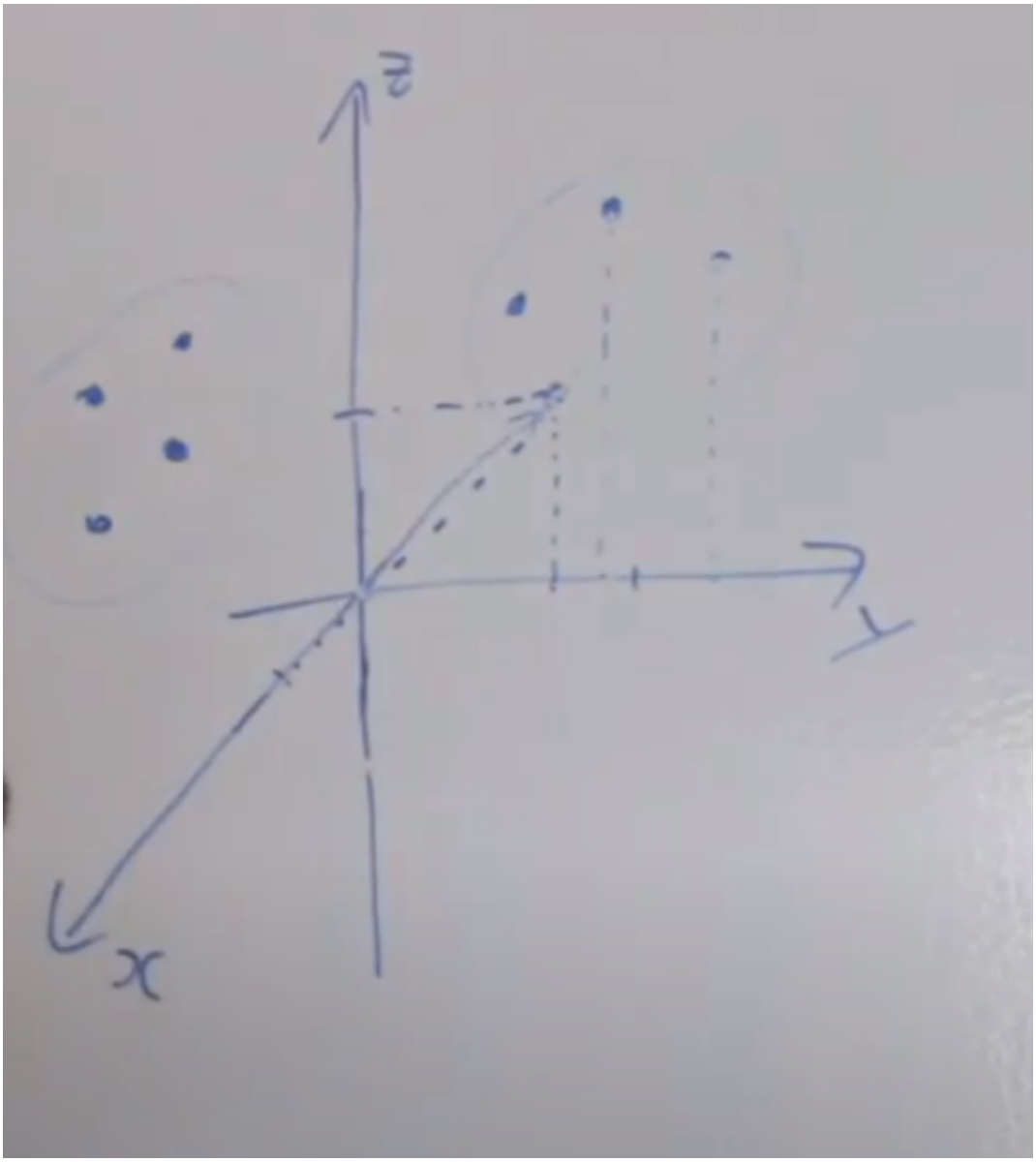
Clustering K Means

O objetivo desse algoritmo é dividir os dados em clusters ou grupos, onde é necessário dizer quantos clusters queremos que o algoritmo separe.

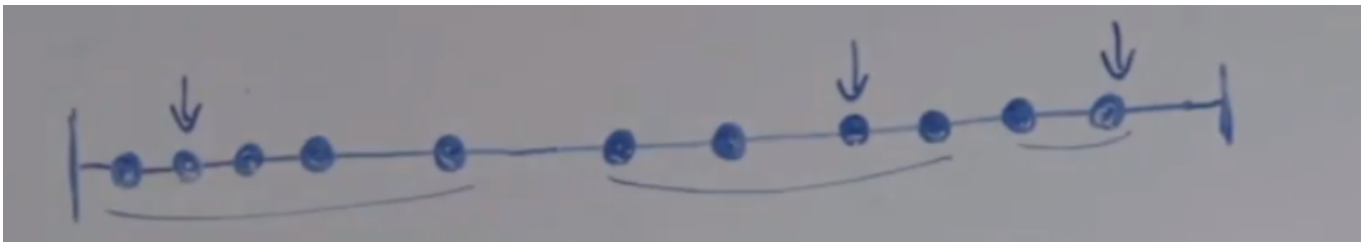
Exemplo em 3 dimensões:



Nesse caso, podemos informar para o algoritmo que queremos 2 clusters, e ele irá calcular a distância entre os pontos para definir se um ponto pertence ao cluster 1 ou 2.

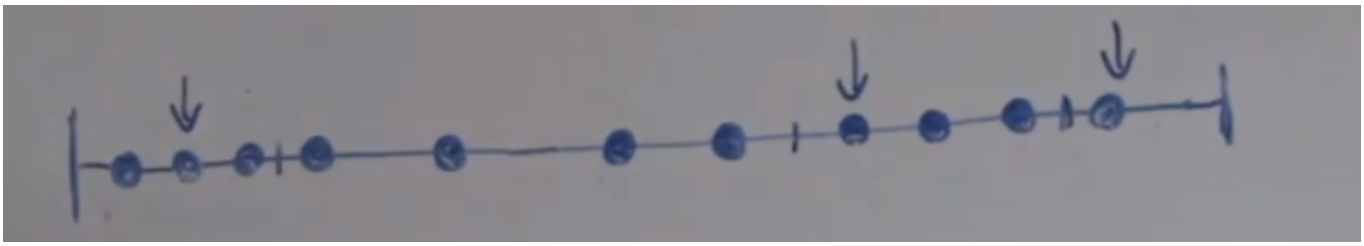
Como é calculado:

Basicamente podemos utilizar uma linha reta com apenas uma variável para exemplificar a separação dos clusters, como mostrado abaixo:

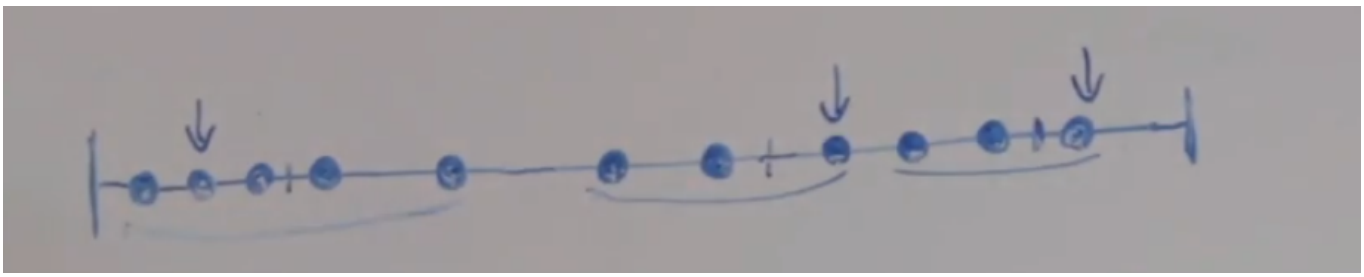


Primeiramente o algoritmo irá sortear 3 números caso o número de clusters escolhido seja 3 e agrupar para cada amostra qual está mais próxima desses 3 pontos.

Após isso é calculado o centro de cada cluster, ou centróide para que seja a nova referência para separação dos grupos.

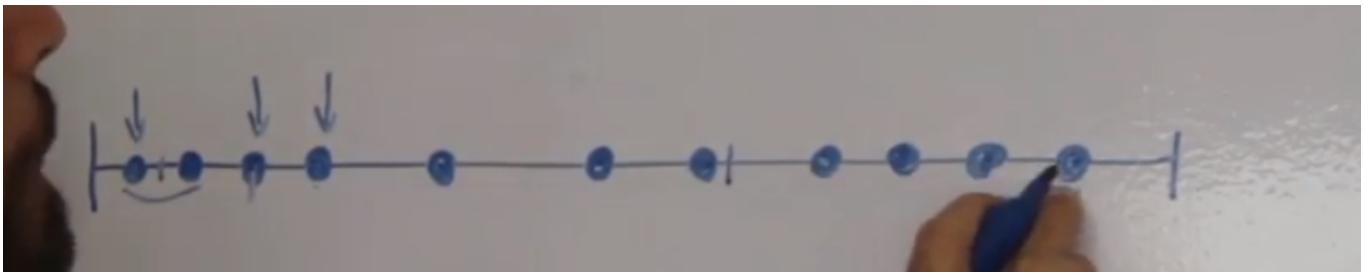


Cada tracinho é um centróide, e é feito o cálculo de distância de cada amostra para cada centróide e verifica qual a menor distância de cada um aquela amostra específica está.



Após a definição do primeiro centroide, podemos fazer outro cálculo dos novos clusters e verificar se houveram mudanças nos grupos. Caso não tenha ocorrido nenhuma mudança, o algoritmo sabe que é hora de parar de separar. Isso ocorre na observação dos centroides, caso eles mudem muito de lugar, ainda não estão separados adequadamente. Esse parâmetro de alteração é chamado de tolerância, caso a mudança em relação as iterações anteriores estejam abaixo da tolerância, não é necessário realizar mais iterações, pois o modelo ainda não está estável.

Um outro exemplo que pode ocorrer durante o sorteio das amostras iniciais é quando são amostras perto uma das outras, como mostrado:



As 2 primeiras serão um cluster, a segunda já é uma amostra inicial que ficará sozinha e o restante será unida com a terceira variável, e calculando o centroide, temos o meio das duas primeiras, a terceira amostra e um outro centroide bem longe dos outros dados.

Para um número grande de variáveis, devemos definir um limite de iterações do algoritmo, pois elas podem ficar alternando seus grupos muitas vezes. E mesmo se começarmos em uma ponta, o algoritmo consegue ir ajustando os centroides até finalizar as separações.

Por fim o algoritmo irá somar as distâncias em relação aos centroides dos grupos de cada iteração realizada e selecionar a menor delas, ou seja, será a melhor divisão de clusters selecionada.

K-means++

Ao invés de selecionar as variáveis aleatoriamente no início, o K-means++ irá atribuir um peso a cada nova escolha. Por exemplo, dado uma variável escolhida, o próximo ponto será selecionado a partir do cálculo da distância de cada ponto em relação ao primeiro escolhido e será proporcional ao quadrado das distâncias em relação ao ponto escolhido, ou seja, quanto mais distante o ponto estiver do ponto escolhido, maior a probabilidade dele ser escolhido como início do novo cluster. A vantagem é tentar diminuir o número de iterações e formar um cluster da forma mais rápida possível pelos dados já estarem longe uns dos outros.