



Movie Recommendation System

Why ML?

- ☐ We are growing at a very faster level and demands are also increasing and require an automated and centralized system.
- ☐ Can we do very difficult computations manually?

- ❑ Machine learning is a branch which deals in with the ability of a machine to self learn and improve according to the past experiences in place of explicit programming.

Traditional programming

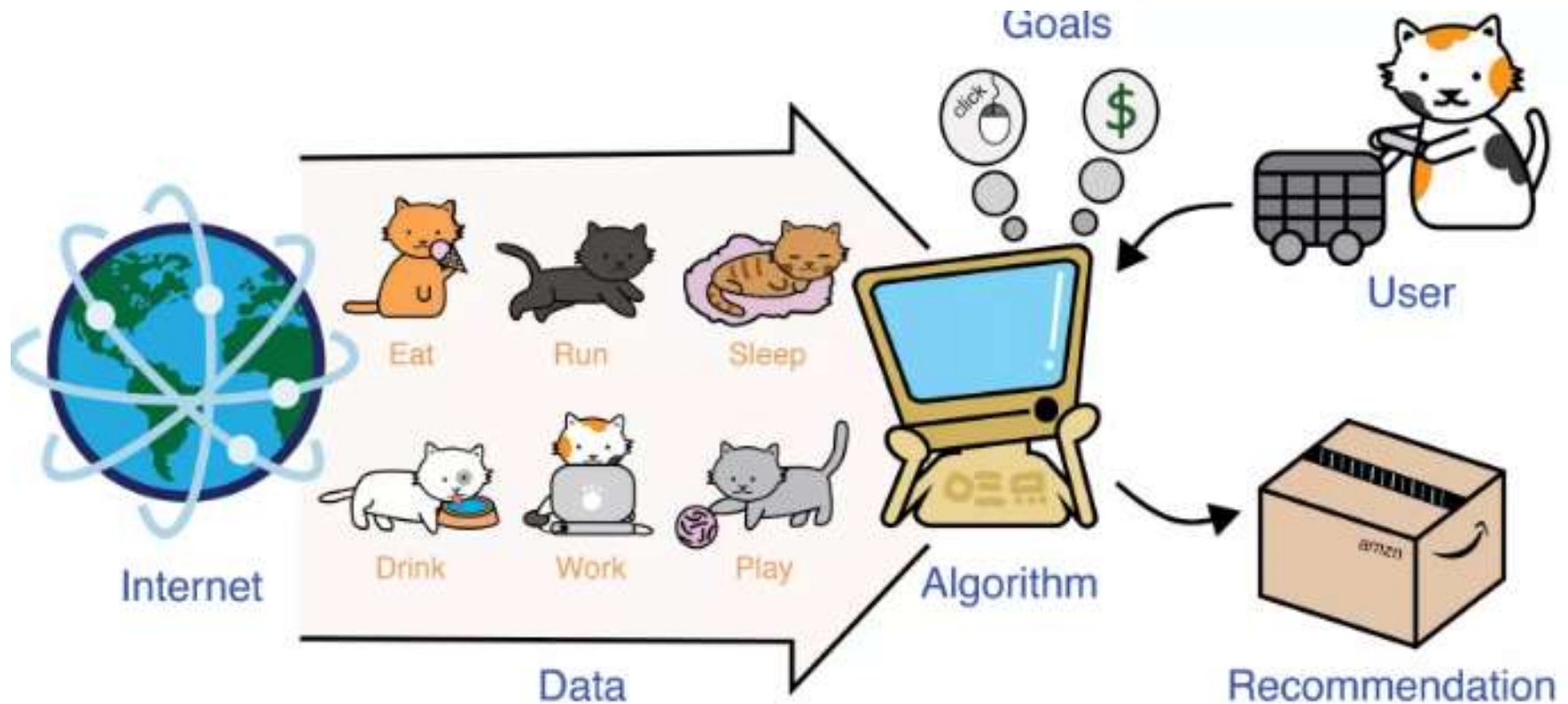


Machine Learning



Goal of ML

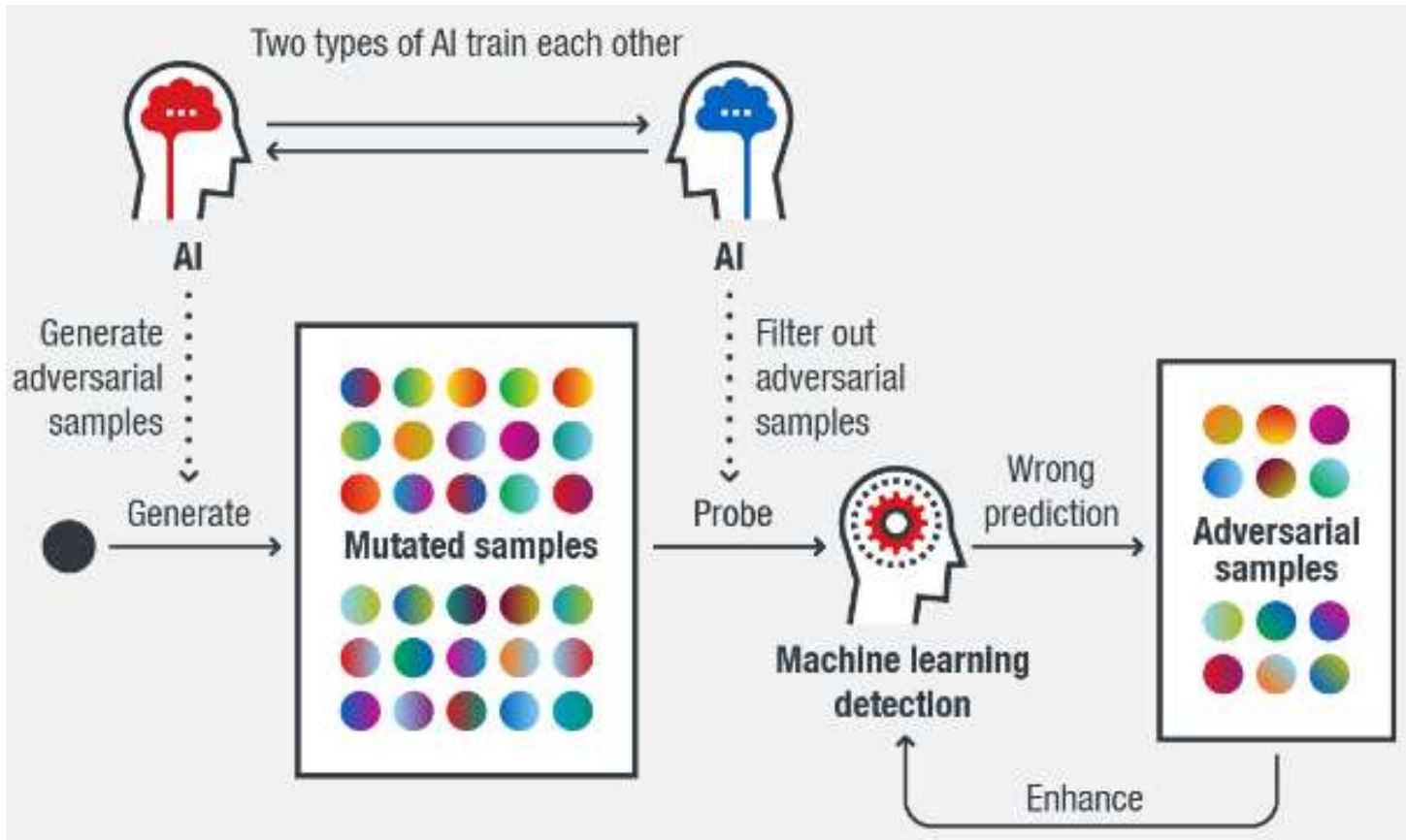
- ❑ Removal of redundancies
- ❑ Dimensionality Reduction



- Map building according to current traffic



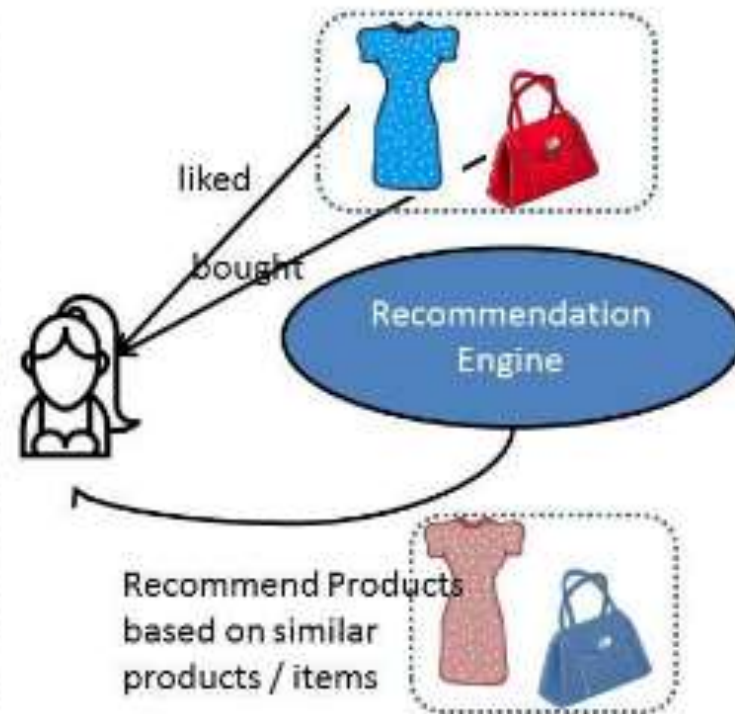
□ Email Spam and Malware Filtering



□ Product recommendation



(a). Collaborative Filtering



(b). Content Filtering

Python programming

Programming languages

☐ Types:

- ☐ High level language(user friendly)
- ☐ Low level language(machine friendly)
- ☐ Assembly language

☐ Converters

- ☐ Compiler
- ☐ Interpreter
- ☐ Assembler

☐ Standard of conversion

- ☐ ASCII

❑ Variables

```
>>> a=10
```

```
>>> type(a)
```

```
<class 'int'>
```

```
>>> a=20.30
```

```
>>> type(a)
```

```
<class 'float'>
```

```
>>> a="hello"
```

```
>>> type(a)
```

```
<class 'str'>
```

- ❑ Operators(Symbols use for computation purpose)

- ❑ Inputting

```
>>> a=input("enter your name")  
enter your nameLiveWire
```

```
>>> a  
'LiveWire'
```

- ❑ Outputting

```
>>> print("hello world")  
hello world
```

❑ Array

❑ List

```
>>> arr=[3,2,4,5,6,7,1,2,4]
```

```
>>> arr
```

```
[3, 2, 4, 5, 6, 7, 1, 2, 4]
```

❑ Tuple

```
>>> tup=(1,4,5,8,9,2)
```

```
>>> tup
```

```
(1, 4, 5, 8, 9, 2)
```

❑ Set

```
>>> se={4,2,5,7,8,9}
```

```
>>> se
```

```
{2, 4, 5, 7, 8, 9}
```


❑ Array

❑ Dictionary

```
>>> dic={'Office':'LiveWire','Place':'Jaipur'}
```

```
>>> dic
```

```
{'Office': 'LiveWire', 'Place': 'Jaipur'}
```

❑ String

```
>>> st="LiveWire"
```

```
>>> type(st)
```

```
<class 'str'>
```

❑ Loops

```
>>> for var in range(5):  
    print("hello") #print hello for 5 times  
  
>>>a=0  
  
>>>while(a<5):  
    print("hello") #print hello 5 times  
    a+=1
```

❑ Conditional statements

```
>>> a=5
```

```
>>> if(a==5):
```

```
    print("value is 5")
```

```
>>> else:
```

```
    print("value is not 5")
```

❑ Functions

```
>>> def funName():
```

```
    print("simple function")
```

- ❑ Modules and packages(Predefined and user defined)

- ❑ Pip and conda package

```
>>>pip install packageName
```

```
>>>conda install packageName
```

- ❑ Numpy

```
>>>import numpy as np
```

```
>>>np.sum([1,2,34,5,6,7,8])
```

- ❑ Pandas(Library for data manipulation and analysis)

- ❑ Matplotlib(Library for making graphical display)

Pre-program preparation

- ☐ Data collection(collecting data from some source)
- ☐ Data exploration and profiling(checking for incomplete)
- ☐ Formatting data to make it consistent
- ☐ Improve data quality(checking erroneous data)
- ☐ Feature engineering(convert raw data into features)
- ☐ Splitting data into training, evaluation sets

How to achieve them?

Data preprocessing

❑ Data collection(collection from source)

```
>>>import pandas as pd  
>>>df=pd.read_excel(open('fileName.xlsx','rb'))  
>>>x=df.iloc[:, :-1]  
>>>y=df.iloc[:, -1]
```

❑ Imputer and Dropna(filling missing value or dropping the tuples)

```
>>>from sklearn.preprocessing import Imputer  
>>>imp=Imputer()  
>>>array=imp.fit_transform(array)
```

❑ Label encoder(converting strings into labels)

```
>>>from sklearn.preprocessing import LabelEncoder  
>>>lbl=LabelEncoder()  
>>>array=lbl.fit_transform(array)
```

Data preprocessing

❑ One hot encoder(making dummies variables)

```
>>>from sklearn.preprocessing import OneHotEncoder
```

```
>>>ohe=OneHotEncoder(categorical_features=[attribute no])
```

```
>>>array=ohe.fit_transform(array)
```

❑ Standard scaler(Normalizing data)

```
>>>from sklearn.preprocessing import StandardScaler
```

```
>>>sc=StandardScaler()
```

```
>>>array=sc.fit_transform(array)
```

❑ Train test split(splitting data into training and testing set)

```
>>>from sklearn.model_validation import train_test_split
```

```
>>>res=train_test_split(input, output, test_size=0.20)
```

❑ Confusion matrix

Actual

		Predicted/Classified	
		Negative	Positive
Actual	Negative	998	0
	Positive	1	1

❑ Recall

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

❑ Precision

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

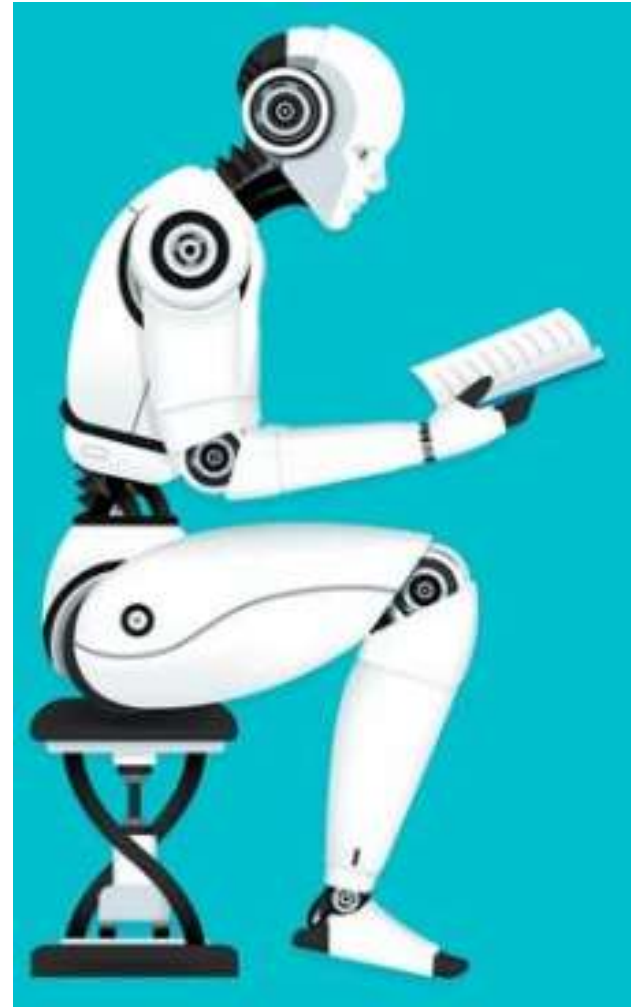
❑ F-Score

$$\text{F-Score} = \frac{2 \cdot \text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}}$$

Machine Learning

Introduction to ML

- ❑ Learning is the conversion of experience into expertise or knowledge. Its an algorithm based technology to solve any problem.



Requirement of ML

- ❑ Programming too complex task

- ❑ Tasks performed by humans

- ❑ whatever task we do in routine are not that easy to elaborate well to transform it into a program.

- ❑ Tasks beyond human capacities

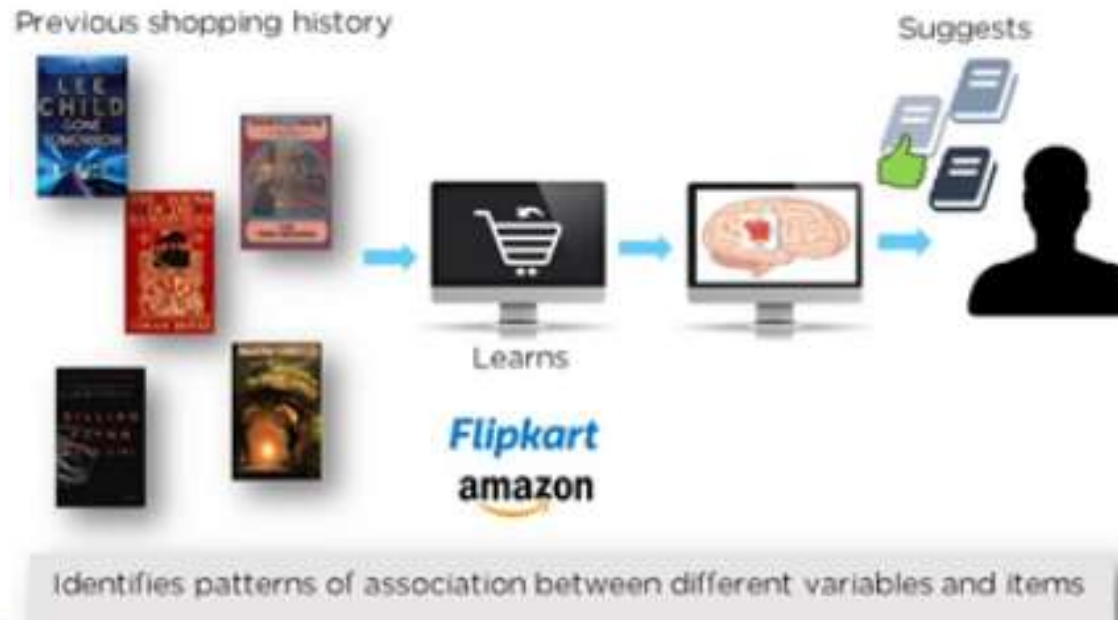
- ❑ Highly complex task

- ❑ Adaptivity

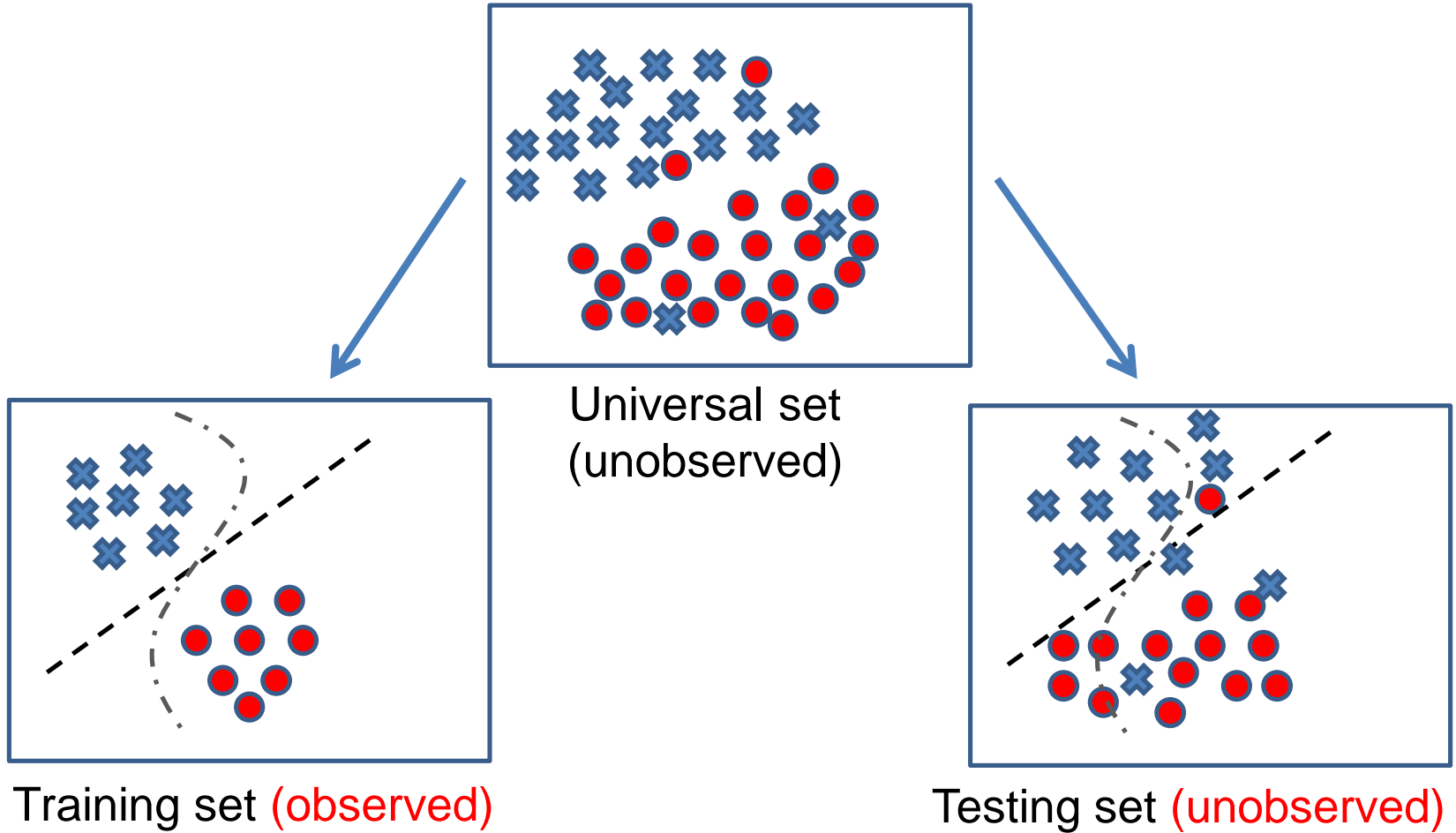
- ❑ Changes in the prediction as per the change in data

What is ML

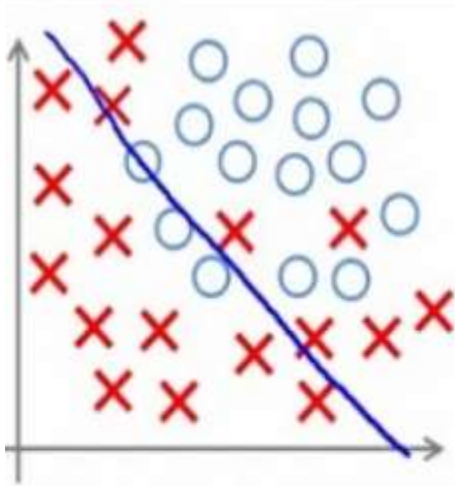
- ❑ It is a branch of artificial intelligence, focus on the design and development of algorithm that allow computers to evolve behaviours based on empirical data.
- ❑ Example: The person who interested in buying a T-shirt can also buy jeans(**www.amazon.in**)



Training and testing set

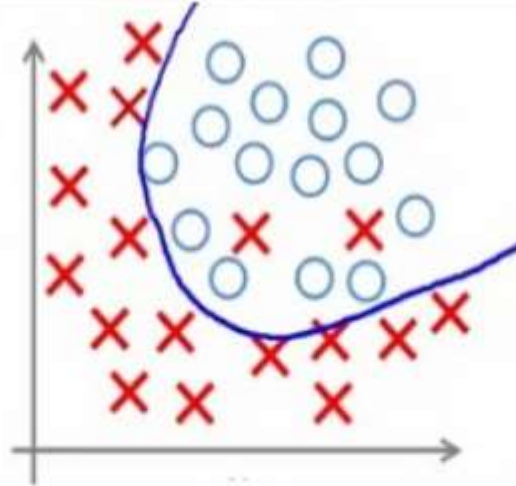


Over-fitting & Under-fitting

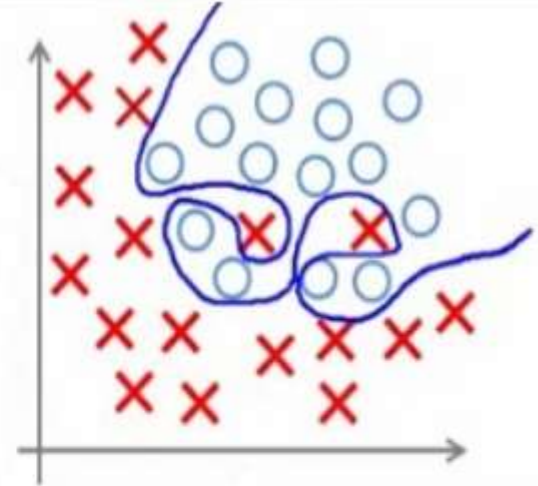


Under-fitting

(too simple to
explain the
variance)



Appropriate-fitting

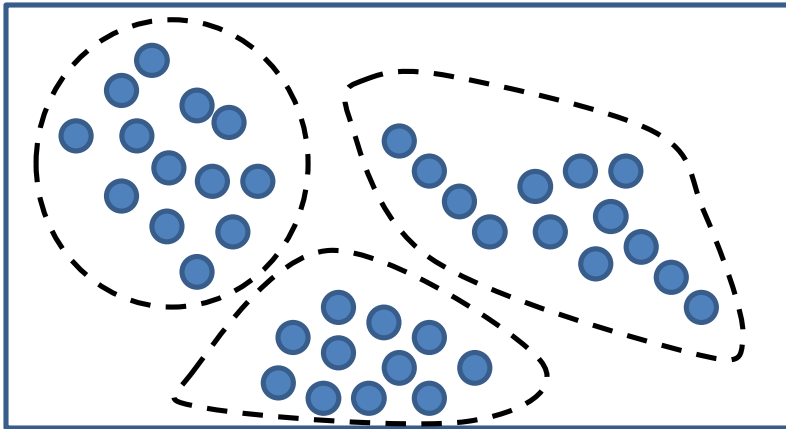


Over-fitting

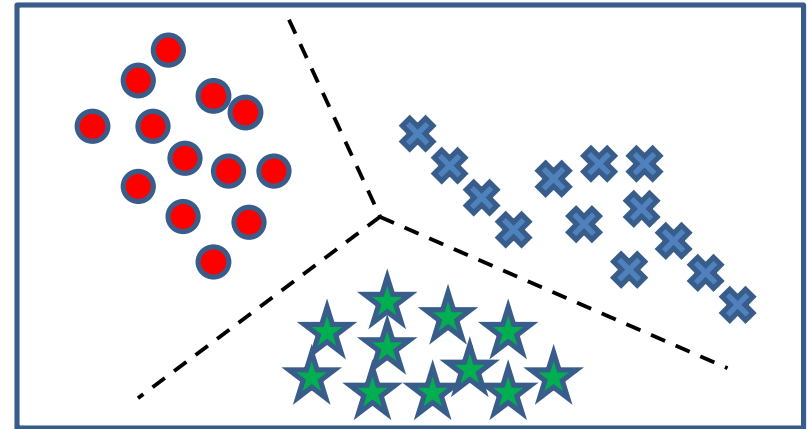
(forcefitting -- too
good to be true)

Types of learning

- ☐ Supervised learning
 - ☐ Classification(For dataset with categorical dependent variable)
 - ☐ Regression(For dataset with continuous dependent variable)
- ☐ Unsupervised learning



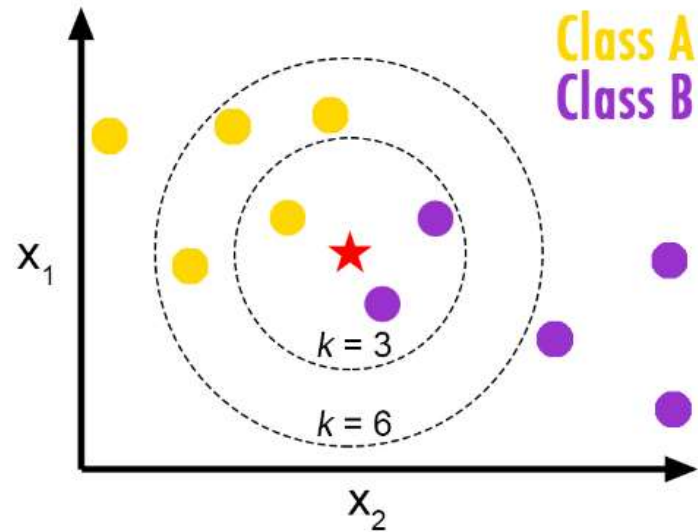
Unsupervised learning



Supervised learning

Supervised learning

- Classification algorithms
 - KNN



Movie Recommendation System

Dataset

1	movielfld	title	genres
2	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy
3	2	Jumanji (1995)	Adventure Children Fantasy
4	3	Grumpier Old Men (1995)	Comedy Romance
5	4	Waiting to Exhale (1995)	Comedy Drama Romance
6	5	Father of the Bride Part II (1995)	Comedy
7	6	Heat (1995)	Action Crime Thriller
8	7	Sabrina (1995)	Comedy Romance
9	8	Tom and Huck (1995)	Adventure Children
10	9	Sudden Death (1995)	Action
11	10	GoldenEye (1995)	Action Adventure Thriller
12	11	American President, The (1995)	Comedy Drama Romance
13	12	Dracula: Dead and Loving It (1995)	Comedy Horror
14	13	Balto (1995)	Adventure Animation Children
15	14	Nixon (1995)	Drama
16	15	Cutthroat Island (1995)	Action Adventure Romance
17	16	Casino (1995)	Crime Drama
18	17	Sense and Sensibility (1995)	Drama Romance
19	18	Four Rooms (1995)	Comedy
20	19	Ace Ventura: When Nature Calls (1995)	Comedy
21	20	Money Train (1995)	Action Comedy Crime Drama Thriller
22	21	Get Shorty (1995)	Comedy Crime Thriller

Dataset

userId	movieId	rating	timestamp
1	1	4	964982703
1	3	4	964981247
1	6	4	964982224
1	47	5	964983815
1	50	5	964982931
1	70	3	964982400
1	101	5	964980868
1	110	4	964982176
1	151	5	964984041
1	157	5	964984100
1	163	5	964983650
1	216	5	964981208
1	223	3	964980985
1	231	5	964981179
1	235	4	964980908

- ☐ Data Loading and Preprocessing
- ☐ Data Filtering
- ☐ Sparse Matrix Conversion
- ☐ Model Training
- ☐ Recommendation Function

case study

Thankyou