# An Exploration of Machine Learning in Libraries

**Craig Boman**[*]

Artificial intelligence (AI) continues to make tremendous leaps forward. Attending coding conferences like PyCon 2018 and PyOhio 2018, attendees like myself witnessed machine learning concepts woven throughout many presentations. Despite this progress, few libraries and fewer librarians are prepared to take full advantage of the benefits of using AI.

One specific challenge that is ripe is improvement of library metadata generation. Libraries, through various vendors as part of the purchasing and acquisitions process, acquire thousands of pieces of metadata for print and digital resources made available to their library users. In cases where an e-book vendor or platform does not include metadata, libraries purchase metadata from vendors that generate metadata or they generate their own (original cataloging). These vendors include two major user-centric metadata types necessary for providing access to library resources: metadata directly describing the resource (a bibliographic record) and supplemental metadata about the author or subjects (authority records). For the increasing majority of born-digital resources, machine learning provides an array of possible tools to help libraries generate metadata for digital resources, allowing cataloging to not only increase the speed of metadata generation but also vastly improve the depth and breadth of subject terms.

## Research Goal

The purpose of my project will be to explore the use of LDA (latent Dirichlet allocation), a type of machine learning model, in the generation of library subject headings. The full-text e-book collection to be used (Project Gutenberg or PG) contains both fiction and nonfiction e-books. The PG e-book data was retrieved and extracted using wget and unzip bash command and were transformed and loaded using a combination of bash and Python functions.

Additionally, this research will describe not only outcomes but also processes taken to begin implementing a machine learning workflow for librarians. The outcomes will be less important than collectively describing my workflow and encouraging the exploration by librarians of machine learning methodologies. In places where explanations are abbreviated for space, readers will be directed to more information.

## Me?

Machine learning is not the realm of just data scientists. I am not a data scientist. I do not have a computer science degree. All I have learned about machine learning has been either on the job or at various developer conferences. Adding to my false sense of confidence are my two years of PhD statistics coursework, which echo many of the predictive algorithms used by machine learning researchers, albeit by less appealing names, like *structural equation modeling* or *factor analysis*. Through these experiences, I am excited to explore the use of machine learning in libraries and help explain some concepts. Put simply, machine learning need not be beyond the reach and understanding of technical-minded librarians like myself. Libraries have access to an increasingly vast

* **Craig Boman** is the Discovery Services Librarian and Assistant Librarian at Miami University Libraries. In addition to writing and speaking on technology and leadership in libraries, he has completed coursework towards an educational leadership Ph.D. at the University of Dayton. Between classes, he organizes technology conferences and hackathons. He is the founding organizer of the Python Dayton Meetup, focused on developing a community around the Python programming language in Dayton, Ohio.

amount of data, for which librarians must take on this burden with bigger and better tools. Machine learning is one of those tools.

For this discussion, it may be practical to cover some concepts that will be repeated often.

## Terms

### Metadata

*Metadata* includes traditional bibliographic MARC records. Library metadata may also be "documentation that describes data," which we make available to our library users.[1] Library metadata most often takes the form of structured data, but could also take the form of unstructured data. Structured library metadata could take the form of MARC, but could also be BIBFRAME, Linked Data, Dublin core, RDF, XML, or any other metadata schema. For the purposes of this discussion, the metric against which we measure our machine learning output will be that of a MARC record. Additionally structured metadata could be external to or be included as part of the overall unstructured data that we call an e-book.

### MARC Record

A *MARC record* is the cataloging standard for bibliographic metadata, informed by RDA and previously AACR2. Despite MARC being ubiquitous and anachronistic,[2] from a postgreSQL database standpoint, I would argue that MARC as a format is effectively dead. Nowhere in the Sierra integrated library system can one point to a specific postgreSQL table containing a MARC record, yet Miami University continues to have 1.8 million MARC records. MARC as a data transmission standard is further diluted by the use of MARC derivatives such as MARC XML or MARC JSON. This is important to consider since traditional MARC21 records will act as a piece of structured training data by which I will encourage a machine to learn or make sense of the unstructured data contained in a full-text e-book.

### Subject Headings

When describing a book, metadata librarians and catalogers use *subject headings*. These subject headings are access points used by library patrons to browse by subject. Catalogers and metadata librarians go to great efforts to describe resources, especially in subject areas for which they are not subject specialists. However, determining what a book is about and determining subject headings is best done by experts or practitioners in that subject. When including emerging fields of knowledge, this further complicates attempts to contextualize resources within larger fields of study.

Most attempts by catalogers to generate subject headings include some amount of bias. In some cases, catalogers are limited to the use of subject headings that are anachronistic vestiges of cultures past. For instance, the term *illegal aliens* as a subject heading received some publicity and political pushback.[3] This further complicates an exploration of generating subject headings through machine learning, but should not necessarily be seen as a reason to avoid the possibility entirely. Subject headings as they stand now also have been noted as containing heteronormative bias.[4] The use of machine learning to generate subject headings will not resolve bias, but it is important to acknowledge this concern as part of the conversation.

According to authors like Thomas Padilla, Chris Bourg, and Safiya Noble, considerations must be made by machine learning researchers to make sure that the new systems we are building do not continue to reinforce systemic oppression and systemic bias.[5] For the purposes of this article, this concern will be paramount. However, more will be written at a later date concerning the full integration of these ethical dilemmas.

### Artificial Intelligence and Machine Learning

This author will use the description used by Chris Bourg stating machine learning is the use of "computer programs and algorithms that can extract/derive meaning and patterns from data."[6] Although machine learning is roughly defined as a subset of AI, preference will be given to the term *machine learning.*

## Brief Literature Review

According to Rong Ge, machine learning encompasses two major approaches. These are supervised and unsupervised machine learning.[7] These two major classifications of machine learning depend on the project goals and type of data of which you are attempting to make sense. For instance, if you are trying to analyze data that is already structured, you are more likely approaching a supervised machine learning problem. While analyzing data or metadata that is entirely unstructured, you are more likely approaching an unsupervised machine learning challenge. Both of these methodologies include distinct technical tools and machine learning (ML) algorithms.

Potential tools among ML researchers include R, Java, Scala, Python, Go, Clojure, Matlab, and Javascript. Among these, Python may be the most accessible. Python offers a unique set of well-documented modules. Oftentimes, the best tool for the job is the tool you are most comfortable with. Because I was already learning Python and am comfortable with Bash or Linux command line scripting, much of the justification for my tools may result from familiarity, rather than objectively being the best tool for the job.

It is often repeated that when preparing data for analysis, you could make a seemingly endless number of improvements when extracting, transforming, and loading (ETL) your data. It was recommended to me by ML instructors like Alice Zhao to avoid this trap, especially in exploratory ML research. As the idiom goes, *do not let perfect get in the way of good enough.*

*Alice Zhao GitHub profile*
https://github.com/adashofdata

When approaching machine learning in libraries, many potential projects may allow for a machine learning approach. One particular challenge that is ripe for libraries and machine learning is topic modeling. There is not room to compare and contrast various topic modeling methodologies; instead, I will focus on one machine learning methodology. According to David Blei, Andrew Ng, Michael Jordan, and John Lafferty, their latent Dirichlet allocation (LDA) model improves other machine learning topic models by allowing for a specific document to be classified not just into one topic but into many, often overlapping topics.[8] This LDA model "treats each document as a mixture of topics, and each topic as a mixture of words."[9] As a result, an LDA approach to topic modeling in libraries should allow machine learning librarians to determine the aboutness of a resource approximating the application of official library subject headings.

## Development Environment

When working with Python, it is always important to consider your development environment needs. Python virtual environments (not to be confused with a full virtualized operating system) have the benefit that they isolate your environment from any Python used by your operating system. Tools like pipenv also provide developers an ability to track the installation of module dependencies, of which there will be many, and batch pip install them at a later date (pip being a Python package manager). Further, tracking any file changes to your list of Python module dependencies file (a Pipfile or requirements.txt) is also recommended using Git or a version control of your choice. At the moment, this project is being synced to a Github repository. For directions on setting up a development environment, a quick Google search could get you started toward that goal. Additionally, there are numerous cloud service providers to pick from, for varying costs. My choice was a Digital Ocean virtual machine (VM) preconfigured for machine learning researchers.

*Pipenv*
https://pypi.org/project/pipenv

*Pipfile*
https://github.com/craigboman/gutenberg/blob/master/Pipfile

*Github repository for Gutenberg project*
http://github.com/craigboman/gutenberg

## Gutenberg, the Gathering

For my purposes, my source data is the English language e-books from Project Gutenberg. Mirrors of Project Gutenberg are available for downloading the entire collection, but to get the most recent e-books, scraping its entire collection, while respecting its robot recommendations, was preferred. Project Gutenberg (PG) requires a five-second delay between each download, and scraper bots are directed to a specific URL. As a result, the entire PG collection was scraped using one wget command, initiated from terminal in my Digital Ocean VM. The entire PG e-book collection took a weekend of continuous scraping. All of these e-books have to be programmatically unzipped. A few Stack Overflow posts later, a terminal command was found that navigates recursively into subdirectories. This was necessary for the bizarre local file directory structure created when scraping. The command unzips the files and moves all files to one parent directory. All of these scripts are available in my processing-pg-texts log.

*Processing-pg-texts log*
https://github.com/craigboman/gutenberg/blob/master/processing-pg-texts-log.txt

## Data Cleanup

Data cleanup is where data scientists arguably spend much of their time. Although machine learning looks glamorous, behind the scenes are seemingly endless extracting, transforming, and loading (ETL) data tasks from other systems into the system that will be doing your machine learning work. As noted, bash commands are not always the fastest, but are well tested and documented. Additionally, it is often trivial to output or pipe the results from one bash command into another command, creating complex and conditional terminal scripting. My data cleanup tasks included removal of non-English e-books and trimming from the end of each e-book the terms of use license. This was no small task, encompassing 85,000 PG e-books. Making backup copies should be mixed between steps as necessary, controlling your file version histories. My sed command makes files changes while simultaneously making a backup.

1. Download (`wget` command)—both files and metadata from PG
2. Unzip (`unzip/while` command)
3. Remove non-English e-books (`grep` and `cat` commands)
4. Trim terms of service from the end (`sed` command)
5. Serializing (`python jsn.py &`)

A full list of data cleanup tasks are in the processing-pg-text logs at the URL in the gray box.

Serializing, for lack of a better word, is the representation or encoding of an entire e-book as a Python object for easier loading and analysis at a later time. This could include, but is not limited to, encoding in JSON or Pickle. Pickle encoding is a little slower than JSON, but it is still a strong serialization option.[10] I chose JSON encoding due to its speed advantage and its being more common among Python users than Pickle encoding.

There are numerous online tutorials that describe how to go about serializing Python objects in JSON. Part of my Python scripts are inspired by Jason Brownlee's machine learning mastery tutorials, of which there are many.[11] This serialization also included some text normalization features, removing arguably irrelevant text for my ML needs (capitalization, punctuation, etc.). The Python script used for object serialization is available at my Git repository. Calling python jsn.py & in command line results in a useful serialization loop pointed at the directory containing the cleaned collection of English language PG e-books. Other researchers can customize jsn.py, altering the loop() directory path for your local needs. There are better ways to do this, but this works.

*Python script for object serialization*
https://github.com/craigboman/gutenberg/blob/master/python/jsn.py

## Tokenization

Tokenizing is the process of reducing your full-text e-book collection down to the least common denominators. This could be a reduction down to sentences, phrases, words, or characters and may also include removing stopwords or using unique numbers as proxies for words. Instead of processing e-books as text, it is often more efficient to feed e-book data into an ML algorithm as if it were numbers representing distinct words in a book, emphasis on *distinct*. Similarly, when a machine learning algorithm analyzes a picture, instead of actually analyzing the seemingly meaningless order of colors in a photo, a visual machine learning algorithm is analyzing a photo based on a two-dimensional graph of numbers representing a photo. This is essentially what we are doing with an e-book. Instead of analyzing the seemingly meaningless order of words, we reduce the words down to an ordered series of numbers that reflects that e-book. Then a machine learning algorithm can more easily analyze a string of numbers representing an e-book and pull out metadata or structure about said e-book. In this analogy, the differences between a photo of a cat and a separate photo of a dog represent unique two dimensional arrays of numbers; whereas the possible differences between Plato's *Republic* and Homer's *Iliad,* after both are reduced to tokens or word vectors, may contain similar and predictable differences in the two-dimensional arrays representing a cat versus the numerical arrays representing a dog.

## Finally . . . Machine Learning

Now that we have reduced our entire full-text PG e-books collection to JSON-encoded files, we can begin the application of LDA to model e-book topics. In this case the gears making LDA work include Keras, TensorFlow, and NLTK (natural language tool kit). This has been a process of surmountable failures—both of cleaning and preparing my data. Most ML tutorials online leave something to be desired or in the least they did not include all the required Python module dependencies. ML researchers could do more to fully document their processes and workflows. This would be improved by the use of Jupyter Notebooks, but those do not work well for the production workflows I have in mind, where Python scripts are closely integrated into an ILS through webhooks and API calls.

After trying dozens of general ML tutorials, I began by preparing my data in a particular format, in this case basic word-level Python objects encoded into JSON files discussed above. Only recently, stumbling into LDA tutorials, have I determined that all of my data preparation for other ML tutorials was meaningful but functionally useless. I now need to loop back into the data preparation task and reformat all of my data. Until I can reformat all of my data into panda data frames, I will not be able to fully explore the use of LDA at this moment.

## Recommendations for Future Research

Despite my data preparation failures toward machine learning, I remain enthusiastic. Although it was not possible in this first attempt to approximate the official Library of Congress subject headings, this is certainly a goal for future research. LDA continues to be an alluring tool librarians may use to improve our library metadata for resources to which we have full-text access. LDA-enhanced subject headings will not solve bias in libraries, but it will allow catalogers to provide greater services to their patrons.

I would also like to explore the potential use of cloud services, which provide greater support for machine learning researchers. In many ML tutorials, pandas appears to be another strong tool worth exploring. This would also improve serialization as panda dataframes rather than basic word-level JSON encoding. In addition, spaCy has some promising ML tools.

*pandas*
https://pandas.pydata.org

*spaCy*
https://spacy.io

Considering the increasing use of machine learning, bias needs to be a greater part of this discussion. As described in *The Decolonized Librarian,* "Algorithms Don't Think about Race. So Tech Giants Need To."[12] While we as a library industry have a bias in the library subject headings discussion, we should continue to explore the use of automating our generation of library subject headings to improve our library services.

## Notes

1. Cornell University Research Data Management Service Group, "Metadata and Describing Data," accessed October 10, 2018, https://data.research.cornell.edu/content/writing-metadata.
2. Roy Tennant, "MARC Must Die," Digital Libraries, LJ Infotech, *Library Journal* 127, no. 17 (October 15, 2002): 26–27.
3. Jasmine Aguilera, "Another Word for 'Illegal Alien' at the Library of Congress: Contentious," *New York Times*, July 22, 2016, https://www.nytimes.com/2016/07/23/us/another-word-for-illegal-alien-at-the-library-of-congress-contentious.html; Derek Hawkins, "The Long Struggle over What to Call 'Undocumented Immigrants' or, as Trump Said in His Order, 'Illegal Aliens,'" *Washington Post*, February 9, 2017, https://www.washingtonpost.com/news/morning-mix/wp/2017/02/09/when-trump-says-illegals-immigrant-advocates-recoil-he-would-have-been-all-right-in-1970/?noredirect=on&utm_term=.f080a2218603.
4. Melissa A. Adler, "The ALA Task Force on Gay Liberation: Effecting Change in Naming and Classification of GLBTQ Subjects," *Advances in Classification Research Online* 23, no. 1 (2013): https://doi.org/10.7152/acro.v23i1.14226.
5. Thomas G. Padilla, "Collections as Data: Implications for Enclosure," *College and Research Libraries News* 79, no. 6 (June 2018): 296–300, https://crln.acrl.org/index.php/crlnews/article/view/17003/18740; Chris Bourg, "What Happens to Libraries and Librarians When Machines Can Read All the Books?" *Feral Librarian* (blog), March 16, 2017, https://chrisbourg.wordpress.com/2017/03/16/what-happens-to-libraries-and-librarians-when-machines-can-read-all-the-books; Safiya Umoja Noble, Algorithms of Oppression: How Search Engines Reinforce Racism (New York: New York University Press, 2018).
6. Bourg, "What Happens to Libraries?"
7. Rong Ge, "Lecture 1: Machine Learning Basics" (slide presentation, COMPSCI 590.7—Algorithmic Aspects of Machine Learning, Duke University Department of Computer Science, Fall 2015), https://www2.cs.duke.edu/courses/fall15/compsci590.7/lecture1.pdf.
8. David M. Blei, Andrew Y. Ng, Michael I. Jordan, and John Lafferty, "Latent Dirichlet Allocation," *Journal of Machine Learning Research* 3, no. 4/5 (2003): 993–1022.
9. Julia Silge and David Robinson, "Topic Modeling," chapter 6 in *Text Mining with R: A Tidy Approach* (Sebastopol, CA: O'Reilly Media, 2017), https://www.tidytextmining.com.
10. Théo Vanderheyden, "Pickle in Python: Object Serialization," DataCamp, April 5, 2018, https://www.datacamp.com/community/tutorials/pickle-python-tutorial.
11. Jason Brownlee, Machine Learning Mastery website, accessed October 10, 2018, https://machinelearningmastery.com.
12. Michael Dudley, "Algorithms Don't Think about Race. So Tech Giants Need To," *The Decolonized Librarian* (blog), *February* 7, 2017, https://decolonizedlibrarian.wordpress.com/tag/bias.