DATA MINING REVIEW

REFERENCES

- Slides adapted from
  - **Chapter 4 of Python M*achine learning by Sebastian***

PREDICTIVE MODELS

- Classification and class probability estimation
- Regression
- Clustering

CLASSIFICATION AND CLASS PROBABILITY ESTIMATION

- Goal
  - Predict the class, an individual from a population belongs to. Usually there are few mutually exclusive classes.
- Example
  - Among all the patients treated with chemotherapy, Which are likely to survive for 5 years?
  - Two classes: "Will survive" and " Will not survive"
- Procedure: Learn from historical data with known features and use it to predict future data with unknown class labels

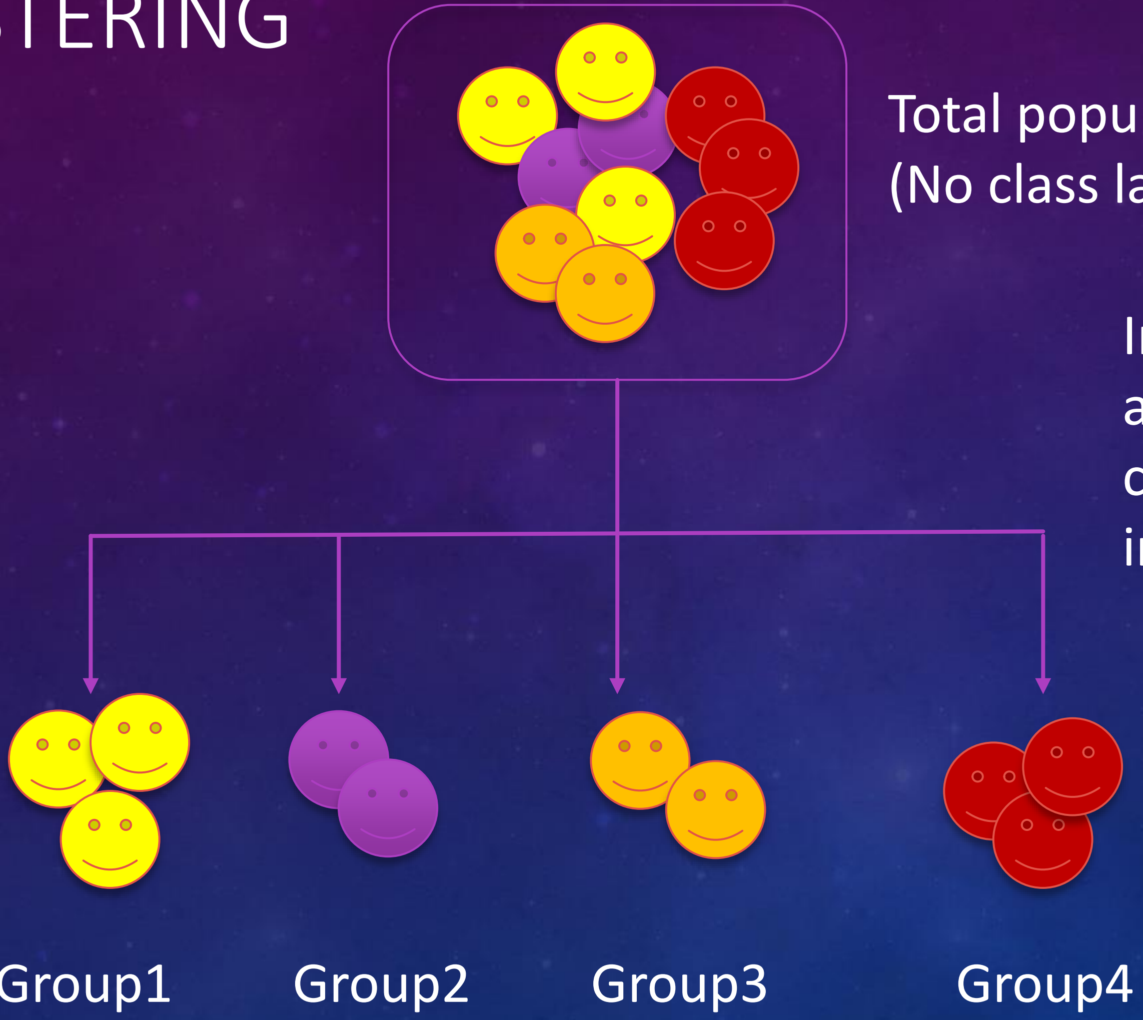
REGRESSION

- Problem/Goal
  - For each individual, estimate/predict the numerical value of some variable for that individual.
- Example
  - "How much will a given customer use the service".
  - The variable of the individual is *service usage*.

CLASSIFICATION VERSUS REGRESSION

- Classification predicts if something will happen or not.
  - Classification assigns a class.
- Regression predicts how much something will happen.
  - Regression estimates an amount of an event.

CLUSTERING

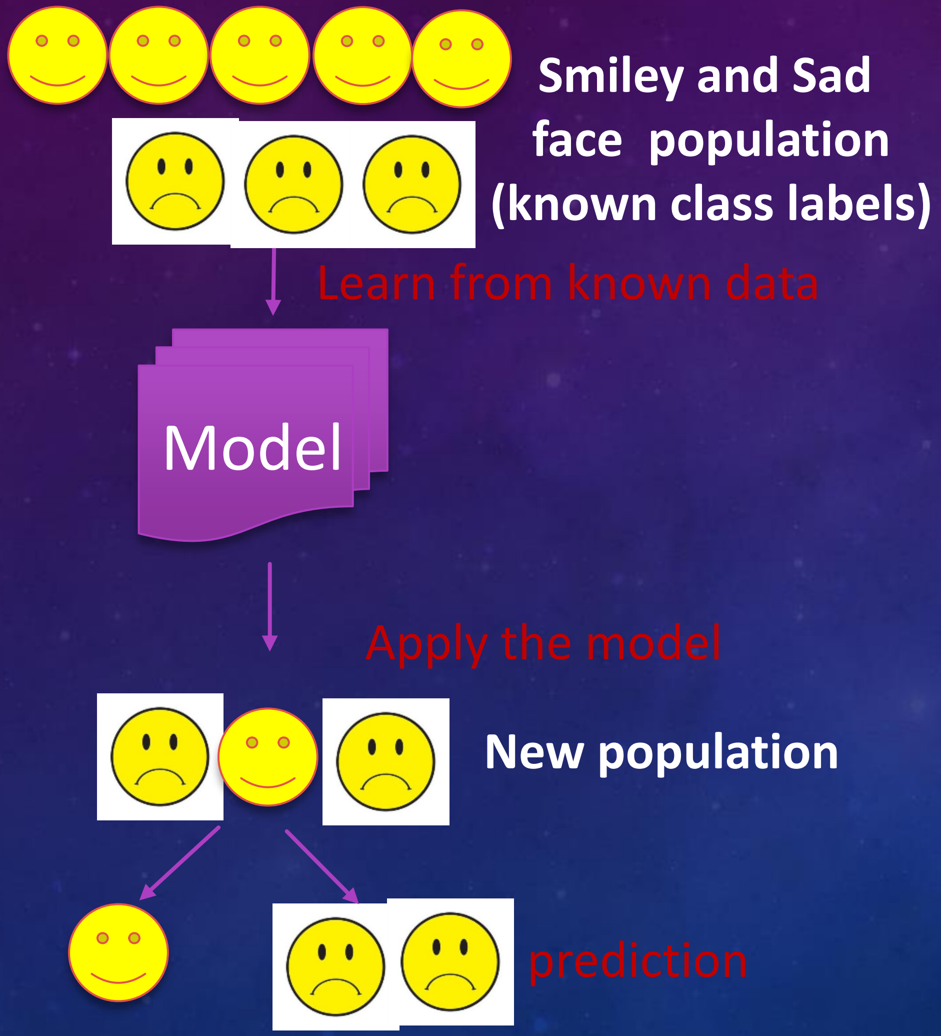


Individuals within the group are more similar to each other compared to the individuals in other groups.

MORE EXAMPLES OF CLUSTERING

- Social network analysis
  - Detect communities.
- Image Processing
  - Group different pixels into groups for object recognition.
- Medical imaging
  - Cluster different pixels of Scans to differentiate different types of blood, tissues etc.
- Crime analysis
  - Cluster analysis can be used to identify areas where there are greater incidences of particular types of crime.

SUPERVISED



Smiley and Sad face population (known class labels)

Learn from known data

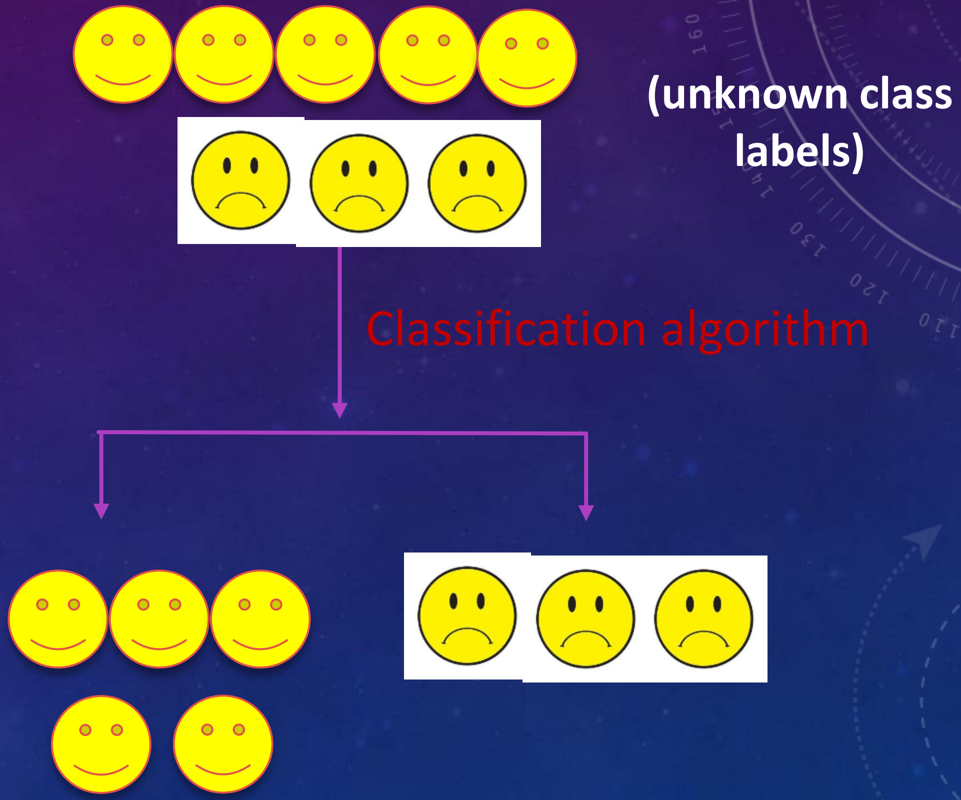
Model

Apply the model

New population

prediction

Unsupervised



(unknown class labels)

Classification algorithm

PREDICTION: PROBLEM

A customer nearing the end of a phone contract

Minutes usage / month	Number of Late payments	Number of phone lines	Switched service ?
300	2	3	?

We want to predict if this person will switch or not ?
(The Problem of churn)

STEP 1: PREPARE HISTORICAL DATA FOR MANY INDIVIDUALS

Target Attribute

Attributes/Features

Training Data Set

Minutes usage / month	Number of Late payments	Number of phone lines	Switched service ?
110	2	1	YES
50	0	3	NO
..	..	..	..
100	6	1	YES

- Collect and prepare data in a particular format (usually Tabular format).
- Make sure there are no missing values.
- Make sure there is enough data available.

STEP 1: PREPARE HISTORICAL DATA FOR MANY INDIVIDUALS

Attributes/Features

Target Attribute

Training Data Set

Minutes usage / month	Number of Late payments	Number of phone lines	Switched service ?
110	2	1	YES
50	0	3	NO
..	..	..	..
100	6	1	YES

- Each row is an instance or example
- Feature vector <100, 6 , 1, YES>.
- Attribute can be numeric or categorical.

STEP 2: BUILD A MODEL THAT DESCRIBES THE HISTORICAL DATA

Training Data Set

Minutes usage / month	Number of Late payments	Number of phone lines	Switched service ?
110	2	1	YES
50	0	3	NO
..	..	..	..
100	6	1	YES

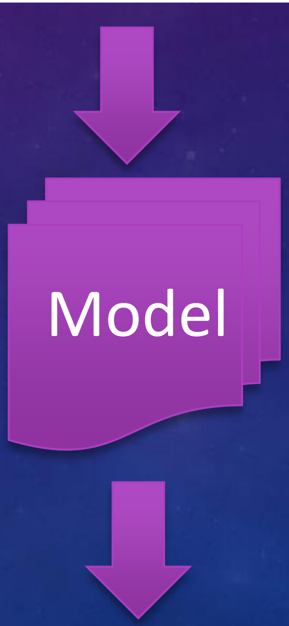
Data mining/ Exploration

Model

STEP 3: PREDICTION USING THE MODEL

Minutes usage / month	Number of Late payments	Number of phone lines	Switched service ?
300	2	3	?

to be predicted?



Will switch
Probability: 0.88

PREDICTIVE MODELS

- Tree-like
  - Decision Trees
  - Random Forest
  - Boosted Trees
- Probabilistic methods
  - Naïve Bayes
  - Logistic Regression
- Kernel methods
  - Support Vector Machines (SVM)
- Neural Networks (Deep Learning)

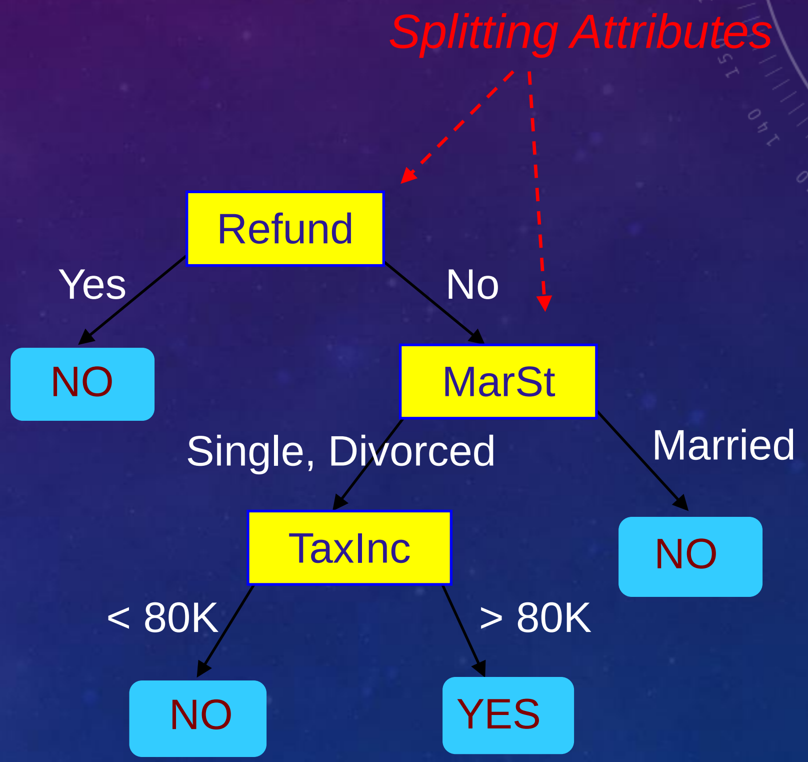
TREE MODELS AND ENSEMBLE TREE MODELS

- Decision Trees (weak Learner)
- Random Forest (Collection of weak learners)
- Boosting: Boosted Trees (Collection of weak learners)

DECISION TREES

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	**No**
2	No	Married	100K	**No**
3	No	Single	70K	**No**
4	Yes	Married	120K	**No**
5	No	Divorced	95K	**Yes**
6	No	Married	60K	**No**
7	Yes	Divorced	220K	**No**
8	No	Single	85K	**Yes**
9	No	Married	75K	**No**
10	No	Single	90K	**Yes**

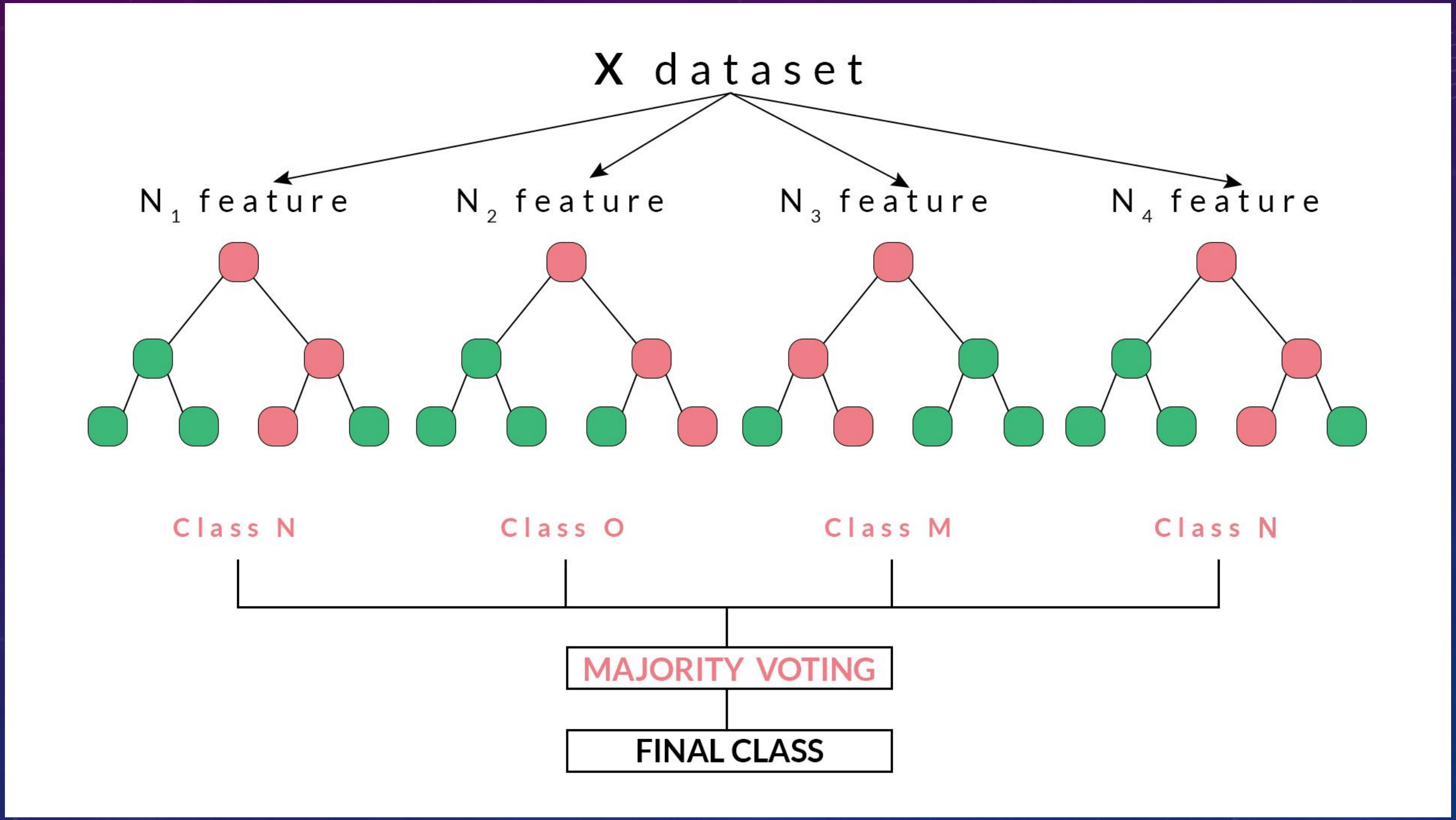
Training Data



Model: Decision Tree

Tan et al, Intro to data mining

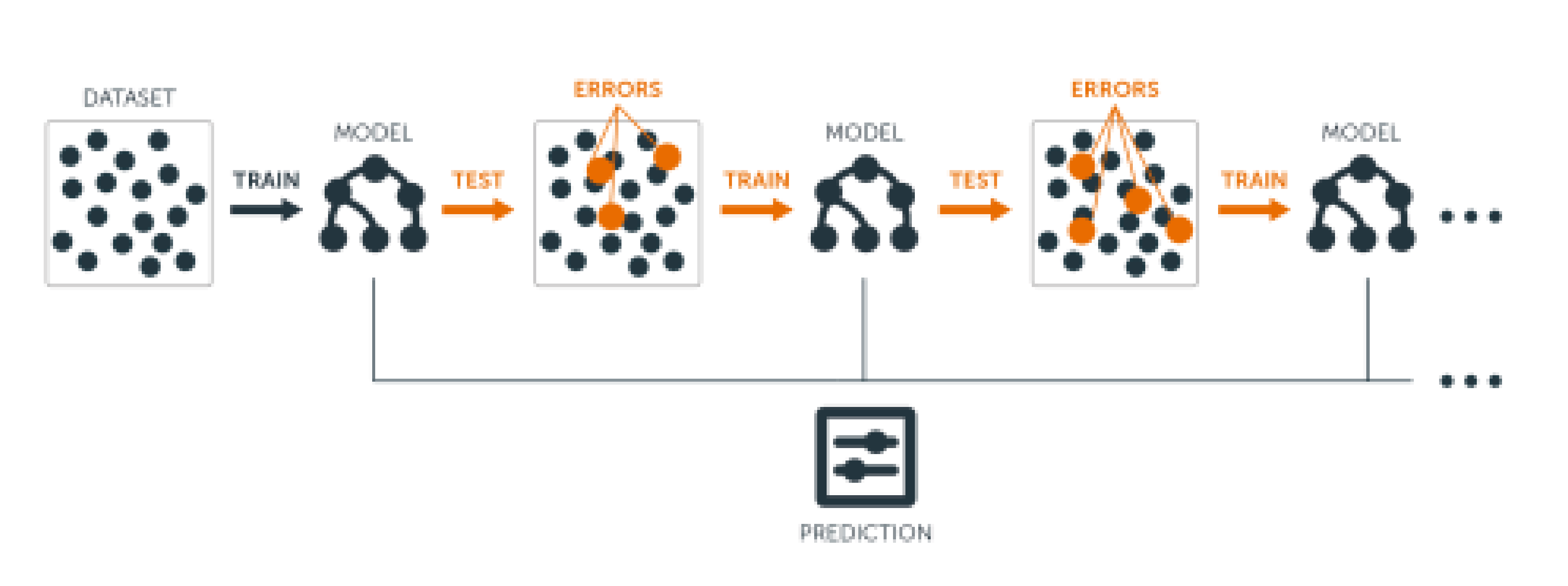
RANDOM FOREST



Features chosen independently for each tree

https://blog.quantinsti.com/random-forest-algorithm-in-python/

BOOSTED TREES

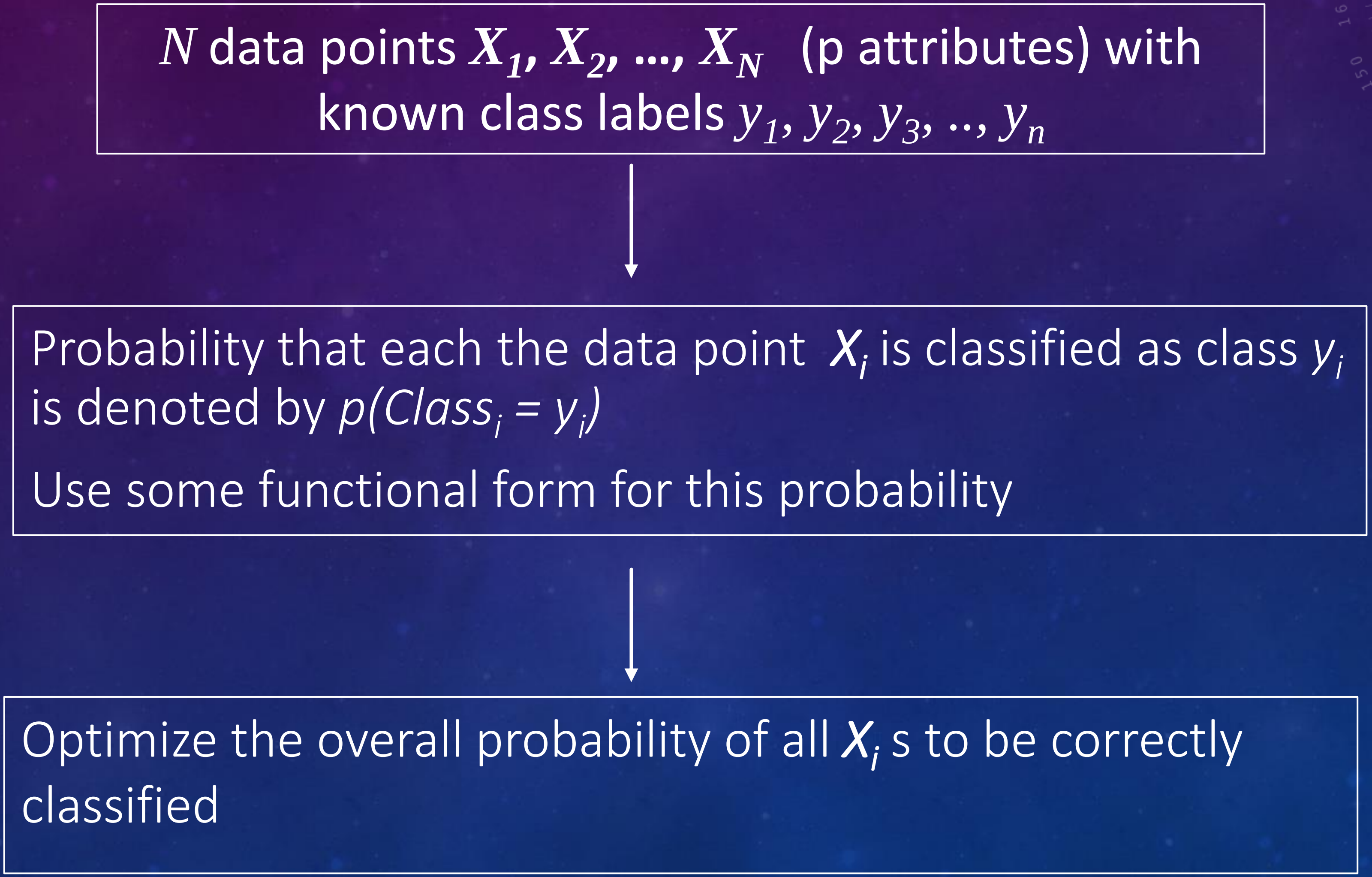


Individual Trees built sequentially and added

PROBABILISTIC MODELS

- Naive Bayes
- Logistic regression

PROBABILISTIC MODELS



N data points $\boldsymbol{X_1, X_2, \ldots, X_N}$ (p attributes) with known class labels $y_1, y_2, y_3, .., y_n$

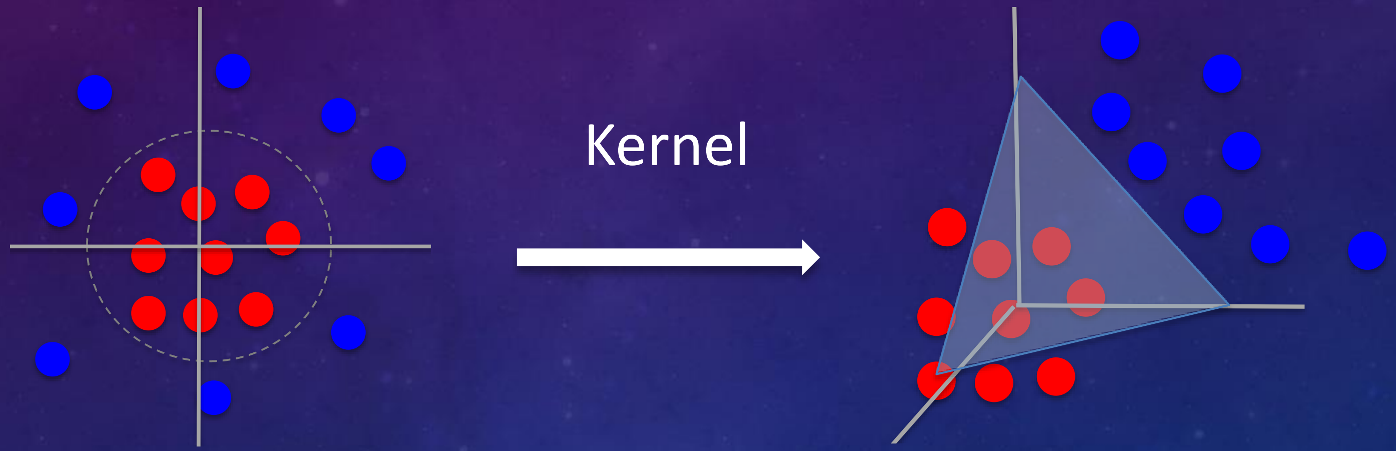
Probability that each the data point X_i is classified as class y_i is denoted by $p(Class_i = y_i)$

Use some functional form for this probability

Optimize the overall probability of all X_i s to be correctly classified

KERNEL METHODS

- Support vector machines



Kernel

Not linearly separable data with two classes

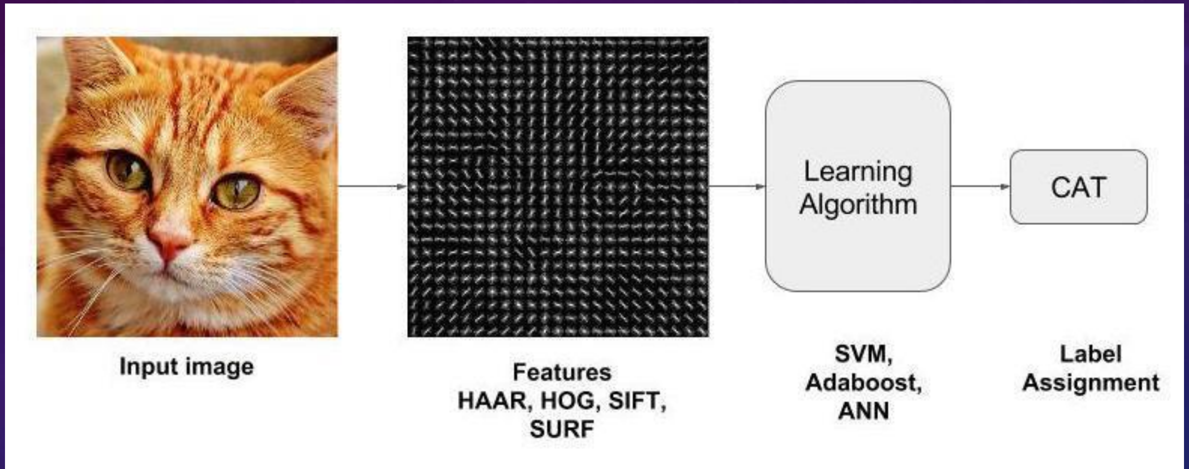
linearly separable data in higher dimensions
Then find the linear optimal Hyperplane

ISSUES: VALIDATING OR EVALUATING PA MODELS

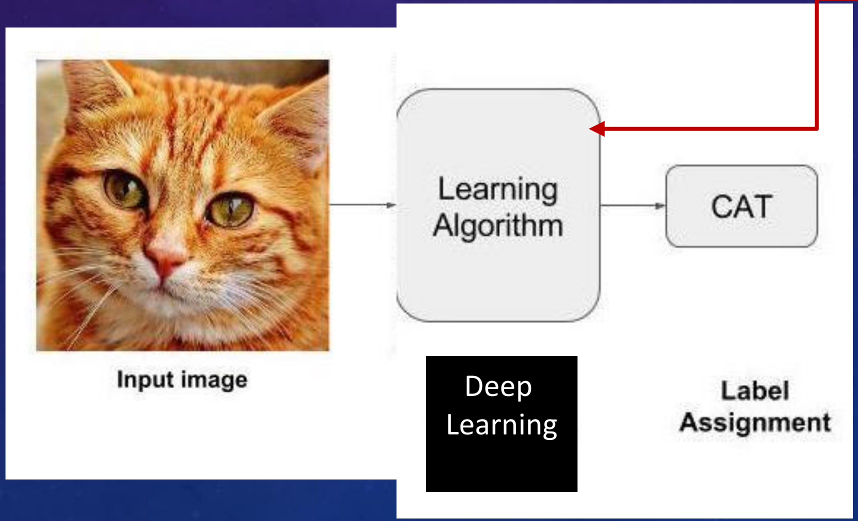
- There can be multiple models to solve the same problem.
- Predictive accuracy.
  - How well the model can predict ?
- Speed and scalability.
  - Time needed to build the model.
  - Time used by the model in predicting.
- Interpretability.
  - Can you understand the rules of the model ?

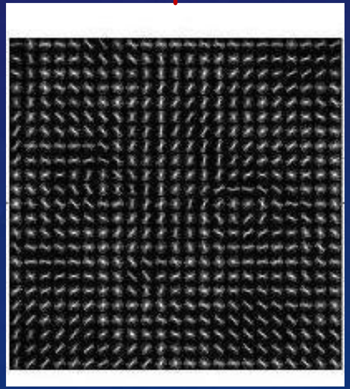
TRADITIONAL MODEL VS DEEP LEARNING

Traditional Models



Deep Learning

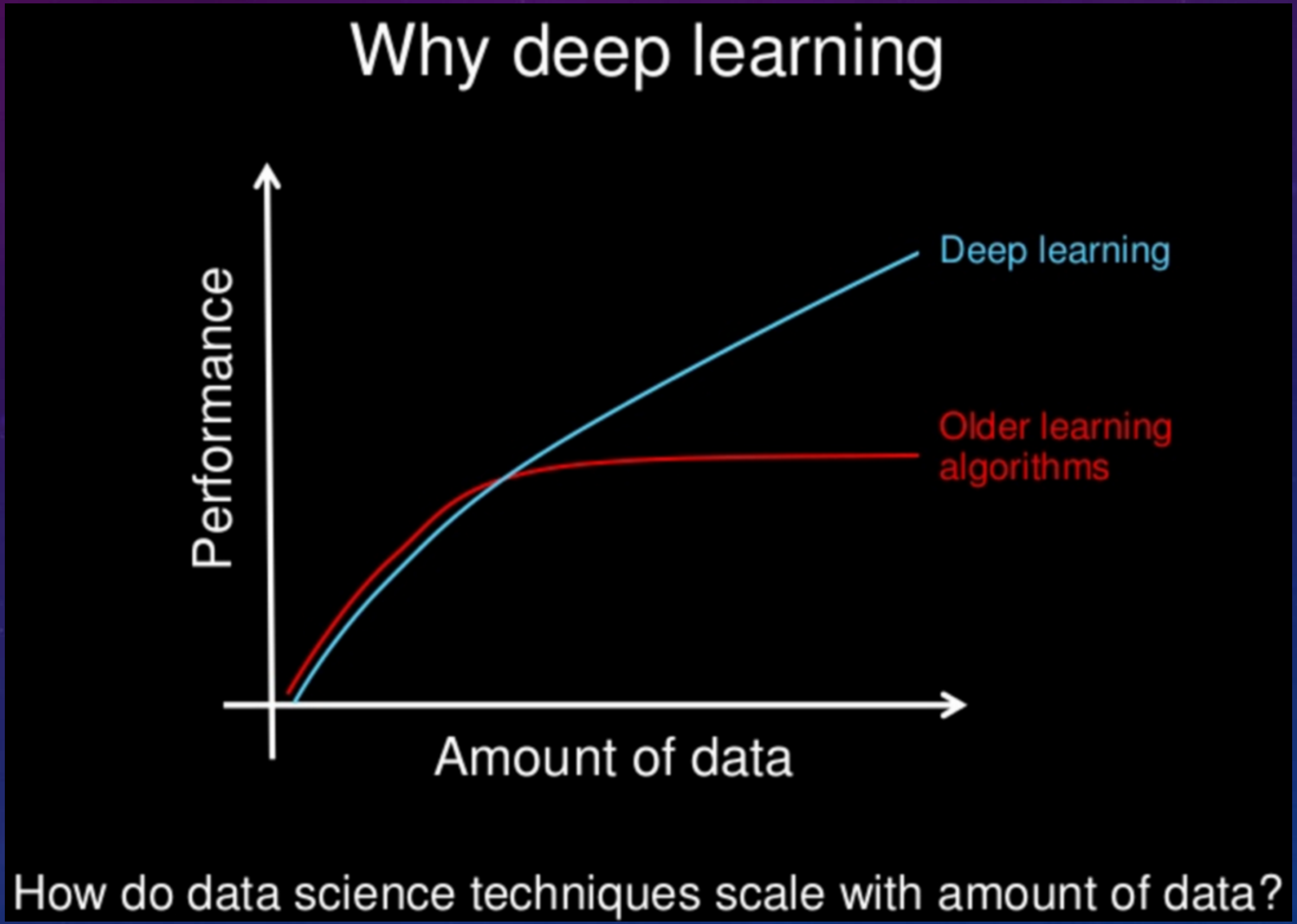




Features are learned automatically

https://www.learnopencv.com/image-recognition-and-object-detection-part1/

DEEP LEARNING NEEDS LARGE AMOUNT OF DATA



PREPROCESSING

- Read the data (features and class labels) from a file and store in a data structure
  - Pandas Data Frame or Numpy matrix (can store only numerics) in Python
  - Data Frame in R
- User can choose to remove any unwanted features or samples (primarily manual at this point).

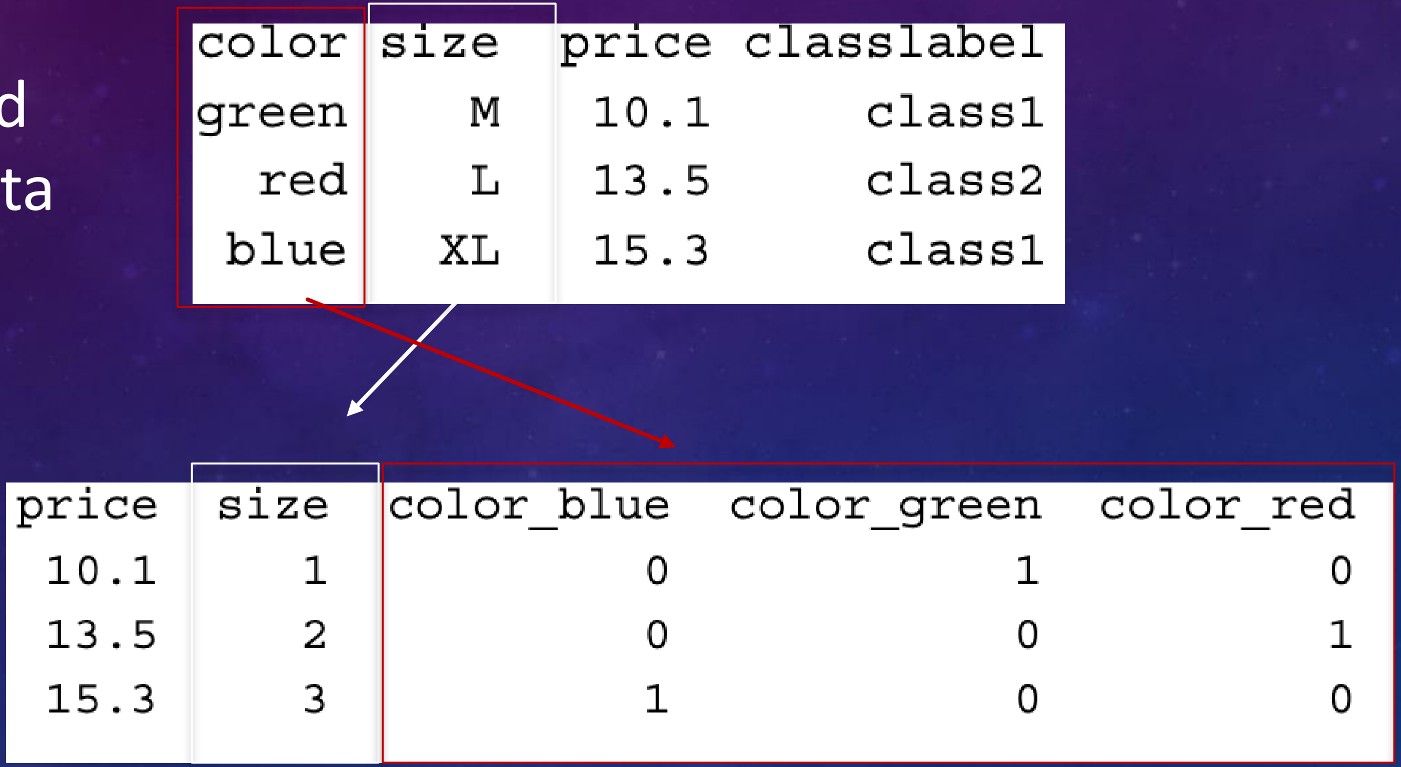
DEALING WITH MISSING VALUES

- Remove rows with missing values in features.
- You may also want to remove features (columns) that have too many missing value in many samples.
- Imputation of missing values
  - Different interpolation techniques to estimate the missing values from the other training samples in our dataset.

DEALING WITH CATEGORICAL FEATURES

- Categorical features should be converted to numerical values.
- Categorical data: **nominal** and **ordinal** features.
- Ordinal features: can be sorted or ordered.
  - For example, *T-shirt size: XL > L > M*
  - Assign XL = 3, L = 2, and M = 1
- Nominal features: can't be sorted.
  - For example: *T-shirt color (Red, Green, Blue)*
  - One-hot encoding: Red -> [1,0,0], Green -> [0,1,0], Blue = [0, 0, 1]
  - This will require additional features to be added.

DEALING WITH CATEGORICAL FEATURES



Old data

color	size	price	classlabel
green	M	10.1	class1
red	L	13.5	class2
blue	XL	15.3	class1

New data

price	size	color_blue	color_green	color_red
10.1	1	0	1	0
13.5	2	0	0	1
15.3	3	1	0	0

SPLIT DATA AND FEATURE SCALING

- Split the data into training (80%) and testing (20%)
  - Make sure 80% of each class go to training
- Feature scaling: bring features to the same scale.
  - Note this step should be done after splitting the data into training and testing sets
  - First, scale the training set.
  - Use the parameters used in scaling training set to scale the testing set.
  - Don’t scale the testing set independently from training set.

FEATURE SCALING

- Feature scaling: Process of bringing all the features to the same scale.
- Why ?: So that model can “weigh” and “interpret” the features in a non bias way.
- Normalization: scale between 0 and 1 (min-max) scaler

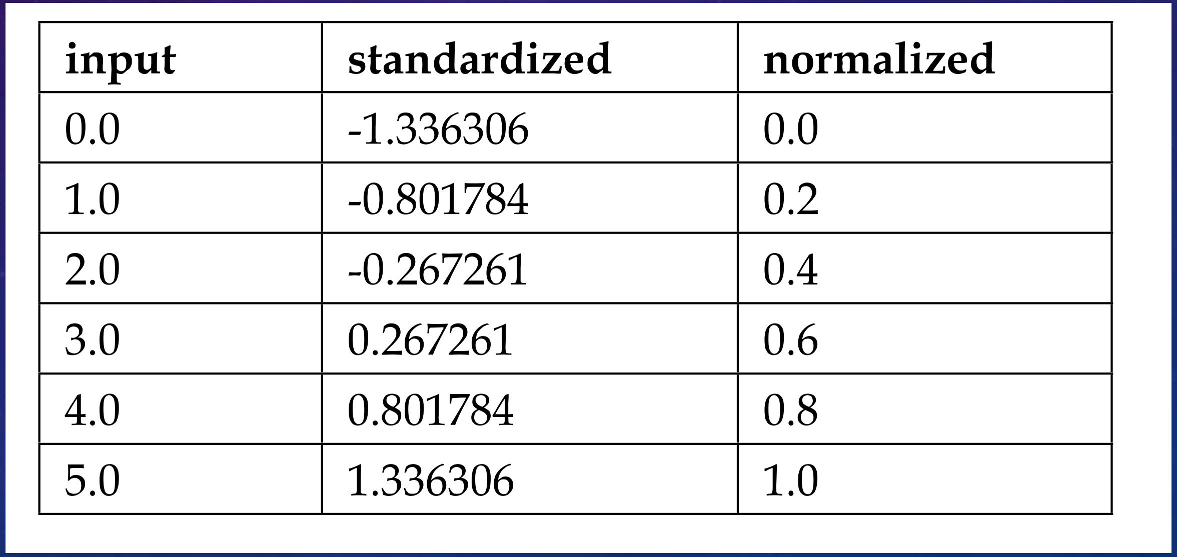
$$x_{norm}^{(i)} = \frac{x^{(i)} - x_{\min}}{x_{\max} - x_{\min}}$$

- Standardization: scale so that values have mean = 0 and standard deviation = 1

$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

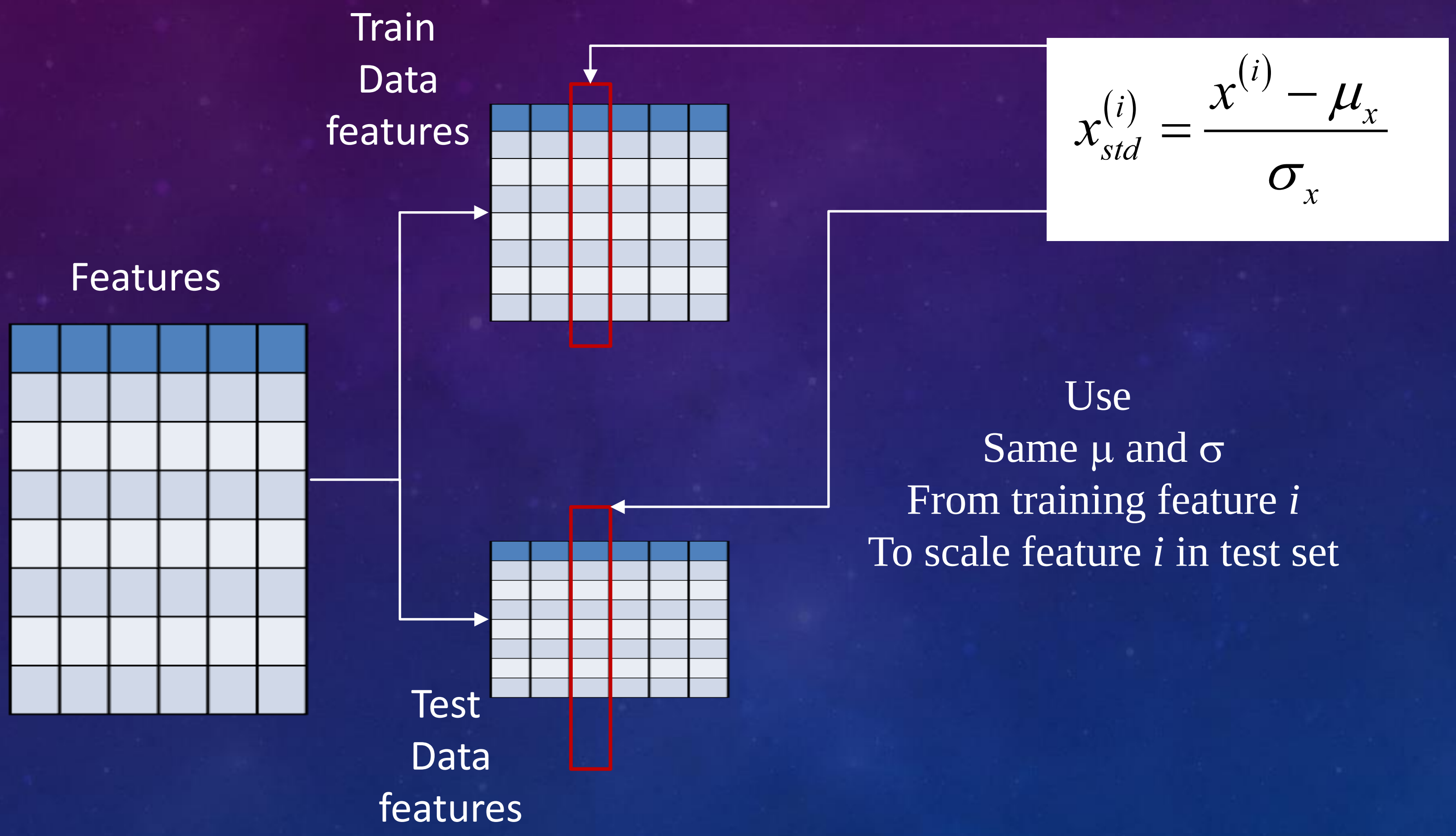
FEATURE SCALING

- The following table illustrates the difference between the two commonly used feature scaling techniques, standardization and normalization on a simple sample dataset consisting of numbers 0 to 5:



input	standardized	normalized
0.0	-1.336306	0.0
1.0	-0.801784	0.2
2.0	-0.267261	0.4
3.0	0.267261	0.6
4.0	0.801784	0.8
5.0	1.336306	1.0

FEATURE SCALING



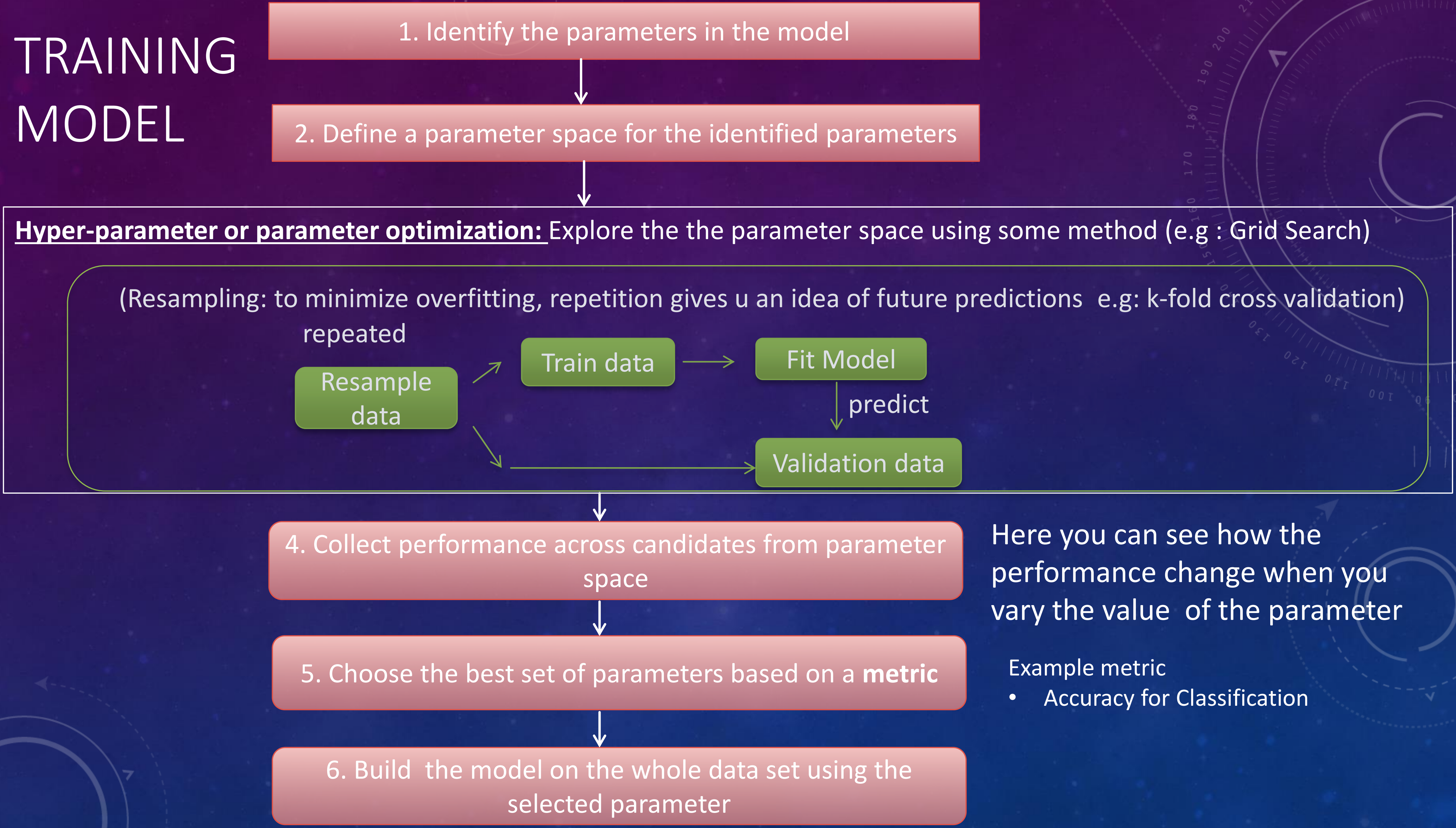
$$x_{std}^{(i)} = \frac{x^{(i)} - \mu_x}{\sigma_x}$$

Use
Same μ and σ
From training feature *i*
To scale feature *i* in test set

CLASS IMBALANCE

- If there is class imbalance, make the training set balanced
  - Up sampling.
  - Down sampling .
  - SMOTE (Synthetic Minority Oversampling Technique).
  - Generative Adversarial Networks (GANs) – Deep Learning method.
- Train the balanced training set.
- Keep the testing set imbalanced.
  - This should represent an accurate class composition in the future test data.

TRAINING MODEL



1. Identify the parameters in the model

2. Define a parameter space for the identified parameters

Hyper-parameter or parameter optimization: Explore the the parameter space using some method (e.g : Grid Search)

(Resampling: to minimize overfitting, repetition gives u an idea of future predictions e.g: k-fold cross validation)

4. Collect performance across candidates from parameter space

5. Choose the best set of parameters based on a **metric**

6. Build the model on the whole data set using the selected parameter

Here you can see how the performance change when you vary the value of the parameter

Example metric

- Accuracy for Classification

PARAMETER (HYPER-PARAMETER) OPTIMIZATION

- Grid search cross validation
  - Try every possible candidate from parameter space
  - Ideal for small space
- Random search
  - Try a random set of possible candidates from parameter space
  - Ideal for small space
- **Bayesian optimization**
  - **Automated and “wise” choice of possible candidates from parameter space**

SAMPLING (VALIDATION) METHODS

- K-fold Cross Validation
- Leave one out cross-validation
- Boot Strap Cross Validation

K-FOLD (BELOW IS K=3 FOLD)

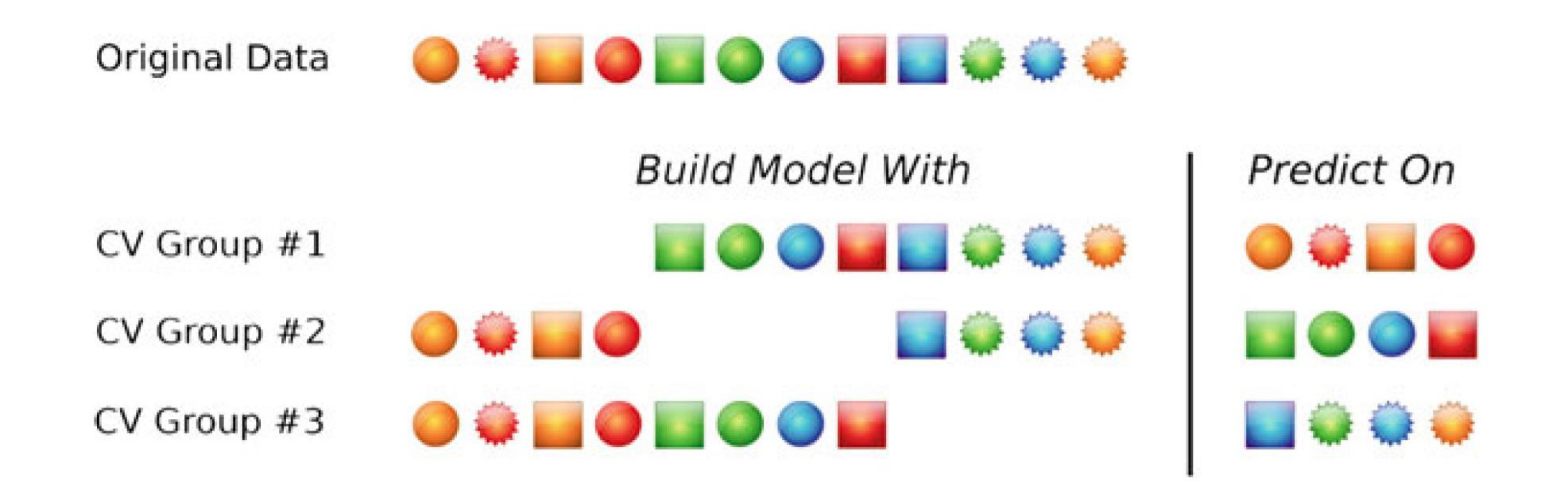


Fig. 4.6: A schematic of threefold cross-validation. Twelve training set samples are represented as symbols and are allocated to three groups. These groups are left out in turn as models are fit. Performance estimates, such as the error rate or R^2 are calculated from each set of held-out samples. The average of the three performance estimates would be the cross-validation estimate of model performance. In practice, the number of samples in the held-out subsets can vary but are roughly equal size

LEAVE-ONE-OUT CROSS-VALIDATION (LOOCV)

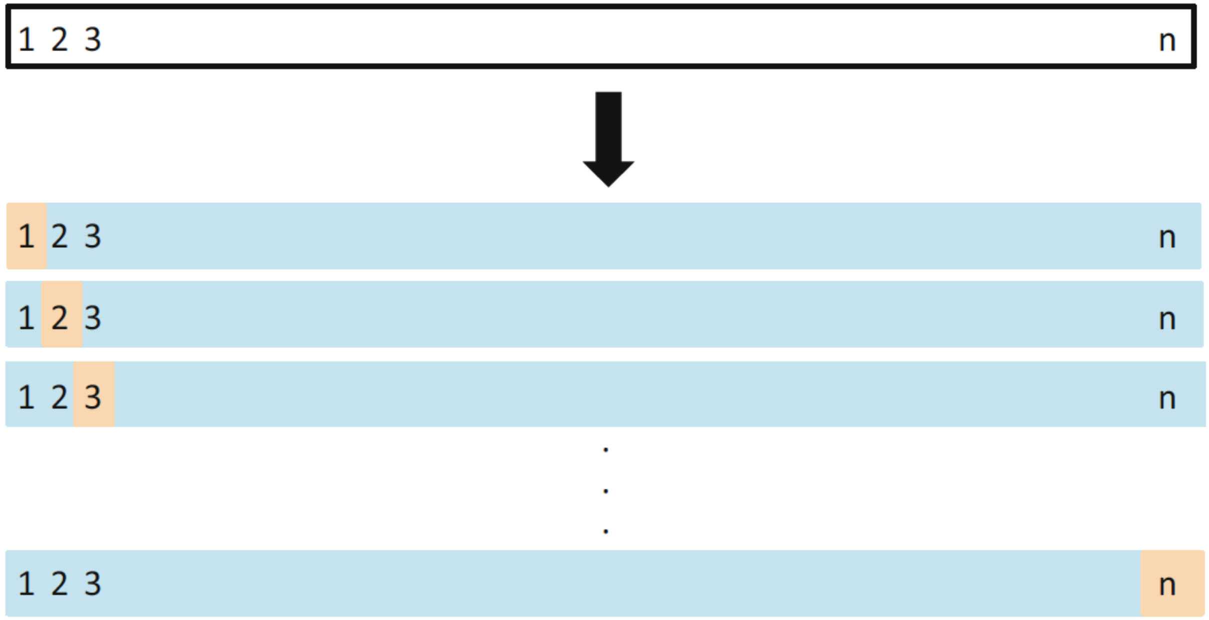


FIGURE 5.3. *A schematic display of LOOCV. A set of n data points is repeatedly split into a training set (shown in blue) containing all but one observation, and a validation set that contains only that observation (shown in beige). The test error is then estimated by averaging the n resulting MSE's. The first training set contains all but observation 1, the second training set contains all but observation 2, and so forth.*

BOOT STRAP CROSS VALIDATION

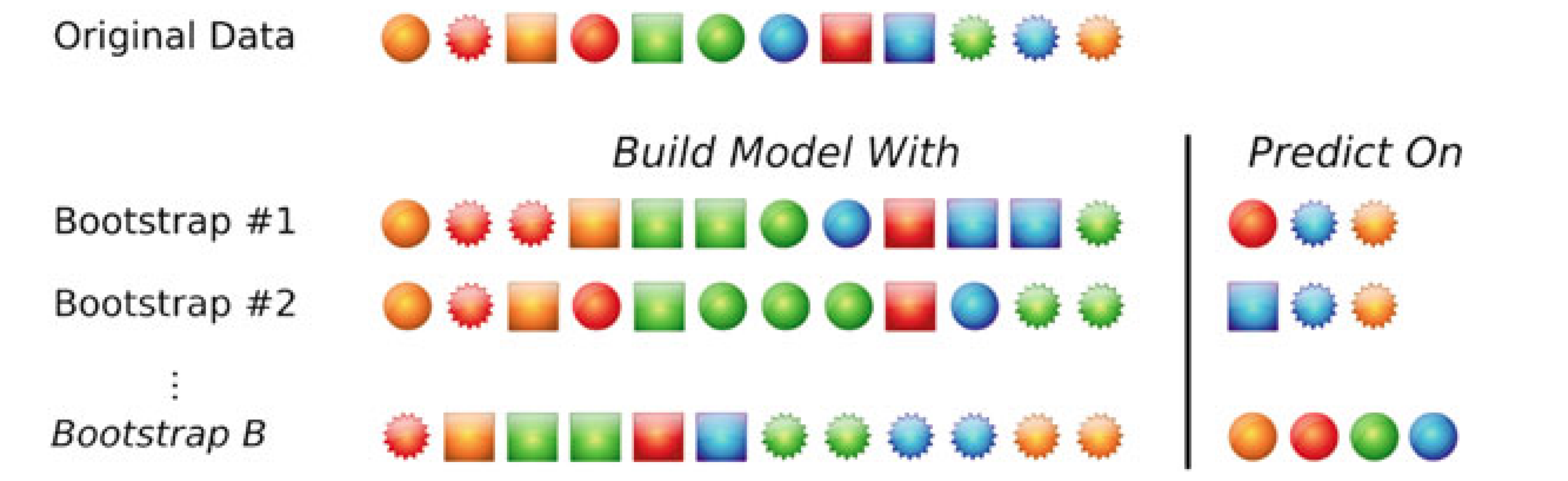


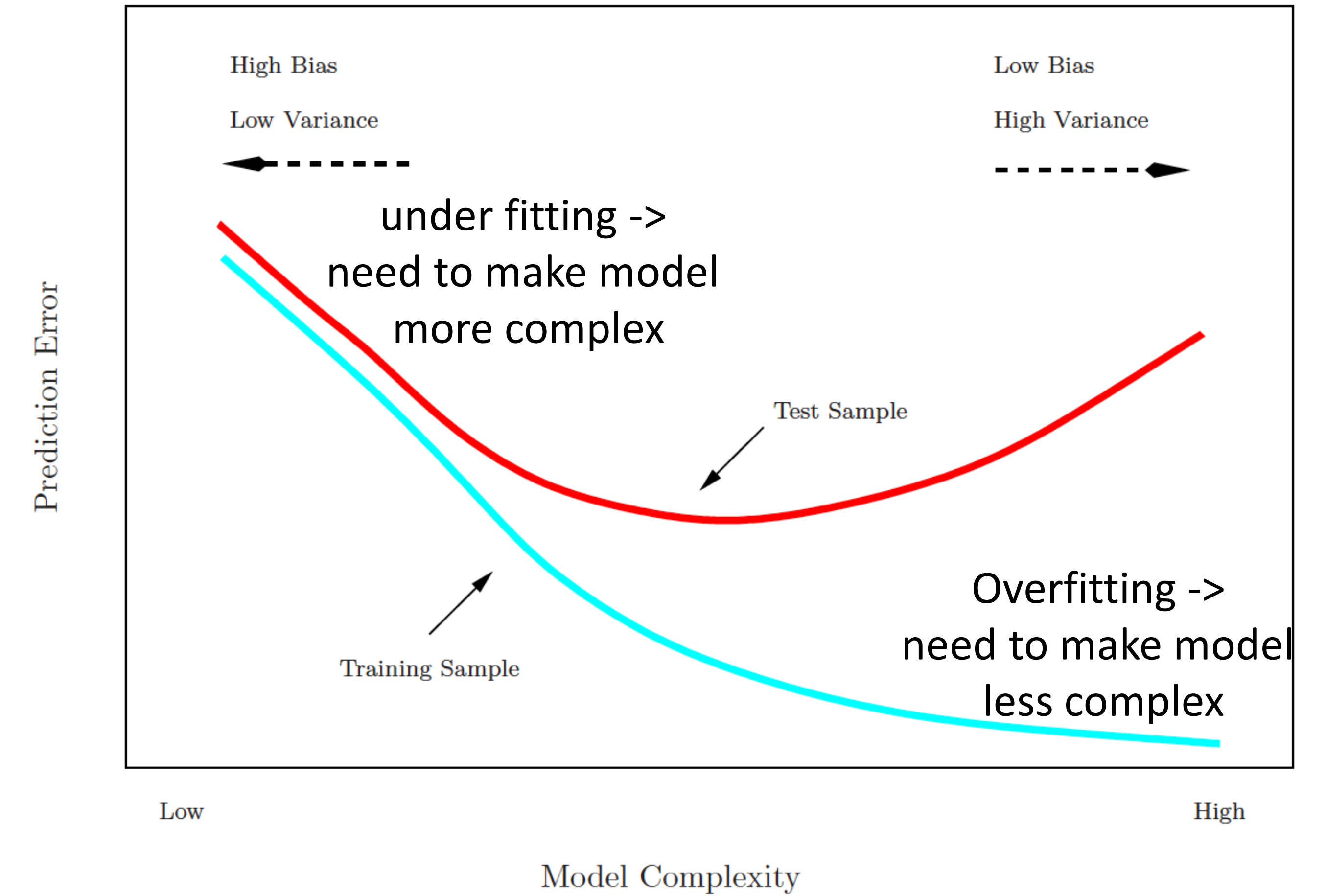
Fig. 4.8: A schematic of bootstrap resampling. Twelve training set samples are represented as symbols and are allocated to B subsets. Each subset is the same size as the original and can contain multiple instances of the same data point. Samples not selected by the bootstrap are predicted and used to estimate model performance

PROS CONS OF SAMPLING TECHNIQUES

- LOOCV
  - All training samples overlap too much
  - Less bias in training (less training error)
  - High variance in testing errors (not consistent with different test data)
- K-fold
  - Training samples overlap less than LOOCV
  - bias-variance balance
  - Repeated K-fold may perform better than k-fold
- Bootstrap
  - Less variance (consistent with different test data)
  - May be high bias (higher training error)
- But depends on the data.

TRAINING AND TEST PERFORMANCE

- **Bias** is the difference between the average prediction of our model and the correct value which we are trying to predict.
- **Variance** is the variability of model prediction for a given data point or a value which tells us spread of our data.



under fitting -> need to make model more complex

Overfitting -> need to make model less complex