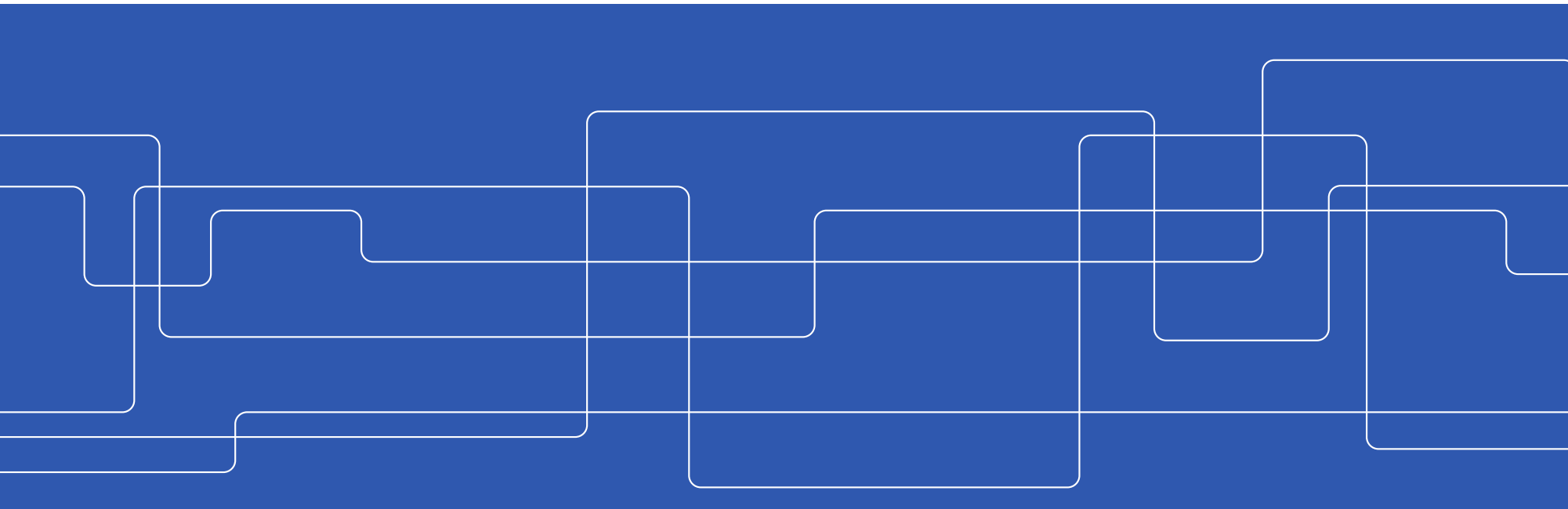
ID2214/FID3214

Programming for Data Science – Introduction

Henrik Boström

Prof. of Computer Science - Data Science Systems
Division of Software and Computer Systems
School of Electrical Engineering and Computer Science
KTH Royal Institute of Technology
bostromh@kth.se

October 28, 2019



Outline

Data science - a Definition

Data science concerns methods and technology for **collecting**, **organizing** and **analyzing** data with the purpose of extracting new **knowledge**

Collecting Data

Traditionally, knowledge is extracted from data that has been collected to answer a specific question

- Is drug A more effective than drug B?
- Do the customers like web site A more than B?

but in many cases, the data one has access to has been collected for other purposes

- Customer orders
- Electronic patient records
- Operational data

Organizing data

Often, the data has a rich (or lack of) structure, troubling techniques that assume fixed-length feature vectors

- Social networks
- Biological sequences
- Chemical compounds
- Free text

or needs to be processed to meet other requirements

- Time and space limitations
- Quality issues

or comes from several different sources or time points

- Early or late fusion
- Aggregation

Analyzing Data

The result of the analysis can be interpretable and provide novel insights (declarative knowledge)

- Customers that buy product A will usually later buy product B
- Drug A in combination with drug B increases risk for dizziness

or opaque (black-box) and may be use to automate processes (procedural knowledge)

- What registration number does the car on the image have?
- Should this credit card transaction be blocked?
- Is this machine about to break down?

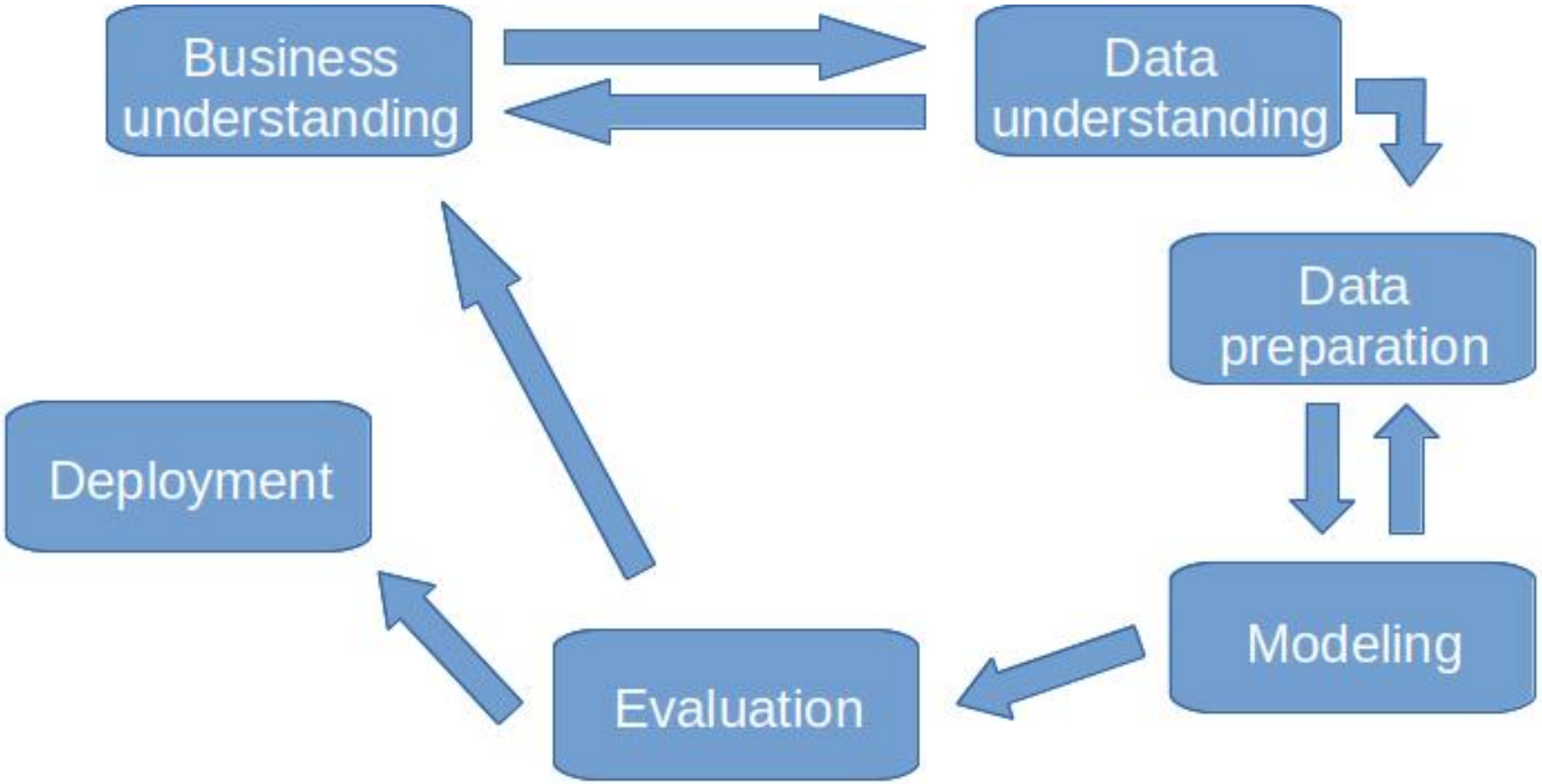
Extracting Knowledge from Data

Algorithms for extracting knowledge from data have been developed during several decades, partly in parallel within machine learning (artificial intelligence) and statistics

The big challenge in a data science project is typically not to choose algorithm, but to understand the problem and organize data so that useful knowledge can be extracted and exploited

- Analysis of electronic patient records to be able to automatically detect adverse drug events
- Analysis of operational vehicle data to predict remaining useful lifetime of components and optimize maintenance

Cross Industry Standard Process for Data Mining



Shearer C., The CRISP-DM model: the new blueprint for data mining, Journal of Data Warehousing 5 (2000) 13-22

Intended Learning Outcomes

Having passed the course, the student should be able to

1. account for and discuss the application of, and
2. implement and apply
   - techniques to convert data to appropriate format for data analysis
   - algorithms to analyse data through supervised and unsupervised machine learning
   - techniques and performance measurements for evaluation of data analysis results

Lectures

L1 i) Introduction ii) Introduction to Python
L2 i) Introduction to Python (cont.) ii) NumPy and pandas
L3 i) Data Preparation ii) Evaluating Predictive Models
L4 Linear models
L5 Naïve Bayes and k-Nearest Neighbors
L6 Artificial Neural Networks
L7 Decision Trees
L8 Combining Models
L9 i) Exploratory data analysis ii) Hyperparameter tuning
L10 Methodology
L11 Unsupervised Learning

Literature

- I. Witten, E. Frank, M. Hall and C. Pal, Data Mining: Practical Machine Learning Tools and Techniques (4th ed.), Morgan Kaufmann, 2016 ISBN: 9780128042915.
- J. VanderPlas, Python Data Science Handbook: Essential tools for working with data (1st ed.), OReilly Media Inc., 2016 ISBN: 9781491912058.

Examination

- Assignments, 4.5 ECTS (ID2214: A-F, FID3214: P/F)
  - Four assignments; the three first may give up to two points each (one is required), while the last may give up to three points (one is required). The sum is converted to a grade.
  - The assignments are done in project groups (ID2214: 3 students, FID3214: 1-2 students)
- Written examination, 3 ECTS (ID2214: A-F, FID3214: P/F)
  - A check that the student is able to account for, discuss the application of, implement and apply central concepts, algorithms and techniques.
- Course grading: (ID2214: A-F, FID3214: P/F)
  - ID2214: at least E is required on the assignment and written examination, and the course grade is the average of the these grades (rounded upwards)
  - FID3214: for P, a P is required on both the assignments and written examination

Assignments

1. Data Preparation and Experimental Setup
   - Implement routines for normalization, discretization, one-hot encoding and imputation
   - Implement routines for evaluating predictive models
2. k-Nearest Neighbors and Naïve Bayes
   - Implement routines for generating and applying predictive models
3. Decision Trees and Forests
   - Implement routines for generating and applying predictive models
4. Data Science Project
   - ID2214: Use source data and routines for generating features to develop the strongest possible model with an estimate of its performance and present results at seminar and in a report
   - FID3214: Choose a topic, present preliminary results at a seminar and final results in a short paper

Programming Languages for Data Science

Obviously, any programming language can be used for data science projects. However, some languages have become more popular than others, possible reasons being:

- they allow for rapid implementation and testing
- they support easy manipulation of matrix/tabular data
- there is a large number of packages/libraries available, including bridges to software written in other languages
- there is an active community of users and developers

Languages frequently listed as suitable for data science include: Python, R, Julia, Java, SAS, SQL, Matlab, Scala, F#

Popular Software for Data Science

In a recent poll[1], asking *What Analytics, Big Data, Data Science, Machine Learning software you used in the past 12 months for a real project?*, the answer by 2052 participants was:

Python (66%), RapidMiner (53%), R (48%), SQL (40%), Excel (39%), Anaconda (33%), Tensorflow (30%), Tableau (26%), scikit-learn (24%), and Keras (22%)

Some identified clusters:

- Python, Anaconda, scikit-learn, Tensorflow, Keras, Spark
- SQL, Excel,Tableau

[1] `www.kdnuggets.com/2018/05/poll-tools-analytics-data-science-machine-learning-results.html`

Python

The development of Python started in the late 80's, by Guido van Rossum, version 1.0 was released in 1994, 2.0 in 2000, and 3.0 in 2008. There are two current branches (v. 2.7 and and 3.7).

Some notable features of the language are:

- multi-paradigm, general purpose, programming language; object-oriented, structured, (largely) functional, etc.
- dynamic typing; typed objects and untyped variable names
- built to be highly extensible
- uses white-space indentation to delimit blocks
- statements can not be part of expressions
- very large standard library
- The official repository (PyPI) of third-party Python software contains over 130 000 packages

Python Example

```
def hello(name,n):
    print("Hello, {}!".format(name))
    for i in range(n):
        print(i)
```

R

R was developed by Ross Ihaka and Robert Gentleman, with a first version released in 1995. It is an open-source version of the S language, which was created by John Chambers in 1976.

Some notable features of the language are:

- developed for statistical analysis
- one of the richest ecosystems to perform data analysis; very large set of statistical libraries and very well developed visualization packages
- huge user community
- Public repositories of third-party R software, most notably the Comprehensive R Archive Network (CRAN), contains over 15 000 packages

R Example

```
hello <- function(name,n){
            print(paste("Hello,",name,"!",sep=""))
            for (i in 1:n){
               print(i)
            }
          }
```

Julia

Julia developent started in 2009, by Jeff Bezanson, Stefan Karpinski, Viral B. Shah, and Alan Edelman, and 1.0 was released in 2018.

Some notable features of the language are:

- multi-paradigm programming language; object-oriented, structured, functional, etc.
- multiple dispatch and dynamic typing; optionally typed
- designed for high performance and parallelism
- not so large user community (yet)
- a growing number of packages, in particular within data science and machine learning, currently just over 1 900 registered packages

Julia Example

```
function hello(name,n)
    println("Hello, $(name)!")
    for i = 1:n
        println(i)
    end
end
```

Summary

- During the course you will learn concepts and techniques from the area of data science, which you will apply in practice through implementing and evaluating algorithms for organizing and analyzing data.
- By actually implementing algorithms, rather than just using existing libraries, you will gain a deeper understanding of the techniques and algorithms and learn how to experiment with new variants and combinations, which is a prerequisite for developing novel solutions with increased efficiency or effectiveness.