Machine Learning

Definition:

The field of study that gives computers the ability to learn without being explicitly programmed. - *Arthur Samuel.*

Types:

1. Supervised Learning
2. Unsupervised Learning

Generally supervised learning is used most in real-world applications which have rapid advancements.

Then the other course, which will give us deep insights about unsupervised machine learning, reinforcement learning and recommended systems.

Supervised Learning:

It refers to algorithms that learn input to output mappings.

The key characteristic of supervised learning is that you give your learning algorithms to learn from inputs. That includes right answers, which mean, the correct label of 'y' for a given input of 'x'. Here, ‘x’ is the input and ‘y’ is the desired outputs that we give.

By seeing the correct pairs of input x and desired output label y, the learning algorithm eventually learns to take just the input alone without the output label and gives a reasonably accurate prediction of the output.

common examples of supervised machine learning:

Input (x)	output (y)	Application
email	Spam	spam filtering
audio	Text transcripts	speech recognition
English	Spanish	machine translation
image, radar	position of cars	self-driving cars
phone image	Defects (0/1)	visual representation

Supervised Machine learning is classified into the following types:

1. **Regression**: Predict a number form infinitely many possible outputs.

Example: Housing prices prediction.

These gives of idea how the algorithm fits curves or lines with the given inputs and deserved outputs. Finally, it gives us the output for any given input using the data that we provided that is, x and y from the graph.

2. **Classification**: Predict categories from small number of possible outputs. It has a very small number of possible outputs which make it different from regression.

" So, to summarize classification algorithms we predict the categories."

Example: Breast Cancer detection.

In this example, there are only two possible outputs the cancer may be Benign or Malignant. we can also provide as many inputs as we can for example we can gives the age, the sizes of the tumor.

Supervised Machine Learning

Regression:

1. Linear Regression Model:

Example: Housing prices prediction

Here, in this example we are going to use a data set on house sizes and prices.

We need to estimate the price of a certain house based on the data we provide. So, we can build a Linear regression model form this dataset.

Your model will fit a straight line to the data, based on this we can train the model.

Notations for describing the data:

1. Training set: The dataset that is used to train the model.
2. x: Input variable aka. Feature or an input feature.
3. y: Target variable, which denotes the output variable which we are trying to predict.
4. m: Total number of training examples used.
5. (x, y): to indicate the single training example.

(x^i, y^i) = Here 'i' indicates the index of the training set and refers to row 'i' in the table.

Function: Both the input variable and output(target) variable will generate some function "f".

The function 'f' is called the model.

X is called the input or input feature.

y-hat (y^): The output of the model is the "prediction". The model's prediction is estimated value of 'y'.

The formula to compute **f** is:

$$f\ w\ b\ (X) = w*X + b$$

(or)

$$f(X) = w*X + b$$

This represents the Linear regression with One variable. Also called as, Univariate linear regression.

Cost function Formula:

The cost function evaluates the disparity between the model's prediction (y) and the actual true value (y).

$$y^{(i)} = f_{w,b}(x^{(i)}, f_{w,b}(x^{(i)}) = w\,x^{(i)} + b$$

Hence, the cost function utilizing the Squared Error Cost Function is represented as:

$$J_{(w,b)} = 1/2m \sum_{i=1}^{i=m} (y^{(i)} - f_{w,b}(x^{(i)}))^2$$

Where m represents the number of training examples.

The simplified version of the cost function is:

$$J(w,b) = 1/2m \sum_{i=1}^{i=m} (f_{w,b}(x_{(i)}) - y_{(i)})^2$$

Gradient Descent

Gradient Descent is employed for minimizing the cost function, particularly for linear regression or any similar function.

Outline:

Begin with initial values for w and b (e.g., $w=0$, $b=0$), then iteratively update w and b to minimize $J(w,b)$ until reaching or approximating a minimum.

How to Implement the Algorithm:

Repeat until convergence:

$$w = w - \alpha\, d/dw J_{(w,b)}$$

$$b = b - \alpha\, d/db J_{(w,b)}$$

Where α represents the learning rate.

Learning rate in Gradient descent:

if alpha is too small: Gd may be slow.

if alpha is too large: Gd may be:

- Overshoot, never reach minimum
- fail to converge, diverge

So, we can replace the derivate part with the following, rewriting the above formula.

repeat until convergence {

$$w=w-\alpha \cdot m1\sum i=1m(fw,b(x(i))-y(i)) \cdot x(i)$$

$$b=b-\alpha \cdot m1\sum i=1m(fw,b(x(i))-y(i))$$

}

Gradient Descent:(Gd)

if we have multiple features the we can use vectorization to find the cost.

An alternative to Gd:

=> Normal equation:

- . Only for linear regression
- . Solve for w b without iterations

=> Disadvantages:

- . Doesn't generalize to other learning algorithms
- . Slow when no of features is large (>10k)

=> What you need to know:

. Normal equation method may be used in machine learning libraries that implement linear regression.

. Gd is the recommended method for finding parameters w b

Feature Engineering:

Using intution to design new features, by transforming or combining original features.

Logistic Regression:

It is used in predicting the outputs of the classification models where we determine the output from the probability of cost formula.

for example:

To identify whether a tumor is malignant or benign. we plot a graph with tumor size on x-axis and where y-axis shows 0 or 1 i.e; 0 => tumor is benign and 1 => tumor is malignant.

Point to remember: Since the prediction values may not always lies on y=1 and y=0. There may be a probability that the values may fall between 0 and 1.

To predict the output for those values we use logistic regression which determines the output by the threshold value.

Interpretation of logistic regression output:

$$f_{w\,b}(x) = 1 / (1 + e^{-1(w*x + b)})$$

which gives the probability.

Example:

x is "tumor size"

y is 0 (not malignant)

or 1(malignant)

$f_{w\,b}(x) = 0.7$

70% chance that y is 1

logistic regression uses Sigmoid function which is also known as **Logistic function**.

The sigmoid function which maps all input values between 0 and 1.

Formula for Sigmoid function:

$g(z) = 1 / (1 + e^{(-z)})$

$0 < g(z) < 1$

Decision Boundary:

It is the separator above which the values fall under positive region or close to 1. Below which the values fall under negative region or close to 0.

$$f_{w\,b}(x) = g(z) = g(w_1x_1 + w_2x_2 + b)$$

$$Db: z = w^*x + b = 0$$

Non-linear Decision boundaries:

There is a probability that the curve may not be a straight line.

$$f_{w\,b}(x) = g^*(w_1x_1^2 + w_2x_2^2) + b)$$

let $w_1=w_2=1$ and $b= -1$

$$z = x_1^2 + x_2^2 - 1 = 0 \Rightarrow x_1^2 + x_2^2 = 1$$

In such cases, we may even can get complex boundaries. So, the curve anything.

Cost function for logistic regression which is non-convex.

Logistic loss function:

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

it is valid for both 0 and 1.

Gradient Descent Implementation:

$$w_j = w_j - \alpha\,[\,1/m\; {}^*\textstyle\sum\,(1\Rightarrow m)\;(fw,b(xi)-yi)xj(i)]$$

$$b = b - \alpha\,[\,1/m\; {}^*\textstyle\sum\,(1\Rightarrow m)\;(fw,b(xi)-yi)]$$