The Machine Learning Solutions Architect Handbook : Practical Strategies and Best Practices on the ML Lifecycle, System Design, MLOps, and Generative AI /

Ping, David,
author

Monografía

Design, build, and secure scalable machine learning (ML) systems to solve real-world business problems with Python and AWS Purchase of the print or Kindle book includes a free PDF eBook Key Features Solve large-scale ML challenges in the cloud with several open-source and AWS tools and frameworks Apply risk management techniques in the ML life cycle and learn architecture patterns for solutions Understand the challenges and risks of implementing generative AI Book Description David Ping, Head of GenAI and ML Solution Architecture for global industries at AWS, provides expert insights and practical examples to help you become a proficient ML solutions architect, linking technical architecture to business-related skills. You'll learn about ML algorithms, cloud infrastructure, system design, MLOps , and how to apply ML to solve real-world business problems. David explains the generative AI project lifecycle and examines Retrieval Augmented Generation (RAG), an effective architecture pattern for generative AI applications. You'll also learn about open-source technologies, such as Kubernetes/Kubeflow, for building a data science environment and ML pipelines before building an enterprise ML architecture using AWS. As well as ML risk management and the different stages of AI/ML adoption, the biggest new addition to the handbook is the deep exploration of generative AI. By the end of this book , you'll have gained a comprehensive understanding of AI/ML across all key aspects, including business use cases, data science, real-world solution architecture, risk management, and governance. You'll possess the skills to design and construct ML solutions that effectively cater to common use cases and follow established ML architecture patterns, enabling you to excel as a true professional in the field. What you will learn Apply ML methodologies to solve business problems across industries Design a practical enterprise ML platform architecture Gain an understanding of AI risk management frameworks and techniques Build an end-to-end data management architecture using AWS Train large-scale ML models and optimize model inference latency Create a business application using artificial intelligence services and custom models Dive into generative AI with use cases, architecture patterns, and RAG Who this book is for This book is for solutions architects working on ML projects, ML engineers transitioning to ML solution architect roles, and MLOps engineers. Additionally, data scientists and analysts who want to enhance their practical knowledge of ML systems engineering, as well as AI/ML product managers and risk officers who want to gain an understanding of ML solutions and AI risk management, will also find this book useful. A basic knowledge of

Python, AWS, linear algebra, probability, and cloud infrastructure is required before you get started with this handbook

https://rebiunoda.pro.baratznet.cloud:28443/OpacDiscovery/public/catalog/detail/b2FpOmNlbGVicmF0aW9uOmVzLmJhcmF0ei5yZW4vMzU5NTUzNTY

Título: The Machine Learning Solutions Architect Handbook Practical Strategies and Best Practices on the ML Lifecycle, System Design, MLOps, and Generative AI David Ping

Edición: Second edition

Editorial: Birmingham, England Packt Publishing [2024] 2024

Descripción física: 1 online resource (603 pages)

Contenido: Cover -- Copyright -- Contributors -- Table of Contents -- Preface -- Chapter 1: Navigating the ML Life Cycle with ML Solutions Architecture -- ML versus traditional software -- ML life cycle -- Business problem understanding and ML problem framing -- Data understanding and data preparation -- Model training and evaluation -- Model deployment -- Model monitoring -- Business metric tracking -- ML challenges -- ML solutions architecture -- Business understanding and ML transformation -- Identification and verification of ML techniques -- System architecture design and implementation -- ML platform workflow automation -- Security and compliance -- Summary -- Chapter 2: Exploring ML Business Use Cases -- ML use cases in financial services -- Capital market front office -- Sales trading and research -- Investment banking -- Wealth management -- Capital market back office operations -- Net Asset Value review -- Post-trade settlement failure prediction -- Risk management and fraud -- Anti-money laundering -- Trade surveillance -- Credit risk -- Insurance -- Insurance underwriting -- Insurance claim management -- ML use cases in media and entertainment -- Content development and production -- Content management and discovery -- Content distribution and customer engagement -- ML use cases in healthcare and life sciences -- Medical imaging analysis -- Drug discovery -- Healthcare data management -- ML use cases in manufacturing -- Engineering and product design -- Manufacturing operations - product quality and yield -- Manufacturing operations - machine maintenance -- ML use cases in retail -- Product search and discovery -- Targeted marketing -- Sentiment analysis -- Product demand forecasting -- ML use cases in the automotive industry -- Autonomous vehicles -- Perception and localization -- Decision and planning -- Control Advanced driver assistance systems (ADAS) -- Summary -- Chapter 3: Exploring ML Algorithms -- Technical requirements -- How machines learn -- Overview of ML algorithms -- Consideration for choosing ML algorithms -- Algorithms for classification and regression problems -- Linear regression algorithms -- Logistic regression algorithms -- Decision tree algorithms -- Random forest algorithm -- Gradient boosting machine and XGBoost algorithms -- K-nearest neighbor algorithm -- Multi-layer perceptron (MLP) networks -- Algorithms for clustering -- Algorithms for time series analysis -- ARIMA algorithm -- DeepAR algorithm -- Algorithms for recommendation -- Collaborative filtering algorithm -- Multi-armed bandit/contextual bandit algorithm -- Algorithms for computer vision problems -- Convolutional neural networks -- ResNet -- Algorithms for natural language processing (NLP) problems -- Word2Vec -- BERT -- Generative AI algorithms -- Generative adversarial network -- Generative pre-trained transformer (GPT) -- Large Language Model -- Diffusion model -- Hands-on exercise -- Problem statement -- Dataset description -- Setting up a Jupyter Notebook environment -- Running the exercise -- Summary -- Chapter 4: Data Management for ML -- Technical requirements -- Data management considerations for ML -- Data management architecture for ML -- Data storage and management -- AWS Lake Formation -- Data ingestion -- Kinesis Firehose -- AWS Glue -- AWS Lambda -- Data cataloging -- AWS Glue Data Catalog -- Custom data catalog solution -- Data processing -- ML data versioning -- S3 partitions -- Versioned S3 buckets -- Purpose-built data version tools -- ML feature stores -- Data serving for client consumption -- Consumption via API -- Consumption via data copy -- Special databases for ML -- Vector databases -- Graph databases -- Data pipelines Authentication and authorization -- Data governance -- Data lineage -- Other data governance measures -- Hands-on exercise - data management for ML -- Creating a data lake using Lake Formation -- Creating a data ingestion pipeline -- Creating a Glue Data Catalog -- Discovering and querying data in the data lake -- Creating an Amazon Glue ETL job to process data for ML -- Building a data pipeline using Glue workflows -- Summary -- Chapter 5: Exploring Open-Source ML Libraries -- Technical requirements -- Core features of open-source ML libraries -- Understanding the scikit-learn ML library -- Installing scikit-learn -- Core components of scikit-learn --

Understanding the Apache Spark ML library -- Installing Spark ML -- Core components of the Spark ML library -- Understanding the TensorFlow deep learning library -- Installing TensorFlow -- Core components of TensorFlow -- Hands-on exercise - training a TensorFlow model -- Understanding the PyTorch deep learning library -- Installing PyTorch -- Core components of PyTorch -- Hands-on exercise - building and training a PyTorch model -- How to choose between TensorFlow and PyTorch -- Summary -- Chapter 6: Kubernetes Container Orchestration Infrastructure Management -- Technical requirements -- Introduction to containers -- Overview of Kubernetes and its core concepts -- Namespaces -- Pods -- Deployment -- Kubernetes Job -- Kubernetes custom resources and operators -- Services -- Networking on Kubernetes -- Security and access management -- API authentication and authorization -- Hands-on - creating a Kubernetes infrastructure on AWS -- Problem statement -- Lab instruction -- Summary -- Chapter 7: Open-Source ML Platforms -- Core components of an ML platform -- Open-source technologies for building ML platforms -- Implementing a data science environment -- Building a model training environment Registering models with a model registry -- Serving models using model serving services -- The Gunicorn and Flask inference engine -- The TensorFlow Serving framework -- The TorchServe serving framework -- KFServing framework -- Seldon Core -- Triton Inference Server -- Monitoring models in production -- Managing ML features -- Automating ML pipeline workflows -- Apache Airflow -- Kubeflow Pipelines -- Designing an end-to-end ML platform -- ML platform-based strategy -- ML component-based strategy -- Summary -- Chapter 8: Building a Data Science Environment Using AWS ML Services -- Technical requirements -- SageMaker overview -- Data science environment architecture using SageMaker -- Onboarding SageMaker users -- Launching Studio applications -- Preparing data -- Preparing data interactively with SageMaker Data Wrangler -- Preparing data at scale interactively -- Processing data as separate jobs -- Creating, storing, and sharing features -- Training ML models -- Tuning ML models -- Deploying ML models for testing -- Best practices for building a data science environment -- Hands-on exercise - building a data science environment using AWS services -- Problem statement -- Dataset description -- Lab instructions -- Setting up SageMaker Studio -- Launching a JupyterLab notebook -- Training the BERT model in the Jupyter notebook -- Training the BERT model with the SageMaker Training service -- Deploying the model -- Building ML models with SageMaker Canvas -- Summary -- Chapter 9: Designing an Enterprise ML Architecture with AWS ML Services -- Technical requirements -- Key considerations for ML platforms -- The personas of ML platforms and their requirements -- ML platform builders -- Platform users and operators -- Common workflow of an ML initiative -- Platform requirements for the different personas -- Key requirements for an enterprise ML platform Enterprise ML architecture pattern overview -- Model training environment -- Model training engine using SageMaker -- Automation support -- Model training life cycle management -- Model hosting environment -- Inference engines -- Authentication and security control -- Monitoring and logging -- Adopting MLOps for ML workflows -- Components of the MLOps architecture -- Monitoring and logging -- Model training monitoring -- Model endpoint monitoring -- ML pipeline monitoring -- Service provisioning management -- Best practices in building and operating an ML platform -- ML platform project execution best practices -- ML platform design and implementation best practices -- Platform use and operations best practices -- Summary -- Chapter 10: Advanced ML Engineering -- Technical requirements -- Training large-scale models with distributed training -- Distributed model training using data parallelism -- Parameter server overview -- AllReduce overview -- Distributed model training using model parallelism -- Naïve model parallelism overview -- Tensor parallelism/tensor slicing overview -- Implementing model-parallel training -- Achieving low-latency model inference -- How model inference works and opportunities for optimization -- Hardware acceleration -- Central processing units (CPUs) -- Graphics processing units (GPUs) -- Application-specific integrated circuit -- Model optimization -- Quantization -- Pruning (also known as sparsity) -- Graph and operator optimization -- Graph optimization -- Operator optimization -- Model compilers -- TensorFlow XLA -- PyTorch Glow -- Apache TVM -- Amazon SageMaker Neo -- Inference engine optimization -- Inference batching -- Enabling parallel serving sessions -- Picking a communication protocol -- Inference in large language models -- Text Generation Inference (TGI) -- DeepSpeed-Inference -- FastTransformer Hands-on lab - running distributed model training with PyTorch

ISBN: 1-80512-482-X

Materia: Machine learning

Enlace a formato físico adicional: 1-80512-250-9

Baratz Innovación Documental

- Gran Vía, 59 28013 Madrid
- (+34) 91 456 03 60
- informa@baratz.es