Machine Learning
Text Classification
Hands-on introduction

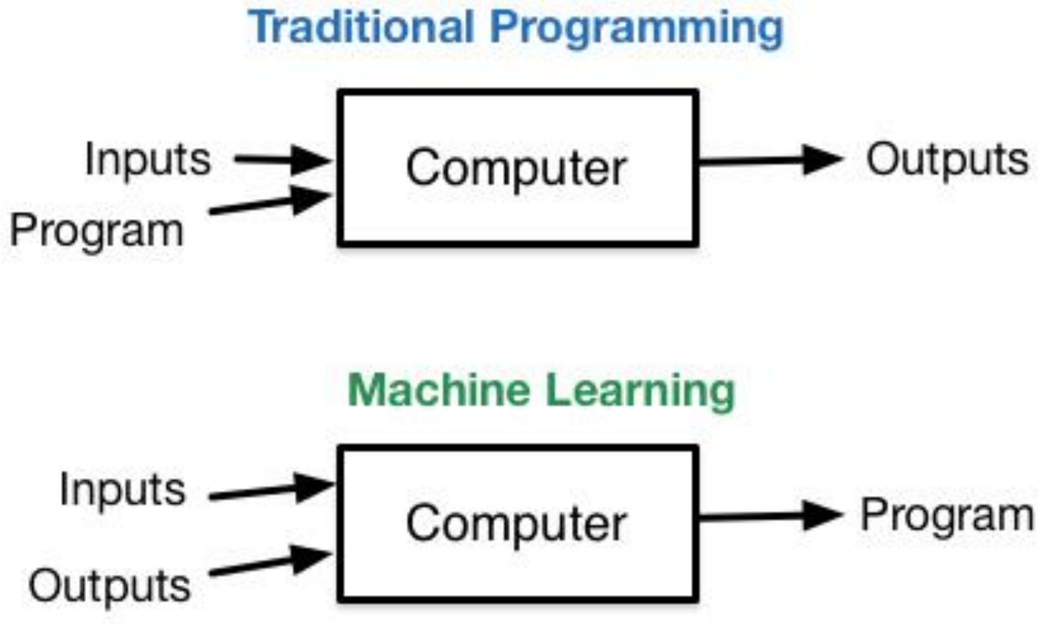
Noureddine Loukil

02 May 2018

What is machine learning ?

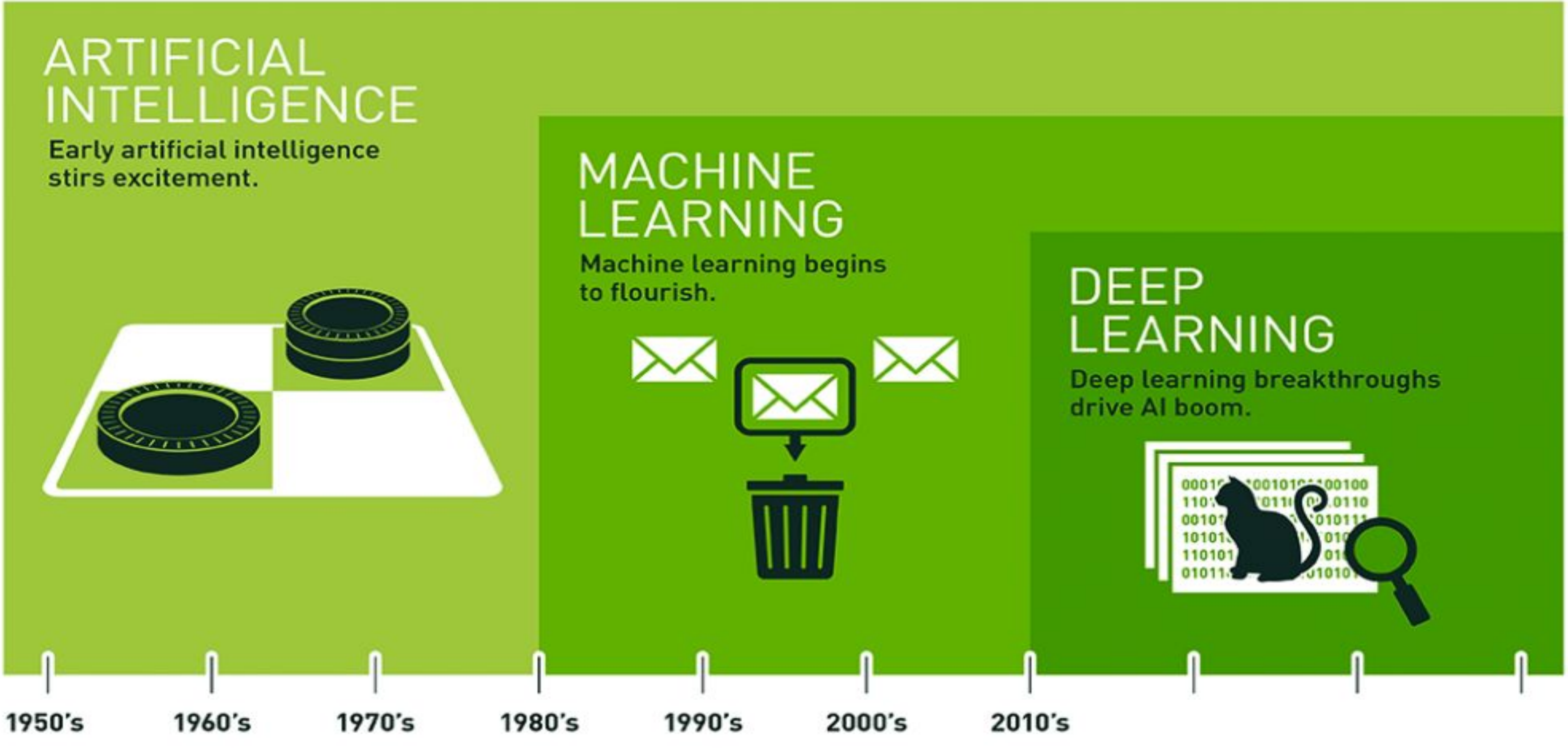
- ML is an application of artificial intelligence
- ML is about automatically learn and improve from data
- ML do learn without being explicitly programmed.

Traditional programming or Machine Learning ?



(Optional)

History of Machine Learning



Early artificial intelligence stirs excitement.

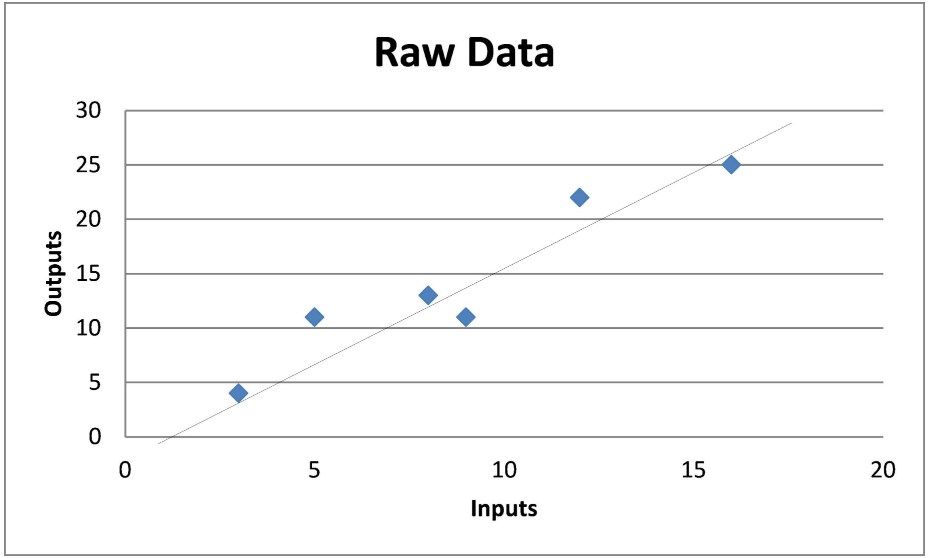
Machine learning begins to flourish.

Deep learning breakthroughs drive AI boom.

Since an early flush of optimism in the 1950s, smaller subsets of artificial intelligence – first machine learning, then deep learning, a subset of machine learning – have created ever larger disruptions.

Learning and prediction exercise

X Input	Y Output
3	4
5	11
8	13
9	11
13	22
17	25
10	?



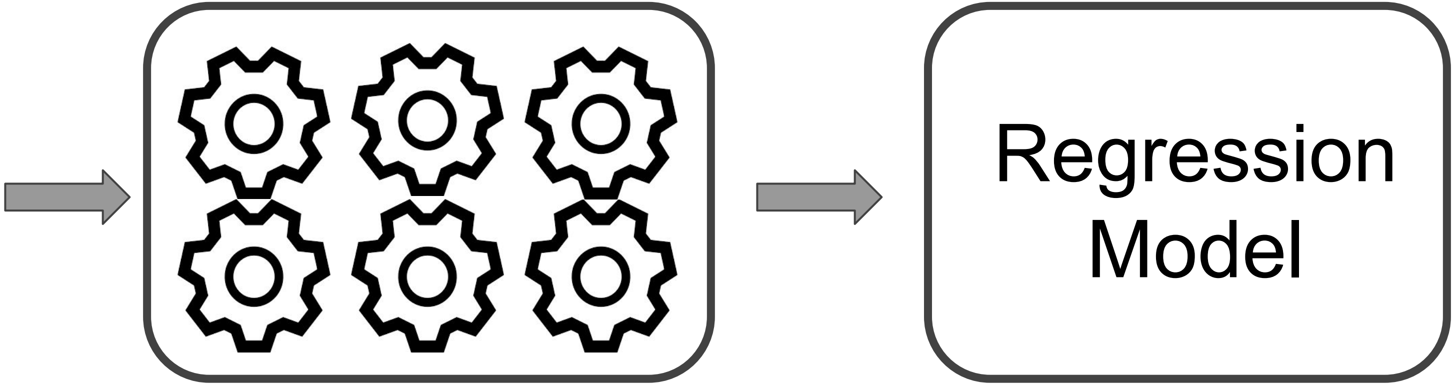
Y = a X + b

Linear Model

ML is trained to get a model

X Data	Y Output
3	4
5	11
8	13
9	11
13	22
17	25

Learning relations between data (input) and output

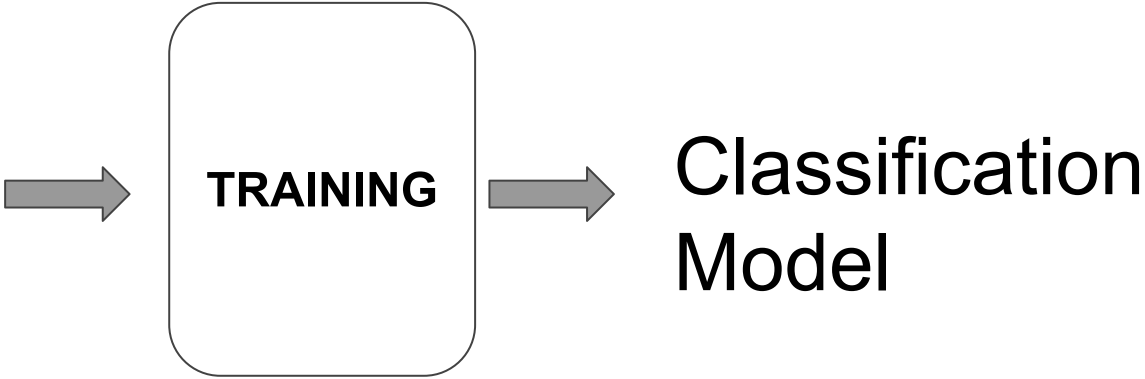


Regression Model

Classification of natural language text

Data	Y Output
Hello my friend	+
Damn the machine	-
Oh my god	-
Thanks a lot	+
That was a shame	-
I am very angry	?

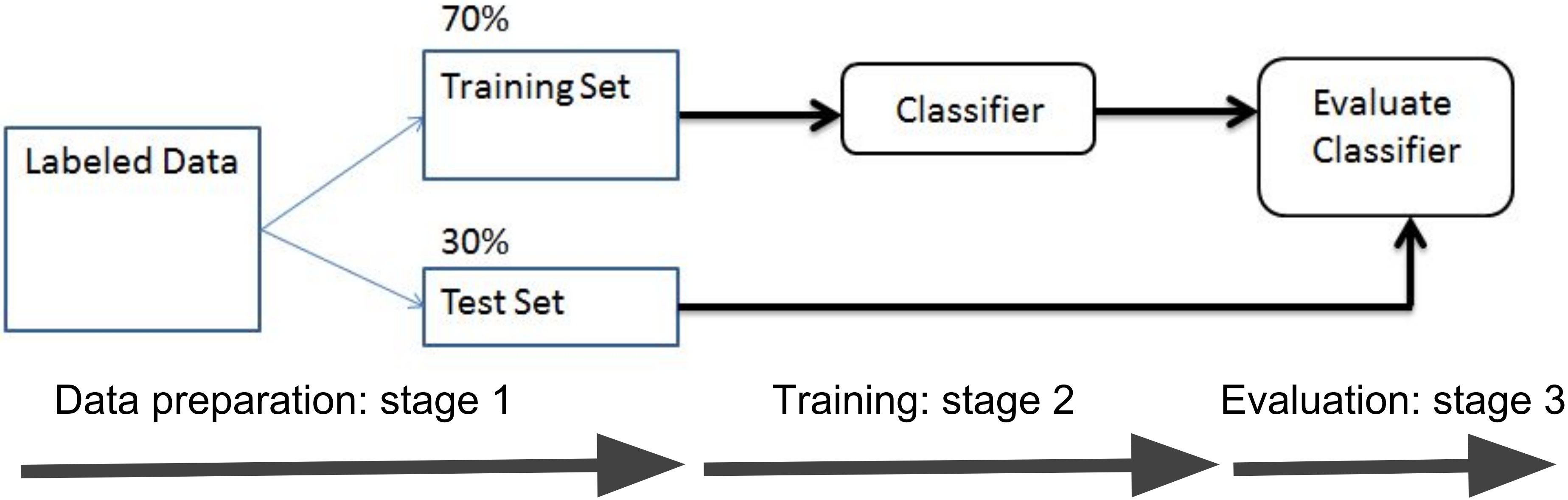
Learning relations between data (input) and output



TRAINING

Classification Model

Classification process



Data preparation: stage 1

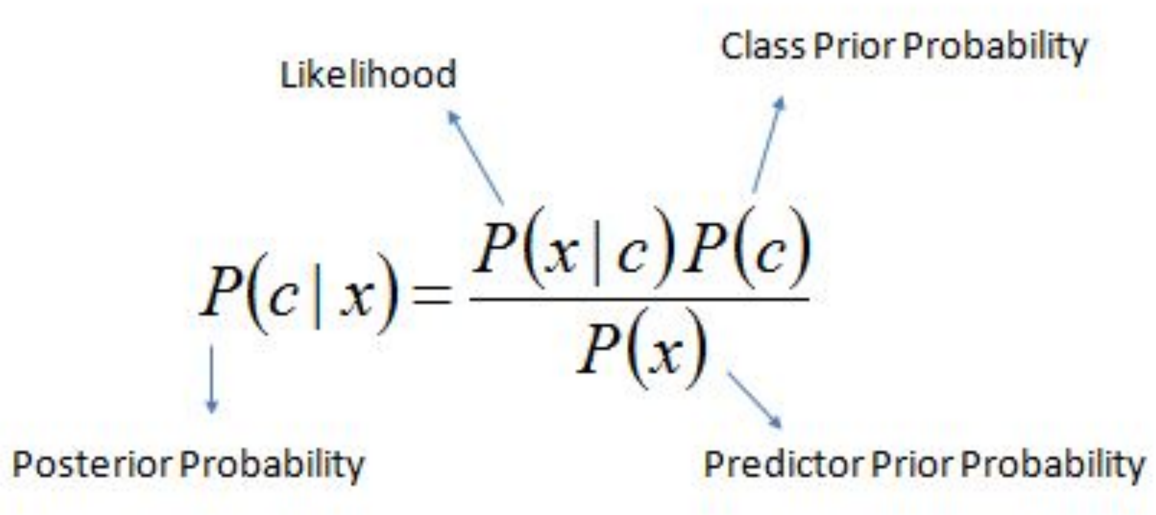
Training: stage 2

Evaluation: stage 3

Classification models

1. Linear Classifiers: Logistic Regression, Naive Bayes Classifier
2. Support Vector Machines
3. Decision Trees
4. Boosted Trees
5. Random Forest
6. Neural Networks (deep learning)
7. Nearest Neighbor

Naive (why?) Bayes model

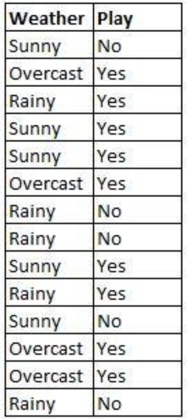


$$P(c \mid x) = \frac{P(x \mid c)\,P(c)}{P(x)}$$

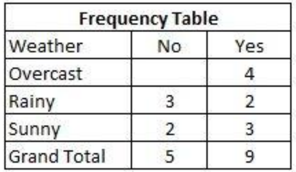
$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

How Naive Bayes algorithm works?

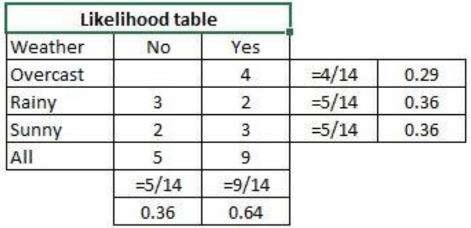
1. Convert the data set into a frequency table
2. Create Likelihood table
3. Use Naive Bayesian equation to calculate the posterior probability for each class.
4. The class with the highest posterior probability is the outcome of prediction.



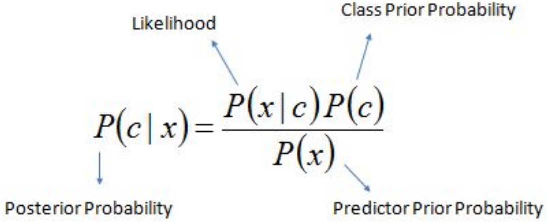
Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No



Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9



Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		



$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Tool suite for the data scientist

- Python vs R
- IDE (e.g. pyCharm) vs Live code (e.g. Jupyter Notebook)
- Statistical vs Neural methods
- Scikit learn library (Machine learning in python)
- Natural Language ToolKit library (NLP in python)

Scikit Learn a.k.a. sklearn

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts (classification, regression, clustering,...)
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable under BSD license
- `>>> import sklearn`
- `>>> from sklearn import something`

Natural Language ToolKit a.k.a NLTK

- leading platform for building Python programs for Natural language processing
- easy-to-use interfaces to over 50 corpora and lexical resources
- text processing libraries for classification, tokenization, stemming, tagging, parsing,semantic analysis, …
- >>> import nltk
- >>> from nltk.corpus import treebank

Lab: News classifier

- Create an account on https://repl.it
- Fork or clone the Workshop material: https://github.com/noureddine01/MLWorkshopMaterial
- Create MLWorkshopLab1.py on your repl account
- Train the model
- Do a prediction
- Assess the Model by yourself
- Use the model ->

Use the model

```
# new instances where we do not know the answer

Xnew, _ = [["new event in the weekend", "hello world"],[]]

# make a prediction

ynew = model.predict(Xnew)

# show the inputs and predicted outputs

for i in range(len(Xnew)):

    print("X=%s, Predicted=%s" % (Xnew[i], news.target_names[ ynew[i]]))
```

Where to go from here?

- Install Jupyter Notebook http://jupyter.org
- Explore other data sets
  http://scikit-learn.org/stable/auto_examples/index.html#dataset-examples
- Try other algorithms/models
- Try other sklearn examples
  http://scikit-learn.org/stable/auto_examples/index.html#classification
- Go for real classification problems
  www.kaggle.com

Thanks for your attention