

# Exercise 4: High Level Synthesis Intro

## KNN based Movie Recommendation System



AGGELOS FERIKOGLU  
MANOLIS KATSARAGAKIS



# Outline

- Recommendation Systems (RS) Basics
- K Nearest Neighbors (KNN) Algorithm
- Exercise

---

# Recommendation Systems Basics

- Most internet products we use today are powered by RS
- Lots of internet products rely on RS to:
  - a) filter millions of contents
  - b) make personalized recommendations to users
- E.g.

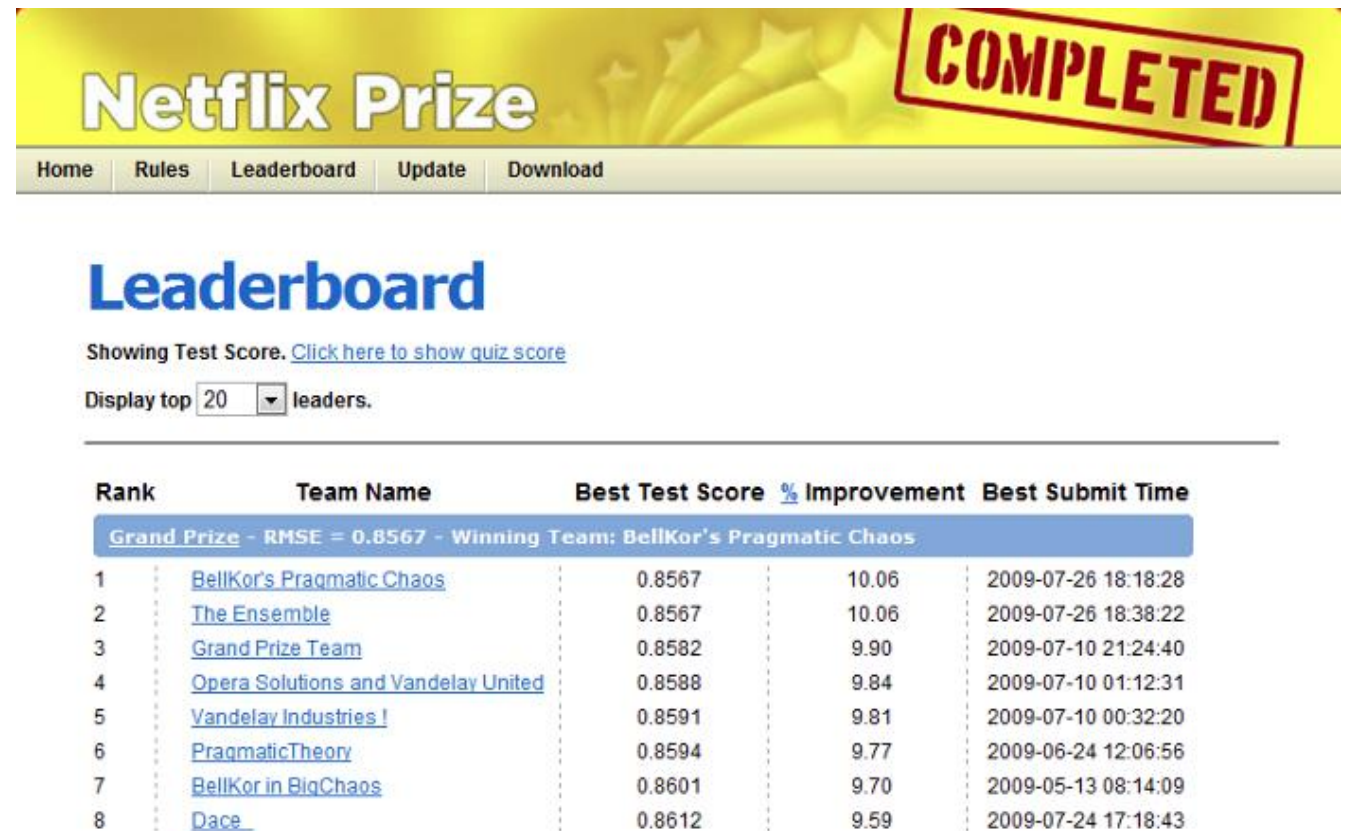


# More on Netflix...



## More on Netflix...

- Netflix awarded a **1 million \$ prize** to a developer team in 2009, for an algorithm that **increased the accuracy** of the company's recommendation system by **10%**



Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	<a href="#">BellKor's Pragmatic Chaos</a>	0.8567	10.06	2009-07-26 18:18:28
2	<a href="#">The Ensemble</a>	0.8567	10.06	2009-07-26 18:38:22
3	<a href="#">Grand Prize Team</a>	0.8582	9.90	2009-07-10 21:24:40
4	<a href="#">Opera Solutions and Vandelay United</a>	0.8588	9.84	2009-07-10 01:12:31
5	<a href="#">Vandelay Industries!</a>	0.8591	9.81	2009-07-10 00:32:20
6	<a href="#">PragmaticTheory</a>	0.8594	9.77	2009-06-24 12:06:56
7	<a href="#">BellKor in BigChaos</a>	0.8601	9.70	2009-05-13 08:14:09
8	<a href="#">Dace</a>	0.8612	9.59	2009-07-24 17:18:43

---

# Recommendation System Approaches

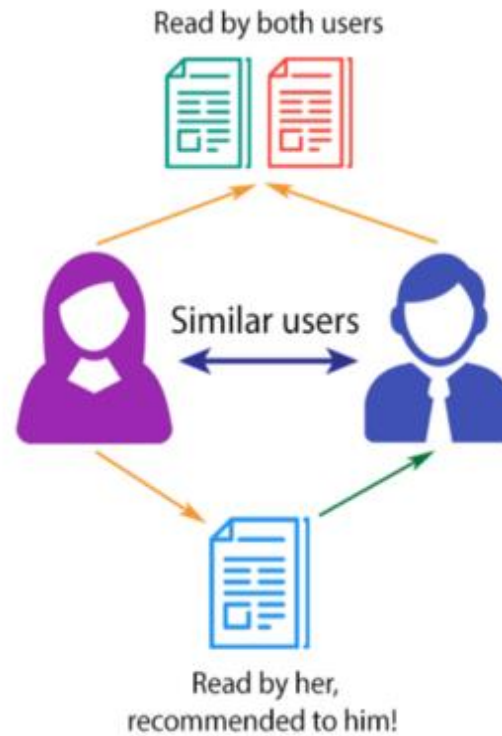
Content based Systems

Collaborative filtering Systems

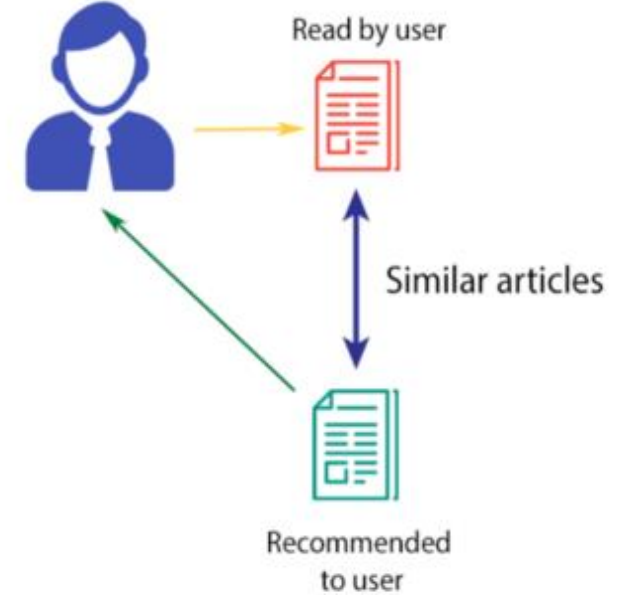
Hybrid Systems

# Content-based vs Collaborative Filtering

COLLABORATIVE FILTERING

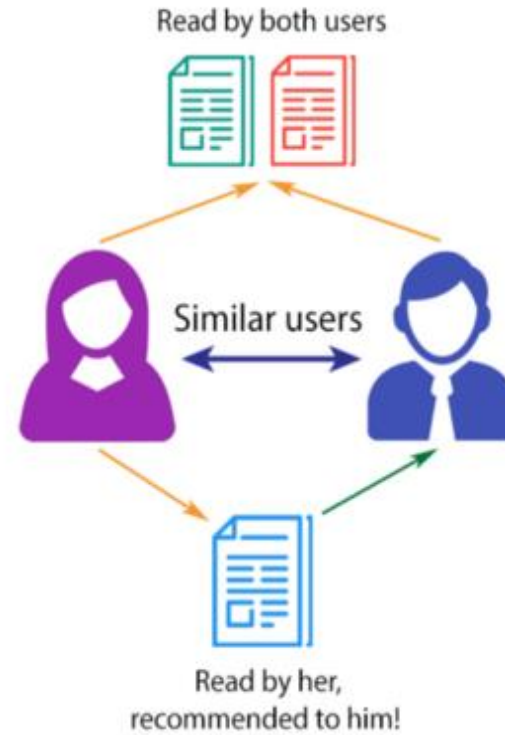


CONTENT-BASED FILTERING



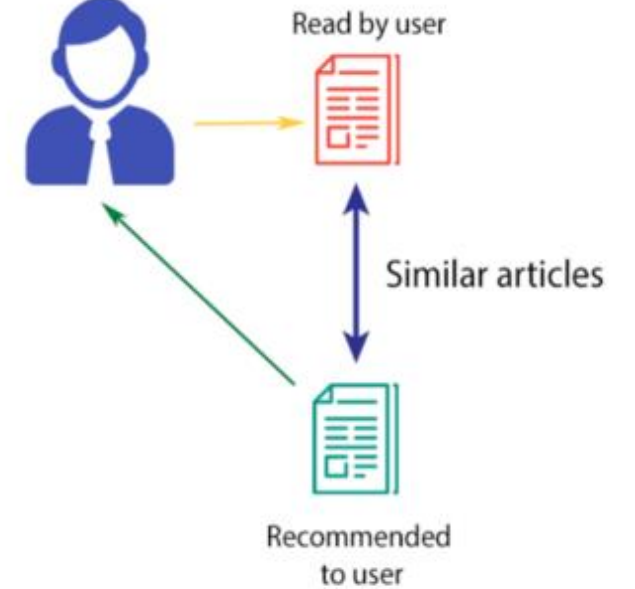
# Content-based vs Collaborative Filtering

## COLLABORATIVE FILTERING



User-based

## CONTENT-BASED FILTERING



Item-based



---

# Recommendation System Common Challenges

Cold start

Sparsity

Synonymy

Privacy

Scalability



# K Nearest Neighbors Algorithm

- Non-parametric, lazy learning method

# K Nearest Neighbors Algorithm

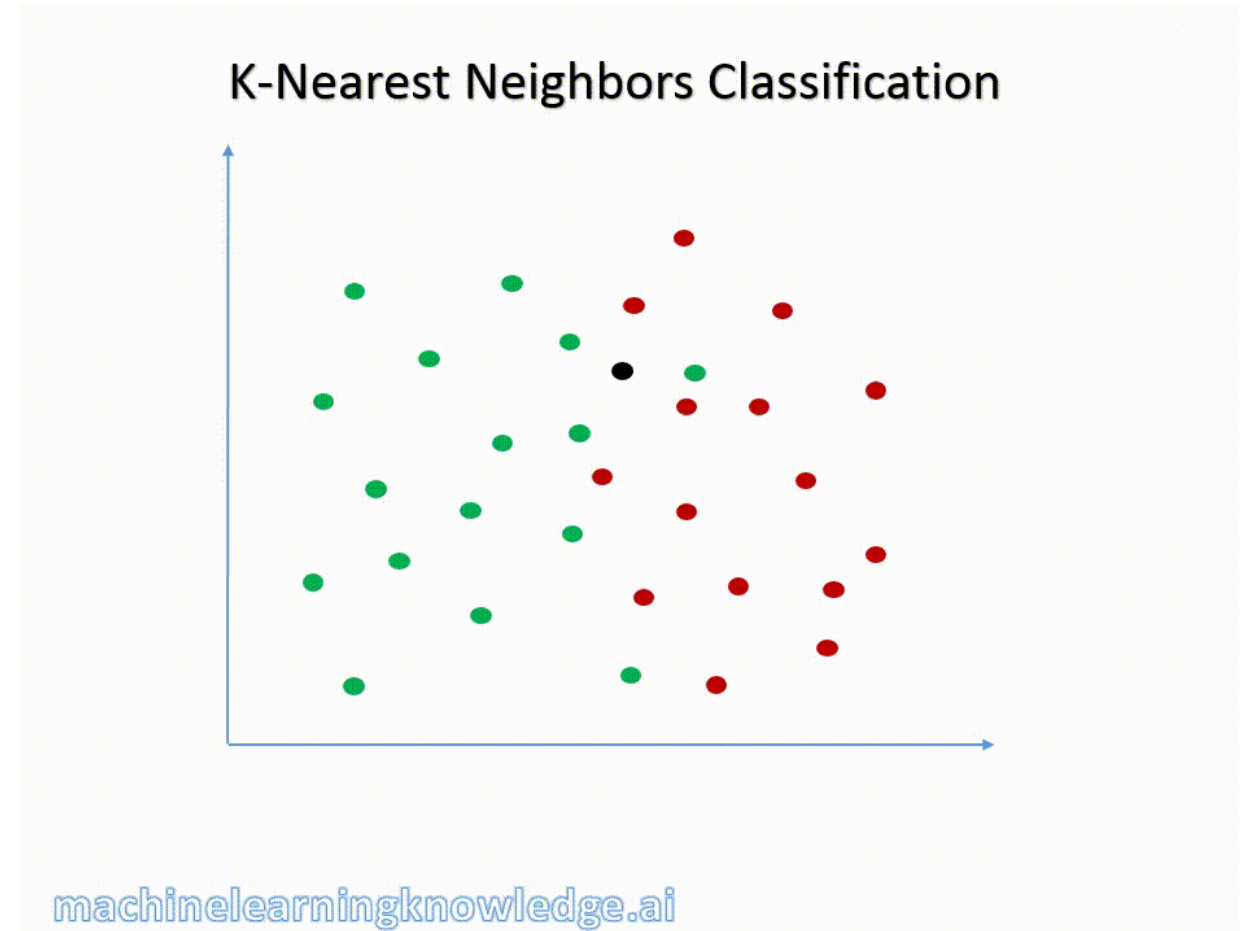
- Non-parametric, lazy learning method
- We can use different **distance metrics**

1. Euclidean  $\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$
2. Cosine  $\frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$
3. Minkowski  $D(X, Y) = \left( \sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}$
4. Manhattan  $d_1(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}\|_1 = \sum_{i=1}^n |p_i - q_i|$
5. Hamming  $D_H = \sum_{i=1}^k |x_i - y_i|$

$$x = y \Rightarrow D = 0$$

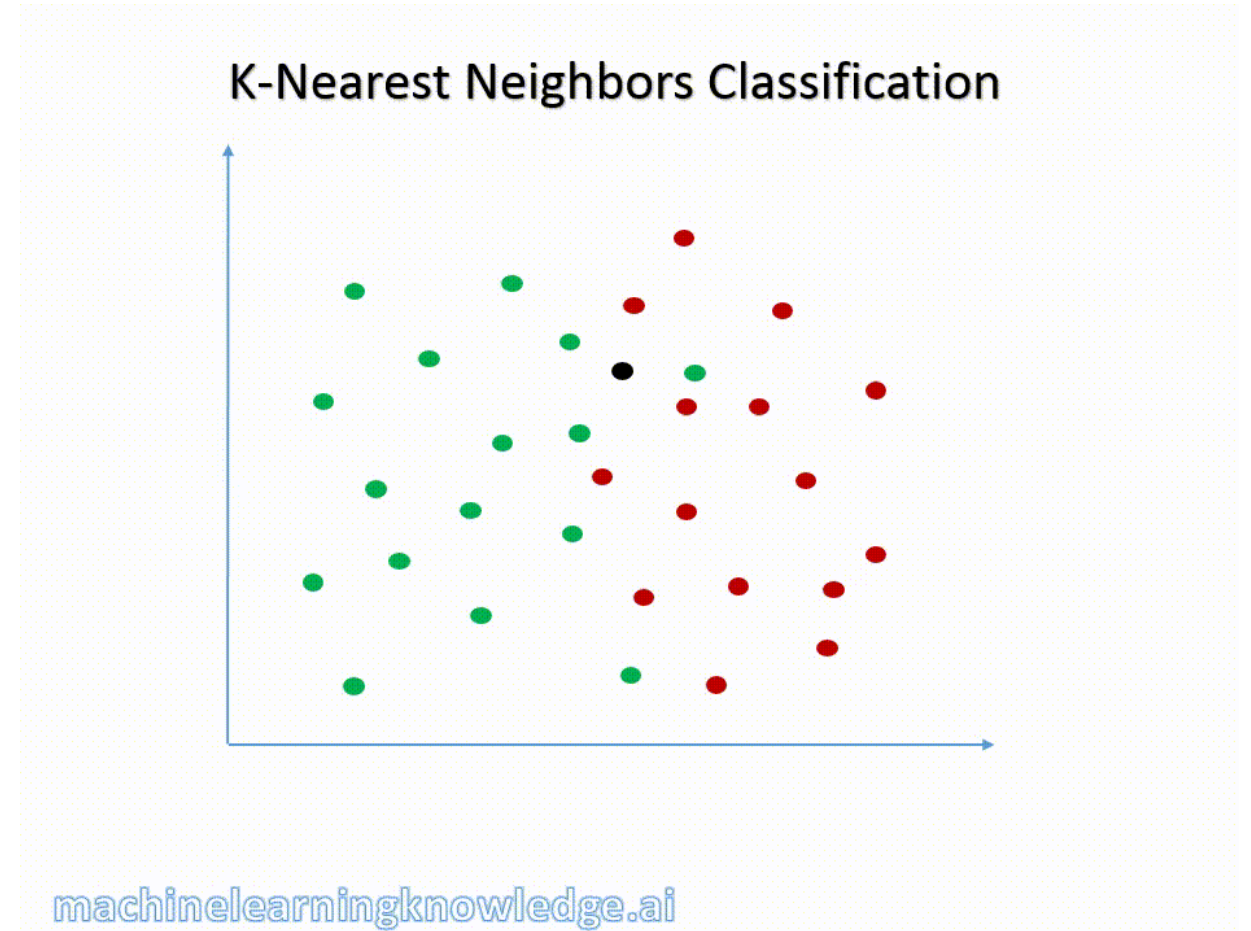
$$x \neq y \Rightarrow D = 1$$

# K Nearest Neighbors Algorithm Animation

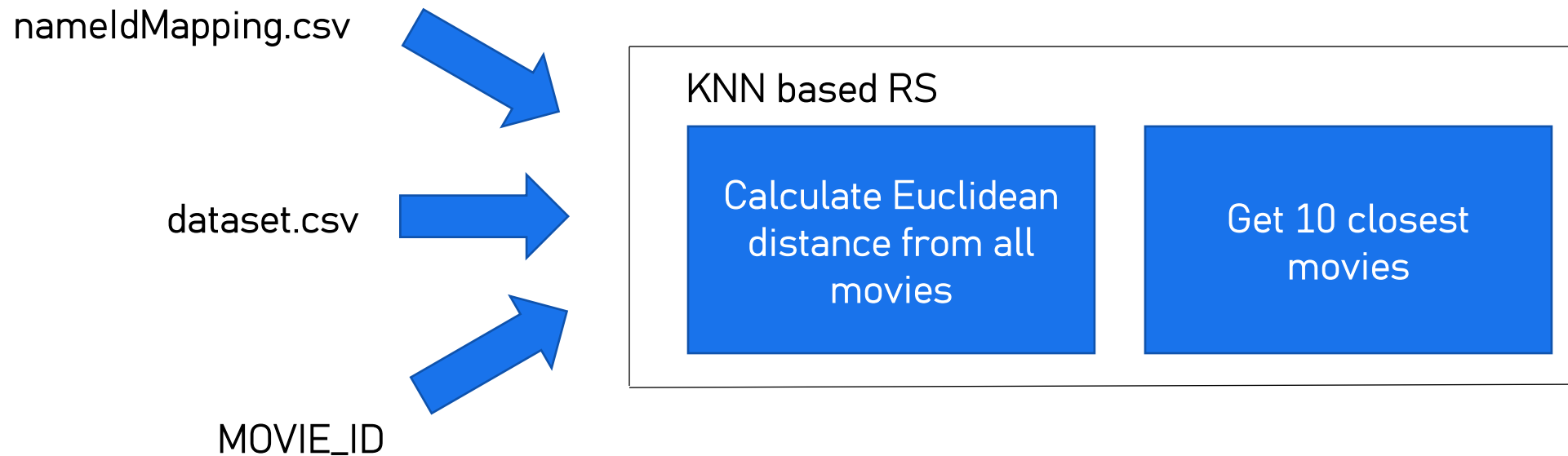


# K Nearest Neighbors Algorithm Animation

Calculating Euclidean Distances for all the data points is time consuming !



# KNN based Recommendation System Architecture





# Input Data

- **dataset.csv** is a subset of **Movie Lens** dataset
- For each movie it contains the rating for different users
- The rating is a value in  $\{0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0, 4.5, 5.0\}$
- **nameIdMapping.csv** contains movie id – movie name mapping



You are going to use HLS to accelerate the  
Euclidean Distances calculation...





# Useful Links

- [Recommendation Systems Basics](#)
- [K Nearest Neighbors Algorithm](#)
- [Exercise Github Repository](#)



