***Question :***

Detail Case study on Pinterest recommendation system? How it works, If you wanted to clone it how will you do it. Mention the process. Which recommendation system do you like and why?

***Answer :***

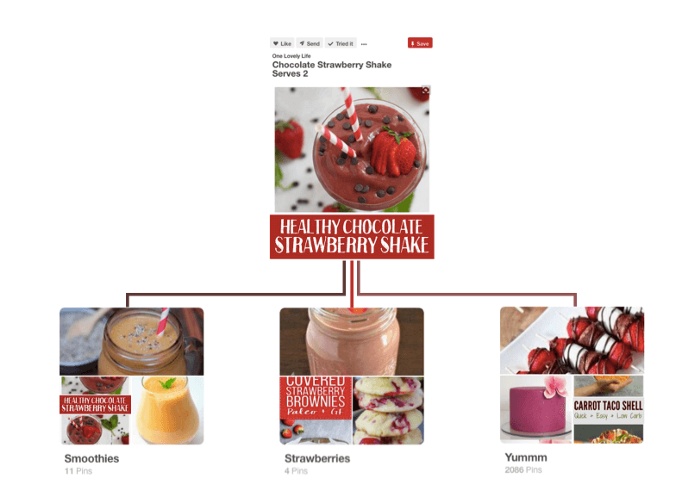
**What is pinterest** : It is a website which allows us to organize the data in a properly and sorted manner. It shows/recommends the related posts from across the internet. User need to create an account on the website.

In Pinterest – Pins and boards are important things.

Pin is nothing but a post which are created by multiple users.

Board is nothing but a collection of similar kind of pins which are created by users.

While saving any pin, user has to save it in related board. It completely depends on user how he/she wants to organize these saved pins.



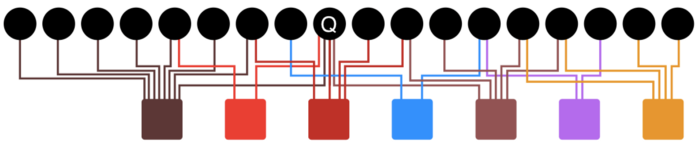
Healthy Chocolate Strawberry Shake is a pin which I can save it in smoothies board, somebody else can save it in board of strawberries or Yummm.

**Problem Statement:**

For a query pin, the system should show relevant pin recommendations.

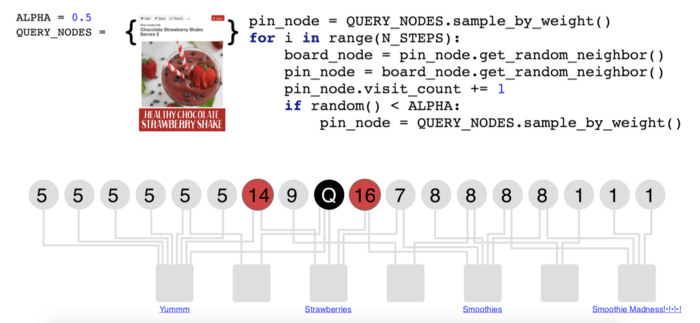
**Pixie: Pinterest Recommendation Algorithm**

Here they have used Graph based algorithm. All the pins and the boards are connected to each other and forms bipartite graph.



When a user saves or click on particular pin to view, system considers it as a query pin(Q). Pixie uses Random Walk algorithm internally.

**Random Walk Algorithm**: In this algorithm we have to traverse this bipartite graph starting from query point till it completes some count of walks. In Pixie it traverses 100K walks. To complete this random walk it traverses by randomly choosing nearest neighbors.



**Algorithm :**

**------------------------------------------------------------------------------------------------------------------------------------------**

For 100K times{

Board = Randomly select a nearest board

Pin = Randomly select a nearest neighbor pin of the selected board

Keep a visit count and increase it every time the pin gets visited in the random walk

If Random num < 0.5 (its more like a toss) then :

start the same process again from the query pin

else :

continue walking with the same randomly selected pin

}

Once all the 100K steps completed, pins which has the highest visit count are more related to the query pin.

**------------------------------------------------------------------------------------------------------------------------------------------**

There are 100B+ edges in this graph. To reduce the response time they have used few optimization techniques.

**Early Stopping:** Ideally, only top 1000 related pins should get retrieved. So we don’t need a walk to complete 100K steps. The algorithm can be stopped when 1000 pins completes the visit count of 20.

**Graph Pruning:** Here, they have implemented a function which provides cap for the number of neighbors a pin has. This downscales the effect of popular pins.

Pixie is very unique and easy to use recommender system algorithm which provides radical personalization in **real time**.

To clone this recommender system, first we need to have a data as number of pins and its corresponding board names. We will get it by scraping the website.

Once we have the data in pin, board format. We can then convert it into bipartite graph and apply the same algorithm on top of it to fetch the related pins.

There are multiple types of recommender systems. Out of which I do like Hybrid Recommender system, where recommendations from more than one technique are presented together, so the user can choose from a wide range of recommendations. Here we can have recommendations based on user-user similarity as well as item-item recommendations, SVD recommendations and in addition to this if we have an algorithm like Pixie we can show results of that too. Because of this we will have wide range of recommendations which covers all the perspectives of recommender system.