Exercice 2 Régression Logistique

2.1 Implémentation

**Régression Logistique Classique**

**Apprentissage**

Pour ce problème de régression logistique classique binaire, nous avons 2 classes (). En donnant un individu x, nous supposons que la probabilité

que x appartenant à la classe et celle que x appartenant à la classe

sont :

Dans notre jeu de données, x est un vecteur de 2 dimensions, écrit comme . est le vecteur des paramètres du X, écrit comme . Nous définissons aussi une variable aléatoire T comme étiquète, qui est comme :

Actuellement, nous avons un échantillon des individus , et nous utilisions la méthode de maximum vraisemblance à estimer . La formule est comme :

Et puis, nous prenons le Log :

Le gradient de la log-vraisemblance s’écrit :

= où

Comme est un vecteur de p+1 dimension, nous ne pouvons pas résoudre ce système directement. Par conséquent, nous utilisons l’algorithme de *Newton-Raphson* à résoudre.

Pour cela, nous avons la formule comme :

, où

,

Dans chaque itération, nous renouvelons P et jusqu’à est convergent. Comme cela, peut avoir du maximum. Du coup, ce est le vecteur des paramètres optimaux.

**Classement**

Après la procédure d’apprentissage, nous avons un vecteur qui maximise le .

Dans le classement, nous pouvons calculer les probabilités avec ce

Par résultat, pour chaque individu dans l’ensemble du test, nous calculons leur probabilités et . D’après le règle de Bayes, si nous supposons que appartient à la classe . Sinon, nous supposons que appartient à la classe.

Comme cela, nous avons un classifieur qui peut faire une discrimination entre deux classes.

**Régression Logistique Quadratique**

Le principe de la régression logistique quadratique est la même comme une régression logistique classique. La différence est que la régression logistique classique ne prend en compte que des facteurs linéaires. Par exemple, dans notre jeu de données, nous avons le vecteur X de 2 dimensions, . Dans la régression logistique classique, nous ne prend en compte que les facteurs linéaires. Cela nous donne . Du coup, la frontière de décision de la régression logistique classique est un droit pour 2 dimensions.

Cependant, la régression logistique quadratique prend également des facteurs non-linéaire, et cela nous donne . Par conséquent, la frontière de décision de la régression logistique quadratique est souvent une courbe.

L’avantage de la régression logistique quadratique est qu’il donne une classification plus fiable et précise que celle de classique, car elle prend en compte un modèle plus complexe. Par contre, elle n’est pas assez robuste comme celle de classique.

2.2 Test sur données simulées

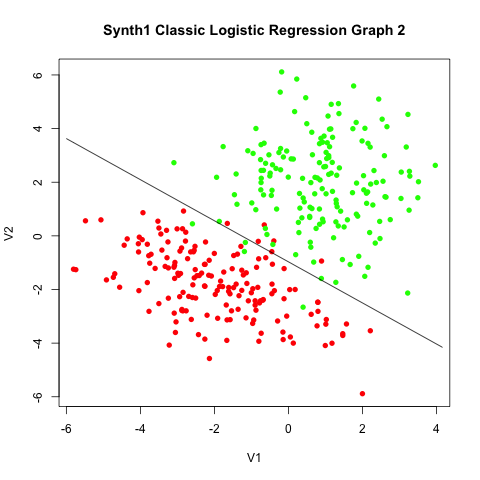
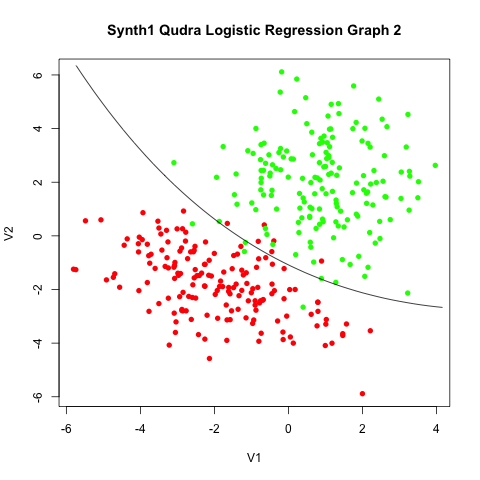
Dans cette question, nous appliquons nos deux modèles de régression logistique sur les trois jeux de données. Pour chaque jeu de données, nous répétons 20 fois avec des séparations de données différentes.

**Jeu de données : Synth1-1000**

Pour le jeu de données Synth1-1000, nous lançons le calcul et nous obtenons le taux d’erreur moyen comme ci-dessous :

|  |  |
| --- | --- |
| Le taux d’erreur pour modèle classique | Le taux d’erreur pour modèle quadratique |
| 0.02612613 | 0.02627628 |

Nous prenons également les frontières de décision sur une répétition de calcul :



D’après les taux d’erreur obtenus, nous trouvons que les deux modèles sont presque même performant pour ce jeu de données.

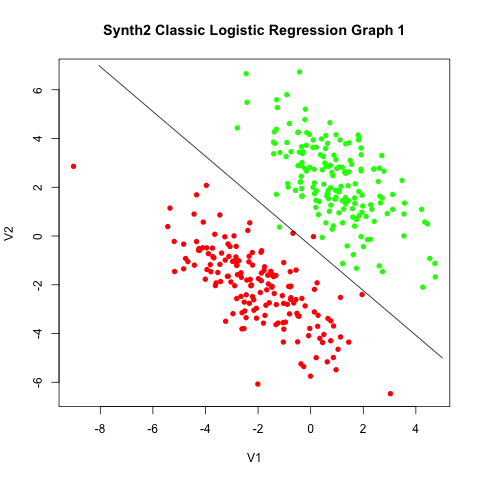
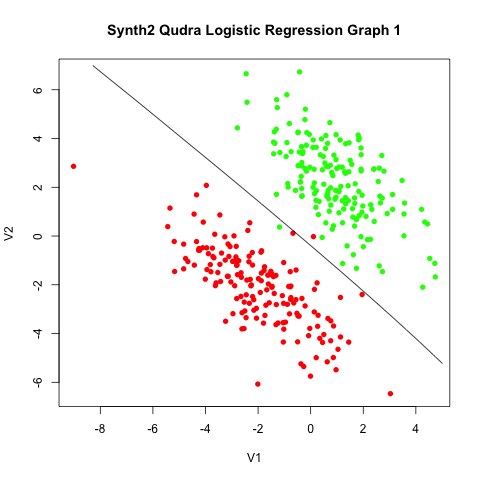
Selon le graphe, nous constatons que le jeu de donnée Synth1-1000 est généralement séparable, sauf qu’il y a quelques points mélangés près de la frontière de deux classes. Du coup, les deux modèles ont presque des mêmes performances.

**Jeu de données : Synth2-1000**

Ensuite, nous analysons le jeu de données Synth2-1000, cela nous donne :

|  |  |
| --- | --- |
| Le taux d’erreur pour modèle classique | Le taux d’erreur pour modèle quadratique |
| 0.01242515 | 0.01332335 |

Nous prenons également les frontières d’une répétition :



Par rapport au jeu de données Synth1-1000, le jeu de données Synth2-1000 est plus séparé. Par conséquent, les taux d’erreur sont plus bas pour Synth2-1000.

D’ailleurs, les taux d’erreur pour les deux modèles sont également similaires.

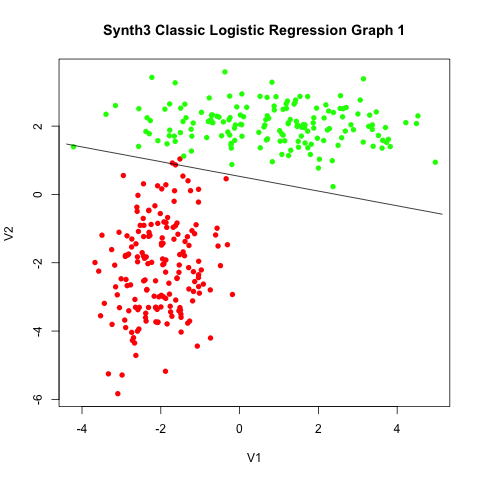
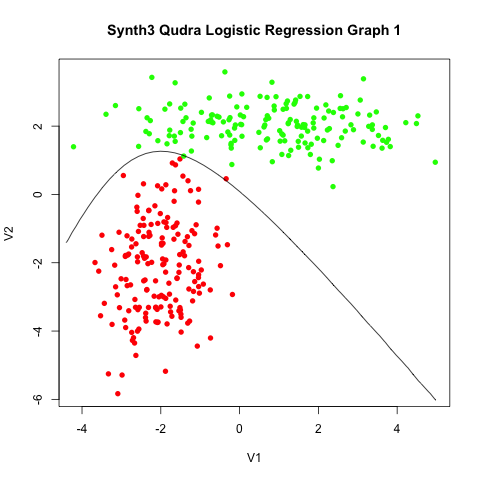
Selon les graphes, nous constatons que la frontière de décision de régression logistique quadratique est un peu près comme un droit. (Cela veut dire que les facteurs non-linéaires sont très faibles dans ce jeu de données.)

**Jeu de données : Synth3-1000**

En finale, nous regardons le jeu de données Synth3-1000, cela nous donne :

|  |  |
| --- | --- |
| Le taux d’erreur pour modèle classique | Le taux d’erreur pour modèle quadratique |
| 0.02117117 | 0.01576577 |

Nous avons également les frontières de décision pour une répétition :



Pour le jeu de données synth3-1000, nous remarquons le taux d’erreur de régression logistique quadratique est plus faible que ceux de régression logistique linéaire.

Nous regardons ensuite les frontières de décision. Nous trouvons que les deux classes n’est pas parfaitement linéairement séparables. Par conséquent, la régression logistique quadratique est meilleure que celle de classique.