

Chapitre IV

Conception

IV.1 Introduction

Après l'étude de la gestion des relations avec les clients dans les banques, et après l'identification des problèmes qui se posent dans la GRC, nous avons déterminé les techniques de data mining nécessaires dans la résolution de ces problèmes. Dans ce chapitre nous allons présenter la conception de notre système permettant d'appliquer ces techniques.

IV.2 Architecture générale

Afin d'atteindre les modèles que nous cherchons, et qui permettent de résoudre les trois problèmes de la GRC à savoir l'acquisition, la rétention et la segmentation des clients, nous avons conçu une architecture générale pour ce système (figure 4.1). Il se compose principalement de trois modules, un module pour chaque problème.

Module d'acquisition (figure 4.3): Permet de trouver un modèle qui spécifie les perspectives qui peuvent devenir des clients, grâce au processus d'apprentissage et assure la validité du modèle obtenu par le processus de validation.

Module de rétention (figure 4.7): le module de rétention permet d'avertir l'administrateur si il ya des clients susceptible de quitter la banque à partir du contrôle du comportement des clients.

Module de segmentation (4.8): Ce module permet de trouver les groupes qui composent les clients de la banque, ce qui donne à l'administrateur un regard approfondi sur les différents types de ses clients.

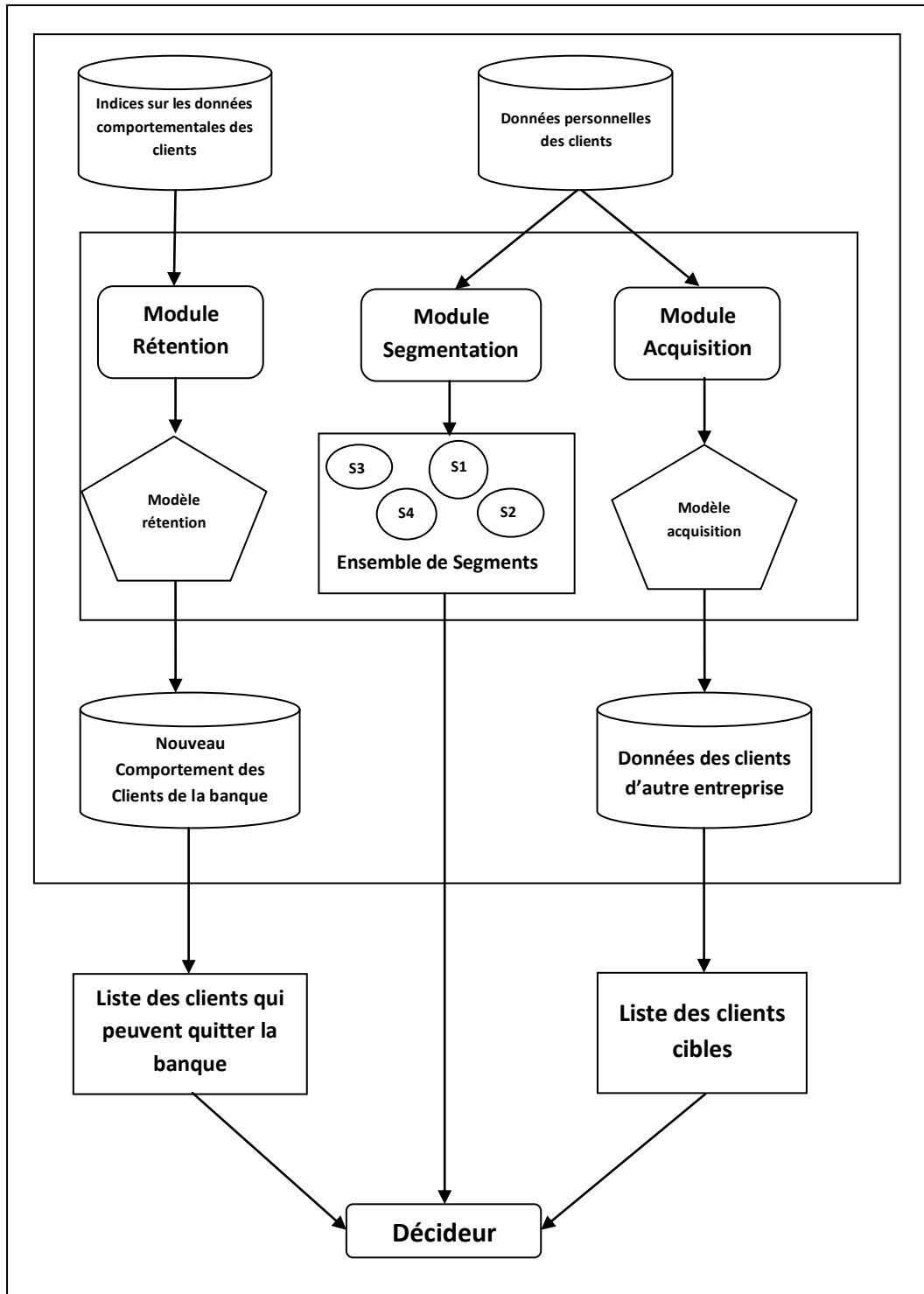


Figure 4.1: Architecture générale du système

IV.3 Architecture détaillé

IV.3.1 Bases de données

- **Base d'apprentissage** : La base de données d'apprentissage est l'ensemble des données des clients de la banque mentionnées dans la section « données du client ». Où cette base est divisée en deux parties une pour l'entraînement, généralement comprise 70 à 90% où il devrait être disponible sur certaines caractéristiques pour un processus d'apprentissage juste. Ces informations ne doivent pas être bruitées ou erronées.

La base de données de test sera semblable à la base de données d'entraînement et représente entre 10 et 30% pour l'application du modèle obtenus, afin de vérifier la validité de ce modèle.

- Pour la segmentation on a besoin d'une seule base de données pour faire la segmentation.

- **Base d'utilisation** : La base d'utilisation est l'ensemble de données d'autres sociétés obtenues par un accord entre la banque et cette société. Ces données sont préparées par la sélection des données similaires entre la base de données de la banque et la base de données d'utilisation pour la possibilité d'appliquer le modèle.

IV.3.2 Module d'acquisition

IV.3.2.1 Données d'apprentissage

Les données d'apprentissage se composent dans ce cas des informations personnelles des clients, y compris les détails des données démographiques et de contact.

Les attributs des clients prospectifs obtenus de sources externes doivent être similaires aux informations des clients de la banque.

IV.3.2.2 Apprentissage SVM mono-classe

Dans la cas d'acquisition, la base de données de la banque contient des exemples d'un seul type de clients, c'est l'ensemble des clients de la Banque avec l'absence de contre-exemples pour cela on va utiliser l'apprentissage SVM mono-classe.

La technique de SVM mono-classe permet de trouver un hyperplan sépare les exemples de la classe cible de l'origine, et qui maximise la marge entre les deux comme dans la figure suivante (figure 4.2).

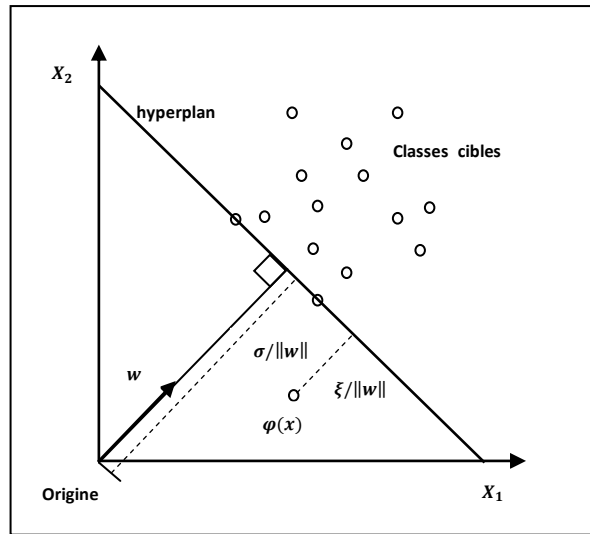


Figure 4.2 : SVM mono-classe

Le problème est de modéliser par le problème primal de programmation quadratique de l'équation (4.1) dont l'objectif est de maximiser la marge et minimiser les erreurs de classification. La contrainte est la bonne classification des exemples d'entraînement.

$$\begin{cases} \min_{w, \xi, \rho} \frac{1}{2} \|w\|^2 + \frac{1}{\nu N} \sum_{i=1}^l \xi_i - \rho \\ \langle w, \phi(x_i) \rangle \geq \rho - \xi_i \\ \xi_i \geq 0 = 1, 2, \dots, N \end{cases}$$

4.1

Où N est le nombre d'exemples de la classe cible, (w, ρ) les paramètres permettant de localiser l'hyperplan, ξ_i représentent les erreurs permises sur les exemples, pénalisées par le paramètre ν et ϕ est une transformation d'espace semblable à celle du cas binaire. Une fois (w, ρ) déterminés, tout nouvel exemple pourra être classé par la fonction de décision de l'équation (4.2) :

$$f(x) = \langle w, \phi(x_i) \rangle - \rho$$

4.2

x appartient à la classe cible si $f(x)$ est positive.

En fait, la résolution du problème de l'équation 4.1 est réalisée par l'introduction des multiplicateurs de Lagrange pour obtenir le problème dual de l'équation (4.3) :

$$\left\{ \begin{array}{l} \text{Minimiser } \alpha \quad \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(x_i, x_j) \\ \text{sous contraintes} \quad \sum_{i=1}^n \alpha_i = 1 \\ 0 \leq \alpha_i \leq \frac{1}{vN} \end{array} \right.$$

4.3

Où K est un noyau qui représente la transformation d'espace ϕ .

Une fois les α_i déterminés ils peuvent être dans l'un des trois cas suivants :

- $\alpha_i = 0$: correspondent aux exemples bien classés c-à-d qui se situent au dessus de l'hyperplan,
- $i = 1/vN$ correspondent aux exemples qui se situent à l'intérieur de la marge (au dessous de l'hyperplan),
- $0 < i < 1/vN$ correspondent aux exemples vecteurs support qui se situent sur l'hyperplan.

La fonction de décision pour tout exemple x est donnée par l'équation (4.4) :

$$f(x) = \sum_{i=1}^l \alpha_i K(x_i, x) - \rho$$

4.4

Où ρ peut être déterminé à partir d'un exemple x_i d'apprentissage dont $\alpha_i \neq 0$ par l'équation (4.5).

$$\rho = \sum_j \alpha_j K(x_j, x_i)$$

4.5

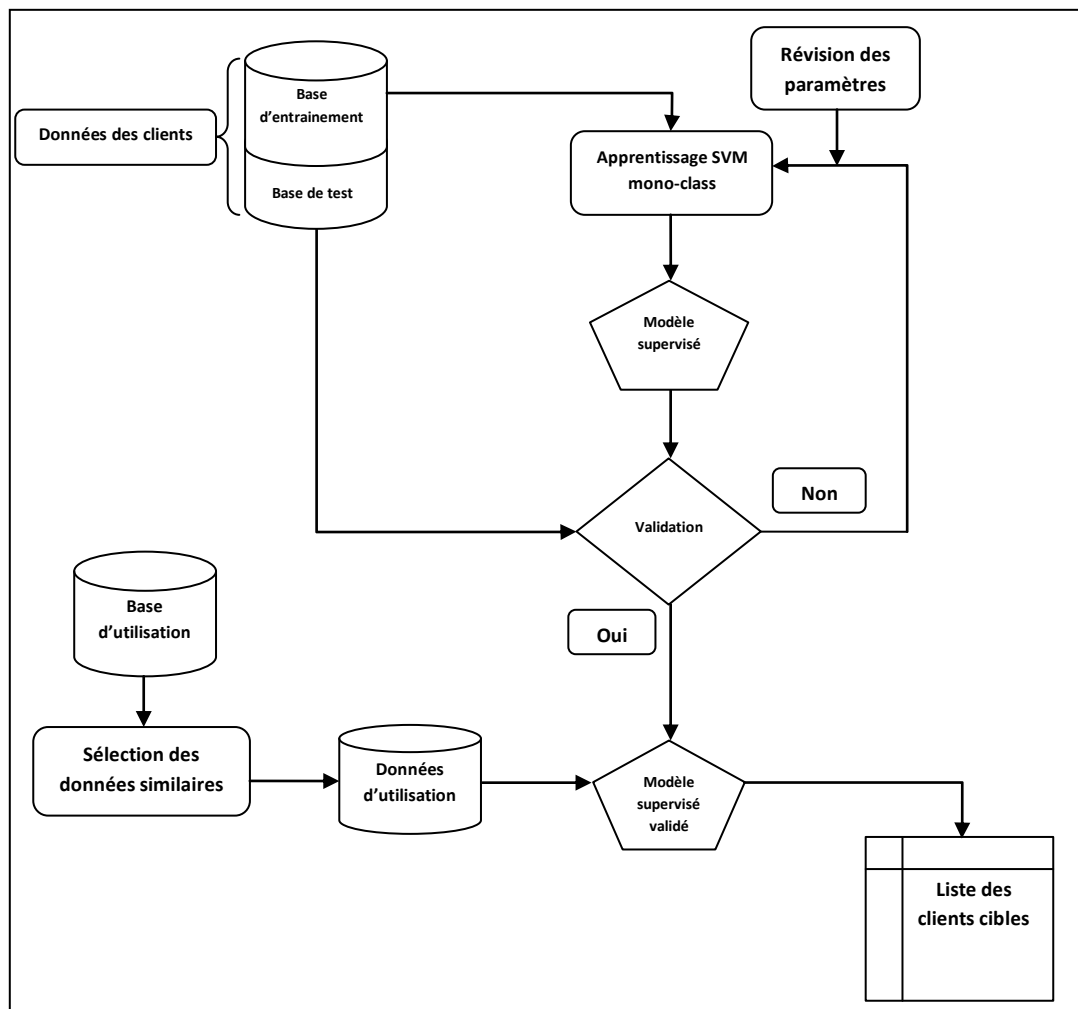


Figure 4.2 : Module d'acquisition

IV.4 Module de rétention

IV.4.1 Données d'apprentissage

La construction du modèle pour la rétention des clients utilise les données relatives au comportement du client (ses transactions). Les indices sur le comportement du client peuvent être le maximum de retrait, le maximum de virement, la moyenne mensuelle des retraits, le nombre moyen mensuelle de retrait, ...etc. Ces données peuvent être conclues à partir de la base de transaction des clients comme dans le tableaux suivant (tableaux 4.1).

N° C	Versement	Retrait
1	10000	0
2	0	5000
1	12000	0
3	100000	0
4	0	50000
3	0	10000

devient

N° C	Max Vers	Max ret	Moy Vers	Moy ret
1	12000	0	11000	0
2	0	5000	0	2500
3	100000	10000	100000	10000
4	0	50000	0	50000

Tableau 4. 1 : Transformation d'un tableau de comportement à un tableau des indices

A chaque nouvelle transaction d'un client, ses indices sont recalculés et exposés au modèle construit, si le modèle retourne une réponse négative, les mesures de rétention sont lancées.

IV.4.2 Apprentissage SVM binaire

Dans le cas de rétention, la banque dispose des données des clients fidèles et ceux infidèles ce qui signifie l'existence de deux classes et que la méthode d'apprentissage appropriée et celle binaire. Nous proposons d'utiliser dans ce cas les SVMs binaires.

L'idée des SVMs binaires est de rechercher un hyperplan (droite dans le cas de deux dimensions) qui sépare le mieux ces deux classes (figure 4.4).

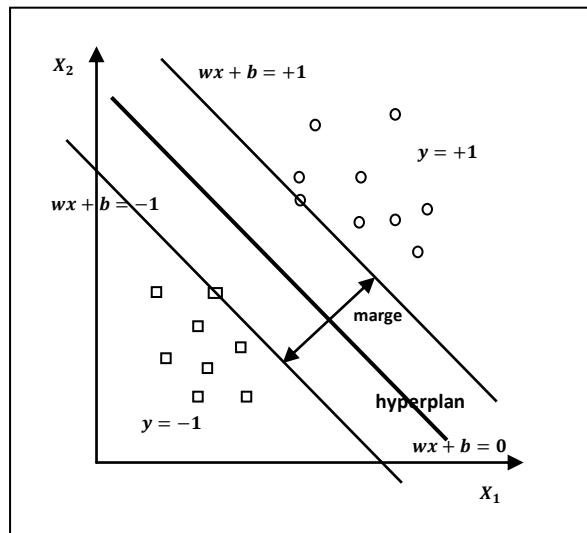


Figure 4.3 : SVM binaire

Dans ce cas l'hyperplan séparateur se représenter par l'équation suivante (4.6) :

$$H(x) = w^T x + b$$

4.6

Où w est un vecteur de m dimensions et b est un terme. La fonction de décision, pour un exemple x , peut être exprimée comme suit (4.7):

$$\begin{cases} \text{Classe} = 1 & \text{Si } H(x) > 0 \\ \text{Classe} = -1 & \text{Si } H(x) < 0 \end{cases}$$

4.7

Puisque les deux classes sont linéairement séparables, il n'existe aucun exemple qui se situe sur l'hyperplan, c-à-d qui satisfait $H(x) = 0$. Il convient alors d'utiliser la fonction de décisions suivante (4.8):

$$\begin{cases} \text{Classe} = 1 & \text{Si } H(x) > 1 \\ \text{Classe} = -1 & \text{Si } H(x) < -1 \end{cases}$$

4.8

Le but de cette machine est de maximiser la marge $\frac{2}{\|w\|^2}$ avec le sous contrainte :

$$y_i(w^T x_i + b) \geq 1 \quad \forall i = 1..n$$

4.9

Dans ce cas on parle sur une machine SVM a marge dure parce qu'il n'y a pas des exemples entre les deux hyperplans.

En réalité, un hyperplan séparateur n'existe pas toujours, et même s'il existe, il ne représente pas généralement la meilleure solution pour la classification. En plus une erreur d'étiquetage dans les données d'entraînement (un exemple étiqueté +1 au lieu de -1 par exemple) affectera crucialement l'hyperplan.

Dans le cas où les données ne sont pas linéairement séparables, ou contiennent du bruit (outliers : données mal étiquetées) les contraintes de l'équation (4.9) ne peuvent être vérifiées, et il y a nécessité de les relaxer un peu. Ceci peut être fait en admettant une certaine erreur de classification des données (figure 4.5) ce qui est appelé "SVM à marge souple (Soft Margin)".

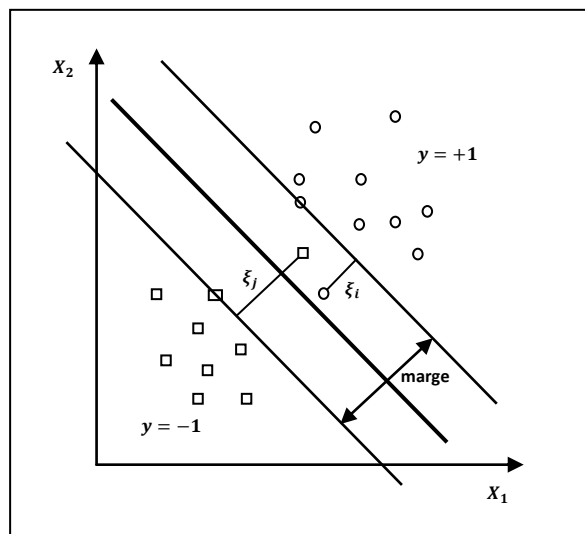


Figure 4.4 : SVM binaire à marge souple

On introduit alors sur les contraintes des variables ξ_i dites de relaxation pour obtenir la contrainte de l'équation (4.10) :

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1..n$$

4.10

Grâce aux variables de relaxation non négatives ξ_i , un hyperplan séparateur existera toujours. Si $\xi_i < 1$, x_i ne respecte pas la marge mais reste bien classé, sinon x_i est mal classé par l'hyperplan. Dans ce cas, au lieu de rechercher uniquement un hyperplan séparateur qui maximise la marge, on recherche un hyperplan qui minimise aussi la somme des erreurs permises c-à-d minimiser $Q(w) = \sum_{i=1}^n \xi_i$

On obtient le problème primal de l'équation (4.11) suivant :

$$\left\{ \begin{array}{l} \text{Minimiser} \quad \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i \\ \text{sous contraintes} \\ y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1..n \\ \xi_i \geq 0 \end{array} \right.$$

4.11

Où C est un paramètre positif libre (mais fixe) qui représente une balance entre les deux termes de la fonction objective (la marge et les erreurs permises) c-à-d entre la maximisation de la marge et la minimisation de l'erreur de classification. On obtient le problème dual de l'équation (4.12) où on introduit les multiplicateurs de Lagrange α_i et β_i :

$$Q(w, b, \alpha, \xi, \beta) = \frac{1}{2} w^T w + C \sum_{i=1}^n \alpha_i y_i (w^T x_i + b) - 1 + \xi_i - \sum_{i=1}^n \beta_i \xi_i$$

4.12

À la solution optimale, les dérivées par rapport aux variables w, b, α, β s'annulent ainsi que le produit des contraintes aux multiplicateurs. Les conditions suivantes (4.13) sont alors vérifiées :

$$\left\{ \begin{array}{l} \frac{\partial Q(w, b, \alpha, \xi, \beta)}{\partial w} = 0 \quad (a) \\ \frac{\partial Q(w, b, \alpha, \xi, \beta)}{\partial b} = 0 \quad (b) \\ \alpha_i \{y_i(w^T x_i + b) - 1 + \xi_i\} = 0 \quad (c) \\ \beta_i \xi_i = 0 \quad (d) \\ \alpha_i \geq 0; \beta_i \geq 0; \xi_i \geq 0 \quad (e) \end{array} \right.$$

4.13

On déduit :

$$\begin{cases} w = \sum_{i=1}^n \alpha_i y_i x_i \\ \sum_{i=1}^n \alpha_i y_i = 0 \\ \alpha_i + \beta_i = 0 \end{cases}$$

4.14

En remplaçant l'équation (4.14) dans l'équation (4.12), on obtient le problème dual (4.15) suivant :

$$\begin{cases} \text{Maximiser} & Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j x_i^T x_j \\ \text{sous contraintes} & \sum_{i=1}^n \alpha_i y_i = 0 \\ & 0 \leq \alpha_i \leq C \end{cases}$$

4.15

La seule différence avec la SVM à marge dure est que les α_i ne peuvent pas dépasser C , ils peuvent être dans l'un des trois cas suivants :

$\alpha_i = 0 \Rightarrow \beta_i = C \Rightarrow \xi_i = 0 : x_i$ est bien classé,

$0 < \alpha_i < C \Rightarrow \beta_i > 0 \Rightarrow \xi_i = 0 \Rightarrow y_i(w^T x_i + b) = 1 : x_i$ est un vecteur support est appelé dans ce cas vecteur support non borné,

$\alpha_i = C \Rightarrow \beta_i = 0 \Rightarrow \xi_i \geq 0 \Rightarrow y_i(w^T x_i + b) = 1 - \xi_i : x_i$ est un vecteur support appelé dans ce cas vecteur support borné. Si $0 \leq \xi_i < 1$, x_i est bien classé, sinon x_i est mal classé.

La fonction de décision est alors calculée de la même manière que dans le cas des SVMs à marge dure mais uniquement à base des vecteurs supports non bornés par :

$$H(x) = \sum_{i \in U} \alpha_i y_i x_i^T x + b$$

4.16

Pour les vecteurs supports non bornés, nous avons :

$$b = y_i - w^T x_i$$

4.17

Pour garantir une bonne précision, on prend la moyenne de b pour tous les vecteurs supports non bornés :

$$b = \frac{1}{|U|} \sum_{i \in U} y_i - w^T x_i$$

4.18

IV.4.3 Utilisation des noyaux

Dans certains exemples nous ne pouvons pas trouver une bonne généralisation pour un hyperplan même si ce dernier est optimisé.

La détermination d'une telle fonction non linéaire est très difficile voire impossible. Pour cela les données sont amenées dans un espace où cette fonction devient linéaire (figure 4.6), cette astuce permet de garder les mêmes modèles de problèmes d'optimisation vus dans les sections précédentes, utilisant les SVMs basés essentiellement sur le principe de séparation linéaire. Cette transformation d'espace est réalisée souvent à l'aide d'une fonction $F = \{\phi(x) | x \in X\}$ appelé "Mapping function" et le nouvel espace est appelé espace de caractéristiques "Features space".

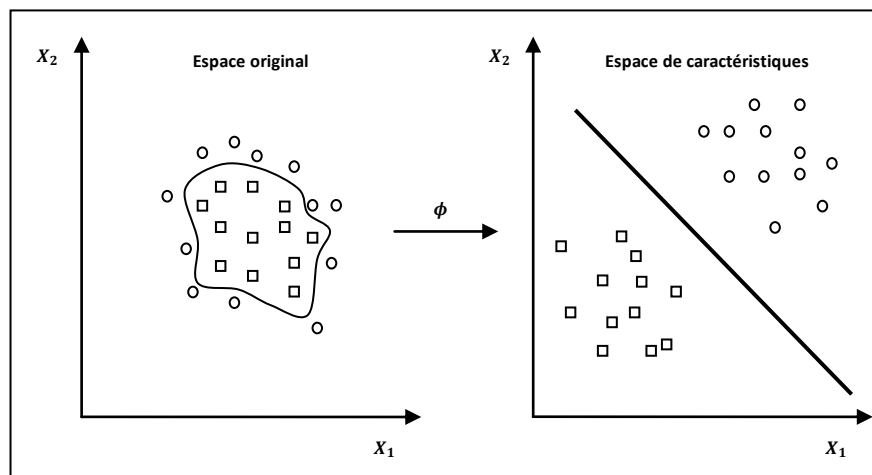


Figure 4.5 : Transformation d'espace

L'utilisation du noyau permet de surmonter le problème de détermination de la transformation ϕ et permet d'apprendre des relations non linéaires par des machines linéaires. En pratique, il existe certains noyaux qui sont très utilisés et qui sont considérés comme standards. Une fois le noyau choisi, la fonction objective peut être calculée comme suit :

$$Q(\alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

4.19

Et la fonction de décision devient :

$$H(x) = \sum_{i \in S} \alpha_i y_i K(x_i, x) + b$$

4.20

Parmi les noyaux qui existent, il y a :

Le noyau polynomial qui élève le produit scalaire à une puissance naturelle d :

$$K(x_i, x_j) = (x_i^T x_j)^d$$

4.21

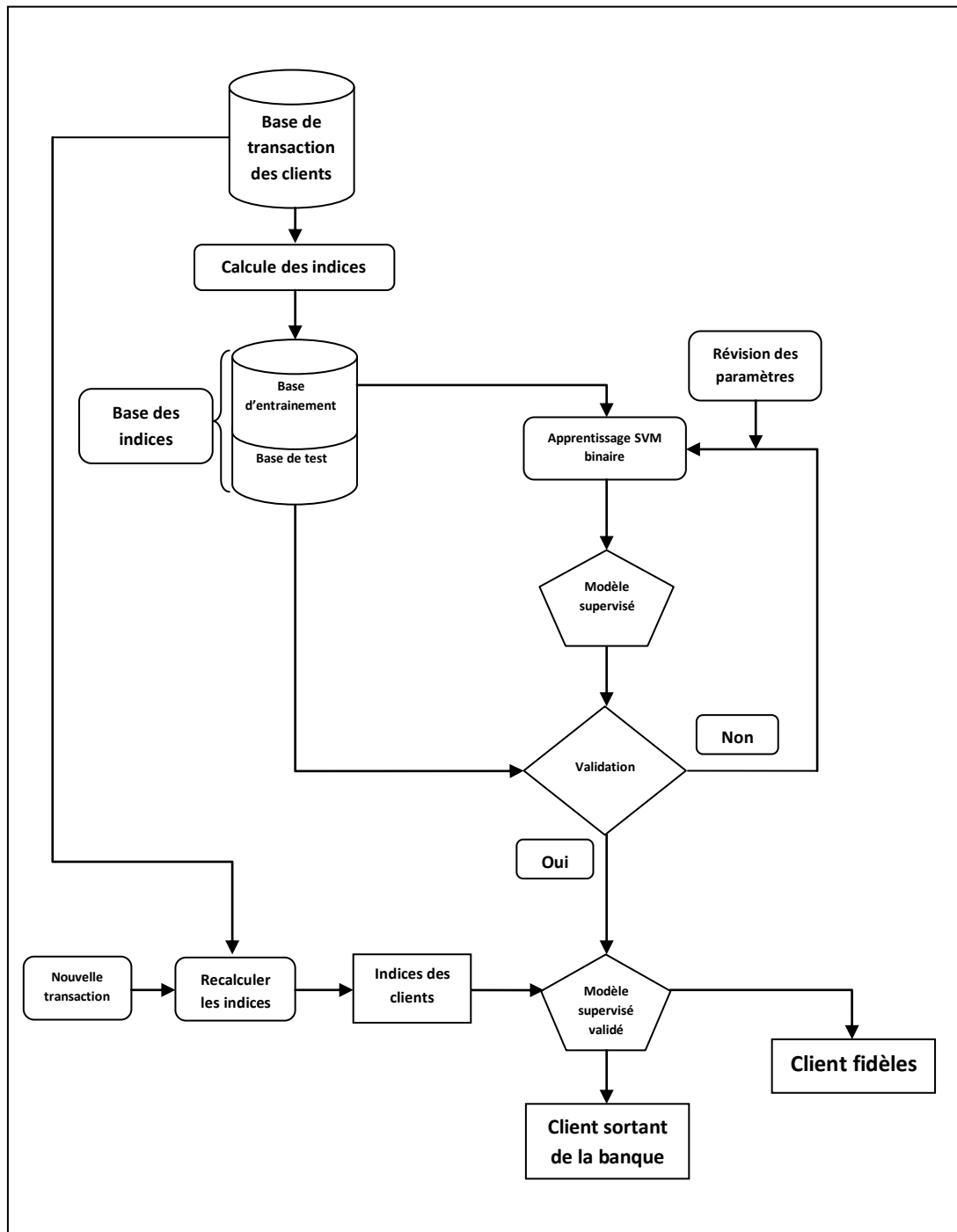


Figure 4.6 : Module de rétention

IV.4.4 Validation du modèle

Pour tester la validité de nos modèles nous avons choisi la validation croisées. On subdivise l'ensemble d'entraînement initial en k sous ensemble disjoints D_1, D_2, \dots, D_k de même taille. A chaque itération i on réserve le sous-ensemble D_i pour le test et le reste des exemples sont utilisés pour entraîner le modèle. La précision finale du modèle est égale à la moyenne des k précisions de test.

On calcule la précision de chaque itération avec la formule suivante (4.1) :

$$P = \frac{1}{N} \sum_{i=1}^N L(y_i, \hat{f}(x_i))$$

4.22

$$\text{avec } L = \begin{cases} 1 & \text{si } y_i = \hat{f}(x_i) \\ 0 & \text{sinon} \end{cases}$$

IV.4.5 Paramètres d'apprentissage

Il y'a plusieurs paramètre d'apprentissage qui permet d'influencer les résultats de cette opération. Les paramètres des fonctions noyau et les paramètres concernant le type d'SVM utilisé. Dans le chapitre suivant nous allons parler en détail de ces paramètres.

IV.5 Module de segmentation

IV.5.1 Base de donnée

Dans le premier chapitre nous avons parlé de deux types de données. Données de comportement et données personnelles du client. Nous allons choisir les données personnelles comme données de segmentation et la méthode K-means comme méthode de segmentation.

IV.5.2 K-means

Le k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centroïdes conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. K-means change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer. Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur K du nombre de clusters [FRE05].

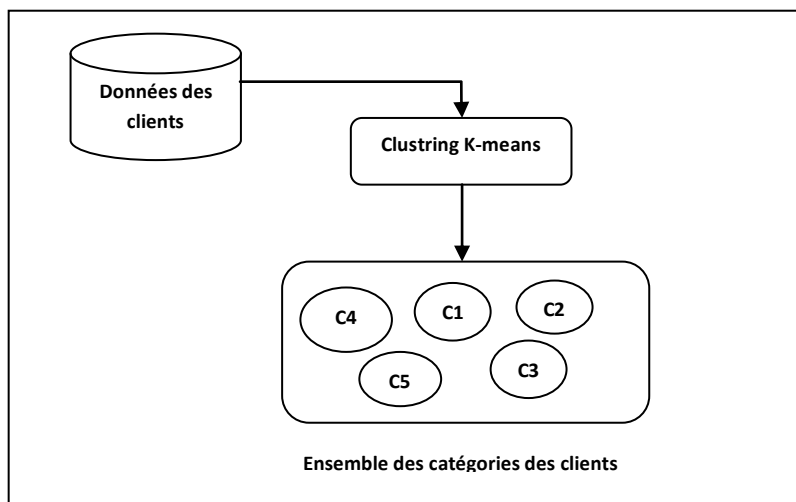


Figure 4.7 : Module de segmentation

IV.6 Conclusion

Dans ce chapitre nous avons étudié toutes les parties de notre système, ses composantes, ses opérations. En plus, nous avons parlé des données des clients et comment les préparer, et des techniques utilisées. Dans le prochain chapitre, nous allons présenter la réalisation de ce système.