

MACHINE LEARNING WITH RUST

*A practical attempt to explore Rust and
its libraries across popular machine
learning techniques*

Keiko Nakamura



Copyright © 2024 by GitforGits

All rights reserved. This book is protected under copyright laws and no part of it may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without the prior written permission of the publisher. Any unauthorized reproduction, distribution, or transmission of this work may result in civil and criminal penalties and will be dealt with in the respective jurisdiction at anywhere in India, in accordance with the applicable copyright laws.

Published by: GitforGits

Publisher: Sonal Dhandre

www.gitforgits.com

support@gitforgits.com

Printed in India

First Printing: January 2024

ISBN: 9788119177936

Cover Design by: Kitten Publishing

For permission to use material from this book, please contact GitforGits at support@gitforgits.com.

Prologue

Prior to starting this writing expedition, I had a clear but challenging objective: to discover how the language Rust, which is known for being fast and safe, can work with the constantly changing field of machine learning. This book was written out of a mix of interest and a strong desire to go where no one has gone before, mapping out the places where Rust and machine learning meet. It's more of an exploration and story of finding than a claim to mastery. As the author, I'm not inviting you on this journey as a perfect teacher, but as a fellow learner eager to find out what can be done by combining the strength of Rust with the flexibility of machine learning methods.

It was important to make it clear from the start that this book is not meant to be the only way to learn machine learning with Rust. It's more of a starting point or introduction for people who are interested in what Rust can do for machine learning projects. Sharing knowledge and trying new things is what these pages are all about. The goal is to get you interested and give you a base from which to build.

We start with the basics and show you how Rust works so that even people who are new to the language can see what it can do. The steps in the book are similar to how I learned Rust: from getting to know its syntax and features to looking into how its ideas can be used to make machine learning methods work better, they are similar.

The exploring spirit of this book stays true as we learn about different machine learning techniques, from basic algorithms to more complex neural networks. The datasets of COVID, CIFAR-10 and practical examples are meant to show the way by showing how Rust's features, such as its speed, memory safety, and concurrency, can be used to deal with the problems that come up in machine learning projects.

The main idea of this book is to ask questions, learn, and discover what hasn't been explored before. Developers and experts alike think that combining Rust and machine learning can lead to new discoveries. Although there are technical details, the story does not present them as a declaration of knowledge. Instead, it does so as a conversation, encouraging you to try new things, dig deeper, and help us learn more.

Finally, I want to recommend this book to anyone who is interested in what can happen when Rust and machine learning come together. It opens the door to new ideas by showing how the rules of a systems programming language can be combined with machine learning to encourage new ideas. As we turn the pages together, let's start this journey with an open mind and a desire to learn, ready to discover the huge possibilities that lie ahead. At any point, you encounter correction, please do not forget to highlight them up to us at support@gitforgits.com

Content

<i>Preface</i>	<i>xii</i>
<i>GitforGits</i>	<i>xiv</i>
<i>Acknowledgement.....</i>	<i>xv</i>
Chapter 1: Rust Basics for Machine Learning	1
<i>Machine Learning Landscape.....</i>	<i>2</i>
Structure and Components of Machine Learning	2
Types of Machine Learning.....	2
Why Rust for ML?.....	2
Prospects of Machine Learning.....	3
<i>Basics of Rust Programming</i>	<i>3</i>
Memory Safety and Ownership	3
Concurrency	3
Error Handling	4
Comprehensive Ecosystem and Package Management	4
Interoperability with Different Languages.....	5
<i>Cargo and Popular Rust Packages.....</i>	<i>5</i>
Cargo Overview.....	5
Managing Dependencies	5
Cargo's Extensibility	6
ndarray and ndarray-linalg	6
Linfa	6
tch-rs and tract.....	6
<i>Setting up Rust Environment.....</i>	<i>7</i>
Installing Rust and Cargo.....	7
Selecting IDE or Text Editor	7
Initializing New Project.....	8
Installing Essential Packages	8
Testing Environment with Simple Program	8
<i>Structure Machine Learning Projects</i>	<i>8</i>
Root Directory and Workspace Configuration.....	9
Source Code Organization	9
Data Folder for Datasets and Resources	9
Model Artifacts and Evaluation Metrics.....	9
Documentation and Notebooks	10
<i>My First Rust Program.....</i>	<i>10</i>
Setting up Project Directory	10

Creating Main Rust File	11
Importing Library and Initializing Arrays.....	11
Performing Array Operations	11
Running the Program	12
Organizing Dependencies in Cargo.toml.....	12
Creating Feature Flags	13
Utilizing Dev and Build Dependencies.....	13
Versioning in Cargo	14
Rust IDEs.....	14
Why IntelliJ?	14
Downloading and Installing IntelliJ IDEA.....	15
Installing Rust Plugin	15
Configuring IDE.....	15
Creating First Rust Project in IntelliJ.....	15
Debugging and Testing in IntelliJ	15
Preparing ML Environment for Release	16
Optimizing Code for Release	16
Managing Dependencies and Updating Crates.....	16
Setting Version Number and Metadata	16
Creating Release Profile	17
Packaging and Distributing the Project.....	17
Summary.....	18
Chapter 2: Data Wrangling with Rust.....	19
Introduction.....	20
Read CSV Files	22
Adding CSV Crate and File I/O Libraries	22
Reading CSV File	22
Dealing with Headers and Specific Fields	23
Data Validation	24
Storing Records in a Data Structure	24
Data Cleaning with Polars	25
Installing Polars and Importing Data	25
Handling Missing Values.....	26
Removing Outliers	26
Normalizing and Scaling Features	27
Data Transformation	27
Feature Engineering	27
Creating Polynomial Features	28
Creating Interaction Terms	28
Categorical Variable Encoding	28
Binning and Discretization	29
Normalization and Standardization.....	29

<i>Plotters and Data Visualization.....</i>	<i>30</i>
Install Plotters	31
Basic Plot Creation.....	31
Populating Data Points	31
Advanced Customization	32
Plotting Multiple Series	33
<i>Data Export: JSON, Parquet, Feather and Binary</i>	<i>34</i>
JSON (JavaScript Object Notation)	34
Parquet	34
Feather	34
Native Binary	35
<i>Data Serialization and Deserialization.....</i>	<i>35</i>
Understanding Serialization	35
Using Serde for Serialization.....	36
Deserialization in Rust.....	37
Custom Serialization and Deserialization	37
<i>Summary</i>	<i>37</i>
Chapter 3: Linear Regression by Example	39
<i>Introduction to Linear Regression.....</i>	<i>40</i>
Overview	40
<i>Linear Regression: Problem Statement.....</i>	<i>41</i>
Definition	41
Variable Selection and Feature Engineering.....	41
Data Splitting.....	42
Feature Scaling.....	42
Target Variable.....	42
<i>Popular Linear Regression Strategies</i>	<i>43</i>
Exploratory Data Analysis (EDA).....	43
Hypothesis Testing.....	43
Model Selection	43
Feature Interaction Terms	44
Model Training and Validation	44
Iterative Refinement.....	44
Interpreting Results and Making Decisions	44
<i>Implement Linear Regression Model</i>	<i>44</i>
Model Training	45
Interpreting Coefficients.....	45
Model Prediction and Validation.....	45
Iterative Refinement.....	46
<i>Other Linear Regression Libraries</i>	<i>46</i>
Rustlearn.....	46
Smartcore	47

Alumina	47
Evaluate Linear Regression Performance	48
Root Mean Square Error (RMSE)	48
Coefficient of Determination (R-squared)	49
Mean Absolute Error (MAE)	49
Residual Plots	50
Save and Reload Model	50
Serialization using Serde	50
Writing and Reading Files	52
Deserialization and Model Loading	52
Linear Regression Model Deployment	53
Packaging the Model	53
Creating a Web Service	53
Deploying to Server	54
Monitoring and Logging	54
Versioning and Rollback	54
Summary	55
Chapter 4: Logistic Regression for Classification	56
Overview	57
Introduction to Logistic Regression	57
Implement Logistic Regression	59
Problem Statement and Data Splitting	59
Model Training	59
Making Predictions	60
Evaluation Metrics	60
Hyperparameter Tuning	60
Performance Metrics Deep Dive	60
Understanding Accuracy	61
Precision and Recall	61
F1 Score and Confusion Matrix	61
ROC Curve and AUC-ROC	61
Comparing Metrics	62
Model Hyperparameter Tuning Process	62
Grid Search Methodology	62
Random Search Methodology	62
Cross-Validation in Tuning	62
Fine-Tuning and Iterative Process	63
Final Model Training and Verification	63
Sample Program: Hyperparameter Tuning	63
Define Hyperparameter Space and Cross-Validation	63
Grid Search Loop	63

Train and Evaluate the Model	64
<i>Serialize and Deserialize the Model.....</i>	65
<i>Summary.....</i>	66
Chapter 5: Decision Trees in Action	68
<i>Introduction to Decision Trees.....</i>	69
Interpretable and Transparent Model.....	69
Feature Importance and Selection	69
Handling Categorical and Numerical Data.....	70
Ensemble Methods and Random Forests.....	70
<i>Online Purchase Retail Dataset.....</i>	70
<i>Data Processing for Decision Trees.....</i>	72
Reading the Excel File and Initial Exploration	72
Handling Missing Values.....	72
Encoding Categorical Variables	73
Data Splitting.....	73
Data Normalization	73
<i>Performing Feature Extraction.....</i>	73
Creating Customer Lifetime Value.....	73
Temporal Feature Engineering.....	74
Average Purchase Value.....	74
Seasonality Features.....	75
Product Categories	75
Geographical Features	76
<i>Initialize and Train Decision Trees.....</i>	76
Separate Features and Labels.....	76
Initialize Decision Tree Model.....	77
Training the Model	77
Model Evaluation	78
Making Predictions.....	79
Batch Predictions.....	80
Model Interpretation.....	80
<i>Summary.....</i>	81
Chapter 6: Mastering Random Forests.....	82
<i>Introduction to Random Forests.....</i>	83
<i>Random Forest Classifier Architecture.....</i>	84
<i>Building Random Forest Model.....</i>	86
Up and Running with RustLearn	86
Defining Random Forest Model	86
Training and Predict Model.....	86

RustLearn's Model Evaluation.....	87
Comparison with Decision Tree Model.....	87
Creating Respective Models	87
Evaluating Models.....	88
Model Interpretability.....	88
Computational Cost.....	89
Overfitting	89
Feature Importance	89
Tuning Random Forests Hyperparameters.....	90
Manual Grid Search	90
Randomized Search.....	91
Cross-Validation	92
Errors and Troubleshooting.....	92
#1 Overfitting.....	92
Solution	92
#2 Memory Exhaustion	92
Solution	93
#3 Imbalanced Data	93
Solution	93
#4 High Dimensionality.....	93
Solution	93
#5 Parallelization Errors	93
Solution	93
#6 Numerical Instabilities.....	93
Solution	93
#7 Non-Numerical Data	94
Solution	94
#8 Incomplete or Missing Data	94
Solution	94
#9 Incorrect Evaluation Metrics.....	94
Solution	94
Summary.....	94
Chapter 7: Support Vector Machines In Action.....	96
Introduction to Support Vector Machines.....	97
Types of Support Vector Machines.....	98
Linear SVM	98
Architecture of Linear SVM.....	98
Kernel SVM.....	99
Architecture of Kernel SVM	99
Differences and Commonality	99
SVM Modeling: Data Load, Preprocess and Model Build	99
Coronavirus Tweets Dataset	99
Loading Data with CSV Crate	100

Using Rust's String Methods	101
Using Bag-of-Words or TF-IDF.....	101
SVM Model Building.....	101
<i>Perform Training and Predictions.....</i>	102
Splitting Data into Training and Test Sets	102
Training the SVM Model.....	102
Making Predictions.....	103
Assessing Model Accuracy	103
<i>Predictions Evaluation and Hyperparameter Tuning.....</i>	103
Calculating Accuracy.....	104
Calculating Precision and Recall	104
Grid Search in Rust.....	105
<i>Dataset and Model Update</i>	106
Incremental Learning.....	106
Data Retention Policies	107
Versioning Models	107
Automating Updates.....	107
<i>Model Rollbacks</i>	108
Model Versioning System	108
Metadata Storage	108
Version Catalog	109
Rollback and Testing	109
<i>Summary</i>	110
Chapter 8: Simplifying Naive Bayes and k-NN	111
<i>Naive Bayes and k-NN Overview</i>	112
<i>Naive Bayes Classifier Architecture.....</i>	112
<i>Building Naive Bayes Model.....</i>	113
<i>Building k-NN Model.....</i>	115
<i>Naive Bayes and k-NN Model Comparison.....</i>	116
<i>Errors and Troubleshooting</i>	117
#1 Model not Converging	117
<i>Solution</i>	117
#2 File not Found during Serialization/Deserialization.....	117
<i>Solution</i>	118
#3 Memory Overflow	118
<i>Solution</i>	118
#4 Poor Generalization	118
<i>Solution</i>	118
#5 Inconsistent Data Types.....	119
<i>Solution</i>	119
#6 Data Imbalance for Classification.....	119

<i>Solution</i>	119
#7 Feature Scaling	119
<i>Solution</i>	119
#8 Concurrency Issues	120
<i>Solution</i>	120
#9 Incorrect Label Encoding.....	120
<i>Solution</i>	120
Summary	121
Chapter 9: Crafting Neural Networks with Rust	122
Introduction to Neural Networks	123
Genesis of Neural Networks	123
Breakthroughs and Modern Applications	123
Neural Networks in Today's World	124
Components of Neural Networks.....	125
<i>Neurons</i>	125
<i>Layers</i>	126
<i>Weight and Biases</i>	126
<i>Activation Functions</i>	126
<i>Loss Function</i>	126
<i>Optimizer</i>	126
<i>Backpropagation</i>	126
NeuroFlow for Neural Networks.....	127
Install NeuroFlow	127
Adding NeuroFlow to Rust Project.....	127
Creating Single-Layer Neural Network	127
PyTorch for Neural Networks	130
Core Features of PyTorch	130
<i>Dynamic computational graph</i>	130
<i>Simplicity and Flexibility</i>	130
<i>Strong GPU Acceleration</i>	130
<i>Extensive Libraries</i>	130
<i>TorchScript for Easy Deployment</i>	131
Creating a Neural Network in PyTorch	131
Training a Model in PyTorch.....	131
Using PyTorch in Rust.....	132
<i>Exporting PyTorch Model with TorchScript</i>	132
<i>Using Model with tch-rs</i>	132
Setup LibTorch	133
Download LibTorch.....	134
Extract LibTorch Archive.....	134
Set Environment Variables	134
Install tch-rs Crate	134
Verify Installation	135
Building Multi-Layer Neural Networks or MLPs	135

Understanding Multi-Layer Neural Networks	136
Using PyTorch to build Multi-Layer Neural Network.....	136
Training a Multi-Layer Neural Network	138
<i>Using ndarray for MLP.....</i>	139
Up and Running with ndarray	139
Defining MLP Structure.....	139
Forward Propagation	141
<i>Convolutional Neural Networks (CNNs)</i>	142
Understanding Convolutions	142
Differentiation from Traditional Neural Networks	143
Convolutional Neural Network Architecture.....	143
<i>Convolutional Layers.....</i>	<i>143</i>
<i>Activation Layers.....</i>	<i>143</i>
<i>Pooling Layers.....</i>	<i>144</i>
<i>Fully Connected Layers.....</i>	<i>144</i>
<i>Dropout Layers</i>	<i>144</i>
<i>Normalization Layers.....</i>	<i>144</i>
Application in Real World	144
<i>Building CNN using PyTorch</i>	145
Define CNN Architecture	145
Initialize Model and Specify Loss Function/Optimizer	147
Train the Model.....	147
<i>Summary.....</i>	148
<i>Epilogue.....</i>	150

Preface

In this stimulating book, you'll learn how to use the Rust programming language in conjunction with machine learning. It's not a full guide to learning machine learning with Rust. Instead, it's more of a journey that shows you what's possible when you use Rust to solve machine learning problems. Some people like Rust because it is quick and safe. This book shows how those qualities can help machine learning a lot.

To begin, we will show you what Rust is and how it works. This is so that everyone, even those who are new to Rust, can follow along. Then, we look at some basic machine learning concepts, such as linear and logistic regression, and show how to use Rust's tools and libraries to make these ideas work.

You will learn more complex techniques like decision trees, support vector machines, and how to work with data as we go along. It goes all the way up to neural networks and image recognition, and we show you how to use Rust for these types of tasks step by step. We use real-world examples, such as COVID data and the CIFAR-10 image set, to show how Rust works with issues that come up in the real world.

This book is all about discovery and experimentation. To see what you can do with them, we use various Rust tools for machine learning. It's a fun way to see how Rust can be used in machine learning, and it will make you want to try new things and learn more on your own. This is only the beginning; there is so much more to uncover as you continue to explore machine learning with Rust.

In this book you will learn how to:

- Exploit Rust's efficiency and safety to construct fast machine learning models.
- Use Rust's ndarray crate for numerical computations to manipulate complex machine learning data.
- Find out how Rust's extensible machine learning framework, linfa, works across algorithms to comprehend the ecosystem.
- Use Rust's precision and speed to construct linear and logistic regression.
- See how Rust crates simplify decision trees and random forests for prediction and categorization.
- Learn to implement and optimize probabilistic classifiers, SVMs and closest neighbor methods in Rust.
- Use Rust's computing power to study neural networks and CNNs for picture recognition and processing.

- Apply learnt strategies to COVID and CIFAR-10 datasets to address realistic problems and obtain insights.

GitforGits

Prerequisites

This book is for aspiring Rust programmers, newbies to machine learning, and to machine learning professionals who want to witness the possibilities of Rust in machine learning. The book just expects you to be eager to explore Rust in machine learning and no other thing is desired to pull this book.

Codes Usage

Are you in need of some helpful code examples to assist you in your programming and documentation? Look no further! Our book offers a wealth of supplemental material, including code examples and exercises.

Not only is this book here to aid you in getting your job done, but you have our permission to use the example code in your programs and documentation. However, please note that if you are reproducing a significant portion of the code, we do require you to contact us for permission.

But don't worry, using several chunks of code from this book in your program or answering a question by citing our book and quoting example code does not require permission. But if you do choose to give credit, an attribution typically includes the title, author, publisher, and ISBN. For example, "Machine Learning with Rust by Keiko Nakamura".

If you are unsure whether your intended use of the code examples falls under fair use or the permissions outlined above, please do not hesitate to reach out to us at support@gitforgits.com.

We are happy to assist and clarify any concerns.

Acknowledgement

I owe a tremendous debt of gratitude to GitforGits, for their unflagging enthusiasm and wise counsel throughout the entire process of writing this book. Their knowledge and careful editing helped make sure the piece was useful for people of all reading levels and comprehension skills. In addition, I'd like to thank everyone involved in the publishing process for their efforts in making this book a reality. Their efforts, from copyediting to advertising, made the project what it is today.

Finally, I'd like to express my gratitude to everyone who has shown me unconditional love and encouragement throughout my life. Their support was crucial to the completion of this book. I appreciate your help with this endeavour and your continued interest in my career.

CHAPTER 1: RUST BASICS FOR MACHINE LEARNING