21CSA511 - MACHINE LEARNING

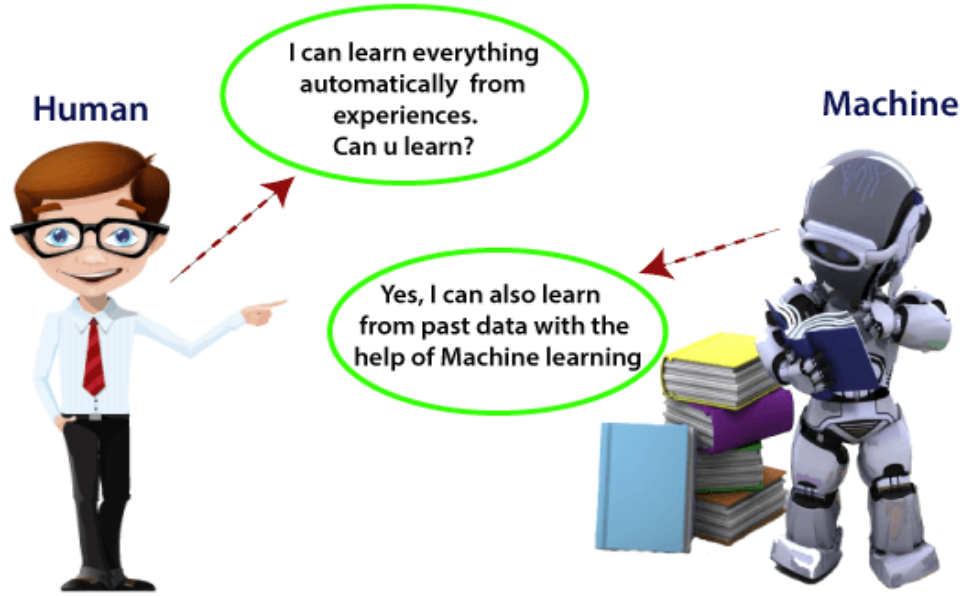
Machine Learning

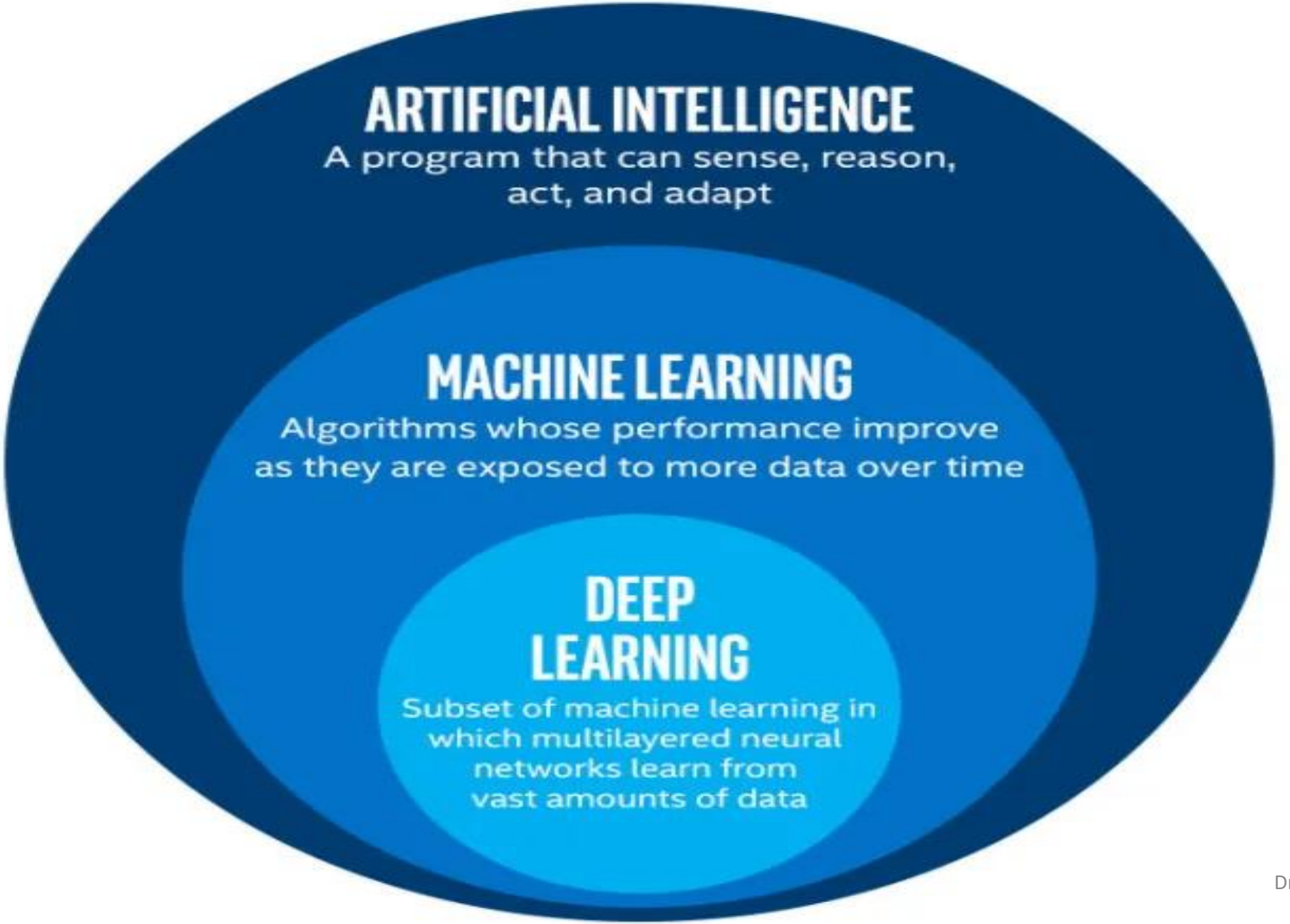
- A rapidly developing field of technology, machine learning **allows computers to automatically learn from previous data**. For building mathematical models and making predictions based on historical data or information, machine learning employs a variety of algorithms. It is currently being used for a variety of tasks, including **speech recognition, email filtering, auto-tagging on Facebook, a recommender system, and image recognition.**

- **A subset of artificial intelligence known as machine learning** focuses primarily on the creation of algorithms that enable a computer to **independently learn from data and previous experiences. Arthur Samuel first used the term "machine learning" in 1959.**

- **Without being explicitly programmed, machine learning enables a machine to automatically learn from data, improve performance from experiences, and predict things.**

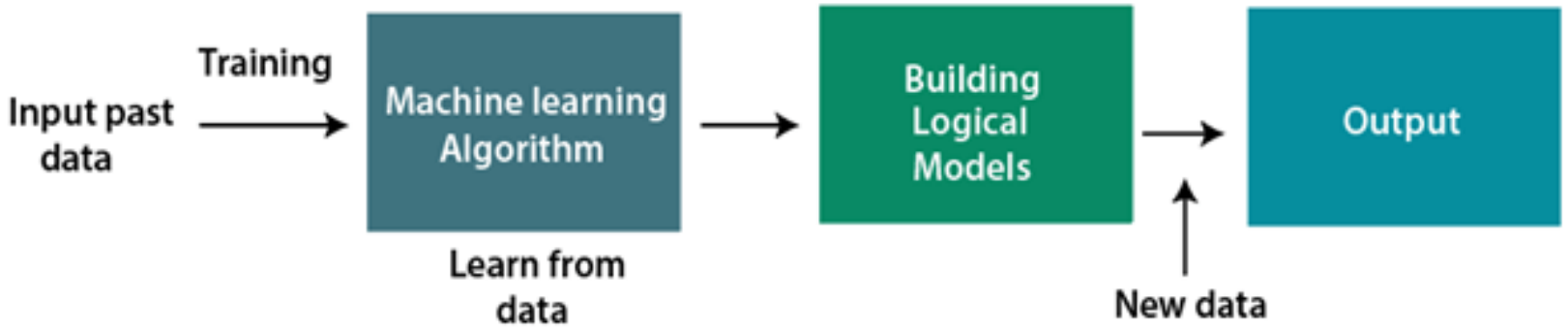
- Machine learning algorithms create a mathematical model that, without being explicitly programmed, aids in making predictions or decisions with the assistance of sample historical data, or training data. **For the purpose of developing predictive models, machine learning brings together statistics and computer science. Algorithms that learn from historical data are either constructed or utilized in machine learning.**




ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

How does Machine Learning work

- A machine learning system builds prediction models, learns from previous data, and predicts the output of new data whenever it receives it. The amount of data helps to build a better model that accurately predicts the output, which in turn affects the accuracy of the predicted output.



Features of Machine Learning:

•Machine learning uses data to **detect various patterns in a given dataset.**

•It can **learn from past data and improve automatically**.

•It is a data-driven technology.

•Machine learning is much similar to data mining as it also deals with the huge amount of the data.

Different types of Machine Learning

It has a main research area that focuses on computer programs that will automatically learn based on the given input data and make intelligent decisions. There are similarities and interrelations between machine learning and data mining. For classification and clustering approaches, machine learning is often applied to predict accuracy. Typical machine learning problems that are utilized in mining are:

1. Supervised learning that makes use of class labels to predict information.

Supervised learning algorithms are used for numerous tasks, including the following:

- **Binary classification.** This divides data into two categories.
- **Multiclass classification.** This chooses among more than two categories.
- **Ensemble modeling.** This combines the predictions of multiple ML models to produce a more accurate prediction.
- **Regression modeling.** This predicts continuous values based on relationships within data.

2. **Unsupervised learning doesn’t use class labels similar to clustering but it will discover new classes within data.**

Unsupervised learning is effective for various tasks, including the following:

- Splitting the data set into groups **based on similarity using clustering algorithms**.
- Identifying **unusual data points in a data set using anomaly detection algorithms.**
- Discovering sets of items in a data set that **frequently occur together using association rule** mining.
- Decreasing the number of variables in a data set using **dimensionality reduction techniques**.

3. **Semi-supervised learning will redefine the boundaries between two classes and makes use of both labeled and unlabeled examples.**

Semi-supervised learning can be used in the following areas, among others:

- **Machine translation.** Algorithms can learn to translate language based on less than a full dictionary of words.
- **Fraud detection.** Algorithms can learn to identify cases of fraud with only a few positive examples.
- **Labeling data.** Algorithms trained on small data sets can learn to automatically apply data labels to larger sets.

4. Reinforcement learning involves programming an algorithm with a distinct goal and a set of rules to follow in achieving that goal. The algorithm **seeks positive rewards for performing actions that move it closer to its goal and avoids punishments for performing actions that move it further from the goal.**

Reinforcement learning is often used for tasks such as the following:

- **Helping robots** learn to perform tasks in the physical world.
- **Teaching bots** to play video games.
- Helping **enterprises plan the allocation** of resources.

Machine learning models and their training algorithms

Supervised learning

Data scientists provide input, output and feedback to build model (as the definition).

EXAMPLE ALGORITHMS:

Linear regressions
- Sales forecasting.
- Risk assessment.

Support vector machines
- Image classification.
- Financial performance comparison.

Decision trees
- Predictive analytics.
- Pricing.

Unsupervised learning

Use deep learning to arrive at conclusions and patterns through unlabeled training data.

EXAMPLE ALGORITHMS:

Apriori
- Sales functions.
- Word associations.
- Searcher.

K-means clustering
- Performance monitoring.
- Searcher intent.

Artificial neural networks
- Generate new, synthetic data.
- Data mining and pattern recognition.

Semi-supervised learning

Builds a model through a mix of labeled and unlabeled data, a set of categories, suggestions and exampled labels.

EXAMPLE ALGORITHMS:

Generative adversarial networks
- Audio and video manipulation.
- Data creation.

Self-trained Naïve Bayes classifier
- Natural language processing.

Reinforcement learning

Self-interpreting but based on a system of rewards and punishments learned through trial and error, seeking maximum reward.

EXAMPLE ALGORITHMS:

Q-learning
- Policy creation.
- Consumption reduction.

Model-based value estimation
- Linear tasks.
- Estimating parameters.

Data in Machine Learning

Data refers to the set of observations or measurements to train a machine learning models. The performance of such models is heavily influenced by both the quality and quantity of data available for training and testing.

Why is Data Crucial in Machine Learning?

Machine learning models cannot be trained without data. Modern advancements in Artificial Intelligence (AI), automation, and data analytics rely heavily on vast datasets.

For instance, **Facebook acquired WhatsApp for $19 billion primarily to access user data, which is critical for enhancing services.**

Properties of Data

Volume: The scale of data generated every millisecond.

Variety: Different data types like healthcare, images, videos, and audio.

Velocity: The speed of data generation and streaming.

Value: The meaningful insights data provides.

Veracity: The accuracy and reliability of data.

Viability: Data's adaptability for integration into systems.

Security: Protection against unauthorized access or manipulation.

Accessibility: Ease of access for decision-making.

Integrity: Accuracy and consistency throughout its lifecycle.

Usability: Simplicity and interpretability for end-users.

Types of Data in Machine Learning Based on Structure

1. Structured Data

- Structured data is organized and stored in a tabular format, such as rows and columns. This type of data is **common in databases and spreadsheets.**
- **Examples**: Sales records, customer information, financial transactions.
- **Usage**: Useful in **supervised learning tasks like regression and classification.**

2. Unstructured Data

- Unstructured data lacks a predefined format, making it **more challenging to process.**
- **Examples**: Text documents, images, videos, audio files.
- **Usage**: Found in **applications like image recognition, natural language processing, and speech-to-text systems.**

3. Semi-Structured Data

- Semi-structured data lies between structured and unstructured data. It has **organizational elements but does not fit neatly into a tabular format.**
- **Examples**: JSON files**, XML files, and NoSQL databases**.
- **Usage**: Often used in web scraping, **API responses, and social media analysis**

Types Data Based on Labeling

- **Labeled Data**: Includes input variables and corresponding target outputs. **Example: Features like "age" and "income"** with a label like "loan approval status."

- **Unlabeled Data**: Contains only input variables without any target labels. **Example: Images without annotations.**

Data into Information and Knowledge

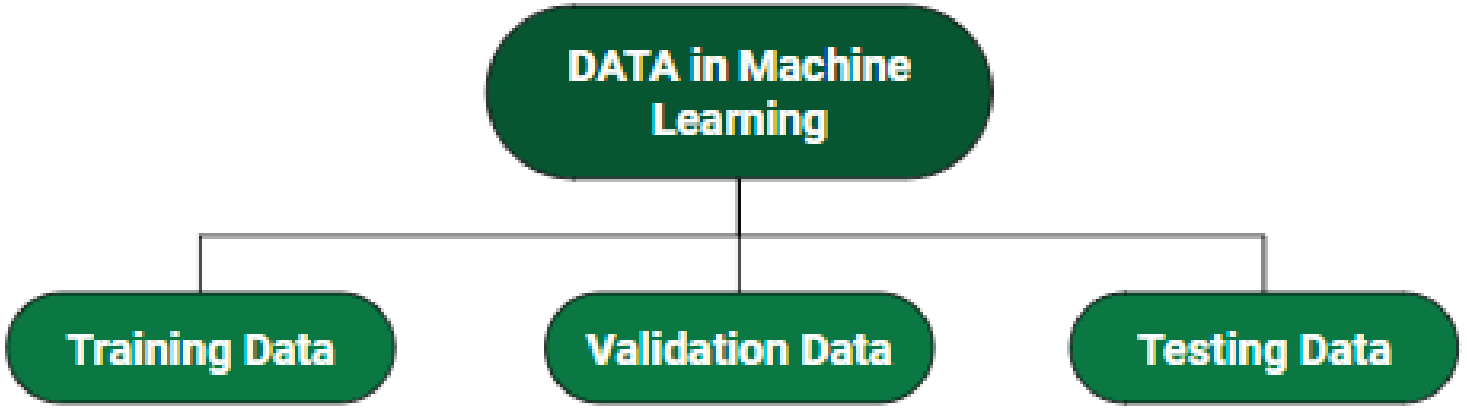
- A shopping mart owner gathers raw survey responses from customers—this is **data**. Manually analyzing thousands of responses is inefficient, so the data is processed into meaningful insights **using tools and calculations, turning it into information**. When **combined with experience and individual perspectives, this information becomes knowledge**, enabling informed decision-making.

Data split in Machine Learning :

Training Data: The part of data we use to train our model. This is the data that – **the model actually sees (both input and output) and learns from.**

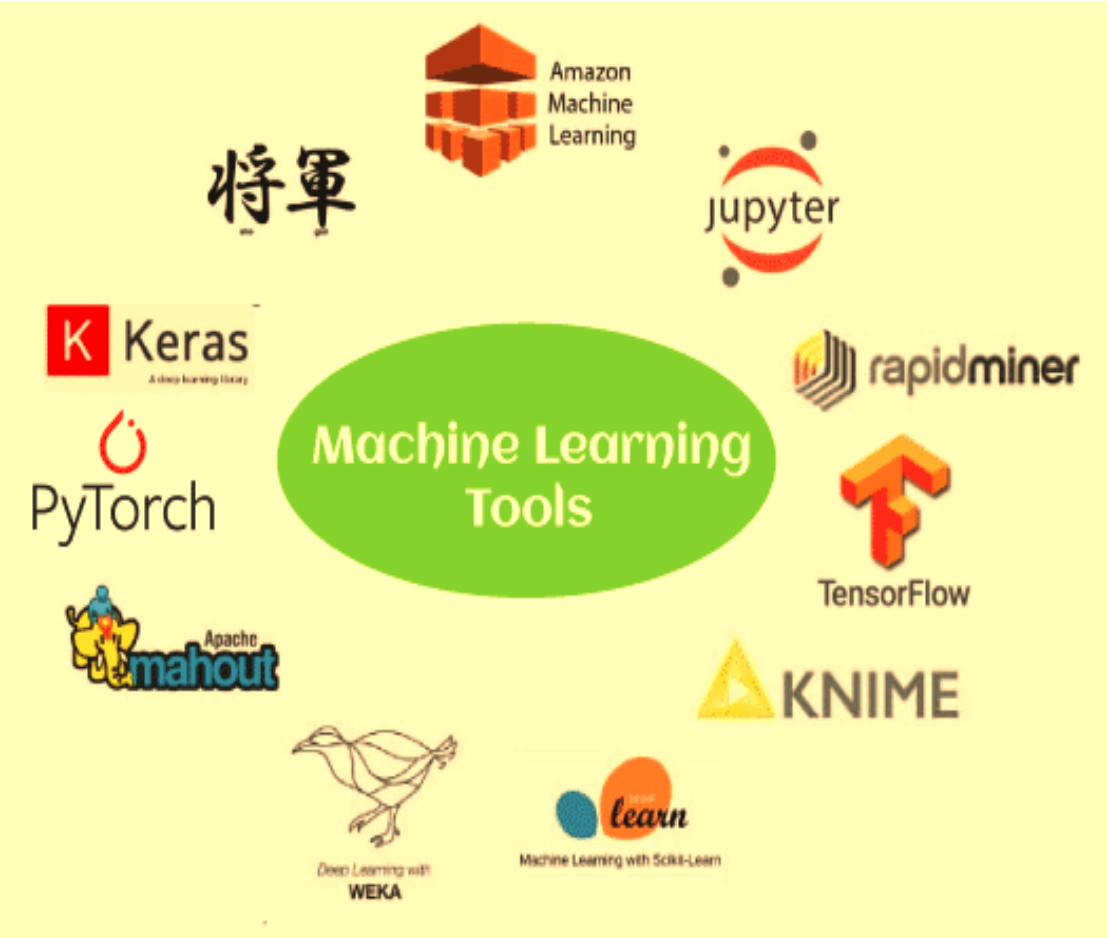
Validation Data: The part of data that is used **to do a frequent evaluation of the model, fit on the training dataset along with improving involved hyperparameters (initially set parameters before the model begins learning)**. This data plays its part when the model is actually training.

Testing Data: Once our model is completely trained, testing data provides an unbiased evaluation. **When we feed in the inputs of Testing data, our model will predict some values(without seeing actual output). After prediction, we evaluate our model by comparing it with the actual output present in the testing data**. This is how we evaluate and see how much our model has learned from the experiences feed in as training data, set at the time of training.



Tools

Machine learning tools help simplify the process of building and deploying machine learning models. They provide ready-made frameworks, libraries, and platforms that make handling tasks like data processing, model training, and deployment easier.



Need of the ML tools:

1.Data Analysis and Interpretation: With the explosion of data in recent years, ML tools are critical for analyzing and interpreting vast amounts of data quickly and efficiently, uncovering patterns and insights that would be impossible for humans to find.

2.Automation: ML enables the automation of decision-making processes and **can perform tasks without human intervention**, increasing efficiency and productivity in various industries.

3.Personalization: ML tools are at the heart of personalization technologies used in e-commerce, content platforms, and marketing. They provide tailored experiences to users based on their behaviors and preferences.

4.Innovation and Competitive Advantage: Businesses that leverage ML tools can innovate faster, creating new products and services that more effectively meet customer needs.

5.Solving Complex Problems: ML tools have the potential to solve complex problems in diverse domains, including **healthcare, finance, environmental protection,** and more, by finding solutions that are not apparent through traditional methods.

1. TensorFlow is an **open-source software library that facilitates numerical computation through data flow graphs**. Developed by the **Google Brain team's researchers and engineers**, it is utilized in both research and production activities within Google.

Key Features

- Extensive library for **deep learning and machine learning**.
- Strong support for research and production projects.

2. Weka is an **open-source Java software suite designed for data mining tasks**. It includes a variety of machine learning algorithms for tasks such as **data pre-processing, classification, regression, clustering, discovering association rules, and data visualization.**

Key Features

- User-friendly interface for **exploring data** and models.
- Wide range of **algorithms for data analysis** tasks.
- **Suitable for developing new machine learning schemes.**

3.Microsoft Azure is a cloud-based environment for training, deploying, automating, managing, and tracking ML models. It is designed to help data scientists and ML engineers leverage their existing data processing and model development skills and frameworks.

Key Features

- Drag-and-drop visual interface (Azure ML Studio).
- Support for popular **ML frameworks** and languages

4.Amazon Machine Learning is a cloud service that makes it easy for professionals of all skill levels to use machine learning technology. It provides **visualization tools and wizards** to create machine learning models **without learning complex ML algorithms** and technology.

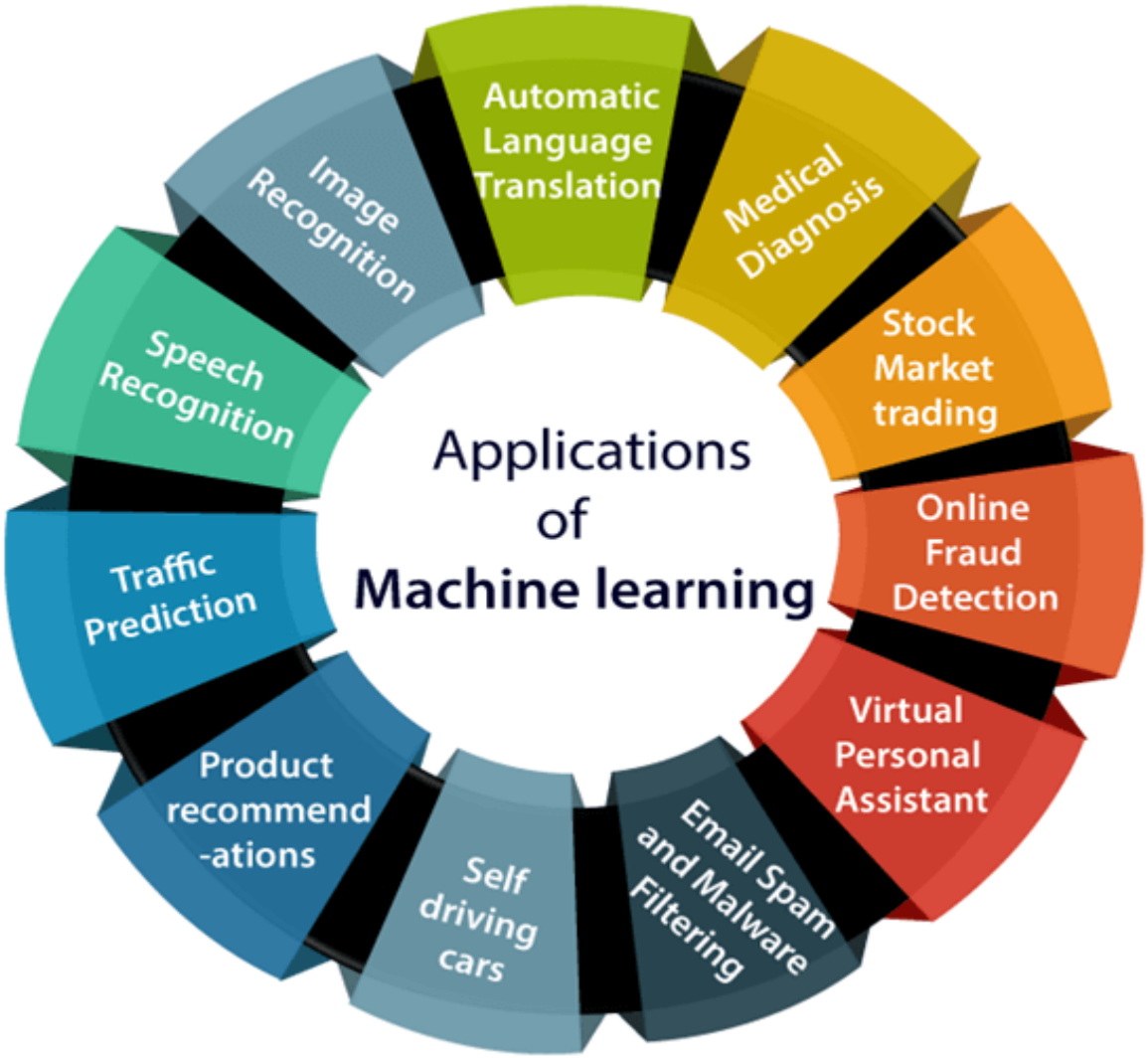
- Key Features
- Easy to **use for creating** ML models.
- **Automatic data transformation and model evaluation.**

5. **jupyter, is a Free software, open standards, and web services for interactive computing across all programming languages**

- A Jupyter Notebook is an open source web application that allows data scientists to create and share documents that include live code, equations, and other multimedia resources.

- JupyterLab is the latest web-based interactive development environment for notebooks, code, and data. Its flexible interface allows users to configure and arrange workflows in data science, scientific computing, computational journalism, and machine learning.

Applications of Machine Learning



1. Image Recognition:

Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:

Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.

It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

- While using **Google, there is an option of** "**Search by voice**," which comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction

- To visit a new place, one takes the help of **Google Maps**, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:
- **Real Time location of the vehicle form Google Map app and sensors**
- **Average time has taken on past days at the same time**.
- **Everyone who is using Google Map is helping this app to make it better**. **It takes information from the user and sends back to its database to improve the performance.**

4. Product recommendations:

- Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, **etc., for product recommendation to the user.** Whenever we search for some product on Amazon, then we started getting **an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.**
- **Google understands** the user interest using various machine learning algorithms and **suggests the product as per customer interest**.
- **As similar, when we use Netflix, we find some recommendations** for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

- One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. **Tesla,** the most popular car manufacturing company is working on self-driving car. **It is using unsupervised learning method to train the car models to detect people and objects while driving.**

6. Email Spam and Malware Filtering:

- Whenever we receive a new email, it is filtered automatically as important, normal, and spam. **We always receive an important mail in our inbox with the important symbol and spam emails in our spam box, and the technology behind this is Machine learning**. Below are some spam filters used by Gmail:
- **Content Filter**
- **Header filter**
- **General blacklists filter**
- **Rules-based filters**
- **Permission filters**

Some machine learning algorithms such as Multi-Layer Perceptron, Decision tree, and Naïve Bayes classifier are used for email spam filtering and malware detection.

7. Virtual Personal Assistant

- There are lot of virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.
- These virtual assistants use machine learning algorithms as an important part.

These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

8. Online Fraud Detection

- Machine learning is making our online transaction safe and secure by detecting fraud transaction. Whenever we perform some online transaction, there may be various ways that a fraudulent transaction can take place such as **fake accounts, fake ids, and steal money in the middle of a transaction. So to detect this, Feed Forward Neural network helps us by checking whether it is a genuine transaction or a fraud transaction.**
- **For each genuine transaction, the output is converted into some hash values, and these values become the input for the next round. For each genuine transaction, there is a specific pattern which gets change for the fraud transaction hence, it detects it and makes our online transactions more secure.**

9. Stock Market trading

Machine learning is widely used in stock market trading. In the stock market, there is always a risk of up and downs in shares, **so for this machine learning's long short term memory neural network is used for the prediction of stock market trends.**

10. Medical Diagnosis

In medical science, machine learning is used for diseases diagnoses. **With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.**

It helps in finding brain tumors and other brain-related diseases easily.

11. Automatic Language Translation

When new places are visited and we are not aware of the language then it is not a problem at all, as for this also **machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.**

The technology behind the automatic translation is a **sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.**

Major Issues (Challenges) in Machine Learning Systems

- The major issues in machine learning systems (data mining research), partitioning them into five groups: mining methodology, user interaction, efficiency and scalability, diversity of data types, and data mining and society.

Mining Methodology

- ***Mining various and new kinds of knowledge:*** Data mining covers a wide spectrum of data analysis and knowledge discovery tasks, from data characterization and discrimination to association and correlation analysis, classification, regression, clustering, outlier analysis, sequence analysis, and trend and evolution analysis. **These tasks may use the same database in different ways and require the development of numerous data mining techniques.**

- **Mining knowledge in multidimensional space: one can search for interesting patterns among combinations of dimensions (attributes) at varying levels of abstraction**. Such mining is known as (exploratory) multidimensional data mining.

- ***Data mining—an interdisciplinary effort:*** The power of data mining can be substantially enhanced by integrating new methods from multiple disciplines. **For example, to mine data with natural language text, it makes sense to fuse data mining methods with methods of information retrieval and natural language processing.**

- ***Boosting the power of discovery in a networked environment:*** Knowledge **derived in one set of objects** can be used to **boost the discovery of knowledge in a "related" or semantically linked set of objects.**

- ***Handling uncertainty, noise, or incompleteness of data:*** Errors and noise may confuse the data mining process, leading to the derivation of erroneous patterns. Data cleaning, data preprocessing, outlier detection and removal, and uncertainty reasoning are examples of techniques that need to be integrated with the data mining process.

- ***Pattern evaluation and pattern- or constraint-guided mining:*** What makes a pattern interesting may vary from user to user. Therefore, **techniques are needed to assess the interestingness of discovered patterns based on subjective measures** - by using interestingness measures or user-specified constraints to *guide* the discovery process, we may **generate more interesting patterns and reduce the search space.**

User Interaction

- **Interactive mining:** It would allow users to dynamically change the focus of a search, to refine mining requests based **on returned results, and to drill, dice, and pivot through the data and knowledge space interactively**, dynamically exploring "cube space" while mining.

- **Incorporation of background knowledge :** To guide discovery process and to express the discovered patterns, the background knowledge can be used. **Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple levels of abstraction**.

- Ad hoc data mining and data mining query languages: **Query languages (e.g., SQL) have played an important role in flexible searching because they allow users to pose ad hoc queries.** Optimization of the processing of such flexible mining requests is another promising area of study.

- Presentation and visualization of data mining results − Once the patterns are discovered it needs to be expressed in high level languages, and visual representations. **These representations should be easily understandable.**

Efficiency and Scalability

- **Efficiency and scalability of data mining algorithms** − In order to effectively extract the information from huge amount of data in databases, **data mining algorithm must be efficient and scalable.**

- **Parallel, distributed, and incremental mining algorithms** − The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. **These algorithms divide the data into partitions which is further processed in a parallel fashion. Then the results from the partitions are merged. The incremental algorithms, update databases without mining the data again from scratch.**

- **Cloud computing and cluster computing**, which use computers in a distributed and collaborative way to tackle very large-scale computational tasks, are also active research themes in parallel data mining.

Diverse Data Types Issues

- **Handling of relational and complex types of data** − The database may contain complex data objects, multimedia data objects, spatial data, temporal data etc. **It is not possible for one system to mine all these kind of data.**

- **Mining dynamic, networked, and global data repositories-** The data is available at different data sources on LAN or WAN. These **data source may be structured, semi structured or unstructured.** Therefore mining the knowledge from them adds challenges to data mining.

Data Mining and Society

- ***Social impacts of data mining-*** The improper disclosure or use of data and the **potential violation of individual privacy and data protection rights are areas of concern that need to be addressed.**

- ***Privacy-preserving data mining-*** **it poses the risk of disclosing an individual's personal information.** Studies on privacy-preserving data publishing and data mining are ongoing. The philosophy is to observe data sensitivity and preserve people's privacy while performing successful data mining.

- ***Invisible data mining-*** **Intelligent search engines and Internet-based stores perform such *invisible data mining* by incorporating data mining into their components to improve their functionality and performance**. This is done often unbeknownst to the user. For example, when purchasing items online, users may be unaware that the store is likely collecting data on the buying patterns of its customers, which may be used to recommend other items for purchase in the future.

Goals Of ML

The goal of machine learning is to develop algorithms that can automatically learn from data and improve their performance on a specific task. The process of machine learning involves four key steps: data preparation, model training, model evaluation, and model deployment.

• **Data preparation** is the process of collecting and cleaning data to make it usable for machine learning. The quality and quantity of the data used in machine learning are crucial to the success of the algorithm. The data should be relevant to the problem being solved, accurate, and representative of the population.

• **Model training** involves feeding the algorithm with the prepared data, and it is the process by which the algorithm learns the patterns and relationships in the data. The algorithm can use different methods such as supervised, unsupervised, and reinforcement learning to learn from the data.

• In supervised learning, the algorithm is trained on labeled data, where the correct output is already known.

• In unsupervised learning, the algorithm is trained on unlabeled data and has to find patterns and relationships on its own. Reinforcement learning involves learning through trial and error by rewarding or punishing the algorithm for its actions.

• **Model evaluation** involves testing the algorithm's performance on a separate set of data. This is done to ensure that the algorithm can generalize well to new data and has not overfit to the training data. Overfitting occurs when the algorithm performs well on the training data but fails to generalize to new data.

• **Model deployment** involves integrating the trained algorithm into a real-world application. This is done by making the algorithm available through an **application programming interface (API)** or embedding it into a larger system.

Goals Of ML

Machine learning has several advantages, including:

- **Improved decision-making:** Machine learning algorithms can analyze vast amounts of data and **provide insights that would be difficult or impossible for humans to identify.** This can help businesses make better decisions, improve customer experiences, and increase efficiency.

- **Increased efficiency:** Machine learning algorithms can **automate repetitive tasks and processes, freeing up time and resources for other tasks.** This can lead to increased efficiency and productivity in the workplace.

- **Personalization:** Machine learning algorithms can analyze customer data to provide **personalized recommendations, offers, and experiences.** This can improve customer satisfaction and loyalty.

- **Scalability:** Machine learning algorithms can process large amounts of data quickly and efficiently, making them scalable to handle big data applications.

- **Continuous improvement:** Machine learning algorithms can learn from new data and improve their accuracy over time. This can lead to continuous improvement in performance and results.

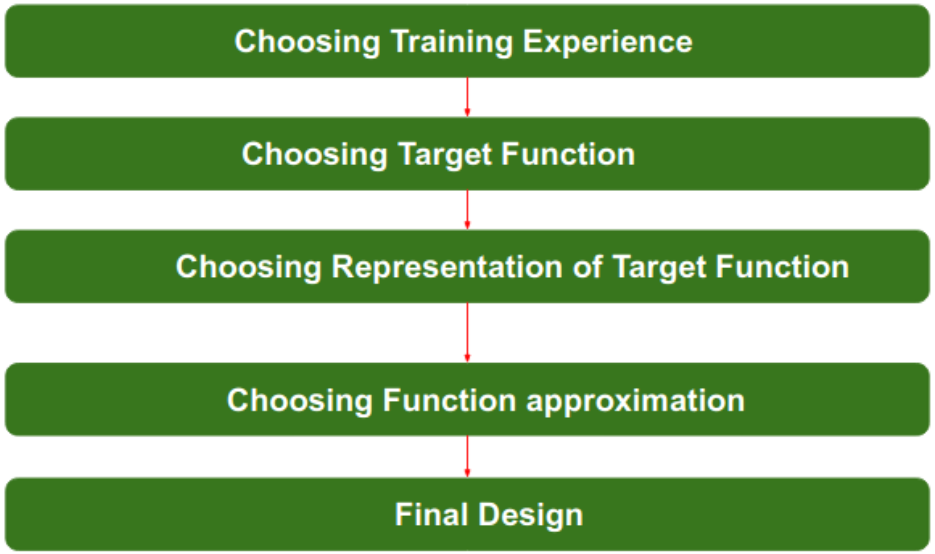
Aspects of developing systems (Machine Learning)

According to Tom Mitchell, "A computer program is said to be learning from experience (E), with respect to some task (T). Thus, the performance measure (P) is the performance at task T, which is measured by P, and it improves with experience E."

Example: In Spam E-Mail detection,

- **Task, T:** To classify mails into Spam or Not Spam.
- **Performance measure, P:** Total **percent of mails being correctly classified as being "Spam" or "Not Spam".**
- **Experience, E: Set of Mails with label "Spam"**

Steps for Designing Learning System are:



Step 1 - Choosing the Training Experience: The very important and first task is to choose the training data or training experience which will be fed to the Machine Learning Algorithm. **It is important to note that the data or experience that we fed to the algorithm must have a significant impact on the Success or Failure of the Model. So Training data or experience should be chosen wisely.**

- **Below are the attributes which will impact on Success and Failure of Data:**
- The training experience will be able to provide **direct or indirect feedback regarding choices**. For **example:** While **Playing chess the training data will provide feedback** to itself like instead of this move if this is chosen the chances of success increases.
- **Second important attribute is the degree to which the learner will control the sequences of training examples**. For example: when **training data is fed to the machine then at that time accuracy is very less but when it gains experience while playing again and again with itself** or opponent the machine algorithm will get feedback and control the chess game accordingly.
- **Third important attribute is how it will represent the distribution of examples over which performance will be measured**. For example, a Machine learning algorithm will get experience while going through a number of different cases and different examples. **Thus, Machine Learning Algorithm will get more and more experience by passing through more and more examples and hence its performance will increase.**

Step 2- Choosing target function: The next important step is choosing the target function. It means according to **the knowledge fed to the algorithm the machine learning will choose NextMove function which will describe what type of legal moves should be taken.** For example : While playing chess with the opponent, when opponent will play then the machine learning algorithm will decide what be the number of possible legal moves taken in order to get success.

Step 3- Choosing Representation for Target function: When the machine algorithm will know all the possible legal moves **the next step is to choose the optimized move using any representation i.e. using linear Equations, Hierarchical Graph Representation, Tabular form** etc. The NextMove function will move the Target move like **out of these move which will provide more success rate**. For Example : while playing chess machine have 4 possible moves, so the machine will choose that optimized move which will provide success to it.

Step 4- Choosing Function Approximation Algorithm: An optimized move cannot be chosen just with the training data. The training data had to go through with set of example and through these examples the training data will approximates which steps are chosen and after that machine will provide feedback on it. For Example : When a training data of Playing chess is **fed to algorithm so at that time it is not machine algorithm will fail or get success and again from that failure or success it will measure while next move what step should be chosen and what is its success rate.**

Step 5- Final Design: The final design is **created at last when system goes from number of examples , failures and success , correct and incorrect decision and what will be the next step etc.** Example: **DeepBlue is an intelligent computer which is ML-based won chess game against the chess expert Garry Kasparov**, and it became the first computer which had beaten a human chess expert.

Training data and Testing data

- The ***training data is the biggest (in -size) subset of the original dataset, which is used to train or fit the machine learning model***.

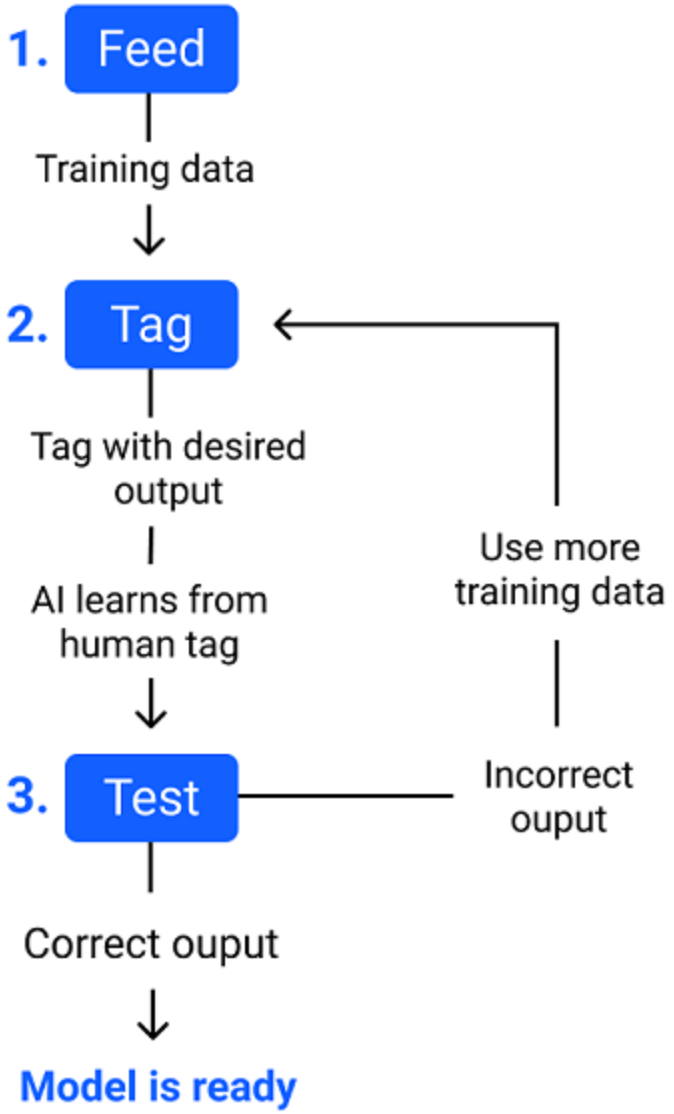
- ***The test dataset is another subset of original data, which is independent of the training dataset***. However, it has some similar types of features and class probability distribution and uses it as a benchmark for model evaluation once the model training is completed. **Test data is a well-organized dataset that contains data for each type of scenario for a given problem that the model would be facing when used in the real world**. Usually, the test dataset is approximately 20-25% of the total original data for an ML project.

- **Training data teaches a machine learning model how to behave, whereas testing data assesses how well the model has learned.**

- **Training Data:** The machine learning model is **taught how to generate predictions** or perform a specific task using training data. Since it is usually identified, every data point's output from the model is known. **In order to provide predictions, the model must first learn to recognize patterns in the data.** Training data can be compared to a student's textbook when learning a new subject. **The learner learns by reading the text and completing the tasks, and the book offers all the knowledge they require.**

- **Testing Data:** The **performance of the machine learning model is measured using testing data**. Usually, it is labeled and distinct from the training set. This indicates that for every data point, the model's result is unknown. **On the testing data, the model's accuracy in predicting outcomes is assessed. Testing data is comparable to the exam a student takes to determine how well-versed in a subject they are**. The test asks questions that the student must respond to, and the test results are used to gauge the student's comprehension.

How do training and testing data work in ML



Need of Splitting dataset into Train and Test set

- **Splitting the dataset** into train and test sets is one of the **important parts of data pre-processing**, as by doing so, one can improve the performance of our model and hence give **better predictability**.

- one can understand it as if, one trains the model with a training set and **then test it with a completely different test dataset, and then the model would not be able to understand the correlations** between the features.

- Therefore, **if one trains and tests the model with two different datasets, then it will decrease the performance of the model**. Hence it is important to split a dataset into two parts, i.e., train and test set.

Training data vs. Testing Data

- The main difference between training data and testing data is that training data is the subset of original data that is used to train the machine learning model, whereas testing data is used to check the accuracy of the model.

- **The training dataset is generally larger in size compared to the testing dataset. The general ratios of splitting train and test datasets are 80:20, 70:30, or 90:10.**

- **Training data is well known to the model as it is used to train the model, whereas testing data is like unseen/new data to the model.**

Classification Errors

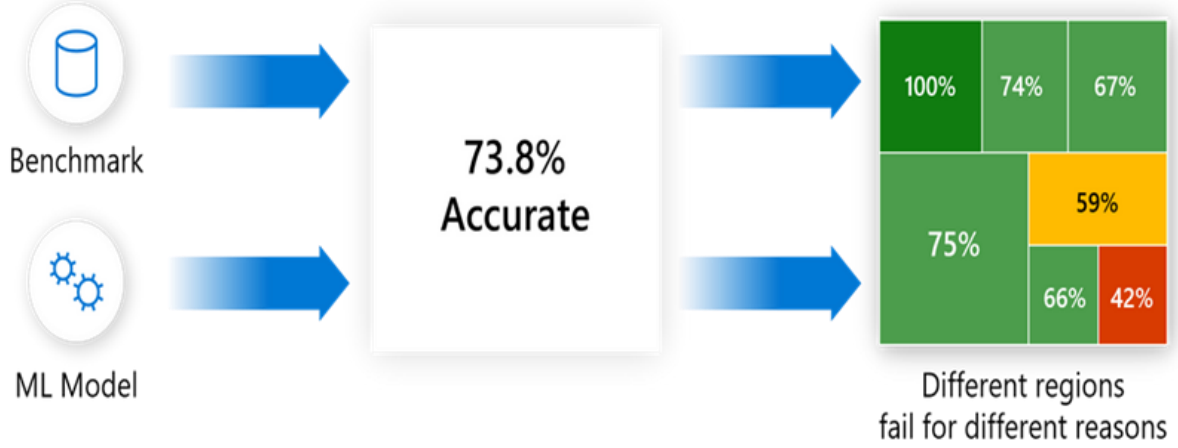
Classification error refers to the misclassification that occurs when data points belonging to different classes overlap in a probability distribution, leading to incorrect assignment of entities to classes based on the highest probability measure.

In machine learning, classification errors occur when a model incorrectly predicts the class of an instance. These errors can be categorized in several ways, depending on the nature of the problem and evaluation metrics.

Error Analysis

Error analysis is the process to isolate, observe and diagnose erroneous ML predictions thereby helping understand pockets of high and low performance of the model. **When it is said that "the model accuracy is 90%" it might not be uniform across subgroups of data and there might be some input conditions which the model fails more.**

An example might be that a dog detection image recognition model might be doing better for dogs in an outdoor setting but not so good in low-lit indoor settings. The below illustration provides a view of how moving from aggregate to group-wise errors provides a better picture of model performance.



Types of Classification Errors

a. Misclassification Error

- **Definition:** When the predicted class is not the same as the true class.
- **Formula: Misclassification Error Rate=Number of Incorrect Predictions/Total Predictions**
- Related Metric: **Accuracy = 1- Misclassification Rate.**

b. Type I and Type II Errors (Binary Classification)

- **Type I Error (False Positive):**
  - The model **incorrectly predicts a positive class when the true class is negative.**
  - Example: Predicting a healthy patient as sick.
- **Type II Error (False Negative):**
  - The model **incorrectly predicts a negative class when the true class is positive.**
  - Example: Predicting a sick patient as healthy.

False Positive and False Negative Rates

- **False Positive Rate (FPR):** FPR=False Positives/True Negatives + False Positives
- **False Negative Rate (FNR):** FNR=False Negatives/True Positives + False Negatives

Causes of Classification Errors

- **Insufficient or Imbalanced Data**: Poor class representation leads to bias in predictions.
- **Overfitting or Underfitting:** Overfitting causes the model to perform well on training data but poorly on unseen data.Underfitting means the model cannot capture the patterns in the training data.
- **Model Complexity**: Too simple or too complex models may lead to classification errors.
- **Feature Selection**: Irrelevant or missing features can mislead the model.
- **Noise in Data**: Incorrect labels or noisy features can degrade performance.

Metrics to Evaluate Classification Errors

- **Confusion Matrix:**
  - Provides a detailed breakdown of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

- **Precision:**Measures the fraction of **correctly predicted positive instances:**

  Precision= True Positives / False Positives + True Positives

- **Recall (Sensitivity):**Measures the fraction **of true positives correctly predicted:**

  Recall= True Positives/ False Negatives + True Positives

- **F1-Score:**Harmonic **mean of precision and recall:**

  F1-Score=2·Precision·Recall/Precision + Recall

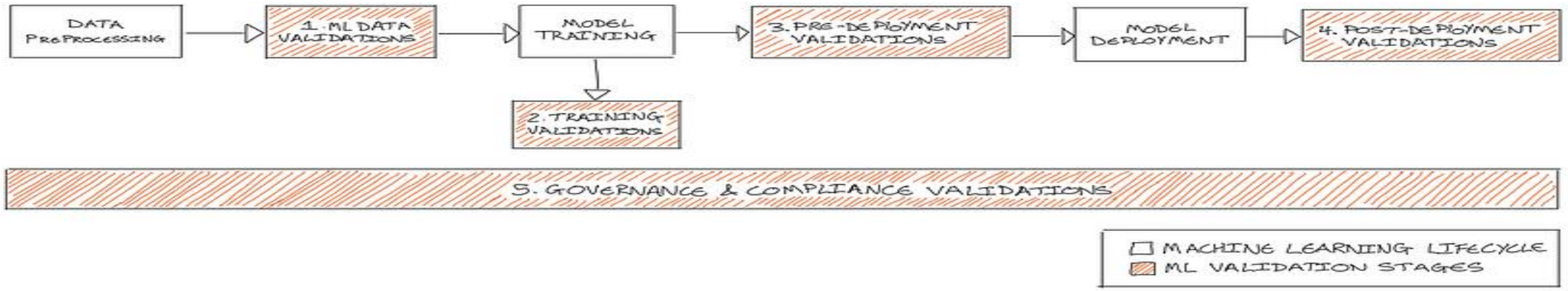
- **ROC Curve and AUC: Evaluates model performance across different classification thresholds.**

Minimizing Classification Errors

- **Improve Data Quality:**
  - Ensure **balanced datasets and correct labeling.**
- **Feature Engineering:**
  - **Select relevant features and create new ones** if necessary.
- **Hyperparameter Tuning:**
  - Optimize parameters using methods like **grid search or Bayesian optimization**.
- **Model Selection:**
  - Use **ensemble methods** (e.g., Random Forest, Gradient Boosting) for better performance.
- **Regularization:**
  - **Prevent overfitting by adding penalties to the loss function** (e.g., L1, L2 regularization).
- **Cross-Validation:**
  - Evaluate the model on multiple subsets of data to ensure generalization.

Validation

Validation in machine learning is a crucial step to evaluate how well a model generalizes to unseen data and to ensure it is neither overfitting nor underfitting. It involves splitting the available dataset into parts to train and assess the model's performance before final deployment.



Purpose of Validation

Generalization Check: Ensures the model **performs well on new, unseen data**.

Hyperparameter Tuning: Helps **identify the best set of hyperparameters** for the model.

Prevent Overfitting: **Avoids memorizing training data instead of learning patterns.**

Model Selection: Enables comparison of multiple models to choose the best-performing one.

Types of (Cross-) Validation

a. Holdout Validation

In Holdout Validation, **we perform training on the 50% of the given dataset and rest 50% is used for the testing purpose. It's a simple and quick way to evaluate a model**. The major drawback of this method is that we perform training on the 50% of the dataset, it may possible that the remaining 50% of the data contains some important information which we are leaving while training our model **i.e. higher bias.**

Advantages: Simple and fast.

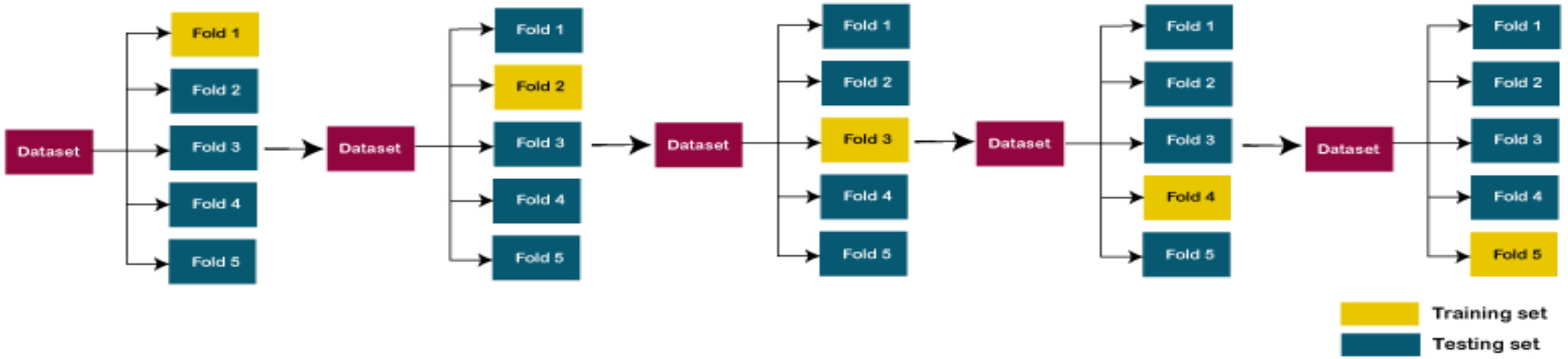
Disadvantages: **Performance depends heavily on how the data is split.**

b. K-Fold Cross-Validation

- K-fold cross-validation approach divides **the input dataset into K groups of samples of equal sizes. These samples are called folds**. For each learning set**, the prediction function uses k-1 folds, and the rest of the folds are used for the test set.**

The steps for k-fold cross-validation are:

- Split the input dataset into K groups
- For each group:
  - Take one group as the reserve or test data set.
  - Use remaining groups as the training dataset.
  - Fit the model on the training set and evaluate the performance of the model using the test set.



Advantages: **Reduces bias and variance due to splitting.**
Suitable for small datasets.
Disadvantages:**Computationally expensive for large datasets.**

Note:
While making predictions, **a difference occurs between prediction values made by the model and actual values/expected values,** and this difference is known as **bias errors or Errors due to bias**

Variance tells that how much **a random variable is different from its expected value.**

C. Stratified k-fold cross-validation

- This technique is similar to k-fold cross-validation with some little changes. **This approach works on stratification concept, it is a process of rearranging the data to ensure that each fold or group is a good representative of the complete dataset. To deal with the bias and variance, it is one of the best approaches.**
- It can be understood with an example of housing prices, such that the price of some houses can be much high than other houses. To tackle such situations, a stratified k-fold cross-validation technique is useful.

D. Leave one out cross-validation - LOOCV

- There is a **need to take 1 dataset out of training**. It means, in this approach, **for each learning set, only one datapoint is reserved, and the remaining dataset is used to train the model.** This process repeats for each datapoint. **Hence for n samples, we get n different training set and n test set.** It has the following features:
- In this approach, **the bias is minimum** as all the data points are used.
- The process is executed for **n times; hence execution time is high.**
- This approach leads to **high variation** in testing the effectiveness of the model as we iteratively check against one data point.
- Advantages:
  - Exhaustive and **unbiased.**
- Disadvantages:
  - **Computationally expensive.**
  - **Sensitive to noise.**

Common Challenges in Validation

- **Data Leakage:** Ensure **no information from the validation set leaks into the training process.**
- **Imbalanced Datasets:** Use **stratified sampling to ensure proper class representation.**
- **Overfitting on Validation Set: Avoid excessive hyperparameter tuning on the validation set**, as it can lead to overfitting.

Best Practices for Validation

- **Shuffle Data: Randomize data before splitting** (except for time-series data).
- **Handle Imbalanced Data: Use stratified splits or class weighting** for balanced representation.
- **Avoid Data Leakage: Ensure no overlap between training and validation/test data** (e.g., same user IDs in training and testing).
- **Combine Techniques:** Use a **mix of validation strategies** for robust evaluation.
- **Early Stopping:** Use **validation performance to stop training** when the model begins overfitting.

Validation Metrics

Common metrics used to evaluate model performance on the validation set:

- **Classification Metrics:**
  - Accuracy, Precision, Recall, F1-Score
  - ROC-AUC
- **Regression Metrics:**
  - **Mean Absolute Error (MAE)** - It's a measurement of the typical **absolute discrepancies between a dataset's actual values and projected values.**
  - **Mean Squared Error (MSE)** - It measures the square root of the **average discrepancies between a dataset's actual values and projected values.**
  - **R-squared** - It quantifies **the percentage of the dependent variable's variation that the model's independent variables contribute to.**

Linear Regression in Machine learning

- **Linear regression** is also a type of **supervised machine-learning algorithm** that learns from the labelled datasets and **maps the data points with most optimized linear functions which can be used for prediction on new datasets.** It **computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation with observed data.** It predicts the continuous output variables based on the independent input variable.
- For example if we want to predict house price we consider **various factor such as house age, distance from the main road, location, area and number of room, linear regression uses all these parameter to predict house price** as it consider a linear relation between all these features and price of house.
- Linear Regression is a fundamental supervised learning algorithm used for predicting continuous numerical values. It establishes a linear relationship between input features (X) and the target variable (Y).

Types of Linear Regression

- Simple linear regression is the simplest form of linear regression involving **only one independent variable and one dependent variable.**
- Multiple linear regression involves **more than one independent variable and one dependent variable**.

Equation of Linear Regression

The general form of a linear regression model is:

$$Y=\beta 0+\beta 1X1+\beta 2X2+...+\beta nXn+\epsilon$$

Where:

Y = Target variable (dependent variable)

X 1 ,X 2 ,...,X n = Input features (independent variables)

$\beta 0$ = Intercept (bias term)

$\beta 1,\beta 2,...,\beta n$ = Coefficients (weights) for the input features

ϵ = Error term (random noise)

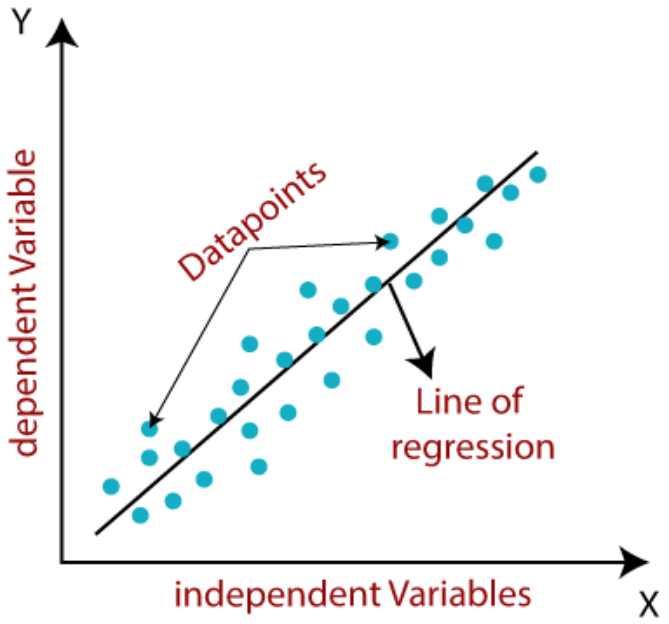
For a simple linear regression (one feature X):

$$Y=\beta 0+\beta 1X+\epsilon$$

For multiple linear regression (multiple features X 1 ,X 2 ,...,X n):

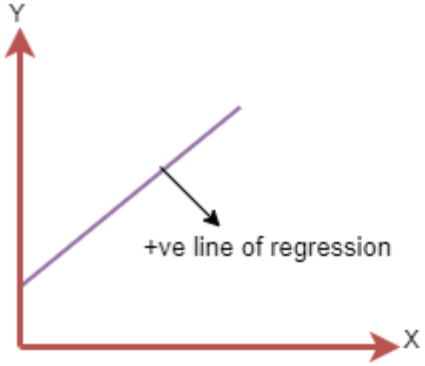
$$Y=\beta 0+\sum(i=1 - n)\ \beta iXi+\epsilon$$

The linear regression model **provides a sloped straight line representing the relationship** between the variables.



Positive Linear Relationship:

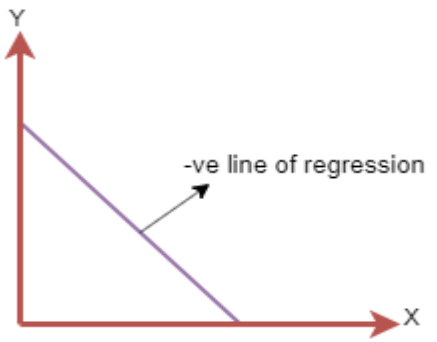
If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relatio



The line equation will be: $Y= a_0+a_1x$

Negative Linear Relationship:

If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear



The line of equation will be: $Y= -a_0+a_1x$

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).

The different values for weights or the coefficient of lines (a_0, a_1) gives a different line of regression, so we need to calculate the best values for a_0 and a_1 to find the best fit line, so to calculate this we use cost function.

Cost function

- **The different values for weights or coefficient of lines (a_0, a_1) gives the different line of regression, therefore there is a need of cost function, to estimate the values of the coefficient for the best fit line.**
- Cost function **optimizes the regression coefficients or weights**. It measures how a linear regression model is performing.
- We can use the **cost function to find the accuracy of the mapping function, which maps the input variable to the output variable. This mapping function is also known as Hypothesis function.**
- For Linear Regression, we **use the Mean Squared Error (MSE) cost function,** which is the mean of squared error occurred between the predicted values and actual values. It can be written as:
- For the above linear equation, MSE can be calculated as:

$$MSE = 1\frac{1}{N}\sum_{i=1}^{n}(y_i - (a_1x_i + a_0))^2$$

Where,
N=Total number of observation
Yi = Actual value
$(a1x_i+a_0)$= Predicted value.

Residuals:

- The **distance between the actual value and predicted values is called residual.**
- If the observed **points are far from the regression line, then the residual will be high, and so cost function will high.**
- If the scatter points **are close to the regression line, then the residual will be small and hence the cost function.**

Gradient Descent:

- **Gradient descent is used to minimize the MSE by calculating the gradient of the cost function.**
- A **regression model uses gradient descent to update the coefficients of the line by reducing the cost function.**
- It is done by a **random selection of values of coefficient and then iteratively update the values to reach the minimum cost function.**

Sum of Squared Errors (SSE) in Machine Learning

One of the most common metrics used for evaluating regression models is **Sum of Squared Errors (SSE)**. **SSE measures how well a model's predictions align with actual values. Understanding SSE and its role in model evaluation can help in building more accurate and reliable machine learning models.**

The **Sum of Squared Errors (SSE)** is a measure of how well a model's predictions match the actual values. It is commonly used in regression models to quantify the total squared difference between the predicted values and the actual values.

Formula:

$$\text{SSE} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

- y_i is the actual value for the i-th data point,
- $\hat{y}_i$ is the predicted value for the i-th data point,
- n is the total number of data points.

- The result of SSE is a single number that represents the total error in the predictions.

Key Insights:

•**Lower SSE** means the model's predictions are close to the actual values (better fit).

•**Higher SSE** indicates poor model performance.

•**SSE is used in optimization algorithms** like Gradient Descent to minimize error.

Simple Linear Regression

Dependent Variable (Y)

Num months of BONUS

$$Y_i = b_0 + b_1 X_i + \varepsilon_i$$

Least Squares Method

Estimate line of best fit

Criteria: **min**$(\sum \varepsilon_i^2)$

Sum of Squared Errors (SSE)

ε_i

b_1 *slope coefficient*

intercept b_0

0

Company PROFIT

Independent Variable (X)

Why Use Sum of Squared Errors

1. **Minimizing Error**: SSE provides a straightforward way to measure the accuracy of a model. By minimizing SSE, we can improve a model's predictions. This approach is commonly used in optimization algorithms like **Gradient Descent** to update model parameters and reduce prediction error.
2. **Sensitivity to Outliers**: Because SSE squares each error, it gives more weight to larger errors. This sensitivity can be advantageous if we want our model to be highly responsive to large deviations, but it also means SSE may be affected by outliers. If a few predictions are way off, they will significantly increase the SSE.
3. **Comparing Models**: SSE can be used to compare different models. Given the same dataset, the model with the lower SSE is generally considered to be the better model, as it indicates a closer fit to the data.

Limitations of Sum of Squared Errors

- While SSE is a useful metric, it has some limitations:
- **Sensitive to Outliers**: Since SSE squares each error, it gives a high weight to large errors. Outliers can significantly increase SSE, which may lead to inaccurate conclusions about model performance.

- **Does Not Scale Well**: SSE values depend on the number of data points, which means they can vary widely depending on dataset size. To address this, metrics like Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) are sometimes preferred, as they average the squared errors and are easier to interpret.
- **Does Not Provide Interpretability on its Own**: SSE on its own does not provide an easily interpretable measure of model performance relative to the range of the data. Other metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE) are often used alongside SSE for a more complete evaluation.

- **Example of Sum of Squared Errors**

Given the Actual and predicted values, Calculate the **Sum of Squared Errors**

Actual Value (y_i)	Predicted Value ($\hat{y}_i$)
5	4.8
7	7.5
3	2.9
6	5.7
8	8.2

SSE - ?

Python Code for SSE Calculation

```
import numpy as np

# Actual values (y) and Predicted values (y_hat)
y_actual = np.array([3, -0.5, 2, 7])
y_pred = np.array([2.5, 0.0, 2, 8])

# Compute SSE
sse = np.sum((y_actual - y_pred) ** 2)

print("Sum of Squared Errors (SSE):", sse)
```

-- SSE - ?

Gradient Descent in Machine Learning

- **Gradient Descent is known as one of the most commonly used optimization algorithms to train machine learning models by means of minimizing errors between actual and expected results. Further, gradient descent is also used to train Neural Networks.**

- In mathematical terminology, Optimization algorithm refers to the task of minimizing/maximizing an objective function f(x) parameterized by x. Similarly, in machine learning, optimization is the task of minimizing the cost function parameterized by the model's parameters. **The main objective of gradient descent is to minimize the convex function using iteration of parameter updates.** Once these machine learning models are **optimized, these models can be used as powerful tools for Artificial Intelligence and various computer science applications.**

Gradient Descent or Steepest Descent

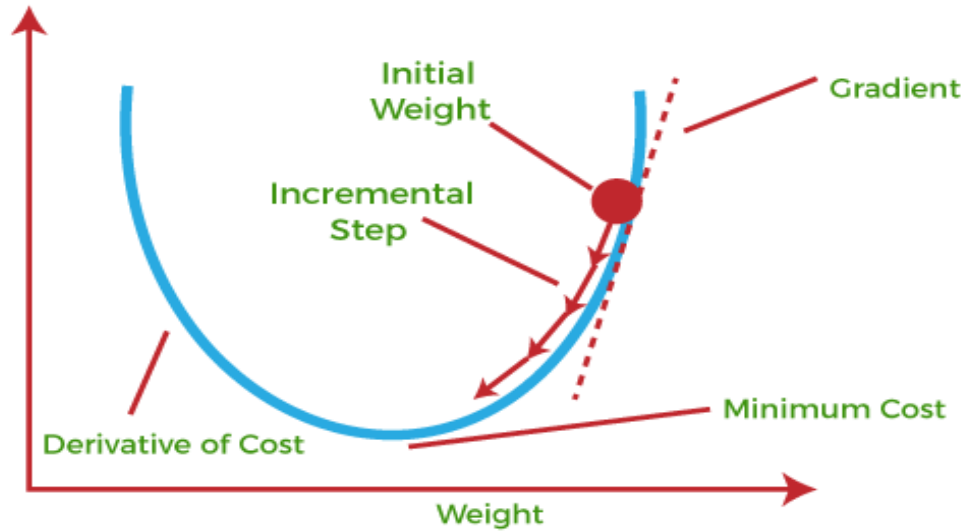
- Gradient descent was initially **discovered by "Augustin-Louis Cauchy" in mid of 18th century.** Gradient Descent is defined as one of the most commonly used iterative optimization algorithms of machine learning to train the machine learning and deep learning models. **It helps in finding the local minimum of a function.**

The best way to define the local minimum or local maximum of a function using gradient descent is as follows:

- If one moves toward a **negative gradient or away from the gradient of the function** at the current point, it will give the **local minimum** of that function.
- Whenever one moves toward a **positive gradient or towards the gradient of the function** at the current point, we will get the **local maximum** of that function.

***The main objective of using a gradient descent algorithm is to minimize the cost function using iteration*. To achieve this goal, it performs two steps iteratively:**

- Calculates the **first-order derivative of the function to compute the gradient or slope of that function.**
- Move away from the direction of the gradient, **which means slope increased from the current point by alpha times, where Alpha is defined as Learning Rate.** It is a tuning parameter in the optimization process which helps to decide the length of the steps.



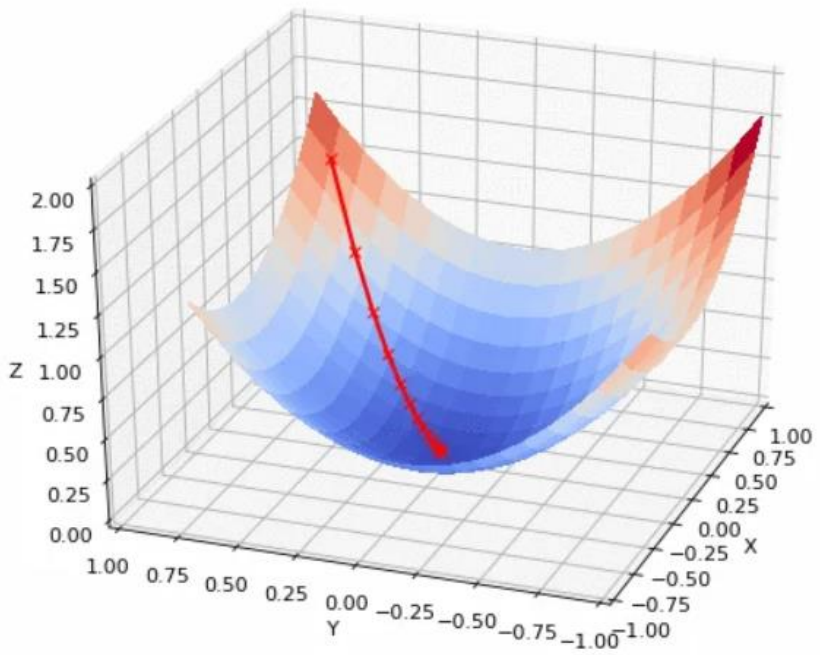
- **The cost function is defined as the measurement of difference or error between actual values and expected values at the current position and present in the form of a single real number.** It helps to increase and improve machine learning efficiency by providing feedback to this model so that it can minimize error and find the local or global minimum.

- Further, **it continuously iterates along the direction of the negative gradient until the cost function approaches zero**. **At this steepest descent point, the model will stop learning further**

- **The cost function is calculated after making a hypothesis with initial parameters and modifying these parameters using gradient descent algorithms over known data to reduce the cost function.**

The primary **function of a gradient is to measure the change in each weight against the change in the errors**. Think of gradients as the slope of a function. **The slope will be steeper the higher the gradient - this is a favorable condition for models because they can learn quickly. However, the model will stop learning if the slope becomes zero.**

In the **implementation part**, one has to write **two functions.** One will be the **cost function which takes the actual output and predicted output as input and returns the loss. The second one will be the actual gradient descent function which takes the independent variable and the target variable(dependent variable) as input and finds the best fitting line using the gradient descent algorithm.**



Types of Gradient Descent

(a) Batch Gradient Descent (BGD)

Batch gradient descent (BGD) is used to **find the error for each point in the training set and update the model after evaluating all training examples.**

Uses the **entire dataset** to compute the gradient.

Converges **smoothly but is slow for large datasets.**

(b) Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) is a type of gradient descent that runs **one training example per iteration.**

As it requires only **one training example at a time, hence it is easier to store in allocated memory.** However, it shows some computational efficiency losses

Due to frequent updates, it is also treated as a noisy gradient.

It is more efficient for large datasets.

It is relatively fast to compute than batch gradient descent.

Faster but has **high variance**, **leading to noisy updates.**

(c) Mini-Batch Gradient Descent

Mini Batch gradient descent is the **combination of both batch gradient descent and stochastic gradient descent.**

It divides the training datasets into small batch sizes then performs the updates on those batches separately.

Splitting training datasets into smaller batches make a balance to maintain the **computational efficiency of batch gradient descent and speed of stochastic gradient descent.**

Hence, we can achieve a special type of gradient descent with higher computational efficiency and **less noisy gradient descent.**

Example of Gradient Descent

- Assume, Players are playing a game in which - the players are at the top of a mountain and asked to reach the lowest point of the mountain. Additionally, they are blindfolded. So, what approach do you think would make one to reach the lake?

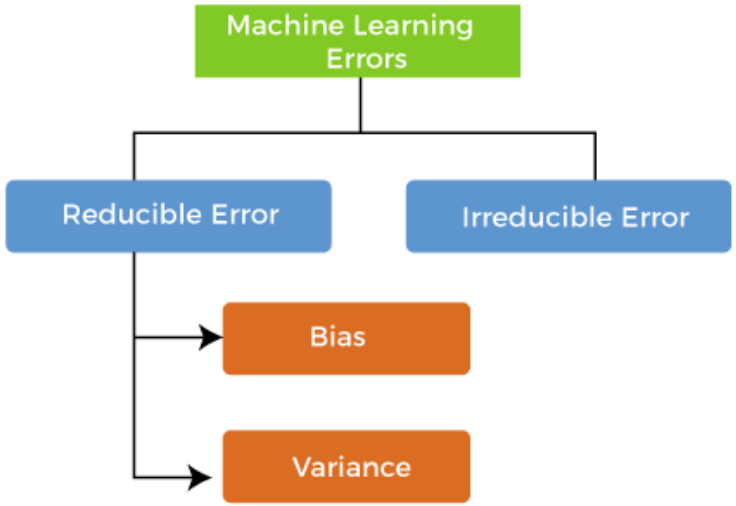
- **The best way is to observe the ground and find where the land descends. From that position, step in the descending direction and iterate this process until we reach the lowest point.**
- Finding the lowest point in a hilly landscape.
- **Gradient descent is an iterative optimization algorithm for finding the local minimum of a function.**
- To find the local minimum of a function using gradient descent, **one must take steps proportional to the negative of the gradient (move away from the gradient) of the function at the current point**.
- **If one take steps proportional to the positive of the gradient (moving towards the gradient), one will approach a local maximum of the function, and the procedure is called Gradient Ascent.**

Bias and Variance in Machine Learning

If the **machine learning model is not accurate, it can make predictions errors, and these prediction errors are usually known as Bias and Variance**. **In machine learning, these errors will always be present** as there is always a slight **difference between the model predictions and actual predictions**. The main aim of ML/data science analysts is to reduce these errors in order to get more accurate results.

Errors in Machine Learning

- An error is a measure of how accurately an algorithm can make predictions for the previously unknown dataset. On the basis of these errors, the machine learning model is selected that can perform best on the particular dataset. There are mainly two types of errors in machine learning, which are:
- **Reducible errors:** These **errors can be reduced to improve the model accuracy**. Such errors can further be classified into bias and Variance.
- **Irreducible errors:** These **errors will always be present in the model - regardless of which algorithm has been used**. The cause of these errors **is unknown variables whose value can't be reduced**.



Bias

While making predictions, a difference occurs between prediction values made by the model and actual values/expected values, and this difference is known as bias errors or Errors due to bias.

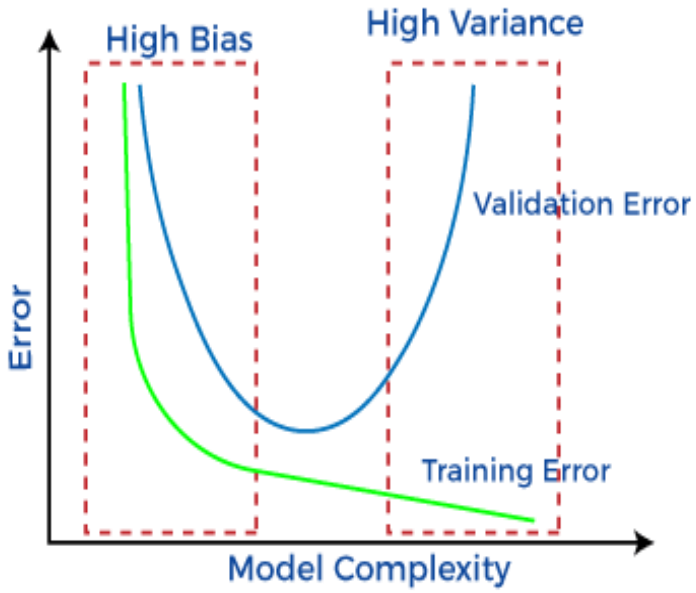
For example, training data for a facial recognition algorithm that over-represents white people may create errors when attempting facial recognition for people of color.

•**Low Bias:** A low bias model will make **fewer assumptions about the form of the target function**.
•**High Bias:** A model with a high bias makes **more assumptions**, and the model becomes unable to capture the important features of our dataset. **A high bias model also cannot perform well on new data.**

Examples of machine learning algorithms with **low bias are Decision Trees, k-Nearest Neighbours and Support Vector Machines**. At the same time, an algorithm with **high bias is Linear Regression, Linear Discriminant Analysis and Logistic Regression.**

Ways to reduce High Bias:

•**Increase the input features** as the model is underfitted.
•**Decrease the regularization term.** (Additional penalty added to the loss function)
•Use more complex models, such as including some polynomial features.



Variance Error

Variance tells that how much a random variable is different from its expected value.

Low variance means there is a **small variation in the prediction of the target function with changes in the training data** set. At the same time, **High variance** shows a **large variation in the prediction of the target function with changes in the training dataset.**

Example: A model that shows **high variance learns a lot and perform well with the training dataset**, and does not generalize well with the unseen dataset. As a result, such a model gives good results with the training dataset but shows **high error rates on the test dataset**.

Since, with high variance, the model learns too much from the dataset, it leads to overfitting of the model. A model with high variance has the below problems:

- A high variance model leads to overfitting.
- Increase model complexities.
- Some examples of machine learning algorithms with **low variance are, Linear Regression, Logistic Regression, and Linear discriminant analysis**. At the same time, algorithms with **high variance are decision tree, Support Vector Machine, and K-nearest neighbours.**

Ways to Reduce High Variance:

- **Reduce the input features** or number of parameters as a model is overfitted.
- Do not use a much complex model.
- **Increase the training data.**
- **Increase the Regularization term.**

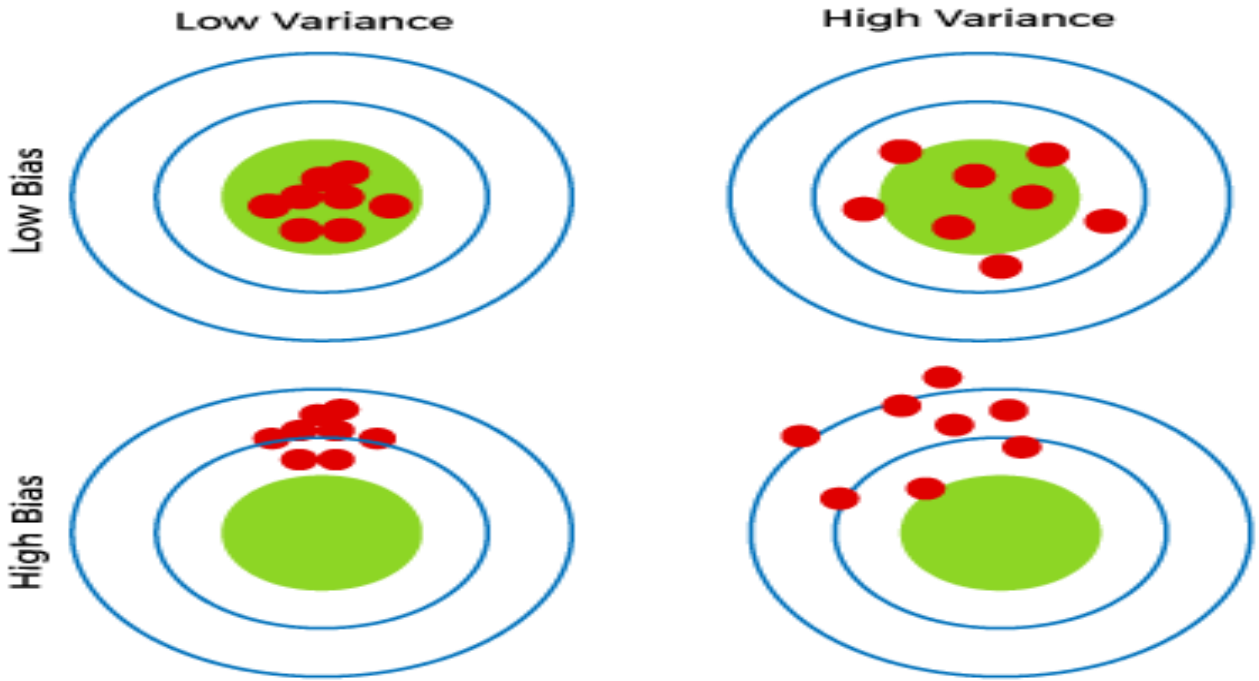
Different Combinations of Bias-Variance

1. Low-Bias, Low-Variance:
The combination of low bias and low variance shows an ideal machine learning model. However, **it is not possible practically.**

2. Low-Bias, High-Variance: With low bias and high variance, **model predictions are inconsistent and accurate on average**. This case occurs when the model learns with a large number of parameters and hence leads to an **overfitting**

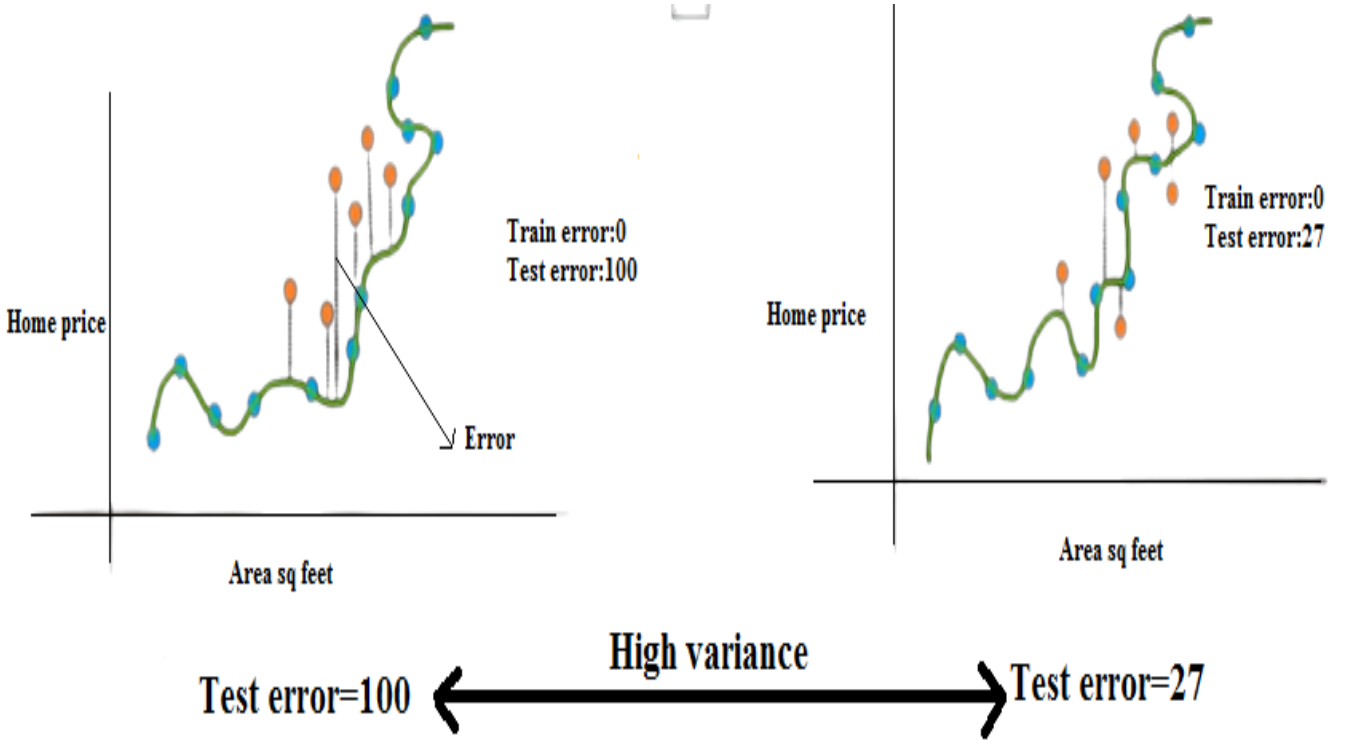
3. High-Bias, Low-Variance: With High bias and low variance, **predictions are consistent but inaccurate on average**. This case occurs when a model does not learn well with the training dataset or uses few numbers of the parameter. It leads to **underfitting** problems in the model.

4.High-Bias,High-Variance:
With high bias and high variance, **predictions are inconsistent and also inaccurate on average.**



Understanding with example

- Suppose one wants to predict house price with respect to the house area. Let's say all these **blue dots** are **training samples**, the **orange dots** are **test samples** as shown in the figure below. We can train a model that fits these blue dots perfectly which means our model is an overfitted model. An overfit model tries to fit exactly to the training samples but not to the test samples that's why training error becomes close to zero and test error is high.



Why there are high errors as compared to other model, even after using the same methodology and same data.

This is because **the test error varies greatly based on the selection of training data points. And this is called high variance because there is high variability in the test error based on what kind of training samples one is selecting**. Now when one is selecting training samples at random - test error varies randomly which is not good and this is the common issue with overfit model.

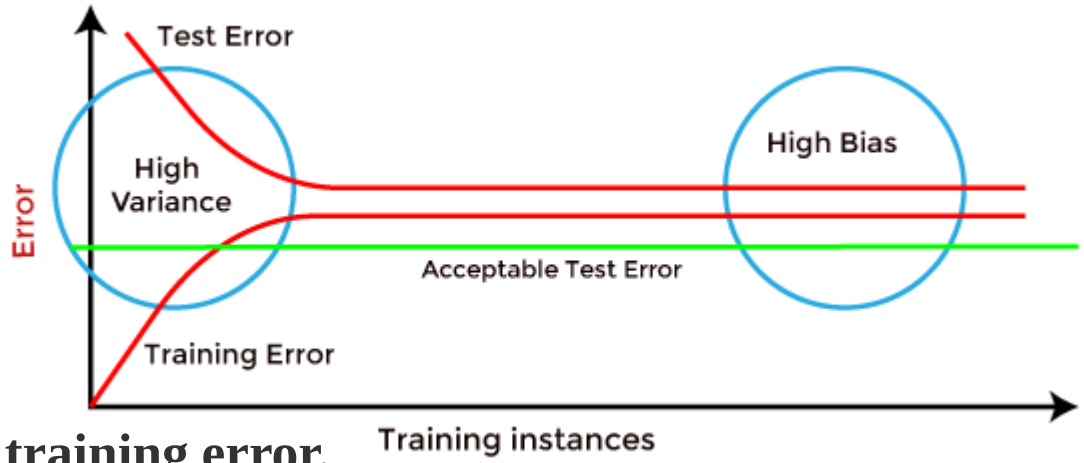
Bias——>Underfitting—->High train and test error

Variance—->Overfitting——>High test error

How to identify High variance or High Bias?

High variance can be identified if the model has:

- **Low training error and high test error.**



High Bias can be identified if the model has:

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

Bias-Variance Trade-Off:

While building the machine learning model, it is really important to take care of bias and variance in order to avoid overfitting and underfitting in the model.

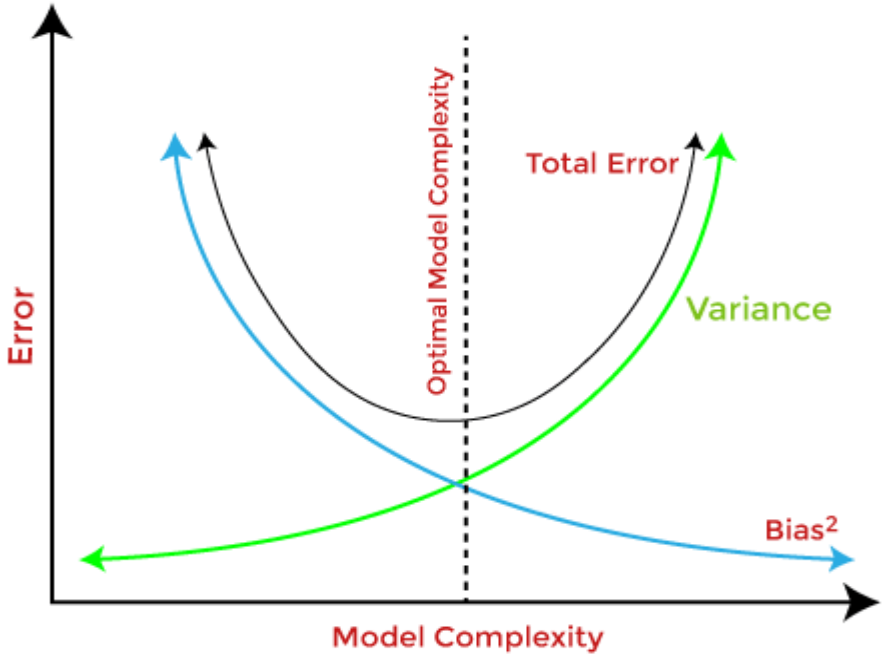
If the model is very simple with **fewer parameters, it may have low variance and high bias. Whereas, if the model has a large number of parameters, it will have high variance and low bias.**

So, it is required to make a balance between bias and variance errors, and this balance between the bias error and variance error is known as the Bias-Variance trade-off.

For an accurate prediction of the model, algorithms **need a low variance and low bias. But this is not possible because bias and variance are related to each other:**

- **If we decrease the variance, it will increase the bias.**
- **If we decrease the bias, it will increase the variance.**

Bias-Variance trade-off is a central issue in supervised learning. Ideally, we need a model that accurately captures the regularities in training data and simultaneously generalizes well with the unseen dataset. Unfortunately, doing this is not possible simultaneously. Because a **high variance algorithm** may perform well with training data, but it **may lead to overfitting to noisy data**. Whereas, **high bias algorithm** generates a much simple model that **may not even capture important regularities in the data.** So, we need to find a sweet spot between bias and variance to make an optimal model.



Overfitting and Underfitting in Machine Learning

Overfitting and Underfitting are the two main problems that occur in machine learning and degrade the performance of the machine learning models.

The main goal of each machine learning model is **to generalize well**. Here **generalization defines the ability of an ML model to provide a suitable output by adapting the given set of unknown input. It means after providing training on the dataset, it can produce reliable and accurate output.** Hence, the underfitting and overfitting are the two terms that need to be checked for the performance of the model and whether the model is generalizing well or not.

Key Terms:

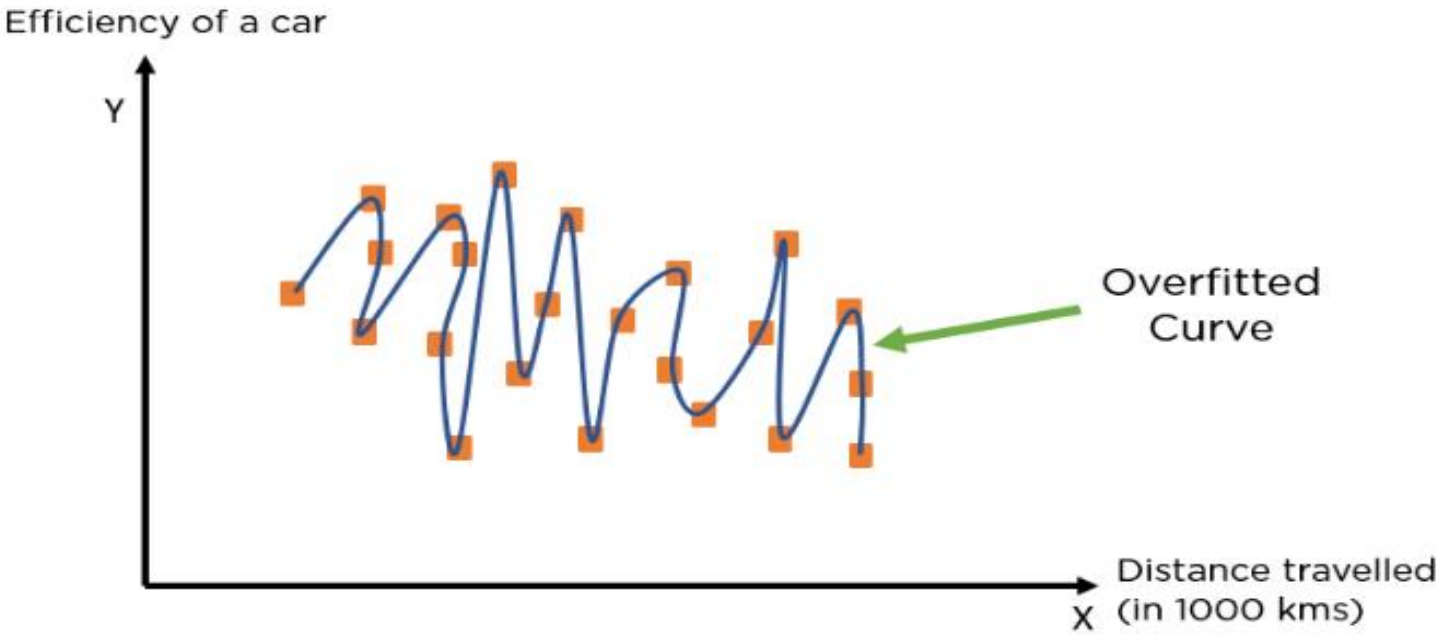
- **Signal:** It refers to the true **underlying pattern of the data** that helps the **machine learning model to learn from the data.**
- **Noise:** Noise is **unnecessary and irrelevant data** that **reduces the performance** of the model.
- **Bias:** Bias is a prediction error that is introduced in the model **due to oversimplifying the machine learning algorithms. Or it is the difference between the predicted values and the actual values.**
- **Variance:** If the machine learning model **performs well with the training dataset, but does not perform well with the test dataset, then variance occurs.**

Overfitting

- **When a model performs very well for training data but has poor performance with test data (new data), it is known as overfitting**. In this case, the machine learning model learns the details and noise in the training data such that it negatively affects the performance of the model on test data.

- Overfitting can happen due to low bias and high variance.

- **Overfitting models are like students who memorize answers instead of understanding the topic. They do well in practice tests (training) but struggle in real exams (testing).**



Reasons for Overfitting:

1.High variance and low bias.

2. The model is **too complex**.

3. The size of the training data.

Techniques to Reduce Overfitting:

1. **Improving the quality of training data** reduces overfitting by focusing on meaningful patterns, mitigate the risk of fitting the noise or irrelevant features.

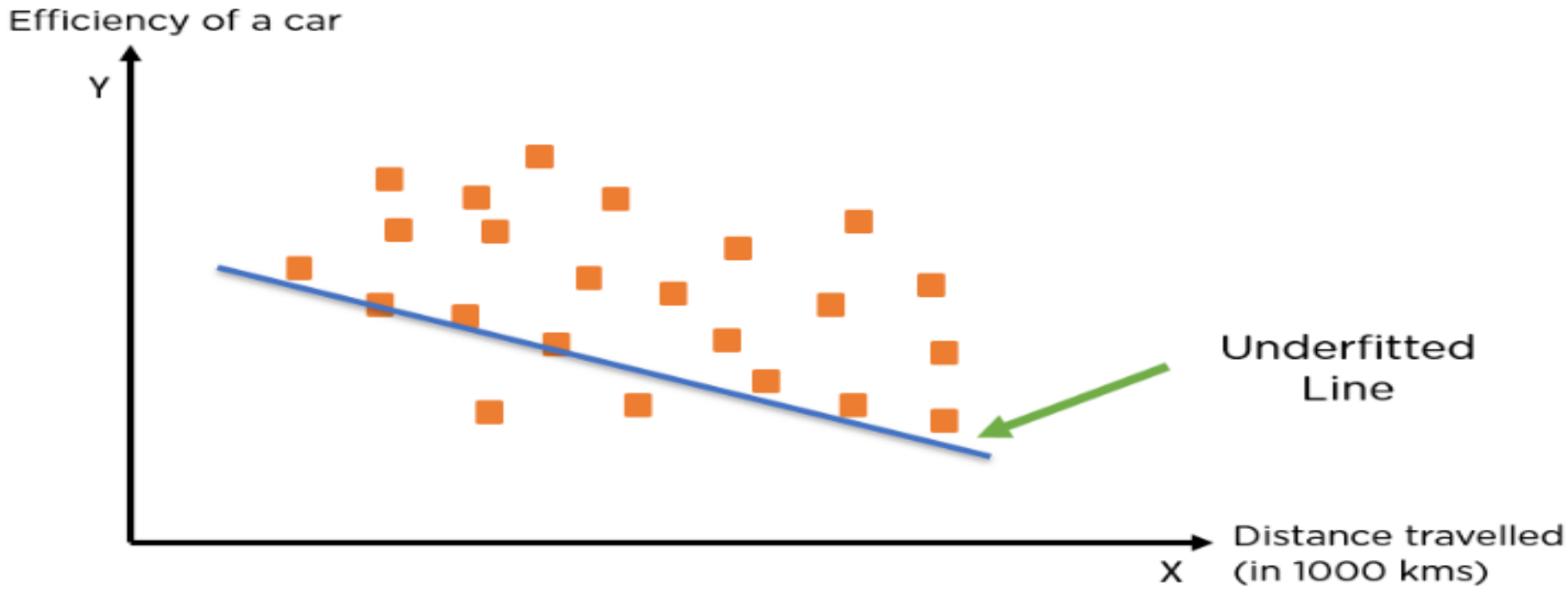
2. **Increase the training data can improve the model's ability to generalize to unseen data** and reduce the likelihood of overfitting.

3. **Reduce model complexity**.

4. **Early stopping** during the training phase (have an eye over the loss over the training period as soon as **loss begins to increase stop training).**

5. Adopting **ensembling** techniques.

Underfitting : When a model has not learned the patterns in the training data well and is unable to generalize well on the new data, it is known as underfitting. An underfit model has poor performance on the training data and will result in unreliable predictions. Underfitting occurs due to **high bias and low variance**. **Underfitting models are like students who don't study enough. They don't do well in practice tests or real exams.**



Reasons for Underfitting:

1.The **model is too simple**, So it may be not capable to represent the complexities in the data.

2.The **input features** which is used to train the model is **not the adequate representations of underlying factors** influencing the target variable.

3.The **size of the training dataset** used is **not enough**.

4.Excessive regularization are used to prevent the overfitting, which constraint the model to capture the data well.

5.Features are not scaled.

Techniques to Reduce Underfitting:

1.Increase model complexity.

2.Increase the number of features, performing feature engineering.

3.Remove noise from the data.

4.Increase the number of epochs or increase the duration of training to get better results.

Good Fit In Machine Learning

To find the good fit model, one needs to look at the performance of a machine learning model over time with the training data. **As the algorithm learns over time, the error for the model on the training data reduces, as well as the error on the test dataset.** If **one trains the model for too long, the model may learn the unnecessary details** and the noise in the training set and hence lead to overfitting. In order to **achieve a good fit, one needs to stop training at a point where the error starts to increase.**