PROBABILISTIC GRAPHICAL MODEL

Guillaume PETIT
Thomas FAURÉ

22 Novembre 2019

1 Learning in discrete graphical models

La log-vraisemblance s'écrit :

$$L(\pi, \theta) = \sum_{n=1}^{N} \sum_{m=1}^{M} z_m^n log(\pi_m) + \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{k=1}^{K} z_m^n x_k^n log(\theta_{mk})$$

On note N_m le nombre de réalisations où $z = m$ et N_{mk} le nombre de réalisations où $z = m$ et $x = k$.
On réécrit donc :

$$L(\pi, \theta) = \sum_{m=1}^{M} N_m log(\pi_m) + \sum_{m=1}^{M} \sum_{k=1}^{K} N_{mk} log(\theta_{mk})$$

On veut maximiser la log-vraisemblance par apport π_m et θ_{mk} sous les contraintes que $\pi_m > 0$ et $\theta_{mk} > 0 \ \forall$ m , k. De plus $\sum_{m=1}^{M} \pi_m = 1$ et $\sum_{k=1}^{K} \theta_{mk} = 1 \forall m$. On peut négliger les contraintes d'inégalités car lorsque π et θ tendent vers 0 le lagrangien décroît vers $-\infty$.
Le lagrangien s'écrit :

$$L(\mu, \lambda_1, ..., \lambda_M) = \sum_{m=1}^{M} N_m log(\pi_m) - \mu(\sum_{m=1}^{M} \pi_m - 1) +$$

$$+ \sum_{m=1}^{M} \sum_{k=1}^{K} N_{mk} log(\theta_{mk}) - \sum_{m=1}^{M} \lambda_m (\sum_{k=1}^{K} \theta_{mk} - 1)$$

On peut optimiser séparément. On dérive par apport à π_i :

$$\frac{\partial L}{\partial \pi_i} = \frac{N_i}{\pi_i} - \mu$$

On cherche la valeur de π_i qui annule la quantité ci-dessus et on trouve :

$$\pi_i = \frac{N_i}{\mu}$$

En sommant par apport à i on trouve $1 = \frac{N}{\mu}$ et finalement $\widehat{\pi_i} = \frac{N_i}{N}$

De la même manière on trouve $\frac{\partial L}{\partial \theta_{mk}} = \frac{N_{mk}}{\theta_{mk}} - \lambda_m$ et finalement $\widehat{\theta_{mk}} = \frac{N_{mk}}{N_m}$

2 Linear Classification

2.1 Generative model LDA

Question a

Pour ce modèle, les paramètres sont $\pi \in [0,1]$, $\mu_1 \in \mathbf{R}^2$, $\mu_2 \in \mathbf{R}^2$ et $\Sigma \in S^2_{++}$. Soit $\Theta = (\pi, \mu_0, \mu_1, \Sigma) \in \mathbf{R}^9$ et n le nombre de variables iid des observations $(x_i, y_i) \in \mathbf{R}^2 \times \{0,1\}$. La vraisemblance de notre modèle est donc :

$$L(\Theta) = \prod_{i=1}^{n} p_\theta(x_i, y_i)$$

Or on sait que

$$\begin{aligned} p(x_i, y_i) &= p_\theta(x_i|y_i)p(y_i) \\ p(y_i) &= \pi^{y_i}(1-\pi)^{1-y_i} \\ p(x_i|y_i) &= \mathcal{N}(\mu_1, \Sigma)^{y_i}\mathcal{N}(\mu_0, \Sigma)^{1-y_i} \end{aligned}$$

Donc on peut réécrire :

$$L(\Pi, \Theta) = \prod_{i=1}^{n} \mathcal{N}(\mu_1, \Sigma)^{y_i}\mathcal{N}(\mu_0, \Sigma)^{1-y_i}\pi^{y_i}(1-\pi)^{1-y_i}$$

Ainsi la log-vraisemblance de notre problème est égale à :

$$l(\Theta) = \underbrace{\sum_{i=1}^{n} y_i log(\pi) + (1-y_i)log(1-\pi)}_{A(\pi)} + \sum_{i=1}^{n} y_i log(\mathcal{N}(\mu_1, \sigma)) + (1-y_i)log(\mathcal{N}(\mu_0, \sigma))$$

$$l(\Theta) = A(\pi) + \sum_{i=1}^{n} -\frac{y_i}{2}(x_i-\mu_1)^T\Sigma^{-1}(x_i-\mu_1) - \frac{1-y_i}{2}(x_i-\mu_0)^T\Sigma^{-1}(x_i-\mu_0) - \frac{n}{2}log(2\pi) - \frac{n}{2}log(|\Sigma^{-1}|)$$

$$l(\Theta) = A(\pi) + G(\mu_0, \mu_1, \Sigma)$$

On remarque que $l(\Theta)$ est une fonction concave en chacun des paramètres. De plus, la log-vraisemblance est continue et dérivable. Pour la maximiser dans le but de trouver $\widehat{\Theta}$, on a juste à dériver la fonction.

• **Maximisation par rapport à π** :

$$\begin{aligned} \frac{\partial l}{\partial \pi} = 0 &\iff \sum_{i=1}^{n} \frac{y_i}{\pi} - \frac{1-y_i}{1-\pi} = 0 \\ &\iff \widehat{\pi} = \frac{1}{n}\sum_{i=1}^{n} y_i \end{aligned}$$

• **Maximisation par rapport à μ_1** :

$$\frac{\partial l}{\partial \mu_1} = 0 \iff \sum_{i=1}^{n} \frac{y_i}{2} \left(2_i^T \Sigma^{-1} + 2\mu_1 \Sigma^{-1}\right)$$
$$\iff \widehat{\mu_1} = \frac{\sum_{i=1}^{n} y_i x_i^T}{\sum_{i=1}^{n} y_i}$$

• **Maximisation par rapport à μ_0** : Par symétrie, on a

$$\widehat{\mu_0} = \frac{\sum_{i=1}^{n} (1-y_i) x_i^T}{\sum_{i=1}^{n} (1-y_i)}$$

• **Maximisation par rapport à Σ** :

Pour démontrer ce résultat, on fait référence aux notes de cours qui nous expliquent que la fonction :

1. $f(A) = Trace(BA)$ avec $(A,\, B)$ des matrices, a pour gradient $\nabla f(A) = B$
2. $g(A) = log(det(A))$ avec $A \in S_{++}^n$ a pour gradient $\nabla g(A) = A^{-1}$

On peut commencer par réécrire la log-vraisemblance

$$l(\Theta) = A(\pi) - \frac{1}{2} Tr\left(A_1 \Sigma^{-1}\right) - \frac{1}{2} Tr\left(A_0 \Sigma^{-1}\right) + \frac{n}{2} log\left(|\Sigma^{-1}|\right)$$

avec $A_1 = \sum_{i=1}^{n} y_i (x_i - \mu_1)^T (x_i - \mu_1)$ et $A_0 = \sum_{i=1}^{n} (1-y_i)(x_i - \mu_0)^T (x_i - \mu_0)$

En appliquant le cours, on obtient

$$\frac{\partial l}{\partial \Sigma} = 0 \iff \frac{n}{2}\Sigma - \frac{1}{2} A_1 - \frac{1}{2} A_0 = 0$$
$$\iff \widehat{\Sigma} = \frac{1}{n} (\widehat{A_0} + \widehat{A_1})$$

où $\widehat{A_0}$ et $\widehat{A_1}$ sont les estimateurs de A_0 et A_1 (μ_0 et μ_1 deviennent $\widehat{\mu_0}$ et $\widehat{\mu_1}$)

Question b

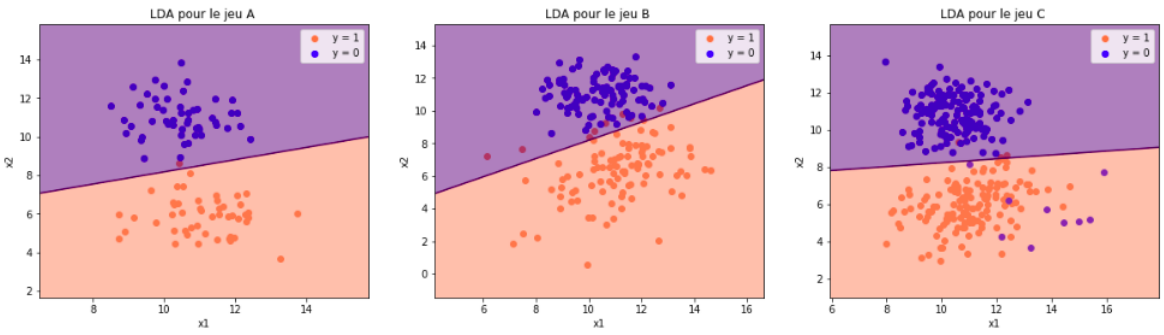
Par la formule de Bayes, on peut déterminer la valeur de $\mathbb{P}(Y = 1|X)$

$$\begin{aligned}
\mathbb{P}(y = 1|x) &= \frac{\mathbb{P}(x|y = 1)\mathbb{P}(y = 1)}{\mathbb{P}(x)} \\
&= \frac{\pi\mathcal{N}(\mu_1, \Sigma)}{\pi\mathcal{N}(\mu_1, \Sigma) + (1 - \pi)\mathcal{N}(\mu_0, \Sigma)} \\
&= \frac{1}{1 + \frac{1 - \pi}{\pi}\frac{\mathcal{N}(\mu_0, \Sigma)}{\mathcal{N}(\mu_1, \Sigma)}} \\
&= \frac{1}{1 + e^{-(w^T x + b)}} = \sigma(w^T x + b)
\end{aligned}$$

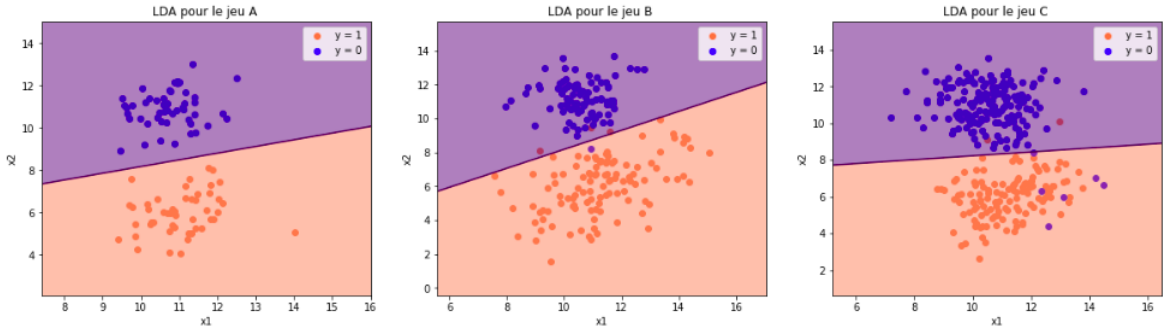
avec $w^T = \Sigma^{-1}(\mu_1 - \mu_0)$ et $b = -\frac{1}{2}\left(\mu_1^T\Sigma^{-1}\mu_1 + \mu_0^T\Sigma^{-1}\mu_0\right) + ln\left(\frac{\pi}{1 - \pi}\right)$

Question c

On obtient les résultats suivants pour la partie test et train respectivement



Classification LDA pour les dataset Test



Classification LDA pour les dataset Train

2.2 Logistic Regression

On veut implémenter une régression logistique pour une fonction affine $f(X) = w^T X + b$, que l'on peut réécrire $f(X) = \tilde{w}^T \tilde{X}$ avec $\tilde{w}$ est le vecteur w auquel on rajoute b en premiere coordonnée et $\tilde{x}$ la matrice X à laquelle on rajoute une colonne de 1. D'après le cours, Y suit une loi de Bernoulli de paramètre $\theta = \sigma(\tilde{w}^T \tilde{X})$. On cherche donc à estimer le paramètre $\tilde{w}^T$.

Dans la suite on notera :
- X la matrice des observations (plutôt que $\tilde{X}$)
- x_i l'observation i
- Y le vecteur des y_i
- w plutôt que $\tilde{w}$

On a :

$$L(\theta) = \prod_{i=1}^{n} \theta^{y_i}(1-\theta)^{(1-y_i)}$$

En passant au log :

$$l(\theta) = \sum_{i=1}^{n} y_i log(\theta) + (1-y_i) log(1-\theta)$$

On remarque que c'est une fonction concave comme combinaison linéaire de fonctions concaves. Içi, on a une formule beaucoup plus compliqué qui rend les calculs analytiquement impossible pour trouver notre estimateur. On va donc passer par Newton. On calcule alors le gradient et la hessienne de $l(\theta)$ par apport à w.
D'après le cours nous avons :

$$\frac{\partial l}{\partial w}(\sigma(w^T X)) = \sum_{i=1}^{n} x_i^T (y_i - \sigma(w^T x_i))$$

$$\frac{\partial^2 l}{\partial w^2}(\sigma(w^T X)) = X^T D X$$

ou D est la matrice diagonal tel que $D_{ii} = \sigma(w^T x_i)(1 - \sigma(w^T x_i))$.

On implémente donc :

$$w_{new} = w_{old} + (X^T D_{old} X)^{-1} X^T (Y - \sigma(w_{old}^T X))$$

question a

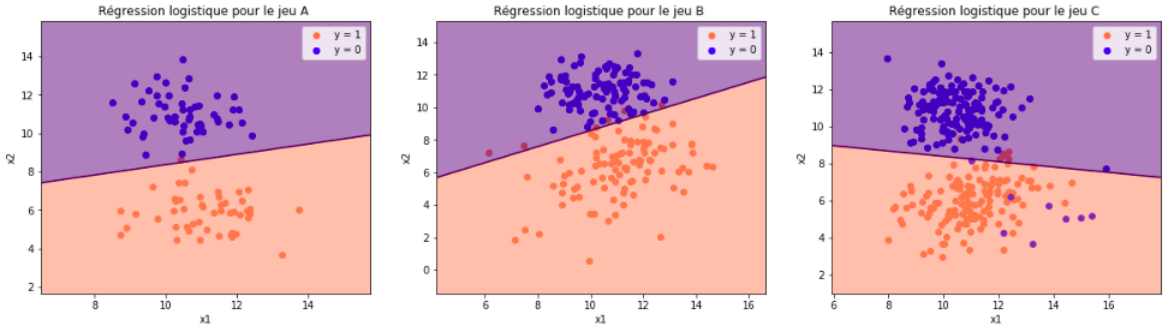
$w_A = [106.95, 5.05, -18.83]$, $w_B = [13.41, 1.84, -3.70]$ et
$w_C = [18.80, -0.27, -1.91]$

Question b

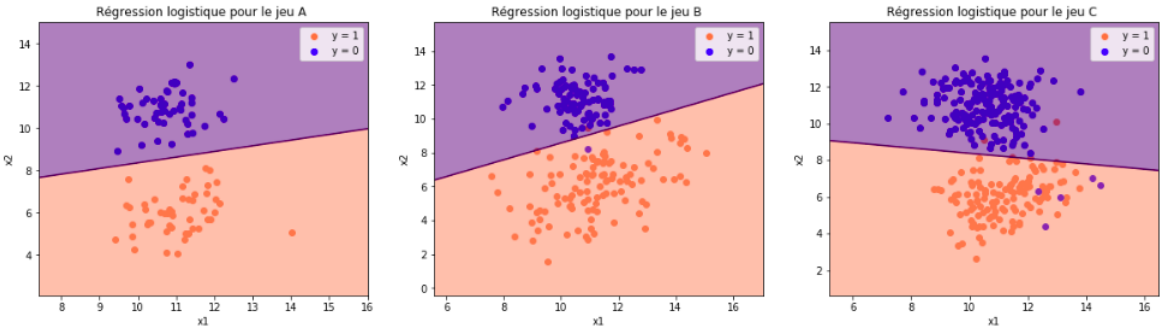
On a :

$$\mathbb{P}(y = 1|x) = 0.5 \Leftrightarrow w^T x = 0$$

et on obtient les graphiques ci-dessous :



Régression logistique pour les dataset Test



Régression logistique pour les dataset Train

2.3 Régression Linéaire

Dans le cadre de la régression linéaire, notre modèle est de la forme suivante :

$$Y = Xw + b + \epsilon = Xw + \epsilon \text{ avec } \epsilon \overset{iid}{\sim} \mathcal{N}(0, \sigma^2 Id)$$

après reparamétrisation. Comme pour la régression logistique, nous conservons X au lieu de $\tilde{X}$ et w au lieu de $\tilde{w}$. On obtient les résultats suivants :

$$\widehat{w} = \left(X^T X\right)^{-1} X^T Y$$
$$\widehat{\sigma^2} = \frac{1}{n} ||Y - X\widehat{w}||^2$$

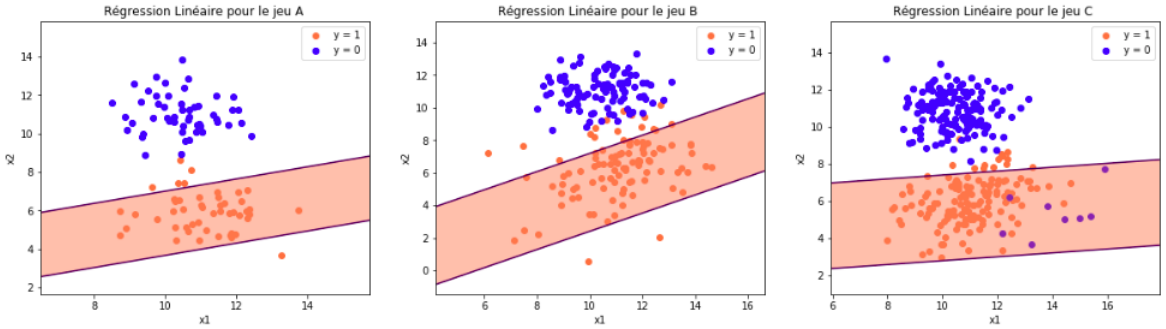
Question a on obtient les paramètres suivants :

$$w_A = [1.38, 0.05, -0.17] \text{ et } \sigma_A^2 = 0.027$$
$$w_B = [0.88, 0.082, -0.14] \text{ et } \sigma_B^2 = 0.048$$
$$w_C = [1.64, 0.016, -0.15] \text{ et } \sigma_C^2 = 0.055$$

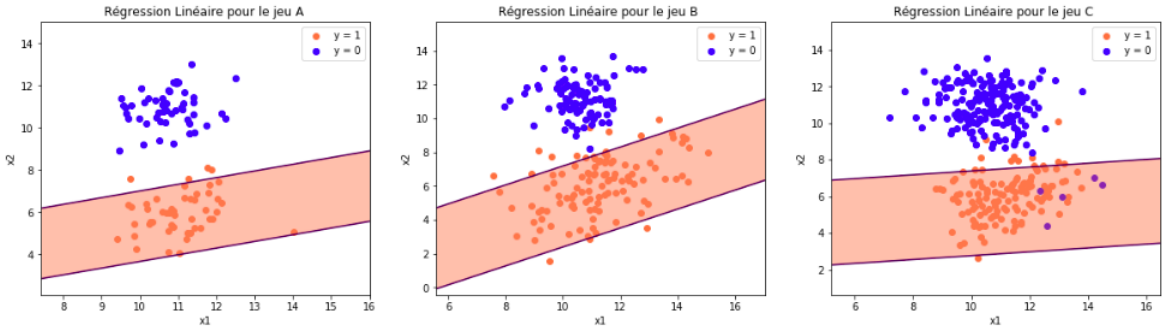
Nous avons vu au début de la régression linéaire que $Y|X \sim \mathcal{N}(Xw, \sigma^2 Id)$.
Ainsi l'hyperplan séparateur est de la forme :

$$P(y=1|x) = 0.5 \iff \frac{1}{\sqrt{2\pi\sigma^2}} exp\left(-\frac{1}{2\sigma^2}(1-w^T x)^2\right) = 0.5$$

On trace cette droite et on obtient les résultats suivants :



Régression linéaire pour les dataset Test



Régression linéaire pour les dataset Train

2.4 Application

Question a

On obtient les erreurs suivantes pour les 3 différents modèles :

Dataset	LDA	Logistique	Linéaire
Test A	1 %	1 %	6 %
Test B	4.5 %	3.5 %	9.5 %
Test C	4 %	4.67 %	7 %
Train A	0 %	0 %	4 %
Train B	2 %	1 %	7 %
Train C	2.67 %	3 %	6.67 %

Question b

D'après la question a), on remarque que la régression linéaire est la moins performante de toute, faisant même des erreurs sur le jeu de données A qui est pourtant assez simple à classifier.

Dataset A

Pour le premier Dataset, à l'exception de la régression linéaire, tous les modèles se comportent de la même manière : ils apprennent parfaitement et font une erreur très faible de l'ordre de 1 pourcent. Ce jeu de données est assez simple et linéairement séparable, il n'y a rien d'étonnant à ceux que le modèle cluster correctement les données. Les résultats sont similaires à l'entrainement et en phase de test.

Dataset B

Sur le second Dataset on remarque que l'estimateur le plus efficace est la régression logistique qui ne fait que 3.5 pourcent d'erreur sur le test quand LDA en fait 4.5 et la régression linéaire 9.5. En phase d'entraînement les erreurs étaient relativement moins importantes LDA faisait 2 pourcent d'erreur la régression logistique en faisait 1 et la linéaire 7. Sur ce Dataset les données ne sont plus linéairement séparables, ce qui explique une telle différence avec le Dataset A.

Dataset C

Sur le troisième Dataset, le modèle le plus efficace est la LDA qui se trompe de 4 pourcent dans le test (2.67 dans le train) alors que la régression logistique et linéaire se trompent de 4.67 et 7 pourcent respectivement (et de 3 et 6.67 respectivement dans le train). Ici, à l'instar du second Dataset, les données ne sont pas séparables et donc les estimateurs ne peuvent que se tromper. Ceci dit on remarque que l'écart entre la phase d'entraînement et de test est moindre que pour le Dataset B. De plus, de manière générale, la régression linéaire est la moins performante. Ceci vient du fait qu'elle "borne" les données tel que y = 1 deux fois (supérieurement et inférieurement). Elle augmente donc les chances de se tromper.

2.5 QDA model

D'après le modèle LDA nous remarquons que les paramètres autre que Σ_0 et Σ_1 gardent les mêmes estimateurs respectivement.

La log-vraisemblance du modèle QDA est similaire au modèle LDA à quelques détails près :

$$l(\Theta) = \sum_{i=1}^{n} -\frac{y_i}{2} Tr\left((x_i-\mu_1)(x_i-\mu_1)^T\Sigma_1^{-1}\right) + \frac{y_i}{2} log\left(|\Sigma^{-1}|\right) - \frac{1-y_i}{2} Tr\left((x_i-\mu_0)(x_i-\mu_0)^T\Sigma_0^{-1}\right) + \frac{1-y_i}{2} log\left(|\Sigma_0^{-1}|\right) - log(2\pi)$$

On remarque très vite que les estimations de π, μ_0 et μ_1 vont être les mêmes.
Par un raisonnement similaire à LDA, on obtient

$$\widehat{\Sigma_0} = \sum_{i=1}^{n} -\frac{1-y_i}{2}(x_i - \mu_0)(x_i - \mu_0)^T$$
$$\widehat{\Sigma_1} = \sum_{i=1}^{n} -\frac{y_i}{2}(x_i - \mu_1)(x_i - \mu_1)^T$$

Question a

•Pour le jeu A

$\pi_A = 0.48$, $\mu_{0_A} = [10.73, 10.93]$, $\mu_{1_A} = [11.032, 5.99]$, $\Sigma_{0_A} = [[0.46, 0.098], [0.098, 0.713]]$

et $\Sigma_{1_A} = [[0.72, 0.18], [0.18, 0.93]]$

•Pour le jeu B

$\pi_B = 0.55$, $\mu_{0_B} = [10.58, 11.17]$, $\mu_{1_B} = [11.24, 6.09]$, $\Sigma_{0_A} = [[0.76, 0.053], [0.053, 1.107]]$

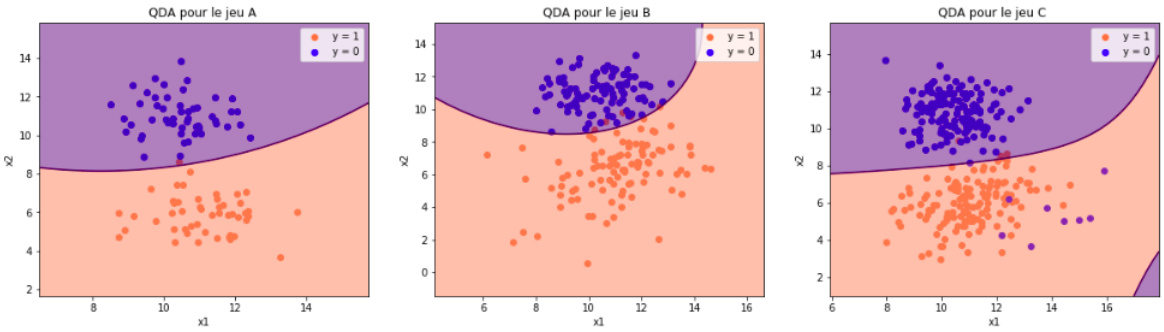
et $\Sigma_{1_B} = [[2.36, 1.23], [1.23, 2.84]]$

•Pour le jeu C

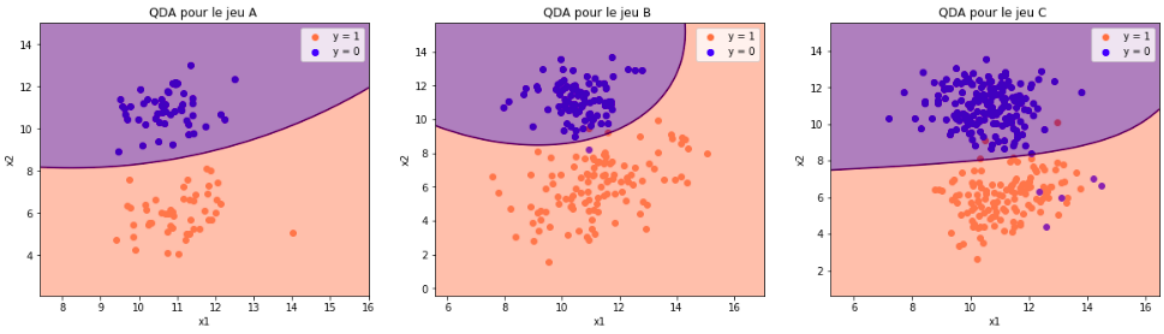
$\pi_C = 0.4167$, $\mu_{0_C} = [10.61, 10.83]$, $\mu_{1_C} = [11.18, 6.04]$, $\Sigma_{0_C} = [[1.28, -0.43], [-0.43, 1.82]]$

et $\Sigma_{1_C} = [[1.26, 0.45], [0.45, 1.44]]$

Question b
Toujours par un raisonnement similaire, on a trace la droite $\mathbb{P}(y = 1|x) = 0.5$
et on obtient la classification suivante :



QDA pour les dataset Test



QDA pour les dataset Train

Question c On obitent les erreurs suivantes :

Dataset	Training	Test
A	0 %	1 %
B	1,5 %	2.5 %
C	2,67%	4.3 %

question d

Premièrement, on remarque que QDA surpasse LDA pour tous les datasets. Ce qui est normal puisque apprendre sur deux tailles différentes d'ellipsoïdes (Σ_1 et Σ_2) est plus général qu'apprendre avec une taille commune (Σ). De plus la séparation n'est plus linéaire et permet "d'attraper" des points qui ne sont pas accessibles si la séparation est linéaire. Cette particularité vient du fait que l'équation $ax + b = 0$ de LDA devient une équation du second degré. Ainsi cette méthode est quasi optimale à chaque fois (cf le tableau ci-dessus).