Features Extraction and Preprocessing

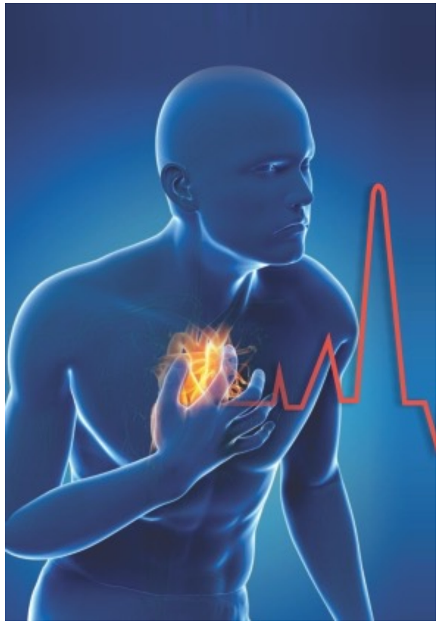
PLAKSHA UNIVERSITY

What are features in ML?

Features are the independent variables in ML models.

Example 1

Dependent Variable



Risk of Cardiac Disease

Independent Variables (Features)

- Age
- Weight
- Sex
- Whether the person smokes
- Whether the person consumes alcohol
- Whether the person has diabetes
- Whether the person is heavily stressed
- Whether the person is physically active
- ...

What are features in ML?

Features are the independent variables in ML models.

Example 2

Dependent Variable

Probability of getting the job

Independent Variables (Features)

- Educational Qualification
- Number of years of professional experience
- Strength of the letters of recommendation
- Whether they have leadership skills
- Strength of their communication skills
- …

What are features in ML?

Features are the independent variables in ML models.

Example 3

Dependent Variable

Plaksha University Happiness Index

Independent Variables (Features)

- Vibrancy of Student Life
- Student's Mental Health
- Student Achievements
- Hostel Infrastructure
- Ease of interaction with faculty
- …

Features Representation in a ML Dataset

As a general convention in ML, rows in a dataset denote instances while columns in a dataset denote features and classes.

Features ⟷ **Targets/ Classes**

Instances

1 Sex	2 Age	3 Weight	4 BloodPressure		5 Smoking
Male	38	79	124	93	1
Male	43	73	109	77	0
Female	38	59	125	83	0
Female	40	60	117	75	0
Female	49	53	122	80	0
Female	46	64	121	70	0
Female	33	64	130	88	1
Male	40	81	115	82	0
Male	28	82	115	78	0
Female	31	59	118	86	0
Female	45	57	114	77	0
Female	42	62	115	68	0
Male	25	78	127	74	0
Male	39	91	130	95	1
Female	36	58	114	79	0
Male	48	81	130	92	1
Male	32	86	124	95	1
Female	27	59	123	79	1
Male	37	81	119	77	0
Male	50	77	125	76	0
Female	48	60	121	75	0
Female	39	53	123	79	0
Female	41	62	114	88	0
Female	44	66	128	90	1
Female	28	55	129	96	1
Male	25	85	114	77	0

Features are of two types

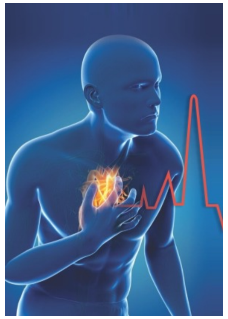
1. Continuous Features

   Numerical values that can take on any value in a certain range.

2. Categorical (Discrete) Features

   Features that can be divided into categories.

Dependent Variable



Risk of Cardiac Disease

Continuous Features

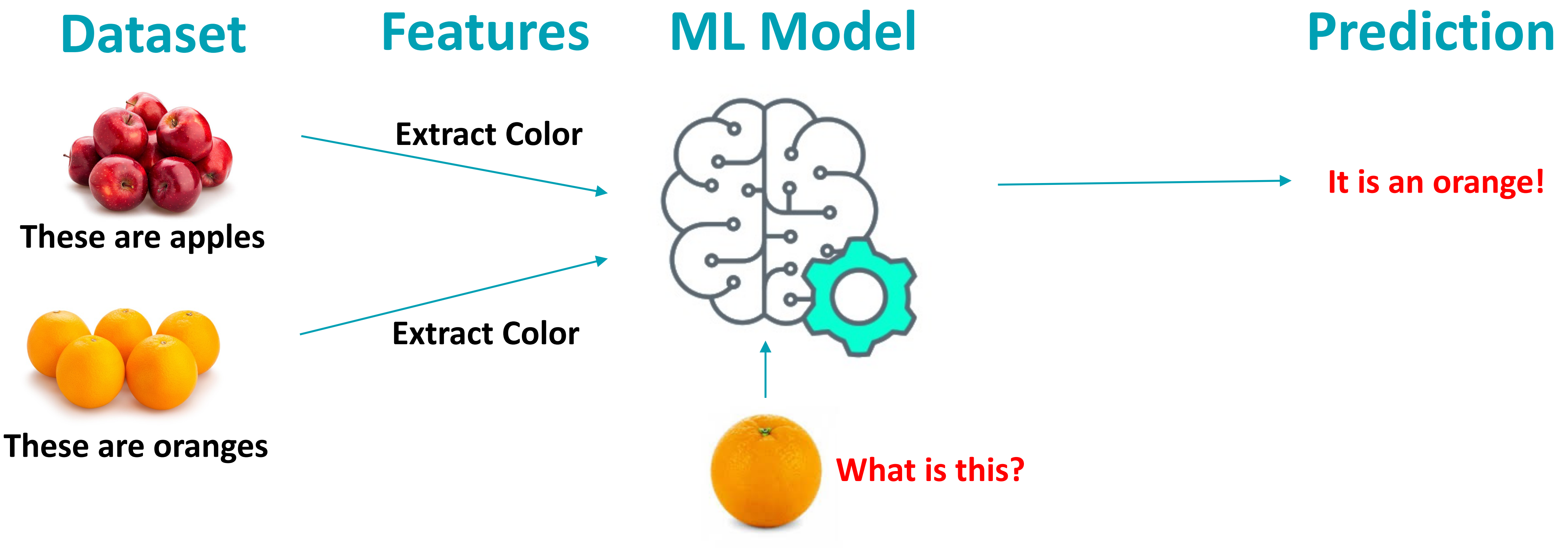
- Age (18-90)
- Weight (45-110)
- Cigarettes they smoke/day (0-30)
- Hours of exercise/day (0-6)

Categorical (Discrete) Features

- Sex (Male/Female/Intersex)
- Whether the person smokes (True/False)
- Whether the person has diabetes (True/False)
- Whether the person is heavily stressed (True/False)
- Whether the person is physically active (True/False)

Why use features in ML?

1. Features represent relevant information in data that could be useful in interpreting it. For example: Color could be a great feature to classify fruits into apples and oranges.



Why use features in ML?

2. Choice of features has a big impact on the quality of insights you get from ML models. For example:

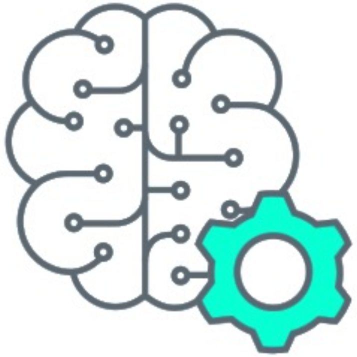
Dataset | **Features** | **ML Model** | **Prediction**

These are apples

~~Extract Color~~
Extract shape

These are bananas

~~Extract Color~~
Extract shape



What is this?

It is a banana!

Features Preprocessing

- Features Preprocessing are steps used to transform the features data so that it is easily parsed by the ML model.

- Real-world datasets are highly susceptible to issues such as missing values, duplicates, outliers, etc. These need to be taken care of before sending the features into the ML model, else the ML model may not work efficiently.

- Categorical features have to be transformed into a "numerical" representation so that the ML model can understand them. For example, a feature like *Car Model* that can have values like *Swift/Thar/Nexon/Scorpio* needs to be converted into a representation that the ML model could understand.

Features Preprocessing

I. Missing values in the dataset

Missing feature values in a dataset must be filled for the ML model to work.

a) If many instances are missing, ignore the feature.

	Height	Weight	Country	Place	Number of days	Some column
0	12.0	35.0	India	Bengaluru	1.0	NaN
1	NaN	36.0	US	New York	2.0	NaN
2	13.0	32.0	UK	London	NaN	NaN
3	15.0	NaN	France	Paris	4.0	NaN
4	16.0	39.0	US	California	5.0	12.0
5	NaN	NaN	NaN	Mumbai	NaN	NaN
6	NaN	NaN	NaN	NaN	6.0	NaN

Features Preprocessing

I. Missing values in the dataset

Missing feature values in a dataset must be filled for the ML model to work.

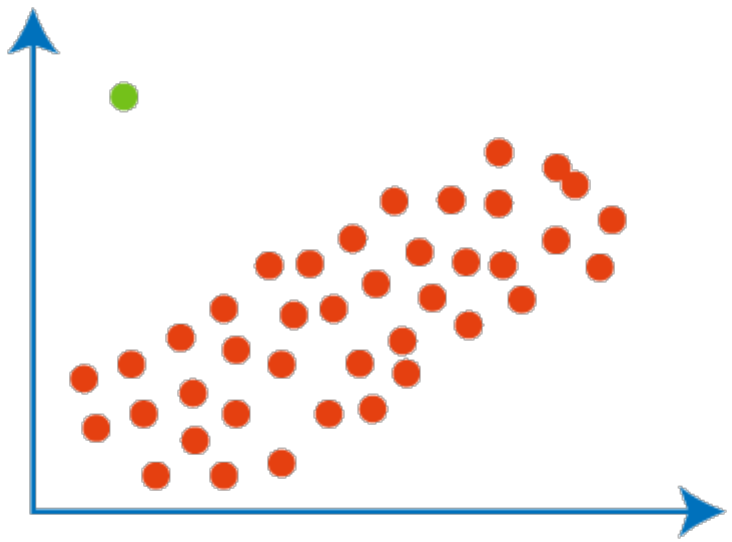
b) Fill in the missing values by regression/interpolation

Row/ Col	1	2	3	4	5	6
1	0.24	-0.1		0.18	0.42	-0.25
2	0.19	-0.22	-0.2	0.12	0.21	-0.26
3	0.21	0.09	0.57	-0.14	0.29	0.01
4	0.76	0.07	0.04	-0.06	0.3	-0.47
5	0.46	0.12	0.49	-0.42	0.28	-0.3
6	0.43	-0.23	-0.3	-0.24	0.23	
7	0.44	-0.32	0.26	-0.77	0.31	-0.09
8	0.11	0.03		-0.24	0.36	-0.11
9	0.32	0	0.26	-0.5	0.31	0.1
10	0.12	-0.01	-0.13	0.12	0.47	-0.3
11	0.53	0.25	0.49	-0.3	0.13	-0.12
12	0.17	0.06	0.06	0.28	0.38	-0.23
13	0.19	-0.06	0.05	-0.25	0.23	-0.05

Features Preprocessing

II. Outlier removal

It is important to remove outliers because they interfere with model fitting and may inflate the error metrics while assessing performance of a ML model.



An outlier

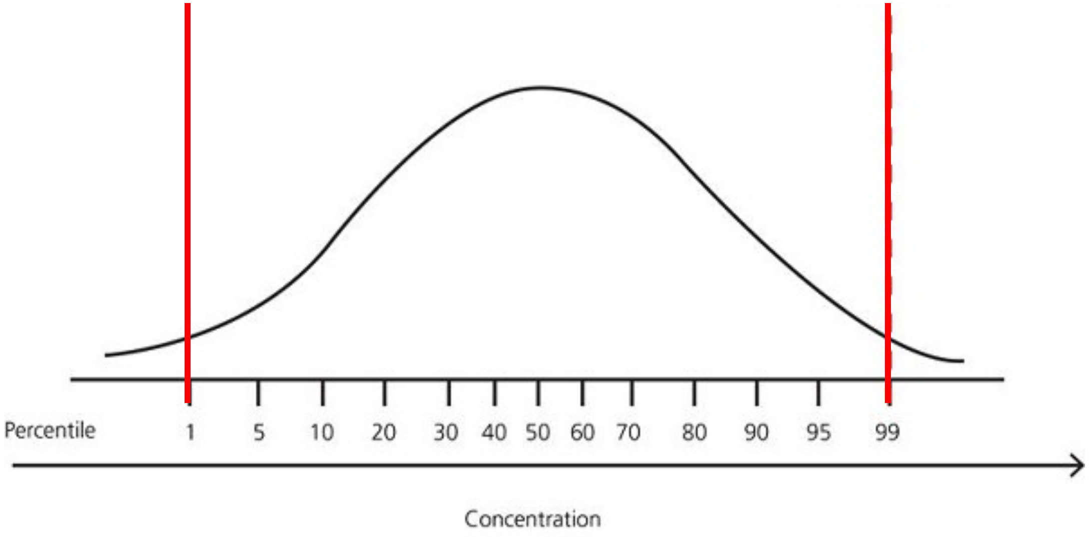
Can occur due to noise/errors in data collection.

Can occur due to a valid natural outlier in data.

Features Preprocessing

II. Outlier removal

a) Using Bell Curve



An easy way to detect outliers is to use the Bell Curve

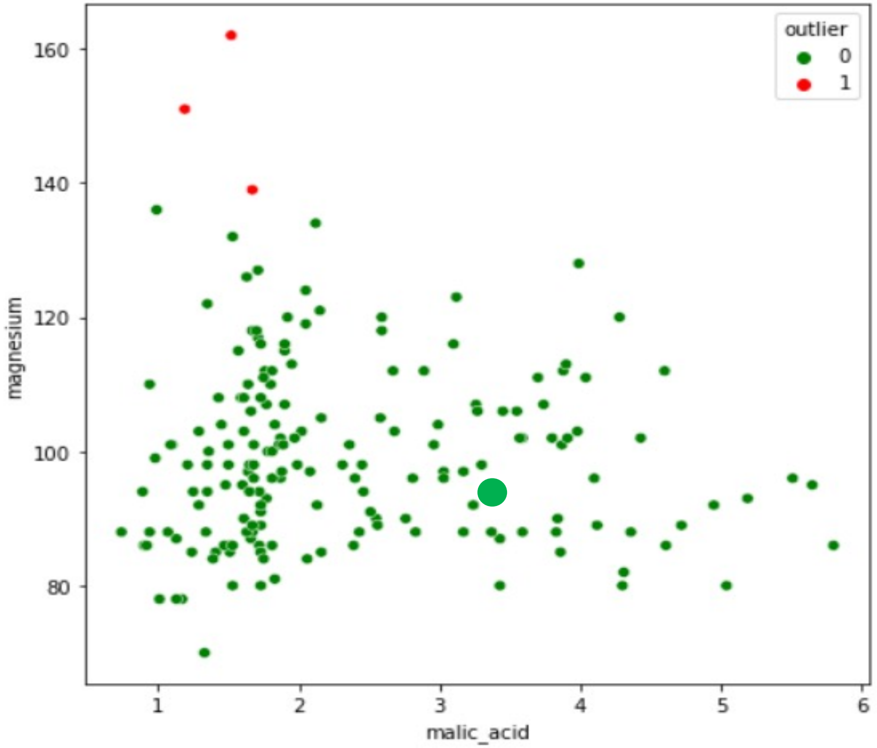
For e.g., labels datapoints that lie outside [1-99] percentile of data as outliers.

Good for Gaussian data.

Features Preprocessing

II. Outlier removal

b) Using distance from mean



Find distance (Euclidean) of each point from the mean of the dataset.

If $dist_i$ > threshold, label $data_i$ as an outlier.

Sensitive to the cutoff threshold.

Graphic courtesy: https://towardsdatascience.com/outlier-detection-methods-in-machine-learning-1c8b7cca6cb8

Features Preprocessing

III. Features Scaling

Features Scaling is necessary to ensure that all features have comparable scale and ranges to efficiently train the ML model.

a) Normalization

$$x_{\text{norm}} = \frac{x - \min(x)}{\max(x) - \min(x)}$$

#	Emp	Age	Salary
1	Emp1	44	73000
2	Emp2	27	47000
3	Emp3	30	53000
4	Emp4	38	62000
5	Emp5	40	57000
6	Emp6	35	53000
7	Emp7	48	78000

Normalization

Age	Normalized Age	Salary	Normalized Salary
44	0.80952381	73000	0.838709677
27	0	47000	0
30	0.142857143	53000	0.193548387
38	0.523809524	62000	0.483870968
40	0.619047619	57000	0.322580645
35	0.380952381	53000	0.193548387
48	1	78000	1
	Range 0-1		**Range 0-1**

How to calculate Normalized value?
X = 35, min = 27, max = 48 for column Age.
Xnorm(for 35) = $\frac{35-27}{48-27}$ = 0.3809

Features are scaled between 0 and 1.

Good for non-Gaussian data.

Prone to outliers.

Graphic courtesy: https://ashutoshtripathi.com/2021/06/12/what-is-feature-scaling-in-machine-learning-normalization-vs-standardization/

Features Preprocessing

III. Features Scaling

Features Scaling is necessary to ensure that all features have comparable scale and ranges to efficiently train the ML model.

b) z-Scoring (Standardization)

$$x_{\text{stand}} = \frac{x - \text{mean}(x)}{\text{standard deviation }(x)}$$

#	Emp	Age	Salary
1	Emp1	44	73000
2	Emp2	27	47000
3	Emp3	30	53000
4	Emp4	38	62000
5	Emp5	40	57000
6	Emp6	35	53000
7	Emp7	48	78000
		Mean = 37.42857 Std. Dev. = 6.883876	Mean = 60428.5714 Std. Dev = 10499.7570

Standardization

How to calculate Standardized value?
X = 35, mean = 37.42, Std. Dev. = 6.88
for column Age.
Xstd(for 35) = $\frac{35-37.42}{6.88}$ =-0.3527

Age	Standardized Age	Salary	Standardized Salary
44	0.954611636	73000	1.197306616
27	-1.514927162	47000	-1.278941158
30	-1.079126198	53000	-0.707499364
38	0.083009708	62000	0.149663327
40	0.373543684	57000	-0.326538168
35	-0.352791257	53000	-0.707499364
48	1.535679589	78000	1.673508111
	Mean = 0 Std. dev. = 1		Mean = 0 Std. dev. = 1

Features are scaled such that mean = 0, std = 1.

Good for Gaussian data.

More robust to outliers.

Graphic courtesy: https://ashutoshtripathi.com/2021/06/12/what-is-feature-scaling-in-machine-learning-normalization-vs-standardization/

Features Preprocessing

IV. Encoding for Categorical Features

ML models can only understand numbers. How to convert categorical data into numbers?

Human-Readable	Machine-Readable
Pet	
Cat	1
Dog	2
Turtle	3
Fish	4
Cat	1

But, does that mean

Fish (4) > Dog (2)?

Obviously not! These are categories and not numbers.

Features Preprocessing

IV. Encoding for Categorical Features

a) One-Hot Encoding

Human-Readable

Pet
Cat
Dog
Turtle
Fish
Cat

Machine-Readable

Cat	Dog	Turtle	Fish
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1
1	0	0	0

Every unique value in category becomes a feature.

But, dimensionality increases.

Bad when there are a lot of categories.

Graphic courtesy: https://medium.com/analytics-vidhya/stop-one-hot-encoding-your-categorical-variables-bbb0fba89809

Features Preprocessing

IV. Encoding for Categorical Features

b) Target Encoding

	Animal	Target	Encoded Animal
0	cat	1	0.40
1	hamster	0	0.50
2	cat	0	0.40
3	cat	1	0.40
4	dog	1	0.67
5	hamster	1	0.50
6	cat	0	0.40
7	dog	1	0.67
8	cat	0	0.40
9	dog	0	0.67

	Animal Group	Target 0	Target 1	Probability of 1
0	cat	3	2	0.40
1	dog	1	2	0.67
2	hamster	1	1	0.50

Every categorical feature is substituted by a probability value.

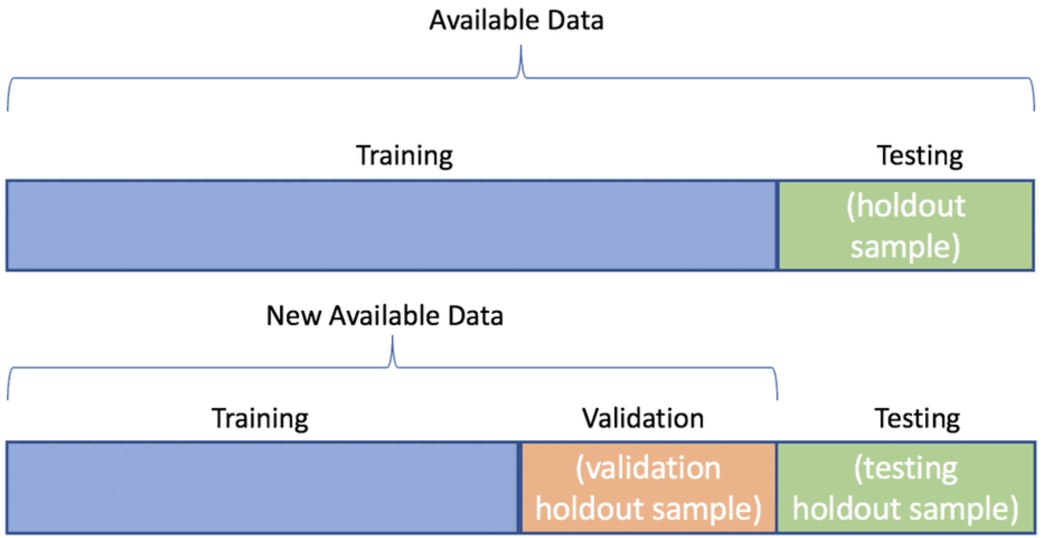
Dimensionality does not increase.

Prone to distribution of the target.
Bad when dataset is heavily skewed.

Graphic courtesy:
https://medium.com/analytics-vidhya/target-encoding-vs-one-hot-encoding-with-simple-examples-276a7e7b3e64

Splitting Datasets

- To evaluate the performance of a ML model, we split datasets into Train/Validation/Test sets.
- This is done so that we train our ML model on one set of data but test it on another test so that we can be sure if the ML model will work for new data.
- Usually, we randomly take 80% of the total dataset (rows) for training+validation and the remaining 20% for testing.



Splitting Datasets (N-fold Cross Validation)

- Training and Validation set is usually randomly broken N (usually 10) times to validate the ML model again and again before testing it on the Test set.