Project acronym:	**EDSA**
Project full name:	**European Data Science Academy**
Grant agreement no:	**643937**

D2.3 Data Science Curricula 3

Deliverable Editor:	**Huw Fryer (SOTON)**
Other contributors:	**Elena Simperl (SOTON), Module curricula prepared by partners as noted in document: ODI, FRAUNHOFER, TU/e**
	Joos Buijs (TU/e)
Deliverable Reviewers:	**Alex Mikroyannidis (OU)**
Deliverable due date:	**31/07/2017**
Submission date:	**26/07/2017**
Distribution level:	**Public**
Version:	**1.0**

This document is part of a research project funded
by the Horizon 2020 Framework Programme of the European Union

Change Log

Version	Date	Amended by	Changes
0.1	03/06/2016	Huw Fryer	Structure of document and initial content.
0.2	05/07/2017	Huw Fryer	Added analysis about skills from EDISON and demand analysis
0.3	09/07/2017	Huw Fryer	Version submitted for internal review
0.4	24/07/2017	Huw Fryer	Responses to comments from internal review
0.5	25/07/2017	Elena Simperl	Scientific Director approval
1.0	26/07/2017	Aneta Tumilowicz	Final QA

Table of Contents

List of Tables

1. Executive Summary

This deliverable presents the final version of the EDSA curriculum, which comprises of 15 core data science topics that are being delivered by the 3-year project. Previously, we released the first version of the curriculum containing 6 of 15 modules, each of which had a specific curriculum containing learning objectives, topic descriptions and various resources and materials that been identified. Modules are separated into four stages: foundations, data storage and processing, data analysis and data interpretation and use. These categories remain in version 2 and provide structure to our courses. We present new modules in this deliverable, reflecting a number of changes to the curriculum's list of topics, along with revisions to the previously released modules based on learner feedback, teaching experience and increased amounts of demand analysis.

2. Introduction

In this document, we present the final version of the EDSA curriculum, based on other individual curricula from EDSA partners. Four new curricula which have been added: Programming/computational thinking; Data intensive computing; Social media analytics; and Data exploitation including data management.

We will begin by providing a recap of the previous two iterations of the curriculum, before discussing the processes involved in developing the curriculum, including the response to demand analysis; and the skills frameworks we have developed. There were only very minor changes to existing courses for this iteration, so these are discussed briefly, before we introduce the new curricula for this deliverable. We finish by presenting our conclusions about the final EDSA curricula.

2.1 Recap of curricula versions 1 and 2

Before presenting the new version of the EDSA curriculum, we revisit the previous versions which were released in M6 and M18.

2.1.1. Modules released in year 1

The 15 topics to make up the core EDSA curriculum was divided up into four stages: Foundations, Storage and Processing, Analysis, and Interpretation and Use. The modules were divided amongst consortium partners according to their expertise, with the release plan as follows:

Table 1 - Core EDSA Curriculum, version 1

Topic	Stage	Schedule	Allocated Partner
Foundations of Data Science	Foundations	M6	SOTON
Foundations of Big Data	Foundations	M6	JSI
Statistical / Mathematical Foundations	Foundations	M18	JSI
Programming / Computational Thinking (R and Python)	Foundations	M30	SOTON
Data Management and Curation	Storage and Processing	M18	TU/e
Big Data Architecture	Storage and Processing	M6	Fraunhofer
Distributed Computing	Storage and Processing	M6	KTH

Data Intensive Computing	Storage and Processing	M30	KTH
Machine Learning, Data Mining and Basic Analytics	Analysis	M6	Persontyle
Big Data Analytics	Analysis	M18	Fraunhofer
Process Mining	Analysis	M6	TU/e
Data Visualisation	Interpretation and Use	M18	SOTON
Visual Analytics	Interpretation and Use	M30	Fraunhofer
Finding Stories in Open Data	Interpretation and Use	M18	ODI
Data Exploitation including data markets and licensing	Interpretation and Use	M30	ODI

Following the schedule outlined in Table 1, curricula for the first six modules were released in M6:

- Foundations of Data Science
- Foundations of Big Data
- Big Data Architecture
- Distributed Computing
- Machine Learning, Data Mining, and Basic Analytics
- Process Mining

Each was released by a different EDSA partner as a syllabus, containing the following information:

- Learning Objectives
- Syllabus and Topic Descriptions
- Existing Courses
- Existing Materials
- Example Quizzes and Questions
- Description of Exercises
- Further Reading

This template for releasing the curricula means that partners can work according to their specialty, and additionally target different audiences or sectors. It was retained for version 2 of the EDSA syllabus, and for this version as well.

2.1.2. Modules released in year 2

In year 2, we made modifications to the curricula as a result of feedback from demand analysis and industry advisory board, and to the overall EDSA curriculum as a whole. The following changes were made to the EDSA curriculum in M18:

- The 'Data Visualisation' and 'Finding Stories in Open Data' modules were merged, since their intended aims held considerable overlap
- An additional module 'Linked Data and the Semantic Web' was released, since there was very little coverage of this area of data science. The syllabus and resources from this module were adapted from the previous EU project EUCLID[1], in which a number of EDSA partners participated.

[1] http://euclid-project.eu/

- The 'Visual Analytics' module to be delivered in M30 by Fraunhofer was replaced with 'Social Media Analytics', in order to broaden the curriculum to ensure a greater coverage of key data science topics.

As such, according to the planned schedule, and the modifications described, the following modules were released in M18, and their curricula described in D2.2:

- Statistical / Mathematical Foundations
- Data Management and Curation
- Big Data Analytics
- Data Visualisation and Storytelling
- Linked Data and the Semantic Web[2]

The core curriculum at M18 was as follows:

Table 2 - Core EDSA Curriculum, version 2

Topic	**Stage**	**Schedule**	**Allocated Partner**
Foundations of Data Science	Foundations	M6	SOTON
Foundations of Big Data	Foundations	M6	JSI
Statistical / Mathematical Foundations	**Foundations**	**M18**	**JSI**
Programming / Computational Thinking (R and Python)	Foundations	M30	SOTON
Data Management and Curation	**Storage and Processing**	**M18**	**TU/e**
Big Data Architecture	Storage and Processing	M6	Fraunhofer
Distributed Computing	Storage and Processing	M6	KTH
Stream Processing	Storage and Processing	M30	KTH
Linked Data and the Semantic Web	**Storage and Processing**	**M18**	**SOTON**
Machine Learning, Data Mining and Basic Analytics	Analysis	M6	Persontyle
Big Data Analytics	**Analysis**	**M18**	**Fraunhofer**
Process Mining	Analysis	M6	TU/e
Social Media Analytics	Interpretation and Use	M30	Fraunhofer
Data Visualisation and Storytelling	**Interpretation and Use**	**M18**	**ODI**
Data Exploitation including data markets and licensing	Interpretation and Use	M30	ODI

[2] This was described in D2.2, but the curriculum was already largely complete from previous work in the EUCLID project.

3. Insights from demand analysis, community feedback, and EDSA advisory board

In version 2, the curricula developed were assisted by the release of D1.4, which provided insights into the demand for data science and was able to be fed into the curriculum. In particular, the deliverable produced seven recommendations in regard to EDSA curriculum development as indicated in Table 3.

Table 3 - Recommendations for EDSA curriculum development from D1.4

Title	Intervention level	Summary description
Holistic training approach	General training approach	Refine the EDSA's training approach and curriculum cycle to strengthen skills along the full data exploitation chain.
Open source based training	Existing curriculum design	Continue current technical and analytical training based on open source technologies; apply cross-tool focus to deliver overarching training.
Soft skills training	Expansion of curriculum	Implement soft skill training to increase performance and organisation impact of data scientists / data science teams.
Basic data literacy training	Expansion of curriculum	Develop basic data literacy and data science training for non-data scientists to improve basic skills across organisations and facilitate uptake of data-driven decision making and operations.
Blended training	Course delivery	Develop blended training approaches including sector-specific exercises and examples to increase effectiveness of training delivery.
Data science skills framework	Training approach and delivery	Implement data science skills framework to structure skills requirements, assess skills of data scientists, and identify individual skills needs.
Navigation and guidance	Training market	Develop quality assessment of third party courses; provide navigation support to identify relevant trainings from EDSA and third parties.

The implementation of these recommendations was incomplete, with the intention that the final version of the curriculum be able to complete the process of acting upon the demand analysis. Version 2 was influenced by rows 2-4 inclusive of Table 3. This version has built upon the work from version 2, with an increased focus on the first recommendation 'Holistic training approach', and the final two 'Data science skills framework', and 'Navigation and guidance'.

The fifth recommendation 'Blended training' in regards to the course delivery was considered for the development as well. However, following feedback from the European Commission at the midway point

of the project, it was decided that a greater emphasis be placed upon the curation rather than delivery of courses. Therefore, it is no longer relevant for consideration in this document.

The improvements made to our approach in this iteration are discussed below.

3.1 Data science skills framework

In D2.2, we presented an initial version matching modules to the skills that they train, based on the curricula released in the first two versions of the EDSA curriculum. For the following version, we expanded this by mapping the dashboard skills to the comprehensive classification in the EDISON project's Data Science Body of Knowledge[3], using their "Skill Areas" to map a path for participants through the courses on offer within EDSA. The EDISON data science skills areas are also to be used for the learning pathways, where we seek to guide a prospective job applicant through the range of EDSA courses with the prerequisites and outcomes towards a particular job. The implementation of this is part of the dashboard, and will be discussed in more detail in D2.6.

A summary of the coverage of the EDSA curriculum according to the EDISON skills areas is presented in Table 4. This contains the individual knowledge areas, and the amount of the curricula which cover that particular skill.

Table 4 - Coverage of EDSA curricula according to EDISON Data Science Skills Framework

EDISON Knowledge Area	Amount of Curricula
Big data technologies and systems	6
Computing methodologies	5
Infrastructure and platforms for data science applications group	5
Information systems	4
Computer systems organisation for big data applications	3
Big data systems organisation and management	3
Data management and Enterprise data infrastructure	3
General principles and concepts in data management and organisation	3
Modelling and simulation	2
Theory of computation	2
Big data (data science) applications design	2
Data management systems	2
Scientific/research methods	2
Business analysis and enterprise organisation	1
Digital libraries and archives	1
Big data software organisation and engineering	1
Mathematics of computing	1
Business process management	1
Business analysis organisation and management	0
Software engineering and management	0

[3] http://edison-project.eu/sites/edison-project.eu/files/filefield_paths/edison_ds-bok-release1-v0.3.pdf

3.2 Demand analysis

The following process was established to ensure that the EDSA curriculum, and by extension courses, remains consistent with demand analysis:

1. The EDSA curriculum is updated based on the demand analysis every 12 months and specifically on M6, M18 and M30 (delivered as D2.1, D2.2 and D2.3 respectively). This is led by SOTON and other consortium partners in WP2 contribute.
2. The curriculum defines the topics for which we offer courses. SOTON assigns responsibilities to WP2 partners to address these topics according to their expertise.
3. Based on this assignment, partners develop new courses or survey existing external courses. New and existing courses have to adhere to the demand analysis recommendations (see D1.4 & D2.5).
4. New and existing courses are added to the EDSA courses portal as soon as they are ready and launched on M12, M24 and M36 (delivered as D2.4, D2.5 and D2.6 respectively). This is led by the OU and WP2 partners contribute.
5. The OU is responsible for maintaining and updating the courses portal, as well as collecting and updating the metadata of new & existing courses. For this reason, a survey is circulated regularly among partners, asking them to map their courses to the top dashboard skills.

For this version of the curriculum, the demand dashboard was used to establish the levels of demand for particular skills. The most popular skills in demand from the EDSA dashboard were collected, and distributed to partners in the consortium on a bimonthly basis. From there, consortium members identified which skills were covered by their courses, and which areas of the curriculum which may require modifications. Skills areas from the EDISON framework (see above) were also used for this purpose. The process for updating the curriculum in response to this is as follows:

1. A discussion led by SOTON is conducted, where the gaps which have been identified are analysed. It is decided between the partners whether it is appropriate to update either their individual curricula, or the curriculum in general.
2. Individual partners identify which of the missing skills can be covered within their areas of the curriculum, and if so, their curricula are updated in the next version of the curriculum deliverable.
3. In the event that none of the curricula can accommodate the required skills, the leader of the curriculum development task (SOTON) analyses the missing areas in the context of the overall curriculum. A proposal is made to all partners to update the curriculum
4. Proposed changes are discussed, and if approved, the EDSA curriculum is modified.

For this version of the curriculum, areas were identified from the demand dashboard, and EDISON mapping, as not being covered by the EDSA curriculum. From the EDISON skills areas, the following had little or no coverage:

- Software engineering and management
- Business analysis organisation and management
- Business analysis and enterprise organisation

The consortium considered the gaps in the curriculum, and considered that there was in fact a distinction between what core “data science” should comprise; and what skills might be useful for a job in data science. The other skills and knowledge areas were considered to be out of scope, as not being part of data science. It was decided by the consortium that areas such as business organisation, or software engineering were too general, and were not directly associated with data science.

A similar issue was discovered with skills from the demand dashboard. Since these were skills requested by employers, they were not necessarily data science, or vague and general (e.g. computer science), or overly specific and tied to a specific technology, such as jQuery, MySQL, HTML5, Hadoop, etc. Whilst the consortium does include references to these technologies, these are as an example of more general data science principles. Tying the curriculum to a new technology was not considered appropriate, and as such the curriculum remained unchanged.

The process was modified slightly to include a longer tail of skills, and partners were to select skills they felt should be in a data science curriculum, before a second part of the survey was distributed as before. The latest survey had similar issues, but the filtered skills were able to better capture the demand for data science skills.

The comparison between the two processes can be seen in Tables 5 and 6, containing the responses from two surveys with the skills covered in the left column, and the missing skills in the right column. Table 6 demonstrates that the survey in July 2017 had a greater coverage than previous surveys, with 25 out of 57 skills covered, as opposed to 18 out of 50. Nevertheless, the same problems remained, where individual technologies continued to dominate the list of skills "missing" from the EDSA curriculum.

Table 5 - Coverage of skills from demand analysis in EDSA curriculum, from survey April 2017

Skills covered by EDSA Curriculum	Skills missing from EDSA Curriculum
.net	android
.net framework	architect
analysis	artificial intelligence
analyst	assurance
analytics	automation
computer science	backend
data analysis	business intelligence
data management	c++
data mining	cloud
data science	compiler
database	data warehouse
html	design
java	devops
nosql	finance
python	hardware
relational database	html
scripting language	ios
sdlc	javascript
	jquery

	leadership
	linux
	machine learning
	matlab
	metadata
	mysql
	oracle
	php
	project management
	reverse engineering
	sales

Table 6 - Coverage of skills from demand analysis in EDSA curriculum, from survey July 2017

Skills covered by EDSA Curriculum	Skills missing from EDSA Curriculum
amazon web services	.net
analysis	.net framework
analytics	analyst
artificial intelligence	asp.net
cloud	backend
computer science	c
data analysis	c++
data management	data modeling
data mining	data visualization
data science	data warehouse
database	design
distributed computing	git
hadoop	java
json	jquery
machine learning	matlab
nosql	metadata
python	mongodb

r	mysql
relational database	node.js
scripting language	oracle
simulation	postgresql
sql	project management
	scrum
	sdlc
	security
	sharepoint
	software development
	software engineer
	unit testing
	user interface
	vmware

The main concession the consortium has made to this issue has been the inclusion of the "Programming/Computational Thinking" curriculum, released in this version of the curriculum. This is different to the other curricula, since it teaches the absolute fundamentals of computer science, and looks at far more general principles than other modules, such as databases; algorithms; and abstraction. Therefore, where there is demand for a general computer science skill in a job advertisement, the EDSA curriculum does satisfy the demand, but at a basic level.

4. Updated curricula released for year 3

The final version of the curriculum deliverable includes no major modifications to existing modules. Changes have been made as follows:

1. For Data Intensive Computing, the focus of the module was shifted towards Data Intensive Systems where Stream processing (the old name of the module) is a part of the module, as a result of demand analysis.
2. In Data Visualisation and Storytelling, the data visualisation part of the course was slightly modified to include more information on theories from Tufte, and chart junk, whilst also increasing the emphasis on data processing. The content relating to data stories has further been streamlined, since there was replication of content in some areas.

The following four new modules have been released, with Table 7, showing the released curricula within their context of the remainder of the EDSA curriculum.

- Programming / Computational Thinking (R and Python)
- Data Intensive Computing
- Social Media Analytics
- Data Exploitation including data markets and licensing

Table 7 - Core EDSA Curriculum, version 3

Topic	**Stage**	**Schedule**	**Allocated Partner**
Foundations of Data Science	Foundations	M6	SOTON
Foundations of Big Data	Foundations	M6	JSI
Statistical / Mathematical Foundations	Foundations	M18	JSI
Programming / Computational Thinking (R and Python)	**Foundations**	**M30**	**SOTON**
Data Management and Curation	Storage and Processing	M18	TU/e
Big Data Architecture	Storage and Processing	M6	Fraunhofer
Distributed Computing	Storage and Processing	M6	KTH
Data Intensive Computing	**Storage and Processing**	**M30**	**KTH**
Linked Data and the Semantic Web	Storage and Processing	M18	SOTON
Machine Learning, Data Mining and Basic Analytics	Analysis	M6	Persontyle
Big Data Analytics	Analysis	M18	Fraunhofer
Process Mining	Analysis	M6	TU/e
Social Media Analytics	**Interpretation and Use**	**M30**	**Fraunhofer**
Data Visualisation and Storytelling	Interpretation and Use	M18	ODI
Data Exploitation including data markets and licensing	**Interpretation and Use**	**M30**	**ODI**

4.1 Programming/computational thinking - SOTON

This is a course about getting people to think like computer scientists. Many people think computer science is only about writing code. Whilst this is a part of it, in this course we aim to show more of the processes behind writing the code, adding the code as an afterthought. We base the course on one of our modules at the University of Southampton, which we use to get people from backgrounds other than computer science up to speed on our Web Science programmes.

We also use the work of Wing (2006) about computational thinking.

In addition to being able to focus on thinking like a computer scientist, we also wish to introduce learners to the practical skills they need to put this thinking into practice. As such, each week we have included instructions about using the Python programming language to apply core computational and programming concepts.

This course is an exception when compared to the other courses in the EDSA curriculum, in that it is a foundational course which provides background to complete the remainder of the data science courses. As such, it is slightly further away from what might be considered a "core" of data science.

4.1.1. Learning objectives

After completion of this course, participants will be able to:

1. Describe how computing is possible through a series of abstractions
2. Assess the most effective algorithm for a particular task
3. Design simple algorithms for common computing tasks
4. Retrieve data stored from a Web API, and store it in a relational database
5. Develop simple applications using the Python programming language
6. Approach everyday problems with a computational focus

4.1.2. Syllabus and topic descriptions

This is an introductory course, which introduces the concepts required to be able to work as a computer scientist. The layout of the course is as follows:

Algorithmic Thinking

In this section, students are introduced to the concept that there are different ways of performing tasks, some of which are better than each other. In addition, with these different ways, the students will learn how an algorithm may be evaluated

Algorithms

- **Introduction** - Introduction to algorithms, and why they are useful. Compares a computational algorithm to a food recipe
- **Different Methods** - There are many ways of doing the same thing, each with advantages and disadvantages. Shows that a better algorithm can lead to a greater improvement in performance than faster computers
- **Evaluation** - Having established that more than one algorithm can perform the same task, we introduce some formal means of evaluating the average case, best case, and worst case scenarios.
- **Tractability** - Not all algorithms can complete in a reasonable time, sometimes an approximate solution is a better option

Data Structures

- **Introduction** - Data structures have different forms, to optimise a particular type of function given certain constraints
- **Simple data structures** - Introduces the stack and queue as data structures, and how efficiently they perform certain tasks

Computing as an abstraction

In this section, the course looks at ways in which computers make use of black boxes and other abstractions to build upon what has been completed before:

- **Computer architecture** - A computer is a clear example of an abstraction, discusses how it gets from switches and gates to an operating system
    - Binary/machine code, logic gates, hex, assembly
- **Programming as an abstraction** - Computers are good at automation, here we see how the automation of some principles leads to
    - Compiled vs interpreted
    - Different types of programming, e.g., procedural, functional, object oriented, event based
    - Higher level languages
- **Object oriented programming** - A class is an abstraction of a thing, containing representations only of the things we want it to.
- **Data as an abstraction**
    - How data or information are abstracted from binary to represent text, statistics, or other media content

The Web and the Cloud

The World Wide Web (the Web) is central to modern data processes. In this section, we introduce the Web as an instance of a computer network. With the ubiquity of Internet connections, an inevitable extension was the concept of the Cloud, where online operators provide and manage services which would previously have been done on a business's own infrastructure.

- **What are computer networks?** An introduction to basic concepts of how data are transported across a network
- **The WWW** - The most widely known network
    - Client/server model - Explains the model which is used by the Web
    - HTTP - Explains the background to the protocol handles request and response to connect to the Web, and the use of different REST verbs
    - Displaying pages on the Web
        - HTML & CSS - separation of concerns between content and presentation
- **The Cloud**
    - **Introduction to the cloud** - Services provided by cloud companies vary from provision of hardware to fully running services. This section explains the cloud business model.
    - **Types of cloud services** - This section introduces the different types of services provided by cloud companies, such as Iaas, PaaS, SaaS

Data Management

This section uses "data management" in a broad way, as to define all different means of manipulating and representing data.

- **Introduction to data management**
    - Store data somewhere, whilst being able to have access to it
    - Different trade-offs with different situations
    - Types of data: Unstructured, semi-structured, structured
    - Different scenarios for data management
- **Data Representation** - Provides examples of a variety of scenarios of how data may be represented, such that it can be manipulated for analysis in the future.
    - **Text files** - The simplest way of representing data. Introduces the following data formats: csv/tsv/json/xml, and good practices for log files

- **Images, videos** - Following analysis of data it may be represented in an image or video; or these could themselves be used as data to be analysed. Discussion of common image and video encoding.
- **Relational database** - The most common way of storing data, using SQL to manipulate and retrieve.
- **NoSQL** - Introduces the CAP theorem, and discusses trade-offs between relational and non-relational data.
- **APIs** - Data stored in some manner can be made available from an API. Builds on the discussion from HTTP and introduces REST APIs

Python Programming

This is done alongside the other sections, as a means of applying the theoretical concepts introduced in the other week. It will not give expertise in Python programming, but will provide a foundation to develop expertise in the future. The intention is not to teach how to program in Python so much, but rather to apply the principles in the remainder of the course.

Python is chosen above R for this curriculum, since it is more general purpose, and more relevant to the principles of programming generally as opposed to R which fits better as a language for the mathematical side of programming. In addition, there are other courses within this curricula which additionally cover R, such as '*Statistical and Mathematical Foundations*'. Teaching Python will cover only the essentials sufficient to perform these tasks, with students referred to other courses should they wish to gain a more detailed understanding.

- Setting up a development environment
- Primitive types
- Control flow and loops
- Classes and functions

4.1.3. Existing courses

This is an introductory course, and offers an introduction to the different areas described in each week, in addition to basic Python programming. The online course is based upon the module delivered for Masters students at the University of Southampton, which is itself based on a course at Berkeley. More generally, in the UK, it is part of the national curriculum for schools to include content on "computational thinking", and as such most schools could also be regarded as having an existing course[4].

Given the nature of the course, as covering a broad area without much depth, there are courses which will cover the individual elements, a selection of which have also been included.

Computational Thinking - University of Southampton

This is an MSc module taught to Web Science students at the University of Southampton, and is a course is aimed at students who do not have a background in computer science. This is distinct from EDSA, aiming to provide a broader understanding of computer science and topical issues, and be able to understand and communicate this technical information effectively [Link].

The Beauty and Joy of Computing

This is a course which was developed at Berkeley [Link], but now additionally is provided as an open curriculum at http://bjc.berkeley.edu/website/curriculum.html

Introduction to Computational Thinking - Open University

[4] https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study/national-curriculum-in-england-computing-programmes-of-study

This self-study course is based strongly from Wing's work, and expounds on the principles she describes in a presentation she gives. [Link]

Introduction to Computational Thinking and Data Science - MIT OCW

This course is an online version of the 6.002 course taught at MIT. It is a more advanced course than suggested by the syllabus, expected for people who already have familiarity with algorithms and complexity, and advances those skills. [Link]

4.1.4. Existing materials

Google provides a series of resources for computational thinking educators, who are educating children up to the age of 18. This includes an overview of their curriculum for what they consider computational thinking to be. [Link]

4.1.5. Sources for further reading

Guttag, John. *Introduction to Computation and Programming Using Python: With Application to Understanding Data*. 2nd ed. MIT Press, 2016. ISBN: 9780262529624

Wing, J.M., 2006. Computational thinking. *Communications of the ACM*, *49*(3), pp.33-35.

Tedre, M. and Denning, P.J., 2017. The long quest for computational thinking. *Communications of the ACM*, 60(6), pp.33-39.

4.2 Data Intensive Computing - KTH

The module contains three parts.

4.2.1. Part I. Data Intensive Computing foundations

(this syllabus is built using materials from courses developed by Dr. Amir Payberah and Prof. Seif Haridi - KTH).
In this course, we describe the critical technology trends that are enabling cloud computing and the services and applications they offer. The course covers a wide variety of advanced topics in data intensive computing, including distributed file systems, NoSQL databases, processing data-at-rest (batch data) and data-in-motion (streaming data), graph processing, and resource management. The course is mainly based on research papers. The following list also shows the main platforms we will cover in this course.

4.2.1.1. Intended learning outcomes

The course complements distributed systems courses, with a focus on processing, storing and analysing massive data. It prepares the students for master projects, and Ph.D. studies in the area of data-intensive computing systems. The main objective of this course is to provide the students with a solid foundation for understanding large scale distributed systems used for storing and processing massive data.
More specifically after the course is completed the student will be able to:

- explain the architecture and properties of the computer systems needed to store, search and index large volumes of data
- describe the different computational models for processing large data sets for data at rest (batch processing) and data in motion (stream processing)
- use various computational engines to design and implements nontrivial analytics on massive data
- explain the different models for scheduling and resource allocation computational tasks on large computing clusters

- elaborate on the tradeoffs when designing efficient algorithms for processing massive data in a distributed computing setting.

4.2.1.2. Syllabus and topic descriptions

- Introduction
  - Concepts and principles of cloud computing and data intensive computing. Cloud computing and Big Data (main trends, definitions and characteristics). Cloud Computing Models: IaaS, PaaS and SaaS. Cloud deployment models. Dimensions of Big Data: Volume, Velocity, Variety, Vacillation. Big Data Stack: Processing, Storage, Resource Management.
- Storage
  - Distributed File Systems. Google File System (GFS). Master Operations. System Interations. Fault Tolerance. Flat Datacenter Storage (FDS). Databases and Database management. NoSQL. Dynamo. Data Consistency. Big Table. Cassandra.
- Parallel Processing.
  - Programming Languages: Crash course in Scala. MapReduce. FlumeJava. Dryad. Spark and Spark SQL. Project Tungsten.
- Stream Processing
  - Introduction to stream processing. SPS programming model. Data Flow Composition and Manipulation. Parallelization. Fault Tolerance. Distribute messaging System. Kafka. Storm. SEEP. Naiad. Spark Streaming. Structured Streaming. Flink Stream. MillWheel. Google Cloud Dataflow.
- (Large) Graph processing
  - Graph Algorithms Characteristics. Data-parallel vs. Graph-Parallel Computation. Pregel. GraphLab. PowerGraph (GrpahLab2). GraphX. X-Stream. Edge-Centric programming Model. Streaming partitions. Chaos. Storage and computation models.
- Machine learning with Mllib and Tensorflow.
  - Data and Knowledge. Knowledge Discovery from Data. Data Mining Functionalities. Classification and Regression (Supervised Learning). Clustering (Unsupervised learning). MLlib. Tensorflow.
- Resource Management
  - Mesos. YARN.

4.2.1.3. Systems considered in the course

- Data processing
  - Graph Data: Pregel, GraphLab, PowerGraph, GraphX, X-Streem, Chaos
  - Structured Data: Spark, SQL
  - Machine learning: Mlllib, Tensorflow
  - Batch Data: MapReduce, Dryad, FlumeJava, Spark
  - Streaming Data: Strom, SEEP, Naiad, Spark Streaming, Flink, Millwheel, Google Dataflow
- Data storage
  - Distributed File System: GFS, Flat FS
  - NoSQL Databases: Dynamo, Big Table, Cassandra
  - Distributed messaging Systems: Kafka
- Resource Management
  - Mesos, YARN

4.2.1.4. Existing courses

- Data Intensive Computing, Royal Institute of technology, KTH
- Data-intensive Computing Systems, Duke University, https://www.cs.duke.edu/courses/spring15/compsci516/
- Data-Intensive Computing, Illinois Institute of Technology, http://www.cs.iit.edu/~iraicu/teaching/CS554-F13/
- Data-Intensive Scalable Computing, Brown University, http://cs.brown.edu/courses/csci2950-u/f11/
- Data-Intensive Computing, The City University of HongKong, https://www.cityu.edu.hk/ug/201415/course/CS4480.htm
- Data Intensive Computing and Clouds, University of Central Florida, http://www.eecs.ucf.edu/~jwang/Teaching/EEL6938-s12/
- Data Intensive Computing, University of Buffalo, http://www.cse.buffalo.edu/shared/course.php?e=CSE&n=587

4.2.1.5. Existing materials

- Sources for Further Reading
- P. Mell, and T. Grance. The NIST Definition of Cloud Computing *800-145. National Institute of Standards and Technology (NIST), Gaithersburg, MD,* (*September 2011*)
- Armbrust, M.,Fox, A., Griffith, R. , Joseph, A. D. , Katz, R. H. and Konwinski, A., Lee, G., Patterson, D. A. , Rabkin, A., Stoica, I., and Zaharia, M. Above the Clouds: A Berkeley View of Cloud Computing. EECS Department, University of California, Berkeley, Technical Report No. UCB/EECS-2009-28, February 10, 2009
- S. Ghemawat, H.d Gobioff, and Shun-Tak Leung. The Google File System. 19th ACM Symposium on Operating Systems Principles, Lake George, NY, October, 2003
- E. B. Nightingale, J. Elson, J. Dean and S. Ghemawat. Flat Datacenter Storage. USENIX Association 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI '12).
- DeCandia, G., Hastorun, D., Jampani, M., Kakulapati, G., Lakshman, A., Pilchin, A., Sivasubramanian, S., Vosshall, P., and Vogels, W. Dynamo: Amazon's Highly Available Key-value Store. Proceedings of Twenty-first ACM SIGOPS Symposium on Operating Systems Principles, SOSP'07
- Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., Chandra, T., Fikes, A., Gruber, R. E. Bigtable: A Distributed Storage System for Structured Data. ACM Trans. Comput. Syst., 4:1--4:26, 2008
- Lakshman, and P. Malik. Cassandra: a decentralized structured storage system. *Operating Systems Review 44 (2): 35-40* (*2010*)
- J. Dean , S. Ghemawat. MapReduce: simplified data processing on large clusters. OSDI'04: PROCEEDINGS OF THE 6TH CONFERENCE ON SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION
- Chambers, A. Raniwala, F. Perry, S. Adams, R. Henry, R. Bradshaw and Nathan. FlumeJava: Easy, Efficient Data-Parallel Pipelines. *ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI)*, ACM New York, NY 2010, 2 Penn Plaza, Suite 701 New York, NY 10121-0701 (2010), pp. 363-375
- Zaharia, M., Chowdhury, M., Das, T., Dave, A., Ma, J., McCauley, M., Franklin, M. J., Shenker, S, Stoica, I. Resilient Distributed Datasets: A Fault-tolerant Abstraction for In-memory Cluster

Computing. Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation. 2012
- G. Cugola, Alessandro Margara. Processing flows of information: From data stream to complex event processing. ACM Comput. Surv. 44(3): 15:1-15:62 (2012)
- J.-H.Hwang, M. Balazinska, A. Rasin, U. Çetintemel, M. Stonebraker, S. B. Zdonik:
- High-Availability Algorithms for Distributed Stream Processing. ICDE 2005: 779-790
- R. C. Fernandez, M. Migliavacca, E. Kalyvianaki, and P. Pietzuch. 2013. Integrating scale out and fault tolerance in stream processing using operator state management. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data* (SIGMOD '13). ACM, New York, NY, USA, 725-736.
- M. Zaharia, T. Das, H. Li, S. Shenker, and I. Stoica. 2012. Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters. In *Proceedings of the 4th USENIX conference on Hot Topics in Cloud Ccomputing* (HotCloud'12). USENIX Association, Berkeley, CA, USA, 10-10.
- T. Akidau, R. Bradshaw, C. Chambers, S. Chernyak, R. J. Fernández-Moctezuma, R. Lax, S. McVeety, D. Mills, F. Perry, E. Schmidt, and S. Whittle. 2015. The dataflow model: a practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. *Proc. VLDB Endow.* 8, 12 (August 2015), 1792-1803.
- X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, DB Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M.Zaharia, and A. Talwalkar. 2016. MLlib: machine learning in apache spark. *J. Mach. Learn. Res.* 17, 1 (January 2016), 1235-1241.

4.2.2. Part II. Advanced Topics in Distributed Systems

(this syllabus is built using materials from courses developed by Dr. Šarūnas Girdzijauskas and Prof. Seif Haridi - KTH).

The objective of the course is to deeper understand and study the behavior of the networks arising within Distributed Systems. In particular, the course will focus on the concepts of graph theory which will allow to explain the connectivity and dynamics of many real world networks. The course will also cover the topics of Distributed Data Management, Large Graph Processing, Publish/Subscribe Systems, Navigable Small-World Overlays.

The students will also learn how to read and review a scientific paper. The students will have to make one presentation based on a recent research paper.

4.2.2.1. Learning objectives

The students shall after the course:

- be able to describe research areas in the field of distributed systems.
- be able to, through reading research articles, understand a problem description and its solution.
- be able to discuss problem descriptions and proposed solutions.
- have the skill to acquire relevant material to better understand a topic.
- be able to summarize and present a research report.

4.2.2.2. Syllabus and topic descriptions

- Machine learning with Mllib and Tensorflow.
    - Data and Knowledge. Knowledge Discovery from Data. Data Mining Functionalities. Classification and Regression (Supervised Learning). Clustering (Unsupervised learning). MLlib. Tensorflow.

- Introduction
  - Networks: behavior and dynamics. Analysis of Networks. Networks vs. graphs. Complexity of networks.
- Main Concepts
  - Basic definitions. Paths. Cycles. Connectivity. (Giant) Components. Distance
- Network models
  - G(n,m) model. Erdos-Renyi random graph. Random graphs and real world. Watts-Strogatz model.
  - Small-Worlds model. Preferential attachment model. Random walks.
- PageRank, Graph Spectra
  - Convergence of Random Walk. Relation to Web search. Google page rank. Topic specific page rank. Graph spectrum. Spectral graph partitioning.
- Graph Exploration, Navigable Small-World Networks
  - How to explore big networks? Influence of node degree. Milgram's (Small-worlds) experiment. Implications for P2P systems. Navigation in Watts-Strogatz Small-Worlds. Kleinberg's model of Small-Worlds.
- Navigable Structured Overlays, Gossiping Algorithms, Cyclon
  - Small-Worlds based P2P overlays. Approximation of Kleinberg's model. Traditional DHTs and Kleinberg's model. Basic navigation principles. Gossiping algorithms. Cyclon.
- Topology construction by gossiping. Publish/Subscribe Systems
  - Proactive gossip framework. Topology creation. T-man. Publish/Subscribe systems. Topic based publish/subscribe. Content Topic based publish/subscribe. Filter based routing. Overlay-Per-Topic publish/subscribe. SpiderCast. Interest-Aware (greedy) Links. Tera. Rendezvous based publish/subscribe.
- Hybrid Pub/Sub Systems
  - Scribe. Decentralized publish/subscribe. Vitis. Building Navigable Structure.

4.2.2.3. Existing materials

- Advanced Topics in Distributed Computing, at Royal Institute of Technology, KTH
- Network Analysis, University of Michigan, https://www.icpsr.umich.edu/icpsrweb/sumprog/courses/0131
- Social and Information Network Analysis, Stanford center for Professional Development, http://scpd.stanford.edu/search/publicCourseSearchDetails.do?method=load&courseId=7932016
- Social Network Analysis, University of Michigan, Coursera, https://www.class-central.com/mooc/338/coursera-social-network-analysis
- Network Analysis and Modelling, Santa Fe Institute, http://tuvalu.santafe.edu/~aaronc/courses/5352/

4.2.2.4. Further reading

- "Networks, Crowds, and Markets: Reasoning About a Highly Connected World" by David Easley and Jon Kleinberg
- "Networks: An Introduction" by Mark Newman
- "Foundations of Data Science" by J. Hopcroft and R.Kannan

4.2.3. Part III. Scalable Machine Learning and Deep Learning

(this syllabus is built using materials from courses developed by Dr. Jim Dowling - KTH).

The course studies fundamentals of distributed machine learning algorithms and the fundamentals of deep learning. We will cover the basics of machine learning and introduce techniques and systems that enable machine learning algorithms to be efficiently parallelized. The course complements courses in machine learning and distributed systems, with a focus on both the topic of Deep Learning as well as the intersection between distributed systems and machine learning. The course prepares the students for master projects, and Ph.D. studies in the area of Data Science and distributed computing.

The main objective of this course is to provide the students with a solid foundation for understanding large-scale machine learning algorithms, in particular, Deep Learning, and their application areas.

4.2.3.1. Intended learning outcomes

- On successful completion of the course, the student will:
- be able to re-implement a classical machine learning algorithm as a scalable machine learning algorithm
- be able to design and train a layered neural network system
- apply a trained layered neural network system to make useful predictions or classifications in an application area
- be able to elaborate the performance trade-offs when parallelizing machine learning algorithms as well as the limitations in different network environments
- be able to identify appropriate distributed machine learning algorithms to efficiently solve classification and pattern recognition problems.

4.2.3.2. Syllabus and topic descriptions

Main Topics:

- Machine Learning (ML) Principles
- Using Scalable Data Analytics Frameworks to parallelize machine learning algorithms
- Distributed Linear Regression
- Distributed Logistic Regression
- Distributed Principal Component Analysis
- Linear Algebra, Probability Theory and Numerical Computation
- Feedforward Deep Networks
- Regularization in Deep Learning
- Optimization for Training Deep Models
- Convolutional Networks
- Sequence Modelling: Recurrent and Recursive Nets
- Applications of Deep Learning

4.2.3.3. Detailed Content

- Introduction:
    - Brief history and application examples of deep learning, Large-Scale Machine Learning: at Google and in industry, ML background, brief overview of Deep Learning, understanding Deep Learning Systems, Linear Algebra review, Probability Theory review.
- Distributed ML and Linear Regression:
    - Supervised and Unsupervised learning, ML pipeline, Classification pipeline, Linear regression, Distributed ML, Computational Complexity

- Gradient Descent and SparkML:
  - Optimization theory review, gradient descent for Least Squares Regression, The Gradient, Large-Scale ML Pipelines, Feature Extraction, Feature Hashing, Apache Spark and Spark ML.
- Logistic Regression and Classification:
  - Probabilistic Interpretation, Multinominal Logistic Classification, Classification Example in Tensorflow, Quick Look in Tensorflow
- Feedforward Neural Nets and Backprop:
  - Numerical Stability, Neural Networks, Feedforward Neural Networks, Feedforward Phase, Backpropagation
- Regularization and Debugging:
  - A Flow of Deep Learning, Techniques for Training Deep Learning Nets, Regularization, Why does Deep Learning work?
- Convolutional Neural Networks:
  - How Convolutional Neural Nets Work, ConvNets with Depth, Memory Complexity, Case Studies, ConvNets for Everything
- Recurrent Neural networks:
  - Sequence-to-Sequence Learning and Autoencoders, Long Short-term Memory (LSTM), Attention, Autoencoders: Unsupervised Feature Learning,
- Deep Reinforcement Learning:
  - Markov decision Processes, Overview of Reinforcement learning, Supervised Learning vs Reinforcement Learning, Deep RL, Q-learning, Deep Policy Networks, Distributed RL Architecture, Asynchronous RL.
- Case Study:
  - AlphaGo, When will Deep Learning become Intelligent?

4.2.3.4. Existing courses

- Scalable machine learning and Deep Learning, at Royal Institute of technology, KTH
- Scalable Machine Learning, edX, https://courses.edx.org/courses/BerkeleyX/CS190.1x/1T2015/info
- Distributed Machine Learning with Apache Spark, edX https://www.edx.org/course/distributed-machine-learning-apache-uc-berkeleyx-cs120x
- Deep Learning Systems, University of Washington, http://dlsys.cs.washington.edu/
- Scalable Machine Learning, University of Berkeley, https://bcourses.berkeley.edu/courses/1413454/

4.2.3.5. Existing materials

- Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning, MIT Press
- SparkML Pipelines, http://spark.apache.org/docs/latest/ml-pipeline.html
- SparkML Overview, https://www.infoq.com/articles/apache-sparkml-data-pipelines
- SparkML under the Hood, http://spark.tc/machine-learning-in-apache-spark-2-0-under-the-hood-and-over-the-horizon-2/?cm_mc_uid=02948943737214702940997&cm_mc_sid_50200000=1472964414
- Tensorflow, https://www.tensorflow.org/versions/r0.11/tutorials/

4.3 Social media analytics - Fraunhofer

4.3.1. Learning objectives

The participants get a sound overview of social media analysis problems and analysis techniques. Target is the extraction and summarization of user statements from social media sources. The relevant text data mining concepts and deep learning approaches will be introduced. The underlying model assumptions are discussed as far as they are important for the practical application and interpretation of results. Special emphasis is put on the performance evaluation of procedures. To be able to process comprehensive collections specific Big Data implementations will be utilized. Most approaches are demonstrated by stepping through small Python scripts to show the necessary computations. The participants will be able to realistically assess the application of social media analysis technologies for different usage scenarios and can start with their own experiments using different analysis packages with Python.

4.3.2. Syllabus and topic descriptions

Syllabus	**Concepts and methods**
Introduction • Social Media • Media Analytics • Recent success stories	Types of social media User base of social media Relevance of social media analysis Analysis by machine learning Main machine learning paradigms Different types of analysis questions Important applications
Download and preparation • Document formats • Corpora • Google, Twitter, Facebook • Preparation of data	Search engines for document collection crawling packages parsing of different document formats language detection sentence splitting and tokenization part of speech tagging Example scripts
Classify social media posts • Common approaches • Performance evaluation • Application	Definition Different types: logistic regression, support vector machine Overfitting test and performance measures feature selection Big Data approaches: Spark, TensorFlow Example Scripts

Word similarity in posts • Group words in posts • Detecting topics in posts	Unsupervised learning of word properties topic models: Assuming underlying topics word2vec: predicting words in the neighborhood stochastic optimization negative sampling Big Data implementations: Spark, Tensorflow Example scripts
Clustering social media posts • Similarity of post • visualization	Defining similarity measures using topic models and word embeddings Clustering approaches Visualization and specialization Interpretation of results Example scripts
Detecting names, products • Detecting entities • Long range dependencies	Predict properties of words Probabilistic: Conditional Random Field recurrent neural network automatic feature selection word models bidirectional recurrent neural network test and performance measures practical training and evaluation Example scripts
Opinions in social media • Predict user sentiment • opinions and aspects • phrases and their relations	Attaching labels to phrases and sentences Types of opinion mining Aspect based opinions Evaluation Example scripts
Practical social media analytics • Lessons learnt • selecting algorithms • combining approaches	Description of a practical evaluation Combining different approaches and algorithms Evaluation and performance assessment visualization

4.3.3. Existing courses

- Face-to-face training at Fraunhofer: http:/http://www.iais.fraunhofer.de/socialmediaanalytics.html
- https://www.diygenius.com/10-free-online-courses-in-social-media-and-inbound-marketing/
- https://apps.ep.jhu.edu/course-homepages/3523-605.433-social-media-analytics-piorkowski-mcculloh·
- https://www.coursera.org/specializations/social-media-marketing

4.3.4. Sources for further reading

- G.F. Khan Seven Layers of Social Media Analytics (2015): Mining Business Insights from Social Media Text, Actions, Networks, Hyperlinks, Apps, Search Engine, and Location Data. Kindle
- T.Schreck, D.Keim (2013), Visual Analysis of Social Media Data,
- C.C. Aggarwal, C.X. Zhai (2012) Mining Text Data. Springer
- CD Manning, P Raghavan, H Schütze (2008): Introduction to Information Retrieval
- C. Biemann, A. Mehler (2015): Text Mining: From Ontology Learning to Automated Text Processing Apllications
- http://www.iais.fraunhofer.de/data-scientist.html
- https://www.tensorflow.org/
- http://scikit-learn.org/stable/

According to the changes to the workplan from January 19, 2017, the development of learning material for this course was dropped. The quality would be comparable to that of "big data architecture" and "big data analytics", which had not been considered as sufficient for self-learning in the project review in 2016.

4.4 Data Exploitation including data markets and licensing - ODI

4.4.1. Learning objectives

Data science is fundamentally about generating impact from data. Impact is achieved when data acts as a catalyst for change either for social, environmental or economic gain.

At the heart of driving change is finding and telling stories in data, as covered in the finding stories course. The combination of this along with strong communication and leadership skills can help lead to change in business practice in either profitability, market positioning and/or efficiency. Data science can inform and bring evidence for new and changing business models and effective data exploitation can also unlock new and unexpected markets.

In this curated set of resources, we look at how exploiting a world of data can lead to unexpected benefits for businesses and citizens alike. We will guide you through the latest set of resources that can take you through the process of exploiting data science to assist in business innovation.

Curated content in the syllabus will be accompanied with narration to help guide a learner in their interpretation of the resources. Additionally, supporting materials and guidance from those in the community will help explain how data science has opened up new markets.

4.4.2. Syllabus and topic descriptions

Syllabus	**Concepts and methods**
Creating value from data	Managing knowledge assets Managing risk and reward Creating value by sharing The role of open innovation Agglomeration economics and data science
Data means business	Forming new businesses based on data science Diversifying product portfolios with data science Embracing data science in large companies Business approaches data science products and services Challenges for data centric and non-data centric companies
Business models in data	Creating value propositions Freemium / Cross subsidy and Premium business models Razors/Blades and data science Scaling for demand Data science in the cloud
The european data startup hub	The landscape for startups in europe From startup funding to seed funding Pitching a value proposition
Unlocking new markets	Building new partnerships with data science Unlocking a knowledge economy rather than data economy Efficiency and growth Data science and operational insight
Data licensing	Emerging licensing models for data Selling open data Building an open business with a closed offering Exploiting the knowledge economy with data
Building an ecosystem around your data	Publishing open data for commercial standing Competing with the community Creating data ecosystems Exploiting communities and collaboration

Public services and public data	Take advantage of public, free open data Building a business in the public sector Improving public services with data science Competing in a public sector market with open contracting

4.4.3. Existing courses

Data Science for Business, MSc, Postgraduate Diploma, Postgraduate Certificate, University of Stirling [Link]

Using Open Data for Digital Business, University of Holloway, FutureLearn [Link]

Data Science for Business [Data Science as Value Amplifier for CXOs], Persontyle [Link]

Data Business, Irish Management Institute [Link]

4.4.4. Sources for further reading:

How data science projects deliver business impacts: Gartner Report [Link]

Open data means business: UK innovation across sectors and regions. Open Data Institute (2015) [Link]

England Makes 3D Data of the Entire Country Free After Minecrafters Ask For It [Link]

Creating value with identifiers in an open data world, Thomson Reuters [Link]

Our big data challenge, Met Office [Link]

5. Conclusions

In this deliverable we presented the final version of the EDSA curriculum. There were four new courses introduced, but very few changes to existing courses. In addition, a process was developed to ensure that the curriculum is able to adapt to changes in demand, and a preliminary skills framework based on the EDISON skills areas was implemented to assist navigation through the courses offered by EDSA.

The final deliverable in WP2 is D2.6, which is due for release in M36. This will expand further on the implementation of the skills framework, and include further updates of courses implemented, and external courses identified by partners as part of the curation effort for data science.