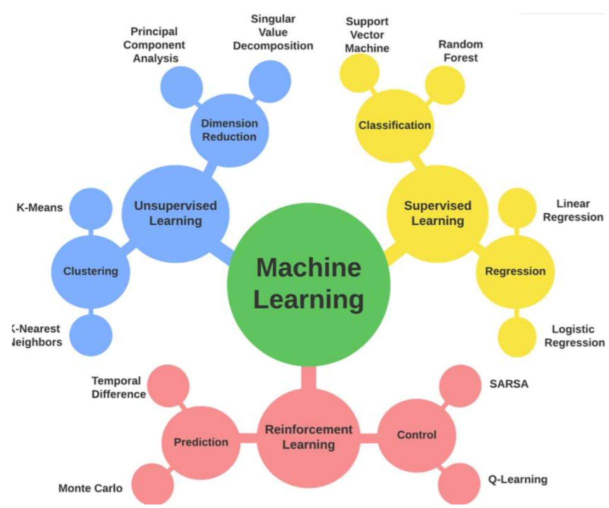
Các phương pháp học máy
Machine learning methods

4 TC: 2 LT – 2 TH

Giảng viên: **Tạ Hoàng Thắng**

tahoangthang@gmail.com

0975399307



Classification (ML)

Definitions

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

- The model is fully trained using the training data, and then it is evaluated on validation data before predicting on new unseen data (test data).
- https://www.datacamp.com/blog/classification-machine-learning

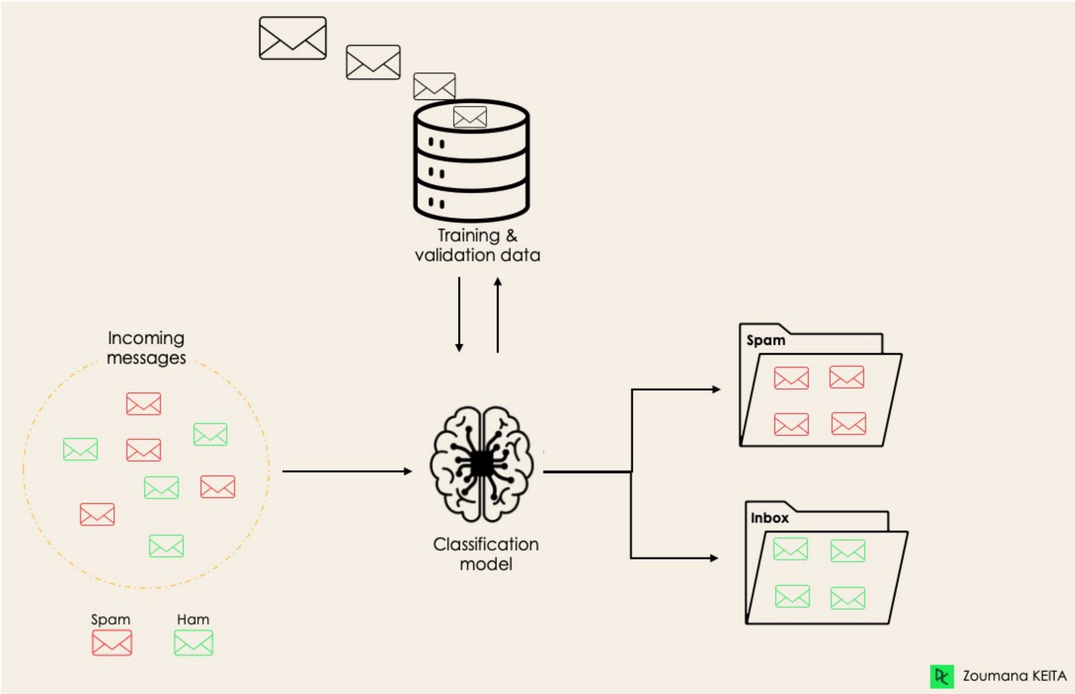
Wikipedia: When classification is performed by a computer, statistical methods are normally used to develop the algorithm.

- The individual observations are analyzed into a set of quantifiable properties, known variously as explanatory variables or features.
- These properties may variously be categorical (e.g. "A", "B", "AB" or "O", for blood type), ordinal (e.g. "large", "medium" or "small").
- https://en.wikipedia.org/wiki/Statistical_classification

Classification (ML)

Examples

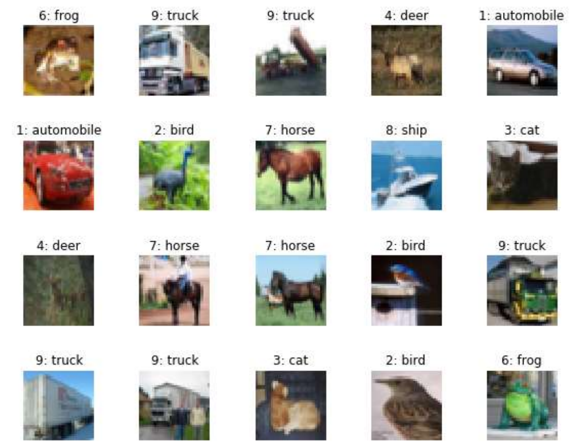
An algorithm can learn to predict whether a given email is **spam or ham (no spam).**



Classification (ML)

Other examples

- Image Classification: CIFAR10, ImageWoof
- Text Classification: GLUE
- Audio Classification: BirdSet, UrbanSound8K



Check your tasks and datasets in HuggingFace.

- https://huggingface.co/datasets?language=language:en&sort=trending

Classification (ML)

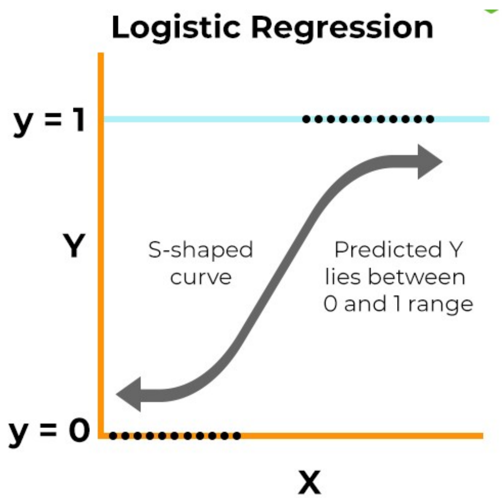
Lazy Learners Vs. Eager Learners

Eager learners are machine learning algorithms that first build a model from the training dataset before making any prediction on future datasets.

- **More time during the training process**
- **Require less time to make predictions**

Some examples:

- **Logistic Regression**
- Support Vector Machine
- Decision Trees
- Artificial Neural Networks



Classification (ML)

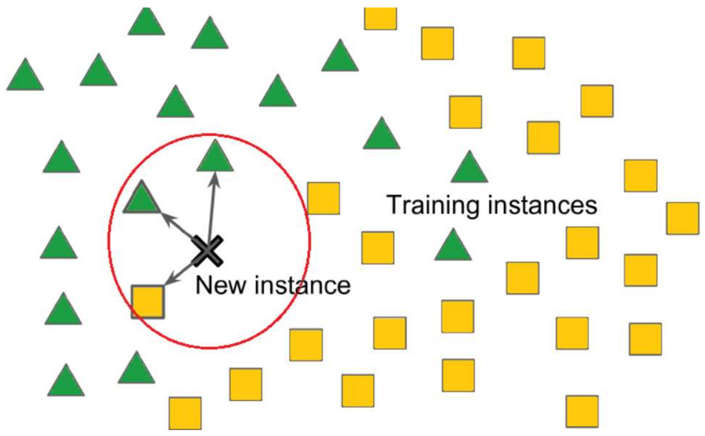
Lazy Learners Vs. Eager Learners

Lazy learners or instance-based learners, on the other hand, do not create any model immediately from the training data, and this is where the lazy aspect comes from.

- memorize the training data, and each time there is a need to make a prediction, they search for the nearest neighbor from the whole training data
  - very slow during prediction.

Some examples of this kind are:

- K-Nearest Neighbor
- Case-based reasoning



Classification (ML)

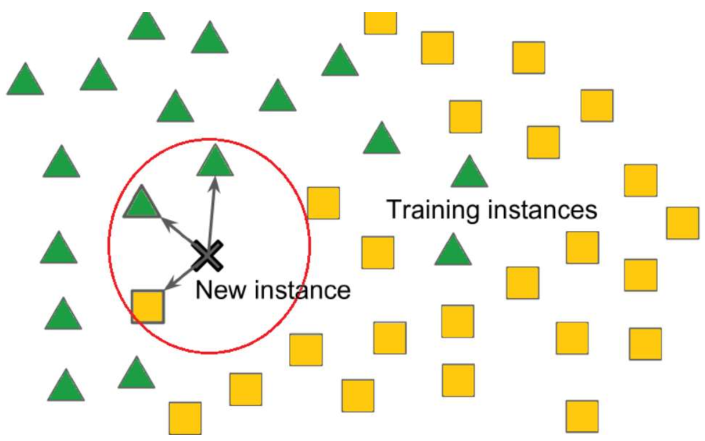
Lazy Learners Vs. Eager Learners

Lazy learners or instance-based learners, on the other hand, do not create any model immediately from the training data, and this is where the lazy aspect comes from.

- memorize the training data, and each time there is a need to make a prediction, they search for the nearest neighbor from the whole training data
  - very slow during prediction.

Some examples of this kind are:

- K-Nearest Neighbor
- Case-based reasoning

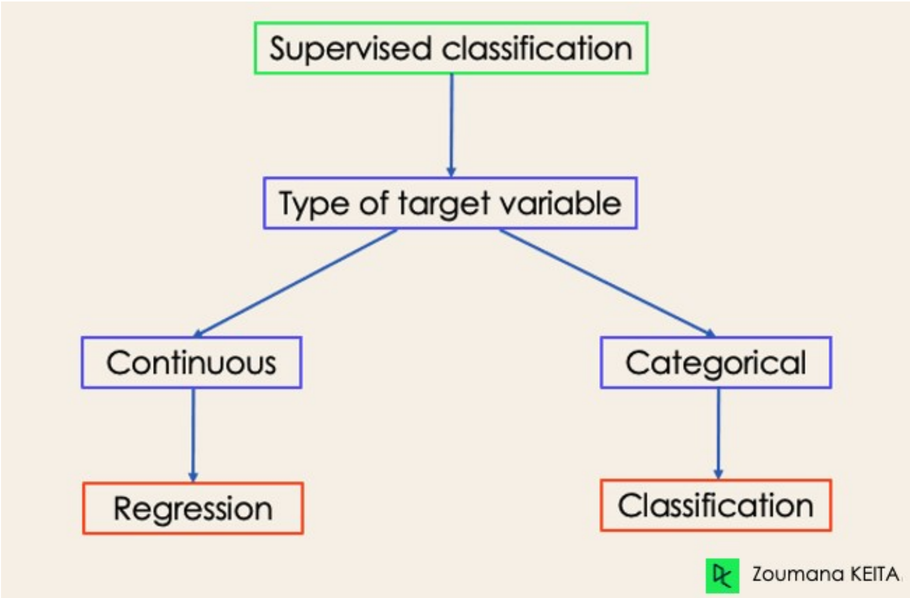


Classification (ML)

Machine Learning Classification Vs. Regression

There are **four main categories of Machine Learning algorithms**: supervised, unsupervised, semi-supervised, and reinforcement learning.

- Even though classification and regression are both from the category of supervised learning, they are not the same.

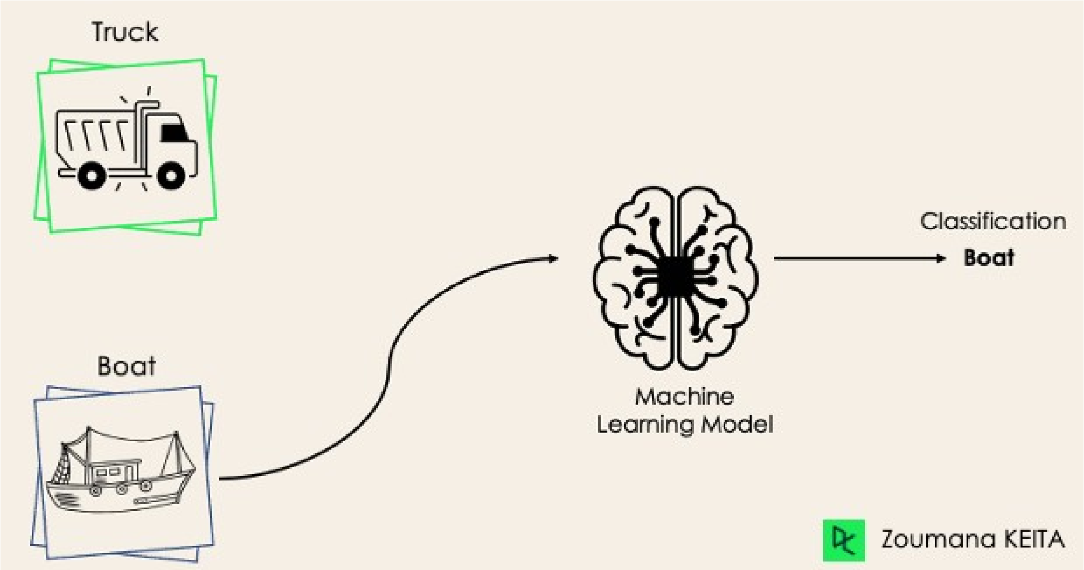


Classification (ML)

Different Types of Classification Tasks in Machine Learning

Binary Classification: In a binary classification task, the goal is to classify the input data into **two mutually exclusive categories**.

- The training data in such a situation is labeled in a binary format: **true and false; positive and negative; O and 1; spam and not spam**…

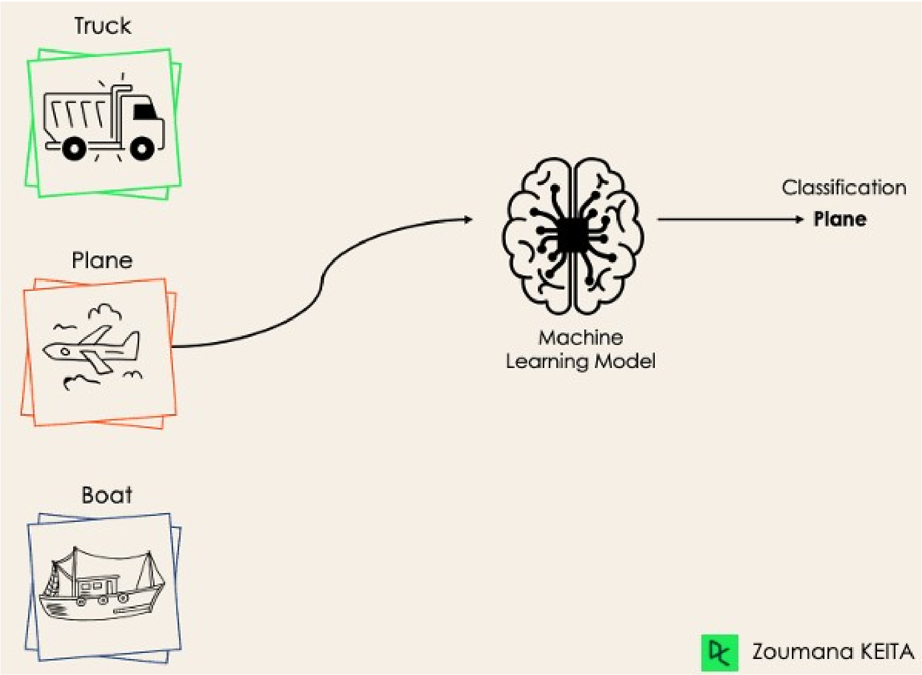


Classification (ML)

Different Types of Classification Tasks in Machine Learning

Multi-Class Classification: The multi-class classification, on the other hand, has at least two mutually exclusive class labels,

- **The goal is to predict to which class a given input example belongs to.**



Classification (ML)

Different Types of Classification Tasks in Machine Learning

Multi-Class Classification: The multi-class classification, on the other hand, has at least two mutually exclusive class labels,

- **The goal is to predict to which class a given input example belongs to.**

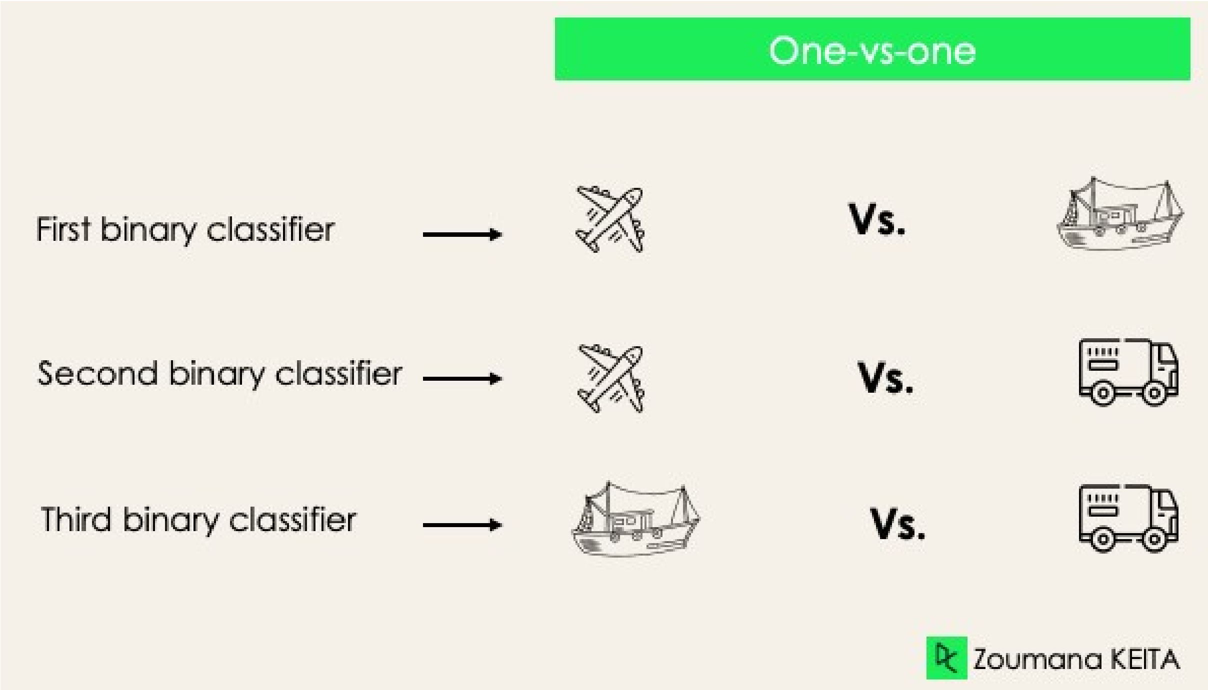
Most of the binary classification algorithms can be also used for multi-class classification.

- Random Forest, Naive Bayes
- K-Nearest Neighbors, Gradient Boosting
- SVM, Logistic Regression
- **CatBoost, LightGBM, XGBoost**

Classification (ML)

Different Types of Classification Tasks in Machine Learning

One-versus-one: this strategy trains as many classifiers as there are pairs of labels. If we have a 3-class classification, we will have three pairs of labels, thus three classifiers, as shown below.



Classification (ML)

Different Types of Classification Tasks in Machine Learning

One-versus-one: this strategy trains as many classifiers as there are pairs of labels. If we have a 3-class classification, we will have three pairs of labels, thus three classifiers, as shown below.

In general, for N labels, we will have N(N-1)/2 classifiers.

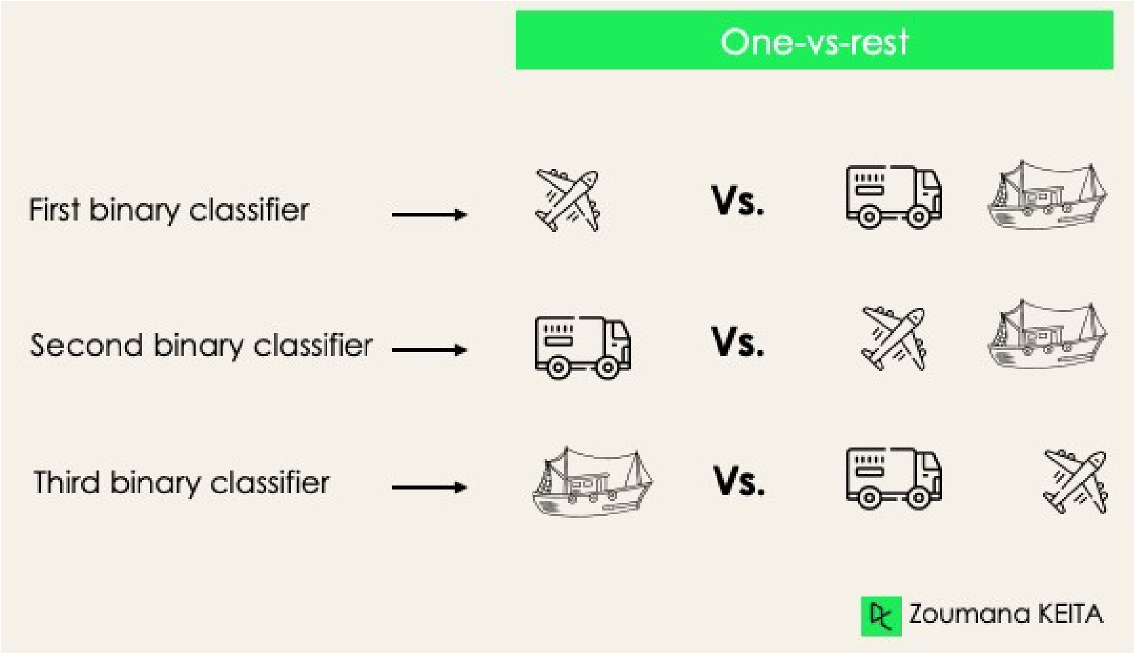
- **Each classifier is trained on a single binary dataset, and the final class is predicted by a majority vote between all the classifiers.**

One-vs-one approach works best for SVM and other kernel-based algorithms.

Classification (ML)

Different Types of Classification Tasks in Machine Learning

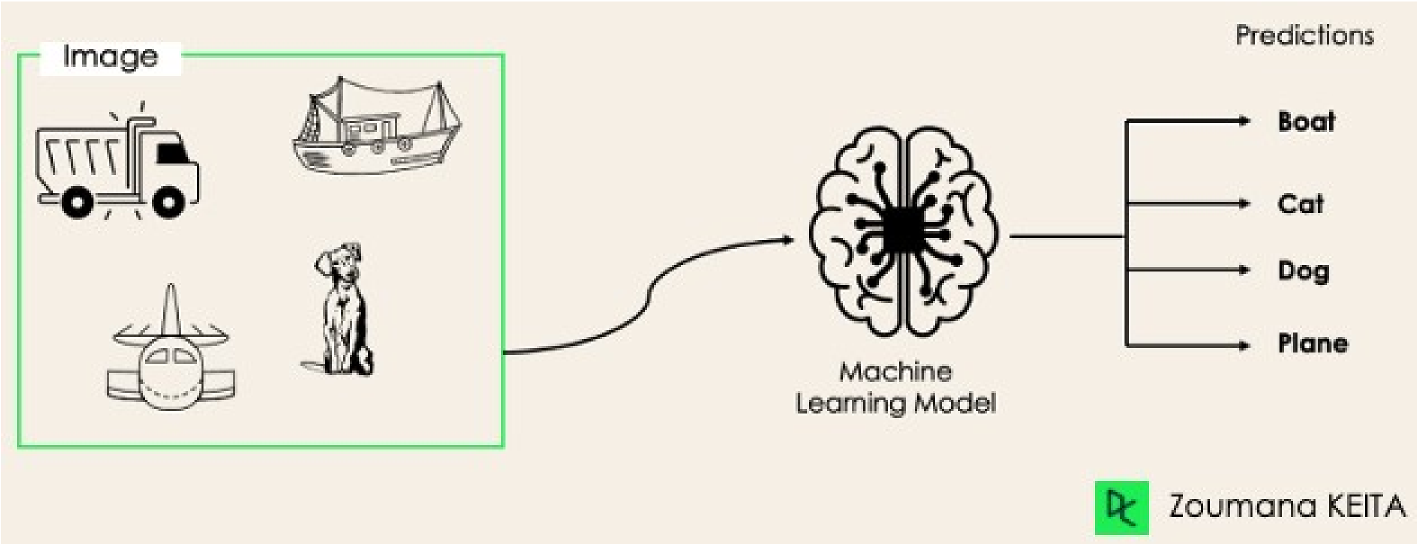
One-versus-rest: at this stage, we start by considering each label as an independent label and consider the rest combined as only one label. With 3-classes, we will have three classifiers.



Classification (ML)

Different Types of Classification Tasks in Machine Learning

Multi-Label Classification: In multi-label classification tasks, we try to predict 0 or more classes for each input example. In this case, there is no mutual exclusion because the input example can have more than one label.



Classification (ML)

Different Types of Classification Tasks in Machine Learning

Multi-Label Classification: In multi-label classification tasks, we try to predict 0 or more classes for each input example. In this case, there is no mutual exclusion because the input example can have more than one label.

Some versions of multi-label algorithms.

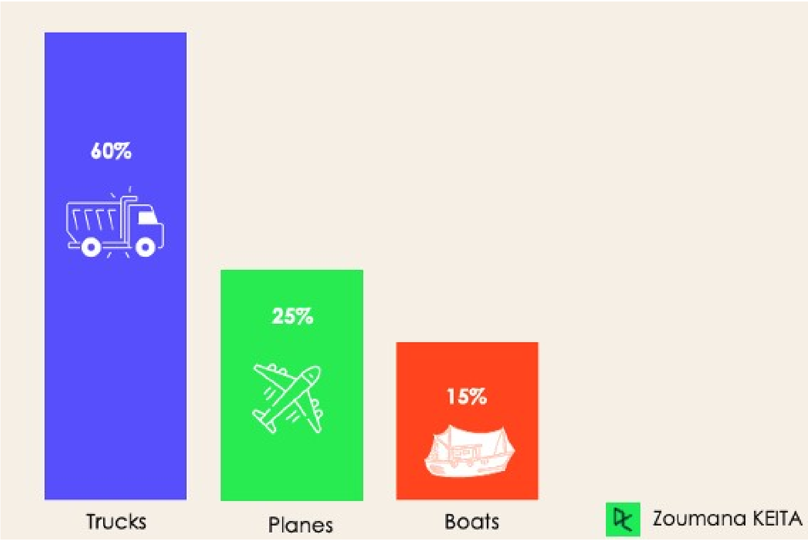
- Multi-label Decision Trees
- Multi-label Gradient Boosting
- Multi-label Random Forests

Classification (ML)

Imbalanced Classification

For the imbalanced classification, the number of examples is unevenly distributed in each class, meaning that we can have more of one class than the others in the training data.

Let's consider the following 3-class classification scenario where the training data contains: 60% of trucks, 25% of planes, and 15% of boats.



Classification (ML)

Example of DepressionEmo

- https://github.com/abuBakarSiddiqurRahman/DepressionEmo
- SVM, Light GBM, and XGBoost, BERT, GAN-BERT, BART

Table 1: Several chosen examples in DepressionEmo.

Text (Title + Post	Date	Emotions
it's okay to cry ### After 8 months of break up and 4 months of no contact, I finally cried for the first time. It was a strange kind of cry.... a cry that I knew I had no emotions and I do not want her back in my life (even after she dumped me through an email). To all that are going through a break up... you will have relapses and will go through different stages of break up... stay strong, be kind to yourself, speak to love ones and remember guys... it's okay to cry.	2020-06-29 01:57:02	sadness, emptiness, hopelessness

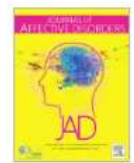
Classification (ML)

Example of DepressionEmo

- https://github.com/abuBakarSiddiqurRahman/DepressionEmo
- SVM, Light GBM, and XGBoost, BERT, GAN-BERT, BART

ELSEVIER

Journal of Affective Disorders
Volume 366, 1 December 2024, Pages 445-458



Research paper

DepressionEmo: A novel dataset for multilabel classification of depression emotions

Abu Bakar Siddiqur Rahman [a], Hoang-Thang Ta [b], Lotfollah Najjar [a], Azad Azadmanesh [a], Ali Saffet Gönul [c]

Show more

Logistics Regression

Definitions

Wikipedia: In statistics, the logistic model (or logit model) is a **statistical model that models the log-odds of an event as a linear combination of one or more independent variables.** In regression analysis, logistic regression (or logit regression) estimates the parameters of a logistic model (the coefficients in the linear or non linear combinations).

Geeksforgeeks: Logistic regression is a supervised machine learning algorithm used for classification tasks where **the goal is to predict the probability that an instance belongs to a given class or not**.

- Logistic regression is a statistical algorithm which analyze the relationship between two data factors.

Logistics Regression

Some references:

- https://machinelearningcoban.com/2017/01/27/logisticregression/
- https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html
- https://www.geeksforgeeks.org/understanding-logistic-regression/
- https://en.wikipedia.org/wiki/Logistic_regression

Logistics Regression

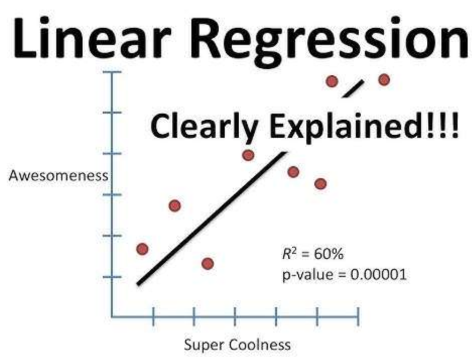
Logistic regression is used for **binary classification** where we use **sigmoid function**, that takes input as independent variables and produces a probability value between 0 and 1.

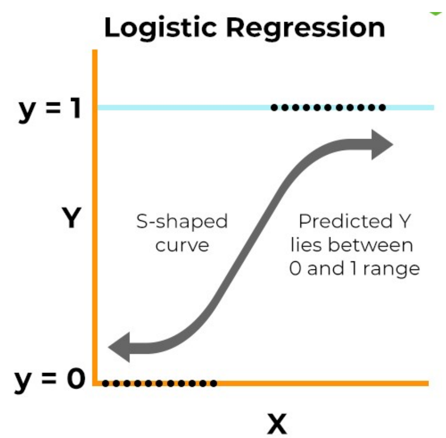
For example, we have two classes **Class 0** and **Class 1** if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 otherwise it belongs to Class 0.

Logistics Regression

Key Points:

- Logistic regression predicts the output of a categorical dependent variable. Therefore, **the outcome must be a categorical or discrete value**.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the **probabilistic values** which lie between 0 and 1.
- In logistic regression, instead of fitting a regression line, we fit an “S” shaped logistic function, which predicts two maximum values (0 or 1).





Logistics Regression

Logistic Function – Sigmoid Function

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the “S” form.
- The S-form curve is called the **Sigmoid function** or the logistic function.
- In logistic regression, we use the concept of the **threshold value**, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Logistics Regression

Logistic Function – Sigmoid Function

Formula:

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

Key Characteristics:

1. **Output Range**: The sigmoid function outputs values in the range (0, 1). This makes it particularly useful for modeling probabilities.
2. **Shape**: The curve is S-shaped (sigmoidal), with the following properties:
   - As x approaches negative infinity, $\sigma(x)$ approaches 0.
   - As x approaches positive infinity, $\sigma(x)$ approaches 1.
   - At $x = 0$, the output is $\sigma(0) = 0.5$.
3. **Derivative**: The derivative of the sigmoid function is:

$$\sigma'(x) = \sigma(x)(1 - \sigma(x))$$

   This property is useful for optimization algorithms, such as gradient descent, in neural networks.

Logistics Regression

Logistic Function – Sigmoid Function

Formula:

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

Calculate sigma(x) when **x = 2, x = -2, x = 5, and x = -5**.

Logistics Regression

Logistic Function – Sigmoid Function

Python code:

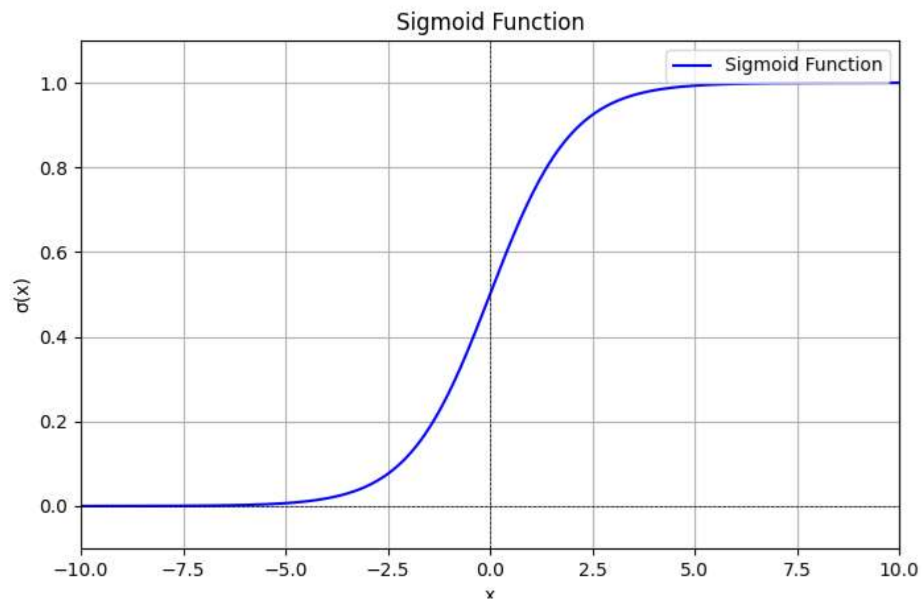
```python
import numpy as np
import matplotlib.pyplot as plt

# Define the sigmoid function
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Generate x values
x = np.linspace(-10, 10, 400)
# Calculate the corresponding sigmoid values
y = sigmoid(x)

# Create the plot
plt.figure(figsize=(8, 5))
plt.plot(x, y, label='Sigmoid Function', color='blue')
plt.title('Sigmoid Function')
plt.xlabel('x')
plt.ylabel('σ(x)')
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.xlim(-10, 10)
plt.ylim(-0.1, 1.1)
plt.grid()
plt.legend()
plt.show()

```



Logistics Regression

Logistic Function – Sigmoid Function

Python code:

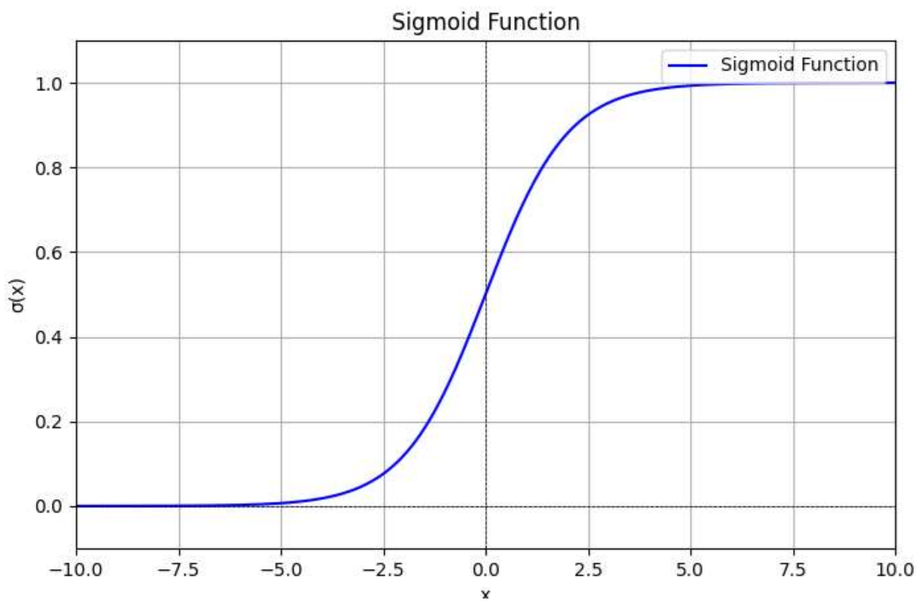
```python
import numpy as np
import matplotlib.pyplot as plt

# Define the sigmoid function
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Generate x values
x = np.linspace(-10, 10, 400)
# Calculate the corresponding sigmoid values
y = sigmoid(x)

# Create the plot
plt.figure(figsize=(8, 5))
plt.plot(x, y, label='Sigmoid Function', color='blue')
plt.title('Sigmoid Function')
plt.xlabel('x')
plt.ylabel('σ(x)')
plt.axhline(0, color='black', linewidth=0.5, ls='--')
plt.axvline(0, color='black', linewidth=0.5, ls='--')
plt.xlim(-10, 10)
plt.ylim(-0.1, 1.1)
plt.grid()
plt.legend()
plt.show()

```



Logistics Regression

Types of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

- **Binomial:** In binomial Logistic regression, there can be **only two possible types of the dependent variables**, such as 0 or 1, Pass or Fail, etc.
- **Multinomial:** In multinomial Logistic regression, there can be **3 or more possible unordered types of the dependent variable**, such as “cat”, “dogs”, or “sheep”
- **Ordinal:** In ordinal Logistic regression, there can be **3 or more possible ordered types of dependent variables**, such as “low”, “Medium”, or “High”.

Logistics Regression

Terminologies involved in Logistic Regression

- **Independent variables**: The input characteristics or predictor factors applied to the dependent variable's predictions.
- **Dependent variable:** The target variable in a logistic regression model, which we are trying to predict.
- **Logistic function:** The formula used to represent how the independent and dependent variables relate to one another. The logistic function transforms the input variables into a probability value between 0 and 1, which represents the likelihood of the dependent variable being 1 or 0.
- **Odds:** It is the **ratio of something occurring to something not occurring. It is different from probability as the probability is the ratio of something occurring to everything that could possibly occur**.

Logistics Regression

Terminologies involved in Logistic Regression

Calculate odds when p = 0.7.

Example 1: Calculating Odds from Probability

Suppose you have a probability of an event occurring, p, such as the probability of a basketball player making a free throw.

- **Probability of making the shot**: $p = 0.8$ (80%)

To calculate the odds of making the shot, use the formula:

$$\text{Odds} = \frac{p}{1-p}$$

Plugging in the values:

$$\text{Odds} = \frac{0.8}{1-0.8} = \frac{0.8}{0.2} = 4$$

This means the odds of the player making the shot are 4 to 1 (4:1).

Logistics Regression

Terminologies involved in Logistic Regression

- **Log-odds:** The log-odds, also known as the logit function, is the natural logarithm of the odds. In logistic regression, the log odds of the dependent variable are modeled as a linear combination of the independent variables and the intercept.
- **Coefficient:** The logistic regression model's estimated parameters, show how the independent and dependent variables relate to one another.
- **Intercept:** A constant term in the logistic regression model, which represents the log odds when all independent variables are equal to zero.
- **Maximum likelihood estimation:** The method used to estimate the coefficients of the logistic regression model, which maximizes the likelihood of observing the data given the model.

Logistics Regression

How does Logistic Regression work?

The logistic regression model transforms the **linear regression function** continuous value output into categorical value output using a sigmoid function, which maps any real-valued set of independent variables input into a value between 0 and 1.

- This function is known as **the logistic function**.

Let the independent input **X** and the dependent variable is **Y** having only binary value (i.e. 0 or 1), and the multi-linear function **z** be:

$$X = \begin{bmatrix} x_{11} & \cdots & x_{1m} \\ x_{21} & \cdots & x_{2m} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nm} \end{bmatrix} \qquad z = \left(\sum_{i=1}^{n} w_i x_i\right) + b \qquad Y = \begin{cases} 0 & \text{if } Class\ 1 \\ 1 & \text{if } Class\ 2 \end{cases}$$

Logistics Regression

How does Logistic Regression work?

Let the independent input **X** and the dependent variable is **Y** having only binary value (i.e. 0 or 1), and the multi-linear function **z** be:

Here x_i is the ith observation of X, $w_i = [w_1, w_2, w_3, \cdots, w_m]$ is the weights or Coefficient, and b is the bias term also known as intercept. simply this can be represented as the dot product of weight and bias.

$$z = w \cdot X + b$$

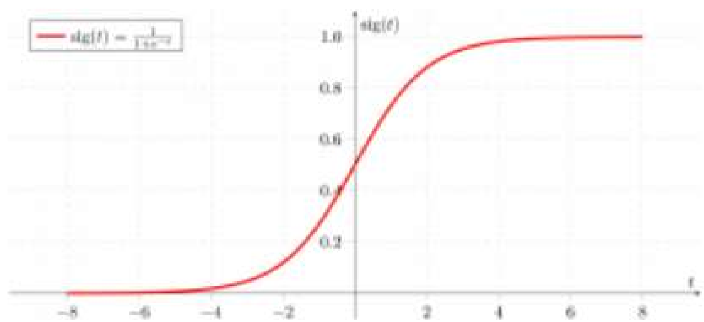
Logistics Regression

How does Logistic Regression work?

Apply the sigmoid function on the output of the multi-linear function **z**.

Now we use the sigmoid function where the input will be z and we find the probability between 0 and 1. i.e. predicted y.

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Sigmoid function

Logistics Regression

Logistic Regression Equation

- The odd is the ratio of something occurring to something not occurring.
- It is different from probability as the probability is the ratio of something occurring to everything that could possibly occur.

$$\frac{p(x)}{1-p(x)} = e^z$$

Applying natural log on odd. then log odd will be:

$$\log\left[\frac{p(x)}{1-p(x)}\right] = z$$

$$\log\left[\frac{p(x)}{1-p(x)}\right] = w \cdot X + b$$

$$\frac{p(x)}{1-p(x)} = e^{w \cdot X + b} \quad \cdots \text{Exponentiate both sides}$$

$$p(x) = e^{w \cdot X + b} \cdot (1 - p(x))$$

$$p(x) = e^{w \cdot X + b} - e^{w \cdot X + b} \cdot p(x))$$

$$p(x) + e^{w \cdot X + b} \cdot p(x)) = e^{w \cdot X + b}$$

$$p(x)(1 + e^{w \cdot X + b}) = e^{w \cdot X + b}$$

$$p(x) = \frac{e^{w \cdot X + b}}{1 + e^{w \cdot X + b}}$$

then the final logistic regression equation will be:

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

Logistics Regression

Likelihood Function for Logistic Regression

The predicted probabilities will be:

- for y=1 The predicted probabilities will be: p(X;b,w) = p(x)
- for y = 0 The predicted probabilities will be: 1-p(X;b,w) = 1-p(x)

$$L(b, w) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

Taking natural logs on both sides

$$\begin{aligned}
\log(L(b,w)) &= \sum_{i=1}^{n} y_i \log p(x_i) \ + \ (1-y_i)\log(1-p(x_i)) \\
&= \sum_{i=1}^{n} y_i \log p(x_i) + \log(1-p(x_i)) - y_i \log(1-p(x_i)) \\
&= \sum_{i=1}^{n} \log(1-p(x_i)) + \sum_{i=1}^{n} y_i \log \frac{p(x_i)}{1-p(x_i} \\
&= \sum_{i=1}^{n} -\log 1 - e^{-(w \cdot x_i + b)} + \sum_{i=1}^{n} y_i(w \cdot x_i + b) \\
&= \sum_{i=1}^{n} -\log 1 + e^{w \cdot x_i + b} + \sum_{i=1}^{n} y_i(w \cdot x_i + b)
\end{aligned}$$

Logistics Regression

Gradient of the log-likelihood function

To find the maximum likelihood estimates, we differentiate w.r.t w,

$$\frac{\partial J(l(b,w)}{\partial w_j} = -\sum_{i=n}^{n} \frac{1}{1+e^{w\cdot x_i+b}} e^{w\cdot x_i+b} x_{ij} + \sum_{i=1}^{n} y_i x_{ij}$$

$$= -\sum_{i=n}^{n} p(x_i; b, w) x_{ij} + \sum_{i=1}^{n} y_i x_{ij}$$

$$= \sum_{i=n}^{n} (y_i - p(x_i; b, w)) x_{ij}$$

Logistics Regression

Further examples:

- https://www.youtube.com/watch?v=C5268D9t9Ak&t=247s
- https://www.youtube.com/watch?v=yIYKR4sgzI8&t=357s

Softmax Regression

Softmax function

The softmax function is a mathematical function commonly used in machine learning, especially in the context of multi-class classification problems. It transforms a vector of real numbers into a probability distribution over multiple classes.

Given a vector $z = [z_1, z_2, \ldots, z_K]$, where K is the number of classes, the softmax function $\sigma(z)$ is defined as:

$$\sigma(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

for each element z_i in the vector. The key properties of the softmax function are:

1. **Output Range**: The output values are between 0 and 1.
2. **Sum to One**: The sum of all the output probabilities equals 1, making it suitable for interpreting the results as probabilities.

Softmax Regression

Softmax function

Example

Suppose we have a vector of scores (logits) from a model's output for three classes:

$$z = [2.0, 1.0, 0.1]$$

Step 1: Exponentiate Each Score

First, we apply the exponential function to each element:

$$\begin{aligned} e^{z_1} &= e^{2.0} \approx 7.39 \\ e^{z_2} &= e^{1.0} \approx 2.72 \\ e^{z_3} &= e^{0.1} \approx 1.11 \end{aligned}$$

So, the exponentiated scores are:

$$e^z \approx [7.39, 2.72, 1.11]$$

Softmax Regression

Softmax function

Step 2: Sum the Exponentiated Scores

Next, we calculate the sum of these exponentiated values:

$$\text{Sum} = 7.39 + 2.72 + 1.11 \approx 11.22$$

Step 3: Calculate the Softmax Probabilities

Now, we can compute the softmax probabilities for each class:

$$\sigma(z_1) = \frac{e^{z_1}}{\text{Sum}} = \frac{7.39}{11.22} \approx 0.659$$
$$\sigma(z_2) = \frac{e^{z_2}}{\text{Sum}} = \frac{2.72}{11.22} \approx 0.242$$
$$\sigma(z_3) = \frac{e^{z_3}}{\text{Sum}} = \frac{1.11}{11.22} \approx 0.099$$

Result

The resulting probabilities from the softmax function are:

$$\sigma(z) \approx [0.659, 0.242, 0.099]$$