INFO 1590: Data Science OnRamp

Instructor

Professor Ying Ding
Associate Director of Data Science Online Programs
Email: dingying@indiana.edu
Phone: (812) 855-5388

Course Outline and Objectives

This course is comprised of self-paced modules, spread over ~two week periods. Each unique lesson introduces students to a different level of understanding amongst various programming and database skills and will cover C++, MongoDB, R, Java, Python, Tableau, SQL, Hadoop/MapReduce, Spark, Scala, Github, Web Scraping, and Text Mining (NLP). These lessons will assist in building foundational and strengthening core competencies necessary for current and future Data Science curricula.

Lessons -

**Pick any one (1) lesson for successful completion of 1 credit hours, two (2) lessons for successful completion of 2 credit hours, or three (3) lessons for successful completion of 3 credit hours per term. Course may be repeated twice, maximum 6 credit hours **

This is a comprehensive list explaining the lesson structures, list of learning objects, and difficulty level for each.
Beginner-level Courses:

Introduction to R

The goal of this course is to teach you how to program in R and how to use R for effective data analysis. We will also cover in detail the plotting systems in R as well as some of the basic principles of constructing data graphics. The course consists of 3 parts with exercises, quizzes in each chapter and a final project of student's choice.

1. First we cover basics of R programming - Data Structures supported by R, Control structures, Functions, Debugging techniques. These topics will help you get started with R and help you in writing good reusable R codes. This will also serve as basis for what is taught in the later sections.

2. The second part is the major focus of the course, exploratory data analytics and visualization. In this section you will learn how to manipulate the data. You will also learn how to use base plotting system of R, lattice and ggplot packages to plot graphs. We will not only teach you how to plot graphs in R but also provide general rules you can follow to build appropriate graphs and be able to tell the story of what is happening in the data using graphs.

3. In this section we will teach how to showcase your work. First you will learn how to use R markdown to make documents that can be made into a webpage. Then you will learn to make web apps and showcase your work using shiny package. By the end of this course you will be making your very own web app!

Course Structure:

- Getting started with R (installing R and R studio, writing some basic commands, and installing libraries in R)
- Data structures supported by R
- Operators, control structures, and functions
- Introduction of some utilities, including basic statistics in R and basic debugging
- How to tidy and manipulate the data sets in R
- How to make plots in R
- The R work showcase-R markdown and shiny package

Basics of Python

In this course, we are going to learn about the basics of python. In industries, most of the computer programmers use two important approaches to writing complex applications, recursive approach, and iterative approach. You will learn about these important concepts from modules as well as from programming assignments. It is good to have hands on experience of the concepts you have learned so, at the end, you will
get introduced to one of the most commonly performed tasks of classification in data mining and machine learning and we will implement a recommendation system using Scikit-learn package. In this course, we ill also introduce you to other useful packages, like Matplotlib and Scikit-learn, which are being widely used in the industry.

**Course Structure:**

- What is Python and how to install it
- Basic operators in python
- Data structures and metrics in Python
- Some basic algorithm concepts including statements, functions and recursion
- Two basic python packages: Numpy and Pandas
- How to visualize in Python
- Basic machine learning concepts
- Run regression model ins Python

**Tableau**

Tableau is a leading data analysis software used by analytics, banking and consulting organizations for data analysis. Tableau helps users to design/develop/deploy data science algorithms without writing huge chunks of code. The visualization of data joins and merges provides an easy way for a non-technical user to work on Data without worrying about coding in traditional scripting languages. In this course, we will learn about Tableau visualization from scratch to a professional level of understanding. We will also understand the techniques for building effective visualizations on various public data sets. The course consists of bi-weekly assignments which mainly focus towards a target problem and building visualization to discovery significant insights. There is also a final project for students to apply knowledge for a practical dataset and present their story-telling skills through interesting data visualizations.

This course will enable all the students gain all the important skills needed for building data visualizations and effective story telling. It will make the students proficient in using the tableau visualization tool and build impressive visualization story boards in their professional careers.

**Course Structure:**

- Introduction to data visualization and its usage
- Familiarizing with the Tableau visualization tool
- Importing data in tableau, working with sample data set, exploring features in tableau. Building simple visualizations in tableau
- Working on features like filters
- Effective use of Details feature, sorting options, view tool bar, worksheet options.
- Creating dashboard and worksheets
- Creating calculated fields, groping set, creating hierarchy
• Working with Time Series data set
• Building effective geo maps and other custom visualizations
• Implementing K-means clustering and classification, prediction in tableau
• Final Project

**Introduction to SQL**

This is a course on SQL (Structured Query Language) which covers the basic concepts of databases and SQL as a language for accessing databases. This course is intended for beginners and assumes no background whatsoever in databases/SQL. We will start the course by introducing the need to use databases, building entity-relationship models, installing MySQL and then we will proceed to programming (querying) the database using SQL. At the end of the course, you will work to create a sample database, input data and perform complex SQL operations on the database to retrieve the desired output. The assignments are included to provide a working knowledge of the concepts taught throughout the course.

**Course Structures:**

• Introduction to databases
• Entity-Relationship Model basics and examples
• Introduction to SQL including examples of CREATE, INSERT, UPDATE, DELETE and DROP statements
• Introduction to MySQL
• Examples of specifying constraints like UNIQUE, NOT NULL, Primary and Foreign keys
• Mapping of entities to tables using SQL
• Write basic SQL queries using SELECT and use of AND/OR and LIKE operators.
• JOINs in SQL
• Operators in SQL – Union, INTERSECT and EXCEPT
• Views in SQL
• Discussion on sub queries and introduction to aggregate functions
• Modeling SQL queries using ORDER BY, GROUP BY and HAVING clauses
• Brief explanation on transaction management using SQL
• A SQL demo by an example of a UNIVERSITY database

**Basics of Java**

Java is one of the most fundamental programming languages used in industry. In all programming rankings, Java always ranks in the top 3, which can clearly show its importance. One advantage for Java is that Java is an object oriented programming language, which makes it easier to integrate Java codes together. As it runs on JVM, it is so easy to set up the running environment, which means it is good for code immigration.
This is an intro-level Java course. It is intended for people without any Java programming experience. The course is an interweaving of two parts. The first part is knowledge about basic programming with Java, and the second part is to use the knowledge to work on a project. The two parts will not be strictly separated. Once we are equipped with enough knowledge to take a small step, we make some progress in our project. In the process, we may encounter some confusion and wonder what could be done next. Then we learn something new that helps us resolve the problem, and make further progress. This cycle repeats.

**Course Structure:**

- Install Java and run Eclipse
- Basic concepts such as variable and command in Java
- How to write a method in Java
- Basic data structures in Java
- Practice your knowledge with the first project
- Practice your knowledge with the second project
- Latent semantic analysis concepts & code in Java

**Introduction to C++**

R, Python, Java, and MATLAB are the primary languages used by data scientists, but sometimes they are just not fast enough! One of the strategies to optimize them is to rewrite the most time critical code in a language closer to the hardware, including C/C++. Among these languages, **Cython** and **Rcpp** are two main methods to speed up Python and R by C/C++. While Cython is mainly a dialect of Python, Rcpp is to inline C/C++ program into a R program, so Rcpp users really need to write a fraction of programs in C/C++.

This course requires no prior knowledge of any programming languages, and we will not cover programming languages other than C/C++. However, we will go through the C++ appendix of Eddelbuettel's "**Seamless R and C++ integration with Rcpp**," which provides the necessary knowledge of C/C++ for reading how to use Rcpp in the main text. After learning a solid knowledge of standard C/C++, you will have a basic idea how to write a C/C++ program. Whenever you need to optimize their R and Python, you will have enough knowledge to start learning Rcpp and Cython by yourselves.

**Course Structure:**

- Why C++ is important
- How to compile and debug C++ program
- Control structures in C++
- Basic C++ functions
- Pointers and memory management in C++
- What is object oriented and how to use it in C++
- Generic programming and the STL
Intermediate-level Courses:

**Introduction to Spark**

Through this online course, we will introduce you what Apache Spark is, how it can be helpful, and where its power resides. The course is designed to be simple, to the point and instructive for the beginners. We will not be surprised to see many students who has already tried other online tutorials or coerces about Apache Spark, but very soon has found the concepts very confusing. However, here we understand this fact and it is number one priority to express all key concepts in a very straightforward language and try to avoid unnecessary and confusing fancy statements. Additionally, our preference has been to use real world examples to make sure that students actually can imagine how the skills will be helpful in a real-world setting. We want to provide you some hands-on experience by developing simple programs that can be easily deployed in many other situations only by being modified slightly. Moreover, we have tried to make the course easy to proceed by covering every basic concept and skill you need to develop Spark programs so you do not need to look for other resources frequently while taking the course.

**Course Structure:**

- Introduction to Apache Spark
- Apache spark components: Spark Core, Spark SQL, Spark Streaming, Spark MLlib
- Installation of Apache Spark
- Writing your first spark application
- Resilient Distributed Datasets (RDD) in Spark
- Data partitioning in Spark
- Importing and exporting data into Spark
- Accumulators and Broadcast variables
- Spark interaction with R
- Introduction to Spark SQL

**Basics of Scala**

Scala is a very fancy and new programming language. It is pretty popular especially in industry in the recent years. As a functional programming language, it is kind of similar to Java but with more flexibility. It can even run on JVM (Java virtual machine). This course was designed to get you familiar with Scala constructs and features. This course doesn't require any prerequisites but students should have a basic understanding of object-oriented programming. This course uses a data-centric approach to Scala. All content in this course is standard basics in Scala. If you can follow each session closely, you are guaranteed to get some useful knowledge about Scala at the end. And you are able to use Scala to solve some real-world problems.

**Course Structure:**
• Basic background of Scala
• Install Scala in your local environment
• Create a project in Scala IDE
• Scala REPL to run code in terminal
• OOP in Scala
• Write methods in Scala
• What is object in Scala
• Scala-particular basic concepts such as access modifiers and companion objects
• What are case object and case class
• Some synthetic methods
• Collections in Scala
• Sequences and sets in Scala
• Tuple and map in Scala
• Higher order functions in Scala

**Machine Learning with Python**

Machine learning is a technique which is used to teach computers, without being explicitly programmed. In this course, you will learn about basics of python and extending them to use different important packages like Matplotlib, Scikit-learn in python and about different kinds of classifications and classifiers used in machine learning.

We will begin our course with basic python programs because it is good to have some basic level of python experience before we go into advance concepts like machine learning. In industries, most of the computer programmers use two important approaches to write complex applications, recursive approach and iterative approach. You will learn about these important concepts from modules as well as from programming assignments. Scikit-learn is one of the best open source machine learning package in python with large active open source community. We will use this package to learn machine learning in applied fashion. At last, we will show you how you can build recommendation system using Scikit-learn package.

**Course Structure:**

• Introduction to Python
• Installing Python and setting up PyCharm IDE and Anaconda
• Python strings, constants, variables and scope
• Arithmetic and binary operations
• Control structures, functions and exception handling
• Using NumPy and Pandas library in Python
• Introduction to Matplotlib in Python
• Machine learning with Scikit-Learn and Scipy
• Concepts and implementation of linear regression using Numpy and Logistic Regression
• Introduction to Scipy
• Overfitting of curve and Ridge Regression using Python
- K-Means algorithm and its implementation using Scikit-Learn
- Implementation of SVM and Decision tree using Scikit-Learn
- Expectation and Maximization Algorithm and implementing it using Scikit-Learn
- Principal Component Analysis (PCA) and its implementation using Scikit-Learn
- Neural Networks and its implementation using Scikit-Learn

**NLP in Python**

Text mining starts generally with the process of information retrieval. We need to identify the source of data and then collect from this source. General sources are web, blogs, social media platforms, reviews and comments, etc. Once we collect the data, we need to clean the noise in it, such as the removal of duplicate data entries, unwanted information such as url's, image links, etc. There are number of steps involved in denoising the data and this depends on the kind of data that you have at hand. Once we clean the text data, we can apply natural language processing techniques such as parsing, pos tagging, etc.

The whole idea is to convert something not so structured into something meaningful and structured. Once we have such a structured output, we can perform various tasks such as:

- Sentiment analysis
- Topic detection
- Document summarization
- Entity relational modelling
- Pattern recognition
- Predictive analytics
- Text categorization

In this course we will cover some primary concepts in sentiment analysis. The above-mentioned tasks are extremely useful for gaining insights into textual data. We will explore the topics in detail.

**Course Structure:**

- Use regular expression to match string patterns
- Basic linux commend line functions
- Set up python and install packages using pip
- Basic functions in Python
- Basic python function working on strings
- Twitter APT to grab tweets
- Handle Json format and how to deal with it in Python
- Clean a tweet's content by removing non-useful characters
- Use nltk to run semantic analysis on sentences
- Two projects
**Machine Learning with Java**

Machine learning is current one of the hottest topics now. As the data science is deeply involved with machine learning algorithms and programming languages, it is important to master at least one programming language skills to play with machine learning algorithms or to solve real-life problems. In this course, we introduce how to use Java to build machine learning models to solve regression, classification and clustering problems. We also introduce how to evaluate the machine learning models and interpret results. Although in fact there are many Java packages supporting machine learning algorithms, in this course we only focus on the most popular one-Weka, which is a Java package containing many fancy algorithms and is widely used in recent years. One nice thing about Weka is that the package offers not only a Java library so that you can develop your own code to build a model but a well-developed GUI tool so that even for those people who are not familiar with Java, they can even build up a machine learning model very quickly by just clicking several buttons in the GUI.

**Course Structure:**

- What is Weka and install the GUI on your local computer
- What is Java install Java JRE and JDK so that you can run your code in the later sessions
- Input file format in Weka such as ARFF and XRFF
- Generate artificial data in Weka Java package
- Filter data in Weka
- Some basic classification methods using in Weka
- Tree based classification methods in Weka
- Advanced classification methods particularly in Weka
- Basic clustering methods using in Weka

Learn how to visualize your results in Weka

**Advanced-level Courses:**

**Introduction to Hadoop Framework**

Unlike many of the online articles that you may have already seen, here we do not want to talk about how you can improve your resume by acquiring Hadoop MapReduce knowledge and skills, nor do we want to emphasize the importance of Hadoop and MapReduce to the information technology industry, etc. We know that you already understand how important it is from different aspects, in fact that is probably why you are taking this course.

Our goal in this course is trying to teach you some practical skills so you can actually do something cool using Hadoop, like developing a program to rank some documents based on their relevance to a search query. We will start the course in the form of questions and answers, that is we assume that you have already faced with some questions when wanted to learn about Hadoop and MapReduce by yourself, but never
found a clear answer for them. Then we will proceed by introducing different aspects of MapReduce and other systems designed on top of Hadoop. Throughout the course, we will make sure that you get hands on experience by developing simple programs to work on real-world data and scenarios. Moreover, we have tried to make the course easy to proceed by covering every basic concept and skill you need to develop Hadoop and MapReduce programs so you do not to look for other resources frequently while taking the course.

**Course Structure:**

- Basics of MapReduce
- Developing MapReduce programs in Java
- Installing Hadoop on your computer and running your first Hadoop program
- HDFS (Distributed File storage systems) and Yarn concepts
- MapReduce application development and configuration
- MapReduce Job architecture
- Inverted indexing technique for text retrieval
- Graph processing in Hadoop
- Analyzing stack exchange posts dataset using Hadoop
- Introduction to Apache HBase
- Writing MapReduce jobs on HBase
- Introduction to Apache Hive
- Analyzing Stack exchange dataset using Hive
- Final project-Implementing Pagerank algorithm using MapReduce

**Machine Learning with Spark**

Through this online course, we will introduce you how to do Machine Learning on large scale using Apache Spark. The course is designed to be simple, to the point and instructive for the beginners in Spark. We hope you enjoyed the "Introduction to Spark course" which is a prerequisite for the "Machine Learning with spark" course.

The "Machine Learning with spark" course starts with introduction to Linear Algebra and Python in Spark to brush-up your skills. The course discusses the MLlib which is Spark’s scalable machine learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives. The course ends with topics like Text Mining, building a machine learning project pipeline and a final project. Our preference has been to use real world examples to make sure that students can imagine how the skills will be helpful in a real-world setting. We want to give you some hands-on experience by developing simple programs that can be easily deployed in many other situations only by being modified slightly. Moreover, we have tried to make the course easy to proceed by covering every basic concept and skill you need to develop your Machine Learning models in Spark.

**Course Structure:**
• Introduction to Linear Algebra
• Introduction of Python for spark
• Developing word count application of large data set using Spark
• Decision trees implementation in Spark
• Linear regression
• Logistic regression
• Unified view on Linear methods
• Unsupervised machine learning: Clustering
• Text analysis using Spark RDD
• Frequent patterns and occurrences in Spark
• Machine learning pipelines