**Reinforcement learning - RL**

In reinforcement learning, developers devise a method of rewarding desired behaviors and punishing negative behaviors. This method assigns positive values to the desired actions to encourage the agent and negative values to undesired behaviors. This programs the agent to seek long-term and maximum overall reward to achieve an optimal solution.

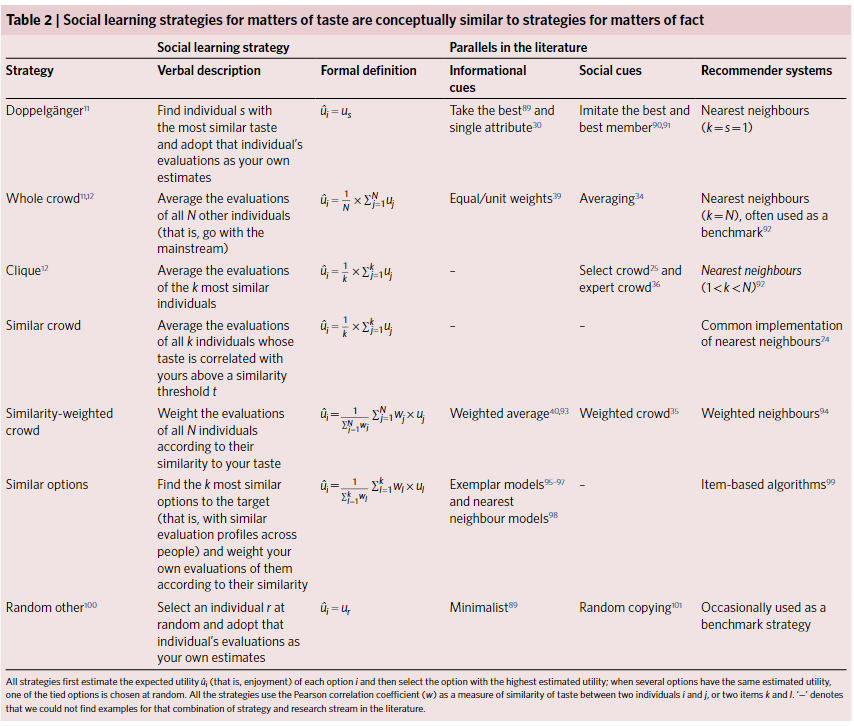
**Optimal stopping**

Reject the first 37%, then choose the best option you see after that. For example: choosing the best apartment - If you want the best odds of getting the best apartment, spend 37% of your apartment hunt (eleven days, if you’ve given yourself a month for the search) noncommittally exploring options. Leave the checkbook at home; you’re just calibrating. But after that point, be prepared to immediately commit — deposit and all — to the very first place you see that beats whatever you’ve already seen.

**Wisdom of crowds**

Wisdom of crowds is the idea that large groups of people are collectively smarter than individual experts when it comes to problem-solving, decision-making, innovating, and predicting. The idea is that the viewpoint of an individual can inherently be biased, whereas taking the average knowledge of a crowd can result in eliminating the bias or noise to produce a clearer and more coherent result.

**Recommender systems - Social learning**



**Content-based recommender systems**

Such systems are recommending items similar to those a given user has liked in the past, regardless of the preferences of other users. Basically, there are two different types of feedback.

* Explicit feedback is intentionally provided by users in form of clicking the “like”/”dislike” buttons, rating an item by number of stars, etc. In many cases, it is hard to obtain explicit feedback data, simply because the users are not willing to provide it. Instead of clicking “dislike” for an item which the user does not consider interesting, he/she will rather leave the web page or switch to another TV channel.
* Implicit feedback data, such as “user viewed an item”, “user finished reading the article” or “user ordered a product”, however, are often much easier to collect and can also help us to compute good recommendations.

**Popularity-Based Recommendation System**

It is a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly recommend those.

For example, if a product is often purchased by most people then the system will get to know that that product is most popular so for every new user who just signed it, the system will recommend that product to that user also and chances becomes high that the new user will also purchase that.

**Collaborative Filtering Systems**

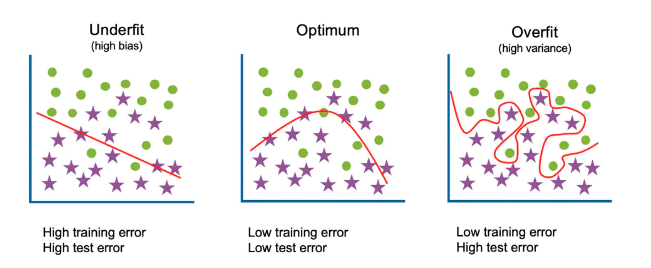
Collaborative filtering methods for recommender systems are methods that are solely based on the past interactions between users and the target items. Thus, the input to a collaborative filtering system will be all historical data of user interactions with target items. This data is typically stored in a matrix where the rows are the users, and the columns are the items. The core idea behind such systems is that the historical data of the users should be enough to make a prediction. I.e we don’t need anything more than that historical data, no extra push from the user, no presently trending information, etc

**Bias / variance**

**Overfitting and underfitting**

**Overfitting** is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose.

**Underfitting** is a scenario in data science where a data model is unable to capture the relationship between the input and output variables accurately, generating a high error rate on both the training set and unseen data.

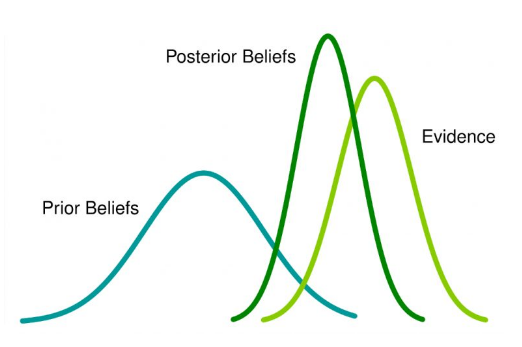


**Pagerank**

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites. An important webpage is hereby not only defined by the number of other pages pointing towards it, its importance is also heavily defined by the importance of the other webpages pointing towards it.

**Bayesian thinking**

Bayesian thinking is a branch of logic applied to decision making and inferential statistics that deals with probability inference: using the knowledge of prior events to predict future events.



**Ensemble learning**

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. Ensemble methods usually produces more accurate solutions than a single model would. There are two main reasons to use an ensemble over a single model, and they are related; they are:

* Performance: An ensemble can make better predictions and achieve better performance than any single contributing model.
* Robustness: An ensemble reduces the spread or dispersion of the predictions and model performance.

Ensembles are used to achieve better predictive performance on a predictive modeling problem than a single predictive model. The way this is achieved can be understood as the model reducing the variance component of the prediction error by adding bias

**Exploration - Exploitation (RL)**

Exploitation and exploration are two possible behaviours when facing a decision making problem that both have pros and cons. On one hand, exploitation consists of taking the decision assumed to be optimal with respect to the data observed so far. This « safe » approach tries to avoid bad decisions as much as possible but also prevents from discovering potential better decisions. On the other hand, exploration consists of not taking the decision that seems to be optimal, betting on the fact that observed data are not sufficient to truly identify the best option. This more « risky » approach can sometimes lead to poor decisions but also makes it possible to discover better ones, if there exists any

**Epsilon-greedy**

Epsilon-Greedy is a simple method to balance exploration and exploitation by choosing between exploration and exploitation randomly. The epsilon-greedy, where epsilon refers to the probability of choosing to explore, exploits most of the time with a small chance of exploring. The algorithm continuously balances exploration with exploitation. (In ‘greedy’ experiments, the lever with highest known payout is always pulled except when a random action is taken). A randomly chosen arm is pulled a fraction ε of the time. The other 1-ε of the time, the arm with highest known payout is pulled.

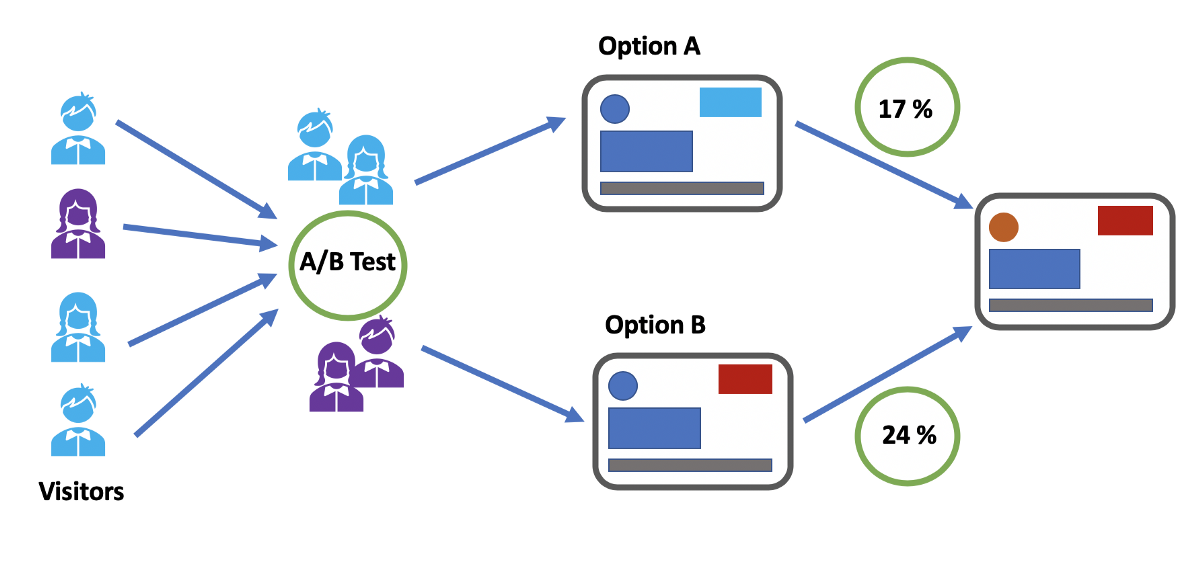
**Multi-armed bandit**

Multi-armed bandit problem is a simple RL problem. At every time step, the agent can choose one of K actions. The agent then receives a reward that is drawn from an unknown (to the agent) probability distribution corresponding to the said action. The goal of the agent is to choose actions such that the total reward received within a certain number of time steps is maximized.

**Win-stay, lose-shift**

in discrimination learning, a mental or behavioral strategy in which an organism continues to give the same response as long as it is being rewarded for doing so but changes the response once it is no longer being rewarded.

**A/B testing**



**Hill Climbing - Randomness**

**Simulated annealing - Stochastic hill climbing**  
Implementation of SA is surprisingly simple. The algorithm is basically hill-climbing except instead of picking the best move, it picks a random move. If the selected move improves the solution, then it is always accepted. Otherwise, the algorithm makes the move anyway with some probability less than 1.

**Steepest ascent hill climbing**

The steepest-Ascent algorithm is a variation of the simple hill-climbing algorithm. This algorithm examines all the neighbouring nodes of the current state and selects one neighbour node which is closest to the goal state. It can only go upwards, and it keeps climbing until it finds a maximum and can not improve. That means if it finds a local maximum, it will stay there, as it can not move downwards. Not ideal if there are multiple local maximums and a global maximum.

**Relaxation**

**Constraint Relaxation**

In this technique, researchers remove some of the problem’s constraints and set about solving the problem they wish they had. Then, after they’ve made a certain amount of headway, they try to add the constraints back in. That is, they make the problem temporarily easier to handle before bringing it back to reality, i.e, you solve the problem you wish you had instead of the one you actually have, and then you see how much this helped you.

**Continuous Relaxation**

Turns discrete or binary choices into continua: when deciding between iced tea

and lemonade, first imagine a 50–50 “Arnold Palmer” blend and then round

it up or down. We could, for example, choose to simply round them as

necessary, sending invitations to everyone who got “half an invitation” or

more in the relaxed scenario. We could also interpret these fractions as

probabilities—for instance, flipping a coin for every location where the

relaxed solution tells us to put half a fire truck, and actually placing a truck

there only if it lands heads.

**Sorting**

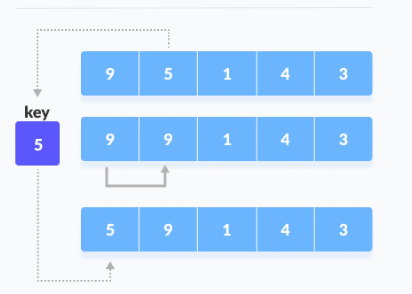
**bubble sort**  
Bubble sort is a basic algorithm for arranging a string of numbers or other elements in the correct order. The method works by examining each set of adjacent elements in the string, from left to right, switching their positions if they are out of order. The algorithm then repeats this process until it can run through the entire string and find no two elements that need to be swapped.

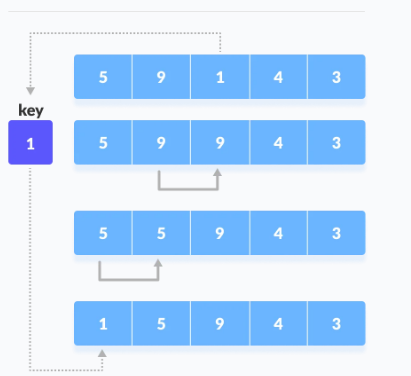


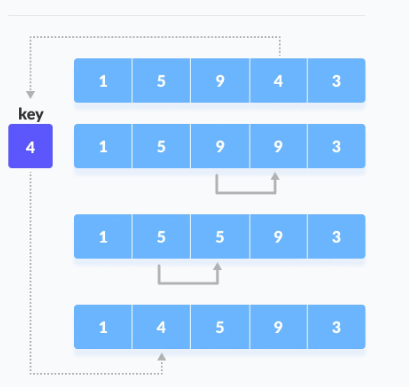
**Insertion sort**

Insertion sort works similarly as we sort cards in our hand in a card game.

We assume that the first card is already sorted then, we select an unsorted card. If the unsorted card is greater than the card in hand, it is placed on the right otherwise, to the left. In the same way, other unsorted cards are taken and put in their right place.





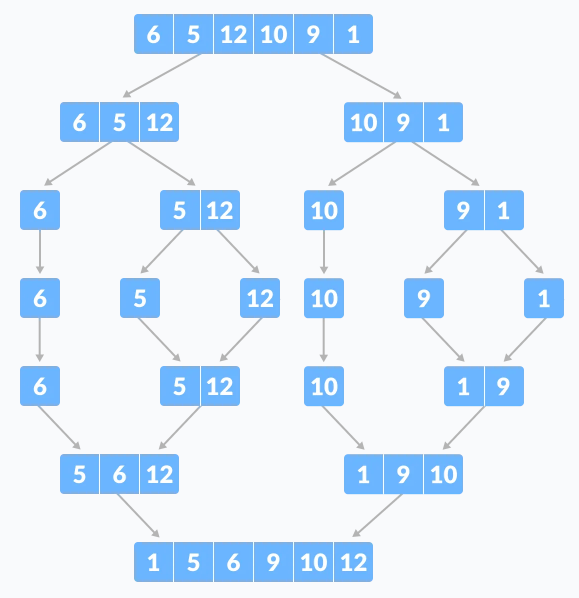


**Difference between bubble and insertion**

The main difference between bubble sort and insertion sort is that bubble sort performs sorting by checking the neighboring data elements and swapping them if they are in wrong order while insertion sort performs sorting by transferring one element to a partially sorted array at a time.

**Mergesort**

Merge Sort is a divide and conquer algorithm. It works by recursively breaking down a problem into two or more sub-problems of the same or related type, until these become simple enough to be solved directly. The solutions to the sub-problems are then combined to give a solution to the original problem. So Merge Sort first divides the array into equal halves and then combines them in a sorted manner.



**Bucket sort**

Bucket Sort is a sorting algorithm that divides the unsorted array elements into several groups called buckets. Each bucket is then sorted by using any of the suitable sorting algorithms or recursively applying the same bucket algorithm.



**Game Theory**

**Nash Equilibrium**

a concept of game theory where the optimal outcome of a game is one where no player has an incentive to deviate from their chosen strategy after considering an opponent's choice.Overall, an individual can receive no incremental benefit from changing actions, assuming other players remain constant in their strategies. A game may have multiple Nash equilibria or none at all. Cooperate & defect, prisoner’s dilemma example