M.Tech Program

Advanced Industry Integrated Programs

Jointly offered by University and LTIMindTree

Applied Machine Learning

Knowledge partner

LTIMindtree

Implementation partner

L&T EduTech

Course Objective:

1. To know about Supervised Learning, Support Vector Machines, Unsupervised Learning.

2. Get the knowledge about Feature Engineering, Statistical Data Analysis, Outlier Analysis and Detection

3. Learn about ML Model Development, Model Evaluation Techniques, Model Deployment and Inferences, Model Explainability

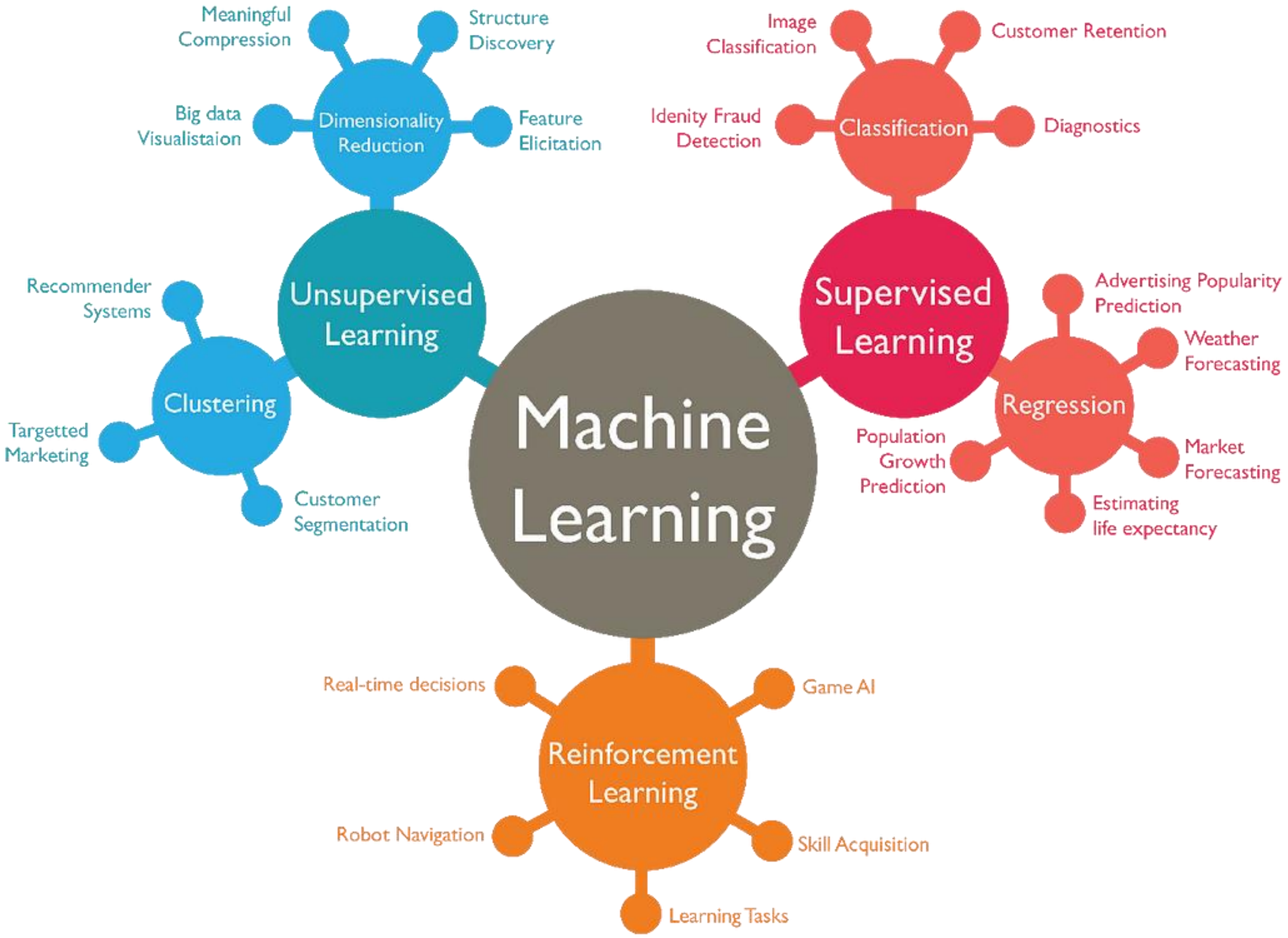
Modules to cover

1. Supervised Learning

2. Advanced Learning Algos

3. Unsupervised Learning and Recommender Systems

ML Algorithms



Supervised Learning

Supervised Learning– *Learning Outcomes..*

- Learners will be able to identify the difference between supervised and unsupervised learning and regression and classification tasks.
- Learners will be able to explain the purpose of a cost function and the process of gradient descent how it is used to train a machine learning model.
- Learners will be able to implement Simple linear regression, Linear regression with multiple features, polynomial regression , logistic regression.
- Learners will be able to explain the concept of regularization and its importance in improving the performance of machine learning models.

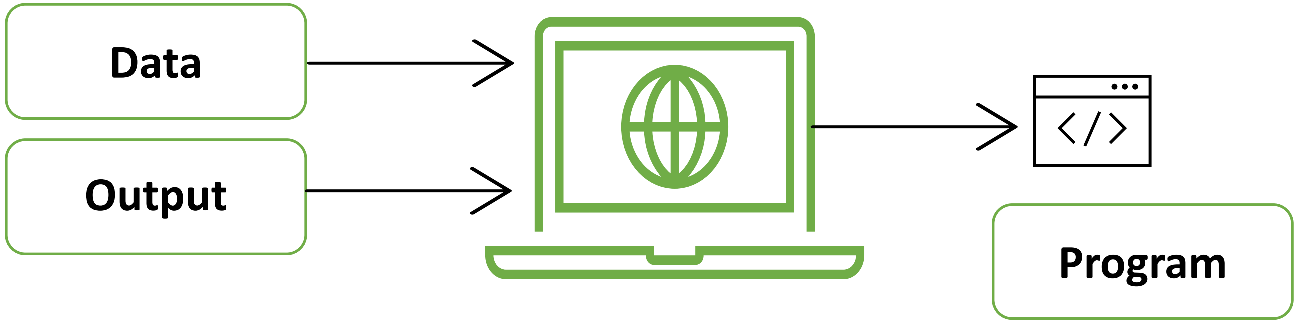
Supervised Learning

Machine Learning Overview

Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so.

Machine learning algorithms use **historical data as input** to **predict new output** values.

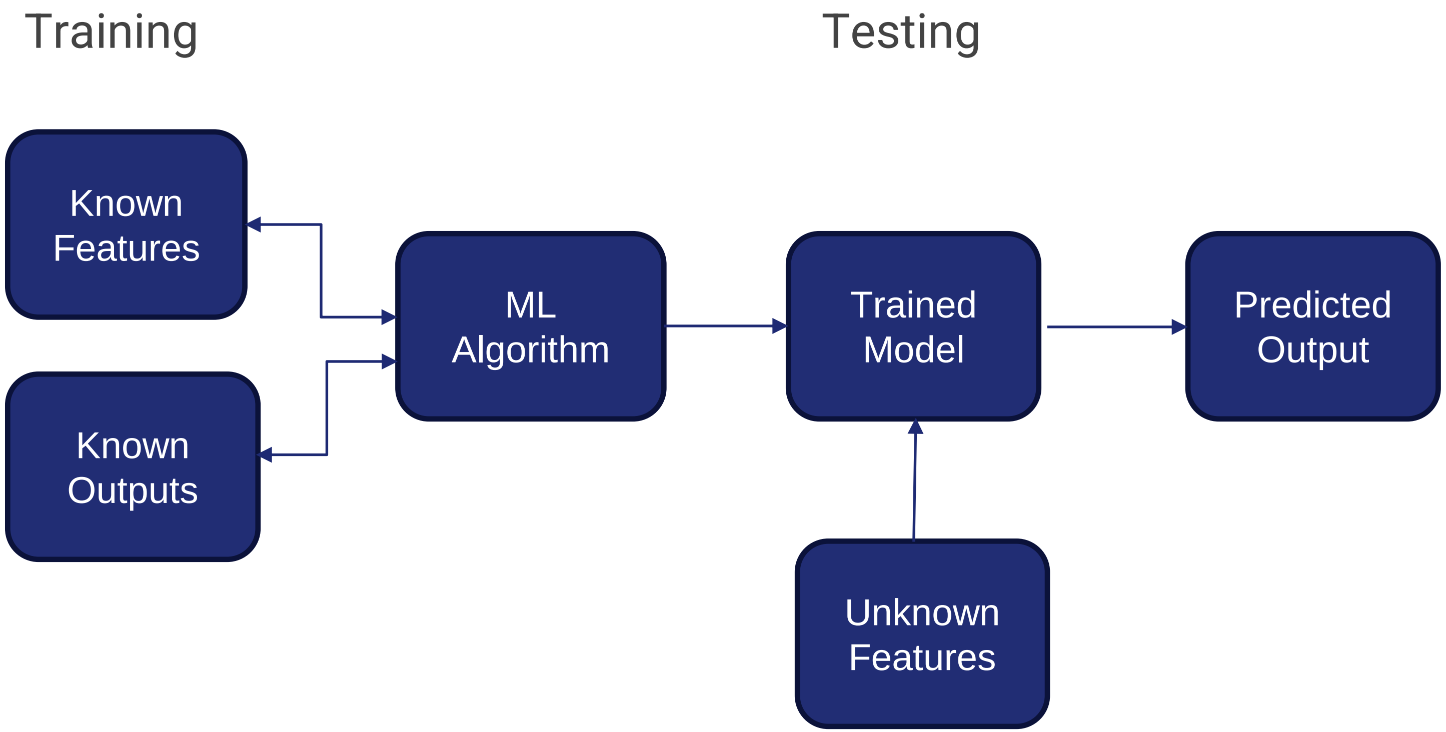
Machine learning helps analyze this data easily and quickly.



Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

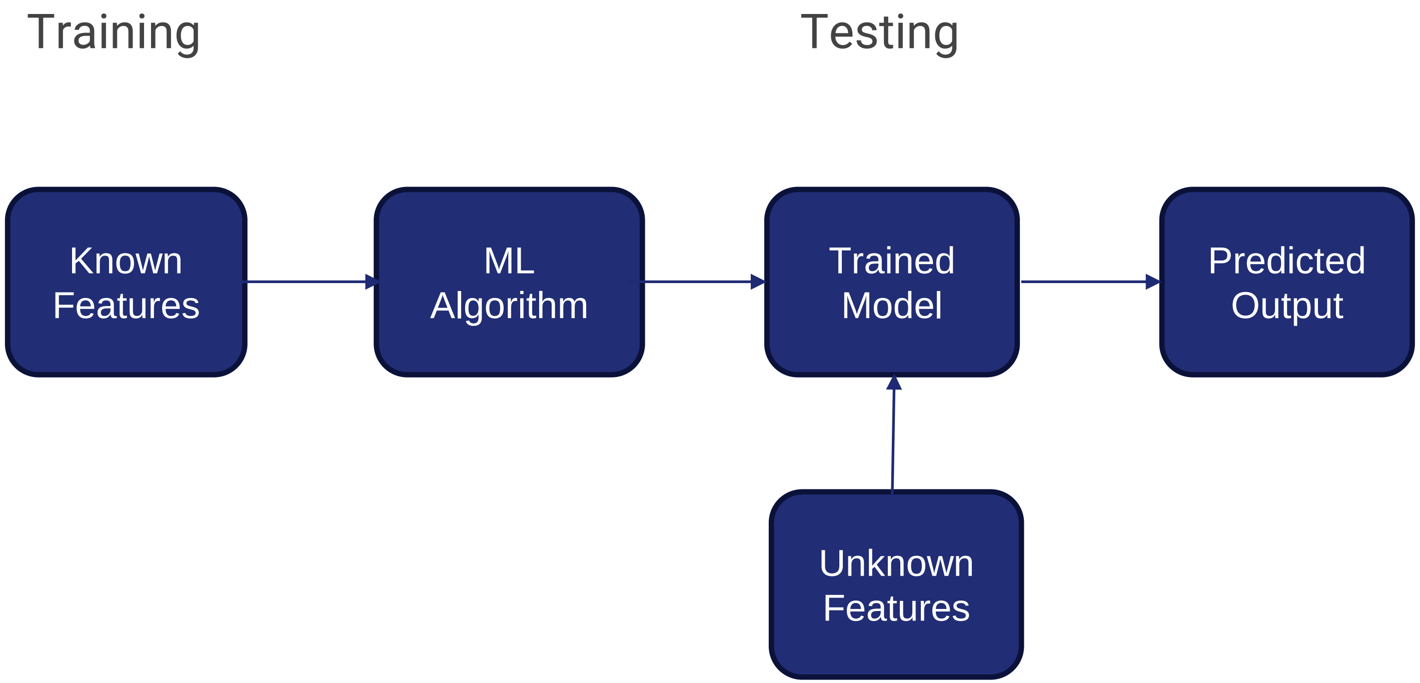
Supervised Learning



Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Unsupervised Learning



Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Regression

- Regression is a fundamental technique in machine learning used for **predicting continuous values** based on one or more independent variables.
- The primary goal of regression is to establish a mathematical model that can accurately predict continuous outputs given a set of input features.

Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Classification

It is the process of categorizing things on the basis of properties. The main goal of a classification problem is to identify the category/class to which a new data will fall under.

Example of Classification :

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe. (Customer Profile)

Supervised Learning

Features	Regression	Classification
Main goal	Predicts continuous values like salary and age.	Predicts discrete values like stock and forecasts.
Input and output variables	Input: Either categorical or continuousOutput: Only continuous	Input: Either categorical or continuousOutput: Only categorial
Types of algorithm	Linear regressionPolynomial regressionLasso regressionRidge regression	Decision treesRandom forestsLogistic regressionNeural networksSupport vector machines
Evaluation metric	R2 scoreMean squared errorMean absolute errorAbsolute percentage error (MAPE)	Receiver operating characteristic curveRecallAccuracyPrecisionF1 score

Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Classification Application

1. Image Recognition
2. Natural Language Processing (NLP)
3. Finance
4. Healthcare
5. Retail
6. Manufacturing

Supervised Learning

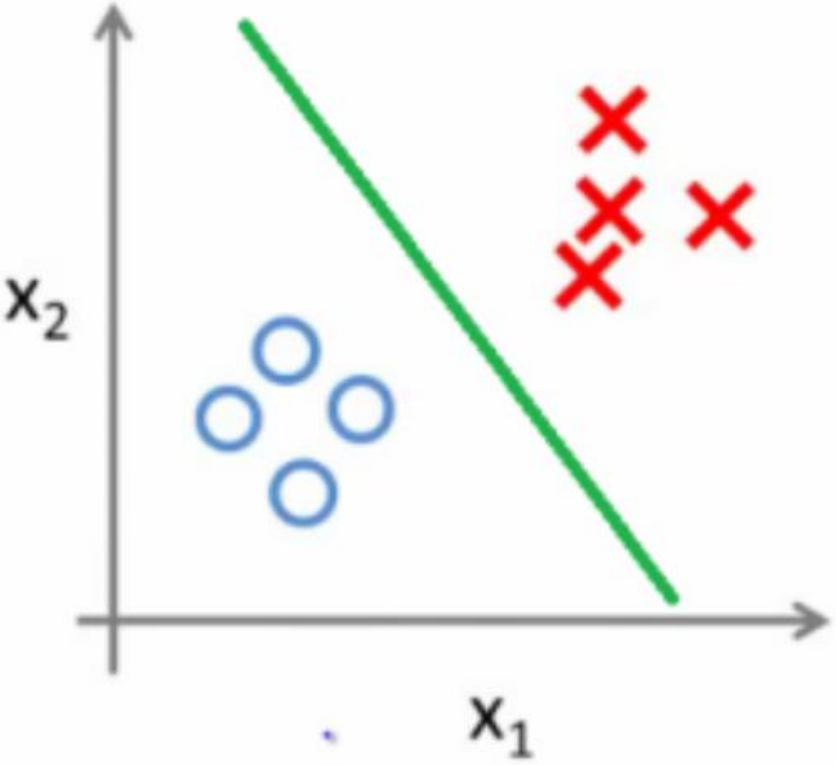
Difference between supervised and unsupervised learning and regression and classification tasks.

Types of classification

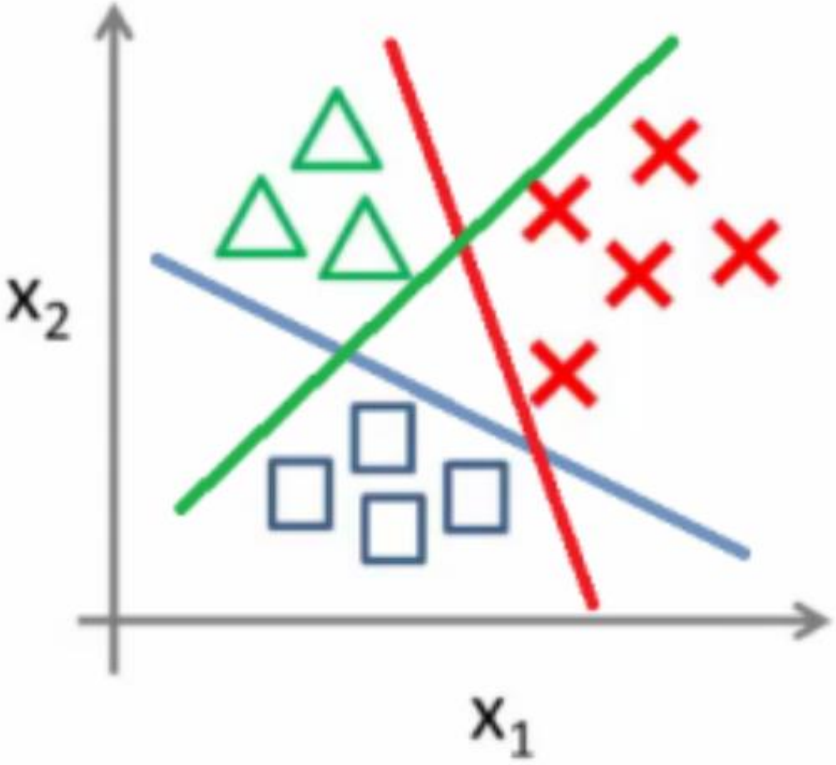
- **Binary Classification** – It is a type of classification with two outcomes, for e.g. – either true or false.
- **Multi-Class Classification** – The classification with more than two classes, in multi-class classification each sample is assigned to one and only one label or target.
- **Multi-label Classification** – This is a type of classification where each sample is assigned to a set of labels or targets.

Supervised Learning

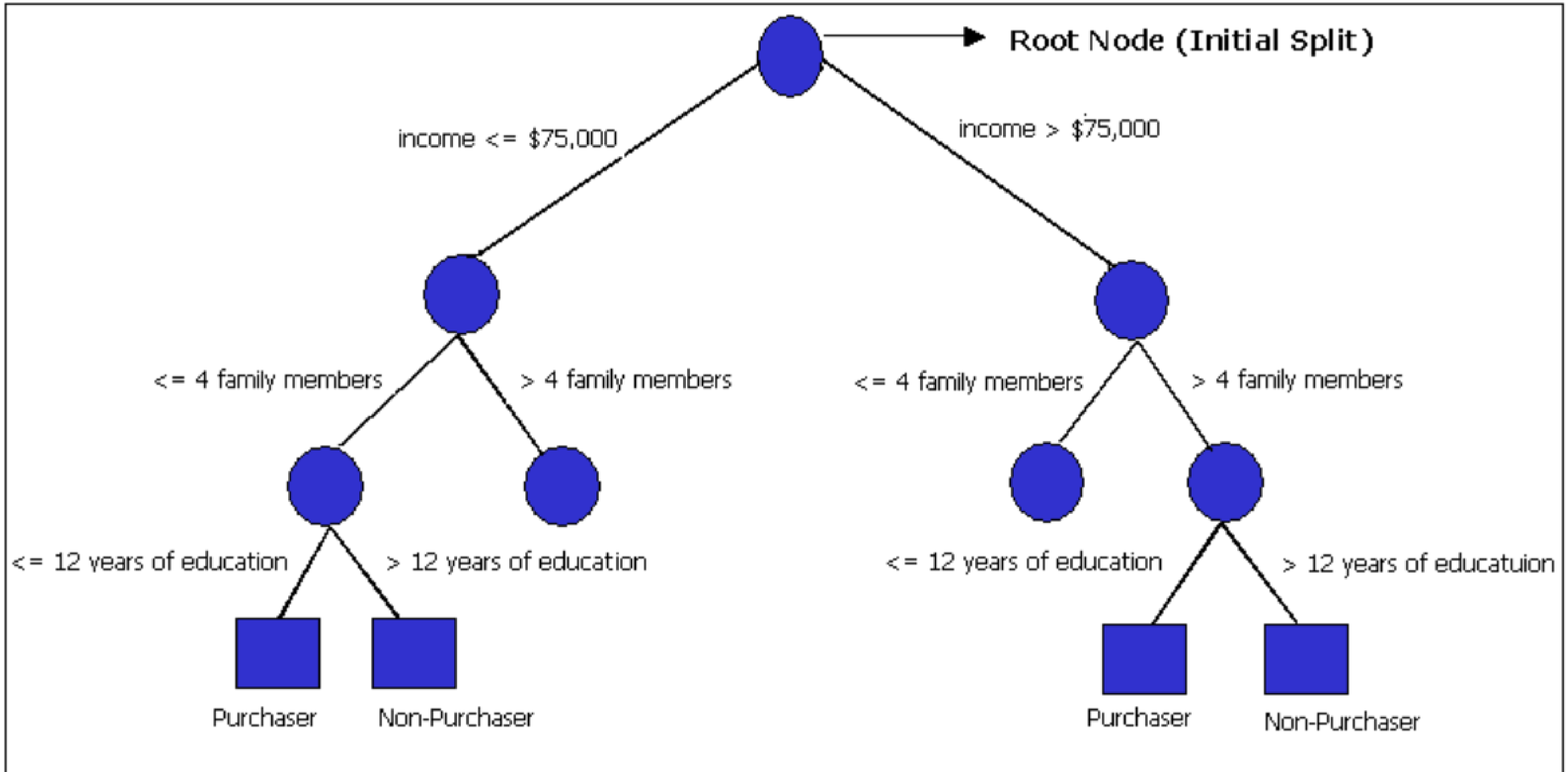
Binary classification:



Multi-class classification:



Supervised Learning – Classification (Example)



Hypothetical Classification Tree

Initially, a Training Set is created where the classification label (i.e., purchaser or non-purchaser) is known (pre-classified) for each record. Next, the algorithm systematically assigns each record to one of two subsets on the some basis (i.e., income > $75,000 or income <= $75,000). The object is to attain an homogeneous set of labels (i.e., purchaser or non-purchaser) in each partition. This partitioning (splitting) is then applied to each of the new partitions. The process continues until no more useful splits can be found. The heart of the algorithm is the rule that determines the initial split rule (displayed in the following figure).

Supervised Learning – Regression (Example)

Income (X)								$E(Y\|X)$
80	55	60	65	70	75			65
100	65	70	74	80	85	88		77
120	79	84	90	94	98			89
140	80	93	95	103	108	113	115	101
160	102	107	110	116	118	125		113
180	110	115	120	130	135	140		125
200	120	136	140	144	145			137
220	135	137	140	152	157	160	162	149
240	137	145	155	165	175	189		161
260	150	152	175	178	180	185	191	173

TABLE 4.1: Weekly consumption of 60 households with respect to income level

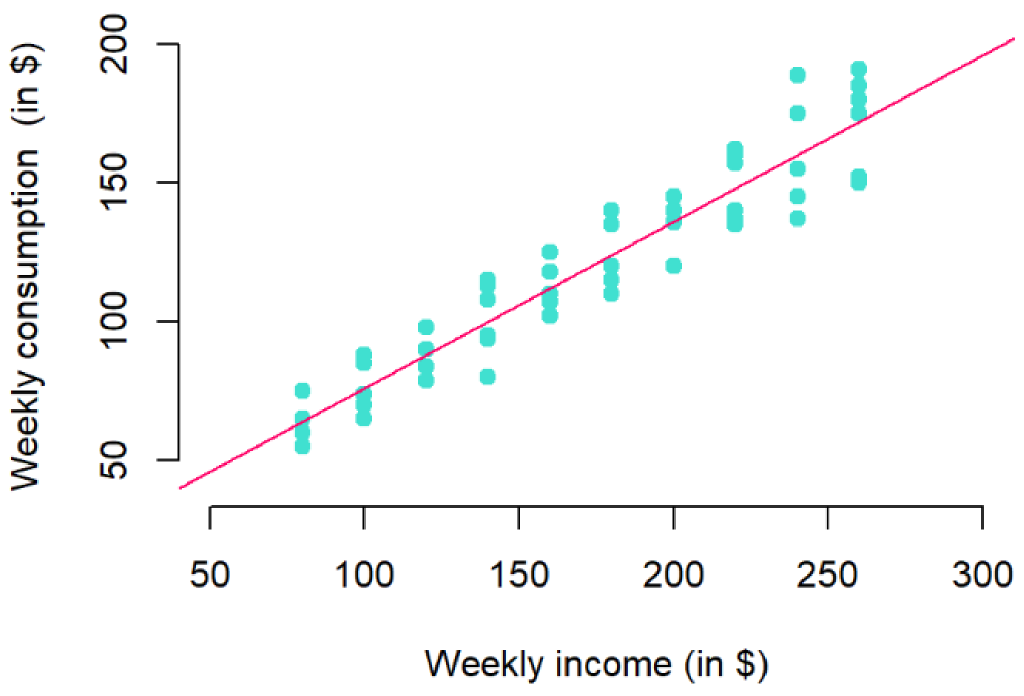


FIGURE 4.1: Population regression line

Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Knowledge Check:

Which of the following statements is TRUE about supervised learning?

(a) It uses labeled data where the desired output is known for each data point.

(b) It focuses on identifying patterns and structures within unlabeled data.

Supervised Learning

Difference between supervised and unsupervised learning and regression and classification tasks.

Knowledge Check:

Which of the following statements is TRUE about supervised learning?

(a) It uses labeled data where the desired output is known for each data point.

(b) It focuses on identifying patterns and structures within unlabeled data.

Supervised Learning

Reference

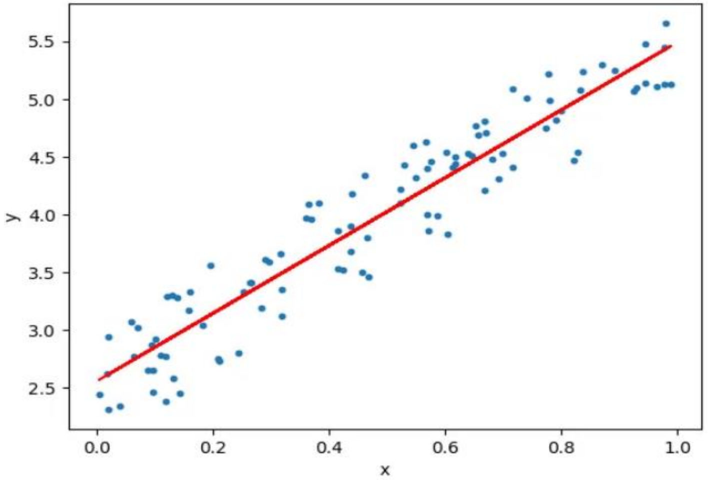
Regression vs. Classification in Machine Learning
Regression vs. Classification in Machine Learning for Beginners | Simplilearn

Supervised Machine Learning: Classification and Regression
Supervised Machine Learning: Classification and Regression | by Nimra Shahzadi | Medium

Supervised Learning

Linear regression model

- Linear regression is a type of **Supervised machine learning** algorithm that computes the linear relationship between the dependent variable and one or more independent features by fitting a linear equation to observed data.
- When there is only one independent feature, it is known as **Simple Linear Regression**, and when there are more than one feature, it is known as **Multiple Linear Regression.**



Supervised Learning

Linear regression model

Simple Linear Regression

This is the simplest form of linear regression, and it involves only one independent variable and one dependent variable. The equation for simple linear regression is:

$$y = \beta o + \beta_1 X$$

Where, **Y** is the dependent variable

β0 is the intercept

β1 is the slope

X is the independent variable

Supervised Learning

Linear regression model

Evaluation metrics

Where:

- n is the number of data points.
- yi represents the actual target value for data point i.
- y^i represents the predicted value for data point i.

Mean Absolute Error

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \widehat{Y}_i\right|$$

Mean Squared Error(MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2$$

Root Mean Squared Error(RMSE)

$$RMSE = sqrt(\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{Y}_i)^2)$$

R Squared Error(R2)

$$R^2 = 1 - \frac{\sum(y_i - \widehat{y}_i)^2}{\sum(y_i - \bar{y})^2}$$

Reference: Evaluation metrics & Model Selection in Linear Regression | by NVS Yashwanth | Towards Data Science

Supervised Learning

Linear regression model

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Where:

- n is the number of data points.
- yi represents the actual target value for data point i.
- $y^\wedge i$ represents the predicted value for data point i.

- **Robustness to Outliers**: Unlike some other metrics, MAE is less sensitive to extreme values (outliers) in the data. This makes it a suitable choice when your dataset contains outliers that might skew other metrics like Mean Squared Error (MSE).

- **Interpretability**: MAE is in the same unit as the original target variable, making it easy to interpret. For example, if your model predicts house prices in dollars, the MAE will also be in dollars, providing a tangible understanding of the error magnitude.

- **Simple and Intuitive**: MAE is straightforward to calculate and understand. Each absolute difference contributes equally to the final score, making it easy to grasp the overall performance of the model.

Supervised Learning

Linear regression model

Each metric treats the differences between observations and expected results in a unique way. The distance between ideal result and predictions have a penalty attached by metric, based on the magnitude and direction in the coordinate system. For example, a different metric such as RMSE more aggressively penalizes predictions whose values are lower than expected than those which are higher. Its usage might lead to the creation of a model which returns inflated estimates.

So how do MAE and MSE treat the differences between points? To check, let's calculate the cost for different weight val

w	-3.0	-2.0	-1.0	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0
MAE	7.5	6.0	4.5	3.0	1.5	0.0	1.5	3.0	4.5	6.0	7.5
MSE	43.75	28.0	15.75	7.0	1.75	0.0	1.75	7.0	15.75	28.0	43.75

- MAE doesn't add any additional weight to the distance between points. The error growth is linear.
- MSE errors grow exponentially with larger values of distance. It's a metric that adds a massive penalty to points that are far away and a minimal penalty for points that are close to the expected result. The error curve has a parabolic shape

Supervised Learning

Cost function

Cost function measures the performance of a machine learning model for given data. Cost function quantifies the error between predicted and expected values and present that error in the form of a single real number. Depending on the problem, cost function can be formed in many different ways. The purpose of cost function is to be either:

- **Minimized:** The returned value is usually called cost, loss or error. The goal is to find the values of model parameters for which cost function return as small a number as possible.

- **Maximized:** In this case, the value it yields is named a reward. The goal is to find values of model parameters for which the returned number is as large as possible.

Supervised Learning

Cost function

- In linear regression, you're trying to fit a straight line to your training data. This line is represented by the equation **f(x) = w * x + b**, where w and b are the model's parameters.
- The cost function, denoted by J(w, b), measures how well a particular choice of w and b fits the training data.
- The goal of linear regression is to find the values of w and b that minimize the cost function J(w, b), making the line fit the data points as closely as possible.

Supervised Learning

Cost function

- The most common cost function used in linear regression is the mean squared error cost function:

$$J(w,b) = \frac{1}{2m}(\sum_{i=1}^{m} \widehat{y_i} - y_i)^2$$

where m is the number of training examples

$\widehat{y_i}$ is the predicted value for the i^{th} data point

y_i is the actual y value for the i^{th} data point in your training set.

Supervised Learning

The purpose of a cost function

1 **Guides the Learning Algorithm**

The cost function acts as a guide for the learning algorithm in linear regression.

2 **Minimizes Squared Errors**

By minimizing the cost function, you essentially minimize the overall squared errors between the predicted values (f(x)) and the actual y values.

3 **Indicates Better Fit**

A lower cost function indicates a better fit between the line and the data points.

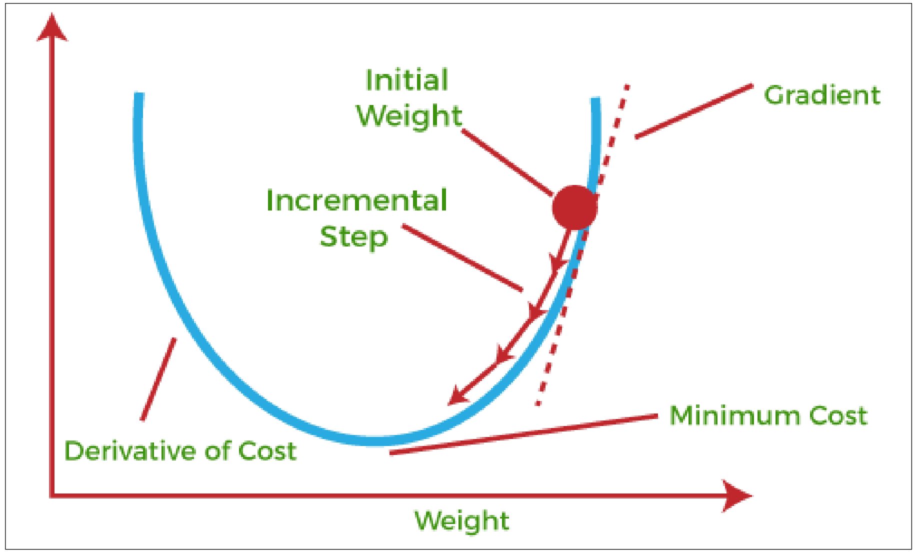
Supervised Learning

How gradient descent is used to train a machine learning model

- 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 using a gradient descent algorithm is to minimize the cost function using iteration. To achieve this goal, it performs two steps iteratively.

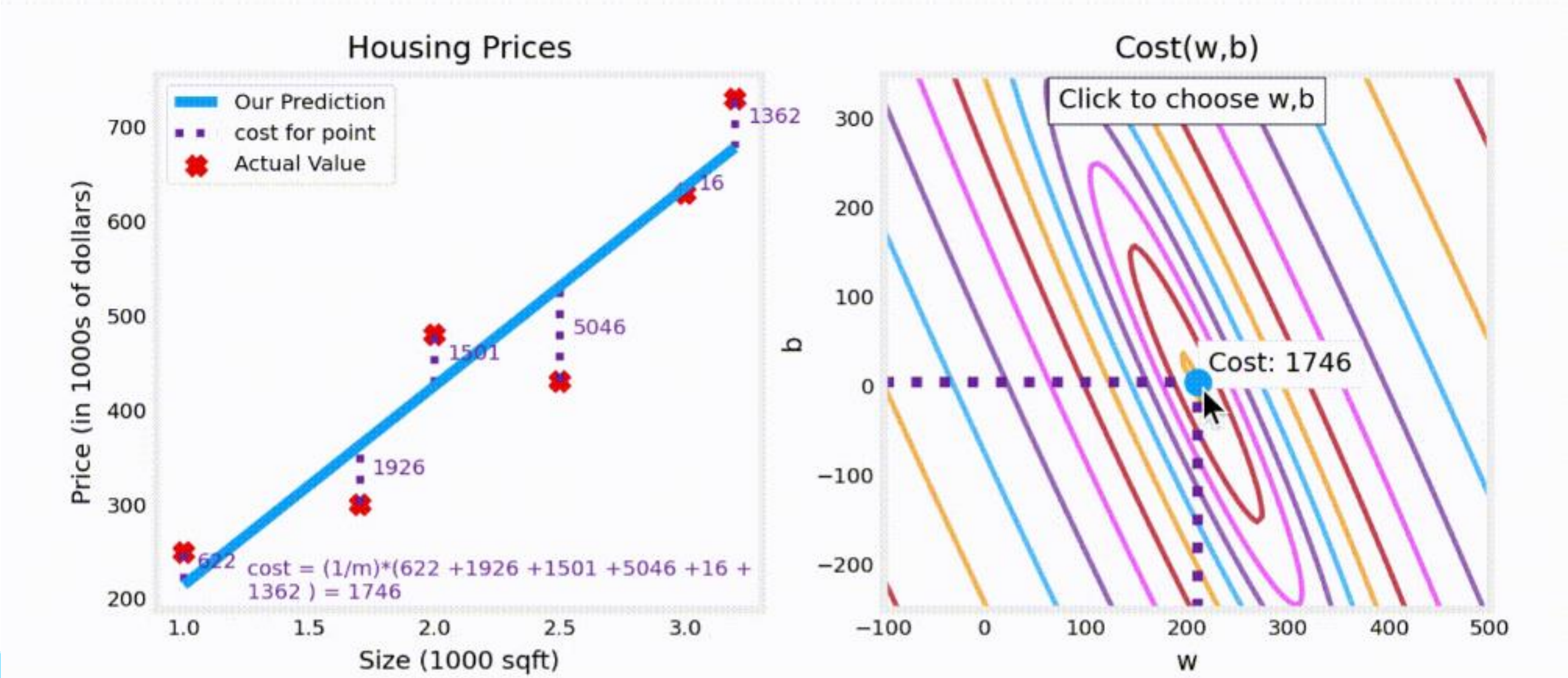
    - Calculates the first-order derivative of the function to compute the gradient or slope of that function.
    - Move away from the direction of the gradient, which means slope increased from the current point by alpha times, where Alpha is defined as Learning Rate. It is a tuning parameter in the optimization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



Supervised Learning

How gradient descent is used to train a machine learning model

Minimizing the cost function involves adjusting the parameters iteratively until convergence, using techniques such as **gradient descent**.



[# 3. Understanding the Cost Function in Linear Regression for Machine Learning Beginners | by Yennhi95zz | Medium]()

Supervised Learning

How gradient descent is used to train a machine learning model

- Gradient descent is an **iterative optimization algorithm** that's used when training a machine learning model. It is used to find the values of a function's parameters that minimize a cost function as far as possible.
- Repeat Until convergence {

$$w = w - \alpha \frac{\partial}{\partial w} J(w, b)$$

$$b = b - \alpha \frac{\partial}{\partial b} J(w, b)$$

}

Supervised Learning

How gradient descent is used to train a machine learning model

starts with an initial guess for the parameters of the model (weights and biases)

calculates the gradient of the cost function with respect to the parameters.

The gradient points in the direction of the steepest ascent of the cost function

Gradient descent takes a small step in the opposite direction (downhill direction), which is the direction of steepest descent.

This process is repeated until the cost function converges to a minimum value

Supervised Learning

Cost function and gradient descent

Knowledge Check:

What is the purpose of gradient descent in linear regression?

(a) To find the equation for the best fit line.

(b) To minimize the cost function and improve the model's fit.

Supervised Learning

Cost function and gradient descent

Knowledge Check:

What is the purpose of gradient descent in linear regression?

(a) To find the equation for the best fit line.

(b) To minimize the cost function and improve the model's fit.

Supervised Learning

Multiple Linear Regression

Problem Description:
We have a dataset of **50 start-up companies**. This dataset contains five main information: **R&D Spend, Administration Spend, Marketing Spend, State, and Profit for a financial year**. Our goal is to create a model that can easily determine which company has a maximum profit, and which is the most affecting factor for the profit of a company.

Since we need to find the Profit, so it is the dependent variable, and the other four variables are independent variables.

R&D Spend	Administration	Marketing Spend	State	Profit
165349	136898	471784	New York	192262
162598	151378	443899	California	191792
153442	101146	407935	Florida	191050
144372	118672	383200	New York	182902
142107	91391.8	366168	Florida	166188
131877	99814.7	362861	New York	156991
134615	147199	127717	California	156123
130298	145530	323877	Florida	155753
120543	148719	311613	New York	152212
123335	108679	304982	California	149760
101913	110594	229161	Florida	146122
100672	91790.6	249745	California	144259

[Multiple Linear Regression in Machine learning - Javatpoint]

Supervised Learning

Multiple Linear Regression

This involves more than one independent variable and one dependent variable. The equation for multiple linear regression is: $f_{\vec{w},b}(\vec{x}) = w_1x_1 + w_2x_2 + ... + w_nx_n + b$

In vector notation,

$$f_{\vec{w},b}(\vec{x}) = \vec{w}.\vec{x} + b$$

$\vec{w} = [w1, w2, w3, ..., wn]$ Parameters of the model

$\vec{x} = [x1, x2, x3, ..., xn]$ Vector

b is a parameter

Supervised Learning

Cost function and gradient descent for multiple linear regression

Cost function for multiple linear regression

The cost function used for multiple linear regression is the **Mean Squared Error (MSE)**. It measures the average squared difference between the predicted values by the model and the actual target values.

$$J(\overrightarrow{w}, b) = \frac{1}{2m}(\sum_{i=1}^{m} \widehat{y_i} - y_i)^2$$

Supervised Learning

Gradient descent for multiple linear regression

For example, if we have 5 features then the equation of hyperplane is represented by :

$$m_1x_1 + m_2x_2 + m_3x_3 + m_4x_4 + m_5x_5 + b = 0$$

In the above line equation, "m" and "b" are the parameter we need to update using Gradient descent to find the best fit line (when I say the best fit line, it is nothing but finding minima in loss function) and "x" is the given input data. The equation to update weight and bias:

$$\text{w\_new} = \text{w\_old} - \text{learning\_rate}*\textstyle\sum_{i=1}^{n} dl/dw$$

$$\text{b\_new} = \text{b\_old} - \text{learning\_rate}*\textstyle\sum_{i=1}^{n} dl/db$$

where "dl/dw" is derivative of loss w.r.t weight, "dl/db" is derivative of loss w.r.t bias, and "n" is the total number of records. Here, the weight is a vector with size=13(we have 13 features).

[Implementing Gradient Descent for multilinear regression from scratch. | by Gunand Mayanglambam | Analytics Vidhya | Medium]

Supervised Learning

Cost function and gradient descent for multiple linear regression

Gradient Descent for multiple linear regression

Repeat {

$$w_1 = w_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_1^{(i)}$$

$$\vdots$$

$$w_n = w_n - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_n^{(i)}$$

$$b = b - \alpha \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

}

Update the values of w_j (for j = 1 to n) and b simultaneously

Supervised Learning

Learning Rate

- In machine learning, the "loss function" measures the error between the predicted and actual output of a machine learning model. The goal is to minimize this loss function by adjusting the model parameters, which improves the model's accuracy. The learning rate controls the size of these parameter updates and influences the speed and stability of the optimisation process.

- A high learning rate can lead to faster convergence but may also cause the optimisation algorithm to overshoot or oscillate around the optimal solution. On the other hand, a low learning rate can result in slow convergence and may get stuck in suboptimal solutions.

- Selecting the right learning rate requires balancing the trade-off between convergence speed and optimisation stability.

What Is Learning Rate in Machine Learning? | Pure Storage

Supervised Learning

Learning Rate

Learning rate (λ) is one such **hyper-parameter** that defines the **adjustment in the weights of our network with respect to the loss gradient descent**. It determines how fast or slow we will move towards the optimal weights.

The Gradient Descent Algorithm estimates the weights of the model in many iterations by minimizing a cost function at every step. Here is the algorithm:

```
Repeat until convergence {

     Wj = Wj - λ θF(Wj)/θWj

}
```

Where:

- **Wj** is the weight
- **θ** is the theta
- **F(Wj)** is the cost function respectively.

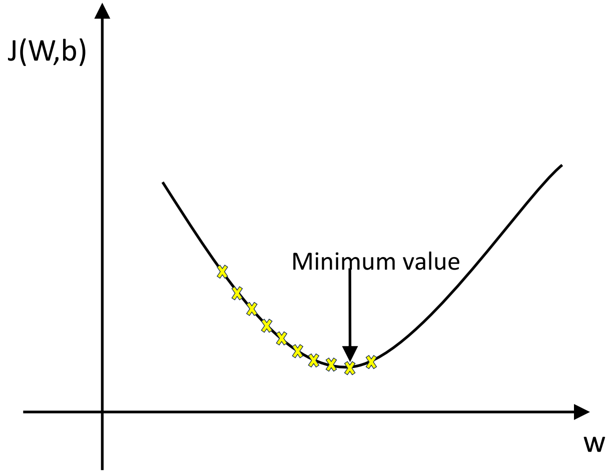
In order for Gradient Descent to work, we must set the learning rate to an appropriate value. This parameter **determines how fast or slow we will move towards the optimal weights**. If the learning rate is very large we will skip the optimal solution. If it is too small we will need too many iterations to converge to the best values. So using a good learning rate is crucial.

In simple language, we can define learning rate as how quickly our network abandons the concepts it has learned up until now for new ones.

Supervised Learning

Methods for improving machine learning models by choosing the learning rate

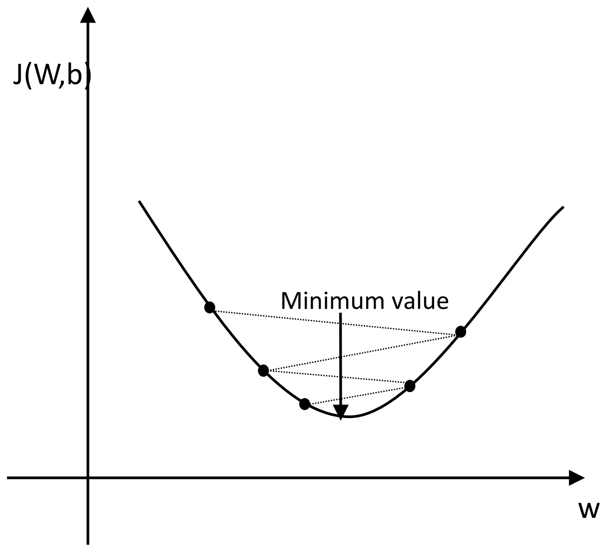
- Consider, $w = w - \alpha \frac{\partial}{\partial w} J(w, b)$
- The choice of the learning rate, alpha will have a huge impact on the efficiency of implementation of gradient descent
- When learning rate is too small , cost J is decreased slowly.



Supervised Learning

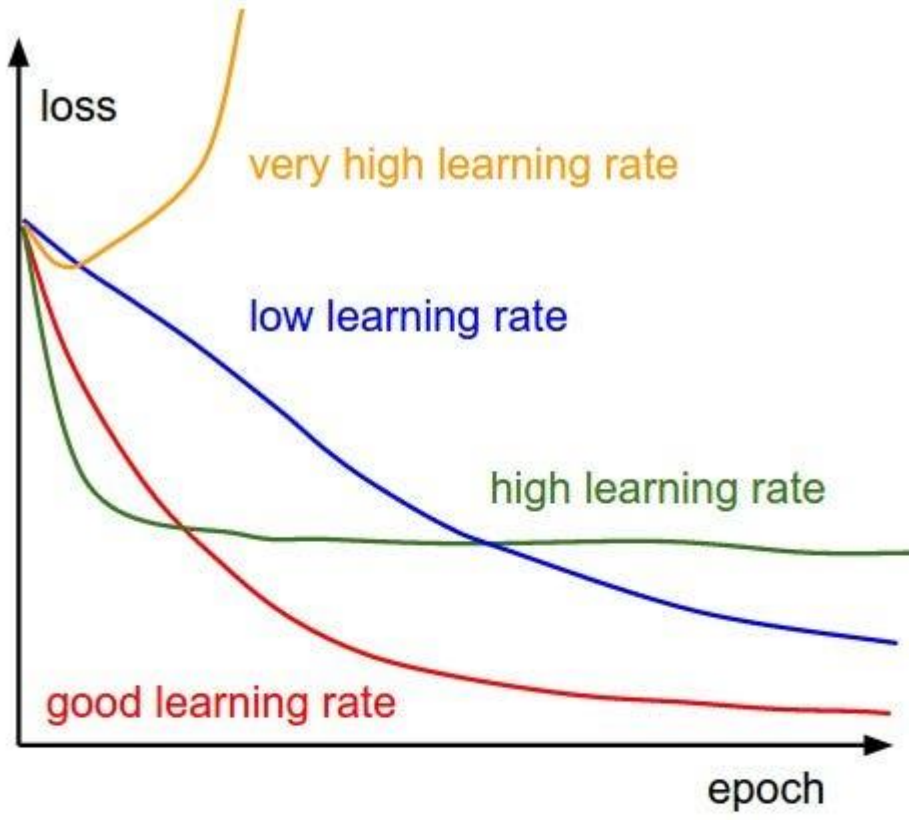
Methods for improving machine learning models by choosing the learning rate

- When the learning rate is too large , great intersect may fail to converge and may even diverge
- Even with a fixed learning rate, gradient descent can automatically take smaller steps as it approaches a minimum because the derivative term gets smaller.



Supervised Learning

Methods for improving machine learning models by choosing the learning rate



The learning rate is the most important hyper-parameter for tuning neural networks. A good learning rate could be the difference between a model that doesn't learn anything and a model that presents state-of-the-art results.

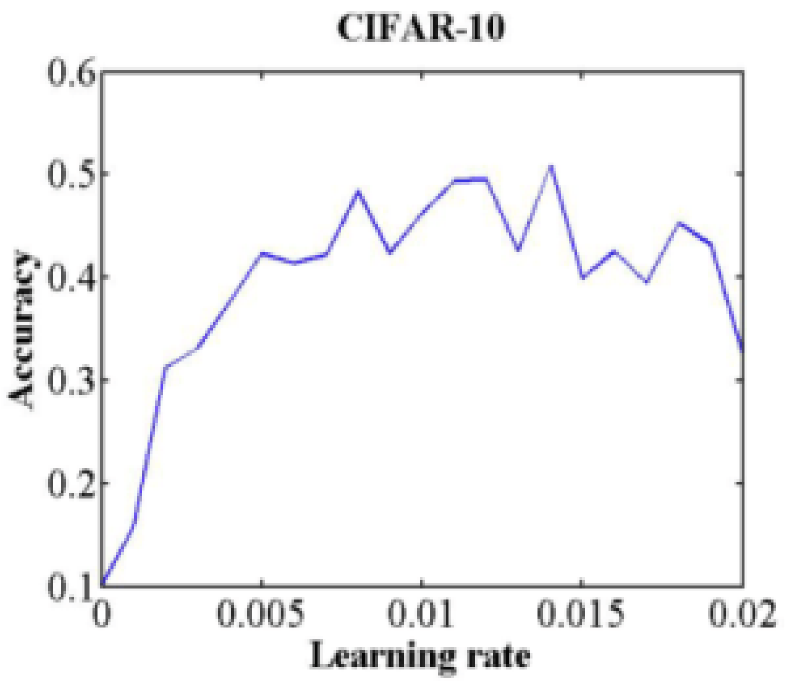
The below diagram demonstrates the different scenarios one can fall into when configuring the learning rate.

The obvious way to find a desirable or optimal learning rate is through trial and error. To do this efficiently, there are a few ways that we should adhere to.

Supervised Learning

Plotting the Learning curve

- Choose a minimum and maximum learning rate to search through (e.g. 1e-7 and 0.1).
- Train the model for several epochs using SGD while linearly increasing the learning rate from the minimum to maximum learning rate.
- At each iteration, record the accuracy (or loss).
- Plot the test accuracy and see where the loss/accuracy starts to improve, and when it starts to get worse/plateau/to become ragged.
- The latter learning rate is the maximum learning rate that converges and is a good value for your initial learning rate.
- The former learning rate, or 1/3–1/4 of the maximum learning rates is a good minimum learning rate that you can decrease if you are using learning rate decay.
- If the test accuracy curve looks like the above diagram, a good learning rate to begin from would be 0.006, where the loss starts to become jagged.



Supervised Learning

Plotting the Learning curve

Learning Curve

- A learning curve is a graphical representation of how the performance of a machine learning model changes as the size of the training set increases.
- It typically involves plotting two error scores:
  - ↳ the **training error**, which reflects how well the model fits the training data
  - ↳ the **validation error**, which reflects how well the model generalizes to unseen data.

Supervised Learning

Plotting the Learning curve

Use of Learning curves

Identifying Underfitting

When the training error is low but the validation error remains high, it suggests the model is underfitting the data. This means it's memorizing the training examples too closely and failing to capture the underlying patterns that generalize to unseen data.

Identifying Overfitting

Conversely, if the training error is very low and close to the validation error, it might indicate overfitting. The model is fitting the training noise and not the actual relationships within the data.

Determining the Right Training Set Size

By observing how the training training and validation errors change with increasing training training set size, you can get a a sense of when the model has has achieved a good balance between fitting the data and generalizing well.

Supervised Learning

Plotting a Learning Curve

Step-by-Step Process

1. Split your data
2. Train models with increasing data sizes
3. Iteratively increase training set size
4. Plot the errors
5. Analyze the curves

Supervised Learning

Learning curve

Knowledge Check:

In machine learning, what role does a learning curve play?

(a) It depicts how long a model takes to train.

(b) It shows how the model's accuracy changes throughout the training process.

(c) It visualizes the decision boundaries learned by the model in the space of its features.

(d) It ranks features based on their importance to the model's predictions.

Supervised Learning

Learning curve

Knowledge Check:

In machine learning, what role does a learning curve play?

(a) It depicts how long a model takes to train.

(b) It shows how the model's accuracy changes throughout the training process.

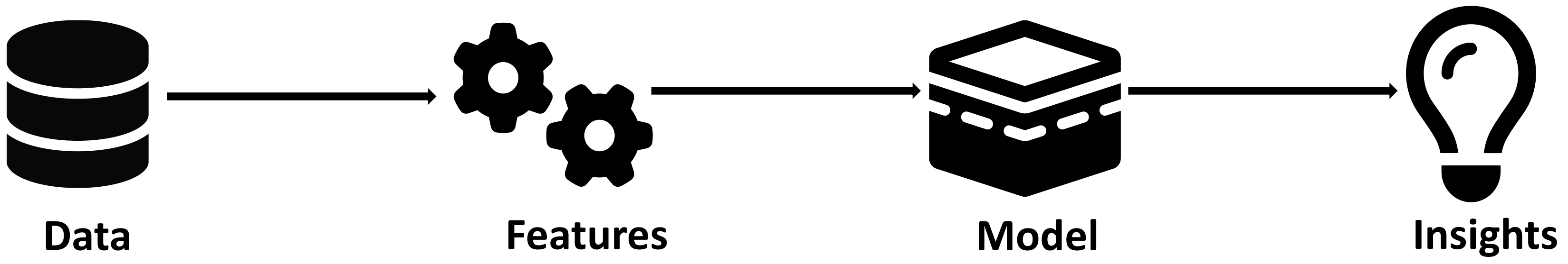
(c) It visualizes the decision boundaries learned by the model in the space of its features.

(d) It ranks features based on their importance to the model's predictions.

Supervised Learning

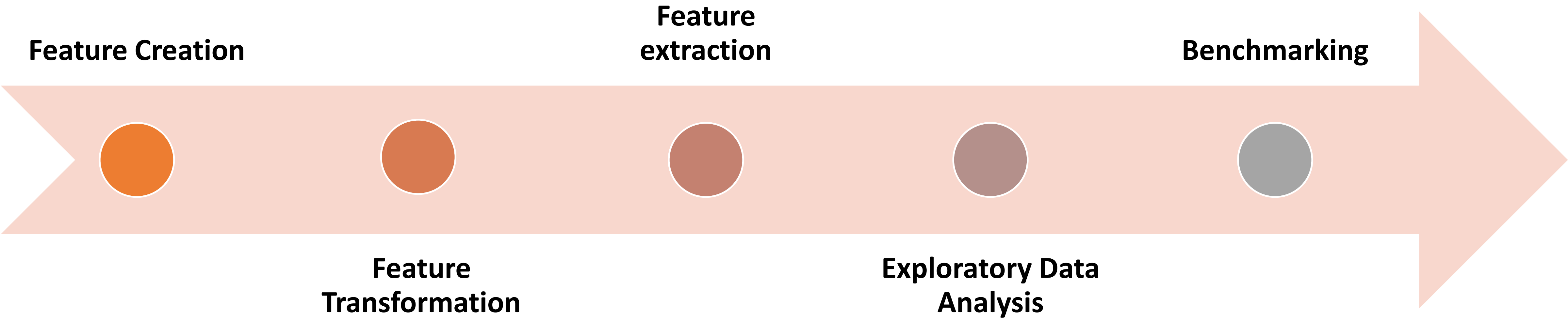
Feature Engineering

- The process of selecting, modifying, and converting raw data into features that may be applied to supervised learning is known as feature engineering.
- Feature engineering leverages data to create new variables that aren't in the training set.



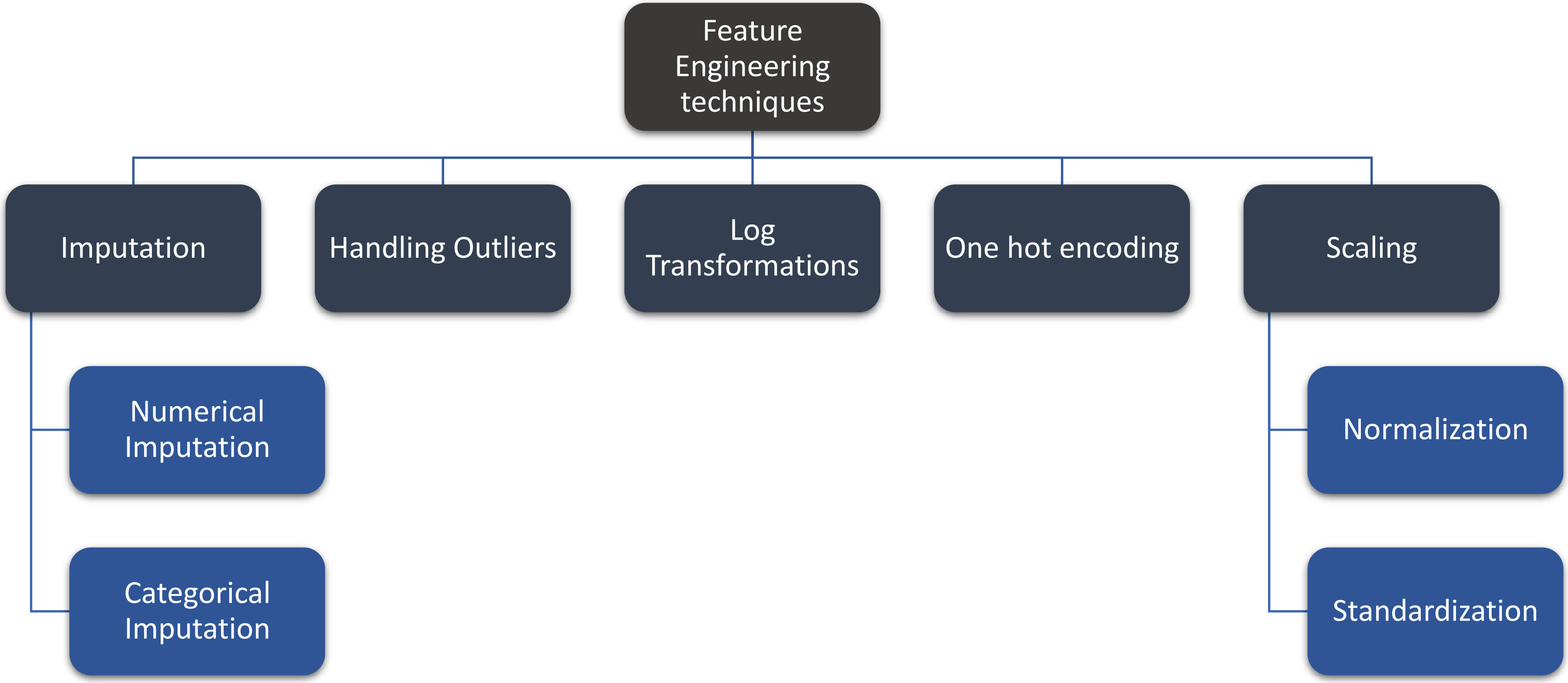
Supervised Learning

Feature Engineering Process



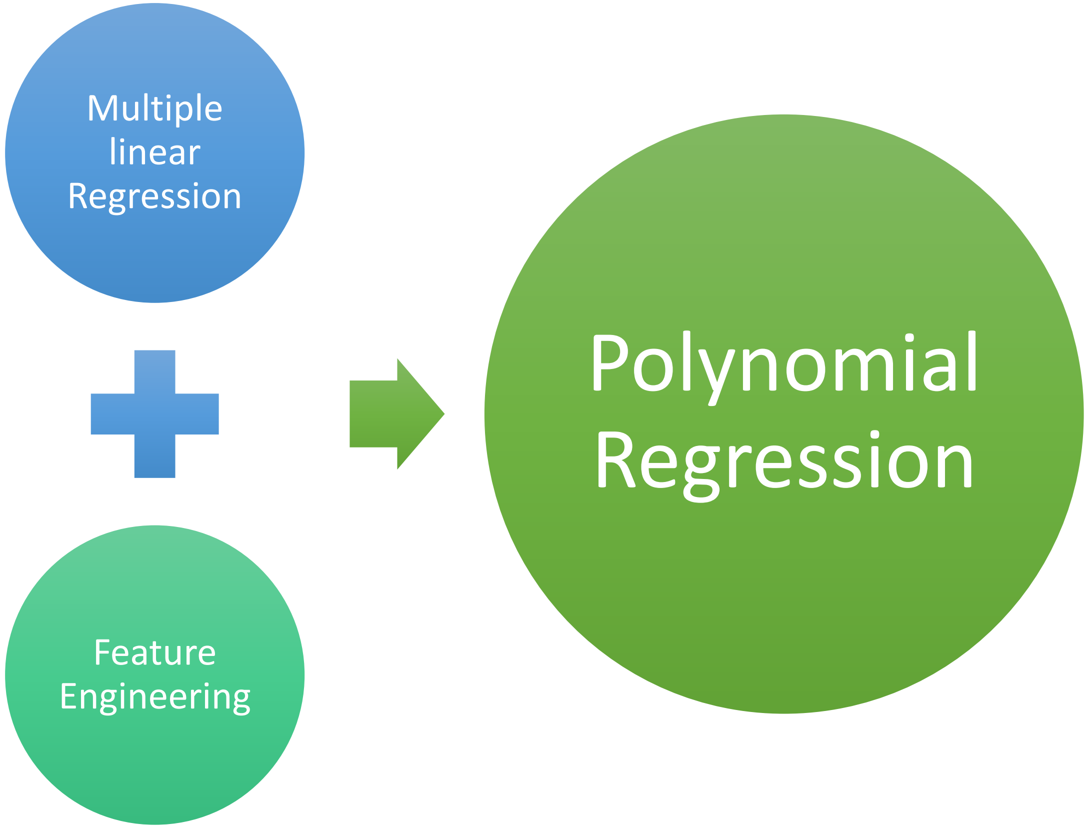
Supervised Learning

Feature Engineering Techniques



Supervised Learning

Polynomial Regression

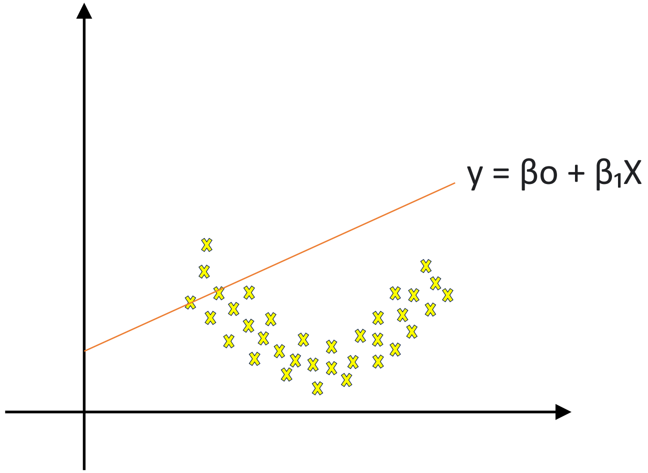


Supervised Learning

Polynomial Regression

Why Polynomial regression?

- Simple Linear Regression works effectively only when the relationship between the data is linear.
- If the data is non-linear then the linear regression fails to draw the best fit line



Supervised Learning

Polynomial Regression

Why Polynomial regression?

- To overcome this problem, Polynomial regression is used.
- Polynomial regression helps to identify the curvilinear relationship between the dependent and independent variables
- Polynomial regression builds upon linear regression to handle non-linear relationships between the independent and dependent variables. It achieves this by introducing terms raised to different powers (polynomials) of the independent variable, transforming the linear model into a non-linear one.

Supervised Learning

Polynomial Regression

Polynomial regression equation

- The polynomial regression equation is:

$$f_{(\vec{w},b)}(\vec{x}) = w_1x_1 + w_2x_2^2 + w_3x_3^3 + \ldots.. + w_nx_n^n + b$$

- Degree of order is considered as the **Hyperparameter**
- Using high degree of polynomial – tries to overfit the data
- Using low degree of polynomial – tries to underfit the data

Supervised Learning

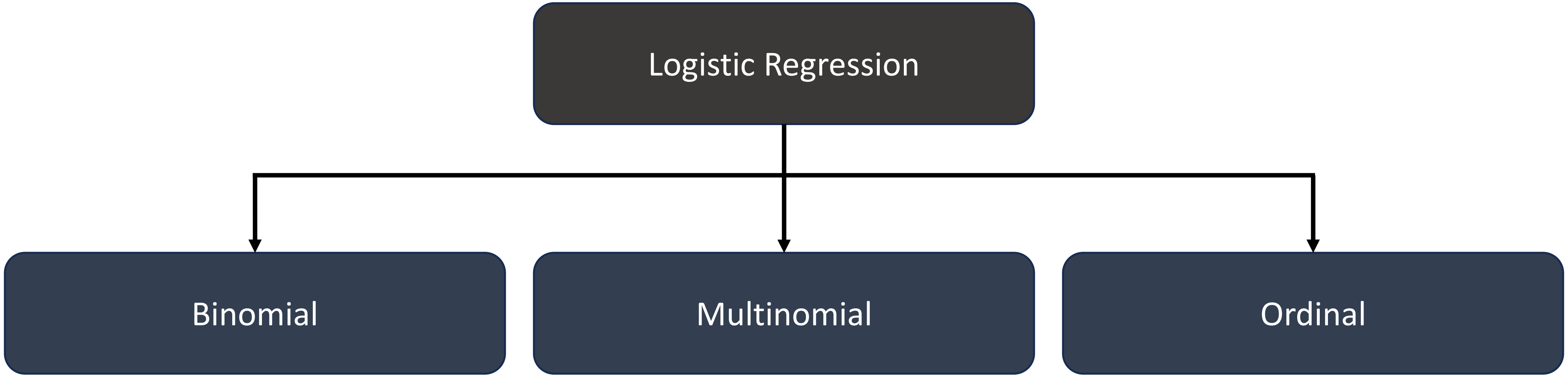
Logistic Regression

- Logistic regression is used for binary classification which takes input as independent variables and produces a probability value between 0 to 1
- Logistic regression uses a sigmoid function
- Instead of fitting a line, we fit a 'S' shaped logistic function, which predicts a maximum of two values 0 or 1
- The sigmoid function is used to map the predicted values to the probabilities, it maps any real value to another value within a range of 0 and 1.

Supervised Learning

Logistic Regression

Types of Logistic Regression



Supervised Learning

Logistic Regression

Assumptions of Logistic Regression

- Independent observations
- Binary dependent variables
- Linearity relationship between independent variables and log odds
- No outliers
- Large sample size

Supervised Learning

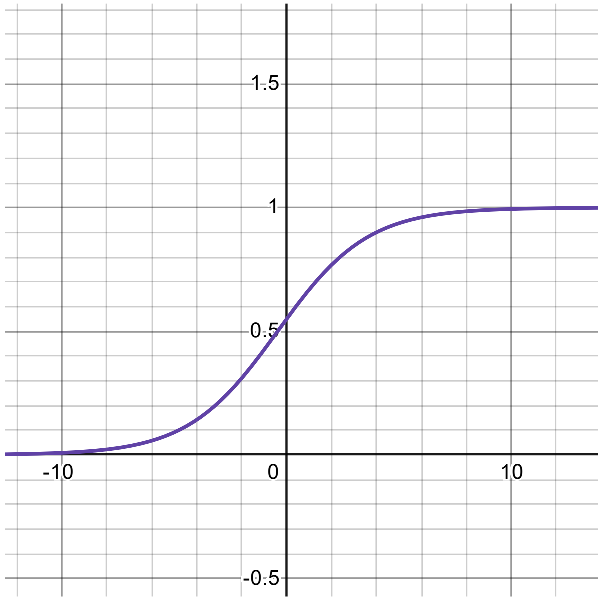
Logistic Regression - Equations

Sigmoid Function

$$S(x) = \frac{1}{1+e^{-x}}$$

S(x) tends towards 1 as x → ∞

S(x) tends towards 0 as x → - ∞



Logistic Regression

$$p(X; b, w) = \frac{1}{1+e^{-w.X+b}}$$

Supervised Learning

Applications of Logistic Regression

Manufacturing

They then plan maintenance schedules based on this estimate to minimize future failures.

Finance

Used to analyze financial transactions for fraud and assess loan applications and insurance applications for risk

Healthcare

They use logistic regression models to compare the impact of family history or genes on diseases

Marketing

Online advertising tools use the logistic regression model to predict if users will click on an advertisement

Supervised Learning

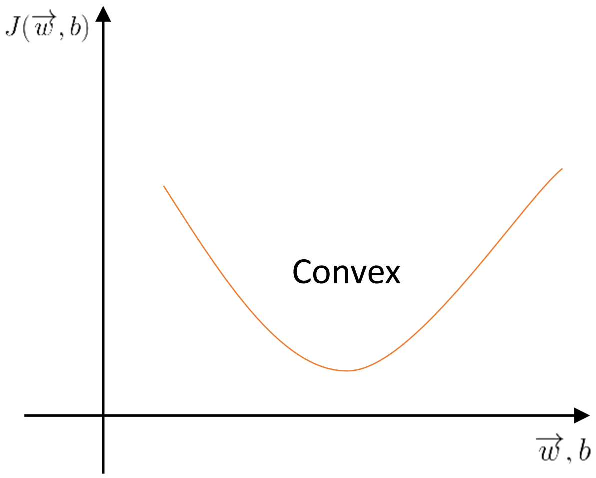
Logistic Regression vs Linear Regression – Which is better suited for classification?

- Classification problems require assigning data points to discrete categories.
- Linear regression, is capable of fitting a line to the data, can't directly predict the categories. Its output (a continuous value) wouldn't tell you definitively whether a data point belongs to class A or class B.
- Logistic regression transforms the linear regression output through a sigmoid function, it generates a probability between 0 and 1. This probability allows you to classify data points based on a threshold.

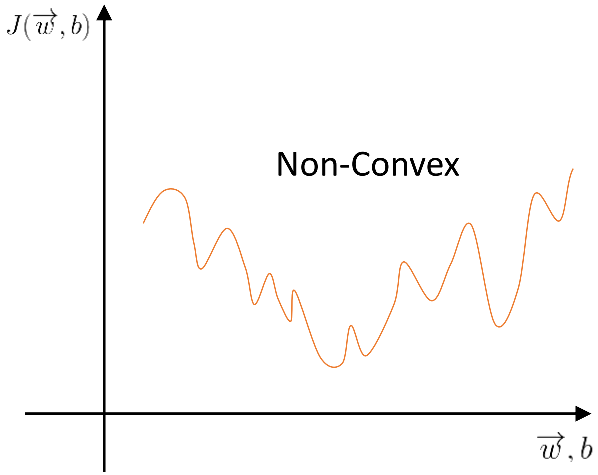
Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

- Why Mean squared error cost function is not suitable for Logistic Regression?
- On using MSE on Linear and Logistic regression,



For linear regression



For logistic regression

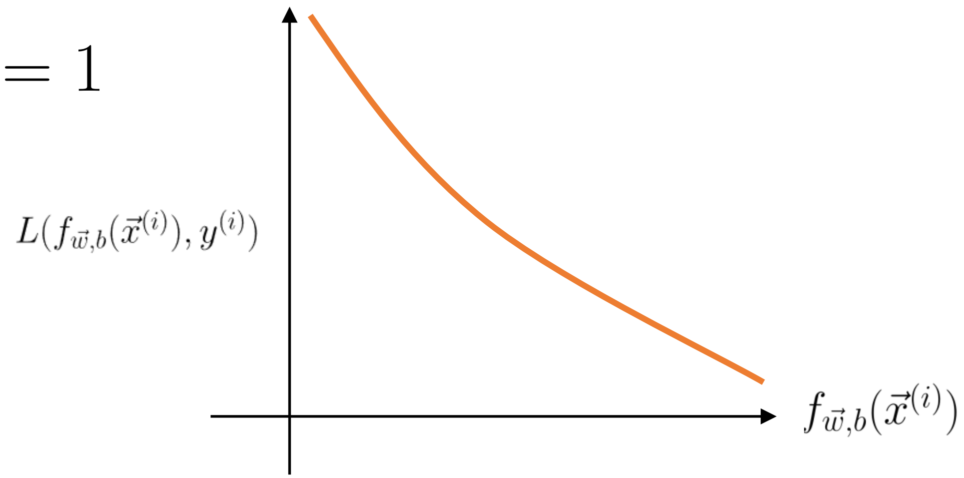
Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

Logistic Loss Function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{If } y^{(i)} = 1 \\ -log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{If } y^{(i)} = 0 \end{cases}$$

When $y^{(i)} = 1$



As $f_{\vec{w},b}(\vec{x}^{(i)})$ → 1, then **loss** → 0

As $f_{\vec{w},b}(\vec{x}^{(i)})$ → 0, then **loss** → ∞

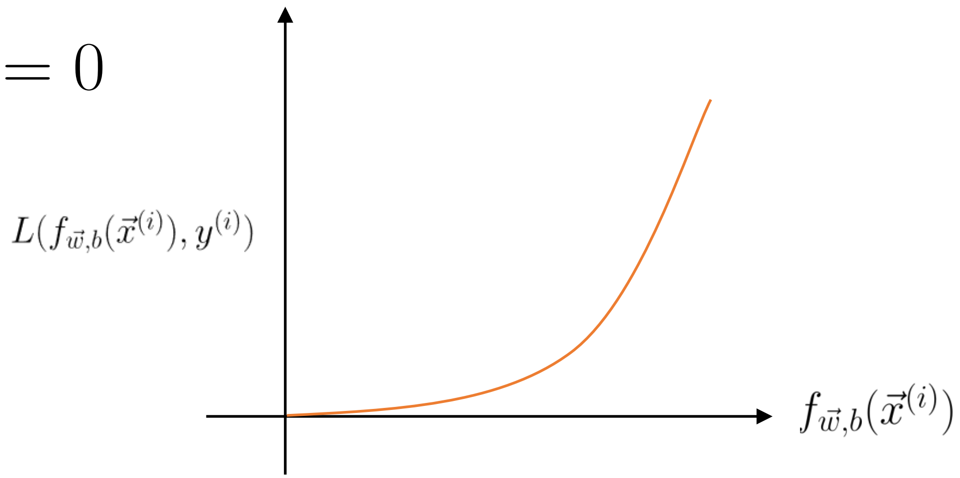
Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

Logistic Loss Function

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = \begin{cases} -log(f_{\vec{w},b}(\vec{x}^{(i)})) & \text{If } y^{(i)} = 1 \\ -log(1 - f_{\vec{w},b}(\vec{x}^{(i)})) & \text{If } y^{(i)} = 0 \end{cases}$$

When $y^{(i)} = 0$



As $f_{\vec{w},b}(\vec{x}^{(i)})$ → 1, then **loss** → ∞

As $f_{\vec{w},b}(\vec{x}^{(i)})$ → 0, then **loss** → 0

Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

Cost function of Logistic Regression

$$J(\vec{w}, b) = -\frac{1}{m}\sum_{i=1}^{m}\left[L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}))\right]$$

$$L(f_{\vec{w},b}(\vec{x}^{(i)}), y^{(i)}) = -y^{(i)}log(f_{\vec{w},b}(\vec{x}^{(i)})) - (1 - y^{(i)})log(1 - f_{\vec{w},b}(\vec{x}^{(i)}))$$

Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

Gradient Descent for Logistic Regression

Repeat{

$$w_j = w_j - \alpha \frac{\partial}{\partial w_j} J(\overrightarrow{w}, b)$$

$$\frac{\partial}{\partial w_j} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)}$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{w}, b)$$

$$\frac{\partial}{\partial b} J(\vec{w}, b) = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)})$$

}

Update Simultaneously

Supervised Learning

Cost Function and Gradient Descent for Logistic Regression

Gradient Descent for Logistic Regression

Repeat{

$$w_j = w_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) x_j^{(i)} \right]$$

$$b = b - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (f_{\vec{w},b}(\vec{x}^{(i)}) - y^{(i)}) \right]$$

} Simultaneous updates

Supervised Learning

Logistic Regression

Knowledge Check:

In Logistic Regression, what is the type of variable being predicted?

(a) categorical independent variable

(b) categorical dependent variable.

(c) numerical dependent variable.

(d) numerical independent variable.

Supervised Learning

Logistic Regression

Knowledge Check:

In Logistic Regression, what is the type of variable being predicted?

(a) categorical independent variable

(b) categorical dependent variable.

(c) numerical dependent variable.

(d) numerical independent variable.

Supervised Learning

Problem of Overfitting

Learning algorithms like linear and logistic regression can face issues such as overfitting and underfitting.

Overfitting happens when the model fits the training data too well, while underfitting occurs when the model doesn't fit the training data well enough.

Supervised Learning

Problem of Overfitting

1

Underfitting

A linear model may not capture the true relationship between house size and price.

2

Overfitting

A high-order polynomial model may fit the training data perfectly but perform poorly on new data.

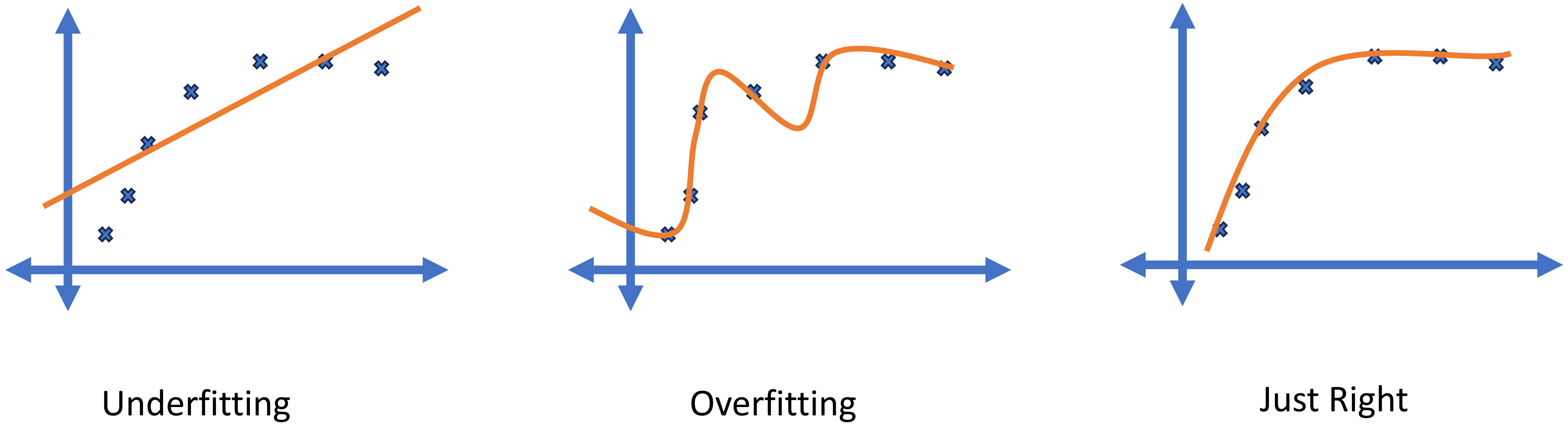
3

Just Right

A quadratic model balances fitting the training data and generalizing to new examples.

Supervised Learning

Problem of Overfitting



Supervised Learning

Addressing Overfitting

Collect More Data

Collecting more training data can help reduce overfitting by making the model less sensitive to noise in the training set.

Use Fewer Features

Use a subset of the most relevant features to reduce overfitting. This is called feature selection.

Apply Regularization

Regularization encourages the learning algorithm to use smaller parameter values, preventing features from having an overly large effect.

Supervised Learning

More Data Collection

Collect More Data

Collecting more training data can help reduce overfitting by making the model less sensitive to noise in the training set.

Larger Training Sets

Larger training sets help the model learn a less wiggly function, improving generalization.

1

2

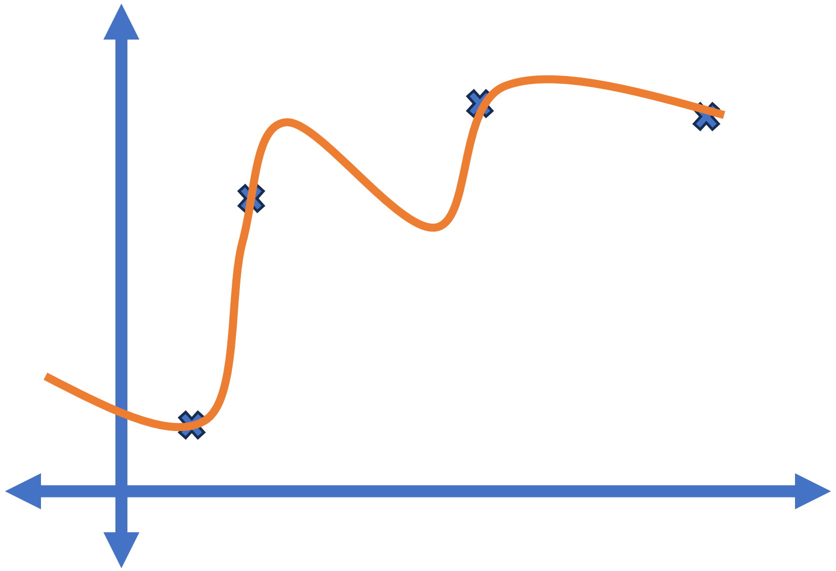
3

Example

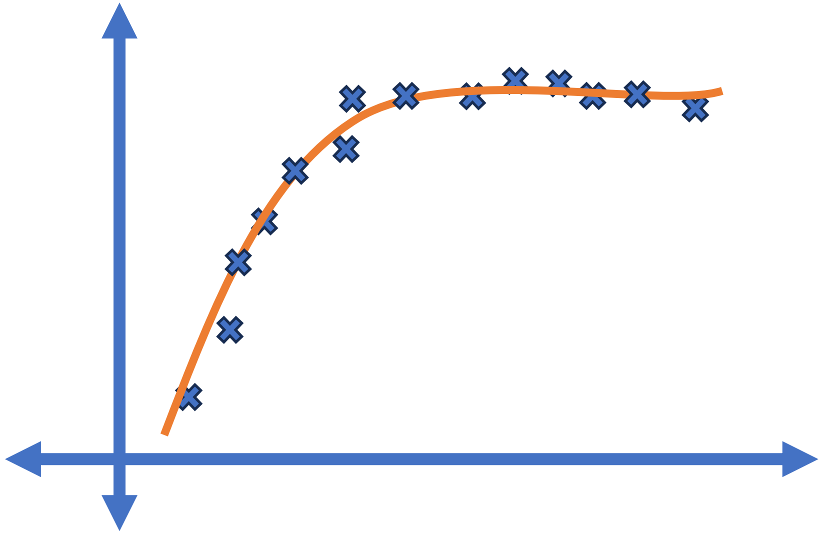
House price prediction with more data on sizes and prices of houses.

Supervised Learning

More Data Collection



Less training of data



More training of data

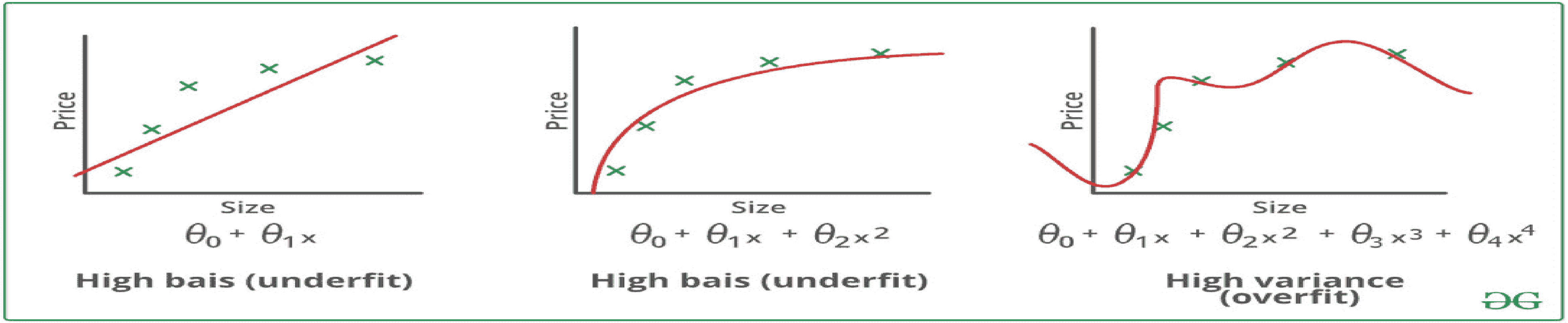
Supervised Learning

Selection of Features

- Use a subset of the most relevant features to reduce overfitting. This is called feature selection.

Supervised Learning

Bias and Variance



Bias refers to the errors which occur when we try to fit a statistical model on real-world data which does not fit perfectly well on some mathematical model.

If we use a way too simplistic a model to fit the data then we are more probably face the situation of **High Bias** which refers to the case when the model is unable to learn the patterns in the data at hand and hence performs poorly.

Variance implies the error value that occurs when we try to make predictions by using data that is not previously seen by the model. There is a situation known as **high variance** that occurs when the model learns noise that is present in the data.

Finding a proper balance between the two that is also known as the Bias-Variance Tradeoff can help us prune the model from getting overfitted to the training data.

Supervised Learning

Bias and Variance

- **High Bias, Low Variance:** A model that has high bias and low variance is considered to be underfitting.

- **High Variance, Low Bias:** A model that has high variance and low bias is considered to be overfitting.

- **High-Bias, High-Variance:** A model with high bias and high variance cannot capture underlying patterns and is too sensitive to training data changes. On average, the model will generate unreliable and inconsistent predictions.

- **Low Bias, Low Variance:** A model with low bias and low variance can capture data patterns and handle variations in training data. This is the perfect scenario for a machine learning model where it can generalize well to unseen data and make consistent, accurate predictions. However, in reality, this is not feasible.

The bias-variance trade-off is a fundamental concept in machine learning. It refers to the balance between bias and variance, which affect predictive model performance. Finding the right tradeoff is crucial for creating models that generalize well to new data.

Supervised Learning

Regularization

Regularization is a technique used to prevent overfitting by adding a penalty term to the loss function, discouraging the model from assigning too much importance to individual features or coefficients.

Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

- Lasso Regularization – L1 Regularization
- Ridge Regularization – L2 Regularization
- Elastic Net Regularization – L1 and L2 Regularization

Supervised Learning

Regulation

Regularization

Regularization: Encourages the learning algorithm to use smaller parameter values, preventing features from having an overly large effect.

Regularization Term

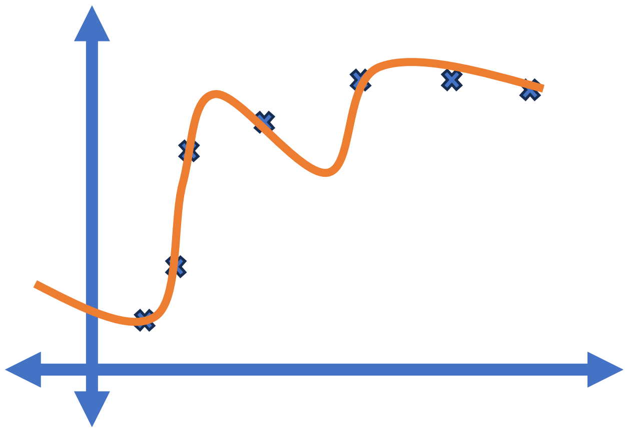
Penalize all features using a regularization term lambda * * sum(Wj^2).

Choosing Lambda

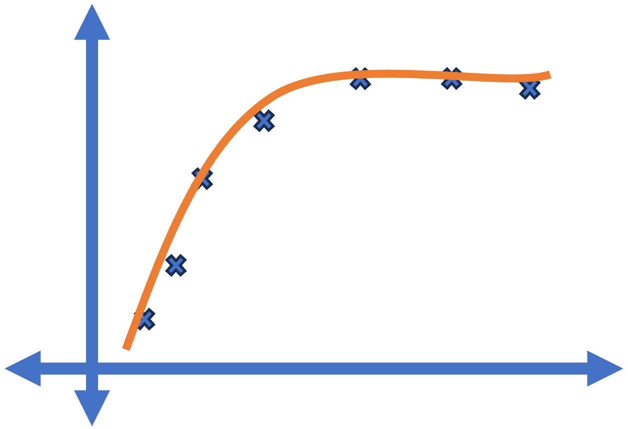
Lambda (λ) is the regularization regularization parameter that that controls the trade-off between fitting the training data and keeping parameters parameters small.

Supervised Learning

Regulation



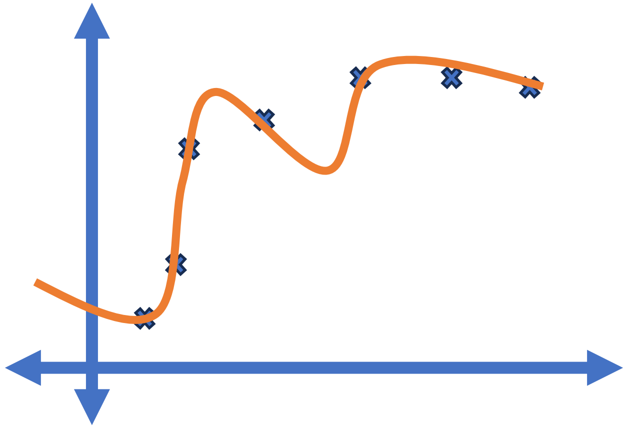
$$f(x) = 90x - 35x^2 + 5x^3 - 17x^3 + 8x^4 - 100$$



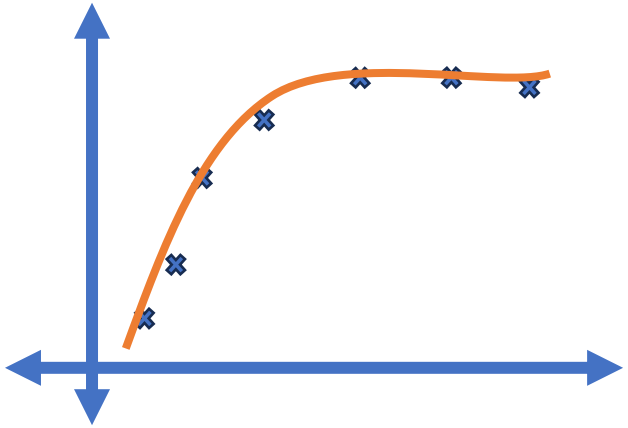
$$f(x) = 17x - 29x^2 + 0.0000065x^3 - 17x^3 + 0.000014x^4 - 100$$

Supervised Learning

Regulation



$$a_1x + a_2x^2 + a_3x^3 + a_4x^4 + b$$



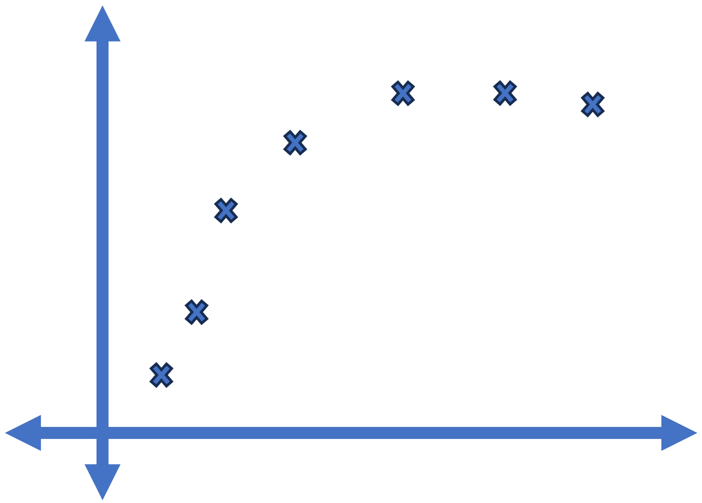
$$a_1x + a_2x^2 + b$$

$$min_{\vec{w},y} 1/2m \sum (f_{\vec{w},b})(\vec{x}^{(i)} - \vec{y}^{(i)})^2 + 1000a_3^2 + 1000a_4^2$$ → Reduce the a_3 and a_4 values, near to 0

Supervised Learning

Regularization

$$min_{\vec{a},b} J(\vec{a}, b) = f_{\vec{a},b}(\vec{x}) = min_{\vec{a},b} \underbrace{1/2m \sum (f_{\vec{a},b})(\vec{x}^{(i)} - \vec{y}^{(i)})^2}_{\text{Mean squared error}} + \underbrace{\lambda/2m \sum_{j=1}^{n} a_j^2}_{\text{Regularization term}}$$



$$f_{\vec{w},y}(\vec{x}) = a_1x + a_2x^2 + a_3x^3 + a_4x^4 + b$$

Supervised Learning

Regularized Linear Regression

$$min_{\vec{a},b} J(\vec{a},b) = f_{\vec{a},b}(\vec{x}) = min_{\vec{a},b} 1/2m \sum (f_{\vec{a},b})(\vec{x}^{(i)} - \vec{y}^{(i)})^2 + \lambda/2m \sum_{j=1}^{n} a_j^2$$

Gradient Descent

repeat{

$$a_j = a_j - \alpha \frac{\partial}{\partial a_j} J(\vec{a},b)$$

$$\frac{\partial}{\partial a_j} J(\vec{a},b) = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{a},b}(\vec{x}^{(i)} - (\vec{y}^{(i)}))x_j^{(i)} + \frac{\lambda}{m} a_j$$

$$b = b - \alpha \frac{\partial}{\partial b} J(\vec{a},b)$$

$$\frac{\partial}{\partial b} J(\vec{a},b) = \frac{1}{m} \sum_{i=1}^{m} (f_{\vec{a},b}(\vec{x}^{(i)} - (\vec{y}^{(i)}))$$

}simultaneous update

Supervised Learning

Regularized Linear Regression

repeat{

$$a_j = a_j - \alpha[\frac{1}{m}\sum_{i=1}^{m}(f_{\vec{a},b}(\vec{x}^{(i)} - \vec{y}^{(i)}))x_j^{(i)} + \frac{\lambda}{m}a_j]$$

$$b = b - \alpha\frac{1}{m}\sum_{i=1}^{m}(f_{\vec{a},b}(\vec{x}^{(i)} - \vec{y}^{(i)}))$$

}

Supervised Learning

Regularized Logistic Regression

$$Z = a_1x_1 + a_2x_2 + a_3x_1^2x_2 + a_4x_1^2x_2^2 + a_5x_1^2x_2^3 + ... + b$$

$$f_{\vec{a},b}(\vec{x}) = \frac{1}{1+e^{-z}}$$

$$J(\vec{a},b) = -\frac{1}{m}\sum_{i=1}^{m}[(y^{(i)}log(f_{\vec{a},b}(\vec{x}^{(i)}))+(1-y^{(i)})log(1-f_{\vec{a},b}(\vec{x}^{(i)}))]+\frac{\lambda}{2m}\sum_{j=1}^{n}a_j^2$$

Supervised Learning

Ovefitting

Knowledge Check:

Which of the following statements accurately describes overfitting in machine learning?

(a) The model performs well on both training and unseen data.

(b) The model performs well on training data but poorly on unseen data due to high variance.

(c) The model performs poorly on both training and unseen data due to low bias.

(d) The model performs well on training data but poorly on unseen data due to capturing noise in the data.

Supervised Learning

Ovefitting

Knowledge Check:

Which of the following statements accurately describes overfitting in machine learning?

(a) The model performs well on both training and unseen data.

(b) The model performs well on training data but poorly on unseen data due to high variance.

(c) The model performs poorly on both training and unseen data due to low bias.

(d) The model performs well on training data but poorly on unseen data due to capturing noise in the data.

Thank You !!!