e-ISBN: 978-93-6252-717-2
IIP Series

INTRODUCTION AND TYPES OF MACHINE LEARNING

Abstract

Machine learning, a cornerstone of artificial intelligence, has evolved into a multifaceted discipline with diverse methodologies catering to distinct problem-solving paradigms. This chapter delves into the various types of machine learning, unraveling the intricacies of supervised, unsupervised, and reinforcement learning, while also exploring hybrid approaches and emerging trends.

This book chapter embarks on an illuminating journey through the intricate realm of machine learning, offering a detailed examination of the foundational principles, types, and practical applications that define this transformative field. The narrative begins with an introduction to the overarching concepts of machine learning, elucidating its significance in the broader landscape of artificial intelligence.

The core of the chapter revolves around a meticulous exploration of various types of machine learning, including supervised learning, unsupervised learning, and reinforcement learning. Each type is dissected, providing a clear understanding of their distinctive methodologies, applications, and underlying algorithms. Real-world case studies and examples showcase the versatility of these types across diverse domains such as image recognition, natural language processing, and autonomous systems.

Ethical considerations take center stage as the chapter delves into the challenges and implications associated with each type of machine learning. Furthermore, the chapter provides a forward-looking perspective, exploring emerging trends and innovations in the field. Discussions on hybrid approaches,

Authors

Dr. Arpana Chaturvedi
Associate Professor
Department of AI/ML/IT
New Delhi Institute of Management
New Delhi. India.
ac240871@gmail.com

Dr. Nitish Pathak
Bhagwan Parshuram Institute of Technology
GGSIPU, New Delhi, India
nitishpathak@bpitindia.com

transfer learning, and continual learning shed light on the evolving nature of machine learning and its potential impact on future technological landscapes. By unraveling the intricacies of machine learning and its diverse types, this chapter aims to serve as a comprehensive guide for both novices and seasoned practitioners, fostering a deeper understanding of the theoretical foundations and practical applications that propel machine learning into the forefront of modern technology.

Keywords: Machine Learning, Supervised Learning, Unsupervised Learning, Reinforcement Learning, Over fitting and Under Fitting, Algorithms

I. OBJECTIVE

To provide a comprehensive understanding of the fundamental types of learning in artificial intelligence, delineating between supervised and unsupervised learning. This chapter aims to introduce the reader to the core concepts and methodologies of these learning paradigms, including classification overview, and to explain the importance of different data sets such as training, test, and validation in the development of AI models. Additionally, the chapter will address common challenges in model training, particularly overfitting and underfitting, and discuss strategies to mitigate these issues. The goal is to equip readers with the knowledge necessary to apply these concepts effectively in various applications of AI.

Structure

- Types of Learning
- Supervised learning
- Unsupervised learning
- Overview of classification
- Setup, Training, Test, Validation dataset,
- Over fitting and Under Fitting

II. INTRODUCTION

Machine learning is increasingly important because it can handle complex tasks that are too much for humans, especially when dealing with big data that we can't process manually. This technology trains computers to make our lives easier by learning from large amounts of data and making predictions. It saves us time and money.

Machine learning is a part of AI that lets machines learn from data and get better over time. It works by training on lots of data to do tasks like predicting trends or spotting patterns.

AI is everywhere now, in self-driving cars, detecting frauds, recognizing faces, and even suggesting friends on social media. It helps companies like Netflix and Amazon recommend things to customers by learning what they like.

III. TYPES OF MACHINE LEARNING

The four main types of machine learning are:

- Supervised Learning
- Unsupervised Learning
- Semi-Supervised Learning
- Reinforcement Learning

A. Supervised Learning

Supervised learning is a method where the computer is given a dataset that's already organized with correct answers. It studies this data to understand patterns, like a student learns from a textbook. The computer is shown data (like pictures or numbers) and the answers (like 'cat' or 'price'). It then learns to connect the two. After it's trained, you can give it new data it hasn't seen before, and it will use what it's learned to make good guesses.

For instance, as shown in Fig 1, if you show it pictures with labels, like 'spoon' or 'knife', the computer looks at the details shape, size, sharpness—and learns to tell them apart. Later, if you show it a new picture without a label, it uses what it learned to guess if it's a spoon or a knife. It's like pairing things up: you show it what an input (like a picture) should lead to, which is the output (like the word 'Spoon' or 'Knife').

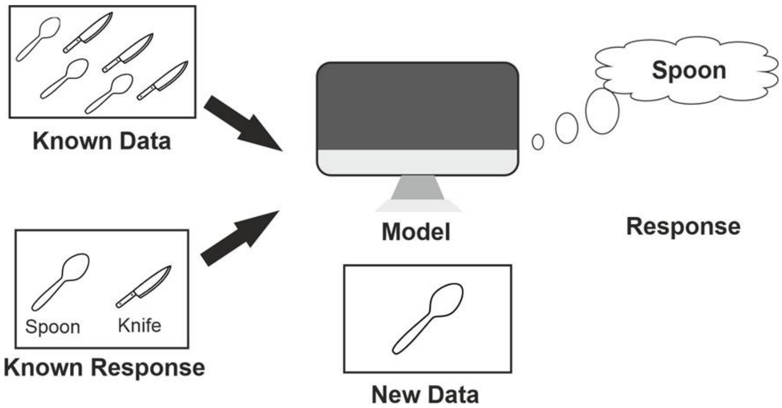


Figure 1: Supervised Learning

In this algorithm the computer recognizes and understand the pattern between the input variable-x (what you show it) and the output variable -y (what it's supposed to say). In the real world, this helps with things like figuring out if a bank transaction is suspicious or if an email is junk.

Working of Supervised Learning

In supervised learning, models undergo training with a labelled dataset, allowing them to recognize and learn from various data types. After training, the model's knowledge is evaluated using **test data** a portion of the dataset originally set aside for this purpose—to ascertain its predictive capabilities. The mechanics of supervised learning are clarified further through specific examples and illustrative diagrams shown in Fig. 2

Suppose there is a dataset comprising various shapes such as squares, rectangles, triangles, and polygons as shown in Fig.2. The initial step involves training a model to recognize each type of shape with the following criteria:

- **Square** - A shape with four equal sides is labelled.
- **Triangle** - A shape with three sides is classified or labelled.
- **Hexagon** - A shape with six equal sides is recognized or labelled .

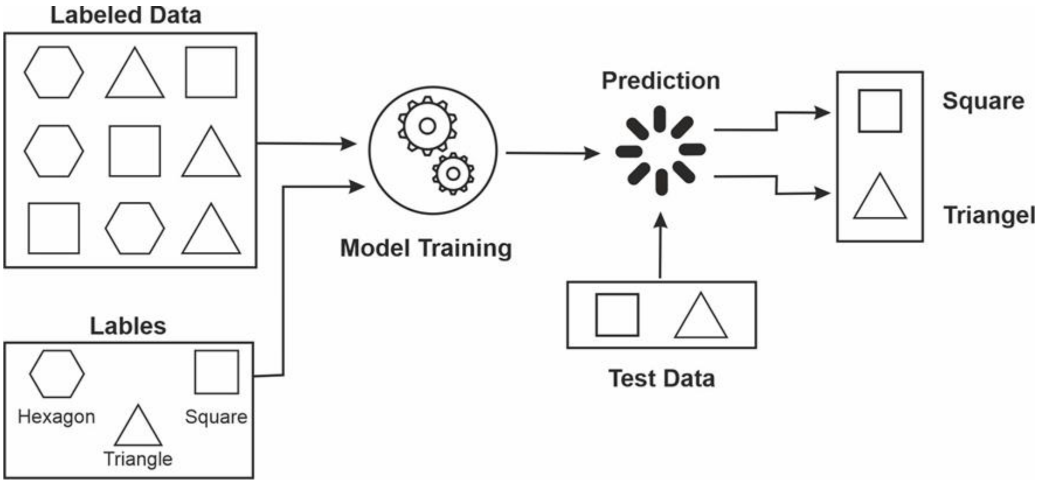


Figure 2: Supervised Learning

After the model has been trained with these rules, it is then tested with a separate set of shapes. The model's task is to determine the type of each new shape it encounters. Having been trained on shape characteristics, the model uses the number of sides and the equality of sides to classify the shape and predict the correct label as output.

Steps Involved in Supervised Learning

The process of supervised learning involves several key steps:

a. Define the Problem

- What kind of information do you want the computer to learn? (e.g., identifying images of animals, predicting the price of a house)
- What kind of data is needed to train the computer? (e.g., labeled images of animals, data with house characteristics and prices)

b. Prepare the Data

- Gather labeled data: Collect examples that are already categorized or have the correct answers.
- Split the data into three parts:
  - ➢ Training set: Used to teach the computer.
  - ➢ Test set: Used to evaluate how well the computer learned.
  - ➢ Validation set: Used to fine-tune the computer's learning process.

c. Choose the Right Tool

- Select an algorithm: This is the recipe the computer uses to learn from the data.
- Different algorithms are better suited for different types of data and problems.

d. Train the Computer

- Feed the computer the training data and let it learn.
- Use the validation set to adjust the algorithm settings and improve learning.

e. Test the Computer

- Give the computer the test data and see how well it predicts the correct answers.
- Evaluate the accuracy of the model and identify areas for improvement.

f. Fine-Tune and Improve

- If the model performs well, it's ready to use!
- If not, try different algorithms, adjust the data, or refine the training process.

Types of Supervised Learning Algorithms:

There are two types of Supervised learning algorithm. These are based on two types of problems:

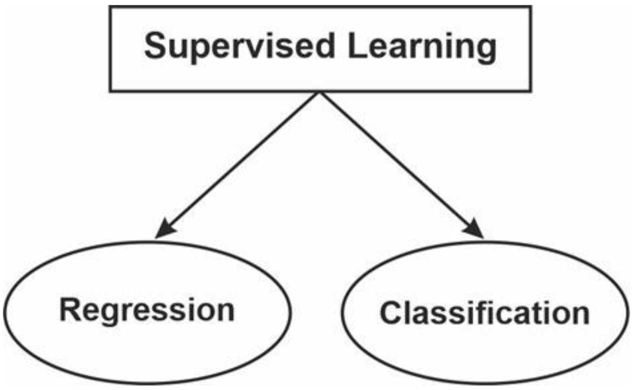


Figure 3: Types of Supervised Learning

There are two main kinds of supervised learning:

1. Classification

Classification involves creating a model that sorts information into distinct categories or classes or discrete values based on certain characteristics in the data. It assigns predefined labels to the data based on these characteristics.

The job in a classification task is to predict specific outcomes, known as class labels, by analysing different or discrete factors or features. Additionally, the task requires defining a clear separation between these categories within the data, known as a decision boundary.

For example, this algorithm can be used when we want to sort data into specific groups or categories, like sorting emails as 'spam' or 'not spam' as shown in Figure 4.These algorithms look at the data we give them and learn to predict the category for new data based on what they've seen.

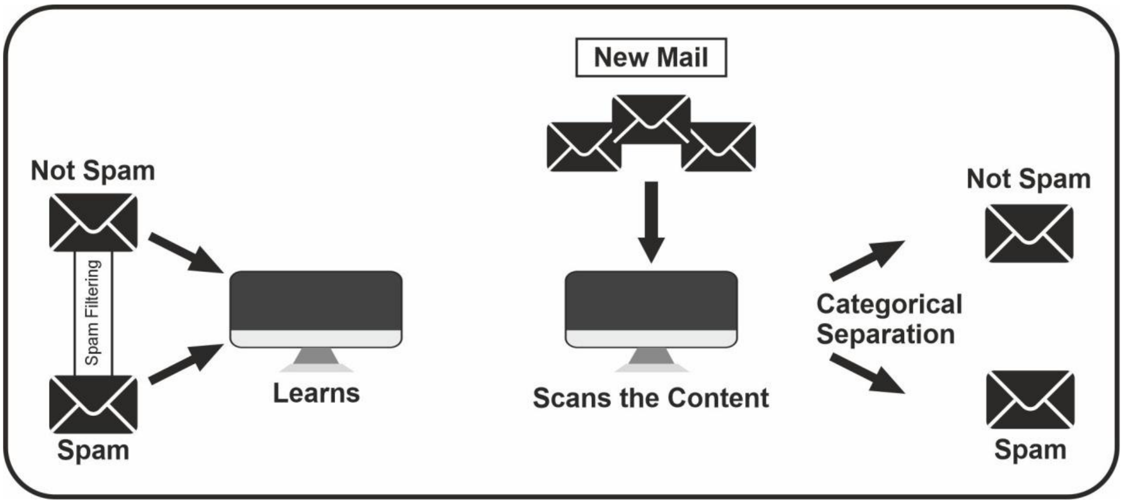


Figure 4: Filtrations of Spam Emails

Another example, a shopkeeper might use a classification algorithm to predict whether a customer will return based on past shopping behaviour.

Here's what happens in classification:

- The computer is given details (features) about something, like stats from previous matches of a sports team.
- It uses those details to predict which group (class) it belongs to, like predicting if a team will win or lose.
- The goal is to find a clear line (decision boundary) that separates the different possible groups or outcomes.
  In a way, the computer is making "IF-THEN" decisions. If certain conditions are met, then it predicts a certain group. For example, if Team A has won a lot of games when it's sunny, the computer might predict a win for Team A when the forecast is sunny. This process is used for decisions where there are fixed options, like a simple yes or no, win or lose.

There are various modern classification techniques or algorithms that have been created to sort data well, and they do this by using special approaches called bagging and boosting.

1.1 Types of Classification Algorithms

Classification Algorithms can be further divided into the mainly two category:

- **Linear Models**
  - Logistic Regression
  - Support Vector Machines

- **Non-linear Models**
  - K-Nearest Neighbours
  - Kernel SVM
  - Naïve Bayes
  - Decision Tree Classification
  - Random Forest Classification

- **Random Forest:** It's like asking a bunch of experts (trees) and combining what they all say to make a final decision.
- **Decision Trees:** This method makes decisions by going through a list of questions and following the answers until it reaches the end.
- **Logistic Regression:** It's usually used when the answer is yes or no—like if a team will win a match or not.
- **Support Vector Machines (SVM):** This algorithm finds the best boundary that separates different categories.
- **K-Nearest Neighbours (KNN):** It looks at the 'K' examples that are most like the one we're trying to figure out and picks the most common category among them.
  Each of these algorithms has its own way of deciding where to put new data based on the examples it's learned from. They can be simple, with just two choices (binary), or they can have many options (multiclass), as shown in Figure 5, like sorting different types of fruits into their own baskets.

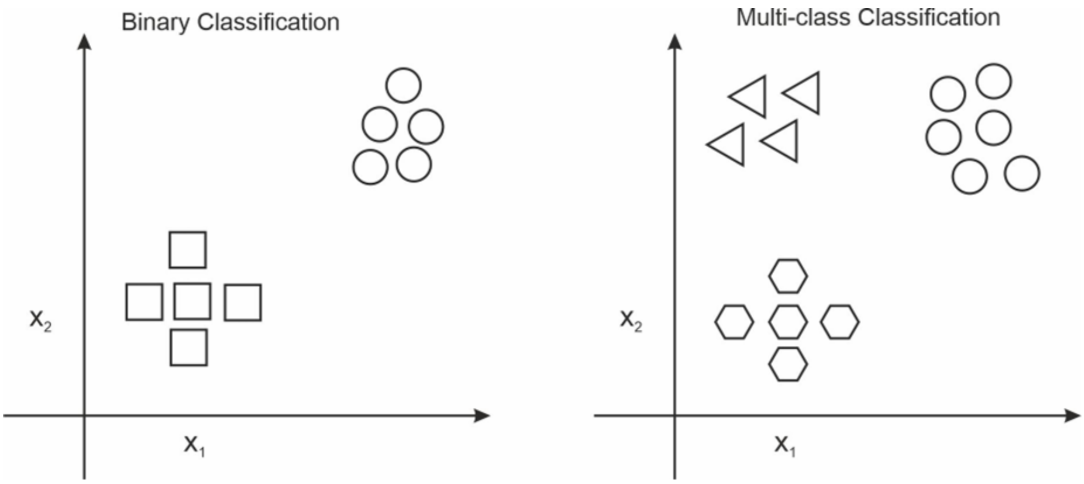


Figure 5: Binary Classification and Multiclass Classification

2. **Regression:** Regression entails establishing a model or function that effectively distinguishes data into continuous real values rather than discrete categories. This process can also reveal patterns and trends within historical data. Since regression models predict quantitative values, their performance is assessed based on the error in these predictions.

   The primary objective of regression is to uncover the relationship between a dependent variable, which is the quantity being predicted, and one or more independent variables, which are the factors influencing the dependent variable. By analysing the relationship between these variables, regression models can be constructed to make accurate predictions of the dependent variable.

Imagine we're looking at two things: how moist the air is (humidity) and how hot it is (temperature). In this case, the heat is the independent variable, and the moisture in the air depends on it. Usually, when it gets hotter, the air becomes less moist.

We put these two pieces of information into a computer model, and it figures out how they're connected. Once the computer has learned this, if we tell it how hot it is, it can guess how much moisture is in the air as shown in Figure 6.

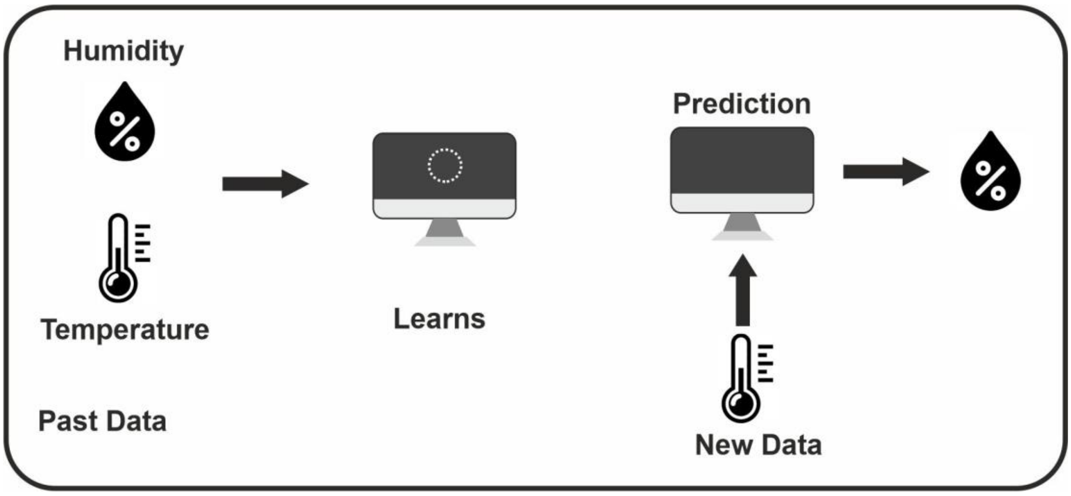
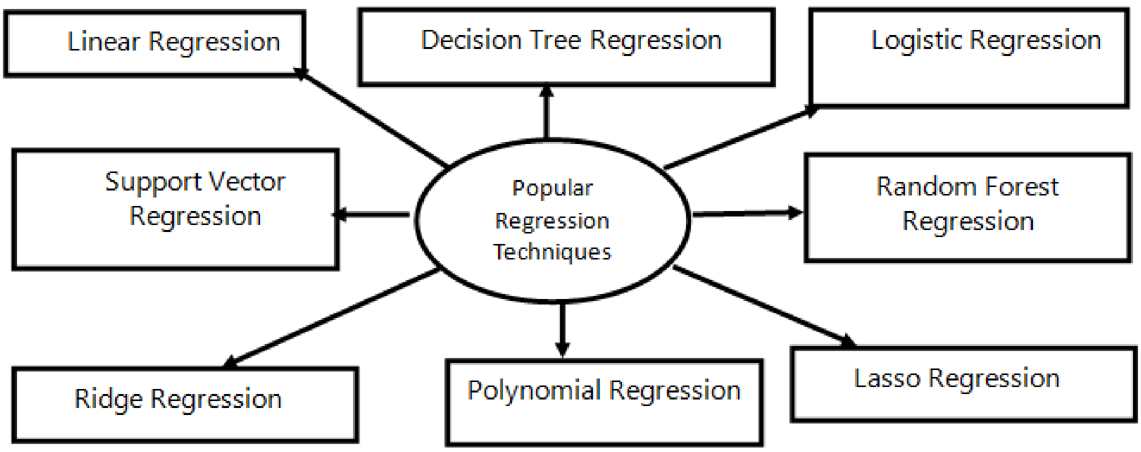


Figure 6: Prediction of Moisture in Air

2.2. Different Kinds of Regression Techniques

In the fields of data science and machine learning, a variety of regression techniques are employed, each significant in its own right depending on the situation. Fundamentally, these methods focus on understanding how independent variables influence dependent variables. Below is a list of some key regression techniques commonly used:



The detailed explanation of few of the popular Regression Algorithm are:

- **Simple Linear Regression Algorithm:** This algorithm predicts a response using a single feature. For example, it can predict a student's test score based on the number of hours they studied.
- **Multivariate Regression Algorithm:** This algorithm uses multiple features to make a prediction. For instance, it could estimate a house’s price based on its size, age, and location.
- **Decision Tree Regression Algorithm:** This method breaks down the data into a tree of decisions. It might predict a car's resale value based on factors like its make, model, and mileage.
- **Lasso Regression:** It is used to not only make predictions but also to understand which features are most important. For example, it could analyse various factors affecting heart health to identify the most crucial ones.

Difference between Classification and Regression Algorithm

Table 1: Difference between Classification and Regression

Aspect	Classification	Regression
Target Variable	Discrete (distinct categories)	Continuous (any numerical value)
Problem Examples	Spam Email Detection, Disease Diagnosis	House Price Estimation, Weather Forecasting
Approach	Finding a decision boundary to separate classes	Finding a trend line that fits the data
Evaluation Metrics	Precision, Recall, F1-Score	Mean Squared Error, R2-Score, MAPE
Problem Types	Binary or Multi-Class	Linear or Non-Linear
Data Types	Categorical outcomes based on independent variables	Continuous outcomes based on independent variables
Task	Mapping inputs to discrete output labels	Mapping inputs to continuous numerical outputs
Objective	Predicting class labels	Predicting numerical values
Use Cases	Image recognition, Sentiment analysis	Stock price prediction, Demand forecasting
Algorithm Examples	Logistic Regression, Decision Trees, SVM, K-NN	Linear Regression, Lasso Regression, Random Forest

When to use Classification Algorithm and When to use Regression Algorithm

Here are some points indicating when to use classification algorithms versus regression algorithms:

a. Classification Algorithm

- When the output has distinct categories.
- When classifying items into groups is the goal.
- When each data point belongs to one of a set of categories.
- Use it for binary decisions (e.g., yes/no, true/false).
- Use it for multi-class problems like sorting animals by species.

b. Regression Algorithm

- When predicting a numerical value.
- When the outcome is a quantity, not a category.
- Use it when there's a continuous relationship between variables.
- Use it for estimating values, like house prices or temperatures.
- Use ordinal regression when dealing with ranked categories.

This should help determine which algorithm to use based on the specific needs of your data analysis or predictive modelling task. You can find below in tabular form also to choose appropriate algorithm as per the type of Problem you have to deal with.

Table 2: Choosing Classification or Regression Algorithm

When to Use	Classification Algorithm	Regression Algorithm
Type of Output	Discrete categories (e.g., Yes/No, Types of Fruit)	Continuous numerical values (e.g., Prices, Age)
Nature of Response	Categorical (e.g., Email: Spam/Not Spam)	Quantitative (e.g., Salary, Weight)
Complexity of Problem	Binary/Multi-class classification	Linear/Non-linear relationships
Examples	Diagnosing diseases	- Predicting real estate values
	Identifying Customer Sentiment	- Estimating stock market trends
Special Cases	Multi-label classification for overlapping classes	- Ordinal regression for ordered categories

Real-Life Applications of Supervised Learning

Supervised learning has many practical uses in the real world. Here are some examples:

- **Risk Assessment:** It's used by banks and insurance companies to figure out how likely it is that something bad will happen, like a loan not being repaid. This helps them lower their chances of losing money.

- **Visual Recognition:** This is the computer's ability to recognize different things in pictures and videos, like finding out where a photo was taken or who's in it.
- **Image Segmentation:** Supervised learning techniques are employed to categorize different parts of images based on predefined categories. This method is crucial for distinguishing various elements within an image.
- **Image Recognition:** An illustrative case is the way platforms like Facebook identify people in photos by matching them to previously tagged images, a process that relies on image classification capabilities of supervised learning algorithms.
- **Diagnostic Medicine:** In healthcare, supervised learning algorithms analyze medical imagery and historical data annotated with disease markers to diagnose conditions in patients. This approach enables the accurate detection of illnesses in new patient cases.
- **Fraud Identification:** Supervised learning is leveraged to pinpoint fraudulent activities, such as irregular transactions or deceptive customers, by examining historical data to uncover patterns indicative of fraud.
- **Transaction Verification**: Supervised learning aids in determining the authenticity of credit card transactions, offering a crucial line of defense against potential financial theft.
- **Identifying Spam:** Classification algorithms play a crucial role in detecting and filtering spam. They categorize emails into spam and non-spam, directing the spam emails to a designated spam folder.
- **Voice Recognition:** In the realm of voice recognition, supervised learning algorithms are extensively utilized. These algorithms are trained using voice samples, enabling them to perform tasks like recognizing voice-activated passwords and interpreting voice commands.

Advantages of Supervised Learning

- Supervised learning algorithms operate on labelled data, providing a clear understanding of the object classes involved.
- They are valuable for forecasting outcomes based on historical data, enhancing decision-making through past insights.

Disadvantages of Supervised Learning

- Supervised learning may struggle with intricate tasks that go beyond the scope of the training data.
- There's a risk of inaccurate predictions if the new data varies significantly from the data used to train the algorithm.
- The training phase for these algorithms is computationally intensive, often requiring significant time and resources.

B. Unsupervised Algorithm

Unsupervised learning is a machine learning approach where the algorithm independently identifies patterns in data without the need for labeled training examples. The primary objective of this method is to uncover hidden structures or distributions within the data set.

e-ISBN: 978-93-6252-717-2
IIP Series

Unsupervised learning, as implied by its name, is a form of machine learning where models operate without guidance from a training dataset. In this technique, the models autonomously discover hidden patterns and insights within the provided data, like how the human brain learns new concepts.

Example

Consider a scenario where an unsupervised learning algorithm is presented with a dataset of various cat and dog images. This algorithm hasn't been previously trained with this specific dataset, so it lacks pre-existing knowledge about the characteristics of these images. The algorithm's challenge is to independently discern features within the images. It accomplishes this by sorting the images into clusters based on the similarities it detects among them.

Functioning of Unsupervised Learning

The functioning of unsupervised learning is illustrated in the Fig. 6 In this example, we use input data that is unlabelled, indicating that it hasn't been classified into categories, and there are no corresponding output labels provided.

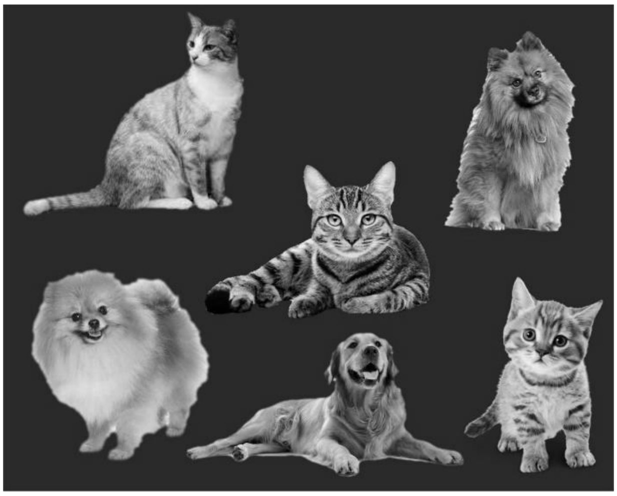


Figure 6: Various Images of Cats and Dogs to identify.

Now, this unlabelled input data is fed to the machine learning model to train it. Firstly, it will interpret the raw data to find the hidden patterns from the data and then will apply suitable algorithms such as k-means clustering, Decision tree, etc.

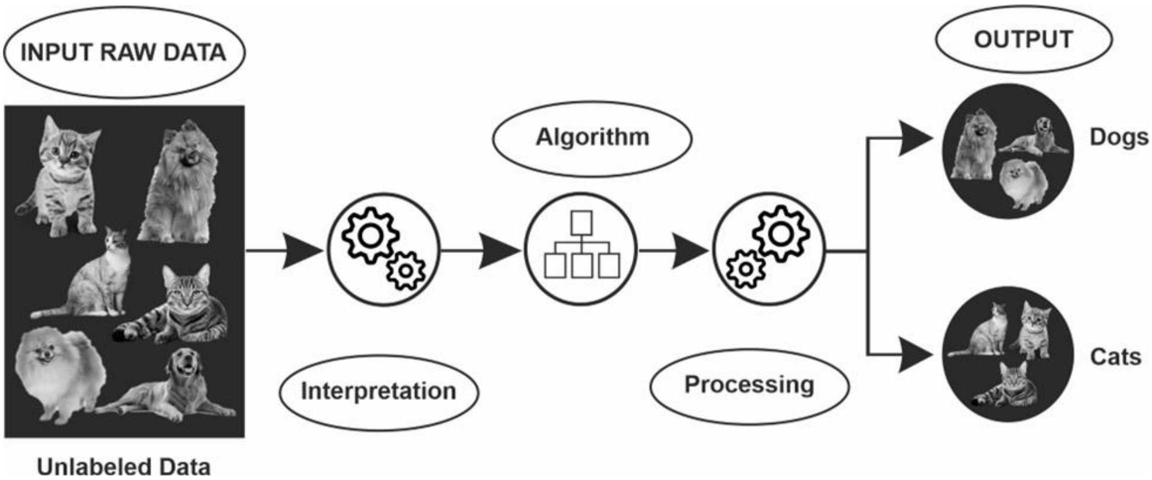


Figure 7: Unsupervised Algorithm- Identifies Unlabelled Input

After implementing the appropriate algorithm, it segregates the data items or objects into clusters based on their similarities and differences.

Types of Unsupervised Learning Algorithm

The unsupervised learning algorithm can be further categorized into two types of problems as shown in Fig. 8:

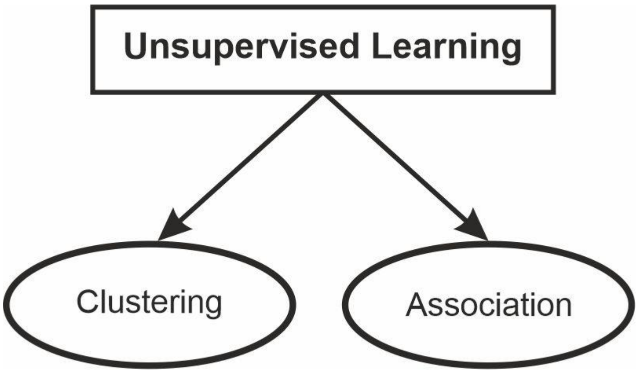


Figure 8: Types of Unsupervised Learning

- **Clustering:** This technique involves grouping objects into clusters where each group contains items that are similar to each other and dissimilar to items in other groups. Through cluster analysis, similarities among data objects are identified, allowing them to be categorized based on these shared traits.
- **Association:** This method in unsupervised learning helps discover relationships among variables within a large database. It identifies sets of items that frequently occur together in the dataset. Association rules enhance marketing strategies by revealing patterns like customers who purchase one item (e.g., bread) often also buy another (e.g., butter or jam). Market Basket Analysis is a classic example of this technique.

Unsupervised Learning Algorithms

There are so many types of Unsupervised Algorithms. Some popular algorithms are given below:

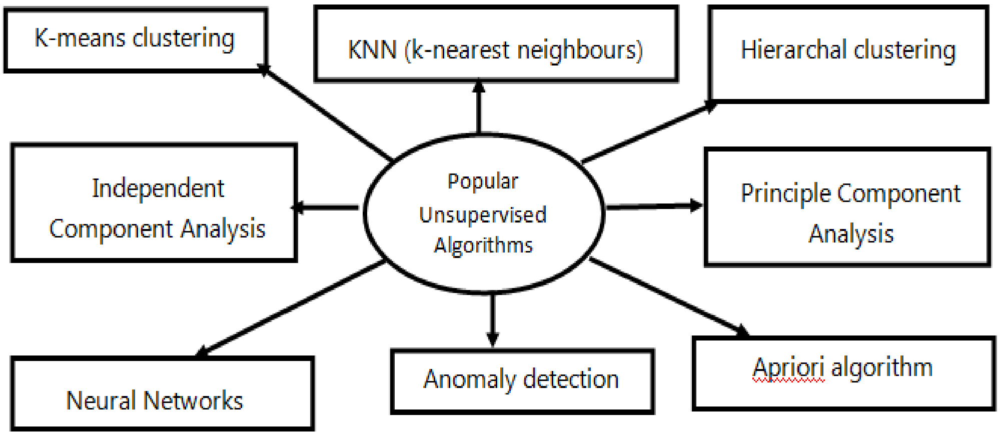


Figure 9: Popular Unsupervised ALgorithm

Dimensionality Reduction

Dimensionality reduction techniques aim to decrease the number of input variables in a dataset, preserving as much valuable information as possible. This simplification helps with data analysis and visualization by addressing the "curse of dimensionality," which is the problem of machine learning models performing worse as the feature space grows. Dimensionality reduction not only combats overfitting by simplifying the models but also improves model generalization.

Dimensionality reduction is a process that reduces the complexity of machine learning models by decreasing the number of input variables. This can lead to improved performance and better generalization to new data. There are two primary methods for reducing dimensionality:

- **Feature Selection**: This method involves choosing a subset of the most relevant features to use in model construction.
- **Feature Extraction**: This technique creates a new set of features from the original ones that capture the most important information in a reduced dimension.

Components of Dimensionality Reduction

There are two components of dimensionality reduction:

a. **Feature Selection:** This involves selecting a subset of the most relevant features from the original dataset. There are three strategies for feature selection:

- **Filter:** This method applies a statistical measure to assign a scoring to each feature. Features are ranked by the score and either selected to be kept or removed from the dataset.
- **Wrapper:** It uses a predictive model to evaluate a combination of features and determine the best subset. The selection process is based on the model's performance.
- **Embedded:** This approach involves algorithms that have built-in feature selection methods during the model training process.

b. Feature Extraction: This technique transforms or projects data from a high-dimensional space to a lower-dimensional space. The transformation is achieved while retaining as much significant information as possible from the original dataset. Methods like PCA (Principal Component Analysis) are used to derive a set of new smaller features that capture the most important variance or structure of the original data.

Methods of Dimensionality Reduction

The There are various techniques for reducing dimensionality, some of which include:

- **Principal Component Analysis (PCA):** This linear technique, introduced by Karl Pearson, projects high-dimensional data to a lower-dimensional space by maximizing the variance in the lower-dimensional representation.
- **Linear Discriminant Analysis (LDA):** LDA aims to find a linear combination of features that best separate two or more classes of objects or events.
- **Generalized Discriminant Analysis (GDA):** This method extends LDA for use with non-linear separations, allowing for dimensionality reduction in more complex datasets.

Dimensionality reduction can be categorized as either linear or non-linear, based on the technique employed. PCA, the most prominent linear method, reduces dimensionality under the condition that the data variance remains as high as possible in the reduced space, thus preserving essential data characteristics.

Advantages of Dimensionality Reduction

- **Data Compression:** Reduces the dataset size, leading to savings in storage space.
- **Faster Computation:** Lowers the computation time required for data processing.
- **Elimination of Redundancy:** Helps in removing duplicate features, streamlining the data.
- **Improved Visualization:** Makes high-dimensional data easier to visualize in 2D or 3D.
- **Prevents Overfitting:** Reduces model complexity, which can help in avoiding overfitting.
- **Feature Extraction:** Aids in identifying significant features from complex datasets.
- **Data Preprocessing:** Serves as an initial step to simplify datasets before applying machine learning algorithms, enhancing model performance.
- **Enhanced Model Performance:** By reducing noise and irrelevant data, it improves the accuracy and efficiency of machine learning models.

e-ISBN: 978-93-6252-717-2
IIP Series

Disadvantages of Dimensionality Reduction

- **Potential Data Loss:** Some information may be lost when reducing dimensions.
- **Linear Correlations:** PCA may not be effective if the variables have non-linear relationships.
- **Inadequacy for Complex Data:** PCA may not work well for datasets where mean and covariances do not capture the full data structure.
- **Uncertainty in Component Selection:** Deciding the number of components to retain can be challenging and often relies on rule-of-thumb methods.
- **Interpretability Issues:** The new, reduced dimensions may be less interpretable than the original features.
- **Risk of Overfitting:** If not done correctly, dimensionality reduction can cause overfitting, especially when component numbers are chosen from the training set.
- **Outlier Sensitivity:** Some methods might be overly influenced by outliers, skewing the reduced dataset.
- **Computational Demands:** Techniques like manifold learning may require extensive computation, particularly with large datasets.

Advantages and Disadvantages of Unsupervised Learning Algorithm

Here are some advantages and disadvantages of Unsupervised Learning Algorithm:

Advantages of Unsupervised Learning

- **Complex Task Handling:** It's adept at handling complex tasks since it doesn't require labelled data.
- **Ease of Obtaining Data:** Unlabelled data is more abundant and easier to acquire than labelled data.

Disadvantages of Unsupervised Learning

- **Potential Inaccuracy:** Without labelled data, the accuracy of the outcomes may suffer.
- **Difficulty in Implementation:** Managing and interpreting unlabelled data can be challenging due to the lack of predefined output.

Real Life Applications of Unsupervised Learning

- **Market Basket Analysis:** Used to understand customer purchase patterns and predict future buying behaviours by analysing items frequently bought together.
- **Semantic Clustering:** Enhances search engine performance by categorizing words and phrases according to their meanings.
- **Logistics Optimization:** Assists businesses in determining demand, managing stock, and planning efficient delivery routes based on historical sales and location data.
- **Safety Enhancements:** Identifies high-risk areas for accidents from historical data, informing where safety measures should be prioritized.

- **Plagiarism Detection:** Analyses documents to uncover plagiarism in academic and scholarly articles.
- **Personalized Recommendations:** Powers recommendation engines on e-commerce and streaming platforms to suggest products or media to users.
- **Fraud Detection:** Detects unusual patterns in financial transactions that could indicate fraudulent behaviour.
- **Data Analysis:** Uses Singular Value Decomposition (SVD) for extracting and analysing specific information from large datasets.

C. Semi-Supervised Learning

Semi-supervised learning is a machine learning approach that incorporates elements from both supervised learning (which uses labelled data) and unsupervised learning (which uses unlabelled data). It is particularly useful when acquiring labelled data is expensive or labour-intensive but there's an abundance of unlabelled data available. In practice, semi-supervised learning algorithms work with a small amount of labelled data supplemented by a larger amount of unlabelled data. The goal is to leverage the structure and distribution of the unlabelled data to better understand the overall dataset and make more accurate predictions.

In a real-world analogy, think of semi-supervised learning as a student who has received some instruction from a teacher (supervised learning) but is also expected to study and learn on their own (unsupervised learning). The combination of these learning methods helps the student to gain a more comprehensive understanding of the subject matter. Similarly, semi-supervised learning aims to create a model that learns from both the guidance provided by the labelled data and the freedom to explore and make inferences from the unlabelled data.

Advantages of Semi-Supervised Learning

- **Simplicity:** The algorithms are generally straightforward and user-friendly, making them easy to grasp.
- **Efficiency:** These algorithms can be highly efficient, as they require fewer labelled instances.
- **Problem-Solving:** They address certain limitations of both supervised and unsupervised learning by utilizing a mix of labelled and unlabelled data.

Disadvantages of Semi-Supervised Learning

- **Stability:** The results across iterations can be inconsistent, leading to potential instability in the model's performance.
- **Data Limitations:** These algorithms are not well-suited for network-level data which requires different analytical approaches.
- **Lower Accuracy:** The accuracy of semi-supervised learning may not match that of fully supervised learning, especially if the labelled data is not representative of the entire dataset.

D. Reinforcement Learning: Learning Through Interaction

Reinforcement learning (RL) is a branch of machine learning where an AI agent learns to make decisions by executing actions and receiving feedback, optimizing for a cumulative reward. This method stands out because it does not need labelled data. Instead, the agent learns through the outcomes of its actions, akin to trial and error.

The RL process parallels human experiential learning, much like how a child learns from daily interactions. For instance, in video games, the agent learns to play better by making moves (actions) in various situations (states) and receiving scores (rewards or penalties) for those moves.

Reinforcement learning has diverse applications across fields such as game theory, operations research, and multi-agent systems. Typically, RL problems are framed as Markov Decision Processes, where the agent's interaction with its environment involves a cycle of states, actions, and feedback, leading to new states and learning opportunities.

Types of Reinforcement Learning

a. **Model-Based Reinforcement Learning:** The agent builds a model of the environment to understand state transitions and rewards. Using this model, the agent plans out actions to maximize rewards. Algorithms like Value Iteration and Policy Iteration fall into this category.
b. **Model-Free Reinforcement Learning:** Here, the agent learns a policy directly from interactions with the environment without modelling it. It updates its policy based on the reward outcomes of its actions. Q-Learning, SARSA, and Deep Reinforcement Learning are key model-free algorithms.

Categories of Reinforcement Learning

Reinforcement learning can be divided primarily into two types based on the nature of the reinforcement provided:

a. **Positive Reinforcement Learning:** This type involves increasing the likelihood that a desired behavior will be repeated in the future by introducing a positive reward or incentive. It strengthens the behavior of the agent and has a beneficial effect on its actions.
b. **Negative Reinforcement Learning:** This category operates on the principle of increasing the probability that a particular behavior will occur again by removing or avoiding a negative condition. It's based on the concept that actions that lead to the removal of an adverse event will be reinforced and hence more likely to recur.

Real-World Use Cases of Reinforcement Learning

a. **Video Games:** Reinforcement learning (RL) is extensively used in video games to achieve performances beyond human capabilities. Notable examples include

AlphaGO and AlphaGO Zero, where RL algorithms were crucial in mastering the game of Go.

b. **Resource Management:** Reinforcement learning has been applied to manage computing resources effectively, as demonstrated in research like "Resource Management with Deep Reinforcement Learning", which discusses using RL to schedule resources efficiently to reduce job slowdowns in computer systems.
c. **Robotics:** RL is pivotal in the field of robotics, particularly in industrial and manufacturing sectors. It's used to enhance the capabilities of robots, driving the vision of creating intelligent, autonomous machines with AI and machine learning.
d. **Text Mining:** Salesforce is one company that employs reinforcement learning in text mining, which is an important aspect of natural language processing (NLP), to improve the extraction of useful information from large volumes of text.

Advantages of Reinforcement Learning

- **Complex Problem Solving:** Reinforcement learning is adept at tackling complex, real-world problems that conventional algorithms may struggle with.
- **Human-like Learning:** The RL model mimics human learning processes, often leading to highly accurate and efficient solutions.
- **Long-Term Benefits:** RL is designed to maximize not just immediate rewards but also long-term gains, making it effective for strategies that unfold over time.

Disadvantages of Reinforcement Learning

- **Not Suited for Simple Tasks:** RL algorithms may be unnecessarily complex for simple problems, where simpler algorithms could suffice.
- **Data and Computation Intensive:** These algorithms require substantial amounts of data and computational power to function effectively.
- **Risk of State Overload:** Excessive use of reinforcement learning can result in a state space that is too large to manage effectively, potentially degrading the performance of the model.

IV. IMPORTANCE OF DATA IN MACHINE LEARNING: UNDERSTANDING, TYPES, AND PROCESSING

Data is the cornerstone of Machine Learning (ML), representing the information collected through observations or measurements used to train models. The effectiveness of an ML model is heavily dependent on the quality and volume of the data it is trained on.
Data becomes **information** once it is processed, given context, and interpreted, which can then lead to actionable insights for users. When information is further synthesized with experience and learning, it evolves into **knowledge**. (Fig.10) This knowledge equips individuals or organizations with a deeper understanding and facilitates the development of new concepts or strategies.

- **Information:** This is data that has been given context and interpreted, yielding meaningful patterns or conclusions for the user.

- **Knowledge:** This represents a synthesis of information, combined with experience, learning, and insight, leading to an enhanced understanding or the formation of new concepts for an individual or organization.

Figure 10: Data-Information-Knowledge

Example

Consider the scenario where a Shopping Mart Owner has amassed a large volume of raw data from customer surveys. This data is made up of a series of questions and their respective answers, embodying the unrefined input that is ripe for analysis. The challenge lies in extracting actionable insights from this extensive dataset, as manually reviewing each response would be inefficient and time-consuming.

However, sifting through this extensive collection of survey responses manually is impractical and time-consuming. To address this, data manipulation tools and methods such as software applications, calculations, and graphical representations come into play. These tools process the raw data, organizing and interpreting it to produce meaningful patterns and insights – this processed output is what we call information. Hence, raw data must be transformed to yield information.

Knowledge then comes into the picture when individuals use this information. It involves understanding the context, drawing on personal experiences, and applying critical thinking to inform decisions or actions. Knowledge reflects the human capacity to interpret information and apply it effectively, which can differ markedly between individuals, even when they have the same information.

Different Forms of Data

Machine learning utilizes different forms of data:

- **Numeric Data:** This type of data is quantifiable and measurable, expressed in numerical terms. For example, age, salary, and temperature are numeric data.
- **Categorical Data:** Also known as nominal data, this refers to data that can be divided into specific categories or groups that do not have a numerical relationship. Examples include gender, race, or the type of browser used.
- **Ordinal Data:** These are categorical data with a clear ordering or ranking. They express attributes in a relative order, such as rankings in a race or satisfaction ratings.
- **Time-Series:** Data that is indexed in time order, often used for forecasting or understanding temporal patterns.

Data Categories in Machine Learning

There are two main categories of data used in machine learning:

a. **Labelled Data:** Used in supervised learning, this data includes both the input features and the corresponding labels (desired outputs).
b. **Unlabelled Data**: Used in unsupervised learning, this data consists of features without any associated labels.

Data Preprocessing in Machine Learning

This involves several steps to prepare the data for the model:

- **Data Cleaning:** Involves correcting or removing incorrect, corrupted, or duplicate data.
- **Normalization:** Adjusts numerical data to fall within a certain range to ensure consistency in scale.
- **Handling Missing Values:** Involves filling in or discarding data points that have missing fields.
- **Feature Selection/Engineering:** Selecting the most impactful features or creating new features from the existing data to enhance the model's learning capability.
  Properly structured and pre-processed data is vital for the success of machine learning models, as it directly affects their ability to learn and make accurate predictions.

Data Splitting in Machine Learning

In machine learning, data is typically split into three distinct sets (Fig.11):

- **Training Data:** This is the dataset that the model is trained on. It includes both the inputs and the expected outputs (labels). The model 'learns' from this data by adjusting its parameters to map the given inputs to their corresponding outputs.
- **Validation Data:** This dataset is used to fine-tune the model's hyperparameters, which are the configuration settings used to structure the learning process. The validation data is used frequently throughout the training phase to gauge model performance and make iterative adjustments.
- **Testing Data:** After the model has been trained and validated, the testing data serves to assess its performance. This dataset is kept separate and unseen by the model during training. The model makes predictions based on the testing data inputs, and its performance is evaluated by comparing these predictions against the actual outputs.

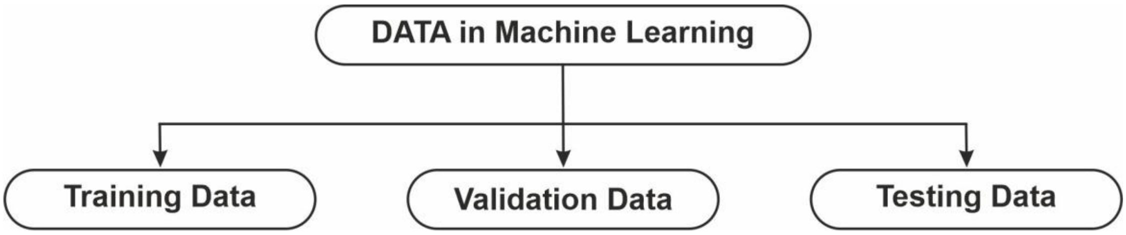


Figure 11: Splitting of Data

In machine learning projects, datasets are crucial and typically divided into two parts: a training set and a test set. The training set is used to teach the machine learning model, and the test set helps evaluate how well the model performs. This split helps determine the model's ability to handle new, unseen data. It's important to make sure these datasets accurately reflect the problem being addressed and are split correctly to prevent bias or overfitting.

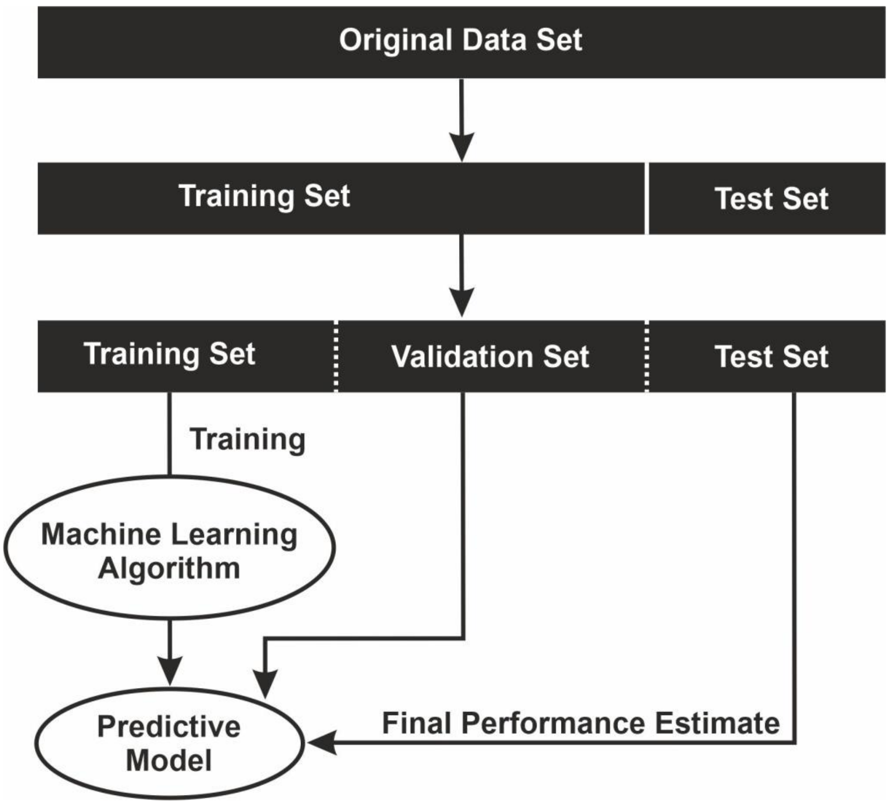


Figure 12: Splitting of Data

Overfitting and Underfitting in Machine Learning

In machine learning, two common issues that can affect a model's performance are overfitting and underfitting. The main aim for any machine learning model is to be able to generalize well. Generalization refers to the model's ability to accurately respond to new, unseen input after being trained on a dataset. Therefore, to ensure a model performs effectively, it's important to check for overfitting and underfitting, which indicate whether the model is generalizing correctly.

Before diving into overfitting and underfitting, let's clarify some basic concepts that are key to understanding these issues:

- **Signal:** This is the real pattern in the data that a machine learning model uses to learn.
- **Noise:** This refers to unnecessary or irrelevant information in the data that can lower the model's performance.
- **Bias:** This is an error in predictions caused by oversimplifying the algorithms in machine learning. It's essentially the gap between what the model predicts and the actual results.
- **Variance:** This happens when a machine learning model works well with its training data but fails to perform effectively with new, unseen data (the test dataset).

e-ISBN: 978-93-6252-717-2

Overfitting

Overfitting happens when a machine learning model learns too much from the training data, including its errors and irrelevant details (noise). This makes the model less effective and accurate because it's too focused on the specific data it was trained on, rather than being adaptable to new data. An overfitted model usually has low bias but high variance. The risk of overfitting grows the more we train the model, as it starts to reflect the training data too closely.

Overfitting is a major issue encountered in Supervised Learning.

Example: The concept of the overfitting can be understood by the below graph of the linear regression output:

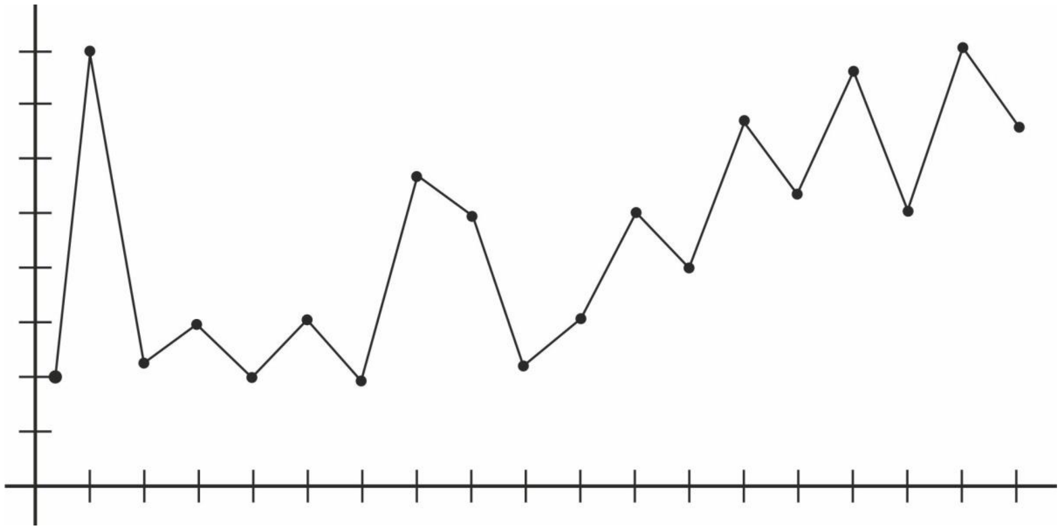


Figure 13: Concept of Overfitting

From the graph shown in Figure 13, it's evident that the model attempts to include every data point in the scatter plot. While this might seem effective at first glance, it isn't ideal. The objective of a regression model is to discover the best fit line, but in this case, no such line is found. As a result, this leads to errors in predictions.

Strategies to Avoid the Overfitting in Model

To prevent overfitting in a machine learning model, which can lead to poor performance, there are several strategies you can use:

- **Cross-Validation:** Use different parts of your data for training and testing to ensure the model works well with unseen data.

- **Training with More Data:** More data can help the model learn patterns better and not just focus on specific details of a smaller dataset.
- **Removing Features:** Simplify your model by removing unnecessary features that might be causing it to learn noise.
- **Early Stopping:** Stop training the model before it begins to learn the noise and inaccuracies in the data.
- **Regularization:** Apply techniques that penalize the model for being too complex, encouraging simplicity.
- **Ensembling:** Combine the predictions from multiple models to balance out individual errors and biases.

Underfitting

Underfitting happens when a machine learning model fails to capture the main trends in the data. This can occur if training is stopped too soon, leading the model to learn too little from the training data. Consequently, the model struggles to identify the most important patterns or trends. In underfitting, the model's learning is inadequate, which reduces its accuracy and leads to unreliable predictions. Typically, an underfitted model will have high bias but low variance.

Example: We can understand the underfitting using below output of the linear regression model:

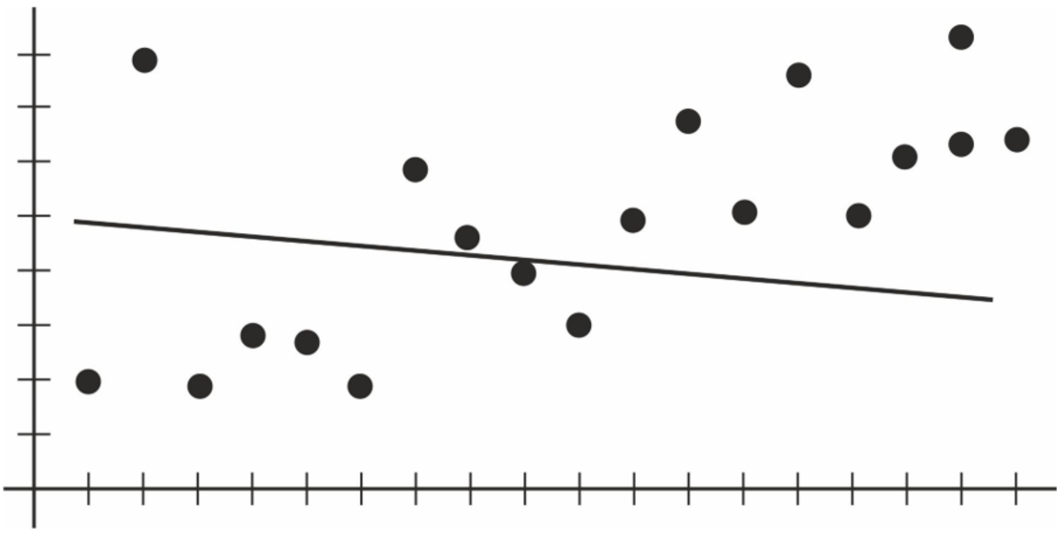


Figure 14: Concept of Underfitting

Looking at Figure 14, it's clear that the model fails to accurately represent the data points in the plot.

To avoid underfitting, you can:

- Extend the training duration of the model.
- Add more features to the model for a better understanding of the data.

Goodness of Fit

"Goodness of fit," a term from statistics, is a goal for machine learning models. It measures how closely a model's predictions match the actual values in the dataset. A well-fitted model falls between being underfitted and overfitted. Ideally, it should predict with zero errors, but achieving this is challenging in practice.

When training a model, errors in the training data typically decrease, and this trend is often mirrored in the test data. However, training for too long can lead to overfitting, where the model starts learning the noise in the data instead of just the relevant patterns. This causes errors in the test data to start increasing. The ideal point to stop training is just before these errors begin to rise, to ensure a good fit.

Two other methods to find this optimal point are using resampling techniques to estimate model accuracy and employing a validation dataset.

Bias and Variance in Machine Learning

Machine learning, a part of Artificial Intelligence, enables machines to analyze data and make predictions. However, these models can sometimes be inaccurate, leading to prediction errors often referred to as Bias and Variance. It's common to have some level of error in machine learning since there's usually a small difference between what the model predicts and the actual outcomes. The primary goal of machine learning and data science professionals is to minimize these errors to achieve more accurate results.

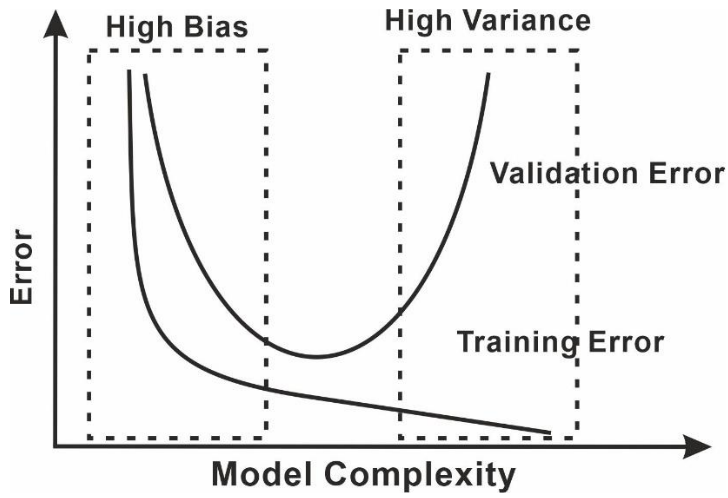


Figure 15: Bias, Variance, Errors

Errors in Machine Learning

In machine learning, errors are used to gauge how well an algorithm predicts outcomes for new, unseen data. The choice of a machine learning model often depends on its error rate. There are two main types of errors:

- **Reducible Errors:** These are the errors you can decrease to improve the model's accuracy. They are further divided into two categories: bias and variance.

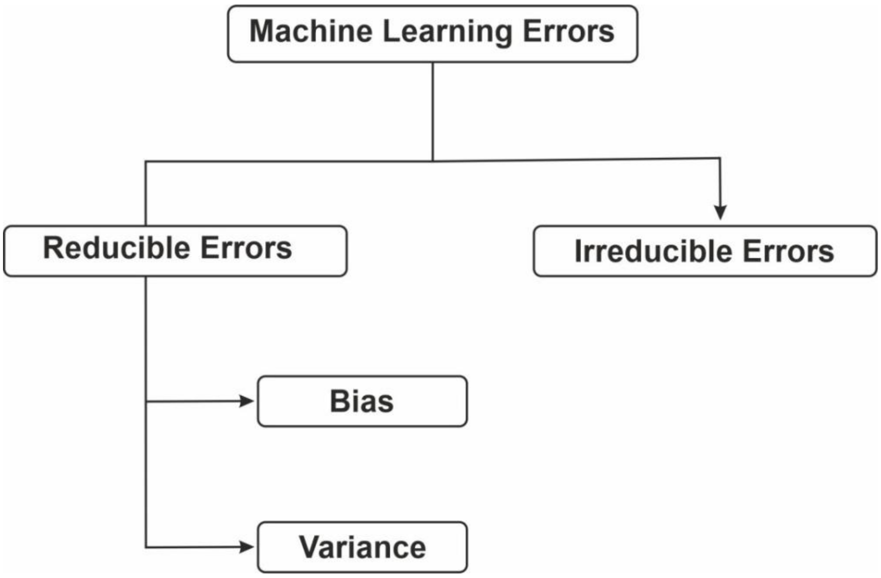


Figure 16: Machine learning Errors

- **Irreducible Errors:** These errors are inherent in the model and will always exist, no matter which algorithm is used. They are caused by unknown factors that cannot be accounted for or reduced.

Bias in Machine Learning

In machine learning, a model looks at data, identifies patterns, and makes predictions. During training, it learns these patterns and then uses them to make predictions on test data. Sometimes, there's a difference between the model's predictions and the actual or expected values. This difference is known as bias error, or simply bias. Bias happens when a machine learning algorithm, like Linear Regression, can't fully understand the true relationship between data points. This is often due to the assumptions the model makes to simplify the learning process. Models can have:

- **Low Bias:** These models make fewer assumptions about the target function's form, trying to be as flexible as possible to fit the data.
- **High Bias:** These models make more assumptions, which might cause them to miss important features in the dataset. They might not perform well with new data. Linear algorithms usually have high bias because they are simpler and learn faster.

For example, Decision Trees, k-Nearest Neighbours, and Support Vector Machines are algorithms with low bias. On the other hand, Linear Regression, Linear Discriminant Analysis, and Logistic Regression typically exhibit high bias.

Ways to Reduce High Bias

High bias often results from using a model that is too simple. Here are some methods to reduce high bias:

- Add more input features, especially if the model is underfitting.
- Reduce the regularization term.
- Switch to more complex models, like those including polynomial features.

Variance Error

Variance in machine learning refers to how much a model's predictions would change if it were trained on different datasets. In other words, it measures how a model's predictions vary from its expected value. Ideally, a model should show consistent performance across different training sets, indicating it understands the relationship between inputs and outputs well. Variance errors can be low or high:

- **Low Variance:** This means there's little change in the model's predictions when the training data changes.
- **High Variance:** This indicates significant changes in predictions with different training data.

A model with high variance learns in detail from the training data but struggles to perform well with new, unseen data. It performs well on the training data but often has high error rates on test data, leading to overfitting. High variance models are usually more complex.

Nonlinear algorithms, which are highly flexible in fitting models, tend to have high variance.

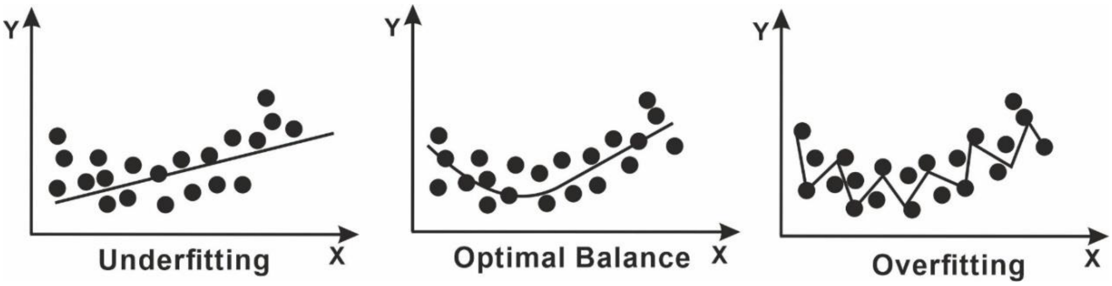


Figure 17: Underfitting, Optimal Balance and Overfitting

Examples of machine learning algorithms that typically have low variance include Linear Regression, Logistic Regression, and Linear Discriminant Analysis. On the other hand, algorithms like Decision Trees, Support Vector Machines, and K-Nearest Neighbour's often exhibit high variance.

Ways to Reduce High Variance

To reduce high variance in a model, you can:

- Decrease the number of input features or parameters, especially if the model is overfitting.

- Avoid using overly complex models.
- Increase the amount of training data.
- Raise the regularization term to prevent the model from fitting too closely to the training data.

Different Combinations of Bias-Variance

There are four possible combinations of bias and variances, which are represented by the below diagram:

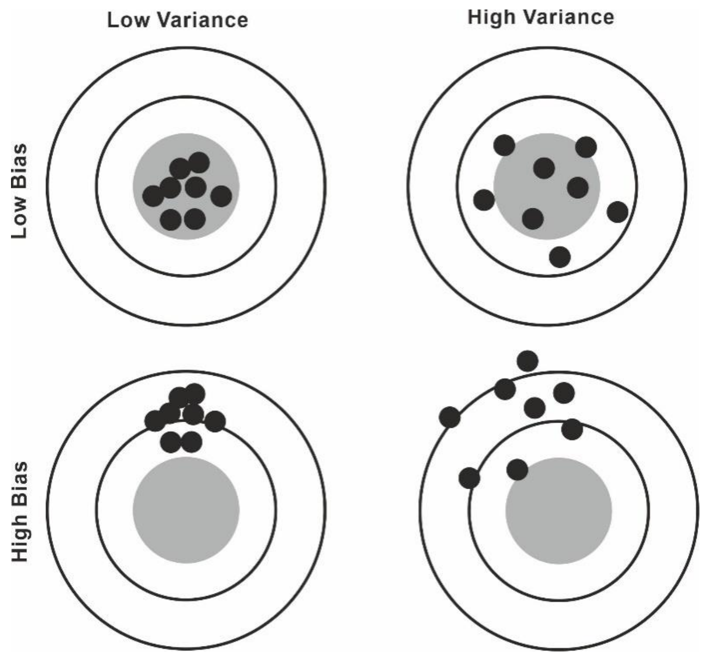


Figure 18: Combination of Bias and Variance

- **Low-Bias, Low-Variance:** This combination represents an ideal machine learning model, but achieving it in practice is challenging.
- **Low-Bias, High-Variance:** In this scenario, the model's predictions are accurate on average but inconsistent. This often happens when the model is trained with many parameters, leading to overfitting.
- **High-Bias, Low-Variance:** Here, predictions are consistent but generally inaccurate. This situation usually arises when the model doesn't learn effectively from the training data or uses too few parameters, resulting in underfitting.
- **High-Bias, High-Variance:** This combination leads to predictions that are both inconsistent and inaccurate on average.

Identification of High variance or High Bias

High variance can be identified if the model has:

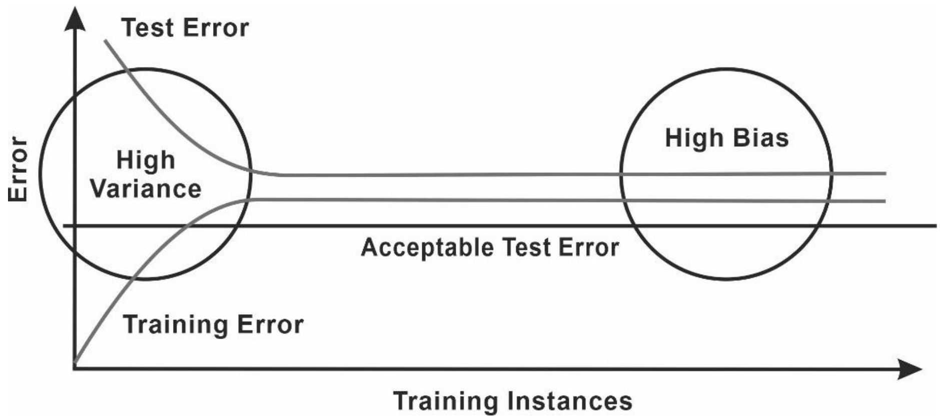


Figure 19: Identification of High Variance and High Bias

- Low training error and high-test error.

High Bias can be identified if the model has:

- High training error and the test error is almost similar to training error.

Bias-Variance Trade-Off

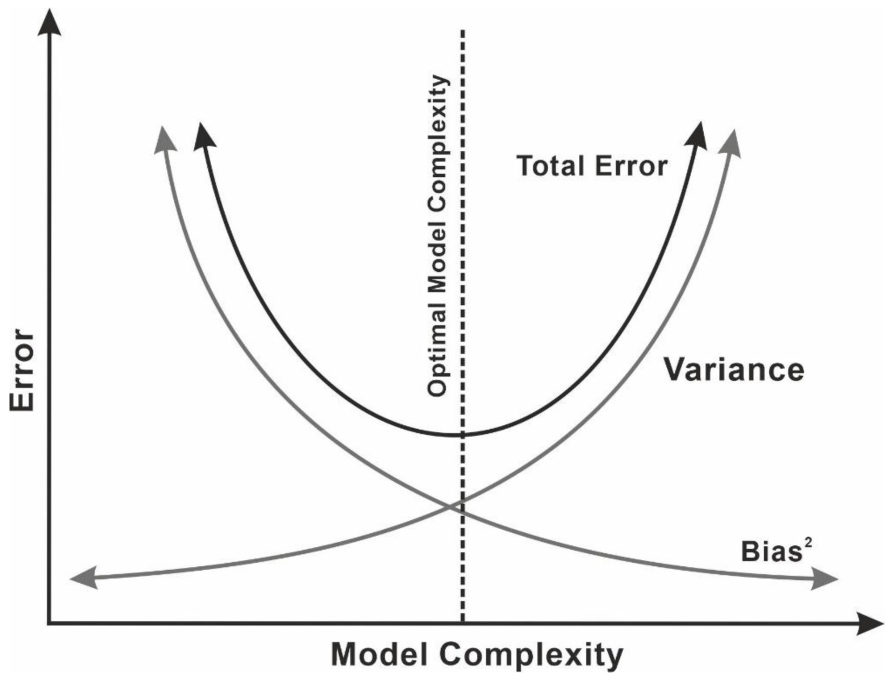


Figure 20: Bias Variance Trade Off

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model. If the model is very simple with fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias. So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as **the Bias-Variance trade-off.**

For a machine learning model to make accurate predictions, it ideally needs both low variance and low bias. However, achieving both simultaneously is challenging because bias and variance are interconnected:

- Decreasing variance often leads to an increase in bias.
- Decreasing bias tends to increase variance.

This interplay is at the heart of the Bias-Variance trade-off, a key concept in supervised learning. The goal is to create a model that not only learns effectively from the training data but also generalizes well to new, unseen data. However, this is difficult to achieve perfectly. A model with high variance might perform well on training data but could overfit, especially to noisy data. On the other hand, a model with high bias might be too simple, failing to capture important patterns in the data.

Therefore, the Bias-Variance trade-off involves finding the optimal balance between bias and variance errors, essentially hitting the 'sweet spot' for the best possible model performance.

V. CONCLUSION

In conclusion, we have established a fundamental understanding of machine learning's core aspects. We explored the crucial types of learning—supervised and unsupervised—and acknowledged their roles in propelling the field of artificial intelligence forward. With a focus on classification, we've seen how algorithms can be adeptly trained to categorize and analyse data.

The vital importance of various datasets, such as training, validation, and testing sets, has been underscored, highlighting their roles in creating strong AI models. We've tackled the challenges of overfitting and underfitting, gaining insight into their effects on model efficacy and examining ways to mitigate these issues.

As we conclude, machine learning is an ever-evolving field, characterized by constant learning and adaptation. The concepts and strategies discussed here are essential, practical tools for solving real-world problems through AI. This knowledge is meant to empower readers to apply machine learning principles effectively, thereby fostering innovation and contributing to advancements in the broad and dynamic domain of artificial intelligence.

VI. LET'S PRACTICE

Case Study: Email Filtering with Supervised Learning

Background: An email service provider aims to improve user experience by automatically filtering out unwanted spam emails. To achieve this, they decide to implement a machine learning-based spam filter.

Challenge: Spam emails are not only a nuisance but can also be dangerous, potentially containing phishing links or malware. The service provider must accurately distinguish between spam and legitimate emails (often referred to as 'ham') to protect users while ensuring important emails are not incorrectly classified as spam.

Approach: A supervised learning model is selected for this task. The model is trained on a labelled dataset where emails are pre-classified as spam or not spam. Features such as the frequency of certain words, the sender's email address, the presence of attachments, and the use of certain phrases that are commonly found in spam emails are used to train the model.

Implementation: The service provider compiles a large dataset of labelled emails. A variety of models, including Naive Bayes, Support Vector Machines (SVM), and neural networks, are trained and validated against a validation dataset. The best-performing model on the validation set is then tested using a separate test dataset to ensure that it can generalize to new, unseen emails.

Outcome: The chosen model demonstrates high accuracy in classifying spam and not spam emails. It is deployed as a part of the email service's pipeline. Regular updates and retraining sessions are scheduled to adapt to evolving spam tactics.

Questions for Discussion

1. What features were most indicative of spam in the emails, and how were they selected?
2. How was the training data collected and labelled, and how did you ensure it was representative of actual email traffic?
3. Which supervised learning model performed the best during validation, and what were the performance metrics?
4. How did the model handle edge cases, such as marketing emails or newsletters that users might have subscribed to?
5. What measures were taken to update the model over time, and how often was retraining required to maintain high accuracy?
6. How can a supervised learning model be used to classify emails into 'spam' and 'not spam'?

Case Study: Customer Segmentation Using Unsupervised Learning

Background: A retail company seeks to better understand its customer base to tailor marketing strategies and improve customer service. The company has collected a variety of data on customer purchase history, demographic details, and browsing behavior.

Challenge: The Company’s diverse range of products and customers makes it difficult to manually segment the market. They require an automated method to divide customers into distinct groups based on similarities in their shopping patterns and preferences.

Approach: Unsupervised learning, specifically clustering algorithms, is chosen to tackle this problem. The aim is to segment customers into clusters that exhibit similar characteristics without prior labelling.

Implementation: The Company compiles a comprehensive dataset, including variables like age, gender, purchase frequency, average spending, and product categories purchased. Clustering algorithms such as K-Means, Hierarchical Clustering, and DBSCAN are applied to this dataset. The number of clusters is determined using methods such as the elbow method for K-Means and silhouette analysis.

Outcome: The unsupervised learning model successfully segments the customer base into distinct groups. These segments reveal insightful patterns, such as a group of high-value customers who purchase frequently and another group of occasional shoppers who buy only sale items. The marketing team uses these insights to craft targeted campaigns, while the product team adjusts inventory based on the preferences of each segment.

Questions for Discussion

1. What pre-processing steps were taken to prepare the data for clustering algorithms?
2. How did you determine the optimal number of customer segments?
3. Which clustering algorithm yielded the most meaningful segmentation, and why was it chosen over others?
4. How did the company ensure that the clusters were actionable and relevant to their marketing strategies?
5. What were some of the key characteristics that differentiated the customer segments?
6. How does the company plan to use these customer segments to influence business decisions going forward?

Case Study: Implementing Predictive Maintenance with Supervised Learning

Background: An industrial equipment manufacturer aims to integrate a predictive maintenance system into their machinery. The goal is to predict potential equipment failures before they occur, thus minimizing downtime and maintenance costs.

Challenge: Predicting equipment failure is complex, as it must account for various factors that could contribute to a machine's malfunction. The manufacturer needs to determine the types of data required for accurate predictions and develop a model that can process this data to predict failures effectively.

Approach: The Company decides to use supervised learning, where a model is trained on historical data that includes instances of equipment failures and normal operations. The data collected comprises machine operational parameters, usage patterns, maintenance records, and failure histories.

Implementation: The team collects a dataset of sensor readings from machinery during operation, maintenance logs, and records of past failures. Features such as temperature, vibration, operating hours, and error codes are included. A variety of machine learning models, including Random Forest, Gradient Boosting Machines, and Neural Networks, are trained on this data. The models are then validated using a cross-validation approach to ensure they can predict failures reliably.

Outcome: The supervised learning model that best predicts upcoming failures is deployed within the machinery's operating system. The system provides real-time alerts to the maintenance team, allowing for proactive repairs. This results in a marked reduction in unplanned downtime and more efficient maintenance scheduling.

Questions for Discussion

1. What specific types of sensor data were most indicative of impending equipment failure?
2. How was the historical data labelled to distinguish between normal operation and failure events?
3. Which supervised learning model provided the highest accuracy and reliability in predicting equipment failures?
4. What challenges were encountered when collecting and preprocessing the data for training the models?
5. How were the models tested to ensure they could generalize well to new, unseen data?
6. What procedures were established for updating the model as new failure data becomes available?

Case Study: Community Detection in Social Networks Using Unsupervised Learning

Background: A social media platform is interested in understanding the natural groupings or communities within its user base to enhance content relevance and advertising targeting. The platform has access to vast amounts of user interaction data but lacks explicit labels for community membership.

Challenge: Identifying communities within a social network is complex due to the vastness of data and the dynamic nature of social interactions. The platform needs a method to discern these communities based on patterns of interactions without predefined labels.

Approach: The company employs unsupervised learning, specifically clustering algorithms that can detect communities based on user interactions such as likes, comments, shares, and messaging patterns. Algorithms considered for this task include K-Means, Hierarchical Clustering, and DBSCAN.

Implementation: The team compiles a dataset consisting of anonymized user interaction data. Features such as frequency of interactions, strength of connections (e.g., number of mutual friends), and similarity in content engagement are extracted. The clustering algorithms are applied to this data to identify distinct user communities.

Outcome: The unsupervised learning approach effectively reveals distinct communities within the social network. These communities show high internal interaction rates and are characterized by shared interests or demographics. The insights gained are used to improve the user experience by personalizing content and optimizing ad targeting.

Questions for Discussion

1. What methods were used to ensure user privacy while conducting the analysis?
2. How did the platform determine the number of communities, and what validation methods were used to assess the quality of the clustering?
3. Which clustering algorithm was most effective in identifying meaningful communities, and what were its key parameters?

4. How did the characteristics of identified communities align with known user demographics and behavior patterns?
5. How will the social media platform utilize these insights to enhance user engagement and business outcomes?
6. What steps will be taken to update and maintain the community detection system over time, considering the evolving nature of social networks?

VII. PRACTICAL EXERCISES

1. Building a Classifier (Supervised Learning)

- **Exercise:** Use a dataset like the Iris dataset to train a classifier to predict the species of an iris plant. Split the data into training and test sets, train a model, and evaluate its performance.

- **Questions**
  - How did you preprocess the data?
  - Which classifier did you choose and why?
  - How did you evaluate the model's performance?

2. Market Basket Analysis (Unsupervised Learning)

- **Exercise:** Perform a market basket analysis on a dataset from a grocery store. Identify frequently purchased items and itemsets.

- **Questions**
  - What patterns did you find in the data?
  - How can these patterns be used to make business decisions?
  - What rules did you derive from the analysis, and what are their confidence and support?

For each case study and practical exercise, the learner is encouraged to consider the full pipeline of machine learning, from data preprocessing, choosing the right model, training the model, and evaluating its performance using appropriate metrics. Additionally, for unsupervised learning tasks, the focus should be on interpreting the results and understanding the actionable insights that can be derived from the models.

REFERENCES

[1] Liu, H., Estiri, H., Wiens, J., Goldenberg, A., Saria, S., & Shah, N. (2019). AI model development and validation. *Artificial Intelligence in Health Care: The Hope, the Hype, the Promise, the Peril*, 119.
[2] Ayodele, T. O. (2010). Types of machine learning algorithms. *New advances in machine learning*, *3*, 19-48.
[3] Ghori, K. M., Abbasi, R. A., Awais, M., Imran, M., Ullah, A., & Szathmary, L. (2019). Performance analysis of different types of machine learning classifiers for non-technical loss detection. *IEEE Access*, *8*, 16033-16048.
[4] Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, *2*(3), 160.
[5] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval* (pp. 21-49). Berlin, Heidelberg: Springer Berlin Heidelberg.

[6] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). Overview of supervised learning. *The elements of statistical learning: Data mining, inference, and prediction*, 9-41.
[7] Liu, B., & Liu, B. (2011). Supervised learning. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*, 63-132.
[8] Niculescu-Mizil, A., & Caruana, R. (2005, August). Predicting good probabilities with supervised learning. In *Proceedings of the 22nd international conference on Machine learning* (pp. 625-632).
[9] Barlow, H. B. (1989). Unsupervised learning. *Neural computation*, *1*(3), 295-311.
[10] Ghahramani, Z. (2003). Unsupervised learning. In *Summer school on machine learning* (pp. 72-112). Berlin, Heidelberg: Springer Berlin Heidelberg.
[11] Celebi, M. E., & Aydin, K. (Eds.). (2016). *Unsupervised learning algorithms* (Vol. 9, p. 103). Cham: Springer.
[12] James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). Unsupervised learning. In *An Introduction to Statistical Learning: with Applications in Python* (pp. 503-556). Cham: Springer International Publishing.
[13] Golbraikh, A., Shen, M., Xiao, Z., Xiao, Y. D., Lee, K. H., & Tropsha, A. (2003). Rational selection of training and test sets for the development of validated QSAR models. *Journal of computer-aided molecular design*, *17*, 241-253.
[14] Bennin, K. E., Keung, J., Monden, A., Kamei, Y., & Ubayashi, N. (2016, June). Investigating the effects of balanced training and testing datasets on effort-aware fault prediction models. In *2016 IEEE 40th annual Computer software and applications conference (COMPSAC)* (Vol. 1, pp. 154-163). IEEE.

