A Taste of Data Science and Machine Learning: a Hands-on Approach

beirut artificial intelligence startupweekend™
Powered by Google for Entrepreneurs

techstars

Elie Kawerk

Outline

- Intro to Data Science and Machine Learning.
- Optimizing functions with gradient descent.
- Linear regression and Logistic Regression.
- Overfitting and ways to combat it.
- Decision Trees and Ensemble Methods.
- Cross Validation for Model Selection and Hyperparameter Tuning.

Data Science

Data Science is an interdisciplinary field that aims at:

- solving problems by extracting insights from structured and unstructured data,
- building predictive and descriptive models of data.

Applications of Data Science

- Spam filtering (email)
- Recommender systems (Amazon, Facebook, …)
- Stock market prediction
- Churn prediction
- Computer vision
- Speech recognition
- Fraud detection
- and more ……..

The Data Science Process

Ask a question about your business and how collecting data can help; Set a **target** and **goals**

↓

Design an **experiment** to collect data

↓

Collect raw **data** and **complement** it with other sources such as the web or a database

↓

Explore, Manipulate, Visualize, Wrangle and **Clean** Data; **Engineer** new meaningful **Features**

↓

Build Machine Learning **Models** & Perform **Statistical Analyses**

↓

Communicate your results clearly through **storytelling** and **visualization; Explain** how your analysis helps solving the problem and achieves the set goals

Machine Learning

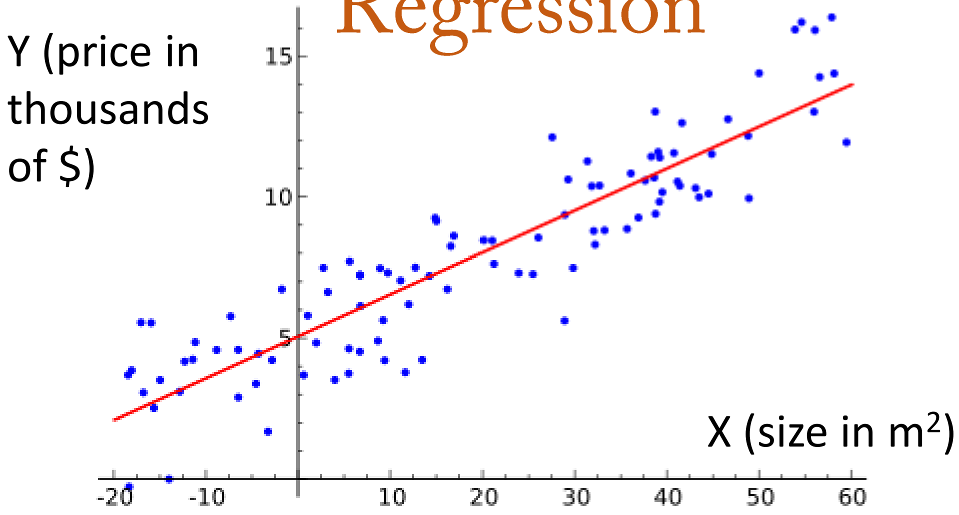
Machine learning is the subfield of computer science that, according to Arthur Samuel in 1959, gives "computers the ability to learn without being explicitly programmed.“

- Wikipedia

Types of Machine Learning

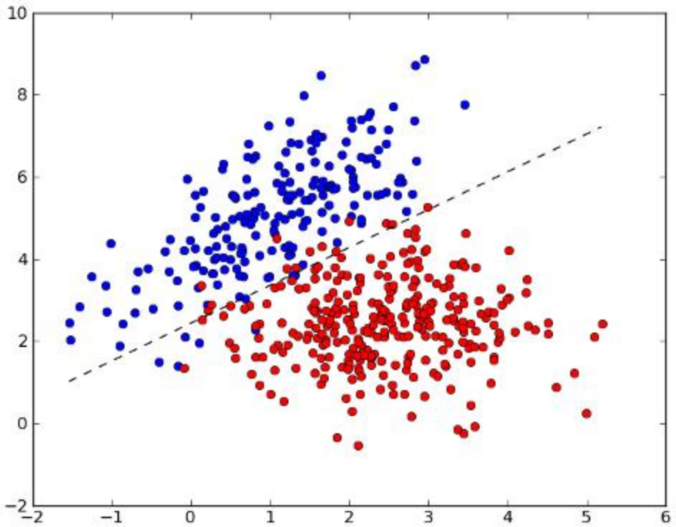
Supervised learning: features (X) and labels (y) are given. The task is to learn the mapping f between X and y.

Regression



f: X→ y

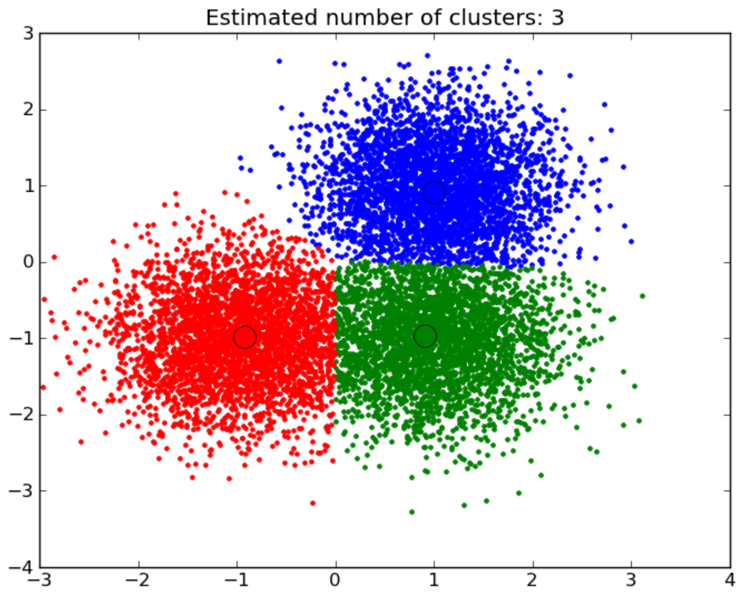
Classification



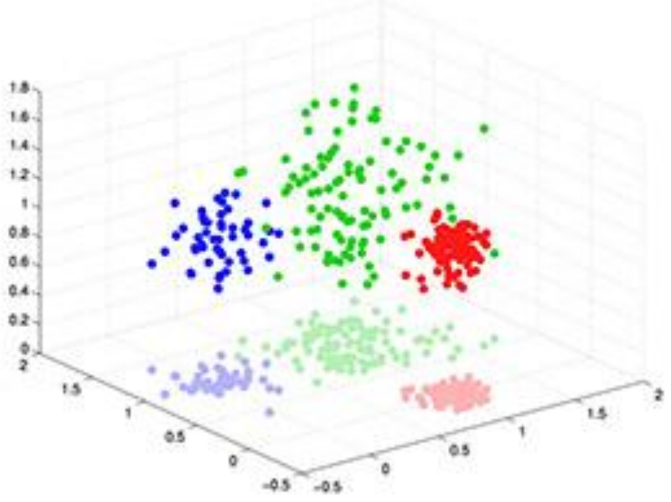
Types of Machine Learning

Unsupervised learning: only features (X) are provided.

Clustering

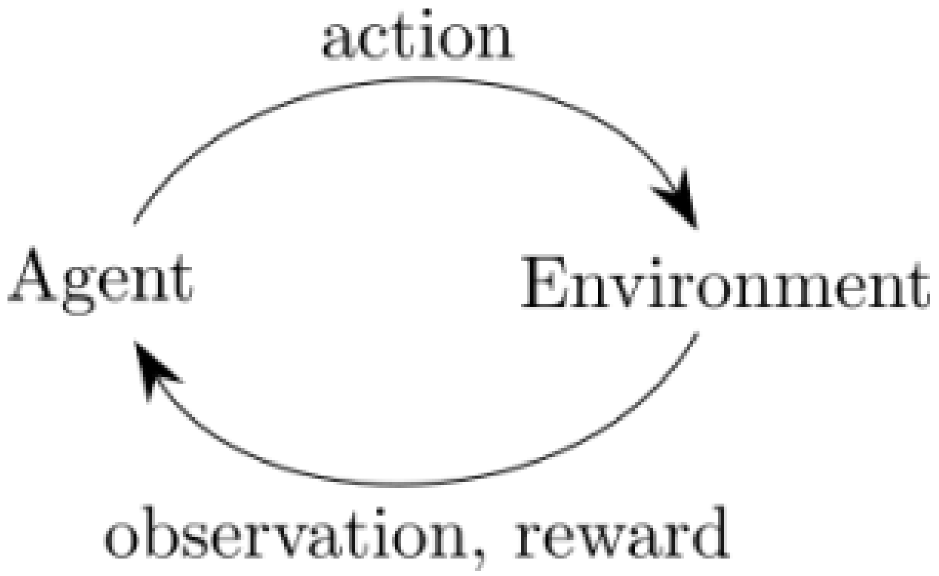


Dimensionality reduction



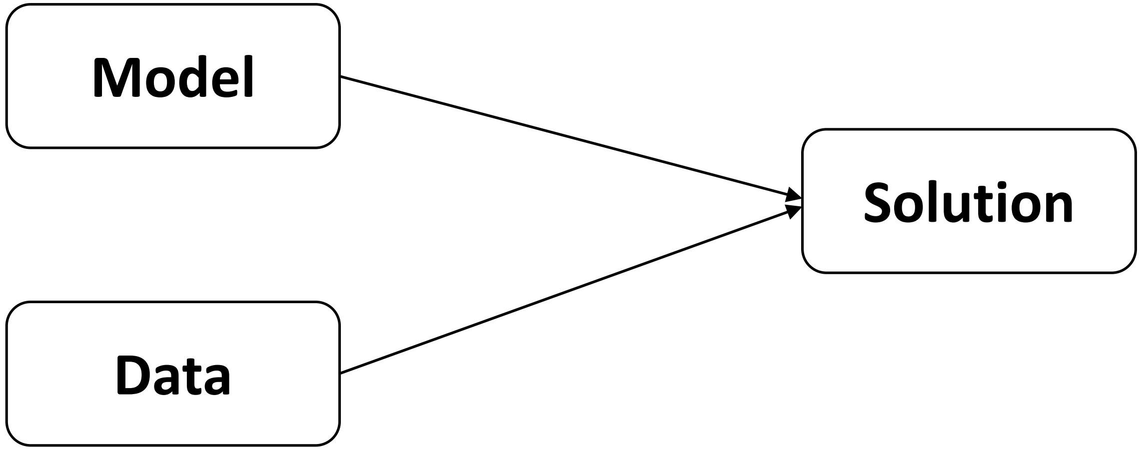
Types of Machine Learning

Reinforcement learning



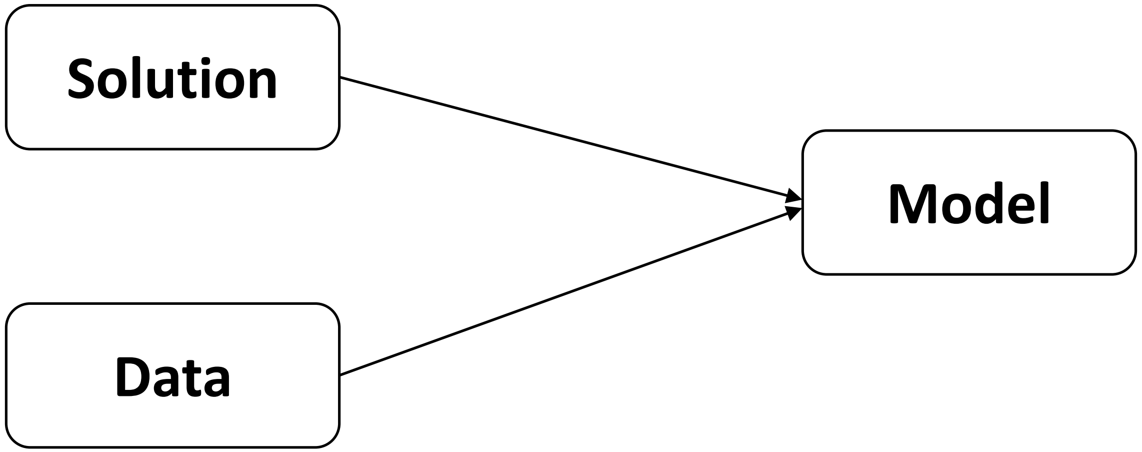
Machine Learning vs. Classical Programming

Traditional Software Engineering



Machine Learning vs. Classical Programming

Supervised Machine Learning



Optimization with Gradient Descent

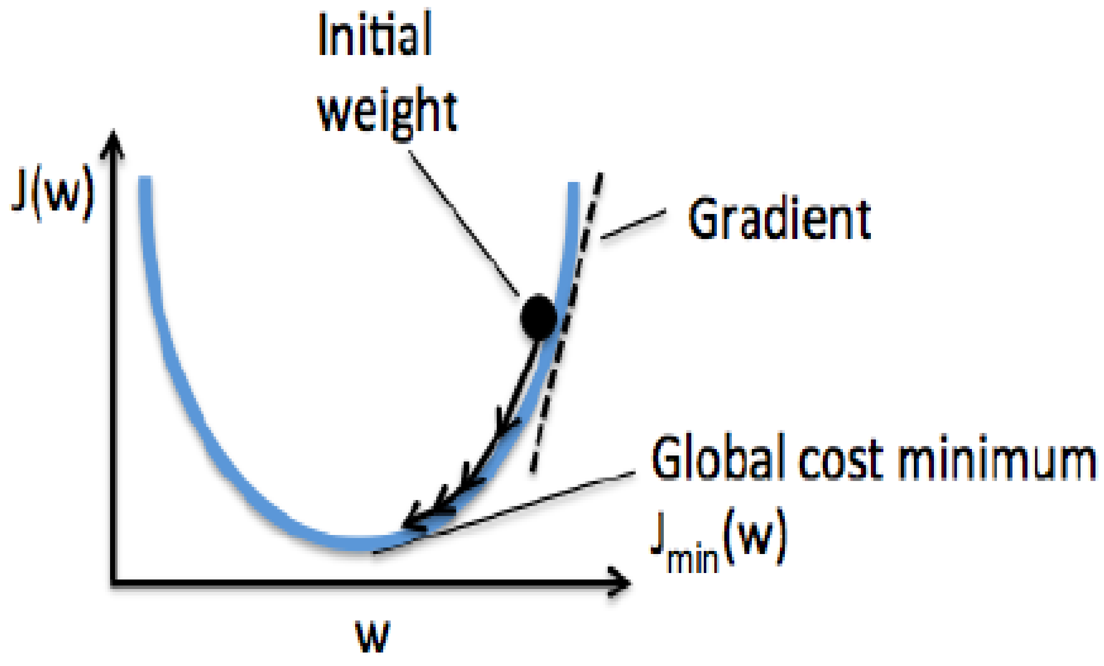
- Machine learning involves learning a model's parameters from data.
- In supervised learning, a model must fit data points.
- In a lot of cases, fitting models to data can be reduced to an optimization problem.
- Usually, we deal with the optimization of convex functions.

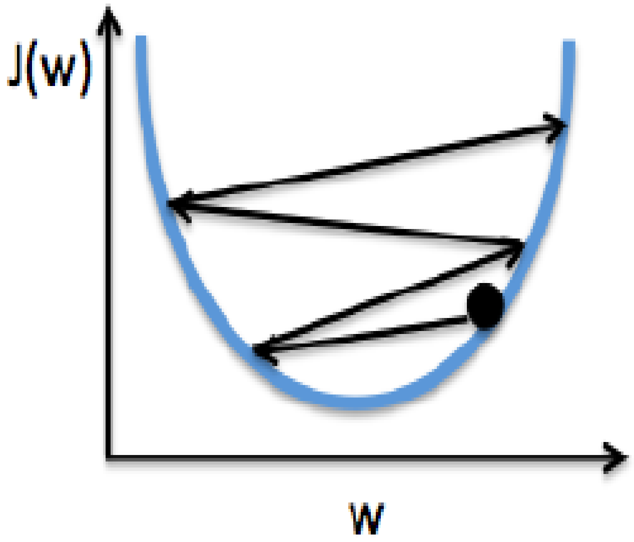
Optimization with Gradient Descent

- In the simplest form, optimization consists of minimizing or maximizing a real function.

- The function may be very complex and multivariate.

Optimization with Gradient Descent





J(w) denotes the function of w that we want to minimize

$$w_{optimal} = argmin_w J(w)$$

Optimization with Gradient Descent

Algorithm

1. Initialize X with some value which can be random.

2. Pick a learning rate α i.e. a shrinkage factor to descend along the tangent to the curve.

3. Update X according to: $$X = X - \alpha \times \frac{\partial J}{\partial X}$$

Optimization with Gradient Descent

Algorithm

4. Repeat 2 and 3 for a certain number of iterations or 'epochs' hopefully converging at the end.

Note that step 3 should be applied on all the components of vector **X** simultaneously.

Let's practice!

Linear and Logistic Regression

- Now that we know how to optimize functions, it's time to put our knowledge into practice!
- Linear Regression is one of the most simple models used for regression, that is predicting continuous valued targets.
- Although it has the term 'regression' in it, Logistic Regression is a linear classification model. It is used for predicting the discrete labels of a target.

Linear Regression

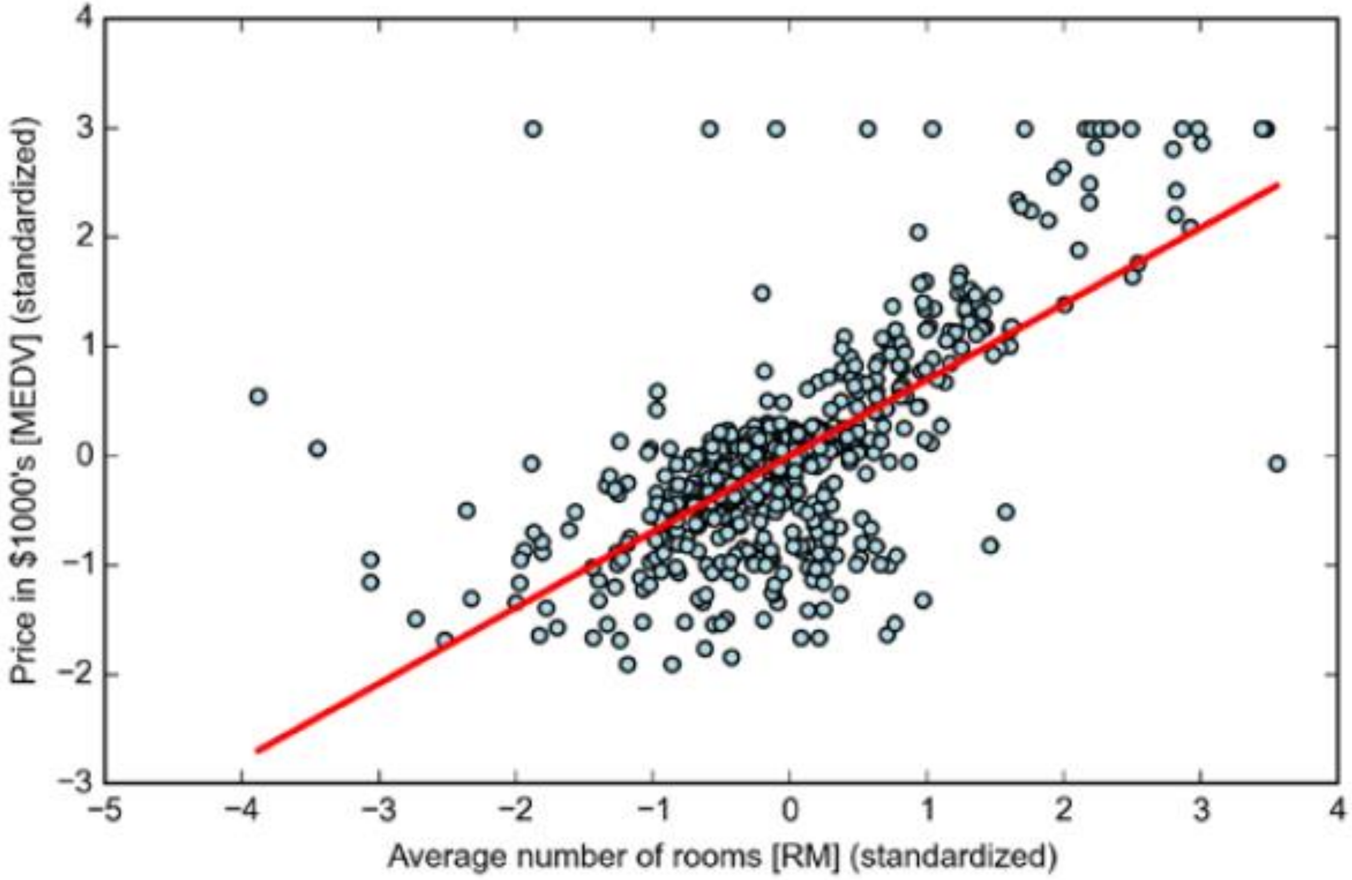
Given the Boston housing dataset (UCI machine learning repository).

X (Average Number of Rooms)

Y (Median House Price)

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0	0.00632	18	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

Linear Regression



Linear Regression

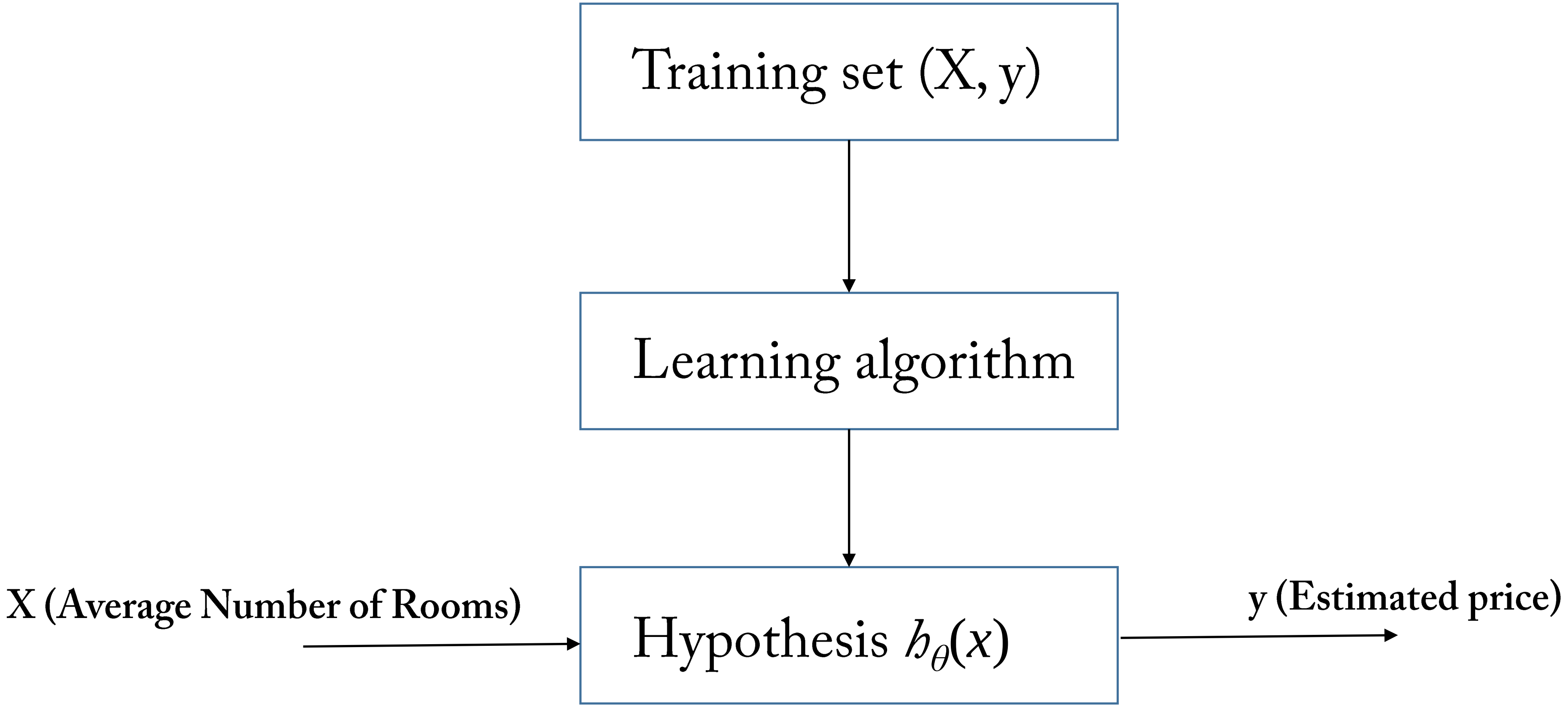
Model

In 1D the linear regression model $h_\theta(x)$ (also known as the hypothesis) has the following form:

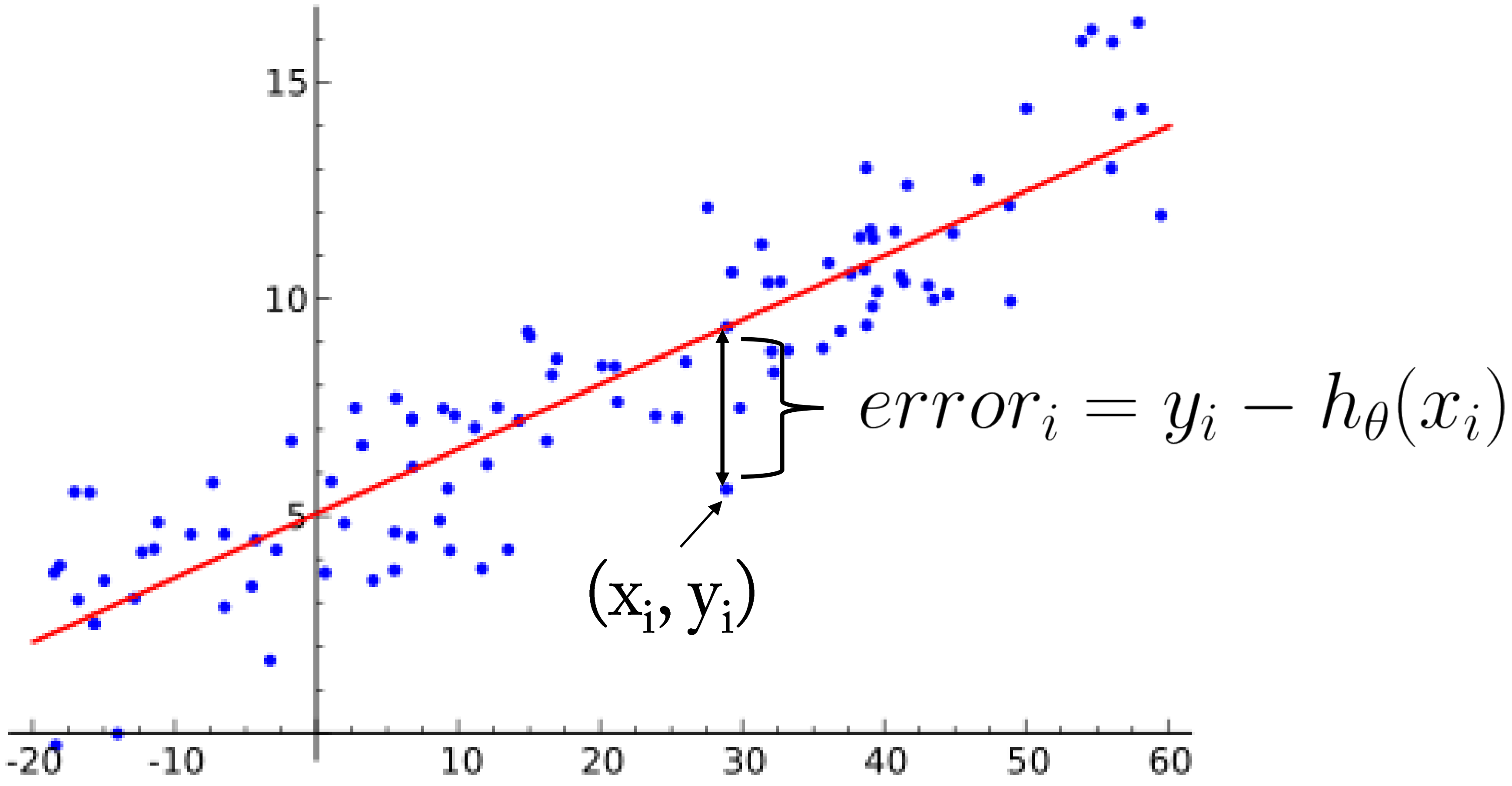
$$h_\theta(x) = \underbrace{\theta_0}_{intercept} + \underbrace{\theta_1}_{slope} \times x$$

The task is to determine the θ_i 's with $\theta = (\theta_0, \theta_1)$

Linear Regression



Cost function



Cost function for Linear Regression

A cost function measures the agreement between the true labels and the predicted ones. Given a dataset of N observations.

$$((x_1, y_1), ..., (x_i, y_i), ..., (x_N, y_N))$$

The cost function for linear regression is the mean squared error:

$$J(\theta) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - h_\theta(x_i))^2$$

Gradient Descent for Linear Regression

In order to find the optimal hypothesis that fits the data, we need to minimize the cost function with respect to θ.

$$min_{\theta_0,\theta_1} J(\theta_0, \theta_1)$$

This minimization is performed using gradient descent.

Gradient Descent for Linear Regression

Algorithm

Repeat until convergence:

$$\theta_j := \theta_j - \underbrace{\alpha}_{\text{learning rate}} \times \underbrace{\frac{\partial J(\theta_j)}{\partial \theta_j}}_{\text{partial derivative along } \theta_j}$$

Gradient Descent for Linear Regression

For linear regression, we can compute the derivative of the cost function to obtain the update we have to make:

$$\theta_j := \theta_j - \frac{\alpha}{N} \sum_{i=1}^{N} (h_\theta(x_i) - y_i) \times x_i^j$$

with $j = 0, 1,\ x_i^0 = 1$ and $x_i^1 = x$

Let's practice!

Logistic Regression

Model

Logistic Regression aims at predicting the probabilities of labels. For a binary classification problems 0 denotes the negative class and 1 denotes the positive class. The hypothesis is the probability of predicting class 1 given a feature vector x.

$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 x_0 + \cdots + \theta_D x_D)}}$$

Cost function for Logistic Regression

Given a dataset of N observations.

$$((x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N))$$

with y_i = 0 or 1

The cost function for logistic regression is given by:

$$J(\theta) = -\sum y_i \log(h_\theta(x_i)) + (1 - y_i)\log(1 - h_\theta(x_i))$$

Let’s practice!

Overfitting

- Overfitting is fitting the training data more than is warranted.
- When you overfit, your model performs well on the training data but fails to generalize on unseen data.
- A model is said to suffer from **high bias** if the hypothesis is **not complex enough** to capture the pattern of the signal in the data.
- On the other hand a **high variance** model is so complex that, not only it fits the signal, but also fits the **noise** present in the training set!

Overfitting

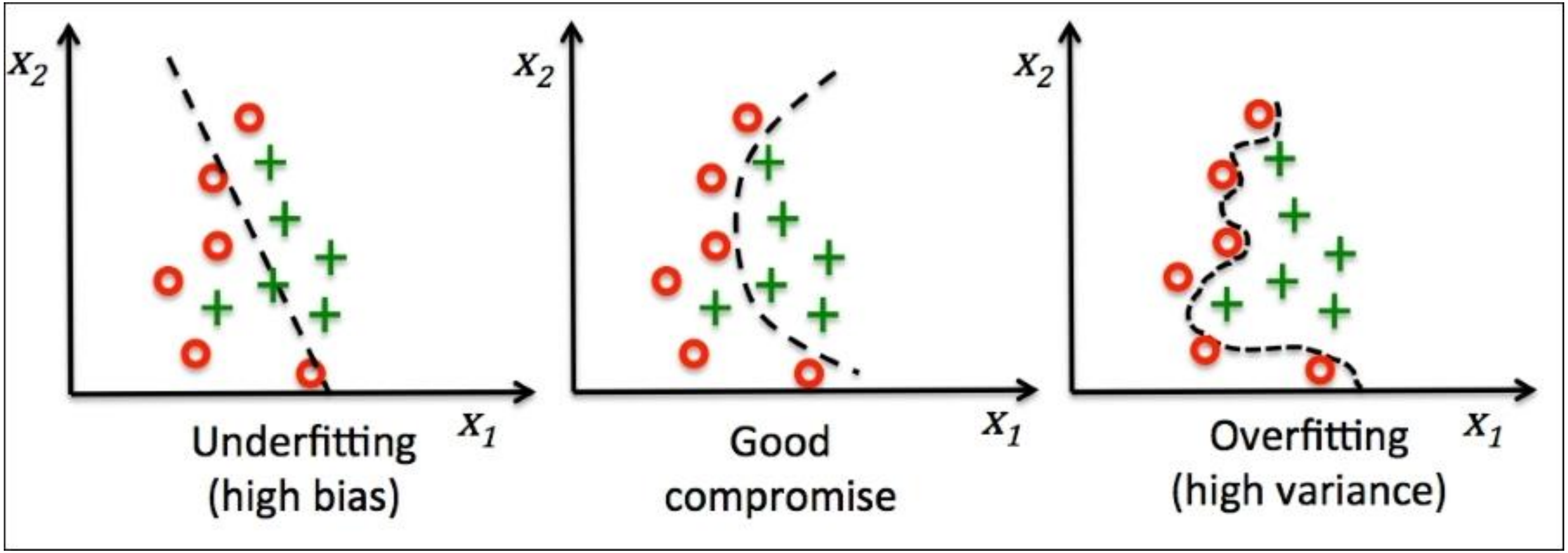
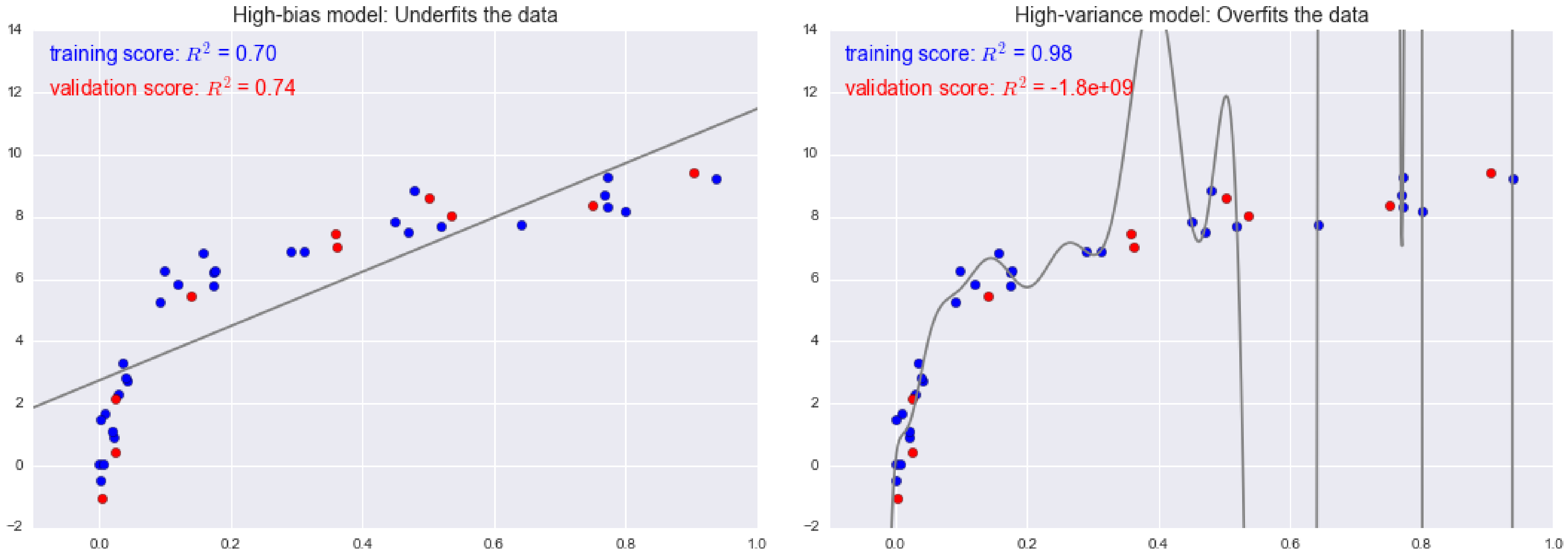


Figure taken from Python Machine Learning, Sebastian Raschka, Packt 2015

Overfitting



Combatting Overfitting

- One way to combat overfitting is to add a regularization term.
- Regularization not only reduces overfitting but also handles collinearity and filters out noise from data.
- In *L2* regularization, we add the following term to the cost function:

$$\frac{\lambda}{2}||\theta||^2 = \frac{\lambda}{2}\sum_{j=1}^{N}\theta_j^2$$

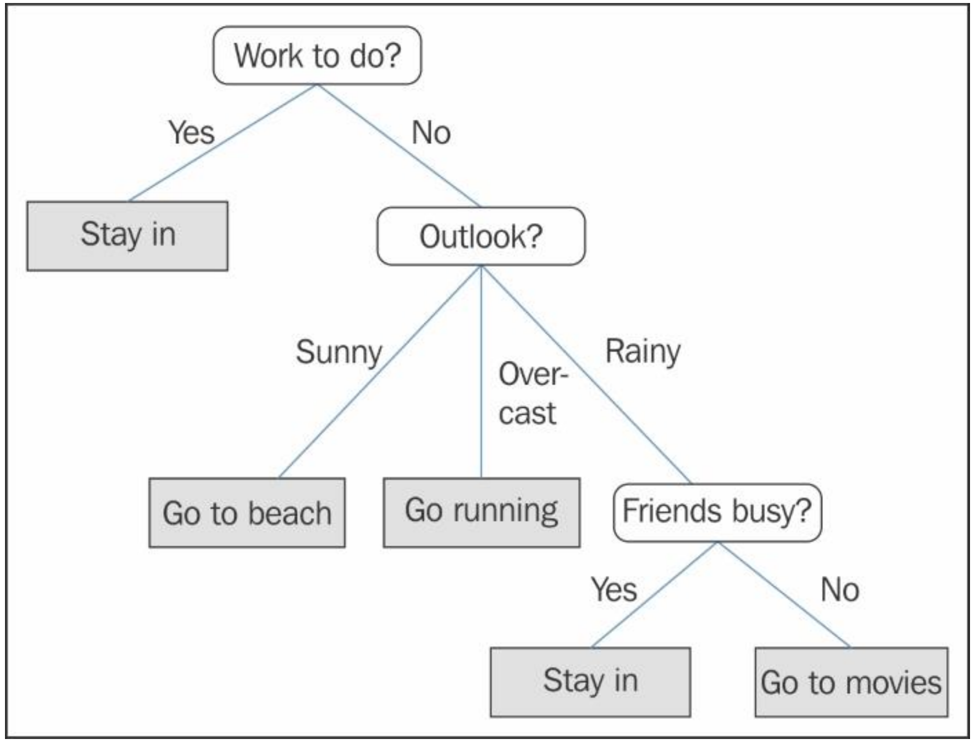
Combatting Overfitting

- Regularization has the effect of shrinking the weights of some features.

Let's practice!

Decision Trees

Make decision based on a series of questions.



Decision Trees

Decision Trees learn by maximizing information gain. At each split, a feature and a split point are selected such that they maximize:

$$IG(D_p, f) = I(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} I(D_j)$$

Using the entropy as a measure of information in a node:

$$I_H(t) = -\sum_{i=1}^{c} p(i|t) \log_2 p(i|t)$$

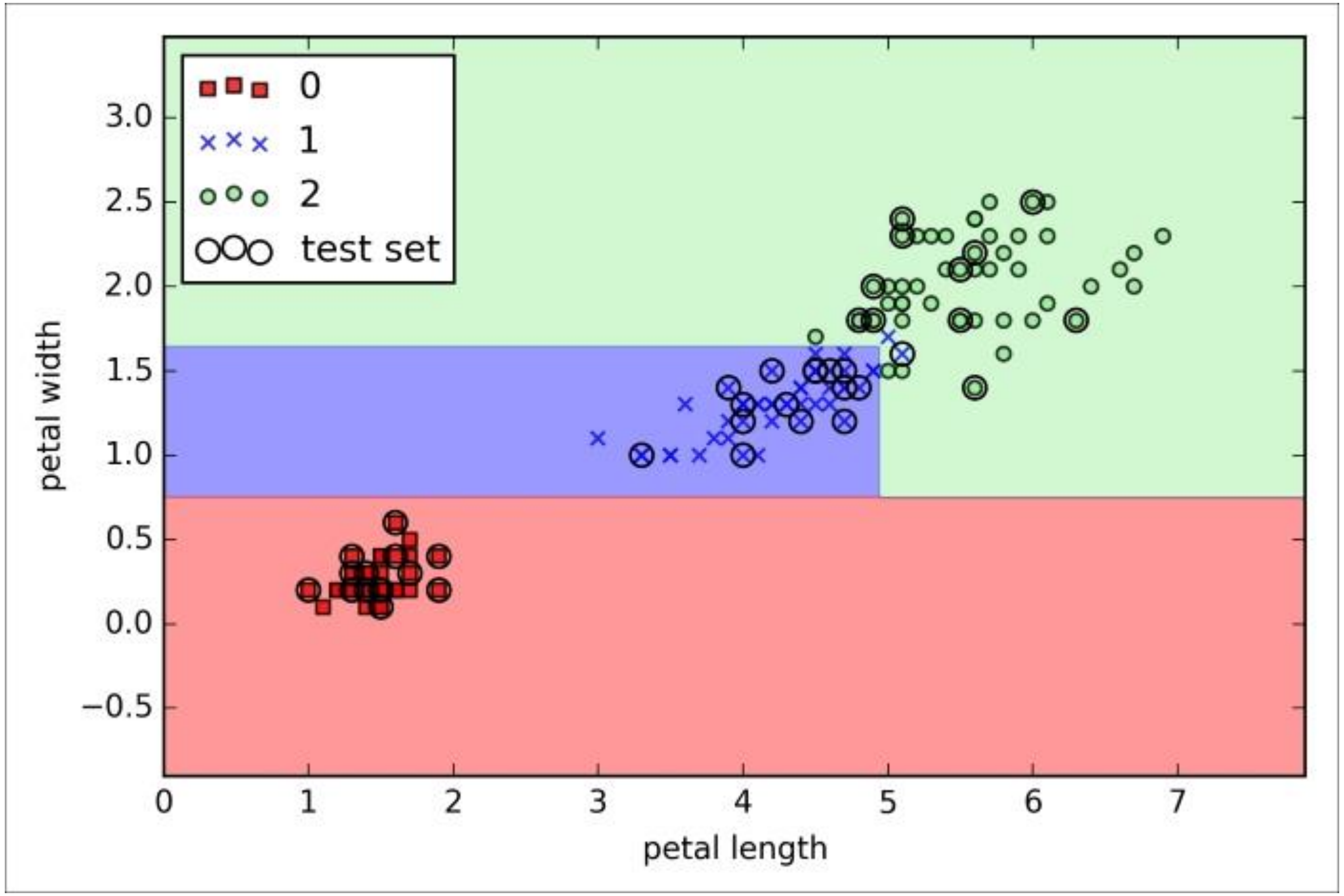
Decision Trees

- Decision trees can build complex decision boundaries by dividing the feature space into rectangles.

- However, we have to be careful since the deeper the decision tree, the more complex the decision boundary becomes, which can easily result in overfitting.

- To reduce overfitting, we should **prune** the tree. That is we impose a maximal depth on its growth.

Decision Trees

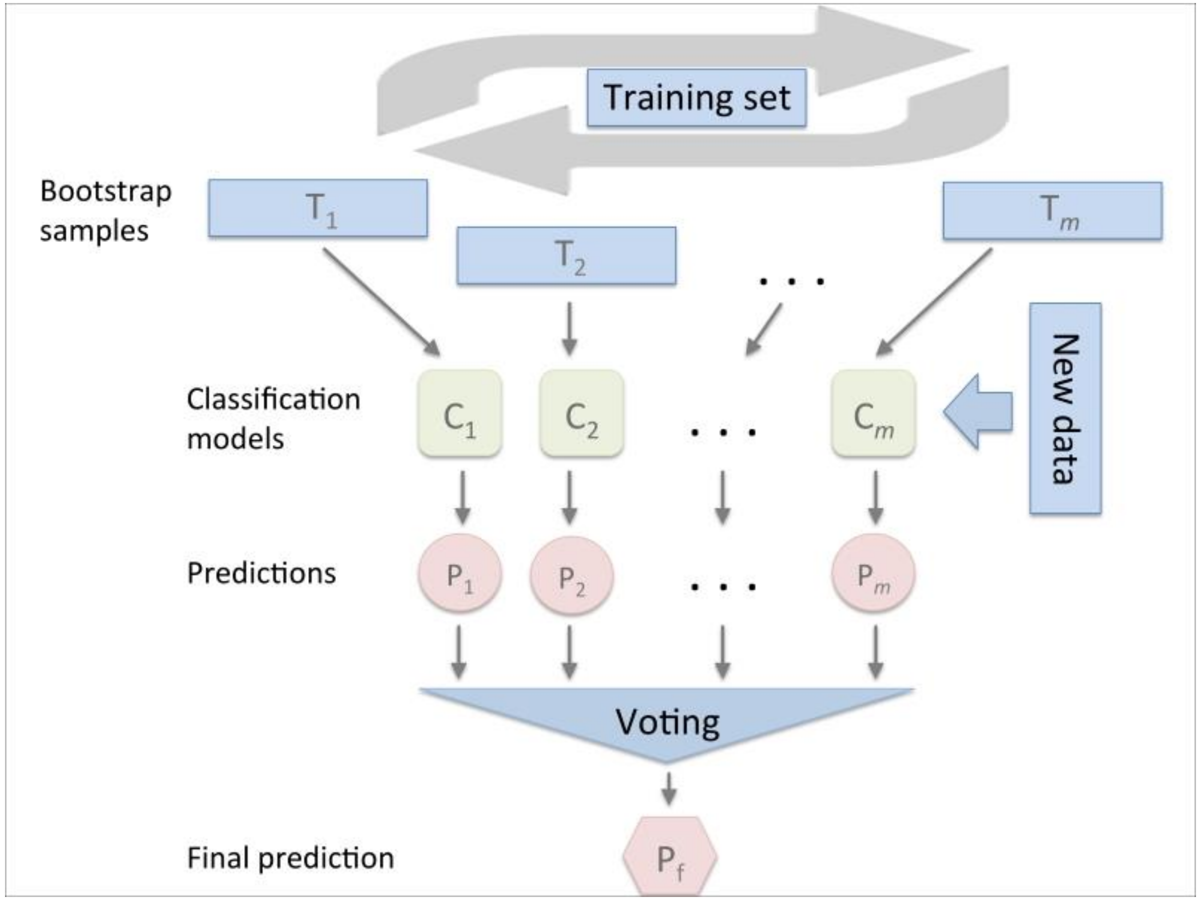


Let’s practice!

Ensemble Methods

- In ensemble methods, we combine the predictions of different classifiers via a meta-classifier.
- Think of it this way: an individual expert may give the wrong prediction.
- On the other hand, when the judgements of many experts are combined strategically, the prediction is more robust and less prone to error.

Ensemble Methods: Bagging



Ensemble Methods: Random Forests

Algorithm

1. Draw a random bootstrap sample of size *n*.

2. Grow a decision tree from the sample. At each node:
   a. Randomly select *d* features without replacement.
   b. Split the node using the feature that provides the best split.

3. Repeat 1 to *k* times.

4. Aggregate results.

Let's practice!

Evaluating Classification Models

- Consider a binary classification problem where there is a class imbalance; that is the negative class has a much higher number of instances than the positive class. Ex: 90 % negative.

- If your evaluation metric is accuracy, then predicting the negative class blindly would achieve a score of 90 %!

- There are many methods used to deal with class imbalance. We will discuss a method in which we use evaluation metrics other than accuracy.

Evaluating Classification Models

		Predicted class	
		P	N
Actual Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$ERR = \frac{FP + FN}{FP + FN + TP + TN}$$

$$ACC = \frac{TP + TN}{FP + FN + TP + TN} = 1 - ERR$$

$$FPR = \frac{FP}{N} = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$$

Evaluating Classification Models

- Precision is a metric that measures the rate of true positive among all predicted positives:

$$PRE = \frac{TP}{TP + FP}$$

- Recall is another name of the TPR: $REC = TPR = \frac{TP}{P} = \frac{TP}{FN + TP}$

- In practice, we use the f1-score defined as follows:

$$F1 = 2\frac{PRE \times REC}{PRE + REC}$$

Cross-validation

- Evaluating you model on the training set gives a biased estimate of your model's skill.
- This is because you are evaluating the model on data that it has already seen.
- Cross-validation is a technique used to obtain a less biased estimate of the skill achieved by your model.

K-fold Cross-validation

- For small to medium sized datasets, the gold standard is K-fold cross validation.
- Split the training data into K folds.
- Train the model K times to obtain a list of K scores.
- At each iteration, train the model on (K-1) folds and evaluate it on an individual fold unseen by in training.

K-fold Cross-validation

The average score of a model obtained using K-fold CV estimates its generalization score.

Model Selection

- After training many models using K-fold cross validation, the winning model should be declared as the one that achieves the highest mean CV score.

- Note that to report a final **unbiased** estimate of your model's skill, you should **evaluate** the model on a **test set** that it has never seen. This step is **crucial**!

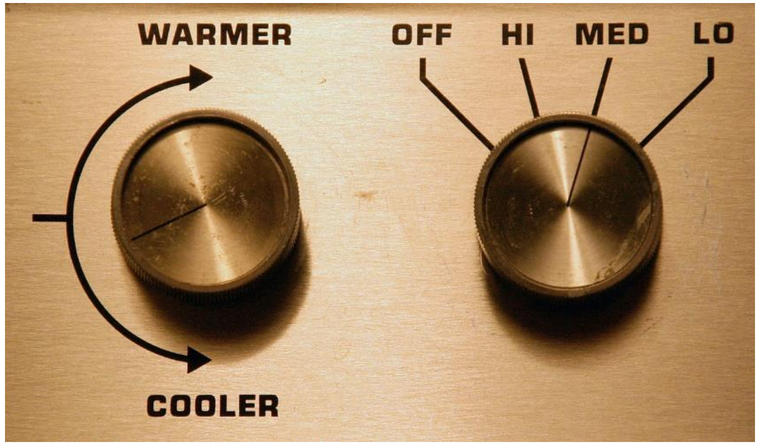
- Do not compromise the last step! otherwise you are **snooping** the data.

Hyperparameter Tuning

- The parameter of your model that are not learned through training are called hyperparameters.
- A model's hyperparameters should be tuned so that the model achieves the best results.
- Examples of hyperparameters include: the regularization term in linear regression, tree depth and number of estimators in random forests.

Hyperparameter Tuning

- Hyperparameter tuning is performed through grid search.
- Think of it as adjusting the control knobs of a thermostat to obtain the 'best' temperature.



Thank you for your attention!

&

Let's practice!