Machine Learning Schemes

Jamal Raiyn

jamal.raiyn@th-ab.de

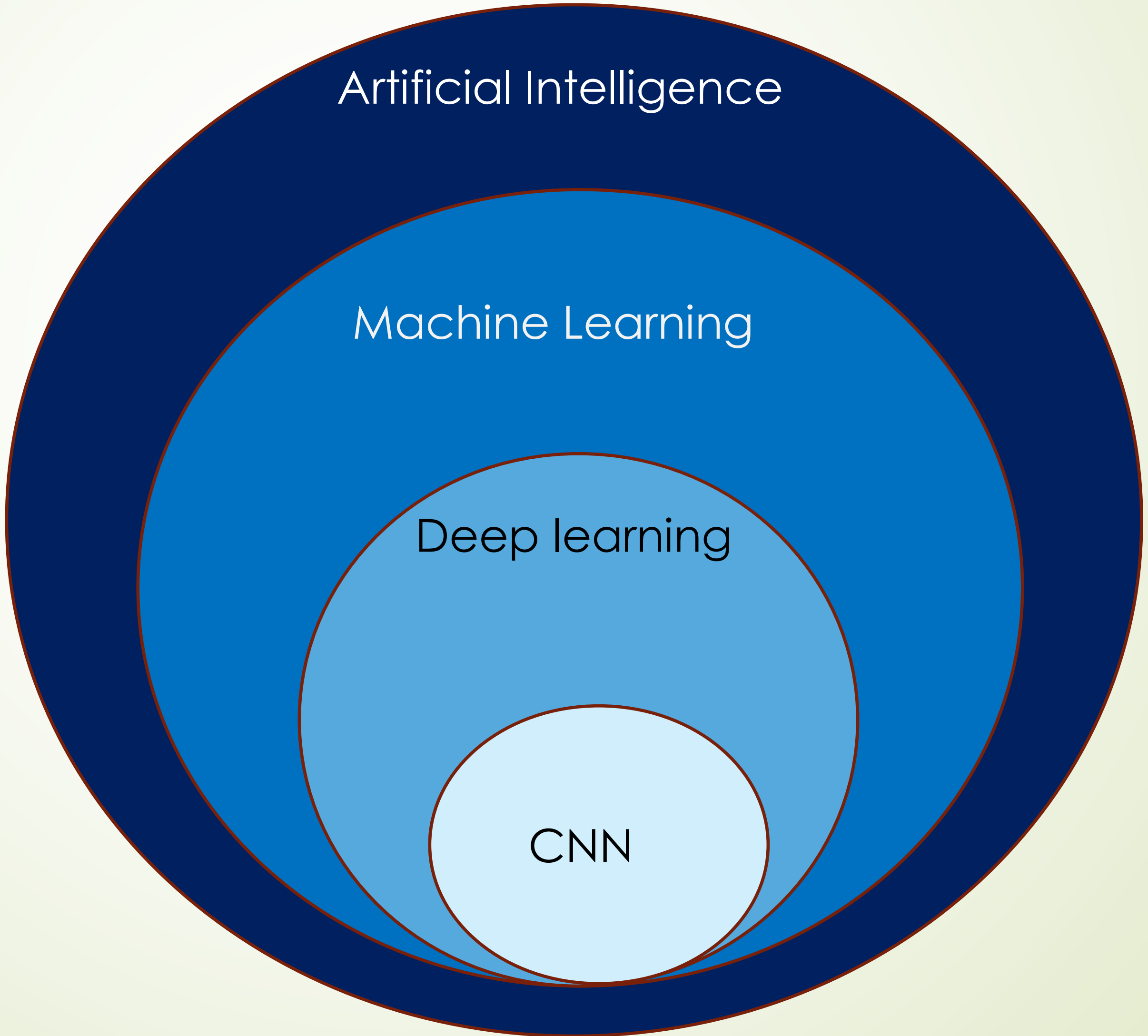
1

Artificial intelligence in applications. Modeling, Machine Learning and Data Classifier Performance

9:45 am - 11:15 am

10-04-2024

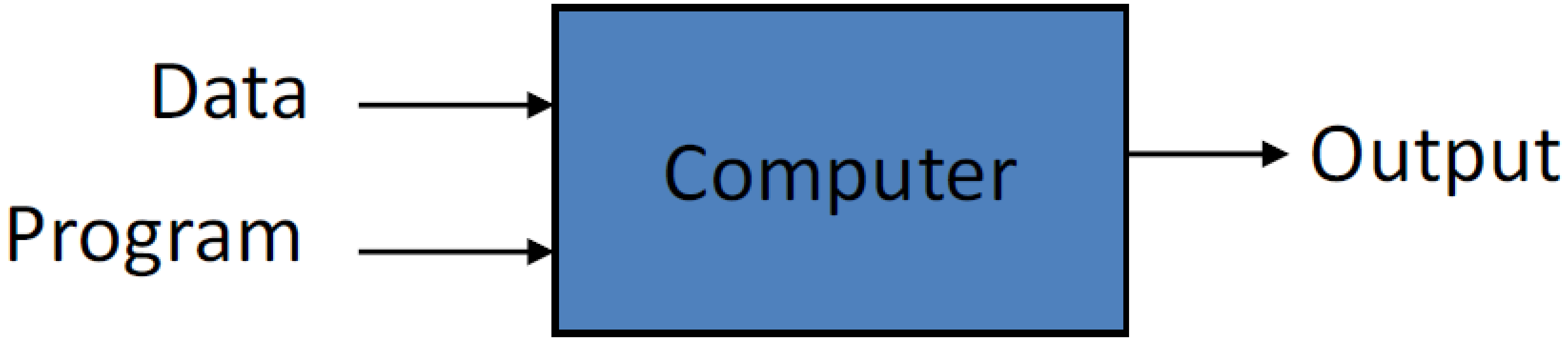
AI, ML


Artificial Intelligence
Machine Learning
Deep learning
CNN

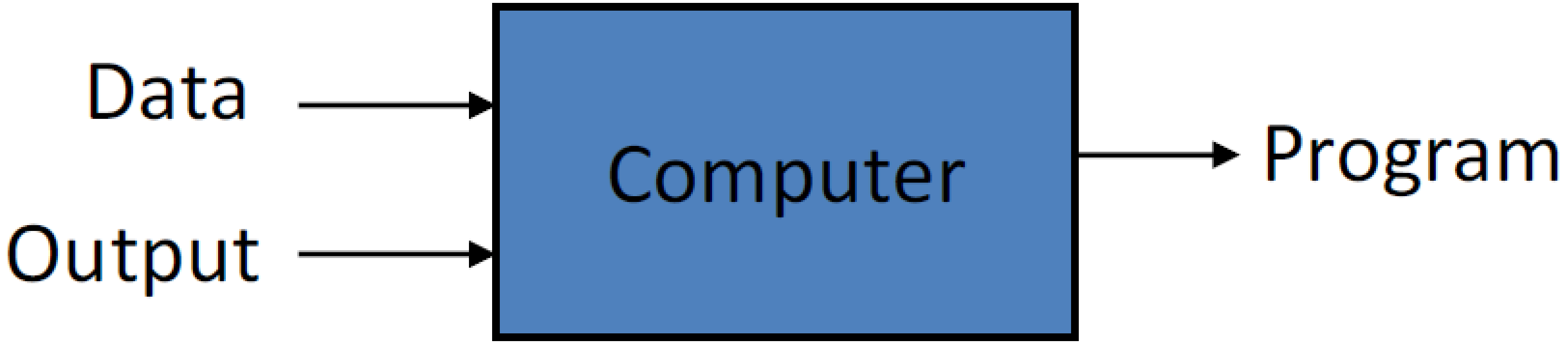
Machine Learning (Herbert Simon)

Learning is any process by which a system improves performance from experience.

Traditional Programming



Machine Learning



Machine learning Algorithm

Machine Learning is an application of artificial intelligence where a computer/machine learns from the past experiences (input data) and makes future predictions. The performance of such a system should be at least human level.

Diverse Data Types

Prediction

Decision Making

Learning from Data

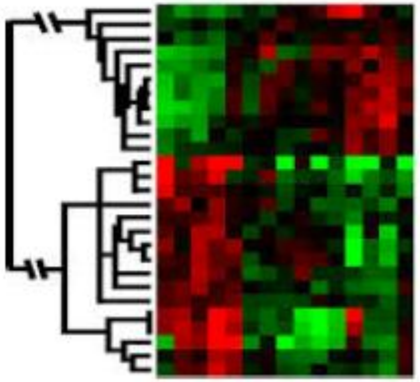
Practical use of ML:
Where can we find ML used?

- Self driving cars (Tesla)
- Voice interfaces (Alexa, Siri)
- Face recognition (Google Photos)
- Recomender systems (Netflix, Amazon)
- Games (AlphaGo)
- Character recognition (Post offices)
- Banking systems
- Medical diagnosis
- ML for Human-Computer Interaction

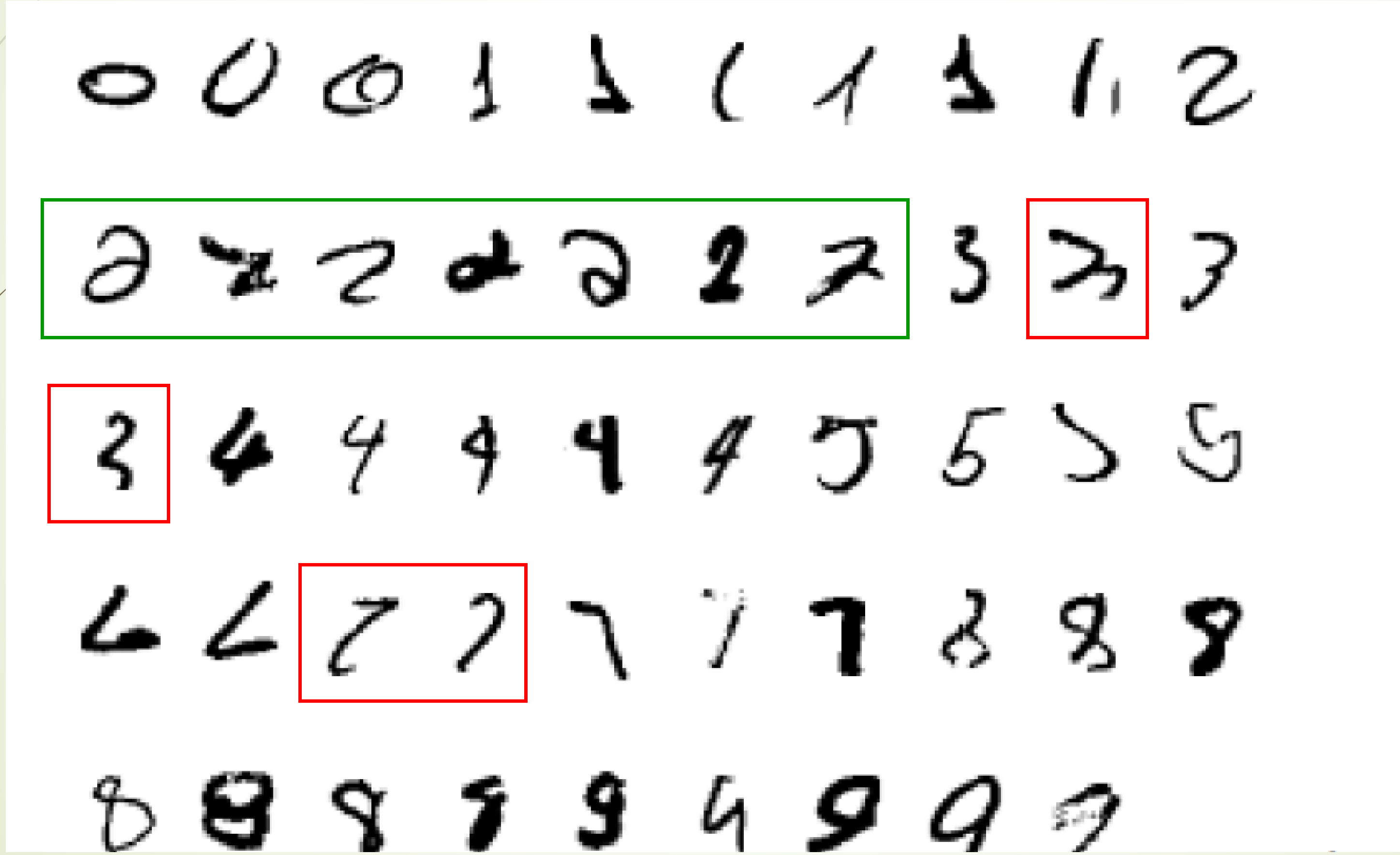
When Do We Use Machine Learning?

- ML is used when:
  - Human expertise does not exist (navigating on Mars)
  - Humans can’t explain their expertise (speech recognition)
  - Models must be customized (personalized medicine)
  - Models are based on huge amounts of data (genomics)



A classic example of a task that requires machine learning:
It is very hard to say what makes a 2

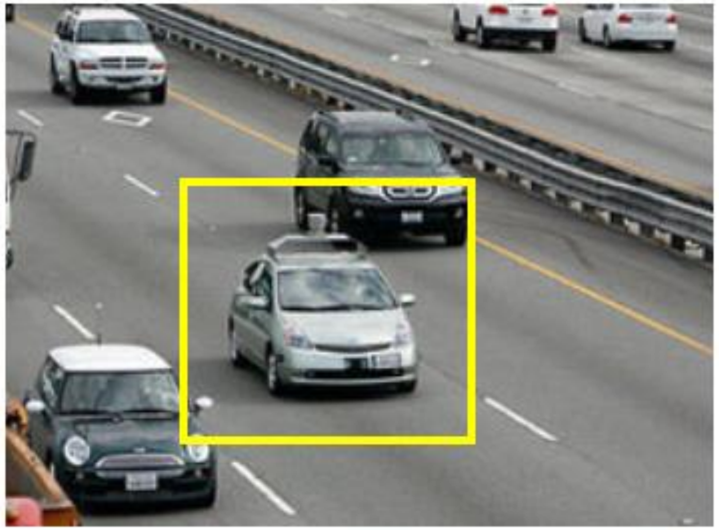


Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - – Facial identities or facial expressions
  - – Handwritten or spoken words
  - – Medical images
- Generating patterns:
  - – Generating images or motion sequences
- Recognizing anomalies:
  - – Unusual credit card transactions
  - – Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - – Future stock prices or currency exchange rates
- Slide credit: Geoffrey Hinton

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software

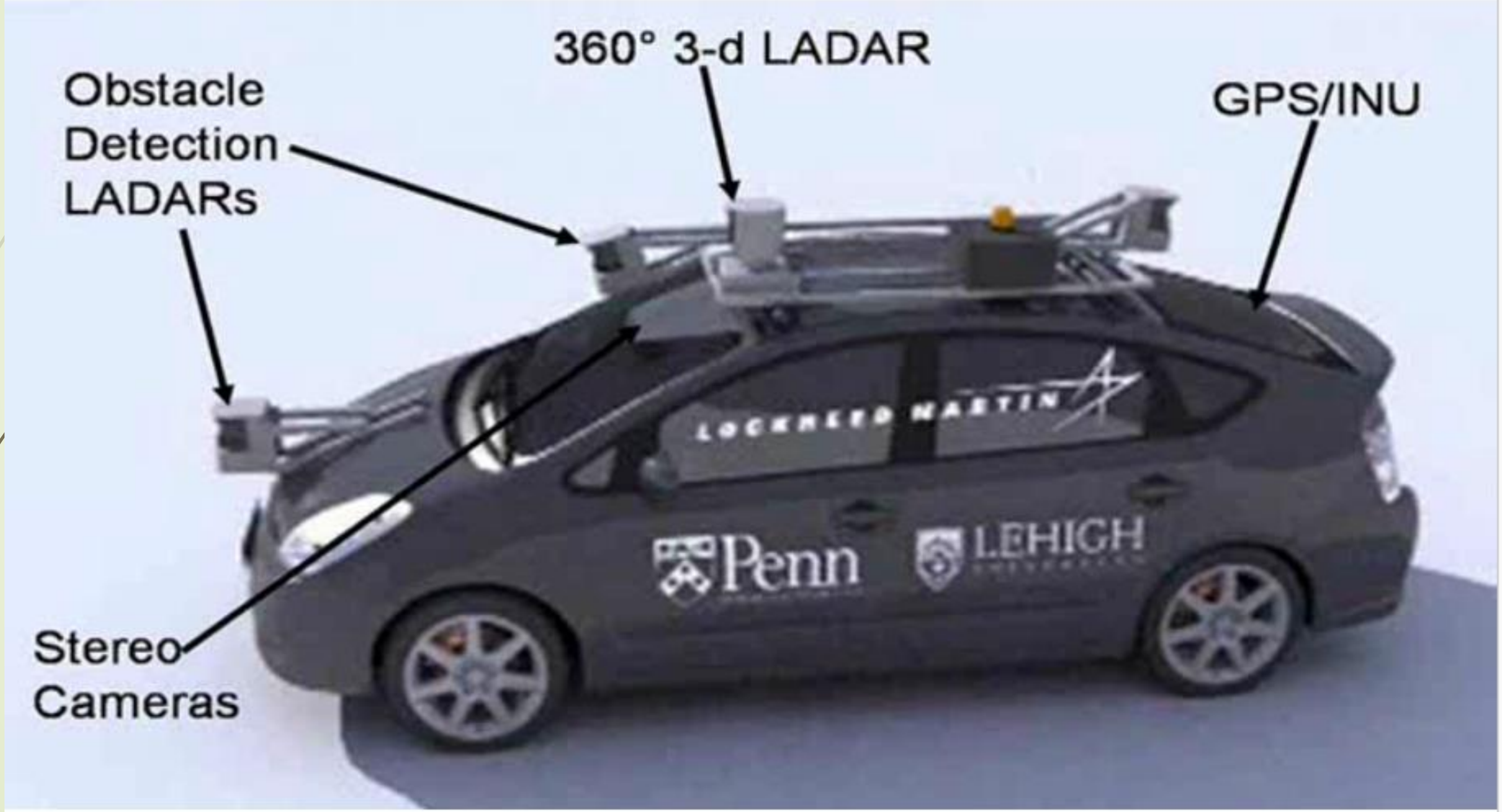
Autonomous Cars



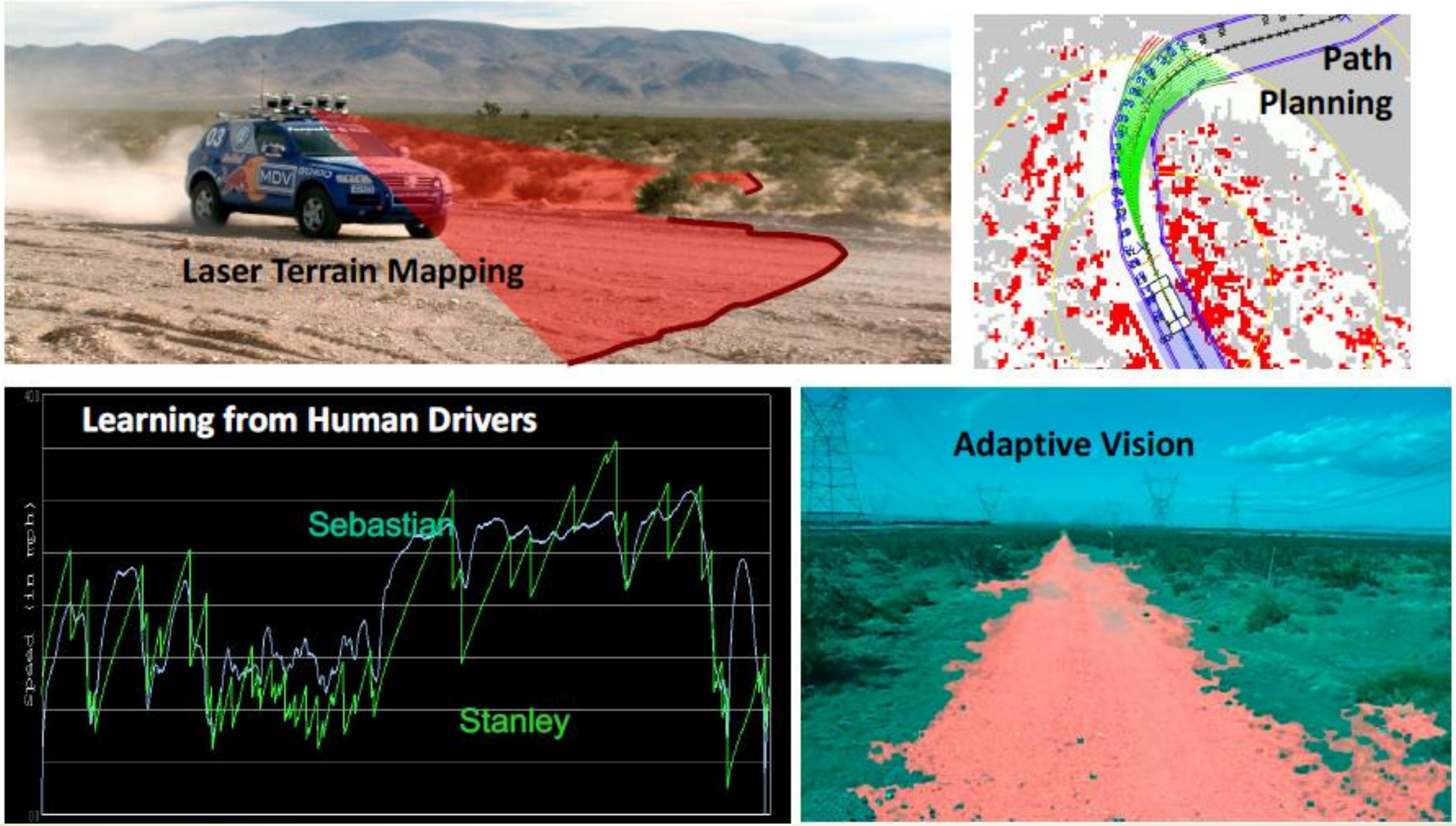
- Nevada made it legal for autonomous cars to drive on roads in June 2011
- As of 2013, four states (Nevada, Florida, California, and Michigan) have legalized autonomous cars

Penn's Autonomous Car →

Autonomous Car Sensors



Autonomous Car Technology



Types of Learning

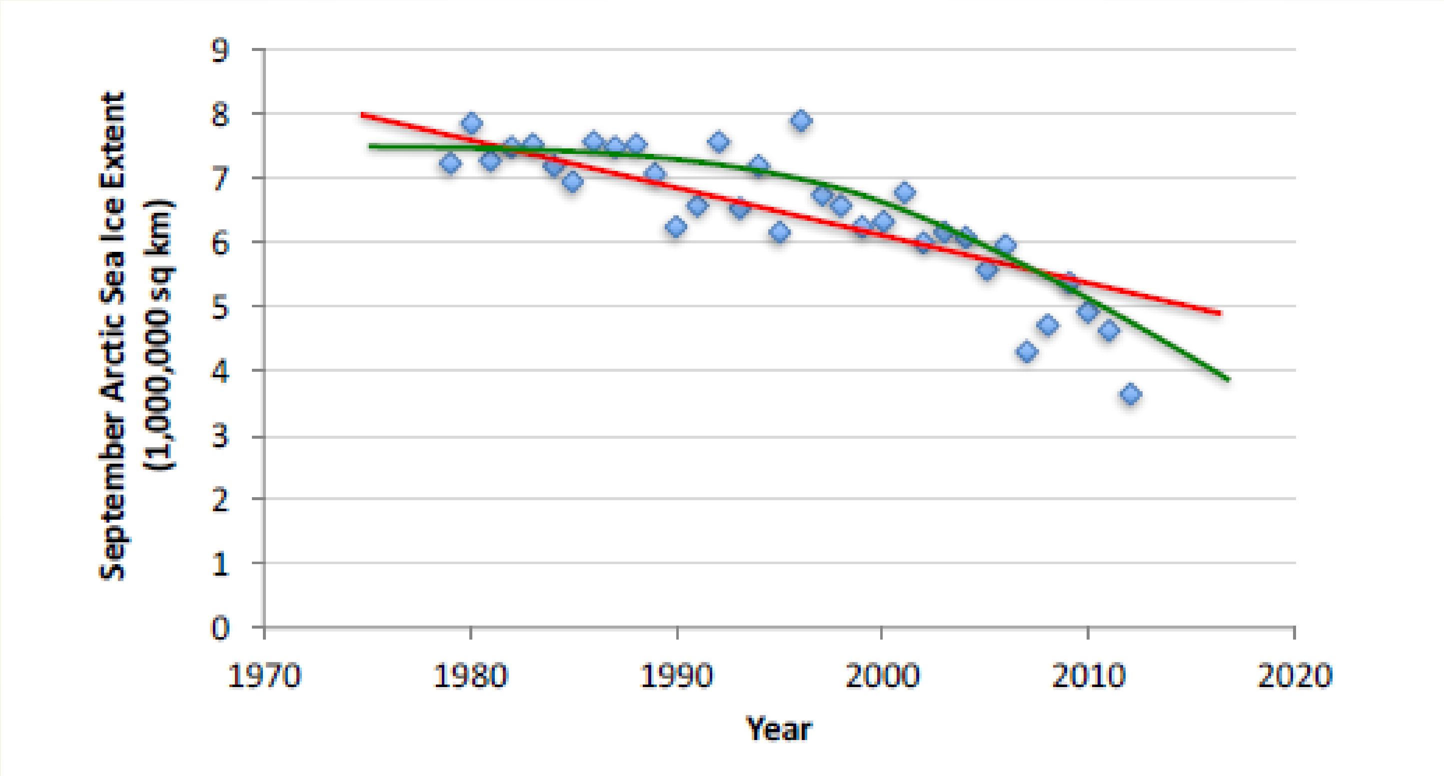
- **Supervised (inductive) learning**
  - Given: training data + desired outputs (labels)
- **Unsupervised learning**
  - Given: training data (without desired outputs)
- **Semi-supervised learning**
  - Given: training data + a few desired outputs
- **Reinforcement learning**
  - Rewards from sequence of actions

Supervised Learning: Regression

• Given (x1, y1), (x2, y2), ..., (xn, yn)

• Learn a function f(x) to predict y given x

– y is real-valued == regression

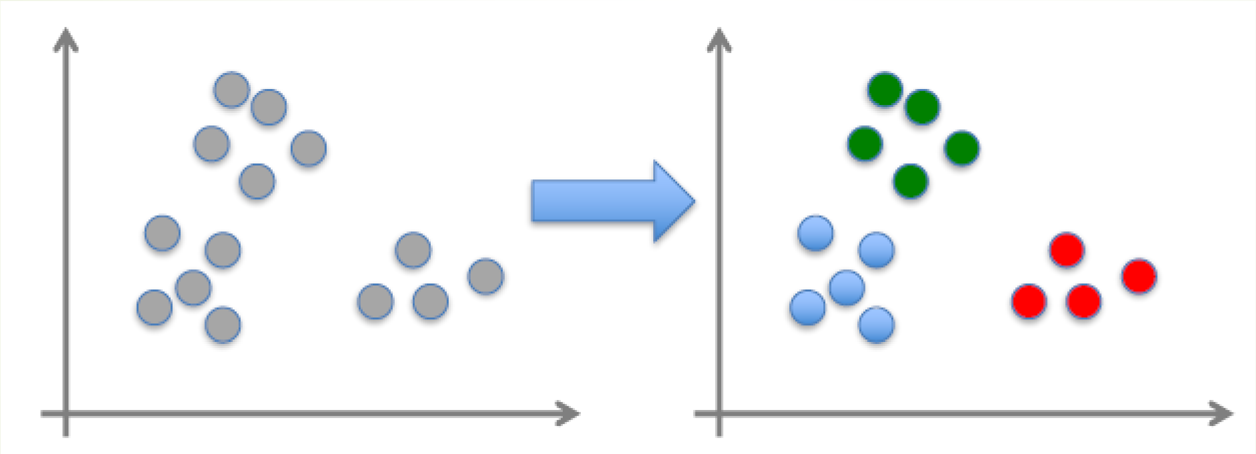


Unsupervised Learning

• Given x1, x2, ..., xn (without labels)

• Output hidden structure

– E.g., clustering



Designing a Learning System

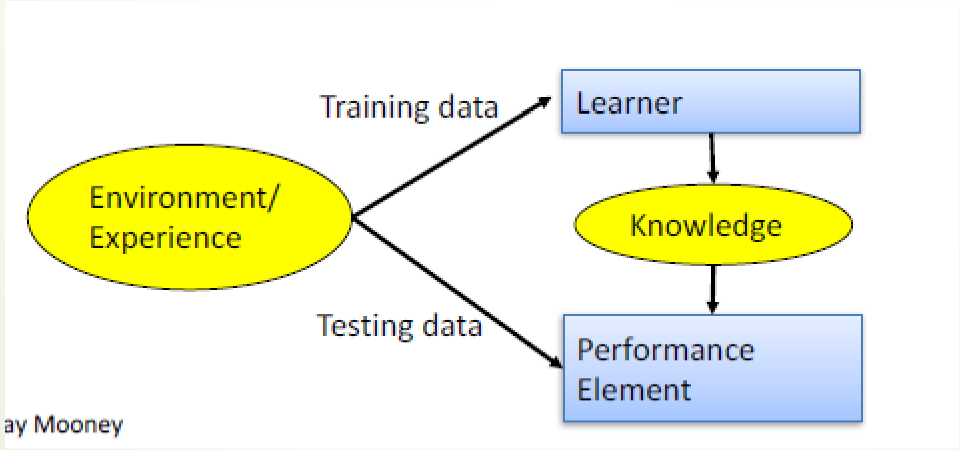
• Choose the training experience

• Choose exactly what is to be learned

– i.e. the ***target function***

• Choose how to represent the target function

• Choose a learning algorithm to infer the target function from the experience

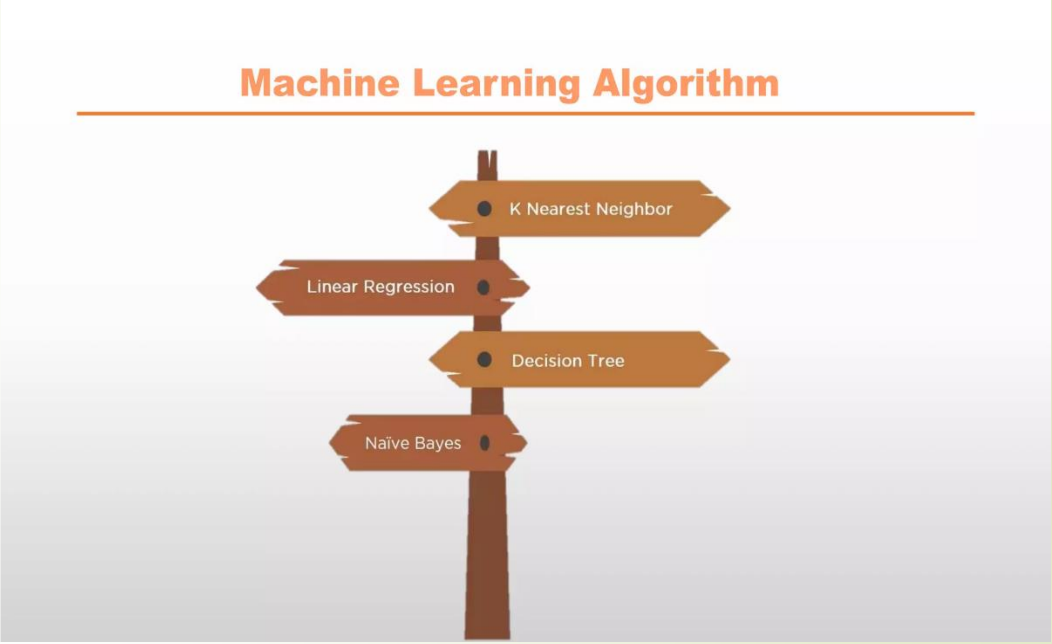


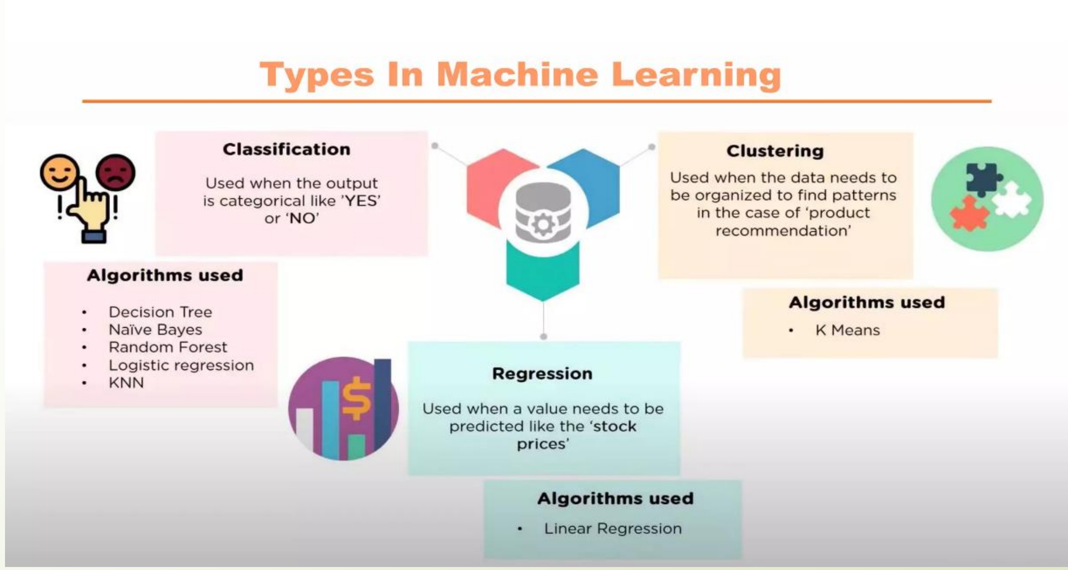
Every ML algorithm has three components:

- – **Representation**
- – **Optimization**
- – **Evaluation**

Various Function Representations

- Numerical functions
  - – Linear regression
  - – Neural networks
  - – Support vector machines
- Symbolic functions
  - – Decision trees
  - – Rules in propositional logic
  - – Rules in first-order predicate logic
  - - Instance-based functions
  - – Nearest-neighbor
- Probabilistic Graphical Models
  - – Naïve Bayes
  - – Bayesian networks
  - – Hidden-Markov Models (HMMs)
  - – Markov network

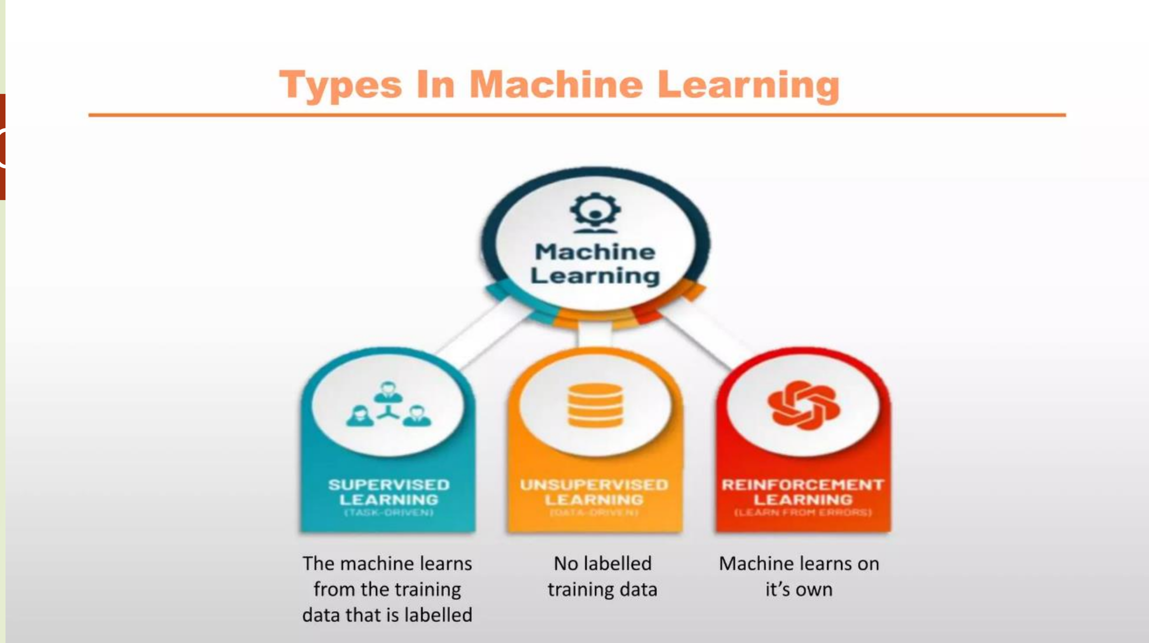


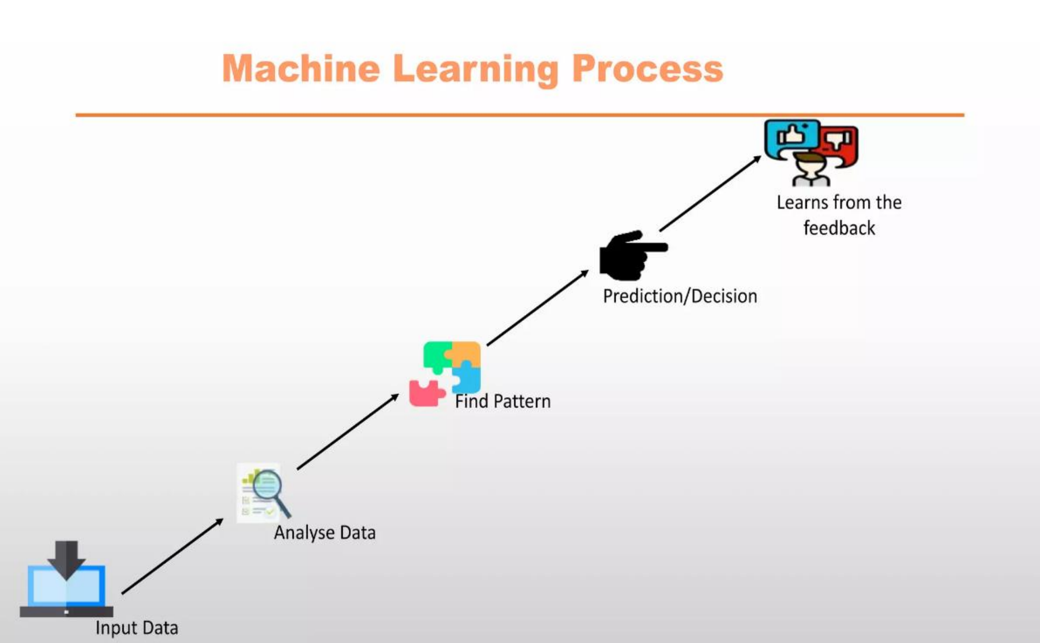


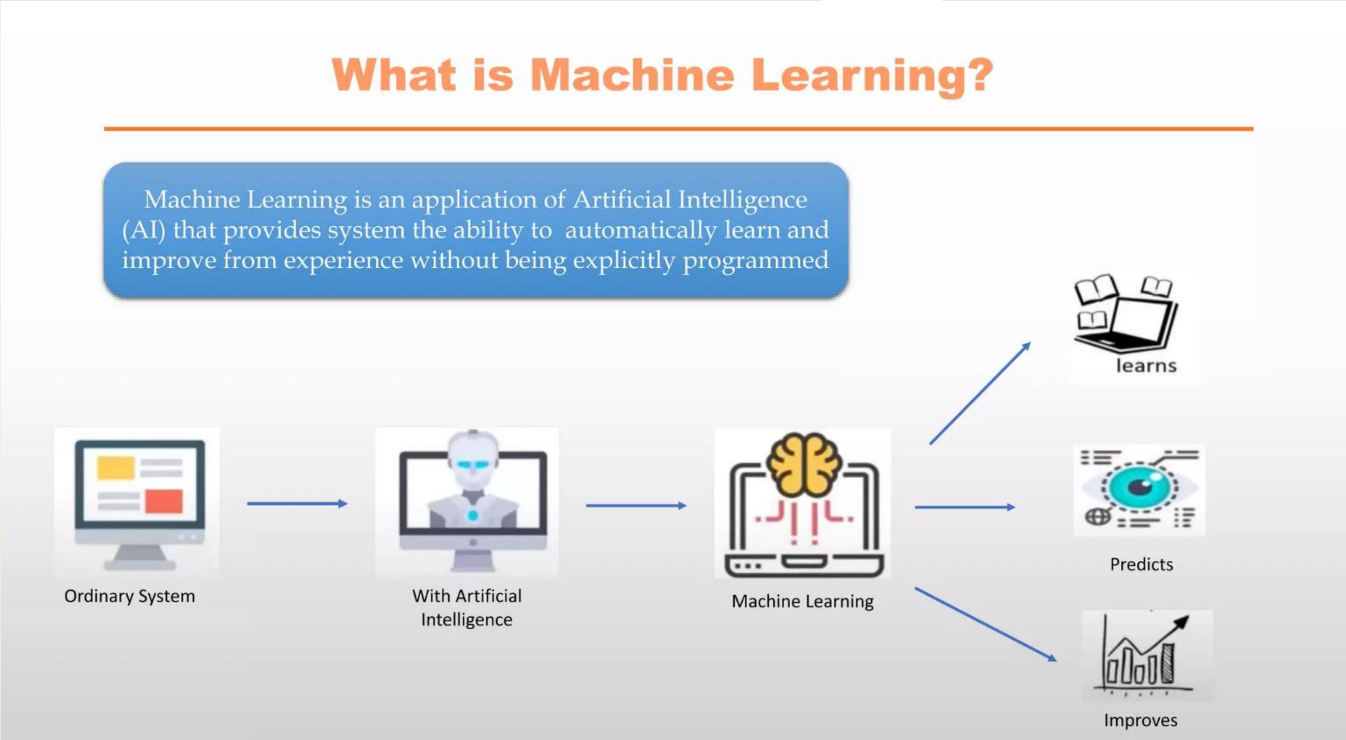
Evaluation

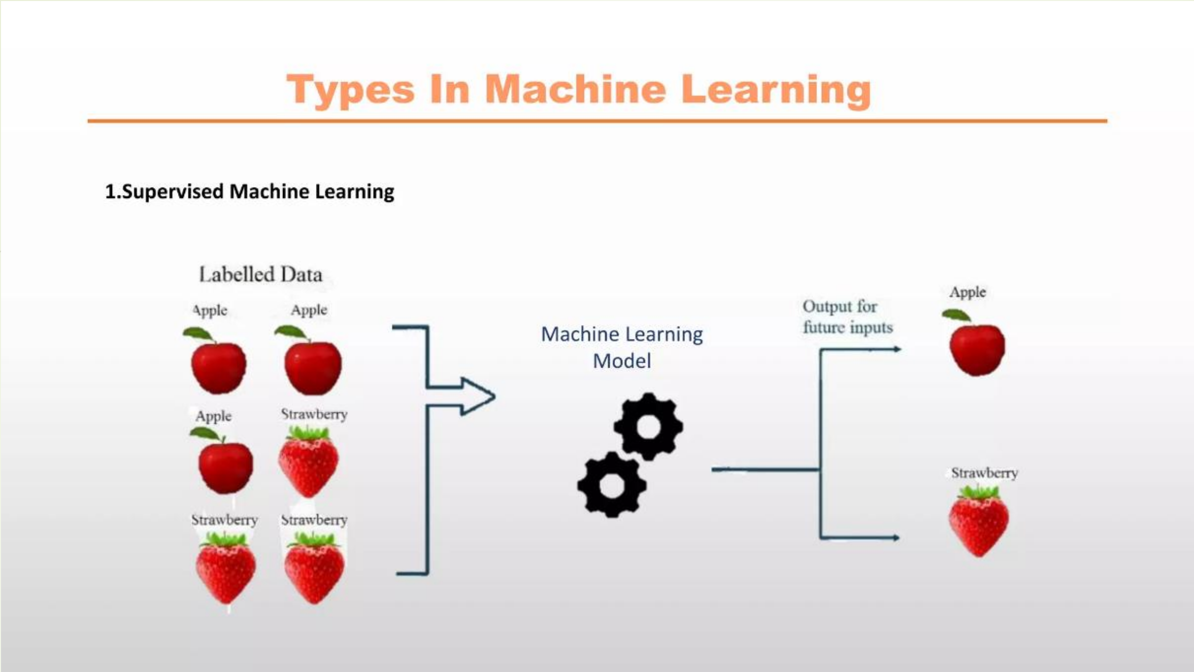
- Accuracy
- Precision and recall
- Squared error
- Likelihood
- Posterior probability
- Cost / Utility
- Margin
- Entropy

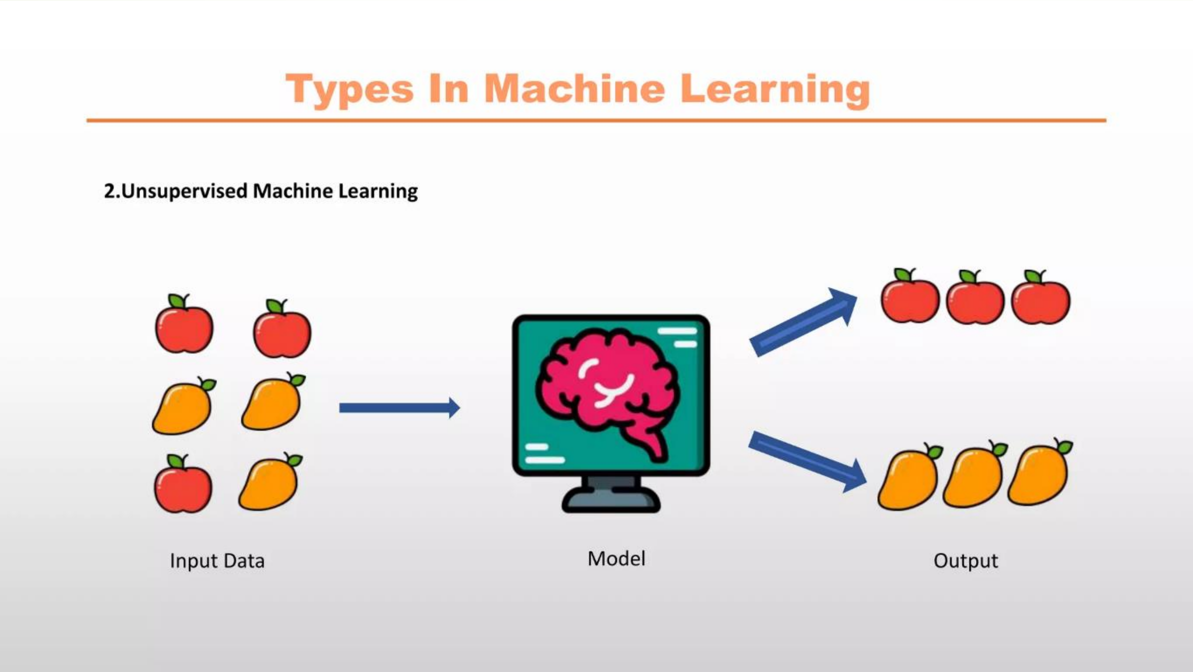
- etc.


Types In Machine Learning
Machine Learning
SUPERVISED LEARNING
(TASK-DRIVEN)
UNSUPERVISED LEARNING
(DATA-DRIVEN)
REINFORCEMENT LEARNING
(LEARN FROM ERRORS)
The machine learns from the training data that is labelled
No labelled training data
Machine learns on it's own


Machine Learning Process
Learns from the feedback
Prediction/Decision
Find Pattern
Analyse Data
Input Data

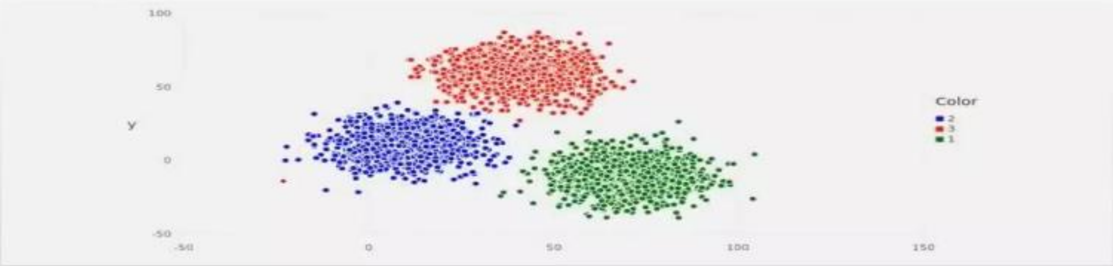

What is Machine Learning?
Machine Learning is an application of Artificial Intelligence (AI) that provides system the ability to automatically learn and improve from experience without being explicitly programmed
learns
Ordinary System
With Artificial Intelligence
Machine Learning
Predicts
Improves


Types In Machine Learning
1.Supervised Machine Learning
Labelled Data
Apple
Apple
Apple
Strawberry
Strawberry
Strawberry
Machine Learning Model
Output for future inputs
Apple
Strawberry

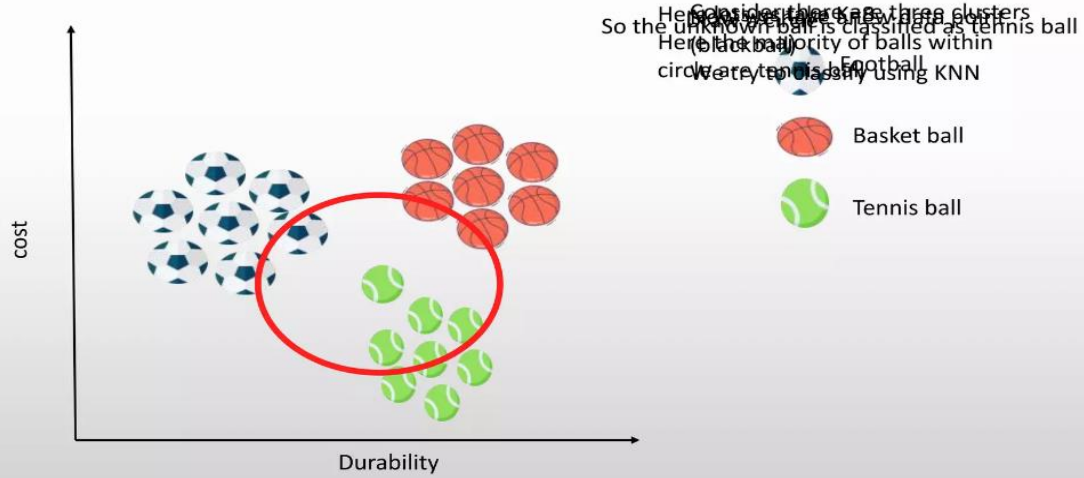

Types In Machine Learning
2.Unsupervised Machine Learning
Input Data
Model
Output

K-Nearest neighbours

- K Nearest Neighbour is a simple algorithm that stores all the available cases and classifies the new data or case based on a similarity measure. It is mostly used to classifies a data point based on how its neighbours are classified.
- In K Nearest neighbour K can be a integer greater than 1.So ,far every new data point we want to classify, we compute to which neighbouring group it is closest.



K-Nearest neighbours Example



Linear Regression

- Linear regression attempts to model the relationship between two variables by fitting a linear equation to observed data. One variable is considered to be an explanatory variable, and the other is considered to be a dependent variable. For example, a modeler might want to relate the weights of individuals to their heights using a linear regression model.

Histogram

Import libraries

I imported several libraries for the project: numpy: To work with arrays pandas: To work with csv files and dataframes matplotlib: To create charts us define parameters using rcParams and color them with cm.rainbow warnings: To ignore all warnings which might be showing up in the notebook du past/future depreciation of a feature train_test_split: To split the dataset into training and testing data StandardScaler: To scale all the features, so t Machine Learning model better adapts to the dataset Next, I imported all the necessary Machine Learn

In [1]:

```
# Basic
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import rcParams
from matplotlib.cm import rainbow
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')

# Other Libraries
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Machine Learning
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
```

Import dataset

After downloading the dataset from Kaggle, I saved it to my working directory with the name dataset.csv. Next, I used read_csv() to read the datase it to the dataset variable. Before any analysis, I just wanted to take a look at the data. So, I used the info() method.

In [4]:

```
df=pd.read_csv('heart.csv')
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
```

Data Processing

To work with categorical variables, we should break each categorical column into dummy columns with 1s and 0s. Let's say we have a column Gender, with values 1 for Male and 0 for Female. It needs to be converted into two columns with the value 1 where the column would be true and 0 where it will be false. Take a look at the Gist below.

To get this done, we use the get_dummi

To get this done, we use the get_dummies() method from pandas. Next, we need to scale the dataset for which we will use the StandardScaler. The fit_transform() method of the scaler scales the data and we update the columns.

In [16]:
```
df = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'])
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
df[columns_to_scale] = standardScaler.fit_transform(df[columns_to_scale])
```

The dataset is now ready. We can begin with training our models.

Machine Learning In this project, I took 4 algorithms and varied their various parameters and compared the final models. I split the dataset into 67% training data and 33% testing data.

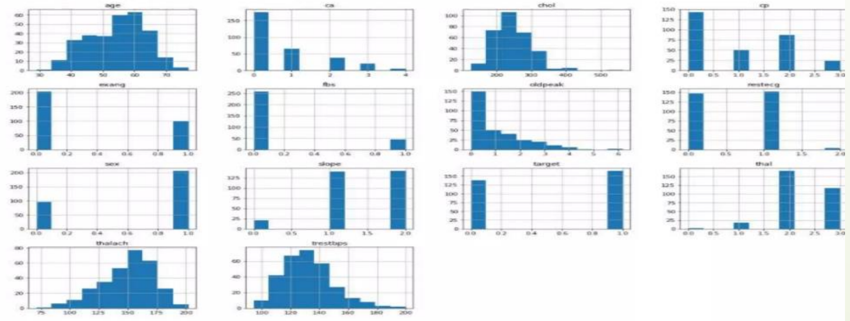
In [18]:
```
y = df['target']
X = df.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

Histogram

The best part about this type of plot is that it just takes a single command to draw the plots and it provides so much information in return. Just use dataset.hist().

In [12]:
```
df.hist()
```

Out[12]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001C76874DEC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C768F3A8C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C768F6EE48>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C768FA6BC8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001C768FDE948>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C7690156C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C76904E348>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C76908CF48>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001C769095048>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C7690CC208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C769130788>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C769169848>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001C7691A1988>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C7691DBA88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C769215888>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001C76924CD88>]],
      dtype=object)
```



Bar Plot for Target Class

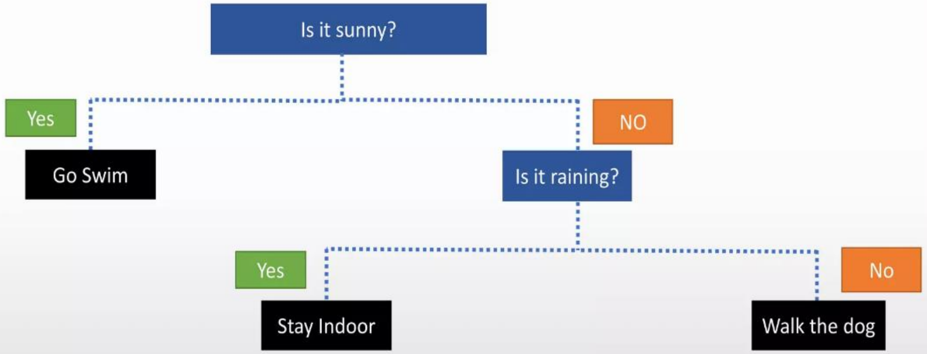
It's really essential that the dataset we are working on should be approximately balanced. An extremely imbalanced dataset can render the whole model training useless and thus, will be of no use.#

Let's say we have a dataset of 100 people with 99 non-patients and 1 patient. Without even training and learning anything, the model can always say that any new person would be a non-patient and have an accuracy of 99%. However, as we are more interested in identifying the 1 person who is a patient, we need balanced datasets so that our model actually learns.

In [14]:
```
rcParams['figure.figsize'] = 8,6
plt.bar(df['target'].unique(), df['target'].value_counts(), color = ['red', 'green'])
plt.xticks([0, 1])
plt.xlabel('Target Classes')
plt.ylabel('Count')
plt.title('Count of each Target Class')
```

Out[14]:
```
Text(0.5, 1.0, 'Count of each Target Class')
```

Decision Tree Example

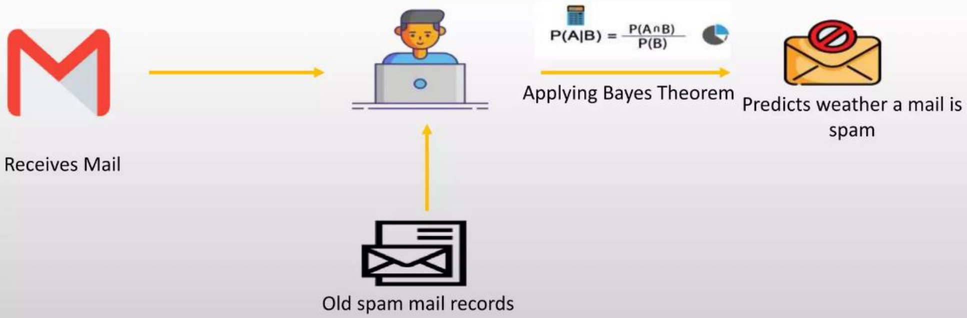


Decision Tree

Decision Tree : Decision tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

Naive Bayes

The Naive Bayes Classifier technique is based on conditional probability and is practically suited when the complexity of inputs is high



Machine Learning

$$x \rightarrow \boxed{f(x)} \rightarrow y$$

Machine Learning

$$2 \rightarrow \boxed{f(x)} \rightarrow 4$$

Machine Learning

$$2 \rightarrow \boxed{f(x) = 2x} \rightarrow 4$$

Machine Learning

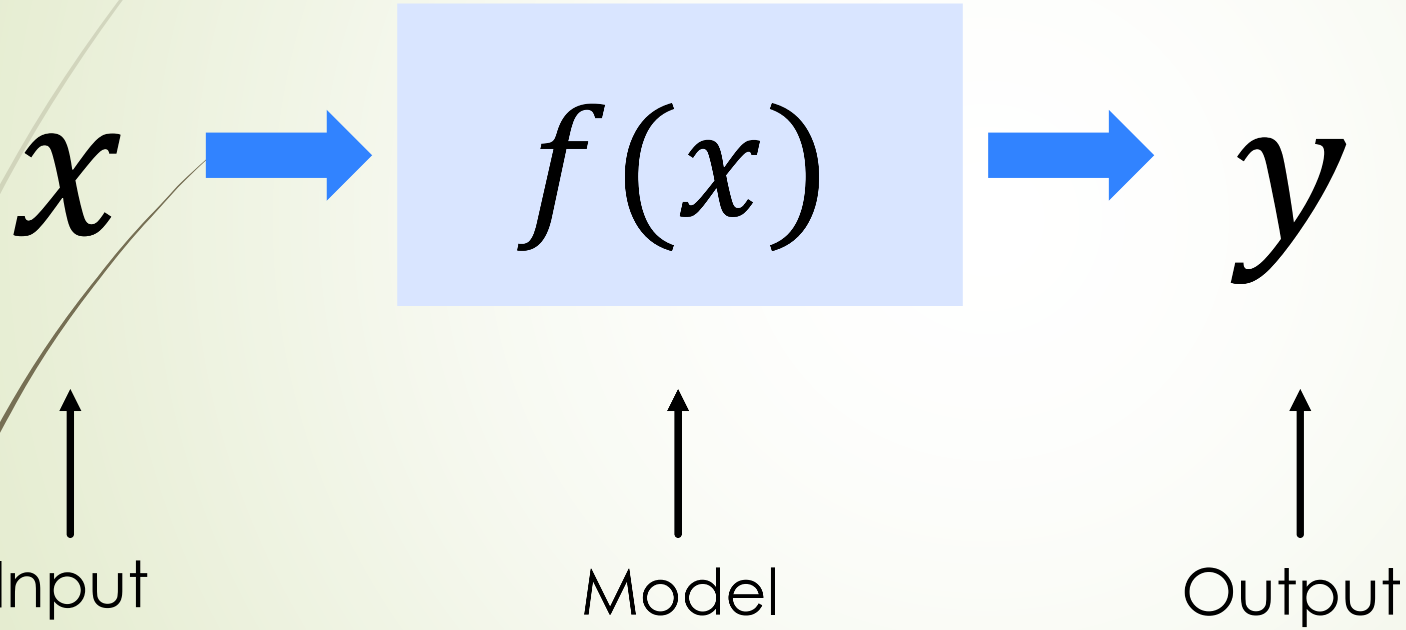
$$2 \rightarrow f(x) = 2x \rightarrow 4$$

Input

Model

Output

Machine Learning



Machine Learning

x → $f(x)$ → Walking

↑

Model?

*human activity recognition

Introduction into Machine Learning

Phone IMU Data → $f(x)$ → Walking

↑

Model?

*human activity recognition

Machine Learning

$$f(x)$$

dog

Model?

*object recognition

Machine Learning

$$f(x)$$

speaker

Model?

*object recognition

Machine Learning

Phone
IMU Data

$f(x)$

30 years

Model?

Gil Levi, and Tal Hassner. "Age and gender classification using convolutional neural networks." In Proceedings of the CVPR workshops. 2015. IEEE: https://doi.org/10.1109/CVPRW.2015.7301352

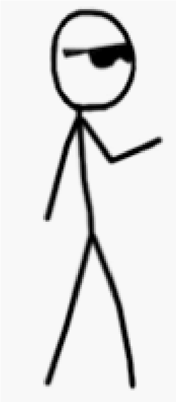
Machine Learning

Phone
IMU Data

$$f(x)$$

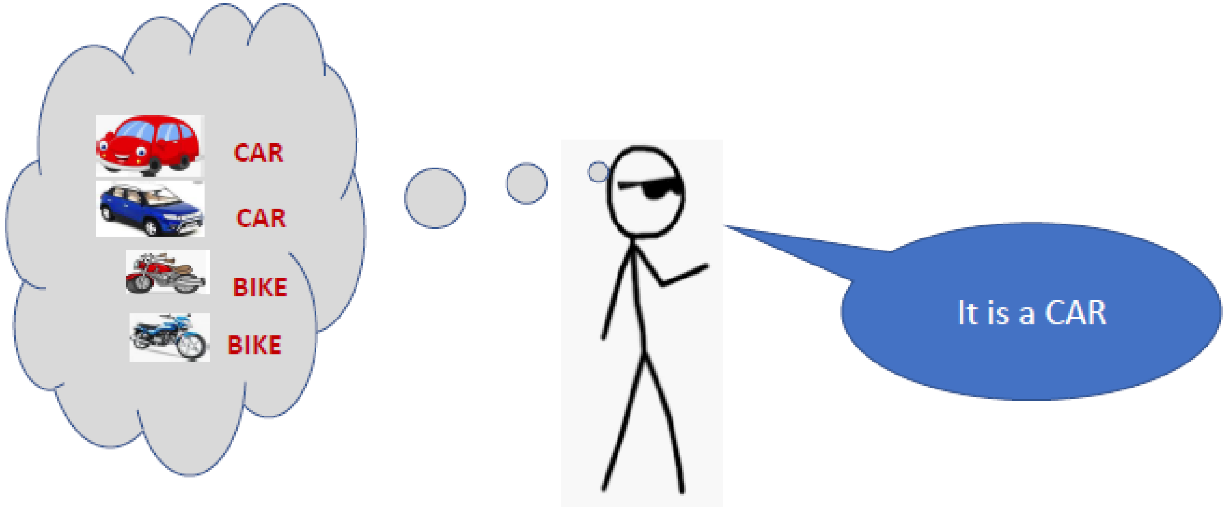
Model?

Ira Kemelmacher-Shlizerman, Supasorn Suwajanakorn, and Steven M. Seitz. 2014. Illumination-aware age progression. In Proceedings of the CVPR 2024. IEEE. DOI: https://doi.org/10.1109/CVPR.2014.426

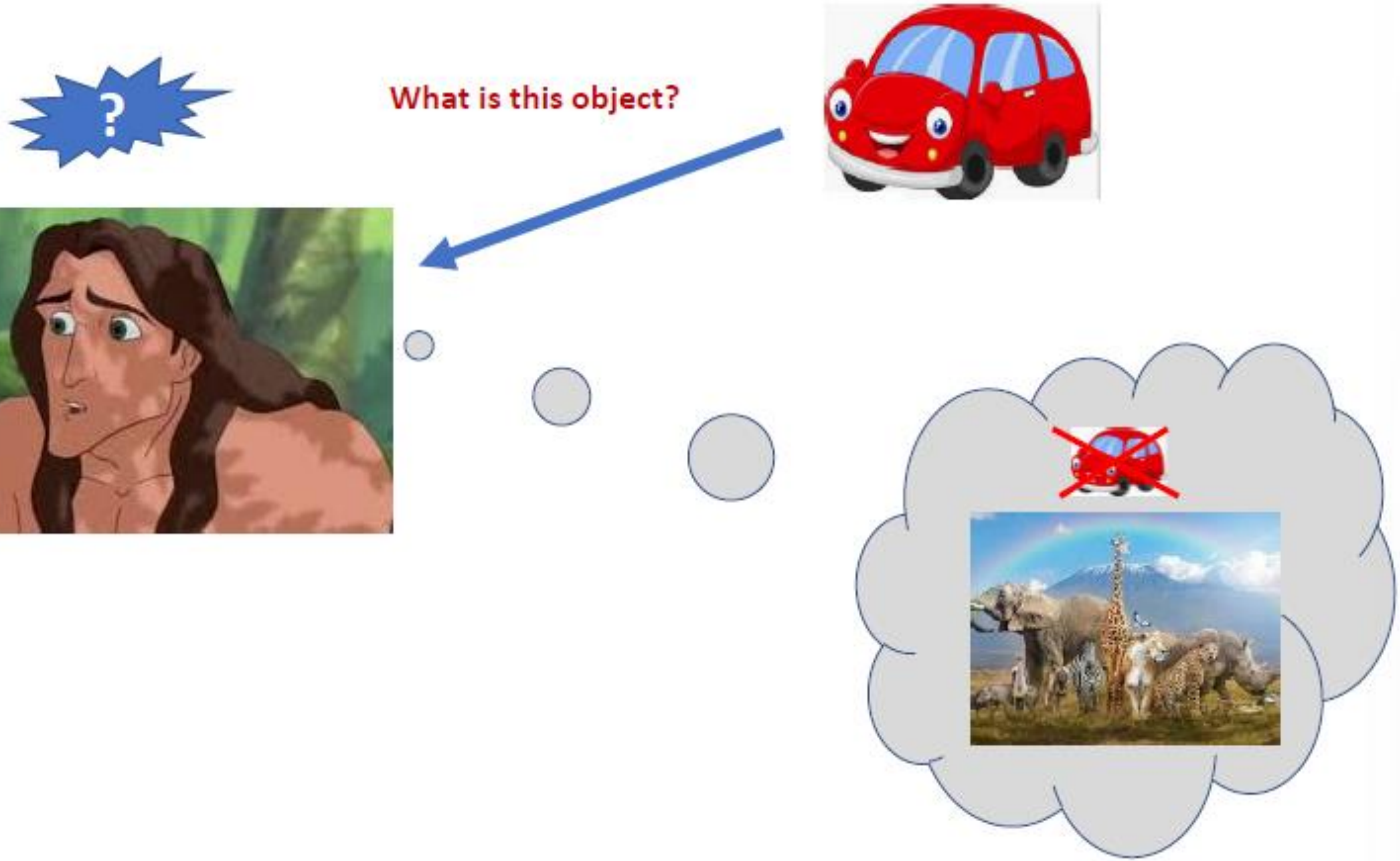


Human can learn from past experience and make decision of its own

What is this object?



Let us ask the same question to him



[But, he is a human being. He can observe and learn]

Let us make him learn

Let us ask the same question now

What is this object?


CAR

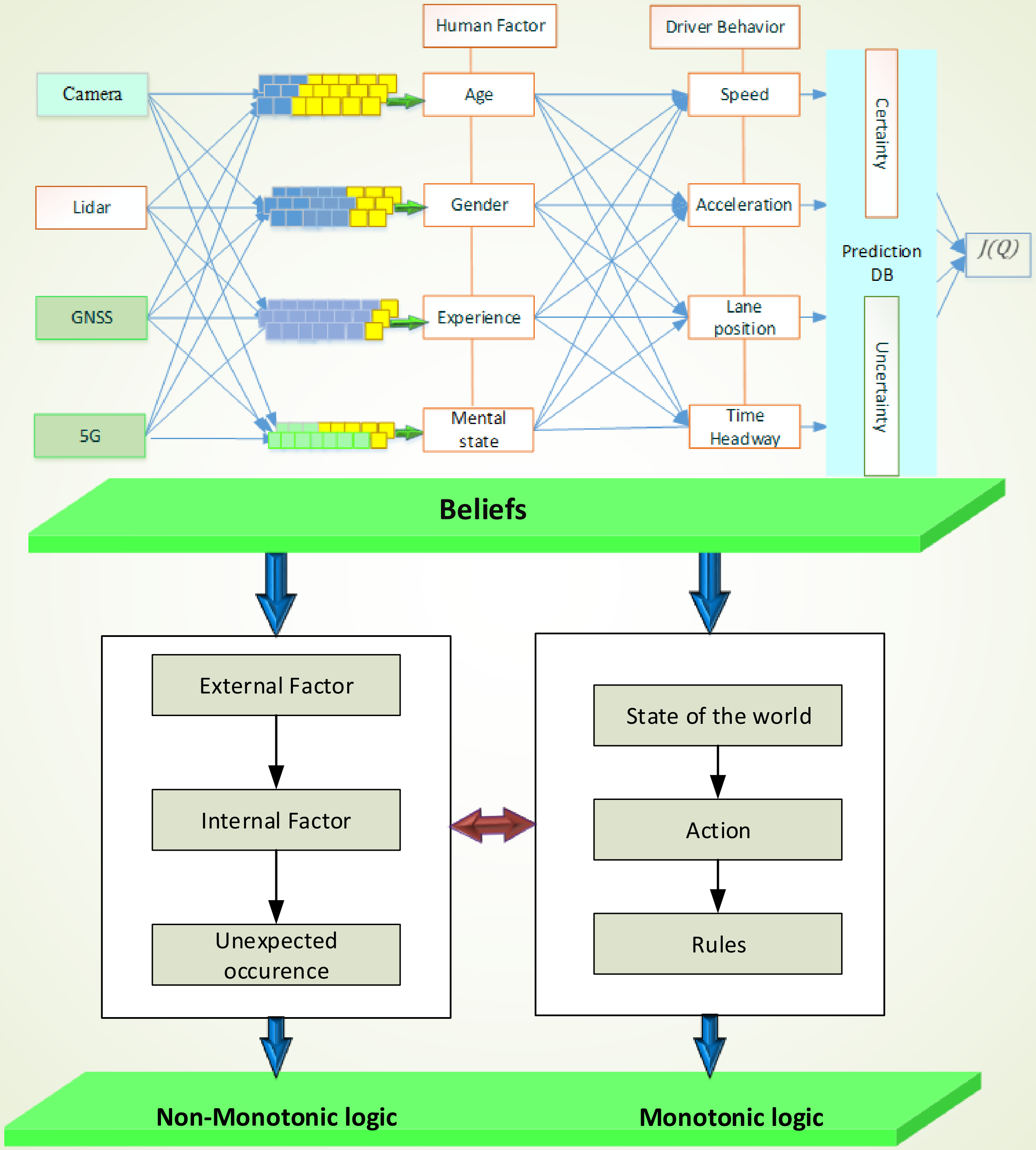

CAR


BIKE

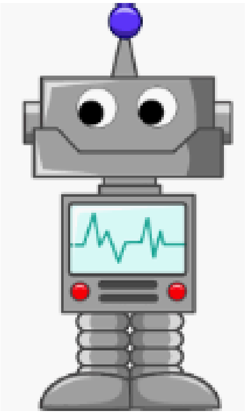

BIKE

Past experience

Let us ask the same question now


Human Factor
Driver Behavior
Camera
Lidar
GNSS
5G
Age
Gender
Experience
Mental state
Speed
Acceleration
Lane position
Time Headway
Certainty
Prediction DB
Uncertainty
J(Q)
Beliefs
External Factor
Internal Factor
Unexpected occurence
State of the world
Action
Rules
Non-Monotonic logic
Monotonic logic

What about a Machine ?



Machines follow instructions

[It can not take decision of its own]

We can ask a machine

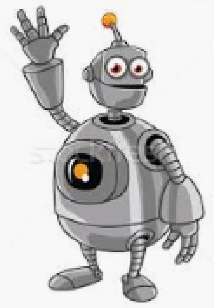
- To perform an arithmetic operations such as
  - Addition
  - Multiplication
  - Division

- Comparison
- Print
- Plotting a chart

[We want a machine to act like a human]

Price in 2025?

[predict the price in future]

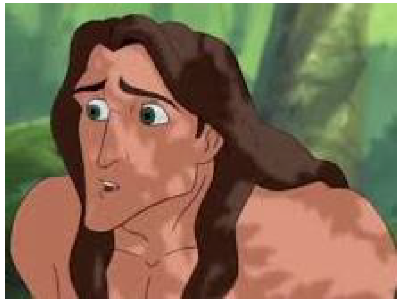


I ~~made~~ met him yesterday

[Natural Language understand, and correct grammar]

recognize face

[Recognize Faces]

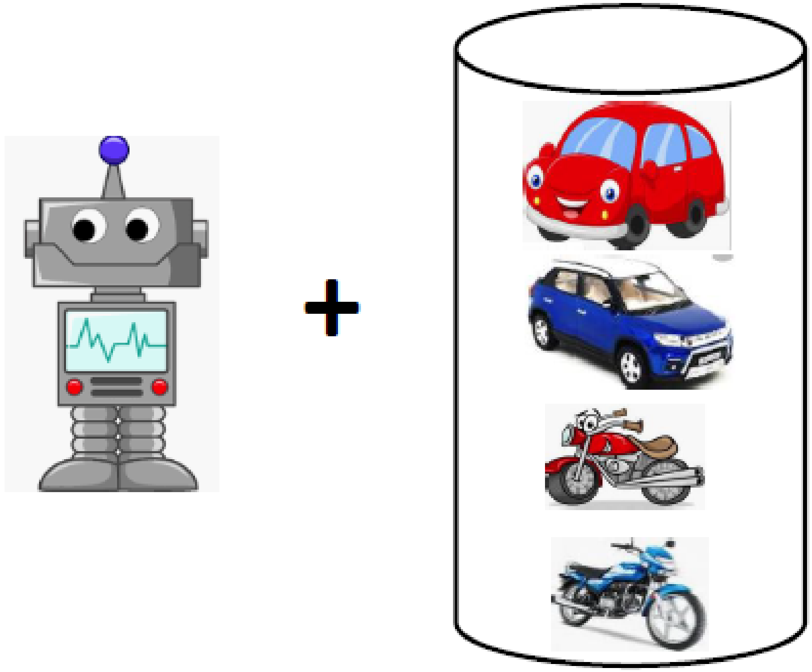


[What do we do?

Just like, what we did to human,

we need to provide experience to the machine.

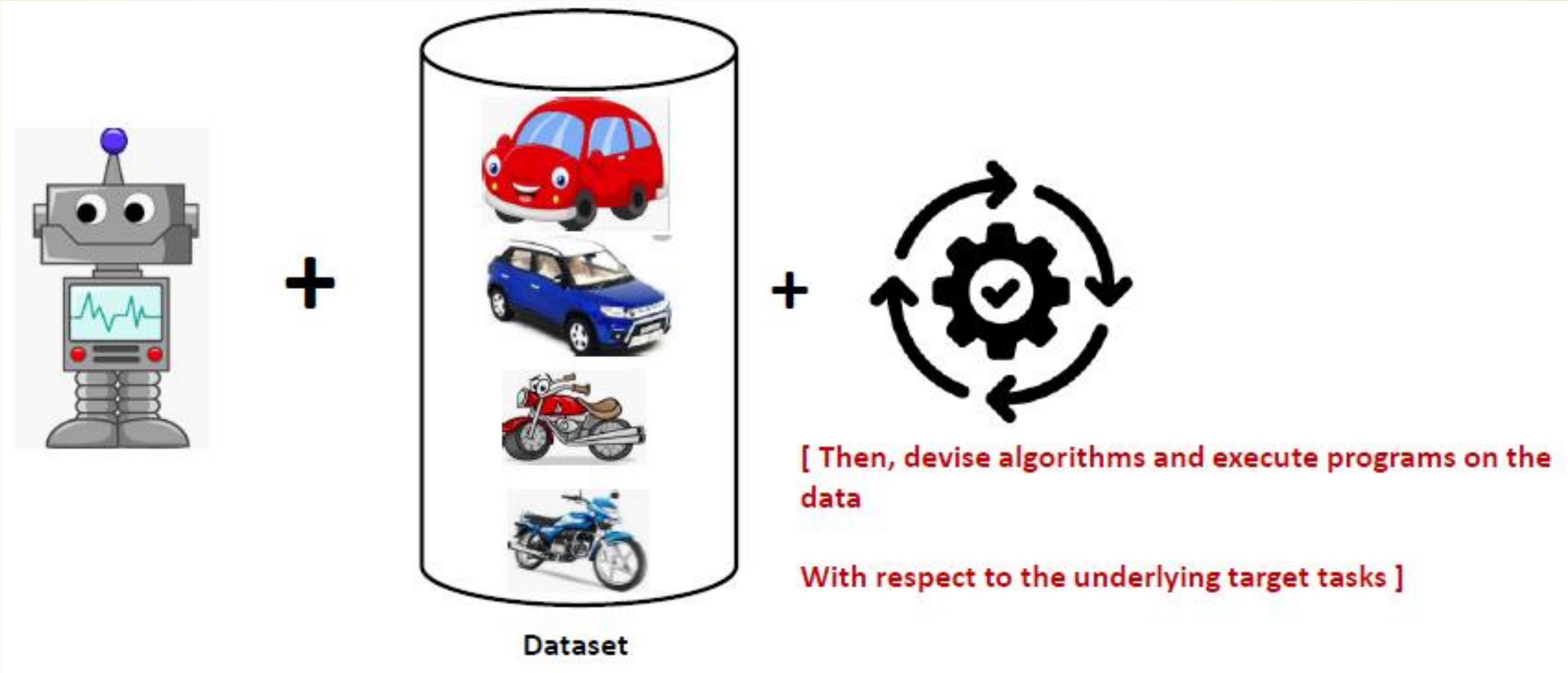
]



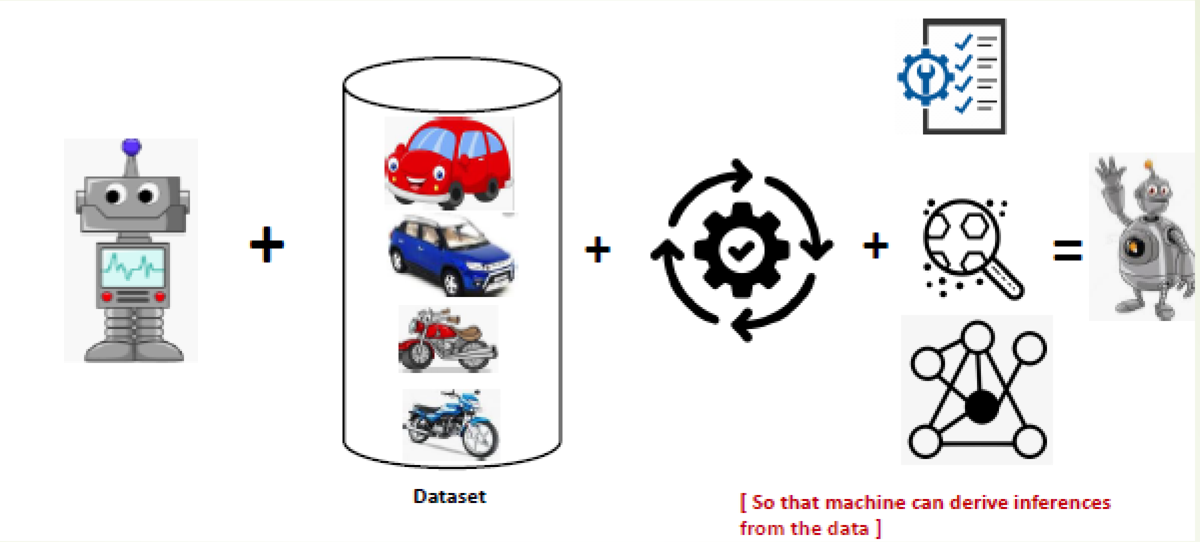
Dataset

[
This what we called as Data or Training dataset

So, we first need to provide training dataset to the machine
]

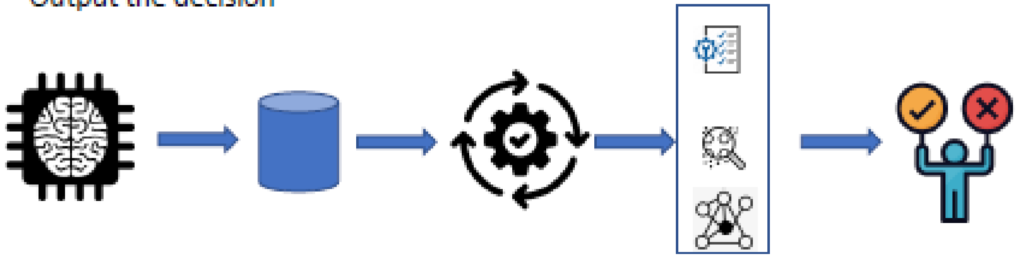


+
Dataset
+
[Then, devise algorithms and execute programs on the data
With respect to the underlying target tasks]

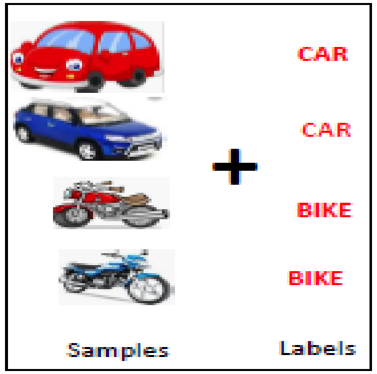


Given a machine learning problem

- Identify and create the appropriate dataset
- Perform computation to learn
  - Required rules, pattern and relations
- Output the decision

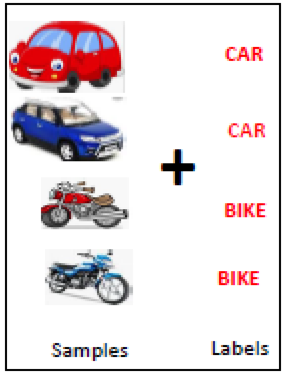


What is Supervised Learning?



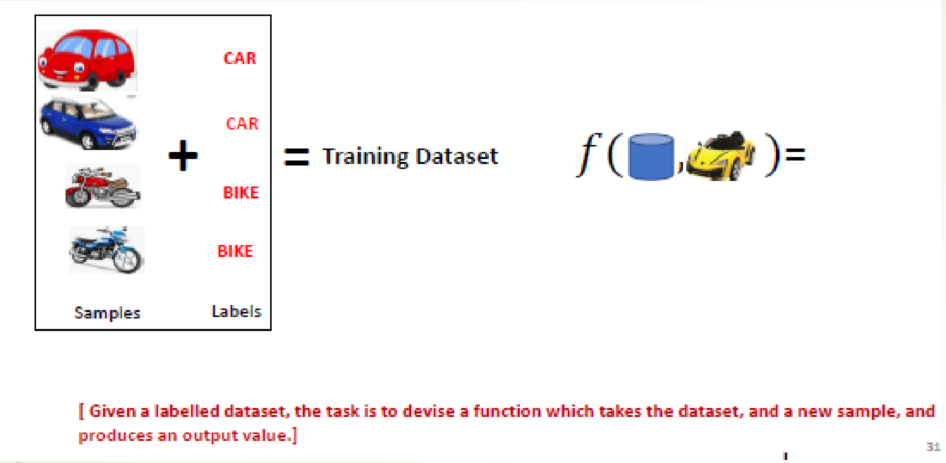
+ = Training Dataset

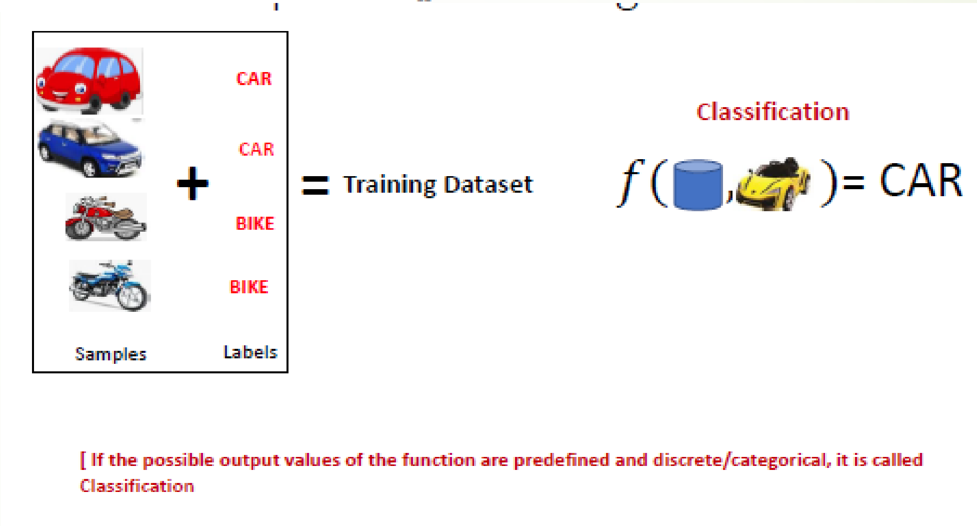
[In supervised learning, we need some thing called a Labelled Training Dataset]

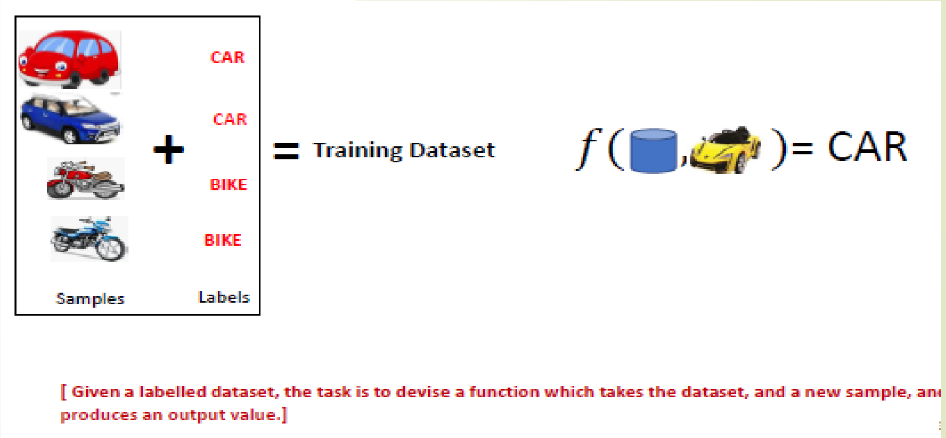


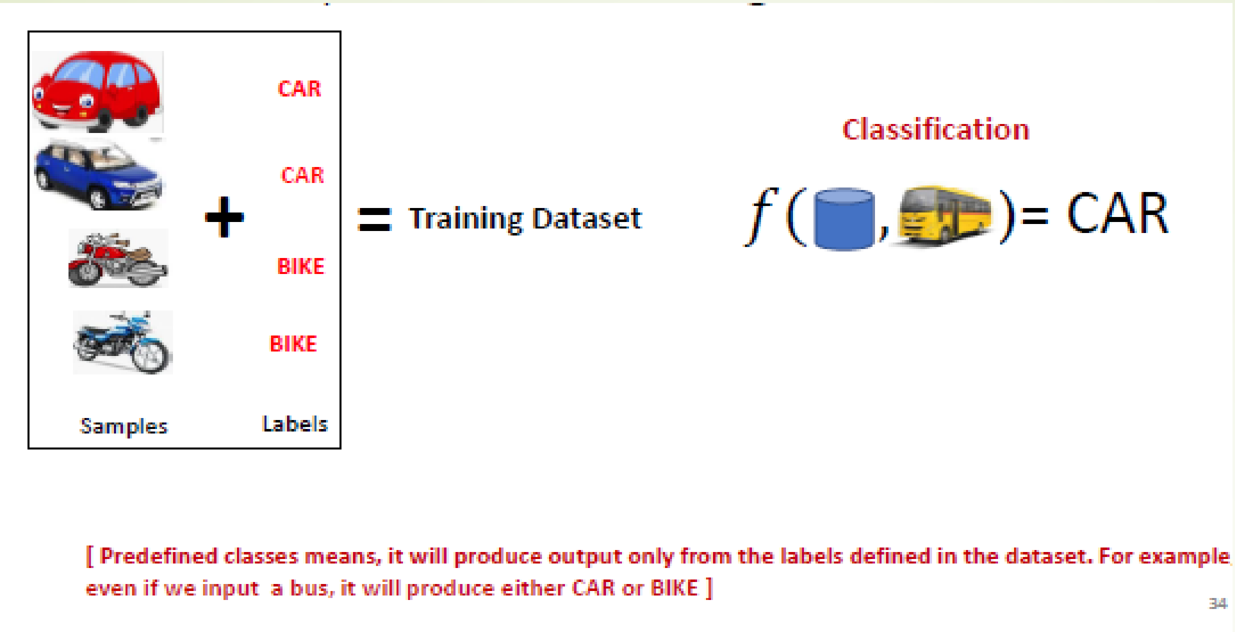
+ = Training Dataset $\quad f(\;,\quad)=$

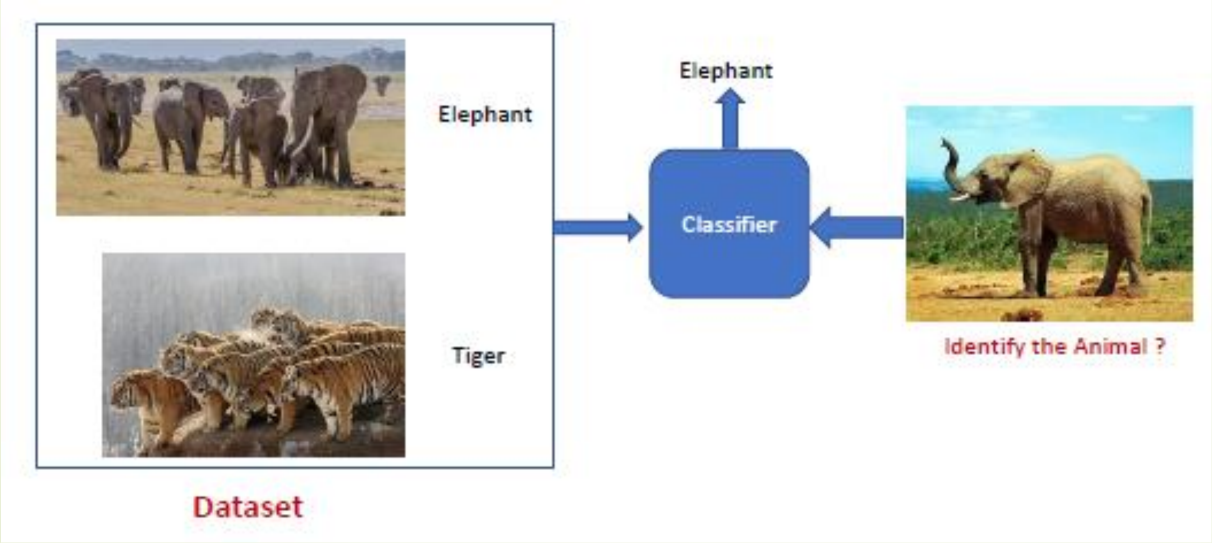
[Given a labelled dataset, the task is to devise a function which takes the dataset, and a new sample, and produces an output value.]

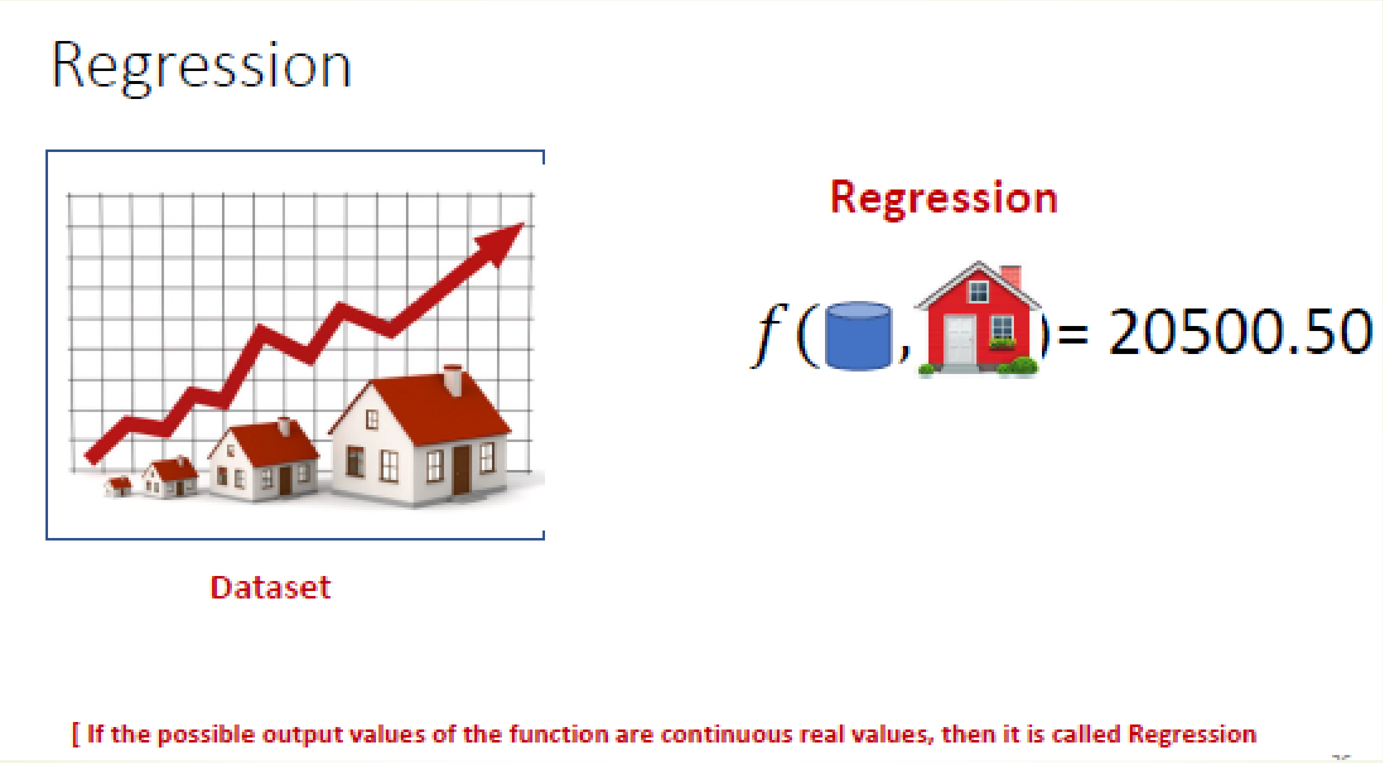

CAR
CAR
BIKE
BIKE
Samples
Labels
+
= Training Dataset
f (,)=
[Given a labelled dataset, the task is to devise a function which takes the dataset, and a new sample, and produces an output value.]


CAR
CAR
BIKE
BIKE
Samples
Labels
+
= Training Dataset
Classification
f (,)= CAR
[If the possible output values of the function are predefined and discrete/categorical, it is called Classification


CAR
CAR
BIKE
BIKE
Samples
Labels
+
= Training Dataset
f (,)= CAR
[Given a labelled dataset, the task is to devise a function which takes the dataset, and a new sample, and produces an output value.]

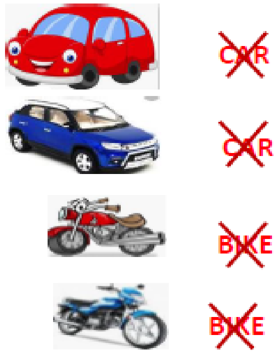

CAR
CAR
BIKE
BIKE
Samples
Labels
+
= Training Dataset
Classification
f(,)= CAR
[Predefined classes means, it will produce output only from the labels defined in the dataset. For example even if we input a bus, it will produce either CAR or BIKE]
34


Elephant
Tiger
Dataset
Elephant
Classifier
Identify the Animal ?



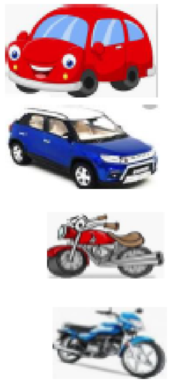
The classification and Regression problems are supervised, because the decision depends on the characteristics of the ground truth labels or values present in the dataset, which we define as experience

What is Unsupervised Learning



Dataset

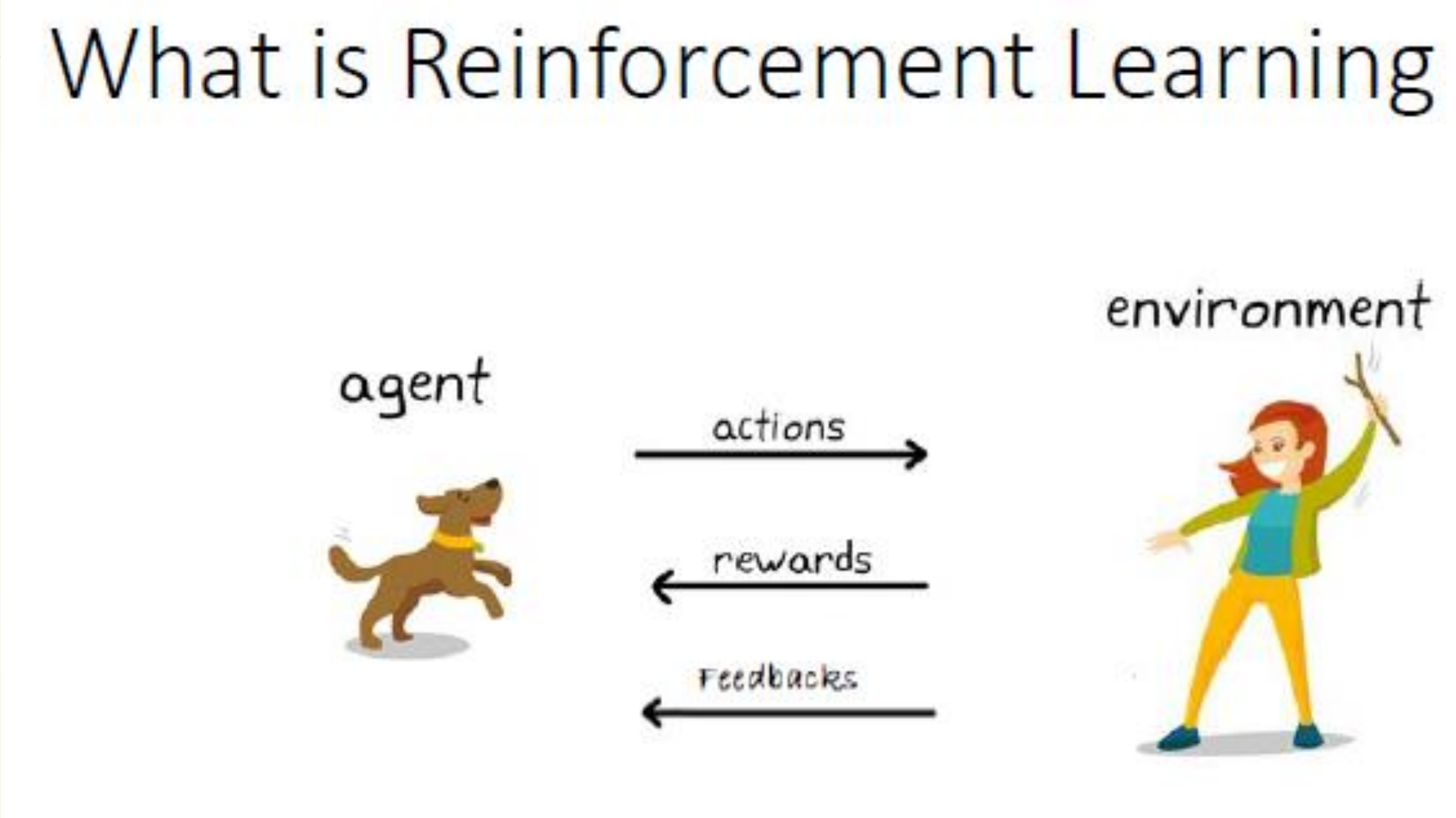
[In the unsupervised learning, we do not need to know the labels or Ground truth values]



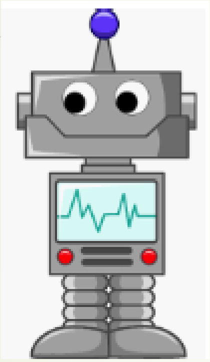
Dataset

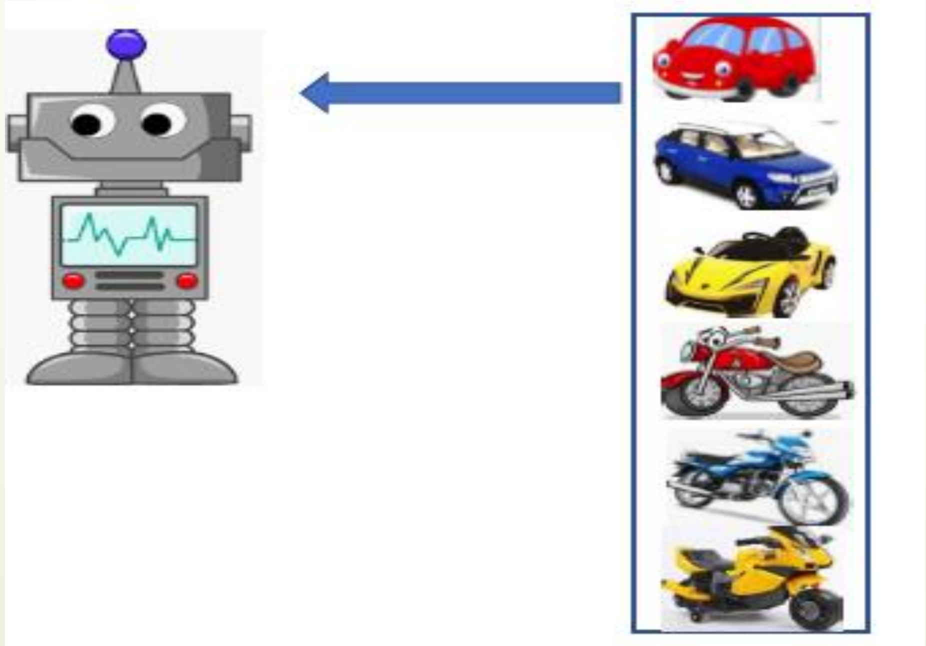
Clustering

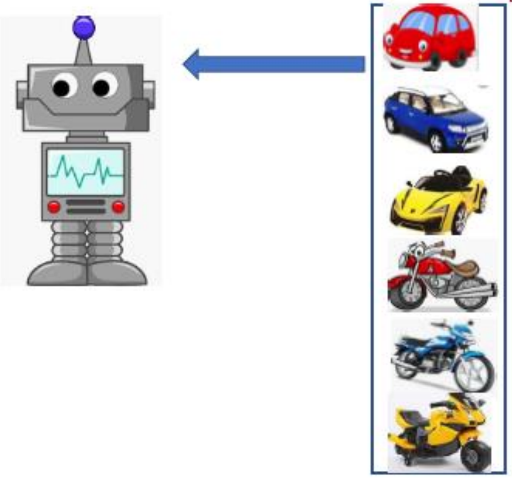
[The task is to identify the patterns like group the similar objects together]


What is Reinforcement Learning
environment
agent
actions
rewards
Feedbacks

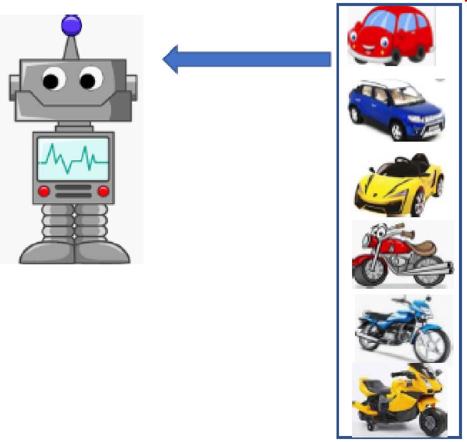
Teach a machine to identify vehicle types







#Wheel Height Weight Color

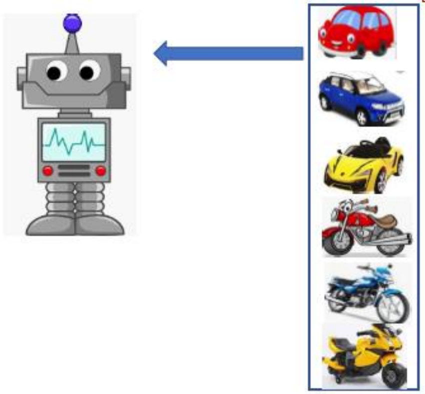


#Wheel Height Weight Color

Identify the features which can represent the objects

$$F = \{f_1\, f_2\, f_3 \dots f_k\}$$

Feature set={ #Wheel Height Weight Color }



#Wheel Height Weight Color

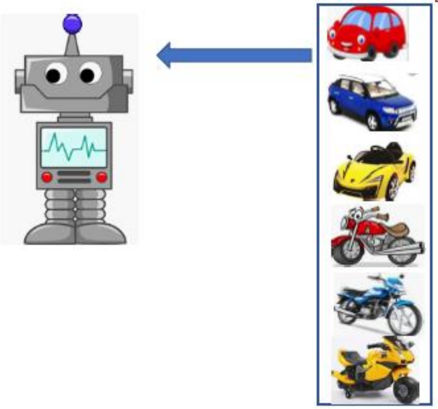
Identify the features which can represent the objects

$$F = \{f_1\, f_2\, f_3 \dots f_k\}$$

For every sample, assign value to corresponding feature

$$v_i = \{w_{i1} w_{i2} w_{i3} \dots w_{ik}\}$$

where w_{ij} is the value assigned for the feature f_j

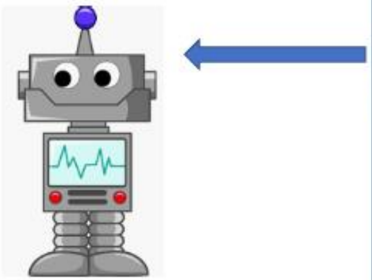


#Wheel	Height	Weight	Color
4	6	500	Red
4	5.5	600	Blue
4	5	550	Yellow
2	3	200	Red
2	3.5	150	blue
2	4	250	Yellow

For every object, assign value to corresponding feature

$$v_i = \{w_{i1} w_{i2} w_{i3} \dots w_{ik}\}$$

where w_{ij} is the value assigned for the feature f_j

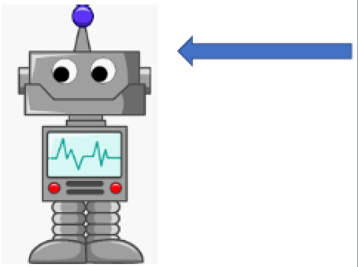
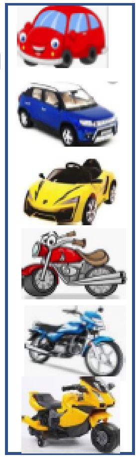



#Wheel	Height	Weight	Color
4	6	500	Red
4	5.5	600	Blue
4	5	550	Yellow
2	3	200	Red
2	3.5	150	blue
2	4	250	Yellow

Features Vectors

This form of representation is called Vector Space Model

Are all features useful?

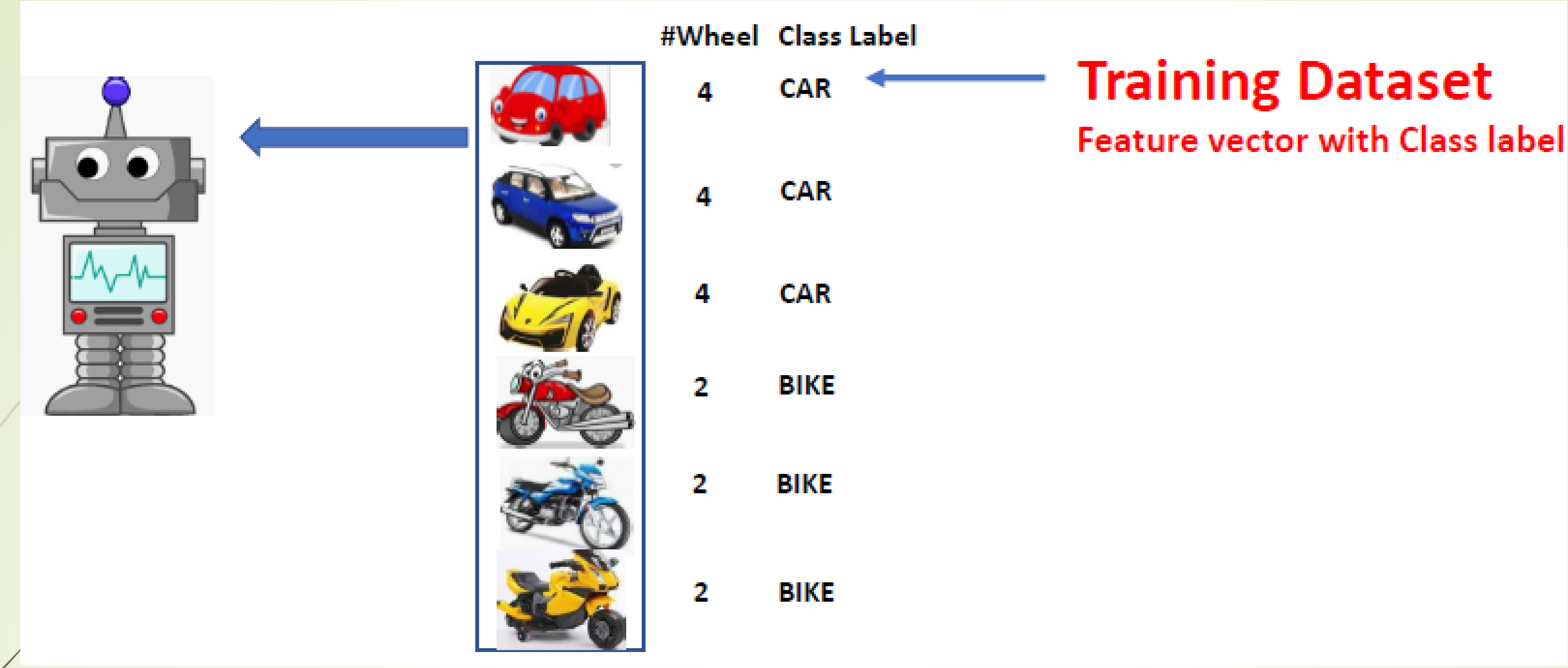
Features

#Wheel	Height	Weight	Color
4	6	500	Red
4	5.5	600	Blue
4	5	550	Yellow
2	3	200	Red
2	3.5	150	blue
2	4	250	Yellow

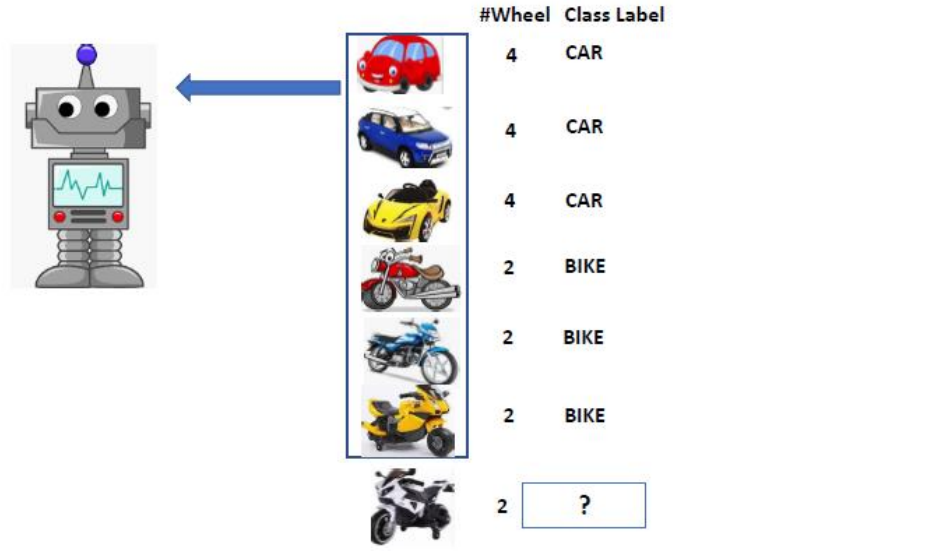
Features Vectors

Good Features
- #Wheel
- Height
- Weight

Bad Feature
- Colour



Given the #Wheel, identify the vehicle

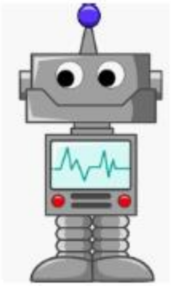


Let us estimate

#Wheel	Class Label
4	CAR
4	CAR
4	CAR
2	BIKE
2	BIKE
2	BIKE

Pr(Vehicle type| #Wheel) = ?

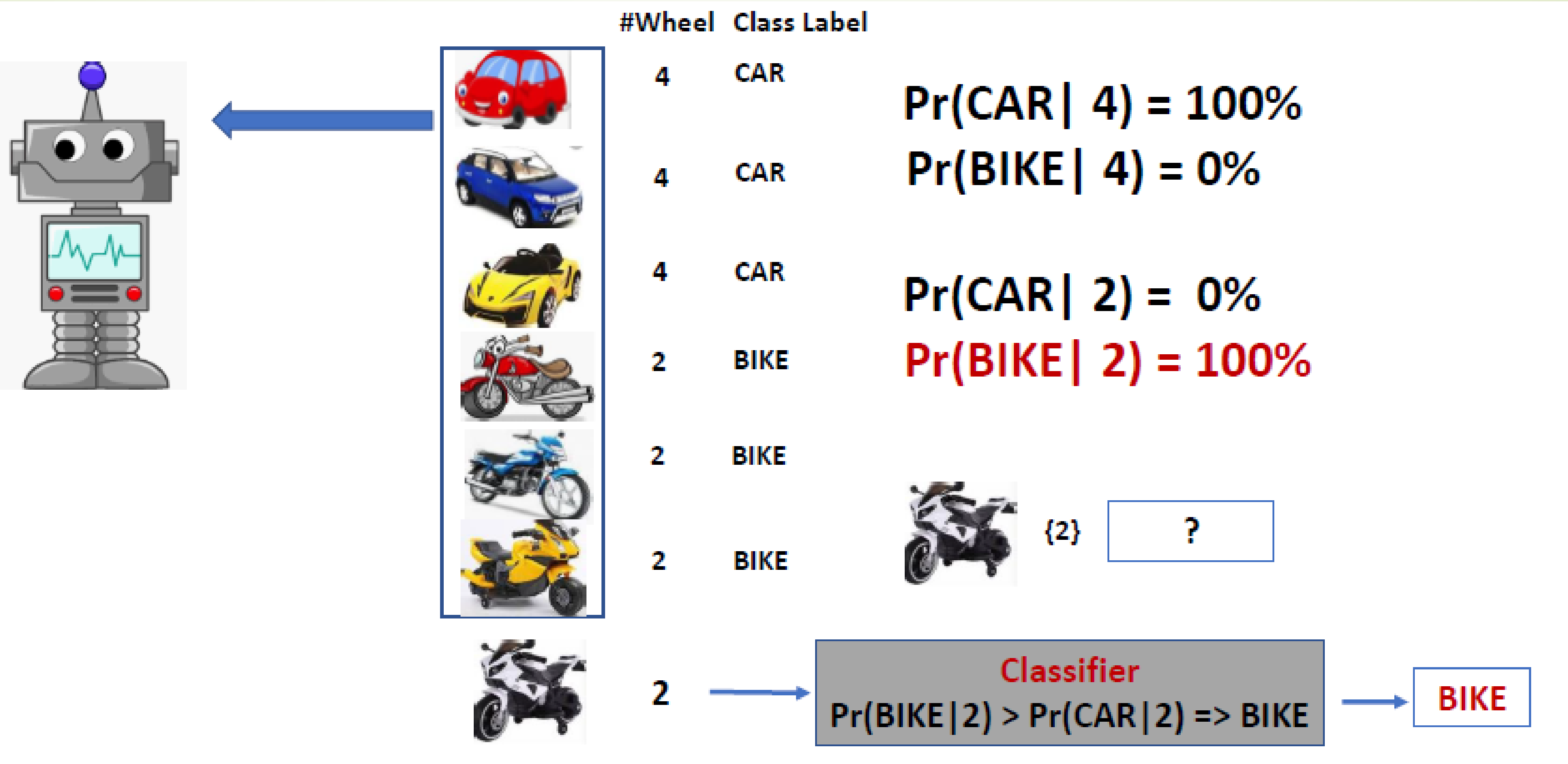
Let us estimate the probability (type|#wheel)

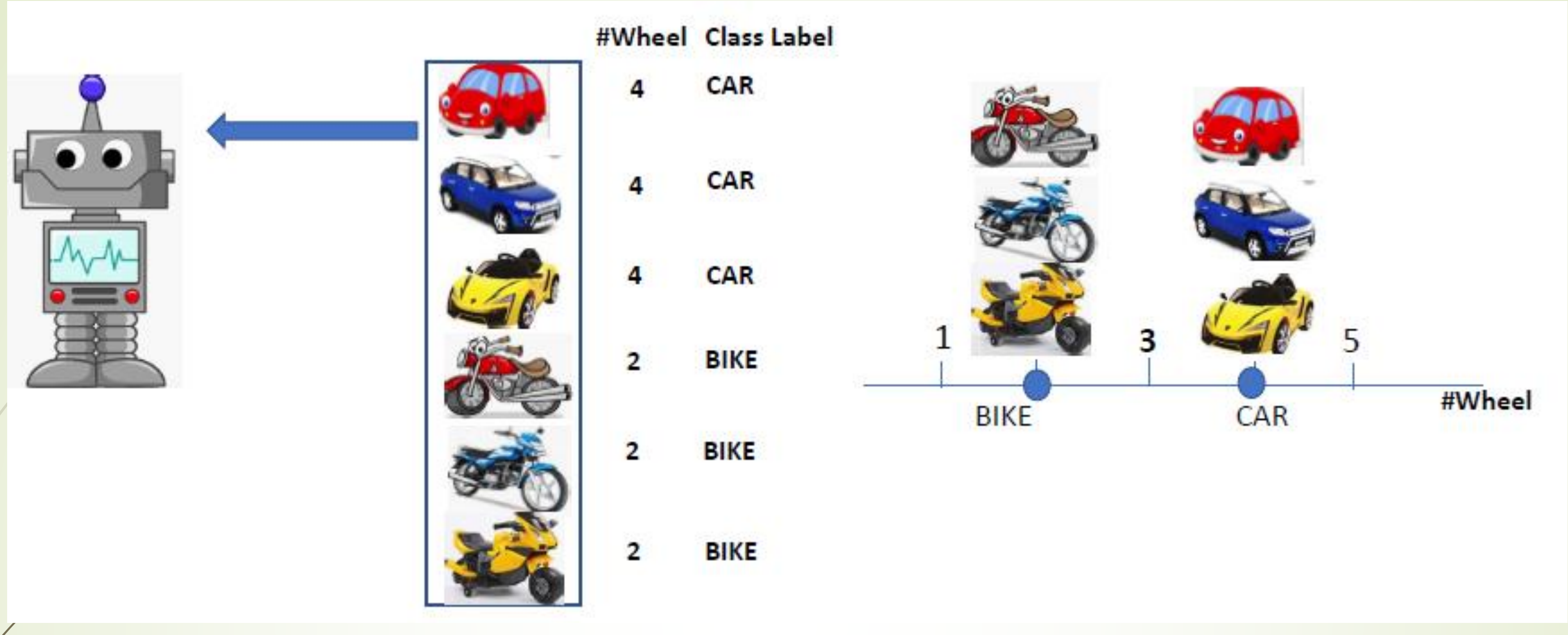


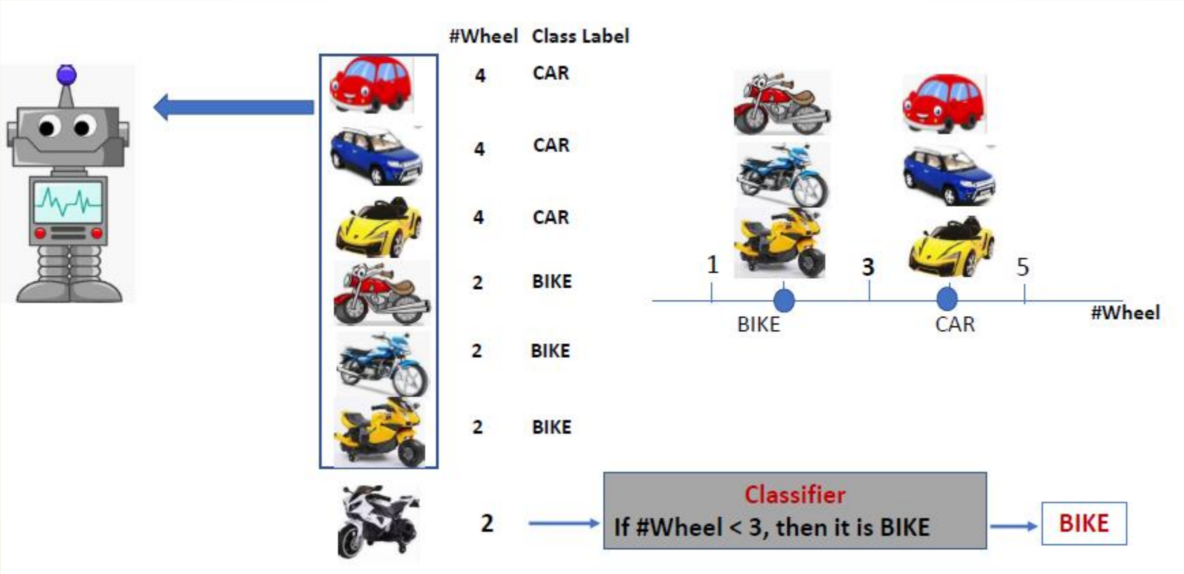
#Wheel	Class Label
4	CAR
4	CAR
4	CAR
2	BIKE
2	BIKE
2	BIKE

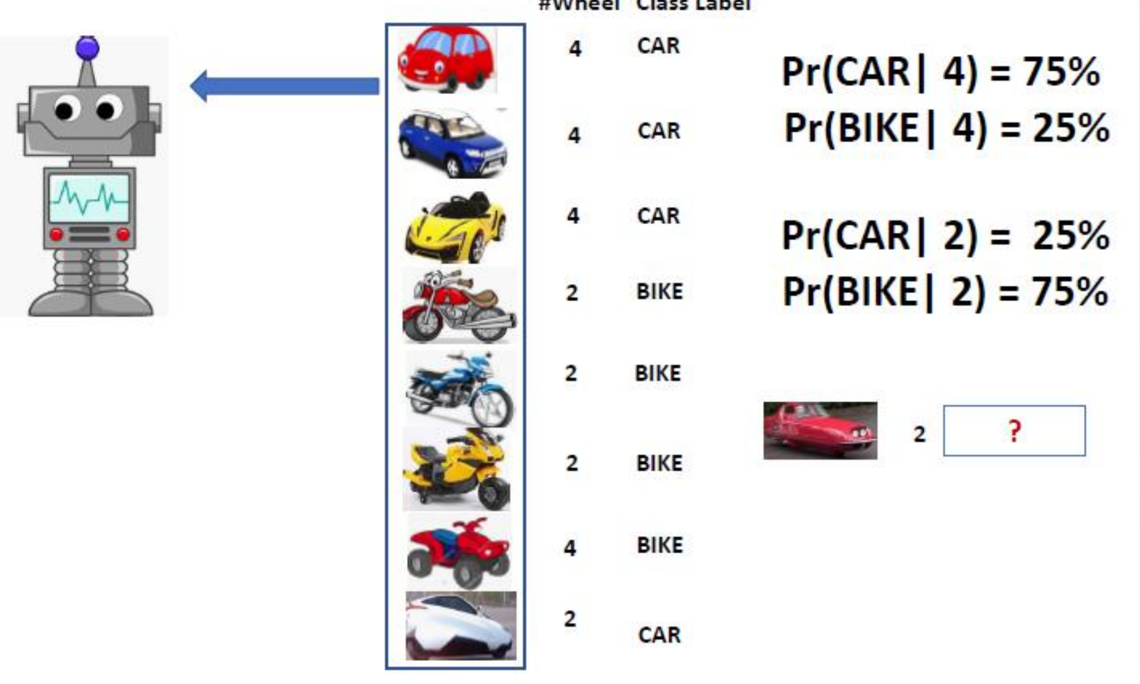
Pr(CAR| 4) = 100%
Pr(BIKE| 4) = 0%

Pr(CAR| 2) = 0%
Pr(BIKE| 2) = 100%

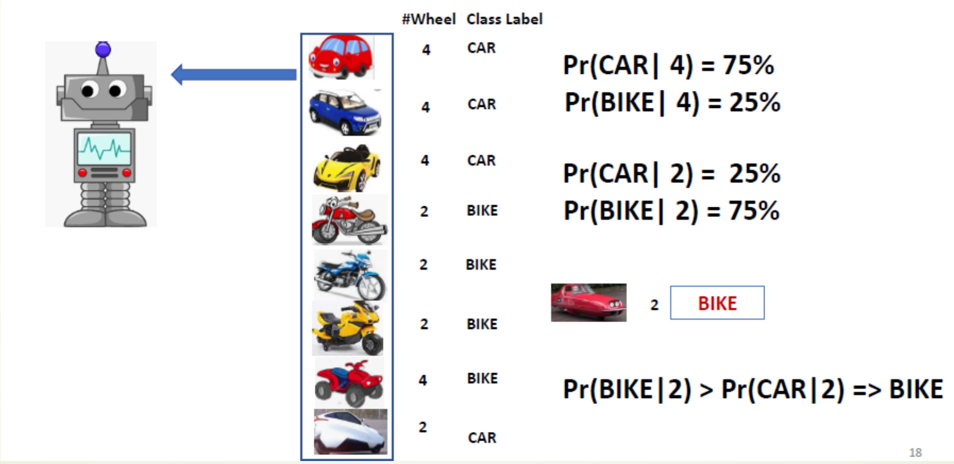

#Wheel Class Label
4 CAR
4 CAR
4 CAR
2 BIKE
2 BIKE
2 BIKE
Pr(CAR| 4) = 100%
Pr(BIKE| 4) = 0%
Pr(CAR| 2) = 0%
Pr(BIKE| 2) = 100%
{2}
?
2
Classifier
Pr(BIKE|2) > Pr(CAR|2) => BIKE
BIKE


#Wheel Class Label
4 CAR
4 CAR
4 CAR
2 BIKE
2 BIKE
2 BIKE
1
3
5
BIKE
CAR
#Wheel

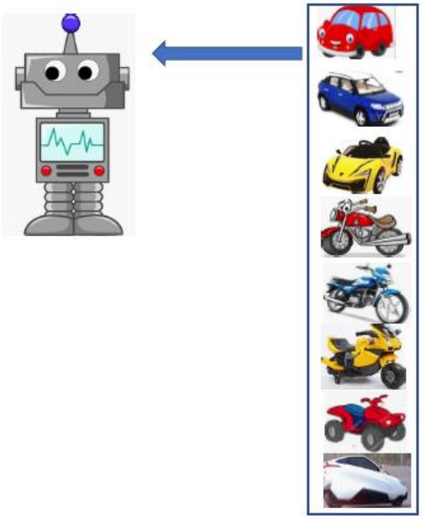

#Wheel Class Label
4 CAR
4 CAR
4 CAR
2 BIKE
2 BIKE
2 BIKE
1
3
5
BIKE
CAR
#Wheel
2
Classifier
If #Wheel < 3, then it is BIKE
BIKE



If selected feature is not sufficient



More Features

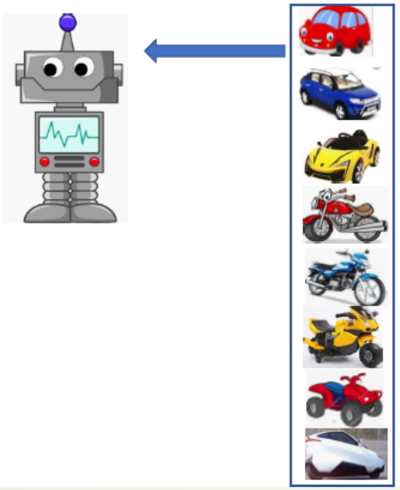


#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

H: High, height >= 5

L: Low, height < 5

Estimate the probabilities, and ask the same question



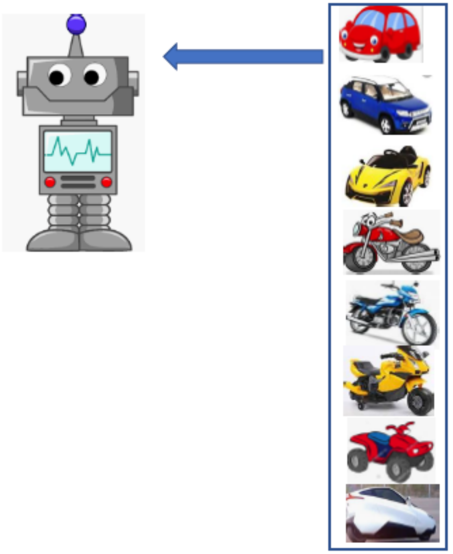
#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Pr(CAR| 4,H) = 100%
Pr(BIKE| 4,L) = 100%
Pr(CAR| 2,H) = 100%
Pr(BIKE| 2,L) = 100%
Pr(CAR| 4,L) = 0%
Pr(BIKE|4,H) = 0%
Pr(CAR| 2,L) = 0%
Pr(BIKE| 2,H) = 0%

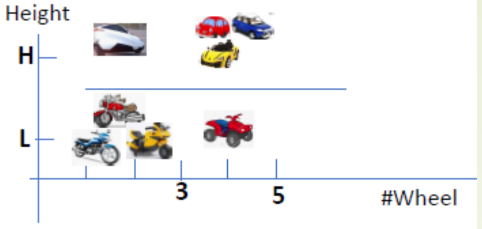
{2 H} ?

20

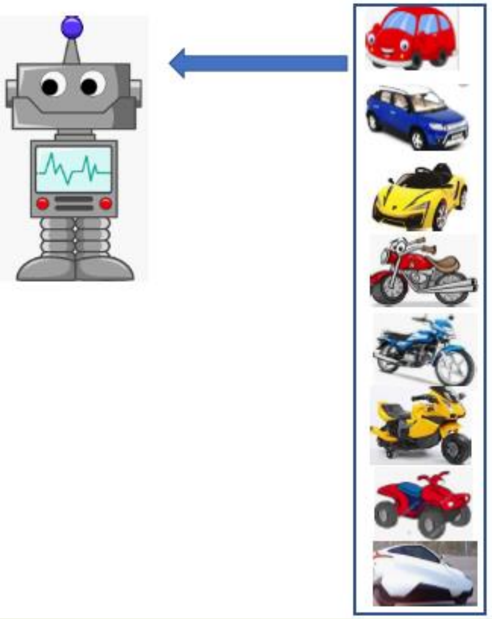
Multiple ways



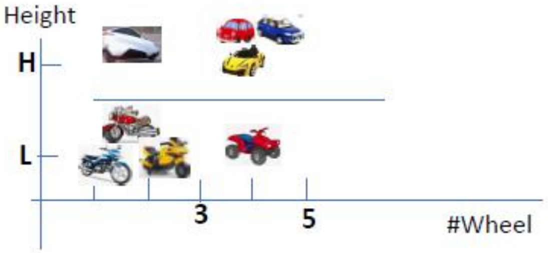
#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

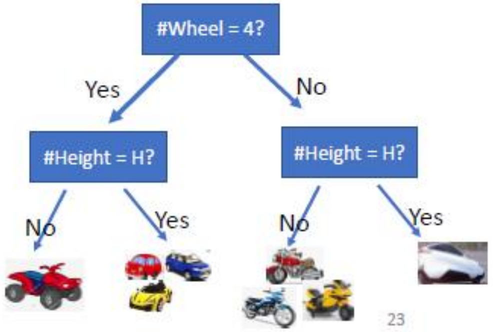


22



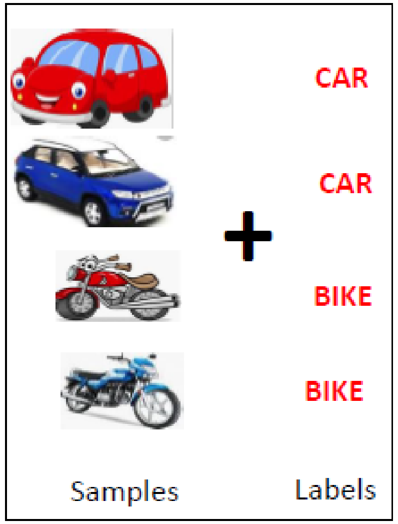
#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR



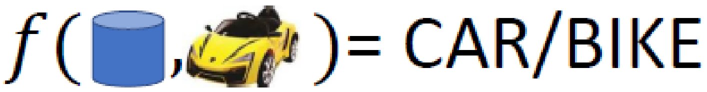


23

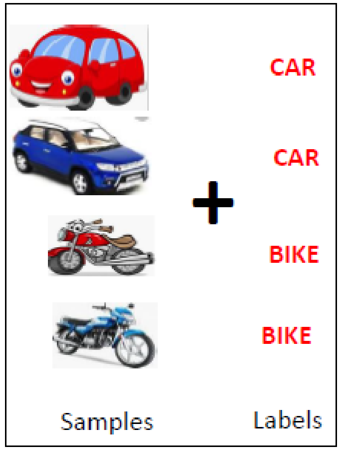
What is Classification?



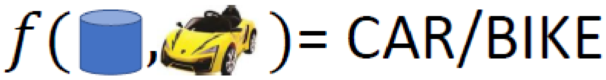
Training Dataset



Classification



Training Dataset

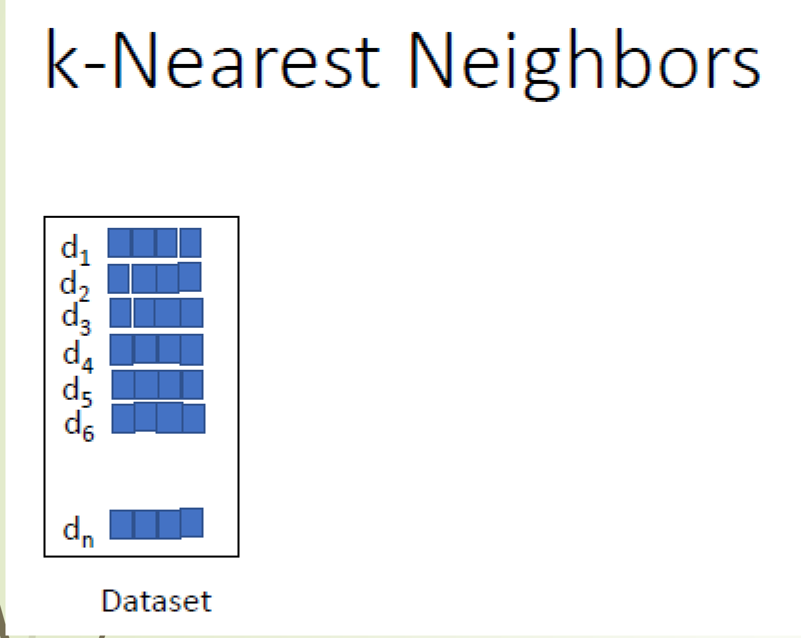


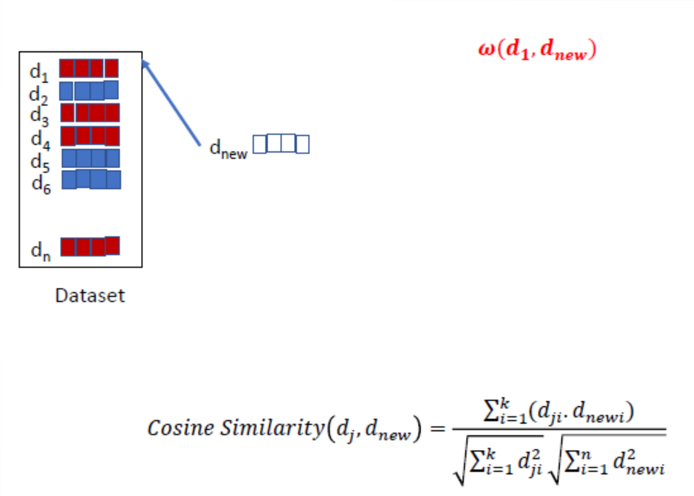
Given a dataset $\boldsymbol{D = \{ x_1, x_2\, x_3 \ldots x_n \}}$ and set of class labels $\boldsymbol{C = \{ c_1\, c_2\, c_3 \ldots c_k \}}$, the task of classification to devise a mapping function $\boldsymbol{f : D \rightarrow C}$.

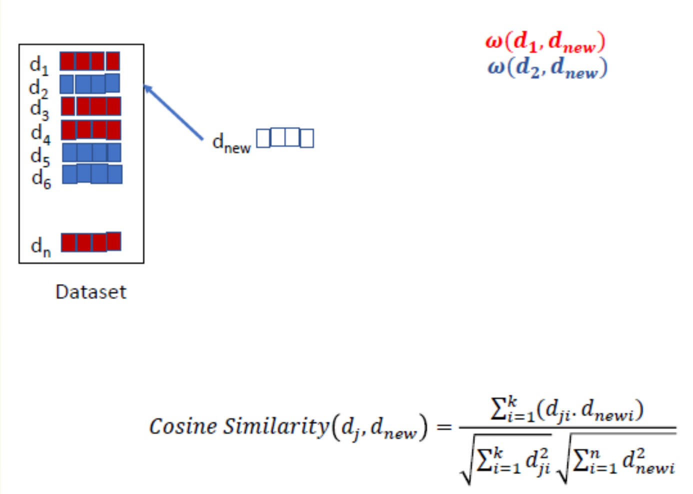
Classification

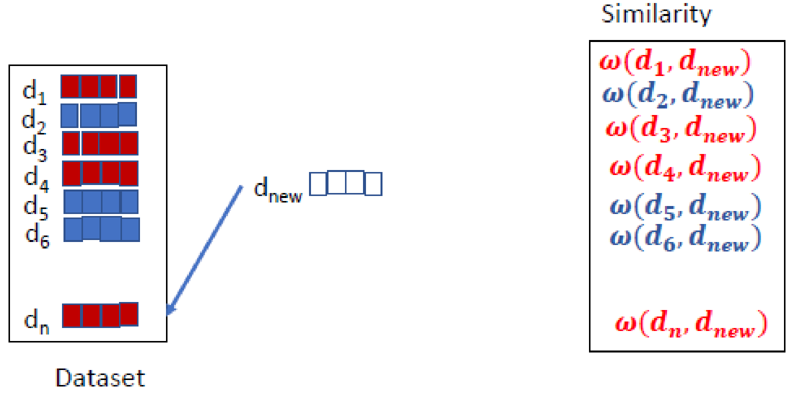
- Bayesian Classifier
- K-Nearest Neighbours
- Decision Tree
- Support Vector Machine
- Neural Network

k-Nearest Neighbors& Centroid Based Classifier

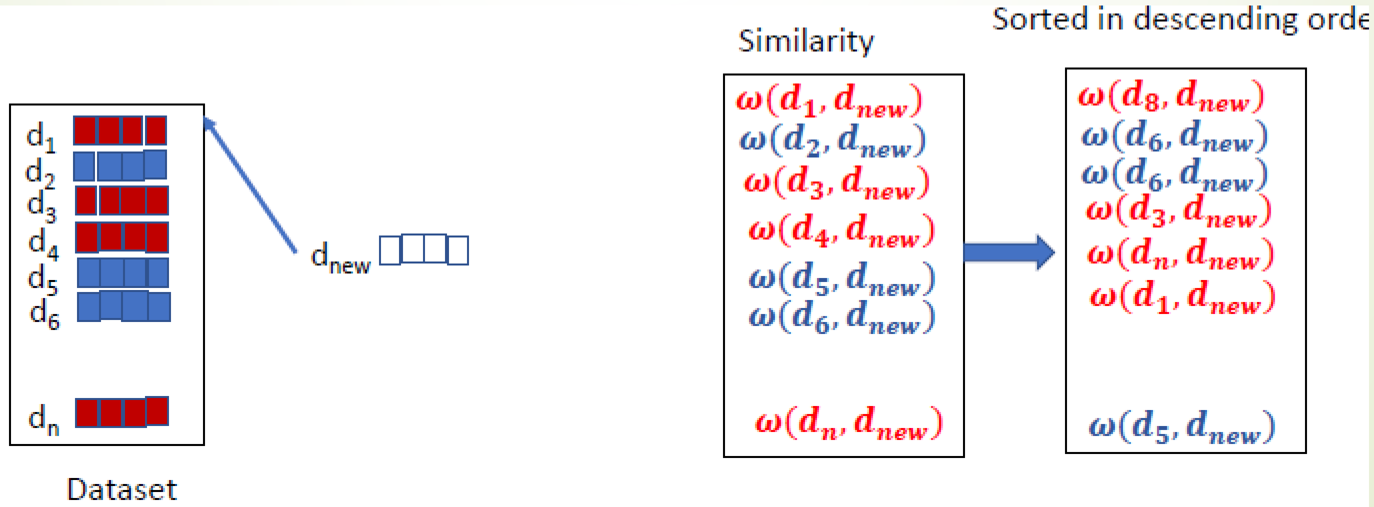


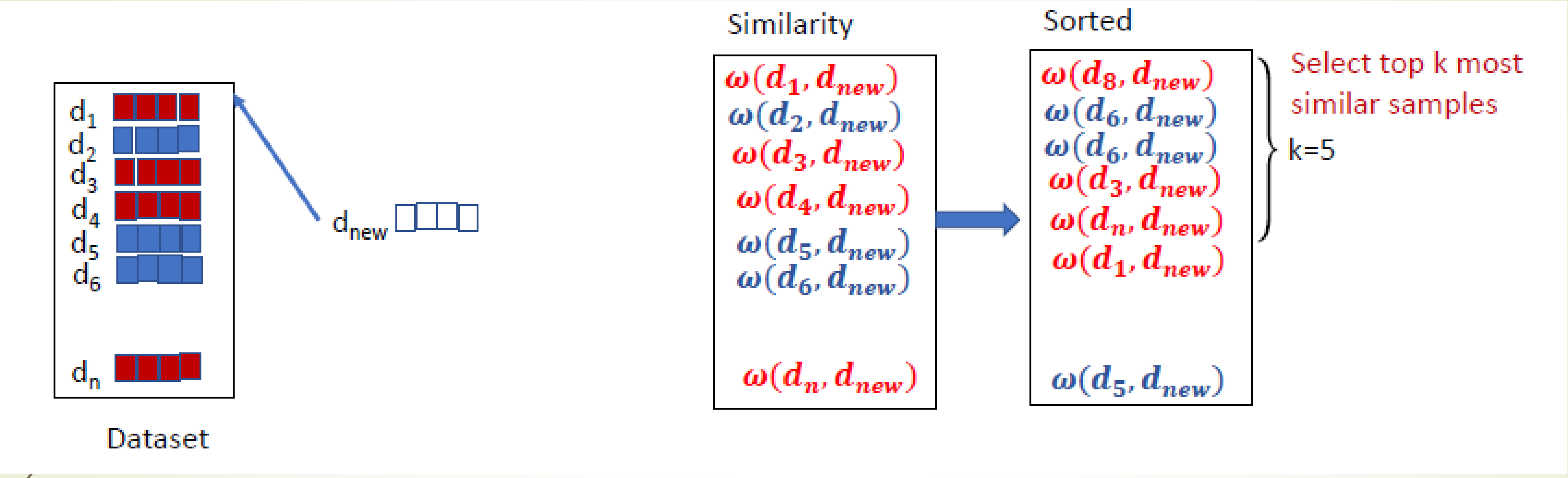


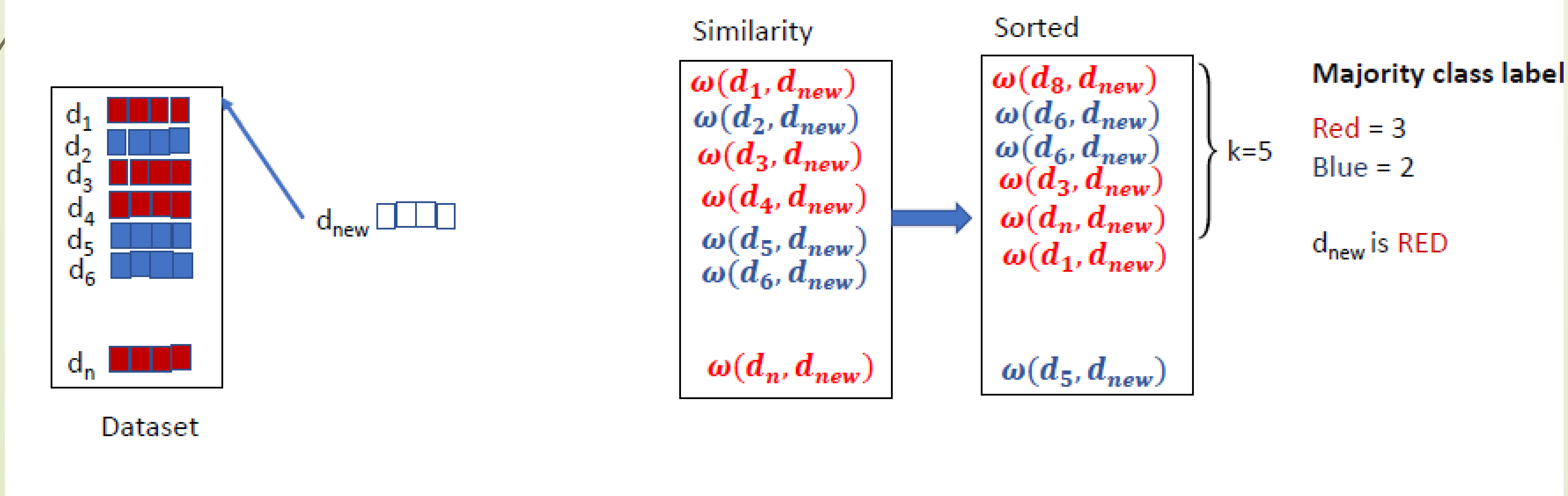




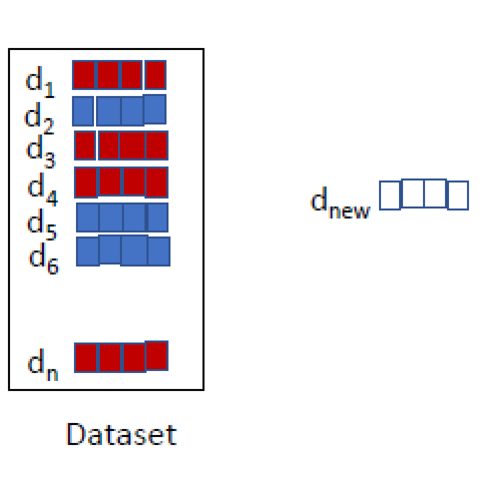
$$Cosine\ Similarity(d_j, d_{new}) = \frac{\sum_{i=1}^{k}(d_{ji}.\, d_{newi})}{\sqrt{\sum_{i=1}^{k} d_{ji}^2}\sqrt{\sum_{i=1}^{n} d_{newi}^2}}$$

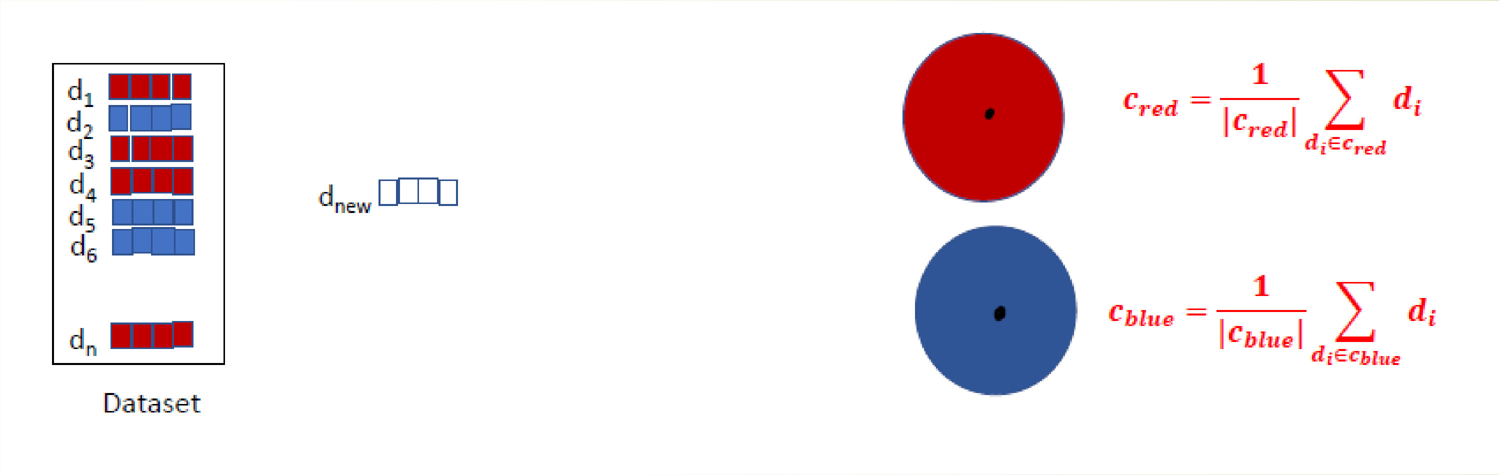


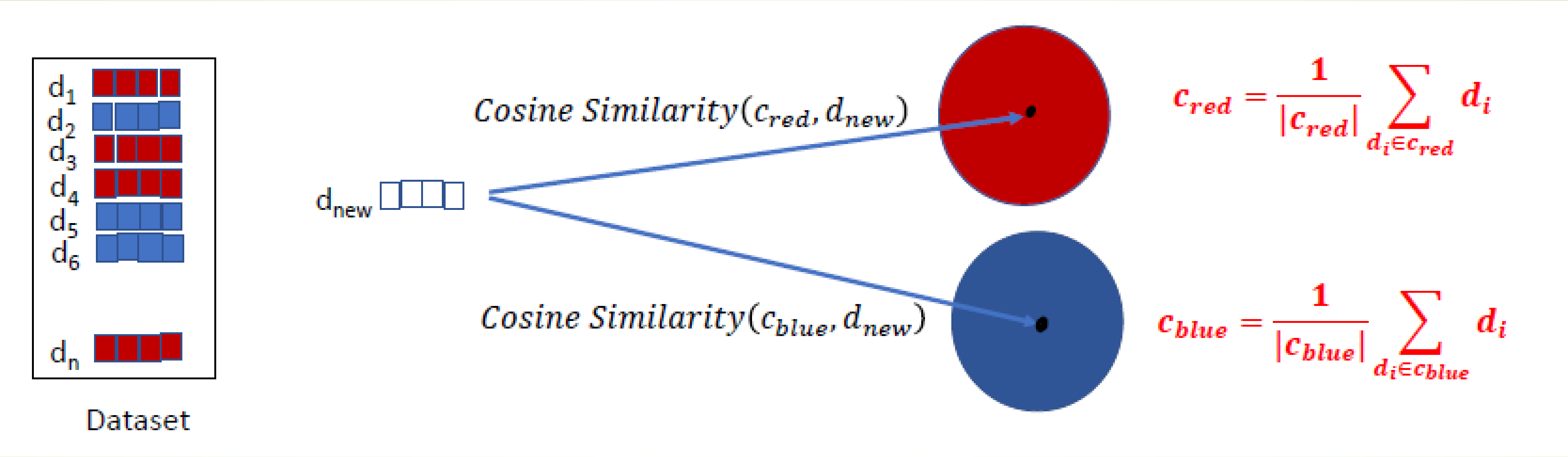

Similarity
Sorted
Select top k most similar samples
k=5
d_1
d_2
d_3
d_4
d_5
d_6
d_n
d_{new}
Dataset
$\omega(d_1, d_{new})$
$\omega(d_2, d_{new})$
$\omega(d_3, d_{new})$
$\omega(d_4, d_{new})$
$\omega(d_5, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_n, d_{new})$
$\omega(d_8, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_3, d_{new})$
$\omega(d_n, d_{new})$
$\omega(d_1, d_{new})$
$\omega(d_5, d_{new})$

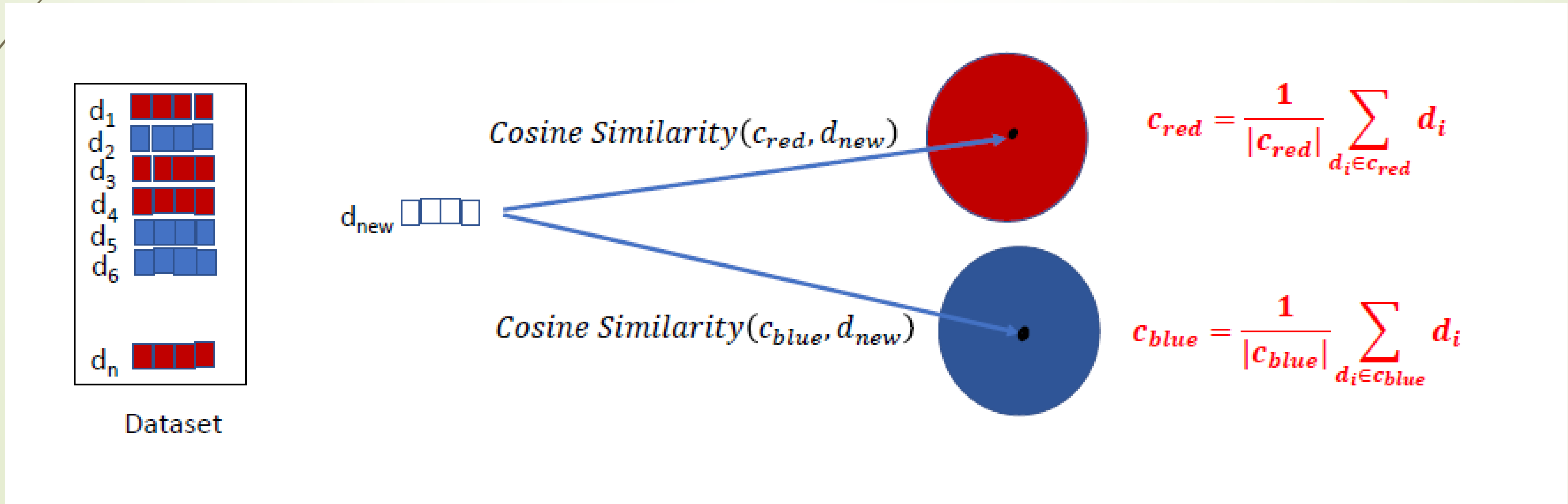

Similarity
Sorted
Majority class label
Red = 3
Blue = 2
d_{new} is RED
k=5
d_1
d_2
d_3
d_4
d_5
d_6
d_n
d_{new}
Dataset
$\omega(d_1, d_{new})$
$\omega(d_2, d_{new})$
$\omega(d_3, d_{new})$
$\omega(d_4, d_{new})$
$\omega(d_5, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_n, d_{new})$
$\omega(d_8, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_6, d_{new})$
$\omega(d_3, d_{new})$
$\omega(d_n, d_{new})$
$\omega(d_1, d_{new})$
$\omega(d_5, d_{new})$

Centroid Based Classifier









Assign Class Label with the nearest Centroid.

Decision Tree (Rule Based Approach)

Example

Features

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Class

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

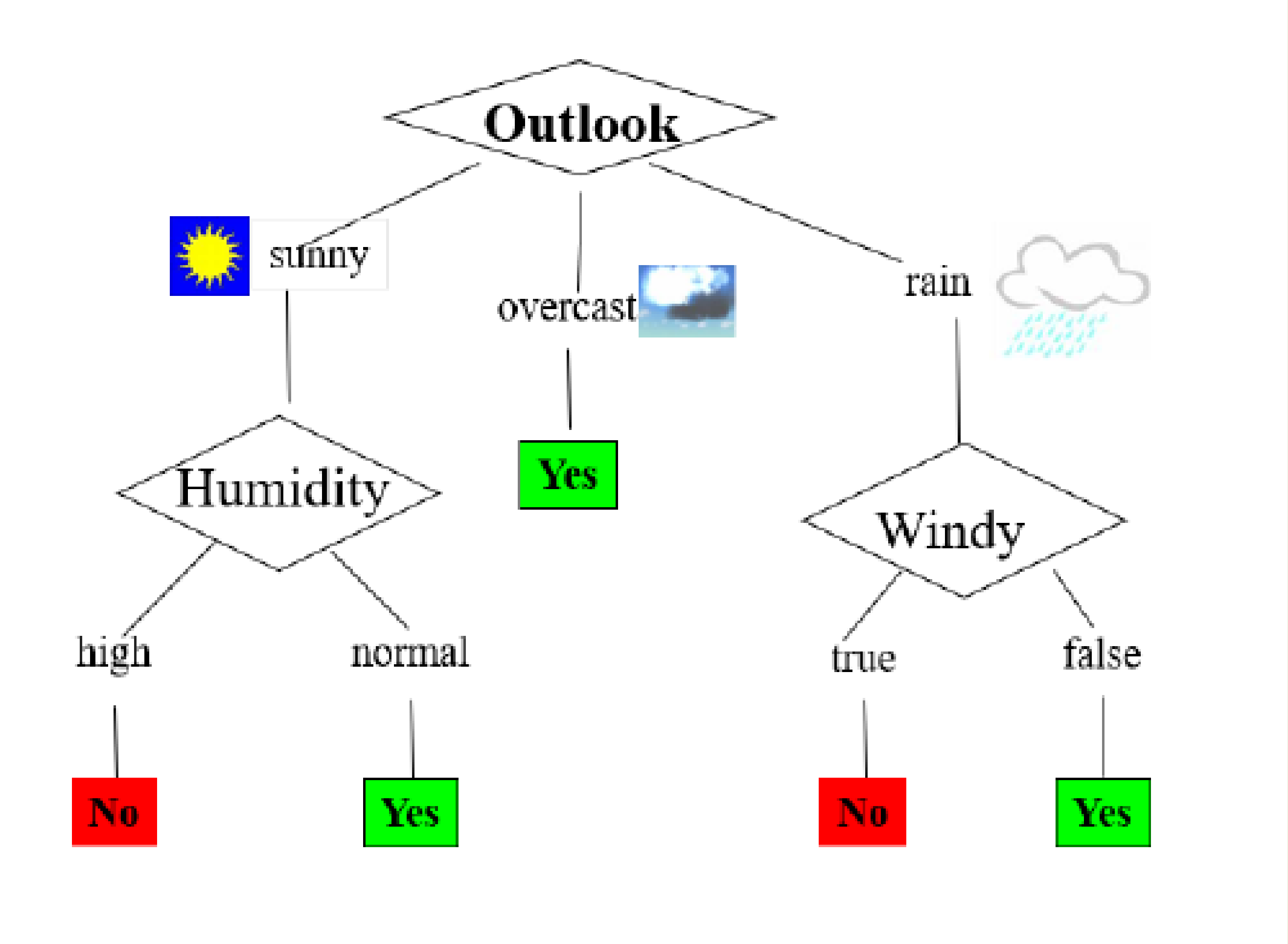
Given : <sunny, cool, high, true>

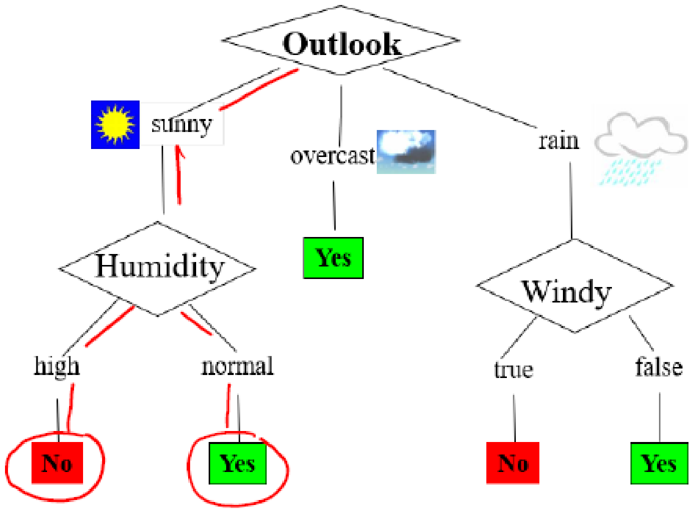
Predict, if there will be a match?

Assume that I have a set of rules:

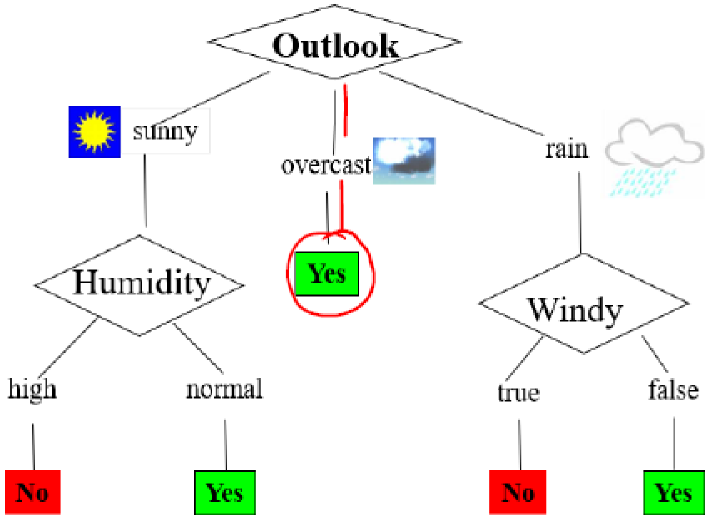
- If *((lookout=sunny)* **and** *(humudity=high)* **and** *(windy=false)) then (yes) else (no)*
- If *(lookout=overcast) then (yes)*
- If *((lookout=sunny)* **and** *(humudity=high)) then (yes) else (no)*
- *so on…..*

Set of rules can be visualized as a tree.



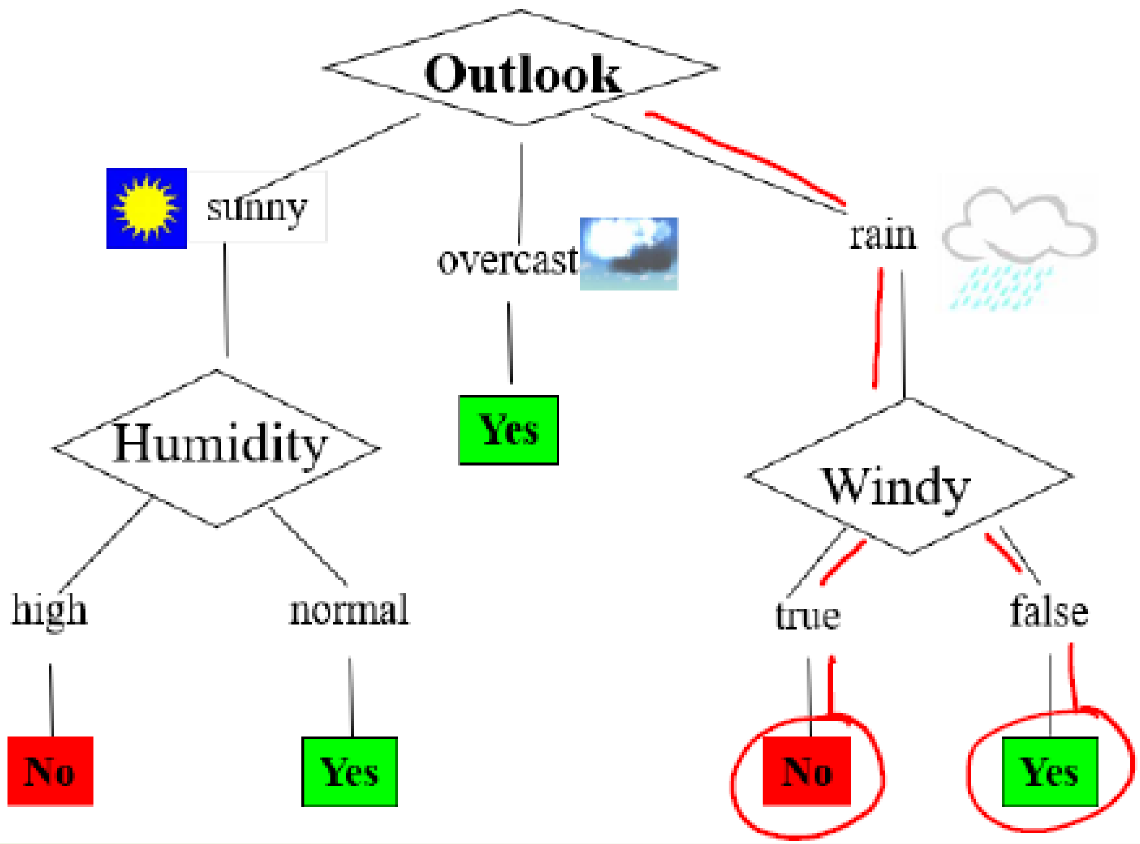


Rule 1: If ((*lookout=sunny*) **and** (*humudity=high*)) *then (yes) else (no)*



Rule 1: If ((*lookout=sunny*) **and** (*humudity=high*)) *then (yes) else (no)*

Rule 2: If *(lookout=overcast) then (yes)*



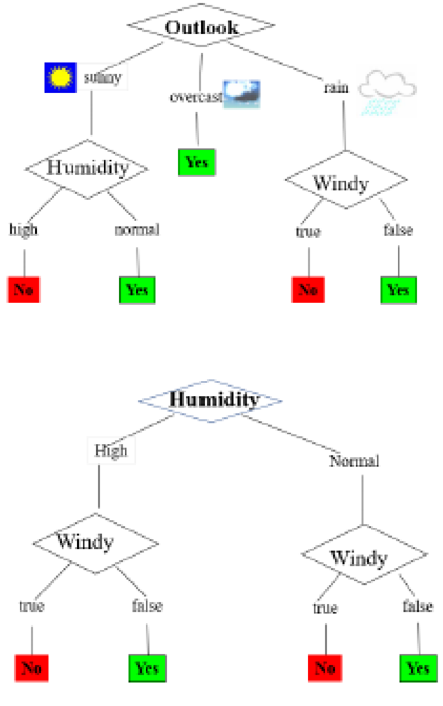
Rule 1: If ((*lookout=sunny*) **and** (*humudity=high*))
then (yes) else (no)

Rule 2: If (*lookout=overcast*) *then (yes)*

Rule 3: If ((*lookout=rain*) **and** (*windy=true*))
then (no) else (yes)

Many possible Trees

outlook	temperature	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Which Tree is the Best?

Models

- **“Traditional” Machine Learning**
  - Support Vector Machines
  - Decision Trees
  - Random Forest
  - …
- **“Deep” Learning Methods**
  - Neuronal Networks
  - Convolutional Neuronal Networks
  - Recurrent Neural Network (RNN)
  - Generative Adversarial Network (GAN)
  - …

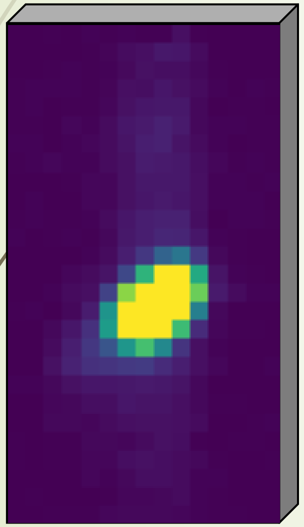
Models

- “Traditional” Machine Learning
  - Support Vector Machines
  - Decision Trees
  - Random Forest
  - …
- “Deep” Learning Methods
  - Neuronal Networks
  - Convolutional Neuronal Networks
  - Recurrent Neural Network (RNN)
  - Generative Adversarial Network (GAN)
  - …

We will look at some of them in this lecture.

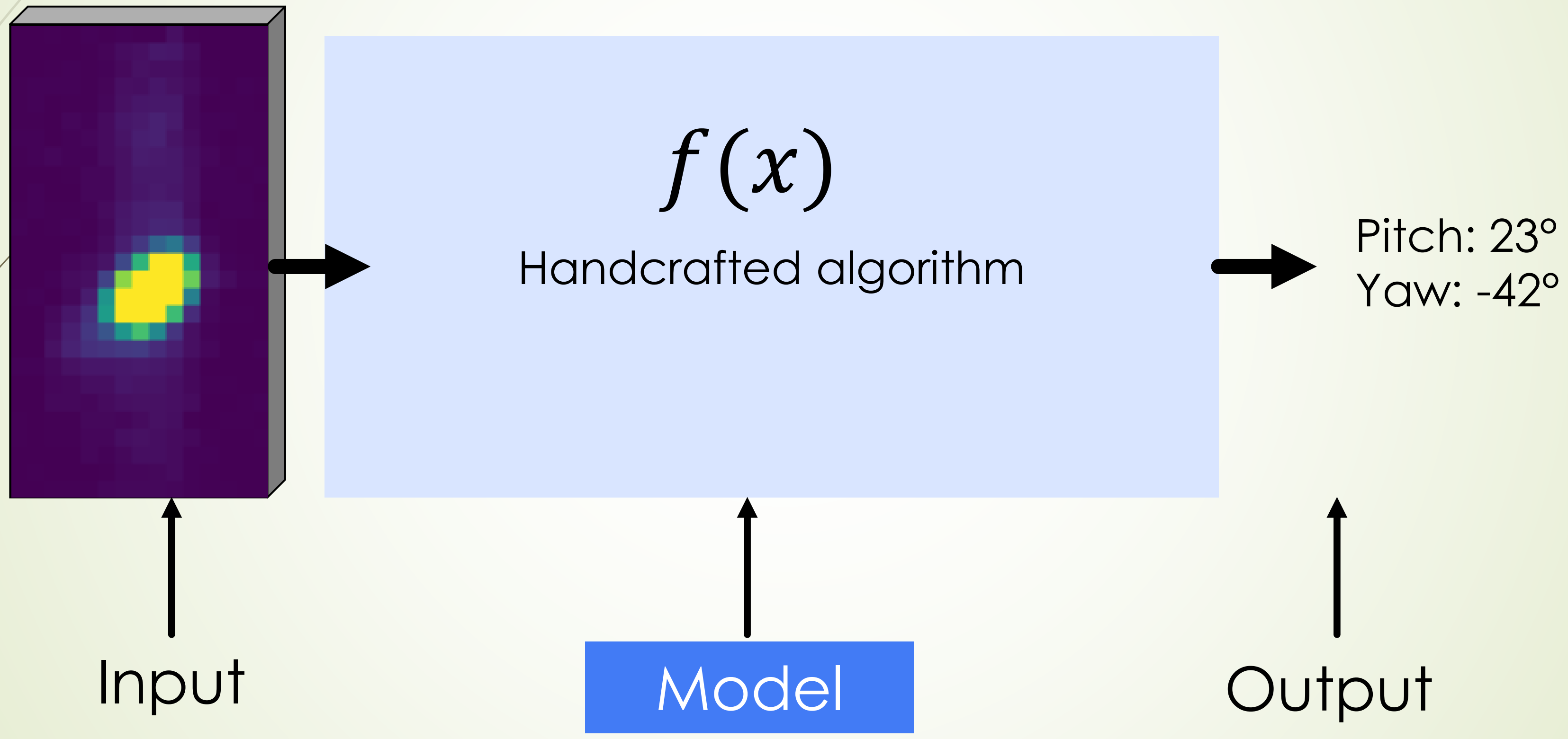
We will focus mainly on Deep Learning Methods.

Example

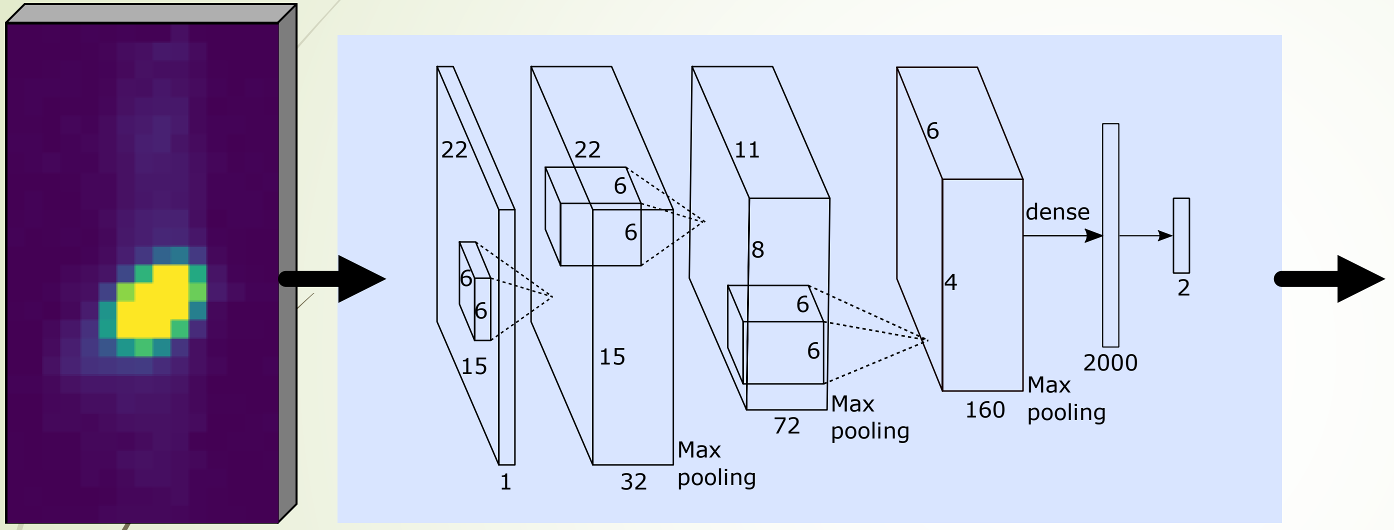


Example

Human-Computer Interaction

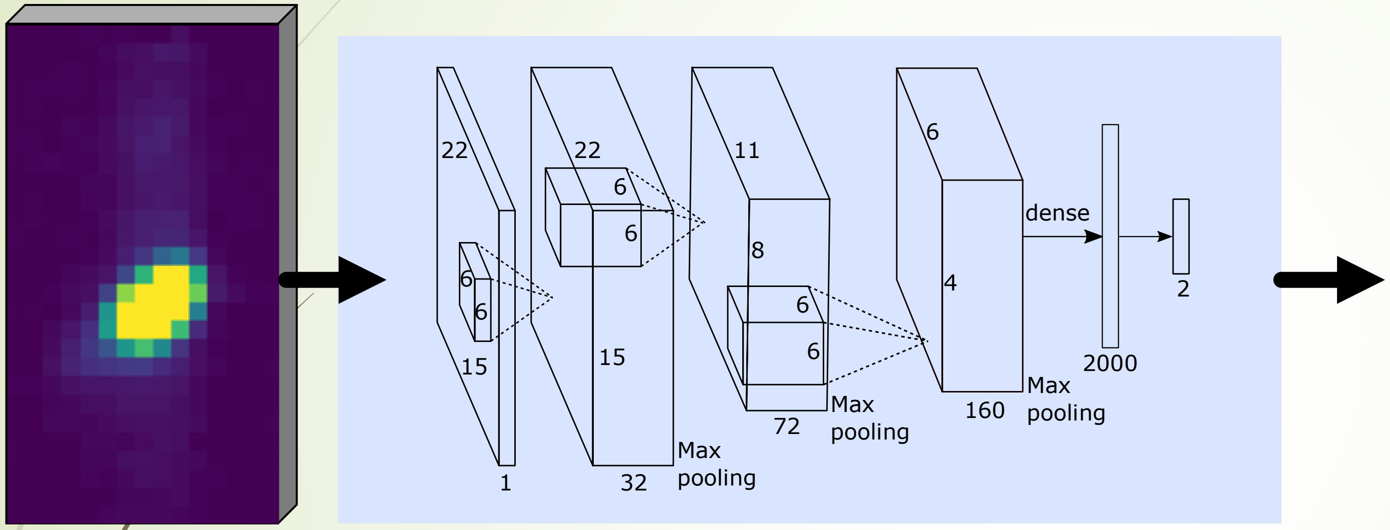


Example



Pitch: 23°
Yaw: -42°

Example



Pitch: 23°
Yaw: -42°

	Pitch			Yaw		
Method	**RMSE**	**MAE**	**SD**	**RMSE**	**MAE**	**SD**
GP reimplementation of Xiao et al. [41]*	14.74	11.78	14.38	56.58	40.51	39.51
pseudo implementation of Xiao et al. [41]**	$\boxed{14.19}$	11.58	8.21	$\boxed{44.53}$	33.39	29.46
CNN + L2	$\boxed{12.8}$	10.09	7.88	$\boxed{24.19}$	17.62	16.58

8.9%

45.7%

Neuronal Networks

What can be trained?

Psychological Review
Vol. 65, No. 6, 1958

THE PERCEPTRON: A PROBABILISTIC MODEL FOR INFORMATION STORAGE AND ORGANIZATION IN THE BRAIN[1]

F. ROSENBLATT

Cornell Aeronautical Laboratory

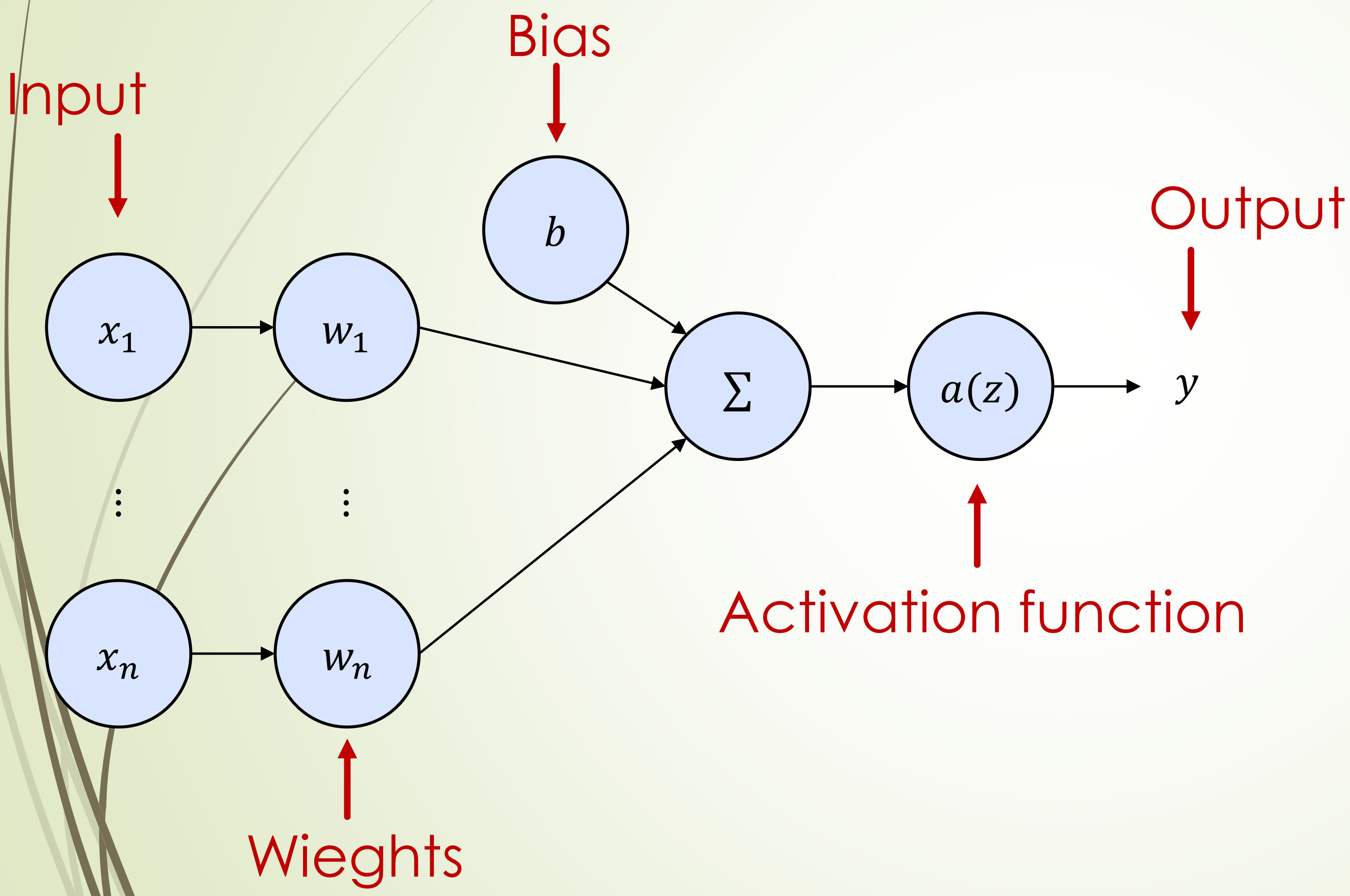
If we are eventually to understand the capability of higher organisms for perceptual recognition, generalization, recall, and thinking, we must first have answers to three fundamental

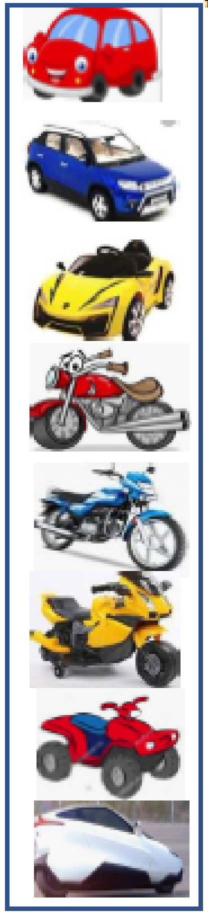
and the stored pattern. According to this hypothesis, if one understood the code or "wiring diagram" of the nervous system, one should, in principle, be able to discover exactlv what an

Frank Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review* 65, no. 6 (1958): 386. DOI: https://psycnet.apa.org/doi/10.1037/h0042519

What is a Perceptron?

Single-Layer Perceptron





#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Pr(CAR| 4,H) = 100%

Pr(BIKE| 4,L) = 100%

Pr(CAR| 2,H) = 100%

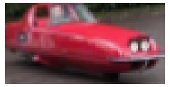
Pr(BIKE| 2,L) = 100%

Pr(CAR| 4,L) = 0%

Pr(BIKE|4,H) = 0%

Pr(CAR| 2,L) = 0%

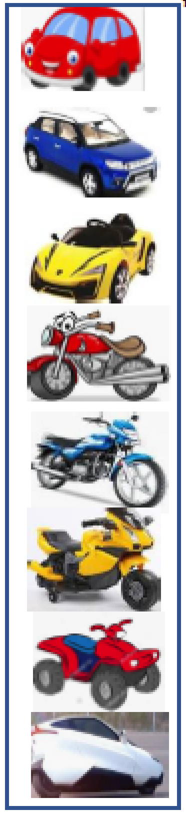
Pr(BIKE| 2,H) = 0%



{2 H} ?

$$Pr(c_i|x), \forall c_i \epsilon C$$

$$class = \arg\max_{c_i} \Pr(c_i|x)$$



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Pr(CAR| 4,H) = 100%

Pr(BIKE| 4,L) = 100%

Pr(CAR| 2,H) = 100%

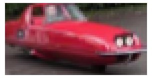
Pr(BIKE| 2,L) = 100%

Pr(CAR| 4,L) = 0%

Pr(BIKE|4,H) = 0%

Pr(CAR| 2,L) = 0%

Pr(BIKE| 2,H) = 0%



{2 H} ?

$$Pr(c_i|x), \forall c_i \epsilon C$$

$$class = \underset{c_i}{\arg\max} \Pr(c_i|x)$$

$Pr(CAR\,|$  $)$

$$Pr(CAR\,|\{2,H\}) = 1$$

$$Pr(BIKE\,|\{2,H\}) = 0$$

$class$ = CAR

7

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

c_i

12

Bayes Rule

Prior

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

c_i

13

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

2

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Bayes Rule

Likelihood

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

2

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Bayes Rule

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

2

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

Bayes Rule

$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

Evidence

x

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

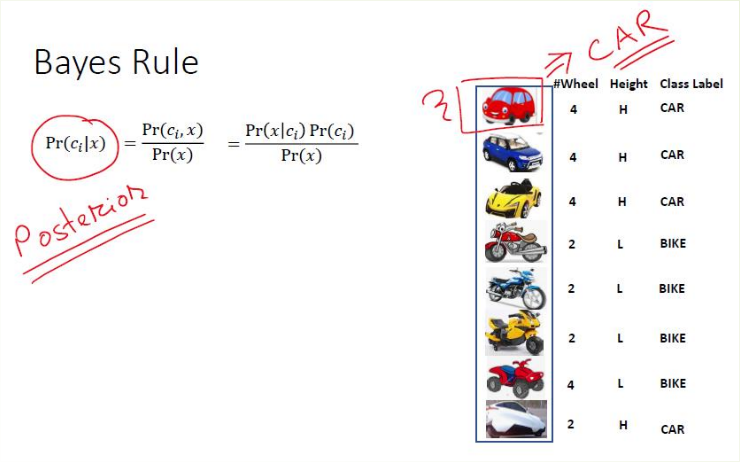
Bayes Rule

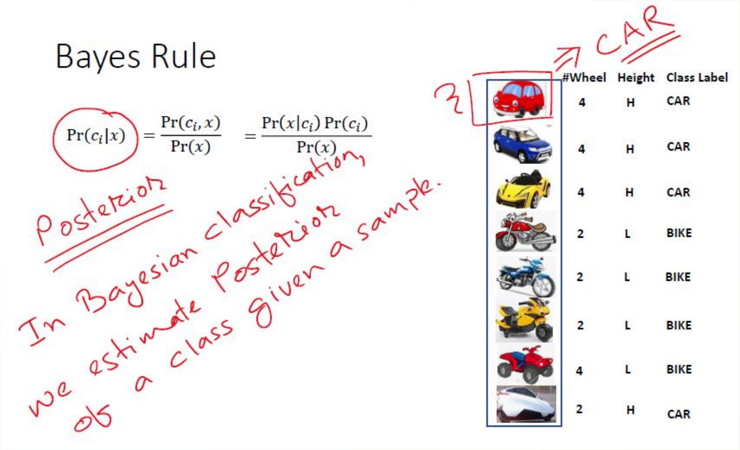
$$\Pr(c_i|x) = \frac{\Pr(c_i, x)}{\Pr(x)} = \frac{\Pr(x|c_i)\Pr(c_i)}{\Pr(x)}$$

$$= \frac{\Pr(x|c_i)\Pr(c_i)}{\underbrace{\Pr(x|c_1)\Pr(c_1) + \Pr(x|c_2)\Pr(c_2) + \ldots + \Pr(x|c_k)\Pr(c_k)}_{\text{Marginalization}}}$$

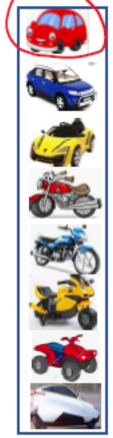
x

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR


Bayes Rule
$Pr(c_i|x) = \frac{Pr(c_i, x)}{Pr(x)} = \frac{Pr(x|c_i)\,Pr(c_i)}{Pr(x)}$
Posterior
? ⇒ CAR
#Wheel Height Class Label
4 H CAR
4 H CAR
4 H CAR
2 L BIKE
2 L BIKE
2 L BIKE
4 L BIKE
2 H CAR


Bayes Rule
$Pr(c_i|x) = \frac{Pr(c_i, x)}{Pr(x)} = \frac{Pr(x|c_i)\,Pr(c_i)}{Pr(x)}$
Posterior
In Bayesian classification, we estimate posterior of a class given a sample.
? ⇒ CAR
#Wheel Height Class Label
4 H CAR
4 H CAR
4 H CAR
2 L BIKE
2 L BIKE
2 L BIKE
4 L BIKE
2 H CAR

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2 w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2 w_3 \dots w_k\}|c_i)\,\Pr(c_i)}{\Pr(\{w_1, w_2 w_3 \dots w_k\})}$$



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(CAR| \;) = \Pr(CAR \mid \{4, H\}) = \frac{\Pr(\{4, H\}|CAR)\,\Pr(CAR)}{\Pr(\{4, H\})}$$

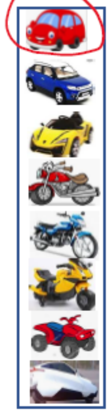
$$= \frac{0.75 \times 0.5}{0.375} = 1$$

$$\Pr(BIKE| \;) = \Pr(BIKE \mid \{4, H\}) = \frac{\Pr(\{4, H\}|BIKE)\,\Pr(BIKE)}{\Pr(\{4, H\})}$$

$$= \frac{0 \times 0.5}{0.375} = 0$$

22

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2 w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2 w_3 \dots w_k\}|c_i)\,\Pr(c_i)}{\Pr(\{w_1, w_2 w_3 \dots w_k\})}$$



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(CAR| \;) = \Pr(CAR \mid \{4, H\}) = \frac{\Pr(\{4, H\}|CAR)\,\Pr(CAR)}{\Pr(\{4, H\})}$$

$$= \frac{0.75 \times 0.5}{0.375} = 1$$

$$\Pr(BIKE| \;) = \Pr(BIKE \mid \{4, H\}) = \frac{\Pr(\{4, H\}|BIKE)\,\Pr(BIKE)}{\Pr(\{4, H\})}$$

$$= \frac{0 \times 0.5}{0.375} = 0$$

$\boxed{CAR}$

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2 w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2 w_3 \ldots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2 w_3 \ldots w_k\})}$$

$\Pr(CAR|$  $)$

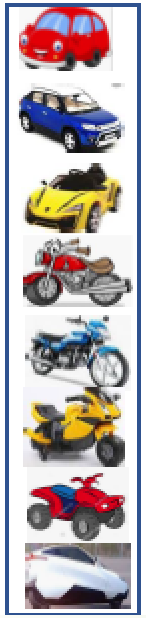
$$= \Pr(CAR \mid \{4, H\})$$

$$= \frac{\Pr(\{4,H\}|CAR)\Pr(CAR)}{\Pr(\{4, H\})}$$

$\Pr(BIKE|$  $)$

$$= \Pr(BIKE \mid \{4, H\})$$

$$= \frac{\Pr(\{4,H\}|BIKE)\Pr(BIKE)}{\Pr(\{4, H\})}$$



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2 w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2 w_3 \ldots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2 w_3 \ldots w_k\})}$$

$\Pr(CAR|$  $)$

$$= \Pr(CAR \mid \{4, H\})$$

$$= \frac{\Pr(\{4,H\}|CAR)\Pr(CAR)}{\Pr(\{4, H\})}$$

$\Pr(BIKE|$  $)$

$$= \Pr(BIKE \mid \{4, H\})$$

$$= \frac{\Pr(\{4,H\}|BIKE)\Pr(BIKE)}{\Pr(\{4, H\})}$$

Same

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2 w_3 \dots w_k\}) = \frac{\dots}{\Pr(\{w_1, w_2 w_3 \dots w_k\})}$

$\Pr(CAR| \quad)$

$= \Pr(CAR \mid \{4, H\})$

$\sim \Pr(\{4, H\}|CAR) \Pr(CAR)$

$\Pr(BIKE| \quad)$

$= \Pr(BIKE \mid \{4, H\})$

$\sim \Pr(\{4, H\}|BIKE) \Pr(BIKE)$

Relation still maintains

#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2 w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2 w_3 \dots w_k\}|c_i) \Pr(c_i)}{\Pr(\{w_1, w_2 w_3 \dots w_k\})}$$

$\Pr(CAR| \quad)$

$= \Pr(CAR \mid \{4, H\})$

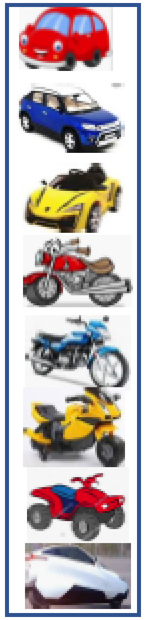
$\sim \Pr(\{4, H\}|CAR) \Pr(CAR)$

$\Pr(BIKE| \quad)$

$= \Pr(BIKE \mid \{4, H\})$

$\sim \Pr(\{4, H\}|BIKE) \Pr(BIKE)$

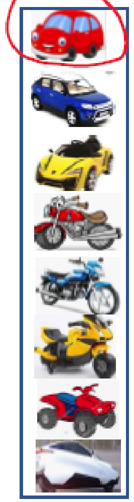
Relation still maintains



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2, w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \dots w_k\})}$$

If ***k*** (the number of classes) is **small,**

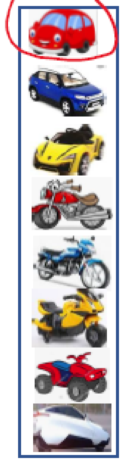


#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2, w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \dots w_k\})}$$

If ***k*** (the number of classes) is **small,**

estimating **likelihood** $\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)$ is **feasible**.



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2, w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)\,\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \ldots w_k\})}$$

However, if ***k*** (the number of classes) is **very large,**

estimating **likelihood** $\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)$ is **a very expensive task** over **a large dataset.**

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2, w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)\,\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \ldots w_k\})}$$

However, if ***k*** (the number of classes) is **very large,**

estimating **likelihood** $\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)$ is **a very expensive task** over **a large dataset.**

$$\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i) \;= \Pr(w_1|w_2, w_3, \ldots, w_3, c_i).Pr(w_2|w_3, w_4, \ldots, w_3, c_i) \ldots .Pr(w_k|c_i)$$

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2, w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \dots w_k\})}$$

However, if ***k*** (the number of classes) is **very large,**

estimating **likelihood** $\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)$ is **a very expensive task** over **a large dataset.**

$$\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i) = \Pr(w_1|w_2, w_3, \dots, w_3, c_i).Pr(w_2|w_3, w_4, \dots, w_3, c_i) \dots .Pr(w_k|c_i)$$

Naïve Bayes Classifier

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2, w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \dots w_k\})}$$

To simplify the estimation, we make an **assumption**

- The features are **conditionally independent**.

Naïve Bayes Classifier

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2, w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \ldots w_k\})}$$

To simplify the estimation, we make an **assumption**

- The features are **conditionally independent**.

$= Pr(w_1|c_i)$ $= Pr(w_2|c_i)$

$$\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i) \;= \Pr(w_1|w_2, w_3, \ldots, w_3, c_i)\, Pr(w_2|w_3, w_4, \ldots, w_3, c_i) \ldots. Pr(w_k|c_i)$$

Naïve Bayes Classifier

$$\Pr(c_i|x) \;= \Pr(c_i \mid \{w_1, w_2, w_3 \ldots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \ldots w_k\})}$$

To simplify the estimation, we make an **assumption**

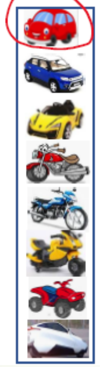
- The features are **conditionally independent**.

Bayesian: $\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i) \;= \Pr(w_1|w_2, w_3, \ldots, w_3, c_i). Pr(w_2|w_3, w_4, \ldots, w_3, c_i) \ldots. Pr(w_k|c_i)$

Naïve Bayes: $\Pr(\{w_1, w_2, w_3 \ldots w_k\}|c_i) \;\sim \Pr(w_1|c_i). Pr(w_2|c_i) \ldots. Pr(w_k|c_i) = \prod_{j=1}^{k} \Pr(w_j|c_i)$

Naïve Bayes Classifier

$$\Pr(c_i|x) = \Pr(c_i \mid \{w_1, w_2, w_3 \dots w_k\}) = \frac{\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i)\Pr(c_i)}{\Pr(\{w_1, w_2, w_3 \dots w_k\})}$$



#Wheel	Height	Class Label
4	H	CAR
4	H	CAR
4	H	CAR
2	L	BIKE
2	L	BIKE
2	L	BIKE
4	L	BIKE
2	H	CAR

$$\sim \prod_{j=1}^{k} \Pr(w_j|c_i)\Pr(c_i)$$

$$\Pr(CAR \mid \{4, H\}) = \Pr(4|CAR) \times \Pr(H|CAR) \times \Pr(CAR)$$

$$= 0.75 \times 1 \times 0.5 = 0.375$$

$$\Pr(BIKE \mid \{4, H\}) = \Pr(4|BIKE) \times \Pr(H|BIKE) \times \Pr(BIKE)$$

$$= 0.25 \times 0 \times 0.5 = 0$$

$\boxed{CAR}$

What is one of the estimate in the likelihood is zero?

$$\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i) \sim \Pr(w_1|c_i).Pr(w_2|c_i) \dots .Pr(w_k|c_i) = \prod_{j=1}^{k} \Pr(w_j|c_i)$$

$$\Pr(CAR \mid \{4, M\}) = \Pr(4|CAR) \times \Pr(M|CAR) \times \Pr(CAR)$$

$$= 0.75 \times 0 \times 0.5 = 0$$

What is one of the estimate in the likelihood is zero?

$$\Pr(\{w_1, w_2, w_3 \dots w_k\}|c_i) \sim \Pr(w_1|c_i).Pr(w_2|c_i) \dots .Pr(w_k|c_i) = \prod_{j=1}^{k} \Pr(w_j|c_i)$$

$$\Pr(CAR|\{4, M\}) = \Pr(4|CAR) \times \Pr(M|CAR) \times \Pr(CAR)$$

$$= 0.75 \times 0 \times 0.5 = 0$$

Smoothing

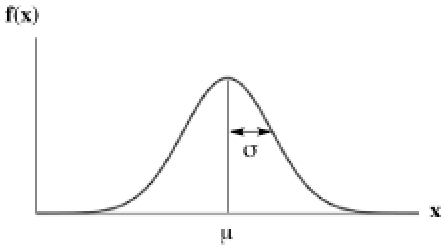
$$Pr(M|CAR) = \frac{0 + Small}{|CAR| + Large}$$

In some of the machine learning tools, you may find

- Naïve Bayes with Gaussian
- Naïve Bayes with Multinomial

In some of the machine learning tools, you may find

- Naïve Bayes with Gaussian



$$f(x) = \frac{1}{\boxed{\sigma}\sqrt{2\pi}} e^{-(x-\boxed{\mu})^2/2\sigma^2}$$

$e = 2.71828$

In some of the machine learning tools, you may find

- Naïve Bayes with Multinomial

$$f(x_1, \ldots, x_k; n, p_1, \ldots, p_k) = \Pr(X_1 = x_1 \text{ and } \ldots \text{ and } X_k = x_k)$$

$$= \begin{cases} \dfrac{n!}{x_1! \cdots x_k!} p_1^{x_1} \cdots p_k^{x_k}, & \text{when } \sum_{i=1}^{k} x_i = n \\ 0 & \text{otherwise,} \end{cases}$$

K Neighbors Classifier

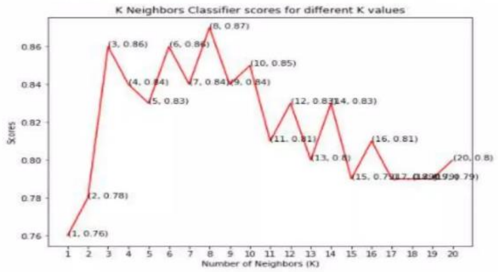
This classifier looks for the classes of K nearest neighbors of a given data point and based on the majority class, it assigns a class to this data point. However, the number of neighbors can be varied. I varied them from 1 to 20 neighbors and calculated the test score in each case.

In [19]:
```
knn_scores = []
for k in range(1,21):
    knn_classifier = KNeighborsClassifier(n_neighbors = k)
    knn_classifier.fit(X_train, y_train)
    knn_scores.append(knn_classifier.score(X_test, y_test))
```

Then, I plot a line graph of the number of neighbors and the test score achieved in each case.

In [20]:
```
plt.plot([k for k in range(1, 21)], knn_scores, color = 'red')
for i in range(1,21):
    plt.text(i, knn_scores[i-1], (i, knn_scores[i-1]))
plt.xticks([i for i in range(1, 21)])
plt.xlabel('Number of Neighbors (K)')
plt.ylabel('Scores')
plt.title('K Neighbors Classifier scores for different K values')
```

Out[20]: `Text(0.5, 1.0, 'K Neighbors Classifier scores for different K values')`



As you can see, we achieved the maximum score of 87% when the number of neighbors was chosen to be 8.

Decision Tree Classifier

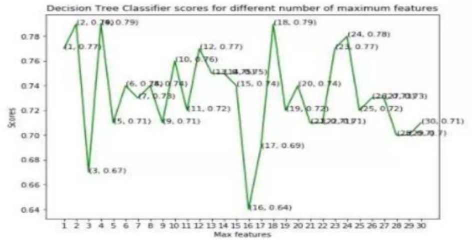
This classifier creates a decision tree based on which, it assigns the class values to each data point. Here, we can vary the maximum number of features to be considered while creating the model. I range features from 1 to 30 (the total features in the dataset after dummy columns were added).

In [24]:
```
dt_scores = []
for i in range(1, len(X.columns) + 1):
    dt_classifier = DecisionTreeClassifier(max_features = i, random_state = 0)
    dt_classifier.fit(X_train, y_train)
    dt_scores.append(dt_classifier.score(X_test, y_test))
```

Once we have the scores, we can then plot a line graph and see the effect of the number of features on the model scores.

In [25]:
```
plt.plot([i for i in range(1, len(X.columns) + 1)], dt_scores, color = 'green')
for i in range(1, len(X.columns) + 1):
    plt.text(i, dt_scores[i-1], (i, dt_scores[i-1]))
plt.xticks([i for i in range(1, len(X.columns) + 1)])
plt.xlabel('Max features')
plt.ylabel('Scores')
plt.title('Decision Tree Classifier scores for different number of maximum features')
```

Out[25]: `Text(0.5, 1.0, 'Decision Tree Classifier scores for different number of maximum features')`



From the line graph above, we can clearly see that the maximum score is 79% and is achieved for maximum features being selected to be either 2, 4 or 18.

Data Processing

To work with categorical variables, we should break each categorical column into dummy columns with 1s and 0s. Let's say we have a column Gender, with values 1 for Male and 0 for Female. It needs to be converted into two columns with the value 1 where the column would be true and 0 where it will be false. Take a look at the Gist below.

To get this done, we use the get_dummi

To get this done, we use the get_dummies() method from pandas. Next, we need to scale the dataset for which we will use the StandardScaler. The fit_transform() method of the scaler scales the data and we update the columns.

In [16]:
```
df = pd.get_dummies(df, columns = ['sex', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'ca', 'thal'])
standardScaler = StandardScaler()
columns_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
df[columns_to_scale] = standardScaler.fit_transform(df[columns_to_scale])
```

The dataset is now ready. We can begin with training our models.

Machine Learning In this project, I took 4 algorithms and varied their various parameters and compared the final models. I split the dataset into 67% training data and 33% testing data.

In [18]:
```
y = df['target']
X = df.drop(['target'], axis = 1)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.33, random_state = 0)
```

Conclusion

Introduction to Machine Learning

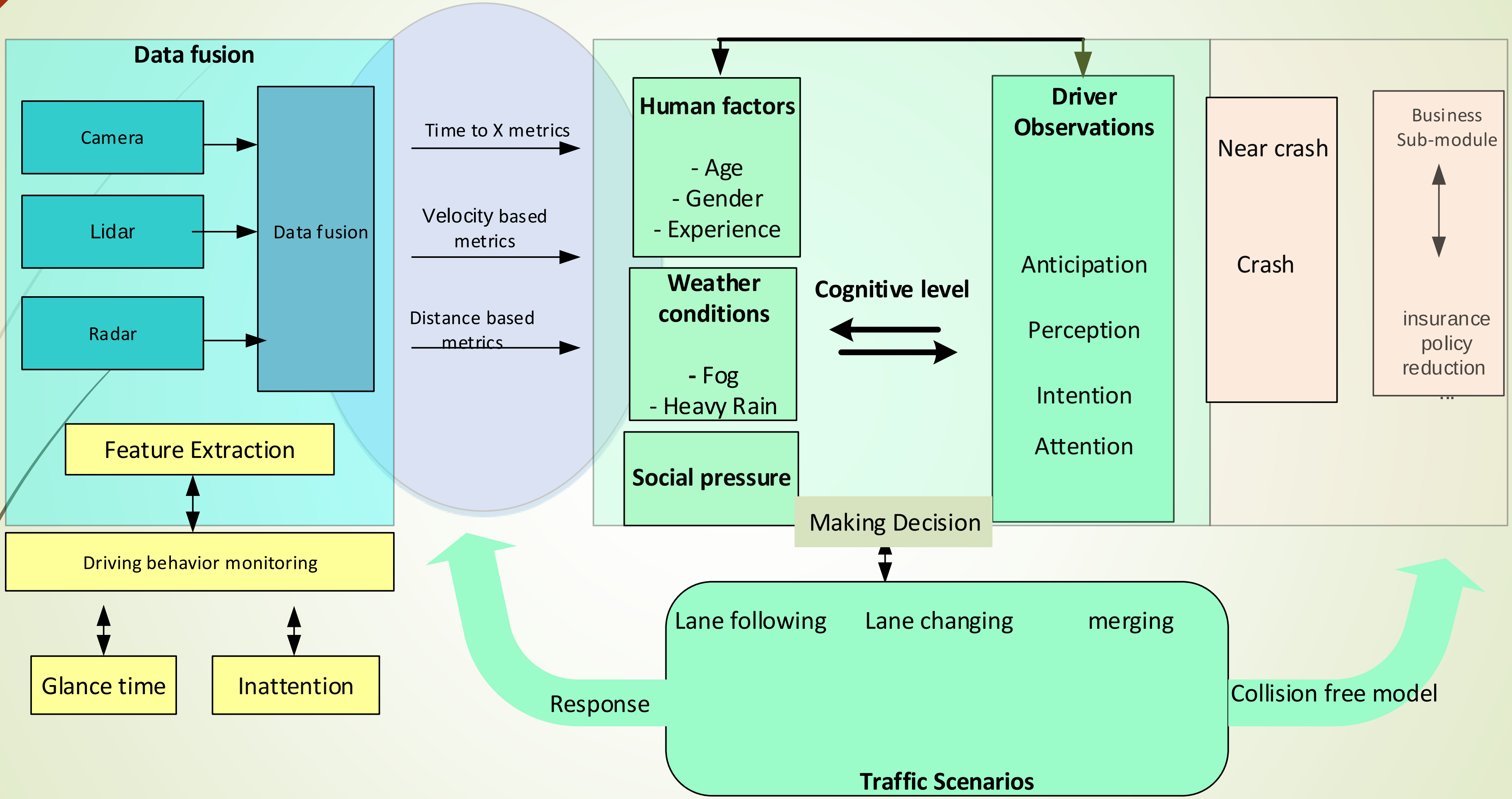
- Practical Examples
- General understanding of the model f(x)
- Deep Learning Approaches
  - Perceptron
  - Weights & biases can be trained
  - Activation function

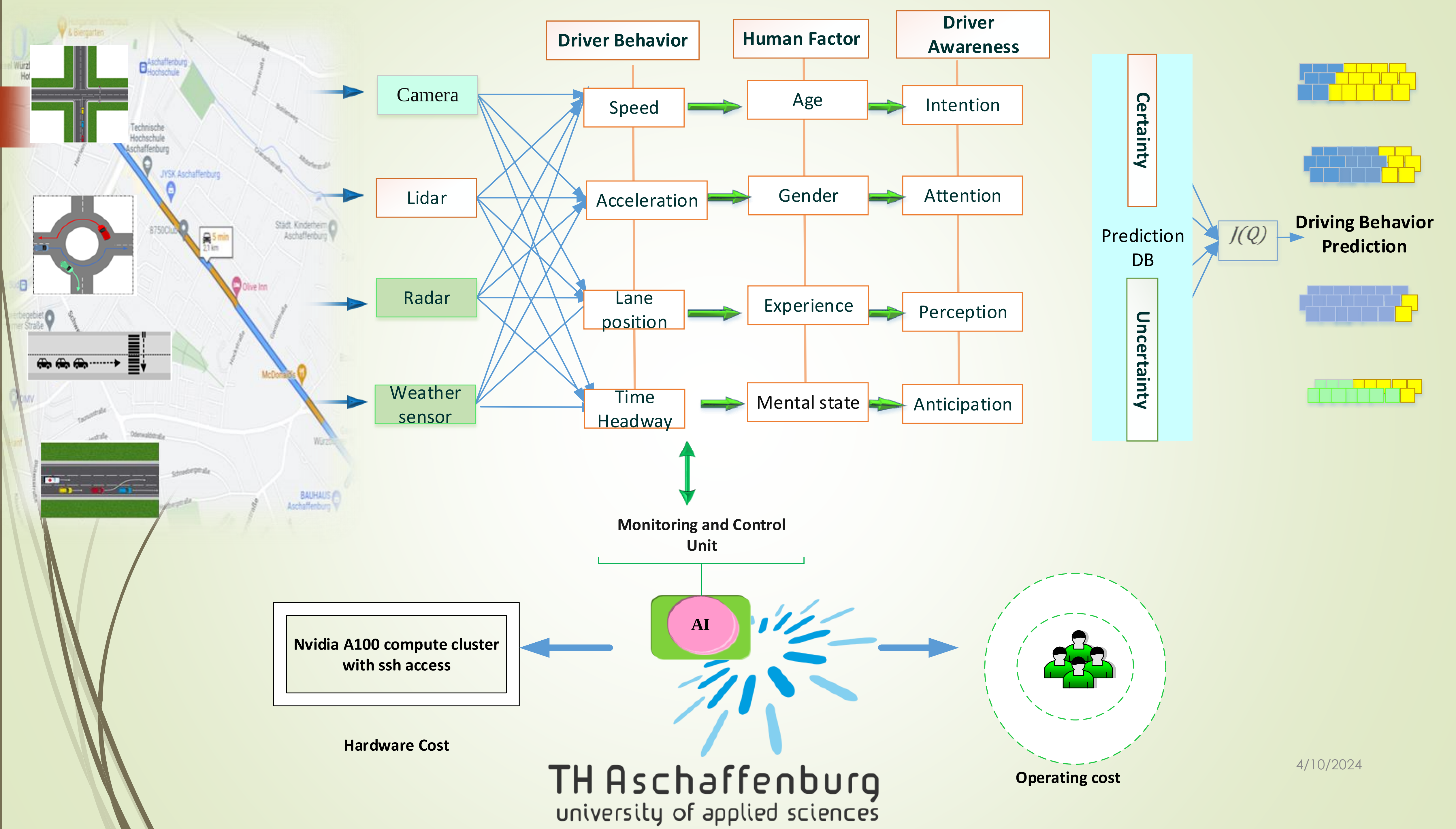
Driving Behavior

Input $\rightarrow$ $f(x)$ $\rightarrow$ Output

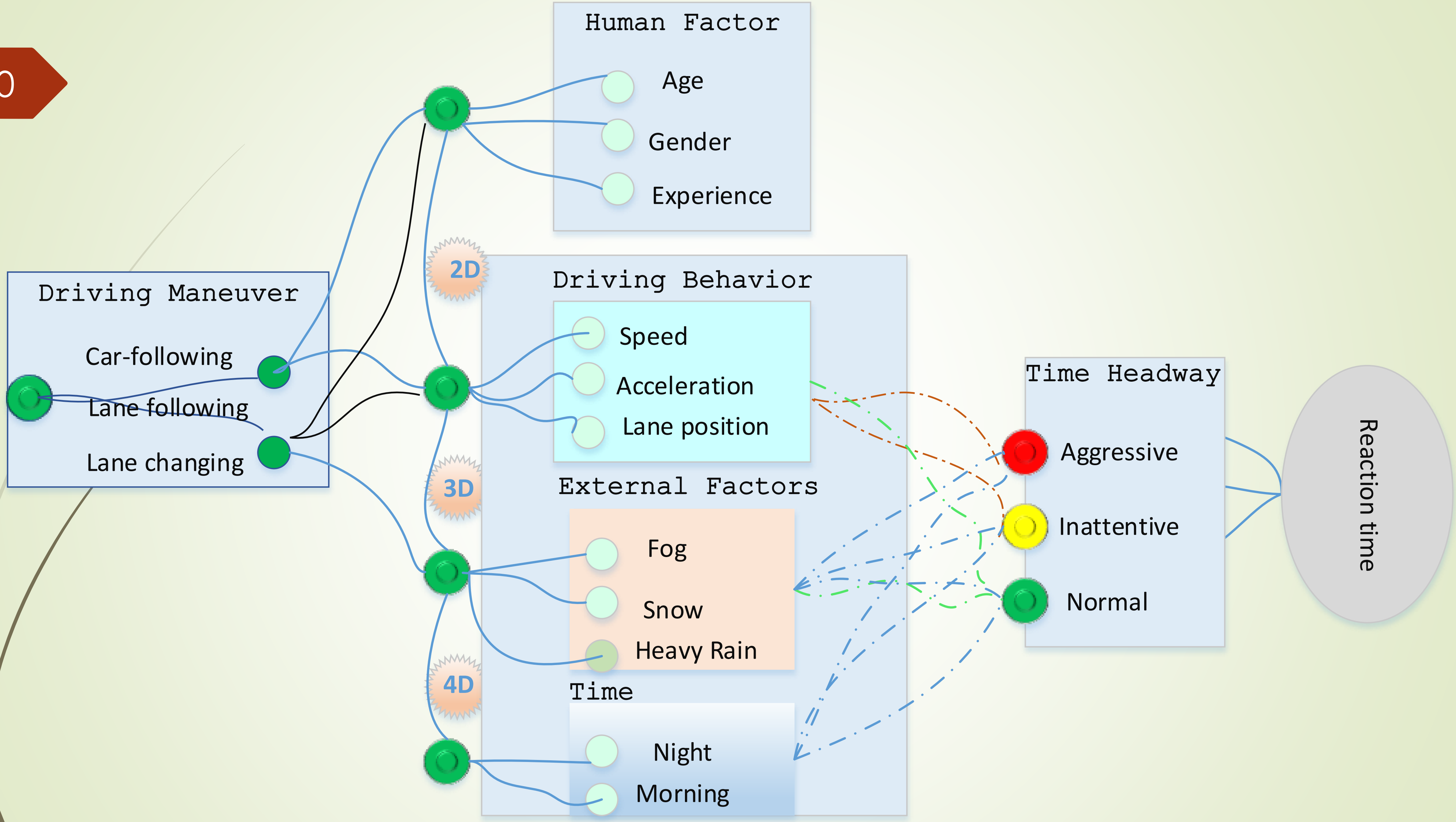
$\uparrow$

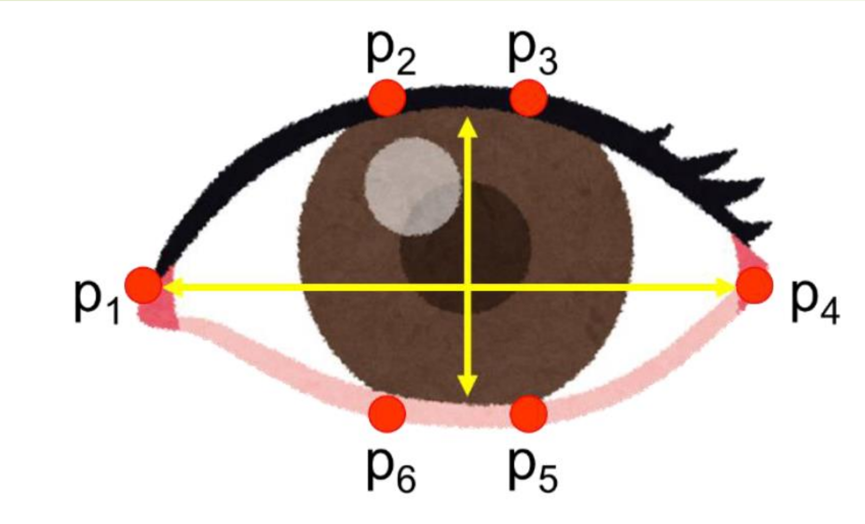
Model?

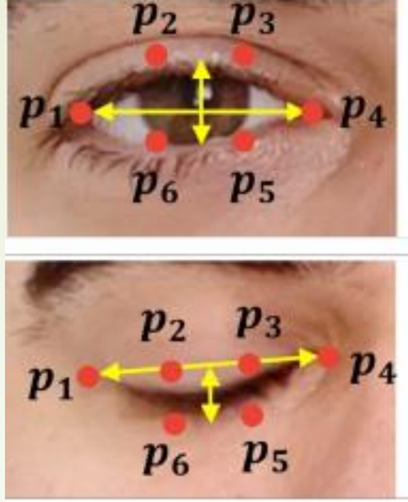

Data fusion
Camera
Lidar
Radar
Data fusion
Time to X metrics
Velocity based metrics
Distance based metrics
Feature Extraction
Driving behavior monitoring
Glance time
Inattention
Human factors
- Age
- Gender
- Experience
Weather conditions
- Fog
- Heavy Rain
Social pressure
Cognitive level
Driver Observations
Anticipation
Perception
Intention
Attention
Making Decision
Near crash
Crash
Business Sub-module
insurance policy reduction
...
Lane following
Lane changing
merging
Traffic Scenarios
Response
Collision free model

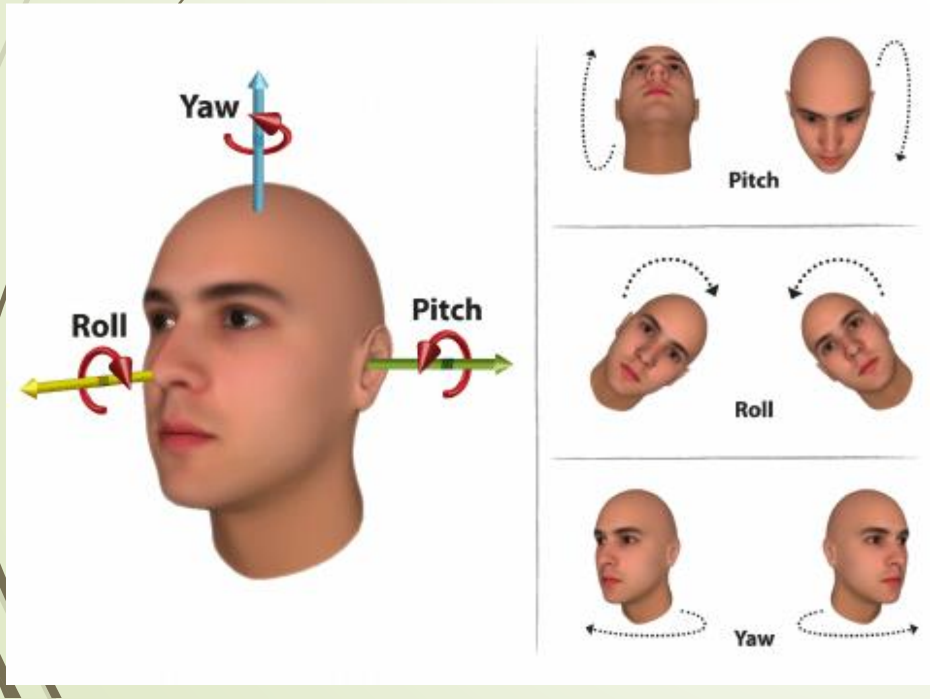


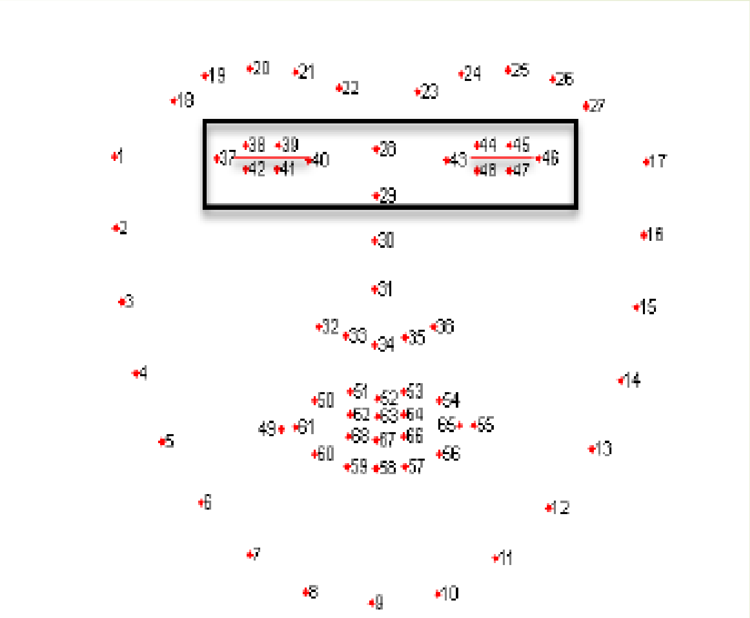
Camera
Lidar
Radar
Weather sensor
Driver Behavior
Speed
Acceleration
Lane position
Time Headway
Human Factor
Age
Gender
Experience
Mental state
Driver Awareness
Intention
Attention
Perception
Anticipation
Certainty
Prediction DB
Uncertainty
J(Q)
Driving Behavior Prediction
Monitoring and Control Unit
AI
Nvidia A100 compute cluster with ssh access
Hardware Cost
Operating cost
TH Aschaffenburg
university of applied sciences


Human Factor
Age
Gender
Experience
Driving Maneuver
Car-following
Lane following
Lane changing
2D
3D
4D
Driving Behavior
Speed
Acceleration
Lane position
External Factors
Fog
Snow
Heavy Rain
Time
Night
Morning
Time Headway
Aggressive
Inattentive
Normal
Reaction time


p2 p3
p1 p4
p6 p5


p2 p3
p1 p4
p6 p5
p2 p3
p1 p4
p6 p5


Yaw
Roll
Pitch
Pitch
Roll
Yaw



Thank You for Your Attention