# **Machine Learning**

## **What is Machine Learning?**

The use and development of computer systems that can learn and adapt without following explicit instructions, by using algorithms and statistical models to analyze and draw inferences from patterns in data.

## **When we use Machine Learning?**

* When you have a problem that requires many long lists of rules (e.g. if-else) to find the solution. In this case, machine-learning techniques can simplify your code and improve performance.
* Very complex problems for which there is no solution with a traditional approach.
* Non- stable environments’: machine-learning software can adapt to new data.

## **Data Types from A Machine Learning Perspective**

### **Numerical Data**

* Numerical data is any data where data points are exact numbers. Statisticians also might call numerical data, quantitative data.
* Numerical data can be characterized by **continuous** or **discrete** data. Continuous data can assume any value within a range whereas discrete data has distinct values.

Diagram

Description automatically generated

### **Categorical Data**

* Categorical data represents characteristics, such as a hockey player’s position, team, hometown. Categorical data can take numerical values. For example, maybe we would use **1** for red and **2** for blue. But these numbers don’t have a mathematical meaning. That is, we **can’t** add them together or take the average.
* In the context of super classification, categorical data would be the *class label*. This would also be something like if a person is a *man* or *woman*, or property is *residential* or *commercial*.

### **Time-Series Data**

* Time series data is a sequence of numbers collected at regular intervals over some period. It is very important, especially in particular fields like finance. Time series data has a temporal value attached to it, so this would be something like a date or a timestamp that you can look for trends in time.

Chart, line chart

Description automatically generated

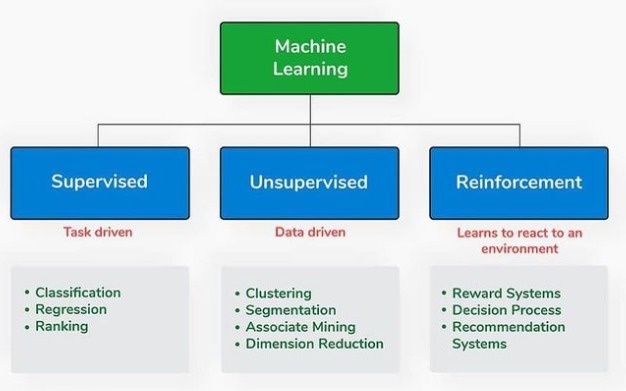
### **Text**

* Text data is basically just words. A lot of the time the first thing that you do with text is you turn it into numbers using some interesting functions like the bag of words formulation.

A picture containing text, accessory

Description automatically generated

## **Types of Systems of Machine Learning**



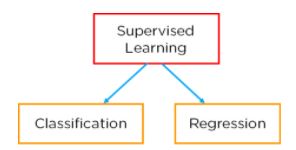
There are different types of machine-learning systems. We can divide them into

categories, depending on whether:

* They have been trained with humans or not
  + Supervised
  + Unsupervised
  + Semi-supervised
  + Reinforcement Learning
* If they can learn incrementally
* If they work simply by comparing new data points to find data points, or can detect new patterns in the data, and then will build a model.

## **Supervised & Unsupervised Learning**

### **Supervised Learning**



In this type of machine-learning system, the data that you feed into the algorithm, with the desired solution, are referred to as “**labels**.”

#### **Regression**

– relationship between dependent and independent variable and the output is continuous. (Predict a number)

Algorithms:

* Linear regression-
* Decision Tree-
* Random Forest-
* Neural Network-

#### **Classification**

Classification is defined as the process of recognition, understanding, and grouping of objects and ideas into preset categories a.k.a “sub-populations.” With the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future datasets into respective and relevant categories.

Classification algorithms used in machine learning utilize input training data for the purpose of predicting the likelihood or probability that the data that follows will fall into one of the predetermined categories. One of the most common applications of classification is for filtering emails into “spam” or “non-spam”, as used by today’s top email service providers.

In short, classification is a form of “*pattern recognition*,”. Here, classification algorithms applied to the training data find the same pattern (similar number sequences, words or sentiments, and the like) in future data sets.

We will explore classification algorithms in detail and discover how a text analysis software can perform actions like sentiment analysis - used for categorizing unstructured text by opinion polarity (positive, negative, neutral, and the like).

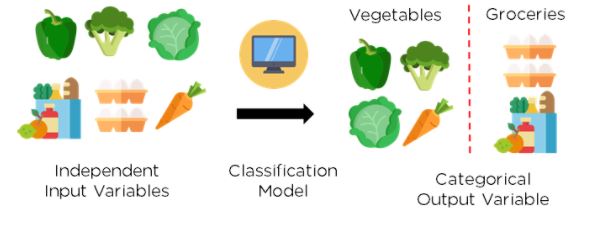


Figure 2: Classification of vegetables and groceries

##### **What is Classification Model/Algorithm?**

Based on training data, the Classification algorithm is a Supervised Learning technique used to categorize new observations. In classification, a program uses the dataset or observations provided to learn how to categorize new observations into various classes or groups. For instance, 0 or 1, red or blue, yes or no, spam or not spam, etc. Targets, labels, or categories can all be used to describe classes. The Classification algorithm uses labeled input data because it is a supervised learning technique and comprises input and output information. A discrete output function (y) is transferred to an input variable in the classification process (x).

In simple words, classification is a type of pattern recognition in which classification algorithms are performed on training data to discover the same pattern in new data sets.

Classification defined as the process of recognition, understanding, and grouping of objects and ideas into preset categories a.k.a “sub-populations.” With the help of these pre-categorized training datasets, classification in machine learning programs leverage a wide range of algorithms to classify future datasets into respective and relevant categories.

Classification Model/algorithm is a Supervised Learning technique used to categorize new observations. In classification, a program uses the dataset or observations provided to learn how to categorize new observations into various classes or groups. For instance, 0 or 1, red or blue, yes or no, spam or not spam, etc.

– the output is discrete. Predict a category

* + Logistics Regression – The output values can only be between 0 and 1
  + Support Vector Machine (SVM) –
  + Naïve Bayes –
  + Decision Tree -
  + Random Forest -
  + Neural Network -
  + K Nearest Neighbors –

##### **4 different types of Classification Tasks:**

* Binary Classification
* Multi-Class Classification
* Multi-Label Classification
* Imbalanced Classification

###### **Binary Classification**

Graphical user interface, application

Description automatically generated

Those classification jobs with only two class labels are referred to as binary classification.

Examples comprise -

* Prediction of conversion (buy or not).
* Churn forecast (churn or not).
* Detection of spam email (spam or not).

Binary classification problems often require two classes, one representing the normal state and the other representing the aberrant state.

For instance, the normal condition is "not spam," while the abnormal state is "spam." Another illustration is when a task involving a medical test has a normal condition of "cancer not identified" and an abnormal state of "cancer detected."

Class label 0 is given to the class in the normal state, whereas class label 1 is given to the class in the abnormal condition.

###### **Multi-Class Classification**

Application, table

Description automatically generated

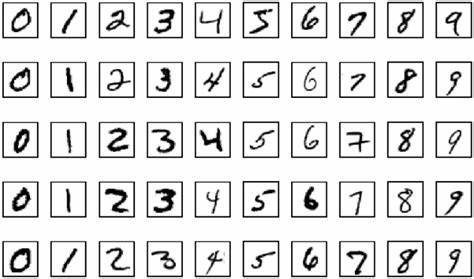
Multi-class labels are used in classification tasks referred to as multi-class classification.

Examples comprise -

* Categorization of faces.
* Classifying plant species.
* Character recognition using optical.

The multi-class classification does not have the idea of normal and abnormal outcomes, in contrast to binary classification. Instead, instances are grouped into one of several well-known classes.

In some cases, the number of class labels could be rather high. In a facial recognition system, for instance, a model might predict that a shot belongs to one of thousands or tens of thousands of faces.



Text translation models and other problems involving word prediction could be categorized as a particular case of multi-class classification. Each word in the sequence of words to be predicted requires a multi-class classification, where the vocabulary size determines the number of possible classes that may be predicted and may range from tens of thousands to hundreds of thousands of words.

###### **Multi-Label Classification**

Table

Description automatically generated

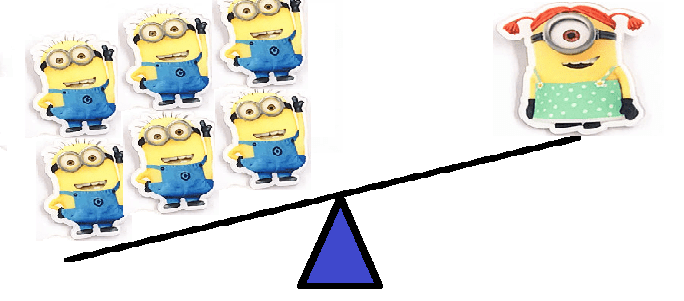
Multi-label classification problems are those that feature two or more class labels and allow for the prediction of one or more class labels for each example.

Think about the photo classification example. Here a model can predict the existence of many known things in a photo, such as “person”, “apple”, "bicycle," etc. A particular photo may have multiple objects in the scene.

This greatly contrasts with multi-class classification and binary classification, which anticipate a single class label for each occurrence.

#### **Imbalanced Classification**

Application

Description automatically generated

The term "imbalanced classification" describes classification jobs where the distribution of examples within each class is not equal.

Most of the training dataset's instances belong to the normal class, while a minority belongs to the abnormal class, making imbalanced classification tasks binary classification tasks in general.

Examples comprise -

* Clinical diagnostic procedures
* Detection of outliers
* Fraud investigation

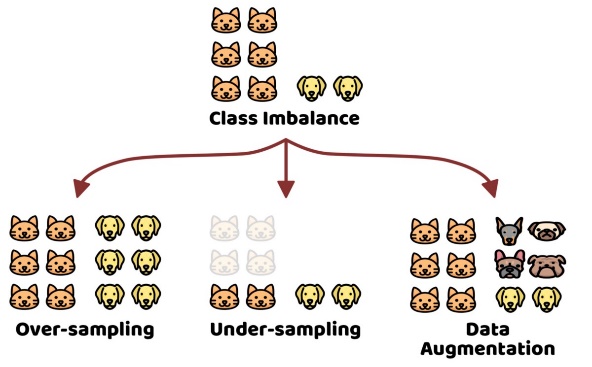
Although they could need unique methods, these issues are modeled as binary classification jobs.

By oversampling the minority class or undersampling the majority class, specialized strategies can be employed to alter the sample composition in the training dataset.

Examples comprise -

* SMOTE Oversampling
* Random Under sampling

It is possible to utilize specialized modeling techniques, like the cost-sensitive machine learning algorithms, that give the minority class more consideration when fitting the model to the training dataset.



### **Unsupervised Learning**

In this type of machine-learning system, you can guess that the data is unlabeled.

* Patterns from input data without references to labeled outcomes
  1. Clustering – Divide by similarity
     + K-means
     + Hierarchical
     + Mean shift
     + Density-based
  2. Association – Identify sequence.
  3. Dimensionality Reduction – a process of reducing the dimension of your feature set. Find hidden dependencies.
     + Feature elimination
     + Feature extraction

PRINCIPAL COMPONENT ANALYSIS (PCA) – popular method of Dimensionality Reduction

### **Note**

Supervised machine learning models make specific predictions or classifications based on labeled training data, while unsupervised machine learning models seek to cluster or otherwise find patterns in unlabeled data.

## **Things to consider in Machine Learning Development**

### **Bad and Insufficient Quantity of Training Data**

Machine-learning systems are not like children, who can distinguish apples and oranges in all sorts of colors and shapes, but they require lot of data to work effectively, whether you're working with very simple programs and problems, or complex applications like image processing and speech recognition.

### **Poor-Quality Data**

If you're working with training data that is full of errors and outliers, this will make it very hard for the system to detect patterns, so it won't work properly. So, if you want your program to work well, you must spend more time cleaning up your training data.

### **Irrelevant Features**

The system will only be able to learn if the training data contains enough features and data that aren’t too irrelevant. The most important part of any ML project is to develop good features “of feature engineering”.

#### **Feature Engineering**

The process of feature engineering goes like this:

* Selection of features: selecting the most useful features.
* Extraction of features: combining existing features to provide more useful features.
* Creation of new features: creation of new features, based on data.

### **Testing**

If you'd like to make sure that your model is working well and that model can generalize with new cases, you can try out new cases with it by putting the model in the environment and then monitoring how it will perform. This is a good method, but if your model is inadequate, the user will complain.

You should divide your data into two sets, one set for training and the second one for testing, so that you can train your model using the first one and test it using the second. The generalization error is the rate of error by evaluation of your model on the test set. The value you get will tell you if your model is good enough, and if it will work properly.

If the error rate is low, the model is good and will perform properly. In contrast, if your rate is high, this means your model will perform badly and not work properly. My advice to you is to use 80% of the data for training and 20% for testing purposes, so that it’s very simple to test or evaluate a model.

### **Overfitting the Data**

If you're in a foreign country and someone steals something of yours, you might say that everyone is a thief. This is an overgeneralization, and, in machine learning, is called “overfitting”. This means that machines do the same thing: they can perform well when they're working with the training data, but they can't generalize them properly.

When does this occur?

Overfitting occurs when the model is very complex for training data given.

#### Solutions

To solve the overfitting problem, you should do the following:

* Gather more data for “training data”
* Reduce the noise level
* Select one with fewer parameters

### **Underfitting the Data**

From its name, underfitting is the opposite of overfitting, and you'll encounter this when the model is very simple to learn. For example, using the example of quality of life, real life is more complex than your model, so the predictions won't yield the same, even in the training examples.

#### Solutions

To fix this problem:

* Select the most powerful model, which has many parameters.
* Feed the best features into your algorithm. Here, I'm referring to feature engineering.
* Reduce the constraints on your model.

## **Machine Learning lifecycle architecture diagram**


      Figure 4 includes the ML lifecycle from Figure 3 and expands its data processing phase 
        into sub-phases of collect data, and prepare data phases. Tje
        prepare data phase is further expanded into pre-process data, and engineer feature.
    

Figure 2: ML lifecycle with data processing sub-phases included

# **References**

<https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/machine-learning-lens.html>

<https://riskspan.com/evaluating-supervised-and-unsupervised-learning-models/#:~:text=Model%20evaluation%20%28including%20evaluating%20supervised%20and%20unsupervised%20learning,or%20appropriately%20flagging%20credit%20card%20transactions%20as%20fraud>.

<https://medium.com/analytics-vidhya/evaluation-criteria-for-machine-learning-models-ce04692cc3e3>

<https://www.datasciencecentral.com/profiles/blogs/9-python-analytics-libraries-1>

<https://towardsdatascience.com/a-feature-selection-tool-for-machine-learning-in-python-b64dd23710f0>

<https://www.simplilearn.com/tutorials/machine-learning-tutorial/classification-in-machine-learning>

<https://learnpython.com/blog/python-libraries-for-data-science/#top>

**Book:**

Step-by-Step Guide To Implement Machine Learning Algorithms with Python, Author: Rudolph Russell

Perry Xiao - Artificial Intelligence Programming with Python from Zero to Hero-John WIley (2022)