Machine Learning

Lesson 3—Data Preprocessing

Learning Objectives

- Recognize the importance of data preparation in Machine Learning
- Identify the meaning and aspects of feature engineering
- Standardize data features with feature scaling
- Analyze datasets and its examples
- Explain dimensionality reduction with Principal Component Analysis (PCA)

Data Preprocessing

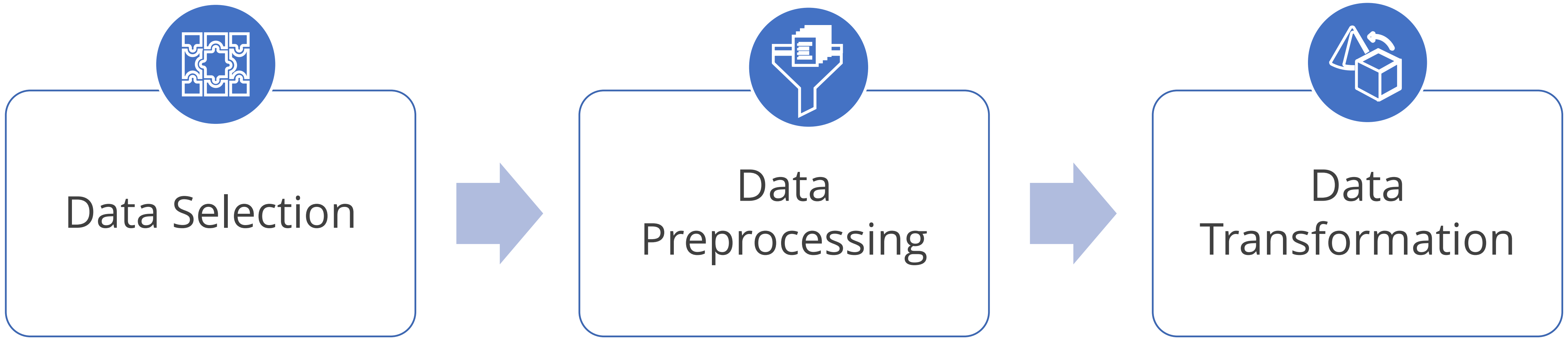
Topic 1: Data Preparation

Data Preparation in ML

- Machine Learning depends largely on test data.
- Data preparation is a crucial step to make it suitable for ML.
- A large amount of data is generally required for the most common forms of ML.
- Data preparation involves data selection, filtering, transformation, etc.

Data Preparation Process

The process of preparing data for Machine Learning algorithm comprises the following:

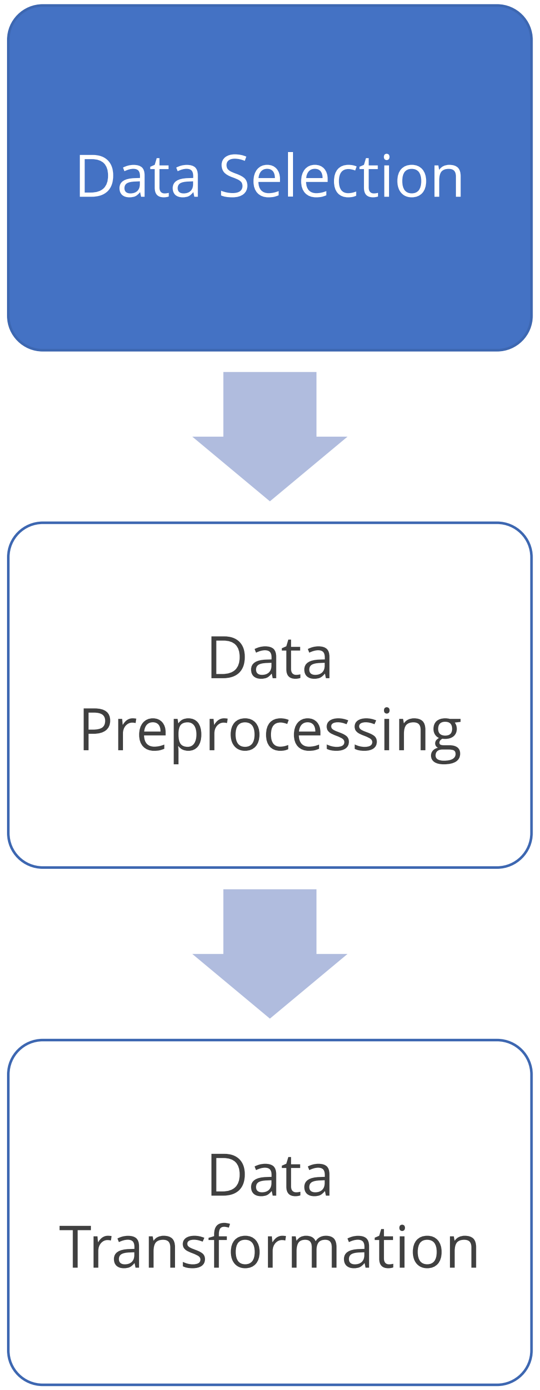


Data Selection

Data Preprocessing

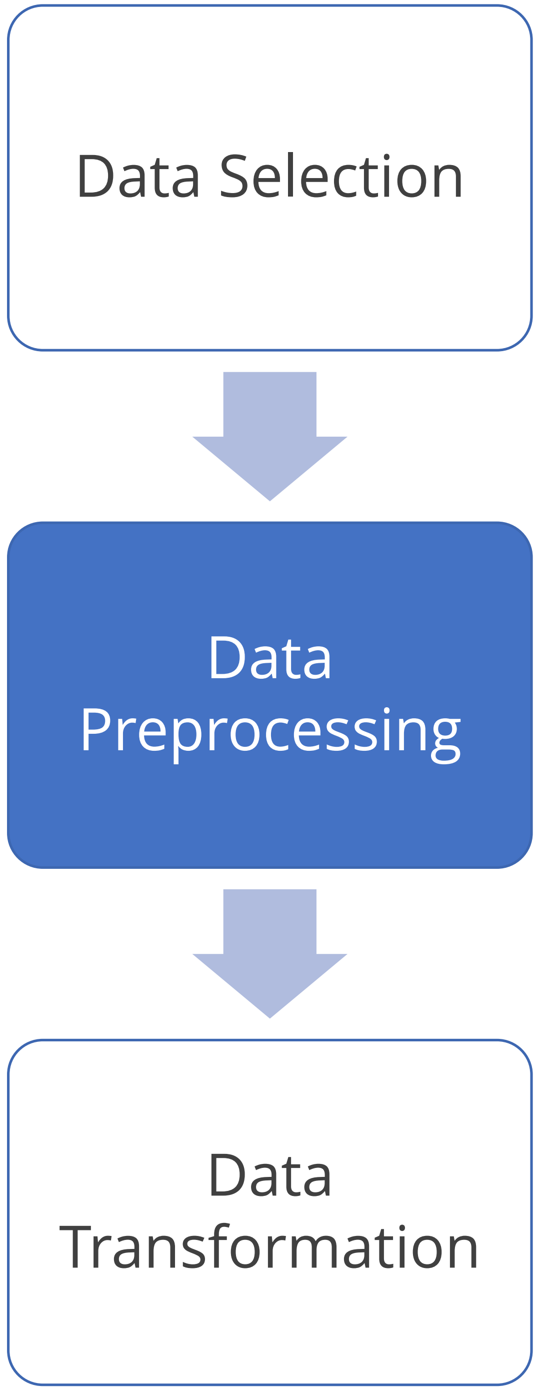
Data Transformation

Data Preparation Process



- There is a vast volume, variety, and velocity of available data for a Machine Learning problem.
- This step involves selecting only a subset of the available data.
- The selected sample must be an accurate representation of the entire population.
- Some data can be derived or simulated from the available data if required.
- Data not relevant to the problem at hand can be excluded.

Data Preparation Process



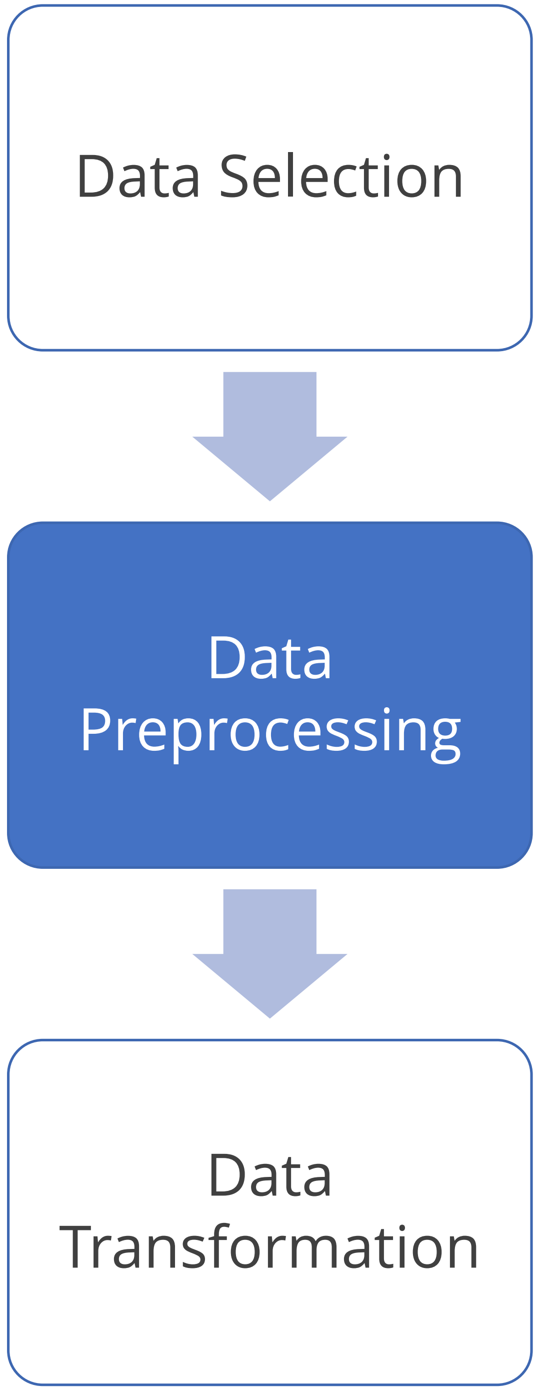
After the data has been selected, it needs to be preprocessed using the given steps:

1. Formatting the data to make it suitable for ML (structured format)

2. Cleaning the data to remove incomplete variables

3. Sampling the data further to reduce running times for algorithms and memory requirements

Data Preparation Process



Data Selection

Data Preprocessing

Data Transformation

Data cleaning at this stage involves filtering it based on the following variables:

Insufficient data

The amount of data required for ML algorithms can vary from thousands to millions, depending upon the complexity of the problem and the chosen algorithm.

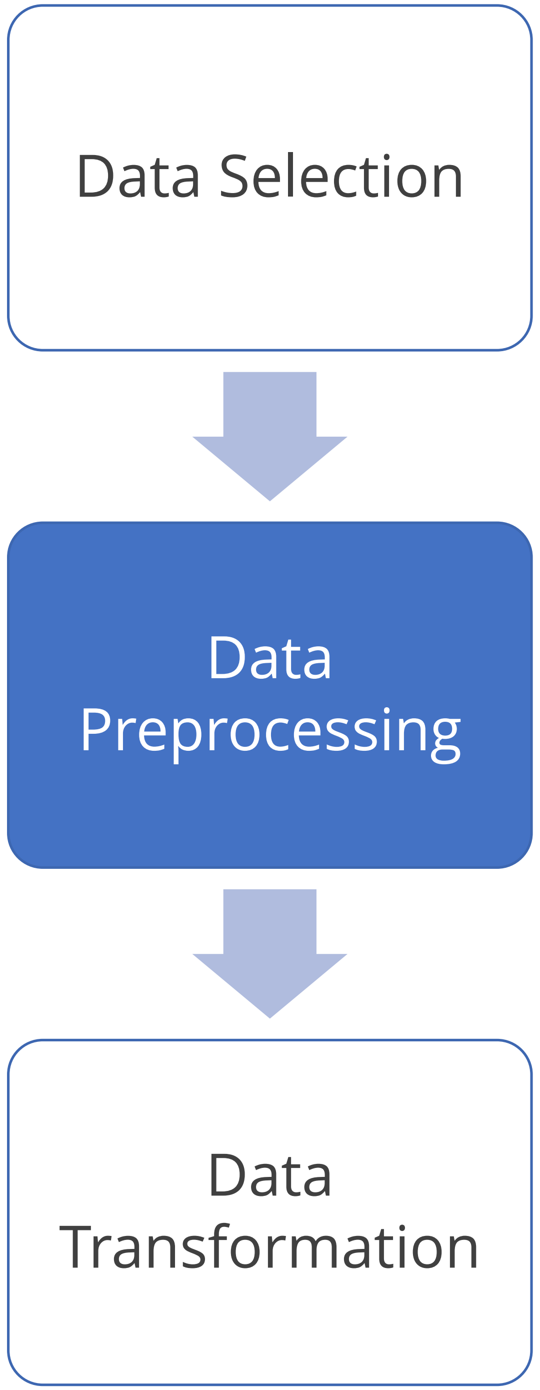
Nonrepresentative data

The sample selected must be an exact representation of the entire data, as nonrepresentative data might train an algorithm such that it won't generalize well on new test data.

Substandard data

Outliers, errors, and noise can be eliminated to get a better fitment of model. Missing features such as age for 10% of audience may be ignored completely, or an average value can be assumed for the missing component.

Data Preparation Process



Data Selection

Data Preprocessing

Data Transformation

Selecting the right size of sample is a key step in data preparation.

Samples that are too large or too small might give skewed results.

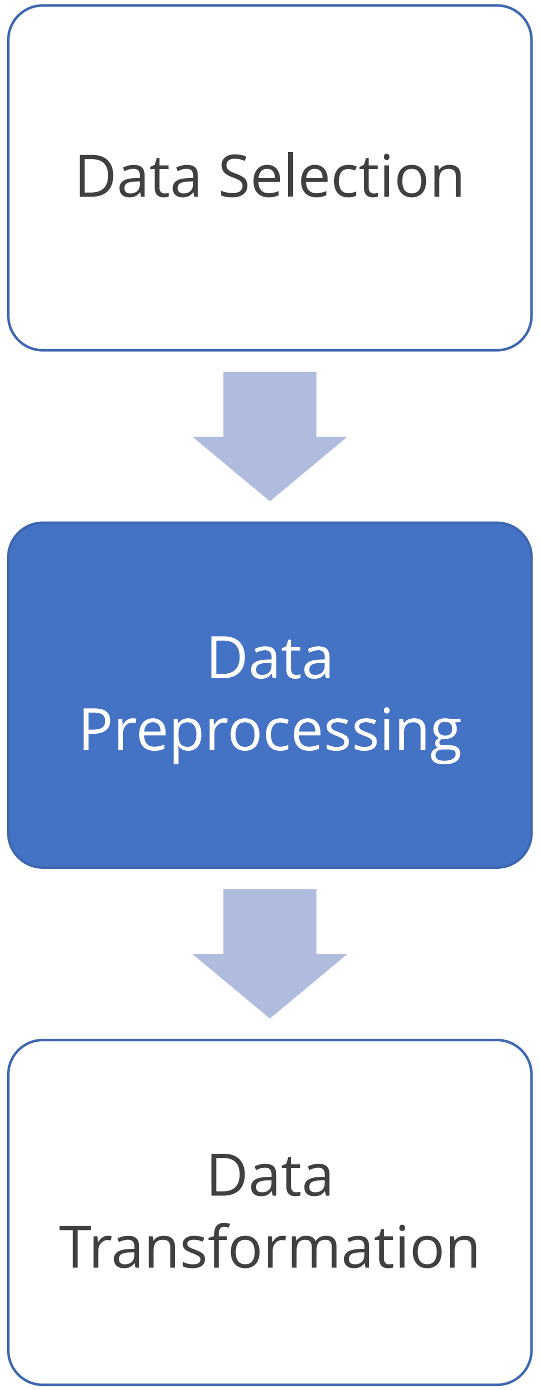
Sampling noise

Smaller samples cause sampling noise since they get trained on nonrepresentative data. For example, checking voter sentiment from a very small subset of voters.

Sampling bias

Larger samples work well as long as there is no sampling bias, that is, when the right data is picked.
For example, sampling bias would occur when checking voter sentiment only for technically sound subset of voters, while ignoring others.

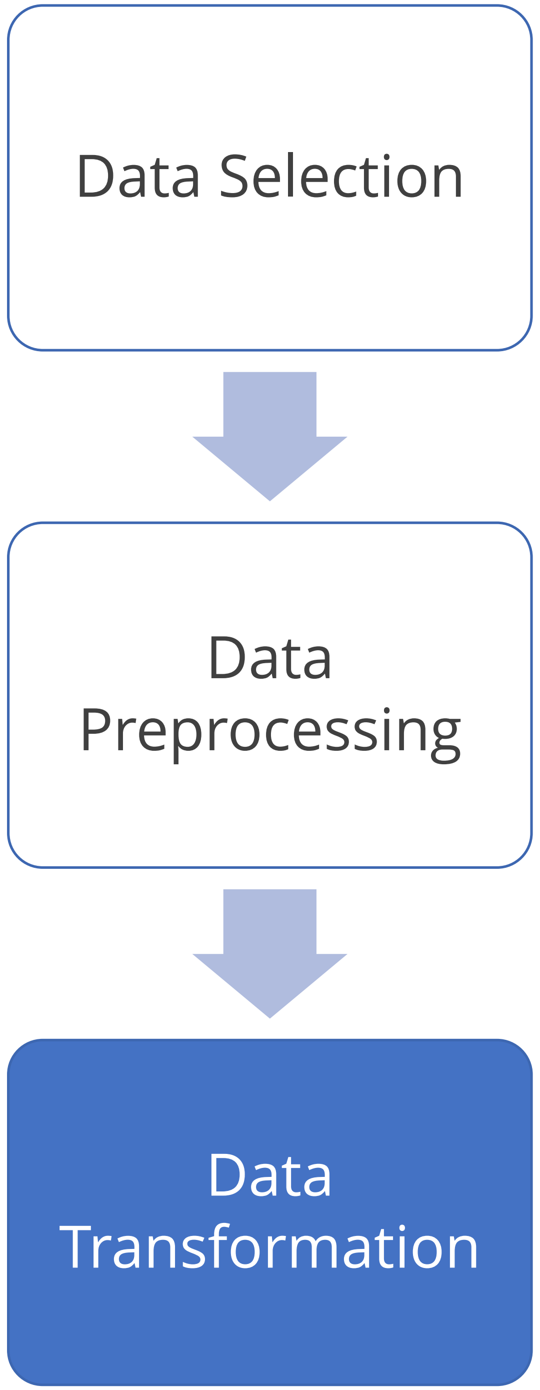
Data Preparation Process



Example: Data sample

Ethnicity	Height (CM)	Weight (Kg)	Will Survive till 70
White	186	90	Yes
African	185	98	No
Asian	175	80	No
African	180	88	Yes
Asian	178		No
Asian	172	72	Yes
White	178	75	No
White		89	Yes
African	186	90	Yes

Data Preparation Process



Data Selection

Data Preprocessing

Data Transformation

The selected and preprocessed data is transformed using one or more of the following methods:

1. Scaling: It involves selecting right feature scaling for the selected and preprocessed data.
2. Aggregation: This is the last step to collate a bunch of data features into a single one.

Types of Data

Labelled Data or Training Data

- It is also known as marked (with values) data.
- It assists in learning and forming predictive hypothesis for future data. It is used to arrive at a formula to predict future behavior.
- Typically 80% of available labelled data is marked for training.

Unlabelled Data

- Data which is not marked and needs real time unsupervised learning is categorized as unlabelled data.

Test Data

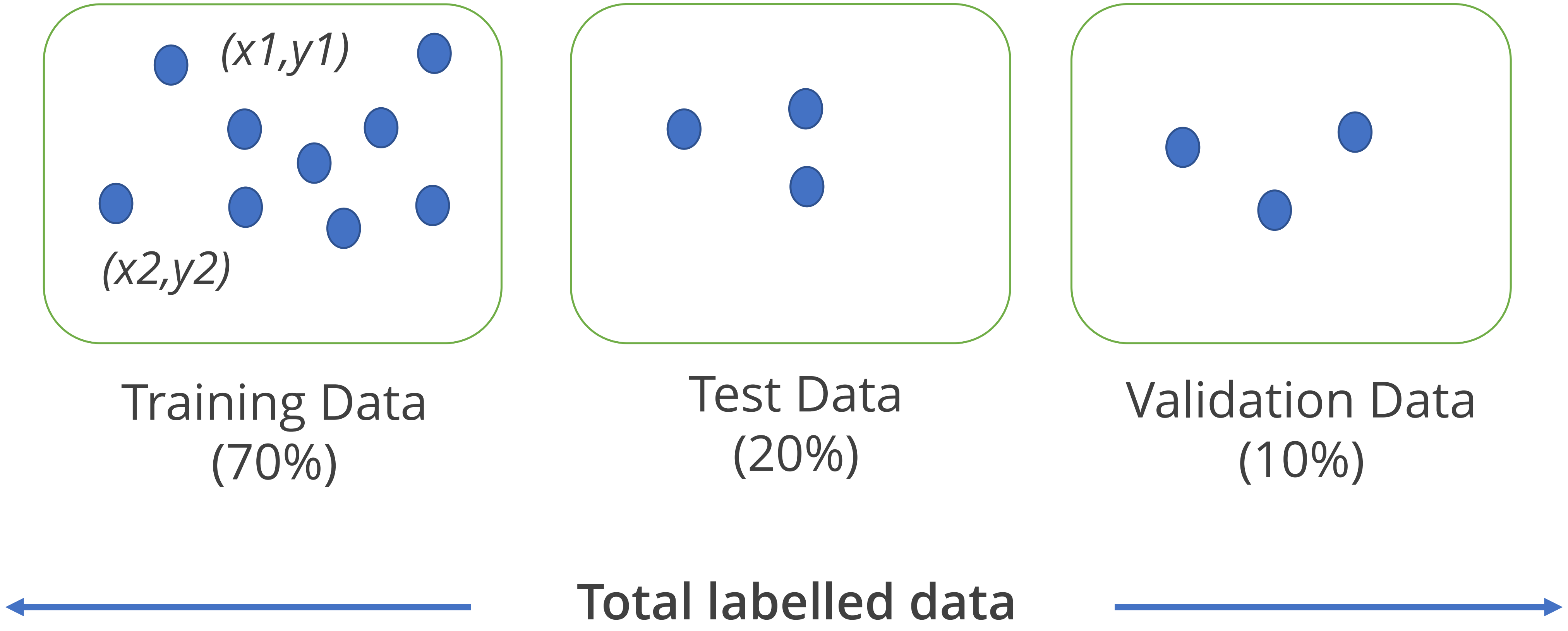
- Data provided to test a hypothesis created via prior learning is known as test data.
- Typically 20% of labelled data is reserved for test.

Validation Data

- It is a dataset used to retest the hypothesis (in case the algorithm got overfitted to even the test data due to multiple attempts at testing).

Types of Data

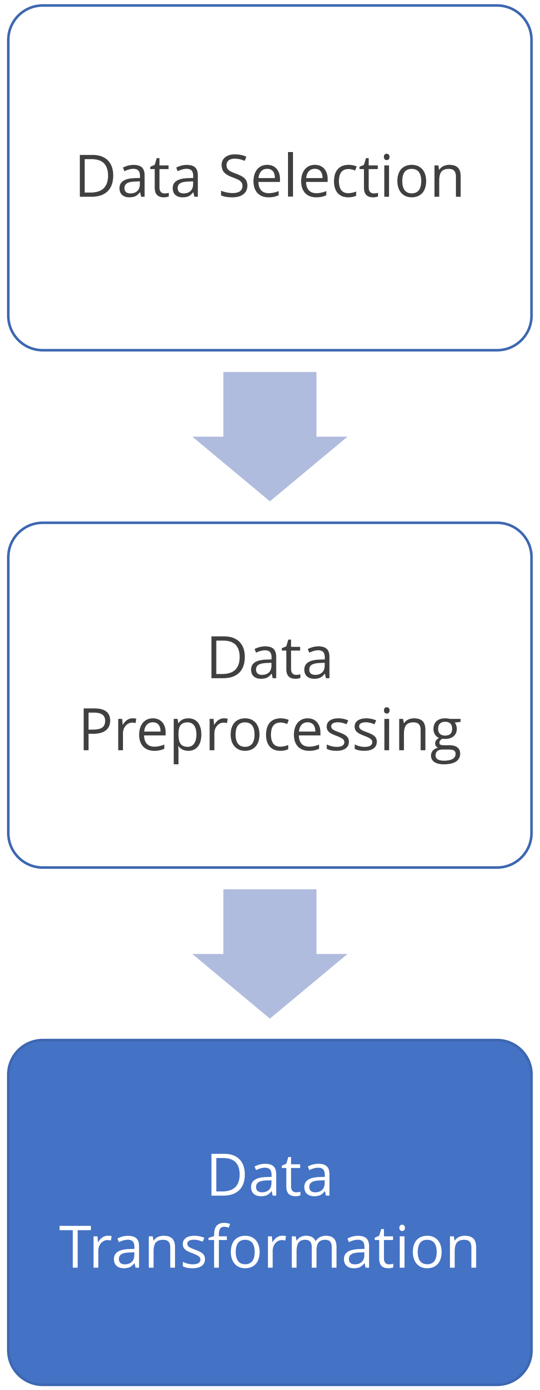
The illustration given below depicts how total available labelled data may be segregated into training dataset, test dataset, and validation dataset.



Data Preprocessing

Topic 2: Feature Engineering

Feature Engineering



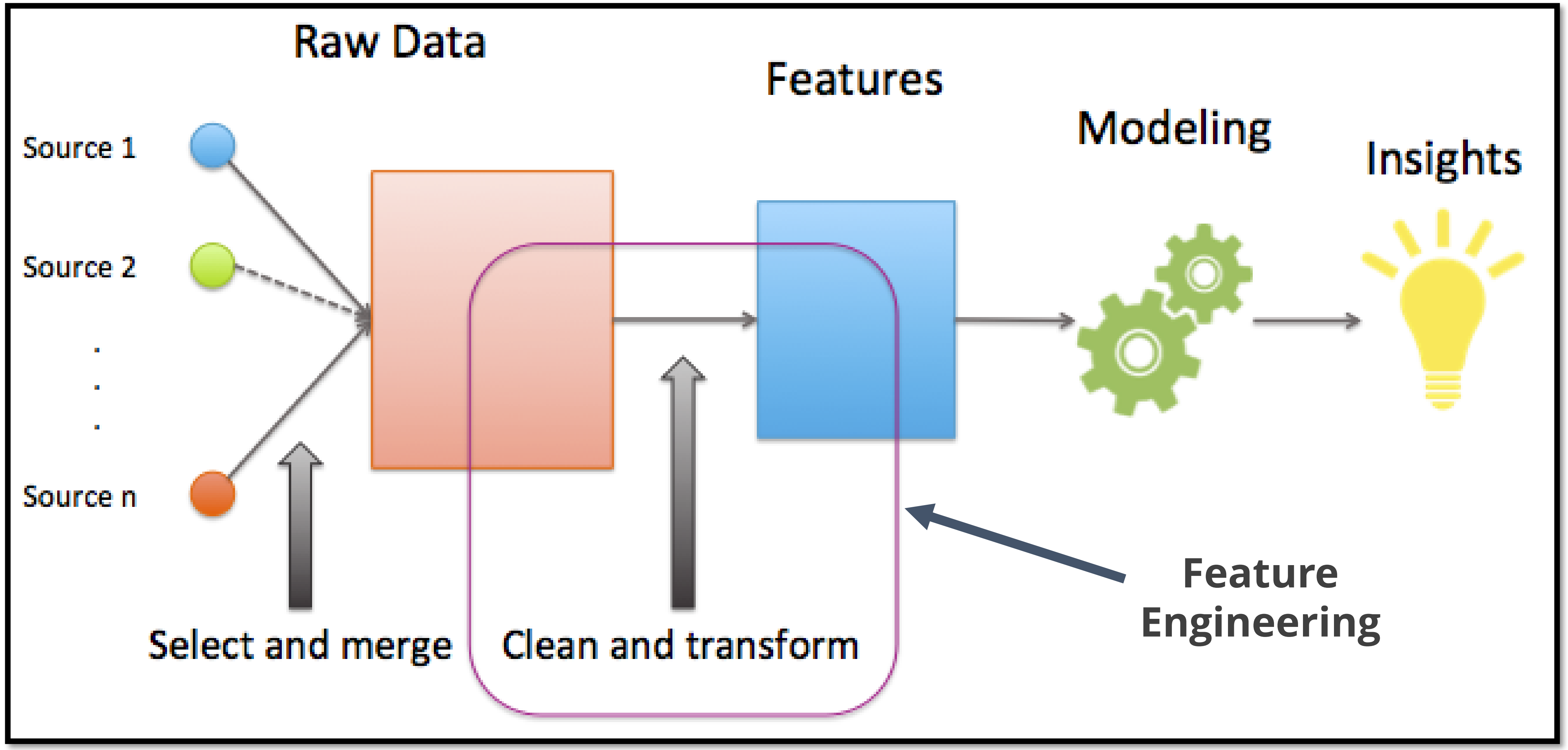
The transformation stage in the data preparation process includes an important step known as **Feature Engineering**.

Definition of Feature Engineering

> Feature Engineering refers to selecting and extracting right features from the data that are relevant to the task and model in consideration.

Feature Engineering in ML

The place of feature engineering in the machine learning workflow is shown below:



Source: safaribooksonline.com

Aspects of Feature Engineering

Feature Selection

Most useful and relevant features are selected from the available data

Feature Extraction

Existing features are combined to develop more useful ones

Feature Addition

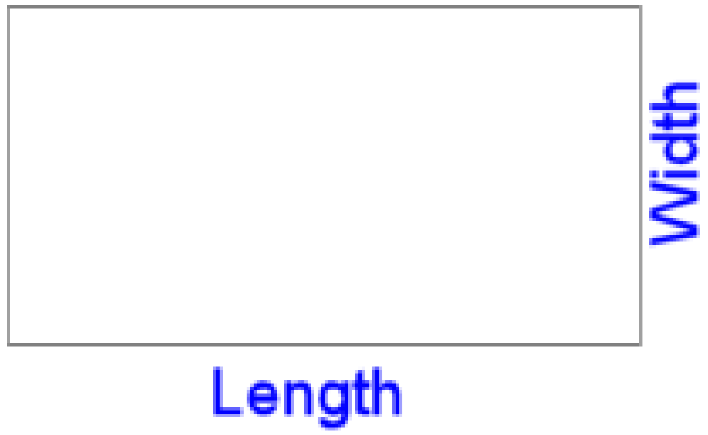
New features are created by gathering new data

Feature Filtering

Filter out irrelevant features to make the modelling step easy

Examples of Feature Engineering

Convert length and width into area

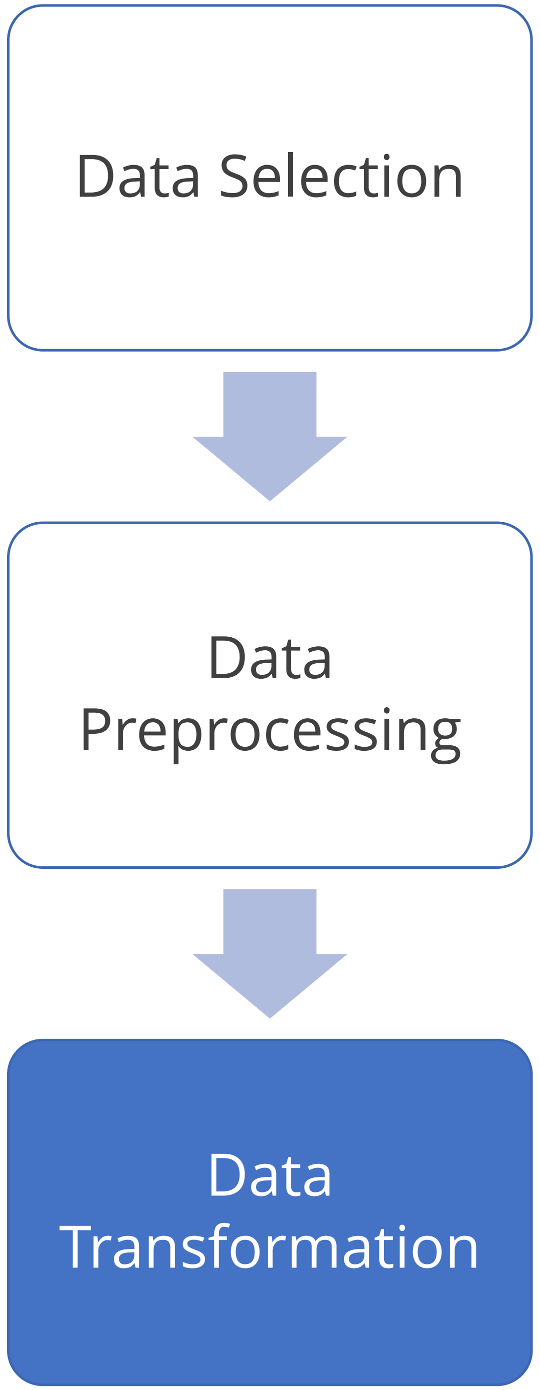


Combine passion for a task and skill for a task into one feature, known as “Task suitability.”

Data Preprocessing

Topic 3: Feature Scaling

Feature Scaling



Feature scaling is an important step in the data transformation stage of data preparation process.

Definition of Feature Scaling

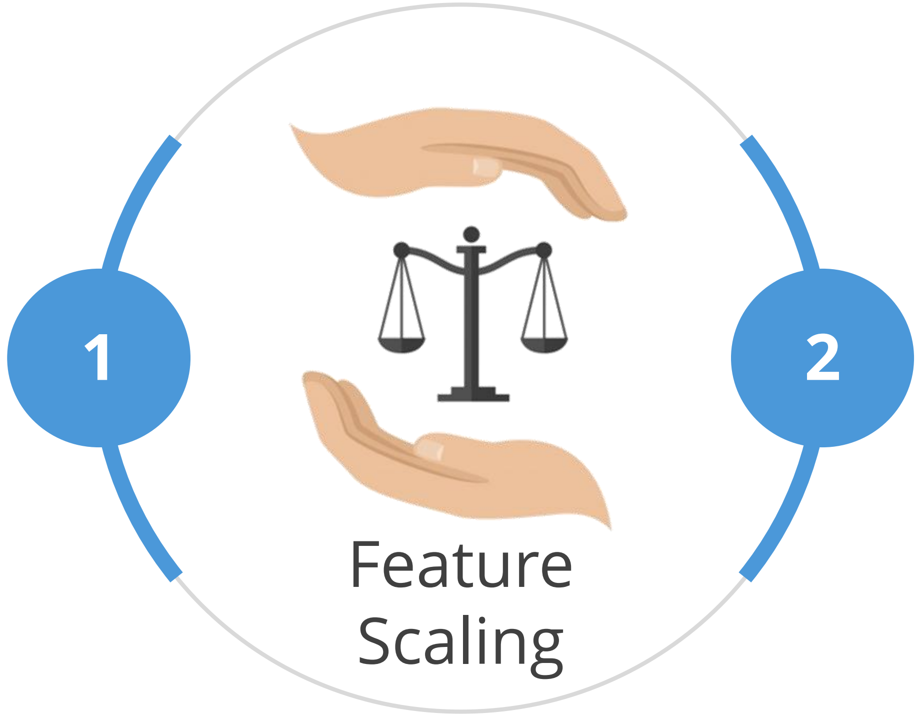
> Feature Scaling is a method used in Machine Learning for standardization of independent variables of data features.

Why Feature Scaling?

- Let’s consider a situation where input data has two features, one ranging from value 1 to 100 and the other from 1 to 10000.
- This might cause error in machine learning algorithms, like mean squared error method, when the optimizer tries to minimize larger errors in the second feature.
- The computed Euclidean distances between samples will be dominated by the second feature axis in K-nearest neighbours (KNN) algorithm.
- The solution lies in scaling all the features on a similar scale like (0 to 1) or (1 to 10).

Techniques of Feature Scaling

The two common techniques to scale features are:



Standardization

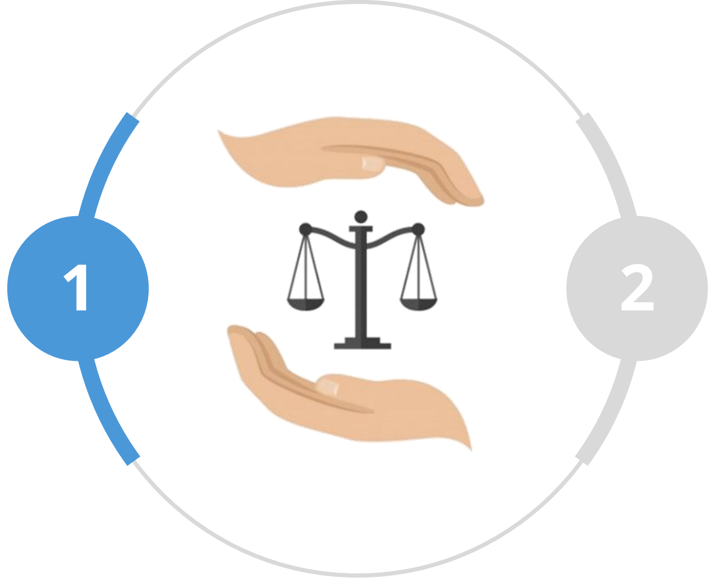
1

2

Normalization

Feature Scaling

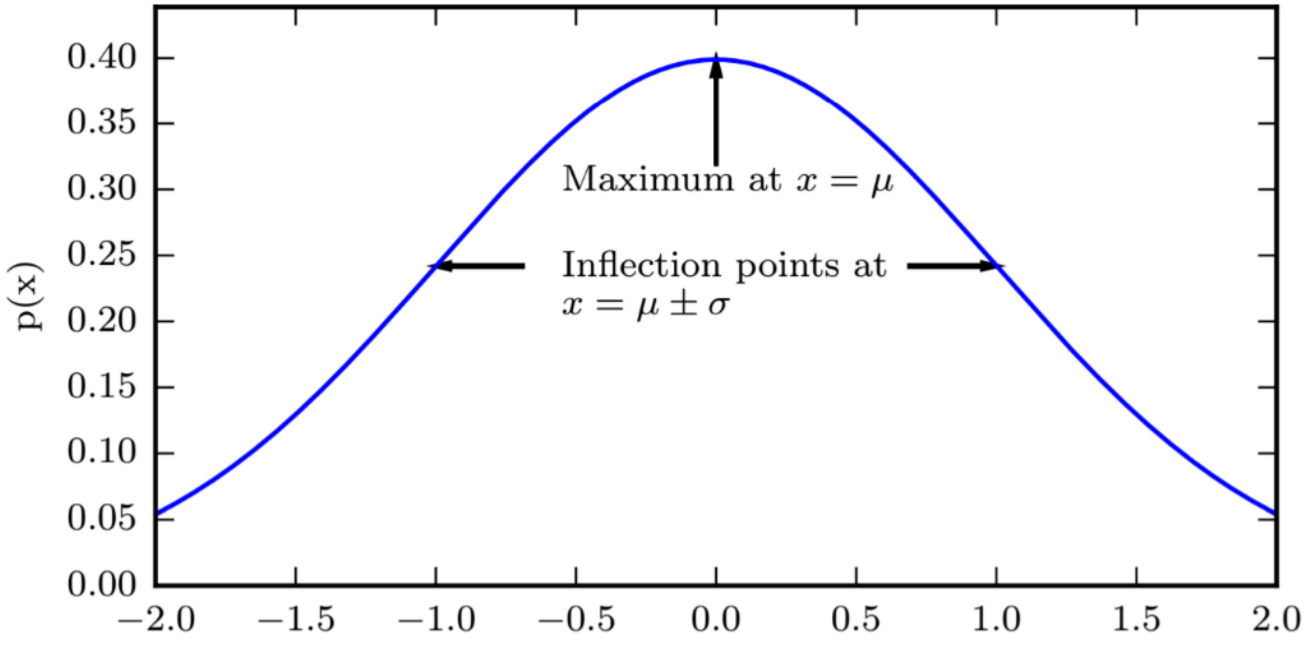
Feature Scaling: Standardization



Standardization

- Standardization is a popular feature scaling method, which gives data the property of a standard normal distribution (also known as Gaussian distribution).
- All features are standardized on the normal distribution (a mathematical model).
- The mean of each feature is centered at zero, and the feature column has a standard deviation of one.

Standardization: Example



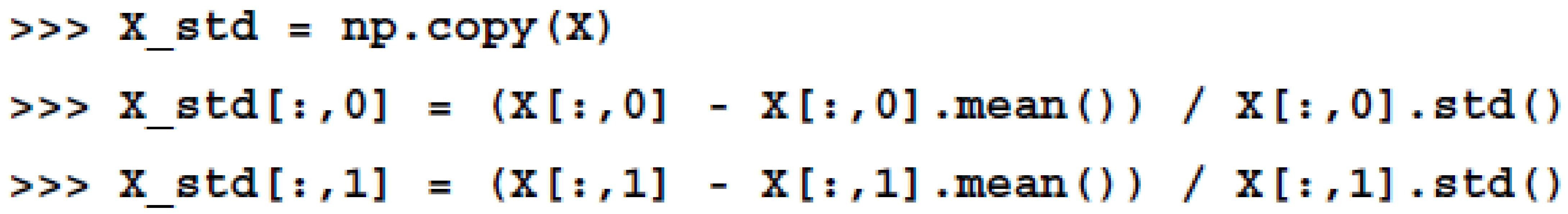
- To standardize the j^{th} feature, you need to subtract the sample mean u_j from every training sample and divide it by its standard deviation σj as given below:

$$x'_j = \frac{x_j - \mu_j}{\sigma_j}$$

- Here, x_j is a vector consisting of the j^{th} feature values of all training samples n

Standardization: Example

Given below is a sample NumPy code that uses NumPy mean and standard functions to standardize features from a sample data set *X* (*x0*, *x1*…) :

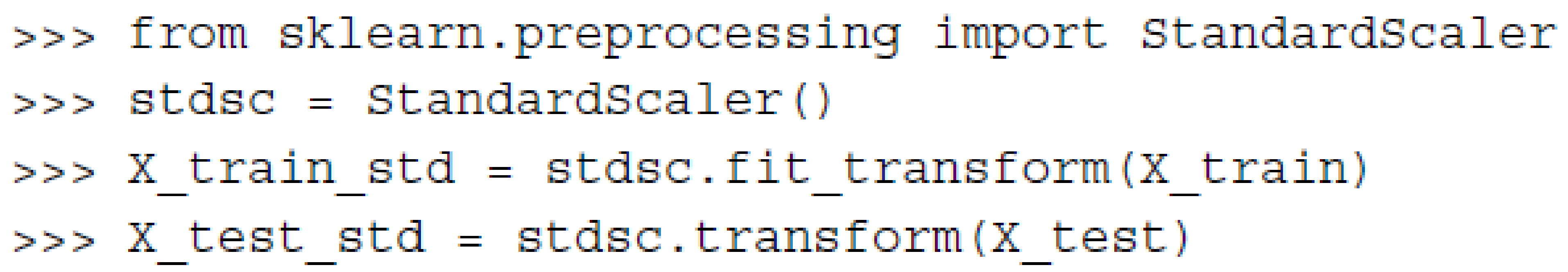


```
>>> X_std = np.copy(X)
>>> X_std[:,0] = (X[:,0] - X[:,0].mean()) / X[:,0].std()
>>> X_std[:,1] = (X[:,1] - X[:,1].mean()) / X[:,1].std()
```

NumPy is a Python library for all sorts of numerical computations.

Standardization: Example

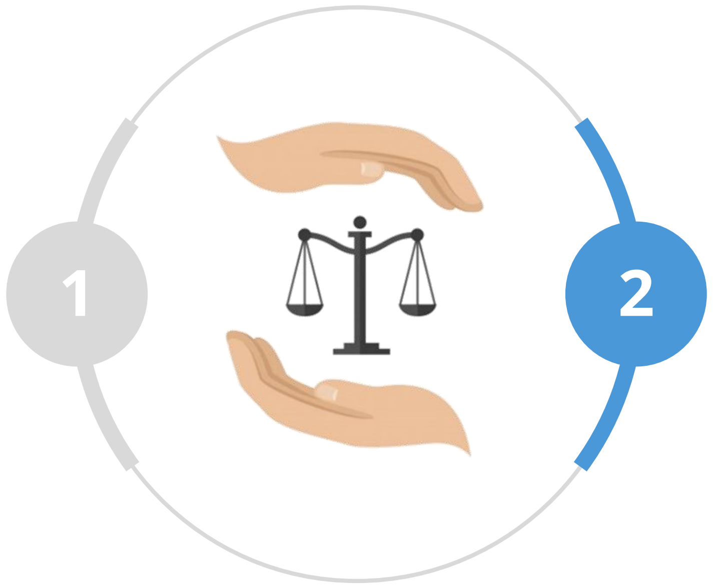
The ML library scikit-learn implements a class for standardization called StandardScaler, as demonstrated here:



```
>>> from sklearn.preprocessing import StandardScaler
>>> stdsc = StandardScaler()
>>> X_train_std = stdsc.fit_transform(X_train)
>>> X_test_std = stdsc.transform(X_test)
```

StandardScaler standardizes the various data features to a uniform scale to allow them to be compared and processed together.

Feature Scaling: Normalization



Normalization

- In most cases, normalization refers to rescaling of data features between 0 and 1, which is a special case of Min-Max scaling.

Normalization: Example

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

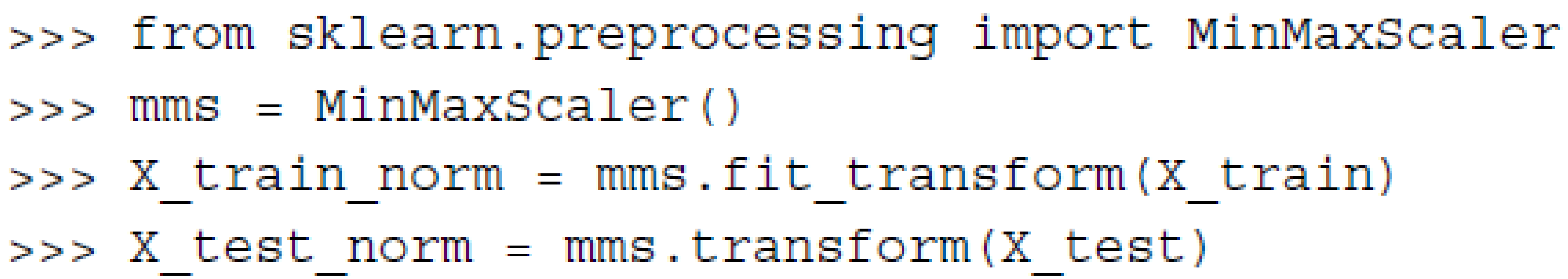
- In the given equation, subtract the min value for each feature from each feature instance and divide by the spread between max and min.

- In effect, it measures the relative percentage of distance of each instance from the min value for that feature.

Normalization: Example

The ML library scikit-learn has a MinMaxScaler class for normalization.

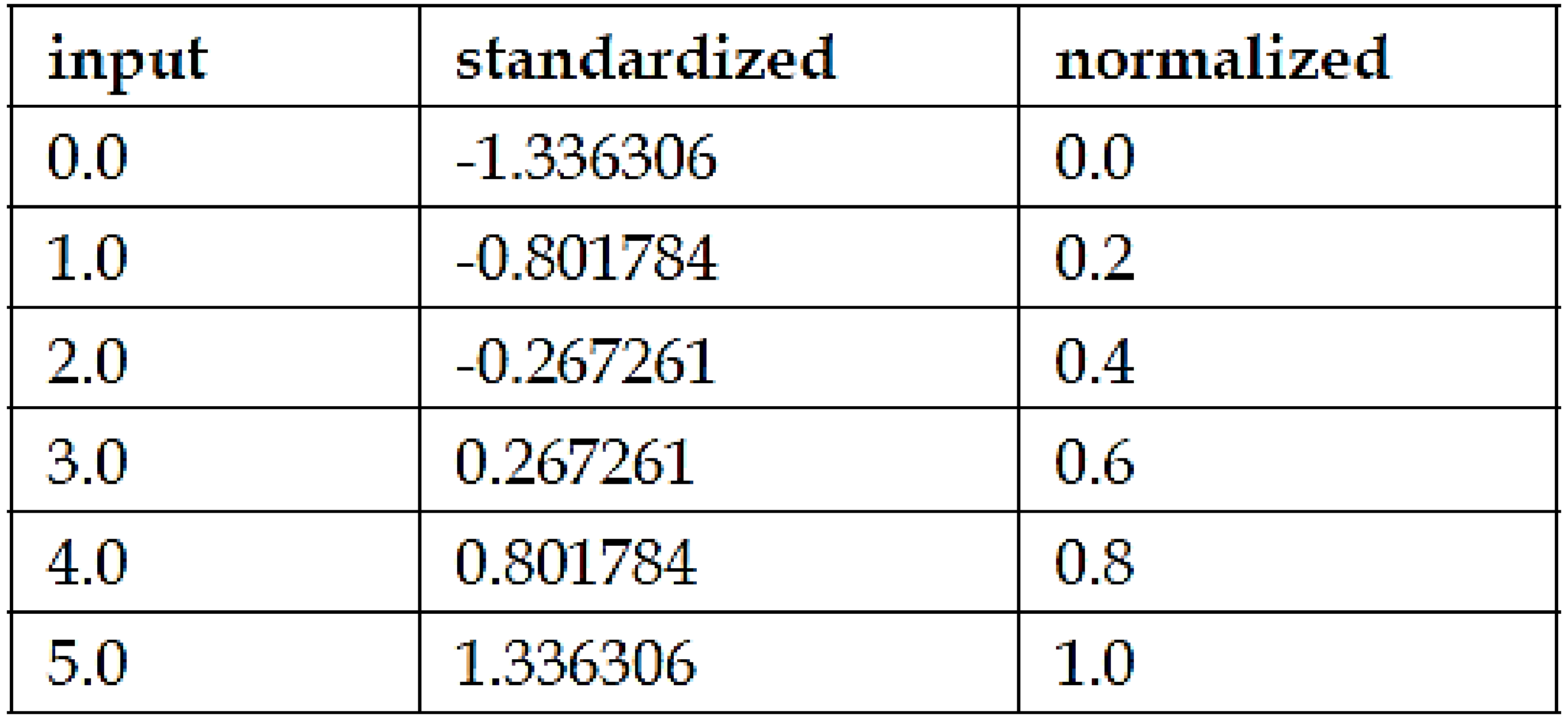
It implements normalization as explained on the previous slide.



```
>>> from sklearn.preprocessing import MinMaxScaler
>>> mms = MinMaxScaler()
>>> X_train_norm = mms.fit_transform(X_train)
>>> X_test_norm = mms.transform(X_test)
```

Difference between Standardization and Normalization

The following table shows the difference between standardization and normalization for a sample dataset with values from 1 to 5:



input	standardized	normalized
0.0	-1.336306	0.0
1.0	-0.801784	0.2
2.0	-0.267261	0.4
3.0	0.267261	0.6
4.0	0.801784	0.8
5.0	1.336306	1.0

Source Credit : “Python Machine Learning” by Sebastian Raschka

Demo

Data Preprocessing

Demonstrate methods to handle missing data, categorical data, and data standardization.

Data Preprocessing

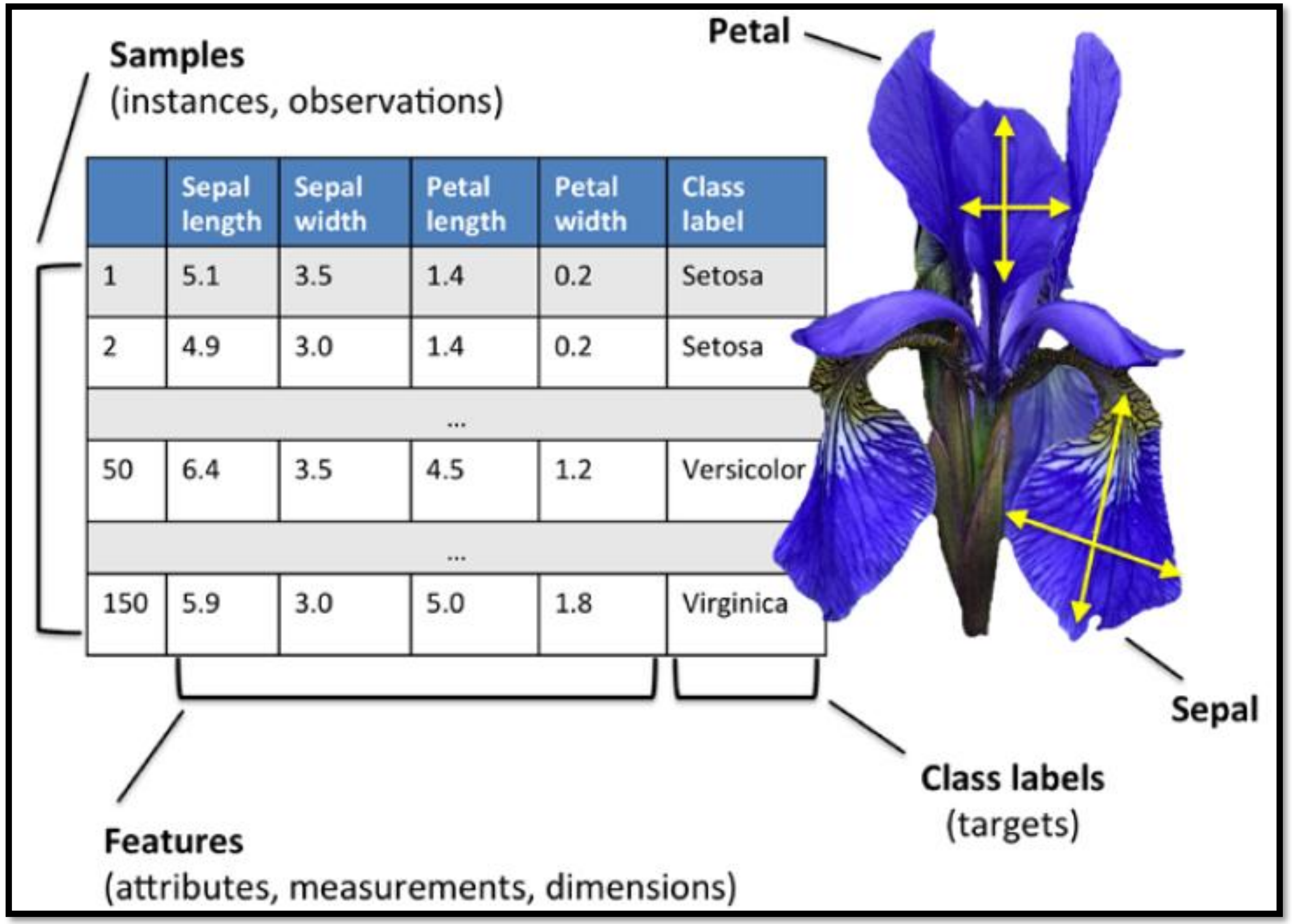
Topic 4: Datasets

Datasets in ML

- Machine Learning problems often need training or testing datasets.
- A dataset is a large repository of structured data.
- In many cases, it has input and output labels that assist in Supervised Learning.

IRIS Dataset

IRIS flower dataset is one of the popular datasets available online and widely used to train or test various ML algorithms.



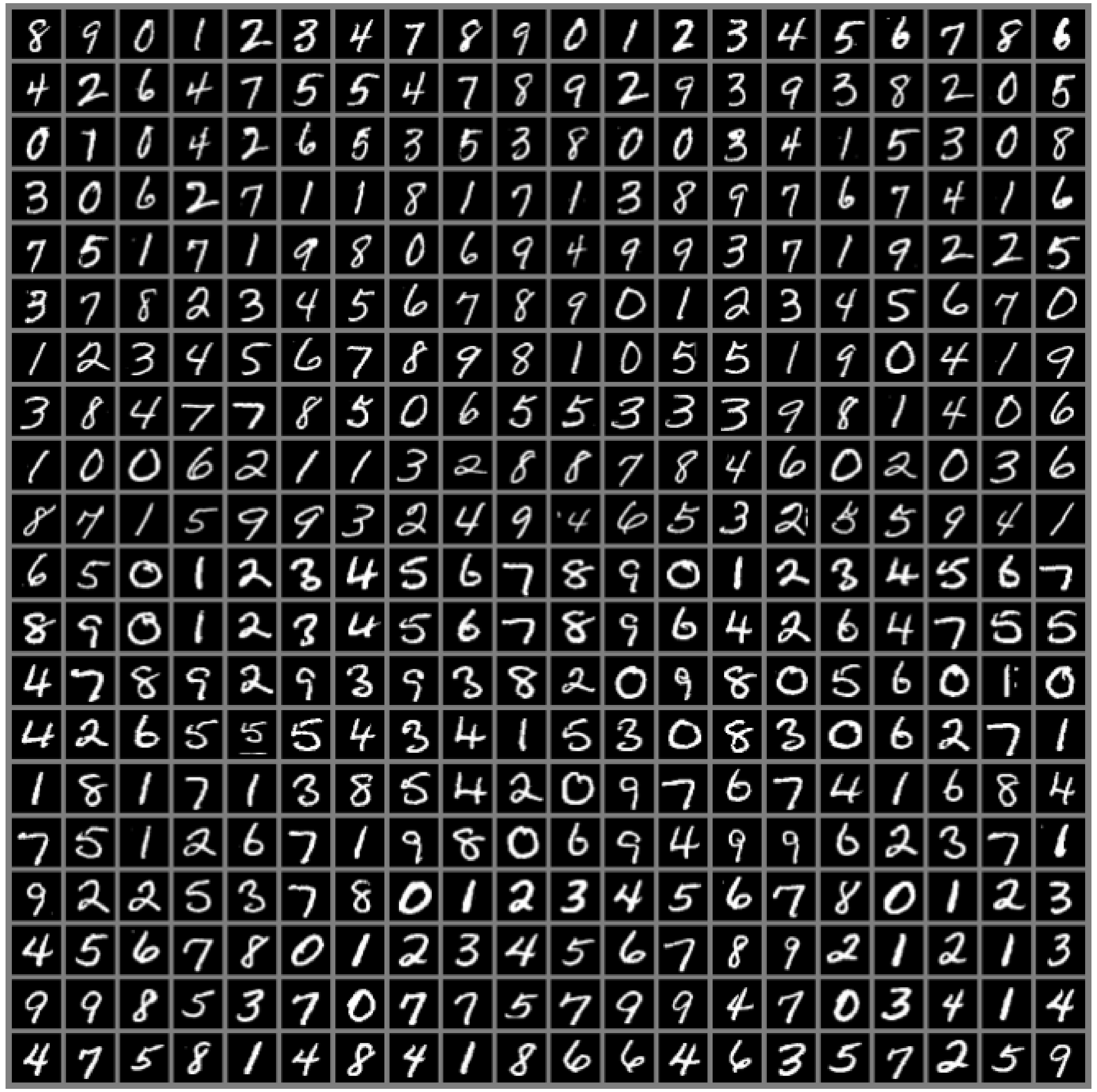
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Samples
(instances, observations)
Petal
Sepal
Class labels
(targets)
Features
(attributes, measurements, dimensions)

Source: "Python Machine Learning" by Sebastian Raschka

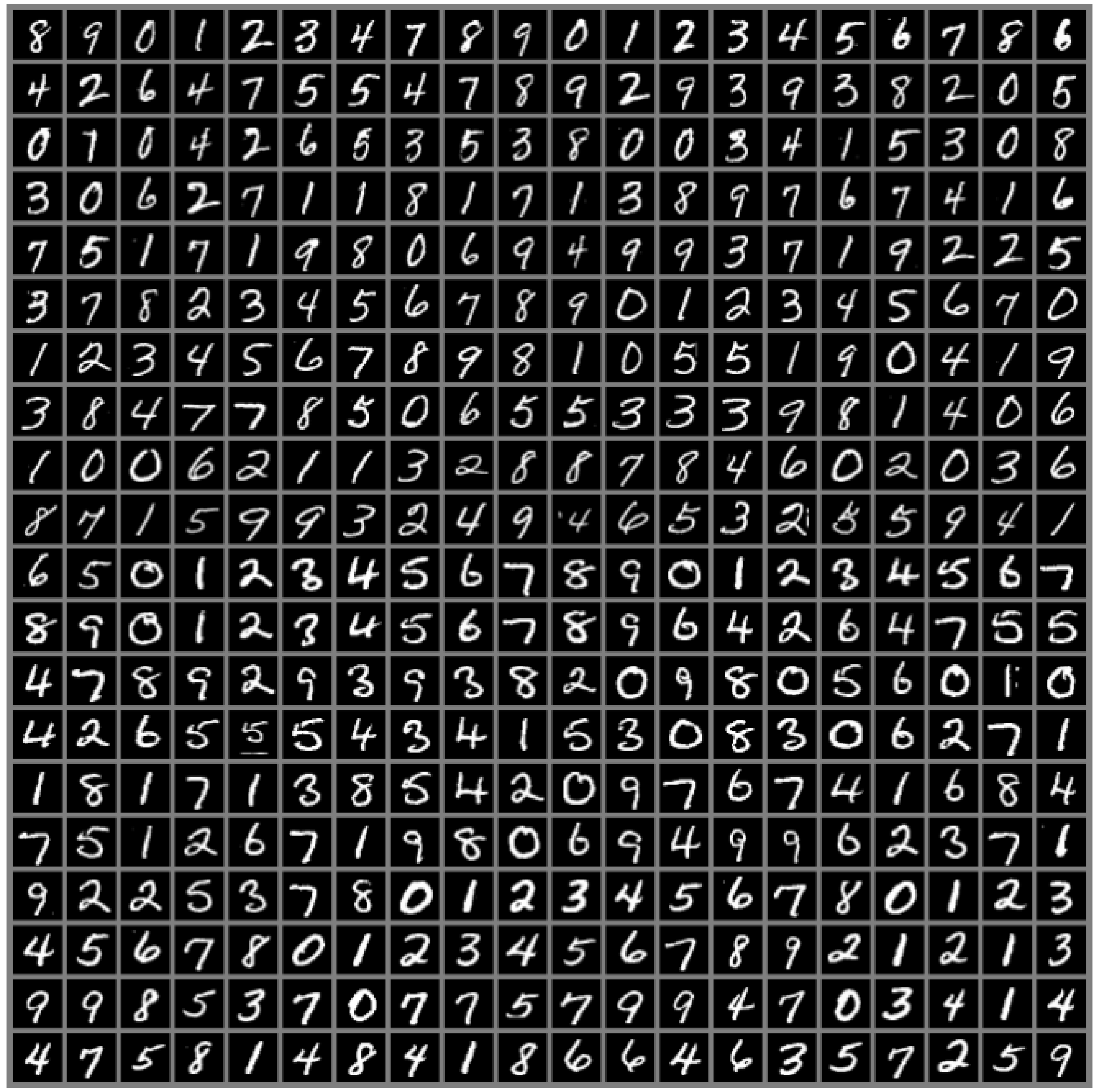
MNIST Dataset

Modified National Institute of Standards and Technology (MNIST) dataset is another popular dataset used in ML algorithms.



Source: "Deep Learning" book by Ian Goodfellow

MNIST Dataset

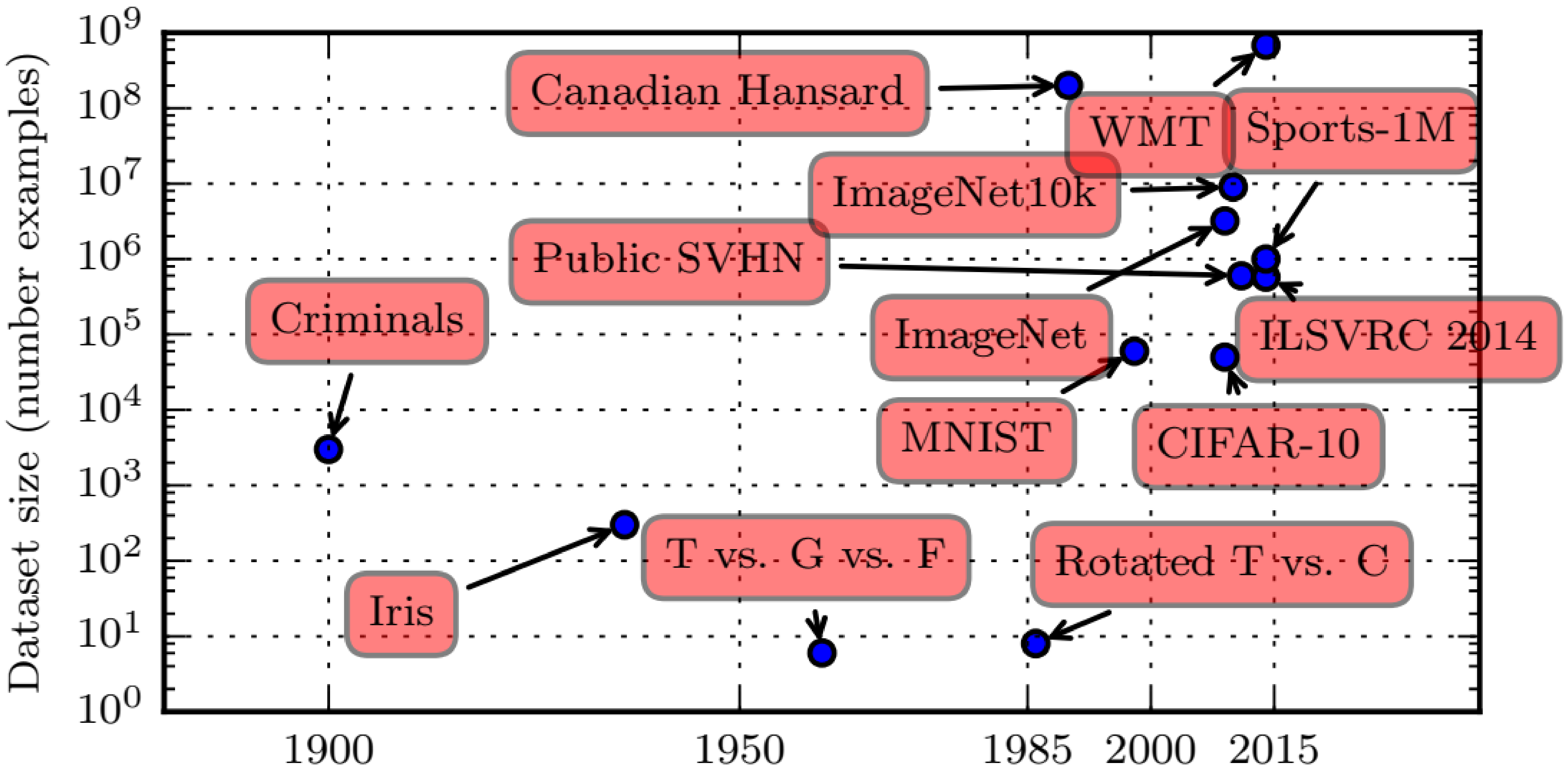


- National Institute of Standards and Technology (NIST) is a measurement standards laboratory, and a non-regulatory agency of US Department of Commerce.
- Modified NIST (MNIST) database is a collection of 70,000 handwritten digits and corresponding digital labels.
- The digital labels identify each of these digits from 0 to 9.
- It is one of the most common datasets used by ML researchers to test their algorithms.

Source: "Deep Learning" book by Ian Goodfellow

Growing Datasets

As the amount of data grows in the world, the size of datasets available for ML development also grows:

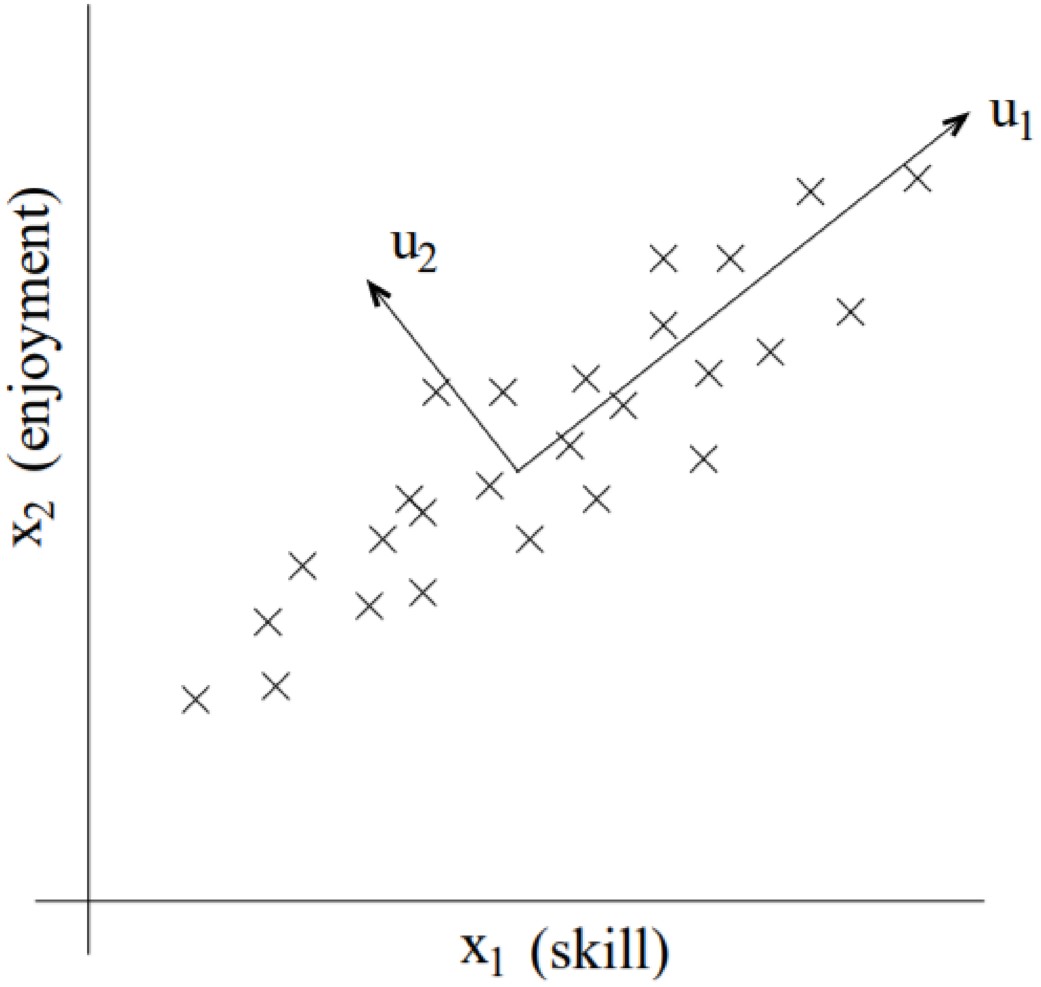


Source: "Deep Learning" book by Ian Goodfellow

Data Preprocessing

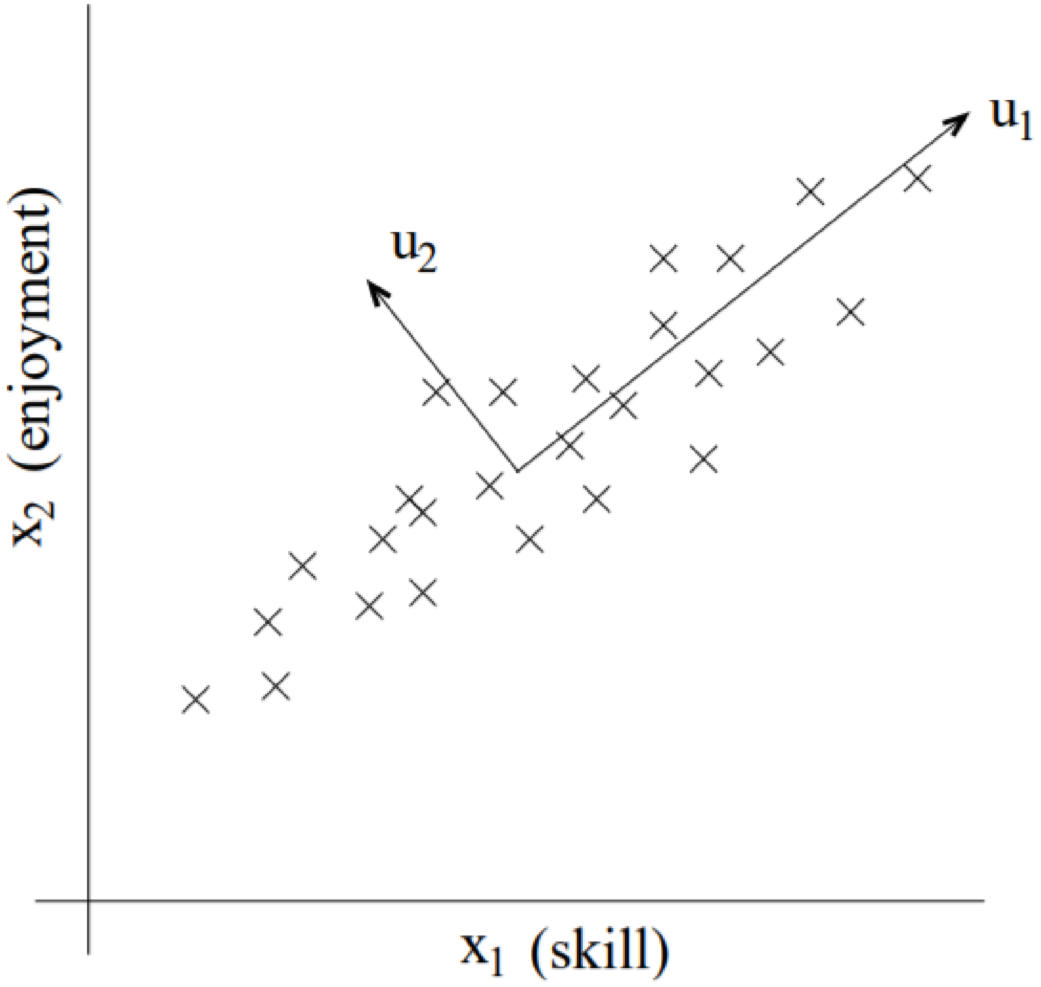
Topic 5: Dimensionality Reduction with Principal Component Analysis

Dimensionality Reduction



- Dimensionality reduction involves transformation of data to new dimensions in a way that facilitates discarding of some dimensions without losing any key information.
- Large scale problems bring about several dimensions that can become very difficult to visualize.
- Some of such dimensions can be easily dropped for a better visualization.
- Example: Car attributes might contain maximum speed in both units, kilometre per hour, and miles per hour. One of these can be safely discarded in order to reduce the dimensions and simplify the data.

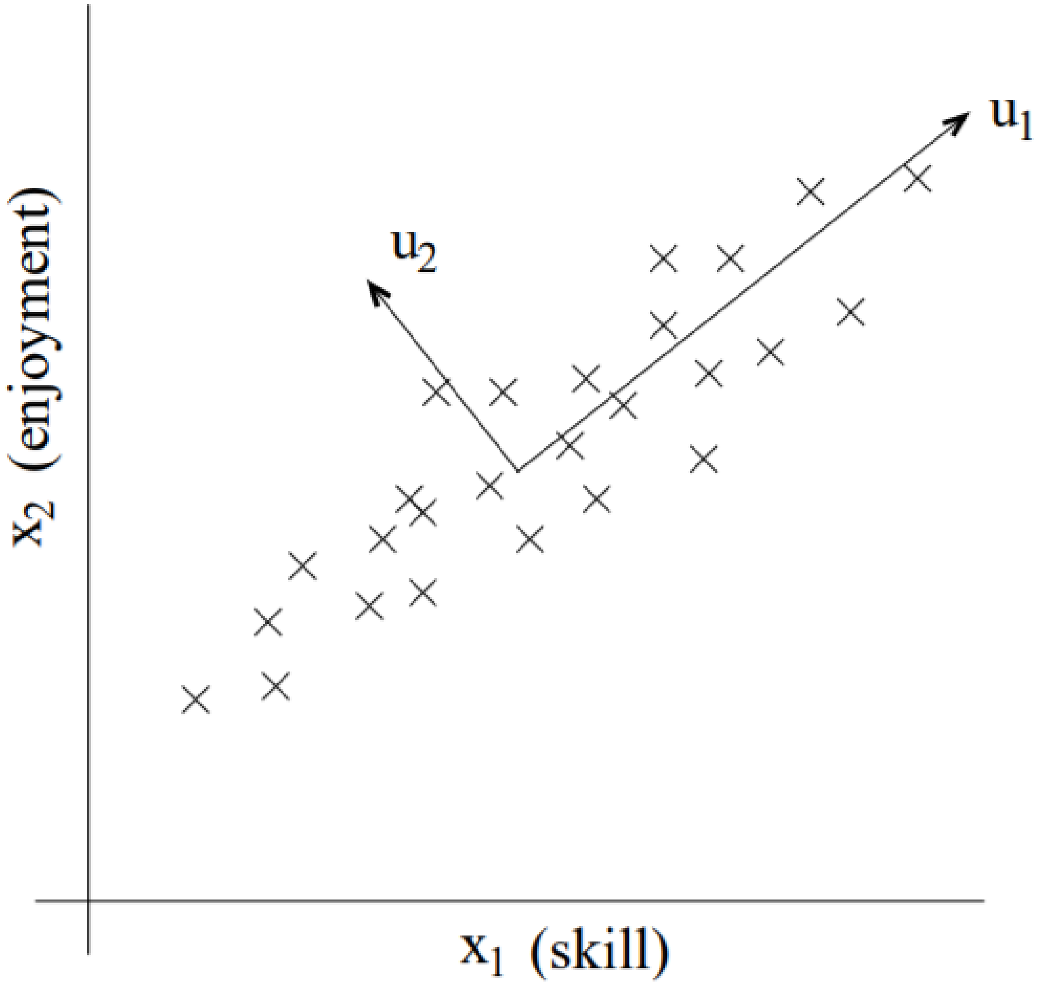
Dimensionality Reduction with Principal Component Analysis (PCA)



Pilots for an RC helicopter

- Principal component analysis (PCA) is a technique for dimensionality reduction that helps in arriving at better visualization models.
- Let's consider the pilots who like to fly radio-controlled helicopters. Assume x_1 = the piloting skill of the pilot and x_2 = passion to fly.
- RC helicopters are difficult to fly and only those students that truly enjoy flying can become good pilots. So, the two factors x_1 and x_2 are correlated, and this correlation may be represented by the piloting "karma" u_1 and only a small amount of noise lies off this axis (represented by u_2).

Dimensionality Reduction with Principal Component Analysis (PCA)



Pilots for an RC helicopter

- Most of the data lies along u_1, making it the principal component.

- Hence, you can safely work with u_1 alone and discard u_2 dimension. So, the 2D problem now becomes a 1D problem.

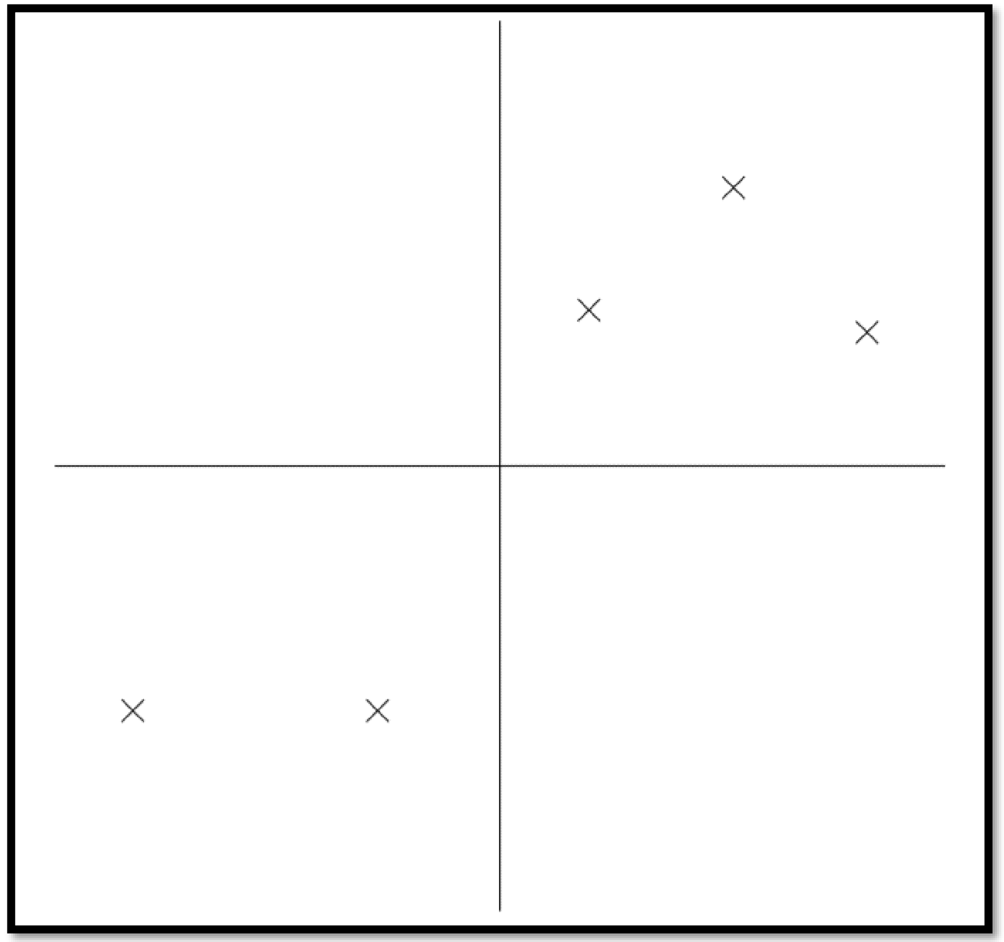
Principal Component Analysis (PCA)

1. Let $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$.
2. Replace each $x^{(i)}$ with $x^{(i)} - \mu$.
3. Let $\sigma_j^2 = \frac{1}{m}\sum_i (x_j^{(i)})^2$
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$.

- Before the PCA algorithm is developed, you need to preprocess the data to normalize its mean and variance.
- Steps 1 and 2 reduce mean of the data, and steps 3 and 4 rescale each coordinate to have unit variance. It ensures that different attributes are treated on the same scale.
- For instance, if x_1 was max speed in mph (taking values in high tens or low hundreds) and x_2 were number of seats (taking values 2-4), then this renormalization rescales the attributes to make them more comparable to each other.

Principal Component Analysis (PCA)

How do you find the axis of variation u on which most of the data lies?



- When you project this data to lie along the axis of the u unit vector, you would like to preserve most of it, such that its variance is maximized (which means most data is covered).
- Intuitively, the data starts off with some amount of variance (information).
- The figure shows this normalized data.

Principal Component Analysis (PCA)

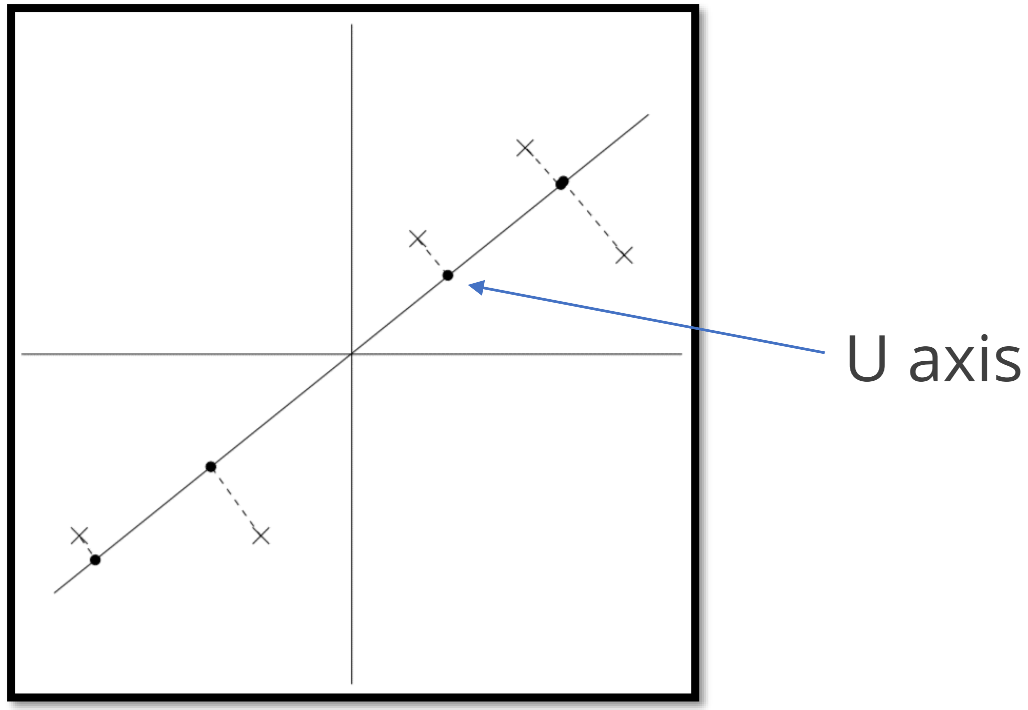


Figure A

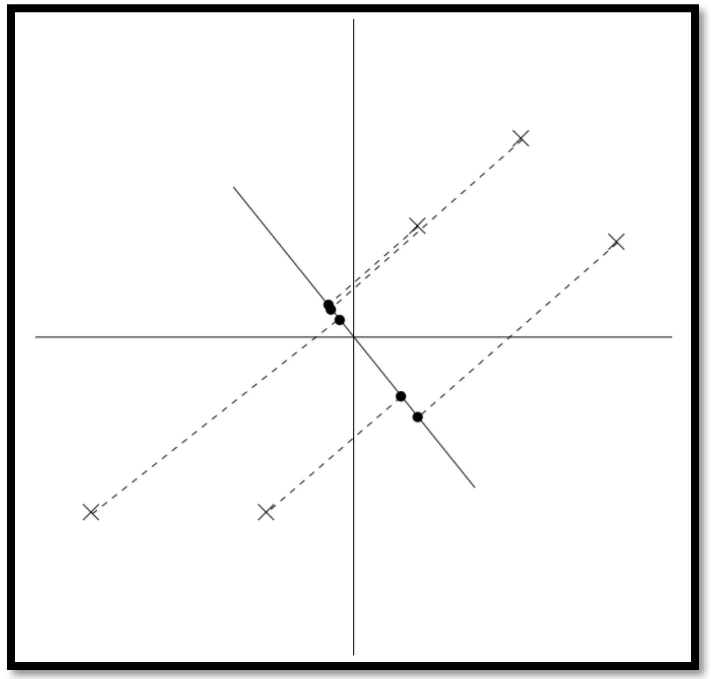


Figure B

- Let’s project data onto different u axes as shown in the charts given on the left.
- Dots represent the projection of data points on this line.
- In figure A, projected data has a large amount of variance, and the points are far from zero.
- In figure B, projected data has a low amount of variance, and the points are closer to zero.
- Hence, figure A is a better choice to project the data.

Principal Component Analysis (PCA)

- The length of projection of x on a unit vector u is given by x^{T}u. This also represents the distance of the projection of x from the origin.

- Hence, to maximize the variance of the projections, you can choose a unit length u:

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}\left(x^{(i)^T}u\right)^2 &= \frac{1}{m}\sum_{i=1}^{m} u^T x^{(i)} x^{(i)^T} u \\ &= u^T\left(\frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^T}\right)u.\end{aligned}$$

- You get the principal Eigenvector* of $\Sigma = \frac{1}{m}\sum_{i=1}^{m} x^{(i)} x^{(i)^T}$

- It is also known as the covariance matrix of the data (assuming that it has zero mean).

Eigenvector of a matrix is a vector which when multiplied to the matrix changes only the scale of the vector.

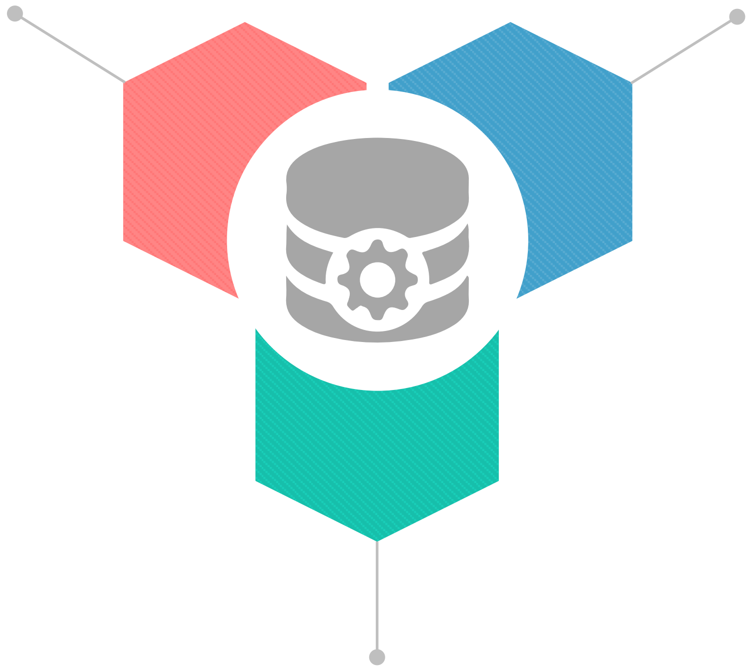
Principal Component Analysis (PCA)

- Generally, if you need to project data onto the *k*-dimensional subspace ($k < n$), you choose $u_1, u_2 \ldots u_k$ to be the top *k* Eigenvectors of $\sum$.
- All the u_i now form a new orthogonal basis for the data.
- Then, to represent $x^{(i)}$ in this new basis, you need to compute the corresponding vector:

$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k$$

- The vector y(i) is a lower k-dimensional approximation of x(i). This is known as the dimensionality reduction.
- The vectors u1,u2...uk are called the first k principal components of the data.

Applications of PCA



Noise reduction

PCA can eliminate noise or non-critical aspects of the data set to reduce complexity. Also, during image processing or comparison, image compression can be done with PCA, eliminating the noise such as lighting variations in face images.

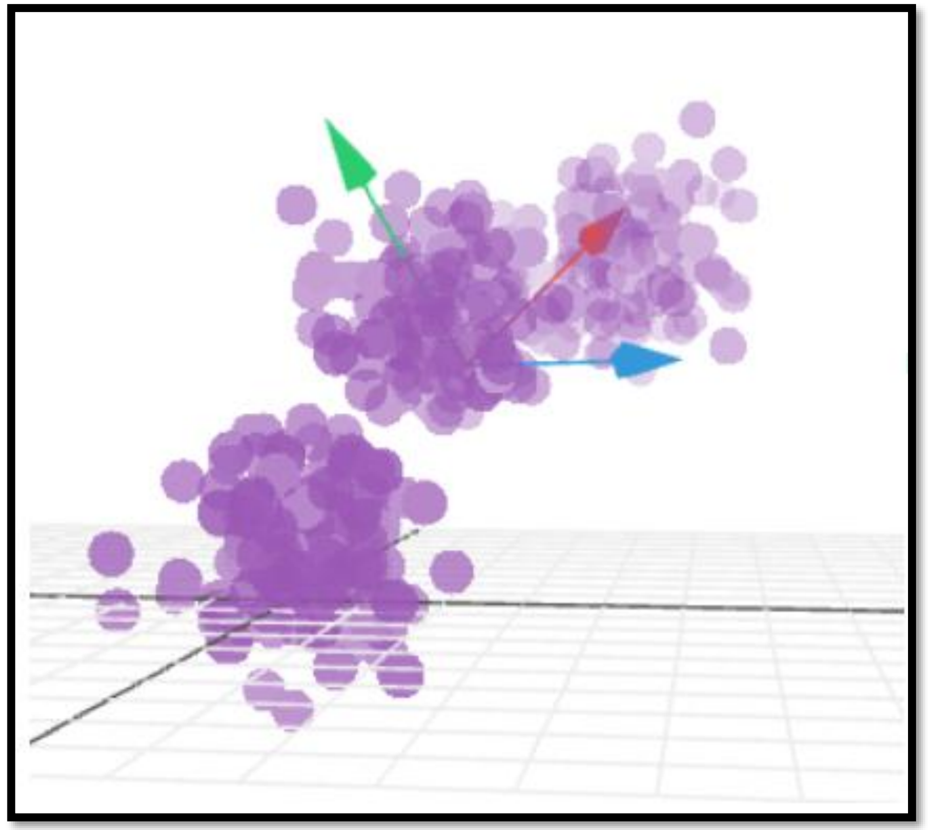
Compression

It is used to map high dimensional data to lower dimensions. For example, instead of having to deal with multiple car types (dimensions), we can cluster them into fewer types.

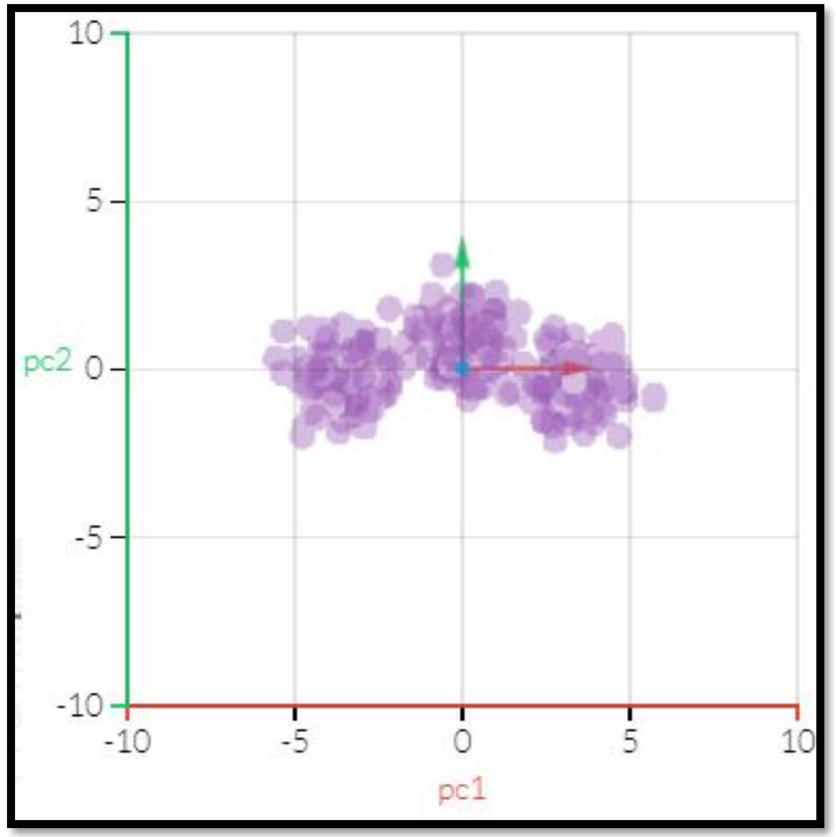
Preprocess

It reduces data dimensions before running a supervised learning program and saves on computations as well as reduces overfitting.

PCA: 3D to 2D Conversion



3D Data



After PCA, one finds only two dimensions being important—Red and Green that carry most of the variance. The blue dimension has limited variance, and hence it is eliminated.

Demo

Dimensionality Reduction

Demonstrate dimensionality reduction for data dimensions from 3D to 2D.

Key Takeaways

- ✓ Data preparation allows simplification of data to make it ready for Machine Learning and involves data selection, filtering, and transformation.
- ✓ Data must be sufficient, representative of real world data, and of high quality.
- ✓ Feature Engineering helps in selecting the right features and extracting the most relevant features.
- ✓ Feature scaling transforms features to bring them on a similar scale, in order to make them comparable in ML routines.
- ✓ Dimensionality Reduction allows reducing dimensions in datasets to simplify ML training.

?
Quiz

QUIZ

1

What are the two techniques of feature scaling?

a. Standardization and Normalization

b. Composition and Decomposition

c. Rooting and Cleaning

d. Climbing and Descending

QUIZ

1

What are the two techniques of feature scaling?

a. Standardization and Normalization

b. Composition and Decomposition

c. Rooting and Cleaning

d. Climbing and Descending

The correct answer is **a.**

The two common techniques to scale features are standardization and normalization.

QUIZ 2

What is Principal Component Analysis?

a. Analysis of data anomalies

b. A way to reduce number of dimensions in the data

c. A technique to scale features

d. None of the above

QUIZ

2

What is Principal Component Analysis?

a. Analysis of data anomalies

b. A way to reduce number of dimensions in the data

c. A technique to scale features

d. None of the above

The correct answer is **b.**

Principal component analysis (PCA) is a technique for dimensionality reduction that helps in arriving at better visualization models.

Hands-on Assignments

Demo File	Assignment	What it demonstrates?
DataPreprocessing.py	Review the training dataset (Excel file). Note that weight is missing for the fifth and eighth row. What is the value computed by the imputer for these two missing rows?	Demonstrates methods to handle missing data, categorical data and data standardization.
	In the tutorial code, find the call to the Imputer class. Replace strategy parameter from 'mean' to 'median' and execute it again. What is the new value assigned to blank fields Weight and Height for the two rows?	Modify the code and strategy parameter and check the output.
	In the code snippet given below in the tutorial, why does the array X has 5 columns instead of 3 columns as before?	Observe the code and find the reason for the change.

Hands-on Assignments

Demo File	Assignment	What it demonstrates?
DimRed.py	What does the hyperplane shadow represent, in the PCA output chart on random data?	Demonstrates how to reduce data dimensions from 3D to 2D.
	What is the reconstruction error after PCA transformation? Give interpretation.	Find the error after modifying the code.

This concludes “Data Preprocessing.”

The next lesson is “Math Refresher.”