Fundamentals of Machine Learning:- Unit 1

(1) What is machine learning?

(2) Why we " "

Introduction to machine learning:-

* Before ML, traditional programming was the primary way we developed applications.

* In traditional programming, a programmer would explicitly write rules & logic for every possible scenario.

* Means, in such applications, programmers used to write if, else conditions to process the data.

* Now, think of a spam filter whose job is to move the incoming email messages to a spam folder if it is spam. What can you do?

* We can make up a blacklist of words that would result in an email being marked as spam. This will be an example of using an expert designed rule system to design an intelligent application.

* But manually crafting decision rules is feasible for some applications, particularly those in which humans have a good understanding of the process to model.

* But it has got certain disadvantages.

  (1) The logic written is specific to a single domain & task. Changing the task, even slightly might require complete rewriting of the whole system.

  (2) Designing rules requires a deep understanding of how a decision should be made by an human expert.

* One example where handcoded approach will fail is in detecting ~~faces~~ objects in images. (say identifying cars in photo)

* If we were to follow traditional programming, to identify cars in an image, we need to write rules that account for every possible feature of a car - shape, color, size, position, lighting conditions etc. Each of these rules need to be coded. For ex,

Rule 1:- If the object is rectangular & has 4 wheels, its a car. (How do we write it? In terms of pixels. The way in which pixels are perceived by computer is different from how we humans perceive a car)

Rule 2:- If the object has headlights, its a car.

Rule 3:- If the object is moving at a high speed, it's probably a car on the road.

variations: cars in different positions, lighting, angles, colors, occlusions (like a tree partly blocking the car), reflections & so on. It is nearly impossible to account for all these variables with just a set of rules. Even a small change in the scenario could break the program.

* This difference in perception of pixels & representation makes it impossible for humans to come up with a good set of rules to describe what constitutes a car in a digital image.

* ML solves this problem by learning from vast amounts of labelled data (images with known labels). Here ML model learns the patterns in the data.

* We train the model on thousands or millions of images of cars (& non-cars).

* The model learns (using algorithm) which features are important to distinguish cars from other objects.

* Over the period of time, it even recognizes cars in a variety of conditions - angles, lighting, occlusions & so on - without having edge cases explicitely programmed.

* ML comes to rescue even when in some cases where rules & patterns change over time. For example, user preferences & Market conditions may evolve. With traditional approach, we have to continuously update the program. ML models can learn & adapt as new data becomes available.

* Sometimes we need to find solutions in an unstructured environment. (like playing a game, diagnosing disease, or recommending products). Traditional approach fails as it is not feasible & level of adaptibility is not achievable.

* When there is a need to process & analyze vast amounts of data quickly & accurately, ML is the best option as it can detect complex patterns that may not be obvious to humans.

* We will understand how humans learn.

But the real world is much messier. There are countless variations: cars in different positions, lighting, angles, color, occlusions (like a tree partly blocking the car), reflections & so on. It is nearly impossible to account for all these variables with just a set of rules. Even a small change in the scenario could break the program.

*. This difference in perception of pixels & representation makes it impossible for humans to come up with a good set of rules to describe what constitutes a car in a digital image.

* ML solves this problem by learning from vast amounts of labelled data (images with known labels). Here ML model learns the patterns in the data.

*. We train the model on thousands or millions of images of cars (& non-cars).

*. The model learns (using algorithm) which features are important to distinguish cars from other objects

*. Over the period of time, it even recognizes cars in a variety of conditions - angles, lighting, occlusions & so on - without having edge cases explicitely programmed

*. ML comes to rescue even ~~when~~ in some cases where rules & patterns change over time. For example, user preferences & market conditions may evolve. With traditional approach, we have to continuously update the program. ML models can learn & adapt as new data becomes available.

* Sometimes we need to find solutions in an unstructured environment. (like playing a game, diagnosing disease, or recommending products). Traditional approach fails as it is not feasible & level of adaptibility is not acheivable.

*. When there is a need to process & analyze vast amounts of data quickly & accurately, ML is the best option as it can detect complex patterns that may not be obvious to humans.

* We will understand how humans ~~that~~ learn.

What is learning?

* How do we, humans learn?

* Learning is the process of acquiring knowledge or skills through study, experience or teaching. It involves the ability to understand, retain & apply new information/ behavior.

* After certain period of time, humans realize that the simple ability to remember things is not intelligence. But converting data stored in the memory into knowledge & applies them to solve a problem in real-world.

Ex:- * Reasoning & adjust to different situations.

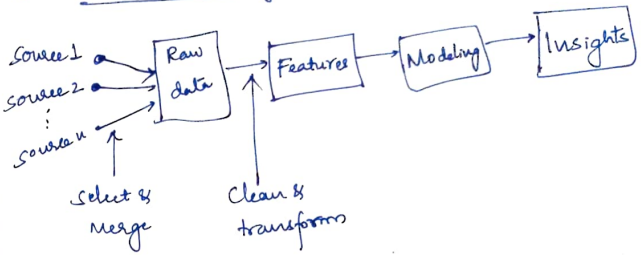
1. Academic learning → laws of physics.
2. Skill acquisition → Playing guitar by practice, observing others & taking music lessons.
3. Social learning → Social norms & behaviors by observing Parents & peers.
4. Professional Development → Employee learning new software
5. Adaptive learning → Learns new culture & language by living in a foreign country.

Now what is ML?

ML is a branch of AI that focuses on developing models & algorithms that let computers learn from data & improve from previous experiences without being explicitly programmed for every task

* We wish to "Program" computers so that they can learn from input available to them.

* Machine Learning Pipeline:-



data:- Observations of real-world phenomena.

Ex. 1. Stock market data → observations of daily stocks prices, announcements of earning by individual company & opinion articles from experts.

2. Biometric data

All these observations are merged. But it is Messy (noise & missing pieces)

Tasks:- Why do we collect data?

To get answers like

1. Which stocks I should invest in 2. How healthy lifestyles?

Due to noise, wrong, missing pieces, data needs to be cleaned & also raw data is not often numeric. So, features come in to picture.

A feature is a numeric representation of

raw data.

Model evaluation:- Features & Model sit between raw data & desired insights. We pick not only the model, but also features. Because choice of one affects the others.

A model that predicts stock prices might be a formula that maps company's earning history, past stock prices & industry to the predicted stock price.

Music ex.

Popular examples of Machine learning:-

1. Virtual Personal Assistants:- Alexa, Siri & Google Assistant use ML to understand & respond to voice commands. Over the period of time, they learn from interactions to give more accurate & personalized responses. ~~over time~~.

2. Movies / Music recommendation systems:- Netflix, Amazon & Spotify use ML to recommend movies & music based on past preferences & behavior.

3. Image recognition:- Facebook's automatic tagging feature uses image recognition to identify & tag people by learning from previous tags & facial features.

4. Self-driving cars:- Self-driving cars use ML to analyze data from sensors & cameras to navigate roads, detect obstacles & make driving decisions.

5. Fraud detection:- Banks & credit card companies use ML to detect fraudulent transactions by analyzing patterns in spending behavior. It tracks unusual behavior & tags them for further investigation.

6. Product recommendations:- Amazon, flipcart use ML to suggest products based on browsing history & purchase behavior.

Case study 1:- Understanding ML in the context of e-commerce (Amazon, flipkart, ebay etc)

They use ML algorithms to analyze human behavior, understand preferences & deliver suggestions.

Step 1:- Visits Amazon website & explores Dell laptops, models, specifications & pricing

Step 2:- Amazon's ML tracks user browsing activity, gets the details of Dell laptop models viewed, time spent on each model & interactions like adding-to-cart, OR favorites.

Step 3:- ML analyzes collected data to identify preferences & patterns.

Step 4:- Builds user profile that [illegible] preferences for Dell laptops, budget constraints, desired features & past purchase history.

Step 5:- Using this data, & ML models, Amazon's generative recommendation system generates suggestions & recommending other Dell laptops & accessories that align with the user's preferences.

Step 6:- Displaying recommendations.

Step 7:- As the user continues to engage with recommended products, the system collects feedback on clicks, views & purchases. This feedback loop refines the ML for more accurate recommendations.

Step 8:- Continuous learning & improvement.

Features of ML

(1) Adaptability:- ML models can adapt to your data & changing environments making them versatile for various applications. Ex:- In autonomous vehicles, ML algorithms continuously learn from real time sensor data to adapt to different driving conditions & improve decision-making.

(2) Automation:- ML enables automation of tasks by allowing systems to learn from data & make decisions without explicit programming instructions.
Ex:- e-mail spam detection

(3) Scalability:- ML algorithms are designed to handle large volumes of data & scale with it. As more data becomes available, ML models can update their predictions & decisions improving their accuracy.
Ex:- Feeds & recommendations

(4) Personalization:- ML enables personalized experiences by tailoring recommendations & content based on individual preferences.
Ex:- Spotify

(5) Predictive Analytics:- ML algorithms makes predictions & forecasts based on historical data
Ex:- Predictions of stock market trends, detecting fraud & weather forecasting.

(6) Continuous Improvement:- As new data becomes more available, ML algorithms improve their performance. Ex:- Alexa, Siri

(7) Decision Making:- ML can take accurate decisions with minimum human intervention.

Ex:- Diagnosing diseases & recommending treatment plans based on patient's data & historical data.

(8) Pattern recognition:- ML excels at recognizing patterns & regularities in data.

Ex:- Facial recognition.

(9) Efficiency:- ML can enhance efficiency by automating repetitive tasks & optimizing process.

Ex:- Manufacturing (ML can analyze production data to predict equipment failure & improve productivity)

Traditional Vs Machine Learning approach examples:-

Spam email filter example:- Spam emails are unwanted messages with advertisements, scams, or malicious content.

(1) Traditional approach:- Specific words/phrases
↳ 4U, credit card, free, exciting & amazing → subject's line, sender's name & email body.
↳ develop detection algorithm for each pattern
↳ test it until it reaches satisfactory level of deployment.

If spammers realize their emails with "4U" are blocked, they switch to "For U" to surpass filter. Continuous update to the spam filter are necessary to reframe the rules.

(2) ML approach:- We train the algorithms on large datasets of spam & non-spam emails to automatically learn patterns & features for filtering.

A spam filter based on ML techniques automatically notices that "For U" has become unusually frequent in spam flagged by users & then it starts flagging them without our intervention.

Why use ML?

(1) Automated learning:- Extracting patterns & insights from data, reducing manual effort.

(2) Adaptibility:- ML models can adapt to new data & changing environments to effectively predict in dynamic environments.

(3) Scalability:- ML can handle large & complex datasets & hence it is suitable for various applications.

(4) Enhanced decision Making:

(5) Innovation & efficiency → ML drives innovation by developing intelligent system.

(6) Streamlined & Accurate processing :-

(7) Complex problem Resolution :- NLP complex tasks.

(8) Flexibility :- TP vs ML

(9) Discovery & learning

(10) Adaptation to fluctuating data environments.

Machine learning problem :-

* A ML problem refers to a specific task | objective that can be addressed using ML techniques or algorithms.
* We need to develop a model that can learn from data, make predictions or perform tasks without being explicitly programmed.

A ~~Machine~~ problem that can be solved using ML involves

(1) Task or objective (T) :- Defining what needs to be achieved such as classification, regression, clustering or pattern recognition

(2) Performance Metric (P) :- Determining how the performance of ML model will be evaluated such as accuracy, precision, recall or F1 score

(3) Training experience (E) :- Providing the model with a dataset of input features & corresponding labels to learn from during training process.

Ex: (1) Handwriting recognition in image

Task (T) :- Recognizing & classifying handwritten words within images.

P :- % of words correctly classified.

E :- Dataset of handwritten words with their corresponding classifications.

(2) A robot's driving learning problem

Task (T) :- Driving on highways using vision sensors.

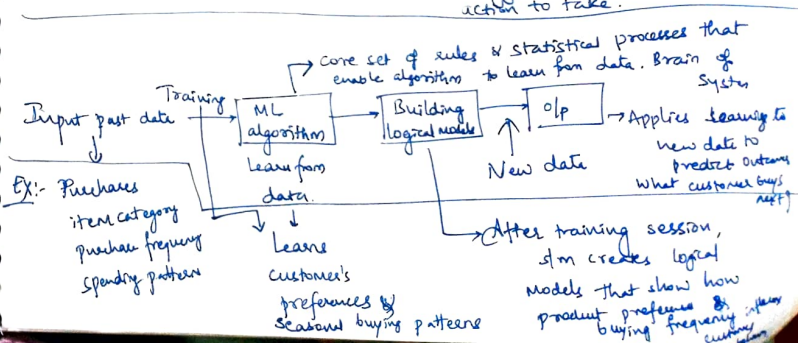
P :- Average distance travelled before an error.

E :- A sequence of images & steering commands recorded while observing a human driver.

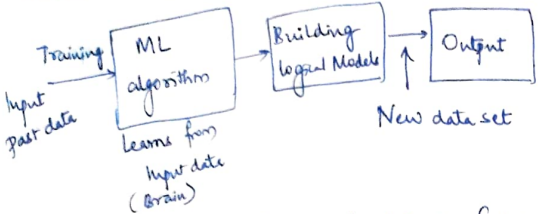
An ML f° can be descriptive → s/m uses the data to explain what happened.

Predictive → s/m uses the data to predict what will happen.

prescriptive → s/m uses the data to make suggestions about what action to take.



Machine learning algorithm's operation (Block diagram)



① Input past data:- Historical data is fed in to ML algorithm for training purposes.

Ex: System is fed with historical data on customer purchase including items bought, purchase frequency & spending patterns

② Training:- The algorithm undergoes training phase where it learn from input data.

Ex:- During training phase, the algorithm learns from the past purchase data to identify trends such as popular products, seasonal buying patterns & customer preferences

③ ML algorithm:- This refers to core set of rules & statistical process that enable the algorithm to learn from data.

Ex:- Rules & statistical processes that enable the system to analyze customer purchase data & make predictions about future buying behavior.

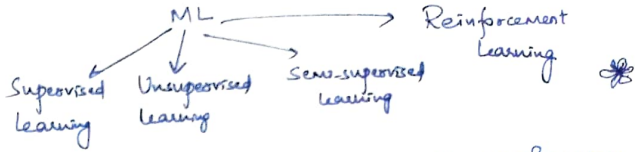
④ Building logical models:- After training phase, the system constructs logical models based on the learned patterns & relationships in the data. These models represent the insights gained from training data.

Ex:- Based on the learned patterns & relationship in the training data of customer purchase data, s/m creates logical models that show how product preferences & purchase frequency influence customer behavior.

⑤ O/p:- When new data is given to the system, logical models created during training are used to predict outcomes.

Eg:- What user is likely to purchase next & helps retail store make personalized recommendations.

Types of ML



① Supervised Learning:- To observe & control the performance of a task. Supervised means working in the presence of Supervision.

* Here the machine is first trained by providing labelled data & then the machine is allowed to predict the outcomes. It is learning by example

Def:- It is a type of machine learning where the algorithm learns to map input data to the correct o/p by being trained on labeled examples.

The training data consists of input-output pairs. I/p data is accompanied by corresponding correct o/p / labels

Here the algorithm has to learn a mapping f? from the i/p to o/p so that it can make accurate predictions on unseen data.

Key Components of Supervised Machine Learning:

① **Labeled data**:- The training data is labeled which means each i/p data point is associated with the correct o/p or target label.

Ex. In Spam filter, each email is labeled as either spam or not spam.

② **Training Phase**:- During the training phase, the algorithm uses the labeled data to learn the relationship bet? input features & the corresponding o/p labels. The algorithm adjusts its internal parameters based on the training data to minimize error bet? its predictions & true labels.

③ **Testing Phase**:- Once the model is trained, it is tested on a separate test dataset that contains new data

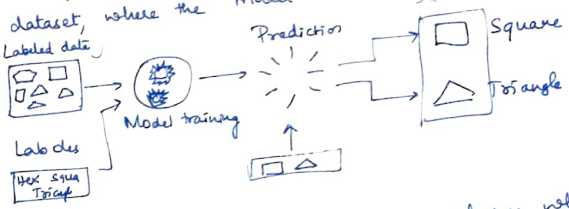
④ **Prediction**:- Once the model is trained, it can be used to make predictions on new, unseen data. The model takes the i/p data & uses learned mapping f? to predict the corresponding o/p label.

⑤ **Evaluation**:- The performance of supervised learning model is evaluated by comparing its predictions on a separate test dataset with the true labels. (accuracy, precision, recall, F1)

L4 - MERN

How supervised learning works?

In supervised learning, models are trained using labelled dataset, where the model learns about each type of data



* We have dataset of different types of shapes which includes triangles, squares, rectangles, & polygon.
* We need to train our model for each shape.

① **Labeled data**:- ① If the given shape has 4 sides & all sides are equal, it will be labelled as a square
② 3 sides, triangle
③ 6 equal sides, hexagon.

② **Training phase**:- During training phase, the model learns the patterns & relationships bet? i/p features (no. of sides) & corresponding class labels.

③ **Testing phase**:- It is tested on separate dataset. It uses mapping f? from training phase

④ **Prediction**:- For new shape in test set, prediction happens.

⑤ Evaluation

Spam/not spam → ① Dataset ② Features like subject line (Congratulations! prize, click here to claim etc, suspicious address)
③ learn feature & labels ④ Algorithm & train model

Types or Categories of supervised ML Algorithms

① Classification Algorithms ② Regression Algorithms

① Classification Algorithm:- Classification algorithm is used when the o/p variable is categorical, meaning it falls into distinct classes or categories. Here the goal is to predict the class label of new data points based on the patterns learned from the training data.

Ex: Classifying emails as spam/not spam.
Predicting whether patient has high risk of heart disease or not.

Following are classification algorithms' lists.

1. K-Nearest Neighbors (KNN)
2. Random forest
3. Logistic Regression
4. Decision trees
5. Naive Bayes
6. Support Vector Machines (SVM)

Ex:

① Spam filtering:- Naive Bayes → based on the presence of some keywords / features in email content.

② Customer Churn Prediction:- Churn (cancel subscription or service): Random forest → based on historical data & customer behavior.

③ Sentiment analysis: SVM to determine sentiment (+ve, -ve & neutral) after analyzing text data expressed in reviews, social media post or customer feedback

④ Image Classification:- Classifying images to pre-defined categories like animals, objects or scenes based on visual features. CNN → learns hierarchical features.

⑤ Medical diagnosis:- Predicting the presence or absence of disease based on patient symptoms, medical history & test results. Decision trees

⑥ Fraud detection:- Identifying fraudulent transactions to prevent financial losses.

Logistic regression.

② Regression algorithms:- Regression algorithms are used when the relationship bet? i/p variables & continuous o/p variables needs to be predicted. Here the goal is to predict a continuous value based on i/p feature.

Ex: Predicting price of house based on its size, location & amenities

Forecasting sales

Following are regression algorithms:

1. Random Forest regression
2. Decision tree regression
3. Support vector regression (SVR)
4. Polynomial regression
5. Lasso
6. Ridge regression

Ex:-

① House price prediction:- Features like area, no. of bedrooms, location etc.

Algo:- Linear regression → uses linear eq? to model the relationship bet? input features & house price

② Stock price forecasting:- based on historical stock data, market trends & other relevant factors.

Algo:- ARIMA(AutoRegressive Integrated Moving Average) or prophet.

③ Demand forecasting:- Algo:- Random forest regression

④ Temperature prediction:- Based on historical weather data & meteorological factors.

Algo: SVR.

⑤ Sales Revenue Prediction:- historical sales data, marketing campaigns & economic indicators.

Algo:- LightGBM.

Applications of supervised learning

1. Image classification:- Recognizing objects, faces & features within images.
2. Natural Language Processing:- Extracting insights from text, including sentiment analysis, entity identification & relationship extraction.
3. Speech recognition:- Spoken language in to text for transcription purposes.
4. Recommendation Systems:- Personalized recommendations to users is given based on their preferences.
5. Predictive analytics:- It is used for forecasting outcomes such as sales figures, customer retention rates & stock market trends.
6. Medical Diagnosis:- Helps in identifying diseases & Medical conditions from patient data.
7. Fraud detection:- Identifies & flags ~~fo~~ fraudulent transactions
8. Autonomous vehicles:- Enables vehicles to detect & respond to objects in their surroundings.
9. Email Spam Detection:- Classifying email as spam/not Spam
10. Credit Scoring
11. Gaming:- Analyzes players behavior, character recognition, & NPC creation in gaming environments.
12. Customer Support:- Automates tasks in customer service for improved efficiency.
13. Weather Forecasting:- Forecasting weather based on previous weather & meteorological details.
14. Sports Analytics:- Analyzes players performance, predicts game outcomes & optimizes strategies for sports team.

Advantages

1. Predictive accuracy:- Learning patterns from labeled training data, they can make accurate predictions
2. Interpretable Results:- Insights into relationship bet? i/p features & target variables.
3. Evaluation Metrics:- Clear EM enable easy model performance assessment.
4. Feature importance:- Identifies most imp features that contribute to prediction outcomes.
5. Generalizations:- Supervised models can generalize well to unseen data, making them suitable for real world appl?s.

Disadvantages

① Data dependendancy

② Overfitting - Dataset may have captured noise or irrelevant patterns which leads to poor generalizations on unseen data. → boat example

③ Limited Flexibility :- Models are constrained by the features & labels provided in training data, limiting their ~~adap~~ ability to adapt to new/unseen patterns

Unsupervised Learning :-

* Word unsupervised means "working not under the supervision".

* Here the machine is provided with the set of data that has not been labelled, classified or categorized & the algorithm needs to act on that data without any supervision.

* The primary goal of unsupervised learning is to discover hidden patterns, similarities or clusters within the data which can then be used for various purposes such as data explorations, visualizations, dimensionality reduction etc. without need of labeled data

* Unsupervised learning is a type of ML where the algorithm learns to identify patterns & relationships in data without being explicitely trained on labeled examples.

Key Components of Unsupervised ML :-

① Unlabeled data - Here the input data points do not have corresponding o/p labels. The algorithm's task is to discover the underlying structure or patterns in the data on its own.

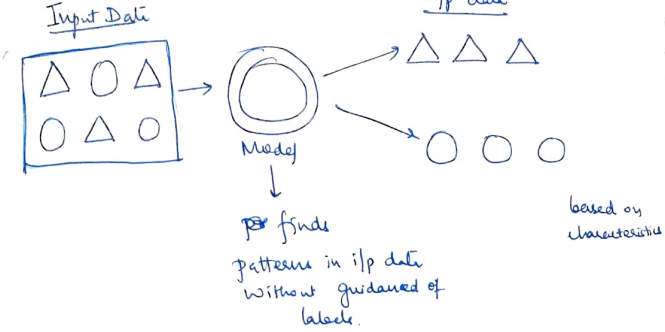
② Clustering - Clustering is a common technique in unsupervised learning where the algorithm groups similar data points together based on their features / characteristics. Clustering algorithms aim to partition data into clusters such that data points within the same clusters are more similar to each other than to those in other clusters.

③ Dimensionality reduction - Dimensionality reduction techniques is a used in unsupervised learning to reduce the no. of features in the data while preserving the important info. This helps not only in removing noise/redundant info but also in visualizing high-dimensional data.

④ Anamoly detection:- Unsupervised algorithms can be used for anamoly detection where the algorithm identifies data points that deviate significantly from the norms or majority of data as compared to unexpected behavior

⑤ Association Rule learning - It is another technique in unsupervised learning that discovers interesting relationships or associations bet^n variables in large dataset. It is used to identify patterns in consumer behavior.

Ex



In unsupervised learning, o/p is not predicted labels but insights from data.

Types or Categories of Unsupervised Machine learning

3 types

① Clustering ② Dimensionality reduction algorithms

③ Association algorithms

① Clustering:- Clustering groups similar data points together based on their features / characteristics. Clustering algorithms aim to partition data in to clusters such that data points within same cluster are more similar to each other than to those in other clusters.

This algorithm learns to identify patterns in the data & assign data points to clusters without any predefined class labels.

Ex: Grouping customers based on purchasing behavior.

List of algorithms

① K-means clustering

② Hierarchical Clustering

③ Gaussian Mixture Models

④ DBSCAN (Density-based spatial clustering of applications with noise)

Ex:- ① Customer Segmentation: Identifying different customer groups based on customers purchasing behavior & preferences to design marketing strategies → K-means is u

② Image segmentation:- Separating an image into different regions for image analysis & object recognition. ~~Clustering~~ Hierarchical clustering → image segmentation for grouping pixels / features that exhibit ~~III~~ char

(3) Anomaly Detection:- Identifying unusual pattern that do not conform to expected behavior such as fraud or n/w intrusions. DBSCAN - algorithm

(2) Dimensionality reduction algorithms:- It is used to reduce the number of features in the dataset without while preserving important information. This helps in not only reducing noise/redundant data points but also in visualizing high-dimensional data.

Algorithms

(1) Principal Component Analysis (PCA) → feature selection
(2) t-Distributed Stochastic Neighbor Embedding (t-SNE) → Data visualization, reducing high-dimensional to 2 or 3 dimensions.
(3) Singular Value Decomposition (SVD)
(4) Independent Component Analysis (ICA)

(3) Association algorithms:- It is used to detect interesting relationships/associations bet? input features in large datasets. Ex: Identifying patterns in customer's behavior. It is extremely useful in scenarios where manual pattern recognition is impractical.

Appl?: Bio~~m~~informatics, text mining, recommendation systems.

Algorithms

(1) Apriori Algorithm
(2) FP-growth "
(3) Eclat.

Ex:

(1) Market basket analysis:- Apriori algorithm is used to identify products often purchased & optimize marketing strategies.

(2) Fraud detection:- Discovers unusual patterns in the data that may indicate fraudulent behavior. Sequential pattern discovery using equivalence clause SPADE algorithm is used.

(3) Recommendation Systems:- Collaborative filtering approaches (Matrix factorization or neural network based recommendation algorithms) are used.

(4) Cross Marketing:- Apriori or FP-growth are used to find associations bet? product categories to drive cross-promotional strategies

(5) Catalog design:- Arranges items in catalog to maximize the discovery of associated items. Eclat algorithm

Applications of unsupervised machine learning:-

1. Clustering:- Group similar data points into clusters
2. Anomaly detection:- Identify outliers or anomalies in data
3. Dimensionality reduction:- Reduce the dimensionality of data while preserving its essential information.
4. Recommendation system:- Suggest products, movies or content to users based on their preferences | historical behavior.
5. Image & video compression:- Reduce the amount of storage required for multimedia content.
6. Data Pre-processing:- Cleaning, imputation of missing values & data scaling.
7. Market basket analysis:- Discover associations bet^n products.
8. Genomic data analysis:- Identify patterns or group genes with similar expression profiles.
9. Image Segmentation:- Segment images into meaningful regions.
10. Community detection in social networks:- Identify communities or group of individuals with similar interests or connections.

Ex:- Customer data → age, income, purchase history & browsing behavior but there are no categories for these customers.

Using K-means, s/m can identify patterns & group customers like "frequent high-spending buyers", "occasional budget shoppers", "new customer" → Then personalized marketing strategy.

Advantages of Unsupervised ML

1. Discover hidden patterns:- Without need for labeled outcomes.
2. Data exploration:- Reducing high-dimensional data to lower dimensions
3. Anomaly Detection:- Identify outliers or anomalies in datasets that deviate from normal patterns.
4. Feature extraction:- For model's improvement in performance & reduce dimensionality.
5. Scalability:- Easily handle large volumes of data & adapt to new data without the need for manual labeling.

Disadvantages of unsupervised learning:-

1. Interpretability:- Models may be harder to interpret as compared to supervised learning models due to the lack of explicit labels.
2. Evaluation metrics:- Lack of clear evaluation metrics can make model's performance assessment challenging.
3. Domain knowledge:- Requires domain expertise to assess & validate the discovered patterns effectively.
4. Computational Complexity:- Some unsupervised algorithms can be computationally intensive & time-consuming especially for large datasets.

Semi-Supervised Machine Learning:-

* Semi-supervised machine learning is a type of machine learning algorithm that lies bet? supervised & unsupervised machine learning.

* It represents the intermediate ground bet? supervised (with labelled training data) & unsupervised learning (with no labelled training data) algorithms & uses a combination of labelled & unlabelled datasets during the training ~~data~~ period.

* Dataset mostly consists of unlabelled data. The algorithm in semi-supervised machine learning is trained on a dataset that contains a small amount of labeled data & a larger amount of unlabeled data.

Definition:-

Semi-supervised learning is a machine learning paradigm where the algorithm learns from a combination of labeled & unlabeled data. The goal is to use the labeled data to learn & utilize unlabeled data to discover the underlying patterns & relationships within the data.

↳ Why semi-supervised machine learning is used? → To overcome the drawbacks of supervised learning & unsupervised learning algorithms

↳ What's the process? → Initially similar data is clustered with an unsupervised learning algorithm, & further it helps to label the unlabelled data in to labeled data. It is because labeled data is comparatively more expensive acquisition than unlabeled data

Key components of Semi-supervised machine learning:-

① Labeled & unlabeled data :- We have some data with labels (like pictures of cats labeled as "cat") & a lot of data without labels. The algorithm learns from both types of data to improve its understanding.

② Label Propagation:- This technique spreads the known labels to similar unlabeled data points. If a labeled picture of a cat looks similar to an unlabeled picture, the algorithm assumes the unlabeled one is also a cat. This helps the algorithm learn more from unlabeled data.

③ Pseudo-labeling:- The algorithm predicts labels for the unlabeled data based on its current understanding. These predicted labels are then used to train the model further. It is like making educated guesses to teach the algorithm.

④ Self-training:- The algorithm trains on the labeled data, then uses its knowledge to predict labels for unlabeled data. If it is confident about these predictions, it adds them to the labeled dataset for future training. This helps the model to learn more from unlabeled data.

⑤ Transfer learning:- In semi-supervised learning, we can use knowledge from pre-trained models on labeled data to assist in tasks with limited labeled examples. This improves overall performance.

IQSE

Example:- How semi-supervised learning works?

Spam filtering:

① Labeled & unlabeled data:- We have a small set of labeled emails where some are marked as spam & others as non-spam. The majority of emails in our dataset are not labeled as spam or non-spam.

② Label propagation:- The algorithm looks at the labeled emails & their characteristics (like keywords, sender information) to identify patterns. It then spreads these patterns to similar unlabeled ~~data~~ emails. For example, if a labeled email with the word "discount" is marked as spam, similar unlabeled emails with "discount" may also considered spam.

③ Pseudo-labeling:- The algorithm predicts labels for unlabeled emails based on its current understanding. If it predicts that an email is likely spam based on its content, it assigns a pseudo-label of "spam" to that email.

④ Self-training:- The algorithm trains on the labeled emails & uses its knowledge to predict labels for unlabeled emails. If it is confident in its predictions, it adds these emails with pseudo-labels to the training set for further training iterations.

⑤ Transfer-learning:- If there are pre-trained models for email classification tasks with labeled data, we can use this knowledge to improve semi-supervised learning model's performance.

Types/Categories of semi-supervised ML algorithms:-

1. Self-training algorithms:- Training iteratively on labeled data & using model to predict labels for unlabeled data.

2. Co-training algorithms:- In co-training, the algorithms trains multiple models on different subsets of features. Each model then provides predictions for the unlabeled data, & the agreement bet? the models helps in labeling unlabeled data points.

3. Semi-supervised support vector machines (S3VM):- It extends traditional support vector machines to incorporate unlabeled data in the learning process. It aims to find decision boundary that not only separates labeled data but also considers distribution of unlabeled data.

4. Graph-based algorithms:- These algorithms represent the data as a graph where nodes are data points & edges represent relationship bet? them. By label propagation, these algorithms can use the structure of data for semi-supervised learning

5. Generative Models:- Generative models such as Generative Adversarial Networks (GANs) or Variational Autoencoders (VAEs) can be used for semi-supervised learning by learning the underlying distribution of the data. They can generate new data points & help in improving model's understanding of data distribution.

⑥ Low-density separation algorithms:- These algorithms aim to find a decision boundary that separates high density regions (labeled data) from low-density regions (unlabeled data). It helps in effective classification of both labeled & unlabeled instances

* Each type in semi-supervised algorithm has its strengths & is suitable for different types of datasets & learning tasks.

Ex:- Medical data

Speech recognition:- podcasts, videos or customer service call recordings. Learns relationship b/w audio features like tones, accents & speech patterns.

Applications of semi-supervised ML

① Text classification:- Spam detection, sentiment analysis, document categorization.

② Image recognition:- Recognizes patterns & objects effectively by propagating labels to similar unlabeled images.

③ Speech recognition

④ Anomaly detection → learning from labeled data & detecting deviations in unlabeled data.

⑤ Drug discovery:- Predict the properties of new drug compounds.

⑥ Recommendation systems:- Utilizes both explicit user ratings (labeled data) & implicit user behavior (unlabeled data) to make accurate predictions.

⑦ Medical Image Analysis:-

⑧ Social network analysis:- To predict connections or identify communities within a networks

Advantages of Semi-supervised Machine Learning

① Efficient use of Unlabeled data:- No labeling efforts, Improvement in Performance

② Cost-effective:- Labeled data is expensive & time-consuming

③ Improved generalizations

④ Enhanced performance:- Better predictions

⑤ Scalability:- Can scale well to large datasets.

Disadvantages of semi-supervised machine learning

① Quality of unlabeled data:- noise/irrelevant information can negatively impact model's performance

② Model Complexity:- Needs additional steps such as label propagation or pseudo-labeling

③ Risk of overfitting:-

④ Limited Control:- Lack of direct supervision makes it difficult to interpret & debug model's performance.

(5) Domain dependency :- Some algorithms may perform better in certain domains while being less effective in others.

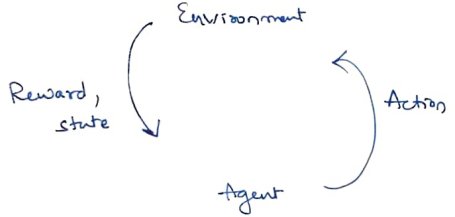
(4) Reinforcement Learning :-

It is a feedback-based learning method, in which a learning agent gets a reward for every right action & gets a penalty for each wrong action. The agent learns from these feedbacks & improves its performance.

* In reinforcement learning, agent interacts with environment & explores it.

* Characteristics of reinforcement learning → Trial, error & delay.

* The model keeps on increasing its performance using reward feedback to learn the behavior / pattern.



* Ex: Robotic dog as agent.

✓ Action :- Movement of arms

✓ environment : Space in which it operates.

By receiving rewards / penalties based on success in arm movements, robotic dog can adjust its actions over time to optimize performance.

Key Components of Reinforcement Learning

(1) Agent :- Decision maker based on environment's feedback.

(2) Environment :- External environment with which agent interacts with. It gives feedback to agent in terms of rewards / penalties based on agent's actions.

(3) Actions :- The set of possible choices that the agent takes in a given state. The agent selects actions based on its policy, where choosing actions in different states is defined.

(4) State :- The current situation or configuration of environment at a particular time step. Action takes from one state to another.

(5) Rewards :- Numeric feedback provided by the environment to the agent after each action. Agent's aim is to maximize the cumulative reward it receives over time.

(6) Policy :- Strategy / set of rules that the agent uses to select actions in different states.

How it works?

(1) Agent :- Smart car (student learning to drive)

(2) Environment :- Simulated environment with which autonomous vehicles interact. It provides feedback to the vehicle based on collisions or successful navigation.

(3) Actions :- Smart car can slow down, speed up, turn left, turn right or keep going straight. These are the choices.

④ State:- Current situation on road, where the car is, how fast its going & if there are any obstacles nearby.

⑤ Rewards:- Reaching ~~a~~ ~~point~~ destination gives it points, but hitting an obstacle takes points away.

⑥ Policy:- How to drive based on what's happening around. "avoid obstacles" or more complex strategies.

Algorithms:-

① Q-learning:- Estimates expected reward for taking action in a given state. (Model-free RL)

② SARSA (State-Action-Reward-State-Action):- Updates a Q-function for the action that was actually taken.

③ Deep Q-learning:- Uses neural network to represent Q-f? which allows it to learn complex relationship bet? states & actions.

Applications of re-inforcement machine learning:- RL can teach

① Game playing ② Robotics → tasks autonomously
③ Autonomous vehicles → navigation & make decisions
④ Recommendation system ⑤ Healthcare:- Optimize treatment plans & drug discovery

⑥ NLP:- Dialogue systems & chatbots.
⑦ Finance & Trading ⑧ Supply chain & Inventory management.
⑨ Energy management ⑩ Game AI ⑪ Personal assistant
⑫ VR & AR ⑬ Industrial Control ⑭ Education:- Adaptive learning systems. ⑮ Agriculture

Advantages:-

① Autonomous decision making for robotics & game-playing
② Long-term results that are difficult to achieve
③ Solves complex problems that can't be solved using conventional techniques.

Disadvantages:-

① ~~Expensive~~ Training reinforcement learning agents can be computationally expensive & time-consuming.
② Not recommended for simple problems.

Difference bet? Supervised & Unsupervised Learning

Machine learning life cycle :-

How does ML works? It can be described using it's life cycle.

*. ML life cycle is a cyclic process to build an efficient ML project. The goal here is to find a solution to the problem or project

* There are 7 major steps:-

① Gathering data ② Data preparation ③ Data wrangling ④ Analyse Data ⑤ Train the model ⑥ Test the model. ⑦ Deployment.

*. To solve the problem, we create ML system called Model & this model is created by providing training. But to train a model, we need data, so life cycle starts with collecting data.

① Gathering data :- here data-related problems are identified.

(i) We need to identify the different data sources as data can be collected from various sources such as files, databases, internet, Mobile devices.

(ii) Quantity & quality affects efficiency of o/p.

(iii) Integrate the data obtained from different sources. which results in coherent set of data "dataset"

② Data Preparation :- Here we put our data into a suitable place & prepare it to use in ML training.

① Data exploration :- Here we understand the nature of data that we have to work with. We need to understand the characteristics, format & quality of data. Here we find correlations, general trends & outliers.

② Data pre-processing :- Analysis of data.

③ Data Wrangling :- It is the process of cleaning, selecting variable & to use converting raw data into a useable format. Cleaning addresses the quality issues.

Issues :

① ~~Data~~ Missing Values
② Duplicate data
③ Invalid data
④ Noise

Various techniques are used to address these issues.

④ Data analysis :- Now, the cleaned & prepared data is passed on to analysis step. This step involves

① Selection of analytical techniques
② Building models
③ Review of result.

Here the aim is to build ML model to analyze the data to analyze the data using various analytical techniques & review the outcomes.

*. Classification / Regression / Association / Cluster analysis are used & build the model using prepared data & evaluate model.

(5) Train Model :- Use datasets to train the model using various ML algorithms. Training is necessary to make the model understand various patterns, rules & features. thereby improving performance.

(6) Test Model :- Here we checks for the % accuracy of our model by providing a test dataset to it.

(7) Deployment :- Deploying model in the real-world system. If the above-designed model is producing an accurate result as per our requirement with acceptable speed, then we deploy model in the real world. We also check whether it is improving its performance using available data.

Applications of ML

(1) Image recognition :- Automatic friend tagging suggestion. ML's face detection & recognition system algorithm is used.
This project was initiated by Facebook & named as "Deep Face"

(2) Speech recognition :- "Search by voice" in Google. Converts speech to text.
Google assistant, Siri, Cortana & Alexa.

(3) Traffic Prediction :- Everyone who is using Google Map is helping this app to make it better :-
(1) Real time location of vehicle from Google Map app & sensors
(2) Average time has taken on past days at the same time.

(4) Product recommendation :- Searching for some product on Amazon & getting same product recommendation or advertisement while internet surfing on same browser. Netflix → entertainment series, movies etc.

(5) Self-driving cars :- Tesla self-driving car → detect people & objects while driving

(6) Email spam & malware filtering :-

Spam filters algorithms → Content filter, header filter, general blacklists filter, rules-based filters, permission filters.

Malware detection → Decision tree, Naive Bayes classifier, Multi-layer Perceptron.

(7) Virtual Personal Assistant :- Uses our voice instructions & performs the tasks.

Record voice → send it over server on cloud → decode using ML algorithm.

(8) Online fraud detection :- online transactions are safe. ~~fake~~ Detects fake accounts, fake ids & steal money in the middle of transaction.

Algorithm → Feed forward Neural network.

* For each genuine transaction, o/p is converted into some hash values & these are i/p to next round.

(9) Stock market trading :- long short term memory neural n/w.

(10) Medical diagnosis :- Builds 3D models that can predict brain tumors in brain.

(11) Automatic language translation :- Even if we aren't aware of new language in a new location, it is not a problem anymore. Google's GNMT (Google neural machine translation) provides this feature of translating text to needed language. (Automatic translation)

Technology :- Sequence to sequence learning algorithm.

Why Python ?

Python is a preferred language for ML due to various reasons :-

(1) Rich ecosystem of libraries : Python includes a vast array of libraries & frameworks which are created specifically for ML & data science tasks. Example libraries includes NumPy, Pandas, Scikit-learn, TensorFlow & PyTorch provide powerful tools for data manipulation, visualization & building ML models.

(2) Ease of learning & use :- Python is popular for its simplicity & readability. It has clean syntax & extensive documentation that enable developers to write code efficiently & speeding up the development process.

(3) Community Support:- Python has a large & active community of developers & data scientists who support/contribute to open-source projects & offer support through forums, tutorials & online resources. This collaborative environment encourages knowledge sharing & innovation.

(4) Versatility:- Python is a versatile language meaning it is not only used in ML but also in various other domains like web development, automation, scientific computing etc.

(5) Integration Capabilities:- Python seamlessly integrates with other languages, & tools & technologies. It can be combined with languages like R, C/C++, Java for building ML applications.

(6) Scalability:- Python offers scalability for ML projects by handling of large datasets & complex algorithms. It supports deployment on different platforms including cloud services.

(7) Platform independence:- Code written in Python can run on different OS without modifications. It provides the flexibility & ease of deployment across various platforms.

Scikit-Learn:-

*. It is a popular open-source ML library in Python that offers a comprehensive set of tools & algorithms for data analysis, modeling & ML tasks.

*. It is built on NumPy, SciPy & Matplotlib which are popular foundational libraries for scientific computing & data visualization.

*. It provides a user-friendly & efficient framework for both beginners & experts in the field of data science.

Features of Scikit-learn:

(1) Simple & Consistent API:- Scikit-learn provides a consistent API for simplifying the process of implementing ML algorithms. This uniform interface allows users to seamlessly switch bet? different models & techniques without need for code modifications.

(2) Diverse algorithms:- Scikit-learn offers a diverse collection of supervised & unsupervised learning algorithms including support vector machines, decision trees, random forests, K-means clustering & more. Developers can choose suitable algorithm. (Classification, regression, clustering & dimensionality reduction)

③ Model Evaluation & Selection:: Scikit learn provides robust tools for model evaluation, parameter tuning & selection. Techniques such as cross-validation, grid search & performance metrics like accuracy, precision, recall & F1 score help users assess & optimize performance of their ML models

④ Preprocessing & feature engineering:- The library includes utilities for data pre-processing, feature scaling & transformation. These capabilities enable users to prepare & clean their data effectively before training ML models to improve model performance & generalization.

⑤ Integration with NumPy & Pandas:: Scikit-learn seamlessly integrates with NumPy arrays, matplotlib & Pandas Dataframe to facilitate data manipulation. Data Pre-processing & model building

⑥ Extensive documentation & Community Support:: Scikit-learn offers detailed documentation, tutorials & examples to help users in understanding & using ML algorithms effectively. It has large community support

⑦ Scalability & Performance:: Scikit-learn can handle small to medium-sized datasets that offers scalability through integration with parallel processing libraries like Dask & joblib. This allows users to handle larger datasets efficiently & distribute resources when necessary

Installing scikit-learn

Step 1 Type 'cmd' in "Run" dialog box & ok.

step 2: Check python is installed on the system | not. Scikit-learn is compatible with python 3.6 & higher. To check, type in command prompt as,

```
Python --version
```

step 3:- Update pip (Optional)
Updating pip to the latest version before installing any package by running following command,

```
python.exe -m pip install --upgrade pip
```

step 4: Scikit-learn can be installed using pip

```
Pip install scikit-learn
```

step 5:- Verify installation of scikit-learn library

```
import sklearn
print(sklearn.__version__)
```

⑧ User friendly & easy to use:: It is designed with a user-friendly interface & simple syntax which makes it accessible for both beginners & experienced ML practitioners.

⑨ Flexibility & Customization::

⑩ Wide adoption & Industry usage

Essential libraries & tools

For effective implementation of ML projects, essential libraries & tools are important.

1. Scikit-learn:- ML learning library in Python that provides a simple & efficient tool for data analysis & modeling.
2. Numpy:- Scientific computing in Python, operations using multi-dimensional arrays & matrices.
3. Scipy:- Library for mathematics, science & engineering. It is built on Numpy. It offers modules for optimization, integration & interpolation etc.
4. Pandas:- Data manipulation & analysis library that offers data structures & functions to efficiently work with structured data.
5. Matplotlib:- Library for creating static, animated & interactive visualizations in Python.
6. Jupyter Notebook:- Interactive web-based tool for creating & sharing docs that contain live code, eq^{n}s, visualizations & narrative text.

Jupyter Notebook

* It is an open source web-based application that is used for creating & sharing documents that contain live code, eq^{n}s, visualizations & narrative text.

Key Features:

1. Interactive coding:- Jupyter Notebook supports over 40 programming languages, including Python, R, Julia & Scala. Code can be written in cells & executed interactively which allows for immediate feedback & iterative data exploration.
2. Rich media Support:- In addition to code cells, Jupyter Notebook supports integrating rich media such as images, videos, HTML, LaTeX & more.
3. Visualization:- It integrates with data visualization libraries like Matplotlib, Seaborn & Plotly.
4. Shareability:- Notebooks can be shared with others via email, Dropbox, Github, or Jupyter Notebook Viewer. This sharing capability facilitates collaboration on projects & education by allowing others to see & execute documents live.
5. Conversion:- JNs can be converted to a number of open standard o/p formats (HTML, presentation slides, LaTeX, PDF, Python script & more) through "nbconvert".

Installation

`pip install notebook`

`jupyter notebook` → Launches a local web server & opens a notebook interface in the default web browser.

② NumPy

* It is a fundamental package for scientific computing in Python.
* It supports large, multi-dimensional arrays & matrices along with a collection of mathematical functions to operate on these arrays.
* In ML, NumPy arrays are used for storing & manipulating data, serving as inputs to ML algorithms for tasks like classification, regression & clustering

Key features

① Multi-dimensional arrays - Numpy's main object is ndarray, which is a " " " that allows efficient manipulation of large datasets. These arrays can be created, indexed, sliced & reshaped easily. It is ideal for storing & processing data in ML algorithms.

② Mathematical functions:- Numpy has f^n's like np.sum, np.mean, np.min, np.dot etc for efficient manipulation of data.

③ Broadcasting:- Numpy's broadcasting feature allows operations to be performed on arrays of different shapes. This helps in performing mathematical operations in ML algorithms.

④ Linear algebra operations:- Matrix multiplication, matrix inversion, eigenvalues, Singular value decomposition f^n's are provided which serve as inputs to algorithms such as regression, clustering & dimensionality reduction.

⑤ Random number generation:- Numpy provides f^n's for random number generation which are useful for tasks like data shuffling, initialization of weights in neural networks & creating synthetic datasets for testing ML models.

⑤ Integration with other libraries:- Pandas, SciPy, Scikit-learn & matplotlib

Common Use Cases

① Data analysis & Statistical operations
② ML
③ Image processing & computer graphics
④ Scientific Simulations

Installation:-

`pip install numpy` → This command instructs pip to download & install numpy package from Python Package Index (PyPI)

```
C:\>python
Python 3.11.1 ----------
>>> import numpy as np
```

→ If no error message is displayed, then it means numpy has been successfully installed.

Matrix Operations on Arrays:

```
import numpy as np

a = np.array([[1, 2], [3, 4]])
b = np.array([[5, 6], [7, 8]])

print("Matrix Addition:\n", a+b)
print("Matrix Subtraction:\n", a-b)
print("Matrix Multiplication: (element-wise)\n", a*b)
print("Matrix Multiplication:\n", np.matmul(a,b))
print("Transpose of a:\n", np.transpose(a))
```

① $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix}$

o/p:
```
[[ 6  8]
 [10 12]]
```

② $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} - \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} -4 & -4 \\ -4 & -4 \end{bmatrix}$

③ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$= \begin{bmatrix} 5 & 12 \\ 21 & 32 \end{bmatrix}$ → Element Wise Multiplication

④ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

$\begin{matrix} 15 \\ 22 \\ +3 \end{matrix}$ $\begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$

⑤ $\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

Transpose $\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix}$ Flips 'a' over its diagonal

Statistical functions on Arrays

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])
print("Standard Deviation:", np.std(a))
print("Variance:", np.var(a))
print("Median:", np.median(a))
print("Percentile:", np.percentile(a, 50))   # 50th percentile also known as median
```

③ Scipy

* It is an open-source python library that is used for scientific & technical computing.
* It builds on the top of NumPy & provides a wide range of f^n's for numerical integration, optimization, signal processing, linear algebra, statistics & more

Key features:

① Integration & optimization:- Scipy provides f^n's for numerical integration, interpolation & optimization.
Optimization:- Parameter tuning in algorithms like support vector machines / neural networks

② Signal Processing:- Scipy offers tools for signal processing tasks like filtering, spectral analysis, & waveform generation. It is used in speech recognition or image processing.

③ Linear algebra:- Scipy provides set of f^n's for linear algebra operations including matrix decomposition, eigenvalue problems & solving linear system of eq^n's. These operations are important for many ML algorithms that involve matrix computations.

④ Statistics:- Scipy includes statistical f^n's for probability distributions, hypothesis testing & descriptive statistics. These f^n's are useful for data analysis, model evaluation & understanding significance of results in ML experiments.

⑤ Sparse Matrices:- SciPy supports sparse matrix representations & provides efficient algorithms for working with large, sparse datasets. This is used in ML tasks like collaborative filtering, text mining & graph analysis.

⑥ Image processing:- Filtering, edge detection & morphology. Useful in preprocessing image data in ML applications like computer vision & object recognition.

⑦ Interoperability with Numpy:- Seamlessly integrates Numpy's array manipulation capabilities with advanced scientific computing f^os of SciPy.

Common Use Cases:-

① Scientific analysis:- Algorithms to do precise calculations & data analysis in physics & chemistry.

② Engineering applications:- Simulating real-world processes, optimizing systems & analyzing data.

③ Academic research:- SciPy's statistical tools to analyze experimental data in fields like economics, sociology & psychology.

④ Image Processing:- SciPy's sub-package ndimage supports tasks in multi-dimensional image processing. (Medical image analysis & computer vision)

Installation:- Python & Numpy should be installed before Scipy.

```
pip install scipy
```

```
import scipy
```
→ If no error appears, installation is successful

Ex:- Solving a system of linear eq^s:-

```
from scipy.linalg import solve

A = [[1, 2], [3, 4]]
B = [8, 18]
solution = solve(A, B)
print("Solution of the system:", solution)
```

o/p: Solution of the system: [2. 3.]

Expl^n:-

* $x + 2y = 8$
  $3x + 4y = 18$
* In a matrix form,
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 8 \\ 18 \end{bmatrix}$$
* Matrix A → coefficients of variables in the system of linear eq^s
* Vector B → Right hand side of eq^s (constant terms)
* o/p → $x = 2$ & $y = 3$ (sol^s to system of eq^s)

Pandas

* It is a popular open-source data manipulation & analysis library for Python

* It is widely used for data cleaning, transformation, exploration & analysis

Key features

(1) DataFrames:- The primary data structure in Pandas is the DataFrame. It is a 2-dimensional labeled data structure with columns of potentially different types.
They are similar to tables in a relational database or Excel spreadsheet. It stores structured data & manipulates them.

(2) Series:- Pandas also provides Series data structures which is a one-dimensional labeled array. It holds values of any data type. Series represent a single column or row in a DataFrame. It is useful to perform operations on individual data elements.

(3) Data Import & Export:- Pandas supports reading & writing data from various file formats including CSV, Excel & SQL databases, JSON & more

(4) Data cleaning & transformation:- Pandas offers various functions for data cleaning & preprocessing tasks such as handling missing values, removing duplicates, transforming data types & reshaping data

(5) Data Exploration:- Pandas offer powerful tools for exploring & summarizing data, including descriptive statistics, grouping & aggregation, filtering, sorting & visualization. These capabilities help developers to gain insights into patterns of data.

(6) Data Manipulation:- Pandas enables efficient data manipulation operations such as merging, joining, concatenating & reshaping datasets. These operations are useful for combining data from multiple sources, creating new features & preparing data for modeling

(7) Time series analysis:- Pandas include specialized data structures & f°s for working with time series data such as date/time indexing, resampling, shifting & rolling window calculations. It is useful for analyzing time-dependent data in ML applications.

(8) Integration with Numpy:- Pandas is built on top of Numpy.

Common Use Cases

1. Data Cleaning:- Efficiently performed by Pandas
2. Data exploration & Analysis:-
3. Data Visualization:- Integrates with matplotlib to plot data directly from DataFrames
4. Building ML models:- Ensures data is in proper format for model training
5. Time Series Analysis:- resampling of time series data to convert frequencies, generating date ranges

Installation:

```
Pip install pandas
```

```
import pandas as pd
print(pd.__version__)
```

Matplotlib:- It is a popular plotting library in Python that provides a wide range of functionalities for creating static, interactive & animated visualizations.

* It is used for creating high quality plots, charts & graphs for data analysis & visualizations in various fields including data science, ML, scientific research & more.

Key Features

1. Simple & Flexible:- Matplotlib offers a simple & intuitive interface for creating a wide variety of plots with just few lines of code. Customization in terms of colors, labels, markers, etc can be done. fonts, plot sizes, line style
2. Support for various plot types:- Matplotlib supports a wide range of plot types including line plots, scatter plots, bar plots, histogram, Pie charts, box plots, heatmaps, & more. Visualization in meaningful ways is made possible.
3. Publication-quality plots:- Matplotlib is used to create publication-quality plots with customizable features such as titles, labels, legends, gridlines, annotations. Users can adjust plot look & feel to meet specific requirements.
4. Multiple Backends:- Matplotlib supports multiple backends for rendering plots including interactive backends for use in Jupyter Notebook & GUI applications as well as non-interactive backends for saving plots to image files or embedding in web applications.
5. Integration with NumPy:- Numpy arrays can be directly plotted by integrating Matplotlib with Numpy.
6. Subplots & Figures:- Matplotlib supports creation of multiple subplots within a single figure so that we can display multiple plots in a grid layout. This feature allows to compare different datasets or to visualize related info in a single plot window.

~~Common Use Cases~~

(7) Interactive Plotting :- Matplotlib can be used in interactive mode to create dynamic plots that respond to user interactions such as zooming, panning & selecting data points.

Common Use Cases :-

(1) Data Visualization :- Time-series data & ordered categories

(2) Scientific Plotting :- High quality plots, charts & figures

(3) Algorithm Visualization :- For diagnosing problems

(4) Interactive applications :- Visualizing data in real-time

(5) Education :- To teach concepts in data science, Statistics & computational mathematics.

Installation

```
pip install matplotlib
```

```
pip show matplotlib
```
→ displays location & version number

```
import matplotlib.pyplot as plt
```