UNIT I:

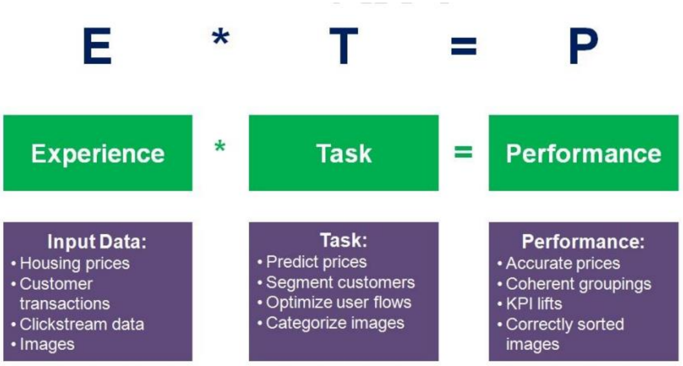
Introduction- Artificial Intelligence, Machine Learning, Deep learning, Types of Machine Learning Systems, Main Challenges of Machine Learning. **Statistical Learning:** Introduction, Supervised and Unsupervised Learning, Training and Test Loss, Tradeoffs in Statistical Learning, Estimating Risk Statistics, Sampling distribution of an estimator, Empirical Risk Minimization

---

INTRODUCTION

1. INTRODUCTION TO MACHINE LEARNING:

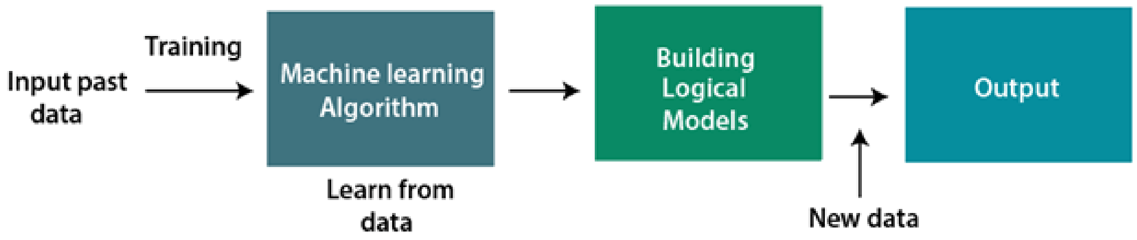
- The term Machine Learning was first coined by Arthur Samuel in the year 1959. Looking back, that year was probably the most significant in terms of technological advancements.
- “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.”



- In simple terms, Machine learning is a subset of Artificial Intelligence (AI) which provides machines the ability to learn automatically & improve from experience without being explicitly programmed to do so. In the sense, it is the practice of getting Machines to solve problems by gaining the ability to think.

MACHINE LEARNING DEFINITIONS:

- **Algorithm:** A Machine Learning algorithm is a set of rules and statistical techniques used to learn patterns from data and draw significant information from it. It is the logic behind a Machine Learning model. An example of a Machine Learning algorithm is the Linear Regression algorithm.
- **Model:** A model is the main component of Machine Learning. A model is trained by using a Machine Learning Algorithm. An algorithm maps all the decisions that a model is supposed to take based on the given input, in order to get the correct output.
- **Predictor Variable:** It is a feature(s) of the data that can be used to predict the output.
- **Response Variable:** It is the feature or the output variable that needs to be predicted by using the predictor variable(s).
- **Training Data:** The Machine Learning model is built using the training data. The training data helps the model to identify key trends and patterns essential to predict the output.
- **Testing Data:** After the model is trained, it must be tested to evaluate how accurately it can predict an outcome. This is done by the testing data set.



MACHINE LEARNING LIFECYCLE:

The lifecycle of a machine learning project involves a series of steps that include:

1. **Study the Problems:** The first step is to study the problem. This step involves understanding the business problem and defining the objectives of the model.
2. **Data Collection:** When the problem is well-defined, we can collect the relevant data required for the model. The data could come from various sources such as databases, APIs, or web scraping.
3. **Data Preparation:** When our problem-related data is collected. then it is a good idea to check the data properly and make it in the desired format so that it can be used by the model to find the hidden patterns. This can be done in the following steps:

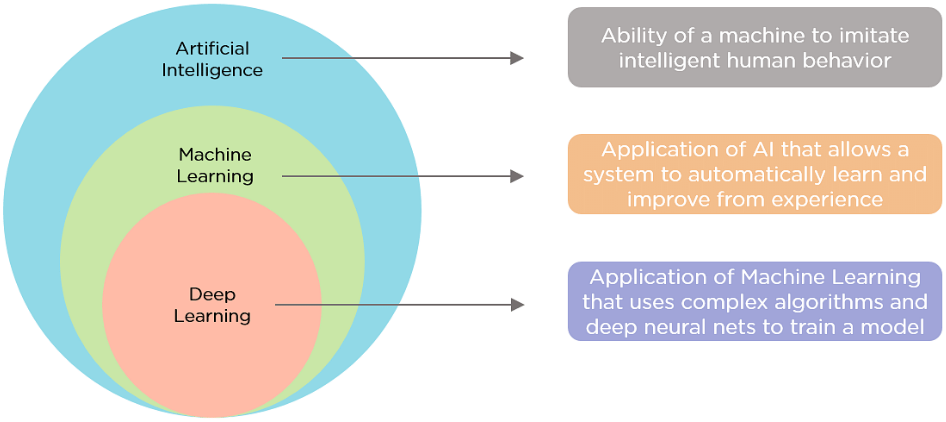
    - Data cleaning
    - Data Transformation
    - Explanatory Data Analysis and Feature Engineering
    - Split the dataset for training and testing.

4. **Model Selection:** The next step is to select the appropriate machine learning algorithm that is suitable for our problem. This step requires knowledge of the strengths and weaknesses of different algorithms. Sometimes we use multiple models and compare their results and select the best model as per our requirements.
5. **Model building and Training:** After selecting the algorithm, we have to build the model.

    - In the case of traditional machine learning building mode is easy it is just a few hyperparameter tunings.
    - In the case of deep learning, we have to define layer-wise architecture along with input and output size, number of nodes in each layer, loss function, gradient descent optimizer, etc.
    - After that model is trained using the preprocessed dataset.

6. **Model Evaluation:** Once the model is trained, it can be evaluated on the test dataset to determine its accuracy and performance using different techniques like classification report, F1 score, precision, recall, ROC Curve, Mean Square error, absolute error, etc.
7. **Model Tuning:** Based on the evaluation results, the model may need to be tuned or optimized to improve its performance. This involves tweaking the hyperparameters of the model.
8. **Deployment:** Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing software system or creating a new system for the model.
9. **Monitoring and Maintenance:** Finally, it is essential to monitor the model's performance in the production environment and perform maintenance tasks as required. This involves monitoring for data drift, retraining the model as needed, and updating the model as new data becomes available.

2. DIFFERENCE BETWEEN ARTIFICIAL INTELLIGENCE VS MACHINE LEARNING VS DEEP LEARNING:


Artificial Intelligence
Ability of a machine to imitate intelligent human behavior
Machine Learning
Application of AI that allows a system to automatically learn and improve from experience
Deep Learning
Application of Machine Learning that uses complex algorithms and deep neural nets to train a model

- **Artificial Intelligence** is basically the mechanism to incorporate human intelligence into machines through a set of rules(algorithm). AI is a combination of two words: “Artificial” meaning something made by humans or non-natural things and “Intelligence” meaning the ability to understand or think accordingly. Another definition could be that **“AI is basically the study of training your machine(computers) to mimic a human brain and its thinking capabilities”**.

  AI focuses on 3 major aspects(skills): learning, reasoning, and self-correction to obtain the maximum efficiency possible.

- **Machine Learning:** Machine Learning is basically the study/process which provides the system(computer) to learn automatically on its own through experiences it had and improve accordingly without being explicitly programmed. **ML is an application or subset of AI.** ML focuses on the development of programs so that it can access data to use it for itself. The entire process makes observations on data to identify the possible patterns being formed and make better future decisions as per the examples provided to them. **The major aim of ML is to allow the systems to learn by themselves through experience without any kind of human intervention or assistance.**
- **Deep Learning:** Deep Learning is basically a sub-part of the broader family of Machine Learning which makes use of **Neural Networks** (similar to the neurons working in our brain) to mimic human brain-like behavior. DL algorithms focus on **information processing patterns** mechanism to possibly identify the patterns just like our human brain does and classifies the information accordingly. DL works on larger sets of data when compared to ML and the **prediction mechanism is self-administered by machines**.

Artificial Intelligence	Machine Learning	Deep Learning
AI stands for Artificial Intelligence, and is basically the study/process which enables machines to mimic human behaviour through particular algorithm.	ML stands for Machine Learning, and is the study that uses statistical methods enabling machines to improve with experience.	DL stands for Deep Learning, and is the study that makes use of Neural Networks(similar to neurons present in human brain) to imitate functionality just like a human brain.
AI is the broader family consisting of ML and DL as it’s	ML is the subset of AI.	DL is the subset of ML.

Artificial Intelligence	Machine Learning	Deep Learning
components.		
AI is a computer algorithm which exhibits intelligence through decision making.	ML is an AI algorithm which allows system to learn from data.	DL is a ML algorithm that uses deep (more than one layer) neural networks to analyze data and provide output accordingly.
The aim is to basically increase chances of success and not accuracy.	The aim is to increase accuracy not caring much about the success ratio.	It attains the highest rank in terms of accuracy when it is trained with large amount of data.
Three broad categories/types Of AI are: Artificial Narrow Intelligence (ANI), Artificial General Intelligence (AGI) and Artificial Super Intelligence (ASI)	Three broad categories/types Of ML are: Supervised Learning, Unsupervised Learning and Reinforcement Learning	DL can be considered as neural networks with a large number of parameters layers lying in one of the four fundamental network architectures: Unsupervised Pre-trained Networks, Convolutional Neural Networks, Recurrent Neural Networks and Recursive Neural Networks
The efficiency of AI is basically the efficiency provided by ML and DL respectively.	Less efficient than DL as it can’t work for longer dimensions or higher amount of data.	More powerful than ML as it can easily work for larger sets of data.
Examples of AI applications include: Google’s AI-Powered Predictions, Ridesharing Apps Like Uber and Lyft, Commercial Flights Use an AI Autopilot, etc.	Examples of ML applications include: Virtual Personal Assistants: Siri, Alexa, Google, etc., Email Spam and Malware Filtering.	Examples of DL applications include: Sentiment based news aggregation, Image analysis and caption generation, etc.
AI refers to the broad field of computer science that focuses on creating intelligent machines that can perform tasks that would normally require human intelligence, such as reasoning, perception, and decision-making.	ML is a subset of AI that focuses on developing algorithms that can learn from data and improve their performance over time without being explicitly programmed.	DL is a subset of ML that focuses on developing deep neural networks that can automatically learn and extract features from data.
AI can be further broken down into various subfields such as robotics, natural language processing, computer vision, expert systems, and more.	ML algorithms can be categorized as supervised, unsupervised, or reinforcement learning. In supervised learning, the algorithm is trained on labeled data, where the desired output is known. In unsupervised learning, the algorithm is trained on unlabeled data, where the desired output is unknown.	DL algorithms are inspired by the structure and function of the human brain, and they are particularly well-suited to tasks such as image and speech recognition.

Artificial Intelligence	Machine Learning	Deep Learning
AI systems can be rule-based, knowledge-based, or data-driven.	In reinforcement learning, the algorithm learns by trial and error, receiving feedback in the form of rewards or punishments.	DL networks consist of multiple layers of interconnected neurons that process data in a hierarchical manner, allowing them to learn increasingly complex representations of the data.

3. TYPES OF MACHINE LEARNING SYSTEMS:

A machine can learn to solve a problem by following any one of the following three approaches. These are the ways in which a machine can learn:

1. Supervised Learning
2. Unsupervised Learning
3. Reinforcement Learning

1. Supervised Learning:

This type of ML involves supervision, where machines are trained on labelled datasets and enabled to predict outputs based on the provided training. The labelled dataset specifies that some input and output parameters are already mapped. Hence, the machine is trained with the input and corresponding output. A device is made to predict the outcome using the test dataset in subsequent phases.

The primary objective of the supervised learning technique is to map the input variable (a) with the output variable (b). Supervised machine learning is further classified into two broad categories:

- **Classification**: These refer to algorithms that address classification problems where the output variable is categorical; for example, yes or no, true or false, male or female, etc. Real-world applications of this category are evident in spam detection and email filtering.

Some known classification algorithms include the Random Forest Algorithm, Decision Tree Algorithm, Logistic Regression Algorithm, and Support Vector Machine Algorithm.

- **Regression**: Regression algorithms handle regression problems where input and output variables have a linear relationship. These are known to predict continuous output variables. Examples include weather prediction, market trend analysis, etc.

Popular regression algorithms include the Simple Linear Regression Algorithm, Multivariate Regression Algorithm, Decision Tree Algorithm, and Lasso Regression.

2. Unsupervised machine learning

Unsupervised learning refers to a learning technique that's devoid of supervision. Here, the machine is trained using an unlabelled dataset and is enabled to predict the output without any supervision. An unsupervised learning algorithm aims to group the unsorted dataset based on the input's similarities, differences, and patterns.

Unsupervised machine learning is further classified into two types:

- **Clustering**: The clustering technique refers to grouping objects into clusters based on parameters such as similarities or differences between objects. For example, grouping customers by the products they purchase.

Some known clustering algorithms include the K-Means Clustering Algorithm, Mean-Shift Algorithm, DBSCAN Algorithm, Principal Component Analysis, and Independent Component Analysis.

- **Association:** Association learning refers to identifying typical relations between the variables of a large dataset. It determines the dependency of various data items and maps associated variables. Typical applications include web usage mining and market data analysis.

Popular algorithms obeying association rules include the Apriori Algorithm, Eclat Algorithm, and FP-Growth Algorithm.

3. Reinforcement learning

Reinforcement learning is a feedback-based process. Here, the AI component automatically takes stock of its surroundings by the hit & trial method, takes action, learns from experiences, and improves performance. The component is rewarded for each good action and penalized for every wrong move. Thus, the reinforcement learning component aims to maximize the rewards by performing good actions.

Reinforcement learning is applied across different fields such as game theory, information theory, and multi-agent systems. Reinforcement learning is further divided into two types of methods or algorithms:

- **Positive reinforcement learning**: This refers to adding a reinforcing stimulus after a specific behavior of the agent, which makes it more likely that the behavior may occur again in the future, e.g., adding a reward after a behavior.
- **Negative reinforcement learning**: Negative reinforcement learning refers to strengthening a specific behavior that avoids a negative outcome.

4. MAIN CHALLENGES OF MACHINE LEARNING:

During the development phase our focus is to select a learning algorithm and train it on some data, the two things that might be a problem are a bad algorithm or bad data, or perhaps both of them. The following are some of the challenges of ML**.**

1. **Not enough training data:**
   Machine Learning is not quite there yet; it takes a lot of data for most Machine Learning algorithms to work properly. Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples.
2. **Poor Quality of data:**
   Obviously, if your training data has lots of errors, outliers, and noise, it will make it impossible for your machine learning model to detect a proper underlying pattern. Hence, it will not perform well. So put in every ounce of effort in cleaning up your training data. No matter how good you are in selecting and hyper tuning the model, this part plays a major role in helping us make an accurate machine learning model. "Most Data Scientists spend a significant part

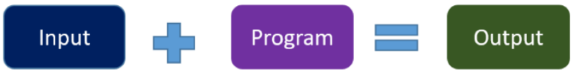
of their time in cleaning data".

3. **Irrelevant Features:**
   "Garbage in, garbage out (GIGO)." if our model is "AWESOME" and we feed it with garbage data, the result will also be garbage(output). Our training data must always contain more relevant and less to none irrelevant features.
4. **Nonrepresentative training data:**
   To make sure that our model generalizes well, we have to make sure that our training data should be representative of the new cases that we want to generalize to. If train our model by using a nonrepresentative training set, it won't be accurate in predictions it will be biased against one class or a group.
5. **Overfitting the Training Data**:
   Overfitting happens when the model is too complex relative to the amount and noisiness of the training data. The possible solutions are: To simplify the model by selecting one with fewer parameters (e.g., a linear model rather than a high-degree polynomial model), by reducing the number of attributes in the training data or by constraining the model
   - To gather more training data
   - To reduce the noise in the training data (e.g., fix data errors and remove outliers)
6. **Underfitting the Training Data**:
   Underfitting is the opposite of overfitting: it occurs when your model is too simple to learn the underlying structure of the data. For example, a linear model of life satisfaction is prone to underfit; reality is just more complex than the model, so its predictions are bound to be inaccurate, even on the training examples.
   The main options to fix this problem are:
   - Selecting a more powerful model, with more parameters
   - Feeding better features to the learning algorithm (feature engineering)
   - Reducing the constraints on the model (e.g., reducing the regularization hyperparameter)

TRADITIONAL VS MACHINE LEARNING

Traditional Programming

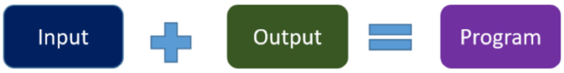
- Traditional programming is a manual process—meaning a person (programmer) creates the program. But without anyone programming the logic, one has to manually formulate or code rules.



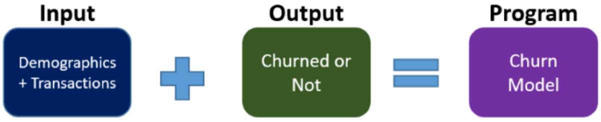
- In machine learning, on the other hand, the algorithm automatically formulates the rules from the data.

Machine Learning Programming

- Unlike traditional programming, machine learning is an automated process.
- It can increase the value of your embedded analytics in many areas, including data prep, natural language interfaces, automatic outlier detection, recommendations, and causality and significance detection.



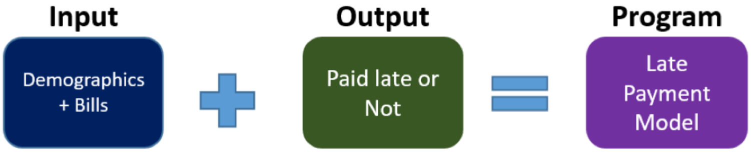
- For example, if you feed in customer demographics and transactions as input data and use historical customer churn rates as your output data, the algorithm will formulate a program that can predict if a customer will churn or not. That program is called a **predictive model**.



This model can be to predict business outcomes in any situation where input and historical output data are known:

1. Identify the business question to ask.
2. Identify the historical input.
3. Identify the historically observed output (i.e., data samples for when the condition is true and for when it's false).

- For instance, if we want to predict who will pay the bills late, identify the input (customer demographics, bills) and the output (pay late or not), and let the machine learning use this data to create your model.



The difference between traditional programming and machine learning lies in their approaches to problem-solving and how they are programmed to handle tasks:

1. **Approach to Problem Solving**:
   - **Traditional Programming**: In traditional programming, a programmer writes explicit rules or instructions for the computer to follow. These rules dictate exactly how the computer should process input data to produce the desired output. It requires a deep understanding of the problem and a clear way to encode the solution in a programming language.
   - **Machine Learning**: In machine learning, instead of writing explicit rules, a programmer trains a model using a large dataset. The model learns patterns and relationships from the data, enabling it to make predictions or decisions without being explicitly programmed for each possibility. This approach is particularly useful for complex problems where defining explicit rules is difficult or impossible.
2. **Data Dependency**:
   - **Traditional Programming:** Relies less on data. The quality of the output depends mainly on the logic defined by the programmer.
   - **Machine Learning:** Heavily reliant on data. The quality and quantity of the training data significantly impact the performance and accuracy of the model.
3. **Flexibility and Adaptability:**
   - **Traditional Programming:** Has limited flexibility. Changes in the problem domain require manual updates to the code.
   - **Machine Learning:** Offers higher adaptability to new scenarios, especially if the model is retrained with updated data.
4. **Problem Complexity:**
   - **Traditional Programming:** Best suited for problems with clear, deterministic logic.

- **Machine Learning:** Better for dealing with complex problems where patterns and relationships are not evident, such as image recognition, natural language processing, or predictive analytics.

5. **Development Process:**
   - **Traditional Programming:** The development process is generally linear and predictable, focusing on implementing and debugging predefined logic.
   - **Machine Learning:** Involves an iterative process where models are trained, evaluated, and fine-tuned. This process can be less predictable and more experimental.
6. **Outcome Predictability:**
   - **Traditional Programming:** The outcome is highly predictable if the inputs and the logic are known.
   - **Machine Learning:** Predictions or decisions made by a machine learning model can sometimes be less interpretable, especially with complex models like deep neural networks.

STATISTICAL LEARNING

1. INTRODUCTION:

- Structuring and visualizing data are important aspects of data science.
- When the goal is to interpret the model and quantify the uncertainty in the data, this analysis is referred to as statistical learning.
- There are two major goals for modeling data:
  1) to accurately predict some future quantity of interest, given some observed data
  2) to discover unusual or interesting patterns in the data.
- To achieve these goals, one must depend on knowledge from three important pillars of the mathematical sciences.
  1) Function approximation
  2) Optimization
  3) Probability and statistics
- **Function approximation:** A mathematical function that is used to represent the relationship between two variables. The data scientists have to understand how best to approximate and represent functions using the least amount of computer processing and memory.
- **Optimization:** Given a class of mathematical models, we wish to find the best possible model in that class. This step usually requires knowledge of optimization algorithms and efficient computer coding or programming.
- **Probability and Statistics:** The knowledge of probability and statistics is needed to fit or train algorithm and generate a model.

2. SUPERVISED AND UNSUPERVISED LEARNING:

- Given an input or feature vector x, one of the main goals of machine learning is to predict an output or response variable y.

  Example:

  1) x: digital signature
  y: Whether the signature is genuine or false

  2) x: weight and smoking habits of an expecting mother.
  y: The birth weight of a baby.
- This prediction is encoded in a mathematical function g, called the prediction function, which takes as an input x and outputs a guess g(x) for y (denoted by $\hat{y}$).
- In regression problems, the response variable y can take any real value.
- In contrast, when y can only lie in a finite set, say $y \in \{0, . . . , c - 1\}$, then predicting y is

conceptually the same as classifying the input x into one of c categories, and so prediction becomes a classification problem.

- We measure the accuracy of a prediction $\hat{y}$ with respect to a given response y by using some loss function Loss(y, $\hat{y}$).
- In a regression setting the usual choice is the squared-error loss (y- $\hat{y}$)2 .
- In the case of classification, the zero–one (also written 0–1) loss function, Loss(y, $\hat{y}$) = 1{y , $\hat{y}$} is often used, which incurs a loss of 1 whenever the predicted class by is not equal to the class y.
- Any mathematical function g will be able to make accurate predictions for all possible pairs (x, y) one may encounter in Nature.
- One reason for this is that, even with the same input x, the output y may be different, depending on chance circumstances or randomness.
- For this reason, we adopt a probabilistic approach and assume that each pair (x, y) is the outcome of a random pair (X, Y) that has some joint probability density f(x, y).
- We then assess the predictive performance via the expected loss, usually called the risk, for g:

$$\ell(g) = \mathbb{E}\,\mathrm{Loss}(Y, g(\boldsymbol{X})).$$

Theorem: Optimal Prediction Function for Squared-Error Loss:

For the squared-error loss Loss(y, $\hat{y}$) = (y −$\hat{y}$)^2, the optimal prediction function g* is equal to the conditional expectation of Y given X = x: g * (x) = E[Y | X = x].

Proof:

Let $g^*(\boldsymbol{x}) = \mathbb{E}[Y \mid \boldsymbol{X} = \boldsymbol{x}]$. For any function g, the squared-error risk satisfies

$$\begin{aligned}\mathbb{E}(Y - g(\boldsymbol{X}))^2 &= \mathbb{E}[(Y - g^*(\boldsymbol{X}) + g^*(\boldsymbol{X}) - g(\boldsymbol{X}))^2]\\ &= \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}[(Y - g^*(\boldsymbol{X}))(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))] + \mathbb{E}(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))^2\\ &\geqslant \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}[(Y - g^*(\boldsymbol{X}))(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))]\\ &= \mathbb{E}(Y - g^*(\boldsymbol{X}))^2 + 2\mathbb{E}\left\{(g^*(\boldsymbol{X}) - g(\boldsymbol{X}))\mathbb{E}[Y - g^*(\boldsymbol{X}) \mid \boldsymbol{X}]\right\}.\end{aligned}$$

By the definition of the conditional expectation, we have E[Y – g* (X) | X] = 0. It follows that E(Y – g(X))^2 ≥ E(Y – g*(X))^2 , showing that g* yields the smallest squared-error risk.

- From the above calculation, conditional on X = x, the (random) response Y can be written as

  $$Y = g^*(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}),$$

  where ε(x) can be viewed as the random deviation of the response from its conditional mean at x.
- Our goal is thus to “learn” the unknown g* using the n examples in the training set T.
- Let us denote by g_T the best (by some criterion) approximation for g* that we can construct from T.
- A particular outcome is denoted as g_T. The function g_T is a learner who learns the unknown functional relationship g* : x → y from the training data T.
- We can imagine a “teacher” who provides n examples of the true relationship between the output Yi and the input Xi for i = 1, . . . , n, and thus “trains” the learner g_T to predict the output of a new input X, for which the correct output Y is not provided by the teacher.
- This is called supervised learning, because one tries to learn the functional relationship between the feature vector x and response y in the presence of a teacher who provides n examples, where x is a vector of explanatory variables variables .
- In contrast, unsupervised learning makes no distinction between response and explanatory variables, and the objective is simply to learn the structure of the unknown distribution of

the data.

3. TRAINING AND TESTING LOSS:

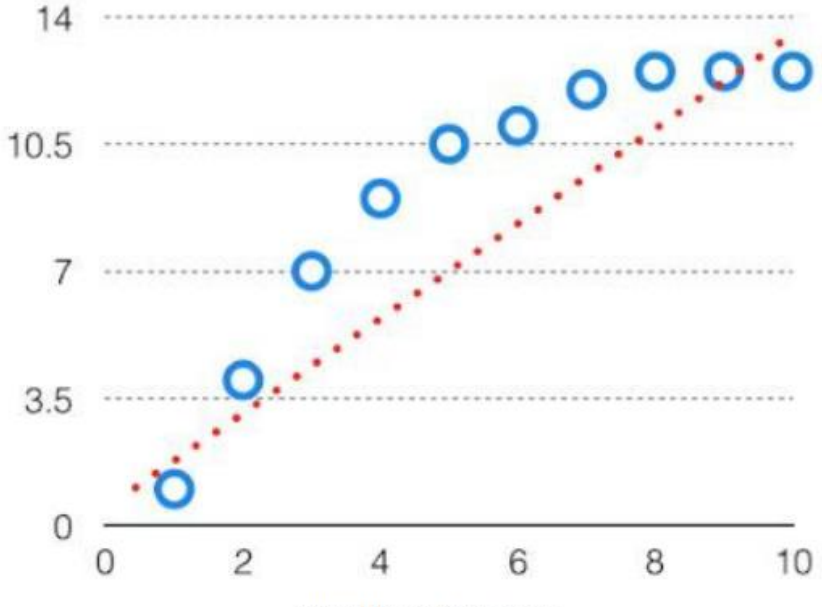
- In machine learning, training and testing loss are used to evaluate the performance of a model.
- The **train loss**, often referred to as the training loss or training error, represents the error or difference between the predicted output and the actual target values during the training phase of a machine learning model.
- The **test loss**, also known as validation loss, is the error or difference between the predicted output and the actual target values on a separate dataset not seen during the training phase.
- The training loss typically decreases during the training process, as the model becomes better at making predictions. However, if the model is overfitting to the trained data, the testing loss may increase even as the training loss decreases.
- This shows that the model is not able to generalize well on the new data.
- The key objective of machine learning is to minimize both the training loss and the testing loss while preventing overfitting.
- This is achieved by using techniques such as regularization, early stopping, and cross validation.

4. TRADEOFFS IN STATISTICAL LEARNING:

- It is important to understand prediction errors (bias and variance) when it comes to accuracy in any machine-learning algorithm.
- There is a tradeoff between a model’s ability to minimize bias and variance.
- A proper understanding of these errors would help to avoid the overfitting and underfitting of a data set while training the algorithm.

➢ **Bias**

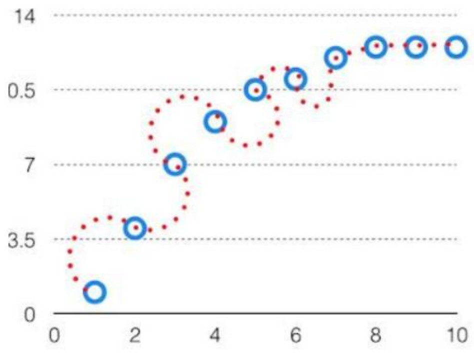
- The bias is known as the difference between the prediction of the values by the Machine Learning model and the correct value.
- Being high in biasing gives a large error in training as well as testing data.
- It recommended that an algorithm should always be low-biased to avoid the problem of underfitting.
- By high bias, the data predicted is in a straight line format, thus not fitting accurately in the data in the data set. Such fitting is known as the **Underfitting of Data**.
- This happens when the hypothesis is too simple or linear in nature. Refer to the graph given below for an example of such a situation.



High Bias in the Model

Variance

- The variability of model prediction for a given data point which tells us the spread of our data is called the variance of the model.
- The model with high variance has a very complex fit to the training data and thus is not able to fit accurately on the data which it hasn't seen before.
- As a result, such models perform very well on training data but have high error rates on test data.
- When a model is high on variance, it is then said to as **Overfitting of Data**.
- Overfitting is fitting the training set accurately via complex curve and high order hypothesis but is not the solution as the error with unseen data is high.
- While training a data model variance should be kept low.

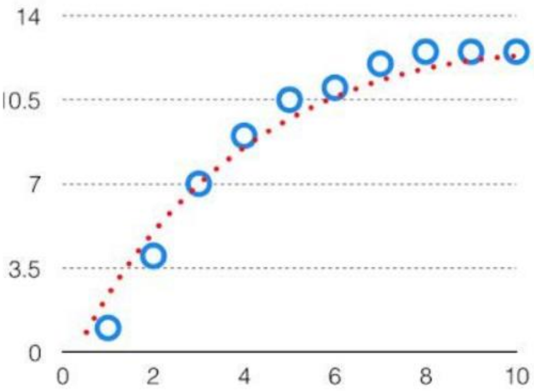


High Variance in the Model

Bias Variance Tradeoff

- If the algorithm is too simple (hypothesis with linear equation) then it may be on high bias and low variance condition and thus is error-prone.
- If algorithms fit too complex (hypothesis with high degree equation) then it may be on high variance and low bias.
- In the latter condition, the new entries will not perform well.

- Well, there is something between both of these conditions, known as a Trade-off or Bias Variance Trade-off.
- For the graph, the perfect tradeoff will be like this.



➢ We try to optimize the value of the total error for the model by using the Bias-Variance Tradeoff.

$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}$$

5. ESTIMATING RISK STATISTICS:

- Risk statistics in machine learning are used to evaluate the performance and generalization of a model.
- Some commonly used risk statistics include the following:

1) **Training error:** This is the error rate of the model on the training data. The training error is used to evaluate how well the model fits the training data.
2) **Testing error:** This is the error rate of the model on the test data. The test error is used to evaluate how well the model generalizes to new unseen data.
3) **Cross-Validation error:** This is the error rate of the model on a validation set that is created by partitioning the data into multiple subsets. It is used to estimate the generalization performance of the model and to prevent overfitting.
4) **Bias:** This is the difference between the prediction of the values by the Machine Learning model and the correct value. Bias measures how well the model captures the true relationship between the input features and the target variable.
5) **Variance:** This is the variability of the model’s prediction for different training sets. Variance measures how sensitive the model is to small changes in the training data.
6) **Mean Square Error:** This is the average of the square differences between the predicted value and the true value. MSE is used to evaluate the overall performance of the model.
7) **Root Mean Squared Error:** This is the square root of the MSE. RMSE is used to measure the average magnitude of the error made by the model.

6. SAMPLING DISTRIBUTION OF AN ESTIMATOR:

What is Sampling Distribution?

Sampling distribution is also known as a **finite-sample distribution**. Sampling distribution is the probability distribution of a statistic based on random samples of a given population. It represents the distribution of frequencies on how spread apart various outcomes will be for a specific population.

Since population is too large to analyze, you can select a smaller group and repeatedly sample or analyze them. The gathered data, or statistic, is used to calculate the likely occurrence, or probability, of an event.

Important Terminologies in Sampling Distribution

Some important terminologies related to sampling distribution are given below:

- **Statistic:** A numerical summary of a sample, such as mean, median, standard deviation, etc.
- **Parameter:** A numerical summary of a population is often estimated using sample statistics.
- **Sample**: A subset of individuals or observations selected from a population.
- **Population:** Entire group of individuals or observations that a study aims to describe or draw conclusions about.
- **Sampling Distribution:** Distribution of a statistic (e.g., mean, standard deviation) across multiple samples taken from the same population.
- **Central Limit Theorem(CLT):** A fundamental theorem in statistics stating that the sampling distribution of the sample mean tends to be approximately normal as the sample size increases, regardless of the shape of the population distribution.
- **Standard Error:** Standard deviation of a sampling distribution, representing the variability of sample statistics around the population parameter.
- **Bias**: Systematic error in estimation or inference, leading to a deviation of the estimated statistic from the true population parameter.
- **Confidence Interval:** A range of values calculated from sample data that is likely to contain the population parameter with a certain level of confidence.
- **Sampling Method:** Technique used to select a sample from a population, such as simple random sampling, stratified sampling, cluster sampling, etc.
- **Inferential Statistics:** Statistical methods and techniques used to draw conclusions or make inferences about a population based on sample data.
- **Hypothesis Testing:** A statistical method for making decisions or drawing conclusions about a population parameter based on sample data and assumptions about the population.

Understanding Sampling Distribution

Sampling distribution of a statistic is the distribution of all possible values taken by the statistic when all possible samples of a fixed size n are taken from the population. Each random sample that is selected may have a different value assigned to the statistics being studied. Sampling distribution of a statistic is the probability distribution of that statistic.

Factors Influencing Sampling Distribution

A sampling distribution's variability can be measured either by calculating the standard deviation(also called the standard error of the mean), or by calculating the population variance. The one to be chosen is depending on the context and interferences you want to draw. They both measure the spread of data points in relation to the mean.

3 main factors influencing the variability of a sampling distribution are:

1. **Number Observed in a Population:** The symbol for this variable is "N." It is the measure of observed activity in a given group of data.
2. **Number Observed in Sample:** The symbol for this variable is "n." It is the measure of observed activity in a random sample of data that is part of the larger grouping.

3. **Method of Choosing Sample:** How you chose the samples can account for variability in some cases.

Types of Distributions

There are 3 main types of sampling distributions are:

- Sampling Distribution of Mean
- Sampling Distribution of Proportion
- T-Distribution

Sampling Distribution of Mean

Mean is the most common type of sampling distribution.

It focuses on calculating the mean or rather the average of every sample group chosen from the population and plotting the data points. The graph shows a normal distribution where the center is the mean of the sampling distribution, which represents the mean of the entire population.

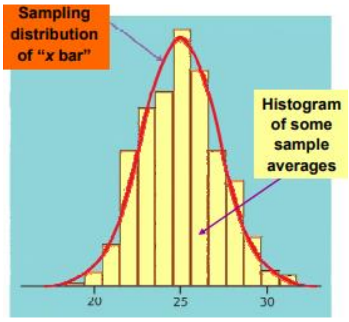
We take many random samples of a given size n from a population with mean μ and standard deviation σ. Some sample means will be above the population mean μ and some will be below, making up the sampling distribution.

For any population with mean μ and standard deviation σ:

- Mean, or center of the sampling distribution of x̄, is equal to the population mean, μ.

$$\mu \bar{x} = \mu$$

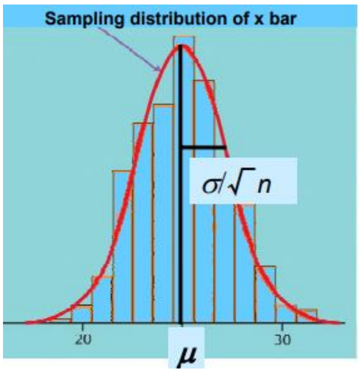
There is no tendency for a sample mean to fall systematically above or below μ, even if the distribution of the raw data is skewed. Thus, the mean of the sampling distribution is an unbiased estimate of the population mean μ.



- Standard deviation of the sampling distribution is σ/√n, where n is the sample size.

$$\sigma x = \sigma/\sqrt{n}$$

- Standard deviation of the sampling distribution measures how much the sample statistic varies from sample to sample. It is smaller than the standard deviation of the population by a factor of √n. Averages are less variable than individual observations.



Sampling Distribution of Proportion

Sampling distribution of proportion focuses on proportions in a population. Here, you select

samples and calculate their corresponding proportions. The means of the sample proportions from each group represent the proportion of the entire population.
Formula for the sampling distribution of a proportion (often denoted as p̂) is:

$$\hat{p} = x/n$$

where:

- **p̂** is Sample Proportion
- **x** is Number of "successes" or occurrences of Event of Interest in Sample
- **n** is Sample Size

This formula calculates the proportion of occurrences of a certain event (e.g., success, positive outcome) within a sample.

T-Distribution

Sampling distribution involves a small population or a population about which you don't know much. It is used to estimate the mean of the population and other statistics such as confidence intervals, statistical differences and linear regression. T-distribution uses a t-score to evaluate data that wouldn't be appropriate for a normal distribution.
Formula for the t-score, denoted as t, is:

$$t = [x - \mu] / [s / \sqrt{(n)}]$$

where:

- **x** is Sample Mean
- **μ** is Population Mean (or an estimate of it)
- **s** is Sample Standard Deviation
- **n** is Sample Size

This formula calculates the difference between the sample mean and the population mean, scaled by the standard error of the sample mean. The t-score helps to assess whether the observed difference between the sample and population means is statistically significant.

7. EMPIRICAL RISK MINIMIZATION:

- The Empirical Risk Minimization (ERM) principle is a learning paradigm which consists in selecting the model with minimal average error over the training set.
- This so-called training error can be seen as an estimate of the risk (due to the law of large numbers), hence the alternative name of empirical risk.
- By minimizing the empirical risk, we hope to obtain a low value of the risk. The larger the training set size is, the closer to the true risk the empirical risk is.
- If we are to apply the ERM principle without more care, we would end up learning by heart, which we know is bad. This issue is more generally, related to the overfitting phenomenon, which can be avoided by restricting the space of possible models when searching for the one with minimal error.
- The most severe and yet common restriction is encountered in the contexts of linear classification or linear regression. Another approach consists in controlling the complexity of the model by regularization.
- While building our machine learning model, we choose a function that reduces the differences between the actual and the predicted output i.e. empirical risk.
- We aim to reduce/minimize the empirical risk as an attempt to minimize the true risk by hoping that the empirical risk is almost the same as the true risk.

Empirical risk minimization depends on four factors:

1. The size of the dataset – the more data we get, the more the empirical risk approaches the true risk.
2. The complexity of the true distribution – if the underlying distribution is too complex, we might need more data to get a good approximation of it.
3. The class of functions we consider – the approximation error will be very high if the size of the function is too large.
4. The loss function – It can cause trouble if the loss function gives very high loss in certain conditions.

The L2 Regularization is an example of empirical risk minimization.

L2 Regularization:

- In order to handle the problem of overfitting, we use the regularization techniques. A regression problem using L2 regularization is also known as **ridge regression**.
- In ridge regression, the predictors that are insignificant are penalized. This method constricts the coefficients to deal with independent variables that are highly correlated.
- Ridge regression adds the "squared magnitude" of coefficient, which is the sum of squares of the weights of all features as the penalty term to the loss function.

$$LossFunction = \frac{1}{N}\sum_{i=1}^{N}(\hat{Y} - Y)^2 + \lambda\sum_{i=1}^{N}\theta_i^2$$

Here λ is the regularization parameter.