CM2604 Machine Learning

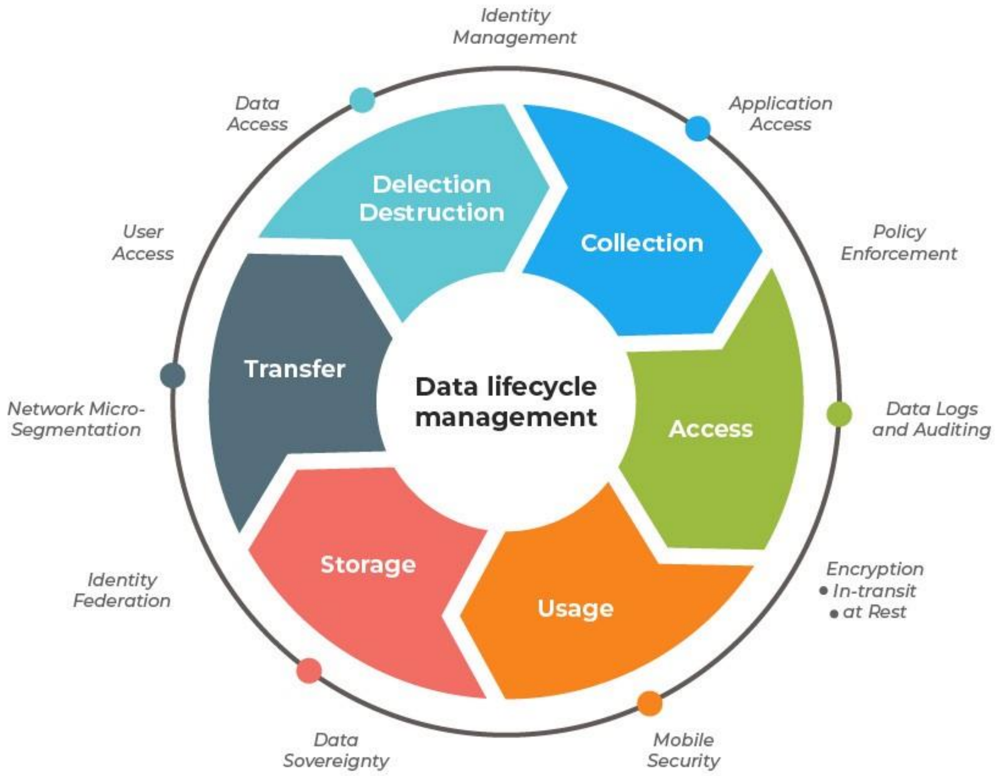
Nature of Data and Dimensionality Reduction

Week 02 | Prasan Yapa

ROBERT GORDON UNIVERSITY ABERDEEN
TEF Gold
INFORMATICS INSTITUTE OF TECHNOLOGY

Overview

- Data in ML
- Data Handling
- Data Cleaning
- Features
- Preparing Dataset
- Dimensionality Reduction

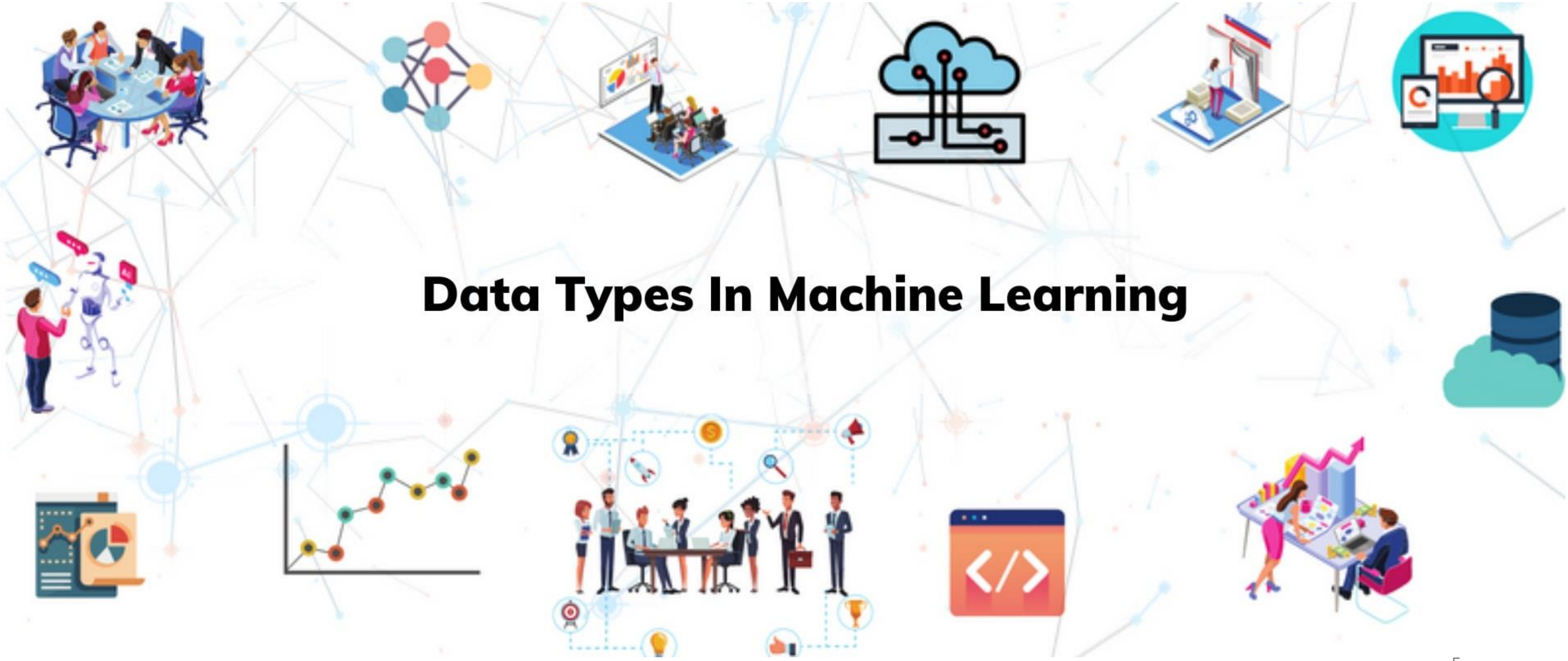


Data in ML

Data in ML

- Any unprocessed fact, value, text, sound, or picture that is not being interpreted and analyzed.
- Most important part of all Data Analytics, Machine Learning, Artificial Intelligence etc.
- Without data, we can’t train any model and all modern research and automation will go in vain.
- Big Enterprises are spending lots of money just to gather as much certain data as possible.

Data in ML



Different Forms of Data

- Numeric Data
  - If a feature represents a characteristic measured in numbers , it is called a numeric feature.
- Categorical Data
  - A categorical feature is an attribute that can take on one of the limited, and usually fixed number of possible values based on some qualitative property.
- Ordinal Data
  - This denotes a nominal variable with categories falling in an ordered list. Examples include clothing sizes such as small, medium, and large.

Properties of Data

- Volume: Scale of Data.
- Variety: Different forms of data.
- Velocity: Rate of data streaming and generation.
- Value: Meaningfulness of data in terms of information.
- Veracity: Certainty and correctness in data.

Some Facts on Data

- As compared to 2005, 300 times i.e., 40 Zettabytes (1ZB=10^21 bytes) of data will be generated by 2020.
- By 2011, the healthcare sector has a data of 161 Billion Gigabytes.
- 400 Million tweets are sent by about 200 million active users per day.
- Each month, more than 4 billion hours of video streaming is done by the users.
- 30 Billion different types of content are shared every month by the user.
- It is reported that about 27% of data is inaccurate and so 1 in 3 business idealists or leaders don’t trust the information on which they are making decisions.

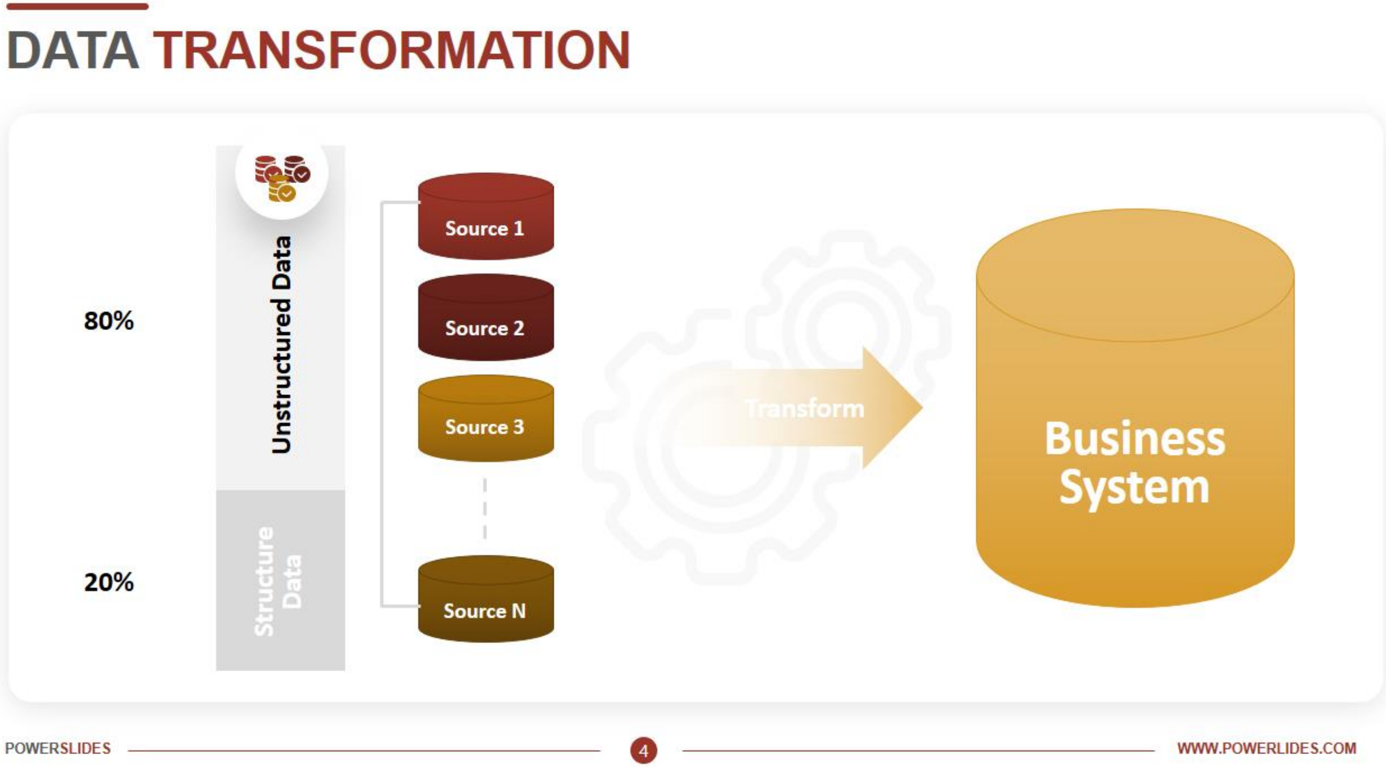
Data Handling

Data Cleaning

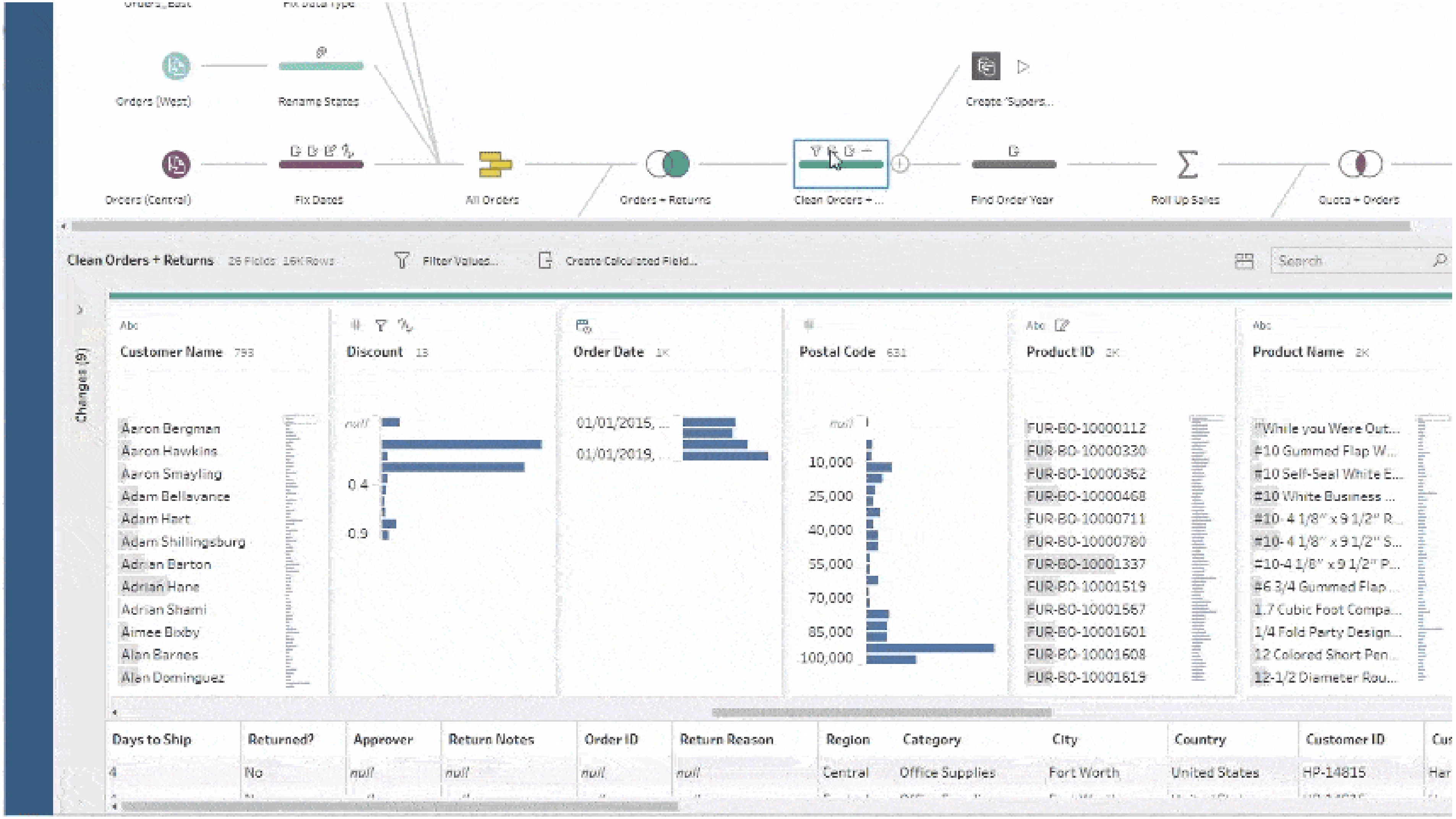
- Process of fixing or removing:
  - Incorrect
  - Corrupted
  - Incorrectly formatted
  - Duplicate
  - Incomplete data within a dataset
- If data is incorrect, outcomes and algorithms are unreliable.
- There is no one absolute way to prescribe the exact steps in the data cleaning process.

Data Cleaning Vs Data Transformation

- DT is the process of converting data from one format or structure into another.



How to Clean Data



Step 1: Removing Duplicate or Irrelevant Observations

- Duplicate observations will happen most often during data collection.
- De-duplication is one of the largest areas to be considered in this process.
- Irrelevant observations are when you notice observations that do not fit into the specific problem.
- Creating a more manageable and more performant dataset.

Step 2: Fixing Structural Errors

- Strange naming conventions.
- Typos.
- Incorrect capitalization.
- Inconsistencies can cause mislabeled categories or classes.
  - “N/A” and “Not Applicable” both appear, but they should be analyzed as the same category.

Step 3: Filtering Unwanted Outliers

- There will be one-off observations where do not appear to fit within the data.
- Just because an outlier exists, doesn’t mean it is incorrect.
- This task is needed to determine the validity.
- If an outlier proves to be irrelevant for analysis or is a mistake, consider removing it.

Step 4: Handling Missing Data

- You can’t ignore missing data because of many algorithms.
- There are a couple of ways to deal with missing data.
  - Observations that have missing values can be dropped, but doing this will drop or lose information, so be mindful of this before removing it.
  - Secondly you can input missing values based on other observations.
  - You might alter the way the data is used to effectively navigate null values.

Step 5: Validation

- Does the data make sense?
- Does the data follow the appropriate rules for its field?
- Does it prove or disprove your working theory, or bring any insight to light?
- Can you find trends in the data to help you form your next theory?
- If not, is that because of a data quality issue?

Stemming Vs Lemmatization in Data Science

- Stemming
  - Process of removing last few characters from a word, often leading to incorrect meanings and spelling.

```
import nltk
from nltk.stem import PorterStemmer
```

```
words=['done','doing','studying','identify','this']
ps=PorterStemmer()
for word in words:
    print(f"{word}: {ps.stem(word)}")
```

```
done: done
doing: do
studying: studi
identify: identifi
this: thi
```

- Lemmatization
  - Considers the context and converts the word to its meaningful base form, which is called Lemma.

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

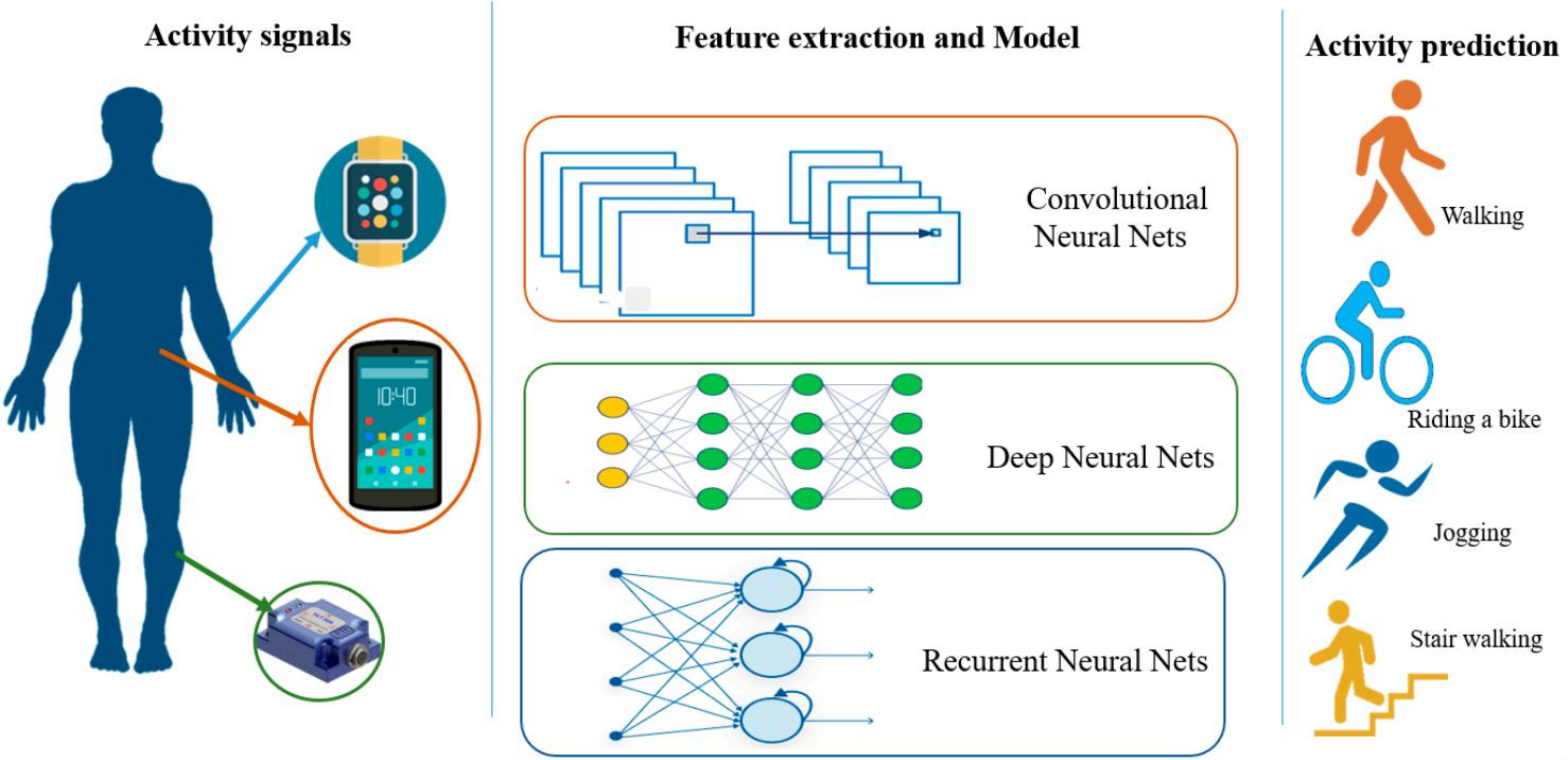
```
words=['feet','dogs','children','identify','this']
for word in words:
    print(f"{word}: {lemmatizer.lemmatize(word)}")
```

```
feet: foot
dogs: dog
children: child
identify: identify
this: this
```

Variables (Features)

Feature

- A feature is a measurable property of the object you're trying to analyze.
- Features are the basic building blocks of datasets.
- Features are also referred to as "variables" or "attributes."
- Different business problems do not necessarily require the same features for understanding the business goals.


Activity signals
Feature extraction and Model
Activity prediction
Convolutional
Neural Nets
Deep Neural Nets
Recurrent Neural Nets
10:40
Walking
Riding a bike
Jogging
Stair walking

Types of Variables

- Independent Variables Vs Dependent Variables.
- If a variable's value changes when another of the variables change then it is dependent, otherwise independent.
- Independent variables are the input for a process that is being analyzed.
- Dependent variables are the output of the process.

Types of Variables

raw_data - DataFrame

Index	Age	Education	Income	Marital Status	Purchased
0	36–55	Masters	High	Single	1
1	18–35	High School	Low	Single	0
2	36–55	nan	High	Single	1
3	18–35	PhD	Low	nan	1
4	nan	High School	Low	Single	1
5	55+	High School	High	Married	0
6	55+	High School	nan	Married	1
7	nan	High School	nan	Married	1
8	55+	High School	High	Married	1
9	< 18	Masters	Low	Single	0

Independent variables

Dependent variables

Format | Resize | Background color | Column min/max | Cancel | OK

Preparing Your Dataset

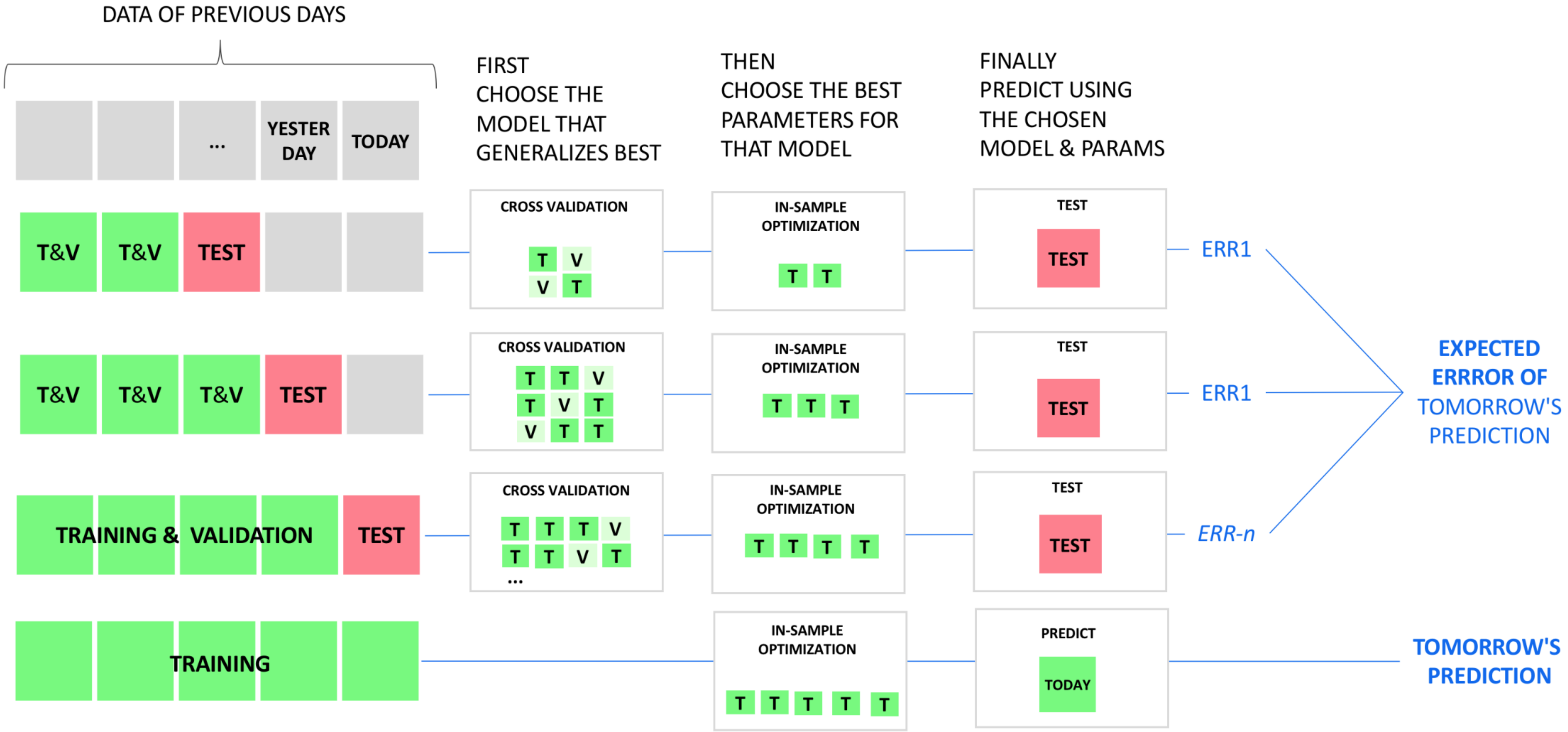
Preparing Your Dataset for ML

- Articulate the problem early.
- Establish data collection mechanisms.
- Check your data quality.
- Format data to make it consistent.
- Reduce data.
- Complete data cleaning.
- Create new features out of existing ones.
- Rescale data.

Training, Validation & Testing Data

- Training data
  - Responsible for building up the machine learning algorithm.
  - The data scientist feeds these data, which corresponds to an output.
- Validation data
  - During training, validation data infuses new data into the model that it hasn't evaluated before.
- Test data
  - After the model is built, testing data once again validates that it can make accurate predictions.
  - If training and validation data include labels, the testing data should be unlabeled.

PREDICT THE NEXT DAY USING PREVIOUS DAYS DATA



Overfitting Vs Underfitting

- Overfitting
  - Good performance on the training data, poor generalization to other data.
  - A high accuracy measured on the training set.
- Underfitting
  - Poor performance on the training data and poor generalization to other data.
  - Reduces the accuracy and produces unreliable predictions.

Overfitting and Underfitting are two of the main reasons machine learning models have poor performance.

Fixing **overfitting:**

- Simplify the model (fewer parameters)
- Simplify training data (fewer attributes)
- Constrain the model (regularization)
- Use cross-validation
- Use Early stopping
- Build an ensemble
- Gather more data

Fixing **underfitting:**

- More complex model (more parameters)
- Increase number of features
- Feature engineer should help
- Un-constrain the model (no regularization)
- Reduce noise on the data
- Train for longer

Dimensionality Reduction

Dimensionality Reduction

- Reducing the number of random variables by obtaining a set of principal variables.
- Eliminating redundancy and reducing the possibility of the model overfitting.
- Two components:
  - Feature Selection - Smaller subsets of features are chosen by filtering, wrapping or embedding.
  - Feature Extraction - Reducing the number of dimensions in a dataset in order to model variables.

Principal Component Analysis (PCA)

- PCA is used to compress a dataset onto a lower-dimensional feature subspace.
- Feature selection finds a subset of features while PCA produces a smaller new set.
- PCA helps us to identify patterns in data based on the correlation between features.
- PCA aims to find the directions of maximum variance in high-dimensional data.

Applications of PCA in Machine Learning

- PCA is used to visualize multidimensional data.
- It is used to reduce the number of dimensions in healthcare data.
- PCA can help resize an image.
- It can be used in finance to analyze stock data and forecast returns.
- PCA helps to find patterns in the high-dimensional datasets.

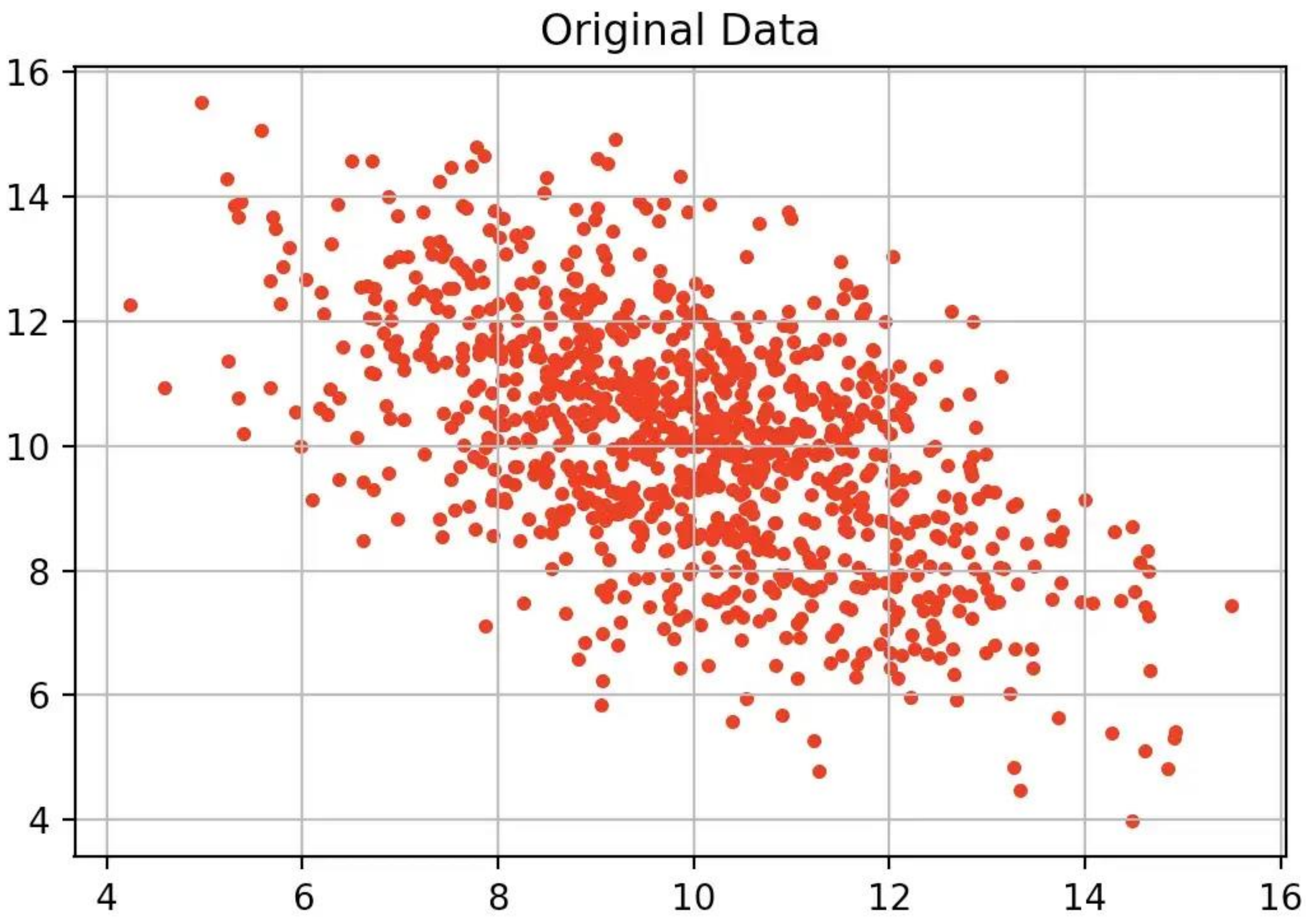
How does PCA Work? – Step 1

- Create data by randomly drawing samples.
- Let's start with 2-dimensional data.

```
import numpy as np
import matplotlib.pyplot as plt
import numpy.random as rnd

# Create random 2d data
mu = np.array([10,13])
sigma = np.array([[3.5, -1.8], [-1.8,3.5]])

print("Mu ", mu.shape)
print("Sigma ", sigma.shape)

# Create 1000 samples using mean and sigma
org_data = rnd.multivariate_normal(mu, sigma, size=(1000))
print("Data shape ", org_data.shape)
```

How does PCA Work? – Step 1

- Mean, mu is: [10, 13].
- Covariance matrix, sigma is: [[3.5 -1.8], [-1.8, 3.5]].
- Here is a scatter plot of data:

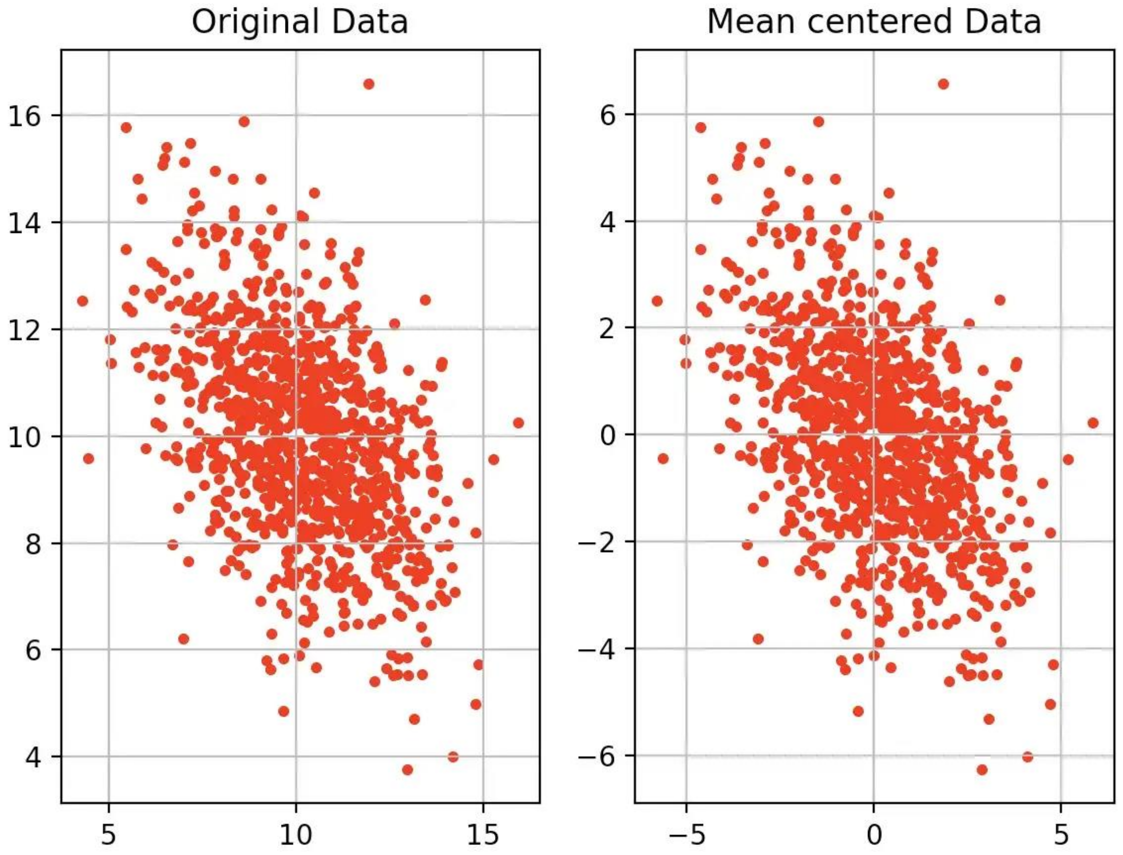


How does PCA Work? – Step 2

- Normalizing data.
- Normalization is done to scale up all features.
- Mean centering is done ensuring the first PC is in the direction of maximum variance.
- Mean centering is done by subtracting mean from all features or channels.

How does PCA Work? – Step 2

```
# Subtract mean from data
mean = np.mean(org_data, axis= 0)
print("Mean ", mean.shape)
mean_data = org_data - mean
print("Data after subtracting mean ", org_data.shape, "\n")
```



Scatter Plot of Original Data (Left) and Mean Centered Data (Right)

How does PCA Work? – Step 3

- Computing the covariance of all features dimensions.
- Every covariance matrix is symmetric and positive semi-definite.
- It has orthogonal eigen vectors.
- The size of the covariance matrix will be (2 x 2).

```
# Compute covariance matrix
cov = np.cov(mean_data.T)
cov = np.round(cov, 2)
print("Covariance matrix ", cov.shape, "\n")
```

How does PCA Work? – Step 4

- Computing eigen vectors of the covariance matrix.
- The number of eigen vectors will be the same as the number of features.
- Each eigen vector represents a direction of variance.

```
# Perform eigen decomposition of covariance matrix
eig_val, eig_vec = np.linalg.eig(cov)
print("Eigen vectors ", eig_vec)
print("Eigen values ", eig_val, "\n")
```

How does PCA Work? – Step 4

- The eigen vector corresponding to the largest eigen value will give the direction of maximum variance.
- This is the first principal component.
- Then, the eigen vector corresponding to the 2^{nd} largest eigen value will give the direction of the second largest variance.
- This is the second principal component. And, so on.

```
# Sort eigen values and corresponding eigen vectors in descending order
indices = np.arange(0,len(eig_val), 1)
indices = ([x for _,x in sorted(zip(eig_val, indices))])[::-1]
eig_val = eig_val[indices]
eig_vec = eig_vec[:,indices]
print("Sorted Eigen vectors ", eig_vec)
print("Sorted Eigen values ", eig_val, "\n")
```

How does PCA Work? – Step 5

- Computing the explained variance and select N components.
- The optimal way is to compute the explained variance of each feature.
- Computing explained variance by dividing the eigen values by the sum of all eigen values.
- Then, we take the cumulative sum of all eigen values.

```
# Get explained variance
sum_eig_val = np.sum(eig_val)
explained_variance = eig_val/ sum_eig_val
print(explained_variance)
cumulative_variance = np.cumsum(explained_variance)
print(cumulative_variance)
```

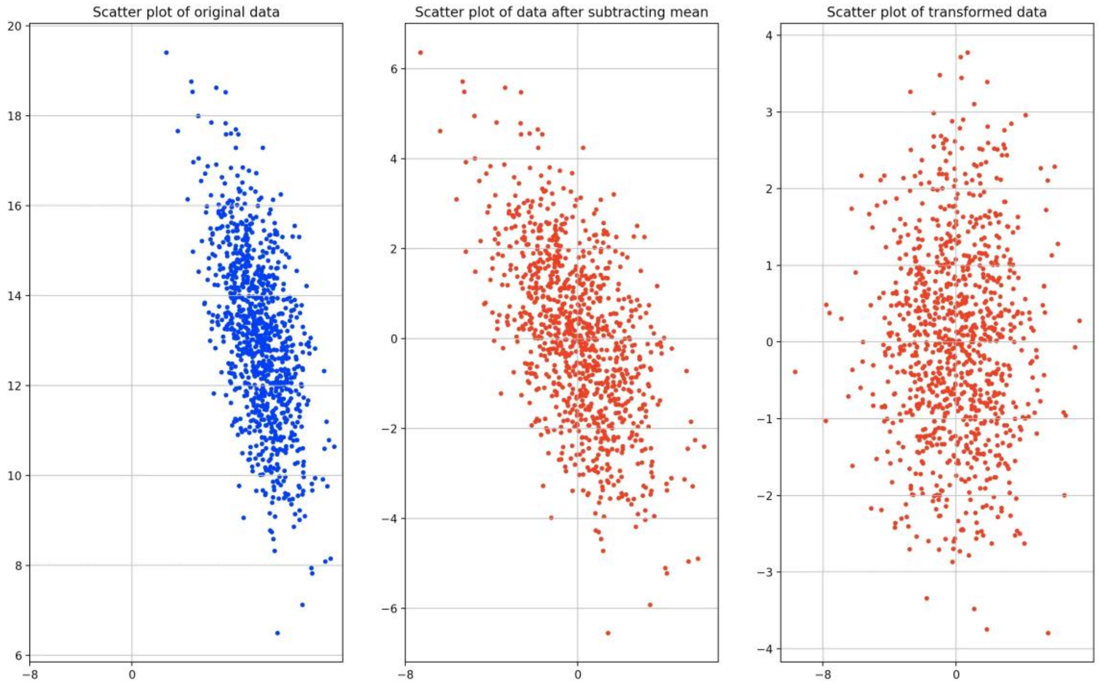
How does PCA Work? – Step 5

- The eigen values here are: [5.50, 1.72].
- The sum of eigen values is: 7.22.
- Explained variance is: [0.76, 0.23].
- Cumulative explained variance is: [0.76, 0.99].
- When we have higher dimensional data, we usually take *k* components in such a way that we get an explained variance of 0.95 or more.

How does PCA Work? – Step 6

- The dot product of data will be taken with the eigen vectors to get projections of the data in the direction of these eigen vectors.

```
# Take transpose of eigen vectors with data
pca_data = np.dot(mean_data, eig_vec)
print("Transformed data ", pca_data.shape)
```

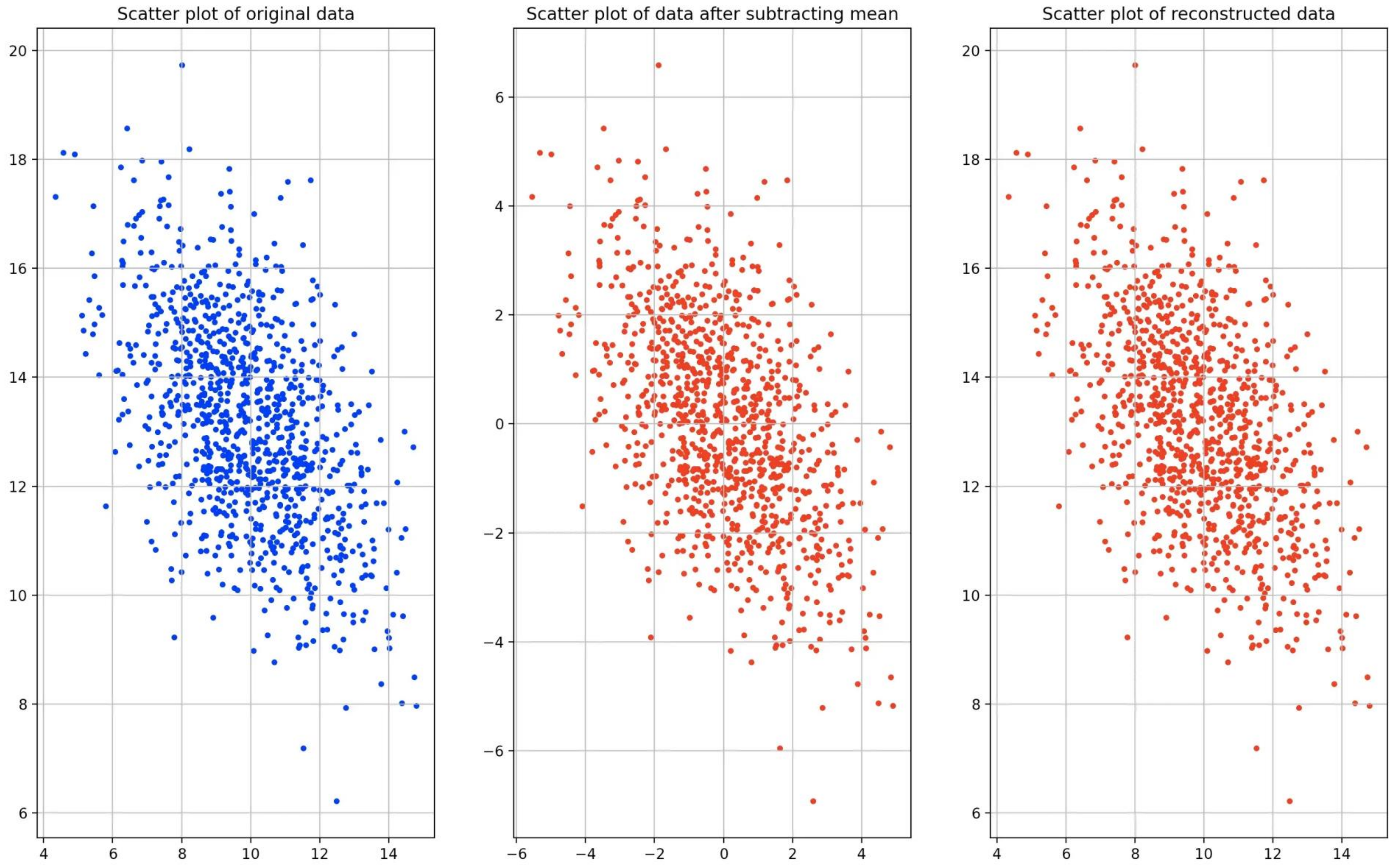


In the scatter plot, we can see that after PCA, the y-axis is the direction of maximum variance.

How does PCA Work? – Step 7

- Inverting PCA and Reconstruct original data.
- Reconstructing the original data by taking the dot product of transpose of eigen vectors.
- Remember to add the mean as well *(mean was subtracted from the data at the beginning to center the data)*.
- When we take the dot product of eigen vectors with itself, we get an identity matrix.

How does PCA Work? – Step 7



How does PCA Work? – Step 7

- We can also compute reconstruction loss:

```
# Compute reconstruction loss
loss = np.mean(np.square(recon_data - org_data))
print("Reconstruction loss ", loss)
```

- In this case, reconstruction loss is: 2.6426840324903897e-32.
- It is very low because we used all the components to reconstruct the data.

Questions