1. How would you define machine learning?
    A. Giving the computer the ability to learn.
    B. Study of algorithms that improve their performance on tasks with experience
2. Can you name four types of applications where it shines?
    A. Anywhere where the traditional algorithms are complex. Problems where the solution requires a lot of maintenance. Problems which require extracting insights with lots of data. Fluctuating environments
    B. Examples: spam filters, tumor detections, summarizing data, forecasting prices
3. What is a labeled training set?
    A. A labeled training set is a set with the known predictions. Where the parameters are paired with the target variable(variable expected to predict)
4. What are the two most common supervised tasks?
    A. Classification (spam or ham), regression(price prediction)
5. Can you name four common unsupervised tasks?
    A. Clustering, Dimensionality Reduction, Association Learning, Anomaly Detection.
6. What type of algorithm would you use to allow a robot to walk in various unknown terrains?
    A. Reinforcement Learning, where an agent with access to the environment will use policy to change its state and learn with positive or negative reinforcement.
7. What type of algorithm would you use to segment your customers into multiple groups?
    A. Clustering Algorithm or Classification depending on whether the data is labeled or not.
8. Would you frame the problem of spam detection as a supervised learning problem or an unsupervised learning problem?

A. I would frame it as a supervised learning problem because the spam detection needs labels and we use classification algorithms to classify them.

9. What is an online learning system?
    A. Incremental Learning with data arriving sequentially or sometimes in mini batches
10. What is out-of-core learning?
    A. Online learning can be used to train large sets of data which cannot fit in machines memory. We use parts of data and train the model iteratively until the data is exhausted which is called out-of-core learning.
11. What type of algorithm relies on a similarity measure to make predictions?
    A. Instance based learning.
12. What is the difference between a model parameter and a model hyperparameter?
    A. Model parameter is the attribute that is learnt while training, while hyperparameters are used to set by the practitioner before the training step.
13. What do model-based algorithms search for? What is the most common strategy they use to succeed? How do they make predictions?
    A. Model based algorithms search for the best set of optimal internal parameters that best represent the data. They essentially find the best mathematical representation of the relationship between the input data and output data.
14. Can you name four of the main challenges in machine learning?
    A. Less data, non-representative data, Poor quality data, irrelevant features, overfitting, underfitting and deployment issues
15. If your model performs great on the training data but generalizes poorly to new instances, what is happening? Can you name three possible solutions?
    A. The model is probably doing that because of overfitting. The three possible solutions are regularization, Hyperparameter training and increasing size and quality of data.

16. What is a test set, and why would you want to use it?
    A. The only way to know if the model is generalizing well to new data without a test set is to deploy the model and see how the model performs to the new data, which is a bad idea. We split the data into two sets: train and test set where we train the model with the train set and see how the model performs with the test set.
17. What is the purpose of a validation set?
    A. If we have to select a hyperparameter with training 100 different models and with generalizing and selecting the best hyperparameter for one test set the model most likely doesn't perform well for new instances because the model is generalized for one test set so we use a hold out validation set which helps us select hyperparameters. More specifically, you train multiple models with various hyperparameters on the reduced training set and you select the model which performs best for the validation set and then you train the model with both training and validation set and evaluate the model with the training set.
18. What is the train-dev set, when do you need it, and how do you use it?
    A. If we hold some of the data points in training set (which is train-dev set), and train the model exclusively on training set not train-dev set if the model performs poorly on train-dev set we know that the model overfit, if the model performs well of train set and train-dev set but poorly on test set we know that there is a data mismatch.
19. What can go wrong if you tune hyperparameters using the test set?
    A. If we use a test set for choosing hyperparameters, let's say we have to choose between 100 combinations of hyperparameters, we are choosing the hyperparameter which performs well on one particular test set. Which more likely doesn't perform as well for new instances.You are choosing the hyperparameters that perform *best* specifically on that test set. The model is therefore indirectly

optimizing its structure and learning process to fit the peculiarities of the test set data.

Preface

- **2006 Breakthrough**: Geoffrey Hinton et al. published a paper on training deep neural networks for handwritten digit recognition, marking the birth of "deep learning."
- **Model Structure**: A deep neural network mimics the cerebral cortex, consisting of layers of artificial neurons.
- **Revival of Interest**: After being thought impossible, deep learning drew renewed interest, leading to significant advancements in machine learning.
- **Industry Impact**: By 2016-2023, machine learning began transforming industries, from web ranking to AI assistants like ChatGPT and others.

Machine Learning in Your Projects

- **Excitement for Learning**: Enthusiasm to implement machine learning (ML) in projects.
- **Potential Uses**:
    - Customer segmentation for marketing.
    - Product recommendations based on similar purchases.
    - Fraud detection in transactions.

    - Revenue forecasting.
    - Chatbot development.

Objective and Approach

- **Target Audience**: Designed for beginners with little to no prior knowledge of machine learning.
- **Content Overview**:
    - Techniques from linear regression to advanced deep learning.
    - Use of Python and popular libraries:
        - **Scikit-Learn**: Easy entry point for various ML algorithms.
        - **PyTorch**: Powerful library for deep learning.
        - Other libraries used:
            - **XGBoost**: For gradient boosting.
            - **Hugging Face**: For datasets and pretrained transformer models.
            - **Gymnasium**: For reinforcement learning.
- **Hands-on Learning**: Emphasis on practical examples alongside theory.

Tip

- Experiment with code examples for better understanding.

Code Examples

- **Repository**: Code examples available on GitHub as Jupyter notebooks.
- **Google Colab**: Recommended platform for running notebooks without installation.

Note

- Assumes usage of Google Colab, but works on other platforms too. Installation instructions available online.

Prerequisites

- **Python Knowledge**: Basic experience required.
- **Familiarity with Libraries**: NumPy, pandas, and Matplotlib.
- **Mathematics**: Basic linear algebra and some calculus beneficial. Tutorials available for additional help.

Roadmap

- **Part I: Fundamentals of Machine Learning**:
    - Overview of ML, problem-solving, project steps, model fitting, cost function minimization, data handling, feature selection, model tuning, challenges like overfitting, and common algorithms.
- **Part II: Neural Networks and Deep Learning**:
    - Understanding neural nets, building deep learning models, important architectures, reinforcement learning, and data preprocessing techniques.

Caution

- Master fundamentals before diving into deep learning techniques.

Changes Between the TensorFlow and PyTorch Versions

- **Library Transition**: Shift from TensorFlow to PyTorch due to its growing dominance.
- **Main Changes**:
    - Updated code for recent library versions.
    - Introduction of new chapters on transformers and advanced techniques.
    - Migration of content from TensorFlow to PyTorch.

Other Resources

- **Courses and Tutorials**:
    - **Andrew Ng's ML course**: Highly recommended for comprehensive ML learning.
- **Recommended Books**:
    - Joel Grus's Data Science from Scratch.
    - Stephen Marsland's Machine Learning: An Algorithmic Perspective.
    - François Chollet's Deep Learning with Python.
- **Online Learning**: Engage in competitions on platforms like Kaggle to practice skills.

Conventions Used in This Book

- **Italic**: New terms, URLs, file names.
- **Constant Width**: Code listings and program elements.
- **Constant Width Bold**: Commands to be typed literally.

- **Constant Width Italic**: User-supplied values or context-specific values.

Tip

- Offers suggestions for better practices.

Note

- General notes or insights.

Warning

- Indicates precautions.

O’Reilly Online Learning

Note

- O’Reilly Media provides extensive resources for training and knowledge sharing.

How to Contact Us

- Contact for comments/questions regarding the book along with publisher address and contact numbers.

Acknowledgments

- Gratitude expressed to readers, reviewers, and family for support during the writing process.

Part I. The Fundamentals of Machine Learning

Chapter 1. The Machine Learning Landscape

- **Definition**: Machine learning is about programming computers to learn from data.
- **Evolution**: From early applications like spam filters to present-day AI assistants.

What Is Machine Learning?

- **General Definitions**:
    - Study of algorithms that improve performance on tasks with experience.

Why Use Machine Learning?

- **Advantages Over Traditional Programming**:
    - Automatic learning and adaptation to new patterns, reducing maintenance effort.

Examples of Applications

- **Common Tasks**:
    - Image classification, tumor detection, text classification, chatbot creation, fraud detection, and customer segmentation.

Types of Machine Learning Systems

- **Categories**:
    - Supervised, unsupervised, semi-supervised, self-supervised, and reinforcement learning.

Training Supervision

Supervised Learning

- **Definition**: Training with labeled data to predict outcomes.

Unsupervised Learning

- **Definition**: Training with unlabeled data to find patterns or groupings.

Semi-supervised Learning

- Combines labeled and unlabeled data for better performance.

Self-supervised Learning

- Generates labels from unlabeled data for training.

Reinforcement Learning

- Agents learn optimal actions through trial and error based on rewards.

Batch Versus Online Learning

Batch Learning

- **Definition**: Training on the entire dataset at once.

Online Learning

- **Definition**: Incremental training with data arriving sequentially.

Instance-Based Versus Model-Based Learning

Instance-Based Learning

- Learns by memorizing training instances and using similarity measures.

Model-Based Learning

- Builds predictive models from training data to generalize.

Main Challenges of Machine Learning

Insufficient Quantity of Training Data

- Requires sufficient examples for training.

Nonrepresentative Training Data

- Training data must accurately reflect the target data.

Poor-Quality Data

- Errors and noise in data can hinder performance.

Irrelevant Features

- Feature selection and engineering are crucial for success.

Overfitting and Underfitting

- **Overfitting**: Model performs well on training data but poorly on unseen data.

- **Underfitting**: Model is too simple to capture underlying data patterns.

Deployment Issues

- Challenges during the model deployment phase.

Testing and Validating

Generalization Error

- Measure of performance on unseen data evaluated with a test set.

Hyperparameter Tuning and Model Selection

- Use validation sets for model evaluation before testing with the test set.

Data Mismatch

- Train and test data distributions must align for effective evaluation.

No Free Lunch Theorem

- No single model outperforms all others across all tasks.

Exercises

- Questions to ensure understanding of concepts covered in the chapter.