UNIT – I

Introduction to Machine Learning: Evolution of Machine Learning, Paradigms for ML, Learning by Rote, Learning by Induction, Reinforcement Learning, Types of Data, Matching, Stages in Machine Learning, Data Acquisition, Feature Engineering, Data Representation, Model Selection, Model Learning, Model Evaluation, Model Prediction, Search and Learning, Data Sets.

1. Evolution of Machine Learning

- Machine learning is the process of learning a model that can be used in prediction based on data.
- Prediction involves assigning a data item to one of the classes or associating the data item with a number. The former activity is classification while the latter is regression.
- Machine learning is important and state-of-the-art topic.
- It is gained prominence because of improved processing speed and storage space of computers and the availability of large data sets for experimentation.
- Deep learning is sub area of machine learning.
- Perceptron was the earliest popular ML tool and it forms the basic building block of various Deep learning architectures including multilayer perceptron network, convolutional neural networks and recurrent neural networks.
- The mathematical logic was the ideal vehicle for building AI systems.
- Some of the initial contributions in this area like General Problem Solver (GPS), Automatic Theorem Proving (ATP), rule-based systems and programming languages like Prolog and LISP were all outcomes of this view.
- Various problem solving and game playing solutions also had this flavor.
- During the twentieth century, a majority of prominent AI researchers were of the view that logic is AI and AI is logic. Most of the reasoning systems were developed based on this view.
- Further, the role of a artificial neural networks in solving complex real-world AI problems was under-appreciated.
- However, this view was challenged in the early twenty-first century and the current view is that AI is deep learning and deep learning is AI.
- TensorFlow and Pytorch are basic frameworks to implement deep learning applications.
- Artificial neural network forms the backbone of Deep Learning.
- A high-level view of AI is shown in figure 1.
- The tasks related to conventional AI are listed separately.
- Here ML may be viewed as dealing with more than just pattern recognition tasks.
- Classification and clustering are the typical tasks of PR system.
- However, ML deals with recognition problems also.
- Data mining is the efficient organization of data in the form of a database.

- Note that data structures and algorithms are basic to both conventional and current AI systems.
- Logic and discrete structures played an important role in the analysis and synthesis of conventional AI systems.
- The importance of other background topics may be summarized as follows:

Machine Learning

- We deal with vector and vector spaces and these topics are better appreciated through linear algebra.
- The data input to ML system may be viewed as matrix, popularly called the data matrix.
- If there are n data items, each represented as an l-dimensional vector, then the corresponding data matrix A is of size n x l.
- Linear algebra is useful in analyzing the weights associated with the edges in a neural network.
- Matrix multiplication and eigen analysis are important in initializing the weights of the neural network and in weight updates.
- It can also help in weight normalization.

Probability and Statistics

- The role of probability and statistics need not be explained as ML is, in fact, statistical ML.
- These topics help in estimating the distributions underlying the data.
- Further, they play a crucial role in analysis and inference in ML.

Optimization

- Optimization along with calculus is essential in training neural networks where gradients and their computations are important.

Information theoretic concepts

- like entropy, mutual information and kullback-leibler divergence are essential to understand topics such as decision tree classifiers, feature selection and deep neural networks.

2. Paradigms for ML

There are different ways or paradigms or ML, such as

1. Learning by rote
2. Learning by induction
3. Reinforcement learning

1. Learning by Rote

- Rote learning is the process of memorizing information based on repetition.
- Rote learning enhances students' ability to quickly recall basic facts and helps develop foundational knowledge of a topic.
- Examples of rote learning include memorizing multiplication tables or the periodic table of elements.
- Chess masters spend a lot of time memorizing the great games of the past. It is this role learning that teaches them how to think in chess. Chess and checkers games are example for rote learning.
- The drawbacks of rote learning are that it can be repetitive, it's easy to lose focus and it doesn't allow for a deeper understanding of a topic.

What is Rote Learning in AI?

- Rote learning in AI refers to a form of memorization where a model simply memorizes information without truly understanding its context.
- Rote Learning is a learning process where the AI system stores and reproduces data or patterns without the ability to generalize and apply knowledge effectively.

2. Learning by Induction

- Inductive learning is a subset of machine learning that uses algorithms to infer generalizations from specific cases.
- The inductive learning algorithm (ILA) creates a model based on a set of training examples that are then used to predict new examples.
- This is the most popular and effective form of ML.
- Here, learning is achieved with the help of examples or observations.
- It may be categorized as follows:
    i. Learning from example
    ii. Learning from observations

i. Learning from examples

It is assumed that a collection of labeled examples are provided and the ML system uses these examples to make a prediction on new ML problems: classification and regression.

a) Classification

- Classification is a supervised machine learning method where the model tries to predict the correct label of a given input data.

➢ In classification, the model is fully trained using the training data, and then it is evaluated on test data before being used to perform prediction on new unseen data.

➢ For example, training a machine learning model on historical patient data can help healthcare specialists accurately analyse their diagnoses:
During the COVID-19 pandemic, machine learning models were implemented to efficiently predict whether a person had COVID-19 or not.

➢ As shown in figure 3 for classification example. In this figure contains two classes Class A and Class B. If new data item came you need to classify either class A or Class B based on class labels.

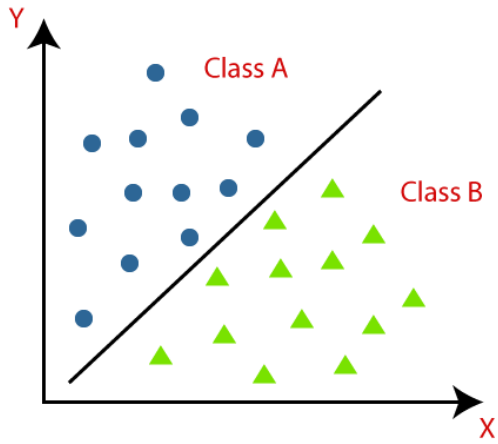


Figure 3. Classification Example

Some Applications of Machine Learning Classification Problems

- ✓ Image classification
- ✓ Fraud detection
- ✓ Document classification
- ✓ Spam filtering
- ✓ Facial recognition
- ✓ Voice recognition
- ✓ Medical diagnostic test
- ✓ Customer behavior prediction
- ✓ Product categorization
- ✓ Malware classification

b) Regression

Regression in machine learning refers to a supervised learning technique where the goal is to predict a continuous numerical value based on one or more independent features. It finds relationships between variables so that predictions can be made. We have two types of variables present in regression:

- Dependent Variable (Target): The variable we are trying to predict e.g house price.
- Independent Variables (Features): The input variables that influence the prediction e.g locality, number of rooms.

Regression analysis problem works with if output variable is a real or continuous value such as “salary” or “weight”.

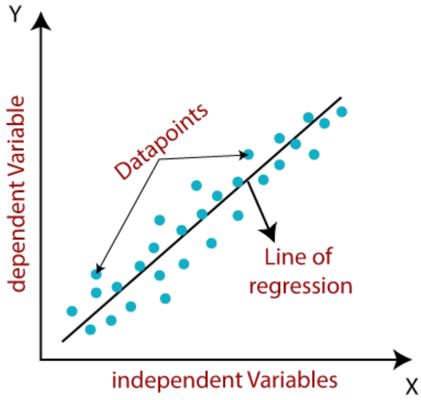


Figure 4 Regression Example

Regression is the process of predicting a continuous value. We can use regression methods to predict a continuous value, such as CO2 emission from a car model, using some other variables.

A s shown in figure 4, In regression, there are two types of variables: a dependent variable and one or more independent variables. The dependent variable is the "state", "target" or "final goal" we study and try to predict, and the independent variables, also known as explanatory variables, are the "causes" of those "states". The independent variables are shown conventionally by X, and the dependent variable is denoted by Y. A regression model relates Y, or the dependent variable, to a function of X, i.e., the independent variables. The key point in regression is that the dependent variable value should be continuous, and not a discrete value. However, the independent variable or variables can be measured on either a categorical or continuous measurement scale.

Common use for machine learning regression models include:

- Forecasting continuous outcomes like house prices, stock prices, or sales.
- Predicting the success of future retail sales or marketing campaigns to ensure resources are used effectively.
- Predicting customer or user trends, such as on streaming services or e-commerce websites.
- Analysing datasets to establish the relationships between variables and an output.
- Predicting interest rates or stock prices from a variety of factors.
- Creating time series visualizations.

ii. Learning from examples

- Observations are also instances like examples but they are different because observations need not be labelled.
- In this case, we cluster or group the observations into a smaller number of groups. Such grouping is performed with the help of clustering algorithm that assigns similar patterns to the same group or cluster.

Clustering

- Clustering or also known as cluster analysis is a type of machine learning technique, which aims to group the unlabelled dataset.
- It can be defined as "A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."

Example of Clustering

In vegetable sections, apples, bananas, Mangoes, and many such items are grouped in separate sections, in an attempt to make it easier for us to find things. The clustering technique also works on the same principle. As shown in figure 5 example of cluster. We have the dataset which is raw dataset. We need to group based on their similarities.

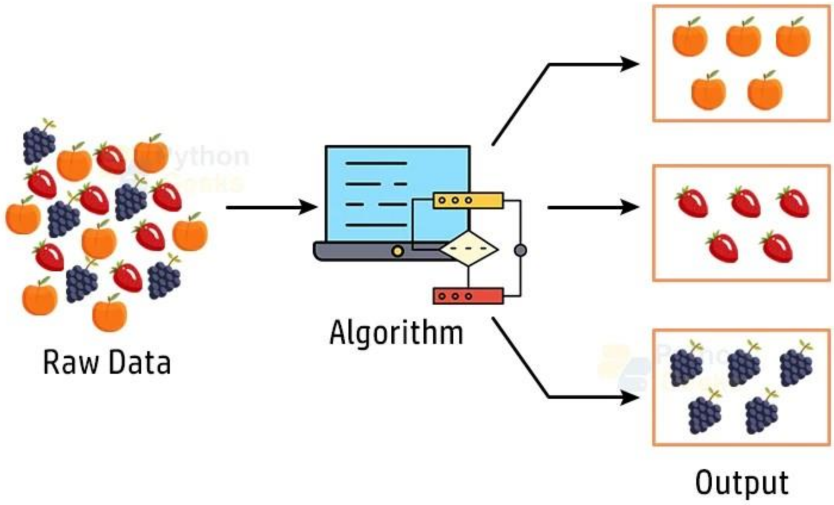


Figure 5. Clustering Example

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection

3. Types of Data

Datasets are a collection of instances that all share a common attribute.
For machine learning models to understand how to perform various actions, training datasets must first be fed into the machine learning algorithm, followed by validation datasets (or testing datasets) to ensure that the model is interpreting this data accurately.

Different Types of data types

The Data type is broadly classified into

1. Quantitative
2. Qualitative

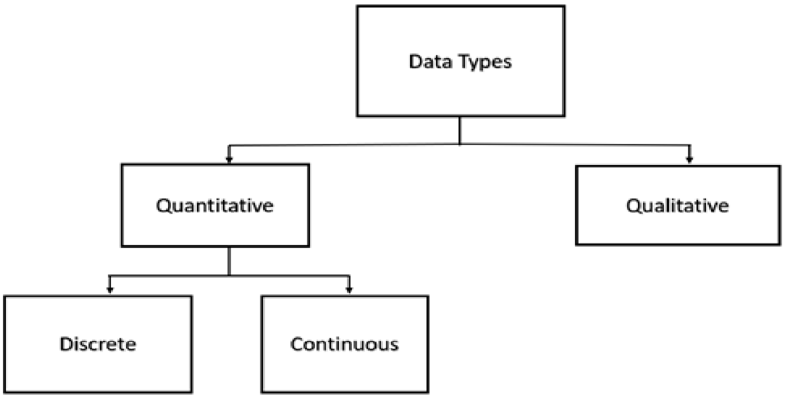


Figure 6. Different Data Types

<u>1. Quantitative data type: –</u>

This type of data type consists of numerical values. Anything which is measured by numbers. E.g., Profit, quantity sold, height, weight, temperature, etc.

This is again of two types

A.) Discrete data type: –

The numeric data which have discrete values or whole numbers. This type of variable value if expressed in decimal format will have no proper meaning. Their values can be counted.

E.g.: – No. of cars you have, no. of marbles in containers, students in a class, etc.

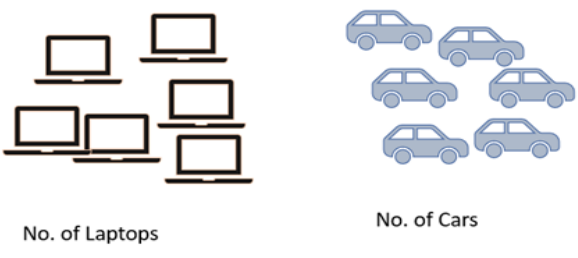


Figure 7. Discrete Data Types

B.) Continuous data type: –

The numerical measures which can take the value within a certain range. This type of variable value if expressed in decimal format has true meaning. Their values cannot be counted but measured. The value can be infinite

E.g.: – height, weight, time, area, distance, measurement of rainfall, etc.

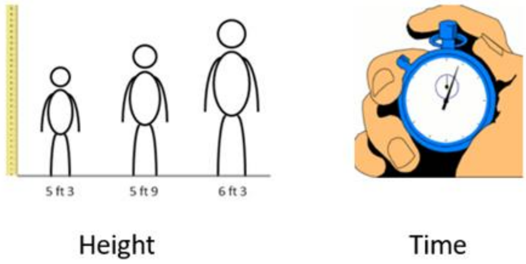


Figure 8. Continuous Data Types

<u>2. Qualitative data type: –</u>

These are the data types that cannot be expressed in numbers. This describes categories or groups and is hence known as the categorical data type.

This can be divided into:-

a. Structured Data:

This type of data is either number or words. This can take numerical values but mathematical operations cannot be performed on it. This type of data is expressed in tabular format.

E.g.) Sunny=1, cloudy=2, windy=3 or binary form data like 0 or1, Good or bad, etc.

ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Figure 9. Structured Data

<u>b. Unstructured data:</u>

This type of data does not have the proper format and therefore known as unstructured data. This comprises textual data, sounds, images, videos, etc.

Figure 10. Unstructured Data

Besides this, there are also other types refer as Data Types preliminaries or Data Measures: -

1. Nominal
2. Ordinal
3. Interval
4. Ratio

These can also be referring different scales of measurements.

I. Nominal Data Type:

This is in use to express names or labels which are not order or measurable. Nominal data are used to label variables without any quantitative value.

E.g., male or female (gender), race, country, etc.

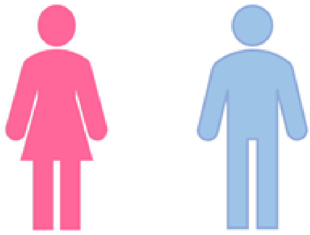


Figure 11. Gender (Female, Male), An Example of Nominal Data Type

II. Ordinal Data Type:

This is also a categorical data type like nominal data but has some natural ordering associated with it. Ordinal scales are often used for measures of satisfaction, happiness, and so on.

E.g., Likert rating scale, Shirt sizes, Ranks, Grades, etc.

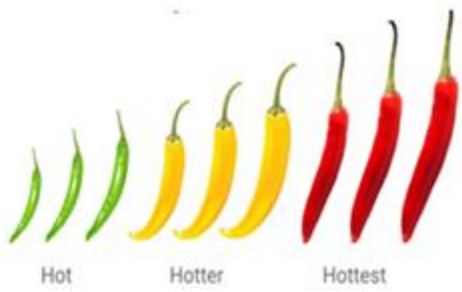


Figure 12. Rating (Good, Average, Poor), An Example of Ordinal Data Type

III. Interval Data Type:

Interval data, also called an integer, is defined as a data type which is measured along a scale, in which each point is placed at equal distance from one another. Interval data always appears in the form of numbers or numerical values where the distance between the two points is standardized and equal.

E.g., Temperature measured in degree Celsius, time, Sat score, credit score, pH, etc. difference between values is familiar. In this case, there is no absolute zero. Absolute Temperature is a common example of interval data. If one day the temperature is 50 degrees and the next it is 60 degrees, we can say that the second day was exactly 10 degrees warmer than the first.

Figure 13. Temperature, An Example of Interval Data Type

IV. Ratio Data Type:

This quantitative data type is the same as the interval data type but has the absolute zero. Here zero means complete absence and the scale starts from zero. This is the global scale. Ratio data is very similar interval data, except zero means none. For ratio data, it is not possible to have negative values.

For instance, height is ratio data. It is not possible to have negative height. If an object's height is zero, then there is no object. This is different than something like temperature. Both 0 degrees and -5 degrees are completely valid and meaningful temperatures.

E.g., Temperature in Kelvin, height, weight, etc.

Figure 14. Weight, An Example of Ratio Data Type

4. Matching

- Matching is an important activity in ML.
- It is used in both supervised learning and unsupervised learning.
- Matching is carried out by using a proximity measure which can be a distance/dissimilarity measure or a similarity measure.
- Two data items, u and v represented as l-dimensional vectors, match better when the distance between them is smaller or when the similarity between them is larger.
- A popular distance measure is the Euclidean distance and a popular similarity measure is the cosine of the angle between vectors.

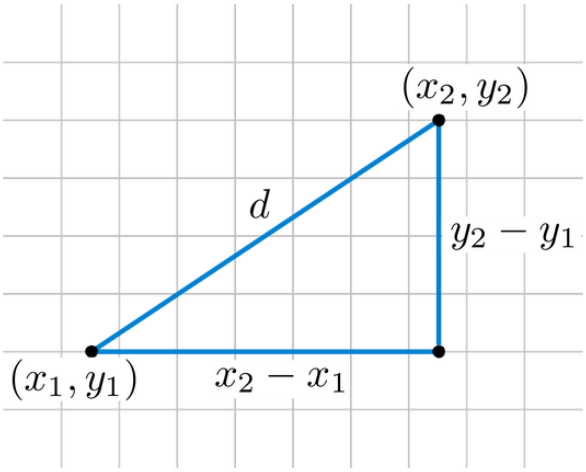
Euclidean Distance:

Euclidean Distance represents the shortest distance between two vectors. It is the square root of the sum of squares of differences between corresponding elements.

This completes our Euclidean distance formula for two points in two-dimensional space.

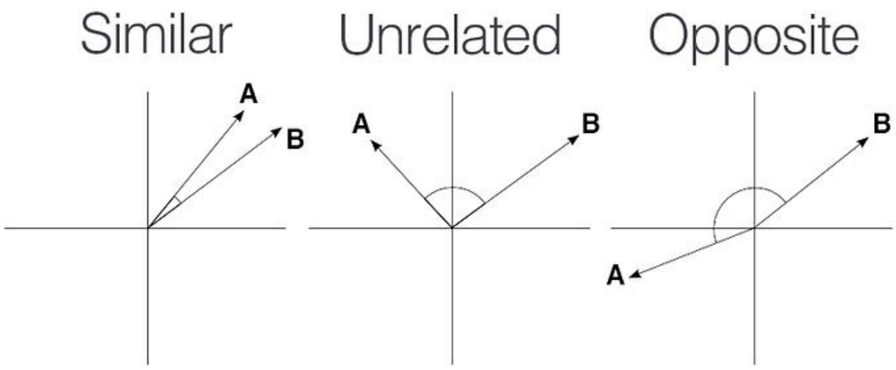
$$d(x, y) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

This defines the Euclidean distance between two points in one, two, three or higher-dimensional space where n is the number of dimensions and x_k and y_k are components of x and y respectively.



<u>Cosine Similarity</u>

- Cosine similarity is a metric that measures the cosine of the angle between two vectors projected in a multi-dimensional space
- The smaller the angle between the two vectors, the more similar they are to each other.
- As the cosine similarity measurement gets closer to 1, and then the angle between the two vectors A and B becomes smaller. In this case, A and B are more similar to each other.



Cosine similarity is described mathematically as the division between the dot product of vectors and the product of the Euclidean norms or magnitude of each vector.

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}},$$

<u>Two applications using matching:</u>

1. Finding the nearest neighbor of a pattern:

- Let x be an l-dimensional pattern vector. Let X={x1,x2,----,xn}be a collection of n data vectors. The nearest neighbor of x from X, denoted by NNx(X) is xj if

  d(x,xj) <=d(x,xi), for all xi belongs to X.
- This is an approximate search where a pattern that best matches x is obtained.

2. Assigning to a set with the Nearest Representative:

5. Stages in Machine Learning

Building a machine learning system involves a number of steps, illustrated in figure

Typically, the available data is spilt into training, validation and test data.
Training data is model learning or training.
Validation data is used to tune the ML model.
Test data is used to examine how the learnt model is performing.

1. Data Acquisition

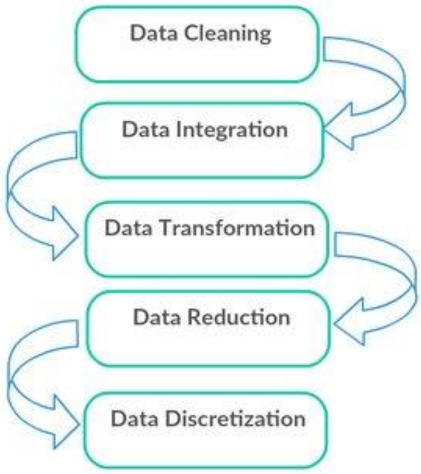
- This depends on the domain of application.
- For example, to distinguish between adults and children, measurements of their height or weight are adequate; to distinguish between normal and COVID-19 infected human, their body temperature and chest congestion may be more important than their height or weight.
- Typically, data collection is carried out before feature engineering.

2. Feature Engineering

This step involves a combination of data preprocessing and data representation.

Data preprocessing:

- Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models.
- Data preprocessing is the process of evaluating, filtering, manipulating, and encoding data so that a machine learning algorithm can understand it and use the resulting output.
- The major goal of data preprocessing is to eliminate data issues such as missing values, improve data quality, and make the data useful for machine learning purposes.



Data Pre-Processing Steps

The common problems encountered with the raw data and missing values, different ranges for different variables and the presence of outliers.

1. Missing data
2. Data from different domains
3. Outliers in the data

1. Missing data:

Missing data is defined as the values or data that is not stored (or not present) for some variable/s in the given dataset.

How is a Missing Value Represented in a Dataset?

1. NaN (Not a Number)
2. NULL or None
3. Empty Strings “ “
4. **Special Indicators**: Datasets might use specific indicators like -999, 9999
5. **Blanks or Spaces**

Different schemes are used for dealing with the prediction of missing values:

a. Use the nearest neighbor
b. Use a larger neighborhood
c. Cluster the data and locate the nearest cluster

a. Use the nearest neighbor:

- It’s based on the idea that similar data points are often close to each other in the feature space.
- It typically uses a distance measure like Euclidean distance to find the closest neighbours.

b. Use a larger neighborhood:

This method finds the closest data points (neighbours) based on available features and uses their values to estimate the missing value.

Example: Consider the set data vectors (1,1,1), (1,1,2), (1,1,3),(1,-,2), (1,1,-), (6,6,1)

- There are 6 three-dimensional pattern vectors in the set. Missing values are indicated by -.
- Let us see how to predict the missing values in (1,-,2).
- Let us use K=3 and find the 3 nearest neighbors (NNs) based on the remaining two feature value.
- The three NNs are (1,1,1), (1,1,2) and (1,1,3).
- Note that the second feature value of all these three neighbors is 1, which is the predicted value for the missing value in (1,-,2). So the vector becomes (1,1,2).

c. Cluster the data and locate the nearest cluster

- This approach is based on clustering the training data and locating the cluster to which x belongs based on the remaining l-1 components.
- Let x with its ith value missing belongs to cluster C^q.
- Let u^q be the centroid of $C^{q.}$
- Then the predicted value of x(i) is u_i^q, ith component of $u^{q.}$

Example Calculation

Let's say we have the following dataset with a missing value in Row 2, Column 2.

Column 1	Column 2	Column 3	Column 4
25	67	45	78
34	NaN	50	80
30	65	48	82
32	68	46	77

In this example:

1. The algorithm will find the nearest neighbors to Row 2 (based on Columns 1, 3, and 4).
2. Suppose Rows 1 and 4 are the closest neighbors.
3. The missing value in Row 2, Column 2 can then be estimated by taking the mean of the values in Column 2 from Rows 1 and 4, giving a value close to the missing one.
   67+65+68=200/3=66.6

2. Data from different domains

- The scales of values of different features could be very different.

- This would bias the matching process to depend more upon features that assume larger values., toning down the contributions of features with smaller values.
- So, in applications where different components of the vectors have different domain ranges, it is possible for some components to dominate in contributing to the distance between any pai of patterns.
- Consider for example, classification of objects into one of two classes: adult or child.
- Let the objects be represented by height in meters and weight in grams.
- Consider the adult represented by the vector (1.6, 75000) and a child represented by the vector (0.6, 5000).
- Assume that domain of height is [0.5, 2.5] and the domain of the weight is [2000, 200000] in this example.
- So, the large difference in the range of values if these two features.

- This can be handled by scaling different components differently and such process of scaling is called normalization.

<u>What is Data Normalization?</u>

- Data normalization is a technique used to organize data in a way that reduces duplicate data (redundancy) and ensures consistency in the process.
- It's particularly important concept for databases and machine learning to make data more structured, efficient and easier to analyze.

- There are two popular normalization schemes:
    i. Scaling using the range
    ii. Standardization

Normalization in Machine Learning

In machine learning, it's about scaling numerical data to bring all features to a similar range (e.g, 0 to 1) for better performance of algorithms.

<u>i. Scaling using the range</u>

This method scales each feature so that all values are within the range of 0 and 1. It achieves this by subtracting the minimum value of the feature and dividing by the range (difference between maximum and minimum values).

Formula:

(x — min(X)) / (max(X) — min(X))

- x is the value you want to normalize
- min(X) is the minimum value in your data set
- max(X) is the maximum value in your data set

Example:

Example of Normalization

Consider a dataset with annual income data for three individuals:

- Person A: $70,000
- Person B: $60,000
- Person C: $52,000

To normalize the income values using the formula:

1. Find the minimum and maximum values:

— Minimum income ({min}(x)) = $52,000

— Maximum income ({max}(x)) = $70,000

2. Apply the normalization formula to each income value:

2. **Apply the normalization formula** to each income value:

- For Person A:
$$x' = \frac{70{,}000 - 52{,}000}{70{,}000 - 52{,}000} = \frac{18{,}000}{18{,}000} = 1$$
- For Person B:
$$x' = \frac{60{,}000 - 52{,}000}{70{,}000 - 52{,}000} = \frac{8{,}000}{18{,}000} \approx 0.444$$
- For Person C:
$$x' = \frac{52{,}000 - 52{,}000}{70{,}000 - 52{,}000} = \frac{0}{18{,}000} = 0$$

After normalization:

- Person A’s normalized income = 1
- Person B’s normalized income ≈ 0.444
- Person C’s normalized income = 0

Normalization transforms data into a standardized range, facilitating fair comparisons across features with different scales. By bringing all values within a consistent range (0 to 1 in this case), normalization enhances the performance and interpretability of machine learning models, ensuring robust and reliable results.

ii. **Standardization**: Here, each feature is transformed to have a mean of 0 and a standard deviation of 1. This is achieved by subtracting the mean value and dividing by the standard deviation of the feature.

The formula for standardization is:

Z = (x — μ) / σ

Where:

Z is the standardized score (also called a z-score)

x is the original value you want to standardize

μ (mu) is the mean of the data set

σ (sigma) is the standard deviation of the data set

Example:

Consider a dataset with age data for three individuals:
- Person A: 45 years
- Person B: 44 years
- Person C: 40 years

To standardize the age values using the formula:

1. **Calculate the mean and standard deviation**:
   - Mean (μ) = $\frac{45+44+40}{3} = 43$
   - Standard deviation (σ) = $\sqrt{\frac{(45-43)^2+(44-43)^2+(40-43)^2}{3}}$
     $\sigma = \sqrt{\frac{4+1+9}{3}} = \sqrt{\frac{14}{3}} \approx 1.53$

2. **Apply the standardization formula** to each age value:
   - For Person A:
     $x' = \frac{45-43}{1.53} \approx \frac{2}{1.53} \approx 1.31$
   - For Person B:
     $x' = \frac{44-43}{1.53} \approx \frac{1}{1.53} \approx 0.65$
   - For Person C:
     $x' = \frac{40-43}{1.53} \approx \frac{-3}{1.53} \approx -1.96$

After standardization:
- Person A's standardized age ≈ 1.31
- Person B's standardized age ≈ 0.65
- Person C's standardized age ≈ -1.96

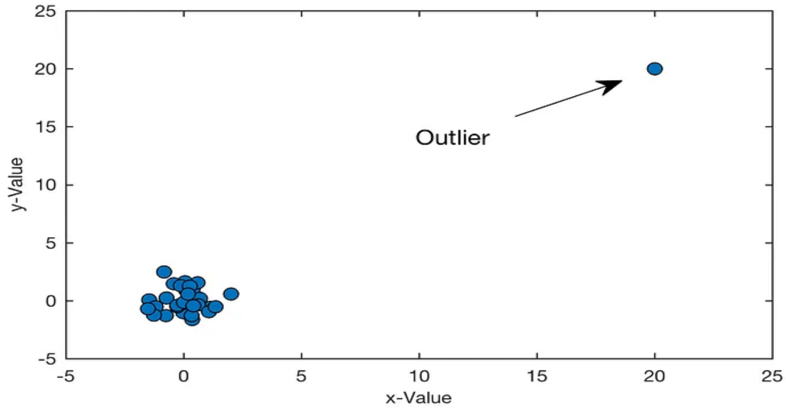
Standardization transforms data to have properties that are more suitable for many machine learning algorithms. By centering the data around 0 and scaling it to have a standard deviation of 1, standardization ensures that all features contribute equally to model training and evaluation. This preprocessing step is particularly useful when features have varying scales and distributions, enhancing the stability and performance of machine learning models.

3. Outliers in the Data

What is an outlier?

- ➢ It is a data point that behaves differently from all other data points in the dataset.
- ➢ Outliers are the datapoints which are significantly different from the rest of the datapoints in the dataset.

- Outliers increase the variance in the dataset which intern results in the decrease in the statistical power.
- So, it is very important to identify these outliers and treat them accordingly.



Reasons for the occurrence of Outliers

Outliers can occur because of various reasons. Some of the most common reasons include:

- Error in data entry.
- Inappropriate scaling of data points.
- Errors caused during measurement.
- Existence of genuine extreme data points.

Effect of outliers on ML algorithms

There are certain set of algorithms can impact badly. Those algorithms are Linear regression, Logistic regression, and Ada-boost.

3. Data Representation

Representation of Data items:

- In machine learning, data is often represented and organized using vectors.
- Each data point is typically represented as a vector, with each component of the vector representing a feature or attribute of the data.
- This vector representation allows machine learning algorithms to process and analyse the data effectively.

What are dimensions?

In the context of data analysis and machine learning, dimensions refer to the features or attributes of data. For instance, if we consider a dataset of houses, the dimensions could include the house's price, size, number of bedrooms, location, and so on.

What problems does it cause?

1. Data sparsity. As mentioned, data becomes sparse, meaning that most of the high-dimensional space is empty. This makes clustering and classification tasks challenging.
2. Increased computation. More dimensions mean more computational resources and time to process the data.
3. Overfitting. With higher dimensions, models can become overly complex, fitting to the noise rather than the underlying pattern. This reduces the model's ability to generalize to new data.
4. Distances lose meaning. In high dimensions, the difference in distances between data points tends to become negligible, making measures like Euclidean distance less meaningful.
5. Performance degradation. Algorithms, especially those relying on distance measurements like k-nearest neighbors, can see a drop in performance.
6. Visualization challenges. High-dimensional data is hard to visualize, making exploratory data analysis more difficult.

What is generalization

In supervised learning, the main goal is to use training data to build a model that will be able to make accurate predictions based on new, unseen data, which has the same characteristics as the initial training set. This is known as generalization.

Generalization error

- ➢ As you know, to train a machine learning model, you split the dataset into 3 sets: training, validation, and testing.
- ➢ As these names imply, you train your models using the training data, then you compare and tune them using the evaluation results on the validation set, and in the end, evaluate the performance of your best model on the testing set.
- ➢ The error rate on new cases is called the generalization error (or out-of-sample error), and by evaluating your models on the validation set, you get an estimate of this error.
- ➢ This value tells you how well your models perform on instances it has never iterated on.
- ➢ A model's generalization error (also known as a prediction error) can be expressed as the sum of three very different errors: Bias error, variance error, and irreducible error.

The concept of bias: Bias error

This type of error results from incorrect assumptions, such as thinking that the data is linear when it is actually quadratic. Bias is defined as a systematic error that happens in the machine learning model as a result of faulty ML assumptions. Bias is also the average squared difference between predictions of the model and actual data. Models with a higher percentage of bias will not match the training data. On the other hand, models with lower bias rates will coincide with the training dataset. Characteristics of a high-bias model include:

- Failure to capture proper data trends
- Potential towards underfitting

- More generalized/overly simplified
- High error rate

The concept of variance: Variance error

Variance, as a generalization error, occurs due to the model's excessive sensitivity to small variations in the training data. In supervised learning, the model learns from training data. So, if you change the training data, the model will also be affected. The variance shows the amount by which the performance of the predictive model will be impacted when evaluating based on the validation data. If your model can generalize well, it shouldn't change too much from one split to another. For example, a model with many degrees of freedom (such as a high-degree polynomial model) is likely to have high variance, while linear models will probably have lower variance. A high-variance model typically has the following qualities:

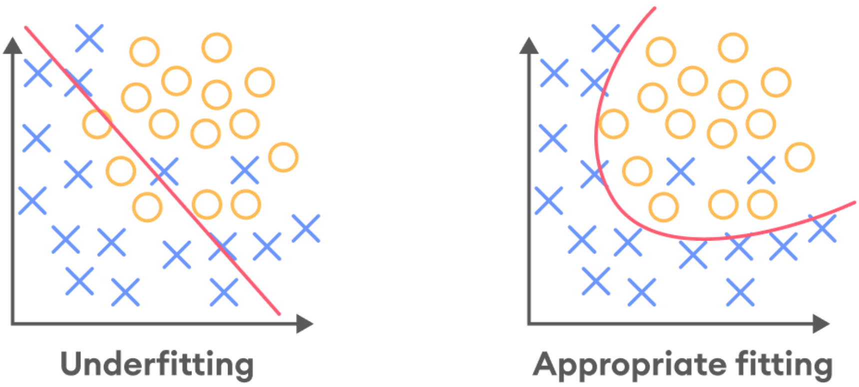
- Noise in the data set
- Potential towards overfitting
- Complex models
- Trying to put all data points as close as possible

<u>Underfitting and Overfitting:</u>

- ➢ Overfitting and underfitting are common problems in machine learning and can impact the performance of a model.
- ➢ Overfitting occurs when the model is too complex and fits the training data too closely. This leads to poor generalization.
- ➢ Underfitting happens when a model is too simple leading to poor performances.

<u>What is underfitting</u>

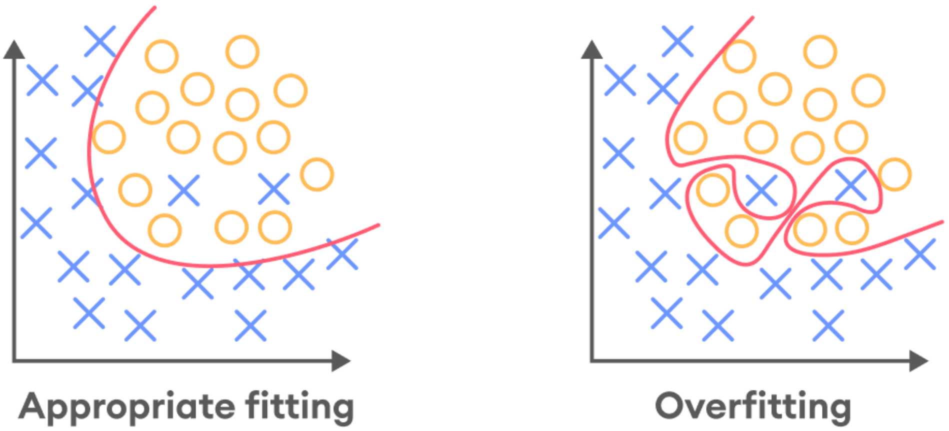
- ➢ Underfitting occurs when a model is not able to make accurate predictions based on training data and hence, doesn't have the capacity to generalize well on new data.
- ➢ Another case of underfitting is when a model is not able to learn enough from training data, making it difficult to capture the dominating trend (the model is unable to create a mapping between the input and the target variable).
- ➢ Machine learning models with underfitting tend to have poor performance both in training and testing sets (like the child who learned only addition and was not able to solve problems related to other basic arithmetic operations both from his math problem book and during the math exam).
- ➢ Underfitting models usually have high bias and low variance.



Underfitting is an issue that occurs when a machine learning model is too simple leading to low model accuracy and poor performance.

What is overfitting

- A model is considered overfitting when it does extremely well on training data but fails to perform on the same level on the validation data (like the child who memorized every math problem in the problem book and would struggle when facing problems from anywhere else).
- An overfitting model fails to generalize well, as it learns the noise and patterns of the training data to the point where it negatively impacts the performance of the model on new data.
- If the model is overfitting, even a slight change in the output data will cause the model to change significantly. Models that are overfitting usually have low bias and high variance.



Overfitting is a machine learning performance issue that occurs when the model is too complex and fits the training data too closely. This leads to poor generalization.

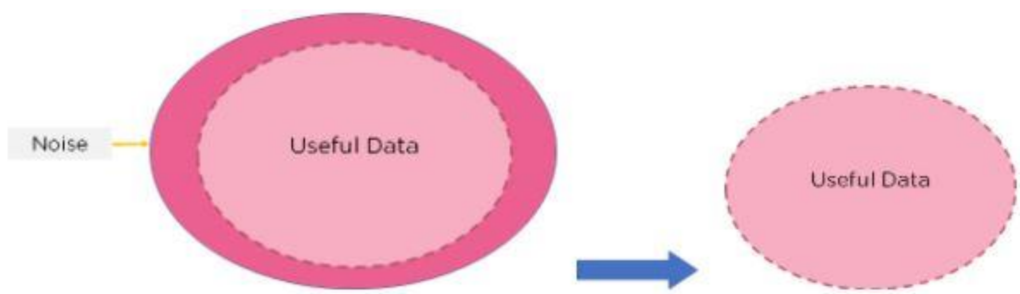
Dimensionality Reduction Methods

Well known dimensionality reduction approaches are:

- Feature Selection
- Feature Extraction

Feature Selection:

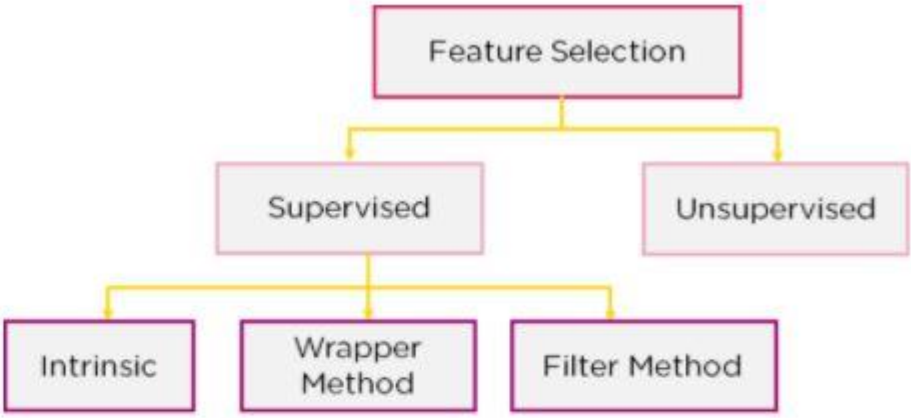
The goal of feature selection techniques in machine learning is to find the best set of features that allows one to build optimized models of studied phenomena.



Feature Selection Models

Feature selection models are of two types:

1. **Supervised Models**: Supervised feature selection refers to the method which uses the output label class for feature selection. They use the target variables to identify the variables which can increase the efficiency of the model
2. **Unsupervised Models:** Unsupervised feature selection refers to the method which does not need the output label class for feature selection. We use them for unlabelled data.



Importance of Feature Selection in Machine Learning

1. Enhanced Model Performance: Selecting relevant features leads to more accurate and efficient models.

2. Reduced Overfitting: Including irrelevant features can lead to overfitting, where the model performs well on the training data but poorly on test data. Feature selection helps mitigate this issue.

3. Improved Interpretability: Models with fewer features are easier to interpret and explain to stakeholders.

4. Faster Training and Inference: Working with a reduced set of features speeds up both the training and inference phases of the machine learning pipeline.

Feature Extraction:

Feature extraction is a process in machine learning and data analysis that involves identifying and extracting relevant features from raw data.

Unlike feature selection, which chooses a subset of existing features, feature extraction creates new features by combining or modifying the original data. This transformation aims to represent the data in a way that simplifies the model's task while retaining as much relevant information as possible.

Why is Feature Extraction Important?

- **Improved Model Performance**: Extracted features are often more informative, leading to better accuracy and generalizability. By focusing on the most important aspects of the data, feature extraction can help avoid overfitting and improve model robustness.
- **Reduced Training Time**: By reducing the number of features or simplifying the representation of the data, feature extraction minimizes computational costs, speeding up both training and inference times.
- **Reduced Data Storage Requirements**: Since feature extraction typically reduces the size of the feature space, it helps save on storage and processing resources, especially when dealing with large datasets.
- **Enhanced Data Understanding**: Extracting features often highlights the underlying patterns or structure of the data, making it easier to interpret and understand. For example, dimensionality reduction techniques like **PCA** help uncover hidden relationships between variables.
- **Improved Handling of High-Dimensional Data**: In fields like image processing or NLP, raw data can be highly dimensional. Feature extraction helps reduce this dimensionality, making it easier to build models without suffering from the curse of dimensionality.

Different Types of Techniques for Feature Extraction

Feature extraction techniques can be divided into several categories based on the type of data and the specific goals of the machine learning task. Below are the most common categories of feature extraction methods:

1. Statistical Methods

Statistical methods aim to extract features by summarizing the statistical properties of the data. These methods are commonly used when the data is numerical or time-series in nature.

- **Mean, Median, Standard Deviation**: Simple statistics that help summarize the central tendency or spread of data.
- **Correlation Coefficient**: Measures the linear relationship between two variables, which can be useful in selecting key features for prediction tasks.

2. Dimensionality Reduction Methods

These methods aim to reduce the number of features while retaining most of the relevant information. They are essential when dealing with high-dimensional data to avoid overfitting and improve model efficiency.

- **Principal Component Analysis (PCA)**: A widely-used technique that transforms the original features into a set of linearly uncorrelated components (principal components) that capture most of the variance in the data.
- **Linear Discriminant Analysis (LDA)**: Focuses on finding a linear combination of features that best separates different classes in classification tasks.

3. Feature Extraction for Textual Data

Text data presents unique challenges, and specific techniques are needed to extract meaningful information for machine learning models.

- **Bag-of-Words (BoW)**: Represents text data as a collection of words without considering grammar or word order. Each word becomes a feature, and its frequency across documents is counted.
- **TF-IDF (Term Frequency-Inverse Document Frequency)**: A refinement of the BoW approach, TF-IDF assigns a weight to each word based on its frequency in a document relative to its occurrence across all documents, helping to distinguish important words from common ones.

4. Signal Processing Methods

In time-series or signal data, specialized methods help extract features that capture the patterns within the data.

- **Fast Fourier Transform (FFT)**: Converts time-domain data into the frequency domain, which is particularly useful for signal processing tasks such as audio analysis or vibration monitoring.
- **Wavelet Transform**: Decomposes a signal into components at different scales, helping capture both frequency and location information.

5. Image Data Extraction

For image data, various techniques help extract meaningful features, focusing on visual aspects like edges, shapes, and colors.

- **Edge Detection**: Identifies boundaries within an image where there is a sharp change in intensity, often used in object detection tasks.
- **Color Histograms**: Represents the distribution of colors in an image, helping models differentiate between images based on color content.
- **Texture Analysis**: Captures the patterns of texture within an image, which can be crucial for applications such as medical imaging or quality control in manufacturing.

6. Principal Component Analysis (PCA)

PCA is a dimensionality reduction technique that transforms the original features into a smaller set of new features called principal components. These components capture the maximum variance in the data, allowing for a simplified feature set while retaining essential information. PCA is particularly useful for high-dimensional datasets where it helps reduce noise and avoid overfitting.

7. Bag of Words (BoW)

BoW is a simple but effective technique for text feature extraction. It represents a text document as a set of words, disregarding grammar and word order. Each word in the vocabulary is treated as a feature, and the frequency of its occurrence in a document is recorded. Although it ignores semantics, it provides a basic numerical representation for text classification tasks.

8. Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF builds on BoW by assigning each word a weight based on how frequently it appears in a document, relative to how often it appears across all documents. This helps distinguish important terms (those frequent in a specific document but rare across others) from common ones, improving the model's ability to differentiate between topics or sentiment in text.

Supplementary Material

Machine Learning Steps

- Machine learning is the process of making systems that learn and improve by themselves, by being specifically programmed.
- The ultimate goal of machine learning is to design algorithms that automatically help a system gather data and use that data to learn more.
- Systems are expected to look for patterns in the data collected and use them to make vital decisions for themselves.

The task of imparting intelligence to machines seems daunting and impossible. But it is actually really easy. It can be broken down into 7 major steps :

1. Collecting Data:

- As you know, machines initially learn from the data that you give them.
- It is of the utmost importance to collect reliable data so that your machine learning model can find the correct patterns.
- The quality of the data that you feed to the machine will determine how accurate your model is.
- If you have incorrect or outdated data, you will have wrong outcomes or predictions which are not relevant.
- Make sure you use data from a reliable source, as it will directly affect the outcome of your model.
- Good data is relevant, contains very few missing and repeated values, and has a good representation of the various subcategories/classes present.

2. Preparing the Data:

After you have your data, you have to prepare it. You can do this by :

- Putting together all the data you have and randomizing it. This helps make sure that data is evenly distributed, and the ordering does not affect the learning process.
- Cleaning the data to remove unwanted data, missing values, rows, and columns, duplicate values, data type conversion, etc. You might even have to restructure the dataset and change the rows and columns or index of rows and columns.
- Visualize the data to understand how it is structured and understand the relationship between various variables and classes present.
- Splitting the cleaned data into two sets - a training set and a testing set. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training.

3. Choosing a Model:

- A machine learning model determines the output you get after running a machine learning algorithm on the collected data.
- It is important to choose a model which is relevant to the task at hand.
- Over the years, scientists and engineers developed various models suited for different tasks like speech recognition, image recognition, prediction, etc.
- Apart from this, you also have to see if your model is suited for numerical or categorical data and choose accordingly.

4. Training the Model:

- Training is the most important step in machine learning.
- In training, you pass the prepared data to your machine learning model to find patterns and make predictions.
- It results in the model learning from the data so that it can accomplish the task set. Over time, with training, the model gets better at predicting.

5. Evaluating the Model:

- After training your model, you have to check to see how it's performing.
- This is done by testing the performance of the model on previously unseen data.
- The unseen data used is the testing set that you split our data into earlier.
- If testing was done on the same data which is used for training, you will not get an accurate measure, as the model is already used to the data, and finds the same patterns in it, as it previously did.
- This will give you disproportionately high accuracy.
- When used on testing data, you get an accurate measure of how your model will perform and its speed.

6. Parameter Tuning:

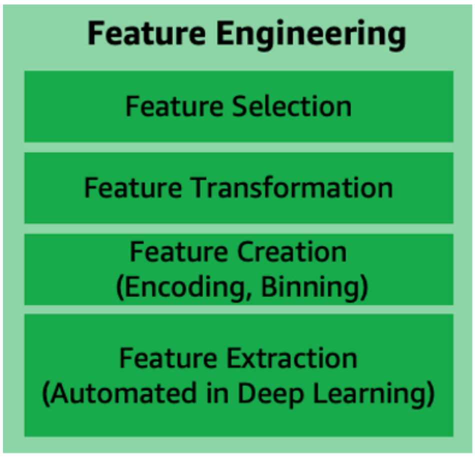
- Once you have created and evaluated your model, see if its accuracy can be improved in any way.
- This is done by tuning the parameters present in your model.
- Parameters are the variables in the model that the programmer generally decides.
- At a particular value of your parameter, the accuracy will be the maximum.
- Parameter tuning refers to finding these values.

7. Making Predictions:

In the end, you can use your model on unseen data to make predictions accurately.

What is Feature Engineering?

- Feature Engineering is the process of creating new features or transforming existing features to improve the performance of a machine-learning model.
- Feature engineering is the process of selecting, manipulating and transforming raw data into features that can be used in supervised learning.
- A “feature” is any measurable input that can be used in a predictive model.



- **Feature creation** refers to the creation of new features from existing data to help with better predictions. Examples of feature creation include: one-hot-encoding, binning, splitting, and calculated features.

- **Feature transformation and imputation** include steps for replacing missing features or features that are not valid. Some techniques include: forming Cartesian products of features, non-linear transformations (such as binning numeric variables into categories), and creating domain-specific features.

- **Feature extraction** involves reducing the amount of data to be processed using dimensionality reduction techniques. These techniques include: Principal Components Analysis (PCA), Independent Component Analysis (ICA), and Linear Discriminant Analysis (LDA). This reduces the amount of memory and computing power required, while still accurately maintaining original data characteristics.

- **Feature selection** is the process of selecting a subset of extracted features. This is the subset that is relevant and contributes to minimizing the error rate of a trained model. Feature importance score and correlation matrix can be factors in selecting the most relevant features for model training.

What Exactly is Data Preprocessing in Machine Learning?

- Data preprocessing in machine learning involves transforming raw, unorganized data into a structured format suitable for machine learning models.
- This step is essential because raw data often contains missing values, inconsistencies, redundancies, and noise.
- Preprocessing addresses these issues, ensuring that data is accurate, clean, and ready for analysis.

Challenges with Raw Data and Their Solutions

Working with raw data often presents challenges that can impact the accuracy and efficiency of machine learning models. Here's how data preprocessing in machine learning tackles these issues:

- **Missing Values**:
  Missing data can lead to inaccurate or biased models.
  Solution: Techniques like mean/mode imputation or advanced methods such as k-nearest neighbors (KNN) imputation can fill in the gaps, ensuring a complete dataset.
- **Inconsistent Formats**:
  Varied scales (e.g., age in years vs. income in dollars) can distort model results.
  Solution: Standardization (e.g., z-scores) or normalization (e.g., min-max scaling) aligns data to consistent formats for fair comparisons.
- **Redundant Information**:
  Duplicate or irrelevant data adds noise and reduces model efficiency.
  Solution: Deduplication techniques and feature selection methods remove unnecessary data, improving processing speed and accuracy.
- **Noisy Data**:
  Irrelevant or erroneous information can obscure meaningful patterns.
  Solution: Noise removal techniques, such as filtering outliers or smoothing data, help retain essential information while eliminating distractions.

Major Tasks Involved in Data Preprocessing in Machine Learning

Data preprocessing consists of multiple steps that prepare data for machine learning. Each task plays a distinct role in refining data and making it suitable for algorithms. Let's explore them one by one.

1. Data Cleaning

Data cleaning focuses on identifying and fixing inaccuracies or inconsistencies in raw data. This step ensures that your dataset is reliable and ready for analysis.

- **Tasks:** Correcting missing values, removing duplicates, and identifying outliers.
- **Techniques:** Imputation methods for missing data, removing duplicate entries, and outlier detection through statistical approaches.
- **Purpose:** Enhances the dataset's reliability, improving the model's ability to generate accurate predictions.

2. Data Integration

Data integration combines information from different sources into a single, cohesive dataset. This is especially critical when working with data collected from multiple systems or platforms.

- **Tasks:** Resolving format differences, aligning schemas, and removing redundancies across datasets.
- **Techniques:** Schema matching to align fields, deduplication processes, and resolving conflicts between datasets.
- **Purpose:** Creates a unified dataset that eliminates inconsistencies across data sources.

3. Data Transformation

Data transformation prepares integrated data for machine learning by converting it into formats that models can interpret effectively.

- **Tasks:** Adjusting data scales, encoding categorical variables, and normalizing distributions.
- **Techniques:** Methods like normalization, standardization, and one-hot encoding.
- **Purpose:** Ensures uniformity across variables, making them comparable and improving model performance.

4. Data Reduction

Data reduction simplifies the dataset by focusing only on the most relevant information while minimizing computational load.

- **Tasks:** Selecting essential features, reducing data dimensions, and sampling smaller subsets.
- **Techniques**: Feature selection methods, dimensionality reduction like PCA, and systematic data sampling.
- **Purpose:** Streamlines datasets for faster processing without losing critical insights.

Normalization schemes:

- In machine learning, algorithms rely on data to learn patterns and make predictions. However, raw data is rarely ready for direct use by these models.
- Data preprocessing is a critical step that can significantly affect the performance of machine learning models.
- Among the various preprocessing techniques, feature scaling is one of the most important.
- Feature scaling ensures that the numerical features of a dataset are on a similar scale, which can prevent models from being biased toward certain features simply because of their magnitudes.
- Without scaling, machine learning algorithms may struggle to converge or produce suboptimal results, especially for distance-based methods like **k-Nearest Neighbours (k-NN)** and **K-Means Clustering**.

- **Feature scaling** is the process of transforming the numerical features in a dataset to a common scale or range. In machine learning, different features in a dataset may have different ranges or units (e.g., age might range from 0 to 100, while income might range in the thousands or millions). Without scaling, these differences can cause models to weigh features unequally, leading to poor performance.
- Feature scaling ensures that all features contribute equally to the model by bringing them to a similar scale. This process can involve either normalizing or standardizing the values, depending on the requirements of the algorithm and the nature of the dataset.

There are several methods for feature scaling, including:

1. **Normalization:** Transforming values to fall within a specific range (e.g., 0 to 1).
2. **Standardization:** Transforming features so that they have a mean of 0 and a standard deviation of 1.
3. **Min-Max Scaling**: Scaling features to a predefined range, typically between 0 and 1.
4. **Robust Scaling:** Scaling data based on percentiles to handle outliers more effectively.

Feature Selection vs. Feature Extraction

While both feature selection and feature extraction are essential processes in machine learning, they serve different purposes and operate in distinct ways:

Feature Selection

- **Definition**: Feature selection focuses on selecting a subset of the existing features from the original dataset. It eliminates irrelevant, redundant, or less important features without altering the data itself.
- **Goal**: The goal is to choose the most important features that contribute the most to the predictive model, helping reduce dimensionality and computational costs without transforming the data.
- **Examples**: Techniques like **Forward Selection**, **Backward Elimination**, and **Recursive Feature Elimination (RFE)** are common in feature selection.
- **When to Use**: Feature selection is preferred when the existing features are sufficient to train the model effectively, but some may be unnecessary or introduce noise.

Feature Extraction

- **Definition**: Feature extraction, on the other hand, involves creating new features by transforming or combining the original features. This transformation reduces the complexity of the dataset while retaining its core information.
- **Goal**: The goal is to transform raw data into a format that can be more effectively processed by machine learning models, often leading to the creation of new features that represent the original data in a more informative way.
- **Examples**: Techniques like **PCA**, **TF-IDF**, and **FFT** are commonly used for feature extraction, creating entirely new representations of the data.
- **When to Use**: Feature extraction is ideal when working with high-dimensional or complex datasets (e.g., image, text, or signal data) where simply selecting from existing features would not be sufficient for model accuracy or efficiency.

Applications of Feature Extraction

Feature extraction plays a vital role in various fields by transforming complex data into usable formats for machine learning models. Below are some real-world applications across different domains:

1 Speech Recognition

- In speech recognition systems, feature extraction helps in identifying key elements such as phonemes and speech patterns. Techniques like **Mel-frequency cepstral coefficients (MFCC)** are used to extract features from raw audio signals, making it easier for machine learning models to recognize and classify speech.

2 Natural Language Processing (NLP)

- Feature extraction in NLP is crucial for tasks like sentiment analysis, topic modeling, and text classification. Techniques such as **TF-IDF** and **Word2Vec** help represent textual data as numerical features that machine learning algorithms can process effectively.

3 Machine Condition Monitoring

- In predictive maintenance and anomaly detection, feature extraction is often applied to sensor data collected from machines. For instance, **Fast Fourier Transform (FFT)** and wavelet transforms help identify patterns in vibration signals that indicate equipment failures or maintenance needs.

4 Biomedical Engineering

- In the healthcare sector, feature extraction is widely used to analyze medical images, signals, and datasets. For example, extracting features from **MRI** or **CT scan** images can help identify early signs of diseases, while signals from **EEG** or **ECG** can be transformed into features for diagnosing neurological or cardiac conditions.

5 Image Processing and Computer Vision

- Feature extraction is a critical component in image processing tasks like object detection, image classification, and facial recognition. Techniques like **edge detection**, **color histograms**, and **texture analysis** transform raw images into feature sets that machine learning models can use to identify objects or make predictions.

6 Financial Market Analysis

- In finance, feature extraction techniques can help in analyzing market trends, stock prices, or trading patterns. By extracting meaningful patterns from time-series data, such as through **PCA** or **Fourier analysis**, financial models can make more accurate predictions regarding stock movements or economic indicators.

Tools and Libraries for Feature Extraction

Several tools and libraries provide efficient implementations for feature extraction, making it easier for data scientists and machine learning practitioners to apply these techniques across various types of data. Below are some popular tools for feature extraction:

1. Scikit-learn

- **Use Case**: Scikit-learn is a powerful Python library that includes a wide range of feature extraction techniques for numerical, categorical, and text data.
- **Techniques**: It offers tools for **PCA**, **TF-IDF**, **Bag-of-Words**, and other dimensionality reduction techniques.

- **Why Use It**: Scikit-learn is widely used for its simplicity, scalability, and integration with other machine learning algorithms.

2. OpenCV

- **Use Case**: OpenCV (Open Source Computer Vision Library) is commonly used for real-time image and video processing.
- **Techniques**: It offers feature extraction techniques like **edge detection**, **color histograms**, and **HOG (Histogram of Oriented Gradients)** for image classification tasks.
- **Why Use It**: OpenCV is an industry-standard tool for computer vision applications and provides a rich set of functionalities for feature extraction from images and video data.

3. TensorFlow / Keras

- **Use Case**: TensorFlow and Keras are deep learning frameworks that offer powerful tools for feature extraction, especially for complex data like images, text, and audio.
- **Techniques**: Both frameworks provide pre-trained models that can be used for feature extraction in image processing (e.g., **convolutional neural networks**) and text analysis.
- **Why Use It**: These frameworks are ideal for large-scale deep learning tasks and offer advanced feature extraction capabilities within their neural network layers.

4. PyTorch

- **Use Case**: PyTorch is another popular deep learning framework that is used for both research and production.
- **Techniques**: Similar to TensorFlow, PyTorch offers a wide variety of neural network layers that can be used for feature extraction, particularly for image and sequence data.
- **Why Use It**: PyTorch is known for its flexibility and ease of use, making it a favorite among researchers for feature extraction in experimental settings.

5. Librosa

- **Use Case**: Librosa is a specialized Python library for audio and music processing, focused on extracting features from audio data.
- **Techniques**: It provides tools for computing audio features like **MFCC**, **chroma features**, and **spectral contrast**, which are crucial for audio analysis and classification.
- **Why Use It**: Librosa is the go-to library for audio feature extraction due to its simple interface and specific focus on music and audio processing.

6. NLTK (Natural Language Toolkit)

- **Use Case**: NLTK is a library tailored for NLP tasks, including text preprocessing and feature extraction.
- **Techniques**: NLTK offers tools for extracting features from text, including **Bag-of-Words**, **n-grams**, and **part-of-speech tagging**.
- **Why Use It**: NLTK is popular in the NLP community for its ease of use and its extensive range of tools for linguistic analysis and text feature extraction.

7. Gensim

- **Use Case**: Gensim is a robust library for topic modeling and text-based feature extraction.
- **Techniques**: It specializes in extracting text features using models like **TF-IDF** and **Word2Vec**, which are essential for tasks like document similarity and topic extraction.
- **Why Use It**: Gensim is widely used in NLP applications where text feature extraction and modeling are required, especially for large corpora.

8. MATLAB

- **Use Case**: MATLAB is a versatile platform for numerical and signal processing.
- **Techniques**: It provides a comprehensive set of tools for feature extraction, particularly for time-series and signal data, including techniques like **Fourier transforms** and **wavelet decomposition**.
- **Why Use It**: MATLAB is favored in industries like engineering and medical fields for its powerful signal and image processing capabilities.

References

https://medium.com/@prasanNH/machine-learning-paradigms-a-comprehensive-overview-55498ff73eff
https://www.wolfram.com/language/introduction-machine-learning/machine-learning-paradigms/#:~:text=Machine%20learning%20is%20commonly%20separated,is%20presented%20to%20the%20computer.
https://www.analyticsvidhya.com/blog/2022/07/machine-learning-paradigms-with-example/
https://bcalabs.org/subject/introduction-to-learning-and-rote-learning-in-ai
https://www.scaler.com/topics/artificial-intelligence-tutorial/inductive-learning/
https://pianalytix.com/data-types-in-machine-learning/
https://www.datarobot.com/blog/the-importance-of-machine-learning-data/
https://www.blog.trainindata.com/knn-imputation-of-missing-values-in-machine-learning/
https://www.datacamp.com/blog/classification-machine-learning
https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning
https://pythongeeks.org/clustering-in-machine-learning/
https://builtin.com/articles/euclidean-distance#:~:text=Euclidean%20distance%20measures%20the%20shortest,neighbors%20and%20k%2Dmeans%20clustering.
https://medium.com/geekculture/cosine-similarity-and-cosine-distance-48eed889a5c4
https://builtin.com/machine-learning/cosine-similarity
https://www.analyticsvidhya.com/blog/2021/10/handling-missing-value/

https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/
https://www.geeksforgeeks.org/ml-feature-scaling-part-2/
https://www.simplilearn.com/tutorials/machine-learning-tutorial/machine-learning-steps

https://domino.ai/data-science-dictionary/feature-engineering
https://www.upgrad.com/blog/data-preprocessing-in-machine-learning/

https://www.appliedaicourse.com/blog/feature-scaling-in-machine-learning/#:~:text=What%20is%20Feature%20Scaling%3F,in%20the%20thousands%20or%20millions).

https://medium.com/@punya8147_26846/understanding-feature-scaling-in-machine-learning-fe2ea8933b66

https://www.iquanta.in/blog/data-normalization-in-machine-learning-techniques-advantages/
https://towardsdatascience.com/scaling-numerical-data-explained-a-visual-guide-with-code-examples-for-beginners-11676cdb45cb/

https://www.appliedaicourse.com/blog/outliers-in-machine-learning/#:~:text=In%20machine%20learning%2C%20an%20outlier,errors%2C%20or%20genuine%20rare%20occurrences.
https://medium.com/analytics-vidhya/outlier-detection-in-machine-learning-382557c775aa
https://viso.ai/deep-learning/representation-learning/
https://datawarrior.wordpress.com/2019/09/15/data-representation-in-machine-learning/
https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning
https://www.superannotate.com/blog/overfitting-and-underfitting-in-machine-learning
https://builtin.com/articles/overfitting-vs-underfitting
https://aws.amazon.com/what-is/overfitting/
https://www.appliedaicourse.com/blog/feature-selection-in-machine-learning/
https://medium.com/@nirajan_DataAnalyst/understanding-feature-selection-techniques-in-machine-learning-02e2642ef63e
https://www.stratascratch.com/blog/feature-selection-techniques-in-machine-learning/
https://www.snowflake.com/guides/feature-extraction-machine-learning/#:~:text=Feature%20extraction%20transforms%20raw%20data,the%20redness%20value%20in%20images.

https://www.appliedaicourse.com/blog/feature-extraction-in-machine-learning/
http://linkedin.com/advice/1/what-feature-extraction-machine-learning-skills-machine-learning