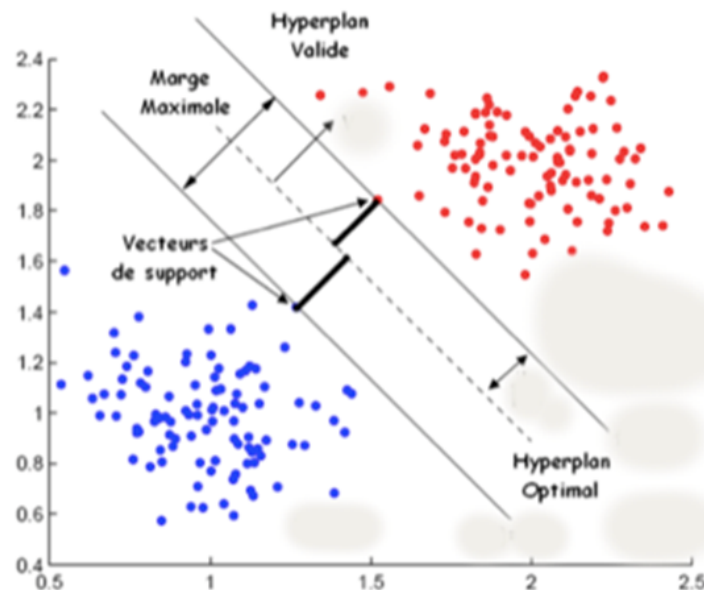


Algorithme LinearSVC

L'algorithme LinearSVC (Support Vector Classification) est un algorithme de classification utilisant des "vecteurs de support". Il fonctionne en trouvant une "frontière de décision" (ou "hyperplan") qui sépare les différentes classes dans l'espace des données. Pour cela, LinearSVC utilise une méthode mathématique appelée "optimisation quadratique" pour trouver la frontière de décision qui maximise la distance entre cette frontière et les données les plus proches des classes, appelées "vecteurs de support". Ces vecteurs de support sont ceux qui sont le plus proches de la frontière de décision et qui ont le plus grand impact sur la position de cette frontière.



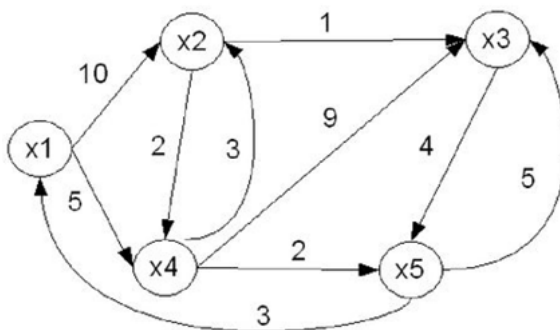
Une fois la frontière de décision trouvée, il est possible de classer de nouvelles données en déterminant de quel côté de la frontière elles se trouvent. Les données de l'autre côté de la frontière sont considérées appartenir à une autre classe. LinearSVC est souvent utilisé pour les problèmes de classification binaire (c'est-à-dire où il y a seulement deux classes possibles), mais il peut également être utilisé pour les problèmes de classification multi-classes en utilisant une technique appelée "un contre un" ou "un contre tous". Cela consiste à entraîner un modèle pour chaque classe par rapport à toutes les autres classes. En général, LinearSVC est un algorithme efficace pour les problèmes de classification linéairement séparables, c'est-à-dire lorsque les données peuvent être séparées en utilisant une frontière de décision linéaire. Il peut également être utilisé pour les problèmes de classification non linéaire en utilisant des techniques de "kernélisation" pour transformer les données dans un espace à plus grande dimension où ils peuvent être séparés linéairement.

Algorithme de Dijkstra

L'algorithme de Dijkstra est un algorithme de recherche de plus court chemin utilisé pour trouver le chemin le plus court entre deux sommets dans un graphe pondéré. Il est souvent utilisé pour des graphes représentant des réseaux de transport, des cartes routières ou des réseaux informatiques. L'algorithme de Dijkstra fonctionne en utilisant une méthode appelée "relaxation". Il commence par considérer que le sommet de départ est le plus proche de lui-même et que la distance vers tous les autres sommets est infinie. Il vérifie alors tous les sommets adjacents au sommet de départ et met à jour la distance vers ces sommets si la distance trouvée est plus courte que celle enregistrée. Il marque ensuite le sommet de départ comme étant visité et sélectionne le sommet non visité le plus proche comme nouveau sommet de départ. Il recommence alors ce processus pour ce nouveau sommet, en vérifiant tous les sommets adjacents et en mettant à jour les distances s'il trouve un chemin plus court. Il continue ce processus jusqu'à ce qu'il atteigne le sommet final ou qu'il n'y ait plus de sommets non visités. A chaque itération de l'algorithme, il stocke la distance minimale à chaque sommet dans une table de distances, avec également le sommet précédent permettant de reconstituer le chemin le plus court une fois l'algorithme terminé. Il est important de noter que l'algorithme de Dijkstra ne fonctionne que pour les graphes pondérés avec des poids positifs sur les arêtes (c'est-à-dire pas de poids négatifs), cependant il existe une variante de l'algorithme Dijkstra appelé Bellman-Ford qui permet de gérer les graphes avec des poids négatifs. En résumé, l'algorithme de Dijkstra est un algorithme de recherche de chemin efficace qui utilise une méthode de relaxation pour trouver le chemin le plus court entre deux sommets dans un graphe pondéré en utilisant une table de distance pour mémoriser les sommets visités et leur distance minimale.

Plus courts chemins – Algorithme de Dijkstra : Exemple

- **Application : plus court chemin à partir de x1**



x1	x2	x3	x4	x5
0	∞	∞	∞	∞
0	10, x1	∞	5, x1	∞
•	8, x4	14, x4	5, x1	7, x4
•	8, x4	12, x5	•	7, x4
•	8, x4	9, x2	•	•
•	•	9, x2	•	•

- **Pour chaque sommet, mémoriser :**

- Son meilleur cout courant,
- le sommet précédent pour ce meilleur cout
- un marquage