1 Course Description
There has been an explosion in volume, velocity, and variety of data, and the increased interest in scaling up data collection and analysis has created a huge demand for professionals who can understand the infrastructure for handling huge amount of data. The course, designed for Master of Information Management (MIM) students who are looking for a possible career in big data analysis, focuses on the principles and techniques of big data technologies and the infrastructure components used to support large-scale data science and business intelligence. The course uses a practical approach to understand the concepts in big data using various tools present in the Hadoop ecosystem. We will look at technologies and architectures for parallel and distributed computing, large-scale data warehousing, and scale-out data analytics platforms. We will review business use cases for big data application development and seek to understand how to apply big data techniques to solve these business challenges.

2 Student Learning Outcomes
Upon completion of the course students are expected to:

1. Understand basic architectures for big data applications and analytics.
2. Understand how business data are collected and organized in data warehouses.
3. Understand techniques for deriving data from business data.
4. Have hands-on experience working with relevant big data technologies.
3 Course Materials and Infrastructure

3.1 Course Readings

Pointers to a selection of readings will be provided via ELMS. Most are EBooks that can be accessed, or
we will provide pdfs of selected chapters.

Selections from the following texts:


“Data Intensive Text Processing with MapReduce,” Lin, Jimmy and Dyer, Chris,


“Mastering MongoDB 3.x,” Giamas, Alex. (ISBN 9781783982615)


“Spark for Data Science,” Srinivas Duvvuri; Bikramaditya Singhal (ISBN 9781785884771)


(ISBN 9781785888281)


9781787287631)

9780596550967)


3.2 Systems and Resources

Students are required to bring a laptop to every class session unless otherwise specified, as we will
regularly do in-class tutorials and exercises that require capabilities not available on tablets or
smartphones.

For specific software resources or access to cloud-based resources, students will be alerted via ELMS
with instruction on how to locate/install/access the resources prior to class.

3.3 Communications

Communication outside of class will use Canvas or your umd.edu email account. Make sure you check
your UMD email regularly, or have it forwarded to your personal email so you do not miss any messages.
Course announcements will be posted on Canvas and individual correspondence will be conducted via email. I will make every effort to send announcements with adequate advance notice; failure to receive email announcements will not be considered a suitable excuse for not being informed. Include “INST767” in email subject lines for prompt response; messages without the course number in the subject line may be overlooked. I will typically reply in two business days, usually less. Telephone is not an effective way to contact me.

4 Course Policies

4.1 Prerequisites

Permission of the instructor.

4.2 (For In-Person Classes) Attendance & Student Conduct

In-class participation is part of the course evaluation, and missing class will negatively affect your course grade. If you must miss class, notify me in advance by email and check with your classmates afterward so that you can catch up.

As a graduate student, I expect you are fully capable of behaving professionally in the classroom, which means treating every person who enters our classroom with the respect that you would like to experience yourself. Since you may need letters of reference for future employment, demonstrating your capacity for professional behavior now is also a great strategy to help ensure that your professors and peers are happy to recommend you for the jobs of your dreams! This means that:

- side conversations are discouraged,
- your cell phone must be silenced before the start of class,
- you should be using your electronic devices for class purposes only, and
- disruptive students will be asked to leave and will forfeit the participation grade for the day.

4.3 Academic Integrity

As per University policy.

4.4 Excused Absences

As per University policy.

4.5 Inclement Weather

As per University policy.
4.6 Emergency Preparedness
As per University policy.

4.7 Academic Assistance
As per University policy.

4.8 Disability Accommodations
As per University policy.

4.9 Intellectual Property
As per University policy.

5 Course Schedule
This course outline and schedule is a general guide for the course and is subject to change with (potentially limited) advance notice. Consult ELMS on a regular basis for notifications of changes.

<table>
<thead>
<tr>
<th>Module</th>
<th>Title</th>
<th>Topics Covered</th>
<th>In-Class Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Module</td>
<td>Introduction to Big Data</td>
<td>Introduction and demonstration of Hadoop Basic HDFS commands Using notebooks</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>What is meant by &quot;big data&quot;? Evolution of scalable computing</td>
<td>Introduction and demonstration of Hadoop Basic HDFS commands Using notebooks</td>
</tr>
<tr>
<td></td>
<td>Module</td>
<td>Data distribution and parallel computing</td>
<td>Introduction and demonstration of Hadoop Basic HDFS commands Using notebooks</td>
</tr>
<tr>
<td></td>
<td>Topics Covered</td>
<td>Big data use-cases</td>
<td>How MapReduce Works? MapReduce demonstration</td>
</tr>
<tr>
<td></td>
<td>Module</td>
<td>Hadoop? Hadoop ecosystem overview Navigating through HDFS (Basics of HDFS)</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>Basics of Python3 Basics of pandas Basics of classes and OOP Lambda Basbas and other relevant topics</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Module</td>
<td>Lambda and other relevant topics</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td>2</td>
<td>Module</td>
<td>MapReduce and Data Parallel Computing - 1</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>What is MapReduce? MapReduce Programming Model How MapReduce Works? MapReduce demonstration</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Topics Covered</td>
<td>MapReduce Features (Counters, Sorting, Joins etc.)</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Module</td>
<td>MapReduce Cluster Deployment Execution and Monitoring of MapReduce Clusters</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>Apache Pig</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td>3</td>
<td>Module</td>
<td>MapReduce and Data Parallel Computing - 2</td>
<td>Deploying MapReduce at Scale</td>
</tr>
<tr>
<td></td>
<td>Title</td>
<td>Deploying MapReduce at Scale</td>
<td>Deploying MapReduce at Scale</td>
</tr>
</tbody>
</table>
| 4 | Databases and SQL on Hadoop | Introduction to databases  
Types of databases  
Relational Databases  
HBase  
Apache Hive  
Apache Impala  
Columnar databases  
Hybrid Transaction Analytical Processing (HTAP) | Create, load, and query a Hive database |
|---|---|---|---|
| 5 | NoSQL Databases | Why the need for NoSQL databases?  
Types of NoSQL databases  
Applications of each of those types  
MongoDB Basics  
Apache Cassandra Basics  
Graph databases | Create, load, and query a MongoDB database  
Create, load, and query a Cassandra database |
| 6 | Data stream processing | What is a data stream?  
Stream processing v/s Batch Processing  
Apache Kafka  
Apache Storm | Create a data pipeline using Kafka/Storm |
| 7 | Big Data Applications  
Work on Projects | Examples of big data applications | Work on group project |
| 8 | Apache Spark - 1 | Introduction to Spark  
Spark vs. MapReduce  
Spark RDDs  
Key-Value Pairs in Spark  
Spark Programming Basics  
Saving data for future use by Spark | Create a Databricks account  
Sample Spark programs using RDDs  
Sample Spark programs with DataFrames |
| 9 | Apache Spark - 2 | Advanced Spark Usage  
Accumulators and Broadcast Variables  
Running Spark on Scale  
Spark SQL  
Spark Streaming | Application with Spark SQL  
Application with Spark Streaming |
| 10 | Machine Learning with Spark | Introduction to MLLib  
Regression with MLLib  
Common classification algorithms  
Common clustering algorithms | Application using Spark MLLib |
| 11 | Natural Language Processing with Spark | Introduction to NLP  
Tokenization  
Bag-of-Words Model  
TF-IDF  
Sentiment Analysis Basics  
Topic Modelling  
Latent Dirichlet Allocation  
Word2Vec  
Recurrent Neural Networks(RNNs) | Application with Python Natural Language libraries |
| 12 | Big Data Visualization | Introduction to data visualization  
JavaScript for D3.js  
The world of D3.js | Data visualization using D3.js |
### 6 Assignments

This course provides an overview of information governance and data quality practices and processes.

To practice valuable professional skills, class members will engage in discussions, readings, and collaborative and individual assignments. Discussions will help you develop your ability to reflect about practical issues and discuss these with colleagues. Readings will provide an introduction to topics and exposure to current issues, debates, issues, and solutions. Written and group assignments serve as skill building exercises.

#### 6.1 Guidelines for Preparing Assignments

If the instructions for an assignment aren't clear, ask, don't assume. If you have questions about assignments, ask before they are due; do not count on getting a reply the day the assignment is due.

Prepare a professional document with tables, graphs, and references that support your content where appropriate. Follow all instructions carefully, and ask questions as soon as they arise if you are uncertain about the assignment requirements. Failure to meet document requirements will be penalized as specified in the rubrics for assignments; content that does not match formatting requirements will be subject to additional scrutiny for potential plagiarism.

Assignments that do not follow these specific requirements (where applicable) will lose points on the assignment grade. In addition to punctuality, the grammar, presentation and your ability to follow instructions are very important, as in the real world, so it is essential that you spell check and proofread your documents. For written assignments, proofreading a printed copy of your work is especially effective for finding errors that you might overlook on the screen. Note that standard Word document templates do not meet the criteria below.

**Document requirements:**

- **Use 11pt Times New Roman** for body text.
- All text must be in black, without highlighting or background colors. Be very careful what you copy and paste!
- Documents must use **1.5 line spacing with 1" margins on all sides in 8.5" x 11" (US letter) format.**
- All written content must have a blank line between paragraphs (block format) OR the first line of the paragraph must be indented.
- You may use larger font sizes, sans serif fonts, boldface, and/or italics for title text and section or
table headers, but it must still be in black type.

- On every page, document headers must include your **name** and **UMD email address or team name** on the left and **page numbering** on the right. Exception: for the written portion of the Team Project, cover pages should not include numbers or headers.
- At the end of the document, insert the **word count**, not including references, appendices, or executive summary (where applicable).
- Use APA format for citations and references. **Web resources must always include the URL and date accessed** regardless of what you may see in examples.
- Most assignments can be submitted in multiple formats; see assignment details for specifics.

These requirements do not apply to R outputs, Markdown documents, or Git repositories. You may choose to submit a Markdown document on Git instead of a word processing document, but it must be nicely formatted and easy to read. The final commit on Git documents must be time-stamped before 6PM on the date the assignment is due; the last commit prior to the cutoff time will be graded.

When you prepare assignments or post on the discussion boards be sure to provide proper bibliographical information for any sources referenced, for direct quotations, and for the sources of key concepts or ideas. Check the UMD citation guide for more details: [https://lib.guides.umd.edu/citation](https://lib.guides.umd.edu/citation)

### 6.2 Assignments

This section provides a high-level, preliminary overview of the assignments for this course. The details of each assignment will be provided via ELMS. Note that the specific assignments are subject to modification.

#### 6.2.1 Assignment #1 (Individual) – 10 points

You will be provided with a collection of text data objects that document events that have occurred over time. You will be provided with a list of key dimensions for organization (such as geography, date, or concept), and you will write a MapReduce program to scan the text of each document, add metatags based on the organizational dimensions, and then organize the documents according to a predefined taxonomy (such as “car accidents” involving “teenagers” in “California”).

#### 6.2.2 Assignment #2 (Individual) – 10 points

You will be provided with a collection of structured data sets with information about health care providers and prescriptions over a period of time. Your task is to create a database model for this data using both a Relational database and a NoSQL database and load the data into the two data models.

#### 6.2.3 Assignment #3 (Group) – 10 points

Revise the application developed in Assignment #1 to use Apache Spark in Python instead of MapReduce.

Write a Spark Python application to search the databases for information related to key phrases found in selected documents.
6.2.4 Assignment #4 (Group) – 10 points
Integrating a real-time data stream that can be used to enhance the data that has already been collected and then subsequently be used for developing predictive models about where future events might take place.

6.2.5 Group Project Implementation Plan – 15 points
Provide a project plan and work breakdown structure for implementing the final project.

6.2.6 Assignment #5 (Group) – 10 points
Use Spark MLlib to develop predictive models associated with the data stream.

6.2.7 Assignment #6 (Group) – 10 points
Develop visualizations (e.g., a dashboard) that is updated in real time that reflects the streaming events being captured and indications of predictions of future events.

6.2.8 Group Project Deliverable (Group) – 15 points
The complete package of design document, code, and demonstration of execution together will account for 15 points.

6.2.9 Final Presentation (Group) – 10 points
Your team will present your group project, describe the approach, and demonstrate the execution of the application.

6.2.10 Class Participation (Individual) – 10 points
Your class participation grade includes class attendance, engagement in class, participation in class activities and discussions, and participation in online discussions.

6.3 Grades and Grading
Assignments are due as defined in the syllabus unless otherwise specified. The penalty for late assignments will be 10% within the first 24 hours, and an additional 25% for each week thereafter. An exception is possible in an extreme circumstance in which there is no reasonable way to anticipate or control the situation. Computers crashing, viruses, lost files, etc. are specifically not grounds for an extension.

Grading rubrics for each assignment are provided on Canvas; please take advantage of them as you prepare your assignments to check whether your work meets grading criteria. If you wish to discuss a grade, submit a written explanation of your argument (email) and arrange for a private conversation. Except for unusual circumstances, no appeals will be considered more than two weeks after the graded paper is returned. For final course grades, no appeal will be considered more than two months after the final day of classes.

Unless announced otherwise, assignments submitted by the due date will be graded within 1-2 weeks. Assignments submitted late will receive lower priority and so will take longer to grade. Final grades will be computed based on the scale below and partial points/percentages will be rounded for final grades.

100%+: A+
96 - 99%: A (4.0)
92 – 95.99%: A- (3.7)
88 – 91.99%: B+ (3.3)
84 – 87.99%: B (3.0)
80 – 83.99%: B- (2.7)
75 – 79.99%: C+ (2.3)
70 – 74.99%: C (2.0)
66 – 69.99%: D (1.0)
0 – 65.99%: F (0.0)