Data Science/Machine Learning/Deep Learning Workshop

Nueva Ecija University of Science and Technology (NEUST)

Rodolfo C. Raga Jr., PhDCS

December 21-23, 2021

Day 1 : Basic Concepts, Intuitions, and Foundations

Topic 1: Introduction to Data Science, Machine Learning, and Deep Learning

- What is Data Science
- What is Machine Learning
- What is Deep Learning

Topic 2: Intro to Python, Jupyter Notebook, Google Colab, Scikit-learn, Tensorflow, and Keras

- Why use Python
- Jupyter Notebook vs. Google Colab
- Understanding sklearn, Tensorflow and Keras

Topic 3: Fundamentals of Data Analysis using sklearn

- Correlation and Distribution Analysis
- Data Preprocessing (Feature Selection, Data Normalization, Data Splitting)
- Building and Training of ML prediction models using sklearn

Topic 4: Fundamentals of Neural Networks

- Definition and NN Architecture
- Perceptron and Multi-layer Perceptron Architecture
- Building NN prediction models using sklearn

Dr. Rodolfo C. Raga Jr.

NEUST Workshop – Dec 21-23, 2021

Outline

- Purpose of Correlation Analysis
- Purpose of Distribution Analysis
- Data Pre-processing
- Supervised Machine Learning
    - Nearest Neighbor Prediction model

Basic Terminologies

- **Dataset:** is a collection of data. Commonly corresponds to the contents of a single database table, or a single statistical data matrix.

- **Data**: The facts and figures collected, analyzed, and summarized for presentation and interpretation

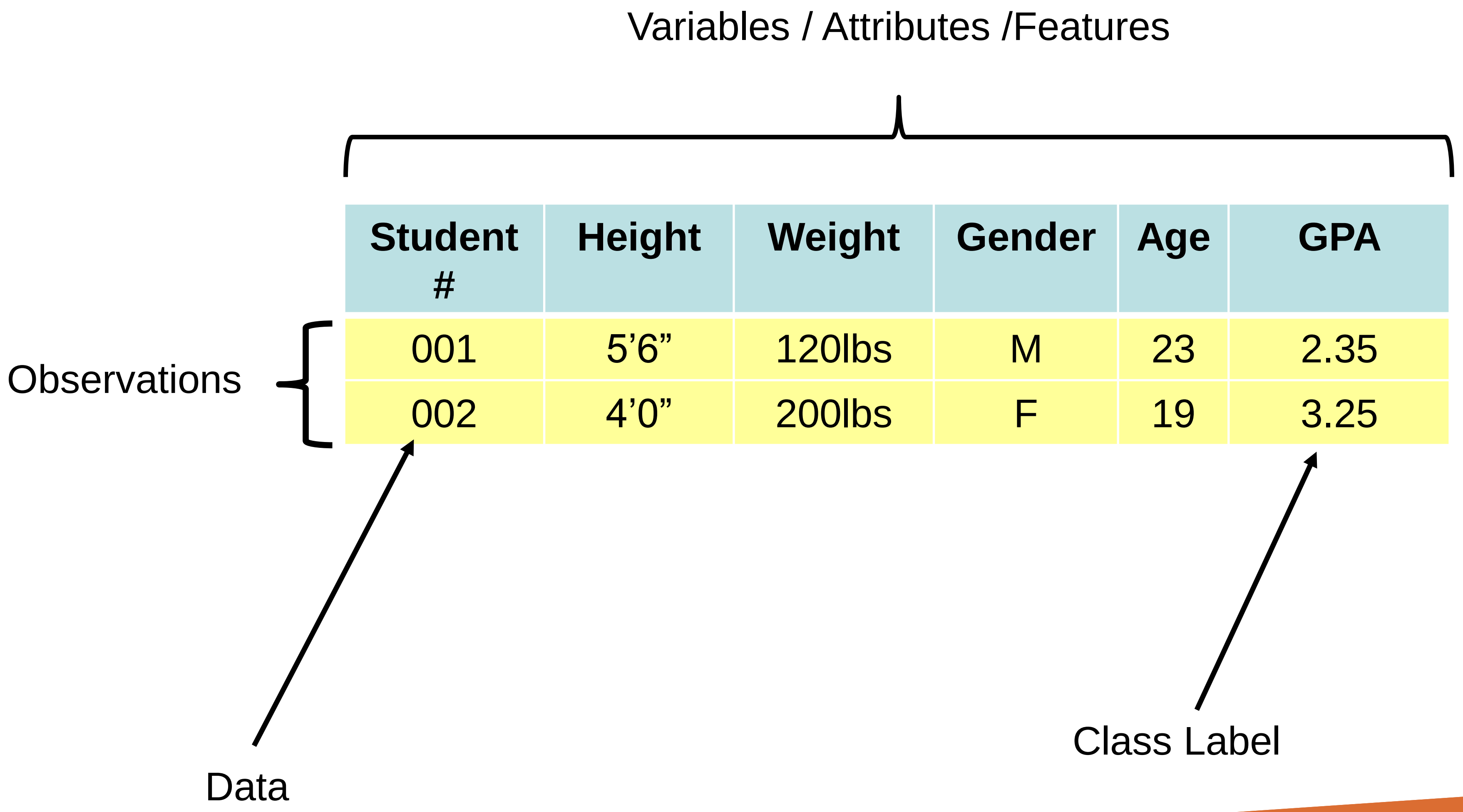
- **Variable**: A characteristic or a quantity of interest that can take on different values

- **Observation**: Set of values corresponding to a set of variables

- **Variation**: The difference in a variable measured over observations

- **Class label:** the discrete attribute having finite values (dependent variable) whose value you want to predict based on the values of other attributes(features)

A historical dataset



Variables / Attributes /Features

Student #	Height	Weight	Gender	Age	GPA
001	5’6”	120lbs	M	23	2.35
002	4’0”	200lbs	F	19	3.25

Observations

Data

Class Label

The five basic questions that Data Science can answer related to New Data

1. Is this A or B? (Classification)
2. Is this weird? (Anomaly Detection)
3. How much or how many? (Regression)
4. How is this organized? (Clustering)
5. What should I do next? (Reinforcement)

Data Quality Predictors

1. Are the data attributes Relevant?
2. Are the data attributes Connected?
3. Are the data attributes Accurate?
4. Is there enough Data?

Frequency Distributions

- After collecting data, the first task for a researcher is to organize and simplify the data so that it is possible to get a general overview of the results.
- This is the goal of descriptive statistical techniques.
- One method for simplifying and organizing data is to construct a **frequency distribution**.

Exploratory Data Analysis

- Exploratory Data Analysis (EDA) is the process of figuring out what the data can tell us.
- EDA can help us find patterns, relationships, or anomalies to inform our subsequent analysis.
- While there are an almost overwhelming number of methods to use in EDA, two of the most common is correlation and distribution analysis.

Correlation

- The term "correlation" refers to a mutual relationship or association between quantities of variables in a dataset.
- It is concerned with strength of the relationship and does not indicate causal effect.
- Correlation is considered as a useful metric:
    - It is used as a basic quantity and foundation for many other modeling techniques
    - It can help in predicting one quantity from another

Correlation

- Variables within a dataset can be correlated for several reasons
  - One variable could cause or depend on the values of another variable.
  - One variable could be lightly associated with another variable.
  - Two variables could depend on a third unknown variable.

Correlation

- A correlation could be positive, negative or neutral.
  - Positive Correlation: both variables change in the same direction.
  - Neutral Correlation: No relationship in the change of the variables.
  - Negative Correlation: variables change in opposite directions.

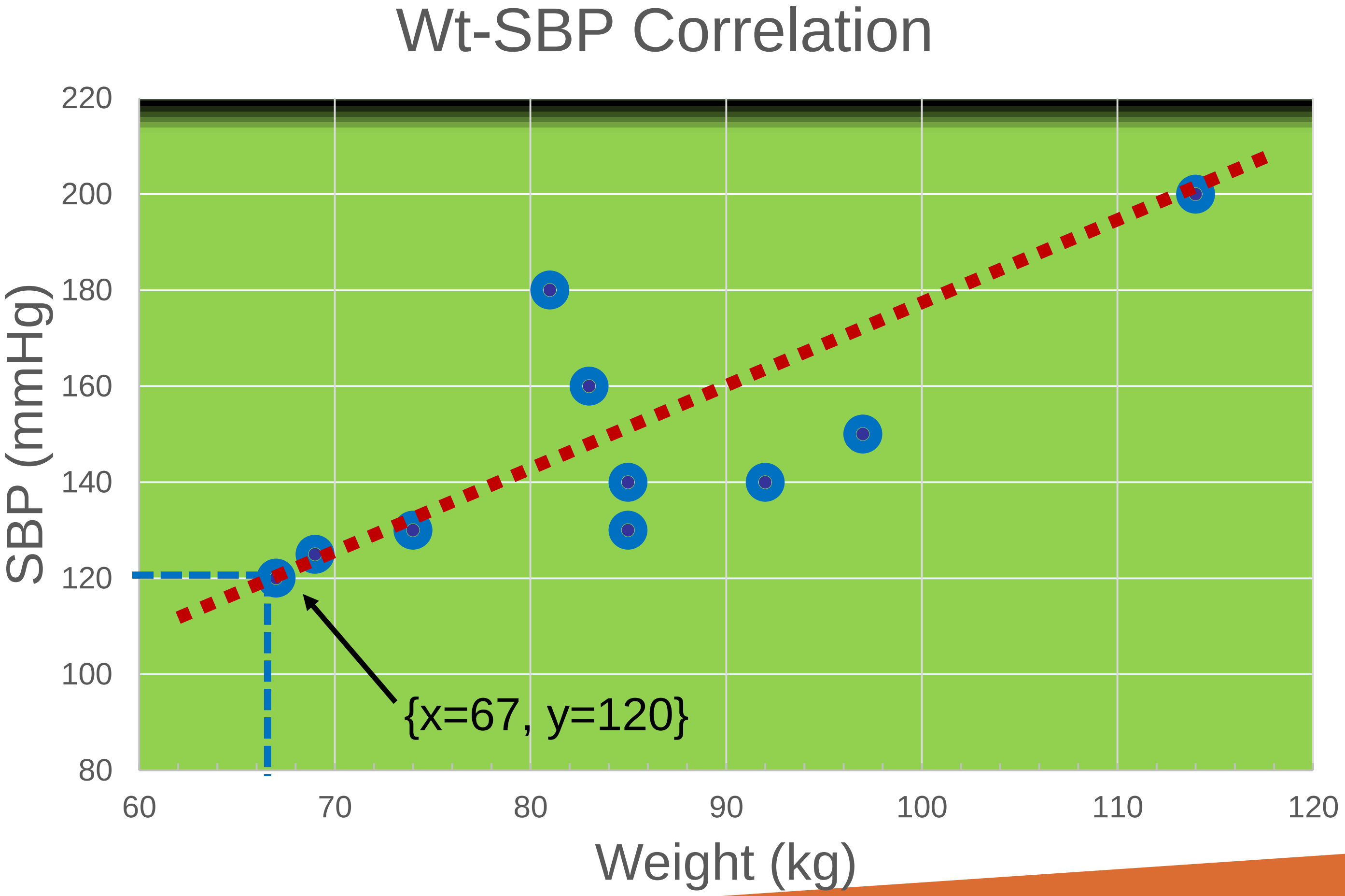
Scatter Plots and Correlation

- It is always a good idea to use visualization techniques to get a better picture of how variables relate to each other.
- A scatter plot (or scatter diagram) is the most often used diagram to graphically depict the relationship between two variables
  - It uses cartesian coordinate
  - Represents two quantitative variables
  - One variable is called independent (X) and the second is called dependent (Y)

Correlation Example

Scatter plot of weight and systolic blood pressure

Wt. (kg)	SBP (mmHg)
67	120
69	125
85	140
83	160
74	130
81	180
97	150
92	140
114	200
85	130

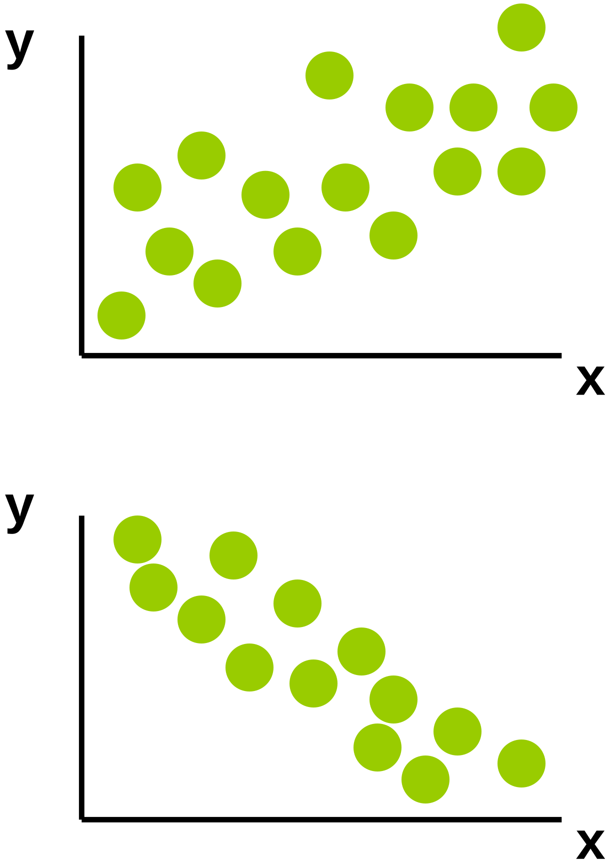


Dr. Rodolfo C. Raga Jr.

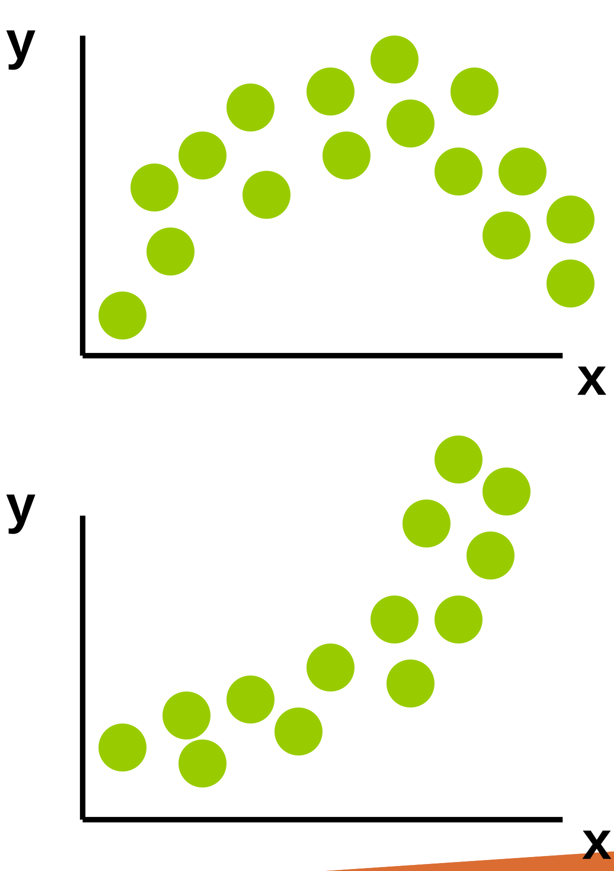
NEUST Workshop – Dec 21-23, 2021

Scatter Plot Examples

Linear relationships

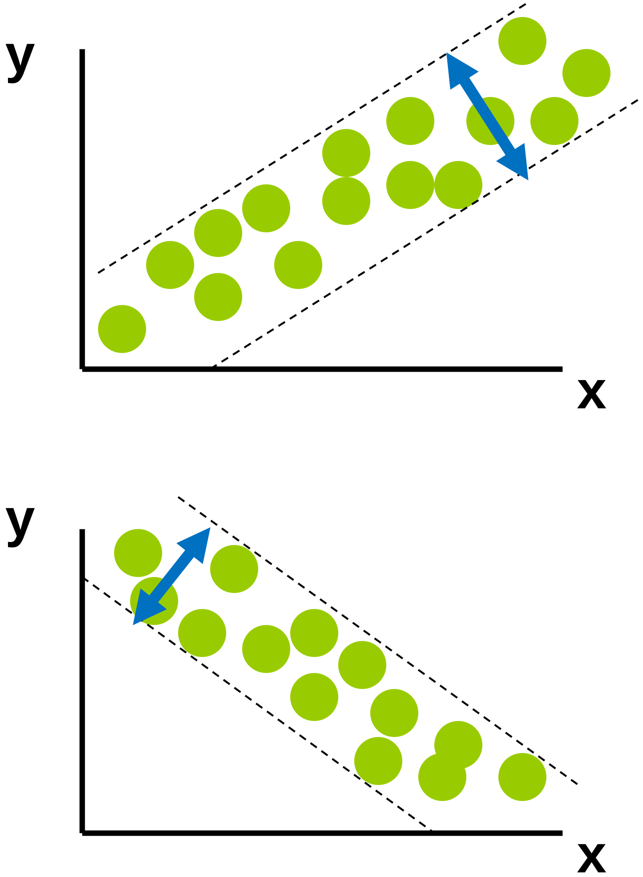


Curvilinear relationships

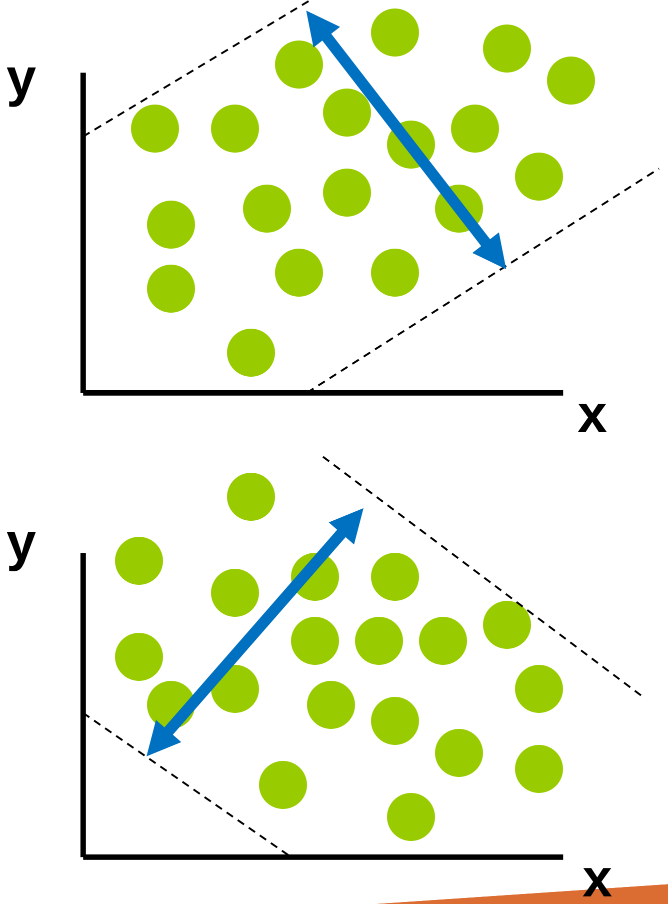


Scatter Plot Examples

Strong relationships

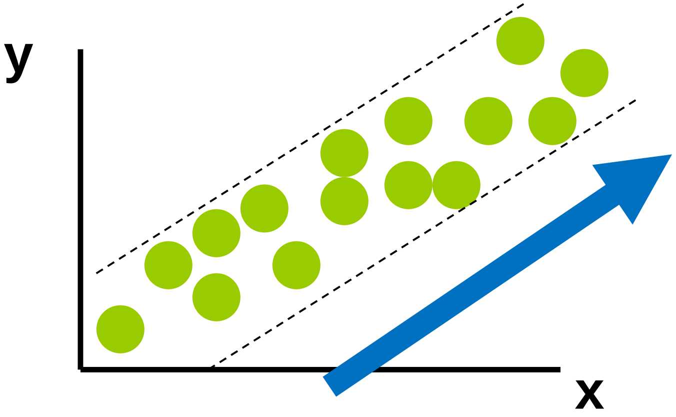


Weak relationships

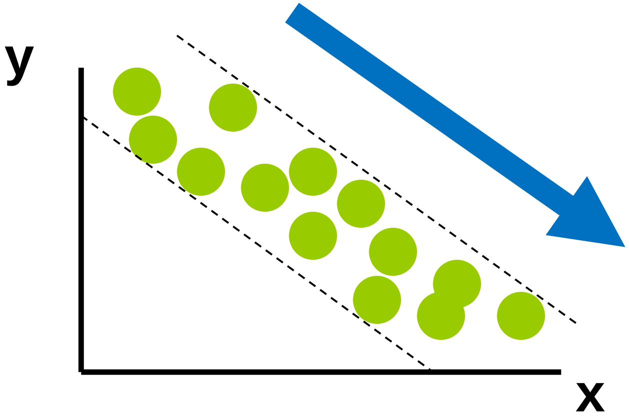


Scatter Plot Examples

Positive relationships

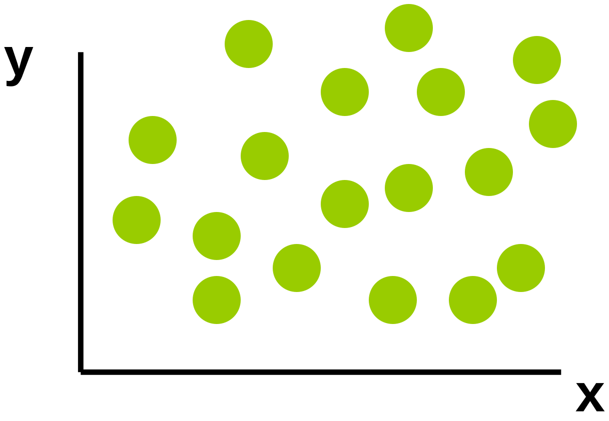


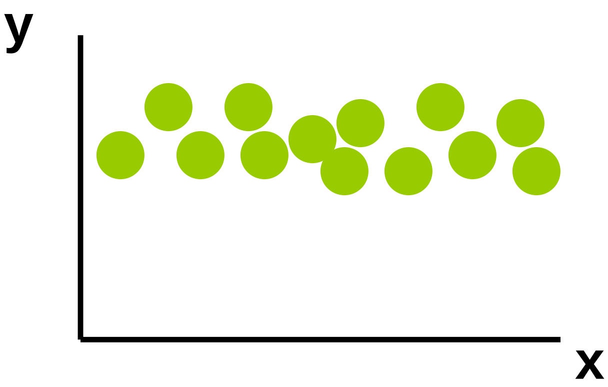
Negative relationships



Scatter Plot Examples

No relationship



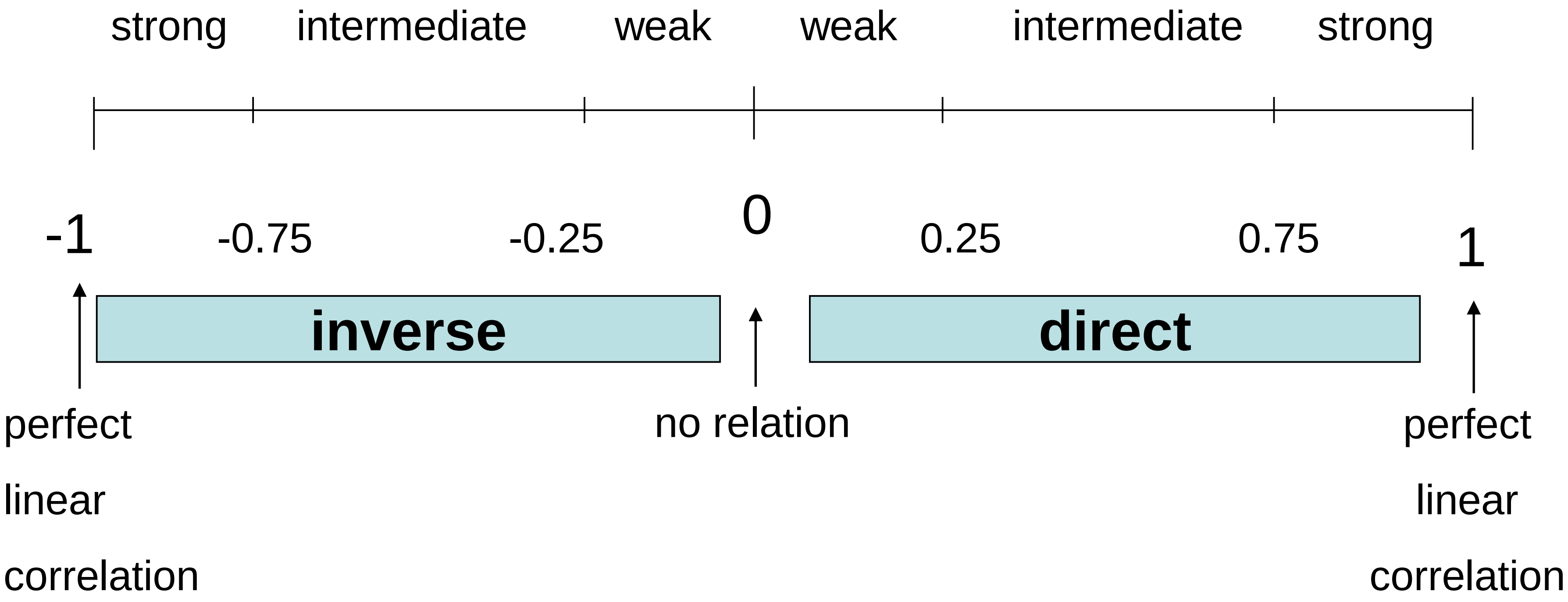


Correlation Coefficient r

- The sample correlation coefficient r is used to measure the strength of the linear relationship in a given sample observations
- This coefficient has values between -1 to 1
  - A value closer to 0 implies weaker correlation (exact 0 implying no correlation)
  - A value closer to 1 implies stronger positive correlation
  - A value closer to -1 implies stronger negative correlation

Interpreting values of r

- The value of r denotes the strength of the association as illustrated by the following diagram.



strong intermediate weak weak intermediate strong

-1 -0.75 -0.25 0 0.25 0.75 1

inverse

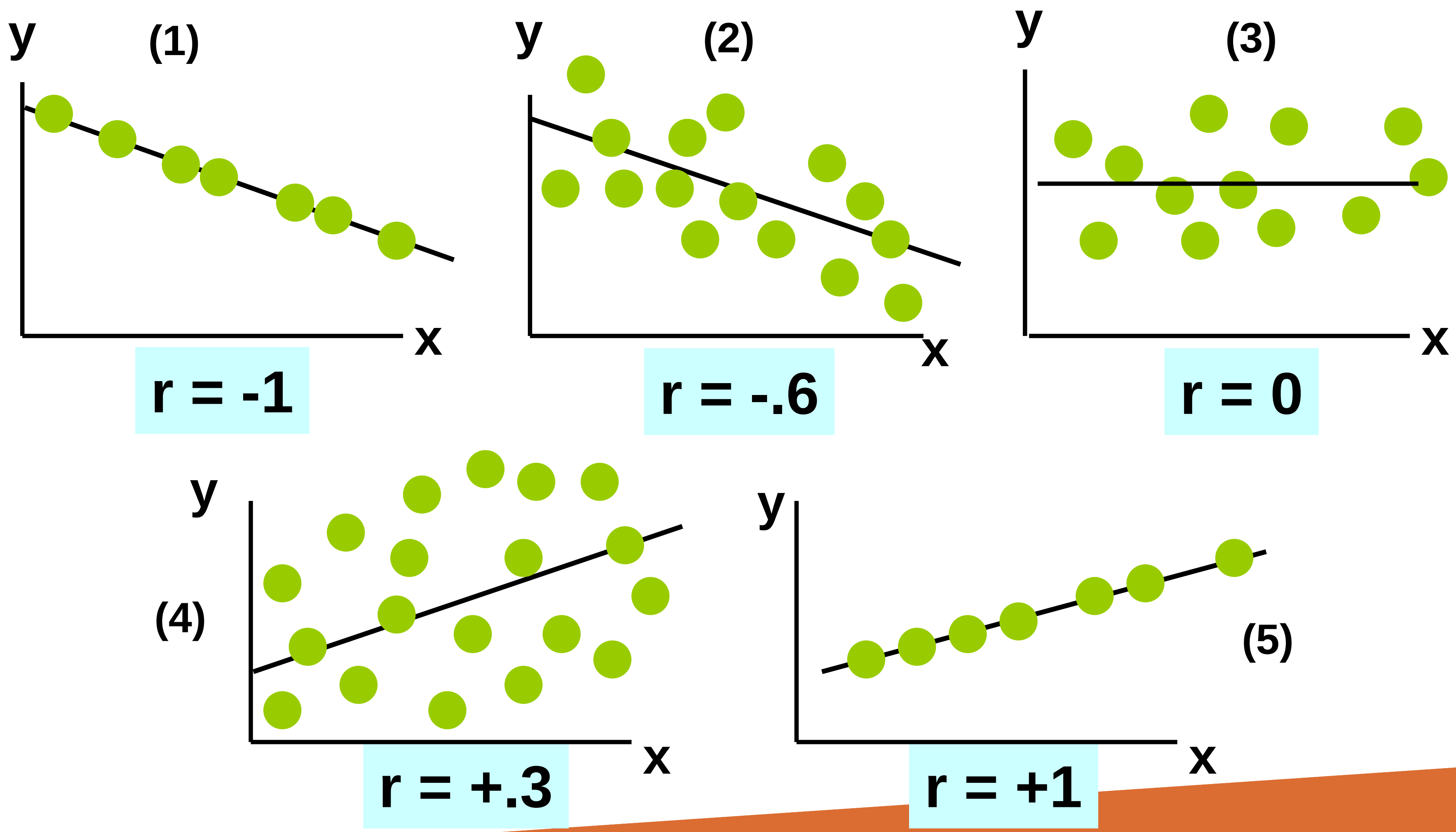
direct

perfect linear correlation

no relation

perfect linear correlation

Scatterplot visualizations of r values



Purpose of Correlation Analysis in ML

- Feature selection is one of the first and important steps while performing any machine learning task.
- Not necessarily every column (attribute) in a dataset will have an impact on the output variable.
- If we use these irrelevant features as predictors, it will just make the prediction model worst (Garbage In Garbage Out).

Frequency Distribution

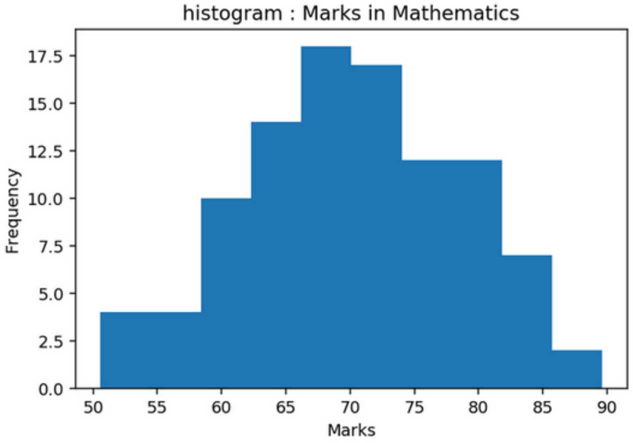
- A frequency distribution is an organized tabulation showing exactly how many individuals are located in each category on the scale of measurement.
- A frequency distribution presents an organized picture of the entire set of scores, and it shows where each individual is located relative to others in the distribution.

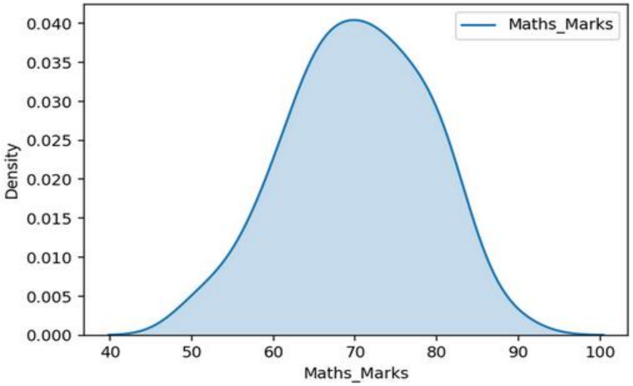
Frequency Distribution Graphs

- In a frequency distribution graph, the score categories (X values) are listed on the X axis and the frequencies are listed on the Y axis.
- When the score categories consist of numerical scores from an interval or ratio scale, the graph should be either a histogram or a polygon.

Histograms and Density Plots

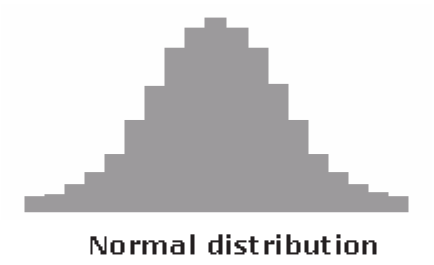
- A Histogram visualizes the distribution of data over a continuous interval
- Each bar in a histogram represents the tabulated frequency at each interval/bin
- The height of each bar represents the frequency for the respective bin (interval)
- A density plot is a smoothed, continuous version of a histogram estimated from the data.
- In this method, a continuous curve (the kernel) is drawn at every individual data point and all of these curves are then added together to make a single smooth density estimation.

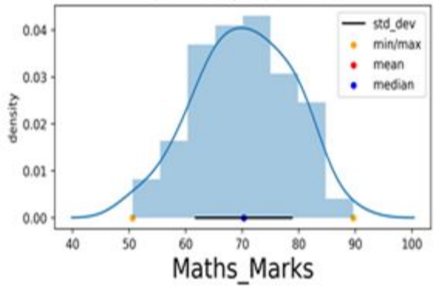




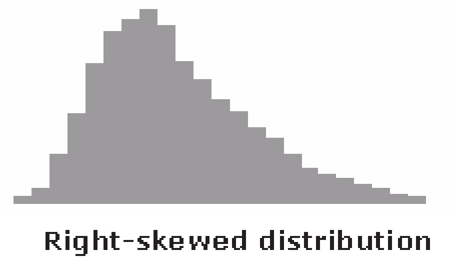
TYPICAL DISTRBUTION SHAPES

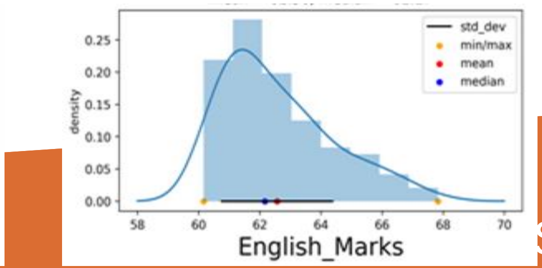
Normal Distribution: In a normal distribution, data is symmetrically distributed with no skew. When plotted on a graph, the data follows a bell shape, with most values clustering around a central region and tapering off as they go further away from the center. This indicates that points are as likely to occur on one side of the average as on the other.

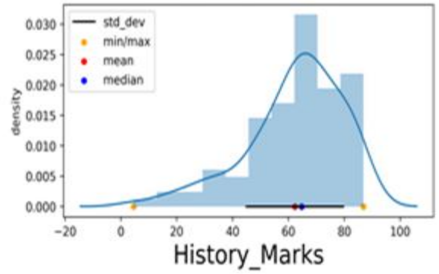

Normal distribution



Skewed Distribution: These distributions are sometimes called asymmetric or asymmetrical distributions as they don't show any kind of symmetry. Symmetry means that one half of the distribution is a mirror image of the other half. The distribution’s peak is off center toward the limit and a tail stretches away from it. These distributions are called right- or left-skewed according to the direction of the tail.

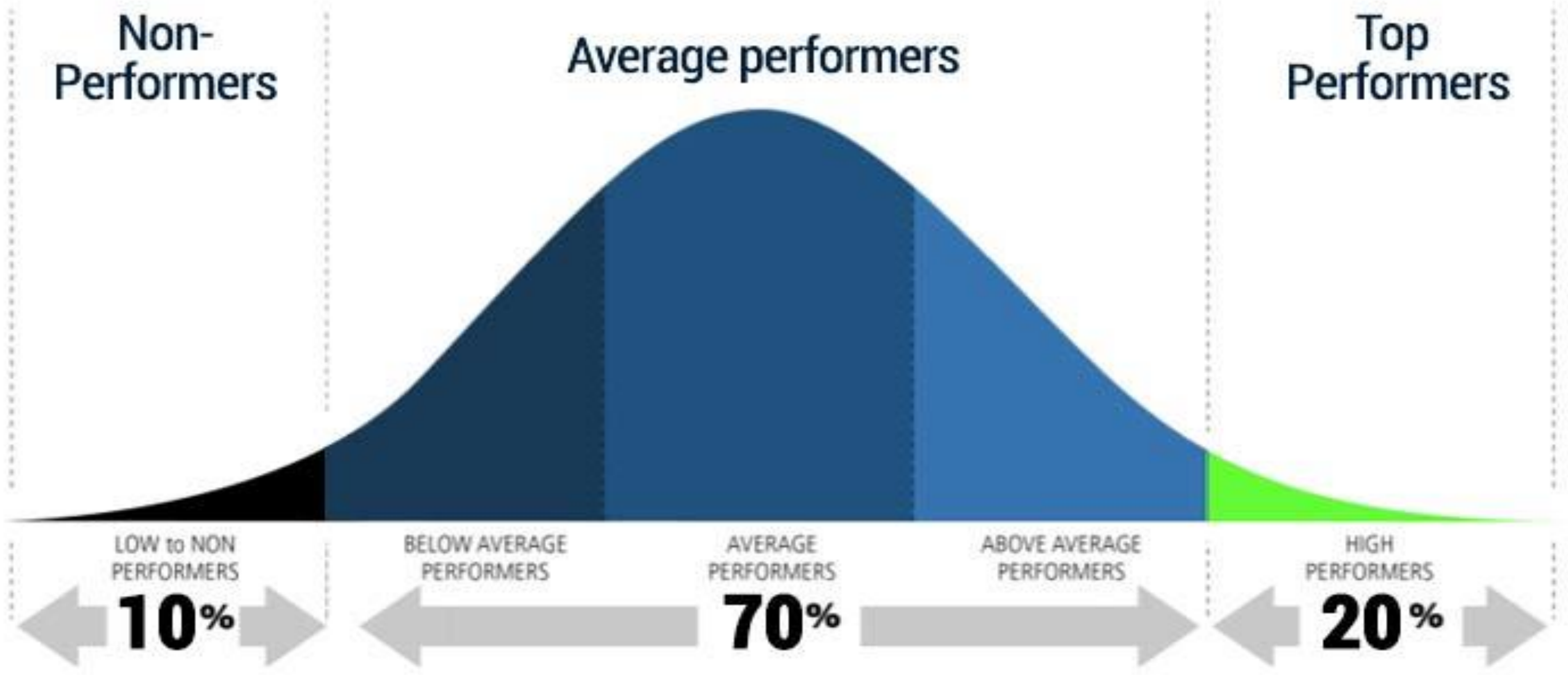

Right-skewed distribution





The Normal Distribution

The normal distribution is a core concept in statistics and the backbone of data science. In exploratory data analysis, it is the most common distribution used to explore data



Purpose of Distribution Analysis in ML

- Data satisfying Normal Distribution is beneficial for building ML models.
- Models like LDA, Gaussian Naive Bayes, Logistic Regression, Linear Regression, etc., are explicitly calculated from the assumption that the distribution is normal. Also, Sigmoid functions work most naturally with normally distributed data.
- So it’s better to critically explore the data and check for the underlying distributions for each variable before going to fit the model.

Purpose of Distribution Analysis in ML (2)

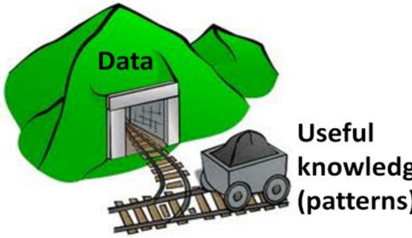
- Normality is an assumption for the ML models. It is not mandatory that data should always follow normality.
- ML models work very well in the case of non-normally distributed data also. Models like decision tree, XgBoost, don’t assume any normality and work on raw data as well.
- Linear regression is statistically effective if only the model errors are Gaussian, not exactly the entire dataset.

Data Pre-processing

- Machine Learning algorithms don’t work well with **raw data**. Before feeding such data to an ML algorithm, it must be **preprocessed**.

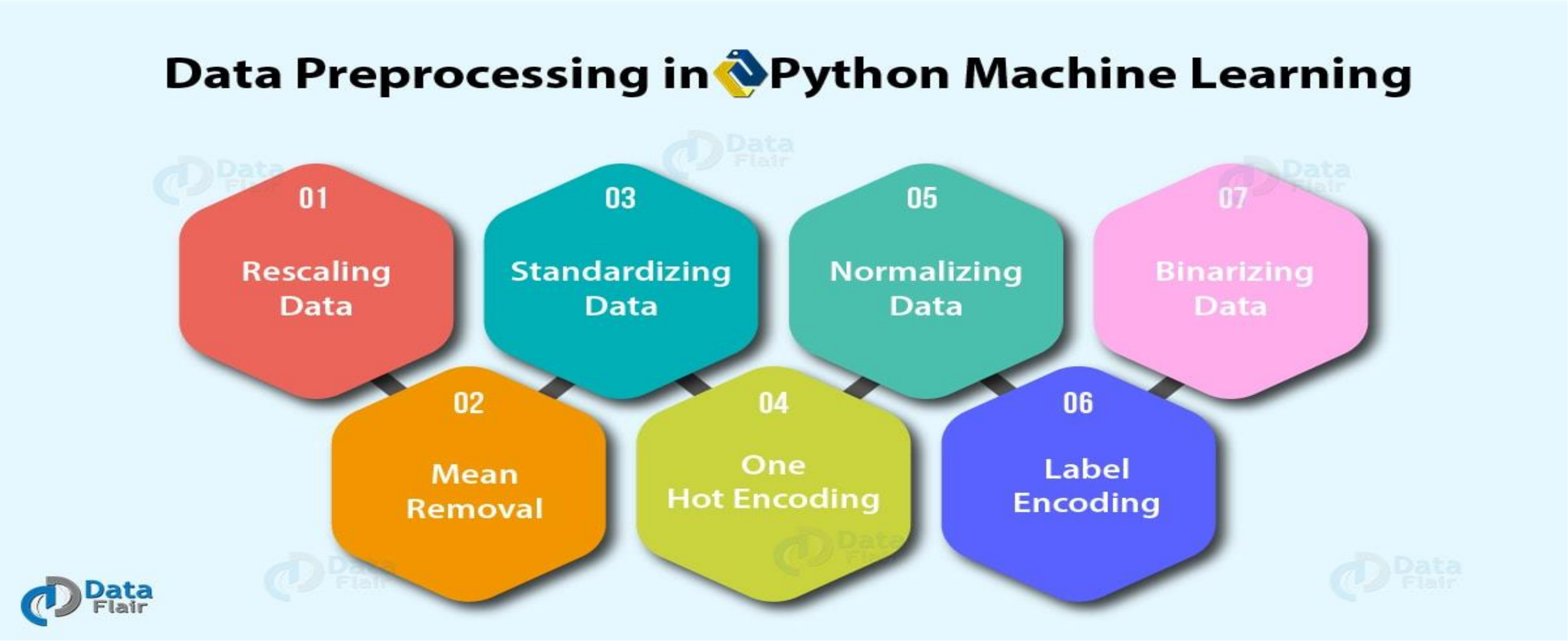
- **Pre-processing** refers to the transformations applied to raw data to transform it into clean data.

- **Data Preprocessing** requires application of techniques that converts the raw data into a clean data set.



- **Need for Data Preprocessing**
  - It can improve model performance
  - Many Machine Learning model require data input in a specific format.
  - Data set should be formatted in such a way that more than one Machine Learning and/or Deep Learning algorithms can work on them.

Major preprocessing techniques



Data Pre-Processing techniques

- Rescale data
- Standardize data
- Binarize data
- Normalization

1. Rescaling Data

- Transforms a dataset so that its data features are rescaled to values between 0 and 1.
- Useful when data is comprised of attributes with varying scales
- Useful when using algorithms that weight inputs like regression and neural networks and with algorithms that use distance measures like K-Nearest Neighbors.
- Also improves the performance of optimization algorithms.

Rescaling Datasets using MinMaxScaler

- To rescale datasets, we first declare the MinMaxScaler library

```
from sklearn.preprocessing import MinMaxScaler
```

- Syntax:

```
MinMaxScaler(feature_range=(0, 1), copy=True)
```

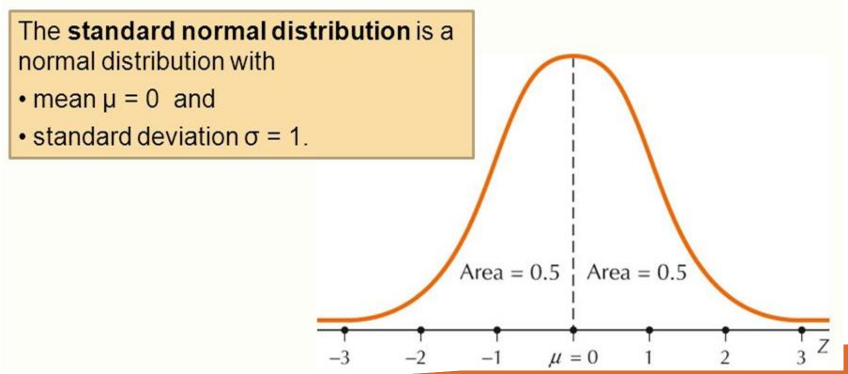
- Sample Use:

```
scaler = MinMaxScaler(feature_range=(0, 1))
rescaledX = scaler.fit_transform(X)
```

2. Standardizing Data

- It transforms the values in the dataset and shifts it so that the original mean value is placed at 0 and the standard deviation is 1.

- Gives data the property of a standard normal distribution (also known as Gaussian distribution).



The **standard normal distribution** is a normal distribution with

- mean μ = 0 and
- standard deviation σ = 1.

Standardizing Datasets using scale

- To standardize datasets, we first declare the scale library

  from sklearn.preprocessing import scale

- Syntax:

  scale(data_to_scale)

- Sample Use:

  rescaledX2 = scale(X)

3. Binarize Data

- **Binarization** is the process of tresholding numerical features to get boolean values. Or in other words, assign a boolean value (True or False) to each sample based on a threshold.
- This is useful is useful as a feature engineering technique for creating new features that indicate something meaningful.
- The Binarizer class in sklearn implements binarization in a very intuitive way. The only parameters you need to specify are the threshold and copy. All values below or equal to the threshold are replaced by 0, above it by 1.

3. Binarizing Datasets using Binarizer

- The Binarizer class in sklearn implements binarization in a very intuitive way. The only parameters you need to specify are the threshold and copy. All values below or equal to the threshold are replaced by 0, above it by 1.
- Example:

```
from sklearn.preprocessing import Binarizer
binarizer = Binarizer(threshold=0, copy=True)
binarizer.fit_transform(X.f3.values.reshape(-1, 1))
```

Machine Learning

Building and Training of ML prediction models using sklearn

Dr. Rodolfo C. Raga Jr.

NEUST Workshop – Dec 21-23, 2021

Machine learning tasks

- Supervised learning
  - **Input**: training data + desired outputs (labels)
  - regression: predict numerical values
  - classification: predict categorical values, i.e., labels
- Unsupervised learning
  - **Input**: training data (without desired outputs)
  - clustering: group data according to "distance"
  - association: find frequent co-occurrences
  - link prediction: discover relationships in data
  - data reduction: project features to fewer features

Supervised Learning

- The aim of **supervised learning** is to build a model that is 'good at' predicting the target variable, given the predictor variables.
- If the target is a continuously varying variable (e.g. price of a house), it is a **regression** task.
- Alternatively, if the target variable consists of categories (e.g. 'click' or 'not', 'malignant' or 'benign' tumor), we call the learning task **classification**.

Regression Algorithms

- Simple Linear Regression
- Multiple Linear Regression
- Support Vector Machines
- Perceptron

Classification

- Logistic Regression
- Support Vector Machines
- Deep Neural Networks

Supervised Machine Learning

Learn to predict **target values** from labelled data.

- Classification (target values are discreet classes)
- Regression (target values are continuous values)

Supervised ML Classification

Uses Labelled Data for Answering Questions

Training Component

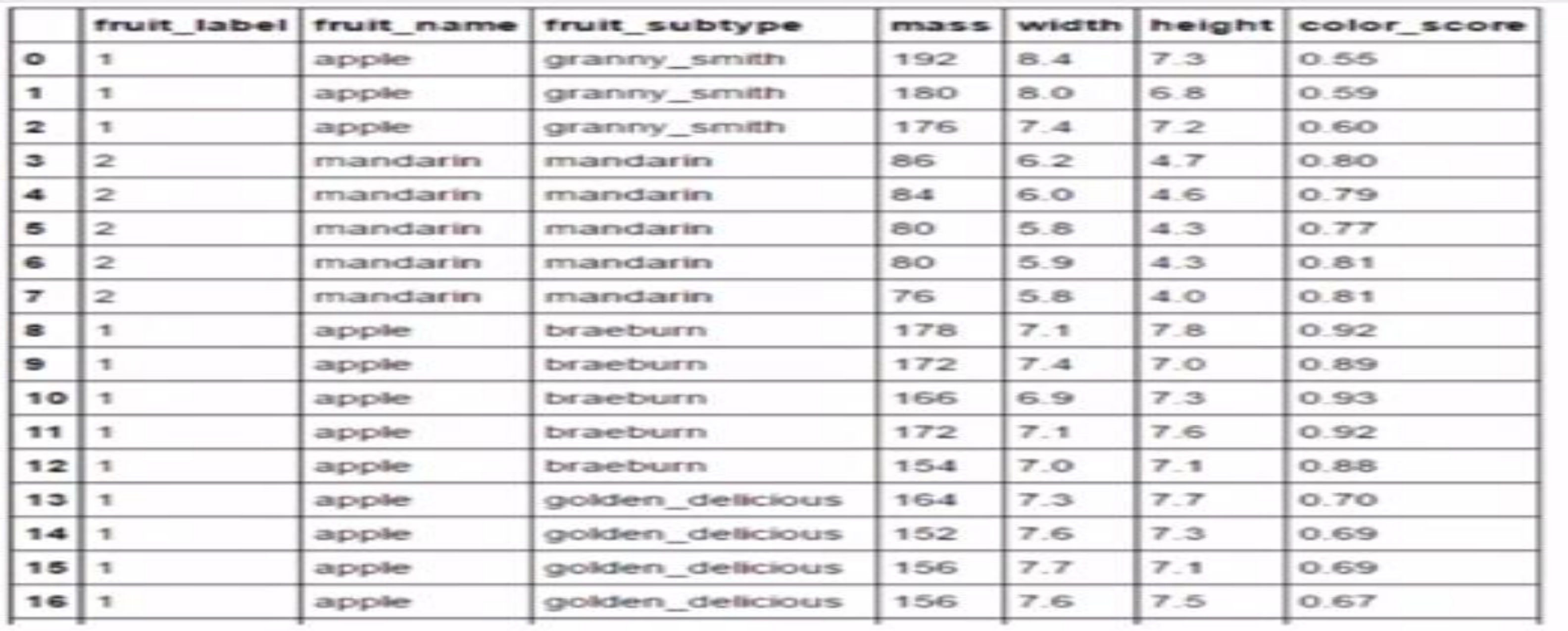
Prediction Component

Example of Questions that can be answered include:

- Is this spam email or not (for text data)
- Is this a dog or a cat (for image data)
- Is the speaker a man or a woman (for audio data)

What is labelled data?

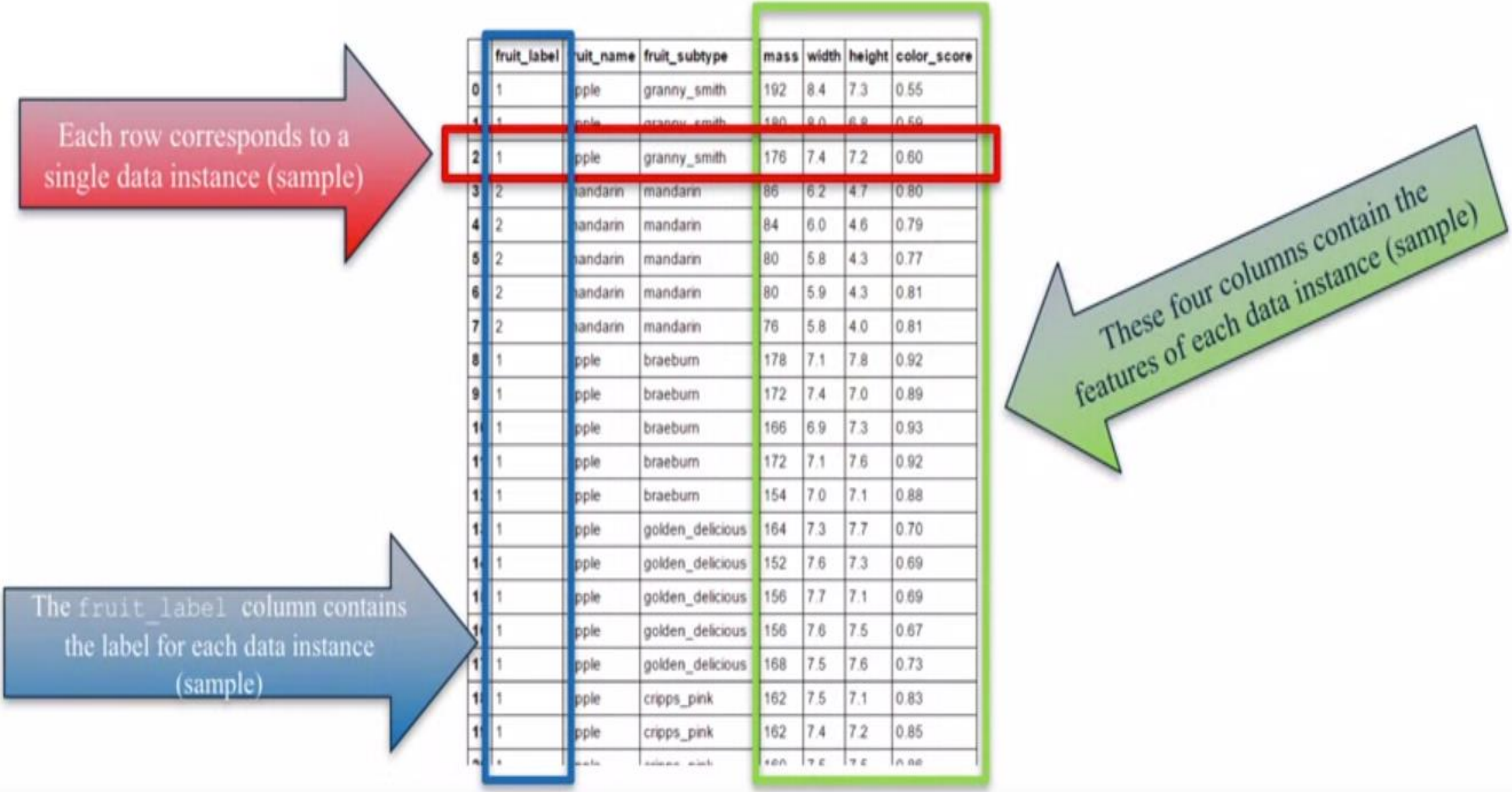
- To apply supervised machine learning to answer a particular problem we need to provide it with training data.
- The training data comes in the form of a table with columns both for the feature values and the target values.
- Feature values describe specific instances of data objects, e.g. to describe specific fruits we can use measurements of its mass, its width and height, etc.
- Target values represent the actual label which can describe the actual group or category of each data object.



	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192	8.4	7.3	0.55
1	1	apple	granny_smith	180	8.0	6.8	0.59
2	1	apple	granny_smith	176	7.4	7.2	0.60
3	2	mandarin	mandarin	86	6.2	4.7	0.80
4	2	mandarin	mandarin	84	6.0	4.6	0.79
5	2	mandarin	mandarin	80	5.8	4.3	0.77
6	2	mandarin	mandarin	80	5.9	4.3	0.81
7	2	mandarin	mandarin	76	5.8	4.0	0.81
8	1	apple	braeburn	178	7.1	7.8	0.92
9	1	apple	braeburn	172	7.4	7.0	0.89
10	1	apple	braeburn	166	6.9	7.3	0.93
11	1	apple	braeburn	172	7.1	7.6	0.92
12	1	apple	braeburn	154	7.0	7.1	0.88
13	1	apple	golden_delicious	164	7.3	7.7	0.70
14	1	apple	golden_delicious	152	7.6	7.3	0.69
15	1	apple	golden_delicious	156	7.7	7.1	0.69
16	1	apple	golden_delicious	156	7.6	7.5	0.67

What is labelled data?

Feature values are also referred to as independent attributes or variables.
Target values are also referred to as dependent attributes or variables



Each row corresponds to a single data instance (sample)

The fruit_label column contains the label for each data instance (sample)

These four columns contain the features of each data instance (sample)

	fruit_label	uit_name	fruit_subtype	mass	width	height	color_score
0	1	pple	granny_smith	192	8.4	7.3	0.55
1	1	pple	granny_smith	180	8.0	6.8	0.59
2	1	pple	granny_smith	176	7.4	7.2	0.60
3	2	andarin	mandarin	86	6.2	4.7	0.80
4	2	andarin	mandarin	84	6.0	4.6	0.79
5	2	andarin	mandarin	80	5.8	4.3	0.77
6	2	andarin	mandarin	80	5.9	4.3	0.81
7	2	andarin	mandarin	76	5.8	4.0	0.81
8	1	pple	braeburn	178	7.1	7.8	0.92
9	1	pple	braeburn	172	7.4	7.0	0.89
10	1	pple	braeburn	166	6.9	7.3	0.93
11	1	pple	braeburn	172	7.1	7.6	0.92
12	1	pple	braeburn	154	7.0	7.1	0.88
13	1	pple	golden_delicious	164	7.3	7.7	0.70
14	1	pple	golden_delicious	152	7.6	7.3	0.69
15	1	pple	golden_delicious	156	7.7	7.1	0.69
16	1	pple	golden_delicious	156	7.6	7.5	0.67
17	1	pple	golden_delicious	168	7.5	7.6	0.73
18	1	pple	cripps_pink	162	7.5	7.1	0.83
19	1	pple	cripps_pink	162	7.4	7.2	0.85
20	1	pple	cripps_pink	160	7.5	7.5	0.86

What is labelled data?

Labelled data refers to a dataset that contains both feature and target values. Making it possible to apply Machine Learning to this data.

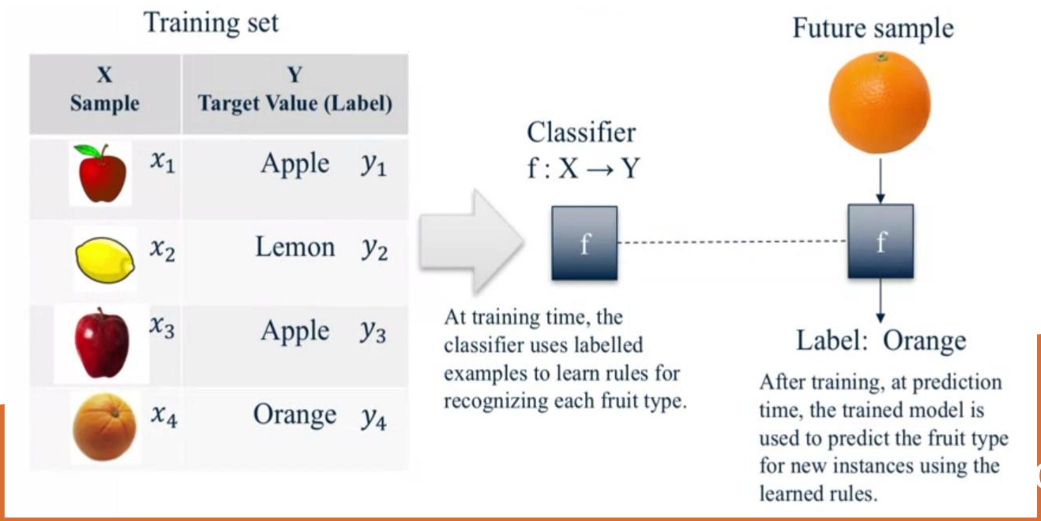
- Feature values are also referred to as independent attributes or variables. Refers to the data that describe the properties and characteristics of the data object. These are values that serve as input to the learning algorithm.
- Target values are also referred to as dependent attributes or variables. It refers to the data that serves as label indicating the status or class of the data object. Within the prediction process, target values serve as values that the prediction model try to predict for each new features that it encounters.

How the machine learns from the labelled data?

Step 1: The computer observes and analyzes the patterns it detects from all the feature values of every data instances in the dataset.

Step 2: It maps those patterns to the corresponding labels of the data instances so it can detect common features/patterns present in every type of data instance.

Step 3: It remembers the mapped patterns between the Features and the labels so it can use this to predict the labels of future data instances.



Training set

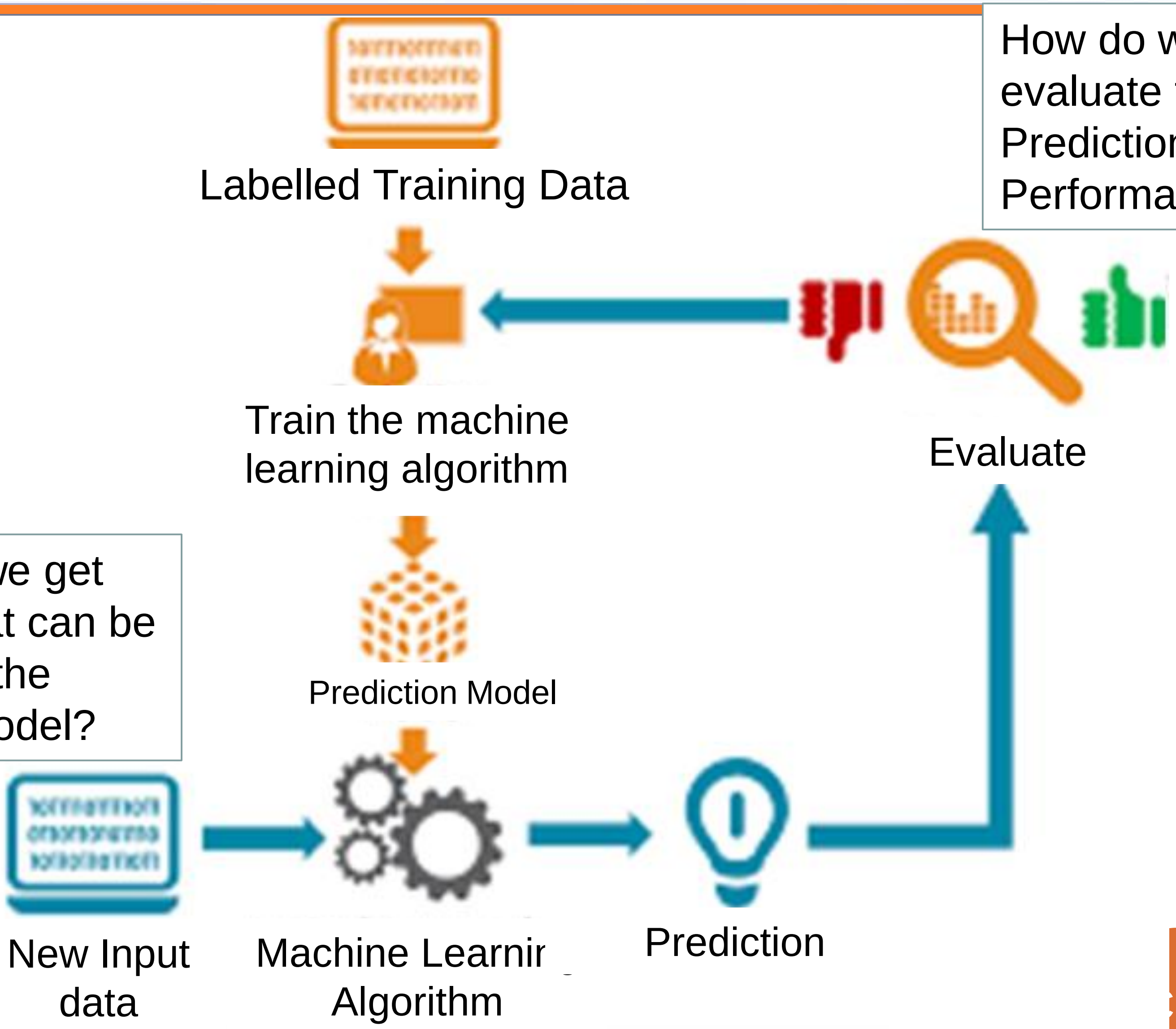
X Sample	Y Target Value (Label)
x_1	Apple y_1
x_2	Lemon y_2
x_3	Apple y_3
x_4	Orange y_4

At training time, the classifier uses labelled examples to learn rules for recognizing each fruit type.

After training, at prediction time, the trained model is used to predict the fruit type for new instances using the learned rules.

The Supervised Training Process

How do we evaluate the Prediction Performance?



Labelled Training Data

Train the machine learning algorithm

Evaluate

Prediction Model

New Input data

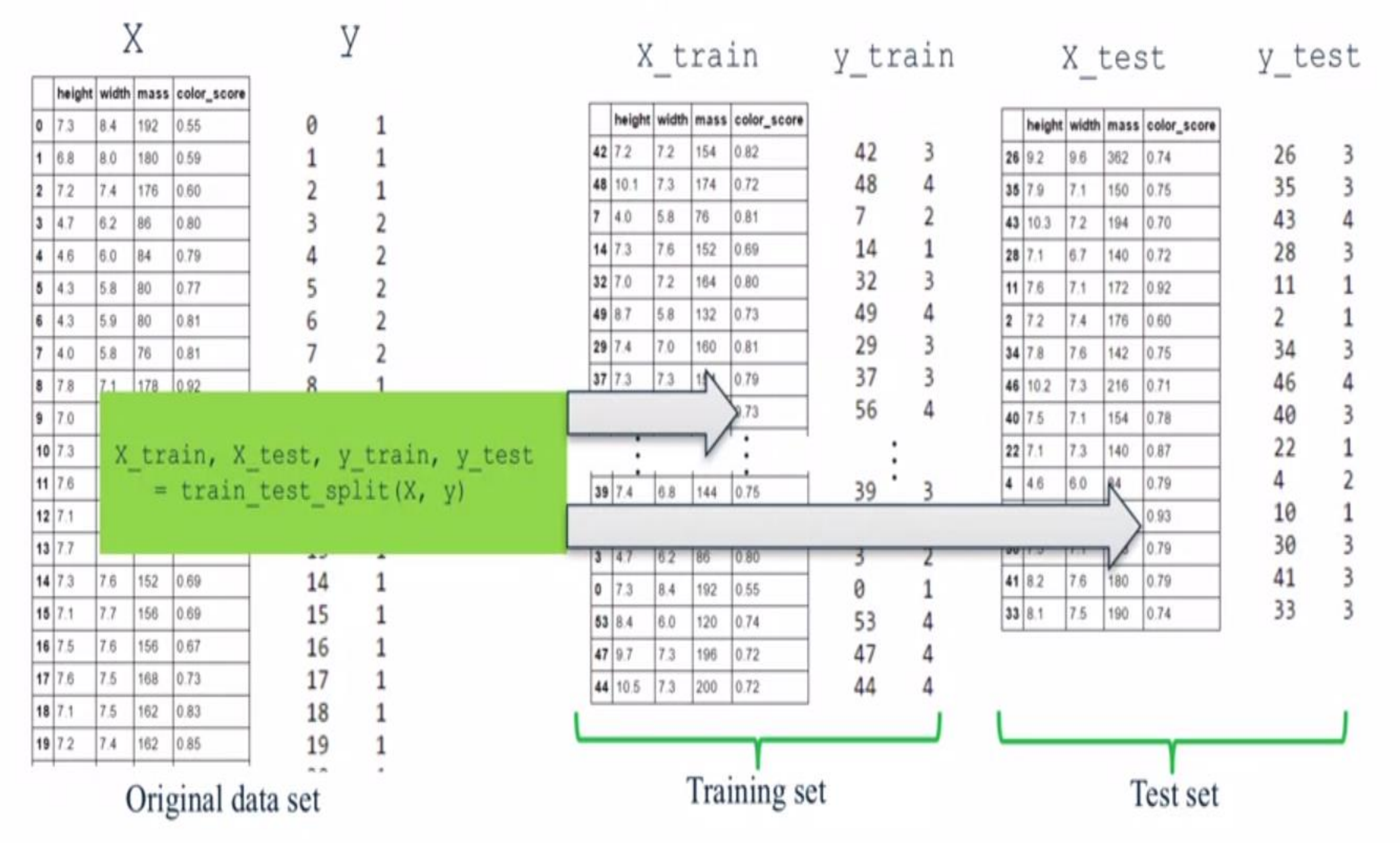
Machine Learnir Algorithm

Prediction

Where can we get new data that can be used to test the prediction model?

Creating Training and Testing datasets

The need to test and evaluate the performance of the prediction model requires dividing the original labelled dataset into several parts.



X

	height	width	mass	color_score
0	7.3	8.4	192	0.55
1	6.8	8.0	180	0.59
2	7.2	7.4	176	0.60
3	4.7	6.2	86	0.80
4	4.6	6.0	84	0.79
5	4.3	5.8	80	0.77
6	4.3	5.9	80	0.81
7	4.0	5.8	76	0.81
8	7.8	7.1	178	0.92
9	7.0	[illegible]	[illegible]	[illegible]
10	7.3	[illegible]	[illegible]	[illegible]
11	7.6	[illegible]	[illegible]	[illegible]
12	7.1	[illegible]	[illegible]	[illegible]
13	7.7	[illegible]	[illegible]	[illegible]
14	7.3	7.6	152	0.69
15	7.1	7.7	156	0.69
16	7.5	7.6	156	0.67
17	7.6	7.5	168	0.73
18	7.1	7.5	162	0.83
19	7.2	7.4	162	0.85

y

0	1
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	1
14	1
15	1
16	1
17	1
18	1
19	1

```
X_train, X_test, y_train, y_test
   = train_test_split(X, y)
```

Original data set

X_train

	height	width	mass	color_score
42	7.2	7.2	154	0.82
48	10.1	7.3	174	0.72
7	4.0	5.8	76	0.81
14	7.3	7.6	152	0.69
32	7.0	7.2	164	0.80
49	8.7	5.8	132	0.73
29	7.4	7.0	160	0.81
37	7.3	7.3	[illegible]	0.79
[illegible]	[illegible]	[illegible]	[illegible]	0.73
⋮				⋮
39	7.4	6.8	144	0.75
3	4.7	6.2	86	0.80
0	7.3	8.4	192	0.55
53	8.4	6.0	120	0.74
47	9.7	7.3	196	0.72
44	10.5	7.3	200	0.72

y_train

42	3
48	4
7	2
14	1
32	3
49	4
29	3
37	3
56	4
⋮	
39	3
3	2
0	1
53	4
47	4
44	4

Training set

X_test

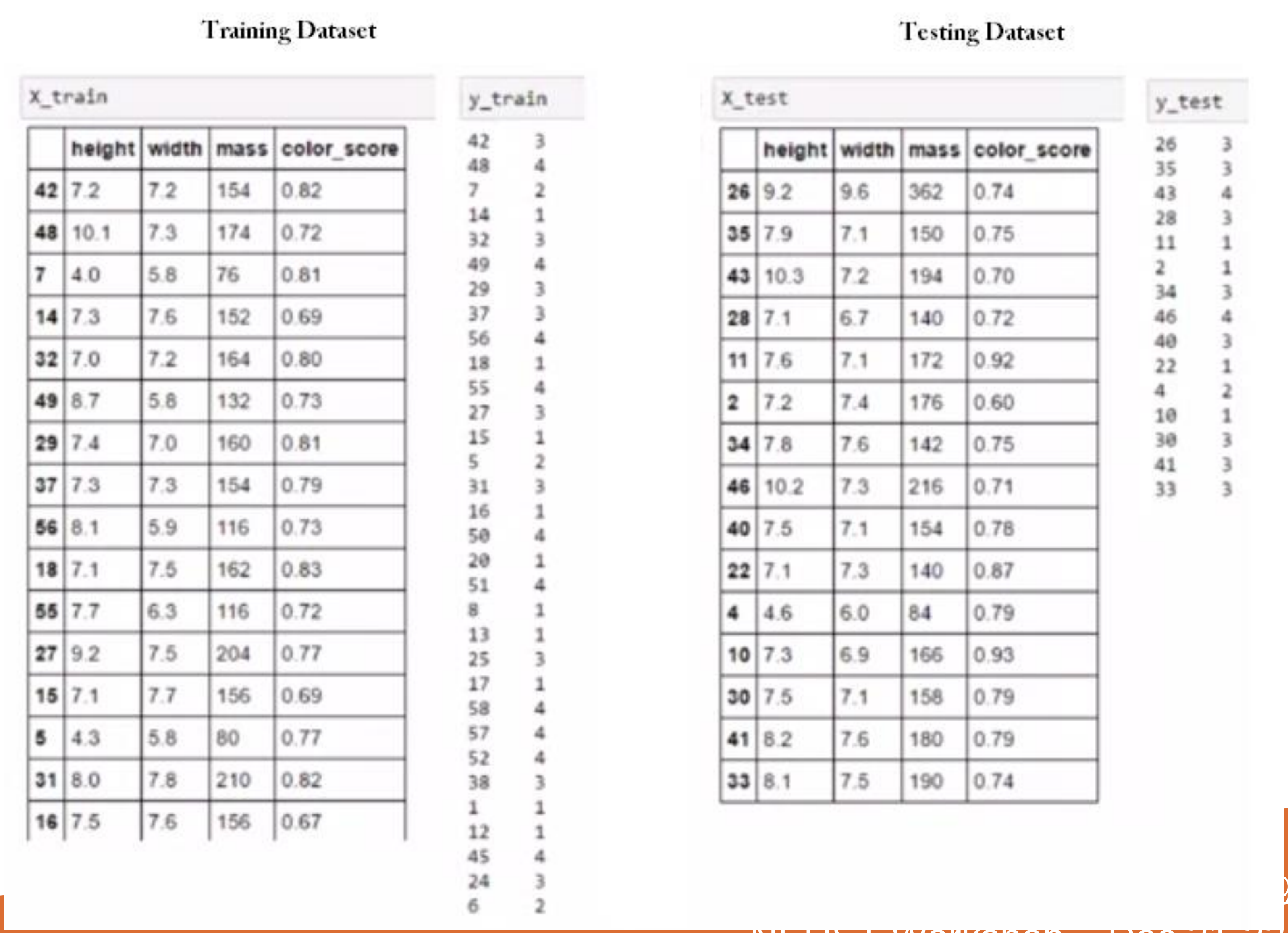
	height	width	mass	color_score
26	9.2	9.6	362	0.74
35	7.9	7.1	150	0.75
43	10.3	7.2	194	0.70
28	7.1	6.7	140	0.72
11	7.6	7.1	172	0.92
2	7.2	7.4	176	0.60
34	7.8	7.6	142	0.75
46	10.2	7.3	216	0.71
40	7.5	7.1	154	0.78
22	7.1	7.3	140	0.87
4	4.6	6.0	84	0.79
[illegible]	[illegible]	[illegible]	[illegible]	0.93
[illegible]	[illegible]	[illegible]	[illegible]	0.79
41	8.2	7.6	180	0.79
33	8.1	7.5	190	0.74

y_test

26	3
35	3
43	4
28	3
11	1
2	1
34	3
46	4
40	3
22	1
4	2
10	1
30	3
41	3
33	3

Test set

Creating Training and Testing datasets



Training Dataset

X_train

	height	width	mass	color_score
42	7.2	7.2	154	0.82
48	10.1	7.3	174	0.72
7	4.0	5.8	76	0.81
14	7.3	7.6	152	0.69
32	7.0	7.2	164	0.80
49	8.7	5.8	132	0.73
29	7.4	7.0	160	0.81
37	7.3	7.3	154	0.79
56	8.1	5.9	116	0.73
18	7.1	7.5	162	0.83
55	7.7	6.3	116	0.72
27	9.2	7.5	204	0.77
15	7.1	7.7	156	0.69
5	4.3	5.8	80	0.77
31	8.0	7.8	210	0.82
16	7.5	7.6	156	0.67

y_train

```
42    3
48    4
7     2
14    1
32    3
49    4
29    3
37    3
56    4
18    1
55    4
27    3
15    1
5     2
31    3
16    1
50    4
20    1
51    4
8     1
13    1
25    3
17    1
58    4
57    4
52    4
38    3
1     1
12    1
45    4
24    3
6     2
```

Testing Dataset

X_test

	height	width	mass	color_score
26	9.2	9.6	362	0.74
35	7.9	7.1	150	0.75
43	10.3	7.2	194	0.70
28	7.1	6.7	140	0.72
11	7.6	7.1	172	0.92
2	7.2	7.4	176	0.60
34	7.8	7.6	142	0.75
46	10.2	7.3	216	0.71
40	7.5	7.1	154	0.78
22	7.1	7.3	140	0.87
4	4.6	6.0	84	0.79
10	7.3	6.9	166	0.93
30	7.5	7.1	158	0.79
41	8.2	7.6	180	0.79
33	8.1	7.5	190	0.74

y_test

```
26    3
35    3
43    4
28    3
11    1
2     1
34    3
46    4
40    3
22    1
4     2
10    1
30    3
41    3
33    3
```

Demo and Exercise