DATA SCIENCE CURRICULUM 2024

CURRICULUM PART ONE:

THE CORE CURRICULUM

Compiled by: Rugshana Madatt

Gqeberha

South Africa

27 November 2021

Table of Contents

FOREWORD 3
1. The Body of Knowledge: 4
1.1 Analysis and Presentation (AP) 4
1.2 Artificial Intelligence (AI) 9
1.3 Big Data Systems (BDS) 15
1.4 Computing and Computer Fundamentals (CCF) 21
1.5 Data Acquisition, Management, and Governance (DG) 25
1.6 Data Mining (DM) 30
1.7 Data Privacy, Security, Integrity, and Analysis for Security (DPSIA) 37
1.8 Machine Learning (ML) 47
1.9 Professionalism (PR) 54
1.10 Programming, Data Structures, and Algorithms (PDA) 60
1.11 Software Development and Maintenance (SDM) 66

FOREWORD

This Data Science curriculum is compiled into two parts.

PART ONE

This document contains the core curriculum approved by the Association for Computing Machinery (ACM): Tier 1 (T1) , Tier 2 (T2) and Electives (E). We will use Tier 1 for quasi-standardised testing, students will be required to engage with and rewrite Tier 2 in the form of essays and or assignments. Electives will be synthesised into assignments and examples as the syllabus progresses.

PART TWO

Courses done online via NEMISA-Coursera will be converted into NQF credits. Students may be required to write scientific reports based on their certificates from Coursera and other platforms such as Linux Foundation, Udemy, Women Who Code, etc.

OUR DATA SCIENCE CURRICULUM PART ONE IS EXTRACTED FROM THE FOLLOWING SOURCE:

Computing Competencies for Undergraduate Data Science Curricula

ACM Data Science Task Force

January 2021

Andrea Danyluk, Co-chair

Paul Leidig, Co-chair

ISBN: 978-1-4503-9060-6 DOI: 10.1145/3453538

Web link: https://dl.acm.org/citation.cfm?id=3453538

+1-800-342-6626 +1-212-626-0500 (outside U.S.)

orders@acm.org Tel: +1 800 272 6657 Fax: +1 714 821 4641
http://computer.org/cspress csbook@computer.org

Sponsoring Society

This report was made possible by financial support from the following society: Association for Computing Machinery (ACM)

The CCDS 2021 Final Report has been endorsed by the Association for Computing Machinery (ACM)

EXTRACTED FROM APPENDIX A, PAGES 42-122:

1. The Body of Knowledge:

Computing Competencies for Data Science

11 Knowledge Areas (KAs):

1. Analysis and Presentation (AP)
2. Artificial Intelligence (AI)
3. Big Data Systems (BDS)
4. Computing and Computer Fundamentals (CCF)
5. Data Acquisition, Management, and Governance (DG)
6. Data Mining (DM)
7. Data Privacy, Security, Integrity, and Analysis for Security (DP)
8. Machine Learning (ML)
9. Professionalism (PR)
10. Programming, Data Structures, and Algorithms (PDA)
11. Software Development and Maintenance (SDM)

The KAs are further divided into sub-domains. Competencies (with "tiers" – i.e., the recommended level of requirement for a Data Science degree) are given for each.

1.1 Analysis and Presentation (AP)

The human computer interface provides the means whereby users interact with computer systems. The quality of that interface significantly affects usability in all its forms and encompasses a vast range of technologies: animation, visualisation, simulation, speech, video, recognition (of faces, of hand-writing, etc.) and graphics. For the data scientist, it is important to be aware of the range of options and possibilities, and to be able to deploy these as appropriate. Through the use of graphs and other forms of diagrams, visualisation can be used in providing readily understood summaries but can also greatly assist in guiding such activities as clustering and classification.

Scope
● Importance of effectively presenting data, models, and inferences to clients in oral, written, and graphical formats.
● Visualization techniques for exploring data and making inferences, as well as for presenting information to clients.
● Effective visualizations for different types of data, including time-varying data, spatial data, multivariate data, high-dimensional multivariate data, tree- or graph-structured data, discrete / continuous data, and text.
● Knowing the audience: the client or audience for a data science project is not, in general, another data scientist.
● Human-Computer Interface considerations for clients of data science products.

Competencies
● Recognize the main strands of knowledge underpinning approaches to Analysis and Presentation

● Summarize the skills and techniques (including tools) that can be employed in addressing each of the challenges of Analysis and Presentation to create efficient and effective interfaces
● Apply a critical demeanor but also confidence and creativity regarding all aspects of the human computer interface
● Execute the selection of tools appropriate for the size of the data/Big Data to be rendered

Sub domains
AP Foundational considerations – T1
AP Visualization – T1
AP User centered design – T2
AP Interaction design – T2
AP Interface design and development – E 44

AP-Foundational considerations – T1

Presenting data in a suitable form is a challenging but important endeavor. For the data scientist this is fundamentally enabling them to display data in a form that is attractive to users / audiences and readily and appropriately understandable, but is also potentially of great value in providing insights and characteristics including underlying structure. Fundamentally it influences usability.

Knowledge

● Contexts for addressing the human computer interface: visualization of data, web pages, multimedia material, instructional material, the general computing environment paying attention to navigational considerations

● Applicable theories, models, principles, guidelines, and standards for interface design and implementation

● Different measures of effectiveness and attractiveness of an interface ● The use of color and multimedia as well as ergonomics and web services

● Cognitive models that influence interaction

● The scope, advantages, and disadvantages of augmented reality

● Software support to assist with perception regarding analysis and presentation

● Accessibility considerations for different groupings of users ***including those with special needs***

Skills

● Justify the adoption of a user centred approach to analysis and presentation

● Critique how considerations of attention, perception, recognition, speech, movement affect the usability of an interface through a variety of contexts.

● Indicate how formal documents (theories, models, guidelines, etc.) affect the analysis and presentation of data

● Explain the desirable impact of differently-abled users and differently aged groups (including children) on interfaces

● Outline ways in which bias may be perceived in interfaces

● Outline the range of software that can be employed in support of analysis and presentation

● Demonstrate the added value and challenges of an augmented reality interface.

Dispositions

● Passionate and responsible recognition of the vital role of an interface in affecting all aspects of usability

AP-Visualization – T1

Different kinds of data benefit from different approaches to their Visualization. Data scientists need to be aware of this and be familiar with the techniques to be employed in any particular situation.

Knowledge

● The role of visualization in Data Science

● Illustrations, including both historical and contemporary examples, of visualization

● Characteristics of effective visualization

● Suitability of different techniques for different data and for different users

● Dashboards and interactive visualisation

● Software to support visualization

● Inference based on visualization

● Preparing for visualization - scaling, the role of color

● Chart types - tables, scatter plots, pie charts, histograms, graphs, data maps including pixel-, glyph-, graph- and map-based representations

Skills

● Interpret famous examples of visualization in common use

● Identify the various roles that visualization can play in Data Science

● Implement an effective visualization, given a set of data that has to be used for a particular purpose

● Describe the role of visualization in classification and categorization and identify approaches that facilitate this

● Create a variety of visualizations for a variety of data-types in a variety of software.

Dispositions

● Appropriate appreciation of the role of visualization

AP-User-centred design – T2

The fundamental approaches to the design of interfaces that benefit users are explored. Inevitably testing is involved to provide assurances about a successful outcome.

Knowledge

● The user-centred design process

● Relevant life cycle models and standards

● Interaction design patterns, visual hierarchy, navigational considerations

● Identification and capturing of functionality and requirements

● Quality considerations including completeness and consistency and checking for these ● Prototyping

● Design for resource constrained situations (e.g. mobile devices)

● Maintenance considerations ● Relevant software support

Skills

● Discuss a range of approaches to prototyping, including the strengths and weaknesses of the various approaches

● Summarize approaches to checking the consistency and completeness of functionality and requirements given a particular application.

● Discuss the role of statistics in evaluating interfaces

● Identify standards, languages and tools that support the design of high-quality user interfaces

● Diagram the life of an interface, dashboard, or visualization including long-term use and maintenance on a variety of devices

Dispositions

● Professional commitment to the design of high-quality interfaces.

● Inventive attitude towards the design of high-quality interfaces.

● Exhibit business acumen in approaches to designing high quality interfaces.

AP-Interaction Design – T2

It is desirable to review the range of issues that have to be addressed and the techniques that can be employed. Best practices (at the time of their creation) will have been captured in appropriate user interface standards

Knowledge

● The various possible roles of an interface; issues associated with addressing the main possibilities

● Implications of collaborative activity

● Characteristics of high-quality interface design

● Approaches to the evaluation of interfaces including walkthroughs, experiments, heuristics

● Consideration of color, multimedia, speech recognition, animation, touch and gestures

● Data driven applications (with database back end)

● Handling failure, help facilities

● Addressing accessibility considerations

● User interface standards

Skills

● Evaluate the effectiveness of interfaces used for a variety of tasks and a variety of purposes and users

● Identify a national and an international user interface standard and the implications of adherence to them

● Explain the possible implications of collaborative activity on interaction design

● Explain the importance of the design parameters that have to be addressed in creating educational material

Dispositions

● Pro-active in having awareness of the possibilities regarding the computer interface

● Responsive to both national and international user interface standards AP-Interface design and development – E The data scientist has to be able to apply a range of programming techniques to the creation of ever more effective interfaces Knowledge

● Software architecture patterns

● Interaction styles and interaction techniques

● GUI libraries

● Software support including GUI libraries

● Interface animation techniques

● Role of animation and multimedia in interfaces

Skills

● Explain the importance of software architecture patterns and interface design patterns to interface design

● Explain the problems associated with navigation in interface design, and how to address these

● Create a GUI interface for a given data science application

● Explain the considerations in creating an interface for a resource constrained device

● Apply animation techniques in an appropriate manner for a user interface.

Dispositions

● A passionate and responsible approach to interfaces

1.2 Artificial Intelligence (AI)

Artificial Intelligence (AI) includes the methodologies for modelling and simulating several human abilities that are widely accepted as representing intelligence. Perceiving, representing, learning, planning, and reasoning with knowledge and evidence are key themes.

Concepts and methods developed for building AI systems are useful in Data Science. For example, knowledge graphs such as semantic ontologies are both used and generated by data scientists. Computer vision algorithms can be used in analysis of image data; speech and natural language processing algorithms can be applied in analysis of speech or text data. Machine learning algorithms are applied extensively to extract patterns from data. Thus, a student who is well versed in AI will be able to apply those techniques in a Data Science context.

Conversely, Data Science methods are applied extensively in AI systems. Data Science students should have an understanding of AI systems and the way they work, if they plan to apply their work to AI.

Due to their centrality in Data Science, AI competencies related to images, text, and machine learning are highlighted elsewhere. Working with images and text is in the Data Acquisition, Management and Governance KA; Machine Learning is its own KA but is also referenced extensively in the Data Mining KA. This knowledge area addresses knowledge representation, reasoning, and planning.

Scope
● Major subfields of AI
● Representation and reasoning
● Planning and problem solving
● Ethical considerations

Competencies
● Describe major areas of AI as well as contexts in which AI methods may be applied.
● Represent information in a logic formalism and apply relevant reasoning methods.
● Represent information in a probabilistic formalism and apply relevant reasoning methods.
● Be aware of the wide range of ethical considerations around AI systems, as well as mechanisms to mitigate problems.

Sub domains	
AI General – T1, T2 AI Knowledge Representation and Reasoning (Logic based models) – T2, E AI Knowledge Representation and Reasoning (Probability based models) – T1, T2, E	AI-Planning and Search Strategies – T2, E

AI-General

Given the utility of AI approaches for knowledge representation and inference, a data scientist should be aware of their range and history. A data scientist should develop a good sense of existing work in order to know where to look for possible solutions to the full range of possible problems one might encounter.

Knowledge

T1:

● History of AI

● Reality of AI (what it is, what it does) versus perception

● Major subfields of AI: knowledge representation, logical and probabilistic reasoning, planning, perception, natural language processing, learning, robotics (both physical and virtual)

Skills

T1:

● Explain how the origins of AI have led to the current status of AI

● Describe major branches of AI in order to recognize useful concepts and methods when needed in Data Science

T2:

● State what AI systems are and that they both collect and use data to implement AI as well as collect and generate data that can be used by data scientists.

● Describe qualitatively how robots (physical or virtual), agents, and multi-agent systems collect and use data to embed, deliver, or implement artificial intelligence.

● Describe data collected and produced by AI systems that can be useful for data science applications.

Dispositions

T1:

● Astute to, and respectful of, the fact that AI is not a new field, but rather one with a long and rich history.

AI-Knowledge Representation and Reasoning (Logic-based Models)

For certain types of problems, methods of formal logic can be appropriate for representing information and performing inference. A data scientist should be aware of such approaches and know how to map them to inference problems.

Knowledge

T2:

● Predicate logic and example uses

● Automated reasoning: forward chaining, backward chaining

● Reasoning integrated into large-scale systems (e.g., Watson)

Elective:

● Ontologies, knowledge graphs (e.g., protege, ConceptNet, YAGO, UMLS)

● Automated reasoning: resolution, theorem proving

● Languages for automated reasoning

Skills

T2:

● Convert natural language statements to predicate logic.

● Convert predicate logic statements to natural language.

● State example uses and limitations of predicate logic.

● Name example algorithms and/or systems for efficient automated reasoning.

● Describe automated reasoning in a logic-based framework by, for example, forward or backward chaining.

● Name examples of cases where reasoning is integrated into large-scale data-driven systems (e.g., Watson)

Elective:

● Describe a specific method for automated theorem-proving.

● Describe what an ontology is, giving examples of existing technologies, contexts in which they can be used (e.g., question answering), and how they are used (e.g., to aid in disambiguation). ● Describe how ontologies are constructed.

● Implement a medium-sized reasoning problem.

Dispositions

T2:

● Respectful of the benefits and limitations of logic-based representations of knowledge.

● Attentive to the rich history behind formal logic and logic-based algorithms, in order to draw upon them for specific applications.

AI-Knowledge Representation and Reasoning (Probability-based Models)

Probability models lie at the heart of many inference techniques for data science. A data scientist should be aware of a wide range of ways in which information can be modeled in formal probability-based systems. [Note: The items designated T1 in this knowledge area will likely move to a new KA when a joint task force develops complete curriculum guidelines for Data Science.]

Knowledge

T1:

● Fundamental concepts: random variables, axioms of probability, independence, conditional probability, marginal probability. (x-ref Probability, a fundamental knowledge area for DS, not computing discipline-specific)

● Causal models

T2:

● Bayesian networks

● Markov Decision Processes (MDPs)

Elective:

● Reinforcement Learning

● Probabilistic logic models (e.g., Markov logic networks)

Skills

T1:

● Justify the need for probabilistic reasoning.

● Define fundamental concepts such as random variables, independence, etc.

● State axioms of probability.

● Use the above fundamental concepts and axioms to model a simple system and answer questions.

● Describe what causal models are, and how they may be used.

T2:

● State what a Bayesian network is, giving a small- or medium-sized example.

● Demonstrate contexts in which Bayesian networks can be useful (e.g., diagnostic problems).

● Demonstrate how Bayesian networks can be used to make inferences; understand that exact reasoning is intractable in most cases; state examples of approaches for more efficient reasoning (e.g., Belief Propagation).

● Identify independence relationships implied by a Bayesian network.

● State what a Markov Decision Process is, giving a small or medium sized example.

● Demonstrate contexts in which MDPs can be useful (e.g., optimization or control problems).

● Demonstrate how MDPs can be used to make inferences.

Elective:

● Construct a Bayesian network for a small- or medium-sized problem.

● Apply a learning algorithm to construct a Bayesian network for a small- or medium-sized problem.

● State how the parameters of a MDP can be learned. Give examples of algorithms that can be used to do so.

● Apply a reinforcement learning algorithm to an appropriate problem.

● State examples of probabilistic logic models, such as Markov logic networks, identifying applications for which they are useful.

● Apply a probabilistic logic model to a small- or medium-sized problem.

Dispositions

T1: ● Respectful of the benefits and limitations of probability-based representations of knowledge and methods for performing inference over them.

AI-Planning and Search Strategies

Beyond representing and reasoning about the world, AI methods allow for planning a step-bystep solution and then carrying it out. A data scientist should be aware of these techniques in order to apply data-driven methods to improve performance or to understand how to gather data from such systems. Note that while several of the methods included here (e.g., breadth- and depth-first search) also appear in the KA on Programming, Data Structures, and Algorithms.

Knowledge

T2:

● State space representation of possible solutions to a problem

● Breadth- and depth-first (i.e., uninformed) search of a state space

● Heuristic (i.e., informed) search of a state space (e.g., A* search) Elective:

● Stochastic search algorithms (e.g., genetic algorithms, simulated annealing)

● Constraint satisfaction problems and methods

Skills

T2:

● Explain how a solution to a problem can be viewed as a state in a space of possible solutions (e.g., assignments of values to variables).

● For a given problem, produce a model of it as search in a multidimensional state space.

● Explain how breadth- and depth-first search can be used to search a space of solutions modeled as a graph.

● Explain how heuristics can be used to (potentially) speed up graph/state space search.

Elective:

● Apply uninformed search to find a solution to a problem modeled as a state space (where the graph representing the space is likely developed as the search is performed, rather than provided as input).

● Design a heuristic for a small problem.

● Apply an informed search approach to a small- or medium-sized problem.

● Apply a stochastic search approach to a small- or medium-sized problem.

● Explain how a stochastic search algorithm addresses issues of exploring a space (e.g., avoiding local minima); explain how a stochastic search algorithm addresses local search in a space of promising solutions.

● Explain how the solution to a problem may involve specific constraints on particular variables as well as their relationships to each other; describe methods for articulating these constraints.

● Implement search algorithms.

● Formulate a model for a small problem as a constraint satisfaction problem.

● Apply a constraint-satisfaction algorithm to a small- or medium-sized problem.

Dispositions

T2:

● Respectful in understanding that there may be multiple acceptable solutions in a state space, as well as multiple ways to find them. Using judgement to evaluate different solutions or problem-solving approaches, depending on external conditions, such as the need for optimality, time constraints, etc.

● Adaptable in utilizing the relationship between algorithm, heuristics, and optimality for designing a solution to a problem.

1.3 Big Data Systems (BDS)

The term ‘Big Data’ has been coined to describe systems that are truly large; these might include, for instance, files of videos, images, handwriting, etc. that cannot be accommodated on a single server. Such systems introduce problems of scale: how to store vast quantities of data, how to be certain the data is of high quality, how to process that in ways that are efficient and how to derive insights that prove useful.

These matters are addressed below under the headings of problems of scale, data storage, high performance computing, and complexity theory. These topics include a range of techniques typically used in addressing the problems of scale. Such systems can be complex and so consideration is given also to software support for Big Data applications.

Scope
● Problems of scale and the implications of Big Data on computation requirements
● Theoretical and methodological issues employed in the context of Big Data
● Appropriate algorithms to harness the processing power of the cluster ● Approaches to simplifying the programming interface used in developing Big Data applications
Competencies
● Describe the main strands of knowledge needed to address Big Data applications, highlighting areas where collaboration is desirable
● Provide familiarity with a range of skills that may be used in the implementation of Big Data applications
● Instil confidence in dealing with the problems of Big Data

Sub domains	
BDS Problems of Scale – T1 BDS Big Data Computing Architectures E BDS Parallel Computing Frameworks E BDS Distributed Data Storage – T2, E BDS Parallel Programming – T2 BDS Techniques for Big Data Applications – T2	BDS-Cloud Computing – T2 BDS-Complexity Theory - E BDS-Software Support for Big Data Applications – T2

BDS Problems of Scale – T1

The computational problems associated with managing and processing very large amounts of data typically increase as the amount of data increases. Measurement provides insights into the rate of increase and the attendant computational consequences.

Knowledge

● The need for measurement in the context of Big Data, including size, capacity and timing

● The concept of the size of a problem

● Consequences of rapid rate of growth considerations for computation ● Storage consequences of rapid rate of data growth

● The need to place an emphasis on simplicity

● Approaches to addressing the problems of coordination with increasing numbers of agents / processes

● Approaches to addressing the problems of scale while accommodating scalability

Skills

● Outline reasons for Big Data applications leading to increased complexity, and give guidance on the nature of that complexity

● Justify the importance of placing an emphasis on simplicity, though not excessive simplicity

● Describe steps that can typically be taken to reduce complexity

● Evaluate data scale and speed for applications according to the descriptions

● Execute a computational task at multiple scale-levels successfully Dispositions

● Adjust in response to changing difficulties created by scale

● Act appropriately in addressing problems of scale

BDS-Big Data Computing Architectures – E

An historical perspective suggests the former existence of two communities: one engaged in I/O intensive activities, the other engaged in compute intensive applications. The systems (preferred hardware and software) used by these communities were largely separate and customised to meet their needs.

Recent developments, e.g. those involving advances in machine learning and deep learning, have tended to bring about a convergence of these communities with them now sharing all the facilities.

Knowledge
● Mechanisms that support fast and efficient input / output
● The concepts and requirements of data-centric high-performance computing 56
● Memory considerations: cache considerations including cache coherence
● The various parallel computing architectures, their strengths and their limitations: multicore, grid computing, GPUs, shared memory, distributed memory, symmetric multiprocessing, vector processing
● Flynn's taxonomy
● Instruction considerations in support of parallelism
● Parallel storage hierarchy
Skills
● Identify approaches to achieving fast input / output
● Explain the nature of any impediments to achieving fast input / output ● Compare and contrast the various parallel computing architectures
● Describe the nature of the applications to which the various parallel architectures are best suited
● Choose the system architecture that best suits a particular computation model and framework as captured in the computation patterns and data features Dispositions
● Thoroughness in addressing hardware issues in support of Data Science applications

BDS-Parallel Computing Frameworks – E

Important high-level support is provided through parallel computation models for the generation of parallel programs.
Knowledge
● Definition and purpose of a parallel computation model
● Classification of models
● Distributed systems
● Grid search
● Process interaction: issues of communication and coordination
● Problem decomposition: task based decomposition, data-parallel decomposition
Skills
● Diagram a parallel computation system ●
● Evaluate a parallel computation's planned process interactions and problem decomposition. for efficiency and effectiveness ● Outline the design and deployment of large-scale data processing parallel systems
Dispositions
● Astute in evaluating or designing potentially complex systems

BDS-Distributed Data Storage
Big Data applications benefit from approaches to data storage that are scalable, accommodate vast amounts of data, possibly straddling various machines, and yet facilitating processing within an appropriate time frame.
Knowledge
T2:

● Approaches to storing vast quantities of data, including storage across a range of devices
● Storage hierarchies
● Ensuring clean, consistent and representative data
● Protecting and maintaining the data
● Retrieval issues
● The benefits and limitations of a range of techniques used in addressing the problems of scale such as hashing, filtering, sampling
● Data backup
Skills
T2:
● Explain the role of the storage hierarchy in dealing with Big Data
● Outline advantages of certain kinds of redundancy in Big Data
● Demonstrate how unwanted redundancy may be removed efficiently from a Big Data set
● Describe approaches to protecting and maintaining data for a Big Data application, ensuring that it remains current and useful
Elective:
● Develop a distributed data storage system, choosing and producing arguments that support mechanisms that will scale
● Design storage systems with related strategies such as backup, migration and compression for data-centric systems to ensure scalability, usability, efficiency and security
Dispositions
T2:
● A strong commitment to the design of storage mechanisms in support of Big Data applications

BDS-Parallel Programming – T2
Parallel programming, whereby several activities may take place simultaneously, is an important approach to increasing the efficiency of programs. Novel forms of programming constructs are 58 required to support this. In practice, new kinds of programming errors may result and there are limitations to the efficiencies that can be achieved. Knowledge
● Concurrency, parallelism, and distributed systems
● Limitations of parallelism including the overheads
● Parallel algorithms and how they best fit particular hardware architectures; load balancing issues
● Typical parallel programming paradigm such as MapReduce
● Complexity of parallel / concurrent algorithms Skills
● Explain the limitations of concurrency / parallelism in dealing with problems of scale
● Identify the overheads and computational complexity associated with parallelism in particular algorithms
● Implement methods for data-centric parallel programs
● Construct data-centric parallel computation systems according to the data scale and data operations
● Develop optimized data-centric parallel programs
● Formulate well-tuned algorithms within a parallel programming paradigm
● Evaluate a parallel algorithm’s load-balance on a variety of hardware architectures

Dispositions
● Attention to detail in factoring in that the overheads of parallelism can become excessive in particular cases
● Astute in dealing with parallel systems in appropriate cases

BDS-Techniques used in Big Data applications – T2
A number of techniques have been devised and, if deployed carefully, have proved valuable in increasing the efficiency of application programs.
Knowledge
● The need for techniques to assist with handling Big Data
● Hashing, Sampling, filtering and their limitations
● Data sketch and synopsis
Skills
● Illustrate the role of hashing in dealing with Big Data
● Explain a range of criteria that may be used in guiding sampling and filtering
● Perform sample selection, to conform to given guidelines, for a particular application involving Big Data
● Critically review a variety of approaches to filtering, illustrating their use 59
● Design a data sketch and synopsis structure according to the available space and permit accuracy loss, and analyze the performance

Dispositions
● Be attentive of pitfalls such as bias in performing sampling and filtering

BDS-Cloud Computing – T2

The Cloud offers a number of advantages (over clusters, for instance) in the context of Big Data. It is important to understand these and be able to exploit them effectively; they include web services.
Knowledge
● The nature of Cloud Computing and its advantages
● The architecture of a data center
● Risks associated with Cloud Computing
● Different approaches to supporting Cloud Computing
● Distributed file-systems
● Cloud Services in support of Big Data applications ● Virtualization technology
● Security issues for cloud including cloud computing, cloud storage and virtual machines
Skills
● Outline the main tasks performed by a cloud-based system
● Design a data center
● Identify the range of Cloud Services typically supplied in support of Big Data applications
● Apply Cloud Services that support particular Big Data applications
● Design security strategies for cloud
● Explain distributed file-systems and virtualization technology Dispositions
● Act appropriately when using Cloud Services Context dependencies
● Different sets of Cloud Services are available, for instance, from Amazon, Google, Microsoft

BDS-Complexity Theory – E

An understanding of how to measure the efficiency of Big Data algorithms, both sequential and parallel, as well as the theoretical limitations to efficiency underpin an informed approach to applications

Knowledge

● Problems of computation and the efficiency of algorithms
● The notion of computational complexity, its use in the context of concurrency / parallelism and its importance in the context of Big Data
● Limitations to the concept of complexity
● Evaluation of the complexity of a range of commonly used algorithms including those exhibiting concurrency / parallelism

Skills

● Explain why mathematical analysis alone is not always sufficient in dealing with efficiency considerations
● Analyze whether the problem could be solved or solved approximately with some ratio bound from the aspect of complexity, given a problem description with data size, time constraints and resource constraints,
● Demonstrate how to evaluate the efficiency of an algorithm to be used in processing Big Data
● Select algorithms appropriate to a particular application involving Big Data, taking account of the problems of scale

Dispositions

● Conviction and initiative in dealing with complexity
● Responsive to the fact that there may be limits to complexity gains

BDS-Software Support for Big Data Applications – T2

Having access to a suite of high-quality software tools that can be deployed and work together effectively can simplify the task of processing large data sets and elevate thinking away from detail and towards greater insight and innovation.

Knowledge

● The need for programming environments to support Big Data applications and the nature of these
● Concepts of auto scaling and serverless computing
● Review of the availability of sophisticated web services for the support of data movement, analytics and machine learning in the context of Big Data

Skills

● Compare and contrast the use of auto scaling and serverless computing
● Identify the relationship between load balancing and auto scaling 61 Dispositions
● Ethical approach to the use of web services including possible bias, and other such deficiencies
● Independent in dealing with Big Data applications
● Attentive to simplicity, but not excessive simplicity, in the context of Big Data applications 62

1.4 Computing and Computer Fundamentals (CCF)

Modern Data Science relies heavily on computing and on computing devices: to gather and store data; to analyze data; to present analyses and conclusions; and to field systems based on analyses and results. Therefore, a Data Scientist should understand -- at least at a high level -- the structure of operating systems, file systems, compilers, and networks, as well as security issues related to them.
Note that many of the competencies in this KA are taken or adapted from CS2013.
Note also that the majority of competencies in this knowledge area are intended to indicate highlevel understanding and appreciation of concepts, rather than deep technical understanding.

Scope
● Digital representation of data
● Processors
● Memory management
● Operating system functions and vulnerabilities
● File organization
● Network structure and communication
● Web programming
● Compilers vs interpreters

Competencies
● Appreciate ways in which digital representations of data affect efficiency and precision
● Recognize that there are different types of processors and configurations of them
● Understand the trade-offs between expensive/fast memory and inexpensive/slower memory
● Summarize the important role of an operating system and the ways in which it is both vulnerable to and can be protected from attack
● Carry out the creation, organization, and protection of files
● Understand at a high level how networks are organized and transmit information
● Recognize the web as an application layer on the internet
● Use the web to gather information and build useful applications
● Understand that while compilers and interpreters are both translators of code, they have relative benefits and limitations

Sub domains	
CCF Basic Computer Architecture – T1, T2	CCF-File Systems – T1, T2
CCF Storage System Fundamentals – T1	CCF-Networks – T1, T2
CCF Operating System Basics – T1, T2	CCF-The Web & Web Programming – T1, T2
	CCF-Compilers and Interpreters – T1

CCF-Basic Computer Architecture

A data scientist will benefit from understanding the ways in which digital representations of data affect precision, as well as the ways that different processor types and configurations can affect the efficiency of computation.
Knowledge
T1:
● “Power wall”
● Bits, bytes, and words
● Representation of numeric data
● CPUs and GPUs
T2:
● Representation of non-numeric data
● Multi-core and multi-processing
● Basic organization of the von Neumann machine
● Parallel architectures (e.g., SIMD, MIMD)
Skills
T1:
● Explain the implications of the “power wall” in terms of further processor performance improvements and the drive towards harnessing parallelism.
● Explain how fixed-length number representations affect accuracy and precision. [x-ref KA: Programming]
● Describe the role of CPUs; compare and contrast with the specialized purpose of GPUs.
T2:
● Describe the internal representation of non-numeric data, such as characters, strings, and images.
● Describe the difference between multi-core and multi-processor systems.
● Explain the organization of the classical von Neumann machine and its major functional units.
● Discuss the concept of parallel processing beyond the classical von Neumann model.
Dispositions
T1:
● Respectful of the benefits and limitations of data representation and processor speed in modern computing devices.

CCF-Storage System Fundamentals – T1

In contexts where data scientists are analyzing large quantities of data, they will benefit from knowing how those data are stored and moved during processing. This may be of help both in understanding the time needed to complete large analyses as well as in selecting hardware infrastructure and configurations to enable such work.
64
Knowledge
● Storage systems and their technology
● Registers, Cache, RAM
● Virtual memory Skills
● Identify major types of memory technology (e.g., SRAM, DRAM, Flash, magnetic disk) and their relative cost and performance.
● Describe how the use of memory hierarchy reduces effective memory latency.
Dispositions

● Proactive and adaptable regarding the trade-off between expensive/fast memory and less expensive/slower memory.

CCF-Operating System Basics

Given the important considerations of security and privacy in data science analyses and applications, the data scientist will benefit from a high-level understanding of operating systems and the ways in which they are vulnerable to attack.
Knowledge
T1:
● Role and purpose of an operating system
● Types of security threats and mitigation approaches
T2:
● Networked, client-server, and distributed operating systems
● Reliability and availability

Skills
T1:
● Describe the objectives and functions of modern operating systems.

● List potential threats to operating systems (e.g., software vulnerabilities, authentication issues, malware) and the types of security features designed to guard against them.
T2:
● Discuss networked, client-server, and distributed operating systems and how they differ from single-user operating systems.
● Discuss the importance of computer reliability and availability; describe methods of fault tolerance for ensuring both.
Dispositions
T1:
● Respectful of the important role of operating systems in providing an interface between humans and system resources as well as between system resources;
● Act appropriately to avoid operating system attacks

CCF-File Systems
File systems provide the mechanism by which data and programs are organized. A data scientist should be aware of how individual files are stored, how they are organized in relationship to each other, and how they can be protected for purposes of security and privacy. A data scientist should know how to select the appropriate file system for the size of the data to be accommodated (e.g., for Big Data, a local file system on a single server would not be a good choice).
Knowledge
T1:
● Files: data, metadata, operations, organization
● Directories: contents and structure
● File protection
T2:
● Files: sequential, nonsequential
Skills

T1:
● Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each.
● Describe levels of file protection and mechanisms for setting them.
T2:
● Compare and contrast sequential and non-sequential file storage. Dispositions
T1:
● Attentive to the importance of good file organization as well as the importance of protecting files from inappropriate access.

CCF-Networks

Data and applications are shared over computer networks. Knowing how they work is helpful for understanding the ways in which data and applications are vulnerable to the introduction of errors, loss of information, or attacks, as well as the ways in which data and applications may be protected from those. In addition, knowledge of networks is important to understand cloud systems, Big Data clusters, and performance.

Knowledge

T2:
● Components of networks: hosts, routers, switches, ISPs, wireless access points, firewalls
● Local area networks; LAN topology (e.g., bus, ring)
● Organization of the Internet: Internet Service Providers (ISPs), Content Providers, etc.
● Circuit- vs packet-switched networks
● Layered network structure
● Naming and address schemes (DNS, IP addresses, Uniform Resource Identifiers, etc.)
● Basic protocols: TCP, IP
● HTTP / HTTPS as application-layer protocols

Skills

T2:
● List major components of standard computer networks
● Recognize that LANs can be organized in a variety of topologies.
● Discuss (at a high level) the organization of the Internet
● Explain the difference between circuit- and packet-switching
● Describe the layered structure of a typical networked architecture
● List the differences and relations between names and addresses in a network
● Describe how basic protocols such as TCP and IP work
● Describe how application-layer protocols such as HTTPS work Dispositions
T1:
● Use discretion concerning the complexity of transmitting information over a network, as well as the mechanisms for mitigating issues that can arise during transmission.

CCF-The Web and Web Programming

Data are frequently obtained via web applications. A data scientist should be able to write and use web applications, as well as appreciate the potential pitfalls of doing so.

Knowledge
T1:
● Relationship between Internet and World Wide Web
● Awareness of web application vulnerabilities and security attacks (e.g., SQL injection, Distributed Denial of Service Attacks)
T2:
● Security attack detection and mitigation
Skills
T1:
● Describe the relationship between the Internet and the World Wide Web ● Design and implement a simple browser-based application
● Describe common web application vulnerabilities and security attacks
● Web programming languages (e.g., HTML5, Java Script, PHP, CSS)
T2:
● Identify and apply methods to protect against security attacks Dispositions
T1:
● Be accurate in avoiding the potential security risks of writing and using web applications in order to do both as securely as possible. 67

CCF - Compilers and Interpreters – T1

Whether for purposes of gathering data, doing analysis, or fielding applications based on analyses, data scientists use and write software. Appreciating the purpose of and differences between compilers and interpreters can be useful in selecting programming languages and tools.
Knowledge
● Programs that take (other) programs as input: interpreters, compilers, type-checkers, documentation generators
● Interpretation vs. compilation to native code vs. compilation to portable intermediate representation
● Syntax and parsing vs. semantics and evaluation
● Examples of languages that fall into interpreted vs. compiled categories Skills
● Explain how programs that process other programs treat the other programs as their input data
● Discuss advantages and disadvantages of interpreted vs compiled code
● Distinguish syntax and parsing from semantics and evaluation
● Identify interpreted and compiled languages
Dispositions
● Accurate evaluating the speed trade-offs of interpreted vs compiled code.
● Accurate evaluating the flexibility trade-offs of compilation to native code vs portable intermediate representations.
● Using judgement evaluating the utility of interpreters during code development.

1.5 Data Acquisition, Management, and Governance (DG)

As the base of data science, data should be acquired, integrated and pre-processed. This is an important step to ensure both quantity and quality of data and improve the effectiveness of the following steps of data processing. Thus, a data scientist must understand concepts and approaches of data acquisition and governance including

data shaping, information extraction, information integration, data reduction and compression, data transformation as well as data cleaning.
In our ever-increasing reliance on the quantity and quality of data in all forms of decision making, the data scientist has an ethical responsibility of protecting the integrity of data and proper use of data.

Scope
● Shaping data and their relationships
● Acquiring data from physical world and extracting data to a form suitable for analysis
● Traditional Data Integration Methods: Pattern Mapping, Data Matching, Entity Recognition
● Integrating heterogeneous data sources
● Pre-processing and cleaning data for applications
● Improving data quality
● Ensuring data integrity including privacy and security

Competencies
● Construct a data governance process according to the requirements of applications, including data preparation algorithms and steps. (Process Construction and Tuning)
● Write semantics rules for data governance, including information extraction, data integration and data cleaning (Rules Definition)
● Develop scalable and efficient algorithms for data governance according to the requirements of applications (including data extraction, integration, sampling, reduction, data compression, transformation and cleaning algorithm (Algorithm Development)
● Diagram the static and dynamic properties of data, changing mechanisms of data and similarity between data. (Property Description and Discovery)
● Develop policies and processes to ensure the privacy and security of data.

Sub-domains	
DG-Data Acquisition – T1, T2 DG-Information Extraction –T1, T2 DG-Working with Various Types of Data – T2	DG-Data Integration – T1 DG-Data Reduction and Compression – T1, T2 DG-Data Transformation – T1 DG-Data Cleaning – T1 DG-Data Privacy and Security – T1

DG-Data Acquisition – T1
As the initial step in data governance policies, data acquisition is the process of obtaining raw data from real-world objects. The process of data acquisition should fully consider the physical properties of the subject, and at the same time consider the characteristics of the data application.
Due to the limited resources available during data acquisition (such as network bandwidth, sensor node energy, website tokens, etc.), it is necessary to effectively design data collection techniques to maximize valuable data within limited resources and minimize valueless data. Also due to resource constraints, the data acquisition process is unlikely to obtain all the information of the data description object, so the

data acquisition technology needs to be carefully designed to minimize the deviation between the collected data and the real objects.
Knowledge
● The sources of data
● Pull-based and push-based approaches
● Various data acquisition with the features of acquired data
● Data acquisition acceleration techniques
● Data discretization method
● Security and Privacy standards and best practices
Skills
T1:
● Select data source for the applications
● Design techniques for data acquisition according to the features of data sources and applications.
● Plan following steps including data discretization, transmission as well as storage to ensure security, privacy, and effective use.
T2:
● Design the acceleration and parallelization strategies for data acquisition according to the applications
Dispositions
● Show business acumen in the ability to assess the trade-off between accuracy and efficiency in data acquisition.

DG-Information Extraction – T2

Information extraction (IE) is the task of automatically extracting structured information from unstructured and/or semi-structured machine-readable documents. It is an important technique to acquire data from documents, web pages, and even multimedia. Information Extraction is relevant to the requirements of data acquisition and governance, but is described elsewhere in this report. See Information Extraction in the Data Mining KA.

DG-Working with Various Types of Data – T2
Data comes in many forms. Some projects will rely completely on numeric data. Others will require processing of text or image or other media data. The data scientist must have an overview of all types of data representation and processing, and must be competent to interact with some types of data as an expert.
Knowledge
● Data representation: numbers, text, images, data precision
● Text data processing: bag-of-words, word-count, TF-IDF, n-grams, Lexical analysis, syntax analysis, semantic analysis, stop word filtering, stemming, basic applications
● Image processing: data representation: multi-dimensional matrices of integers, features, image operators, video operators. Object recognition. Higher order feature extraction

Skills
● Write programs to perform basic operations on data of each type: compute summary statistics, extract n-grams, do modifications to an image, etc.
Dispositions

● Accurate in the choice of data type for encoding information.

DG-Data Integration – T1

In the data acquisition process, since the data may come from an autonomous data source, it is difficult to ensure the consistency of the data mode, modality, semantics, etc.. However, in many applications, these data from multiple autonomous data sources need to be summarized and used together to generate new value, this is the task of data integration, which is a crucial step for data acquisition and governance.
Knowledge
● The concepts and application scenarios of government database, data warehouse and mediator-based information integration
● The concepts and approaches of schema mapping
● The concepts and approaches of data mapping
● The concepts and approaches of data semantic transformation
● The techniques of cross-domain data integration
Skills
● Choose the scheme of data integration, i.e. traditional data integration VS. cross-domain data integration
● Choose the architecture of data integration according to the features of applications
● Select or develop appropriate algorithms for schema mapping, data mapping and data semantic transformation
● Develop proper algorithms for cross-domain data integration

Dispositions
● Astute about the challenges brought by heterogeneous data sources
● Astute about the roles of AI in data integration DG-Data Reduction and Compression The goal of data reduction and compression is to eliminate the redundancy of data and decrease the size of data involved in the next data processing steps. This involves data sampling, filtering and compression.
Knowledge T1:
● The role of reduction and compression in data process
● Various data sampling approaches
● Data filter techniques
● Data compression techniques Skills
T1:
● Examine whether data reduction and compression steps are required
● Perform data sampling and filtering
T2:
●Analyse the properties of data sampling
● Select data compression techniques according to the computation, communication and storage requirements
● Develop query-friendly data compression approach
Dispositions
● Attention to detail evaluating the trade-off between data computation effectiveness and efficiency.

DG-Data Transformation – T1

Data collected from data sources often have different dimensions and ranges. These data may be correct, but they cannot be directly used. It is often necessary to transform the collected data and convert the data into "appropriate" form to understand the data or visualize the data to achieve effective application of the data.

Knowledge

● Data Transformation pipeline
● Simple function transformation methods and their applications
● Data standardization and its applications
● Data normalization and its applications
● Data encoding approaches and their applications
● Data smoothing approaches and their applications

Skills

● Evaluate and compare the dimension and range of data and those of the requirements in the applications.
● Determine the process of data transformation
● Choose proper data algorithms for the task
● Evaluate the effectiveness of data transformation Dispositions

● Astute about the importance of data transformation to data usage
● Astute about the links between data transformation and data quality

DG-Data Cleaning – T1

Data quality is an important aspect of data usability. There is a perception that if data is “suitable for its intended use in operations, decision making, and planning,” it is generally considered to be of high quality. There are also views that if the data correctly represents the real-world entities that it refers to, then it is also considered to be of high quality. Data quality issues and the resulting knowledge and decision-making mistakes have had terrible consequences on a global scale. Data cleaning is an important solution for data quality problems. Knowledge

● The dimensions of data quality
● The approaches to improve data quality
● Data cleaning algorithms including entity resolution, truth discovery, rule-based data cleaning.
● Various forms for data quality rules such as functional dependencies (FD), conditional functional dependencies (CFD), conditional inclusion dependencies (CIND), and matching dependencies (MD) Skills
● Evaluate data quality
● Write rules for data cleaning according to the requirement of applications and data semantics
● Develop a data cleaning pipeline according to the data quality requirements.
● Develop algorithms for efficient and effective data cleaning Dispositions
● Astute about the harm of data quality problems
● Strong commitment to handle the role of data cleaning in data usage.

DG-Data Privacy and Security – T1

Knowledge

● The relationships between individuals, organizations, or governmental privacy requirements
● The cross-border privacy and data security laws and responsibilities

● A comprehension of how organizations with international engagement must consider variances in privacy laws, regulations, and standards across the jurisdictions in which they operate.

Skills
● Explain how laws and technology intersect in the context of the judicial structures that are present – international, national and local – as organizations safeguard information systems from cyberattacks.
● Explain requirements of the General Data Protection Regulation (GDPR), and Privacy Shield agreement between countries, such as the United States and the United Kingdom, allowing the transfer of personal data.
● Describe how [Section 5 of the U.S. Federal Trade Commission, State data security laws, State data-breach notification laws, Health Insurance Portability Accountability Act (HIPAA), Gramm Leach Bliley Act (GLBA), and Information sharing through USCERT, Cybersecurity Act of 2015] and other laws impact data security
Dispositions
● Act ethically in data governance policies and actions
● Accurate about the harm of data loss due to security and privacy failures
● Maintain the utmost ethical standards regarding legal and social responsibility for data

1.6 Data Mining (DM)

At its core, Data Mining involves the processing, analysis, and presentation of data in order to gain valuable information. An important prerequisite is that appropriate data of a high quality has been prepared and is relevant to the task at hand.
The basic types of analysis include clustering, classification, regression, pattern mining, prediction, association and outlier detection with attention being given to various forms of data including time series data and web data. Many of these concepts depend on the notion of data proximity.

Scope
● Data mining and its relationship to data preparation and data management
● Data mining models for a variety of data types and applications
● Selection and application of data mining algorithms for various tasks
Competencies
● Equip students with knowledge about the range of techniques available for mining data as well as the related algorithms and their suitability
● Equip students with the ability to identify and use tools and techniques for mining data which may exist in various forms
● Engender in students a high level of well-founded confidence in mining data

Sub domains

DM-Proximity Measurement – T1, T2 DM-Data Preparation – T1 DM-Information Extraction – E DM-Cluster Analysis – T1, T2 DM-Classification and Regression – T1, T2, E	DM-Pattern Mining – T2 DM-Outlier Detection – T2 DM-Time Series Data – E DM-Mining Web Data – T2 DM-Information Retrieval – T2

DM-Proximity Measurement
Various possibilities exist for measuring differences as well as similarities among data points.

For numerical data the methods are typically phrased in terms of distance between two vectors. But measures for other types of data may include different notions of proximity (such as cosine similarity for text) or correlation. Special definitions may be needed, customized to particular situations. Knowledge
T1:
● Basic properties of metrics

● Lk measure; special cases – Euclidean distance, Manhattan distance
● Use of scores and rankings; desirable characteristics of scores and ranking regimes
● Normalization of data to support comparison 75 T2:
● Metrics involving text
● Metrics such as correlation coefficient for sequences of data
● Metrics such as SimRank for similarity based on relationships, as in graphs
● Graph based metrics
● Metrics for measuring the similarity of time series, e.g. dynamic time warping
Skills
T1:
● Describe and compare measurement concepts and their relevance to different kinds of data – nominal, ordinal, interval and ratios
● Select metrics appropriate for comparison of various kinds of data
Dispositions
T1:
● An accurate, yet inventive, approach to the use of scores and metrics recognizing that typically many approaches exist

DM-Data Preparation – T1
The availability and preparation of high-quality data is essential to data science. There is the initial gathering of relevant data, possibly from a wide variety of sources, and then ensuring the data set is fit for purpose. Knowledge
● Gathering data, its relationship to problem solving, importance of expert knowledge and being open to the views of experts ● Sources of data including databases, the Internet of Things, photographs and videos, online information sources; adequacy of data for particular purposes
● Ethical considerations around obtaining and using data for particular purposes; privacy concerns around collocating data; concerns around potential bias in data

● Munging data - dealing with errors in data, gaps in data, cleansing data, validating data, profiling data, transforming data, and joining datasets as appropriate; quality considerations
● Methods of dealing with dataset issues such as imbalance, insufficient or extraneous attributes; automated and manual approaches and trade-offs between these
● The concept of a ‘feature’; feature extraction and representation; feature selection and feature generation
Skills
● Illustrate the connection between the process of framing a question with the process of obtaining data to answer the question.
● Demonstrate expertise in a particular domain by interacting appropriately with experts.
● Use summary statistics and visualizations in exploratory data analysis to make inferences.
● Illustrate the impact and resolution of issues that may arise with datasets 76
● Explain the benefits and implications of various methods of generating features.
● Describe the similarities and differences between feature selection and feature generation
● Demonstrate how feature generation can result in fewer or more features.
Disposition
● Accurate in the selection and preparation of data as well as an understanding of the importance of dealing with quality data.

DM-Information extraction - E

Information extraction (IE) is concerned with the techniques and processes used to extract structured information from unstructured data that exists in different forms. It is an important technique used to acquire data from documents, web pages and even multimedia.
Knowledge
● Applications where information extraction plays a useful role
● Entity and relation extraction
● Rule-based information extraction approaches and their applications
● Statistics-based information extraction approaches and their applications
● The possible problems in the extracted data Skills
● Design a schema according to the application requirements and data
● Write information extraction rules for an application using both rule-based and statisticsbased approaches
● Apply learning algorithms for information extraction tasks such as rule or model learning and relationship prediction
Disposition
● Astute that there are various approaches to extracting information from data.

DM-Cluster Analysis
Clustering involves grouping together data points that exhibit some element of similarity. This implies some interpretation of proximity and there can be various interpretations of that. Clusters in 2- or 3-dimensions can often be identified on the basis of visualization but that is not always readily available especially in higher

dimensions. Generally, clusters may be compact and well separated but again this is not always the case. (See also ML-Unsupervised Learning.)
Knowledge T1:
● Identification of appropriate similarity measure for clustering activity
● Clustering quality evaluation
● k-means clustering algorithm, including iteration considerations 77
● Density-based algorithms
● Applications of clustering
T2:
● Mean shift clustering
● Agglomerative clustering
● Grid-based algorithms
● Clustering algorithms acceleration and parallelization strategies
Skills
T1:
● Explain the importance of feature selection for clustering.
● Provide guidance on the selection of initialization criteria for the k-means algorithm.
T2:
● Compare clustering approaches, highlighting relative benefits and shortcomings.
● Indicate the circumstances in which the various clustering algorithms should be used, and when other alternatives are preferable.
● Apply algorithms to a test set of data and compare the results.
● Provide illustrations to highlight the utility and value of clustering.
Dispositions
T1:
● Accurate about the role of clustering in Data Science.
● Astute about the importance of scalable and efficient clustering algorithms for real scenarios.

DM-Classification and Regression
There are many application domains that involve assigning a class value to a (possibly complex) instance of data. Similarly, there are many application domains that involve assigning a numeric value to an instance of data. The former is referred to as classification.
Regression involves estimating the relationship between a dependent variable and one or more independent variables. Though these are different tasks, they are related, and many data mining approaches can be adapted to both scenarios. A distinguishing feature of both is that they require labeled training data – i.e., representative samples that have been assigned class / dependent variable values. (See ML-Supervised Learning and ML-Deep Learning.) Knowledge
T1:
● Considerations regarding feature selection for classification
● Instance-based methods such as K-Nearest Neighbor (KNN)
● Decision tree methods
● Probabilistic models, Naïve Bayes
T2: ● Rule-based methods
● Support vector machines 78
● Neural networks
● Real world applications of classification and regression

● Deep learning and related software support (such as Caffe, TensorFlow, PyTorch)

E:
● Acceleration and parallelization strategies
Skills
T1:
● Explain the importance of feature selection for classification and regression.
● Describe criteria that might lead to selection of one method over another, such as predictive accuracy, comprehensibility of the learned model, etc.
T2:
● Identify the relationship between regression and classification.
● Identify critical situations that may benefit from the use of classifiers or regression models.
● Identify software to support each of the approaches and apply the software.
● Demonstrate the ability to select and justify an approach to classification and to apply it to an example of modest complexity.
Dispositions
T1:
● Astute regarding the importance of scalable and efficient classification and regression algorithms for real scenarios.
T2:
● Thoroughness in depicting links between classification and regression, and more generally statistics, as well as machine learning.

DM-Pattern Mining – T2
This topic is concerned with seeking patterns within data. For data collections of considerable size, brute force approaches are often computationally infeasible but selected algorithms provide a way forward. (Pattern matching has important applications in biotechnology through genome sequencing but that is not developed here.)
Knowledge
● The concept of association pattern mining
● Computational complexity considerations
● Association rule mining; Apriori and Frequent pattern (FP) growth algorithms
● Sequential pattern mining; the GSP algorithms
● Efficient and parallel algorithms for pattern mining
● Application areas Skills
● Report a range of areas in which the Apriori algorithm may be used to beneficial effect in day-to-day settings.
● Apply an implementation of the Apriori algorithm to a significant application.
● Compare and contrast the utility of pattern mining algorithms.
Dispositions
● Conviction that pattern mining is a very broad topic with widespread applications.

DM-Outlier Detection – T2
An outlier is a data point that exhibits very different characteristics from the vast majority of other data. It is desirable to identify such data points since excessive attention to these can lead to distortion (and may even suggest maliciousness); though it is also important to understand the domain well enough to determine

whether there are (legitimate) exceptional cases. In what follows it will be assumed that data has already been cleansed and that a true outlier is present.
Knowledge
● Definition of the concept of outlier
● General approach - develop a model of the data and then note that a data point does not fit
● Parametric methods, such as z-score to identify numeric outliers in 1-D
● Use of probability distribution functions
● Use of depth first approaches - having identified the expected convex hull of a set of points, is it inside or outside; use of related graphical approaches
Skills
● Apply algorithms for a range of outlier detection methods.
● Compare and contrast parametric and non-parametric approaches to outlier detection.
● Explain how outlier detection methods might assist with plagiarism detection, cases of financial fraud, network intrusion detection or other application areas.
● Illustrate the importance of outlier detection through appropriate examples.
Disposition
● Thoroughness and astute perspective on outlier analysis and detection.

DM-Time Series Data – E
For certain kinds of data, the inclusion of time or date stamps is important. For instance, this can be used in measuring growth over time, or measuring traffic congestion during particular periods. See also ML-Mixed Methods.
Knowledge
● The nature of time series data, including comparison with sequential temporal data
●Data transformation - noise removal, data normalization of time series data
● Stationary and non-stationary time series
● Converting time series data to discrete sequence data 80
● Time series forecasting - predicting future values on the basis of past values
● Time series motifs - frequently occurring patterns in time series data
● Time series clustering and classification
● Outlier detection in time series - point outliers and shape outliers
Skills
● List a range of situations for which there is relevant time series data and indicate the importance of mining that data.
● Illustrate when converting time series data to sequence data is desirable.
● Explain techniques used for the clustering and classification of time series data.
Disposition
● Attention to detail in that the data mining of time series data is highly relevant in certain critical applications.

DM-Mining Web Data – T2
Increasing amounts of data exist on the web, along with mechanisms for mining that data. As always in doing data collection and mining, ethical considerations should be observed.
Knowledge
● The processes of scraping and spidering / web crawling associated with web access
● Ethical guidelines associated with accessing web data

● The structure and functionality of software libraries for accessing web data
● Knowledge discovery approaches for web data such as community discovery and link prediction Skills
● Compare and contrast community discovery and link prediction
● Use software to scrape precise data from publicly available sites. constraints.
● Develop efficient algorithms to discover knowledge from the web.
Disposition
● Passionate and collaborative access to high quality data taking account of the ethical framework.

DM-Information Retrieval – T2

Information Retrieval includes a disciplined approach to identifying and retrieving information from a larger (usually unstructured) data set. This should be seen to involve searching documents themselves, searching for documents, or searching the web. The documents may take a variety of forms: text, images, videos, sound recordings, etc. The manner in which data is stored initially can significantly influence the efficiency and effectiveness of the processes of retrieving information. Information retrieval is particularly important in certain areas such as in 81 the context of digital libraries, or in extracting information from medical health records. There are strong links with the Data Mining Knowledge Area.
Knowledge
● Techniques used for measuring the efficiency of retrieval processes
● Range of approaches to storing and organizing data so that information can be extracted efficiently; the use of encoding functions
● The concept of a search strategy; the related role of narrowing and broadening
● Keyword(s) selection for the retrieval process; use of Boolean operators ● Search of ordered data
● Techniques for searching text-based material
● Searching a set of documents; strategies for listing the names of selected items
● Feature identification and extraction for non-text-based data; searching strategies used with photographs, sound, video
● Role of hashing, indexing and filtering
● Approaches to searching text-based material
● Techniques for creating and searching relational database systems

● Various relational, non-relational, and other database formats
● Web-based information retrieval; the web viewed as a graph of interconnected nodes; relevant measures from graph theory; PageRank and related metrics that facilitate webbased search
Skills
● Devise a search strategy for a given information retrieval task.
● Explain ethical concerns that may be associated with the information retrieval processes.
● Identify opportunities for the use of parallelism to speed up search.
● Outline the main elements of an effective strategy underpinning web-based search.
● Identify software that can be used in information retrieval tasks associated with images, sound recordings and video clips.
● Create and use a relational database structure using SQL.

● Explain the roles that information retrieval may play in the operation of digital libraries.
Dispositions
● Attention to detail about the importance of a range of considerations that should underpin an efficient and effective approach to information retrieval.

1.7 Data Privacy, Security, Integrity, and Analysis for Security (DPSIA)

Issues around privacy, security, and integrity are cross-cutting – that is, they relate to competencies in all of the Knowledge Areas. Therefore, this KA is somewhat larger than others. It is organized into sub-KAs, which are then further divided into sub-domains.

Data Privacy (DPSIA/DP)

Data scientists should be able to consider data privacy concerns and its related challenges when they acquire, process, and produce data. They should recognize the trade-offs between sharing and protecting sensitive information and how domestic and international privacy rights impact a company's responsibility for collecting, storing, and handling data. Within the extensive area of cybersecurity, there are a number of concepts and subdomains that are cross-referenced within the cybersecurity knowledge areas in addition to Professionalism and Data Acquirement and Governance.

DPSIA / Data Privacy

Scope	Competencies
● Interdisciplinary trade-offs of privacy and security ● Individual rights and impact on needs of society. ● Technologies to safeguard data privacy. ● Relationships between individuals, organizations, and governmental privacy requirements.	● Justify the concept of privacy, including the societal definition of what constitutes personally private information and the tradeoffs between individual privacy and security. ● Summarize the trade-off between the rights to privacy by the individual versus the needs of society. ● Evaluate common practices, technologies, and tools that reduce the risk of data breaches and safeguard data privacy. ● Debate how organizations with international engagement must consider variances in privacy laws, regulations, and standards across the jurisdictions in which they operate. This topic includes how laws and technology intersect in the context of the judicial structures that are present – international, national and local – as organizations safeguard information systems from cyberattacks.

Sub-domains	
DPSIA/DP-Social Responsibility– T1, T2, E DPSIA/DP-Cryptography – T1, T2	DPSIA/DP-Information Systems – T1, T2, E DPSIA/DP-Communication Protocols – T1, T2 83

DPSIA/DP-Social Responsibility

Summarize the trade-off between the rights to privacy by the individual versus the needs of society.

Knowledge

T1:

● Sensitive data that can be exposed by using social engineering and social media

● Trade-offs between the right to privacy and the need of transparency through information dissemination

● Ethical responsibilities about disclosing, transmitting, and sharing information obtained from analytics tools

T2:

● Legal codes that involve privacy concerns of using data to perform certain actions

● International privacy laws that impact society and computing development assets

Skills

T1:

● Demonstrate awareness about data sensitiveness when data is processed as an input.

● Identify scenarios where data cleaning must be considered before processing information.

● Apply techniques to provide data privacy during raw data processing, such as provide ranges or salting techniques.

Elective:

● Demonstrate awareness of global policy and regulations such as HIPAA, FCRA, ECPA, that may affect decision making.

● Demonstrate awareness of well-known search engines and their information storage policies that identify and jeopardize computer users' privacy.

Dispositions

T1:

● Ethical understanding that data provided to any entity may impact the loss of data privacy.

● Accurate and ethical handling of data through computing systems or channels, recognizing the public and private implications in society of inappropriately doing so.

DPSIA/DP-Cryptography

Summarize the usage of cryptographic techniques to emphasize data privacy.

Knowledge

T1:

● Importance of encrypting data before transmitting it through any channel.

● Computational time trade-offs of using encrypted vs unencrypted data for statistical analysis.

T2:

● Differences between symmetric and asymmetric algorithms 84

● Hash functions for privacy checking and protecting authentication data
● Encryption algorithms
Skills
T1:
● Identify tools/mechanisms to encrypt data to reduce the risk of data breaches while keeping in mind computational performance.
● Performing training for different entities such as individuals, organizations, and government agencies about data encryption processes that impact privacy requirements.
● Illustrate the use of cryptography to provide privacy, such as message authentication codes, digital signatures, authenticated encryption, and hash trees.
● Identify the trade-offs between processing plain text data and encrypted data.
T2:
● Analyze which cryptographic protocols, tools, and techniques are appropriate for providing data privacy, protection, integrity, authentication, non-repudiation, and obfuscation.
Dispositions
T1:
● Astute about the need for different mechanisms of encryption.

DPSIA/DP-Information Systems
Summarizing the concept of information systems by contextualizing information and the privacy of such by using well-known models.
Knowledge
T1:
● Concepts and techniques to achieve authentication, authorization, access control, and data privacy.
● Layered defenses to achieve maximum confidentiality, integrity, and availability (CIA).
T2:
● Different access control mechanisms that enforce data privacy such as Bell-LaPadula, Chinese Wall, and Clinical Information Systems Security, to resolve different privacy and transparency conflicts of interest.
● Well-known information system designs and implementations and the impact on data privacy.

Elective:
● Traffic analysis to demonstrate how private information can be jeopardized in a given secure system.
Skills
T1:
● Explain how the data privacy needs of a system might impact the security of the system.

● Discuss the trade-offs between data transparency and data privacy. 85 T2:
● Outline what information should be provided to a computer entity, balancing usability and privacy and how to report information. Dispositions
T1:
● Use discretion when protecting information in a given computer system.
DPSIA/DP-Communication Protocols Summarizing how communication protocols

can be used to guarantee a secure communication over channels (secure and insecure); the consideration of cryptographic protocols used in communication protocols; and recognizing the impact on data privacy by using well-known applications' protocols.

Knowledge

T1:

● The importance of security protocols that enable secure communication over insecure channels

● The importance of privacy protocols to enable private interactions over secure channels

● Internet/communication protocols that can guarantee private communication between applications and servers

T2:

● Balancing security protocols vs privacy protocols by using and not using cryptography

Skills

T2:

● Use security protocols to set up secure channels using different cryptographic primitives.

● Apply privacy protocols to set up private channels using secure transmission techniques.

Dispositions

T2:

● Accurate selection of secure protocols to ensure a private connection between utilities.

● Astute about the secure protocols that interchange data sets without jeopardizing privacy characteristics.

Data Security (DPSIA/DS)

This knowledge unit focuses on the protection of data at rest, during processing, and in transit. It requires the application of mathematical and analytical algorithms to fully implement the necessary security objectives over data-driven applications. This unit allows deeper understanding of data security objectives along with various tools to achieve them.

DPSIA / Data Security

Scope	Competencies
● Cryptographic concepts: • Encryption/decryption, message authentication, data integrity, nonrepudiation; Attack classification (ciphertext-only, known plaintext, chosen plaintext, chosen ciphertext); Secret key (symmetric), cryptography and public-key (asymmetric) cryptography. ● Threat models for data driven applications ● The role mathematical techniques play in producing useful encryption knowledge. ● Public key cryptography for data security ● The data security part of CSEC 2017 document provides additional scope.	● Describe the purpose of cryptography and list ways it is used in data communications; and which cryptographic protocols, tools and techniques that are appropriate for a given situation. ● Understand cipher, cryptanalysis, cryptographic algorithm, and cryptology ● Explain how public key infrastructure supports digital signing and encryption and discuss the limitations/vulnerabilities. ● Exhibit a mathematical understanding behind encryption algorithms ● Explain the difference between, and applications of, Symmetric and Asymmetric ciphers. ● Analyze threats to real-time applications that consume/produce critical data ● Utilize attack vectors and attack tree concepts to model threats ● Explain how data transmissions over a network or the web can be protected

Sub-domains

DPSIA/DS-Data quality and handling for security – T1, T2 DPSIA/DS-Classification of cryptographic tools – T2 DPSIA/DS-Security and performance tradeoff – T2	DPSIA/DS-Network and web protocols – T1 DPSIA/DS-Privacy and data governance – see DPSIA/DP

DPSIA/DS-Data quality and handling for security

Knowledge

T1:

● Qualitative metrics
● Security importance of data assets
● Type of security objectives needed
● Data sources and assets
● Controlling and managing accessibility to data assets

T2:

● Attack vectors and trees
● Threat models of different use cases 87
● Impact of threats on data sources Skills T1:
● Understand data flow in applications.
● Derive important security objectives to achieve.

● Explain the reasons for selecting what data assets to secure.
T2:
● Deduce possible security and privacy threats based on the data flow in applications.
● Implement access control mechanisms to restrict data leaks.
● Implement required authentication processes for securely accessing data assets.
● Assess the significance of data assets based on external and internal factors.
● Perform threat analysis on practical systems.
● Categorize threats based on their impacts.
Dispositions
T2:
● Accurate ability to extract threats on data-driven systems.

DPSIA/DS-Classification of cryptographic tools – T2
Knowledge
● Cryptographic techniques
● Usability of various techniques and tools ●
Cryptographic protocol designs using discrete mathematical concepts
● Public key cryptosystems vs. secret key cryptosystems
Skills
● Apply various cryptographic techniques to achieve necessary security objectives.
● Compare merits and demerits of various techniques.
● Explain performance characteristics of various techniques.
● List attack models for each cryptographic technique.
● Implement data security mechanisms using available cryptographic schemes.

Dispositions
● Recognize the importance and unique characteristics of various crypto protocols.

● Choose the right protocols depending on application requirements.

DPSIA/DS-Security and performance trade-off – T2
Knowledge
● Performance requirements of data driven applications

● Impact of security schemes on performance of applications

Skills

● Apply design principles to balance performance and security needs.
● Investigate the operational environments to characterize critical parameters that affect both performance and security of a system.
● Develop mechanisms that enable high data availability while achieving necessary security.
Dispositions
● Understand the performance and security trade-offs among different protocols.
● Recognize which ciphering technique to opt for based on application requirements.

DPSIA/DS-Network and web protocols – T1
Knowledge

● Insight on data transactions over networks for data-driven applications
● Network and web protocols
● Available and/or enabled security modules in communication protocols
● Operations (storage, retrieval, remote compute) on data network and web
Skills
● Dissect and tune communication protocols to enable security.
● Explain the unique characteristics and working principles of network and web protocols.
● Understand how data gets communicated to various entities over the network or web.
Dispositions
● Strong commitment to network/web protocol security. 89
Data Integrity (DPSIA/DI)

This knowledge unit focuses on the completeness, accuracy, and consistency of data over its entire life cycle starting from generation through transmitting, storing, retrieving, and processing of data. Preservation of data integrity is mandatory in the realm of data science since malicious actions on data can lead to incorrect inference and muddle the decision-making process.
Data scientists must be aware of integrity preservation tools and techniques while understanding their roles and efficiency in order to correctly implement the integrity requirements in data science applications.

DPSIA / Data Integrity

Scope	Competencies
● The accuracy, consistency, and validity of data ● Need for integrity requirements from security perspective ● Techniques and mechanisms to ensure data integrity ● Common security threats in data integrity	● Explain the differences of data integrity, data security, and data privacy ● Describe the main strands of knowledge needed to address data integrity ● Demonstrate the skills to apply commonlyused methods to ensure data integrity ● Perform confidently when dealing with security threats affecting data integrity.

Sub-domains	
DPSIA/DI-Logical integrity – T1 DPSIA/DI-Physical integrity – T1 DPSIA/DI-Security threats affecting data integrity – T1	DPSIA/DI-Methods to ensure data integrity – T1 DPSIA/DI-Data corruption and data validation – T2

DPSIA/DI-Logical integrity – T1 Knowledge
● The concept of logical integrity

● Types of integrity constraints in database systems
● Entity integrity, referential integrity, domain integrity, user-defined integrity
Skills
● Explain concepts in logical integrity
Dispositions
● Accurate in explaining logical integrity

DPSIA/DI-Physical integrity – T1 Knowledge
● The concept of physical integrity 90
● Physical and hardware methods to ensure data integrity such as RAID, redundant hardware, uninterruptible power supply, error-correcting memory, and server cluster
Skills
● Explain concepts in physical integrity
● Describe physical and hardware methods to ensure physical integrity
Dispositions
● Confidence in addressing physical integrity through hardware methods

DPSIA/DI-Security threats affecting data integrity – T1 Knowledge
● Common data integrity threats including human errors, software errors, transmission errors, malware, insider threats, cyber-attacks, and compromised hardware
● Data and information poisoning
● Data provenance assurance
Skills
● List common types of security threats affecting data integrity.
● Describe the potential vulnerabilities behind different hash functions, such as SHA-1 and MD5.
Dispositions
● Confidence in describing common security threats.

DPSIA/DI-Methods to ensure data integrity – T1 Knowledge
● Role of hash algorithms in integrity preservation
● Role of Message Authentication Codes (MACs) and its variants
● CRC and checksum for achieving integrity
● Mechanism behind digital signature schemes (RSA and ECDSA) Skills
● Explain how to use hash algorithms and MAC mechanisms to ensure data integrity.

● Describe digital signature schemes and their needs in integrity preservation context.
● Compare and contrast different integrity preservation techniques in terms of performance and security.
● Understand how to use the integrity models in multiple data ownership domains to ensure provenance and maintain data validity.
Dispositions
● Thoroughness when addressing data integrity through the use of various methods and techniques.

DPSIA/DI-Data corruption and data validation – T2 Knowledge
● The concept of data corruption

● The concept of data validation
● Methods to prevent data corruption including checksums and error correcting codes
● Validation methods including input validation, data type validation, range and constraint validation, and cross-reference validation
Skills
● Explain concepts in data corruption and data validation.
● Describe methods to prevent data corruption and ensure data validation.

Analysis for Security (DPSIA/AS):
This knowledge unit focuses on data science analytical techniques including statistics, probability, machine learning, and data mining, with a specific focus on security and privacy problems. This unit allows deeper understanding of data science tools, algorithms and techniques for security and privacy.

DPSIA / Data Analysis for Security

Scope	Competencies
● Understand security data telemetry and different security applications ● Statistical analysis for security telemetry data ● Machine learning for security telemetry data ● Explainable machine learning methods for security-critical applications ● Machine learning vulnerability and robustness	● Categorize different security-critical applications and understand various security telemetry data. ● Demonstrate in-depth knowledge and strong hands-on implementation skills in machine learning (ML) and statistical methods to improve security applications. ● Recognize when ML explainability and resiliency are necessary in a security application.

Sub-domains	
DPSIA/AS-Machine learning (ML) algorithms and statistical methods for security – T1	DPSIA/AS-Machine learning (ML) robustness and explainability – T1 DPSIA/AS-Categories of security applications – T2 92 DPSIA/AS-Machine learning (ML) algorithms and statistical methods for security – T1

Knowledge
● Statistical methods for exploratory data analysis on security data including descriptive statistics, summary plots, outlier detection, point estimation, hypothesis testing, test statistics, linear regression, and generalized linear regression.
● Computer vision-based approaches such as malware-as-an-image technique, transfer learning, hierarchical ensemble neural network (HeNet) built on hardware for both static and dynamic threat classification and malware detection.
Skills
● Translate security applications into problems that can use ML.

● Design malware detection solutions by employing malware-as-an-image, transfer learning and hierarchical ensemble neural network (HeNet) for static and dynamic detection mechanisms.
● Explain decisions made by ML models for security applications to audiences with different backgrounds.
Dispositions
● Understand different perspectives from computer vision, natural language processing and classical data analysis to approach threat detection, malware intelligence and exploit identification problems

DPSIA/AS-Machine learning (ML) robustness and explainability – T1
Knowledge
● Basic concept of adversarial machine learning, types of attacks against ML models, and
● Adversarial machine learning techniques such as fast gradient sign, iterative fast gradient, universal adversarial perturbation
● Defense techniques such as adversarial training to better protect ML models
● Explainable machine learning methods for security applications. Explanations include local explanation, which is per-sample based, and global explanation, which considers the dataset as a whole. Know how to employ model-agnostic explanations on natural images to vision-based malware detection mechanisms.
Skills
● Evaluate ML resiliency in terms of identifying its blind spot and bypassing its detection.
● Improve ML resiliency by conducting adversarial training.
● Conduct studies on ML algorithms and explain the impact of these models to security experts.
● Communicate with various stakeholders to define ML metrics to address interpretability and vulnerability. 93
● Explain why ML resiliency and vulnerability are a key metric for ML used in security and privacy applications.
● Apply explainable AI methods such as LIME, LEMNA, TCAV to ML models built for security applications.

● Perform and conduct ML model selection based on the trustworthy scores. Especially when using malware-as-an-image approach, be efficient at applying LIME for malware classification model interpretability.
Dispositions
● Attention to detail in ML evaluation using robustness and potential vulnerabilities in addition to typical metrics assessing classification accuracy, false positive, precision, along other characteristics.

DPSIA/AS-Categories of security applications – T2
Knowledge
● Security-critical applications: network analysis, malware intelligence, malware triage, dynamic malware analysis, hardware telemetry analysis.
● Types of security telemetry data: dynamic logs, binary, static code, dynamic code.
Skills
● Select ML methods to use based on the nature of security telemetry data.
Dispositions

● Initiative to increase knowledge of data usage through various security applications and optimal use of datasets.

1.8 Machine Learning (ML)

Machine learning, sometimes known as Statistical Learning, refers to a broad set of algorithms for identifying patterns in data to build models that might then be productionized and possibly productized. These methods are critical for data science.
Data scientists should understand the algorithms they apply and make principled decisions about their use.

Scope	Competencies
● Broad categories of machine learning approaches (e.g., supervised and unsupervised). ● Algorithms and tools (i.e., implementations of those algorithms) for machine learning. ● Machine Learning as a set of principled algorithms (e.g., optimization algorithms), rather than as a "bag of tricks." ● Challenges (e.g., overfitting) and techniques for approaching those challenges. ● Performance metrics. ● Training and testing methodology. ● Algorithmic and data bias, integrity of data, and professional responsibility for fielding learned models.	●Recognize the breadth and utility of machine learning methods ● Compare and contrast machine learning methods ● Select appropriate (classes of) machine learning methods for specific problems. ● Use appropriate training and testing methodologies when deploying machine learning algorithms. ● Explain methods to mitigate the effects of overfitting and curse of dimensionality in the context of machine learning algorithms. ● Identify an appropriate performance metric for evaluating machine learning algorithms/tools for a given problem. ● Recognize problems related to algorithmic and data bias, as well as privacy and integrity of data. ● Debate the possible effects -- both positive and negative -- of decisions arising from machine learning conclusions.
Sub domains	
ML-General – T1, T2, E ML-Supervised Learning – T1, T2, E ML-Unsupervised Learning – T1, T2, E ML-Mixed Methods – E ML-Deep Learning – T1, T2, E	Note that Reinforcement Learning appears in AI-Knowledge Representation and Reasoning (Probability-based Models)

ML-General

Given the centrality of machine learning algorithms to many data science tasks, data scientists should be aware of a wide range of machine learning approaches, as well as the long history of work in this area. A data scientist should be aware of where to look for possible techniques to apply to new problems.

A data scientist should also be aware of cross-cutting concepts, such as the need to evaluate performance and general classes of challenges faced in machine learning.

Knowledge

T1:

● Major tasks of machine learning, including supervised, unsupervised, reinforcement, and deep learning

● Difference between symbolic versus numerical learning, statistical versus structural/syntactic approaches

● Learning algorithms as principled optimization approaches

● “Doing machine learning” as one method of data mining. “Doing machine learning” as a process.

● Importance of robust evaluation

● Challenges for machine learning, including quality of data, need for regularization

Skills

T1:

● Compare the goals, inputs, and outputs of supervised, unsupervised, reinforcement, and deep learning.

● Recognize that different types of data-driven questions can be answered by different approaches;

● For a given data-driven question, explain why a particular approach is appropriate.

● Explain at a high level that ML models and algorithms are principled techniques based on mathematical and statistical foundations.

● Describe the process of “doing machine learning” as a method of data mining: understanding the question / problem a client cares to solve, gathering the data relevant to solving that problem, converting raw data into features, selecting appropriate machine learning methods, tuning those methods, evaluating performance (often against a baseline), and presenting results and insights.

● Discuss the trade-off between fitting to training data and generalizing to new data and how model complexity, as well as the number of examples and features, affect this tradeoff. Relate this trade-off to the role and setting of hyperparameters.

● List trade-offs across performance, interpretability, scalability.

● Recognize that different optimization functions and techniques may yield different tradeoffs in this space. T2:

● Explain a provided derivation of a simple optimization function and learning algorithm from first principles, e.g. decision trees using information theory, logistic regression using maximum likelihood and stochastic gradient descent, PCA using variance 96 minimization and eigenvalues. I.e., the student should be able to follow the derivation and explain it -- not generate it from scratch.

● Analyze performance across models using bootstrapping and statistical significance testing.

● Explain how to efficiently transition a model into production and with appropriate tools that support that transition from the onset.

● Choose which tools to use based on the size of the data -- for Big Data, it is essential to choose a machine learning tool that can run parallelized, otherwise, the learning process may take much longer than is acceptable.
● List state-of-the-art machine learning tools available.
Elective:
● Describe the process of automated (or meta-) learning, specifically how to automate the machine learning pipeline, including data pre-processing, model selection, model structure search, and hyperparameter tuning.
Dispositions
T1:
● Professional use of machine learning. Appreciate that, though recently made popular, machine learning is not a recent innovation. Look for existing solutions before presuming a new invention is required.
● Accurate and ethical use of machine learning (i.e., is not an ad-hoc set of “tricks” and that it should be used responsibly.)
● Strong commitment to applying machine learning as part of a process toward a goal for a client. doing machine learning” is not, in the general case, a simple process of applying a machine learning program to a conveniently-formatted data set. Thoroughness when comparing learned models. There are several dimensions along which learned models may be compared, ranging from empirical loss minimization to model size and complexity to human interpretability.
● Ethically present results that are fair and honest comparisons considering all aspects of model comparison (quality, efficiency, interpretability, etc.).

ML-Supervised Learning
One major class of learning approaches can be described as “supervised” and includes techniques for both classification and regression. A data scientist should be aware of these types of algorithms, including challenges and methodologies that are unique to this type of learning. Note the relationship of this sub-domain with DM-Classification and Regression.
Knowledge
T1:
● Major tasks of supervised learning: regression and classification
● Use cases of regression and classification
● Important considerations and trade-offs in supervised learning, including the relationship between model complexity and generality; the trade-off between bias and variance; Occam’s razor as motivation for simple models.
● The need for separation of training, test, and validation data. Define training error and testing error.
● Common evaluation metrics for classification tasks (e.g., accuracy, sensitivity, specificity, precision, recall, F1, AUROC, regret) and regression tasks (e.g., root mean squared error, mean absolute error, R^2)
● The need for validation data. Cross-validation procedures and goals: tuning hyperparameters and measuring model performance.
● Criteria for assessing the quality of training, test, and validation data, such as number of examples or class stratification.
● Classification and regression algorithms, including at least one linear and one non-linear algorithm for each. (e.g., linear regression/classification, logistic regression, nearest neighbor, Naive Bayes, decision tree learning algorithms).

● Common extensions to basic algorithms, including polynomial features and ensembles (e.g., bagged models, boosted models, random forests). T2:
● Approaches for determining whether a model has high bias or high variance, e.g. training vs test performance, learning curves.

● Reasons to augment or reduce feature set; at least two approaches for each and trade-offs.
● How supervised classifier-learning models can be applied to multiclass problems, including how binary classification models can be extended to multi-class tasks.
● How to express performance using macro- and micro- metrics.
● At least one advanced supervised learning algorithm (e.g. SVMs with kernels, neural networks).
Elective:
● Derivation of supervised learning algorithms from first principles. Skills T1:
● Explain performance of a classifier model using a confusion matrix.
● Compare strengths and weaknesses of evaluation metrics for classification tasks and regression tasks.
● Compare the trade-offs of at least two applied classification algorithms; compare the trade-offs of at least two regression algorithms.
● Apply at least two classification and two regression algorithms to small and medium data sets.
● Compare training and testing error in terms of what they tell us about learned models.
● Compare the performance of algorithms using various metrics.
● Apply at least two extensions (e.g., ensemble methods) to small, medium, and large data sets.
● Justify when extensions such as polynomial features and ensembles are appropriate based on the problems each is able to address.
T2:
● Execute at least two classification and two regression algorithms on a large dataset.
● Illustrate at least one extension to a large dataset.
● Implement methods to mitigate high bias or high variance.
● Perform feature augmentation and selection on a medium or large sized problem.
● Apply advanced supervised learning algorithms (e.g. SVMs with kernels, neural networks).
Elective:
● Devise a simple optimization function and learning algorithm from first principles, e.g. logistic regression using maximum likelihood and stochastic gradient descent.
Dispositions
T1:
● Thorough and astute algorithm selection and evaluation. Know that these choices have implications for and must be made with important stakeholders -- i.e., those for whom models are being developed.
● Apply accurate and ethical evaluation approaches for models in which we can have high confidence.

ML - Unsupervised Learning
A major class of machine learning approaches can be described as “unsupervised” and include techniques for clustering and dimensionality reduction. A data scientist

should be aware of these types of algorithms, including challenges and methodologies that are unique to this type of learning. Note the relationship of this sub-domain with DM-Cluster Analysis.

Knowledge

T1:

● Major tasks of unsupervised learning, including clustering and dimensionality reduction.

● Use cases for both tasks (e.g., data exploration/summarization/visualization, feature selection, data compression, data denoising, prototype learning, recommender systems, topic modeling).

● At least one simple clustering algorithm, e.g. k-means or hierarchical clustering.

● Trade-offs of connectivity-based vs centroid-based clustering.

● At least one simple dimensionality reduction algorithm, e.g. principal component analysis (PCA).

● Similarities and differences between feature transformation, feature selection, and feature projection.

T2:

● At least one advanced clustering algorithm, e.g. density-based methods such as Gaussian mixture models (GMMs).

● At least one advanced dimensionality reduction algorithm, e.g. independent component analysis (ICA) or non-negative matrix factorization (NMF).

Elective:

● At least one mathematical method for implementing algorithms efficiently, e.g. matrix factorization and singular value decomposition (SVD) vs eigendecomposition for PCA.

● At least one advanced algorithm, e.g. spectral clustering, kernel k-means, kernel PCA, latent Dirichlet allocation (LDA).

● The connection of PCA to autoencoders; generalization to non-linear dimensionality reduction.

● Derivation of unsupervised learning algorithms from first principles. Skills

T1:

● Apply at least one clustering and one dimensionality reduction algorithm to small, medium, and large data sets.

● Explain the performance of an unsupervised learning algorithm using various metrics (e.g., visualization; comparison to ground truth, if available; computing metrics such as cluster density; indirect metrics via utility towards another application).

● Implement methods for choosing hyperparameters, e.g. the number of clusters for kmeans or the number of components for PCA.

T2:

● Compare the trade-offs of at least two clustering algorithms.

● Compare the trade-offs of at least two dimensionality reduction algorithms.

Elective:

● Apply advanced unsupervised algorithms.

● Devise a simple optimization function and learning algorithm from first principles, e.g. PCA using variance minimization and eigenvalues. Extend these techniques to similar models.

Dispositions

T1:

● Thorough and astute algorithm selection and evaluation. Appreciate the importance of algorithm choice and evaluation metric on the quality of a learned model. Know that these choices have implications for and must be made with important stakeholders -- i.e., those for whom models are being developed. [See ML - Supervised Learning] ●
● Appreciate the importance of applying accurate and ethical principled evaluation approaches for models in which we can have high confidence.
● [See ML - Supervised Learning]
T2:
● Attention to dealing in unsupervised learning which offers useful techniques for data exploration, understanding, summarization, and visualization.
● Attention to detail in that unsupervised learning which can be a useful pre-processing step to improve the quality or efficiency of supervised learning algorithms.

ML-Applications that Require Mixed Methods – E

Some learning problems and domains have special structure that can be leveraged by specialized techniques. A data scientist should be aware of these broad classes of applications and should know where to turn for possible methods to approach them. Note the relationship of this sub-domain with DM-Time Series Data.
Knowledge
● Examples of learning problems and domains in which the structure of data or interrelatedness of data points may be leveraged in the learned model. For example, time series prediction, sequence prediction, recommender systems.
● How time dependencies or assumptions of shared information across data points may be leveraged in learning.
● Shortcomings of using a supervised or unsupervised approach instead of a mixed approach, e.g. problems of model interpretability or performance.
T2:
● For one such problem, at least one standard approach for learning, e.g., Hidden Markov Models (HMMs) for sequence prediction or Collaborative Filtering for recommender systems.
● The need for separation of training and test data in this context.
● Common evaluation metrics for the selected task, e.g., recall, precision, F1 score for recommender systems.
● Criteria for assessing the quality of training, test, and validation data for the selected problem.
Skills
● Integrate one such problem to a framework for learning. I.e., map data to inputs and outputs, consider settings of hyperparameters, run an appropriate learning algorithm. Dispositions
● Attention to detail regarding challenges (e.g., time inhomogeneity, data sparsity) present in ML models generally may be more salient in specific contexts.
ML-Deep Learning
The availability of data, as well as the availability of computational processing power have led to new and powerful techniques for large-scale learning. A data scientist should be aware of these types of algorithms, including challenges and methodologies that are unique to this type of learning.
Knowledge

T2:
● How multilayer neural networks (including non-deep networks) learn and encode higherlevel features from input features.
● Common deep learning architectures, such as deep feedforward networks, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and LSTMs; purpose and properties of each.
● Practical challenges of common deep learning approaches, e.g., choosing a deep learning architecture, having sufficient data / possibility of overfitting, length of learning time, interpretability.
● Examples of regularization methods for deep learning architectures, such as early stopping, parameter sharing, and dropout.
● Examples of methods for mitigating other challenges of deep learning, such as tools that work with GPUs or on distributed systems.
● Selection of appropriate tools that scale with the size of the data -- specifically, processing Big Data calls for Deep Learning tools that run in a parallelized way.
● Be aware of the state-of-the-art deep learning tools available.
● At least one commonly used algorithm for learning in the context of deep networks, e.g., how backpropagation is used in a deep feedforward network or how backpropagation is used to learn higher-order features in a convolutional network; how backpropagation through time is used in recurrent networks.
● The operation of convolution and why it may be useful, e.g., detecting vertical edges in an image.
● Pooling; examples of pooling functions such as max pooling and use cases.
● Challenge of long- vs short-term dependencies in recurrent neural networks; at least one solution, such as LSTMs.
Elective:
● Deep generative models, such as generative adversarial neural networks (GANNs) and applications for which they may be used.
● Practical challenges of such approaches, e.g., convergence, mode collapse, etc.
● Approaches for handling or mitigating the effects of the above.
Skills
T2:
● Choose the type of deep learning approach(es) that would be most appropriate to apply for a given data set and task.
● Use a deep learning toolkit (e.g., PyTorch, Tensorflow) to study a learned model’s output from a dataset.
● Use a deep learning toolkit (e.g., PyTorch, Tensorflow) to learn a model for a dataset, including configuring a network.
Elective:
● Implement, from scratch, a generative approach for a specific goal with a deep learning toolkit.
● Modify a toolkit to work for a given system architecture. Dispositions
T1:
● Professionalism in machine-learned modeling, understanding the potential negative implications of using a machine-learned model that is difficult or impossible to interpret or explain.
● Responsible use of deep learning, since there are many problems for which the power of deep learning is more than what is necessary.
● Collaborative and ethical commitment to the social and political concerns around deepfakes.

1.9 Professionalism (PR)

In their technical activities, data scientists should behave in a responsible manner that brings credit to the profession. One aspect of this is being positive and proactive in seeking to bring benefit, to undertake positive developments and doing so in a way that is responsible and ethical. Much of this is amplified in general terms in [1]. The section below serves to highlight relevant issues of specific concern to the data scientist.

Scope	Competencies
● The meaning of competency and being able to demonstrate competency ● The acquisition of competencies particularly relevant to the data scientist ● Acquiring expertise / mastery or extending competency; the role of journals, conferences, courses, webinars ● Technological change and its impact on competency ● The role of professional societies in CPD and professional activity	● Recognise the range of knowledge that underpins a professional approach to data science ● Demonstrate the skills that underpins a current and ongoing professional approach to data science ● Construct a set of dispositions that underpin a confident, effective and professional approach to all aspects of data science as well as the wherewithal to maintain such an approach
Sub-domains	
PR-Continuing Professional Development – T1 PR-Communication – T1 PR-Teamwork – T1 PR-Economic Considerations – T2	PR-Privacy and Confidentiality – T1 PR-Ethical Considerations – T1 PR-Legal Considerations – T2 PR-Intellectual Property – E PR-On Automation – E

PR-Continuing Professional Development – T1

The essence of a professional is being competent in certain aspects of data science. It is the responsibility of the professionals to undertake only tasks for which they are competent. There are then implications for keeping up-to-date in a manner that is demonstrable to stakeholders (e.g. employers).

Knowledge

● The meaning of competency and being able to demonstrate competency
● Acquiring expertise / mastery or extending competency; the role of journals, conferences, courses, webinars
● Technological change and its impact on competency
● The role of professional societies in CPD and professional activity

Skills

● Justify the importance to professional data scientists of maintaining competence.

● Describe the steps that professionals would typically take to extend competence or acquire mastery, explaining the advantages of the latter.
● Argue the importance of the role of professional societies in supporting career development.

Dispositions
● Being proactive and passionate about recognition that data science is a rapidly changing field where keeping current, as well as knowing how to stay current, are vital.

PR-Communication – T1

There are various contexts in which the data scientist is required to undertake communication with very diverse audiences. That communication may be oral, written or electronic. There is often the need to engage in discussion about the role that data science can play, to communicate multiple aspects of the data science process with colleagues, to convey results that may lead to change or may provide new insights. Being able to articulate the need for change and being sensitive to the consequences of change are important professional attributes. These activities may entail the ability to have a discussion about limitations in certain contexts and to suggest research activities. Communication from the data scientist must be underpinned by an evidence-based approach to decision making. There is special significance to this in the context of machine learning and automation where the reasons for decisions may require clarification. An important consequence of developments in machine learning is the ability of machines to understand natural language (and so voice input), which can then be employed in such contexts as robotics, word processors or intelligence driven search engines (e.g. Siri, Cortana, Google Assistant, Alexa).
Knowledge
● Different forms of communication – written, oral, electronic - and their effective use
● The technical literature relevant to data science
● Audiences relevant for communication involving the data scientist – including small groups, large groups, experts and non-experts, younger groups, senior managers, machines – and the elements of effective communication in each case
Skills
● Evaluate aspects of the technical literature relevant to data science
● Produce a technical document for colleagues to guide technical development
● Produce presentations for a range of audiences who have an interest in aspects of data science
● Design situation reports for senior managers outlining significant initiatives stemming from a data science investigation including as necessary general issues associated with change management
Dispositions
● Adjust in response to changes in relevant changing technology, know how to do so effectively and be alert to opportunities for new developments
● Proactive and self-motivated determining the significance of new learning and new experiences
● Accurate and respectful about one’s strengths and weaknesses regarding knowledge

PR-Teamwork – T1

The data scientist will often become a member of a team. This may entail being a team leader, or supporting the work of a team (which may be sensitive). It is important to understand the nature of the different team roles as well as the typical dynamics of teams. In terms of teamwork, the data scientist often needs to be able to collaborate not only with data scientists with different tool sets but, in general, with a diverse group of problem solvers.

Knowledge

● Team selection, the need to complement abilities and skills of team members
● The dynamics of teams and team discipline
● Elements of effective team operation

Skills

● Outline steps that could be taken to deal with conflict situations within teams.
● Summarise the considerations involved is selecting a team to undertake a specific data science investigation.
● Recognise the qualities desirable in the team leader for a data science research investigation.

Dispositions

● Respectful, collaborative and act appropriately to sensitivities regarding the formation and operation of teams.
● Be willing to work with others and act appropriately, setting aside unimportant differences when working with others.

PR-Economic Considerations – T2

Data scientists should justify their own positions as well as the kind of activity in which they engage.

Knowledge

● The cost and value of high quality data sets, and of their maintenance ● Justification in cost regarding data science activities
● Estimation of project costs
● Promotion of data science ● Automation stemming from data science activity

Skills

● Assess the value of data sets for organizations, taking into account any requirement for maintenance.
● Argue the case for what data an organization should routinely gather; design a related data collection process identifying the attributes to be included and the form the collection should take, having an eye to quality.
● Assess the cost (in terms of resources generally) of collecting high quality data for a particular purpose.
● Justify the creation of data science activities within an organization and quantify their cost.
● Infer the value to an organization of undertaking a particular investigation or research project.
● Monitor the resources needed to carry out in-house investigations and compare that with outsourcing such activities.
● Evaluate the costs associated with the automation of a particular activity.

Dispositions

● Respectful and act appropriately to costs associated with data science activities.

PR-Privacy and confidentiality – T1

It is possible to gain access to data in a multitude of ways, by accessing databases, using surveys or questionnaires, taking account of conditions of access to some resource, and even with developments such as the Internet of Things, specialized sensors, video capture and surveillance systems. Although gaining access to all kinds of information is important, professionals must do this legally and in such a way that the information is accurate and it protects the rights of individuals, as well as organizations and other groups, are protected. Note the relationship of this sub-domain and the Knowledge Area on Data Privacy.

Knowledge

● Freedom of information
● Data protection regulations including General Data Protection Regulation (GDPR) regulation – see [5] 107
● Privacy legislation
● Ways of maintaining the confidentiality of data
● Threats to privacy and confidentiality
● The international dimension

Skills

● Describe technical mechanisms for maintaining the confidentiality of data.
● Compare the privacy legislation from different countries, highlighting problems arising from any differences.
● Recognize the privacy and confidentiality issues arising from the use of video, voice and face recognition software.
● Summarize the contexts in which particular privacy legislation should be applied, having an eye to international standards.

Dispositions

● Responsible for maintaining privacy and confidentiality to ensure confidence in data science activities. Contextual issues
● The legal framework associated with privacy and security can vary from one country to another.

PR-Ethical Considerations – T1

Ethical issues are of vital importance for all involved in computing and information activities as captured extensively in [1]. Underpinning these activities is a view that professionals should undertake only tasks for which they are competent, and even then should carry out such tasks in a way that reflects good practice in its many forms. Maintaining or extending competence is essential. A heightened awareness of legal and ethical issues must underpin the work of the data scientist. Professionals should consider the ethical issues associated with their decisions as a very important starting point that enables them to recognize themselves as “independent, ethical agents.”

Knowledge

● Ethical issues associated with competence and the maintenance of that competence
● Confidentiality issues associated with data and its use
● General Data Protection Regulation (GDPR) regulation – see [5]
● Need for data, including samples of data, to be truly representative of a situation
● Awareness of, and the possible nature of, bias in data and in algorithms; mechanisms for checking and avoiding bias

● Algorithmic transparency and accountability
Skills
● Illustrate a range of situations in which a data scientist may venture beyond their range of competence and identify steps to mitigate such situations.
● Demonstrate techniques for establishing lack of bias in data sets or in algorithms.
● Debate the merits of joining a network of professionals in the data science area.
Dispositions
● Responsive to the deep ethical issues associated with gathering data and its use.
● Responsive to issues of bias and be proactive in seeking to remove these.
● Self-directed and self-motivated in the advancement of data science.

PR-Legal considerations – T2

Computer crime has continued to increase both in volume and its severity over recent years. In many cases criminals have brought disruption, even chaos, to many organizations. Their threat cannot be ignored and professionals must take steps to counter the possibility of severe disruption. In many cases the law has adjusted to counter these trends but this is an ongoing area of continuous change and adjustment.
Knowledge
● Computer crime relevant to data science
● Cyber security
● Crime prevention
● Mechanisms for detecting criminal activity, including the need for diverse approaches
● Recovery mechanisms and maintaining 100% operation
● Laws to counter computer crime
Skills
● Assess a range of mechanisms for detecting a stated form of criminal activity.
● Justify the desirability of having multiple diverse approaches to countering threats.
Dispositions
● Responsible and ethical, but sensitive and caring attitude, when confronted with possible criminal situations. Contextual issues
● The legal framework can vary from one country to another.

PR-Intellectual property – E

Intellectual Property Rights (IPRs) such as copyright, patents, designs, trademarks and moral rights, exist to protect the creators or owners of creations of the human mind. Moral rights 109 include the right to be named as a creator of intellectual property (IP), and the right to avoid derogatory treatment of creations. For the data scientist the items requiring protection, in possibly different ways, include software, designs including graphical user interfaces (GUIs), data sets, moral rights and reputation. Trade secrets may also be relevant. Knowledge
● Patents, copyrights, trademarks, trade secrets, moral rights and trademarks
● What data science related IP can and cannot be protected, and what kinds of protection are available
● Types of data science related IPs that can and cannot have legal protection and which kind of protection is available

● Regulation related to IP, IP ownership, the territorial nature of IP rights including the effects of international agreements (e.g. the European Directive on trade secrets) and the issue of IP rights being time limited
● Kinds of IP rights that vest automatically and which require registration, including overview of the processes involved in acquiring registered IP rights
● Possibility of infringing the rights of others and validly utilizing protected IP Skills
● Describe those kinds of IP that are relevant to data scientists.
● Argue the difference between patents, copyrights, designs and trademarks and illustrate their use in the context of data science.
● Describe the role of trade secrets in relation to data science.
● Illustrate the processes involved in registering IP rights.
● Explain the issues relating to IP ownership and moral rights.
● Evaluate the risks involved in using protected IP and ways to overcome them validly.
Dispositions
● responsive and astute to the existence and importance of, as well as responsibilities and opportunities afforded by, intellectual property. Contextual issues
● Thoroughness and adaptability dealing with ethical and legal frameworks associated with intellectual property will vary from one country to another. Patent attorneys can typically advise.

PR-On Automation – E

Automation often creates concerns about loss of employment and, in general terms, about machines behaving unreasonably. Professionals should seek explanations about machine behaviour. Related issues are the subject of [3] and [6]. Automation can occur in critical situations where serious loss may be possible, and then typically there is an expectation that machines will operate according to a code of ethics that is in harmony with human behaviour.
Knowledge
● Automation, its benefits and its justification ● The particular concerns of automation in critical situations
● Transparency and accountability in algorithms
Skills
● Explain to a non-technical audience the extent to which automated decision making occurs in a particular situation.
● Analyze the impact on a design requirement to provide insights into decisions made autonomously by machines.
● Argue the benefits of automation for different situations.
● Identify steps needed to ensure that a decision-making system is auditable.
Dispositions
● Responsive and astute to issues of automation and its effect on employment.
● Respectful and ethical approach to issues of automation.

References [1] The ACM Code of Ethics and Professional Conduct, published by ACM on 17th July 2018. m See acm.org [2] When computers decide: European Recommendations on Machine-Learned Automated Decision Making, published by ACM, 2018. See europe.acm.org [3] ACM US Public Policy Council and ACM Europe Policy Council, “Statement on Algorithmic Transparency and Accountability,” 2017. [4] Directive (EU) 2016/943 on protection of undisclosed know-how business

information (trade secrets) against their unlawful acquisition, use and disclosure. See eur-lex.europa.eu June 2016. [5] The EU General Data Protection Regulation, see www.eugdpr.org. Approved by the EU on 14th April 2016 with an implementation date of 25th May 2018. [6] Simson Garfinkel, Jeanna Mathews, Stuart S. Shapiro, Jonathan M. Smith, Towards Algorithmic Transparency and Accountability, Communications of the ACM, September 2017, vol. 60, no. 9, page 5.

1.10 Programming, Data Structures, and Algorithms (PDA)

Data scientists should be able to implement and understand algorithms for data collection and analysis, as well as integrate them with existing software and/or tools. They should understand the time and space considerations of algorithms, as well as particular issues around numerical computing.
Note that this knowledge area draws from various CS2013 knowledge areas but does not duplicate them: Algorithms and Complexity (AL), Computational Science (CN), Programming Languages (PL), and Software Development Fundamentals (SDF).

Scope	Competencies
● Problem solving through algorithmic thinking. ● Development and implementation of programs, including integration with various existing software and/or tools. ● Use of traditional programming languages to integrate existing interfaces between datasets and applications. ● Use of a programming language designed for statistical computing in the context of a data science problem. ● Knowledge and use of Abstract Data Types (ADTs) ● Knowledge and use of numerical computing algorithms ● Algorithm design and analysis ● Factors that influence algorithmic complexity and performance ● Complexity analysis and comparison	● Design an algorithm in a programming language to solve a small or medium size problem. ● Write clear and correct code in a programming language that includes primitive data types, references, variables, expressions, assignments, I/O, control structures, functions, and recursion. ● Implement good documentation practices in programming. ● Use techniques of decomposition to modularize a program. ● Use standard libraries for a given programming language. ● Write appropriate database queries. ● Select appropriate data structures for a given problem. ● Select appropriate algorithms for a given problem. ● Discuss the importance of time and space complexity on the practical utility of an algorithm.

Sub-domains	
PDA-Algorithmic Thinking & Problem Solving – T1, T2 PDA-Programming – T1, T2, E PDA-Data Structures – T1, T2, E	PDA-Algorithms – T1, T2, E PDA-Basic Complexity Analysis – T1, T2 PDA-Numerical Computing – T1, T2

PDA-Algorithmic Thinking & Problem Solving

In order to develop correct, efficient, clear, and usable code -- either in the process of data analysis and presentation or for production-level systems -- a data scientist should have fundamental algorithmic problem-solving skills.
Knowledge
T1:
● Definition of an algorithm
● Importance of algorithms in the problem-solving process
● At least one formal technique for approaching problem solving
● Fundamental object-oriented design concepts and principles
○ Abstraction
○ Encapsulation and information hiding
○ Separation of behavior and implementation
Skills
T1:
● Describe a problem solution using a formalism other than code (e.g., flowcharts or pseudocode).
● Diagram the flow of data (input, transformations, output) through a problem solution in some formalism (e.g., a data flow diagram).
● Identify the inputs (e.g., data, hyperparameters, user responses) and outputs essential to implementing a program to solve a problem
● Identify the data components and behaviors of multiple abstract data types (See PDAData Structures).
T2:
● Use at least one formal technique for approaching problem solving.
Dispositions
T1:
● Accurate descriptions of algorithms and programs, that algorithms are different from programs.
● Accurate understanding that there are principled approaches for breaking large problems into implementable solutions and expressing those solutions in some formalism.

PDA-Programming

In order to collect, analyze, and present data, a data scientist needs to develop programming skills and should be well-versed in fundamental programming constructs. Because the data scientist will interface with many systems, they should be able to develop programs that can either stand alone or integrate with existing software and/or tools.
Knowledge
T1:
● Core coding concepts
○ Variables and primitive data types
○ Expressions and assignments
○ Conditional and iterative control structures
○ Recursive functions
○ Functions and parameter passing
○ Simple I/O, including files or other static data sources ○ Exceptions
● Core practices
○ Documentation

○ Testing
○ Version Control
● Decomposition to break a program into smaller pieces
● Types of errors (syntax, logic, runtime), how they might occur, and how they can be handled
● Methods for querying and parsing data sources
T2:
● Regular refactoring and program maintenance
● Variety of strategies for testing and debugging
● Utility of APIs; when to look for one Elective
● Advanced concepts
○ in-line/anonymous functions (e.g., Lambda functions in Python) ○ variable argument lists for functions and programs
○ classes and objects Skills
T1:
● Write programs that include core concepts and practices listed above. ● Deduce the execution of code segments and articulate summaries of their computation.
● Apply techniques of decomposition to break a program into smaller pieces.
● Manipulate data from selected sources (e.g., databases, spreadsheets, text documents, XML) utilizing appropriate techniques (e.g., database queries, API calls, regular expressions).
● Construct program solutions using recursion and iteration. ● Use consistent documentation and program style standards that contribute to the readability and maintainability of software.
● Apply strategies for testing and debugging programs.
T2:
● Describe the need for regular refactoring and program maintenance.
● Carry out refactoring, maintenance, and improvements on programs following core practices.
● Construct program solutions using classes and objects.
● Develop programs using a modern IDE and associated tools such as unit testing tools and visual debuggers.
● Construct programs using standard libraries available with a programming language.
● Integrate typical Application Program Interfaces (APIs) into software. Elective:
● Effectively design and implement program solutions using templates and generic functions. ● Design and implement unique Application Program Interfaces (APIs). ● Collect and parse data using specialized techniques (e.g. for natural language processing, image processing, etc.). (x-ref KA: Data Acquisition, Management, and Governance) ● Read, understand, write and debug programs that include advanced concepts. Dispositions T1: ● Strong commitment to using software engineering concepts and design principles on the practice of programming. (x-ref KA: Software Development and Maintenance.)
● Proactive in going beyond what has been directly taught. Appreciate that programming constructs and methods are general and useful in many contexts.
● Look beyond simple solutions and be inventive. A data scientist should not be bound by tweaking existing solutions. PDA-Data Structures In order to write effective and efficient code, a data scientist should know a variety of data structures, be able to use them, and understand the implications of choosing one over another. Given

their role in many data science applications, particular attention is given to matrix representations and operations here.
Knowledge
T1:
● Basic data structures and
Abstract Data Types (ADTs) (lists, arrays, stacks, queues, strings, sets, records/structs, maps, hash tables)
○ purpose
○ usage
● Basic matrix representation structures (sparse/dense, row, column)
○ matrix representation types
○ pros/cons of basic matrix operations based on representation types
T2:
● Advanced structures (trees, graphs)
○ purpose
○ usage
Elective:
● Matrix operation optimization
Skills
T1:
● Select basic data structures appropriately in programming
● Appropriately use standard data type libraries for a given programming language
T2:
● Select advanced data types appropriately in programming
● Appropriately use standard libraries for a given programming language Elective:
● Implement a coherent abstract data type, with loose coupling between components and behaviors
● Compare/contrast the time/space of standard operations (e.g., find, insert, delete) for various data structures
Dispositions
T1:
● Thoroughness in implementation and data structure choice and their impact on usage, efficiency (time and space), and readability.

PDA-Algorithms
A data scientist should recognize that the choice of algorithm will have an impact on the time and space required for a problem. A data scientist should be familiar with a range of algorithmic techniques in order to select the appropriate one in a given situation.
Knowledge
T1:
● Simple numerical algorithms, such as computing the average of a list of numbers, finding the min, max, or mode in a list
● Sorting and Searching ○ Sequential and binary search
○ O(n2) (e.g., Insertion) versus O(n log n) (e.g., Merge) sorts.
○ Randomized algorithms for searching and sorting (e.g., Quicksort)
○ Potential efficiency benefits of hash-based searching and sorting
● Properties of graphs: connectedness, betweenness, centrality, etc.
● Graph algorithms
● Basic algorithmic strategies, such as greedy, divide-and-conquer

● Algorithms for solving linear systems
T2:
● Algorithms for combinatorial optimization problems
● Heuristic optimization techniques
Elective:
● Hashing and hash functions
Skills
T1:
● Apply simple numerical algorithms (e.g., computing the average, finding the min, etc.).
● Apply searching and sorting algorithms.
● Contrast the trade-offs of various array-based searching and sorting algorithms.
● Perform a graph or tree traversal using the general framework of a breadth or depth first algorithm.
● Identify a shortest path in a graph or tree using an efficient algorithm, such as a greedy algorithm.
● Apply linear system solvers to appropriate problems.
T2:
● Identify a max- or min-flow through a graph or tree using an efficient algorithm.
● Use common algorithms for combinatorial optimization problems (e.g., Branch and Bound algorithms)
● Apply heuristic optimization techniques (Particle swarm, genetic algs, evolutionary) to appropriate problems.
● Implement Dynamic Programming solutions for appropriate problems.
Elective:
● Implement or use search/sort algorithms on distributed systems or data
● Compare hashing functions in context.
● Graphs
○ Implement traversal, shortest path, and flow algorithms
● Analyze randomized algorithms
Dispositions
T1:

- Agile and accurate when selecting algorithmic techniques. Be aware that there are often a variety of algorithmic techniques that can successfully address a problem.
- Astute that the choice of algorithm has significant implications for efficiency.
- Astute about the implications of efficiency (time, space, etc.) for all code stake-holders such as clients, consumers, and maintainers. PDA-Basic Complexity Analysis Data scientists should be aware of the time and space required to solve a problem and should know that certain problems may not be solvable in a reasonable amount of time. They should also take into consideration how the platform on which they may be running their code will schedule their tasks.

Knowledge
T1:
● Definitions of time and space complexity
● Differences among best, expected, and worst case behaviors of an algorithm
● Trade-offs in managing time and space complexity
● Taxonomies for analyzing algorithms, such as ○ Deterministic vs. Non-Deterministic ○ Time/Space hierarchies

Skills
T1:
● Perform informal comparison of algorithm efficiency (e.g., operation counts).
● Execute algorithms on input of various sizes and compare performance.
● Demonstrate, via examples, that implementation and algorithm choice have an effect on execution time or space.
● Explain how problem representations / data structures and algorithms are related/coupled.
T2:
● Formally apply a variety of classification taxonomies to understand algorithms.
Dispositions
T1:
● Thoroughness in evaluating space/time complexity. There may be trade-offs in managing time and space complexity and appreciate the implications of those trade-offs for clients/users of software.

PDA-Numerical Computing
The types of problems data scientists solve often involve numerical computing. Data scientists should be aware of the power and limits of numerical representations. They should also be aware of standard numerical computing algorithms and their uses.
Knowledge
T1:
● Random Number Generators (RNGs)
● Simulation of probability distributions
● Limitations of numerical representations with bits, and their impact on the accumulation of error (overflow, underflow, round off, truncation) in results
● Implications of numerical representations with respect to their computational complexities
T2:
● Algorithmic and mathematical methods involved in advanced numerical algorithms for data analysis, such as:
○ Principal Component Analysis (PCA) ○ Singular Value Decomposition (SVD)
○ Eigenvalue decompositions
○
Newton's Method
○ Monte Carlo Simulation
● Connection between good problem representations and mathematical models for solving numerical problems.
For example:
○ The use of SVD in representing documents
○ The representation of graphs as adjacency lists or sparse matrices
○ The use of kd-trees to represent metric spaces
Skills
T1:
● Describe how numerical computing algorithms and processes affect the execution of simulations, data sampling, and data generation.
● Describe appropriate numerical computing algorithms to perform data analysis with a recognition of their limitations and numerically driven constraints.

● Use random number generators and simulated probability distributions to
○ Allow reproducibility in data analysis with non-deterministic algorithms
○ Introduce non-determinism into algorithms to ensure proper statistical and numerical conditions
T2:
● Apply appropriate numerical algorithms for solving a variety of problems. Algorithms may include (non-exhaustive, non-ordered):
○ Principal Component Analysis (PCA)
○ Singular Value Decomposition (SVD)
○ Eigenvalue decompositions
○ Newton's Method
○ Monte Carlo Simulation
Dispositions
T1:
● Astute about benefits and limitations of (pseudo)-random number generation
● Astute about the limitations of numerical computing algorithms

1.11 Software Development and Maintenance (SDM)

Data scientists may be expected to build (or contribute to building) deployable systems either for the purposes of data analytics or to put into practice the results of data analytics. To this end, they should be familiar with fundamental software development principles and practices.

Note that this knowledge area draws from the CS2013 knowledge area on Software Engineering (SE).

Note that development and testing are addressed separately below. Testing is integral to the development process. They are separated below only for purposes of readability.

Scope	Competencies
● Software engineering principles, including design, implementation and testing of programs. ● Potential vulnerabilities	● Implement a small software project that uses a defined coding standard. ● Test code by including security, unit testing, system testing, integration testing, and interface usability.

Sub domains	
SDM Software Design and Development – T1, T2, E	SDM Software Testing – T1, T2, E

SDM-Software Design and Development

A data scientist should understand design principles and their implications for issues such as modularization, reusability, and security. Design, implementation, and testing are tightly integrated components of software development. In this KA, we itemize design and testing competencies separately for the sake of readability.

Knowledge

T1:

● Coding and Design Standards
● Integration with Information Management/Database Systems
● Software lifecycle
● Data lifecycle

T2:

● Project management methodology

Elective:

● Integration with Embedded, Process Control, and/or Communications systems

Skills

T1: ● Explain project Coding Standards
● Explain project Design Standards ● Describe how to integrate or interact with Information Management/Database Systems
● Explain the scope and types of different testing paradigms/needs for all areas. [x-ref Testing below]
● Individually, implement a small software project that meets design specifications
● Develop to completion a team software project that meets design specifications
● Implement given design, documentation, and implementation standards
● Execute a basic Software Lifecycle on a simple program
● Execute a basic Data (Science) Lifecycle on a simple data product
● Integrate or interact with Information Management/Database Systems

T2:

● Execute a given project management methodology
● Plan and design a team software project that meets stakeholder specifications
● As leader, develop a project to completion, meeting stakeholder requirements
● Implement the data science lifecycle to build data-driven decisions in appropriate stages of the software lifecycle Elective:
● Integrate or interact with Embedded, Process Control, and/or Communications systems Dispositions

T1:

● Collaborative and ethical team member, recognizing the value of a team built on respect, diversity, and collaboration
● Conviction to adhering to project Coding and Design Standards
● Collaborative and flexible, through good listening skills, the ability to present an idea, and the ability to negotiate
● Strong commitment to approach data and software projects with a lifecycle mindset
● Astute about the benefits of using test-driven development [x-ref Testing below]

T2:

● Professional and ethical leadership. Lead a project to completion following principles of respect, good listening, responsibility, etc.
● Commitment and professionalism in promoting and encourage adherence to project

Coding and Design Standards SDM-Software Testing

A data scientist should understand the importance of good testing in software development and deployment. Knowledge

T1:

● Testing paradigms/needs for
  - Unit/Execution
  - Integration
  - Interface/User
  - Regression/Continuous
  - System 121
  - Security

T2:

● Potential security problems in programs
  - Buffer and other types of overflows
  - Race conditions
  - Improper initialization, including choice of privileges
  - Not checking input
  - Assuming success and correctness
  - Not validating assumptions

Skills

T1:

● Define and explain the scope and types of different testing paradigms/needs for all areas.

● Design basic tests for:
  - Unit/Execution
  - Integration

T2:

● Use or extract representative data from Big Data datasets in order to test algorithms on a small scale before running at scale on a cluster, for example.

● Develop test specifications for:
  - Interface/User
  - Regression Testing
  - System
  - Security

● Execute tests (built by others) for:
  - Interface/User
  - Regression Testing
  - System
  - Security

● Evaluate the results of a program using statistical significance testing

● Describe possible types of risks for a software system

● Describe secure coding and defensive coding practices

Elective:

● Design, develop, and execute tests for all areas

Dispositions

T1:

● Astute about recognizing and value the benefits of using test-driven development.

● Commitment to basic software and data project development from a test-driven perspective, particularly as it pertains to unit/execution and integration tests

T2:

● Commitment to software and data project development from a test-driven perspective, particularly as it pertains to Security, Interface/User, Regression/Continuous, and System tests

Elective:

● Commitment to approaching software and data projects holistically from a test-driven development perspective