**Introduction to Machine Learning: From Math to Code memo**

This book has a terrific concept, and I believe that the idea of giving students both the tools to implement machine learning and the mathematical basis to understand it on a deeper level is a great approach. The exercises and coding assignments that I’ve seen demonstrate real thought about what students will need to be able to do in order to show they’ve understood the material, and many of them are the kinds of exercises that will help students to identify their blind spots.

Most of my suggestions are intended to make the text easier to follow and understand for undergraduate readers, especially those who are unfamiliar with linear algebra, probability and statistics, and optimization. The high-level structure of the book is sound, but I believe that there are some organizational and formatting tweaks within chapters that could make the material more accessible. I’ve also made some suggestions about the language and style of the text which could make the book cleaner and easier for undergraduates to read.

**Definitions**

* More terms should be defined. For instance, the term *stochastic* is used many times throughout the text, and understanding what distinguishes a stochastic from a non-stochastic function is vital to understanding many concepts, but the term is never explicitly defined.
* If possible, definitions should include some plain-language explanation in addition to the mathematic definition.

* Some definitions may need to be followed by examples, especially when the concept is abstract or subtle. These don’t necessarily need to be numbered examples set apart from the text, but brief illustrations. For instance, if I were to define independence of events, I might say that two events are independent if knowledge of one event does not give us knowledge of the other, and then explain that the flip of two fair coins are independent, because knowing that the first coin landed on heads does not tell us anything about where the second coin landed.
* Defined terms should be italicized on their first appearance (and only their first appearance).
* You might consider setting apart the definitions of some key terms (for instance, in the chapter on reinforcement learning, *Markov decision process*) in boxes. This would help readers to understand that a term or concept is important, and help them key in on the definition. It would also help to break up the page, which makes reading easier.
* Please see the comments on the pdfs of Chs. 1 and 8 for more specific examples and suggestions relating to all of the points discussed above.

**Organization**

* Chapters should have some form of introduction to explain to readers the broad idea they are about to learn. The introduction should be a paragraph or two long. It should include a definition of the “big concept” of the chapter, a brief description of how the concept relates to the overall aim of the book, and a road map for what the reader will learn in the chapter. This helps to ground readers in what they’re about to learn. You may even consider including a bulleted list of learning objectives following the introductory paragraphs in each chapter. This would help students know what to expect as they navigate the chapter and give them something to refer back to when reviewing the material.
* Fundamental concepts should be introduced before the more advanced concepts that make use of them. For example, before discussing how to solve a system of equations represented by functions of vectors, explain what a function of a vector is. That way, readers will understand the advanced concepts when they encounter them.
* Some of the content in the first three chapters assumes a level of knowledge from the reader that is unlikely in undergraduate students who are unfamiliar with linear algebra, optimization, and/or probability and statistics. Turning the appendix on linear algebra fundamentals into a proper chapter would make the sections on solving systems of equations much easier for an undergraduate reader to grasp.
* Although math is an important part of this book, some of the mathematical sections (especially in the first chapter) are difficult to parse, both because of their length and the lack of plain-language explanations accompanying them. Unless a mathematical proof or derivation is necessary to understanding a concept, it may be better off moved to an appendix. Mathematical derivations that *are* necessary should be thoroughly explained so that readers understand the concepts underlying them.
* Descriptions of algorithms will be clearer to readers if the algorithm is explained once, in an implementable form, and then explanations are given for why the algorithm works the way it does and how it relates to other algorithms. In other words, first describe *how* it works, then describe *why* it works. For instance, the description of the bisection method starts by describing the method when an arbitrary point is chosen between the two endpoints, and then explains that the midpoint should be used for optimal performance. This is difficult to follow because the initial description of the algorithm doesn’t provide any direction for how someone implementing the algorithm should choose the new endpoint. Describing the algorithm using the midpoint first, and then explaining why the midpoint is used, would be easier to follow. The pseudocode provided in later chapters is extremely clear; methods in earlier chapters would also benefit from a step-by-step explanation.

**Language**

* The word “specifically” is overused to introduce definitions; it’s very rarely necessary for comprehension, and often distracts from the meaning.
* Some of the language is more suited to an academic article in which the authors are working through a new theorem than it is to an introductory textbook in which the authors are conveying established information. Phrases like “we note that,” “we conclude that,” and “we make the following comments” are unnecessary, and may be distracting. They also undermine your authorial voice.
* Active language is clearer and more effective than passive language. For instance, “Reinforcement learning can be considered as one of the three basic machine learning paradigms” can be rewritten as: “Reinforcement learning is one of the three basic machine learning paradigms.” This both shortens the sentence and more clearly conveys the underlying idea.
* The phrase “can be considered as” is overused in general. It’s clearer and more concise to say “X is Y” than “X can be considered as Y.”
* Some of the sentences are a bit too long to be easily comprehended on a first read-through. A sentence that has more than three clauses may be better separated into several shorter sentences.
* “As discussed below” and similar phrases are overused. If a concept will be discussed later in the same chapter, and especially in the same section, it can usually be assumed that readers will continue to that section naturally. If it’s necessary to point out that a concept will be discussed in more detail at some later point, include the specific section, so that readers can seek out the information.

**Headings**

* Capitalization of headings is inconsistent. Some section headings (such as 1.1, “Linear Equation Systems”) are entirely capitalized; some (such as 1.3, “Fixed point iteration”) capitalize only the first word; some (such as 1.2, “The Bisection and Secant methods) capitalize some words but not others. Either capitalizing the first word or capitalizing all words is fine, but it should be consistent throughout the book.
* Ch. 1 only numbers to the level of subheading, but Ch. 8 occasionally numbers to the level of sub-subheading (i.e. 8.4.1). If there is content-driven reason for this (i.e. you have decided that algorithms need their own numbered sub-subheaders but other types of content, such as descriptions of methods such as The Secant Method in Ch. 1, which is simply bolded without a number, do not), then this is okay. But you should make sure that you are using consistent rules throughout the whole book to determine how a section is labeled and/or numbered. The headings and numberings need to follow a consistent logic in order to allow students to follow the map of the chapter.

**Figures**

* Figures and tables should be accompanied by titles and explanatory captions. Captions shouldn’t just describe the figure, but should also contextualize it for the student and highlight specific features students should pay special attention to.
* When figures are referenced in the text, they should be referred to by their specific figure number. Saying “the figure below” may end up being confusing if the text is formatted such that the figure ends up on the next page, or such that there are multiple figures below.

**Formatting**

* Examples should be set apart from the rest of the text by whitespace. This both breaks up the page, which makes reading easier, and makes it easier for the reader to know when an example has ended. (As it currently stands, it’s often very difficult to determine this.) While we are mostly responsible for the formatting and text design and don’t want you to concern yourself with this too much, if there are ways to set the equations apart more clearly in your LaTeK setup, we would encourage you to do this (for reviewing purposes).
* Subheadings below the section level (e.g., “The bisection search” and “The secant search” in Section 1.2) should be separated from the main text with line breaks. This eases readability and makes it clear that the text is moving on to a new subject.
* The pseudocode sections are extremely useful and clear. But they would be easier to find and separate from the rest of the text if they were placed in a box or otherwise set apart.