CG4002

Embedded Systems Design Project

Lecture 2

Software: AI/Machine Learning

+ Dashboard Design

Wang Ye

Associate Professor, School of Computing

NUS National University of Singapore

School of Computing

What is Machine Learning?

Addresses the question of how to build computer programs or algorithms that improve their performance at some task through experience.

Or more simply, we try to make computers learn to solve problems.

Examples of ML Problems

- Voice Recognition
- Medical Diagnosis
- *Activity Recognition*
- Spam Detection
- Face Detection
- OCR
- Weather Forecasting
- And Many More…

How can Machines Learn to Solve Problems?

One can learn from nature: humans or animals

Humans learn from experience to improve ourselves…E.g., a toddler learns to walk

Can we also let machines learn from experience?

E.g., to distinguish sitting ~ walking, dance moves

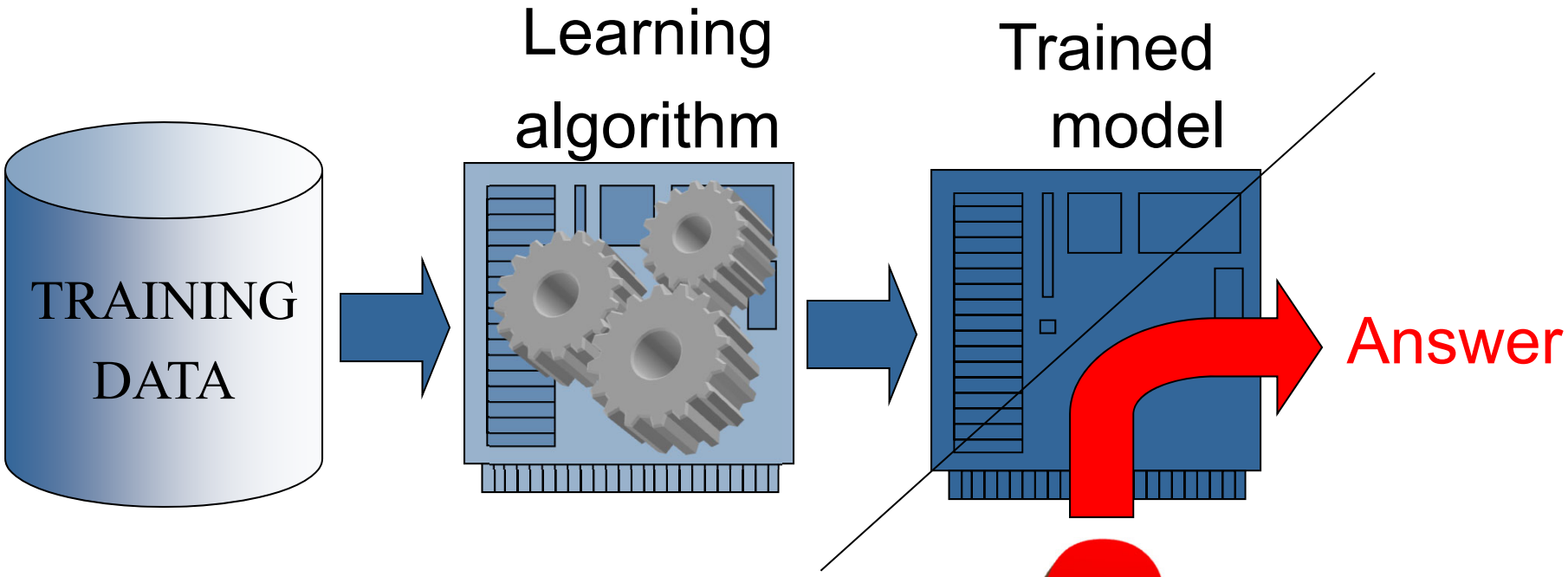
The essence of learning problems

1. A pattern exists (e.g., how we walk or speak tells a lot about us!)

2. Difficult to pin down formally/analytically/mathematically

3. We have data for it (e.g., in our memory)

A Machine Learning Framework
(general)

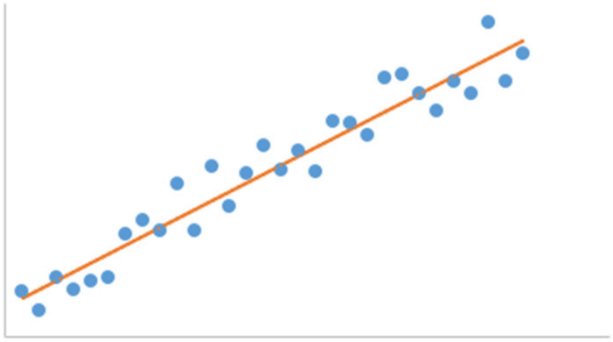


1. How to learn from experience/data?
2. How to make prediction?

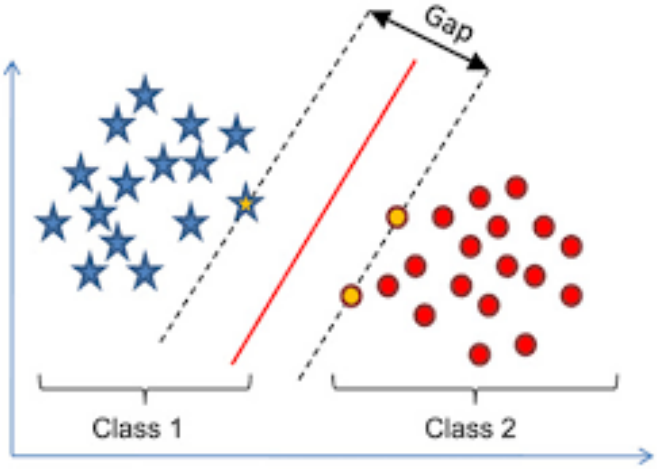
AI/ML is red hot now!

Terminology

Statistical modeling: In simple terms, statistical modeling is a simplified, mathematically-formalized way to approximate reality (i.e., what generates your ***data***) and optionally to make predictions from this approximation. The statistical model is the mathematical equation (in a ***feature space***) that is used.



Regression (curve fitting)



Classification

Terminology

Classification: In machine learning and statistics, classification is the problem of identifying to which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known.

Regression analysis is a statistical process for estimating the relationships among variables. It includes many techniques for modeling and analyzing several variables, when the focus is on the relationship between a dependent variable and one or more independent variables (or 'predictors').

Quiz

Classification or Regression?

Given:	**Decide:**
1) Credit history	Lend money?
2) Picture of a person	Age (fractional age possible)
3) Recording of person singing	Goodness of singing score
4) Recording of person singing	Good or Bad Singer?

Types of Learning

- Supervised Learning (data with annotation)
- Unsupervised Learning (data without annotation)
- Semi-supervised Learning (data partly annotated)
- Reinforcement Learning (long process of exploration, reward & punishment on every action taken)

➢ We will concentrate on **Supervised Learning**

e.g., 10 moves, well defined, and easiest to get high accuracy

Activity Recognition

- Enable new forms of human-computer interaction
- Detect social interactions
- Assist diagnosis and monitoring of patients

Or… track people dancing?

Approaches

- Can use hand coded methods such as thresholding (e.g., drum beat) but…

- Let's make a computer learn the methods for us: A machine learning approach for the dance move classification

Overview

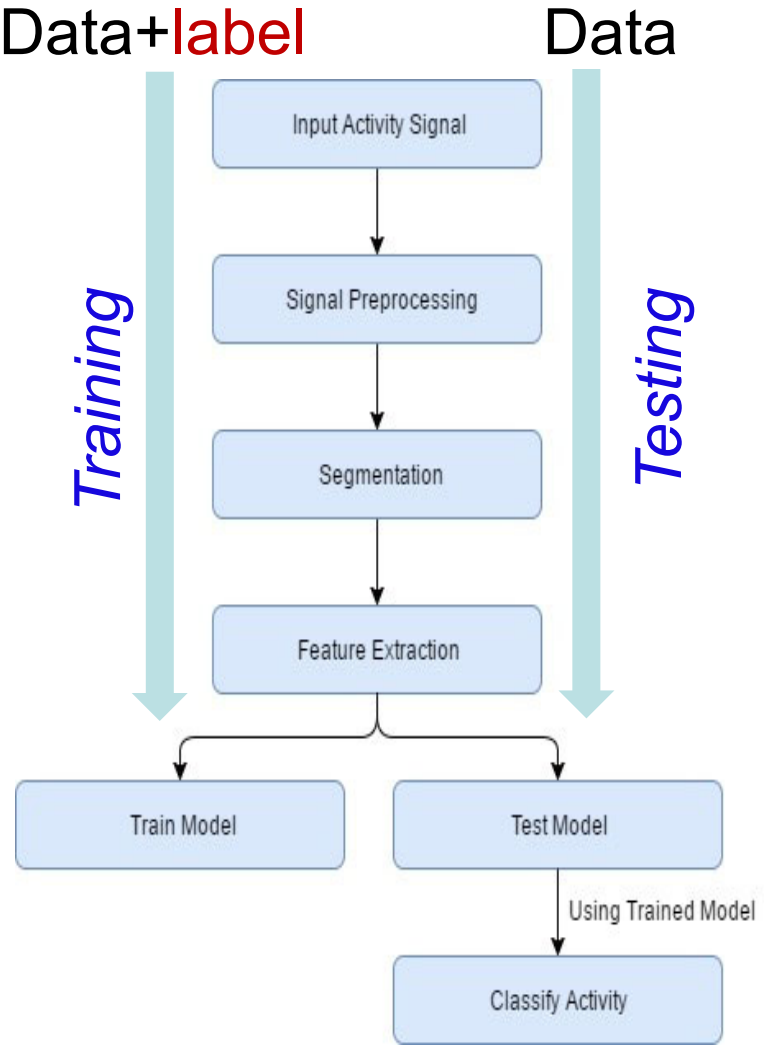
Data set of sensor Inputs

Signal Pre-processing

Segmentation

Feature Extraction

Classification of Activity



Training Dataset

- Need to collect a dataset for supervised training
- Dataset will consist of activity raw data and label
- Will need many different examples of each activity for generalisation

Sensor Inputs

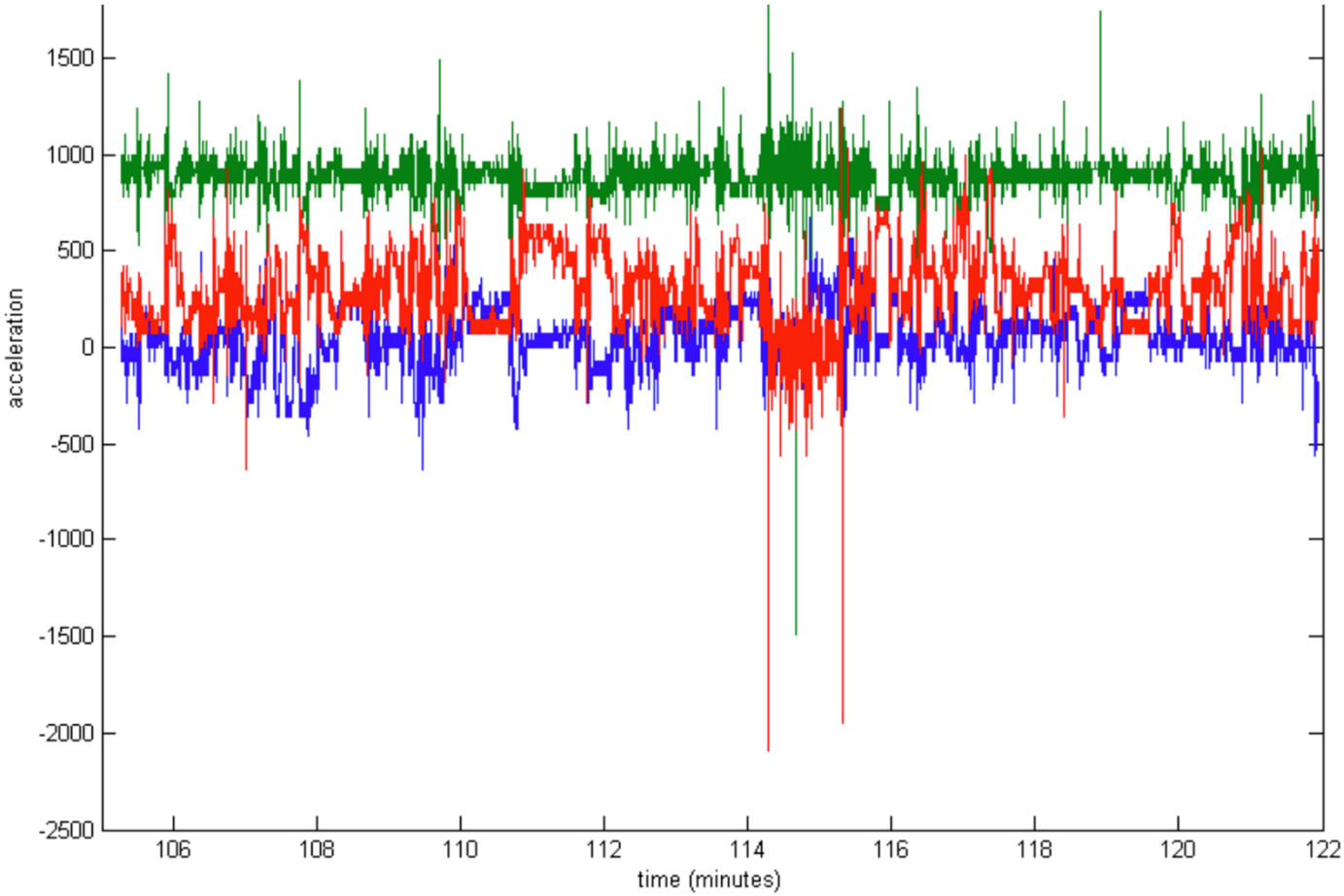
Each sensor provides raw inputs to the system.

Sensors can be of different types e.g. Accelerometers, Gyroscopes, Proximity Sensors, Vision Based Sensors e.g. Camera etc.

Each sensor may have more than one axes.

One or multiple sensors can be used to capture an activity or event.

Sensor Inputs



Preprocessing

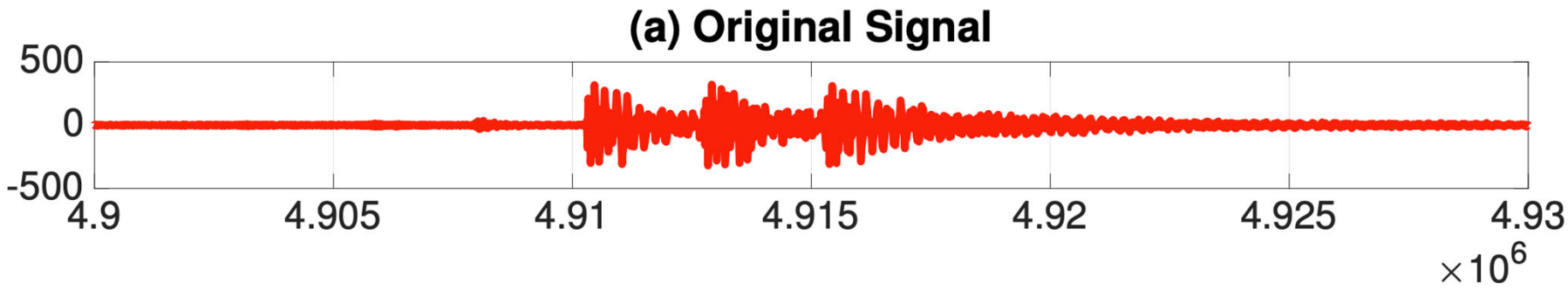
Different preprocessing techniques can be applied to the raw data to smooth the signal or remove the noise.

The signal can be normalized or scaled before being processed.

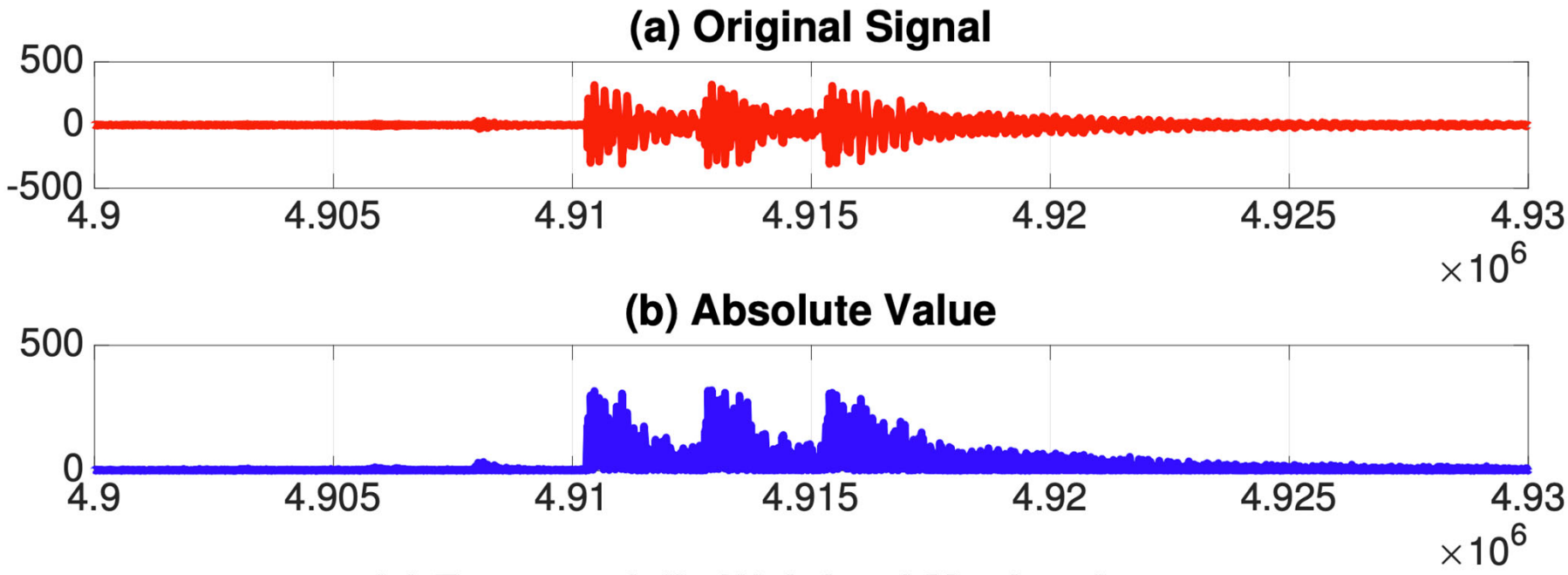
Different types of filters can be used:

- Low Pass Filters: To smooth out the signal
- High Pass Filters: To capture sudden changes

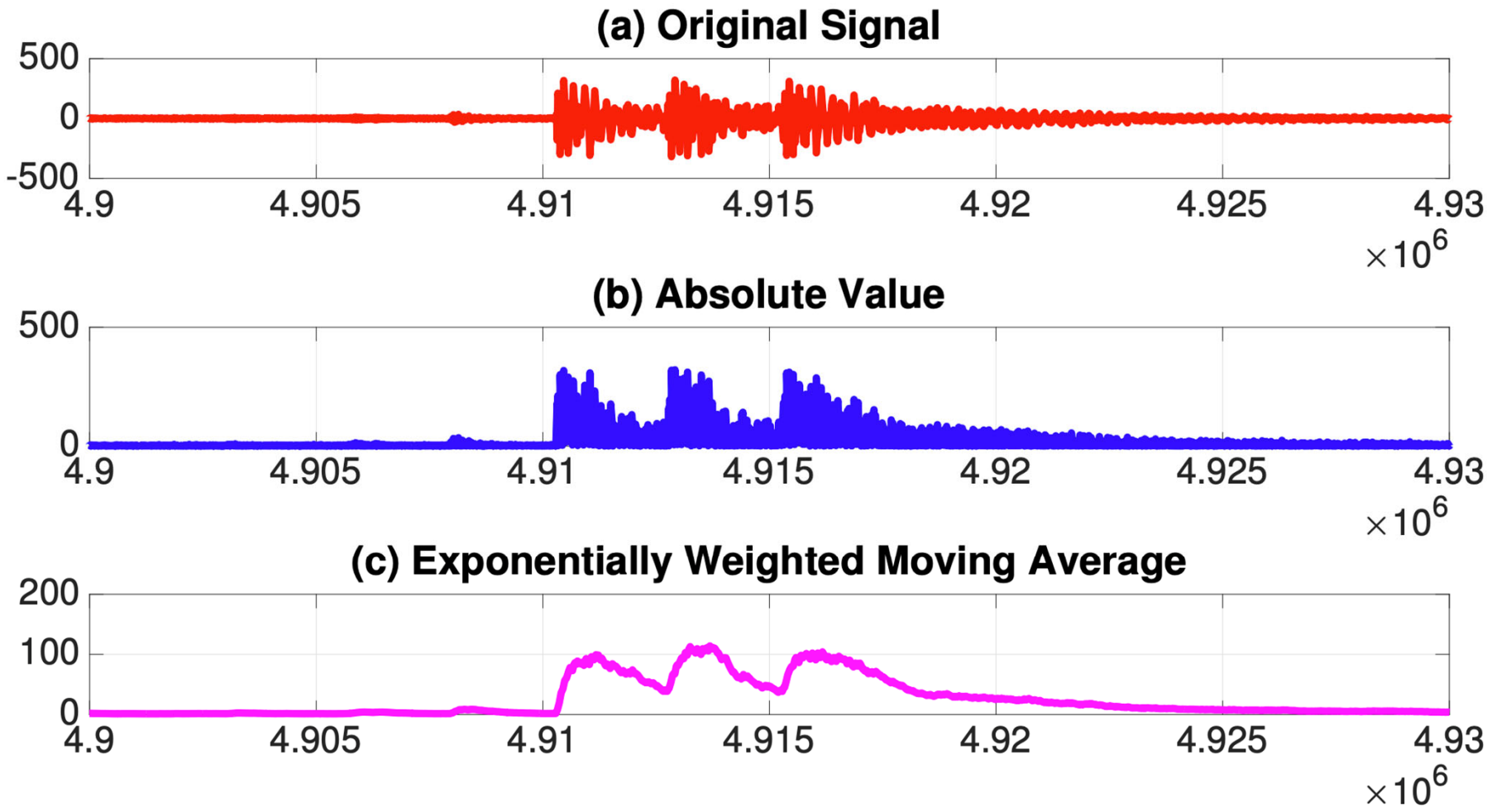
Preprocessing



Preprocessing



Preprocessing



Q: Is this a high pass or low pass filter?

Segmentation

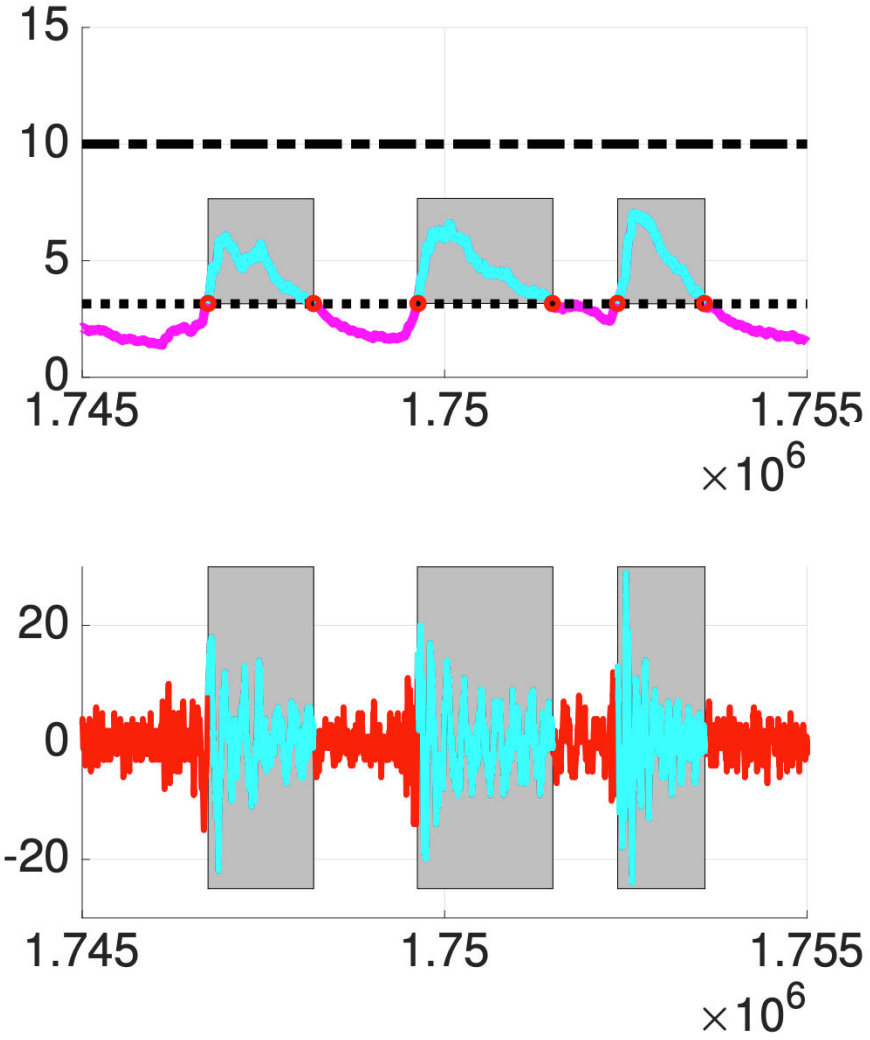
Identifying the start and end of the actual activity from the pre-processed signal (non-trivial task!).

Important for recognizing the activity accurately.

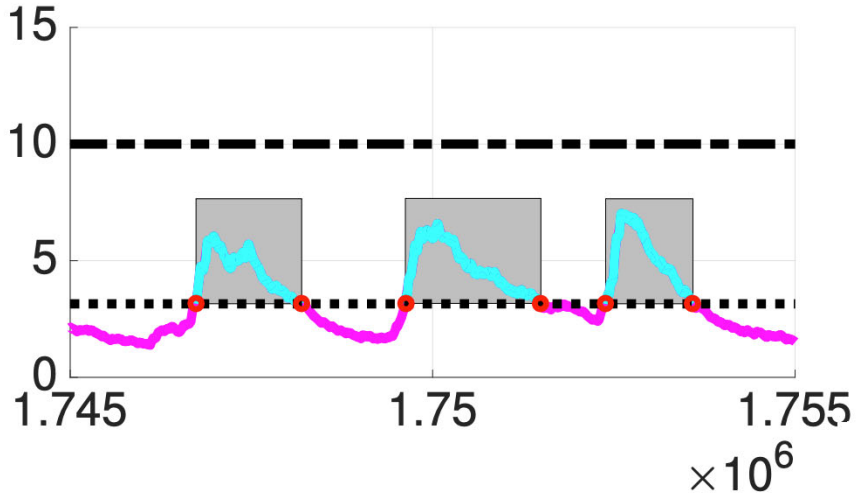
Different segmentation approaches can be used.

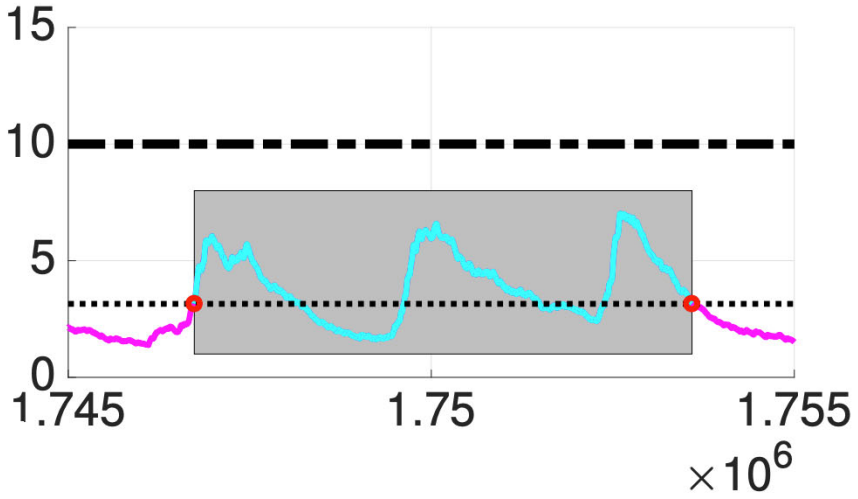
Another technique is to identify when the movement starts. It can be marked by a sudden change in accelerometer values (**onset detection**).

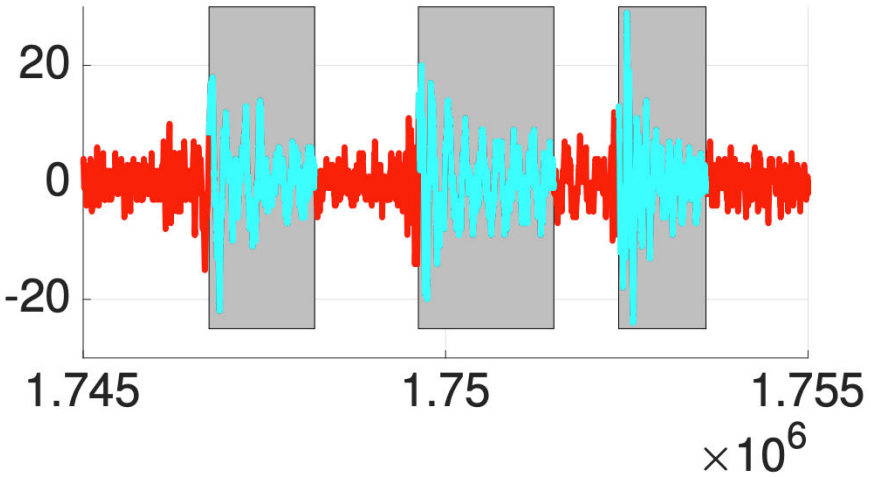
Segmentation

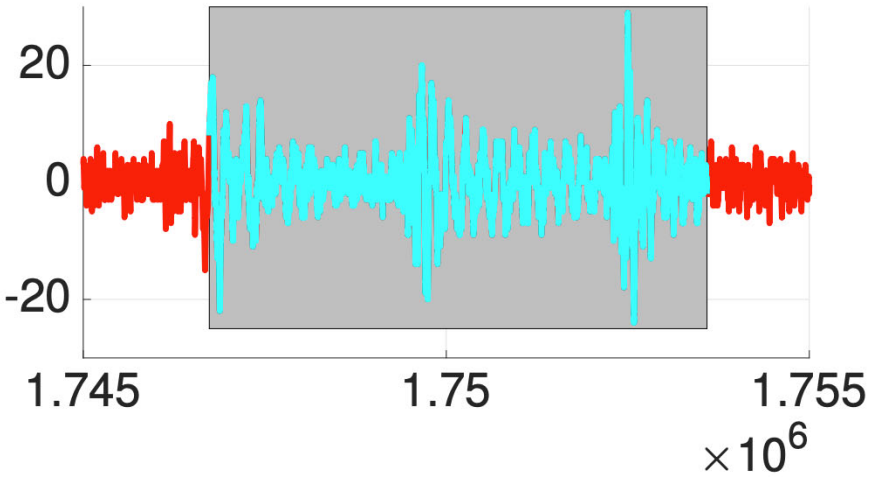


Segmentation









Feature Selection

Different features can be selected to classify the activities.

Selecting good features which distinguish well between the activities and increase the accuracy of the algorithm (**as well as reduce the computational workload**).

Features can be extracted for each axis e.g. x, y and z.

Features are classified into two types:

- Time Domain Features
- Frequency Domain Features

Feature Selection

Time Domain Features:

- Mean
- Variance
- Median
- Mean Absolute Deviation

Feature Selection

Frequency Domain Features:

- Spectral Energy
- Entropy
- Peak Frequency
- Sub-band power

Potential Models

Support Vector Machine (SVM)

K-Nearest Neighbours (KNN)

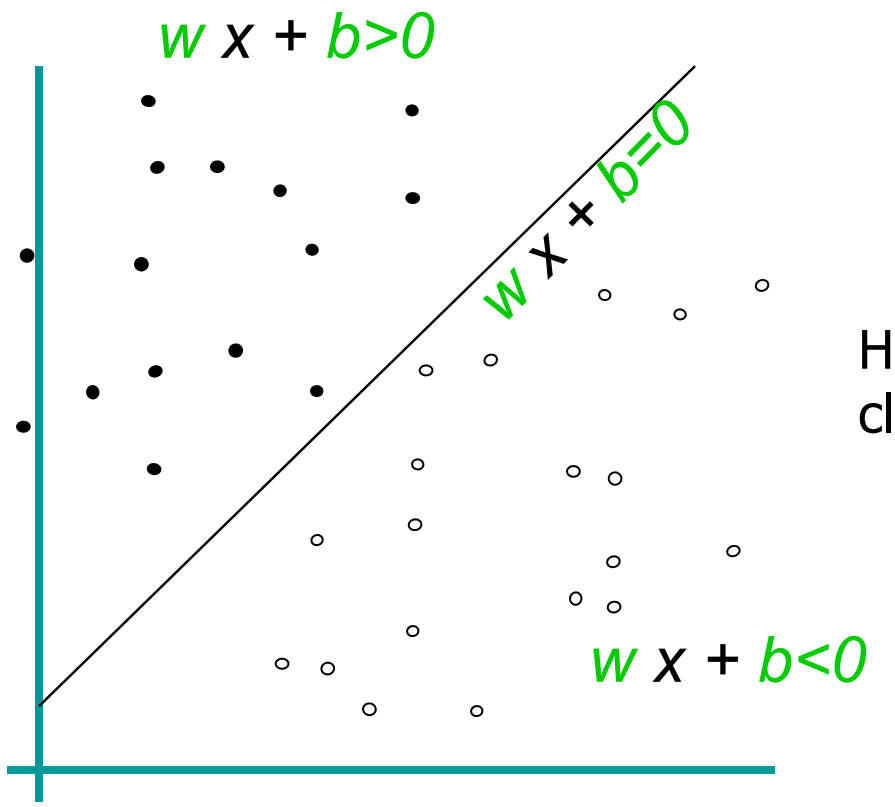
Artificial Neural Network (ANN)

Support Vector Machine (SVM)

- Core idea: Maximize Margin
  - Hard Margin (Linear)
  - Kernel (Nonlinear)

Linear Classifiers

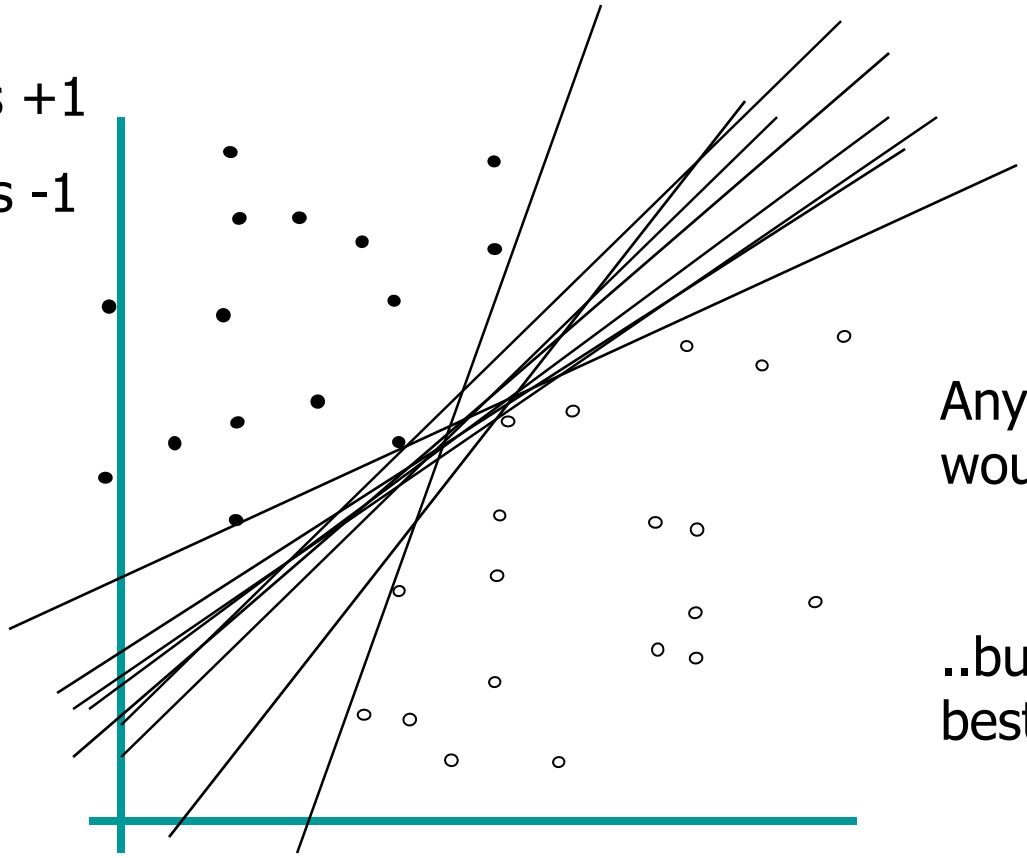
- denotes +1
- denotes -1



How would you classify this data?

Linear Classifiers

- denotes +1
- denotes -1

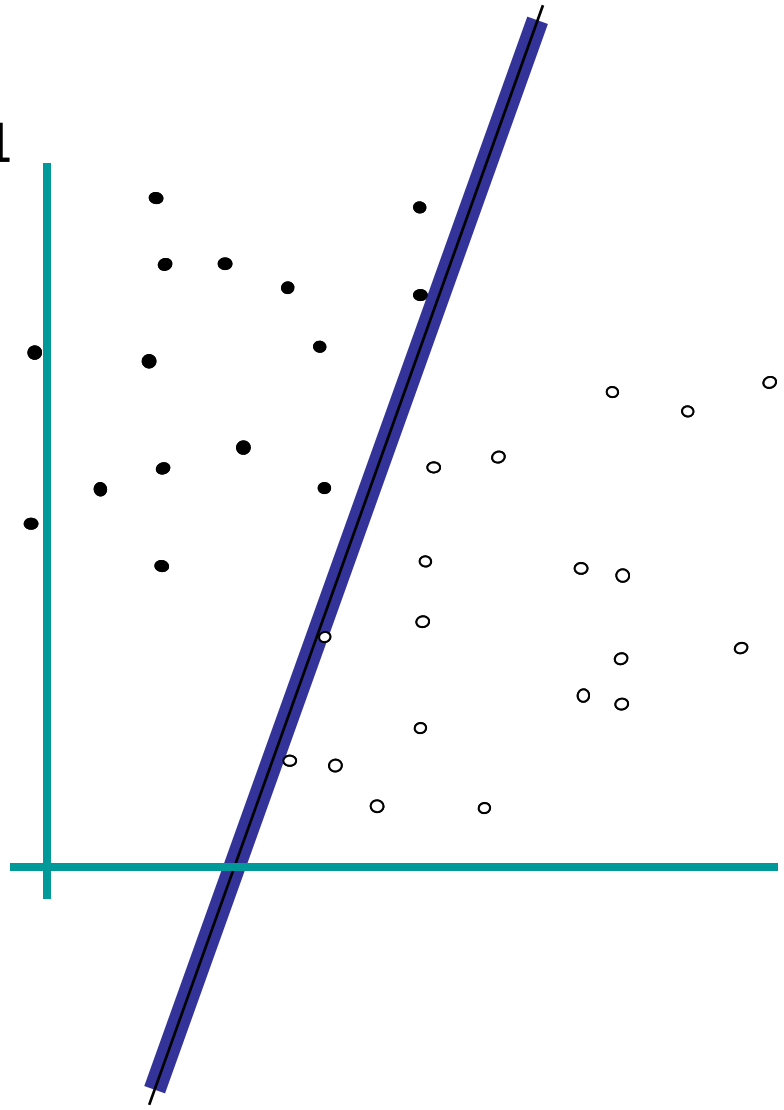


Any of these would be fine..

..but which is best?

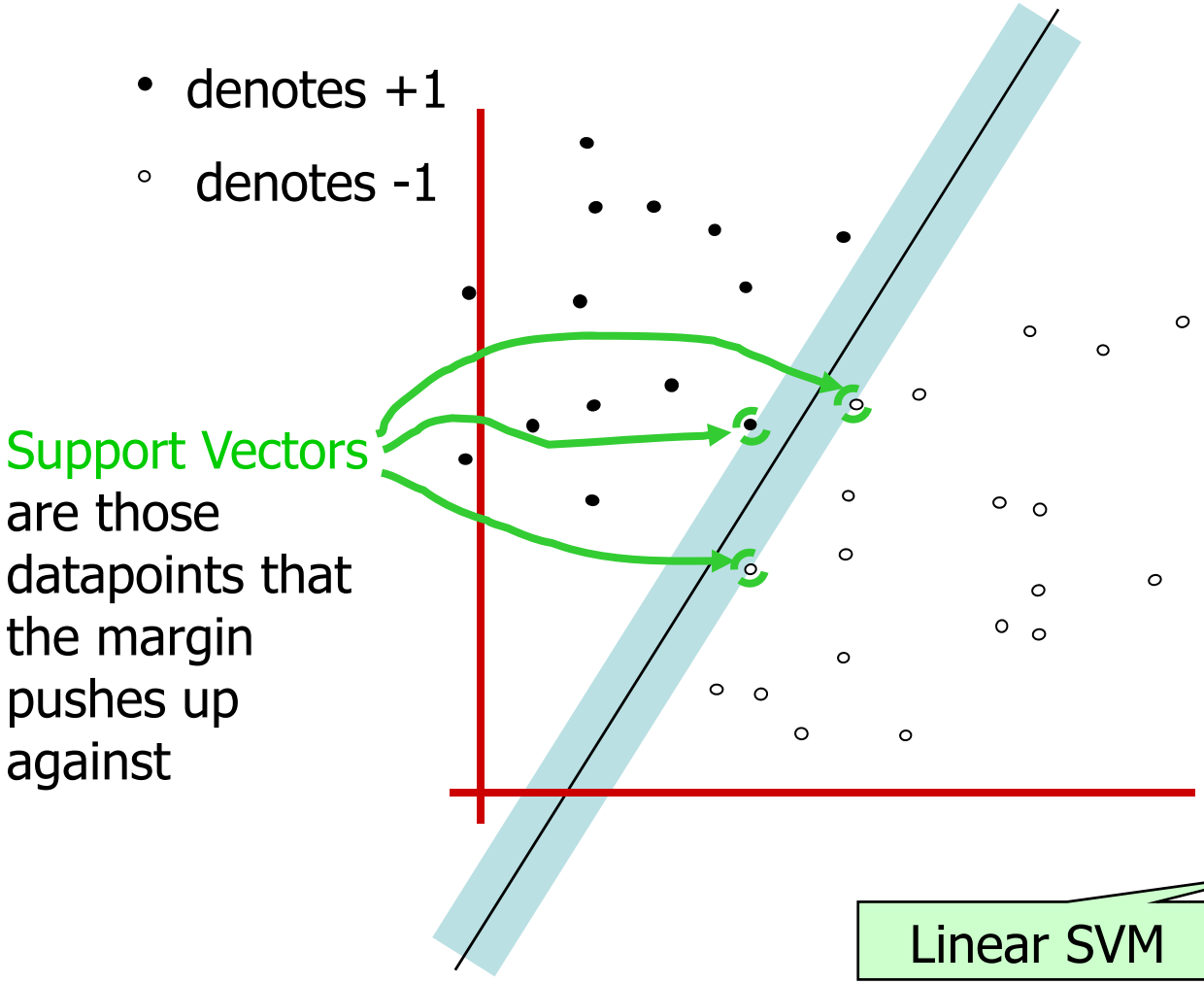
Classifier Margin

- denotes +1
- denotes -1



Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Maximum Margin

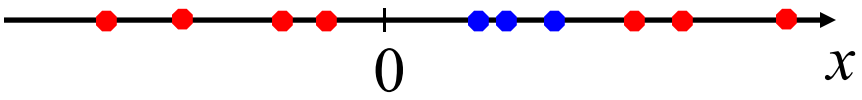


The maximum margin linear classifier is the linear classifier with the, um, maximum margin.

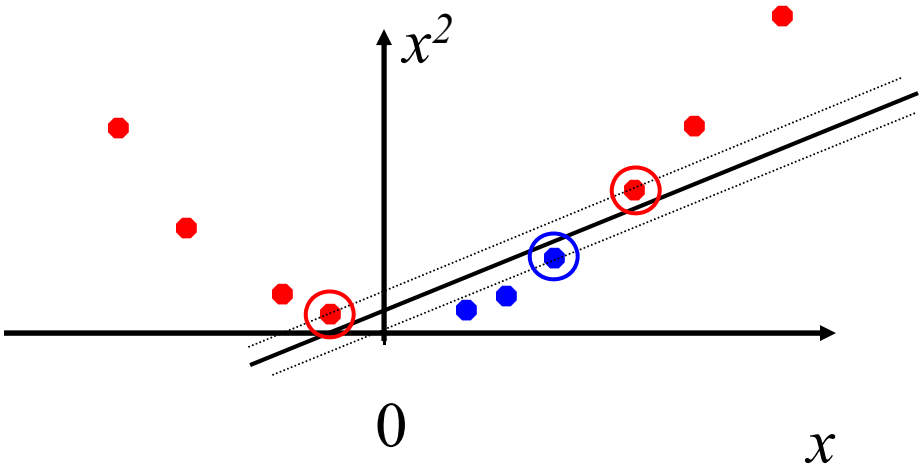
This is the simplest kind of SVM (Called an LSVM)

Non-linear SVMs (Kernel)

- But what are we going to do if the dataset is just too hard?

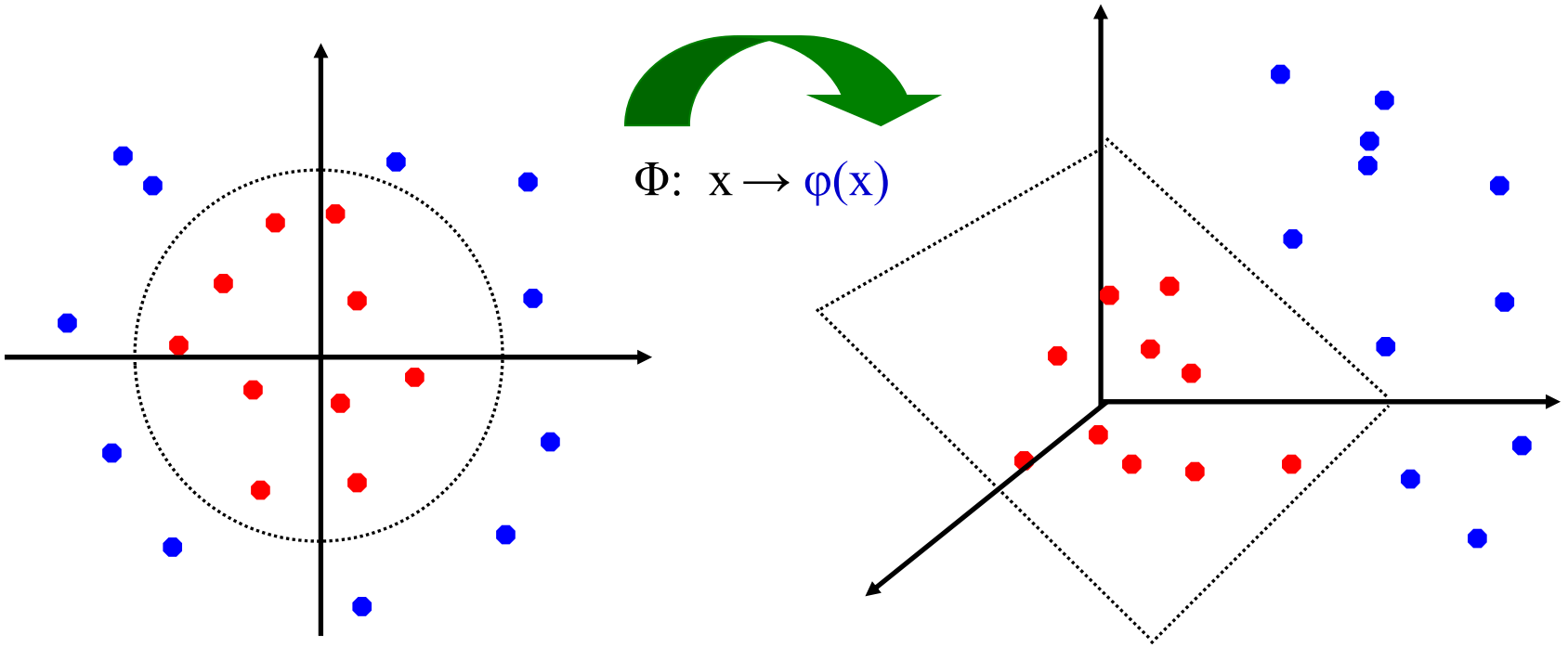


- How about… mapping data to a higher-dimensional space:



Non-linear SVMs: Feature space

- General idea:
  - the original input space can always be mapped to some higher-dimensional feature space where the training set is separable.



Q: Can you define a proper kernel function φ(x) here?

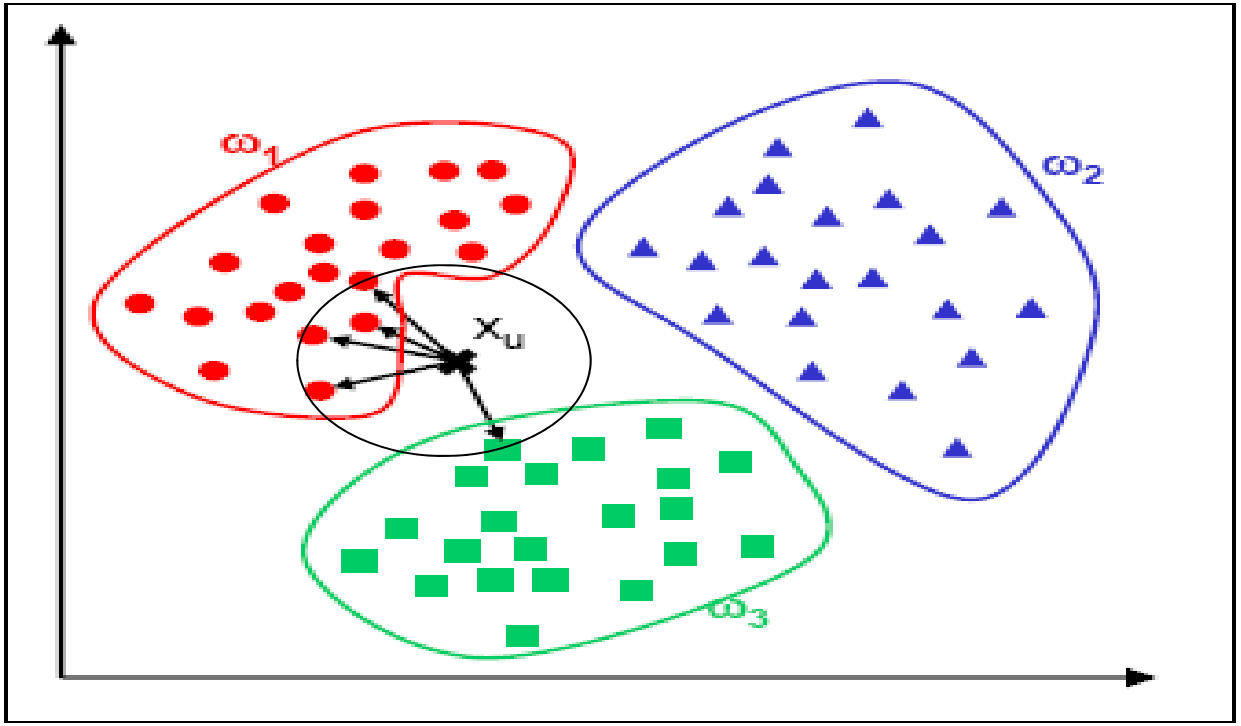
SVM: Pros and Cons

- Naturally suited for 2-class classification
- Simplicity
- But, non-trivial to define a Kernel

Q: Is SVM a good choice for your project to classify 10 dance moves? If yes, how to do it?

K Nearest Neighbor (KNN) Classifier

- **Classify unlabeled data according to the dominant class of the k nearest neighbors in the training data set.**

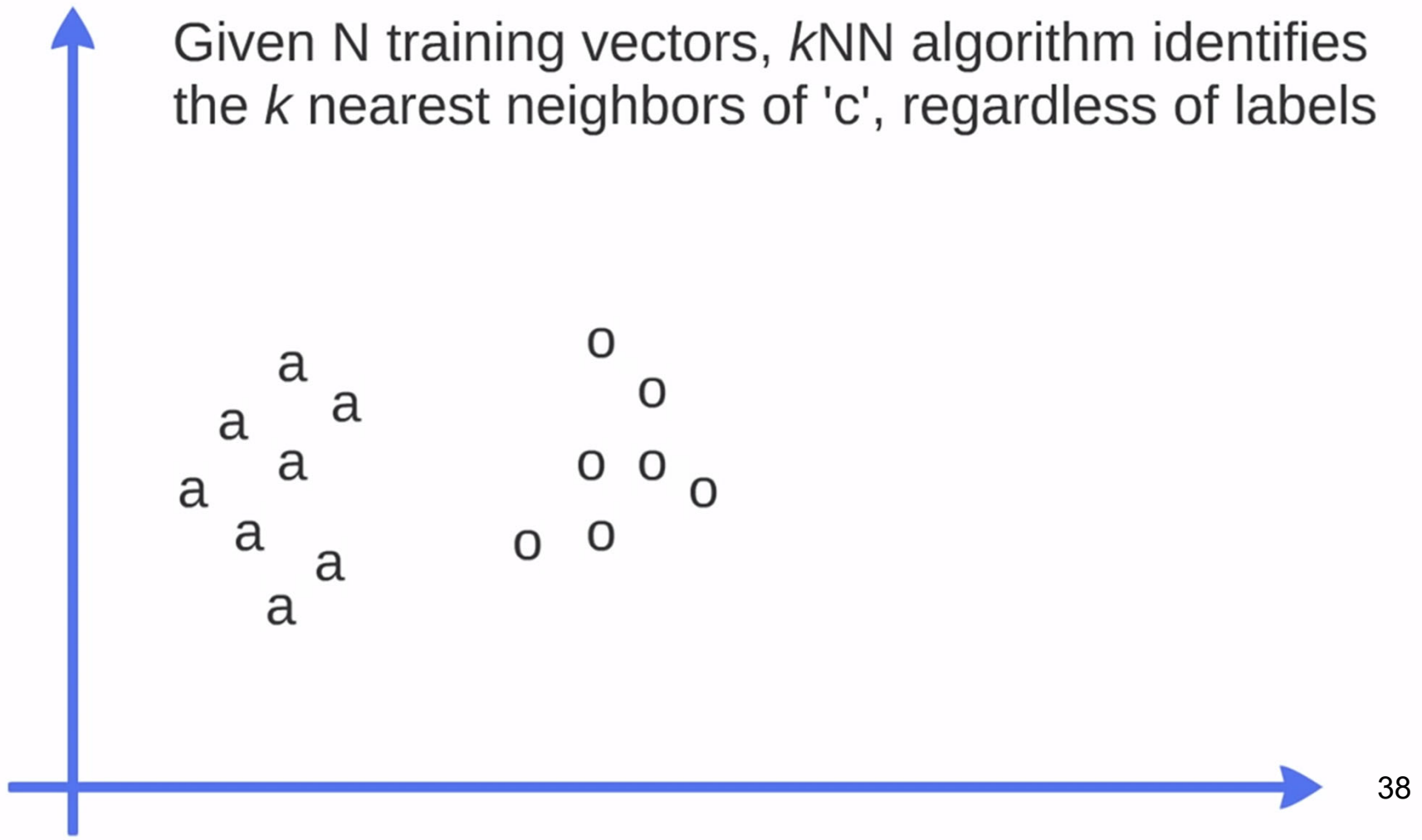


K Nearest Neighbours

- KNN is a classification algorithm.
- Classifies datasets based on distance to its closest K neighbours.
- Euclidean distance is a common distance measure.
- Training consists of storing the feature vectors and class labels of the training samples.
- Testing consists of assigning a label to test object which is most frequent among the k training samples nearest to that point.

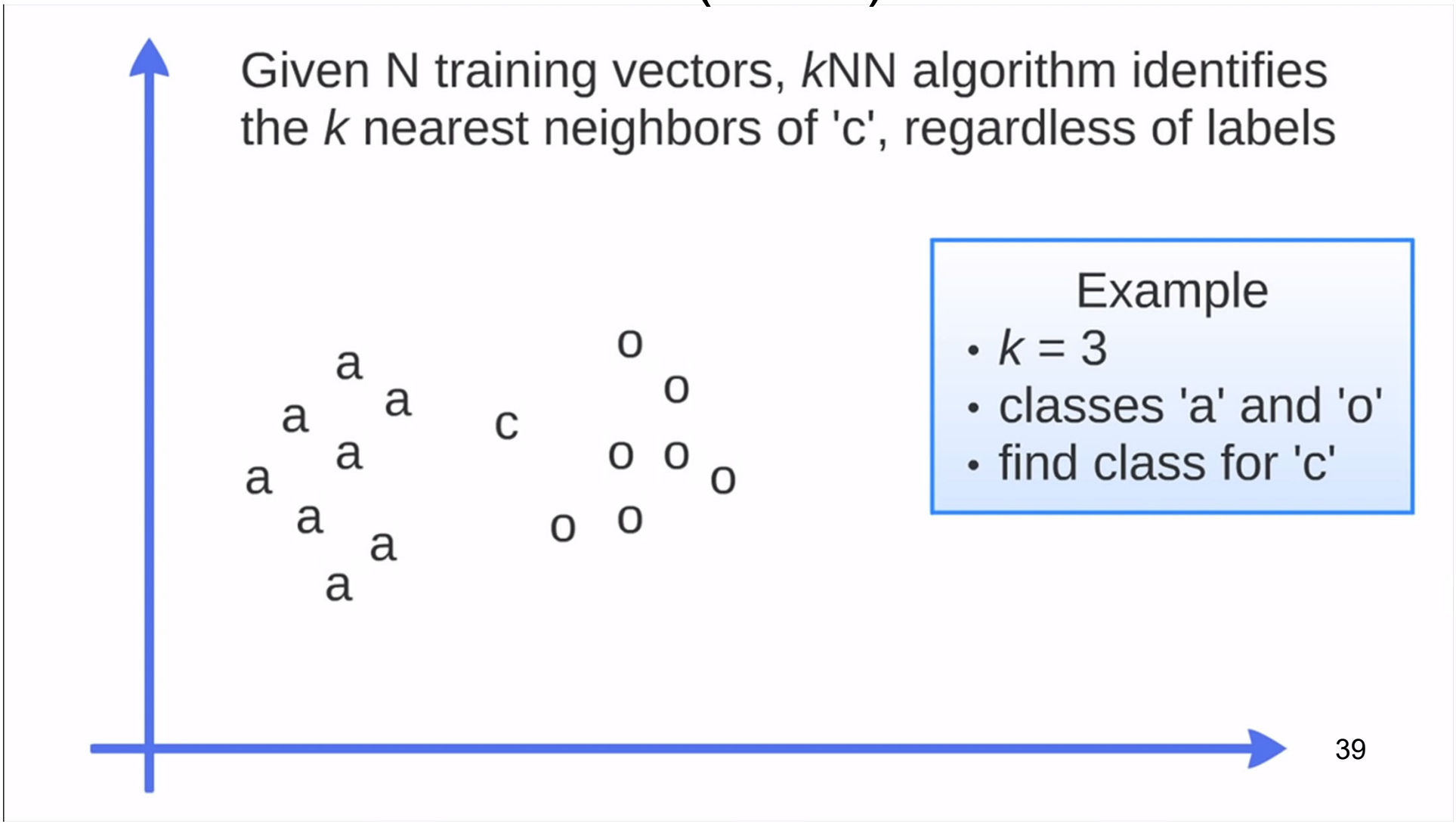
K Nearest Neighbours

2 class classification (K = 3)



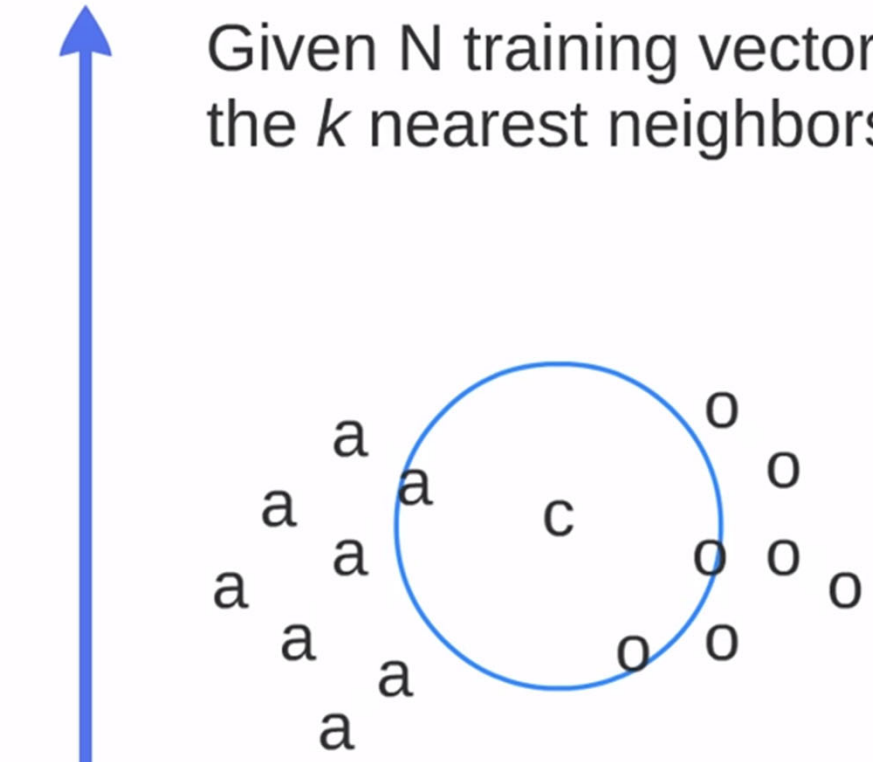
K Nearest Neighbours

2 class classification (K = 3)



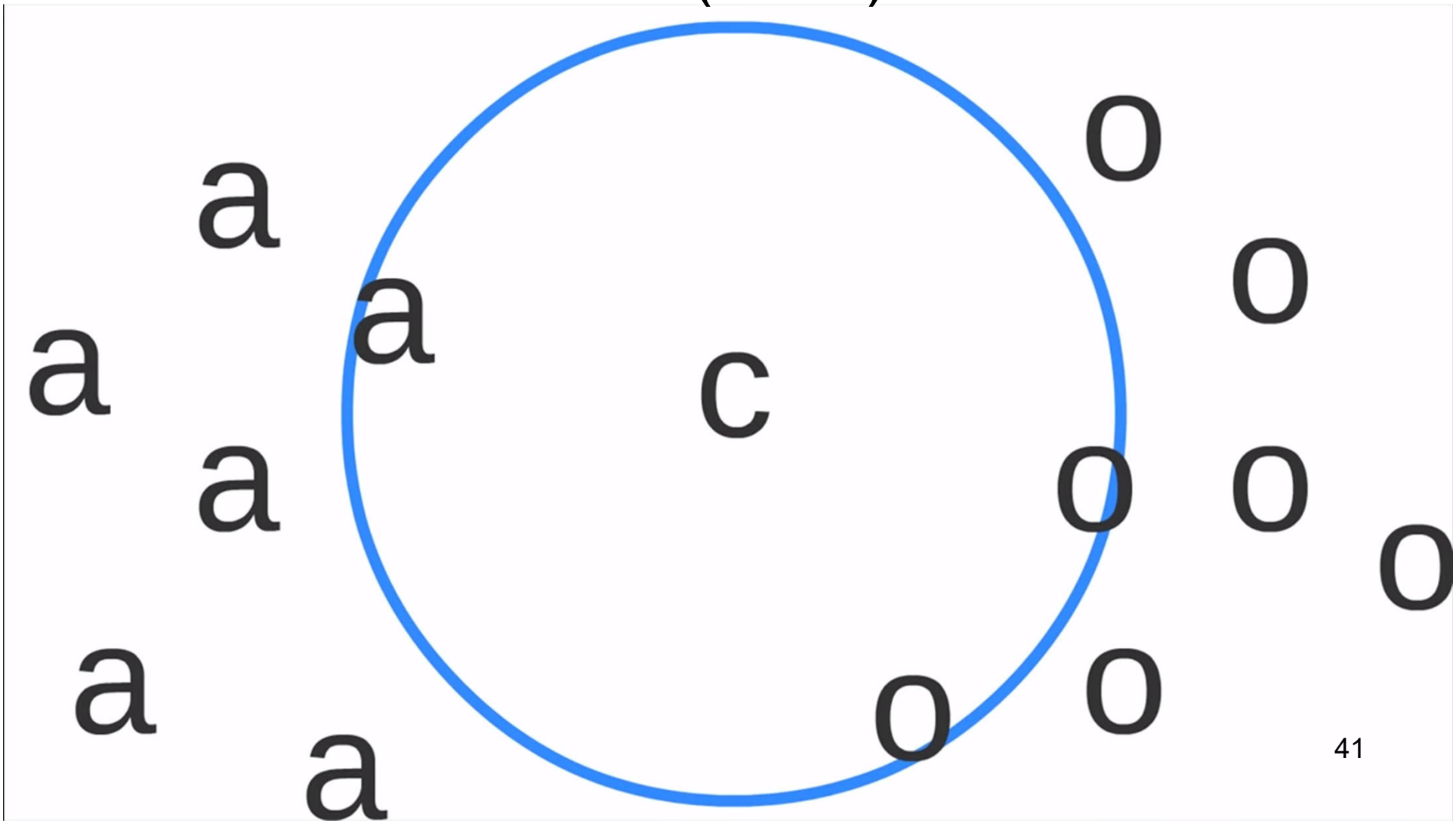
K Nearest Neighbours

2 class classification (K = 3)



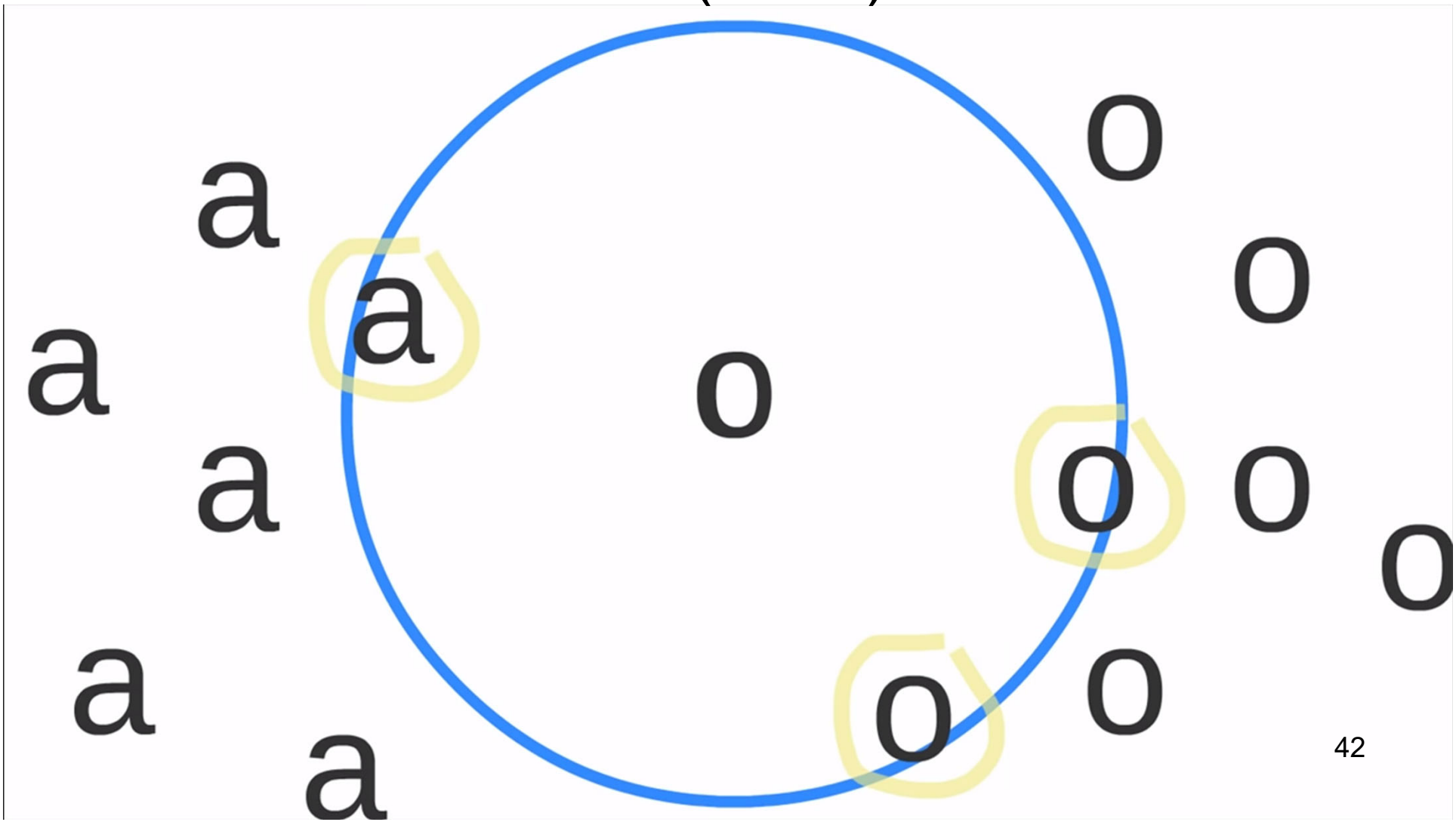
K Nearest Neighbours

2 class classification (K = 3)



K Nearest Neighbours

2 class classification (K = 3)



k-Nearest Neighbors

Given a query item:
Find k closest matches
in a labeled dataset ↓

k-Nearest Neighbors

Given a query item:
Find k closest matches
in a labeled dataset ↓

Return the most
frequent label

k-Nearest Neighbors

k=3

3 votes for “cat”

Cat wins…

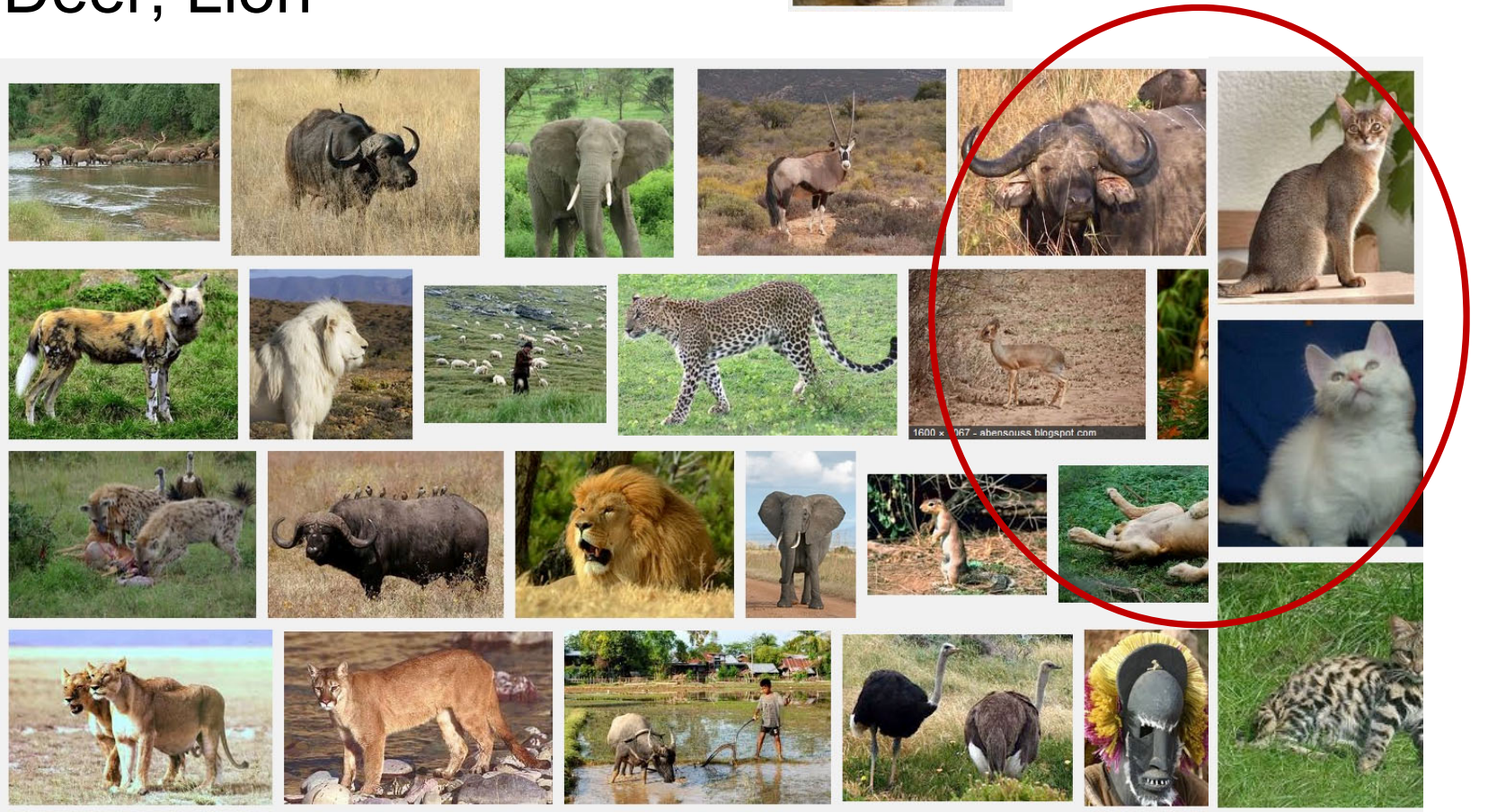
k-Nearest Neighbors

k=5, 2 votes for cat,

1 each for Buffalo,

Deer, Lion

Cat wins…



KNN: Pros and Cons

- Conceptually simple, but
- Large storage requirements
- Computationally intensive recall
- Neighborhood size k is manually assigned

Q: Is KNN a good choice for your project to classify 10 dance moves? Why?

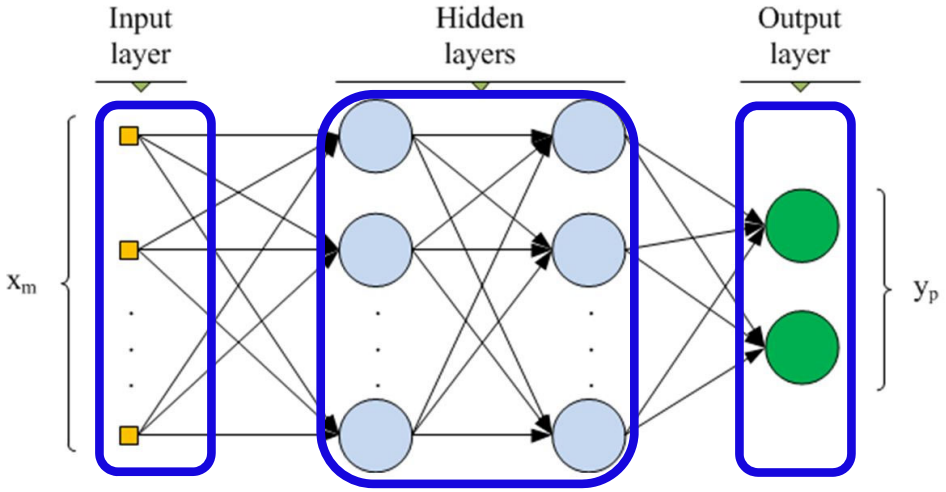
Artificial Neural Network (ANN)

Used in many applications such as speech recognition (e.g., Siri), spam detection, fingerprint recognition systems, finance prediction, activity recognition etc

An artificial neural network (ANN) is a multi-layered structure with the first layer as an input and the last layer as outputs.

Feedforward Neural Network

Each layer consists of multiple neurons.



Input of the (**feedforward**) neural network can be the activity signal input or the features extracted from the signal.

The middle layers do not connect with the outer world, hence are called hidden layers.

Output of the neural network consists of the labels/classes of the activity.

ANN: Pros and Cons

- State-of-the-art performance
- Potentially huge network architecture
- Still considered a black box approach

Q: Is ANN a good choice for your project to classify 10 dance moves? Why?

Classification libraries

There are many libraries/APIs for ML, that support many different types of models

- Scikit-learn (python)
- Tensorflow (standalone, python, etc)
- Keras (python)
- Weka (java, standalone, python)
- PyTorch (Python; C++; CUDA)

Machine learning libraries

Look into classification examples from scikit-learn:

http://scikit-learn.org/stable/supervised_learning.html#supervised-learning

If interested, should also have a look into tensorflow library, which provides more intuition into deep learning models.

https://www.tensorflow.org/api_docs/python/tf/contrib/learn/DNNClassifier

Activity datasets

1. http://archive.ics.uci.edu/ml/datasets.html

Filter with different criteria (e.g. task; instance; attribute.) or searching with certain keywords (e.g. sensor; action recognition).

Cite: **Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set, UC Irvine**

2. https://www.kaggle.com/datasets

Example dataset:

Run or walk [A dataset containing labeled sensor data from accelerometer and gyroscope] (https://www.kaggle.com/vmalyi/run-or-walk)

Many other activity data sets available online. Search and cite!

Model Assessment

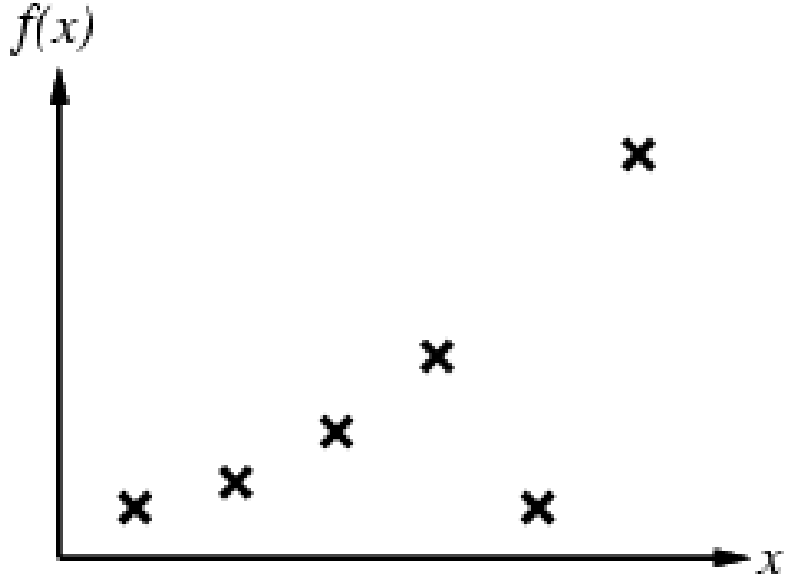
- The *generalization* performance of a machine learning method relates to its prediction capability on independent test sets.
- Assessment of this performance is extremely important in practice, since it guides the choice of the machine learning method or model.
- Further, this gives us a measure of the quality of our chosen model.

Model Assessment (cont.)

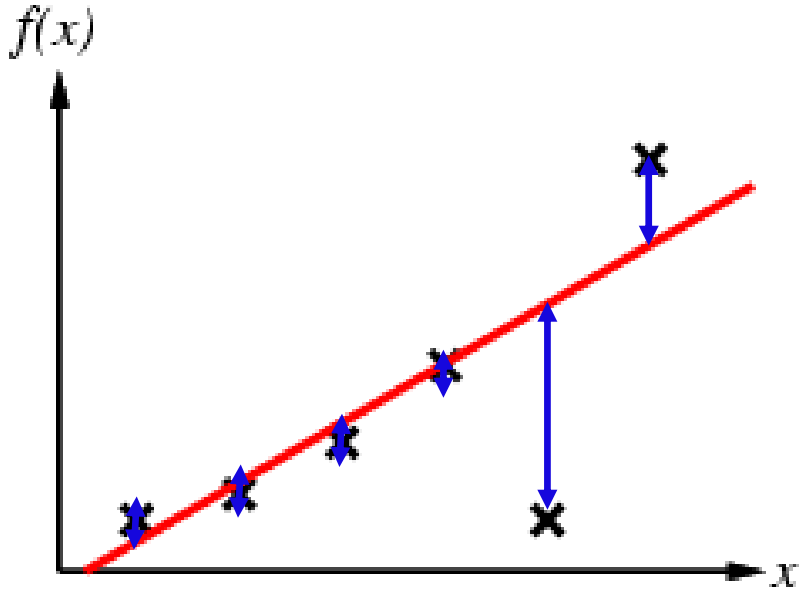
- *Test Error*
  - The average error that results from using a trained model to predict the response on a new observation.
  - The prediction error over an independent test sample.
- *Training Error*
  - The average loss over the training sample
- **Note:** The training error rate can dramatically *underestimate* the test error rate

How are the training & test errors computed?

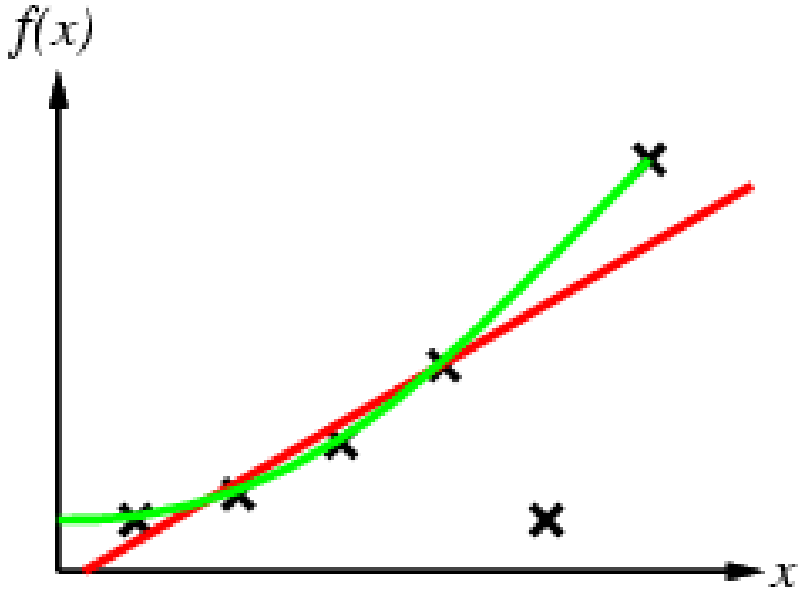
Model Assessment (cont.)



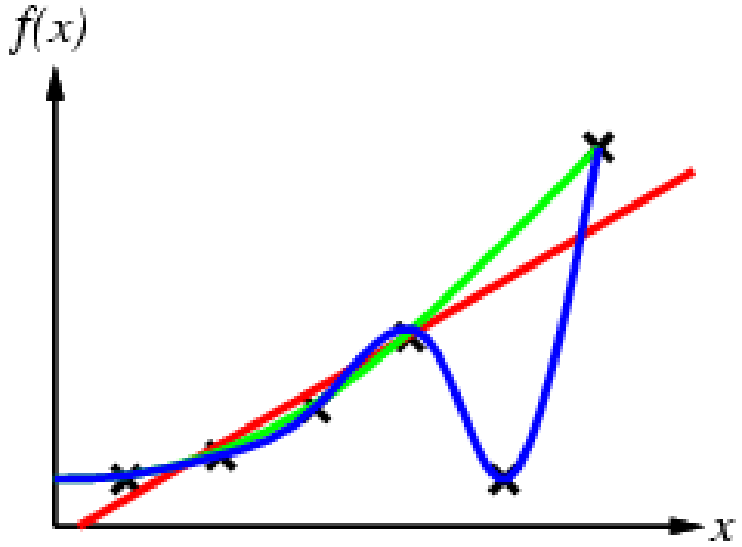
Model Assessment (cont.)



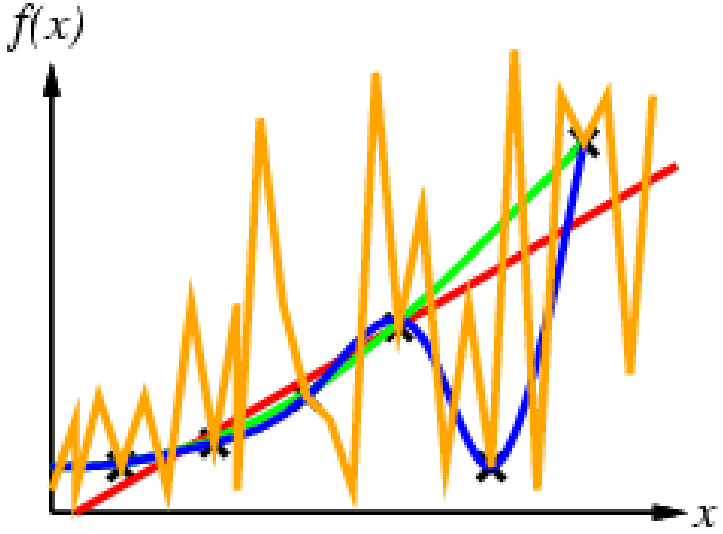
Model Assessment (cont.)



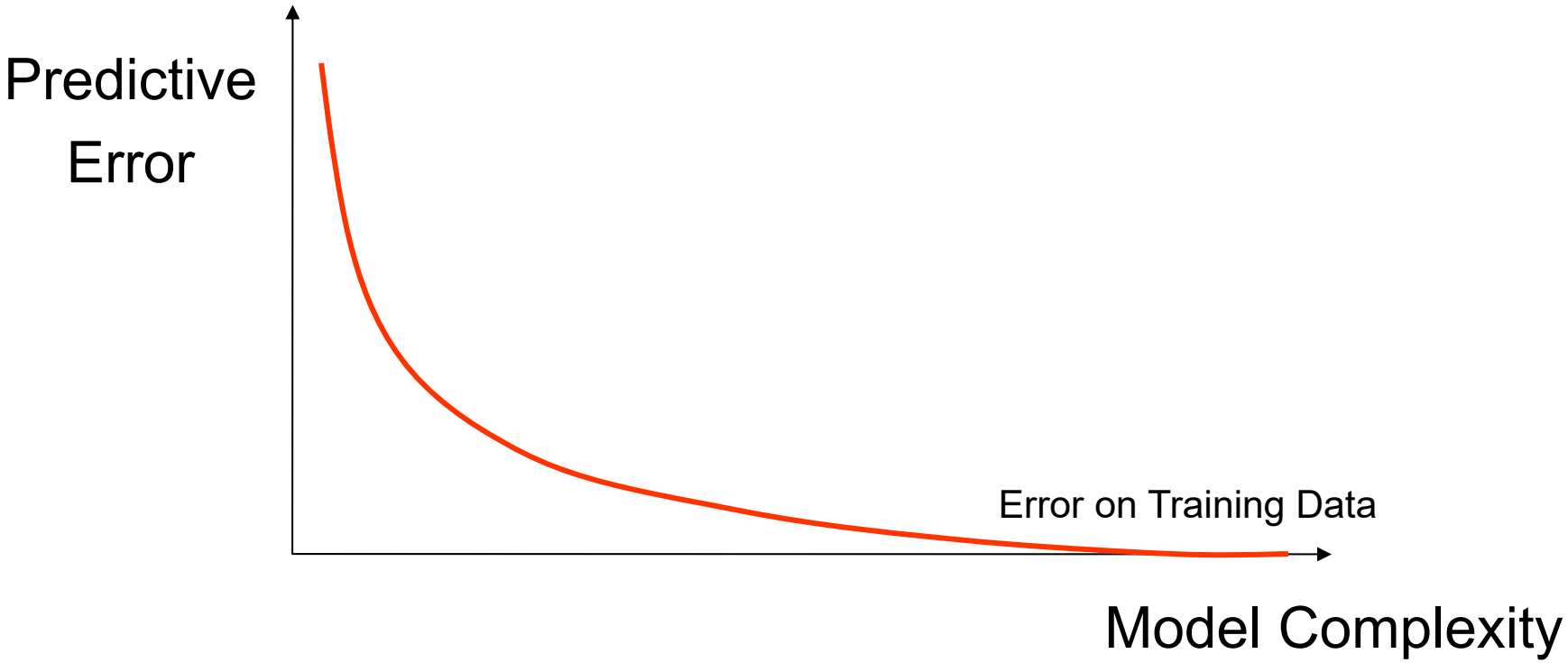
Model Assessment (cont.)



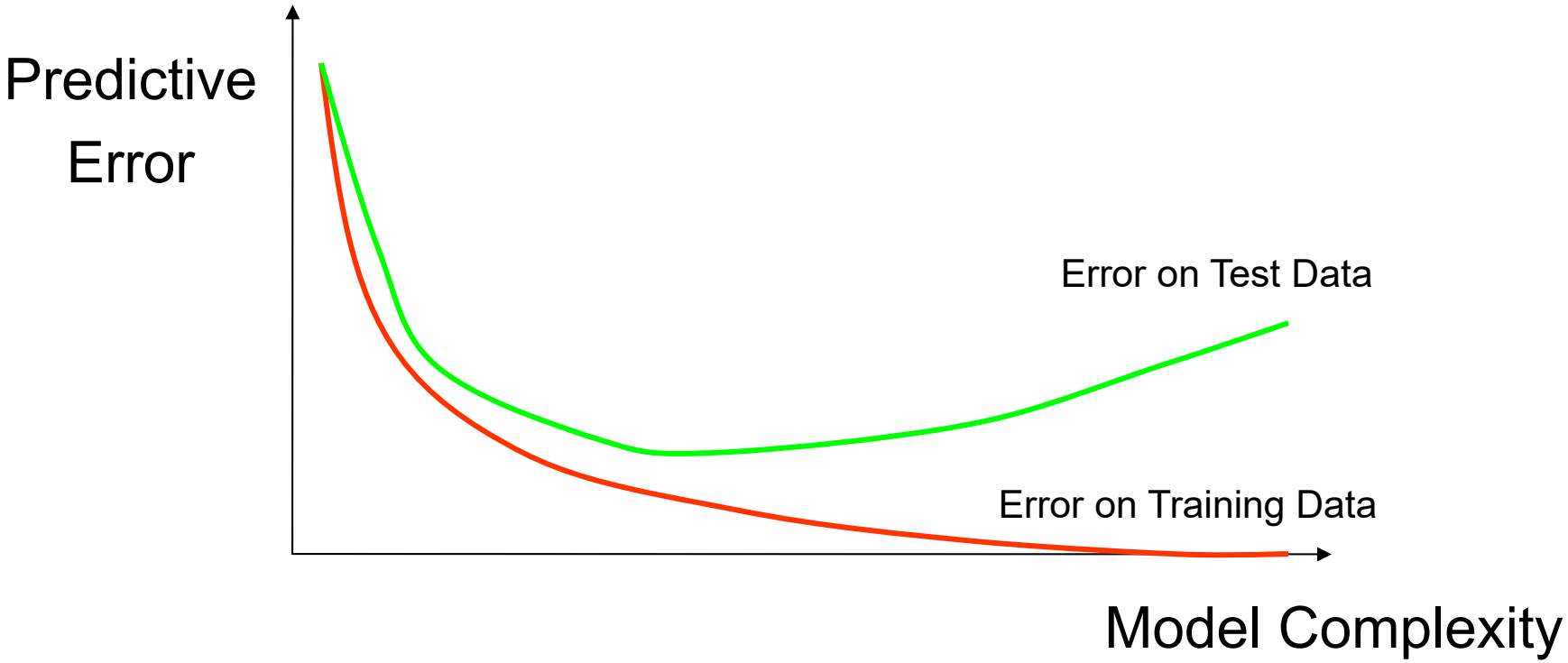
Model Assessment (cont.)



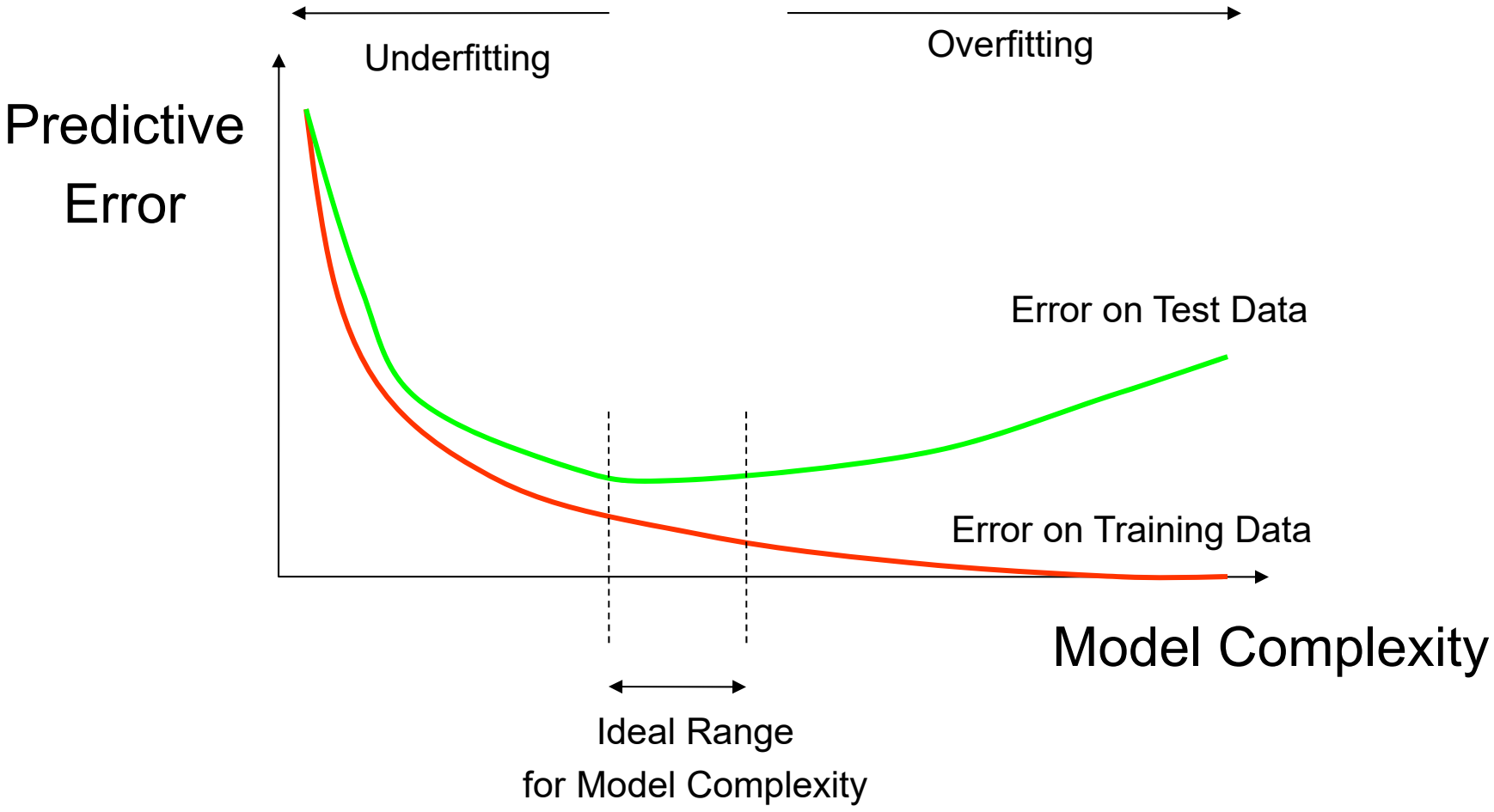
Model Assessment (cont.)



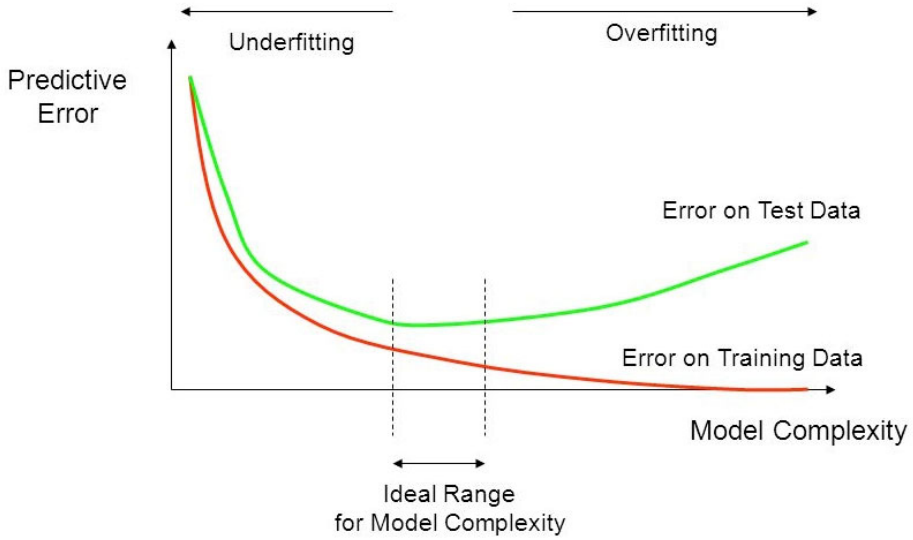
Model Assessment (cont.)



Model Assessment (cont.)



Model Assessment (cont.)



- As the model becomes more and more complex, it uses the training data more and is able to adapt to more complicated underlying structures.

- Training error consistently decreases with model complexity but performs worse on test data (higher test error)

Model Assessment (cont.)

- If we are in a data-rich situation, the best approach for both *model selection* and *model assessment* is to randomly divide the dataset into three parts: training set, validation set, and test set.

- The *training set* is used to fit the models. The *validation set* is used to estimate prediction error for model selection. The *test set* is used for assessment of the prediction error of the final chosen model.

- A typical split might by 50% for training, and 25% each for validation and testing.

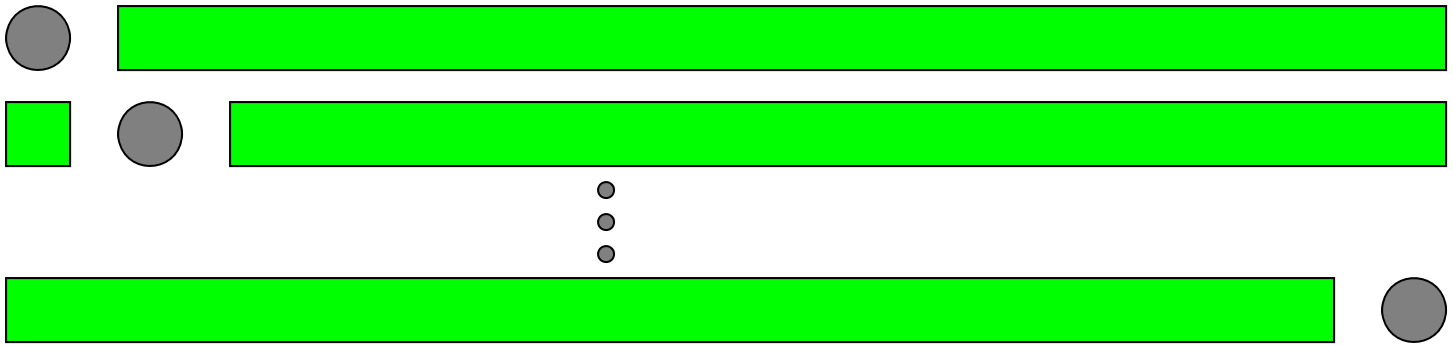
Common Splitting Strategies

- k-fold cross-validation

Common Splitting Strategies

- k-fold cross-validation

- Leave-one-out (n-fold cross-validation)



Evaluation Metrics

		Actual Label	
		Positive	Negative
Predicted Label	Positive	**True Positive (TP)**	**False Positive (FP)**
	Negative	**False Negative (FN)**	**True Negative (TN)**

Evaluation Metrics

- Accuracy = $\frac{TP + TN}{TP + TN + FP + FN}$

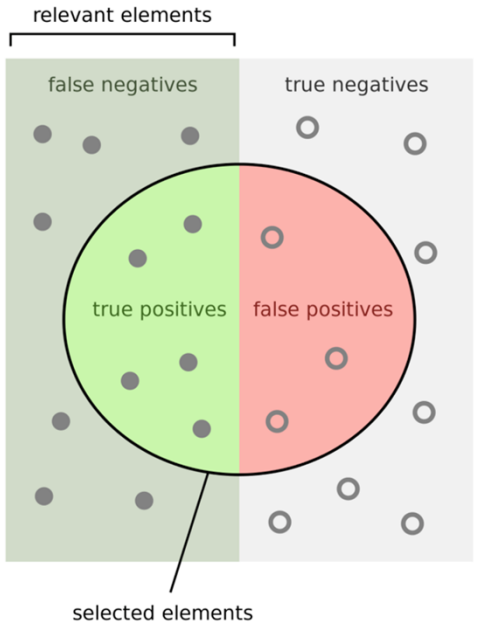
(total number of cases)

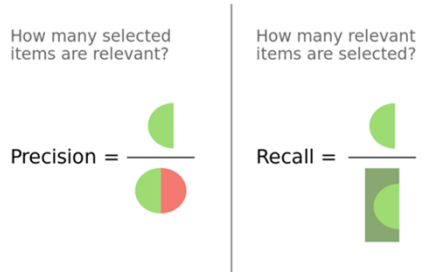
- Recall = $\frac{TP}{TP + FN}$

(actual positives)

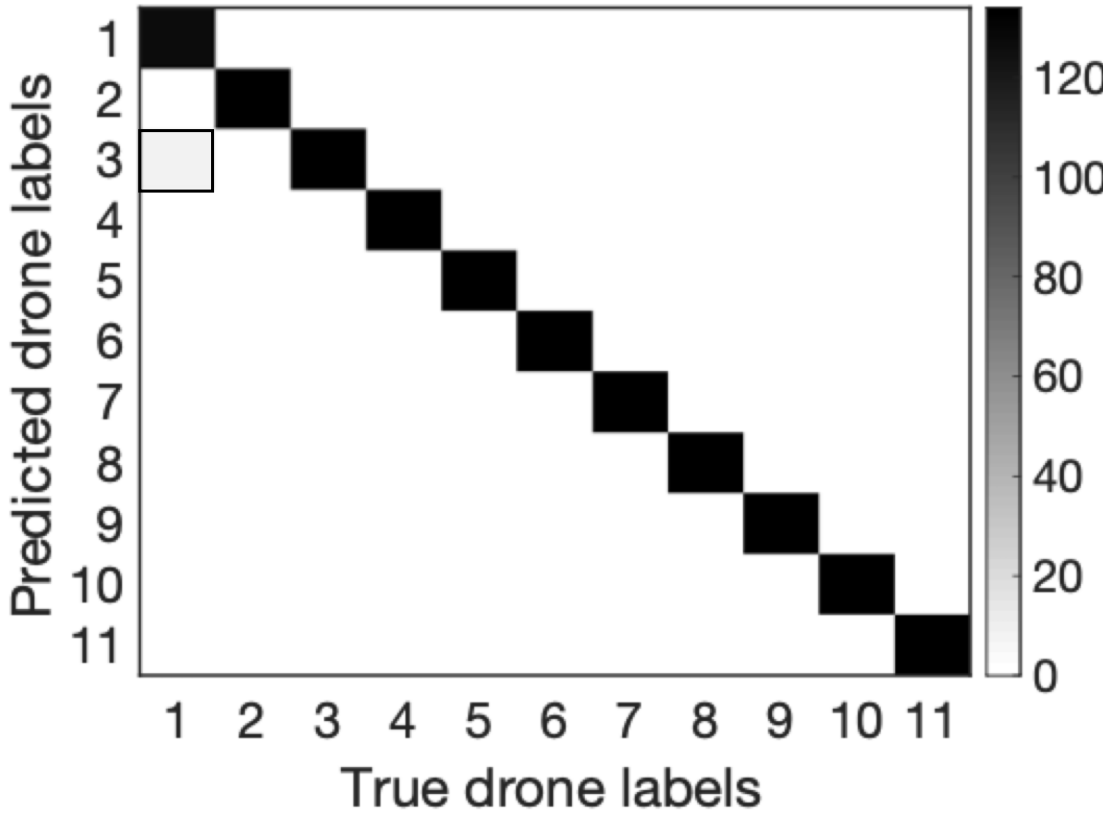
- Precision = $\frac{TP}{TP + FP}$

(predicted positives)



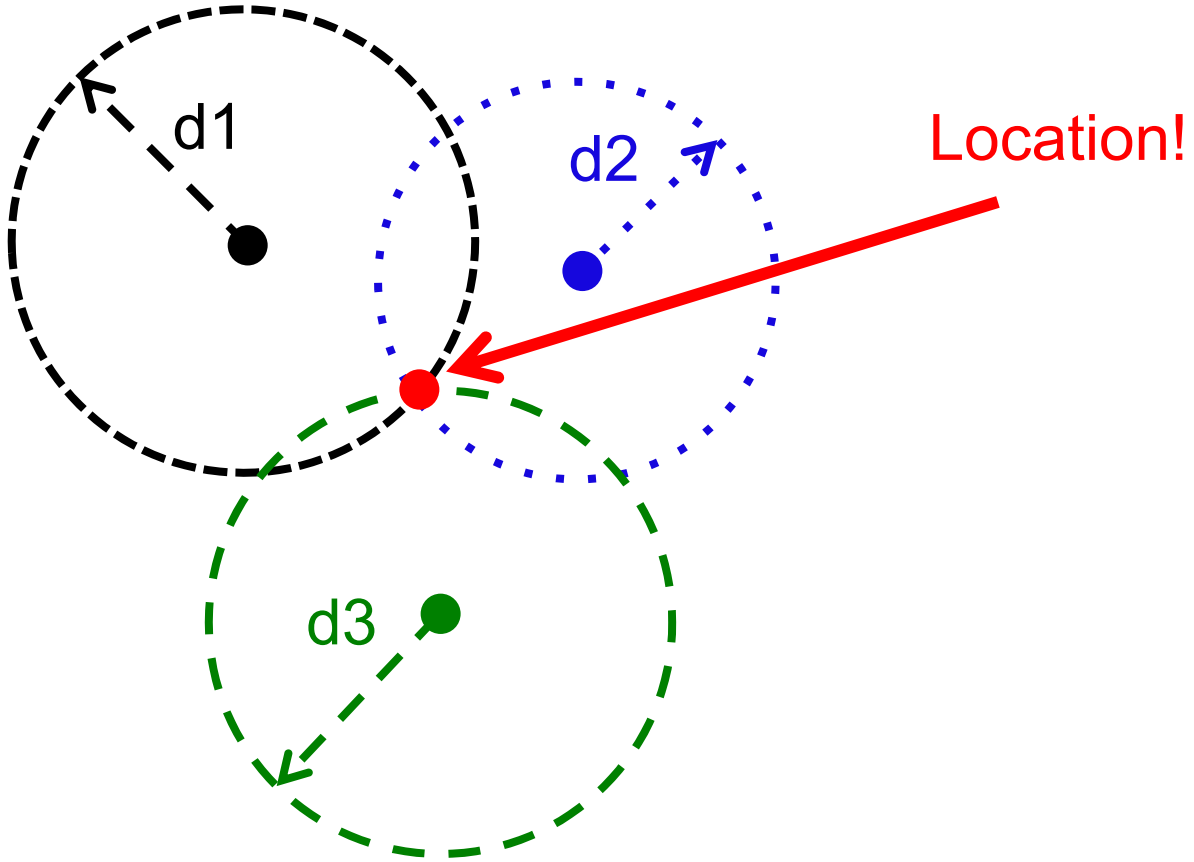


Confusion Matrix



Relative Position

Localization Approach: Triangulation

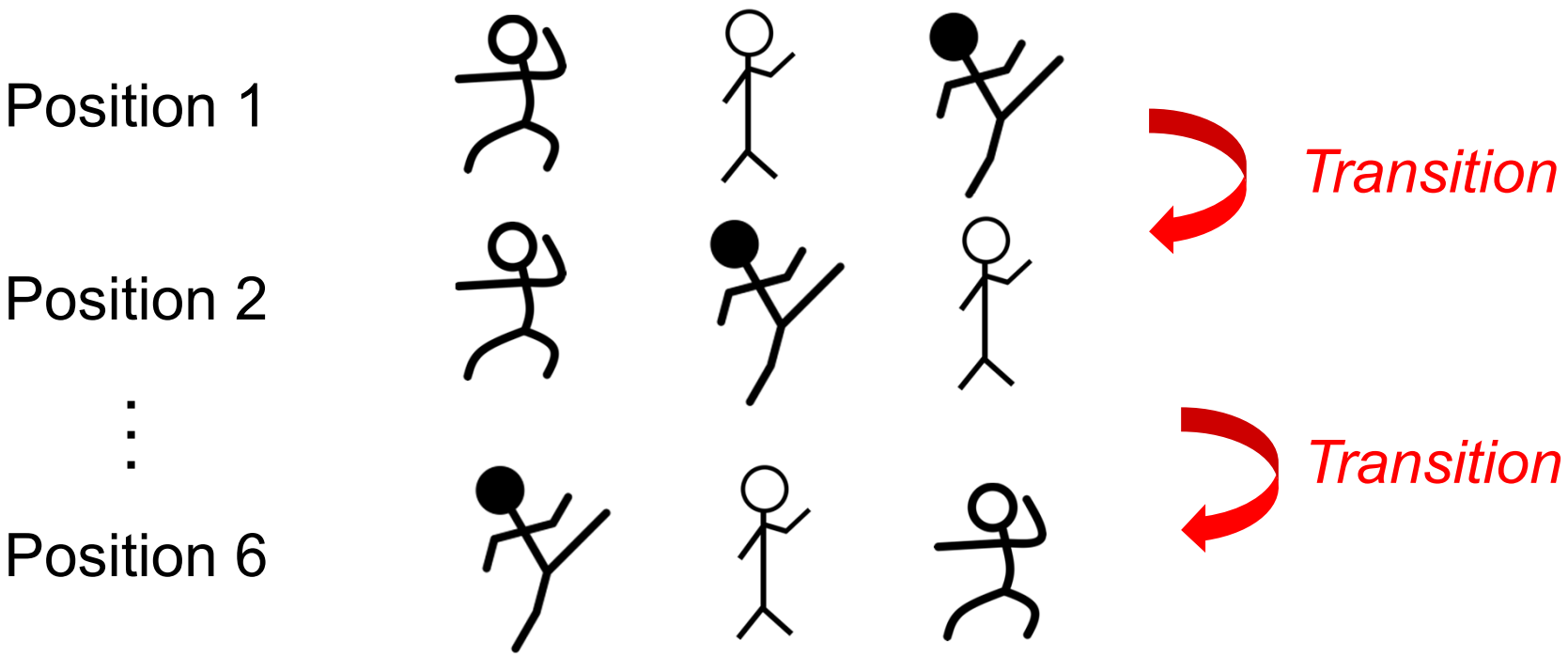


Question: How to find d1, d2, and d3?

Relative Position

Learning Approach:

- Learn the transition moves



Question: What will be your training labels?

Dashboard

- Dashboard is a type of **graphical user interface** which often provides at-a-glance views and **helps the users to quickly gain insights into the most important aspects of their data.**

- Dashboard is **a simple visual display** of **the most important information** that **the users need to achieve their objectives**.

- Be clear about what you're trying to achieve

Dashboard:

Data visualization, HCI, software engineering

- **Who are the users of your dashboard?**
- **What is the purpose of your dashboard?**
- **What do your users want to see from your dashboard?**

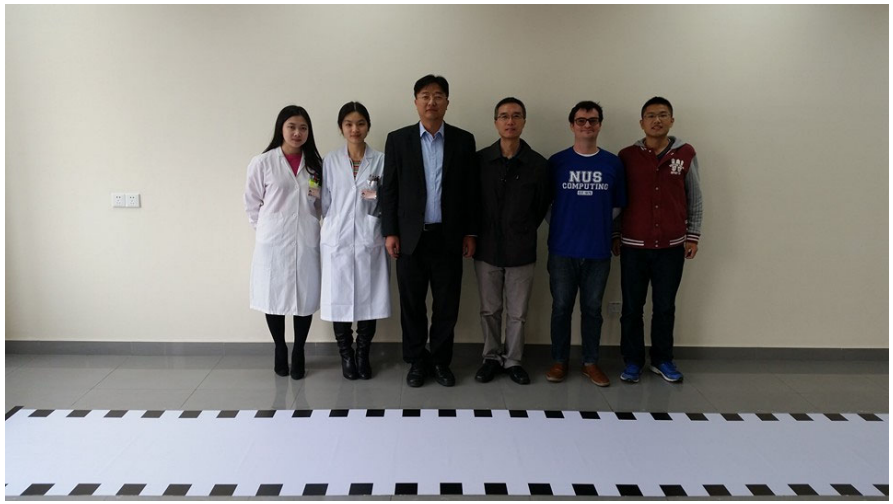
A possible dashboard for CG4002:

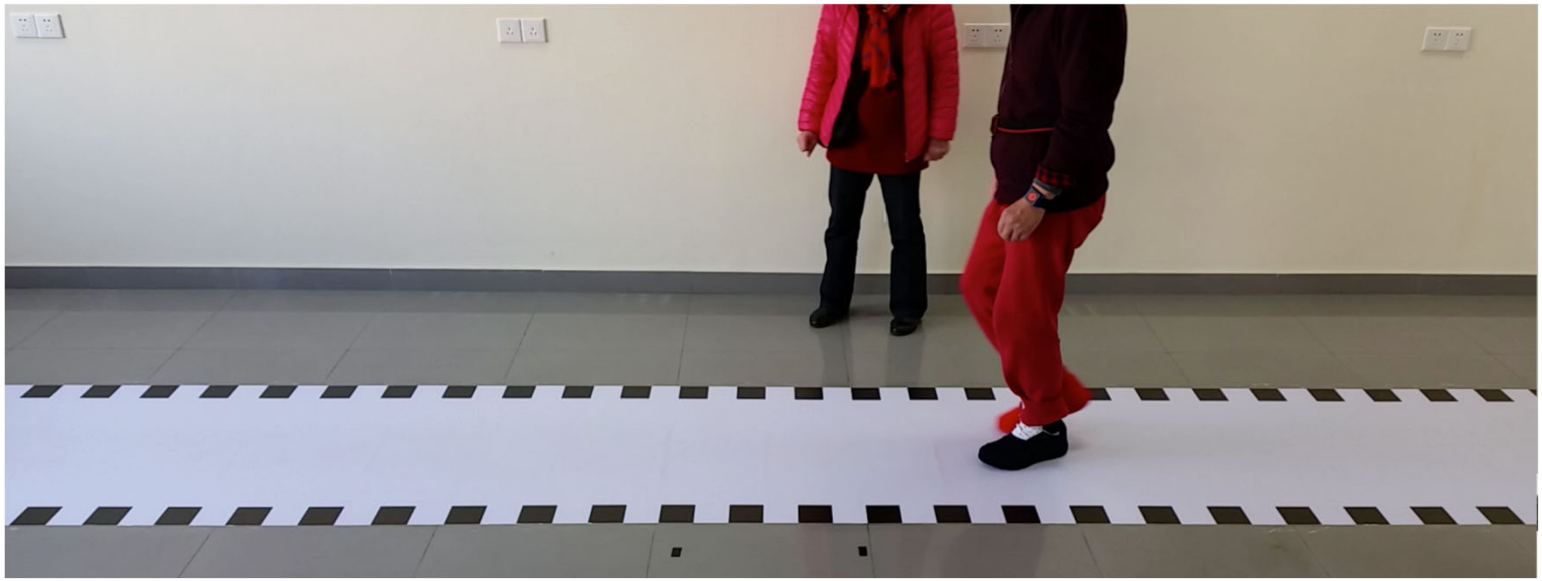
A toy dashboard

User interface that enables one to:

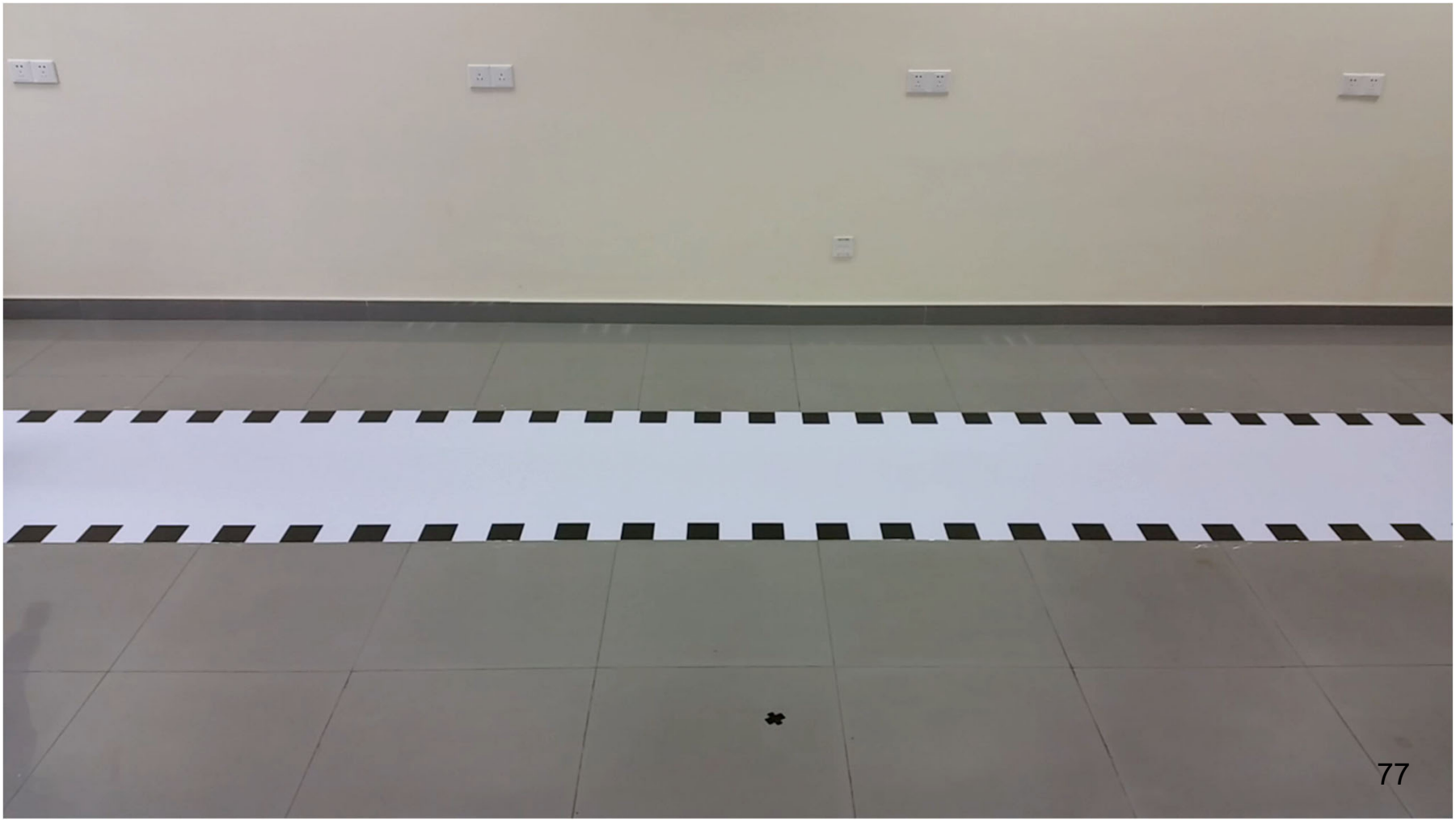
- Monitor real-time sensor readings
  - 3-axis accelerometer values
  - 3-axis gyroscope values
  - EMG signal…
- Monitor real-time predicted dance moves
- Monitor real-time predicted positions
- Input user commands
  - Record readings
  - Clear pages
  - …

MANA gait tracking system

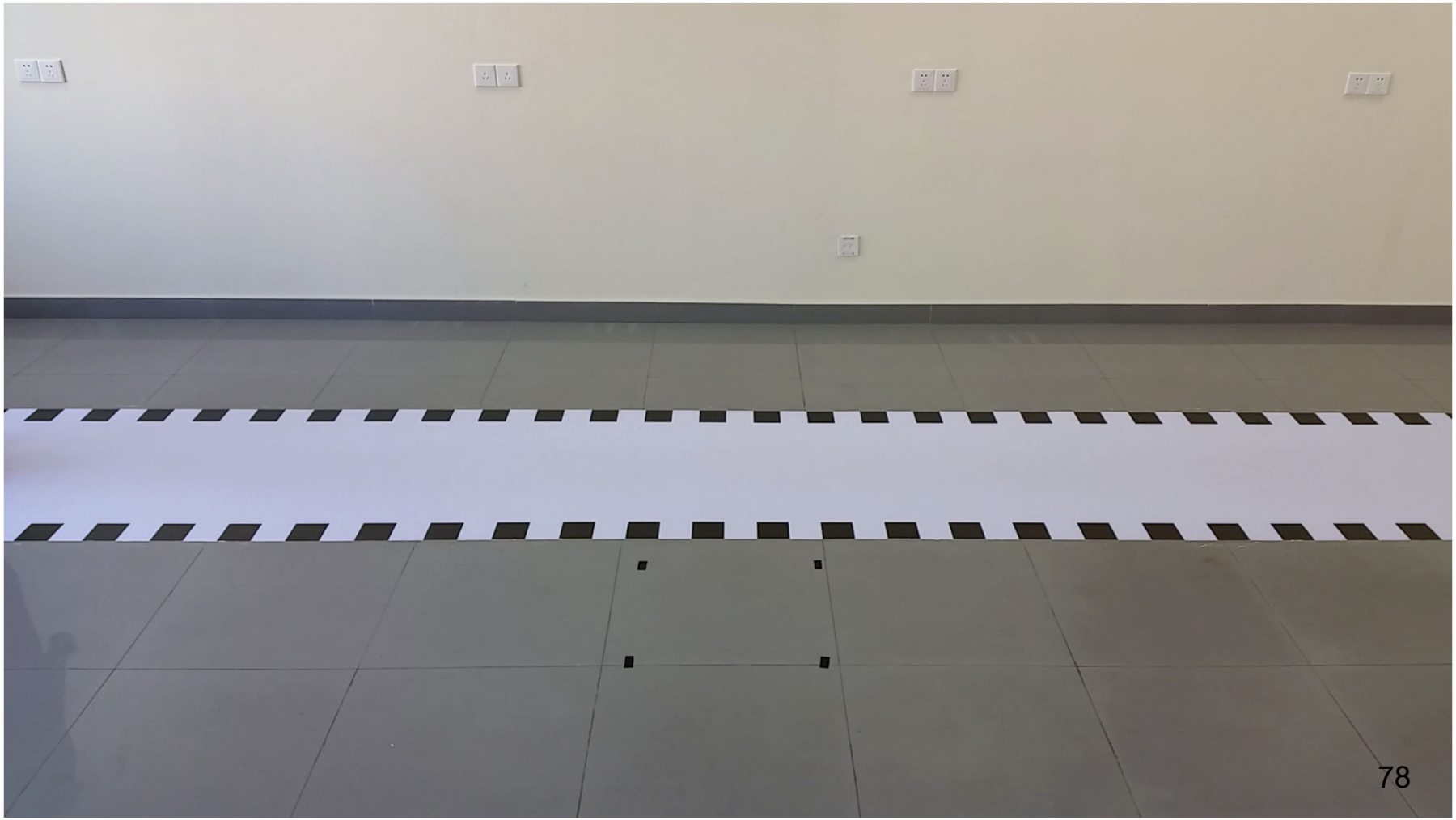




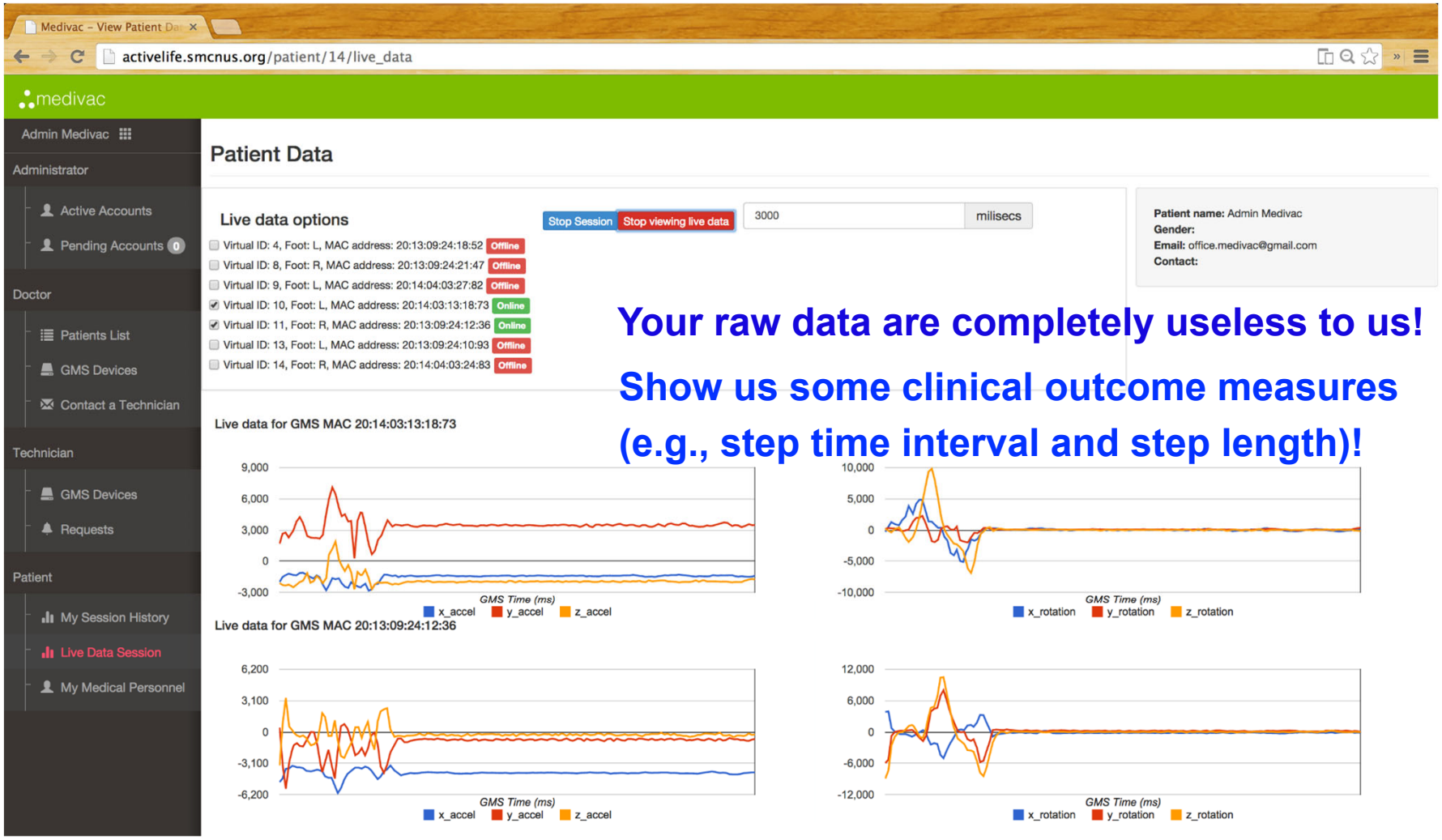
Healthy adults



Parkinson’s disease patients



Dashboard: Initial version

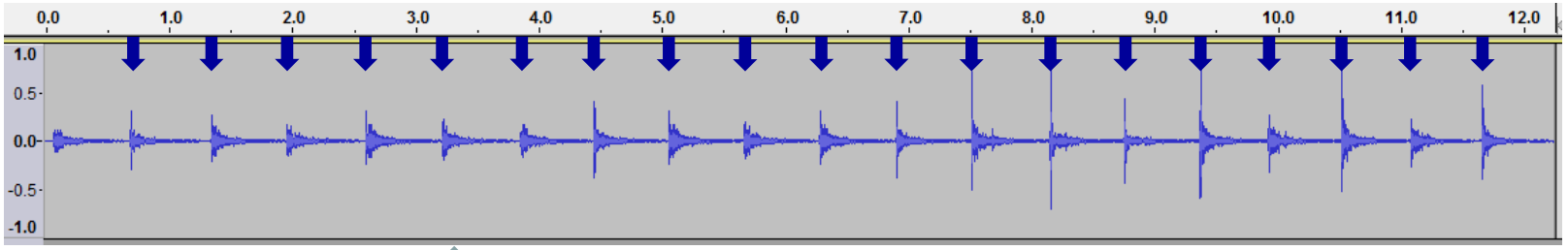


Your raw data are completely useless to us!

Show us some clinical outcome measures (e.g., step time interval and step length)!

We can do real-time visualization of IMU data, cool!

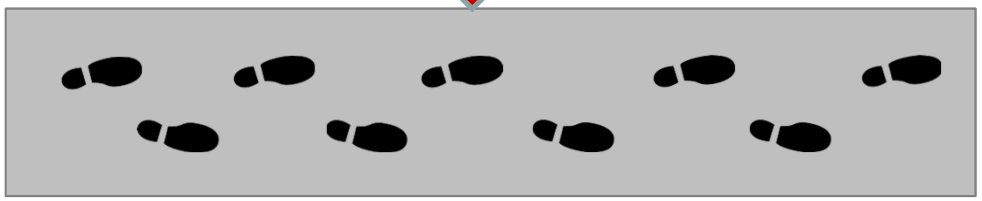
Dashboard: the 2nd version



Entrainment

Music

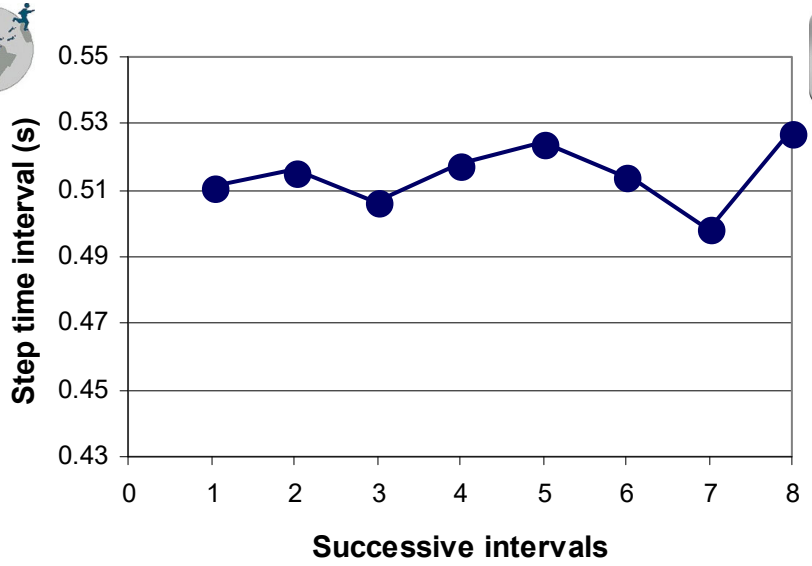
Gait



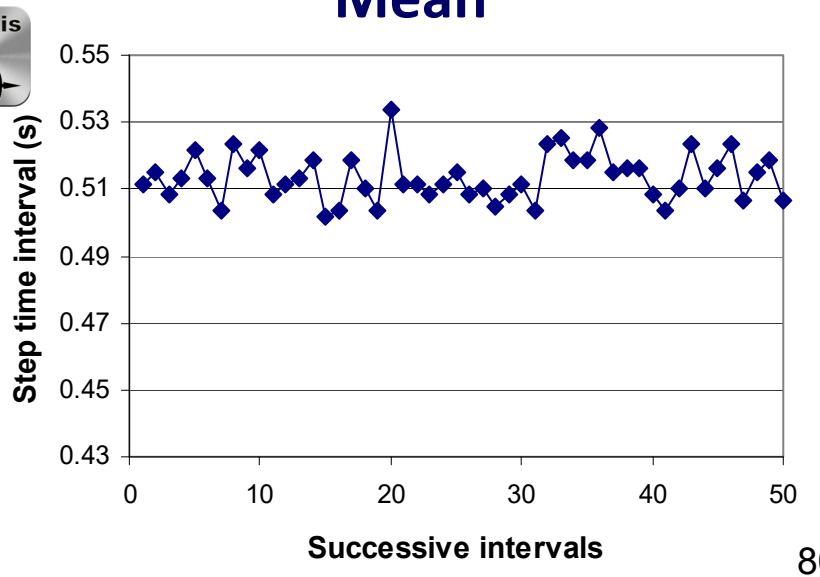
Clinical outcome measure

Coefficient of Variation (CV):

$$CV = \frac{StdDev}{Mean} \times 100 = 1.57\%$$

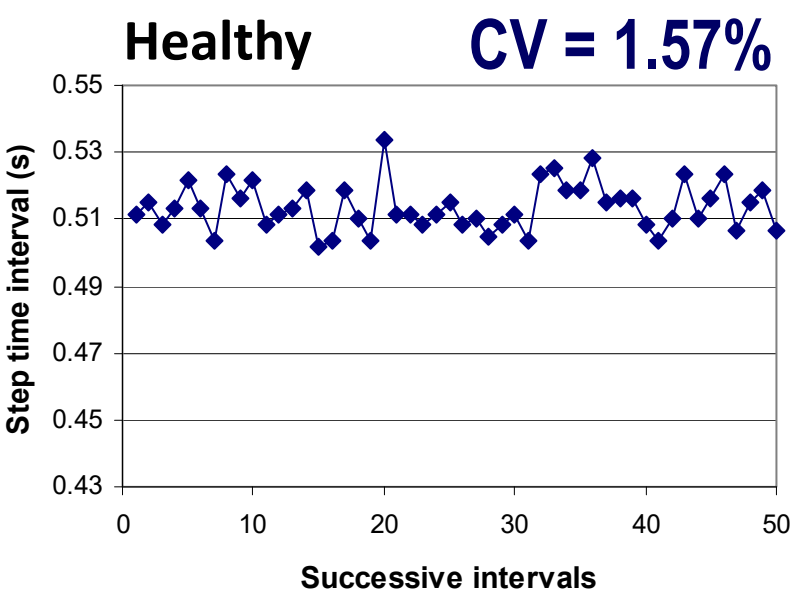


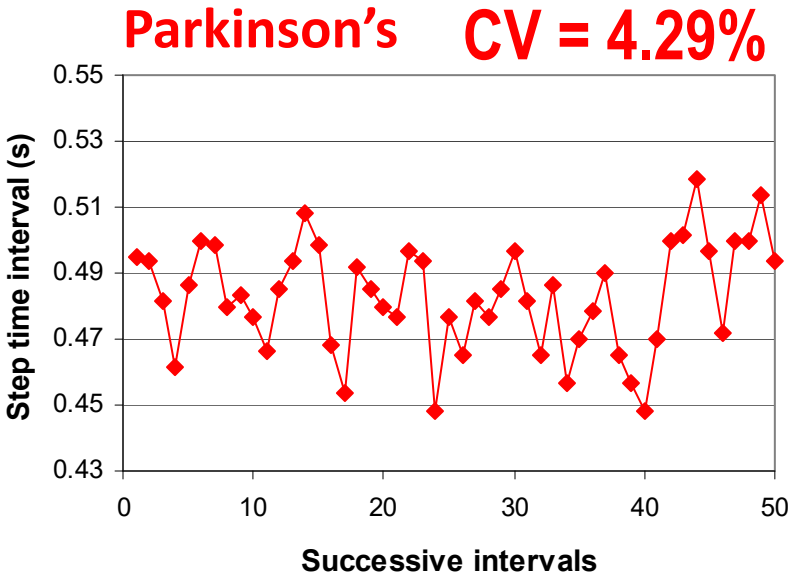
Real-time

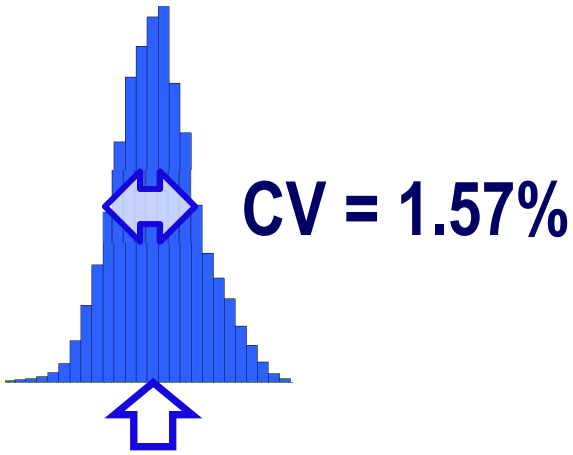


Off-line

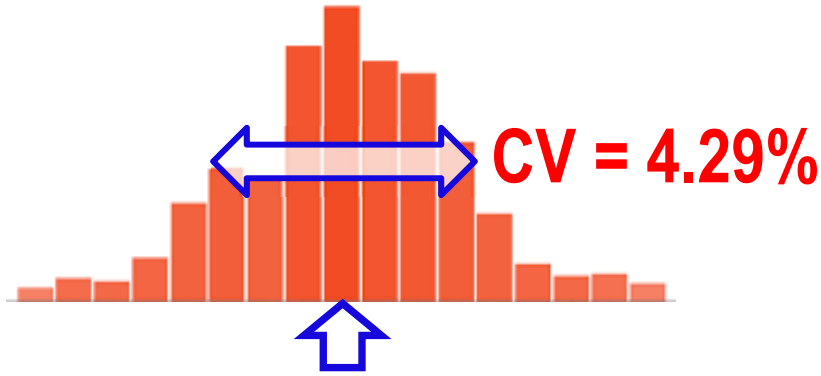
Dashboard: the 3rd version





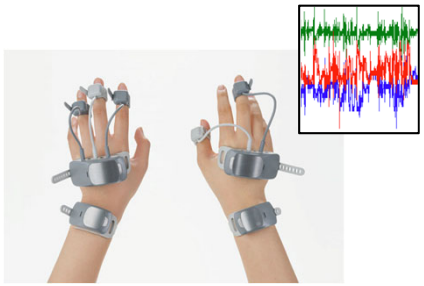


Mean step length: **1m**



Mean step length: **0.5m**

CG4002 Team work distribution

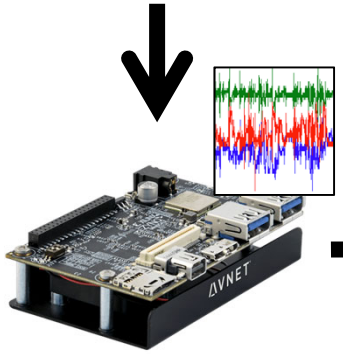


Wearable

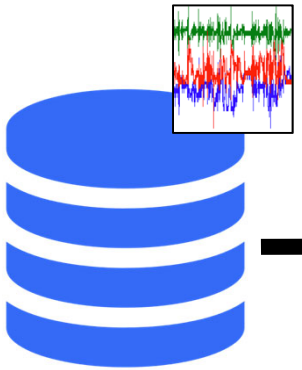
ML

User study & Creativity

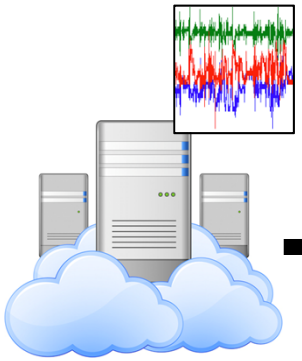
are needed for your dashboard!



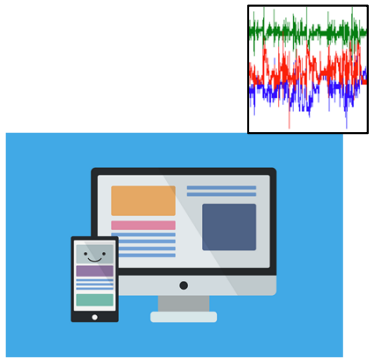
Ultra96



DB



Backend



Frontend

4 key dashboard design principles

•Makes the complex **simple**:

We have lots of information, lots of data that changes all the time and different analytical needs and questions. We want to take all this complexity and make it simple.

•Tells a clear story:

We want to be able to connect data to its context in the business and to answer the viewer's questions. This is where the visual layout of a dashboard plays a crucial role.

•Expresses the meaning of the data:

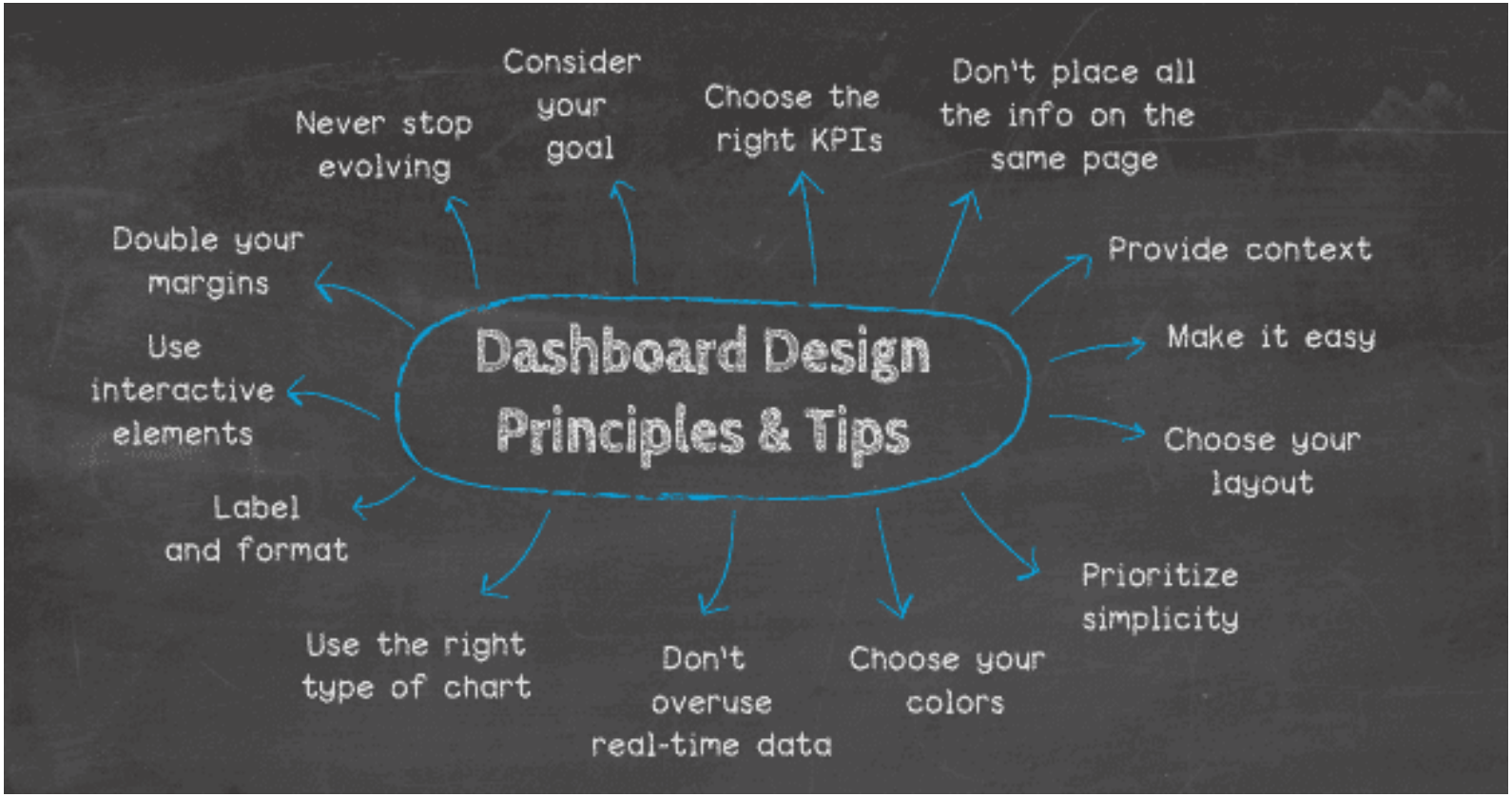
The chosen data visualizations need to correctly represent the data and the information you want to extract from it.

•Reveals details as needed:

We want each viewer to have access to the data they need — no less but also no more. Some users might need to be able to see a more granular view of the data — others could suffice with an overview.

** Adapted from Sisense*

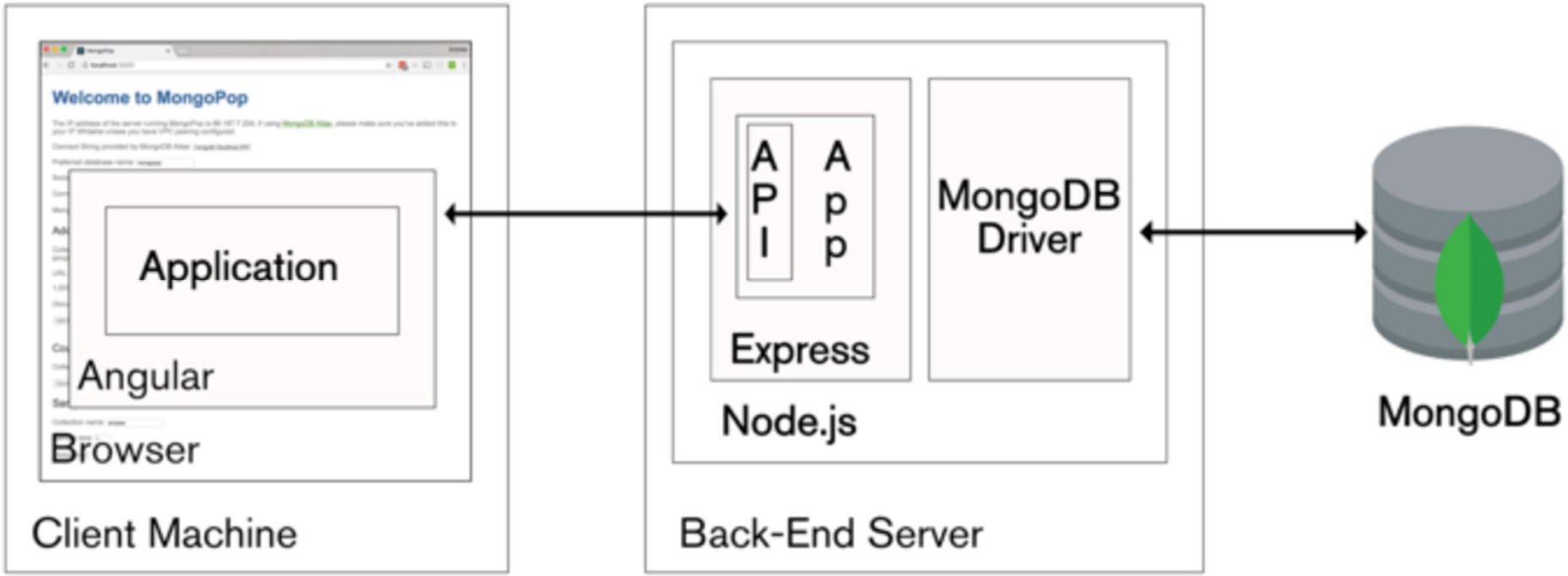
14 dashboard design principles



** Figure adapted from Datapine*

MEAN: a full stack web app

MongoDB **E**xpress **A**ngular **N**ode.js



https://www.youtube.com/watch?v=wtIvu085uU0

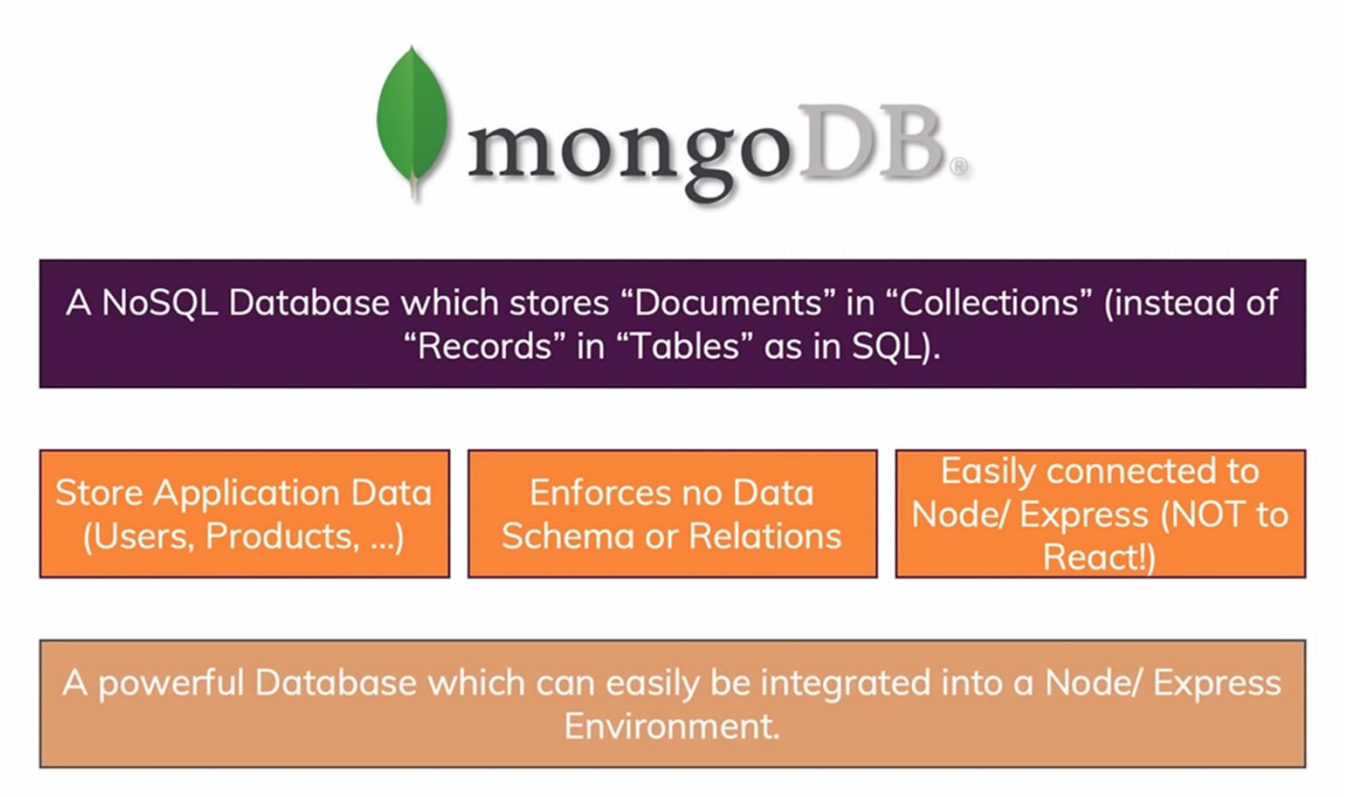
** Slides modified from Acade Mind*

MERN: another full stack web app

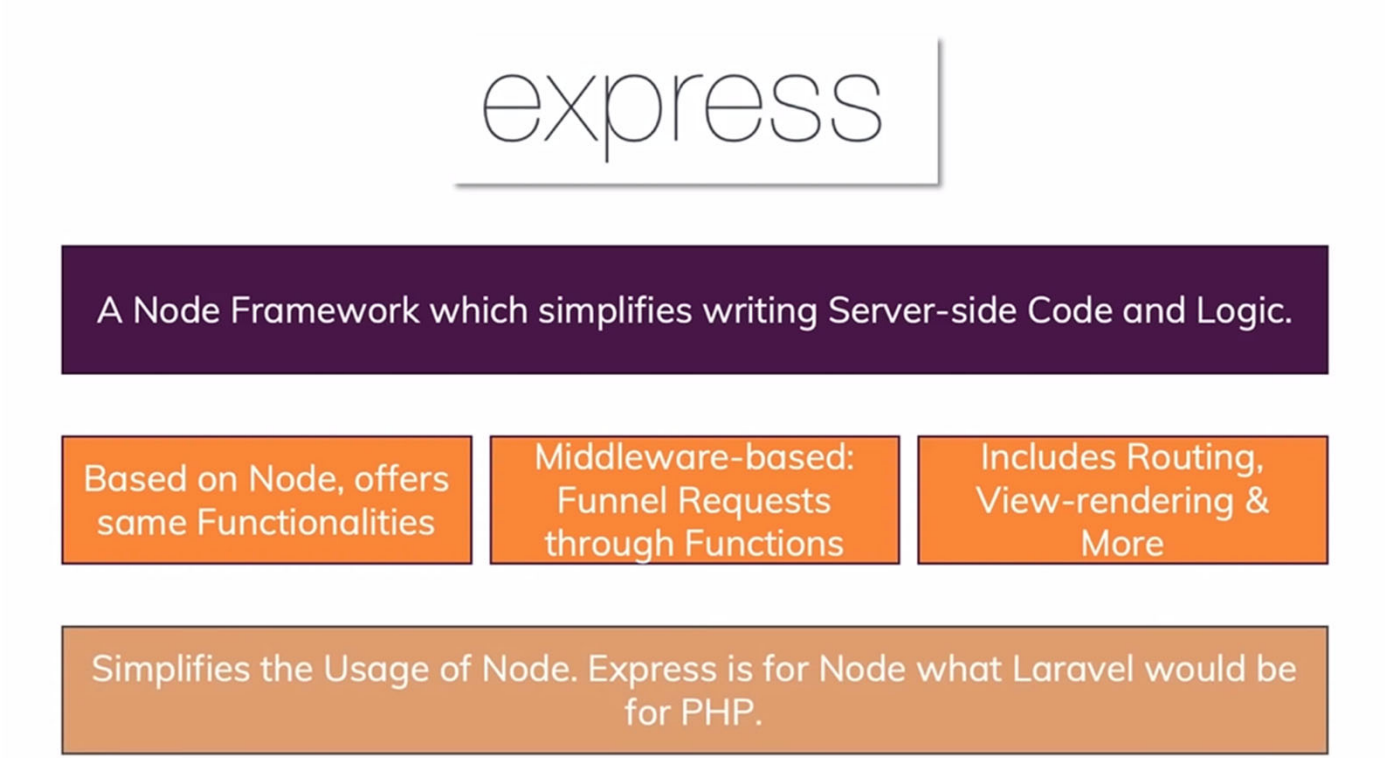
It is a highly scalable and fast architecture and works together particularly well

https://www.youtube.com/watch?v=rpJO0T08Bnc

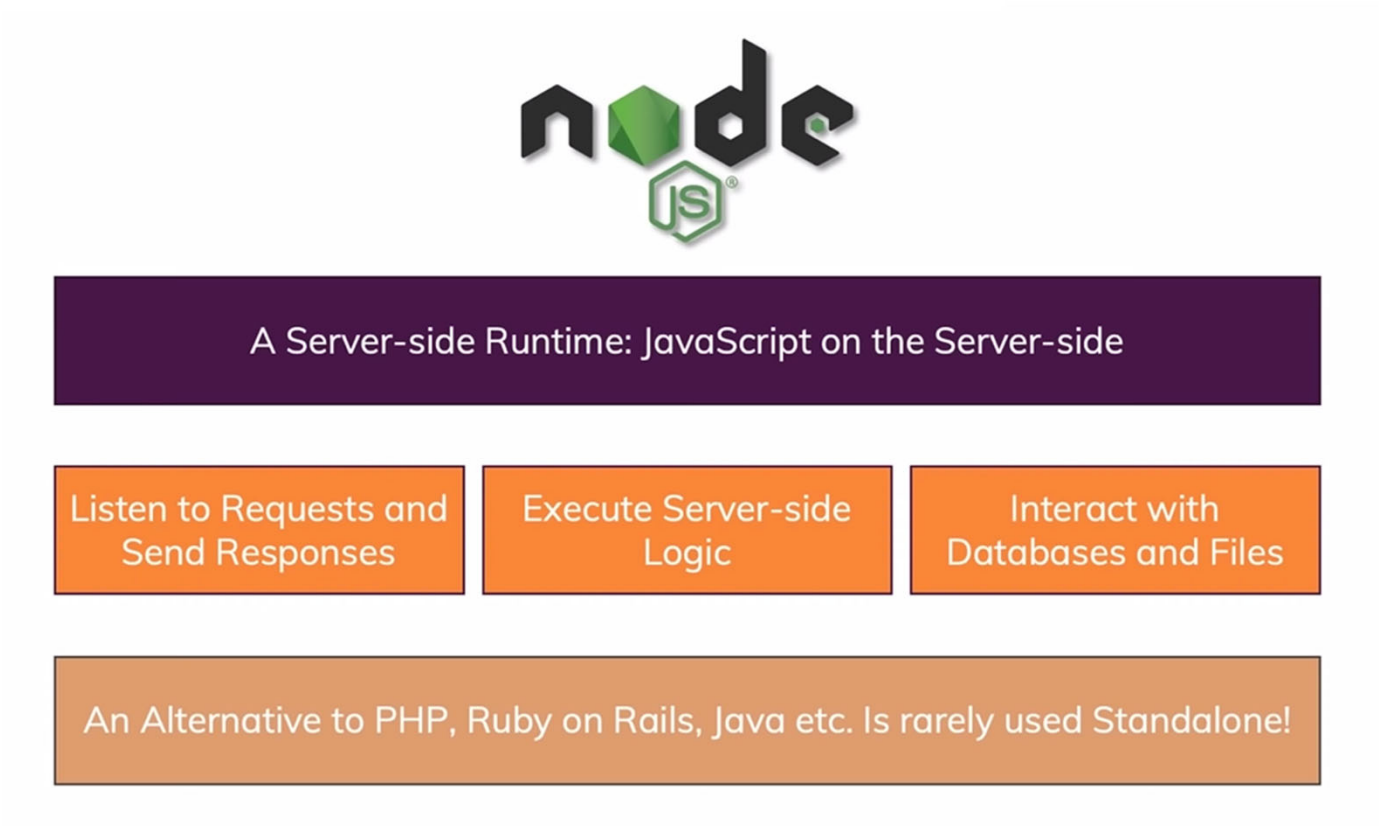
https://www.youtube.com/watch?v=7CqJlxBYj-M



It is a database engine —> all about storing data & its high performance is due to its flexibility

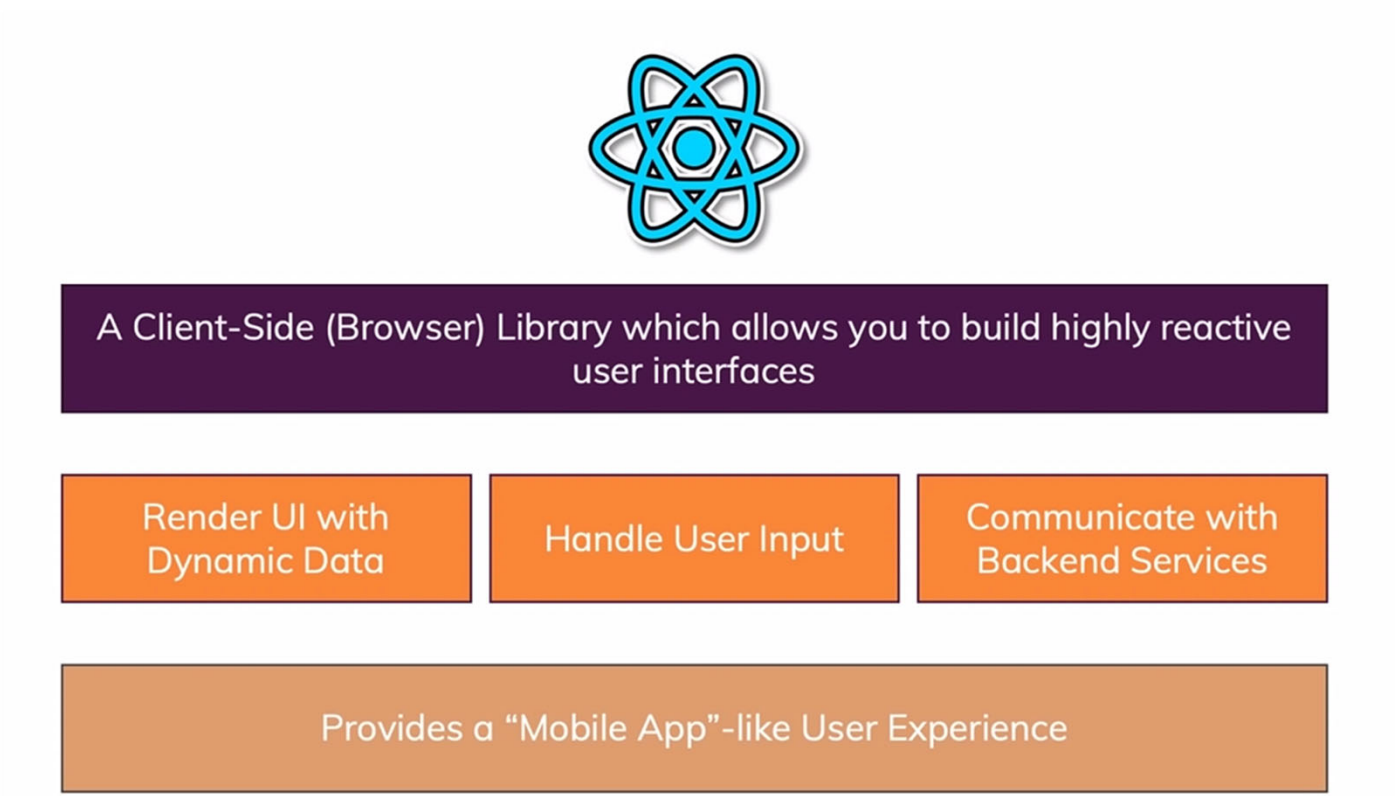


Express is a middleman. It talks to DB server, sends DB queries (using MongoDB library). Client side does NOT send requests to DB directly due to security issues.



Node.js is also a middleman. It executes javascript outside of browser, creates web server to interact with databases. Used in combination with express.

React



It controls what users see on screen and how user Interacts (e.g., button clicks…)

React Native will allow you to build cross platform mobile user Interfaces but not needed for CG4002

Conclusion

- Sensors useful in identifying activities
- Classification model helps to predict dance moves
  - Need features, segmentation techniques, ...
- Many methods to identify positions
  - Localization using triangulation
  - Learning approach
- Dashboard enables user interface