Module: Adv. Machine Learning

Ensemble Learning

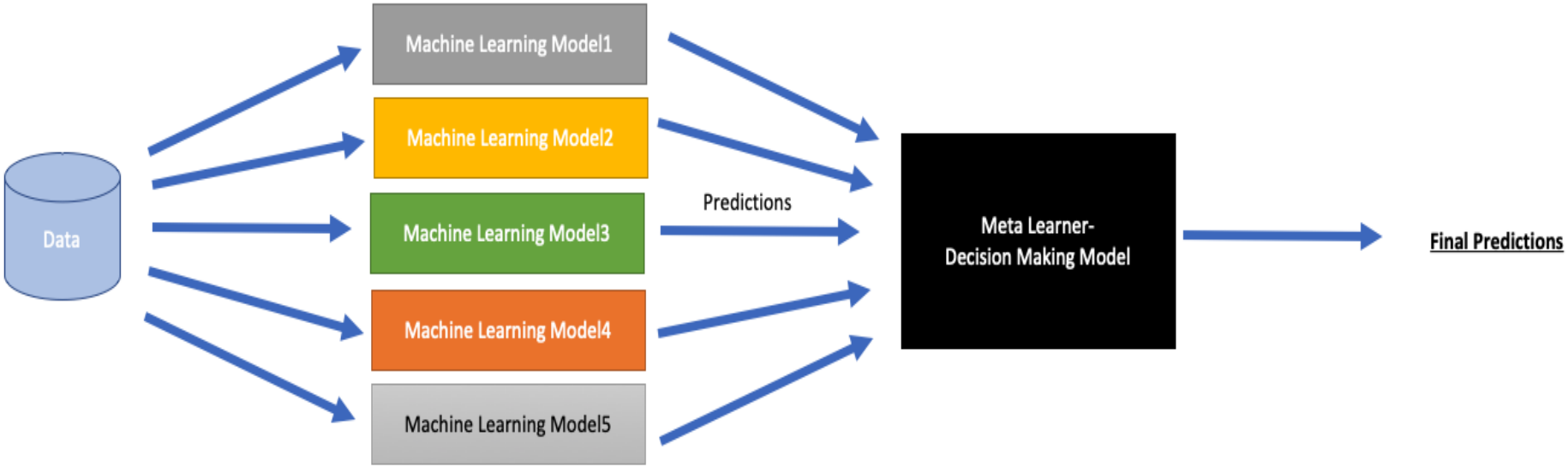
Agenda:
Bagging-based Ensemble
Boosting Classifier (Ada-boost)
Stacking

Why Ensemble Learning:

Simple Models	Under-fitting	High Bias/Low Variance
Complex Models	Overfitting	Low Bias/High Variance

Ensemble Learning:

Ensemble learning is an approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.



Ensemble Learning:

The main ensemble strategies include:

- Bagging
- Boosting
- Stacking

Why Bagging?

Ensemble Learning: Bagging

Bagging, which stands for *bootstrap aggregating*, is a simple ensemble based algorithms, with a surprisingly good performance.

Diversity of classifiers in bagging is obtained by using bootstrapped replicas of the training data.

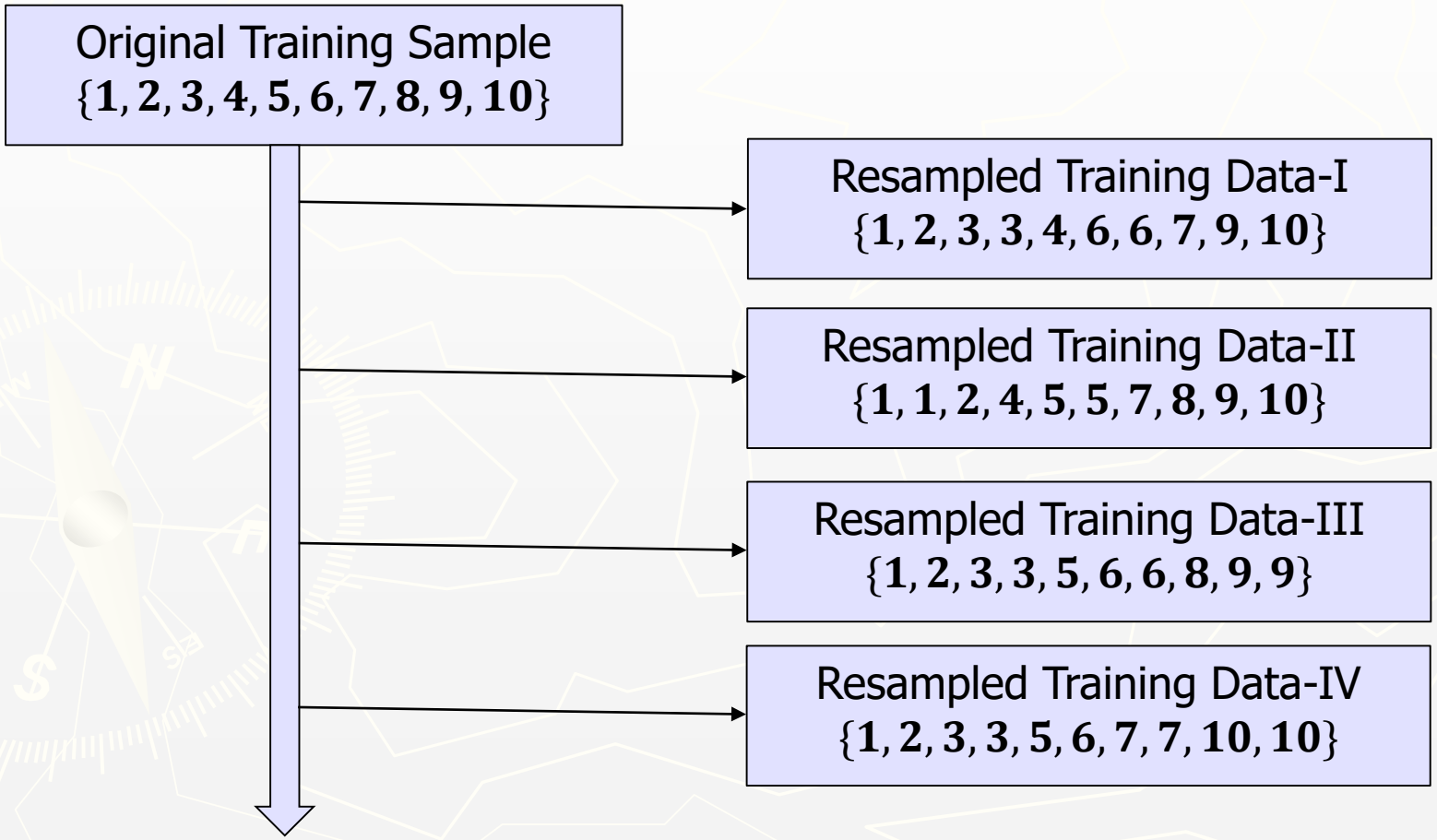
That is, different training data subsets are randomly drawn – with replacement – from the entire training dataset.

Each training data subset is used to train a different classifier of the same type. Individual classifiers are then combined by taking a simple majority vote of their decisions.

For any given instance, the class chosen by most number of classifiers is the ensemble decision or taking the average in case of prediction.

Ensemble Learning: Bagging

Bagging resamples the original training dataset with replacement, some instance(or data) may be present multiple times while others are left out.



Bagging Classifier: Architecture

Bagging Classifier: Hyper-parameters

BaggingClassifier(base_estimator=None, n_estimators=10, max_samples=1.0, bootstrap=True, bootstrap_features=False, n_jobs=None, random_state=None)

base_estimator	The base estimator to fit on random subsets of the dataset. If None, then the base estimator is a DecisionTreeClassifier.
n_estimators	The number of base estimators in the ensemble.
max_samples	The number of samples to draw from X to train each base estimator
bootstrap	Whether samples are drawn with replacement. If False, sampling without replacement is performed.
bootstrap_features	Whether features are drawn with replacement.
n_jobs	The number of jobs to run in parallel for both fit and predict.
random_state	Controls the random resampling of the original dataset (sample wise and feature wise).

Bagging Classifier: An Example

Random Forest: Introduction

Random forest is a supervised learning algorithm.

The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method.

The general idea of the bagging method is that a combination of learning models increases the overall result.

One big advantage of random forest is that it can be used for both classification and regression problems

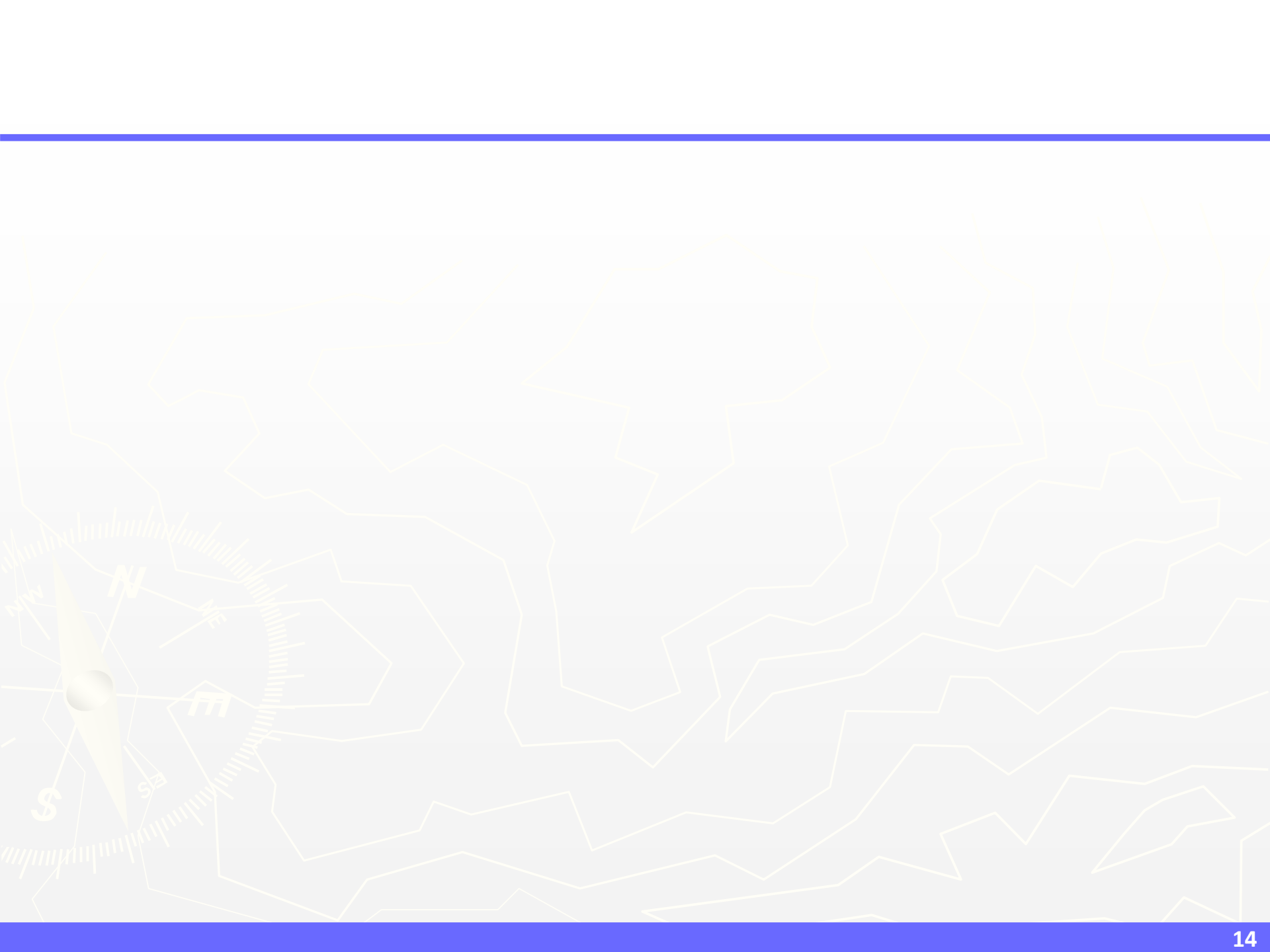
Random Forest v/s Bagging Classifier

Random forest adds additional randomness to the model, while growing the trees.

Instead of searching for the most important feature while splitting a node, it searches for the best feature among a random subset of features.

This results in a wide diversity that generally results in a better model.

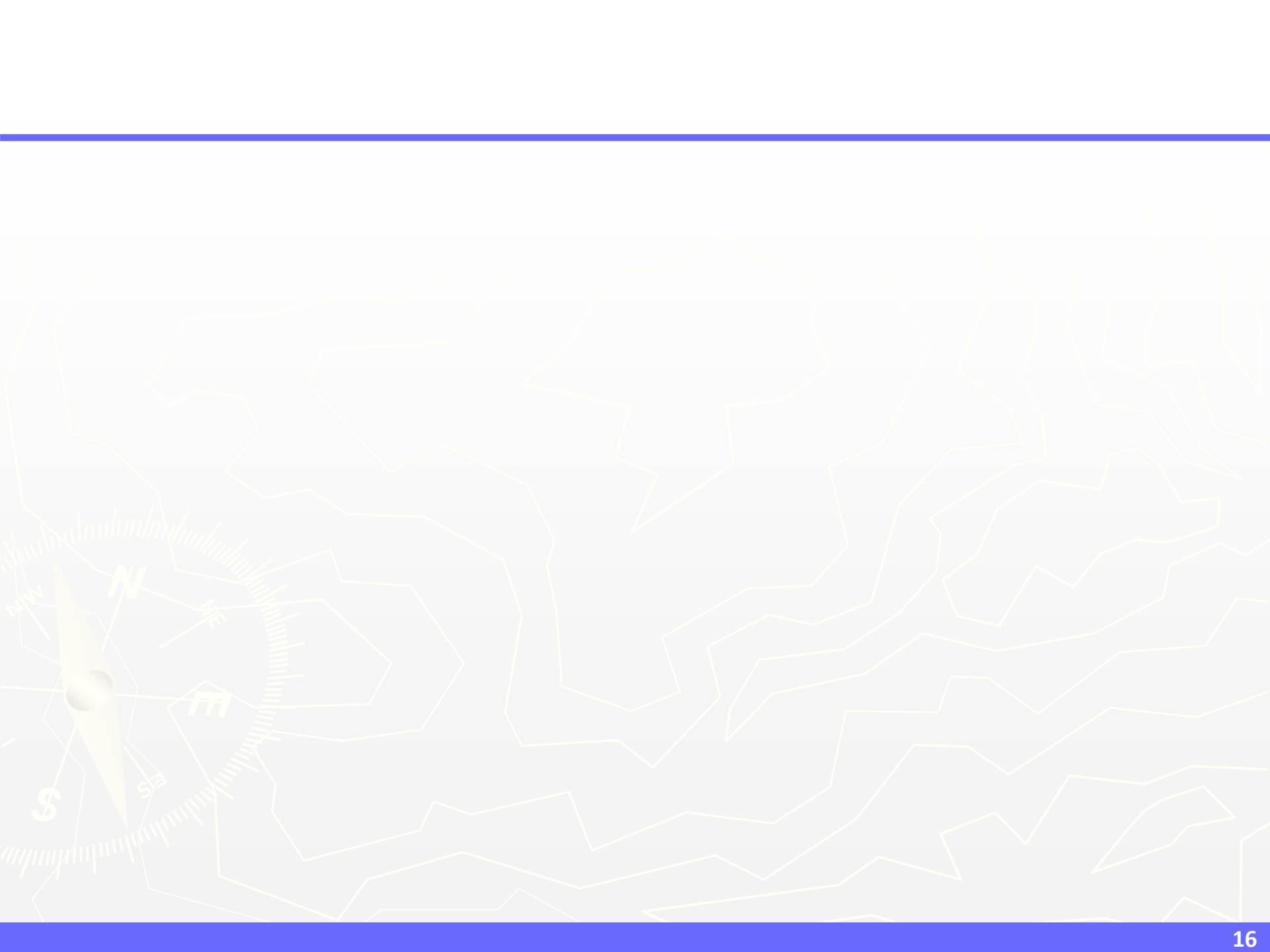
Random Forest: Features selection



Random Forest: Hyper-parameters

class sklearn.ensemble.RandomForestClassifier(n_estimators=100, criterion='gini', max_depth=None, min_samples_split=2, min_samples_leaf=1, oob_score=False, n_jobs=None)

n_estimators:	number of trees the algorithm builds before taking the maximum voting or taking the averages of predictions. In general, a higher number of trees increases the performance and makes the predictions more stable, but it also slows down the computation.
criterion	gini or entropy
max_depth	Maximum depth of each tree
min_samples_ split	
min_samples_leaf	the minimum number of leafs required to split an internal node
max_feature	the maximum number of features random forest considers to split a node
oob_score	is a random forest cross-validation method. In this sampling, about one-third of the data is not used to train the model and can be used to evaluate its performance. These samples are called the out-of-bag samples.
n_jobs	tells the engine how many processors it is allowed to use. If it has a value of one, it can only use one processor. A value of "-1" means that there is no limit.



Random Forest

Boosting:

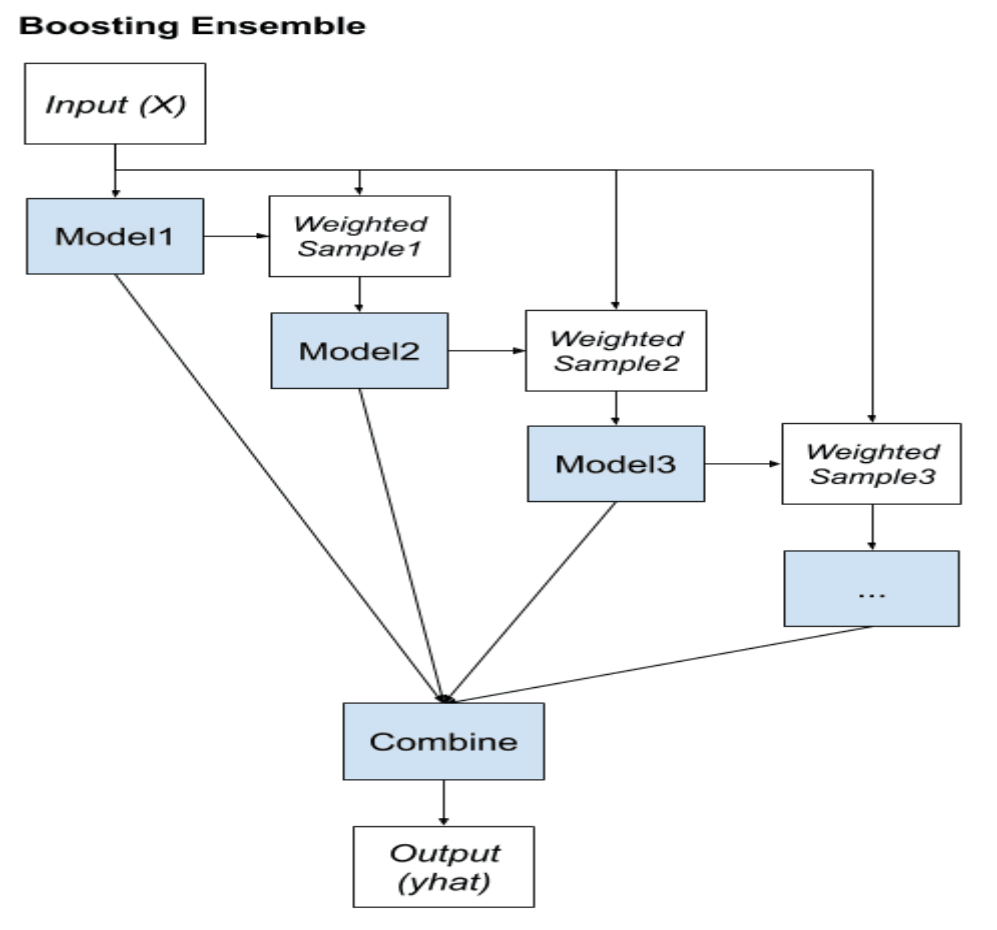
Boosting is an ensemble ML technique which attempts to build a strong estimator from the number of weak estimator.

The weak models are connected in a series.

Firstly, a model is built from the training data. Then the second model is built which tries to correct the errors present in the first model. This procedure is continued and models are added until either the complete training data set is predicted correctly or the maximum number of models are added.

Ensemble Learning: boosting

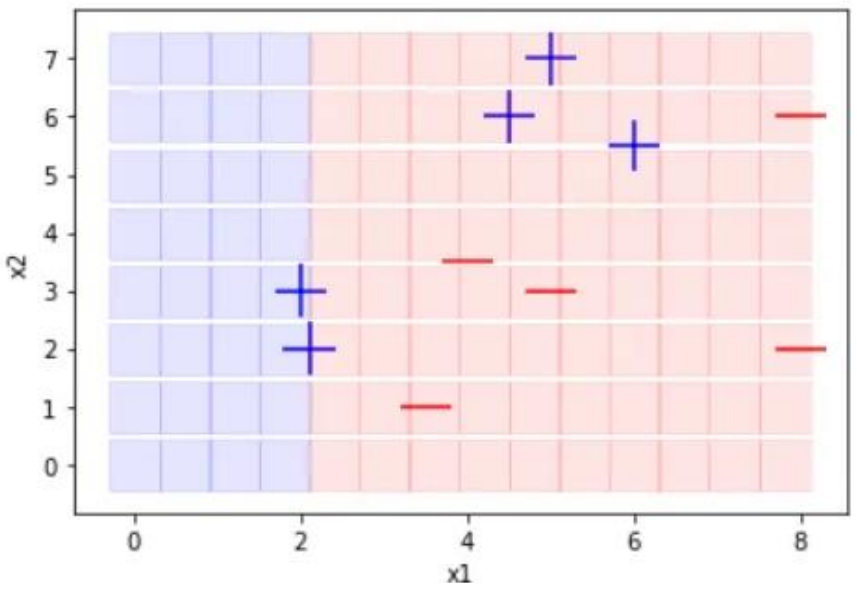
- Bias training data toward those examples that are hard to predict.
- Iteratively add ensemble members to correct predictions of prior models.
- Combine predictions using a weighted average of models.

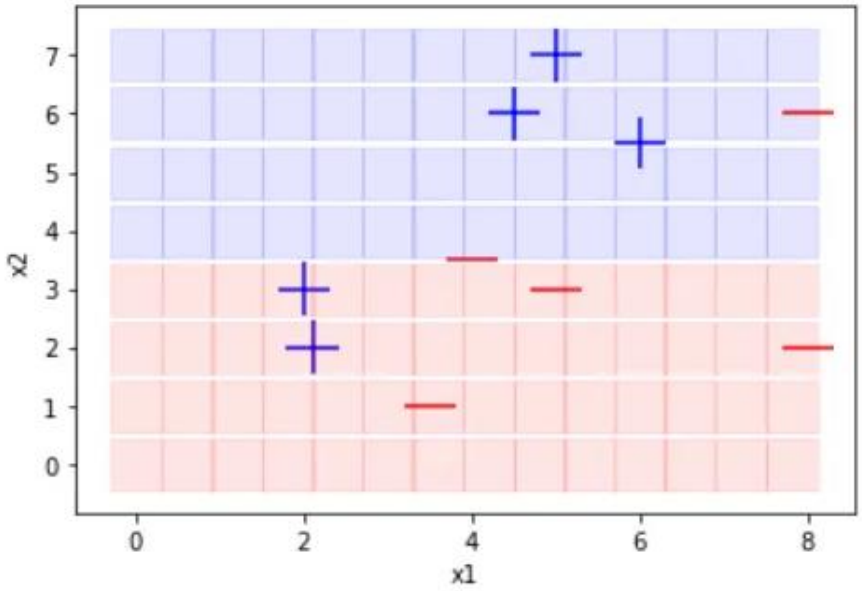


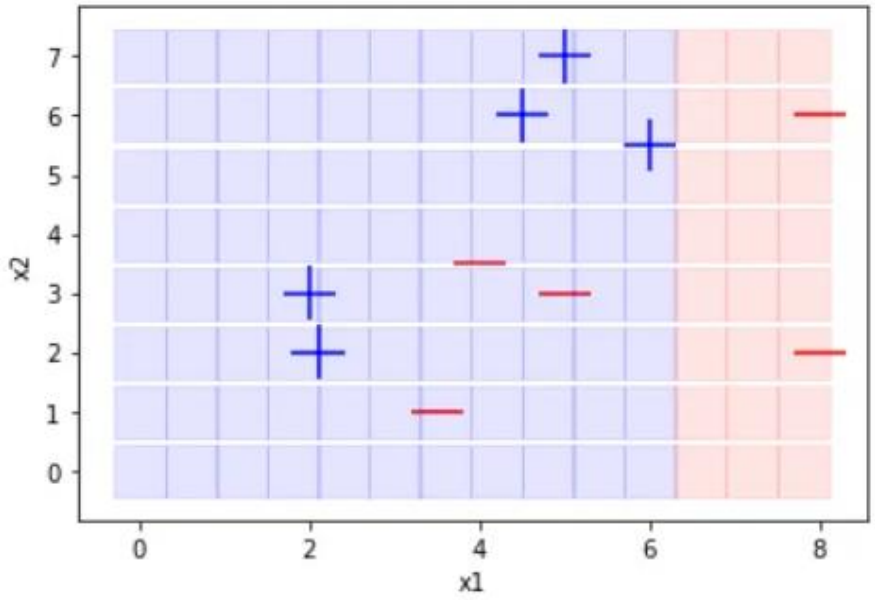
AdaBoost:

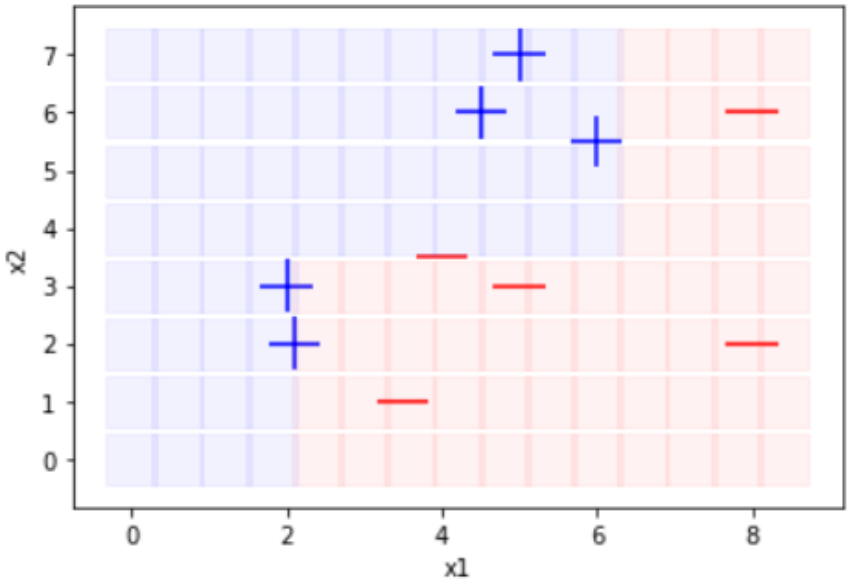
- *AdaBoost* is short for *Adaptive Boosting* and is a very popular boosting technique
- Combine several “weak classifiers” into a single “strong classifier”.
- It was proposed by Yoav Freund and Robert Schapire. They also won the 2003 Gödel Prize for their work.

AdaBoost:

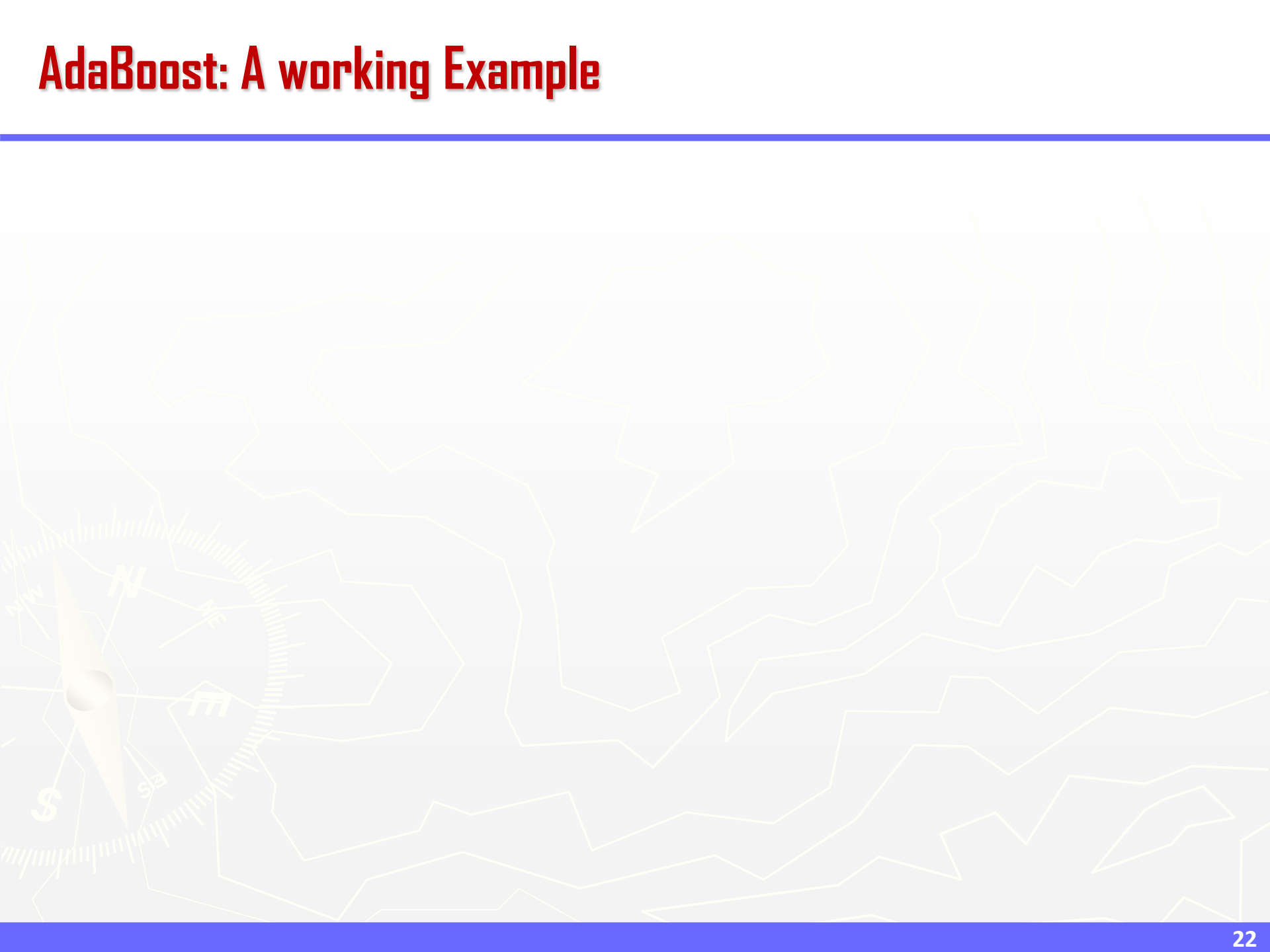

x2
x1
0
1
2
3
4
5
6
7
0
2
4
6
8


x2
x1
0
1
2
3
4
5
6
7
0
2
4
6
8


x2
x1
0
1
2
3
4
5
6
7
0
2
4
6
8

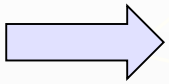

x2
x1
0
1
2
3
4
5
6
7
0
2
4
6
8

AdaBoost: A working Example



AdaBoost: A working Example

x1	x2	Decision
2	3	true
2.1	2	true
4.5	6	true
4	3.5	false
3.5	1	false
5	7	true
5	3	false
6	5.5	true
8	6	false
8	2	false



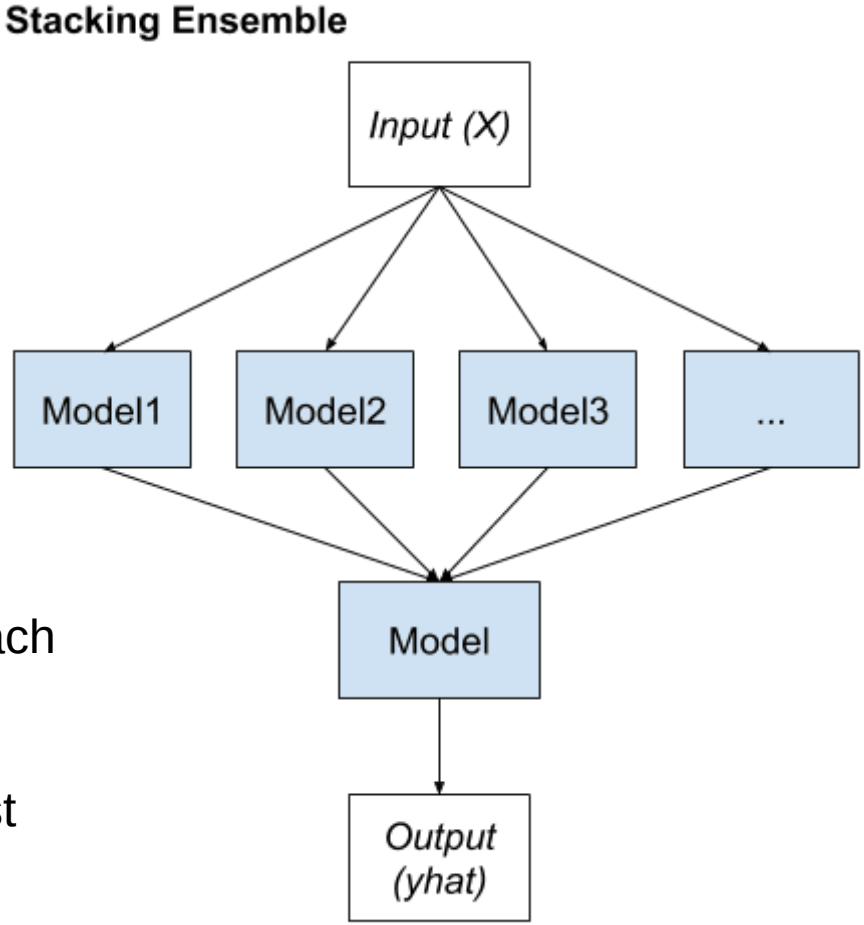
AdaBoost: A working Example

AdaBoost: A working Example

AdaBoost: A working Example

Ensemble Learning: stacking

Stacked Generalization, or stacking for short, is an ensemble method that seeks a diverse group of members by varying the model types fit on the training data and using a model to combine predictions.



- Unchanged training dataset.
- Different machine learning algorithms for each ensemble member.
- Machine learning model to learn how to best combine predictions.

THANKYOU