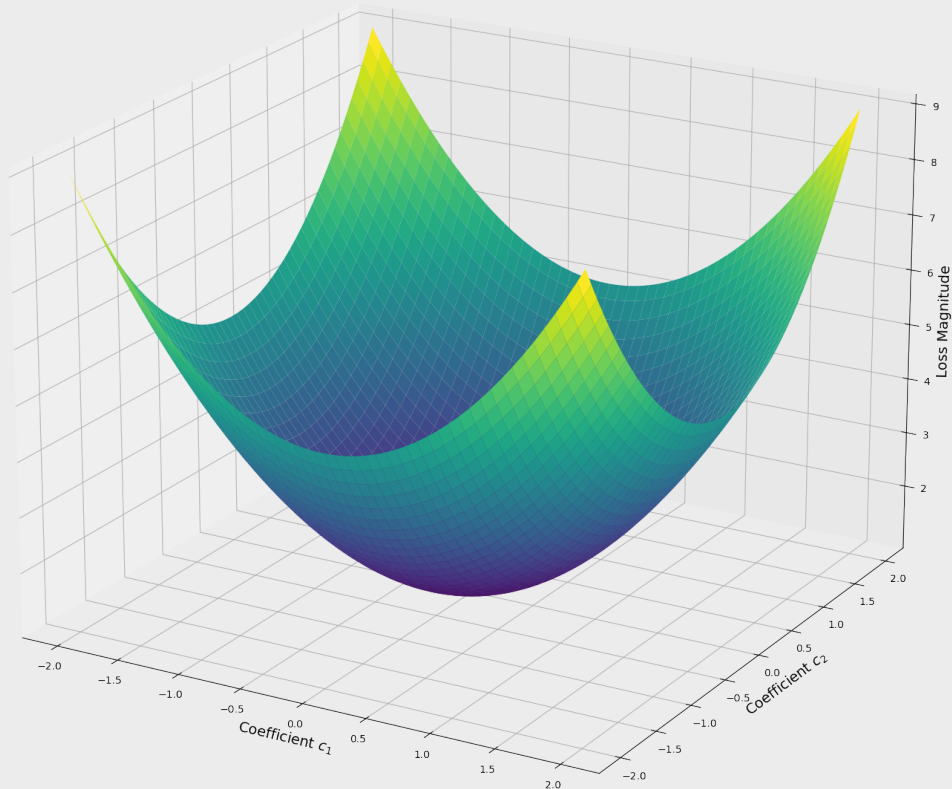


Mastering Data Science Interviews

Explain Any Concept in Data Science, Machine Learning and Artificial Intelligence



William Alston

Copyright

Copyright © 2026 William N. Alston

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the copyright holder, except for brief quotations used in reviews, academic work, or other uses permitted by law.

First published in 2026

coherentnoise publishing
coherentnoise.space

The author has made every effort to ensure that the information in this book is accurate and up to date at the time of publication. However, no responsibility is accepted for any loss, injury, or damage arising from the use of the material contained in this book.

This book is intended for educational purposes only.

Contents

1	The Aim of This Book	1
1.1	Why This Book Exists	1
1.2	The Challenge of Technical Interviews	2
1.3	The Gap Between Coursework and Interviews	2
1.4	Learning Through Interview Questions	3
1.5	What This Book Assumes	3
1.6	The Five Types of Understanding	4
1.7	How Each Question Is Structured	5
1.8	How to Use This Book	7
1.9	Question Difficulty Levels	7
1.10	A Final Note	8
2	How Data Science Interviews Work	9
2.1	Introduction	9
2.2	The Typical Interview Process	9
2.3	Coding and Implementation Interviews	11
2.4	Case Study Interviews	12
2.5	What Interviewers Are Really Testing	12
2.6	An Interview Answer Framework	14
2.7	The Role of Follow-Up Questions	15
2.8	Common Mistakes in Technical Interviews	15
2.9	How This Book Will Help You Prepare	16
2.10	Looking Ahead	16
I	Modelling and Theory	17
3	The Bias–Variance trade-off	19
4	Parametric and Non-Parametric Models	37

5	Linear Regression	54
6	Multicollinearity and Model Stability	71
7	Logistic Regression	86
8	Decision Trees	105
9	Bagging and Random Forest	133
10	Boosting	157
11	Neural Networks	171
II	Optimisation and Training	191
12	Loss Functions	193
13	Gradient Descent	231
14	Momentum and Adaptive Optimisers	252
15	Regularisation	280
16	Training Instabilities	301
17	Weight Initialisation	314
18	Hyperparameters	333
19	Early Stopping	349
III	Evaluation and Metrics	356
20	Train, Validation, and Test Data	358
21	Data Leakage	381
22	Cross Validation	392
23	Classification Metrics	417
24	Confusion Matrices	436
25	ROC Curves	443

26 Precision–Recall Curves	459
27 Regression Metrics	471
28 Model Fitting	484
29 Model Comparison	494
30 Information Criteria	504
31 Statistical Testing	516
32 Calibration	530
33 Log Loss	540
IV Conceptual Foundations	545
34 The Central Limit Theorem	547
35 Independence	553
36 Maximum Likelihood Estimation	558
37 Bayesian Inference	564
38 Probability Foundations	581
39 Feature Engineering	589
40 Feature Scaling	598
41 Curse of Dimensionality	603
42 Principal Component Analysis	611
43 Clustering	624
44 Generative vs Discriminative Models	642
45 Model Interpretability	649
V Applied ML and Systems	664
46 Machine Learning Pipelines	666

47 Real-World Data Challenges	677
48 Missing Data	681
49 Outliers	690
50 Imbalanced Datasets	701
51 Model Deployment and Monitoring	717
52 Production ML Systems	731
53 Online Learning	740
54 Scaling Machine Learning	748
55 Distributed Training	754
56 Experimentation	760
57 Designing ML Systems	775

Preface

This book is intended to help Master's students prepare seriously and confidently for technical interviews in data science, machine learning, and artificial intelligence. It goes beyond short revision-style answers by developing the mathematical ideas, intuitive understanding, and practical interpretation that sit behind common interview topics. The goal is to help readers respond in a way that is not only correct, but also clear, well-structured, and convincing in an interview setting.

My own perspective comes from working across research and teaching for many years. I am an astronomer and a lecturer in data science, and over the past twenty years I have used machine learning in both scientific practice and higher education. Through that experience, I have seen how often students know the vocabulary of machine learning without yet feeling able to explain it fluently or apply it with confidence. This book is my attempt to bridge that gap by bringing together theory, intuition, and implementation in a form designed specifically for technically strong students preparing for demanding interviews.

The Aim of This Book

1.1 Why This Book Exists

Over the past decade, the demand for data scientists, machine learning engineers, and artificial intelligence specialists has grown dramatically. Organisations across technology, finance, health-care, retail, and government increasingly rely on data-driven decision making and predictive modelling. As a result, universities around the world now offer Masters degrees in areas such as:

- Data Science
- Machine Learning
- Artificial Intelligence
- Statistics
- Applied Mathematics
- Computer Science

Students graduating from these programmes typically have strong technical training. They study probability, statistical inference, machine learning algorithms, optimisation, and programming. They complete projects involving real datasets and build predictive models using modern tools such as Python and its scientific libraries.

However, when many graduates begin applying for jobs, they encounter a challenge that their coursework may not have fully prepared them for: the technical interview. Based on decades of experience in development and applications of machine learning techniques to scientific problems and designing several Masters level modules, I have developed this guide in order for newly graduated students to land that dream data scientist role. This guide will also be a valuable resource for any student about to give their Masters coursework viva.

1.2 The Challenge of Technical Interviews

Technical interviews for data science and machine learning roles rarely focus only on whether a candidate has seen a concept before. Instead, interviewers want to determine whether the candidate truly understands the ideas behind the methods they use. Candidates are often asked questions such as:

- What is the bias–variance trade-off?
- Why does logistic regression use the sigmoid function?
- What is the difference between L1 and L2 regularisation?
- How does gradient boosting work?
- What assumptions does linear regression make?
- What does the Central Limit Theorem tell us?

At first glance, these questions appear straightforward. Many students recognise the topics immediately. However, recognising a concept and *explaining it clearly and rigorously* are very different skills.

Many candidates discover during interviews that they can recall definitions but perhaps they struggle to explain the intuition behind an algorithm, the mathematical reasoning that motivates it, the assumptions required for it to work, or the the situations where it may fail. This book was written to help bridge that gap.

Interview Tip

Technical interviews are not only testing whether you know an algorithm. They are testing whether you understand *why it works, when it should be used, and how to explain it clearly.*

Throughout the book, these coloured text boxes will appear to offer additional advice and tips on interviews, mathematical insights and where to go next for a deeper dive on a topic.

1.3 The Gap Between Coursework and Interviews

University and college courses typically focus on teaching the theory and implementation of statistical and machine learning methods. Students learn how to apply algorithms to datasets and evaluate their performance. Once they have gained this understanding, students will typically encounter *real world* data problems in a non-production setting.

This training is essential. However, interviews require an additional skill: the ability to explain technical concepts clearly and logically under pressure. Interviewers often explore a candidate's understanding by asking follow-up questions such as:

- Why does this algorithm work?
- What assumptions does the model make?
- When would you choose this method instead of another?
- How would you diagnose a model that is performing poorly?

These questions reveal whether a candidate truly understands the principles behind machine learning models.

Deep Dive

A strong candidate is not someone who can simply name many algorithms. A strong candidate understands the principles that connect them: optimisation, probability, statistical inference, and linear algebra. Interviews often explore these underlying ideas.

1.4 Learning Through Interview Questions

The central idea of this book is simple:

The best way to prepare for data science interviews is to learn through the questions that interviewers actually ask.

Rather than presenting machine learning purely as an academic subject, this book approaches the material through common interview questions used by companies hiring data scientists and machine learning engineers. Each question becomes an opportunity to explore both the intuition and the mathematics behind the method.

1.5 What This Book Assumes

This book is written primarily for students who have completed, or are currently completing, a Master's degree in data science or a related discipline. Readers should already be familiar with:

- basic probability and statistics
- linear algebra
- Python programming
- fundamental machine learning concepts

The goal of the book is not to introduce these topics from the beginning, but to deepen understanding and prepare students to explain these ideas clearly in technical interviews.

1.6 The Five Types of Understanding

Machine learning interviews often appear to cover a wide range of unrelated topics. In a single interview, a candidate might be asked about linear regression, gradient descent, cross-validation, Bayesian inference, and system design. At first glance these questions may seem disconnected.

In practice, however, most machine learning interview questions fall into a small number of conceptual categories. Interviewers are usually trying to assess whether a candidate understands:

- ✓ how machine learning models represent relationships in data
- ✓ how these models are trained and optimised
- ✓ how their performance is evaluated
- ✓ the statistical ideas that underpin these methods
- ✓ how machine learning systems behave in real-world environments

This book is organised around these five types of knowledge.

Part I: Modelling and Theory introduces the core machine learning models that appear most frequently in interviews. These chapters focus on understanding how different modelling approaches represent patterns in data and how their behaviour relates to concepts such as bias, variance, and model complexity.

Part II: Optimisation and Training explains how machine learning models learn from data. Even a simple model can behave in surprising ways during training, and many interview questions focus on optimisation algorithms, regularisation, and training dynamics.

Part III: Evaluation and Metrics examines how models are assessed and compared. In practical machine learning, it is not enough to train a model; we must also determine whether it generalises well and whether its predictions can be trusted. This section explores evaluation techniques and common pitfalls such as data leakage.

Part IV: Conceptual Foundations develops the statistical ideas that underlie machine learning. Many algorithms can be understood more clearly when viewed through the lens of probability, estimation theory, and statistical reasoning.

Part V: Applied and Systems focuses on the challenges that arise when machine learning is used in real-world systems. Interviewers frequently ask questions about handling imperfect data, deploying models in production, and designing large-scale machine learning systems.

Together, these five parts reflect the different dimensions of knowledge that strong machine learning practitioners are expected to possess. The aim of this structure is not only to prepare

readers for interview questions, but also to help them build a coherent understanding of how machine learning models are developed, evaluated, and used in practice.

1.7 How Each Question Is Structured

Each interview question in this book follows a consistent structure designed to help readers develop both concise explanations and deeper technical understanding.

The Interview Question

Each section begins with a question that commonly appears in technical interviews.

Interview Question

What is the bias–variance trade-off?

Short Interview Answer

In an interview setting, candidates must first give a concise explanation, usually within one or two minutes. Each section therefore begins with a short answer that demonstrates how a strong candidate might respond.

Short Interview Answer

The bias–variance trade-off describes the balance between a model that is too simple to capture the underlying structure of the data and one that is so flexible that it overfits noise in the training data.

Intuition

A clear conceptual explanation helps interviewers see that the candidate understands the idea rather than simply memorising a definition. This often involves some mathematical reasoning.

Mathematical Foundations

Machine learning and data science are built on mathematical ideas from linear algebra, probability theory, statistics and optimisation. Each topic therefore includes a deeper explanation of the mathematical foundations underlying the concept, as can be seen in this example on polynomial regression.

Mathematical Insight

Model complexity is often related to the number of parameters in a model.

Consider a polynomial regression model of degree d :

$$y = \beta_0 + \beta_1x + \beta_2x^2 + \dots + \beta_dx^d.$$

This model contains $d + 1$ parameters.

As d increases, the model becomes capable of representing increasingly complicated functions.

In the limit, a sufficiently high-degree polynomial can interpolate all training points exactly.

However, increasing the number of parameters also increases the variance of the estimator, making the model more sensitive to fluctuations in the training data.

Worked Examples

Concepts become clearer when applied to concrete examples. The book therefore includes examples illustrating how ideas appear in real modelling problems.

Worked Example

Suppose we fit polynomial regression models of degree 1, 3, and 15 to the same dataset. The linear model may miss curvature in the data and have high bias. The degree-15 model may fit almost every fluctuation and have high variance.

Python Implementations

Many interview processes include coding questions or discussions about implementation details. For this reason, full practical Python examples are included throughout the book using libraries such as NumPy, pandas, scikit-learn, PyTorch and statsmodels. These examples are simple in nature, but help understand the topic from a coding perspective. The notebook files can be downloaded directly from github here: <https://github.com/coherentnoise>.

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3
4 X = np.array([[1], [2], [3], [4], [5]])
5 y = np.array([2, 4, 5, 4, 5])
6
7 model = LinearRegression()
8 model.fit(X, y)
9
10 print("Coefficient:", model.coef_)
11 print("Intercept:", model.intercept_)
```

Listing 1.1: Simple Python example

Follow-Up Questions

Interviewers frequently ask additional questions to test deeper understanding. Each section therefore includes examples of typical follow-up questions.

Follow-Up Interview Questions

- How does regularisation affect the bias–variance trade-off?
- Why does more flexible modelling usually increase variance?
- How does cross-validation help identify the right level of complexity?

Common Mistakes

We will highlight any common mistakes in this red textbox that students and interviewees make when trying to answer a particular question.

Common Mistake

A weak answer is to say only that bias is underfitting and variance is overfitting, without explaining why these occur or how the trade-off affects model selection.

1.8 How to Use This Book

Students preparing for interviews may read the chapters sequentially, gradually strengthening their understanding of key ideas in statistics, machine learning, and artificial intelligence.

Alternatively, readers may focus on specific areas relevant to their target roles, such as statistical modelling, machine learning algorithms, deep learning or time series analysis. Because each topic is organised around a specific interview question, the book can also be used as a reference when reviewing individual concepts.

1.9 Question Difficulty Levels

This book is organised around technical interview questions in data science, machine learning, and artificial intelligence. Each question is written at one of three levels:

Question Difficulty Levels

Core

Example: *What is logistic regression?*

This is a core question because it tests a fundamental concept that appears frequently in data science and machine learning interviews. Most candidates are expected to answer it clearly and correctly.

Advanced

Example: *Why does bagging reduce variance?*

This is an advanced question because it requires more than a definition. A strong answer usually involves reasoning about model instability, averaging, and the effect of correlation between estimators.

Deep / Research level

Example: *How does the bias–variance decomposition extend to ensemble models, and why does bagging reduce variance but not bias?*

This is a deep question because it requires the candidate to connect multiple ideas, move beyond standard interview definitions, and reason carefully about the behaviour of ensembles at a more theoretical level.

These examples are not intended to define the difficulty levels rigidly, since the depth expected in an interview will always depend on the role. However, they provide a useful guide to the level of understanding each tag is meant to represent.

A good study strategy is to master the Core questions first, then move on to the Advanced and Deep questions. In a typical interview, these will start by assessing your core understanding of a topic, then ask more advanced questions before asking more open ended questions to assess your deeper understanding of the field.

1.10 A Final Note

Data science sits at the intersection of statistics, mathematics, and computer science. Successful practitioners combine theoretical understanding with practical problem-solving skills. The aim of this book is not simply to help students pass interviews. It is to help them develop the depth of understanding that allows them to **think like professional data scientists**.

By working through the questions in this book, readers will not only improve their interview performance but will also gain a deeper understanding of the mathematical and conceptual foundations of modern data science.

How Data Science Interviews Work

2.1 Introduction

Before preparing for technical interviews in data science or machine learning, it is important to understand how these interviews are typically structured and what employers are actually trying to evaluate. Many students approach interviews as if they were examinations. They attempt to memorise definitions, lists of algorithms, or short explanations. While knowledge is important, this strategy is rarely sufficient.

Technical interviews are designed to evaluate a broader set of abilities. Interviewers want to understand how candidates think, how they reason about problems, and how clearly they can explain technical ideas. In this chapter we will examine:

- ✓ the typical structure of data science interviews
- ✓ the different types of technical questions that may be asked
- ✓ what interviewers are really trying to assess
- ✓ how candidates can prepare effectively

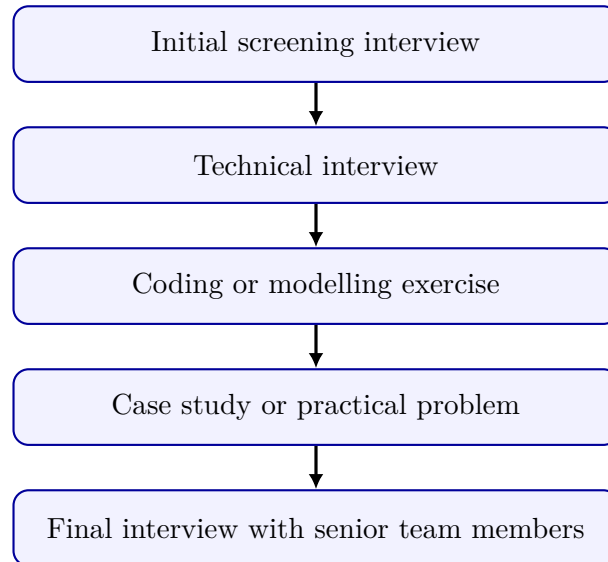
Understanding this process will help you interpret the questions presented throughout this book and see how they relate to real interview situations.

2.2 The Typical Interview Process

Although the exact process varies between organisations, most data science interviews follow a broadly similar structure. You will typically be in contact with a recruitment specialist who will guide you through the interview process. Large companies will likely have their own in-house

recruitment teams who you will liaise with about interview dates and next steps. Independent recruiters specialise in particular fields and will typically put you in touch with small to medium size organisations. Typically, all communication with the interview team will be done through the recruiter or the hiring manager.

A typical hiring process may include several stages:



Each stage focuses on slightly different skills. If you pass one stage then you will move on to the next interview. Typically, there will be few days in-between each interview. It is important to keep in touch with the recruitment specialist about when the next stage will be. In the following we will briefly describe each of these stages.

2.2.1 Initial Screening Interview

The screening interview is often conducted by a recruiter or a member of the data science team. Its purpose is to determine whether the candidate's background matches the requirements of the role. During this stage, candidates may be asked questions such as:

- Can you describe your experience with machine learning?
- What projects have you worked on during your degree?
- What programming languages are you comfortable using?

These questions are not designed to test deep technical knowledge. Instead, they help the interviewer understand whether the candidate has the appropriate academic background and experience. Importantly, your communication and presentation skills as well as other *soft skills* will be assessed at this stage.

Index

A/B testing, 761, 766, 770
activation function, 318
activation functions, 176, 301
AdaBoost, 157
AdaGrad, 262
Adam, 266, 272, 277, 278
AdamW, 277
Adaptive Moment Estimation, 266
Adaptive optimisers, 272, 273
AIC, 504
alternative hypothesis, 525
AUC, 443, 449, 454

backpropagation, 301, 306
bagging, 134, 139
Batch normalisation, 328, 331
Bayes' theorem, 585
Bayesian inference, 565
Bayesian optimisation, 345
bias, 20, 167
bias–variance trade-off, 20, 27
BIC, 504
boosting, 157, 167
bootstrap sampling, 134, 139

Central Limit Theorem, 548
chain rule, 306
class labels, 227
classification, 101, 182, 227, 417, 540
classification metrics, 417
classifier, 159
communication, 13
conditional probability, 582
confusion matrix, 436
cosine distance, 629
credit risk modelling, 95
Cross-Entropy, 193
cross-entropy, 211, 219
cross-validation, 393, 403
curse of dimensionality, 603
customer segmentation, 109

data drift, 724, 728
data leakage, 381, 387, 686
data preprocessing, 668
decision trees, 106
deep neural network, 176, 314
Deep neural networks, 331
derivatives, 307
discriminative, 642, 646
distance measures, 629
distributed ML, 755

early stopping, 349, 354
Elastic net, 296
ensembles, 153
Euclidean, 629
evaluation, 335
evaluation metrics, 668
experimentation, 760
exploding gradient, 310
exploding gradient problem, 301
exploding gradients, 314

F1 score, 428
false negatives, 467
false positives, 467
feature engineering, 590, 668
feature scaling, 599

- forecast demand, 798
- fraud detection, 95, 109, 443, 679, 784
- Gaussian distribution, 202
- generative, 642
- Gini impurity, 105
- gradient boosting, 142, 157, 162
- gradient descent, 193, 231, 237, 244
- grid search, 338, 341
- harmonic mean, 428
- He initialisation, 320
- heteroscedasticity, 69
- homoscedasticity, 69
- hyperparameter optimisation, 341
- hyperparameters, 333, 363
- hypothesis testing, 526
- i.i.d., 403
- imbalanced classification, 432
- imbalanced data, 701
- imbalanced datasets, 418
- imputation, 684
- Independence, 554
- inference, 736
- Information criteria, 504
- instance-based learning, 52
- interpretability, 650
- interviews, 2
- k-fold cross validation, 396, 408
- k-means, 632, 639
- KL divergence, 215
- L1 penalty, 296
- L1 regularisation, 291
- L2 penalty, 289, 296
- L2 regularisation, 279, 286
- label encoding, 594
- Lasso regression, 291, 296
- learning curves, 369, 375
- learning rate, 234, 239, 275
- LightGBM, 165
- likelihood, 65, 202, 228, 586
- linear regression, 53, 54, 59, 91, 197
- Log loss, 541
- log-likelihood, 65, 541
- log-loss, 219
- log-odds, 96
- logistic regression, 49, 86, 91, 197
- Loss function, 707
- loss function, 193, 227, 232, 244
- MAE, 476
- Manhattan, 629
- MAP, 568
- Maximum Likelihood Estimation, 558
- maximum likelihood estimation, 64, 202, 228
- Mean absolute error, 476
- Mean squared error, 472
- mean squared error, 198
- medical diagnosis, 95
- mini-batch gradient descent, 251
- missing data, 686
- MLE, 558
- model deployment, 717
- model drift, 721
- model selection, 412
- model serving, 732
- model training, 668
- model-based learning, 52
- momentum, 253, 258
- monitoring, 668
- MSE, 198, 472
- multi-armed bandit, 770
- multi-class classification, 228
- multicollinearity, 71, 79, 82
- natural language processing, 171
- nested cross-validation, 412
- neural network, 172, 176, 302
- neural networks, 171
- neuron, 176
- non-linear, 172
- non-linear functions, 171
- non-linearity, 176
- non-parametric, 37
- null hypothesis, 525

- OLS, 60
- one-hot encoding, 594
- one-tailed test, 525
- online learning, 740, 745
- optimisation, 223, 231, 286, 301, 314, 315, 334, 335
- Ordinary least squares, 59
- ordinary least squares, 64, 286
- outliers, 690, 696
- overfitting, 20, 280, 281, 333, 369

- p-value, 517, 521
- parametric, 37
- PCA, 611, 616, 620, 696
- polynomial regression, 29
- posterior, 586
- precision, 460
- precision–recall curve, 460
- prediction errors, 55
- Principal Component Analysis, 611
- prior, 568, 586
- probabilistic machine learning, 215
- probabilities, 182
- probability, 96, 582
- production, 731
- PyTorch, 189

- R-squared, 485, 489
- random forests, 142
- random search, 341
- recall, 444, 460
- Receiver Operating Characteristic, 443
- recommendation system, 766, 776
- regularisation, 277, 280, 333, 354
- ReLU, 176, 306, 320, 324
- residual errors, 164
- residuals, 64
- Ridge regression, 286, 296
- RMSE, 480
- RMSProp, 261
- robust regression, 197
- robustness, 650
- ROC curves, 443

- Root mean squared error, 480

- scaling, 749
- scree plot, 622
- SHAP, 654, 658, 661
- sigmoid, 87, 101, 306, 320
- SMOTE, 711
- softmax, 182
- spam detection, 791
- spam filtering, 95
- statistical significance, 516
- stochastic gradient descent, 245, 246
- stratified cross-validation, 408
- supervised learning, 624

- tanh, 306, 320
- target encoding, 594
- technical interviews, 2, 9
- TensorFlow, 189
- testing, 358
- time series, 686
- training, 358, 362
- training error, 350
- two-tailed, 525
- Type I error, 526
- Type II error, 526

- underfitting, 20, 333, 369
- unstable optimisation, 333
- unsupervised learning, 625

- validation, 358, 365
- validation data, 335
- validation error, 350
- validation loss, 372
- vanishing gradient problem, 301
- vanishing gradients, 314
- variance, 20, 324
- variance inflation factor, 79
- VIF, 79

- weak learners, 158
- weight decay, 277
- weight initialisation, 301, 315, 317, 318

Xavier initialisation, [320](#)

XGBoost, [165](#)

z-score, [599](#)