Fuzzy Systems and Soft Computing
ISSN : 1819-4362

EXPLORING MACHINE LEARNING ALGORITHMS: FROM THEORY TO PRACTICE.

Dr.K. Geetha, Associate Professor, Sri Krishna Adithya College of Arts and Science College, Coimbatore.
Aruna Sri S, Department of Data Science, Srikrishna Adithya College of Arts and Science College, Coimbatore
Akshayasri S Department of Data Science, Srikrishna Adithya College of Arts and Science College, Coimbatore

ABSTRACT

Machine learning is the most important subset of AI. This subset enables systems to learn from data and make decisions with minimal human intervention. The paper explores the key categories of ML supervised, unsupervised, and reinforcement learning, highlighting their defining characteristics and widely used algorithms. From linear regression and neural networks to ensemble methods like random forests and gradient-boosting machines, this paper offers a structured overview of ML techniques and their applications. Moreover, it discusses the factors that influence the choice of algorithms, emphasizing their importance in building intelligent systems and solving real-world problems.

INTRODUCTION

Today, the world is data-driven. In that world, ML sits right at the top of innovation in technology and its operation and decision-making process within industries. From streaming personal recommendations to life-saving medical diagnostics, ML has emerged as an essential tool for the domains.

At its core, ML is about equipping systems to learn and improve from experience without explicitly programming them. This capacity is derived from the development of complex algorithms that help in analysing data, recognizing patterns, and making predictive insights. These algorithms come under three broad categories of supervised, unsupervised, and reinforcement learning. This allows for a wide range of applications, from simple classification to game strategy optimization.

It is a subfield of artificial intelligence; it is a revolutionary area which lets computers learn from data and make judgments without explicit programming. It is the basis of many of the modern technologies that exist, from voice recognition and fraud detection to recommendation systems and driverless cars. Machine learning algorithms, in simple words, are designed to identify patterns in data and then generate some form of prediction or judgment. Below are some popular machine learning algorithms.

TYPES OF MACHINE LEARNING

I. Supervised Learning
II. Unsupervised Learning
III. Reinforcement Learning

I. SUPERVISED LEARNING

Supervised learning is the category of machine learning, where the model learns to make predictions or decisions by generalizing from labelled data. The training dataset in this method contains input-output pairs wherein input features are mapped with their corresponding outputs. This learning is mainly done with an objective to learn a function that can predict an output for new, unseen data based on the identified patterns in the training data. Usually supervised learning are to use the tasks like classification of works like class is categorical output and in which regression is continuous kind. Supervised learning is broadly classified into two types:

1. Regression
2. Classification

1. REGRESSION

Regression is supervised learning where the output (target variable) is a continuous numeric value. The main objective of regression is to model the relationship between the input features (independent variables) and the continuous output variable (dependent variable) such that the model can make predictions for new, unknown data.

Types of Algorithms Used in Regression

- Linear Regression
- Polynomial Regression
- Ridge Regression
- Lasso Regression
- Elastic Net
- Support Vector Regression (SVR)
- Decision Tree Regression
- Random Forest Regression
- Gradient Boosting Regression (XGBoost, LightGBM)

- **Linear Regression**

Linear regression is a statistical method to model the relationship between one or more independent variables (predictors) and a continuous dependent variable by fitting a straight line, which is known as a linear equation. The aim is to minimize the error between predicted and actual values.

How It Is Used?

It fits a straight line, which represents the relationship between one or more independent variables (predictors) and a continuous dependent variable by linear regression models. This model predicts the dependent variable according to the values of independent variables.

It predicts the continuous output based on the linear relationship between input features and the output. Often, the equation is in the form

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n + \epsilon$$

Where:

-Y is the dependent variable,

-$X_1, X_2, \ldots, X_n$ are independent variables,

-β_0 is the intercept,

-$B_1, \beta_2, \ldots, \beta_n$ are the regression coefficients.

- **Ridge Regression**

It's a form of linear regression that includes L2 regularization with adding some term on the loss function to contract all model's coefficients. This regularization does reduce overfitting much, especially when there exists multicollinearity-that is, high correlation among features.

How Is Ridge Regression Used?

In the case of ridge regression, the objective is to reduce the loss function while reducing coefficients of the model to stop overfitting.

$$Loss\ Function = Residual\ Sum\ of\ Squares + \lambda \sum_{\{i=1\}}^{n} \quad {\beta_i}^2$$

-where λ is the regularization parameter,

-β_i are the regression coefficients.

- **Lasso Regression**

Lasso regression is the version of ridge regression, with L1 regularization. In this technique, some of the coefficients get shrunk to zero, thereby acting both as feature selector and as a regularize. It is particularly helpful in scenarios where there are too many irrelevant or redundant features.

How is it used?

Lasso (Least Absolute Shrinkage and Selection Operator) regression uses L1 regularization, which can set some coefficients to be exactly zero. This method not only does regularization but also does feature selection, so it is very useful when one has many irrelevant or redundant predictors.

Lasso regression tries to minimize the loss function while adding the constraint that the sum of the absolute values of the coefficients should be less than a constant.

$$Loss\ Function = Residual\ Sum\ of\ Squares + \lambda \sum_{\{i=1\}}^{n} \quad |\, \beta_i \,|$$

Where:
-λ is the regularization parameter
-βi are the coefficients.

- **Support Vector Regression (SVR)**

Support Vector Regression (SVR) is a regression technique that bases on Support Vector Machines (SVM). SVR tries to find a hyperplane best fitted to the data and contains a margin of tolerance epsilon around the predictions. SVR is useful for ignoring minor errors and focusing on bigger ones. It's good especially for non-linear data.

How It Is Used?

SVR wants to fit a function so that it is at the tolerance of epsilon away from the actual observed values as possible and the model needs to be as flat as possible. The goal is minimizing the error outside the margin while maximizing the margin around the hyperplane.

2.CLASSIFICATION

Classification is a type of supervised learning technique in machine learning where the goal is to predict the category or class label of a given input based on its features. In supervised learning, the model is trained using a labelled dataset, where each training instance consists of an input (with features) and a known, corresponding class label (target value). The model learns the mapping from the input features to the class labels and uses this learned relationship to classify new, unseen data. The "classification" term refers to the fact that we are classifying tasks by predefining classes or categories in advance for prediction purposes.

Types of Algorithms used in Classification

- Logistic Regression
- k Nearest Neighbours (k NN)
- Decision Trees
- Random Forest
- Support Vector Machines (SVM)
- Naive Bayes
- Gradient Boosting Classifiers (XGBoost, LightGBM, CatBoost)
- Neural Networks for Classification

- **Logistic Regression**

Logistic Regression is a linear model used for binary classification tasks, where the output is a probability that an input belongs to a specific class. It uses the logistic (sigmoid) function to model the relationship between the input features and the class probability, producing an output between 0 and 1. The predicted class is typically assigned based on a threshold, such as 0.5.

How It Is Used?

In Logistic Regression, the objective is to model the probability of the target variable being in a certain class. The model is trained by minimizing the log-likelihood function, which measures how well the model predicts the actual class labels. The logistic function is applied to a linear combination of input features and their corresponding coefficients, and the model adjusts these coefficients to minimize error.

$$Loss\, Function\ =\ -\sum(y\ *\ log(p)\ +\ (1\ -\ y)\ *\ log(1\ -\ p))$$

Where y is the actual label, p is the predicted probability, and the sum is taken over all data points.

- **k Nearest Neighbours (k NN)**

k-Nearest Neighbours (k-NN) is a non-parametric, instance-based algorithm in machine learning that performs classification and regression. For classification, it looks up the k nearest labelled data points in the feature space with a distance metric, say Euclidean distance, to find the nearest neighbours.

How It's Used?

k-NN works by computing the distance between a query point and all points in the training dataset. The algorithm identifies the k nearest neighbours and classifies the query point based on the majority class among them (for classification). The value of k can significantly impact the model's performance and is often selected using cross-validation.

- **Decision Trees**

A Decision Tree is a structure that resembles a flowchart and is used for solving classification and regression problems. It's essentially breaking down the data into smaller subsets based on certain features. Each branch of the tree is a decision made on a certain feature, and the leaves of the tree are what show the final predicted class or value.

How Do They Work?

The process is an algorithm that repeatedly splits the data into subsets based on the feature that best separates the different classes. This is determined by measures such as Gini impurity or Information Gain. The tree keeps growing until each subset is "pure" enough or until certain conditions are met, such as reaching a maximum depth. Once the tree is constructed, predictions can be made by tracing a path from the root to a leaf based on the input given.

- **Random Forest**

Random Forest is an ensemble learning method that creates multiple decision trees and aggregates their predictions to increase accuracy and avoid overfitting. Each tree is constructed on a random subset of the data and a random subset of features.

How It Is Used?

Random Forest builds several decision trees by bootstrapping (sampling with replacement) and feature bagging, whereby for each tree, the features are randomly selected. In the prediction phase, the output of each tree is aggregated, and the final output is decided based on majority voting for classification tasks or average for regression tasks.

II. UNSUPERVISED LEARNING

Unsupervised learning is another big component of machine learning that tells the algorithm to find structure, correlations, or patterns in a set of data without labels of outputs. Unsupervised learning works with only inputs and looks for any hidden patterns, structures, or correlations without any explicit instruction against supervision where the computer learns its lessons from labelled data - the input-output pairs. This is helpful for tasks where labelled data is either expensive or unavailable.

TYPES OF UNSUPERVISED LEARNING

The main types of unsupervised learning are:

1. Clustering
2. Dimensionality Reduction
3. Association

1.CLUSTERING

Clustering is a strong technique in unsupervised learning that effectively groups the data points into clusters by their similarities. This identifies shared features among the data by placing similar data points in the same cluster and separating dissimilar ones; thus, it is one of the most important techniques in data analysis.

Types of Clustering Algorithms

- K-Means Clustering
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- Hierarchical Clustering
- Gaussian Mixture Models (GMM)

- **K-Means Clustering**

K-means clustering is one of the most popular and widely applied algorithms in machine learning, primarily in unsupervised learning. It is used for group classification of data depending upon their similarity. The central idea is straightforward: we aim to partition a dataset into K unique clusters, where each data point is associated with the cluster whose "center" (or centroid) is nearest.

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

DBSCAN is a clustering algorithm, which groups closely packed data points, known as density-based clustering, and also marks outliers or noise data points that lie alone in low-density regions. The number of clusters need not be specified beforehand as it happens in K-Means. It can even detect clusters of arbitrary shapes.

- **Hierarchical clustering**

Clustering is an unsupervised machine-learning technique used to build a hierarchy of clusters. It is a method of cluster analysis that seeks to group data points based on their similarity. The output of hierarchical clustering is often a tree-like structure called a **dendrogram**, which illustrates the arrangement of the clusters.

Dendrogram

A dendrogram is a tree-like diagram that illustrates how clusters are formed at each stage of merging. The height at which two clusters merge represents the distance (or dissimilarity) between them; the lower the height, the more similar the clusters are. By "cutting" the dendrogram at a specific level, we can determine the number of clusters that best fit the dataset.

- **Gaussian Mixture Model (GMM)**

A **Gaussian Mixture Model (GMM)** is a probabilistic model that assumes that the data is generated from a mixture of several Gaussian distributions (normal distributions), each representing a different cluster. The model doesn’t require the clusters to be of equal size or shape, and each data point is assigned a probability of belonging to each cluster, making it more flexible compared to methods like K-means clustering.

In simple terms, GMM is a way of grouping data points by modelling them as coming from different Gaussian distributions, where each distribution represents a cluster.

2.DIMENSIONALITY REDUCTION

Dimensionality reduction involves reducing the number of features in a dataset while retaining essential information. This technique simplifies high-dimensional data, making it easier to analyse and visualize. It improves computational efficiency, reduces overfitting in machine learning, and reveals hidden patterns by focusing on relevant features. Common methods include Principal Component Analysis (PCA), t-distributed Stochastic Neighbour Embedding (t-SNE), and Linear Discriminant Analysis (LDA).

Techniques for Dimensionality Reduction:

- Principal Component Analysis (PCA)
- t-Distributed Stochastic Neighbour Embedding (t-SNE)
- Linear Discriminant Analysis (LDA)

3.ASSOCIATION

Association is a technique in unsupervised learning where the goal is to find interesting relationships or patterns in datasets, particularly those where items frequently co-occur. Unlike supervised learning, association does not rely on predefined labels or outcomes but instead focuses on uncovering associations among the features or items in the data. The most common application of association is **market basket analysis**, where it identifies sets of products that tend to be bought together. However, the association can also be applied in a wide range of other domains, such as healthcare, web mining, and customer behaviour analysis.

How Association rule works?

- Data Collection
- Data Pre-processing
- Frequent Item set Generation

Support Calculation

Support for an item set X is defined as:

$$Support(X) = \frac{Number\ of\ transactions\ containing\ itemset\ X}{Total\ number\ of\ transactions}$$

Higher support indicates a more frequent item set.

Association Rule Generation

Once frequent item sets are identified, association rules are generated in the form $X \rightarrow Y$, indicating that if X is present, Y is likely present as well.

Confidence: Measures how often Y occurs when X occurs:

$$Confidence(X \rightarrow Y) = \frac{Support(X \cup Y)}{Support(X)}$$

Lift: Indicates how much more likely Y occurs when X is present compared to when it's not. A lift greater than 1 indicates a positive association:

$$Lift(X \to Y) = \frac{Confidence(X \to Y)}{Support(Y)}$$

Pruning

Finally, generated rules are filtered based on minimum thresholds for support, confidence, and lift to retain only the most relevant rules. For instance, rules with low confidence may be discarded as they do not reliably predict the consequent.

Association Rule Mining Algorithms

- Apriori Algorithm
- FP-Growth Algorithm

- **Apriori Algorithm**

The Apriori algorithm is a breadth first search method used to identify frequent item sets and generate association rules. It relies on the Apriori property, which states that if an item set is frequent, all of its subsets must also be frequent, allowing for early pruning of candidate item sets.

- **FP-Growth Algorithm**

FP-Growth is a more efficient algorithm than Apriori because it uses a compact data structure called an FP-tree to represent item sets and their frequencies, avoiding the generation of candidate item sets.

III. Reinforcement Learning

Reinforcement Learning (RL) is a type of machine learning where an **agent** learns to make decisions by interacting with its environment to maximize a cumulative **reward**. The agent takes actions in different states of the environment, and based on these actions, it receives feedback in the form of rewards or penalties. The goal of the agent is to develop a strategy known as a **policy** that maximizes the long-term reward over time, rather than simply optimizing for immediate rewards. RL is characterized by trial and error, with the agent learning from the consequences of its actions.

Types of Reinforcement Learning:

- Model Free vs. Model Based
- On Policy vs. Off Policy
- Value Based vs. Policy Based vs. Actor Critic

1.Model Free vs. Model Based:

Model Free: In model free reinforcement learning, the agent interacts with the environment to learn the best actions to take, without creating an explicit model of the environment's dynamics. This means it does not try to predict the outcomes or rewards associated with its actions based on a model. Instead, it relies solely on the experiences gained from trial and error. Common algorithms in this category include:

Q Learning: This algorithm learns the value of taking a particular action in a specific state, ultimately constructing a policy based on these values.

Policy Gradient Methods: These techniques focus on optimizing the policy directly by adjusting the parameters of the policy model based on the rewards received from actions taken.

Model Based: In model-based reinforcement learning, the agent develops an internal model that simulates the environment's dynamics, including transition probabilities and reward structures. This allows the agent to plan actions more effectively by predicting the outcomes of actions before taking them. The learning process involves:

Building a model to understand how the environment responds to different actions.

Using this model to perform simulated rollouts to explore potential future states and outcomes, ultimately improving decision making.

2. On Policy vs. Off Policy:

On Policy: In on policy learning, the agent evaluates and improves the policy that it is currently following. This means that the agent learns from the actions it takes and the rewards it receives under its policy. A key example of an on-policy algorithm is SARSA (State Action Reward State Action), where the agent learns the action value function based on its experience and policy.

Off Policy: Off policy learning allows the agent to learn from actions taken by a different policy than the one it is currently following. This approach enables the agent to learn from past experiences, even if they were generated by exploratory or different policies. Q learning is a well-known off policy method that leverages this technique, allowing an agent to update its value estimates based on experiences that may not directly reflect its current strategy.

3. Value Based vs. Policy Based vs. Actor Critic:

Value Based: In value-based methods, the agent estimates the value of being in a certain state or taking a specific action in that state. The focus is on learning a value function that will help the agent make decisions aimed at maximizing returns over time. Q learning is a prominent example of this approach, where the agent learns a Q value for each state action pair.

Policy Based: Unlike value-based approaches, policy-based methods involve directly parametrizing the policy that will guide the agent's actions. This allows for the optimization of policies that might not be easily represented by a value function. Policy gradient methods, such as REINFORCE, adjust the policy parameters to maximize a defined objective function based on received rewards.

Actor Critic: The actor critic method combines elements of both values based and policy-based approaches. It features two main components: the actor, which is responsible for selecting actions based on the current policy, and the critic, which evaluates the actions taken by the actor using a value function. The critic provides feedback to the actor, improving the policy iteratively based on the evaluation of the actions taken, fostering a more balanced and efficient learning process.

Reinforcement Learning Algorithms:

1. **Q-learning**:
   - A model-free, off-policy algorithm used for learning action-value functions (Q-values).
   - The Q-value represents the expected cumulative reward for taking action a in state s and following the optimal policy thereafter.
   - The Q-value is updated iteratively using the Bellman equation

$$Q(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma max_{a'}Q(s',a') - Q(s,a)]$$

where:

- α is the learning rate,
- γ is the discount factor,
- $R(s,a)$ is the immediate reward,
- $max_{a'}Q(s',a')$ is the maximum future reward?

Deep Q-Networks (DQN):

- An extension of Q-learning that uses deep neural networks to approximate Q-values for high-dimensional state spaces, such as in video games or robotic control.

SARSA (State-Action-Reward-State-Action):

- A model-free, on-policy algorithm that updates the Q-values based on the actions taken by the agent, not on the optimal actions.
- The update rule is:

$$Q(s,a) = Q(s,a) + \alpha[R(s,a) + \gamma Q(s',a') - Q(s,a)]$$

where a' is the next action chosen by the agent.

Policy Gradient Methods:

- These methods directly optimize the policy by adjusting the parameters of the policy network using gradient-based methods. Examples include REINFORCE and Proximal Policy Optimization (PPO).
- The policy gradient is computed as

$$\nabla\theta J(\theta) = E[\nabla log\pi_{\theta}(s_t,a_t) \cdot G_t]$$

where:

- θ are the parameters of the policy,
- $\pi\theta(s_t,a_t)$ is the probability of taking action a_t in state s_t under the policy,
- G_t the return (total discounted reward) from time step t.

2.Actor-Critic Algorithms:

- Actor-critic methods combine both **value-based** and **policy-based** approaches in reinforcement learning. These algorithms use two main components:
- **Actor**: The part of the algorithm that **selects actions** based on the current policy. The actor chooses actions based on the policy it is currently learning.

- **Critic**: The component that **evaluates** the actions chosen by the actor by estimating the **value function** (state-value or action-value). The critic provides feedback to the actor by indicating how good the chosen action was.

The actor updates the policy using the **advantage** function $A(s_t, a_t)$

$$\Delta\theta = \alpha \nabla_\theta log\pi_\theta(s_t, a_t)A(s_t, a_t)$$

where:

- α is the learning rate,
- $A(s_t, a_t) = R_t - V(s_t)$ is the advantage function (the difference between the actual reward and the value estimate)?
- The critic updates the value function using the Temporal Difference (TD) error:

The critic updates the value function using the Temporal Difference (TD) error

$$\delta t = Rt + \gamma V_{(st+1)} - V(s_t)$$

where:

- δ_t is the TD error,
- $V_{(st)}$ is the value function estimate for state sts_tst,?
- γ is the discount factor.

Conclusion

Machine learning algorithms are necessary for transforming data into actionable insights. This paper reviews supervised, unsupervised, and reinforcement learning, along with some key algorithms: neural networks and ensemble methods. The choice of algorithm strictly depends on the problem, the data, and the resources available. As global technological advancements continue, machine learning's aspect toward innovation and solutions will also spread to all sectors, forming future industries.

Reference

1. https://towardsdatascience.com/machine-learning-types-and-examples-a6a9f835f56e
2. https://www.jair.org/index.php/jair/article/view/11278
3. https://www.analyticsvidhya.com/blog/2021/01/top-10-algorithms-for-machine-learning-in-2021/
4. https://www.geeksforgeeks.org/types-of-machine-learning/