

Index

Symbols

- 1NF (first normal form), [Relational Model](#)
- 2NF (second normal form), [Relational Model](#)

A

- A/B testing, [A/B Testing-A/B Testing](#), [Bandits](#)
- accuracy-related metrics, [Monitoring accuracy-related metrics](#)
- ACID (atomicity, consistency, isolation, durability), [Transactional and Analytical Processing](#)
- active learning, [Active learning-Active learning](#)
- ad hoc analytics, [Experiment tracking](#)
- adaptability, [Adaptability](#)
- adversarial attacks, [Evaluation challenge](#)
- adversarial augmentation, [Perturbation](#)
- AI (artificial intelligence), ethics, [Responsible AI](#), [Discover sources for model biases-Discover sources for model biases](#)
 - data-driven approach limitations, [Understand the limitations of the data-driven approach](#)
 - irresponsible, case studies, [Irresponsible AI: Case Studies-Case study II: The danger of “anonymized” data](#)
 - mitigating biases, [Establish processes for mitigating biases](#)
 - model cards, [Create model cards-Create model cards](#)
 - trade-offs, [Understand the trade-offs between different desiderata](#)
- Airflow, [Data Science Workflow Management-Data Science Workflow Management](#)
- alert fatigue, [Monitoring features](#), [Alerts](#)
- alert policies, [Alerts](#)
- algorithms
 - bandit algorithms, [Bandits-Contextual bandits as an exploration strategy](#)
 - continual learning and, [Algorithm challenge-Algorithm challenge](#)

- feature importance, [Feature Importance](#)
- analytical processing, [Transactional and Analytical Processing](#)
- Apache Iceberg, [Transactional and Analytical Processing](#)
- architectural search, [Hard AutoML: Architecture search and learned optimizer](#)
- Argo, [Data Science Workflow Management-Data Science Workflow Management](#)
- artifacts, [Experiment Tracking and Versioning](#)
- artificial intelligence (see AI)
- asynchronous prediction, [Batch Prediction Versus Online Prediction](#)
- automated retraining, [Stage 2: Automated retraining-Requirements](#)
- AutoML
 - architecture search, [Hard AutoML: Architecture search and learned optimizer-Hard AutoML: Architecture search and learned optimizer](#)
 - hard AutoML, [Hard AutoML: Architecture search and learned optimizer-Hard AutoML: Architecture search and learned optimizer](#)
 - hyperparameter tuning, [Soft AutoML: Hyperparameter tuning-Soft AutoML: Hyperparameter tuning](#)
 - learned optimizer, [Hard AutoML: Architecture search and learned optimizer-Hard AutoML: Architecture search and learned optimizer](#)
 - soft AutoML, [Soft AutoML: Hyperparameter tuning-Soft AutoML: Hyperparameter tuning](#)
- autoscaling, [Scalability](#)

B

- bagging, ensembles, [Bagging-Bagging](#)
- bandit algorithms, [Bandits-Contextual bandits as an exploration strategy](#)
- BASE (basically available, soft state, and eventual consistency), [Transactional and Analytical Processing](#)
- base learners, [Ensembles](#)
- base model, fine tuning, [Transfer learning](#)
- baselines, offline model evaluation, [Baselines](#)
 - existing solutions, [Baselines](#)

- human, [Baselines](#)
- random, [Baselines](#)
- simple heuristic, [Baselines](#)
- zero rule, [Baselines](#)
- batch pipeline, [Unifying Batch Pipeline and Streaming Pipeline-Unifying Batch Pipeline and Streaming Pipeline](#)
- batch prediction, [Batch Prediction Versus Online Prediction-Batch Prediction Versus Online Prediction](#)
 - moving to online prediction, [From Batch Prediction to Online Prediction-From Batch Prediction to Online Prediction](#)
- batch processing, [Batch Processing Versus Stream Processing-Batch Processing Versus Stream Processing](#)
- batches, overfitting, [Versioning](#)
- binary classification, [Binary versus multiclass classification](#)
- binary data, [Text Versus Binary Format](#)
- binary file size, [Text Versus Binary Format](#)
- boosting, ensembles, [Boosting-Boosting](#)
- Borg, [Cron, Schedulers, and Orchestrators](#)
- brand monitoring, [Machine Learning Use Cases](#)
- browsers, ML (machine learning) and, [ML in Browsers](#)
- building versus buying, [Build Versus Buy-Build Versus Buy](#)
- business analysis, [Iterative Process](#)
- business objectives, [Business and ML Objectives-Business and ML Objectives](#)

C

- calibration, [Model calibration-Model calibration](#)
- canary release, [Canary Release](#)
- cardinality, classification tasks and, [Binary versus multiclass classification](#)
- catastrophic forgetting, [Continual Learning](#)
- categorical features, [Encoding Categorical Features-Encoding Categorical Features](#)
- champion model, [Continual Learning](#)
- churn prediction, [Challenges of Class Imbalance](#)
- class imbalance, [Class Imbalance](#)

- algorithm-level methods, [Algorithm-level methods](#)
 - class-balanced loss, [Class-balanced loss](#)
 - cost-sensitive learning, [Cost-sensitive learning](#)
 - focal loss, [Focal loss](#)
- challenges, [Challenges of Class Imbalance-Challenges of Class Imbalance](#)
- evaluation metrics, [Using the right evaluation metrics-Using the right evaluation metrics](#)
- resampling, [Data-level methods: Resampling-Data-level methods: Resampling](#)
- class-balanced loss, [Class-balanced loss](#)
- classification
 - as regression problem, [Using the right evaluation metrics](#)
 - binary, [Binary versus multiclass classification](#)
 - hierarchical, [Binary versus multiclass classification](#)
 - high cardinality, [Binary versus multiclass classification](#)
 - multiclass, [Binary versus multiclass classification](#), [Multiclass versus multilabel classification](#)
 - multilabel, [Multiclass versus multilabel classification](#)
 - sentiment analysis, [Learned Features Versus Engineered Features](#)
- classification models, [Classification versus regression](#)
- cloud computing, [ML on the Cloud and on the Edge](#), [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - elasticity, [Public Cloud Versus Private Data Centers](#)
 - multicloud strategy, [Public Cloud Versus Private Data Centers](#)
- code versioning, [Versioning](#)
- column deletion, [Deletion](#)
- column-major formats, [Row-Major Versus Column-Major Format-Row-Major Versus Column-Major Format](#)
 - pandas, [Row-Major Versus Column-Major Format](#)
 - Parquet, [Row-Major Versus Column-Major Format](#)
- Commuter, [IDE](#)
- compact convolutional filters, [Low-Rank Factorization](#)
- computational priorities, [Computational priorities](#)
- compute-intensive problems, [When to Use Machine Learning](#)
- concept drift, [Types of Data Distribution Shifts](#), [Concept drift](#)
- confidence measurement, [Confidence measurement](#)

- containers, [From Dev to Prod: Containers-From Dev to Prod: Containers](#)
- contextual bandits, [Contextual bandits as an exploration strategy](#)
- continual learning, [Iterative Process](#), [Continual Learning](#), [Why Continual Learning?-Why Continual Learning?](#)
 - algorithms and, [Algorithm challenge-Algorithm challenge](#)
 - evaluation and, [Evaluation challenge-Evaluation challenge](#)
 - feature reuse, [Requirements](#)
 - fresh data access, [Fresh data access challenge-Fresh data access challenge](#)
 - stateful training, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - automated, [Stage 3: Automated, stateful training-Requirements](#)
 - stateless retraining, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - manual, [Stage 1: Manual, stateless retraining](#)
 - training, automated retraining, [Stage 2: Automated retraining-Requirements](#)
 - versus online learning, [Stateless Retraining Versus Stateful Training](#)
- convenience sampling, [Nonprobability Sampling](#)
- cost-sensitive learning, [Cost-sensitive learning](#)
- covariate data distribution shift, [Covariate shift-Covariate shift](#)
- cron, schedulers, [Cron, Schedulers, and Orchestrators-Cron, Schedulers, and Orchestrators](#)
- cross-functional collaboration, teams, [Cross-functional Teams Collaboration](#)
- CSV (comma-separated values), row-major format, [Row-Major Versus Column-Major Format](#)

D

- DAG (directed acyclic graph), [Cron, Schedulers, and Orchestrators](#)
- dashboards, monitoring and, [Dashboards](#)
- data, [When to Use Machine Learning](#), [Data](#)
 - mind versus data, [Mind Versus Data-Mind Versus Data](#)
 - training (see training data)

- unseen data, [When to Use Machine Learning](#)
- data augmentation, [Data Augmentation](#)
 - adversarial augmentation, [Perturbation](#)
 - data synthesis, [Data Synthesis-Data Synthesis](#)
 - perturbation, [Perturbation-Perturbation](#)
 - simple label-preserving transformations, [Simple Label-Preserving Transformations](#)
- data distribution shifts
 - addressing, [Addressing Data Distribution Shifts-Addressing Data Distribution Shifts](#)
 - detection
 - statistical methods, [Statistical methods-Statistical methods](#)
 - time scale windows, [Time scale windows for detecting shifts-Time scale windows for detecting shifts](#)
 - ML system failure, [Data Distribution Shifts](#)
 - concept drift, [Types of Data Distribution Shifts](#), [Concept drift](#)
 - covariate shift, [Covariate shift-Covariate shift](#)
 - feature change, [General Data Distribution Shifts](#)
 - label schema change, [General Data Distribution Shifts](#)
 - label shift, [Types of Data Distribution Shifts](#), [Label shift](#)
- data duplication, data leakage and, [Poor handling of data duplication before splitting](#)
- data engineering, [Iterative Process](#)
- data formats, [Data Formats](#)
 - binary, [Text Versus Binary Format](#)
 - column-major, [Row-Major Versus Column-Major Format-Row-Major Versus Column-Major Format](#)
 - JSON, [JSON](#)
 - multimodal data, [Data Formats](#)
 - relational model, NoSQL, [NoSQL-Graph model](#)
 - row-major, [Row-Major Versus Column-Major Format-Row-Major Versus Column-Major Format](#)
 - text, [Text Versus Binary Format](#)
- data freshness, model updates and, [Value of data freshness-Value of data freshness](#)
- data generation, data leakage and, [Leakage from data generation process](#)

- data iteration, [Stateless Retraining Versus Stateful Training](#)
 - model updates and, [Model iteration versus data iteration](#)
- data leakage, [Data Leakage](#)
 - data duplication prior to splitting, [Poor handling of data duplication before splitting](#)
 - data generation process and, [Leakage from data generation process](#)
 - detecting, [Detecting Data Leakage](#)
 - group leakage, [Group leakage](#)
 - Kaggle competition, [Data Leakage](#)
 - scaling before splitting, [Scaling before splitting](#)
 - statistics from test split, missing data and, [Filling in missing data with statistics from the test split](#)
 - time-correlated data, [Splitting time-correlated data randomly instead of by time](#)
- data models
 - relational, [Relational Model-Relational Model](#)
 - structured data, [Structured Versus Unstructured Data-Structured Versus Unstructured Data](#)
 - unstructured data, [Structured Versus Unstructured Data-Structured Versus Unstructured Data](#)
- data normalization, [Relational Model](#)
- data parallelism, distributed training and, [Data parallelism-Data parallelism](#)
- data scientists, teams, [Approach 2: Data scientists own the entire process-Approach 2: Data scientists own the entire process](#)
- data sources, [Data Sources](#)
 - databases, internal, [Data Sources](#)
 - logs, [Data Sources](#)
 - smartphones and, [Data Sources](#)
 - system-generated data, [Data Sources](#)
 - third-party data, [Data Sources](#)
 - user input, [Data Sources](#)
- data synthesis, [Data Synthesis-Data Synthesis](#)
- data-driven approach, AI ethics and, [Understand the limitations of the data-driven approach](#)
- databases and dataflow, [Data Passing Through Databases](#)
- dataflow, [Modes of Dataflow](#)

- message queue model, [Data Passing Through Real-Time Transport](#)
- passing through databases, [Data Passing Through Databases](#)
- passing through real-time transport, [Data Passing Through Real-Time Transport-Data Passing Through Real-Time Transport](#)
- passing through services, [Data Passing Through Services-Data Passing Through Services](#)
- request driven, [Data Passing Through Real-Time Transport](#)
- DataFrame, pandas and, [Row-Major Versus Column-Major Format](#)
- debugging, [Versioning](#)
- decision trees, pruning, [Pruning-Pruning](#)
- declarative ML systems, [Relational Model](#)
- deep learning
 - ML (machine learning) and, [Overview of Machine Learning Systems](#)
 - ML algorithms and, [Evaluating ML Models](#)
- degenerate feedback loops, ML system failure, [Degenerate feedback loops](#)
 - correcting, [Correcting degenerate feedback loops-Correcting degenerate feedback loops](#)
- dependencies, [Cron, Schedulers, and Orchestrators](#)
 - ML models, model store, [Model Store](#)
- dependency failure, [Software System Failures](#)
- deployment, [Iterative Process, Model Deployment and Prediction Service](#)
 - endpoints, exposing, [Model Deployment and Prediction Service](#)
 - failure, [Software System Failures](#)
 - ML models, [Model Deployment](#)
- myths
 - limited models at once, [Myth 1: You Only Deploy One or Two ML Models at a Time-Myth 1: You Only Deploy One or Two ML Models at a Time](#)
 - model updating, [Myth 3: You Won't Need to Update Your Models as Much](#)
 - performance, [Myth 2: If We Don't Do Anything, Model Performance Remains the Same](#)
 - scale, [Myth 4: Most ML Engineers Don't Need to Worry About Scale](#)

- separation of responsibilities, [Model Deployment and Prediction Service](#)
- shadow deployment, [Shadow Deployment](#)
- development environment, infrastructure, [Infrastructure and Tooling for MLOps](#), [Development Environment](#)
 - containers, [From Dev to Prod: Containers-From Dev to Prod: Containers](#)
 - setup, [Dev Environment Setup](#)
 - IDE, [IDE-IDE](#)
 - standardization, [Standardizing Dev Environments-Standardizing Dev Environments](#)
- directed acyclic graph (DAG), [Cron, Schedulers, and Orchestrators](#)
- directional expectation tests, [Directional expectation tests](#)
- discretization, feature engineering and, [Discretization-Discretization](#)
- distributed training, [Distributed Training](#)
 - data parallelism and, [Data parallelism](#)
 - model parallelism and, [Model parallelism-Model parallelism](#)
- Docker Compose, [From Dev to Prod: Containers](#)
- Docker images, [From Dev to Prod: Containers-From Dev to Prod: Containers](#)
- Dockerfiles, [From Dev to Prod: Containers-From Dev to Prod: Containers](#)
- document model, [Document model](#)
 - schemas, [Document model](#)
- downtime, [Software System Failures](#)
- driver management service, [Data Passing Through Services](#)
- dynamic sampling, [Data-level methods: Resampling](#)

E

- edge cases
 - failure and, [Edge cases-Edge cases](#)
 - outliers and, [Edge cases](#)
- edge computing, [ML on the Cloud and on the Edge](#)
 - model optimization, [Compiling and Optimizing Models for Edge Devices-Using ML to optimize ML models](#)
- EKS (Elastic Kubernetes Service), [Cron, Schedulers, and Orchestrators](#)

- embedding
 - positional embedding, [Discrete and Continuous Positional Embeddings-Discrete and Continuous Positional Embeddings](#)
 - word embeddings, [Discrete and Continuous Positional Embeddings](#)
- endpoint, exposing, [Model Deployment and Prediction Service](#)
- ensembles
 - bagging, [Bagging-Bagging](#)
 - base learners, [Ensembles](#)
 - boosting, [Boosting-Boosting](#)
 - spam classifiers, [Ensembles](#)
 - stacking, [Stacking](#)
- ethics in AI, [Responsible AI-Case study II: The danger of “anonymized” data](#)
- ETL (extract, transform, load), [ETL: Extract, Transform, and Load-ETL: Extract, Transform, and Load](#)
- evaluation, offline
 - confidence measurement, [Confidence measurement](#)
 - directional expectation tests, [Directional expectation tests](#)
 - invariance tests, [Invariance tests](#)
 - model calibration, [Model calibration-Model calibration](#)
 - perturbation tests, [Perturbation tests-Perturbation tests](#)
 - slice-based, [Slice-based evaluation-Slice-based evaluation](#)
- existing data, [When to Use Machine Learning](#)
- experiment artifacts, development and, [Model Store](#)
- experiment tracking, [Experiment tracking-Experiment tracking](#)
 - third-party tools, [Experiment tracking](#)
- exporting models, [Model Deployment and Prediction Service](#)

F

- F1 metrics, [Using the right evaluation metrics](#)
- factorization, low-rank, [Low-Rank Factorization-Low-Rank Factorization](#)
- fairness, [Fairness](#)
- feature change, [General Data Distribution Shifts](#)
- feature engineering, [Learned Features Versus Engineered Features-Learned Features Versus Engineered Features](#)

- categorical features, [Encoding Categorical Features-Encoding Categorical Features](#)
- discretization, [Discretization-Discretization](#)
- feature crossing, [Feature Crossing](#)
- feature generalization, [Feature Generalization-Feature Generalization](#)
- feature importance, [Feature Importance](#)
- missing values and, [Handling Missing Values](#)
 - deletion, [Deletion](#)
 - imputation, [Imputation-Imputation](#)
 - MAR (missing at random), [Handling Missing Values](#)
 - MCAR (missing completely at random), [Handling Missing Values](#)
 - MNAR (missing not at random), [Handling Missing Values](#)
- NLP (natural language processing) and, [Learned Features Versus Engineered Features](#)
- positional embeddings, [Discrete and Continuous Positional Embeddings-Discrete and Continuous Positional Embeddings](#)
- predictive power of features, [Detecting Data Leakage](#)
- scaling, [Scaling-Scaling](#)
- useless features, [Engineering Good Features](#)
- feature scaling, [Scaling-Scaling](#)
- feature store, [Feature Store-Feature Store](#)
- features
 - computation, [Feature Store](#)
 - consistency, [Feature Store](#)
 - extracting, [Monitoring features](#)
 - failures and, [Versioning](#)
 - learned, [Learned Features Versus Engineered Features-Learned Features Versus Engineered Features](#)
 - management, [Feature Store](#)
 - monitoring, [Monitoring features-Monitoring features](#)
 - online, [Batch Prediction Versus Online Prediction](#)
 - reuse, [Requirements](#)
 - streaming, [Batch Prediction Versus Online Prediction](#)
- feedback loops, [Bandits](#)
 - ML system failure, [Detecting degenerate feedback loops](#)
- feedback, users, [Feedback loop length](#)

- fixed positional embeddings, [Discrete and Continuous Positional Embeddings](#)
- fixed-point inference, [Quantization](#)
- FLOPS (floating-point operations per second), [Storage and Compute](#)
- forecasting customer demand, [Machine Learning Use Cases](#)
- Fourier features, [Discrete and Continuous Positional Embeddings](#)
- fraud detection, [Machine Learning Use Cases](#), [Challenges of Class Imbalance](#)

G

- GDPR (General Data Protection Regulation), [Versioning](#)
- generalization, features, [Feature Generalization-Feature Generalization](#)
- GKE (Google Kubernetes Engine), [Cron, Schedulers, and Orchestrators](#)
- Google Translate, [Overview of Machine Learning Systems](#)
- graph model, [Graph model](#)

H

- H2O AutoML, [Relational Model](#)
- hand labels, [Hand Labels](#)
 - lineage, [Data lineage](#)
 - multiplicity, [Label multiplicity-Label multiplicity](#)
- hard AutoML, [Hard AutoML: Architecture search and learned optimizer-Hard AutoML: Architecture search and learned optimizer](#)
- hardware failure, [Software System Failures](#)
- hashed functions, [Encoding Categorical Features](#)
- heuristics, LFs (labeling functions), [Weak supervision](#)
- heuristics-based slicing, [Slice-based evaluation](#)
- hierarchical classification, [Binary versus multiclass classification](#)
- human baselines, [Baselines](#)
- hyperparameters
 - failures and, [Versioning](#)
 - tuning, [Soft AutoML: Hyperparameter tuning-Soft AutoML: Hyperparameter tuning](#)
 - values over time, [Experiment tracking](#)

- IDE (integrated development environment), [IDE](#)
 - cloud dev environment, [Standardizing Dev Environments](#)
 - notebooks and, [IDE](#)
- importance sampling, [Importance Sampling](#)
- infrastructure, [Infrastructure and Tooling for MLOps](#), [Infrastructure and Tooling for MLOps](#)
 - building versus buying, [Build Versus Buy-Build Versus Buy](#)
 - cloud computing and, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - development environment layer, [Infrastructure and Tooling for MLOps](#), [Development Environment](#)
 - setup, [Dev Environment Setup-IDE](#)
 - fundamental facilities, [Infrastructure and Tooling for MLOps](#)
 - ML platform layer, [Infrastructure and Tooling for MLOps](#)
 - requirements, [Infrastructure and Tooling for MLOps](#)
 - resource management layer, [Infrastructure and Tooling for MLOps](#)
 - storage and compute layer, [Infrastructure and Tooling for MLOps](#), [Infrastructure and Tooling for MLOps](#), [Storage and Compute](#)
 - compute resources, [Storage and Compute](#)
 - FLOPS, [Storage and Compute](#)
 - private data centers, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - public cloud, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - units, [Storage and Compute](#)
- input, monitoring, [Monitoring raw inputs](#)
- instances on-demand, [Public Cloud Versus Private Data Centers](#)
- integrated development environment (see IDE)
- interleaving experiments, [Interleaving Experiments-Interleaving Experiments](#)
- internal databases, [Data Sources](#)
- interpretability, [Interpretability](#)
- invariance tests, [Invariance tests](#)
- IR (intermediate representation), [Compiling and Optimizing Models for Edge Devices](#)

- iterative processes
 - model development and, [Iterative Process](#)
 - performance check, [Model Development and Offline Evaluation](#)
 - model updates and, [Model iteration versus data iteration](#)
 - training the model and, [Iterative Process-Iterative Process](#)
 - data engineering, [Iterative Process](#)
 - project scoping, [Iterative Process](#)

J

- JSON (JavaScript Object Notation), [JSON](#)
- judgment sampling, [Nonprobability Sampling](#)

K

- k-means clustering models, [Evaluating ML Models](#)
- Kaggle, data leakage, [Data Leakage](#)
- knowledge distillation, [Knowledge Distillation](#)
- Kubeflow, [Data Science Workflow Management](#)
- Kubernetes (K8s), [From Dev to Prod: Containers, Cron, Schedulers, and Orchestrators](#)
 - EKS (Elastic Kubernetes Service), [Cron, Schedulers, and Orchestrators](#)
 - GKE (Google Kubernetes Engine), [Cron, Schedulers, and Orchestrators](#)

L

- label computation, [Fresh data access challenge](#)
- label schema change, [General Data Distribution Shifts](#)
- label shift, [Types of Data Distribution Shifts](#), [Label shift](#)
- labeling, [Labeling](#)
 - class imbalance and, [Class Imbalance](#)
 - errors, class imbalance and, [Challenges of Class Imbalance](#)
 - hand labels, [Hand Labels](#)
 - lineage, [Data lineage](#)
 - multiplicity, [Label multiplicity-Label multiplicity](#)
 - lack of labels, [Handling the Lack of Labels](#)

- active learning, [Active learning-Active learning](#)
- semi-supervision, [Semi-supervision-Semi-supervision](#)
- transfer learning, [Transfer learning-Transfer learning](#)
- weak supervision, [Weak supervision-Weak supervision](#)
- ML algorithms, [Evaluating ML Models](#)
- natural labels, [Natural Labels](#)
 - feedback loop length, [Feedback loop length](#)
 - recommender systems, [Natural Labels](#)
- perturbation, [Perturbation-Perturbation](#)
- simple label-preserving transformations, [Simple Label-Preserving Transformations](#)
- language modeling, sampling and, [Nonprobability Sampling](#)
- latency, [Computational priorities](#)
- latency versus throughput, [Computational priorities-Computational priorities](#)
- learning, [When to Use Machine Learning](#)
- LFs (labeling functions), [Weak supervision](#)
 - heuristics, [Weak supervision](#)
- logs, [Data Sources](#), [Data Sources](#)
 - experiment tracking, [Experiment tracking](#)
 - monitoring and, [Logs-Logs](#)
 - storage, [Data Sources](#)
- loop tiling, model optimization, [Model optimization](#)
- loss curve, [Experiment tracking](#)
- loss functions, [Objective Functions](#)
 - (see also objective functions)
- low-rank factorization, [Low-Rank Factorization-Low-Rank Factorization](#)

M

- maintainability, [Maintainability](#)
- Manning, Christopher, [Mind Versus Data](#)
- MAR (missing at random) values, [Handling Missing Values](#)
- MCAR (missing completely at random) values, [Handling Missing Values](#)
- merge conflicts, [Versioning](#)

- message queue, dataflow and, [Data Passing Through Real-Time Transport](#)
- Metaflow, [Data Science Workflow Management](#)
- metrics
 - monitoring and, [Monitoring and Observability](#)
 - accuracy-related metrics, [Monitoring accuracy-related metrics](#)
 - features, [Monitoring features-Monitoring features](#)
 - predictions, [Monitoring predictions-Monitoring predictions](#)
 - raw input, [Monitoring raw inputs](#)
 - performance metrics, [Experiment tracking](#)
 - system performance, [Experiment tracking](#)
- mind versus data, [Mind Versus Data-Mind Versus Data](#)
- missing at random (MAR), [Handling Missing Values](#)
- missing completely at random (MCAR), [Handling Missing Values](#)
- missing data, test split statistics and, [Filling in missing data with statistics from the test split](#)
- missing not at random (MNAR), [Handling Missing Values](#)
- ML (machine learning)
 - browsers and, [ML in Browsers, ML in Browsers](#)
 - cloud computing, [ML on the Cloud and on the Edge-ML in Browsers](#)
 - complex patterns, [When to Use Machine Learning](#)
 - deep learning and, [Overview of Machine Learning Systems](#)
 - edge computing, [ML on the Cloud and on the Edge-ML in Browsers](#)
 - existing data and, [When to Use Machine Learning](#)
 - learning, [When to Use Machine Learning](#)
 - model optimization, [Using ML to optimize ML models-Using ML to optimize ML models](#)
 - predictions and, [When to Use Machine Learning](#)
 - production and, [Machine Learning in Research Versus in Production-Discussion](#)
 - repetition, [When to Use Machine Learning](#)
 - research and, [Machine Learning in Research Versus in Production-Discussion](#)
 - scale, [When to Use Machine Learning](#)
 - smartphones and, [Machine Learning Use Cases](#)
 - unseen data, [When to Use Machine Learning](#)

- use cases, [Machine Learning Use Cases-Machine Learning Use Cases](#)
- when to use, [When to Use Machine Learning-Machine Learning Use Cases](#)
- ML algorithms, [Overview of Machine Learning Systems, Model Development and Offline Evaluation](#)
 - deep learning and, [Evaluating ML Models](#)
 - labels, [Evaluating ML Models](#)
 - versus neural networks, [Evaluating ML Models](#)
- ML model logic, [Model Deployment and Prediction Service](#)
- ML models
 - continual learning, [Iterative Process](#)
 - data iteration, [Stateless Retraining Versus Stateful Training](#)
 - debugging, [Versioning](#)
 - deployment, [Model Deployment](#)
 - edge computing, optimization, [Compiling and Optimizing Models for Edge Devices-Using ML to optimize ML models](#)
 - ensembles, [Ensembles](#), [Ensembles](#)
 - bagging, [Bagging-Bagging](#)
 - base learners, [Ensembles](#)
 - boosting, [Boosting-Boosting](#)
 - stacking, [Stacking](#)
 - evaluation, [Evaluating ML Models](#)
 - test in production, [Test in Production-Contextual bandits as an exploration strategy](#)
 - experiment tracking, [Experiment tracking-Experiment tracking](#)
 - exporting, [Model Deployment and Prediction Service](#)
 - failures
 - batches, overfitting, [Versioning](#)
 - components, [Versioning](#)
 - data problems, [Versioning](#)
 - feature choice, [Versioning](#)
 - hyperparameters and, [Versioning](#)
 - poor model implementation, [Versioning](#)
 - random seeds, [Versioning](#)
 - theoretical constraints, [Versioning](#)
 - iteration, [Stateless Retraining Versus Stateful Training](#)

- monitoring, [Iterative Process](#)
- offline evaluation, [Model Offline Evaluation](#)
 - baselines, [Baselines-Baselines](#)
 - methods, [Evaluation Methods-Slice-based evaluation](#)
- optimization, [Using ML to optimize ML models-Using ML to optimize ML models](#)
- parameters, model store, [Model Store](#)
- performance metrics, [Experiment tracking](#)
- selection criteria, [Evaluating ML Models](#)
 - human biases in, [Avoid human biases in selecting models](#)
 - model, [Understand your model's assumptions](#)
 - performance now and later, [Evaluate good performance now versus good performance later](#)
 - simple models, [Start with the simplest models](#)
 - state-of-the-art trap, [Avoid the state-of-the-art trap](#)
 - trade-offs, [Evaluate trade-offs](#)
- speed, [Experiment tracking](#)
- training, [Iterative Process-Iterative Process](#)
 - data engineering, [Iterative Process](#)
 - distributed, [Distributed Training-Model parallelism](#)
- update frequency, [How Often to Update Your Models](#)
 - data freshness and, [Value of data freshness-Value of data freshness](#)
 - data iteration and, [Model iteration versus data iteration](#)
 - model iteration and, [Model iteration versus data iteration](#)
- updates, [Stateless Retraining Versus Stateful Training](#)
- versioning, [Versioning-Versioning](#)
- ML platform, [ML Platform](#)
 - model deployment, [Model Deployment](#)
 - model store, [Model Store-Model Store](#)
- ML platform layer, infrastructure, [Infrastructure and Tooling for MLOps](#)
- ML system failures
 - data distribution shifts, [Data Distribution Shifts](#)
 - addressing, [Addressing Data Distribution Shifts-Addressing Data Distribution Shifts](#)
 - concept drift, [Types of Data Distribution Shifts, Concept drift](#)

- covariate, [Covariate shift-Covariate shift](#)
- detection, [Detecting Data Distribution Shifts-Time scale windows for detecting shifts](#)
- feature change, [General Data Distribution Shifts](#)
- label schema change, [General Data Distribution Shifts](#)
- label shifts, [Types of Data Distribution Shifts](#), [Label shift](#)
- ML-system specific
 - degenerate feedback loops, [Degenerate feedback loops-Correcting degenerate feedback loops](#)
 - edge cases, [Edge cases](#)
 - production data different from training data, [Production data differing from training data-Production data differing from training data](#)
- operational expectation violations, [Causes of ML System Failures](#)
- software
 - crashes, [Software System Failures](#)
 - dependency failure, [Software System Failures](#)
 - deployment failure, [Software System Failures](#)
 - downtime, [Software System Failures](#)
 - hardware failure, [Software System Failures](#)
- ML systems
 - declarative, [Relational Model](#)
 - failures, [Causes of ML System Failures](#)
 - iterative processes, [Iterative Process-Iterative Process](#)
 - requirements
 - adaptability, [Adaptability](#)
 - maintainability, [Maintainability](#)
 - reliability, [Reliability](#)
 - scalability, [Scalability-Scalability](#)
 - versus traditional software, [Machine Learning Systems Versus Traditional Software-Machine Learning Systems Versus Traditional Software](#)
- MLOPs, ML systems design and, [Overview of Machine Learning Systems-Overview of Machine Learning Systems](#)
- MNAR (missing not at random) values, [Handling Missing Values](#)
- model biases, AI ethics, [Discover sources for model biases-Discover sources for model biases](#)

- model calibration, [Model calibration-Model calibration](#)
- model cards, AI ethics, [Create model cards-Create model cards](#)
- model compression, [Model Compression](#)
 - knowledge distillation, [Knowledge Distillation](#)
 - low-rank factorization, [Low-Rank Factorization-Low-Rank Factorization](#)
 - pruning, [Pruning-Pruning](#)
 - quantization, [Quantization-Quantization](#)
- model development, [Iterative Process](#)
- model implementation, failures and, [Versioning](#)
- model parallelism, distributed training and, [Model parallelism-Model parallelism](#)
- model performance, business analysis, [Iterative Process](#)
- monitoring, [Monitoring and Observability](#), [Continual Learning and Test in Production](#)
 - (see also test in production)
 - alerts and, [Alerts](#)
 - dashboards and, [Dashboards](#)
 - logs and, [Logs-Logs](#)
 - metrics and, [Monitoring and Observability](#)
 - accuracy-related metrics, [Monitoring accuracy-related metrics](#)
 - features, [Monitoring features-Monitoring features](#)
 - predictions, [Monitoring predictions-Monitoring predictions](#)
 - raw input, [Monitoring raw inputs](#)
- multiclass classification, [Binary versus multiclass classification](#), [Multiclass versus multilabel classification](#)
- multilabel classification, [Multiclass versus multilabel classification](#)
- multimodal data, [Data Formats](#)

N

- n-grams, [Learned Features Versus Engineered Features](#)
- NAS (neural architecture search), [Hard AutoML: Architecture search and learned optimizer](#)
- natural labels, [Natural Labels](#)
 - feedback loop length, [Feedback loop length](#)
 - recommender systems, [Natural Labels](#)

- natural language processing (NLP) (see NLP)
- neural architecture search (NAS), [Hard AutoML: Architecture search and learned optimizer](#)
- neural networks, [Evaluating ML Models](#)
 - positional embedding, [Discrete and Continuous Positional Embeddings](#)
- newsfeeds
 - ranking posts, [Decoupling objectives](#)
 - user engagement and, [Decoupling objectives](#)
- NLP (natural language processing), [Simple Label-Preserving Transformations](#)
 - data augmentation and, [Data Augmentation](#)
 - feature engineering, [Learned Features Versus Engineered Features](#)
- nonprobability sampling, [Nonprobability Sampling](#)
 - biases, [Nonprobability Sampling](#)
- Norvig, Peter, [Mind Versus Data](#)
- NoSQL, [NoSQL](#)
 - document model, [Document model](#)
 - graph model, [Graph model](#)
- notebooks, IDE and, [IDE](#)
- NSFW (not safe for work) content filtering, [Decoupling objectives](#)
- NumPy, [Row-Major Versus Column-Major Format](#)

O

- objective functions, [Objective Functions-Decoupling objectives](#)
- observability, [Monitoring and Observability](#), [Observability-Observability](#)
- offline evaluation of models, [Model Offline Evaluation](#)
 - baselines, [Baselines](#)
 - existing solutions, [Baselines](#)
 - human, [Baselines](#)
 - random, [Baselines](#)
 - simple heuristic, [Baselines](#)
 - zero rule, [Baselines](#)
- OLAP (online analytical processing), [Transactional and Analytical Processing](#)

- OLTP (online transaction processing) system, [Transactional and Analytical Processing](#)
- on-demand instances, [Public Cloud Versus Private Data Centers](#)
- on-demand prediction, [Batch Prediction Versus Online Prediction](#)
- One Billion Word Benchmark for Language Modeling, [Mind Versus Data](#)
- online features, [Batch Prediction Versus Online Prediction](#)
- online learning, [Stateless Retraining Versus Stateful Training](#)
- online prediction, [Batch Prediction Versus Online Prediction](#)-[Batch Prediction Versus Online Prediction](#), [Bandits](#)
 - moving to from batch prediction, [From Batch Prediction to Online Prediction](#)-[From Batch Prediction to Online Prediction](#)
 - streaming pipeline, [Unifying Batch Pipeline and Streaming Pipeline](#)-[Unifying Batch Pipeline and Streaming Pipeline](#)
- operation expectation violations, [Causes of ML System Failures](#)
- operator fusion, model optimization and, [Model optimization](#)
- orchestrators
 - HashiCorp Nomad, [Cron, Schedulers, and Orchestrators](#)
 - Kubernetes (K8s), [Cron, Schedulers, and Orchestrators](#)
- outliers, edge cases and, [Edge cases](#)
- oversampling
 - overfitting, [Data-level methods: Resampling](#)
 - SMOTE, [Data-level methods: Resampling](#)

P

- pandas, [Row-Major Versus Column-Major Format](#)
- Papermill, [IDE](#)
- parallelization, model optimization and, [Model optimization](#)
- parameter values over time, [Experiment tracking](#)
- Pareto optimization, [Decoupling objectives](#)
- Parquet, [Row-Major Versus Column-Major Format](#), [Text Versus Binary Format](#)
 - binary files, [Text Versus Binary Format](#)
- patterns
 - changing, [When to Use Machine Learning](#)
 - complex, [When to Use Machine Learning](#)

- Pearl, Judea, [Mind Versus Data](#)
- performance metrics, [Experiment tracking](#)
 - system performance, [Experiment tracking](#)
- perturbation, [Perturbation-Perturbation](#)
- perturbation method of semi-supervision, [Semi-supervision](#)
- perturbation tests, [Perturbation tests-Perturbation tests](#)
- positional embedding, [Discrete and Continuous Positional Embeddings-Discrete and Continuous Positional Embeddings](#)
 - fixed, [Discrete and Continuous Positional Embeddings](#)
- precision metrics, [Using the right evaluation metrics](#)
- prediction, [When to Use Machine Learning](#), [Multiple ways to frame a problem](#)
 - asynchronous, [Batch Prediction Versus Online Prediction](#)
 - batch prediction, [Batch Prediction Versus Online Prediction-Batch Prediction Versus Online Prediction](#)
 - moving to online prediction, [From Batch Prediction to Online Prediction-From Batch Prediction to Online Prediction](#)
 - “mostly correct,” user experience, [Combatting “Mostly Correct” Predictions-Combatting “Mostly Correct” Predictions](#)
 - on-demand prediction, [Batch Prediction Versus Online Prediction](#)
 - online, [Batch Prediction Versus Online Prediction-Batch Prediction Versus Online Prediction](#)
 - streaming pipeline, [Unifying Batch Pipeline and Streaming Pipeline-Unifying Batch Pipeline and Streaming Pipeline](#)
 - synchronous, [Batch Prediction Versus Online Prediction](#)
- predictions, monitoring, [Monitoring predictions-Monitoring predictions](#)
- predictive power of features, [Detecting Data Leakage](#)
- price optimization service, [Data Passing Through Services](#)
- problem framing, [Framing ML Problems-Decoupling objectives](#)
- processing
 - analytical, [Transactional and Analytical Processing](#)
 - batch processing, [Batch Processing Versus Stream Processing-Batch Processing Versus Stream Processing](#)
 - ETL (extract, transform, load), [ETL: Extract, Transform, and Load-ETL: Extract, Transform, and Load](#)

- stream processing, [Batch Processing Versus Stream Processing](#)
- transactional, [Transactional and Analytical Processing](#)
 - ACID and, [Transactional and Analytical Processing](#)
- production environment, [Model Deployment and Prediction Service](#)
- production, ML and, [Machine Learning in Research Versus in Production-Discussion](#)
- project objectives, [Business and ML Objectives-Business and ML Objectives](#)
- project scoping, [Iterative Process](#)
- prototyping, batch prediction and, [From Batch Prediction to Online Prediction](#)
- pruning, [Pruning-Pruning](#)
- public cloud versus private data center, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)

Q

- quantization, [Quantization-Quantization](#)
- query languages, [Relational Model](#)
- quota sampling, [Nonprobability Sampling](#)

R

- random baselines, [Baselines](#)
- real-time transport
 - dataflow and, [Data Passing Through Real-Time Transport-Data Passing Through Real-Time Transport](#)
 - streaming data and, [Batch Processing Versus Stream Processing](#)
- reasonable scale, [Infrastructure and Tooling for MLOps](#)
- recall metrics, [Using the right evaluation metrics](#)
- recommender systems, labels, [Natural Labels](#)
- regression
 - class imbalance and, [Class Imbalance](#)
 - tasks, [Multiple ways to frame a problem](#)
- regression models, [Classification versus regression](#)
- relational databases, [Relational Model](#)

- relational models, [Relational Model-Relational Model](#)
 - data normalization, [Relational Model](#)
 - NoSQL, [NoSQL](#)
 - document model, [Document model](#)
 - graph model, [Graph model](#)
 - tables, [Relational Model](#)
- reliability, [Reliability](#)
- repetition, [When to Use Machine Learning](#)
- repetitive jobs, scheduling, [Cron, Schedulers, and Orchestrators](#)
- request-driven data passing, [Data Passing Through Real-Time Transport](#)
- resampling, [Data-level methods: Resampling](#)
 - dynamic sampling, [Data-level methods: Resampling](#)
 - oversampling
 - overfitting and, [Data-level methods: Resampling](#)
 - SMOTE, [Data-level methods: Resampling](#)
 - two-phase learning, [Data-level methods: Resampling](#)
 - undersampling, [Data-level methods: Resampling](#)
- reservoir sampling, [Reservoir Sampling-Reservoir Sampling](#)
- resource management, [Resource Management](#)
- resource management layer, infrastructure, [Infrastructure and Tooling for MLOps](#)
- REST (representational state transfer), [Data Passing Through Services](#)
- ride management service, [Data Passing Through Services](#)
- ROC (receiver operating characteristics) curve, [Using the right evaluation metrics](#)
- Rogati, Monica, [Mind Versus Data](#)
- ROI (return on investment), maturity stage of adoption, [Business and ML Objectives](#)
- row deletion, [Deletion](#)
- row-major format, [Row-Major Versus Column-Major Format-Row-Major Versus Column-Major Format](#)
 - CSV (comma-separated values), [Row-Major Versus Column-Major Format](#)
 - NumPy, [Row-Major Versus Column-Major Format](#)
- RPC (remote procedure call), [Data Passing Through Services](#)

S

- sampling, [Sampling](#)
 - importance sampling, [Importance Sampling](#)
 - nonprobability, [Nonprobability Sampling](#)
 - biases, [Nonprobability Sampling](#)
 - reservoir sampling, [Reservoir Sampling-Reservoir Sampling](#)
 - simple random sampling, [Simple Random Sampling](#)
 - stratified sampling, [Stratified Sampling](#)
 - weighted sampling, [Weighted Sampling](#)
- scalability, [Scalability-Scalability](#)
 - autoscaling, [Scalability](#)
- scale, [When to Use Machine Learning](#)
 - deployment myths, [Myth 4: Most ML Engineers Don't Need to Worry About Scale](#)
- schedulers, [Cron, Schedulers, and Orchestrators-Cron, Schedulers, and Orchestrators](#)
 - Borg, [Cron, Schedulers, and Orchestrators](#)
 - Slurm, [Cron, Schedulers, and Orchestrators](#)
- schemas, document model, [Document model](#)
- scoping a project, [Iterative Process](#)
- self-training, [Semi-supervision](#)
- semi-supervision, [Semi-supervision-Semi-supervision](#)
- sentiment analysis classifier, [Learned Features Versus Engineered Features](#)
- serialization, [Model Deployment and Prediction Service](#)
- services
 - dataflow and, [Data Passing Through Services-Data Passing Through Services](#)
 - driver management, [Data Passing Through Services](#)
 - price optimization, [Data Passing Through Services](#)
 - ride management, [Data Passing Through Services](#)
- SGD (stochastic gradient descent), [Data parallelism](#)
- shadow deployment, [Shadow Deployment](#)
- SHAP (SHapley Additive exPlanations, [Feature Importance](#)
- simple heuristic, offline evaluation, [Baselines](#)

- simple label-preserving transformations, [Simple Label-Preserving Transformations](#)
- simple random sampling, [Simple Random Sampling](#)
- Simpson's paradox, [Slice-based evaluation](#)
- skewed distribution, feature scaling and, [Scaling](#)
- slice-based evaluation, [Slice-based evaluation-Slice-based evaluation](#)
- slicing
 - error analysis, [Slice-based evaluation](#)
 - heuristics based, [Slice-based evaluation](#)
 - slice finders, [Slice-based evaluation](#)
- Slurm, [Cron, Schedulers, and Orchestrators](#)
- smartphones
 - data sources and, [Data Sources](#)
 - ML (machine learning) and, [Machine Learning Use Cases](#)
- smooth failing, user experience, [Smooth Failing](#)
- SMOTE (synthetic minority oversampling technique), [Data-level methods: Resampling](#)
- Snorkel, [Weak supervision](#)
- snowball sampling, [Nonprobability Sampling](#)
- soft AutoML, [Soft AutoML: Hyperparameter tuning-Soft AutoML: Hyperparameter tuning](#)
- software system failure
 - crashes, [Software System Failures](#)
 - dependency, [Software System Failures](#)
 - deployment, [Software System Failures](#)
 - hardware, [Software System Failures](#)
- spam filtering, [Decoupling objectives](#)
- splitting
 - data duplication, [Poor handling of data duplication before splitting](#)
 - data leakage and, [Scaling before splitting](#)
- SQL, [Relational Model](#)
- SQL databases, [Relational Model](#)
- SSD (solid state disk), [Storage and Compute](#)
- stacking, ensembles, [Stacking](#)
- stakeholders, research projects, [Different stakeholders and requirements-Different stakeholders and requirements](#)
- state-of-the-art models, [Avoid the state-of-the-art trap](#)

- stateful training, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - automated, [Stage 3: Automated, stateful training-Requirements](#)
- stateless retraining, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - manual, [Stage 1: Manual, stateless retraining](#)
- stochastic gradient descent (SGD), [Data parallelism](#)
- storage and compute layer, infrastructure, [Infrastructure and Tooling for MLOps, Infrastructure and Tooling for MLOps, Storage and Compute](#)
 - compute resources, [Storage and Compute](#)
 - FLOPS (floating-point operations per second), [Storage and Compute](#)
 - private data centers, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - public cloud, [Public Cloud Versus Private Data Centers-Public Cloud Versus Private Data Centers](#)
 - units, [Storage and Compute](#)
- storage engines, [Data Storage Engines and Processing](#)
- stratified sampling, [Stratified Sampling](#)
- stream processing, [Batch Processing Versus Stream Processing-Batch Processing Versus Stream Processing](#)
- streaming data, real-time transport, [Batch Processing Versus Stream Processing](#)
- streaming features, [Batch Prediction Versus Online Prediction](#)
- streaming pipeline, [Unifying Batch Pipeline and Streaming Pipeline-Unifying Batch Pipeline and Streaming Pipeline](#)
- structured data, [Structured Versus Unstructured Data-Structured Versus Unstructured Data](#)
- Sutton, Richard, [Mind Versus Data](#)
- synchronous prediction, [Batch Prediction Versus Online Prediction](#)
- synthetic minority oversampling technique (SMOTE), [Data-level methods: Resampling](#)
- system performance metrics, [Experiment tracking](#)
- system-generated data, [Data Sources](#)

- tags, model store, [Model Store](#)
- tasks
 - classification, [Classification versus regression](#)
 - binary, [Binary versus multiclass classification](#)
 - high cardinality, [Binary versus multiclass classification](#)
 - multiclass, [Binary versus multiclass classification](#), [Multiclass versus multilabel classification](#)
 - multilabel, [Multiclass versus multilabel classification](#)
 - labels, [Natural Labels](#)
 - regression, [Classification versus regression](#), [Multiple ways to frame a problem](#)
- teams
 - cross-functional collaboration, [Cross-functional Teams Collaboration](#)
 - data scientists, [Approach 2: Data scientists own the entire process-Approach 2: Data scientists own the entire process](#)
 - production management, [Approach 1: Have a separate team to manage production](#)
- telemetry, [Observability](#)
- test in production, [Continual Learning and Test in Production](#), [Test in Production](#)
 - A/B testing, [A/B Testing-A/B Testing](#)
 - bandits, [Bandits-Contextual bandits as an exploration strategy](#)
 - canary release, [Canary Release](#)
 - interleaving experiments, [Interleaving Experiments-Interleaving Experiments](#)
 - shadow deployment and, [Shadow Deployment](#)
- text data, [Text Versus Binary Format](#)
- text file size, [Text Versus Binary Format](#)
- theoretical constraints, failures and, [Versioning](#)
- third-party data, [Data Sources](#)
- time-correlated data, data leakage and, [Splitting time-correlated data randomly instead of by time](#)
- training
 - automated retraining, [Stage 2: Automated retraining-Requirements](#)
 - distributed, [Distributed Training](#)
 - data parallelism and, [Data parallelism-Data parallelism](#)

- model parallelism and, [Model parallelism-Model parallelism](#)
- stateful, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - automated, [Stage 3: Automated, stateful training-Requirements](#)
- stateless retraining, [Stateless Retraining Versus Stateful Training-Stateless Retraining Versus Stateful Training](#)
 - manual, [Stage 1: Manual, stateless retraining](#)
- training data, [Training Data](#)
 - class imbalance, [Class Imbalance](#)
 - algorithm-level methods, [Algorithm-level methods-Focal loss](#)
 - challenges, [Challenges of Class Imbalance-Challenges of Class Imbalance](#)
 - evaluation metrics, [Using the right evaluation metrics-Using the right evaluation metrics](#)
 - resampling, [Data-level methods: Resampling-Data-level methods: Resampling](#)
 - data augmentation, [Data Augmentation](#)
 - perturbation, [Perturbation-Perturbation](#)
 - simple label-preserving transformations, [Simple Label-Preserving Transformations](#)
 - data distributions, [Production data differing from training data](#)
 - data leakage, [Data Leakage](#)
 - labeling, [Labeling](#)
 - hand labels, [Hand Labels-Data lineage](#)
 - lack of labels, [Handling the Lack of Labels-Active learning](#)
 - natural labels, [Natural Labels-Feedback loop length](#)
 - user feedback, [Feedback loop length](#)
 - n-grams, [Learned Features Versus Engineered Features](#)
 - noisy samples, [Perturbation](#)
 - sampling, [Sampling](#)
 - importance sampling, [Importance Sampling](#)
 - nonprobability, [Nonprobability Sampling-Nonprobability Sampling](#)
 - reservoir sampling, [Reservoir Sampling-Reservoir Sampling](#)
 - simple random sampling, [Simple Random Sampling](#)
 - stratified sampling, [Stratified Sampling](#)
 - weighted sampling, [Weighted Sampling](#)

- training the model, iteration and, [Iterative Process-Iterative Process](#)
 - data engineering, [Iterative Process](#)
 - project scoping, [Iterative Process](#)
- transactional processing, [Transactional and Analytical Processing](#)
 - ACID and, [Transactional and Analytical Processing](#)
- transfer learning, [Transfer learning-Transfer learning](#)
- two-phase learning, [Data-level methods: Resampling](#)

U

- undersampling, [Data-level methods: Resampling](#)
- unseen data, [When to Use Machine Learning](#)
- unstructured data, [Structured Versus Unstructured Data-Structured Versus Unstructured Data](#)
- updates, deployment myths, [Myth 3: You Won't Need to Update Your Models as Much](#)
- use cases, [Machine Learning Use Cases-Machine Learning Use Cases](#)
- user experience, [User Experience](#)
 - consistency, [Ensuring User Experience Consistency](#)
 - predictions, mostly correct, [Combating "Mostly Correct" Predictions-Combating "Mostly Correct" Predictions](#)
 - smooth failing, [Smooth Failing](#)
- user feedback, [Feedback loop length](#)
- user input data, [Data Sources](#)

V

- vCPU (virtual CPU), [Storage and Compute](#)
- vectorization, model optimization, [Model optimization](#)
- versioning, [Versioning-Versioning](#)
 - code versioning, [Versioning](#)

W

- WASM (WebAssembly), [ML in Browsers](#)
- weak supervision, [Weak supervision-Weak supervision](#)
 - Snorkel, [Weak supervision](#)
- weighted sampling, [Weighted Sampling](#)

- word embeddings, [Discrete and Continuous Positional Embeddings](#)
- workflow management, [Data Science Workflow Management](#)
 - Airflow, [Data Science Workflow Management-Data Science Workflow Management](#)
 - Argo, [Data Science Workflow Management-Data Science Workflow Management](#)
 - DAG (directed acyclic graph), [Cron, Schedulers, and Orchestrators](#)
 - Kubeflow, [Data Science Workflow Management](#)
 - Metaflow, [Data Science Workflow Management](#)

X

- XGBoost, [Feature Importance](#)

Z

- zero rule baselines, [Baselines](#)
- zero-shot learning, [Transfer learning](#)

[Support](#) [Sign Out](#)