Text Classification

Luciano Barbosa

Centro de Informática UFPE

Universidade Federal de Pernambuco

Example of Text Classification: Spam

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

==========

Click Below to order:
http://www.wholesaledaily.com/sales/nmd.htm

==========

Supervised Learning

- Goal: to infer a function from examples to predict classes of new examples
- Two phases:
  - Training: learn the function from examples
  - Execution: use the function to predict the class of a given instance

Supervised Model

- Training set: instances and labels
- Instance represented by its feature vector
- Learn function f(x)=y that best predicts the value of y given x
- For categorical y -> classification
- For numerical y -> regression

	viagra	learning	the	dating	nigeria	*spam*?
$\vec{x}_1 = ($	1	0	1	0	0 $)$	$y_1 = 1$
$\vec{x}_2 = ($	0	1	1	0	0 $)$	$y_2 = -1$
$\vec{x}_3 = ($	0	0	0	0	1 $)$	$y_3 = 1$

Spam Classification

- Training instances

	viagra	learning	the	dating	nigeria	*spam*?
$\vec{x}_1 = ($	1	0	1	0	0 $)$	$y_1 = 1$
$\vec{x}_2 = ($	0	1	1	0	0 $)$	$y_2 = -1$
$\vec{x}_3 = ($	0	0	0	0	1 $)$	$y_3 = 1$

- Features
  - Words: viagra, learning, the, dating, nigeria
  - Occurrence: 1 or 0
- Class y: spam (1) or non-spam (-1)

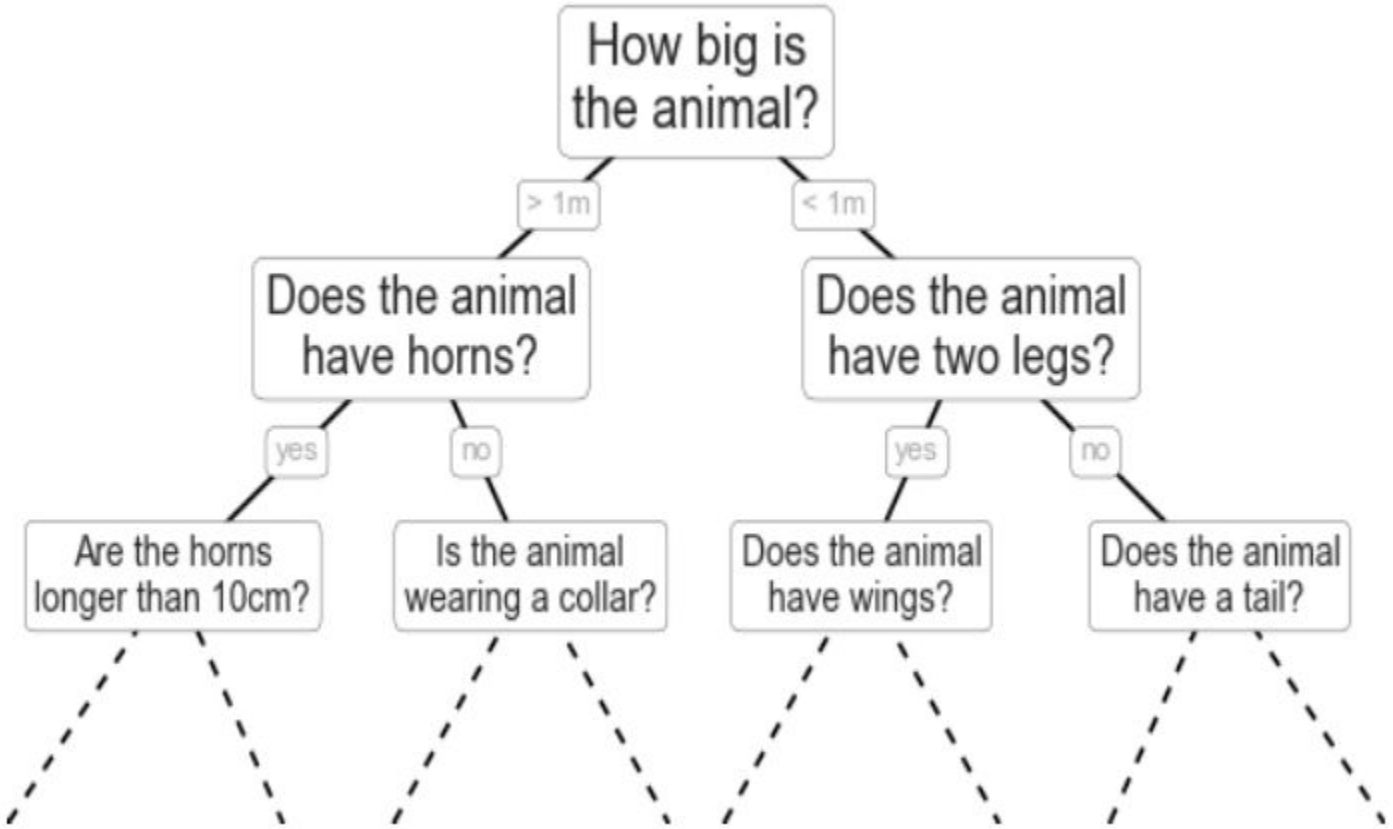
Features

- Great importance in the classification result
- Important: high correlation with the classification output
  - Ex1: rain forecast: temperature, humidity
  - Ex2: sentiment analysis: words with polarity (negative/positive)
- Text classifiers can use any type of feature: words, punctuation, capitalization, etc.

Random Forest

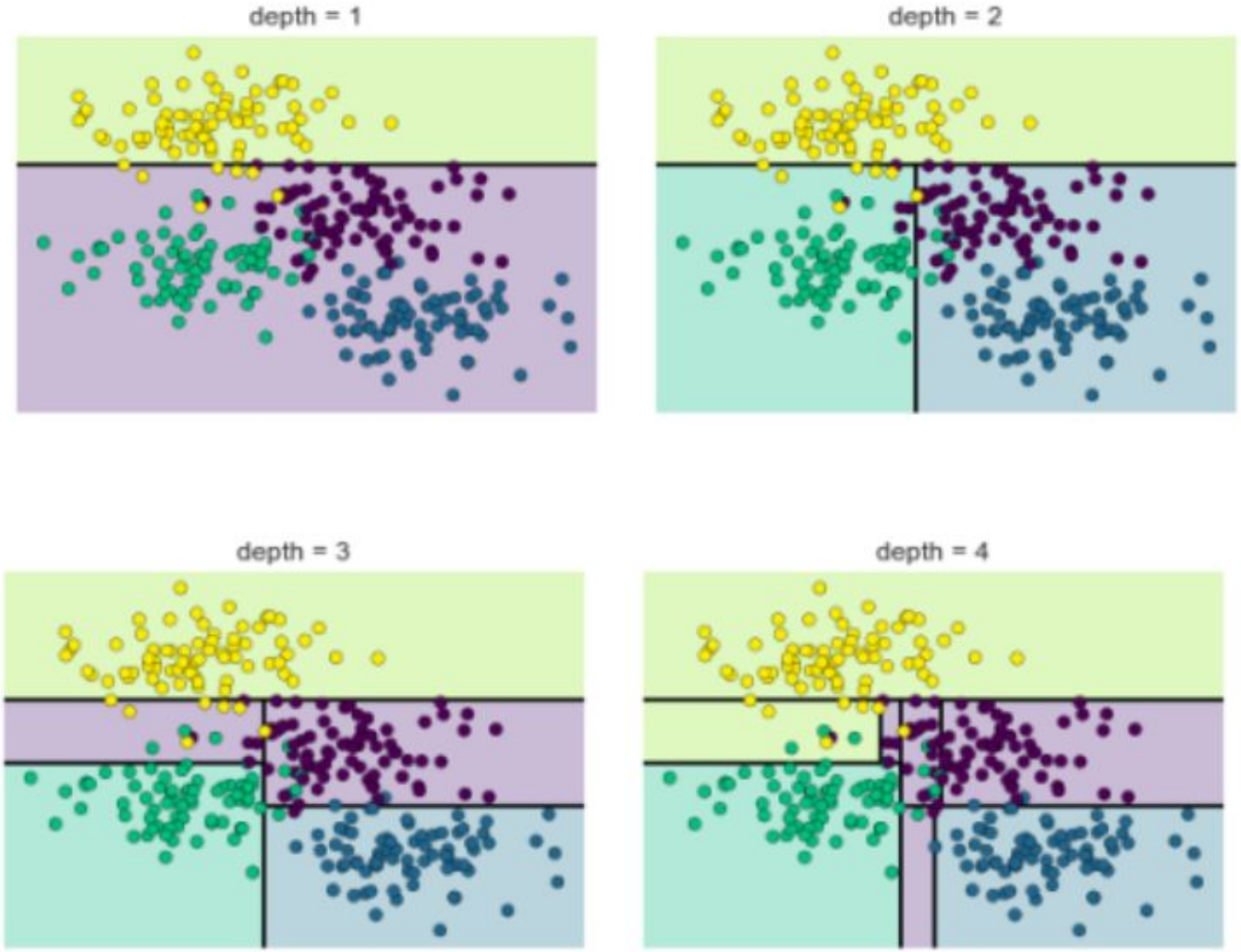
- Decision tree ensemble
- The sum is better than the parts: majority vote is better than models individually

Example of Decision Tree

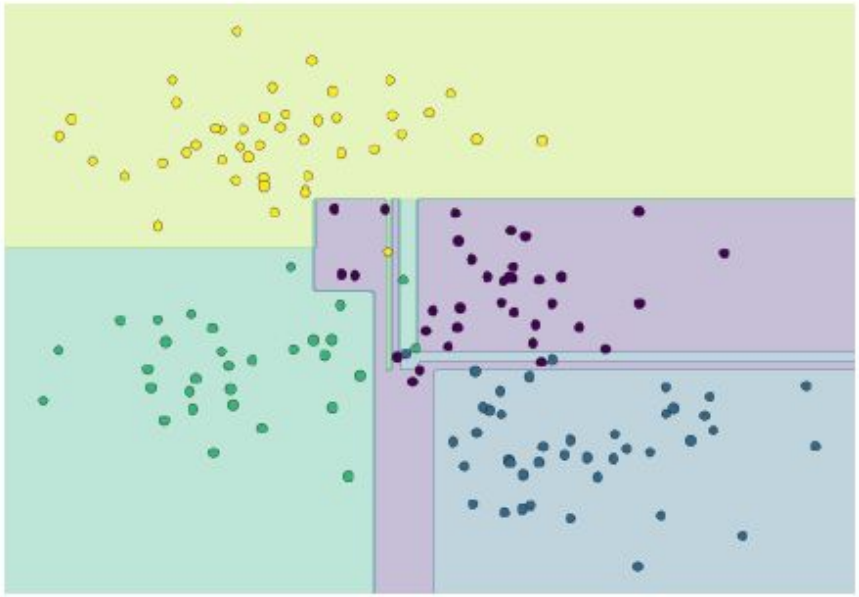


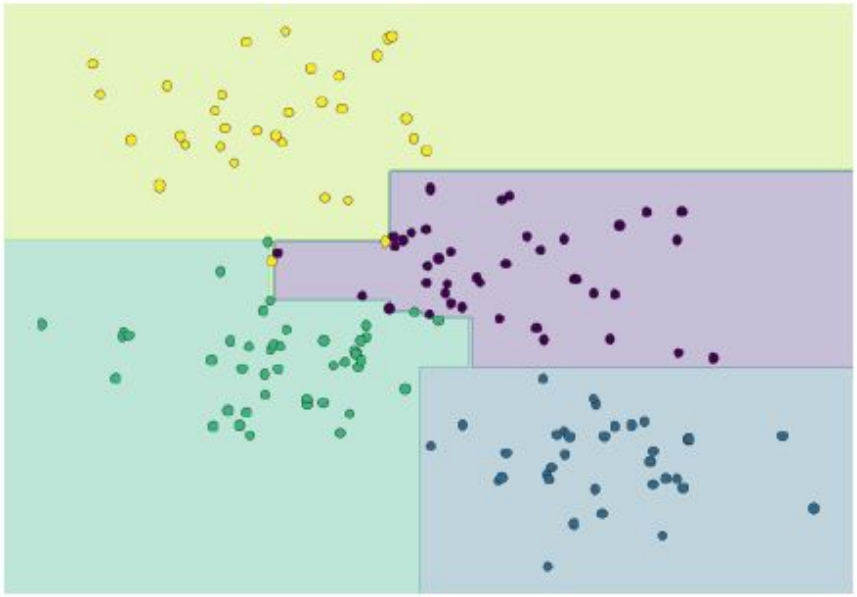
Example of Decision Tree

Splitting the Data



Issue: Overfitting



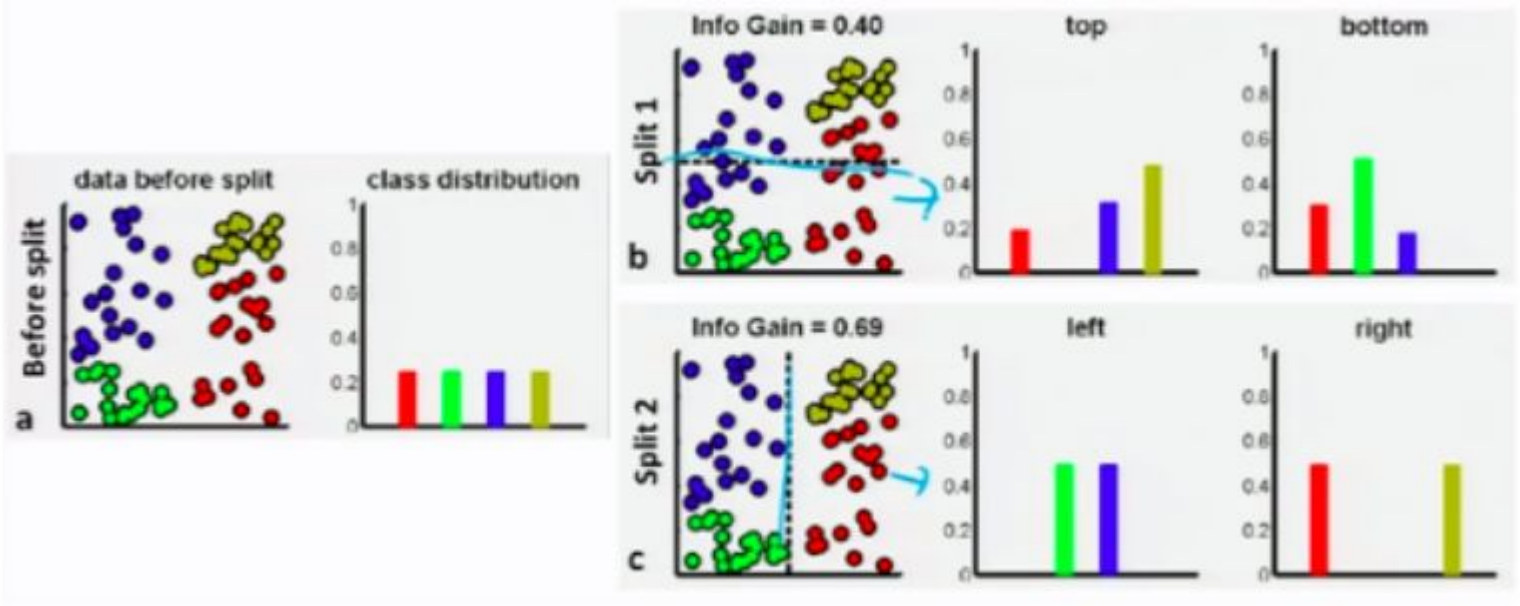


Random Forest

- Decision tree ensemble
- The sum is better than the parts: majority vote is better than models individually
- Random ensemble of decision trees
- Multiple estimators combined to avoid overfitting
- Bagging:
  - Ensemble of parallel estimators that overfit
  - Calculates average of predictions (regression) or majority of votes (ranking)
  - Several overfitting models combined to avoid overfitting
- Does not assume Gaussian distribution, linear relationship, etc.

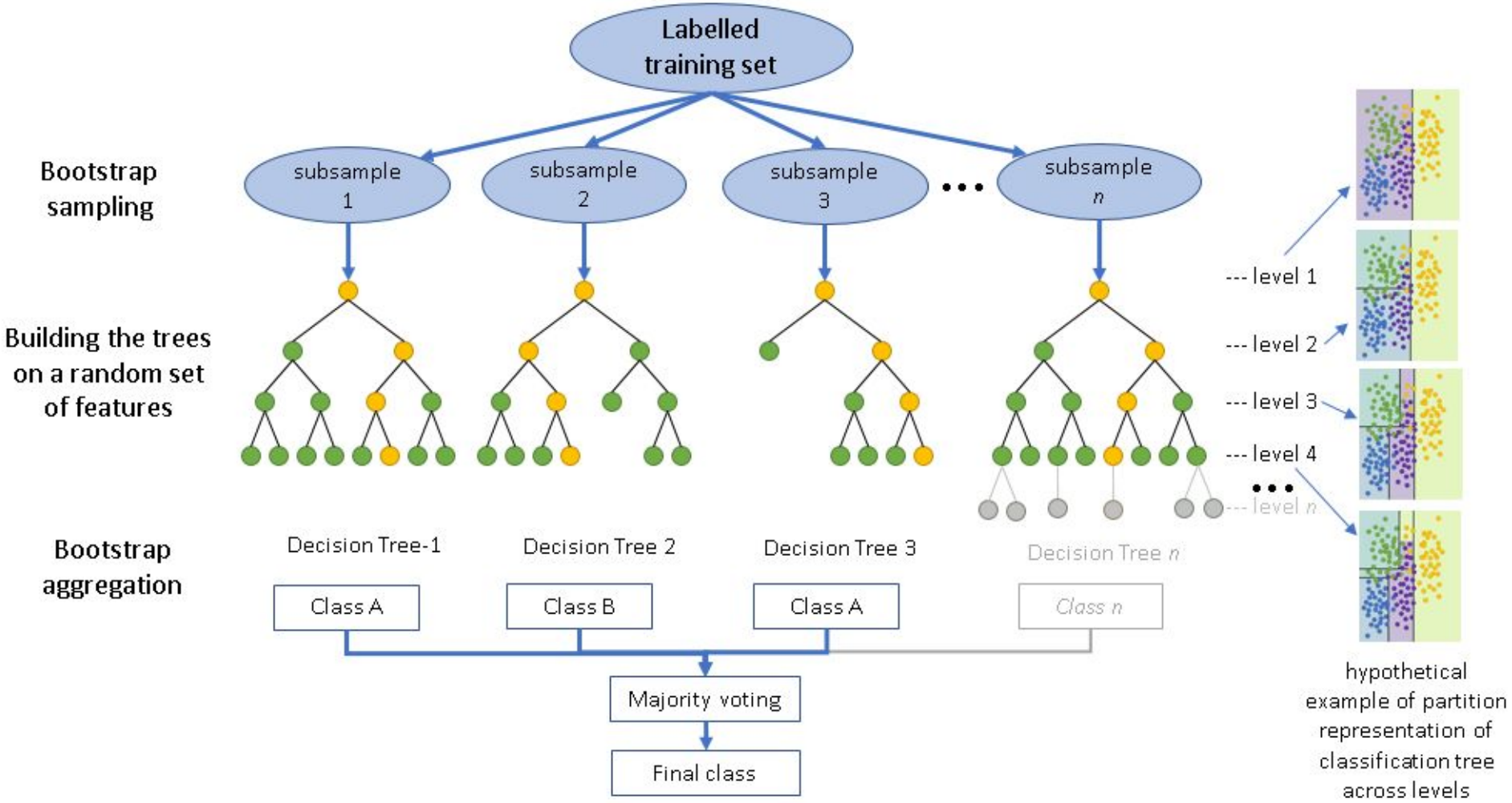
Random Forest: Algorithm

1. Select a training sample
2. For each node, select a sample m from the n features and of these m, the one with the highest information gain or gini is selected
3. Until there is no more data to split
4. Repeat steps 1 to 3 for each tree



Information Gain

Random Forest



Source: https://catalyst.earth/catalyst-system-files/help/concepts/focus_c/oa_classif_intro_rt.html

Benefits

- Training can run in parallel
- Obtained excellent results in several types of datasets
- Fast execution
- Can handle thousands of features
- Shows the importance of features in classification

Classification Evaluation Metrics

- Precision
- Recall
- F1 (F-measure)
- Accuracy

	in the class	not in the class
predicted to be in the class	true positives (TP)	false positives (FP)
predicted to not be in the class	false negatives (FN)	true negatives (TN)

$$\text{precision}: P = TP/(TP + FP)$$

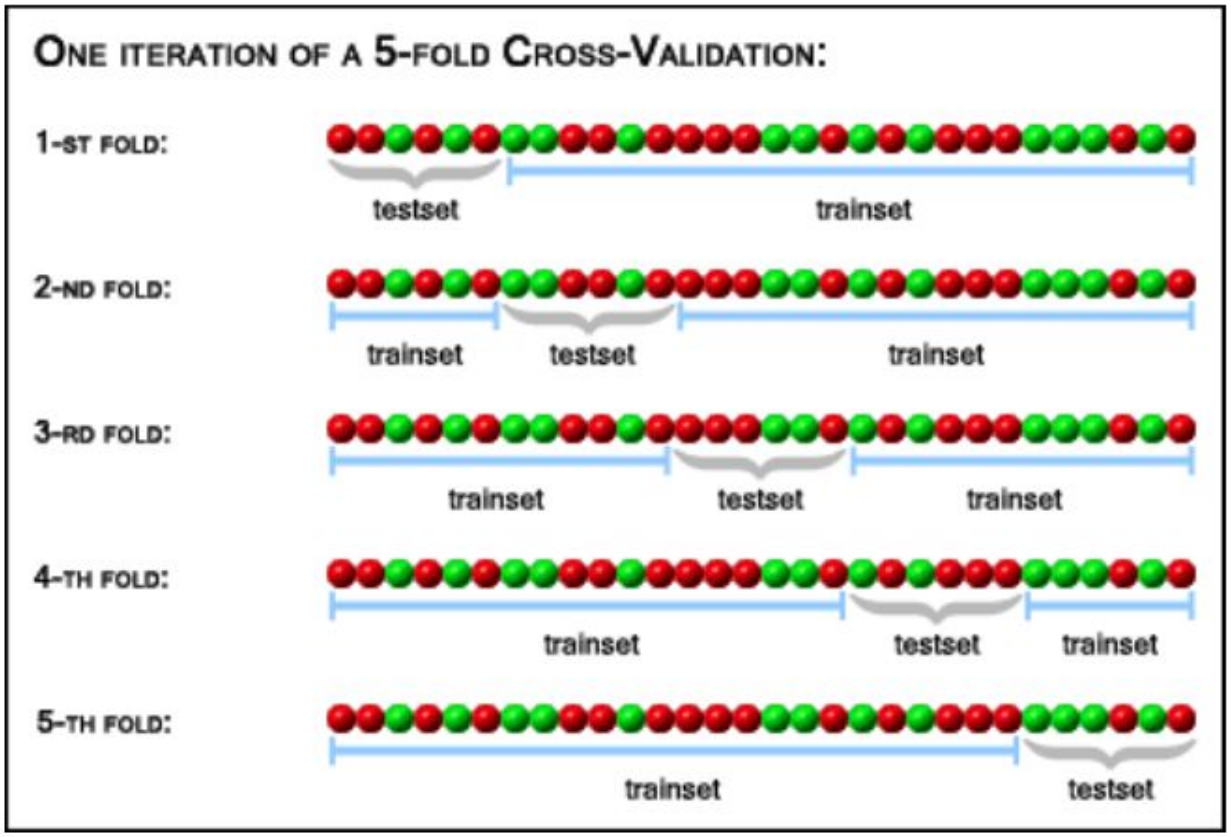
$$\text{recall}: R = TP/(TP + FN)$$

$$F_1 = \frac{1}{\frac{1}{2}\frac{1}{P} + \frac{1}{2}\frac{1}{R}} = \frac{2PR}{P + R}$$

Accuracy = (TP+TN)/(TP+TN + FN + FP)

Evaluation Protocol

- Holdout set: not used for training
- Training/validation/test
- Cross-validation



Procedure for Classifier Creation

- Define and process the features
- Build labeled data
  - Pairs (x,y), where x is a feature vector and y is the label
- Define evaluation protocol
  - Holdout set (training/validation/test)
  - Metrics
- Train the model using an ML software
- Perform model selection: hyper-parameter selection
- Model diagnostics

Model Selection

- Use set validation/cross-validation
- Seek better hyper-parameter values
  - ML Algorithm
  - Parameter space
  - Method for searching candidate values
  - holdout set
  - Evaluation metrics
- Hyper-parameter selection strategies:
  - Grid search
  - Automl: TPOT, SMAC, auto-sklearn, optuna

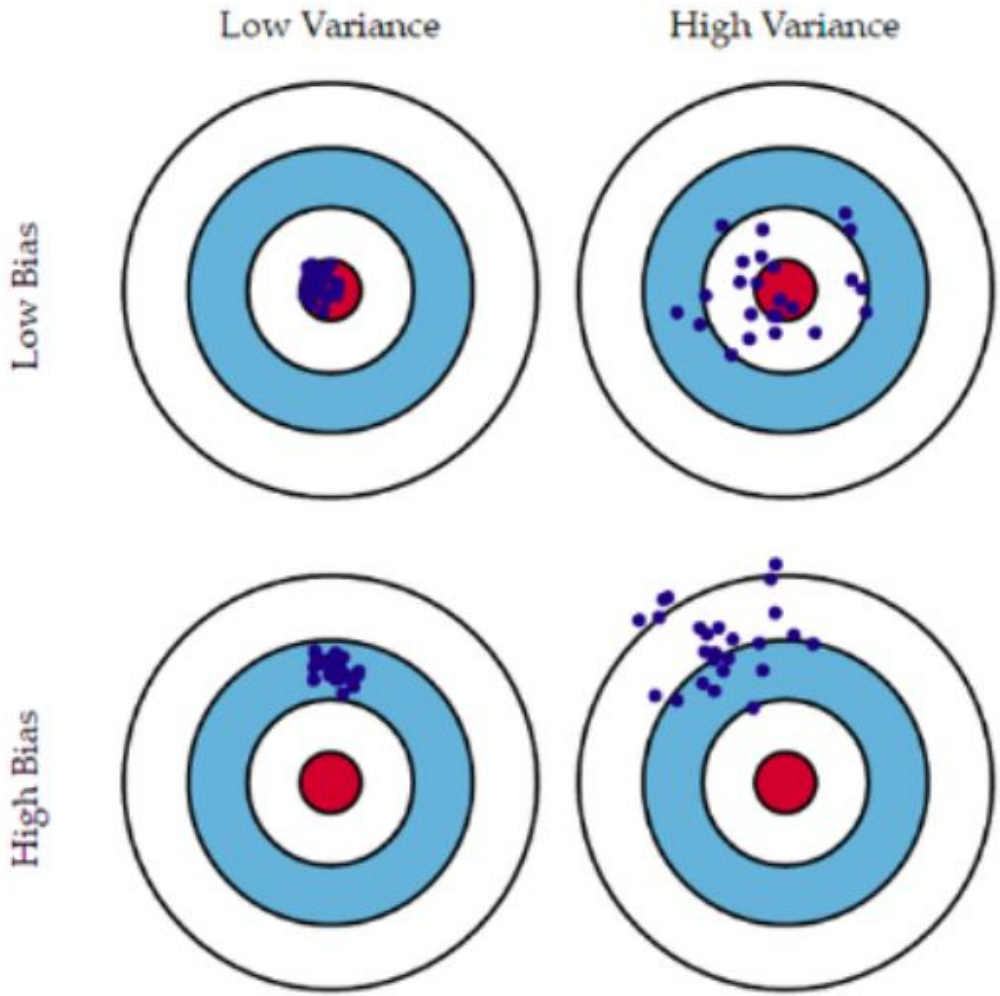
Model Diagnostics

- How to improve the classifier?
  - More/less data
  - More/less features
  - More/less complex models
- Example:
  - Target: 5% error
  - Training error: 15% (bias: 15-5= 10)
  - Test error: 16% (variance: 16-15 = 1)
  - Need to improve performance on training set

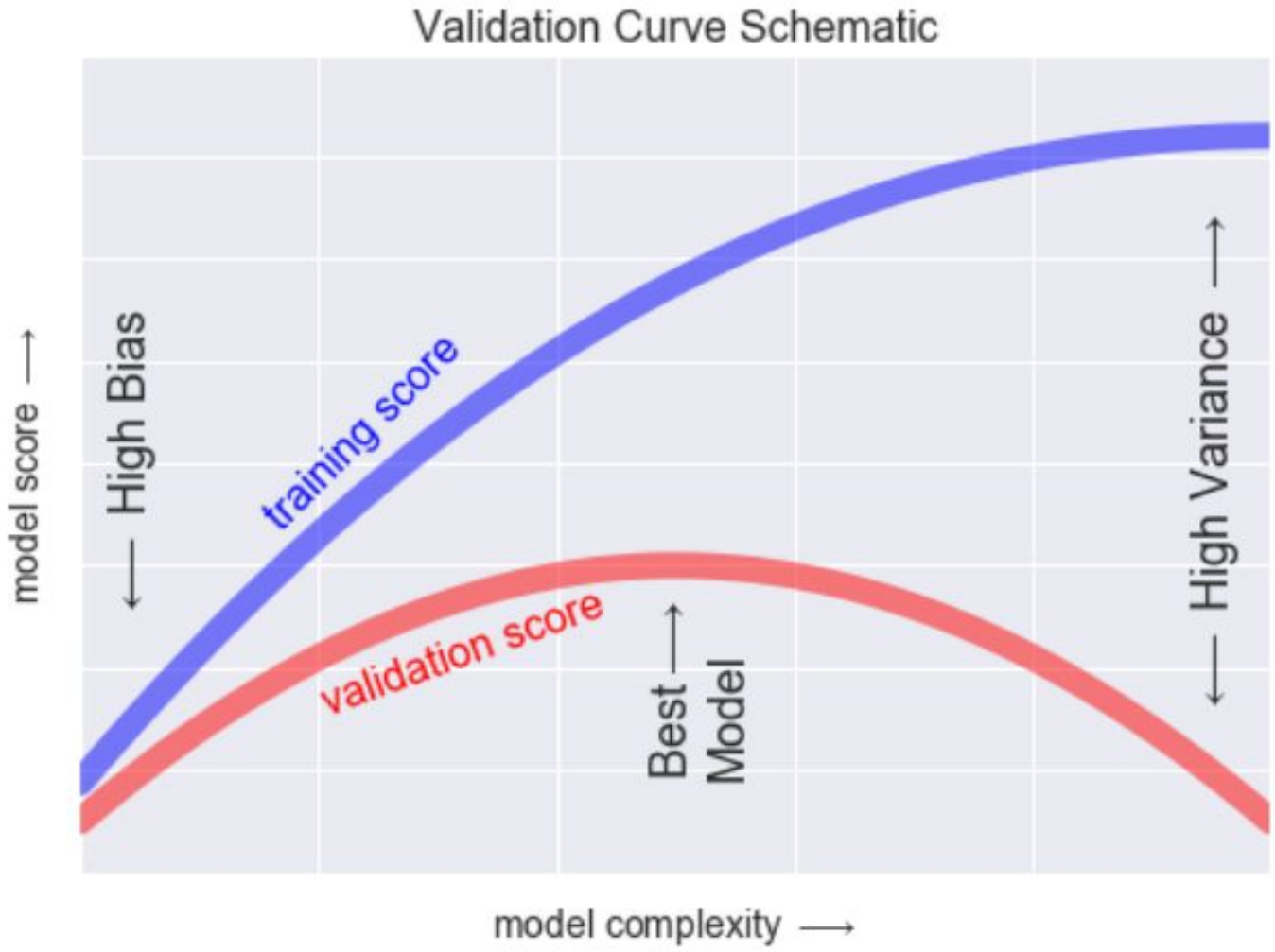
Model Diagnostics: Two Main Concepts

- Bias:
  - Performance on the training set
  - Depends on target value
- Variance: difference in performance between training and testing

Model Diagnostics

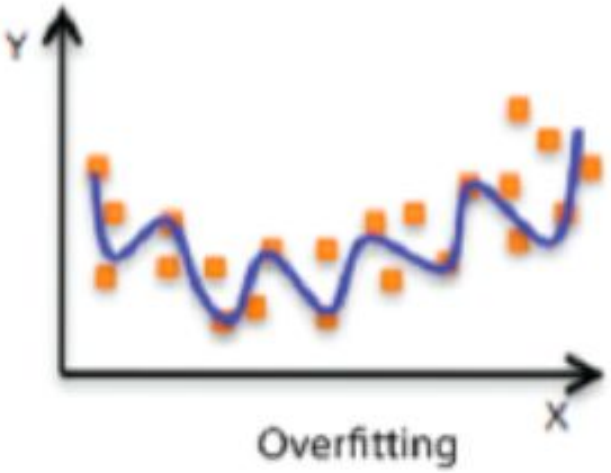


Model Diagnostics



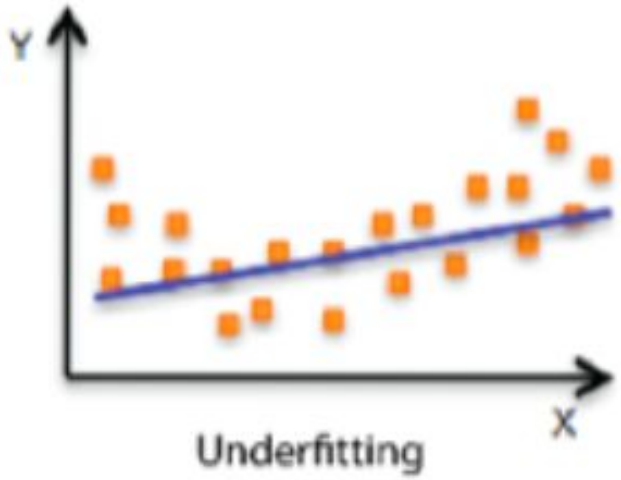
Overfitting

- Good performance on the training set
- Problem generalizing
- Low bias and high variance
- Example
  - Training error: 1%
  - Test error: 11%



Underfitting

- Model does not model the training set well
- High bias and low variance
- Example
  - Error in training: 15%
  - Test error: 16%

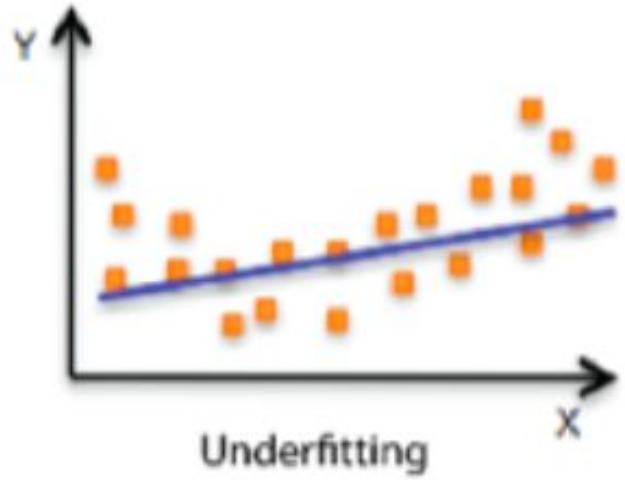


Other Scenarios

- Underfitting e overfitting
  - Example
    - Training error: 15%
    - Test error: 30%
- Ideal
  - Example
    - Training error: 0.5%
    - Test error: 1%

Dealing with Bias

- High bias (underfitting)

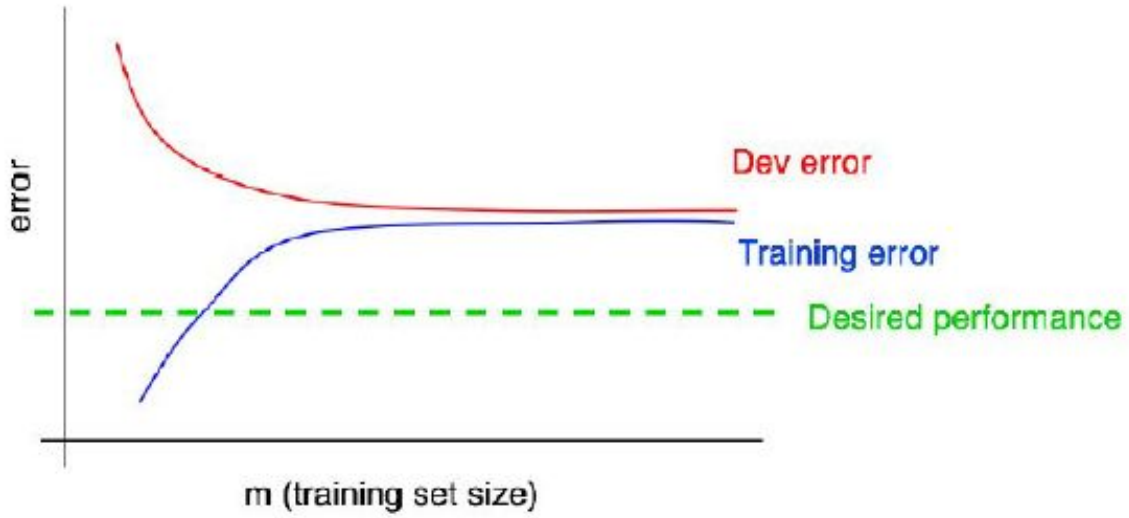


Dealing with Bias

- High bias (underfitting)

Dealing with Bias

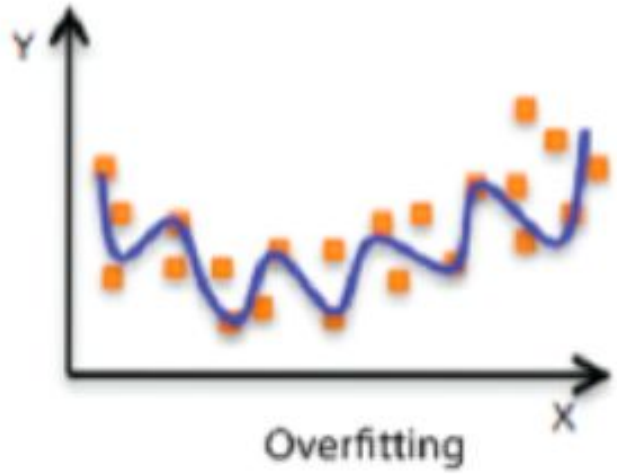
- High bias (underfitting)



  - Increase the complexity of the model
  - More features
  - It doesn't help to add more data to the training data

Dealing with Variance

- High variance (overfitting)



–

Dealing with Variance

- High variance (overfitting)

Dealing with Variance

- High variance (overfitting)

  - Add data to the training set
  - Feature selection
  - Less complex models