TASK

Machine Learning

Visit our website

Introduction

WELCOME TO THE MACHINE LEARNING TASK!

This task introduces you to the most important concepts in Machine Learning, giving you a general overview of the landscape and preparing you to learn about the intention and theory behind specific supervised and unsupervised learning algorithms.

A note from the

HyperionDev Team

The history of machine learning

For decades, visions of machines that can learn the way humans can have captured the imagination of science-fiction authors and researchers alike. But only in recent years have machine learning programs been developed that can be applied on a wide scale, influencing our daily lives.

Machine learning programs are working behind the scenes to produce and curate our playlists, news feeds, weather reports, and email inboxes. They help us find restaurants, translate documents, and even meet potential dates. From a business perspective, machine learning-based software is becoming central to many industries, generating demand for experts.

In 1959, Arthur Samuel, a pioneer in artificial intelligence and gaming, defined machine learning as the "field of study that gives computers the ability to learn without being explicitly programmed." He is known for developing a program capable of playing checkers. Samuel never programmed exactly which strategies the systems could use. Instead, he devised a way in which the program could learn such strategies through the experience of playing thousands of games.

In the 50s, machines were hard to acquire and not very powerful, so machine learning algorithms were mostly an object of theoretical research. Now that computers are vastly more powerful and more affordable, machine learning has become a very active field of study with a variety of real-world applications.

DEFINING MACHINE LEARNING

At its essence, machine learning (ML) can be defined as a computational methodology focused on deriving insights from data. It enables computers to acquire knowledge from past observations and independently make predictions or decisions without relying on explicit programming instructions. By leveraging data-driven patterns and algorithms, machine learning enables automated systems to adapt and improve their performance over time.

Unveiling the line between machine learning and artificial intelligence

The term “machine learning” is often used interchangeably with the term “artificial intelligence” (AI). While the two are very much related, they are not the same thing. There is much debate about the difference between the two, but a simple way to look at it for our purposes is to see Machine Learning as a *type* of artificial intelligence. Any program that completes a task in a way that can be considered human-like can be considered an example of artificial intelligence, but only programs that solve the task by learning without pre-programming are machine learning programs.

A note from our coding mentor

Ridhaa

Did you know that a London-based company called Google **DeepMind** has developed an artificial intelligence-based gamer, which can play 49 video games from Atari 2600 and achieves better than a professional human player’s top score in 23 of them? Yes, you read that right!

According to an article (link provided below), “The software isn’t told the rules of the game. Instead, it uses an algorithm called a deep neural network to examine the state of the game and figure out which actions produce the highest total score.”

One of the most impressive, and probably the eeriest example, is that in the boxing game, the software learned how to pin its opponent on the ropes (which is something only seasoned players of the game knew how to do), and release a barrage of punches until its opponent was knocked out! Extremely ruthless, right? **Give it a read here**.

INPUT AND OUTPUT

Whatever it is that we want a machine learning algorithm to learn, we first need to express it numerically. The machine-readable version of a task consists of an **input** and an **output**. The input is whatever we want the algorithm to learn from, and the output is the outcome we want the algorithm to be able to produce. An example of an input would be the budget or number of awards a movie receives. An example of output would be the box office sales of that movie.

Since machine learning is a young field that overlaps with several other disciplines, including statistics, the input and output may be referred to by several other names.

For input, these include:

- features (named after the fact that inputs typically 'describe' something),
- independent variables, and
- explanatory variables (because the output is usually assumed to depend on or be explained by the input).

For output, alternate terms are:

- labels,
- predictions,
- dependent variables, and
- response variables.

Once we clearly understand the input-output specifics of machine learning models, we can explore different learning algorithms. While there are various types of learning methodologies, our main focus in this context will be on supervised and unsupervised learning. These two approaches are fundamental in the field of machine learning, as they involve training models using labelled or unlabelled data, respectively.

SUPERVISED LEARNING

In supervised learning problems, a program predicts an output given an input by learning from pairs of inputs and outputs (labels); that is, the program learns from examples that have had the right answers assigned to them beforehand. These assignments are often called **annotations**. Because they are considered the correct answers, they are also called **gold labels**, **gold data**, or the **gold standard**.

The collection of data examples used in supervised learning is called a **training set**. A collection of examples used to assess a program's performance is called a **test set**. Like a student learning in a language course that teaches only through exposure, supervised learning problems see a collection of correct answers to various questions. Then they must learn to provide the correct answers to new but similar questions.

Continuing our exploration, we will delve into two common types of supervised learning: regression and classification, which offer valuable tools for predicting continuous values and categorising data into distinct classes.

Regression

Regression is a prediction task where a program learns to estimate and predict a continuous output value. It does this by analysing pairs of input features and their corresponding outputs in a training set. By analysing the training examples, the program tries to identify patterns and associations that allow it to make precise estimations.

The main objective of regression is to understand the relationship between the input variables and a continuous target variable, enabling the program to make accurate predictions for new inputs that are similar to the training data.

Commonly used metrics to assess the accuracy of a regression model, which allow for comparing different models or evaluating the performance of a single model, are:

- R-squared (R2),
- mean squared error (MSE),
- root mean squared error (RMSE),
- mean absolute error (MAE),
- and mean absolute percentage error (MAPE).

R2 is known as the coefficient of determination, quantifying the proportion of variance in the target variable that can be explained by the features in the model. It ranges from 0 to 1, with higher values indicating a better fit.

MSE measures the average squared difference between predicted and actual values, providing an overall measure of prediction accuracy.

RMSE is the square root of MSE and represents the average magnitude of prediction errors.

Another metric that provides a measure of the average magnitude of errors is **MAE**, which calculates the average absolute difference between predicted and actual values.

Finally, **MAPE** measures the average percentage difference between predicted and actual values, which is particularly useful when the magnitude of errors needs to be assessed relative to the actual values. By default, lower values of MSE, RMSE, MAE, and MAPE indicate better model performance.

Classification

Unlike regression tasks, a classification process assumes a program is trained to categorise input data into predefined classes or categories.

By analysing labelled examples, where each example is already assigned to a specific class, the program learns patterns and relationships between input features and classes. This knowledge allows the program to accurately classify new, unseen data, ensuring they are correctly assigned to their respective classes. The ultimate goal of classification is to develop a model that can make reliable predictions for unknown instances, effectively categorising them into the appropriate classes.

To evaluate the effectiveness and performance of classification models, several commonly used evaluation metrics are available. One such metric is the Gini index, which measures the impurity or disorder in a set of categorical data. The confusion matrix is another valuable tool that summarises the model's performance by displaying counts of true positives, true negatives, false positives, and false negatives. Additionally, **precision** and the **F1 score** can be important metrics too. Precision calculates the proportion of true positive predictions among all positive predictions, indicating the model's ability to minimise false positives. The F1 score, as the harmonic mean of precision and recall, assesses the model's performance by considering both of these metrics. Together the discussed metrics provide a comprehensive evaluation of classification models.

Supervised learning algorithms

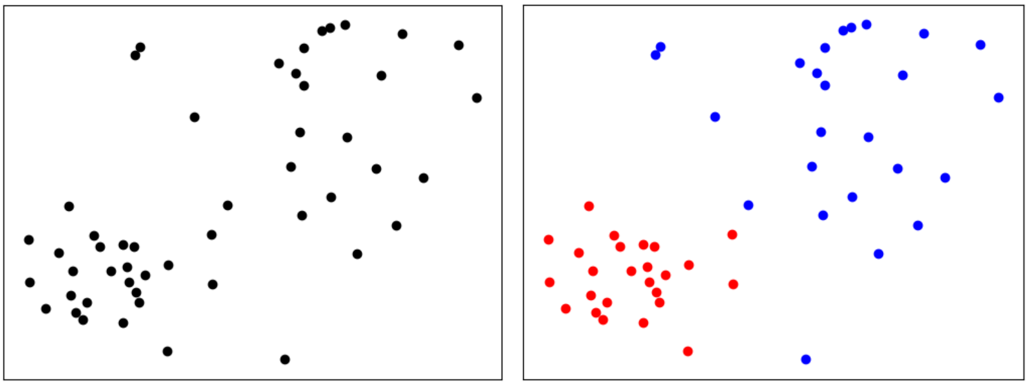
Finally, we offer a list of common supervised learning algorithms and their typical usage:

Supervised learning algorithms	Typical usage	
	Regression	Classification
Linear regression	✔	
Logistic regression		✔
Decision tree	✔	✔
Random forest	✔	✔
Support vector machines (SVM)	✔	✔
Naïve Bayes	✔	
K-nearest-neighbour (KNN)	✔	✔

UNSUPERVISED LEARNING

In unsupervised learning, a program does not learn from labelled data. Instead, it attempts to discover patterns in the data on its own.

For example, suppose you have two classes scattered in a 2-dimensional space (as in the first of the images below) and you want to separate the two data sets (as in the second image on the right-hand side). Unsupervised learning finds underlying patterns in the data, allowing the classes to be separated.



To highlight the difference between supervised and unsupervised learning, consider the following example. Assume that you have collected data describing the heights and weights of people. An unsupervised clustering algorithm might produce groups that correspond to men and women, or children and adults. An example of a supervised learning problem is if we label some of the data with the person’s sex and then try to induce a rule to predict whether a person is male or female based on their height and weight.

Unsupervised learning algorithms

The following algorithms have proven to be highly valuable in practical applications, making them some of the most commonly used methods in unsupervised learning:

- K-means clustering
- Hierarchical clustering
- t-Distributed Stochastic Neighbour Embedding (t-SNE)
- Gaussian Mixture Models (GMM)
- Autoencoders

SEMI-SUPERVISED LEARNING

Semi-supervised learning is an approach that combines labelled and unlabelled data to harness the benefits of both supervised and unsupervised learning. While supervised learning relies on labelled data with known outcomes and unsupervised learning explores unlabelled data to identify patterns, semi-supervised learning uses a smaller set of labelled data to guide the learning process. Simultaneously, it utilises a larger set of unlabelled data containing valuable information.

By leveraging the combined dataset, the algorithm learns from the labelled examples and applies that knowledge to predict outcomes for the unlabelled data, revealing additional patterns and enhancing the model's understanding of the problem's underlying structure. This approach is particularly advantageous when acquiring labelled data is expensive or time-consuming, allowing for optimal resource utilisation and the potential for improved results compared to using just one data type.

Compulsory Task 1

Answer the following in the provided document titled **machine_learning.ipynb**.

1. For each of the following examples describe at least one possible **input and output.** Justify your answers:
   1.1. A self-driving car
   1.2. Netflix recommendation system
   1.3. Signature recognition
   1.4. Medical diagnosis

2. For each of the following case studies, determine whether it is appropriate to utilise **regression** or **classification** machine learning algorithms. Justify your answers:
   2.1. Classifying emails as promotional or social based on their content and metadata.
   2.2. Forecasting the stock price of a company based on historical data and market trends.
   2.3. Sorting images of animals into different species based on their visual features.
   2.4. Predicting the likelihood of a patient having a particular disease based on medical history and diagnostic test results.

3. For each of the following real-world problems, determine whether it is appropriate to utilise a **supervised** or **unsupervised** machine learning algorithm. Justify your answers:
   3.1. Detecting anomalies in a manufacturing process using sensor data without prior knowledge of specific anomaly patterns.
   3.2. Predicting customer lifetime value based on historical transaction data and customer demographics.
   3.3. Segmenting customer demographics based on their purchase history, browsing behaviour, and preferences.
   3.4. Analysing social media posts to categorise them into different themes.

4. For each of the following real-world problems, determine whether it is appropriate or inappropriate to utilise **semi-supervised** machine learning algorithms. Justify your answers:
   4.1. Predicting fraudulent financial transactions using a dataset where most transactions are labelled as fraudulent or legitimate.

4.2. Analysing customer satisfaction surveys where only a small portion of the data is labelled with satisfaction ratings.

4.3. Identifying spam emails in a dataset where the majority of emails are labelled.

4.4. Predicting the probability of default for credit card applicants based on their complete financial and credit-related information.

Rate us

Share your thoughts

HyperionDev strives to provide internationally-excellent course content that helps you achieve your learning outcomes.

Think that the content of this task, or this course as a whole, can be improved, or think we've done a good job?

Click here to share your thoughts anonymously.