The Bachelor of Science in Data Science studies the collection, manipulation, storage, retrieval, and computational analysis of data in its various forms, including numeric, textual, image, and video data from small to large volumes. The program combines computer science, information science, mathematics, statistics, and probability theory into an integrated curriculum that prepares students for careers or graduate studies in big data analysis, data science, and data analytics. The coursework covers exploratory data analysis, data manipulation in a variety of programming languages, large-scale data storage, predictive analytics, machine learning, data mining, and information visualization and presentation. Data science has emerged as a discipline due to the confluence of two major events:

1. The ability to collect, store, prune, process, and transmit large amounts of data in the cloud, and
2. The convergence of programming, statistics, artificial intelligence, and visualization as complementary tools for the analysis and understanding of data.

**Courses**

**Search DS Courses using FocusSearch** ([http://catalog.northeastern.edu/undergraduate/class-search/?subject=DS](http://catalog.northeastern.edu/undergraduate/class-search/?subject=DS))

**DS 1990. Elective. 1-4 Hours.**
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

**DS 2000. Programming with Data. 2 Hours.**
Introduces programming for data and information science through case studies in business, sports, education, social science, economics, and the natural world. Presents key concepts in programming, data structures, and data analysis through Python and Excel. Integrates the use of data analytics libraries and tools. Surveys techniques for acquiring and programmatically integrating data from different sources. Explains the data analytics pipeline and how to apply programming at each stage. Discusses the programmatic retrieval of data from application programming interfaces (APIs) and from databases. Introduces predictive analytics for forecasting and classification. Demonstrates the limitations of statistical techniques.

**DS 2001. Data Science Programming Practicum. 2 Hours.**
Applies data science principles in interdisciplinary contexts, with each section focusing on applications to a different discipline. Involves new experiments and readings in multiple disciplines (both computer science and the discipline focus of the particular section). Requires multiple projects combining interdisciplinary subjects.

**DS 2500. Intermediate Programming with Data. 4 Hours.**
Offers intermediate to advanced Python programming for data science. Covers object-oriented design patterns using Python, including encapsulation, composition, and inheritance. Advanced programming skills cover software architecture, recursion, profiling, unit testing and debugging, lineage and data provenance, using advanced integrated development environments, and software control systems. Use case studies to survey key concepts in data science with an emphasis on machine-learning (classification, clustering, deep learning); data visualization; and natural language processing. Additional assigned readings survey topics in ethics, model bias, and data privacy pertinent to today's big data world. Offers students an opportunity to prepare for more advanced courses in data science and to enable practical contributions to software development and data science projects in a commercial setting.

**DS 2501. Lab for DS 2500. 1 Hour.**
Practices the programming techniques discussed in DS 2500 through hands-on experimentation.

**DS 2990. Elective. 1-4 Hours.**
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.

**DS 2991. Research in Data Science. 1-4 Hours.**
Offers an opportunity to conduct introductory-level research or creative endeavors under faculty supervision.

**DS 3000. Foundations of Data Science. 4 Hours.**
Introduces core modern data science technologies and methods that provide a foundation for subsequent Data Science classes. Covers: working with tensors and applied linear algebra in standard numerical computing libraries (e.g., NumPy); processing and integrating data from a variety of structured and unstructured sources; introductory concepts in probability, statistics, and machine learning; basic data visualization techniques; and now standard data science tools such as Jupyter notebooks.

**DS 3500. Advanced Programming with Data. 4 Hours.**
Offers a deep dive into the design and implementation of enterprise-grade software systems with an emphasis on software architectures for more complex data-driven applications. Covers extensible architectures that support testing, data provenance, reuse, maintainability, scalability, and robustness and building software APIs and libraries for wide-scale adoption and ease of use. Students design, implement, and test complex loosely coupled service-oriented architectures using distributed processing, stream-based data processing, and interprocess communication via message passing. Explores the features, capabilities, and underlying design of popular data analysis and visualization frameworks.

**DS 3990. Elective. 1-4 Hours.**
Offers elective credit for courses taken at other academic institutions. May be repeated without limit.
various types of data, including images and text. Implementations, respectively. Reviews applications of these models to fitting techniques. Emphasizes using these technologies in practice, via stochastic gradient descent and backpropagation, along with related feed forward networks to recurrent neural networks. Also covers networks and introduces standard and new architectures from simple linear algebra that support algorithms, including sampling theory and trees. Studies computational aspects of probability, statistics, and linear algebra that support algorithms, including boosting, and decision regularization, multiclass data and algorithms, and preprocessing. Includes visualizations. Creates visualizations in Excel and Tableau as well as R, Python, and open web-based authoring libraries. Requires programming in Python, JavaScript, HTML, and CSS. Requires extensive writing including documentation, explanations, and discussions of the findings from the data analyses and the visualizations.

Introduces data and information storage approaches for structured and unstructured data. Covers how to build large-scale information storage structures using distributed storage facilities. Explores data quality assurance, storage reliability, and challenges of working with very large data volumes. Studies how to model multidimensional data. Implements distributed databases. Considers multilayer storage design, storage area networks, and distributed data stores. Applies algorithms, including graph traversal, hashing, and sorting, to complex data storage systems. Considers complexity theory and hardness of large-scale data storage and retrieval. Requires use of nonrelational, document, key-column, key-value, and graph databases and programming in R, Python, and C++.

Introduces supervised and unsupervised predictive modeling, data mining, and machine-learning concepts. Uses tools and libraries to analyze data sets, build predictive models, and evaluate the fit of the models. Covers common learning algorithms, including dimensionality reduction, classification, principal-component analysis, k-NN, k-means clustering, gradient descent, regression, logistic regression, regularization, multiclass data and algorithms, boosting, and decision trees. Studies computational aspects of probability, statistics, and linear algebra that support algorithms, including sampling theory and computational learning. Requires programming in R and Python. Applies concepts to common problem domains, including recommendation systems, fraud detection, or advertising.

Continues with supervised and unsupervised predictive modeling, data mining, and machine-learning concepts. Covers mathematical and computational aspects of learning algorithms, including kernels, time-series data, collaborative filtering, support vector machines, neural networks, Bayesian learning and Monte Carlo methods, multiple regression, and optimization. Uses mathematical proofs and empirical analysis to assess validity and performance of algorithms. Studies additional computational aspects of probability, statistics, and linear algebra that support algorithms. Requires programming in R and Python. Applies concepts to common problem domains, including spam filtering.

Offers a hands-on introduction to modern neural network ("deep learning") tools and methods. Covers the fundamentals of neural networks and introduces standard and new architectures from simple feed forward networks to recurrent neural networks. Also covers stochastic gradient descent and backpropagation, along with related fitting techniques. Emphasizes using these technologies in practice, via modern toolkits. Specifically introduces Keras (together with TensorFlow) and PyTorch, which are illustrative of static and dynamic network implementations, respectively. Reviews applications of these models to various types of data, including images and text.
DS 5020. Introduction to Linear Algebra and Probability for Data Science. 4 Hours.
Offers an introductory course on the basics of statistics, probability, and linear algebra. Covers random variables, frequency distributions, measures of central tendency, measures of dispersion, moments of a distribution, discrete and continuous probability distributions, chain rule, Bayes’ rule, correlation theory, basic sampling, matrix operations, trace of a matrix, norms, linear independence and ranks, inverse of a matrix, orthogonal matrices, range and null-space of a matrix, the determinant of a matrix, positive semidefinite matrices, eigenvalues, and eigenvectors.

DS 5110. Introduction to Data Management and Processing. 4 Hours.
Discusses the practical issues and techniques for data importing, tidying, transforming, and modeling. Offers a gentle introduction to techniques for processing big data. Programming is a cross-cutting aspect of the course. Offers students an opportunity to gain experience with data science tools through short assignments. Course work includes a term project based on real-world data. Covers data management and processing—definition and background; data transformation; data import; data cleaning; data modeling; relational and analytic databases; basics of SQL; programming in R and/or Python; MapReduce fundamentals and distributed data management; data processing pipelines, connecting multiple data management and analysis components; interaction between the capabilities and requirements of data analysis methods (data structures, algorithms, memory requirements) and the choice of data storage and management tools; and repeatable and reproducible data analysis.

DS 5220. Supervised Machine Learning and Learning Theory. 4 Hours.
Introduces supervised machine learning, which is the study and design of algorithms that enable computers/machines to learn from experience or data, given examples of data with a known outcome of interest. Offers a broad view of models and algorithms for supervised decision making. Discusses the methodological foundations behind the models and the algorithms, as well as issues of practical implementation and use, and techniques for assessing the performance. Includes a term project involving programming and/or work with real-life data sets. Requires proficiency in a programming language such as Python, R, or MATLAB.

DS 5230. Unsupervised Machine Learning and Data Mining. 4 Hours.
Introduces unsupervised machine learning and data mining, which is the process of discovering and summarizing patterns from large amounts of data, without examples of data with a known outcome of interest. Offers a broad view of models and algorithms for unsupervised data exploration. Discusses the methodological foundations behind the models and the algorithms, as well as issues of practical implementation and use, and techniques for assessing the performance. Includes a term project involving programming and/or work with real-life data sets. Requires proficiency in a programming language such as Python, R, or MATLAB.

DS 5500. Information Visualization: Applications in Data Science. 4 Hours.
Offers students an opportunity to develop effective communication skills with data by drawing from different disciplines including physics, biology, health science, social science, geography, business, and economics. Introduces principles of effective oral and written communication and a wide range of visual data encodings and representations. Covers the foundational principles for visual representations, including human vision and perception and basic interactivity. A semester-long project requires students to translate the domain science or technology problem into the language of data science; design, evaluate, implement, and deploy both static and interactive visualizations of data and data analysis results; translate the results into the language of the original science or technology problem; communicate the findings in oral and written form; and provide constructive criticism of other examples of data communication and visualization.