

## Evaluating Recommender Systems.

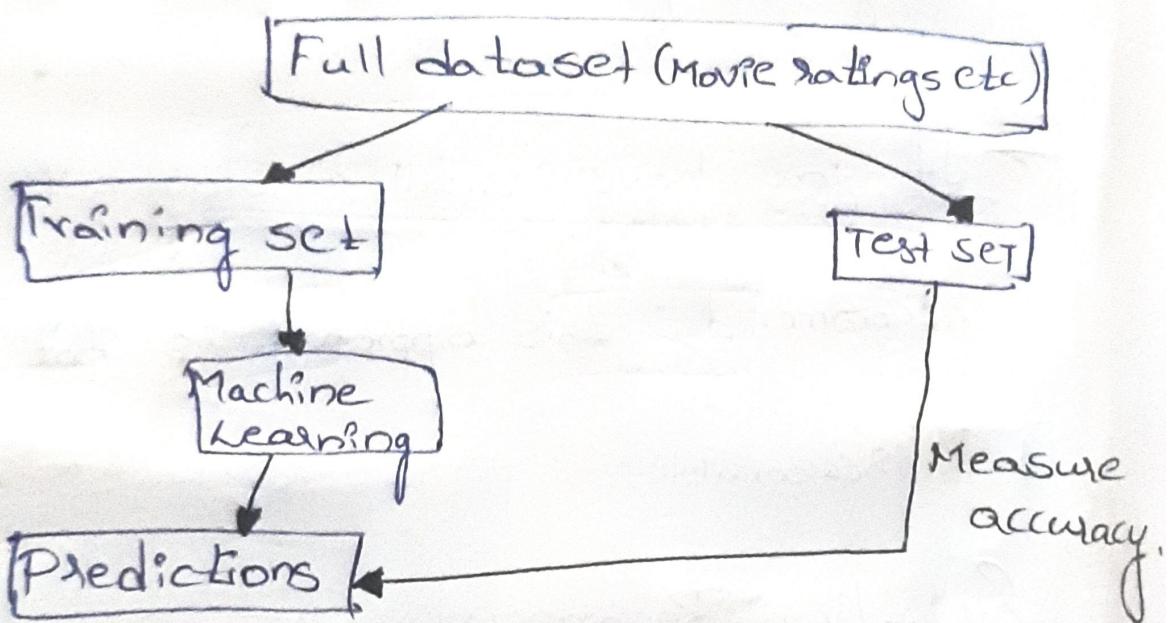
↳ Its difficult to measure how good they are. Because its different from one person to other.

↳ Its hard to say whether a person consider the recommendation ~~product~~ \* to be good or not.

↳ Especially, if we develop recommendation system offline. It's hard to people come up with a lot of different ways to measure the quality of a recommender system.

↳ And different measurements one can be at odds with each other.

### 1<sup>st</sup>) Train/ Test



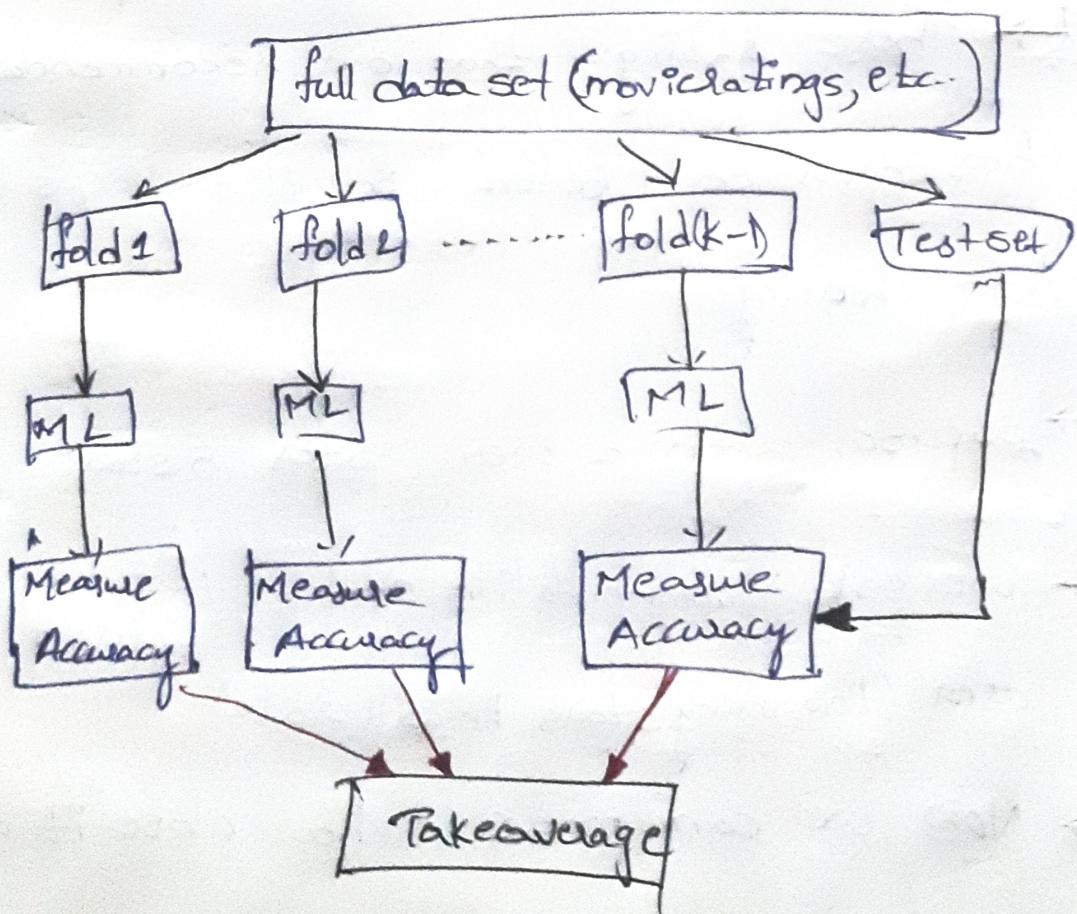
↳ Recommendation System is a Machine Learning system.

↳ Data is splitted into Training & Test sets  
80% 20%

- ↳ Model trains only with 80% of data (training)
- ↳ It learns relationship between Variables (items or users)
- ↳ Once the model is trained, we will predict
- ↳ So, when a new user might rate some item they have never seen before. So, to measure how well it does, we take the data we reserved for testing.
- ↳ These ratings that our recommendation has never seen before. So that keeps it from cheating.
- ↳ Suppose, when a user rates 5 star
  - ↳ we ask recommendation system to predict the user (How many stars, he will rate)
  - ↳ Now we can measure how close it came to the real rating.

- Suppose a new user won't know how much he rated to the recommendation system
- also, we will predict the o/p & we don't know how close to it, whether good/bad.
- To avoid this (to improve this) we use K-fold cross validation.

## 2<sup>nd</sup>) K-fold cross Validation



Note  $k$ -fold cross validation is using

Same idea as train/test, but instead of single training set, we create  $k-1$  training &  $1$  testing sets.

- ↳ Each fold will train the recommender system independently.
  - ↳ at last we will take average of accuracy.
- Disadvantages:-
- ↳ It takes more computing time

### Advantages

- ↳ It avoids over-fitting to a single training set