

Finding Ghosts in Your Data

**Anomaly Detection Techniques
with Examples in Python**

Kevin Feasel

Apress®

Finding Ghosts in Your Data: Anomaly Detection Techniques with Examples in Python

Kevin Feasel
DURHAM, NC, USA

ISBN-13 (pbk): 978-1-4842-8869-6
<https://doi.org/10.1007/978-1-4842-8870-2>

ISBN-13 (electronic): 978-1-4842-8870-2

Copyright © 2022 by Kevin Feasel

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Jonathan Gennick
Development Editor: Laura Berendson
Coordinating Editor: Jill Balzano

Cover photo by Pawel Czerwinski on Unsplash

Distributed to the book trade worldwide by Springer Science+Business Media LLC, 1 New York Plaza, Suite 4600, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

To Mom and Dad, who know a thing or four about anomalies.

Table of Contents

About the Authorxv

About the Technical Reviewerxvii

Introductionxix

Part I: What Is an Anomaly? 1

Chapter 1: The Importance of Anomalies and Anomaly Detection 3

 Defining Anomalies 3

 Outlier 3

 Noise vs. Anomalies 4

 Diagnosing an Example 5

 What If We’re Wrong? 7

 Anomalies in the Wild 8

 Finance 8

 Medicine 11

 Sports Analytics 11

 Web Analytics 14

 And Many More 15

 Classes of Anomaly Detection 16

 Statistical Anomaly Detection 16

 Clustering Anomaly Detection 16

 Model-Based Anomaly Detection 17

 Building an Anomaly Detector 18

 Key Goals 18

 How Do Humans Handle Anomalies? 19

 Known Unknowns 21

Conclusion 22

TABLE OF CONTENTS

Chapter 2: Humans Are Pattern Matchers 23

 A Primer on the Gestalt School 23

 Key Findings of the Gestalt School 24

 Emergence 24

 Reification 25

 Invariance 26

 Multistability 27

 Principles Implied in the Key Findings 28

 Meaningfulness 28

 Conciseness 29

 Closure 30

 Similarity 31

 Good Continuation 32

 Figure and Ground 34

 Proximity 35

 Connectedness 35

 Common Region 35

 Symmetry 36

 Common Fate 37

 Synchrony 38

 Helping People Find Anomalies 39

 Use Color As a Signal 39

 Limit Nonmeaningful Information 40

 Enable “Connecting the Dots” 40

 Conclusion 41

Chapter 3: Formalizing Anomaly Detection 43

 The Importance of Formalization 43

 “I’ll Know It When I See It” Isn’t Enough 43

 Human Fallibility 44

 Marginal Outliers 44

 The Limits of Visualization 45

The First Formal Tool: Univariate Analysis.....	46
Distributions and Histograms	46
The Normal Distribution.....	49
Mean, Variance, and Standard Deviation	51
Additional Distributions	54
Robustness and the Mean	58
The Susceptibility of Outliers.....	58
The Median and “Robust” Statistics.....	58
Beyond the Median: Calculating Percentiles	59
Control Charts	61
Conclusion	62
Part II: Building an Anomaly Detector	63
Chapter 4: Laying Out the Framework.....	65
Tools of the Trade.....	65
Choosing a Programming Language	65
Making Plumbing Choices	66
Reducing Architectural Variables.....	68
Developing an Initial Framework	69
Battlespace Preparation	69
Framing the API	70
Input and Output Signatures	72
Defining a Common Signature.....	73
Defining an Outlier	74
Sensitivity and Fraction of Anomalies	74
Single Solution	75
Combined Arms	75
Framing the Solution.....	76
Containerizing the Solution	79
Conclusion	80

Chapter 5: Building a Test Suite 81

Tools of the Trade 81

 Unit Test Library 82

 Integration Testing 82

Writing Testable Code 83

 Keep Methods Separated 83

 Emphasize Use Cases..... 84

 Functional or Clean: Your Choice 84

Creating the Initial Tests..... 86

 Unit Tests 86

 Integration Tests 90

Conclusion 94

Chapter 6: Implementing the First Methods..... 95

A Motivating Example 95

Ensembling As a Technique 96

 Sequential Ensembling..... 97

 Independent Ensembling..... 98

 Choosing Between Sequential and Independent Ensembling 99

Implementing the First Checks 99

 Standard Deviations from the Mean..... 100

 Median Absolute Deviations from the Median..... 101

 Distance from the Interquartile Range 102

 Completing the run_tests() Function 103

Building a Scoreboard..... 104

 Weighting Results..... 105

 Determining Outliers 106

Updating Tests..... 109

 Updating Unit Tests..... 109

 Updating Integration Tests..... 114

Conclusion 116

Chapter 7: Extending the Ensemble.....	117
Adding New Tests.....	117
Checking for Normality.....	118
Approaching Normality.....	123
A Framework for New Tests	126
Grubbs' Test for Outliers.....	128
Generalized ESD Test for Outliers.....	129
Dixon's Q Test.....	131
Calling the Tests.....	133
Updating Tests.....	135
Updating Unit Tests.....	135
Updating Integration Tests.....	140
Multi-peaked Data	141
A Hidden Assumption	141
The Solution: A Sneak Peek.....	143
Conclusion	144
Chapter 8: Visualize the Results.....	145
Building a Plan	145
What Do We Want to Show?	145
How Do We Want to Show It?	146
Developing a Visualization App	147
Getting Started with Streamlit.....	147
Building the Initial Screen	148
Displaying Results and Details	151
Conclusion	157
Part III: Multivariate Anomaly Detection	159
Chapter 9: Clustering and Anomalies	161
What Is Clustering?	161
Common Cluster Terminology.....	162
K-Means Clustering	163

TABLE OF CONTENTS

K-Nearest Neighbors.....	168
When Clustering Makes Sense	170
Gaussian Mixture Modeling.....	171
Implementing a Univariate Version.....	172
Updating Tests	176
Common Problems with Clusters.....	179
Choosing the Correct Number of Clusters	179
Clustering Is Nondeterministic	180
Alternative Approaches.....	182
Tree-Based Approaches.....	182
The Problem with Trees	183
Conclusion	183
Chapter 10: Connectivity-Based Outlier Factor (COF).....	185
Distance or Density?	185
Local Outlier Factor	187
Connectivity-Based Outlier Factor	189
Introducing Multivariate Support	191
Laying the Groundwork	191
Implementing COF	194
Test and Website Updates.....	197
Unit Test Updates.....	197
Integration Test Updates.....	198
Website Updates.....	198
Conclusion	201
Chapter 11: Local Correlation Integral (LOCI).....	203
Local Correlation Integral.....	203
Discovering the Neighborhood	203
Multi-granularity Deviation Factor (MDEF)	205
Multivariate Algorithm Ensembles	206
Ensemble Types.....	206

COF Combinations	207
Incorporating LOCI	210
Test and Website Updates	213
Unit Test Updates	213
Website Updates	214
Conclusion	215
Chapter 12: Copula-Based Outlier Detection (COPOD)	217
Copula-Based Outlier Detection	217
What's a Copula?	217
Intuition Behind COPOD	218
Implementing COPOD	221
Test and Website Updates	223
Unit Test Updates	223
Integration Test Updates	224
Website Updates	225
Conclusion	228
Part IV: Time Series Anomaly Detection	229
Chapter 13: Time and Anomalies	231
What Is Time Series?	231
Time Series Changes Our Thinking	233
Autocorrelation	233
Smooth Movement	234
The Nature of Change	235
Data Requirements	238
Time Series Modeling	239
(Weighted) Moving Average	239
Exponential Smoothing	239
Autoregressive Models	241
What Constitutes an Outlier?	242
Local Outlier	242

TABLE OF CONTENTS

Behavioral Changes over Time	243
Local Non-outlier in a Global Change	243
Differences from Peer Groups	243
Common Classes of Technique	244
Conclusion	244
Chapter 14: Change Point Detection	247
What Is Change Point Detection?	247
Benefits of Change Point Detection	248
Change Point Detection with ruptures	249
Dynamic Programming	249
PELT	250
Implementing Change Point Detection	250
Test and Website Updates	255
Unit Tests	255
Integration Tests	257
Website Updates	258
Avenues of Further Improvement	260
Conclusion	261
Chapter 15: An Introduction to Multi-series Anomaly Detection	263
What Is Multi-series Time Series?	263
Key Aspects of Multi-series Time Series	264
What Needs to Change?	267
What's the Difference?	267
Leading and Lagging Factors	268
Available Processes	268
Cross-Euclidean Distance	270
Cross-Correlation Coefficient	270
SameTrend (STREND)	271
Common Problems	272
Conclusion	273

Chapter 16: Standard Deviation of Differences (DIFFSTD)	275
What Is DIFFSTD?	275
Calculating DIFFSTD	275
Key Assumptions	276
Writing DIFFSTD	278
Series Processing	278
Segmentation	279
Comparing the Norm	280
Determining Outliers	283
Test and Website Updates	286
Unit Tests	286
Integration Tests	287
Website Updates	289
Conclusion	292
Chapter 17: Symbolic Aggregate Approximation (SAX)	293
What Is SAX?	293
Motifs and Discords	294
Subsequences and Matches	295
Discretizing the Data	296
Implementing SAX	300
Segmentation and Blocking	300
Making SAX Multi-series	303
Scoring Outliers	304
Test and Website Updates	307
Unit and Integration Tests	307
Website Updates	308
Conclusion	308

Part V: Stacking Up to the Competition 311

Chapter 18: Configuring Azure Cognitive Services Anomaly Detector 313

 Gathering Market Intelligence..... 313

 Amazon Web Services: SageMaker 313

 Microsoft Azure: Cognitive Services 314

 Google Cloud: AI Services 315

 Configuring Azure Cognitive Services 316

 Set Up an Account 316

 Using the Demo Application 320

 Conclusion 323

Chapter 19: Performing a Bake-Off 325

 Preparing the Comparison 325

 Supervised vs. Unsupervised Learning 325

 Choosing Datasets 326

 Scoring Results 327

 Performing the Bake-Off 328

 Accessing Cognitive Services via Python 329

 Accessing Our API via Python 331

 Dataset Comparisons 334

 Lessons Learned 336

 Making a Better Anomaly Detector 337

 Increasing Robustness 337

 Extending the Ensembles 338

 Training Parameter Values..... 338

 Conclusion 339

Appendix: Bibliography 341

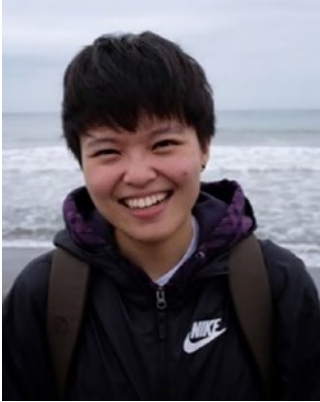
Index..... 345

About the Author



Kevin Feasel is a Microsoft Data Platform MVP and CTO at Faregame Inc., where he specializes in data analytics with T-SQL and R, forcing Spark clusters to do his bidding, fighting with Kafka, and pulling rabbits out of hats on demand. He is the lead contributor to Curated SQL, president of the Triangle Area SQL Server Users Group, and author of *PolyBase Revealed*. A resident of Durham, North Carolina, he can be found cycling the trails along the triangle whenever the weather's nice enough.

About the Technical Reviewer



Yin-Ting (Ting) Chou is currently a Data Engineer/Full-Stack Data Scientist at ChannelAdvisor. She has been a key member on several large-scale data science projects, including demand forecasting, anomaly detection, and social network analysis. Even though she is keen on data analysis, which drove her to obtain her master's degree in statistics from the University of Minnesota, Twin Cities, she also believes that the other key to success in a machine learning project is to have an efficient and effective system to support the whole model productizing process. To create the system, she is currently diving into the fields of MLOps and containers. For more information about her, visit www.yintingchou.com.

Introduction

Welcome to this book on anomaly detection! Over the course of this book, we are going to build an anomaly detection engine in Python. In order to do that, we must first answer the question, “What is an anomaly?” Such a question has a simple answer, but in providing the simple answer, we open the door to more questions, whose answers open yet more doors. This is the joy and curse of the academic world: we can always go a little bit further down the rabbit hole.

Before we start diving into rabbit holes, however, let’s level-set expectations. All of the code in this book will be in Python. This is certainly not the only language you can use for the purpose—my esteemed technical reviewer, another colleague, and I wrote an anomaly detection engine using a combination of C# and R, so nothing requires that we use Python. We do cover language and other design choices in the book, so I’ll spare you the rest here. As far as your comfort level with Python goes, the purpose of this book is not to teach you the language, so I will assume some familiarity with the language. I do, of course, provide context to the code we will write and will spend extra time on concepts that are less intuitive. Furthermore, all of the code we will use in the book is available in an accompanying GitHub repository at <https://github.com/Apress/finding-ghosts-in-your-data>.

My goal in this book is not just to write an anomaly detection engine—it is to straddle the line between the academic and development worlds. There is a rich literature around anomaly detection, but much of the literature is dense and steeped with formal logic. I want to bring you some of the best insights from that academic literature but expose it in a way that makes sense for the large majority of developers. For this reason, each part in the book will have at least one chapter dedicated to theory. In addition, most of the code-writing chapters also start with the theory because it isn’t enough simply to type out a few commands or check a project’s readme for a sample method call; I want to help you understand why something is important, when an approach can work, and when the approach may fail. Furthermore, should you wish to take your own dive into the literature, the bibliography at the end of the book includes a variety of academic resources.

INTRODUCTION

Before I sign off and we jump into the book, I want to give a special thank you to my colleague and technical editor, Ting Chou. I have the utmost respect for Ting's skills, so much so that I tried to get her to coauthor the book with me! She did a lot to keep me on the right path and heavily influenced the final shape of this book, including certain choices of algorithms and parts of the tech stack that we will use. That said, any errors are, of course, mine and mine alone. Unfortunately.

If you have thoughts on the book or on anomaly detection, I'd love to hear from you. The easiest way to reach out is via email: feasel@catallaxyservices.com. In the meantime, I hope you enjoy the book.