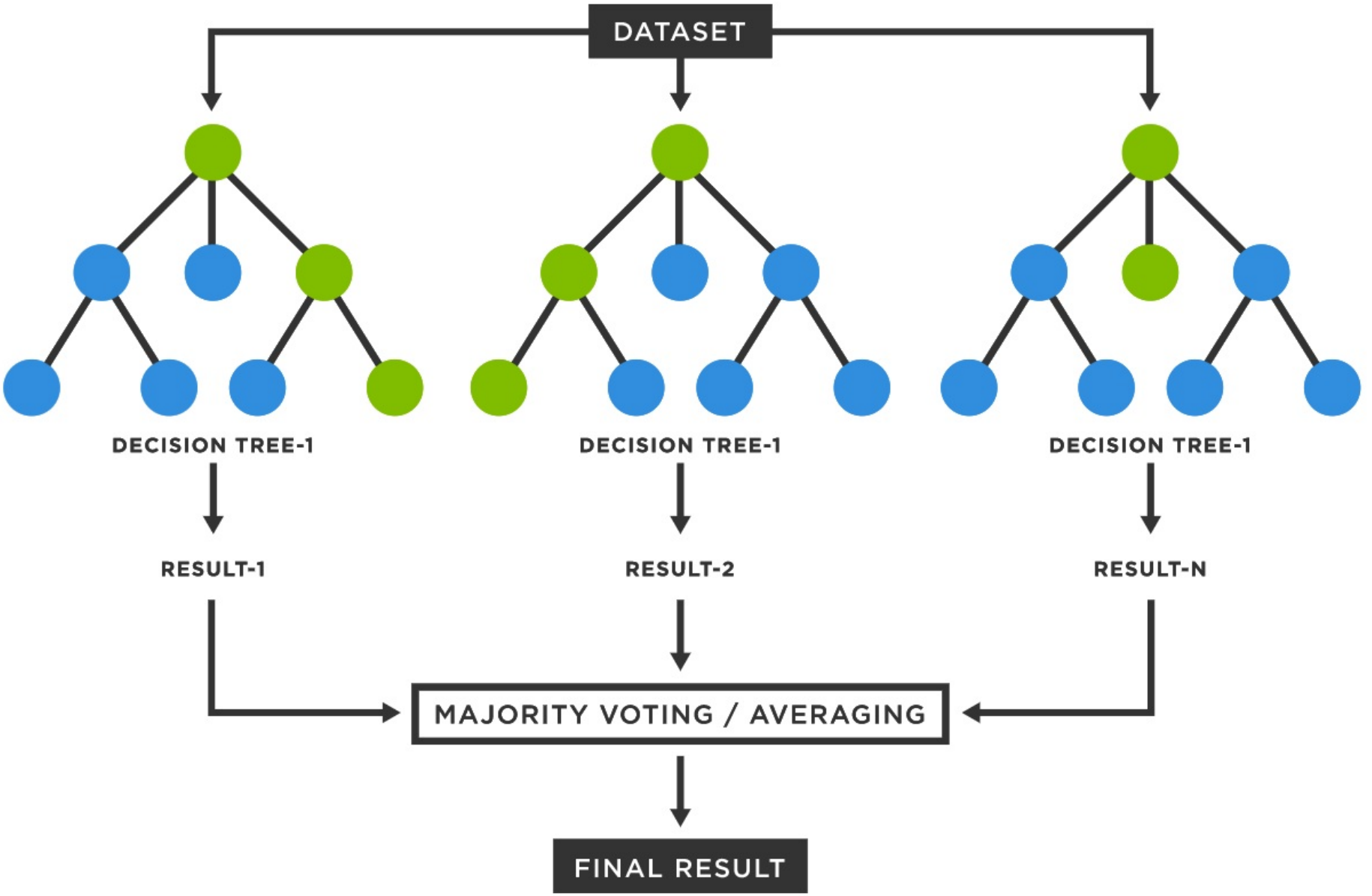
Python: voting classifier

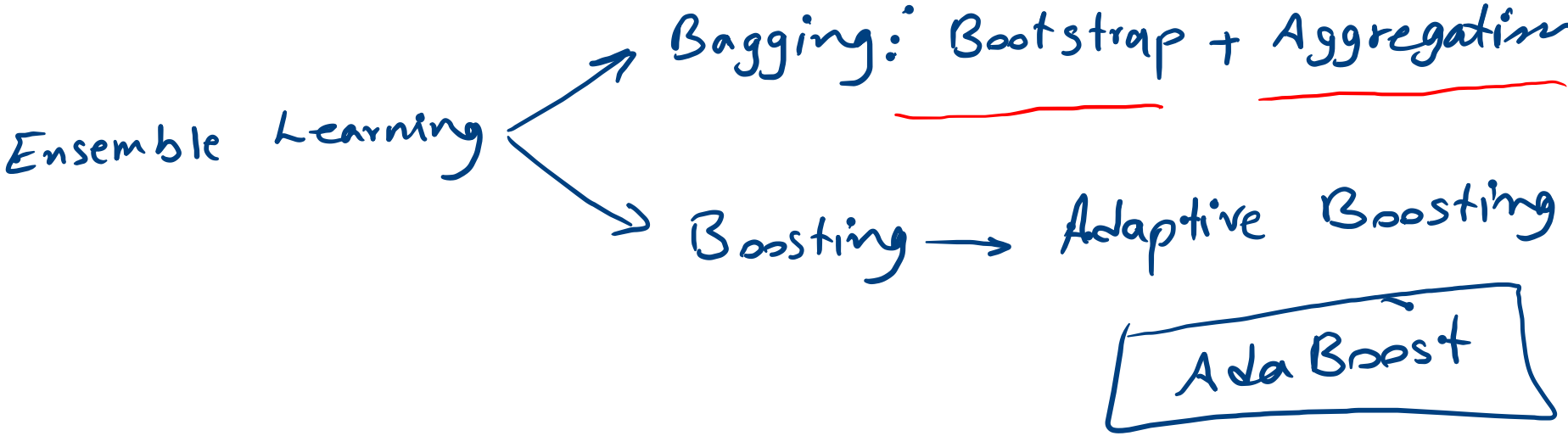
```
# create a voting classifier
from sklearn.ensemble import VotingClassifier
voting_classifier = VotingClassifier(
                                estimators = [('c1', classifier1 ),
                                              ('c2', classifier2 ),
                                              ('c3', classifier3 )],
                                                voting='hard'
)
```

SVM
LR
DT

```
# Train Model
voting_classifier.fit(X_train, y_train)
```

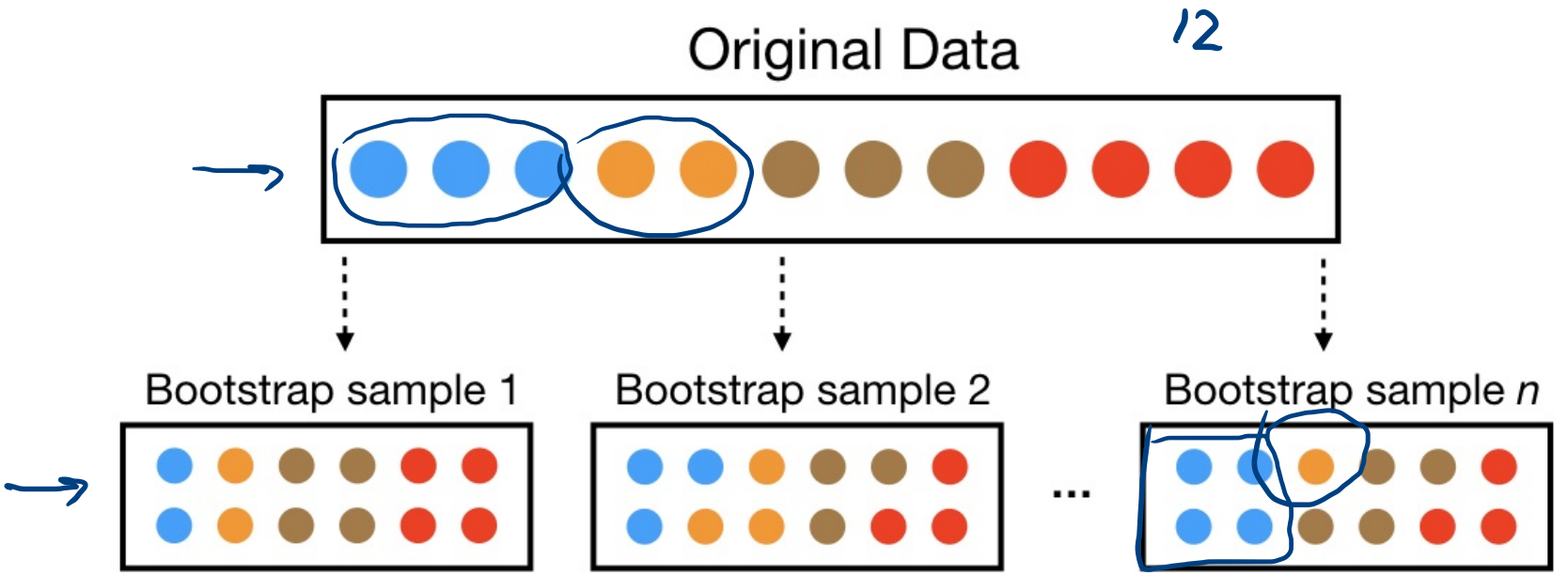
Random Forest



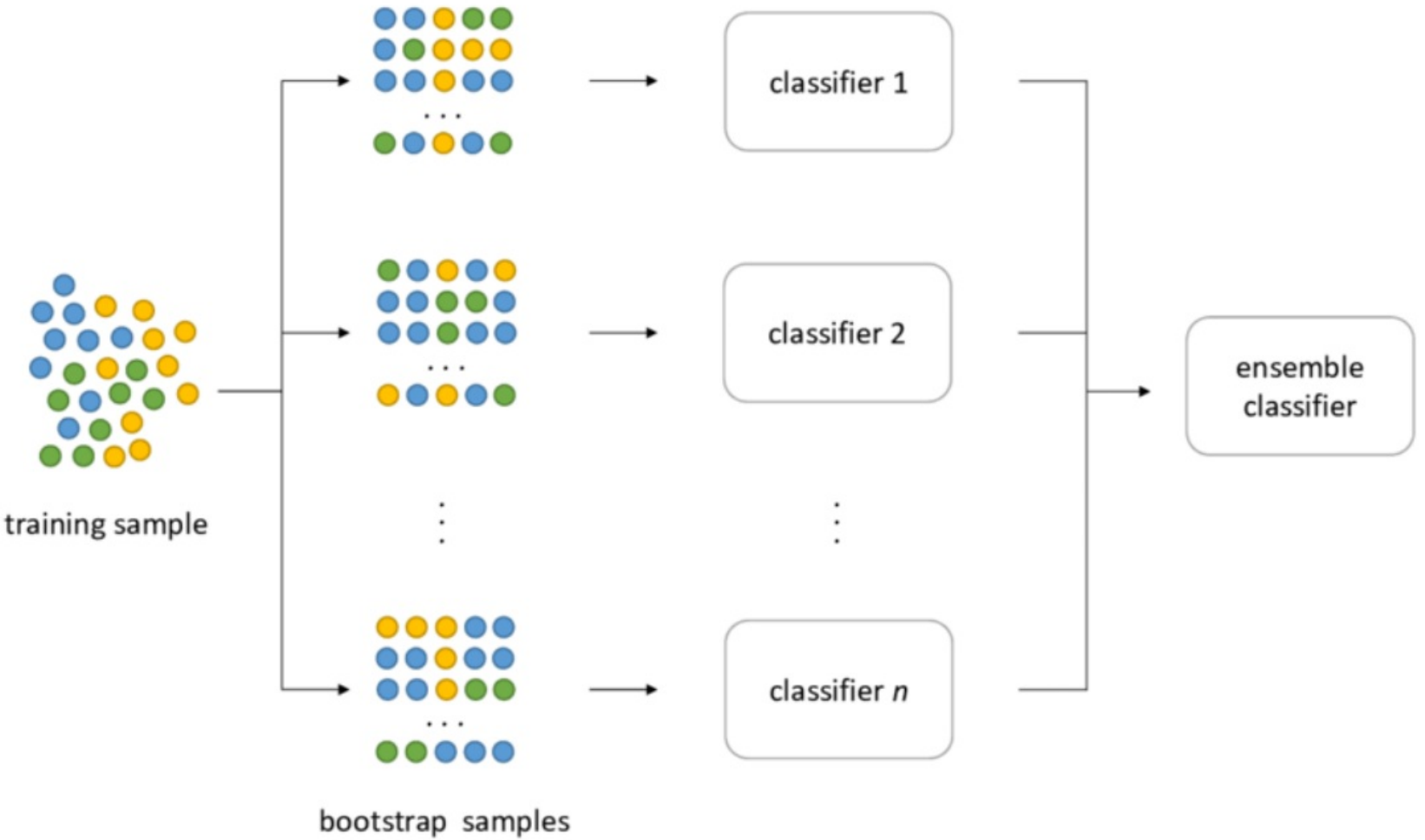

Ensemble Learning
Bagging: Bootstrap + Aggregation
Boosting → Adaptive Boosting
AdaBoost

Bootstrapping

Pasting



Bagging: Bootstrapping + Aggregation

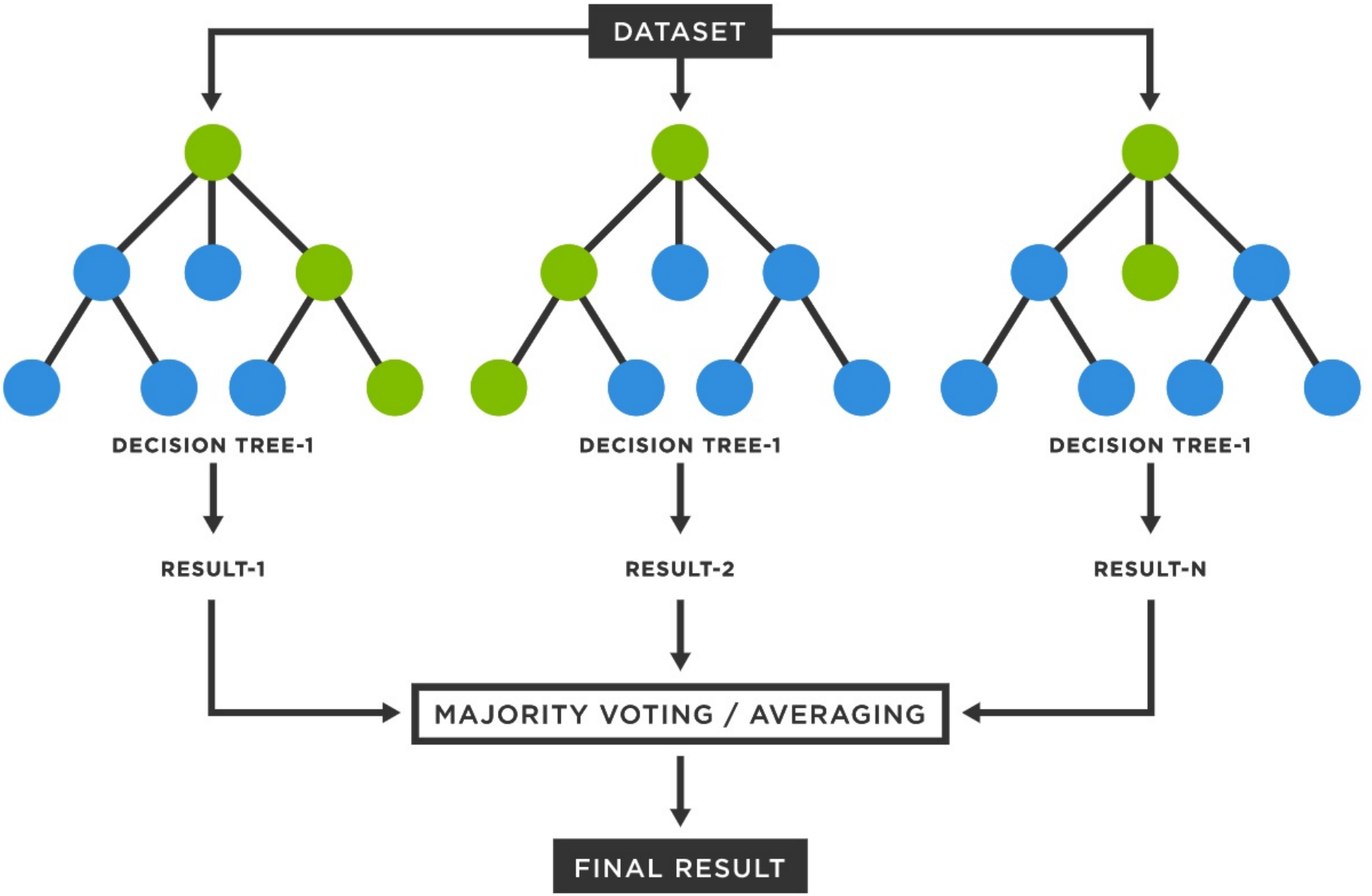


Python: Bagging

```
# create bagging classifier
from sklearn.ensemble import BaggingClassifier
bagging_classifier = BaggingClassifier(LogisticRegression(max_iter=300),
                                       n_estimators = 100, bootstrap=True)
```

```
# Train the Model
bagging_classifier.fit(X_train, y_train)
```

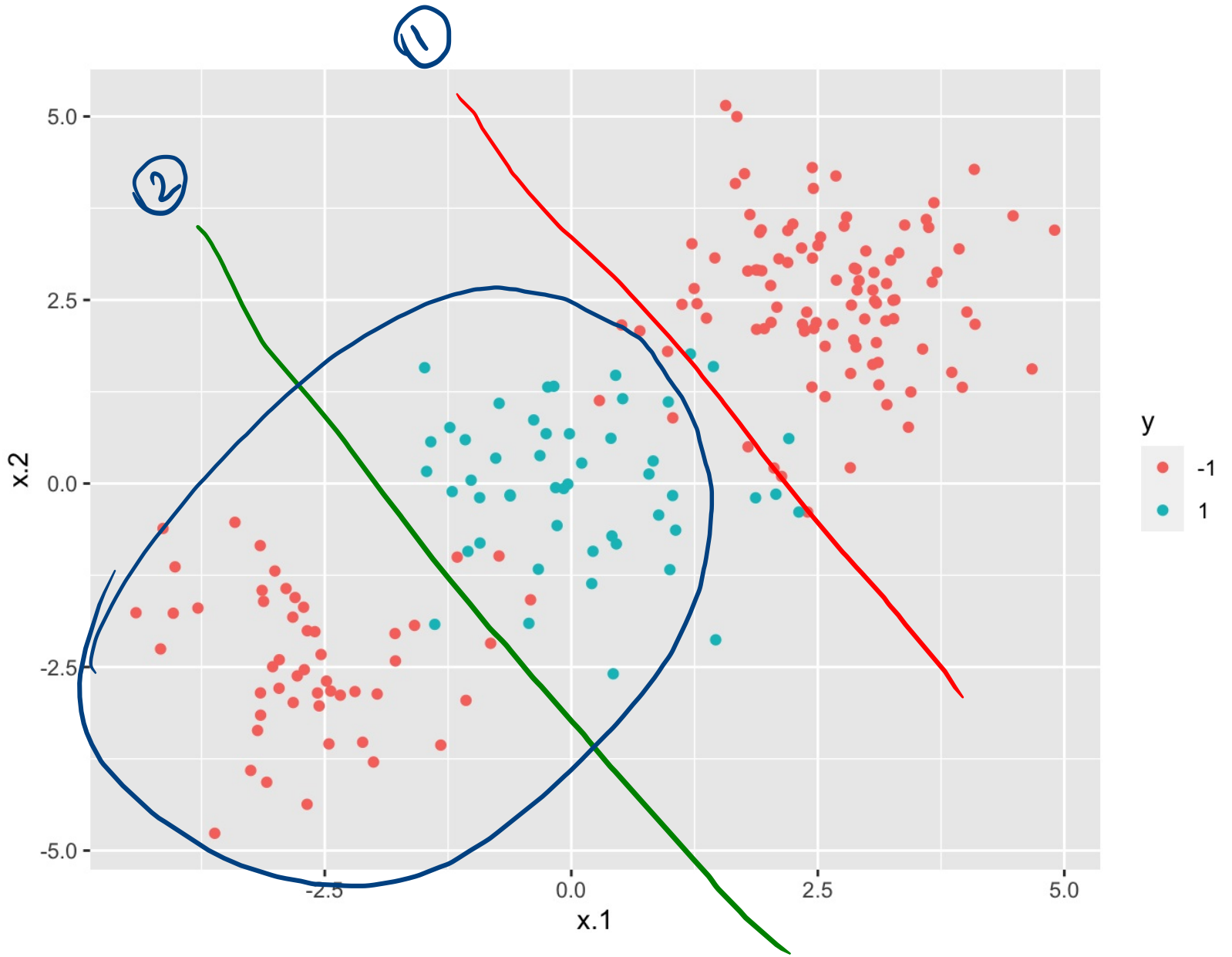
Random Forest

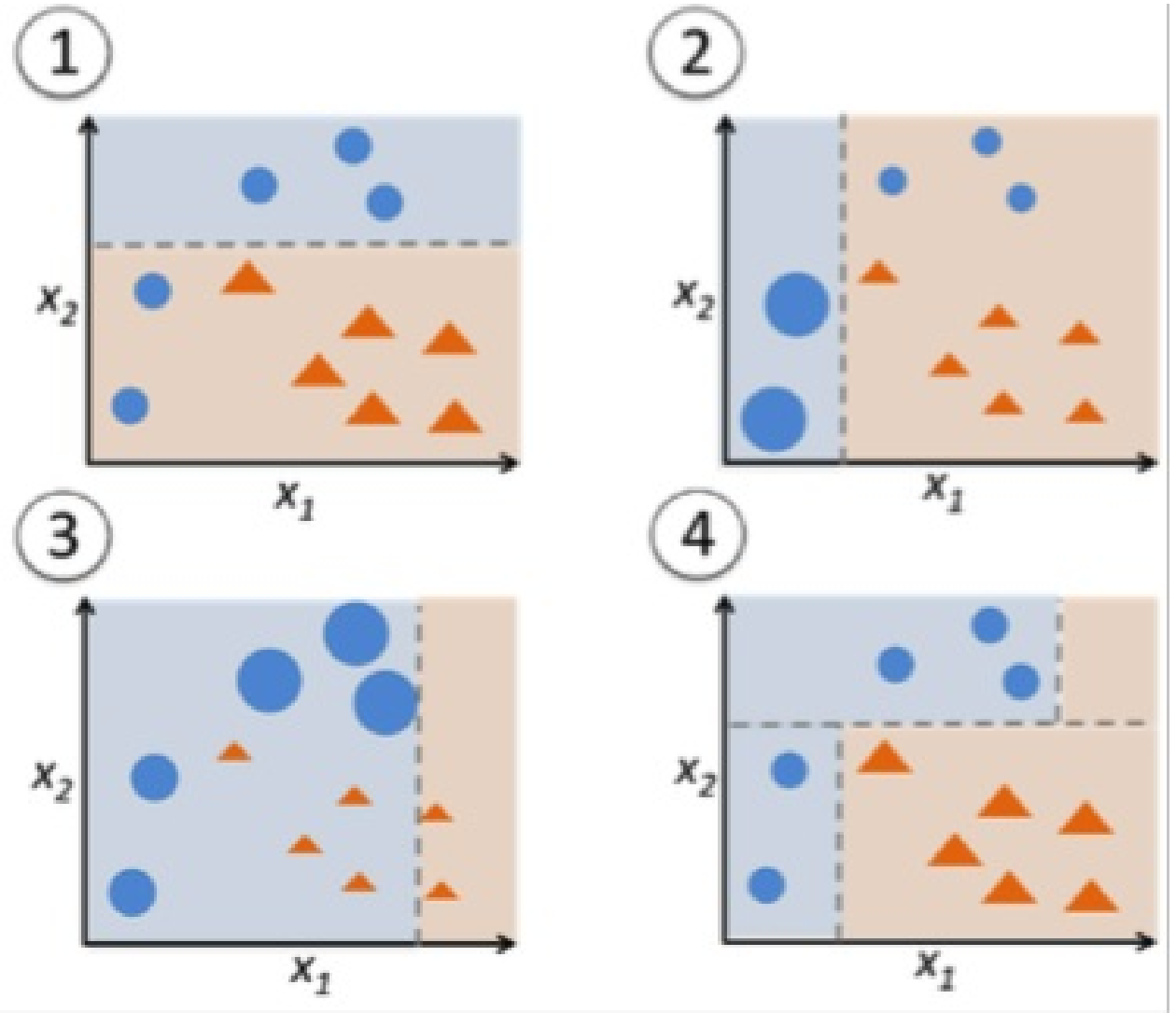


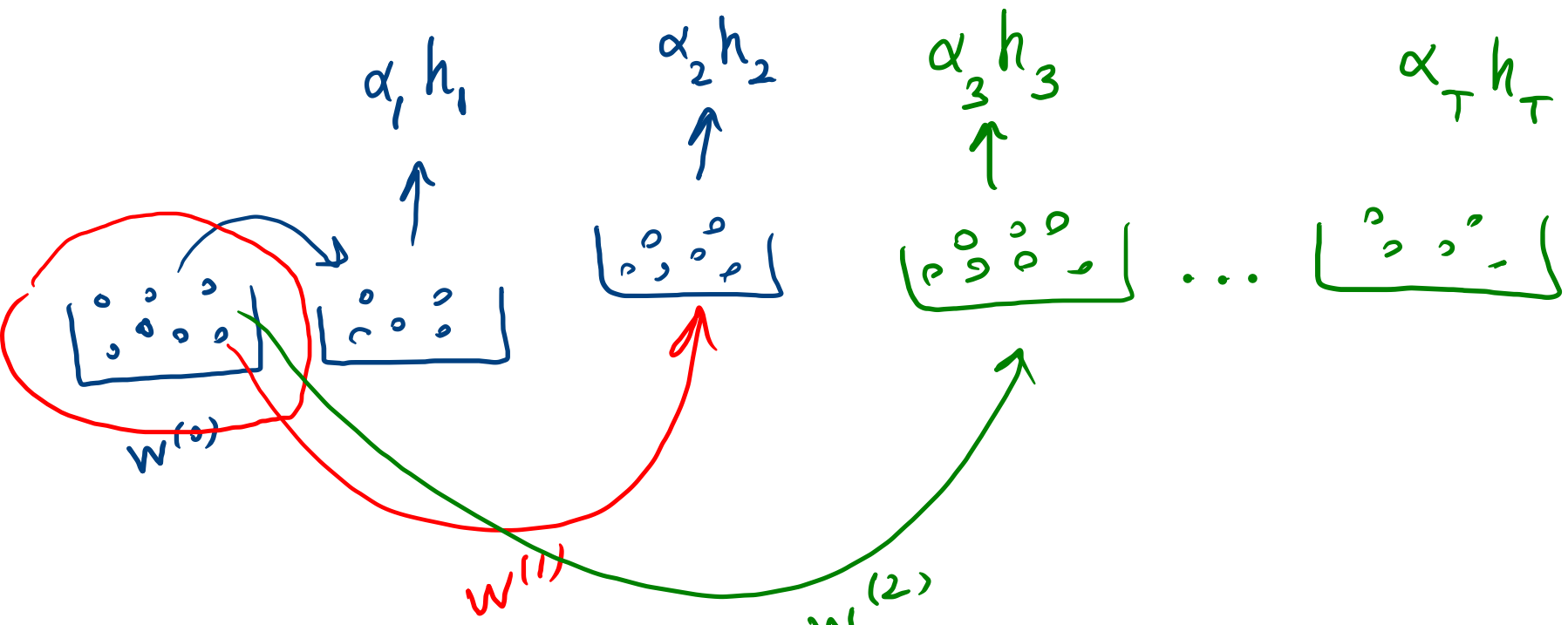
Python: Random Forest

```
# define random forest classifier
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier(n_estimators=20,
                                  max_leaf_nodes=15,
                                  bootstrap=True,
                                  max_samples=100)
```

```
# Train the model
rf_model.fit(X_train, y_train)
```


1
2
5.0
2.5
0.0
-2.5
-5.0
x.2
-2.5
0.0
2.5
5.0
x.1
y
-1
1

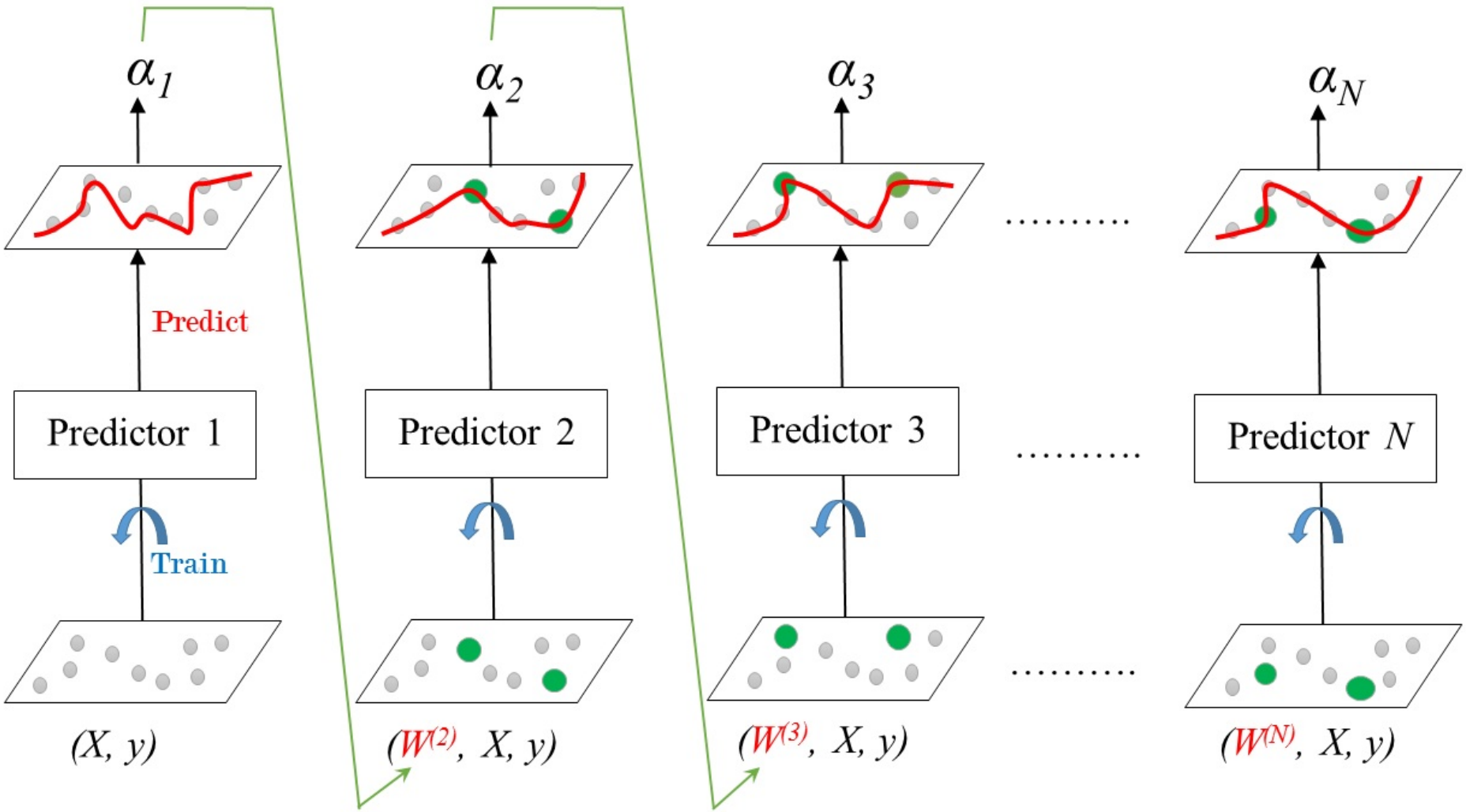

1
x_2
x_1
2
x_2
x_1
3
x_2
x_1
4
x_2
x_1



$\alpha_1 h_1 + \alpha_2 h_2$

$sign\left(\alpha_1 h_1(x) + \alpha_2 h_2(x) \right)$

AdaBoost


α_1
α_2
α_3
α_N
Predict
Predictor 1
Predictor 2
Predictor 3
Predictor N
Train
(X, y)
($W^{(2)}$, X, y)
($W^{(3)}$, X, y)
($W^{(N)}$, X, y)

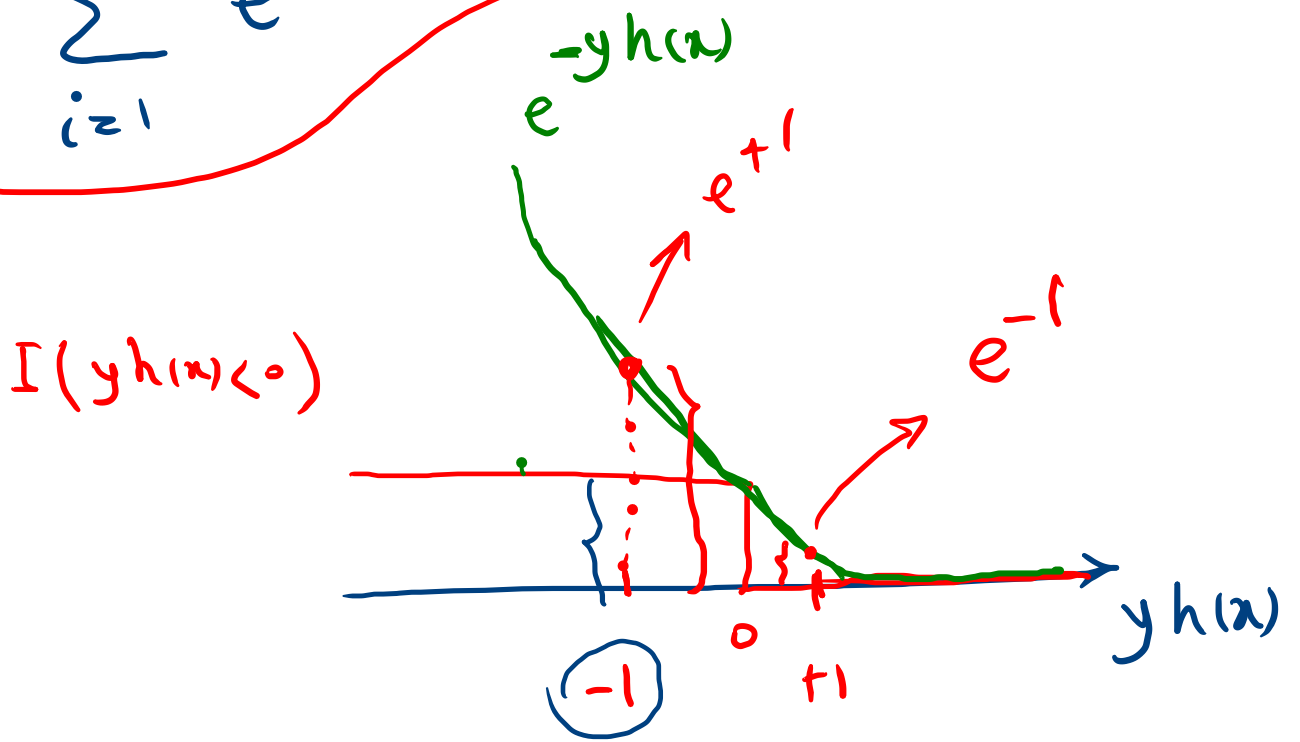
Python: AdaBoost

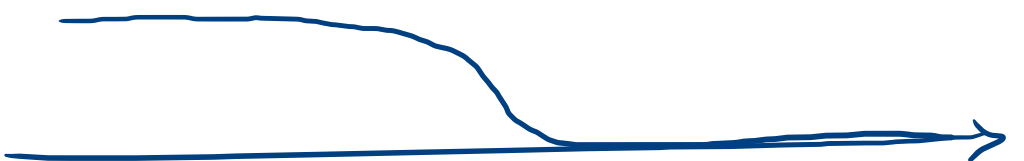
```
# define adaboost model
from sklearn.ensemble import AdaBoostClassifier
adaboost_model = AdaBoostClassifier
                        (base_estimator=LogisticRegression(max_iter=300),
                         n_estimators=100)
```

```
# Train Model
adaboost_model.fit(X_train, y_train)
```

$$E = \sum_{i=1}^{n} I(y_i h(x_i) < 0)$$

$$E = \sum_{i=1}^{n} e^{-y_i h(x_i)}$$





$$F_T(x) = \sum_{t=1}^{T} \alpha_t h_t(x) = \underbrace{\sum_{t=1}^{T-1} \alpha_t h_t(x)}_{F_{T-1}(x)} + \alpha_T h_T(x)$$

$$E = \sum_{i=1}^{n} e^{-y_i F_T(x_i)} = \sum_{i=1}^{n} e^{-y_i \left(F_{T-1}(x_i) + \alpha_T h_T(x_i)\right)}$$

$$= \sum_{i=1}^{n} \underbrace{e^{-y_i F_{T-1}(x_i)}}_{w_i^{(T-1)}} e^{-y_i \alpha_T h_T(x_i)} = \sum_{i=1}^{n} w_i^{(T-1)} e^{-y_i \alpha_T h_T(x_i)}$$

$$= \underbrace{\sum_{i:\, y_i = h_T(x_i)}}_{C} w_i^{(T-1)} e^{-\alpha_T} + \underbrace{\sum_{i:\, y_i \neq h_T(x_i)}}_{M} w_i^{(T-1)} e^{\alpha_T}$$

$$\frac{dE}{d\alpha_T} = -\sum_{i \in C} w_i^{(T-1)} e^{-\alpha_T} + \sum_{i \in M} w_i^{(T-1)} e^{\alpha_T} = 0$$

$$\alpha_T = \frac{1}{2} \ln \frac{\sum_{i \in C} w_i^{(T-1)}}{\sum_{i \in M} w_i^{(T-1)}}$$

$$\epsilon_T = \frac{\sum_{i \in M} w_i^{(T-1)}}{\sum_{i=1}^{n} w_i^{(T-1)}}$$

h_T خطای

$$\alpha_T = \frac{1}{2} \ln \frac{1 - \epsilon_T}{\epsilon_T}$$

h_T

$\epsilon_T = 0 \longrightarrow \alpha_T = +\infty$

$\epsilon_T = 1 \longrightarrow \alpha_T = -\infty$

$\epsilon_T = 1/2 \longrightarrow \alpha_T = 0$