Review of Machine Learning

FAKHRELDIN SAEED

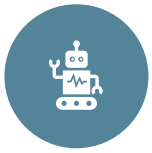
1

Contents

Branches of machine learning.

Evaluating machine-learning models.

Data preprocessing, feature engineering, and feature learning.



Workflow of machine learning.

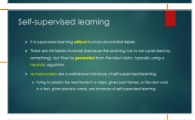
2

Branches of machine learning.

- Supervised learning

- Unsupervised learning

- Self-supervised learning



- Reinforcement learning

3

Supervised learning

- It consists of learning to **map** **input data to known targets**, given a set of examples (often annotated by humans).
- Although supervised learning mostly consists of **classification** and **regression**, there are more exotic variants as well, including the following :
  - Sequence generation—Given a picture, predict a caption describing it.
  - Syntax tree prediction—Given a sentence, predict its decomposition into a syntax tree.
  - Object detection—Given a picture, draw a bounding box around certain objects inside the picture.
  - Image segmentation—Given a picture, draw a pixel-level mask on a specific object.
- **algorithms:**
  - Linear or Logistic regression, Support Vector Machines (SVMs), Naïve Bayes, K-Nearest Neighbors (KNN)

4

Unsupervised learning

- It consists of finding interesting transformations of the input data without the help of any targets.
- for the purposes of data visualization, data compression, or data denoising, or to better understand the correlations present in the data at hand.
- It is the bread and butter of **data analytics**, and it's often a **necessary step** in better understanding a dataset before attempting to solve a supervised-learning problem.
- **Dimensionality reduction** and **clustering** are well-known categories of unsupervised learning.

5

Self-supervised learning

- It is supervised learning **without** human-annotated labels.
- There are still labels involved (because the learning has to be supervised by something), but they're **generated** from the input data, typically using a heuristic algorithm.
- autoencoders are a well-known instance of self-supervised learning.
  - Trying to predict the next frame in a video, given past frames, or the next word in a text, given previous words, are instances of self-supervised learning .

6

Reinforcement learning

- In reinforcement learning, an **agent** receives information about its **environment** and learns to choose actions that will maximize some **reward**.
- For instance, a neural network that "looks" at a **videogame** screen and outputs game actions in order to maximize its score can be trained via reinforcement learning.

7

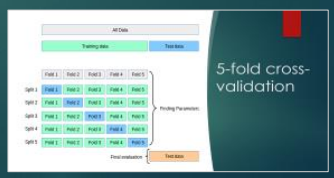
Evaluating machine-learning models (1/3)

- In machine learning, the goal is to achieve models that **generalize** (that perform well on never-before-seen data) and **overfitting** is the central obstacle.
- There are strategies for mitigating overfitting and maximizing generalization.
- **Regularization** : any method to prevent overfitting (simplicity, sparsity, dropout, early stopping) other than adding more data.

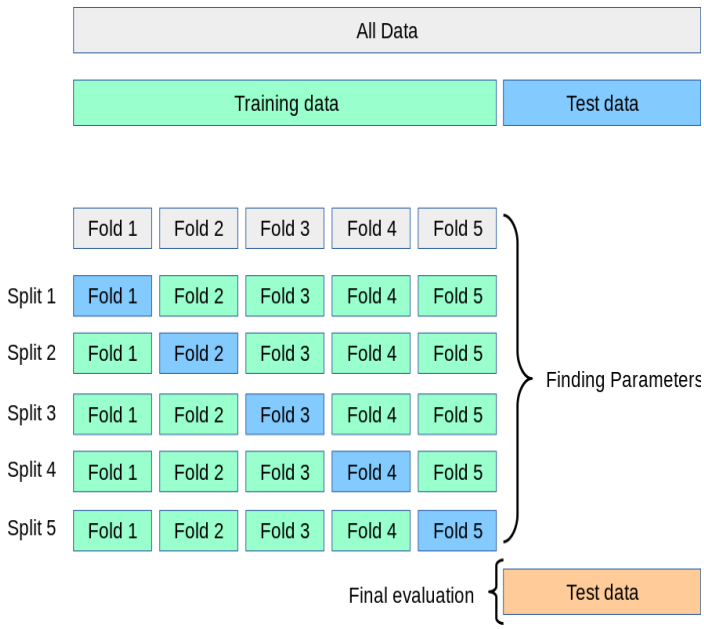
8

Evaluating machine-learning models (2/3)

- Evaluating a model always boils down to splitting the available data into three sets: **training**, **validation**, and **test**.
- You train on the training data and evaluate your model on the validation data. Once your model is ready for prime time, you test it one final time on the test data.
- Three classic evaluation recipes are simple **hold-out validation**, **K-fold** validation, and **iterated K-fold** validation with **shuffling**.



9



5-fold cross-validation

10

Evaluating machine-learning models (3/3)

- Keep an eye out for the following when you're choosing an evaluation protocol:
  - Data representativeness (if the samples are **ordered** by their class)
    - Both the training set and the test set should be representative of the data at hand.
    - Randomly shuffle the data before splitting it into training and test sets.
  - The arrow of time
    - If you're trying to predict the future given the past you should not randomly shuffle your data before splitting it.
  - Redundancy in your data
    - Make sure your training set and validation set are disjoint.

11

Data pre-processing and feature engineering

- **Data pre-processing** aims at making the raw data at hand more amenable to ML model. This includes vectorization, normalization, handling missing values, and feature extraction.
- **Feature engineering** is the process of using your own knowledge about the data and about the machine-learning algorithm at hand to make the algorithm work better by applying hardcoded (nonlearned) transformations to the data before it goes into the model.

12

The universal workflow of machine learning

1. **Defining the problem and assembling a dataset**
   - What will your input data be?
   - What are you trying to predict?
   - problem are you facing?
     - Is it binary classification? Multiclass classification? Scalar regression? Vector regression? Multiclass, multilabel classification? Something else, like clustering, generation, or reinforcement learning?

13

The universal workflow of machine learning

2. **Choosing a measure of success**
   - To achieve success, you must define what you mean by success—accuracy? Precision and recall? Customer-retention rate?
   - For balanced-classification problems, where every class is equally likely, accuracy and (ROC AUC) are common metrics.
   - For class-imbalanced problems, you can use precision and recall.
3. **Deciding on an evaluation protocol**
   - Once you know what you're aiming for, you must establish how you'll measure your current progress.
   - hold-out validation set , K-fold cross-validation or iterated K-fold validation

14

The universal workflow of machine learning

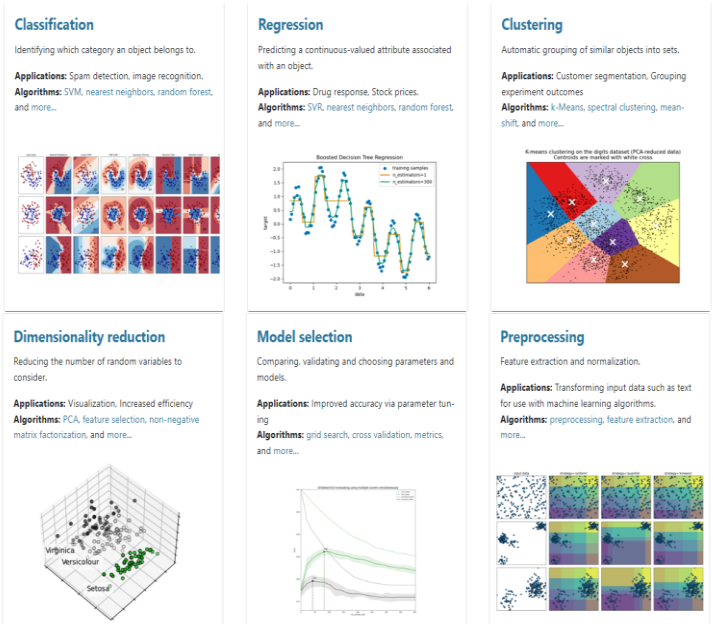
4. **Preparing your data**
   - We should format your data in a way that can be fed into a machine-learning model (formatted as tensors)
5. **Developing a model that does better than a baseline**
   - Our goal at this stage is to achieve statistical power: that is, to develop a small model that is capable of beating a dumb baseline.

15

The universal workflow of machine learning

6. **Scaling up: developing a model that overfits**
   - Once we've obtained a model that has statistical power, the question becomes, is our model sufficiently powerful?
   - the universal tension in machine learning is between optimization and generalization; the ideal model is one that stands right at the border between underfitting and overfitting; between undercapacity and overcapacity.
7. **Regularizing your model and tuning your hyperparameters**
   - We'll repeatedly modify our model, train it, evaluate on our validation data (not the test data, at this point), modify it again, and repeat, until the model is as good as it can get.

16



scikit learn

scikit-learn

Machine Learning in Python

scikit-learn is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

17

Pipelines

- A pipeline in machine learning is a technical infrastructure that allows an organization to organize and automate machine learning operations.
- Stages of building an end-to-end pipeline:
  - Data Ingestion
  - Data Processing
  - Data Splitting
  - Model Training
  - Model Evaluation
  - Model Deployment
  - Monitoring Model Performance

```python
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import make_pipeline
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# create a pipeline object
pipe = make_pipeline(
    StandardScaler(),
    LogisticRegression()
)
# load the iris dataset and split it into train and test sets
X, y = load_iris(return_X_y=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

# fit the whole pipeline
pipe.fit(X_train, y_train)

# we can now use it like any other estimator
accuracy_score(pipe.predict(X_test), y_test)
```

18