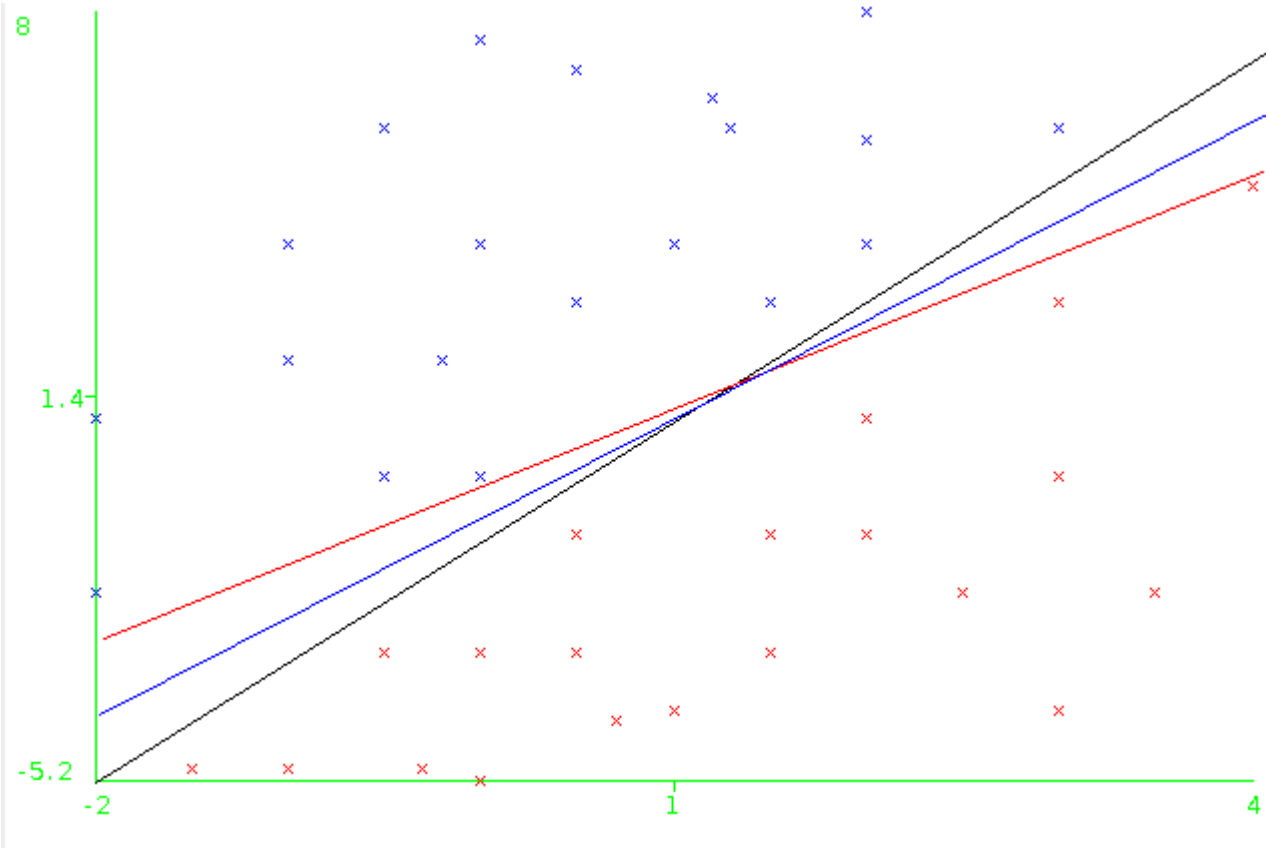
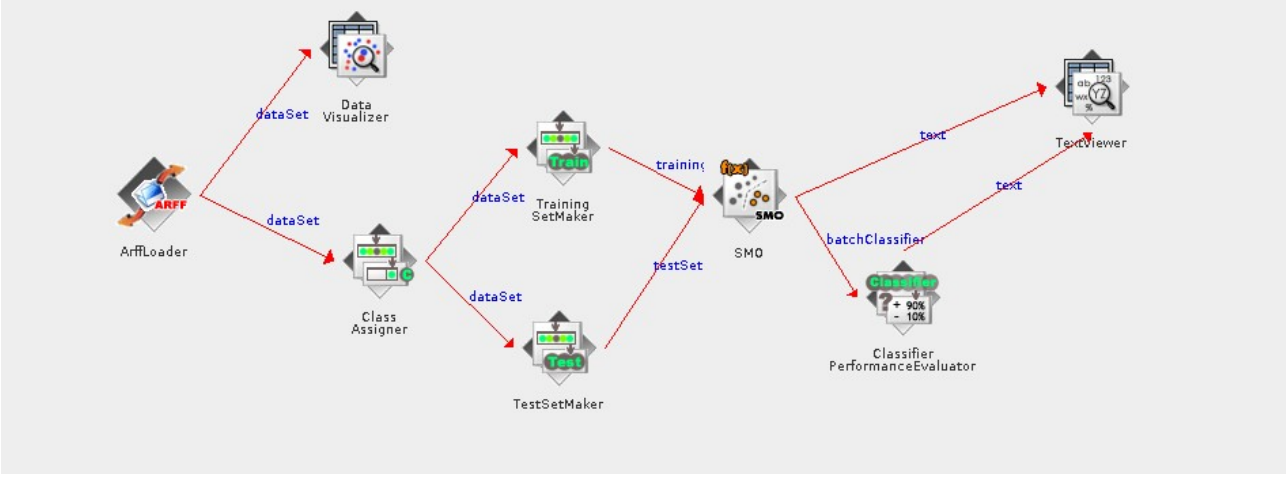
Apprentissage d'un SVM

Données linéairement séparables



3.1.1 On peut tracer un nombre infini de droites dans le but de séparer linéairement les ensembles positif et négatif (3 exemples sur le schéma ci-dessus). La meilleure solution est celle qui maximise l'écart entre les points et la droite. C'es celle qui discrimine le mieux les deux classes.

3.1.2 L'équation de la droite inférée : 1.4531 * x + -0.8174 * y – 0.7266



=== Confusion Matrix ===

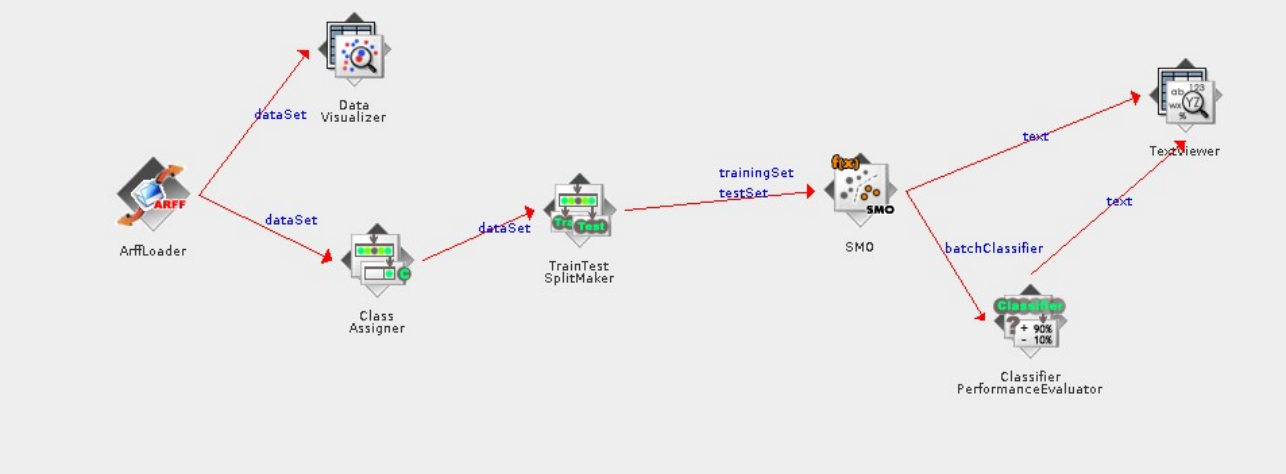
a b <-- classified as

20 0 | a = pos

0 20 | b = neg

Avec cette matrice de confusion, nous observons qu'il n'y a aucun mal classé. Nous pouvons donc calculer le risque empirique, qui ici est nul (0/40). Nous avons utilisé l'échantillon d'apprentissage et de test pour créer notre droite, c'est pour cela que le risque empirique est nul.

3.1.3 Ici nous testons en apprenant avec l'echantillon d'apprentissage et en testant sur celui de test.



Correctly Classified Instances 14 100 %

On voit bien qu'ici le SMO teste bien sur l'échantillon de test (sur 14 données, ce qui correspond à 33 % des données de base).

=== Confusion Matrix ===

a b <-- classified as

8 0 | a = pos

0 6 | b = neg

Le classifier est toujours aussi bon car ici le problème est simple, il est facile de différencier les deux classes.

Avec la cross validation, le résultat reste le même malgré un nombre de données testées plus élevés.

3.2 En modifiant le SMO et en mettant un polynome de degré 2, on obtient :

=== Classifier model ===

Kernel used:

Poly Kernel: K(x,y) = <x,y>^2.0

Number of support vectors: 31

Nous avons un taux de mal classé de 27,5 % ce qui est mieux qu'avec un modèle linéaire qui avait un taux de 41 %.

```
  a  b   <-- classified as
 13  2 |  a = pos
  9 27 |  b = neg
```

Apprentissage d'un arbre de décision et élagage

Construction et évaluation d'arbres

4.1.1 Nous avons 5 attributs et 14 instances. La classe à prédire est play. Tous les attributs sont de type qualitatifs.

4.1.3 Nous avons 2 données de test qui ont été bien classées (Correctly Classified Instances 2 40%). L'apprentissage est plus efficace pour le yes Nous avons une précision de 50 % comme le montre le tableau ci-dessous.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
0.667	1	0.5	0.667	0.571	0.333	yes
0	0.333	0	0	0	0.333	no

4.1.4 En changeant la quantité de donneés d'apprentissage aux alentours de 70 %, on voit que la classification est meilleure. Nous pouvons aussi décider de faire de l'élagage ou non, si nous élaguons l'arbre est moins spécifique. C'est pour cela que nous pouvons attendre un taux de bien classés de 100 % avec un arbre non élagué avec des feuilles pures.

Dans le cadre de ce problème avec peu de données on préfère la cross validation. Ainsi nous obtenons des résultats plus intéressants avec un taux supérieur à 70 % avec un arbre non élagué avec des feuilles pures. Et avec élagué le taux est enore supérieur.

Le second fichier contient des variables quantitatives, ce qui affecte l'arbre : pour certaines branches ce sont des seuils qui sont utilisés.

Apprentissage bayesien

6.1.1 Hypothèse de Bayes : on utilise les probabilités a posteriori calculées à partir des probabilités a priori.

6.1.2 Les résultats fournis sont les paramètres du modèle, ce sont des estimations par . Pour les variables qualitatives, la valeur est un comptage lissé par une Laplacienne.