**Recommendation Systems using K-Nearest Neighbor**

**& Matrix factorization**

**K-Nearest Neighbor (KNN)**

**- Classification Algorithm**

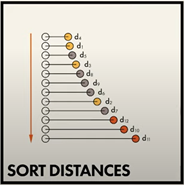
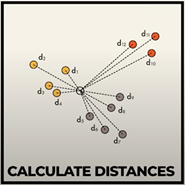
The idea of classification is to split existing data into two or more classes or labels, or determining the classes or labels for incoming data. Unlike the clustering, classification is a supervised learning technique, which means that we know in advance what labels or classes we have, and we need to know the relationship of incoming data to those classes or labels to determine to which one they belong. We use existing data for the process of training to build a model named Classifier, we pick significant attributes of these data to train that model, and when it is ready we give it the significant attributes of the new data which have unknown classes so the classifier can predict them.

**-KNN Introduction**

K-Nearest Neighbors is a supervised algorithm in machine learning which used to make classifications, predictions and regression tasks. It is widely used due to its easy implementation.

**-Working Principle**

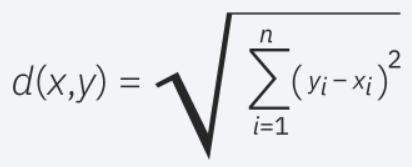
KNN basic assumption is that similar points can be found near one another. Which means if an object is similar to its neighbors, it is likely to belong to the same class and share similar characteristic. The general steps in KNN Algorithm are choosing the value of K, by determine the number of neighbors K values. Calculating the distance between target data points and other points in the set. Select K neighbors with shortest distance to the target point.



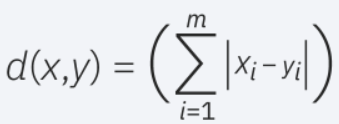
**-Distance determining**

There are many formulas for determining the distance between the new point and each training point**.** Those are three formulas that are used for measuring the distance:

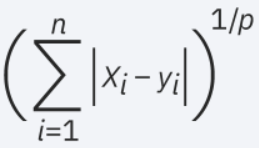
**Euclidean distance**: measures a straight line between the query point and the other point being measured.



**Manhattan distance**: measures the absolute value between two points widely used in city streets.



**Minkowski distance**: generalized form of Euclidean and Manhattan distance metrics



**-What is the best value of K?**

the best approach is to train the system with many K values, starting from low value and going to high values, the accuracy will get better and better, and at some K value, it will start falling down, that would be our K value for the system.

**-Advantage**

KNN algorithm is easy to implement due to its simplicity and accuracy. In addition to its few parameters.

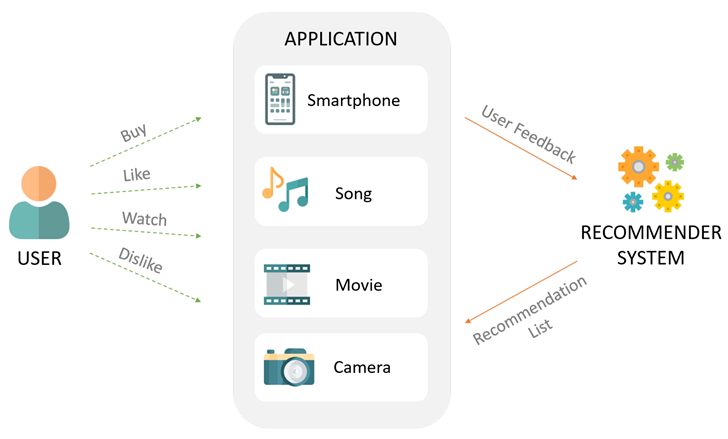
**-Disadvantage**

KNN is a lazy algorithm, it takes up more memory and data storage compared to other classification algorithms. It is not practical for imbalanced datasets. Struggles when dealing with high-dimensional data.

**Recommendation System**

A recommendation system is an artificial intelligence or AI algorithm, usually associated with machine learning, that uses Big Data to suggest or recommend additional products to consumers. These can be based on various criteria, including past purchases, search history, demographic information, and other.

It is also an information system which takes data input and processes it to give the recommended output for the values entered in the system.



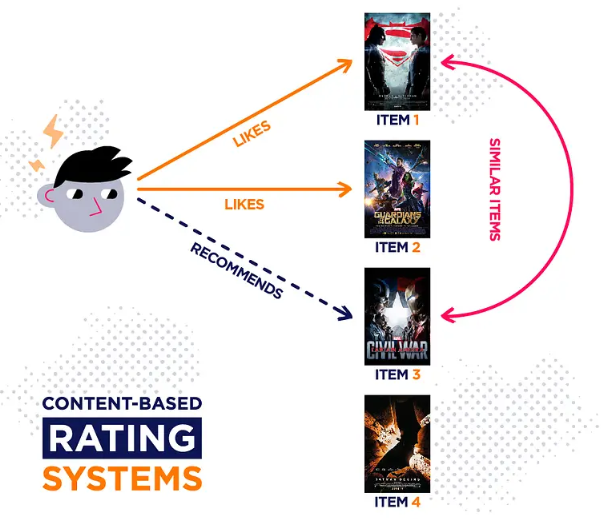
**-What data input needed ?**

Recommendation system can depend on many data such as the user history, user input or many other data depending on the system directly.

**-Types of Recommendation System**

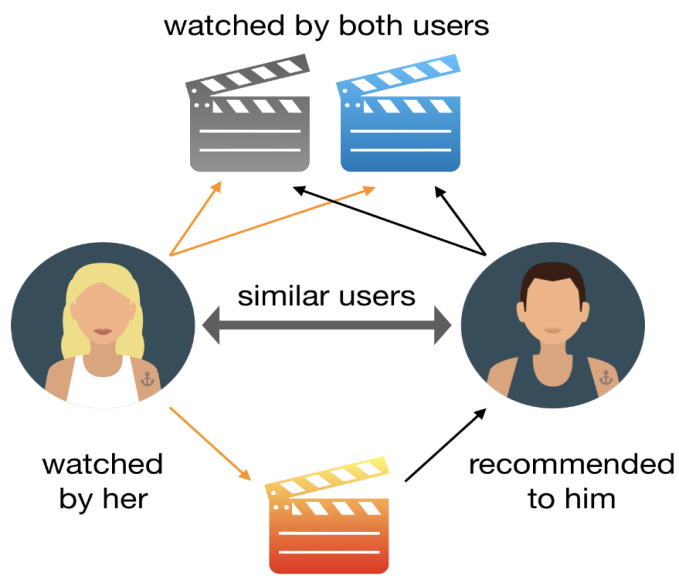
**1)Content Based Recommendation system.**

The system here works on the similar content of the input data , in a movie website if the user liked some movies with genres the recommendation system will recommend movies with similar genres and plots , similar to all the ideas that content based is applied. In KNN for example For a given user, KNN identifies the K-nearest neighbors (users with similar interests) using the calculated similarity.



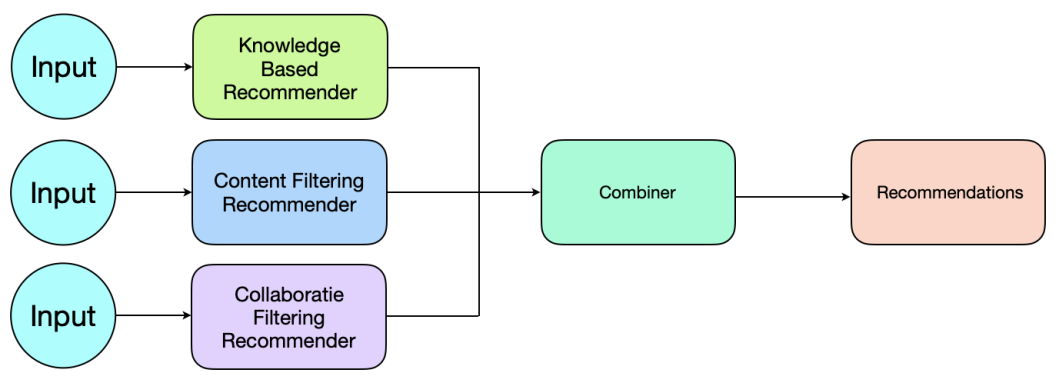
**2) Collaborative based Recommendation system.**

he system works on the similarity between different users , in a movie website if the user liked some movies, the recommendation system will recommend movies that are liked by users who have the same taste with the user, similar to all ideas that collaborative filtering based applied.



**-Hybrid based Recommendation system**

The system works on both content based and collaborative filtering recommendation systems, as both are defined in the system separately and then results of both are combined to give the output, this is the most recommendation system method used.



**-Recommendation System methodology**

Recommendation system contains underneath many steps to get the results as mentioned above as any system contains input , processing and output recommendation systems have these basic steps but all the work is inside the processing part

**-Preprocessing**

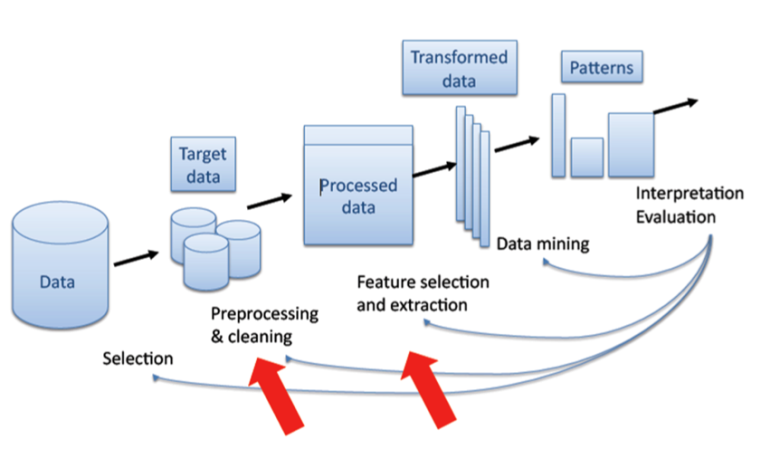
Data is taken from APIs and Databases are relatively not clean for training and needs to be reorganized and cleaned , for example : some data as genres may be split into binary arrays with 1,0 values.

**-Training Stage**

After preprocessing and cleaning the data we need to start train the system on the data we have so it starts to predict well, using a classifier technique such as KNN , Decision Tree, SVM(support vector machine) , the system may need to be trained many times so it can predict well as the error will be stopped on a specific point.

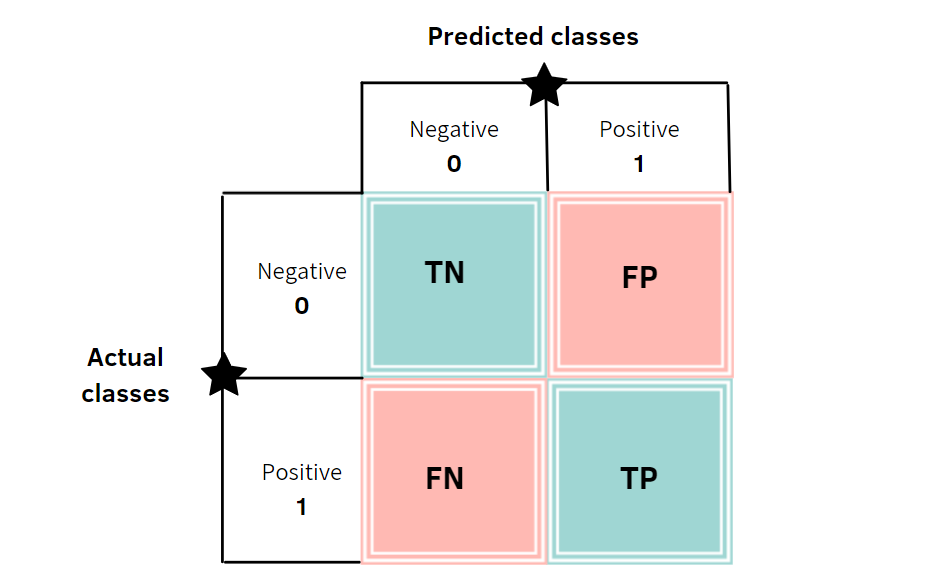
**-Testing Stage**

The data is split into training set and testing set , the testing set data is entered to the system to check if the values are correctly predicted and to calculate error percentage.



**-Confusion Matrices**

Recommendation systems are evaluated using many methodologies : Sum squared error , Mean Squared Error , confusion Matrices or F1 Score, Most used one is the Confusion Matrices to detect the number of error cases and the number of correct predictions in each case.



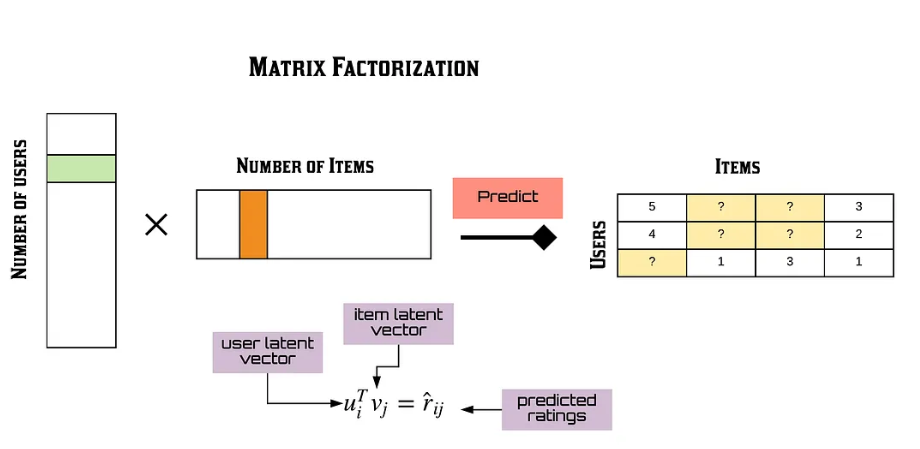
**-KNN Implementation**

KNN algorithm is divided into several parts starting by loading the data then initialize the K value of random for the first time. Looping iteratively on the data and calculate the distance between the input data and the query example and add the resulted distance and it’s index to a Collection and then sort this collection Descending order. Then we get the first K elements of this collection then return the classification of this mode.

**Matrix factorization**

Matrix factorization is a way to generate latent features when multiplying two different kinds of entities. Collaborative filtering is the application of matrix factorization to identify the relationship between items and users entities.

The basic idea is to decompose a user-item interaction matrix into two lower-dimensional matrices, typically referred to as the user matrix and the item matrix. This decomposition helps capture latent factors that represent hidden patterns or features in the data.

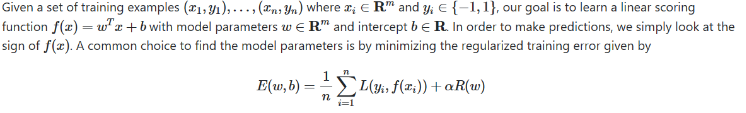


Popular algorithms for matrix factorization in recommendation systems include Alternating Least Squares (ALS), and Stochastic Gradient Descent (SGD) based methods.

Alternating Least Squares (ALS) algorithm approximates the sparse user item rating matrix u-by-i as the product of two dense matrices, user and item factor matrices of size u × f and f × i (where u is the number of users, i the number of items and f the number of latent features) . The factor matrices represent latent or hidden features which the algorithm tries to discover. One matrix tries to describe the latent or hidden features of each user, and one tries to describe latent properties of each movie. For each user and for each item, the ALS algorithm iteratively learns (f) numeric “factors” that represent the user or item. In each iteration, the algorithm alternatively fixes one factor matrix and optimizes for the other, and this process continues until it converges.

Gradient Descent: It is a very popular optimization technique in Machine Learning and Deep Learning and it can be used with most, if not all, of the learning algorithms. A gradient is basically the slope of a function; the degree of change of a parameter with the amount of change in another parameter.

Mathematically, it can be described as the partial derivatives of a set of parameters with respect to its inputs. The more the gradient, the steeper the slope. Gradient Descent is a convex function.



**-Matrix factorization in recommendation system**

factors. Recommender systems are highly useful as they help users discover products and services they might otherwise have not found on their own.

This leads to increasing sales ,helping to form customer habits and trends, speeding up pace of work and so on.

**-Real-world Applications:**

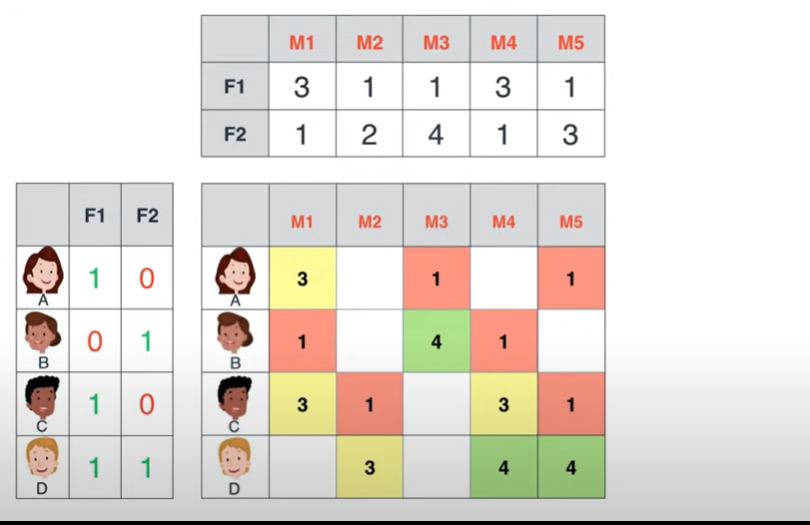
**How we can use matrix factorization algorithm in recommendation systems?**

As I have mentioned above, from an application point of view, matrix factorization can be used to discover latent features underlying the interactions between two different kinds of entities. (or more) And one obvious application is to predict ratings in collaborative filtering.

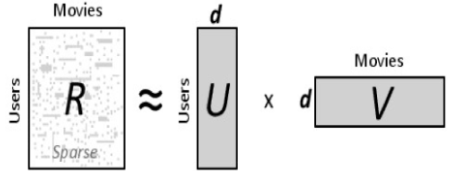
**Why Netflix use recommendation system?**

The company needs to make sure they are promoting videos with a strong likelihood of being viewed. That 80% success rate keeps subscribers happy. Essentially, Netflix believes it could lose $1 billion or more every year from subscribers quitting its service if it didn’t use the Netflix Recommendation Engine. Still it makes sense, personalization is everything in today’s consumer world. Every digital platform utilizes it somehow, from Spotify to Amazon.

In a recommendation system such as Netflix or MovieLens, there is a group of users and a set of items Given that each users have rated some items in the system, we would like to predict how the users would rate the items that they have not yet rated, such that we can make recommendations to the users. In this case, all the information we have about the existing ratings can be represented in a matrix. Assume now we have 4 users ,5 movies and two features, and ratings are integers ranging from 1 to 5, the matrix may look something like this (zero means the user don not like this feature 1 means he/she like it):



As previous picture there is missing ratings in user-movie table, in this case matrix factorization algorithm can be very useful, using matrix factorization to predict the missing rating depending on mapping between user-feature table and movie-feature table using this mathematical relation:





To minimize the difference between *Rij*​ (original) and the approximated ratings (*R*^)*ij*​ This is typically done by minimizing a loss function, often based on the squared error:



When using matrix factorization algorithm to make implicit feedback and

user-specific model, which build recommendation system which all e-commerce company needed to enhanced user experience (improve browsing and encourages exploration) , increase customer satisfaction, generates an insightful report about customer behavior and enhance application optimization ,which leads to increase sales and company’s profit.

**Conclusion**

KNN and matrix factorization both are part of many ai algorithms that have entered new digital life era applications. It evolves all aspects providing better experience in business, education, social media and many other

**Collaboration between:**

**Dima Eid 12029305**

**&&**

**Sema Hodali 12029448**