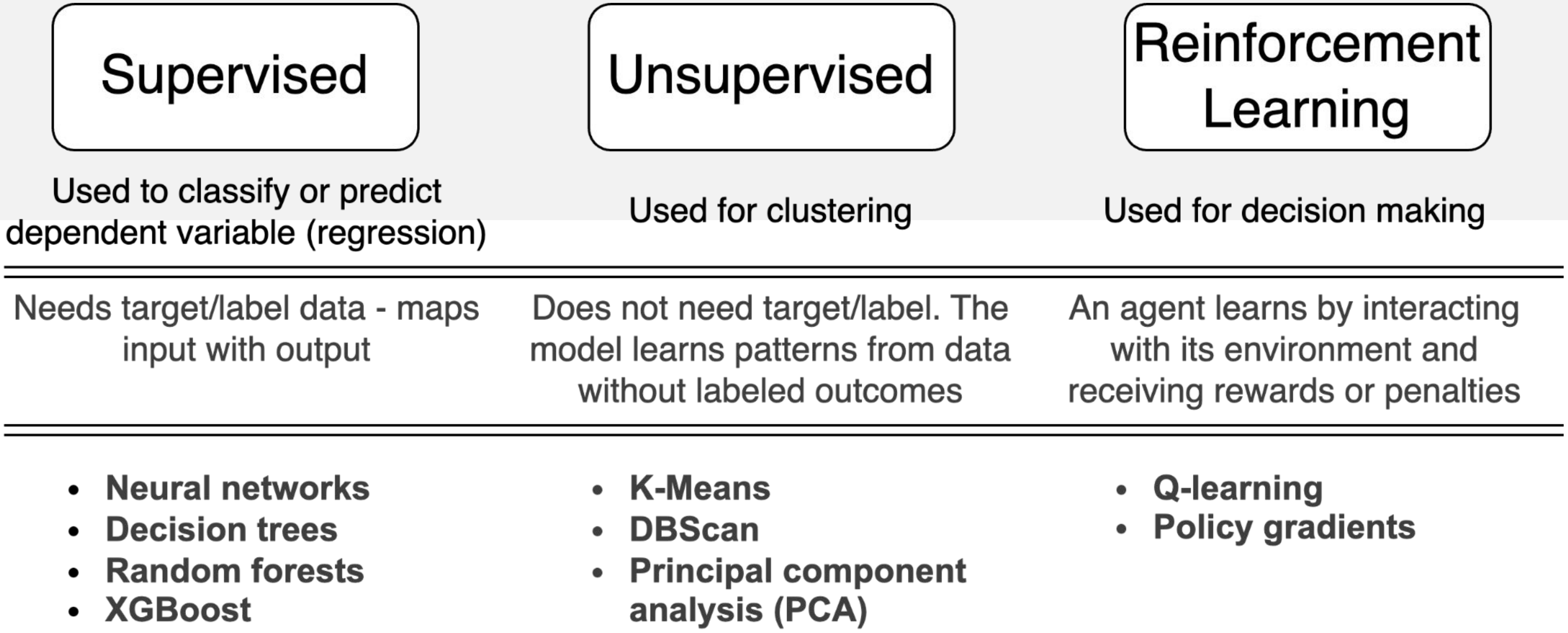
Machine learning algorithms

Prepared by Rytis Augustauskas

rytis.augustauskas@ktu.lt

Kaunas, 2024

Explore a world of ML algorithms



Supervised	Unsupervised	Reinforcement Learning
Used to classify or predict dependent variable (regression)	Used for clustering	Used for decision making
Needs target/label data - maps input with output	Does not need target/label. The model learns patterns from data without labeled outcomes	An agent learns by interacting with its environment and receiving rewards or penalties
• Neural networks **• Decision trees** **• Random forests** **• XGBoost**	**• K-Means** **• DBScan** **• Principal component analysis (PCA)**	**• Q-learning** **• Policy gradients**

Neural Networks

Neural networks are a class of machine learning algorithms inspired by the structure and function of the human brain. They consist of layers of interconnected nodes (neurons) that process information. Neural networks can be used for a wide range of tasks, including classification, regression, and generative modeling.

Sample Use Cases:

- **Image Recognition: Identifying objects in images (e.g., recognizing cats and dogs in photos).**
- **Natural Language Processing (NLP): Sentiment analysis, language translation, and text generation.**
- **Time Series Forecasting: Predicting stock prices or weather conditions based on historical data.**
- **Features interpretation: From sensors data, such as, temperature, movement speed, vibrations predict how likely system will fails to operate.**

Supervised

Decision Trees

Decision trees are a type of supervised learning algorithm that uses a tree-like model of decisions and their possible consequences. Each internal node represents a "test" on an attribute (e.g., is the temperature high?), each branch represents the outcome of the test, and each leaf node represents a decision (e.g., classify as "yes" or "no").

Sample Use Cases:

- **Customer Churn Prediction: Predicting whether a customer will leave a service based on their usage patterns and demographic data.**
- **Medical Diagnosis: Diagnosing diseases based on patient symptoms and medical history.**
- **Credit Risk Assessment: Determining the likelihood of a loan default based on financial and personal data.**

Supervised

Random Forests

Random forests are an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees. This approach helps to reduce overfitting and improve the robustness of predictions.

Sample Use Cases:

- **Fraud Detection: Identifying fraudulent transactions in credit card data.**
- **Stock Market Analysis: Predicting stock price movements based on various economic indicators.**
- **Product Recommendation: Recommending products to users based on their purchase history and preferences.**

Supervised

XGBoost

XGBoost (Extreme Gradient Boosting) is an optimized gradient boosting library designed to be highly efficient, flexible, and portable. It builds an ensemble of weak prediction models, typically decision trees, and combines their predictions to form a strong predictor. XGBoost is known for its performance and speed, making it a popular choice in machine learning competitions.

Sample Use Cases:

- **Abnormal Internet Traffic Prediction: From parameters, such as, upload and download speed, connection IP origins, incoming requests, predict if the internet traffic experiences abnormalities.**
- **Ranking and Search: Improving search engine results by ranking web pages based on relevance.**
- **Bioinformatics: Predicting protein interactions and gene functions based on biological data.**

Supervised

Sample

Dataset Overview

This dataset contains sensor data collected from various machines, with the aim of predicting machine failures in advance. It includes a variety of sensor readings as well as the recorded machine failures.

Columns Description

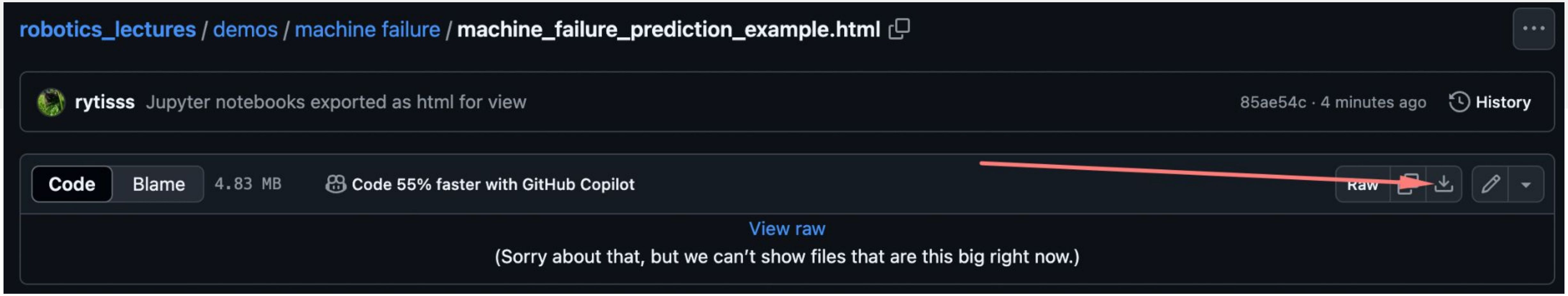
footfall: The number of people or objects passing by the machine.
tempMode: The temperature mode or setting of the machine.
AQ: Air quality index near the machine.
USS: Ultrasonic sensor data, indicating proximity measurements.
CS: Current sensor readings, indicating the electrical current usage of the machine.
VOC: Volatile organic compounds level detected near the machine.
RP: Rotational position or RPM (revolutions per minute) of the machine parts.
IP: Input pressure to the machine.
Temperature: The operating temperature of the machine.
fail: Binary indicator of machine failure (1 for failure, 0 for no failure).

Dataset: https://www.kaggle.com/datasets/umerrtx/machine-failure-prediction-using-sensor-data?resource=download

Supervised

Sample

Rendered Jupyter Notebook with sample is place here. Download it to view in the Internet Browser


robotics_lectures / demos / machine failure / machine_failure_prediction_example.html

rytisss Jupyter notebooks exported as html for view

85ae54c · 4 minutes ago History

Code Blame 4.83 MB Code 55% faster with GitHub Copilot Raw

View raw

(Sorry about that, but we can't show files that are this big right now.)

K-Means Clustering

K-Means is a widely used unsupervised machine learning algorithm for clustering data into K distinct non-overlapping subgroups (clusters). The goal of K-Means is to minimize the within-cluster variance, which is the sum of squared distances between each point in a cluster and the centroid of that cluster.

Sample Use Cases:

- **Customer Segmentation: Grouping customers into segments based on purchasing behavior, demographics, and other attributes.**
- **Document Clustering: Organizing documents into clusters based on their content.**
- **Image Compression: Reducing the color palette of an image to a smaller set of colors.**

DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN is a density-based clustering algorithm that groups together points that are closely packed together, marking points that lie alone in low-density regions as outliers. Unlike K-Means, DBSCAN does not require the number of clusters to be specified in advance and can discover clusters of arbitrary shape.

Sample Use Cases:

- **Geospatial Data Analysis: Clustering geographical data points to identify hotspots or regions of interest.**
- **Medical Imaging: Segmenting medical images to identify regions of interest, such as tumors or lesions.**
- **Market Basket Analysis: Grouping items that are frequently purchased together.**

Unsupervised

Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that transforms a set of possibly correlated variables into a set of linearly uncorrelated variables called principal components. The first principal component captures the most variance in the data, the second captures the second most variance, and so on.

Sample Use Cases:

- **Financial Portfolio Management: Reducing the complexity of financial data to identify the most significant factors affecting portfolio performance.**
- **Sensor Data Analysis: Reducing the dimensionality of sensor data to identify key features and patterns.**
- **Object contour in image analysis: Can show direction along which direction contours are distributed.**

Sample

Supervised

About this Dataset

This case requires to develop a customer segmentation to define marketing strategy. The sample Dataset summarizes the usage behavior of about 9000 active credit card holders during the last 6 months. The file is at a customer level with 18 behavioral variables.

Following is the Data Dictionary for Credit Card dataset:

CUST_ID : Identification of Credit Card holder (Categorical)

BALANCE : Balance amount left in their account to make purchases (

BALANCE_FREQUENCY : How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)

PURCHASES : Amount of purchases made from account

ONEOFF_PURCHASES : Maximum purchase amount done in one-go

INSTALLMENTS_PURCHASES : Amount of purchase done in installment

CASH_ADVANCE : Cash in advance given by the user

PURCHASES_FREQUENCY : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)

ONEOFFPURCHASESFREQUENCY : How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)

PURCHASESINSTALLMENTSFREQUENCY : How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)

CASHADVANCEFREQUENCY : How frequently the cash in advance being paid

CASHADVANCETRX : Number of Transactions made with "Cash in Advanced"

PURCHASES_TRX : Numbe of purchase transactions made

CREDIT_LIMIT : Limit of Credit Card for user

PAYMENTS : Amount of Payment done by user

MINIMUM_PAYMENTS : Minimum amount of payments made by user

PRCFULLPAYMENT : Percent of full payment paid by user

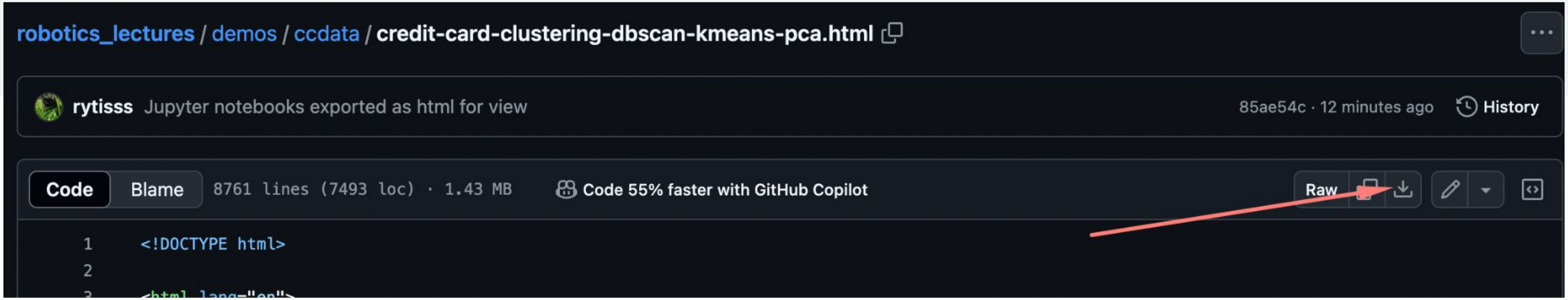
TENURE : Tenure of credit card service for user

Dataset: https://www.kaggle.com/code/salomopolanco/data-mining-clustering

Supervised

Sample

Rendered Jupyter Notebook with sample is place here. Download it to view in the Internet Browser


robotics_lectures / demos / ccdata / credit-card-clustering-dbscan-kmeans-pca.html

rytisss Jupyter notebooks exported as html for view

85ae54c · 12 minutes ago

8761 lines (7493 loc) · 1.43 MB

```
<!DOCTYPE html>

<html lang="en">
```

Q-learning

Q-learning is a type of reinforcement learning (RL) algorithm that allows an agent to learn optimal actions in a given environment by interacting with it. The goal of the agent is to maximize long-term rewards by learning the best actions to take in different states of the environment. Q-learning is a model-free algorithm, meaning it does not require a model of the environment and learns directly from experience.

Sample Use Cases:

- **Game Playing: Teaching an agent to play simple games like Tic-Tac-Toe, chess, or Atari games.**
- **Robotics and Autonomous Control: A robot learning to move through a maze or navigate a room to reach a goal**
- **Resource Management and Optimization: Dynamically managing data center resources (e.g., adjusting server utilization)**

Policy gradients

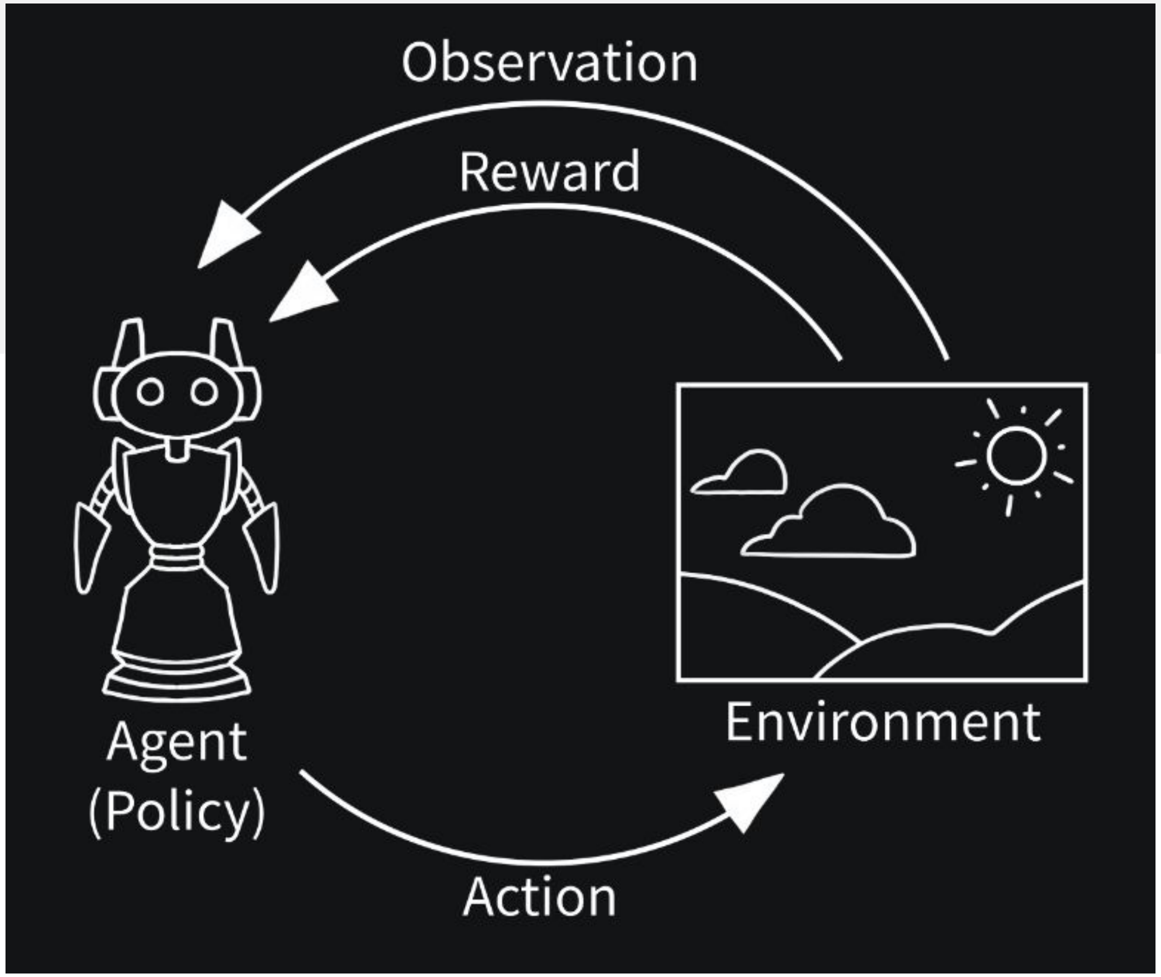
Policy Gradient methods are a class of reinforcement learning (RL) algorithms that directly optimize the policy, which is the strategy an agent uses to decide which actions to take in a given state. Unlike Q-learning, which learns the value of actions indirectly and uses that to derive a policy, policy gradient methods learn the policy directly by maximizing the expected cumulative reward. These methods are often used in environments with continuous action spaces or when it is challenging to compute Q-values (numerical values that represent the reward).

Sample Use Cases:

- **Robotic Control: A robot learning to balance, walk, or manipulate objects.**
- **Autonomous Driving: An autonomous car learning to drive by optimizing its actions (e.g., steering, acceleration, braking) in response to the environment (e.g., other cars, pedestrians, road signs).**

Simplified Reinforcement Learning Concept

In reinforcement learning, the classic "agent-environment loop" pictured below is a simplified representation of how an agent and environment interact with each other. The agent receives an observation about the environment, the agent then selects an action, which the environment uses to determine the reward and the next observation. The cycle then repeats itself until the environment ends (terminates).

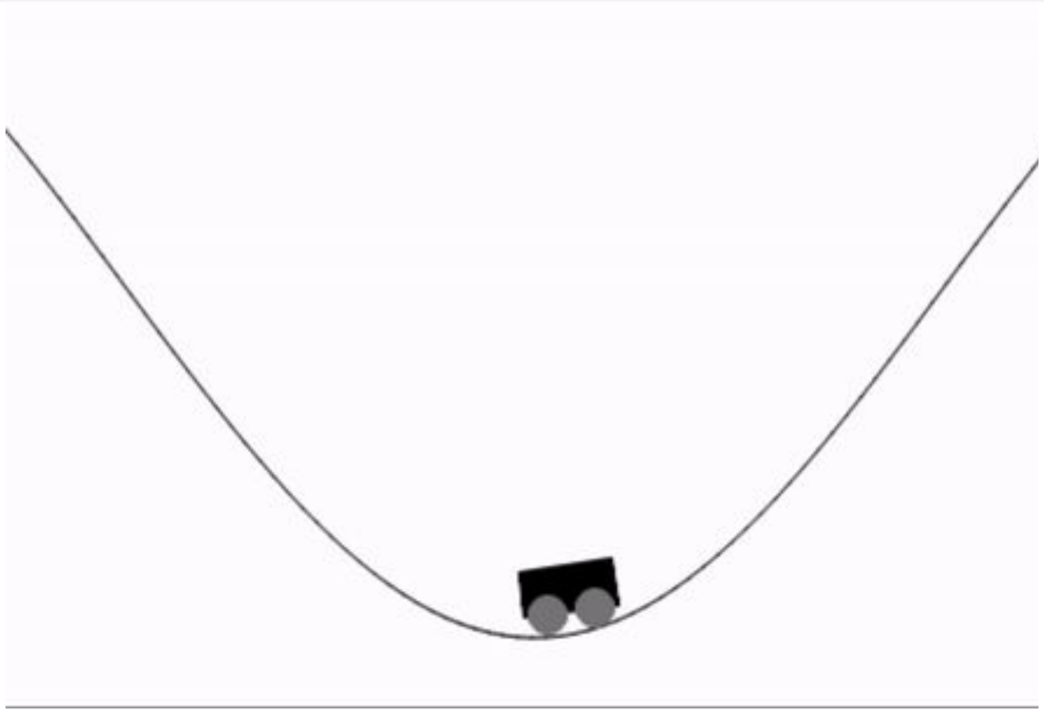


https://gymnasium.farama.org/introduction/basic_usage/

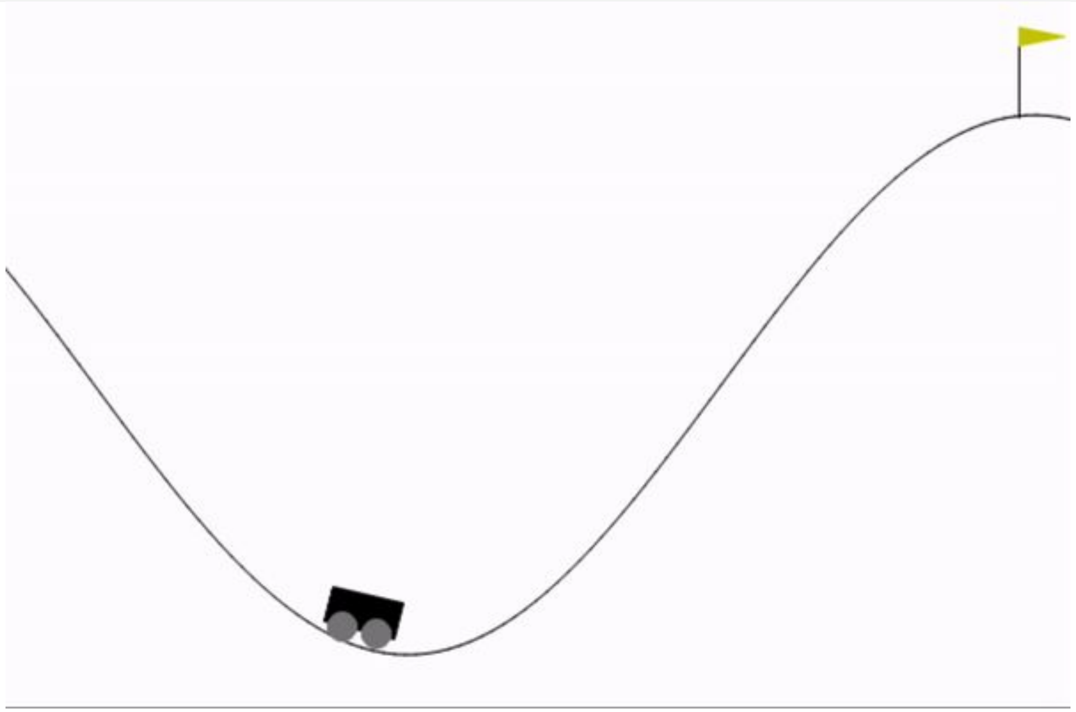
Supervised

Sample

A few sample are given for the OpenAI's Gymnasium environments https://gymnasium.farama.org/ . Q-Learning sample code and videos can be found here



Not trained

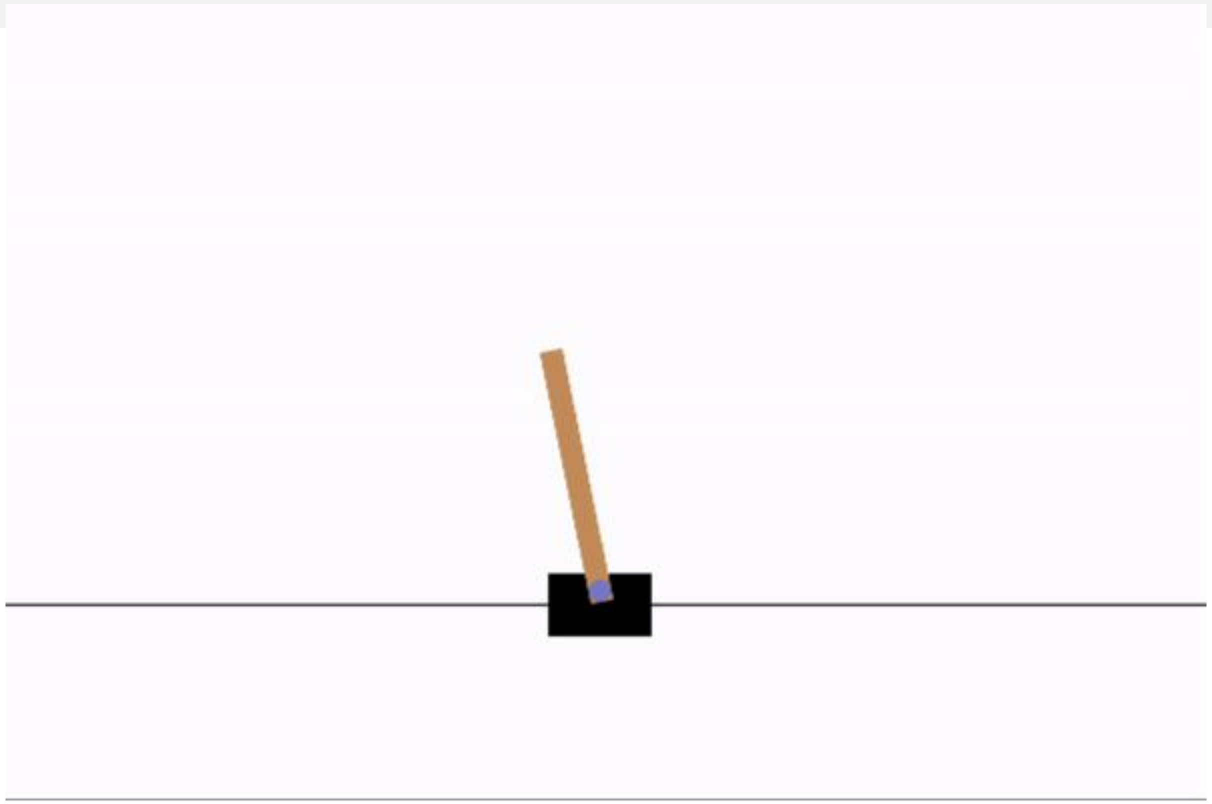


Trained

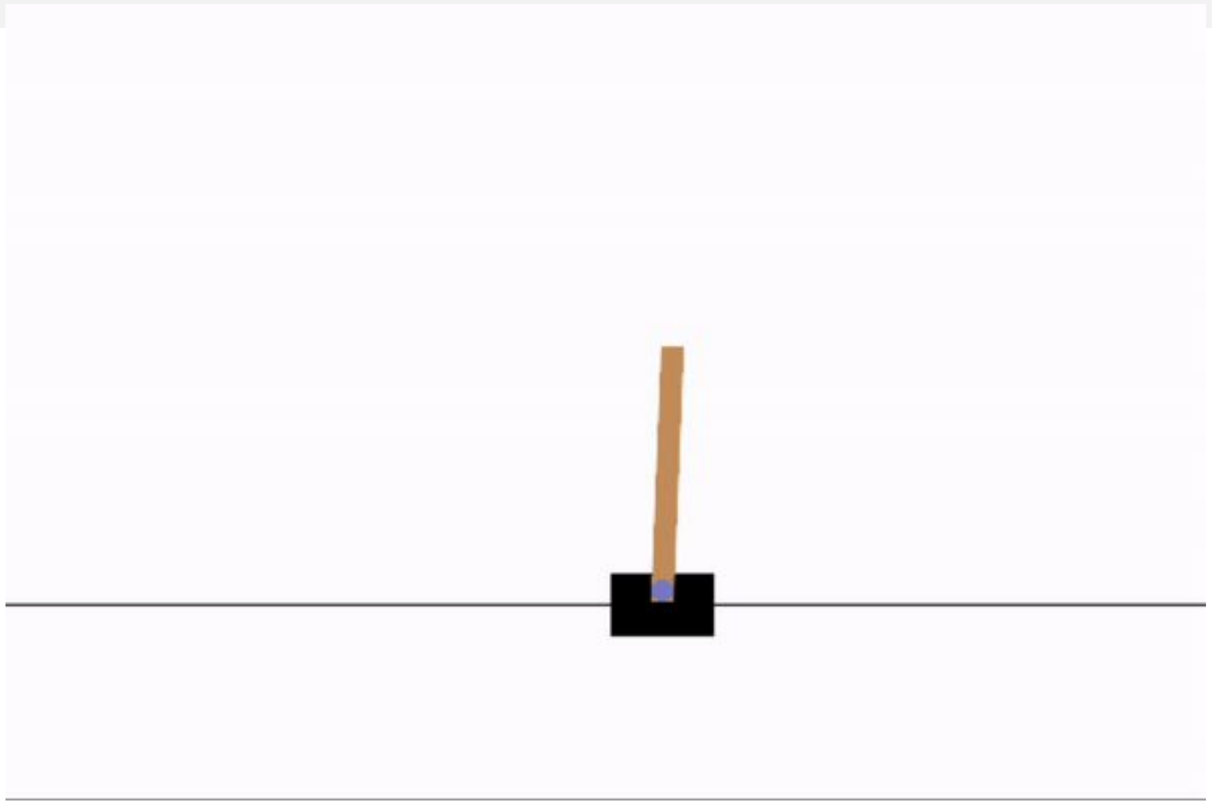
Supervised

Sample

A few sample are given for the OpenAI's Gymnasium environments https://gymnasium.farama.org/ . Q-Learning sample code and videos can be found here



Not trained



Trained

Thank you for your attention! Questions?

email: rytis.augustauskas@ktu.lt