Projet Quatrième Année

Étude de l'analyse discriminante linéaire pour la prédiction de défaut d'emprunt.

Majeure Ingénierie Financière et Statistique

Noélie DEBS
Alexandre DELACHE
Guillaume DURIEU
Matthieu MERINIS

Monsieur OSSONCE

Mai 2019

Table des matières

1 Remerciements

Nous tenons tout d'abord à remercier Mr Ossonce pour son soutien et son aide tout au long de l'élaboration de notre projet.

Nous tenons également à remercier Mr Marie et Mme Halconruy pour les connaissances qu'ils nous ont transmis à travers leurs cours et pour leurs réponses à nos questions.

2 Introduction

L'objectif du projet est, d'abord, l'étude bibliographique de la LDA, qui permettra d'en saisir les fondements statistiques et, à travers eux, les hypothèses sous-jacentes à l'utilisation de la LDA. La LDA sera comparée, dans un premier lieu à un autre outil de classication : la régression logistique. Nous expliquerons d'abord ce qu'est la régression logistique, puis nous nous intéresserons à la LDA.

Nous utilisons la technique du Machine Learning pour travailler sur nos bases de données. Le Machine Learning ou apprentissage automatique est l'art de construire des systèmes pouvant apprendre à partir de données. Cette technique est très pratique pour les problèmes complexes pour lesquels il n'existe pas de solutions algorithmiques, ou pour remplacer de longues listes de règles qu'il faudrait faire évoluer à la main ou encore de construire des systèmes devant s'adapter à des environnements différents. Il existe pleins de types d'apprentissage automatique mais nous utilisons un apprentissage supervisé dans ce projet avec la classification. Dans un apprentissage supervisé, les données d'entraînement que nous fournissons à l'algorithme comportent la ou les solutions désirées. Le but de cette méthode est que l'algorithme puisse apprendre en comparant sa sortie réelle avec les sorties enseignées pour trouver des erreurs et modifier le modèle en conséquence.

Une première implantation en Python sera mise en œuvre sur un jeu de données simple. Par la suite, une base de données comportant un historique de défaut de paiement pour les emprunteurs sera explorée. Après un travail important de préparation des données, la LDA sera utilisée afin d'effectuer des prédictions de défaut d'emprunt avec la bibliothèque sklearn. Enfin, nous essaierons de coder la LDA en utilisans la bibliothéque numpy notamment mais sans utiliser la bibliothéque sklearn.

Tout cela nous permettra d'établir un modèle de classification des demandeurs d'emprunt en fonction de différents paramètres.

3 La régression logistique

3.1 Introduction

La régression logistique, aussi appelée régression binomiale, est une analyse statistique permettant de construire un modèle de prédiction des données. Elle permet de caractériser les relations entre une variable dépendante Y et des variables explicatives $(X_1, ...X_n)$. La variable dépendante est une variable qualitative tandis que les variables explicatives peuvent être quantitatives ou qualitatives. La régression logistique consiste à modéliser la distribution des probabilités afin que Y se réalise.

3.2 Définition mathématique

3.2.1 Hypothèse

Pour rendre calculable la quantité $P(Y = y_k|X)$, il faut introduire une hypothèse fondamentale :

$$\ln \frac{\mathbb{P}(X|Y=1)}{\mathbb{P}(X|Y=0)} = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n \tag{1}$$

Contrairement à la LDA qui émet une hypothèse sur les distributions conditionnelles respectives de $\mathbb{P}(X|Y = 1)$ et $\mathbb{P}(X|Y = 0)$, la régression logistique est qualifiée de méthode semi-paramétrique car l'hypothèse porte uniquement sur le rapport linéaire de ces probabilités. Elle est moins restrictive et donc son champ d'action est plus large.

En pratique, cette hypothèse est rarement vérifiée mais le modèle est assez robuste pour être utilisé.

3.2.2 Régression logistique simple

Le modèle de la régression logistique simple se définit de la façon suivante :

$$p(X) = \mathbb{P}(Y = 1|X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}} = \frac{1}{1 + e^{-\beta_0+\beta_1 X_1}} \tag{2}$$

Les coefficients β_i représentent la combinaison linéaire du prédicteur et de la constante. P(Y|X) représente la probabilité que Y se produise si on prend en compte X. Avec la régression logistique, la prédiction de défault sera comprise dans l'intervalle allant de 0 à 1. En effet, la fonction logistique produira toujours une courbe en forme de S (sigmoïde).

Le modèle de la régression logistique nous permet de définir le rapport de chances (odds ratio) :

$$\frac{p(X)}{1 - p(X)} = e^{(\beta_0+\beta_1 X)} \tag{3}$$

Cette quantité représente le rapport de chances. Il est défini entre 0 et $+\infty$. Si le rapport est proche de 0, cela signifie que la probabilité de défaut est faible

tandis que si le rapport de chance est supérieur à 1, le risque de défaut sera d'autant plus grand que le rapport est élevé.

Si on applique la fonction logarithme des deux cotés de cette équation, on obtient :

$$log(\frac{p(X)}{1-p(X)}) = logit(p) = \beta_0 + \beta_1 X_1 \tag{4}$$

On remarque que lorsque β_1 est positif, $p(X)$ augmente si X augmente. En revanche, $p(X)$ diminue si X augmente lorque β_1 est négatif. Il s'agit bien d'une régression car on veut montrer une relation de dépendance entre une variable à prédire et une série de variables prédictives.

L'estimation des coefficents β_0 et β_1 se fait à partir des données d'entraînement dont on dispose. Contrairement à la régression linéaire qui utilise la méthode des moindres carrés pour estimer ces paramètres, la méthode du maximum de vraisemblance est ici préférée. Cette méthode consiste à chercher des estimations pour β_0 et β_1 telle que la probabilité prédite $p(x_i)$ de défaut pour chaque individu corresponde au maximum du statut par défaut. Il suffit de trouver $\widehat{\beta_0}$ et $\widehat{\beta_1}$ tels que $p(X)$ donne un nombre proche de un pour tous les individus qui ont fait défaut, et un nombre proche de zéro pour tous les individus qui ne l'ont pas fait. Cette méthode peut être formalisée à l'aide de la fonction de vraisemblance où $p(x_i) = \mathbb{P}(Y = 1|X = x_i)$

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} p(x_i)^{y_i} \times \prod_{i=1}^{n} (1 - p(x_i))^{1-y_i} \tag{5}$$

Cela vient du fait que :

$$p(y_i, \beta) = \mathbb{P}(Y = y_i|X = x_i, \beta = \begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix})$$

avec $p(x_i)$ pour $y_i = 1$ et $1 - p(x_i)$ pour $y_i = 0$

On détermine ensuite les estimateurs du maximum de vraisemblance via le log-vraisemblance :

$$l_n(\beta_0, \beta_1) = \sum_{i=1}^{n} y_i log p(x_i) + \sum_{i=1}^{n} (1 - y_i) log(1 - p(x_i))$$

On obtient ainsi les β en annulant $\frac{\partial l_n(\beta_0,\beta_1)}{\partial(\beta_0,\beta_1)}$. Ainsi :

$$\begin{cases} \dfrac{\partial l_n(\beta_0,\beta_1)}{\partial \beta_0} = \displaystyle\sum_{i=1}^{n} y_i - p(x_i) \\ \qquad\qquad\qquad = \displaystyle\sum_{i=1}^{n} (y_i - \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}) = 0 \\ \dfrac{\partial l_n(\beta_0,\beta_1)}{\partial \beta_1} = \displaystyle\sum_{i=1}^{n} x_i(y_i - p(x_i)) \\ \qquad\qquad\qquad = \displaystyle\sum_{i=1}^{n} x_i(y_i - \frac{e^{\beta_0+\beta_1 x_i}}{1+e^{\beta_0+\beta_1 x_i}}) = 0 \end{cases} \tag{6}$$

En résolvant ces deux équations, on trouve nos paramètres $\widehat{\beta_0}$ et $\widehat{\beta_1}$ qui sont choisis afin de maximiser la fonction de vraisemblance. On les utilise ensuite dans le modèle de la régression logistique de départ et nous obtenons la probabilité recherchée.

3.2.3 Généralisation : Régression logistique multiple

Par analogie, le modèle de la régression logistique multiple s'écrit de la façon suivante :

$$p(X) = \mathbb{P}(Y|X_1,...,X_n) = \frac{e^{\beta_0+\beta_1 X_1+...+\beta n X_n}}{1+e^{\beta_0+\beta_1 X_1+...+\beta_n X_n}} = \frac{1}{1+e^{-\beta_0+\beta_1 X_1+...+\beta_n X_n}} \tag{7}$$

avec $(X_1,...,X_n)$, n prédicteurs.

Les coefficients β_i représentent les combinaisons linéaires des prédicteurs et des constantes. $\mathbb{P}(Y|X_1,...,X_n)$ représente la probabilité que Y se produise si on prend en compte les variables prédictives $(X_1,...,X_n)$.

$$log(\frac{p(X)}{1-p(X)}) = logit(p) = \beta_0 + \beta_1 X_1 + ... + \beta_n X_n \tag{8}$$

Le logit permet d'obtenir un résultat compris entre $-\infty$ et $+\infty$ ce qui rend possible l'approximation par une droite réelle et donc plus facile à interpréter. La mise en oeuvre d'une régression logistique permet d'obtenir les coefficients β_i ainsi que leurs rapports de chances et leurs intervalles de confiance.

Là encore, on applique la méthode du maximum de vraisemblance afin d'estimer les coefficients β_i et ainsi ajuster le modèle. La fonction de vraisemblance reste la même :

$$L(\beta_0,...,\beta_n) = \prod_{i=1}^{n} p(x_i)^{y_i} \times \prod_{i=1}^{n} (1-p(x_i))^{1-y_i} \tag{9}$$

En utilisant la vraisemblance conditionnelle $\mathbb{P}(Y = k|X)$, on applique la

log-vraisemblance pour n observations :

$$l(\beta) = \sum_{i=1}^{n} logp(x_i, \beta) = \sum_{i=1}^{n} y_i logp(x_i) + \sum_{i=1}^{n} (1 - y_i) log(1 - p(xi))$$

avec $p(x_i, \beta) = \mathbb{P}(Y = y_i, X = x_i, \beta) = \frac{e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n}}{1 + e^{\beta_0 + \beta_1 X_1 + \ldots + \beta_n X_n}}$

Afin de maximiser le logarithme de vraisemblance et ainsi trouver les β_i, il faut annuler ses dérivées partielles :

$$\frac{\partial l_n(\beta)}{\partial \beta} = \sum_{i=1}^{n} x_i (y_i - p(x_i; \beta)) = 0 \tag{10}$$

Pour résoudre cette équation du dessus (appelée équation de score), il est préférable de la dériver une seconde fois pour trouver plus facilement nos paramètres $\widehat{\beta_i}$. Comme précédemment, on les utilisera par la suite dans le modèle de la régression logistique de départ et obtenir la probabilité recherchée.

En conclusion, la régression logistique est préférable dans les applications où il n'est pas raisonnable de supposer que les variables indépendantes sont normalement distribuées, ce qui est une hypothèse fondamentale de la méthode LDA.

4 LDA : Analyse Discriminante Linéaire

L'analyse discriminante linéaire est un outil de classication qui a pour objectif la prédiction, pour un individu x, de l'appartenance à une classe prédéfinie. La classe qui obtient la probabilité la plus élevée est la classe de sortie. Une prédiction est alors faite.

La régression logistique implique la modélisation directe de $\mathbb{P}(Y = k|X = x)$ à l'aide de la fonction logistique vue en première partie pour le cas de deux classes et en général. Ici nous verrons une approche différente. On suppose que l'échantillon d'apprentissage est issu d'une population en K groupes $G_1, ..., G_k$ avec Y une variable aléatoire compris entre $(1, ..., K)$ et $X = (X_1, ..., X_p)$ un vecteur de variables aléatoires réelles. La variable à prédire Y est catégorielle et les variables prédictives sont à priori continues.

4.1 Théorème de Bayes

Soit le théorème de Bayes suivant :

$$\mathbb{P}(G_k|x) = \mathbb{P}(Y = k|X = x) = \frac{\pi_k \times f_k(x)}{\sum_{l=1}^{K} \pi_l \times f_l(x)}$$

avec :

- $\mathbb{P}(G_k|x) = \mathbb{P}(Y = k|X = x)$ la probabilité conditionnelle appelée la probabilité à posteriori de G_k. Une probabilité a posteriori est la probabilité d'affectation des observations à des groupes d'après les données. Les probabilités à posteriori sont parfois qualifiées de scores et on affecte donc une nouvelle observation au groupe pour lequel le score est le plus grand.
- $\pi_k = \mathbb{P}(Y = k)$ la probabilité globale ou à priori qu'une observation choisie au hasard provienne de la kième-classe avec $\sum_{k=1}^{K} \pi_k = 1$. Une probabilité à priori est la probabilité d'affectation d'une observation à un groupe avant la collecte des données.
- $f_k(x) = \mathbb{P}(X = x|Y = k)$ la fonction de densité de X pour une observation de la kième-classe.

On voit à travers ce théorème de Bayes que son dénominateur est indépendant de k. Il nous suffit donc de maximiser $\pi_k f_k(x)$. Nous allons donc supposer que $f_k(x)$ a une forme paramétrique et estimer les paramètres sur l'échantillon d'apprentissage. Avec la LDA, $f_k(x)$ est une densité $\mathrm{N}(u_k, \sum_k)$.

4.2 LDA pour un prédicteur

Nous allons dans un premier temps, voir comment marche la LDA pour p=1 prédicteur. Le but est d'estimer $f_k(x)$ afin de l'intégrer dans le théorème de Bayes pour enfin estimer $\mathbb{P}(Y = k|X = x)$. Au final, nous classerons une observation dans la classe où $\mathbb{P}(Y = k|X = x)$ est le plus grand. Pour estimer

$f_k(x)$, nous supposons qu'elle est gaussienne c'est à dire qu'elle suit une loi normale de dimension 1.

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} exp(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2)$$

avec μ_k la moyenne et σ_k^2 la variance de la kième classe.

Or avec la LDA, nous admettons que les variances sont communes c'est à dire $\sigma_1^2 = ... = \sigma_k^2$. Nous noterons maintenant σ^2 la variance partagée par toutes les classes. En introduisant l'équation de $f_k(x)$ dans notre théorème de Bayes, nous obtenons :

$$\begin{aligned}\mathbb{P}(Y=k|X=x) &= \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} exp(-\frac{1}{2\sigma^2}(x-\mu_l k)^2)} \\ &= \frac{\pi_k exp(-\frac{1}{2\sigma^2}(x-\mu_k)^2)}{\sum_{l=1}^{K} \pi_l exp(-\frac{1}{2\sigma^2}(x-\mu_l)^2)}\end{aligned}$$

Le classificateur de Bayes implique l'affectation d'une observation $X = x$ à une classe pour laquelle $\mathbb{P}(Y = k|X = x)$ est le plus grand. Le dénominateur de l'équation ci dessus ne dépend pas de k, nous pouvons donc le négliger. En passant en logarithme, nous obtenons :

$$\delta_k(x) = x\frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + log(\pi_k)$$

On cherche $\delta_k(x)$ le plus grand.

Prenons un exemple où nous avons $K = 2$ classes et $\pi_1 = \pi_2$. Le classificateur de Bayes affecte une observation à la classe 1 si $2x(\mu_1 - \mu_2) > \mu_1^2 - \mu_2^2$ et à la classe 2 dans le cas inverse. Dans ce cas-ci, la limite de décision correspond au point où :

$$x = \frac{\mu_1^2 - \mu_2^2}{2(\mu_1 - \mu_2)} = \frac{\mu_1 + \mu_2}{2}$$

Il ne nous reste plus qu'à estimer les paramètres π_k, μ_k et σ^2 que nous intégrerons dans l'équation de $\delta_k(x)$.

- $\widehat{\pi_k} = \frac{n_k}{n}$ où n_k est le nombre d'observations d'apprentissage de la kième classe et n le nombre total d'observations d'apprentissage.
- $\widehat{\mu_k} = \frac{1}{n_k}\sum_{i=k} x_i$ est la moyenne de toutes les observations d'entraînement de la kième classe. Elle représente également l'espérance car nous sommes dans le cas d'une loi normale et E(X)=μ_k
- $\widehat{\sigma^2} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i=k}(x_i - \widehat{\mu_k})^2$ est la moyenne pondérée des variances de l'échantillon pour chacune des classes.

Le classificteur de la LDA nous donne donc, à partir des estimations faites sur les paramètres :

$$\widehat{\delta}_k(x) = x\frac{\widehat{\mu}_k}{\widehat{\sigma}^2} - \frac{\widehat{\mu}_k^2}{2\widehat{\sigma}^2} + log(\widehat{\pi}_k)$$

Le but étant, là encore, d'attribuer une observation $X = x$ telle que $\widehat{\delta}_k(x)$ soit le plus grand. Le mot linéaire dans la LDA vient du fait que les fonctions discriminantes $\widehat{\delta}_k(x)$ sont des fonctions linéaires de x.

4.3 Cas Général Gaussien

Dans le cas général pour la LDA, nous allons modéliser chaque densité de classe grâce aux densités Gaussiennes. X suit une loi normale multidimensionnelle $X \sim \mathrm{N}(u_k, \sum_k)$.

$$f_k(x) = \mathbb{P}(X = x|Y = k) = \frac{1}{(2\pi)^{p/2}|\sum_k|^{1/2}} \times e^{\frac{-1}{2}(x-\mu_k)^T|\sum_k|^{-1}(x-\mu_k)}$$

où $|\sum_k|$ est la matrice de covariance conditionnellement à k et $E(X) = \mu_k$ un vecteur moyen.

La LDA apparaît dans le cas particulier où nous supposons que les classes ont une matrice de covariance communes (hypothèse d'homoscédasticité) c'est à dire $\sum_k = \sum$ pour tout k.
Pour cela, appliquons le logarithme au théorème de Bayes qui nous permettra de comparer deux classes k et l.

$$\begin{aligned}\log\frac{\mathbb{P}(Y=k|X=x)}{\mathbb{P}(Y=l|X=x)} &= \log\frac{f_k(x)}{f_l(x)} + \log\frac{\pi_k}{\pi_l} \\ &= \log\frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k+\mu_l)^T\sum^{-1}(\mu_k-\mu_l) \\ &\quad + x^T\sum^{-1}(\mu_k-\mu_l)\end{aligned}$$

On remarque que les facteurs de normalisation ainsi que la partie quadratique des exposants s'annulent du fait des matrices de covariance égales. Cette fonction logarithme implique que la frontière de décision entre les classes k et l (c'est à dire $\mathbb{P}(Y = k|X = x) = \mathbb{P}(Y = l|X = x)$) est linéaire par rapport à x dans un hyperplan à p-dimensions.

L'objectif étant de déterminer le maximum de la probabilité à posteriori d'affectation, nous négligeons tout ce qui ne dépend pas de k. Nous avons donc :

$$\begin{aligned}\log(\pi_k f_k(x)) &= \log(\pi_k) + \log(f_k(x)) \\ &= \log\pi_k - \frac{1}{2}\mu_k^T\sum^{-1}\mu_k + x^T\sum^{-1}\mu_k\end{aligned}$$

On peut en sortir la règle de décision : $Y(x) = argmax_k \delta_k(x)$ où :

$$\delta_k(x) = x^T \sum^{-1} \mu_k - \frac{1}{2} \mu_k^T \sum^{-1} \mu_k + log\pi_k$$

$\delta_k(x)$ est la fonction linéaire discriminante du groupe k (ou fonction linéaire de classement). Chaque fonction linéaire discriminante définie une fonction score et une nouvelle observation sera affectée pour le groupe où le score sera le plus grand.
Avec tous ces précédents calculs, nous devons estimer les paramètres des distributions Gaussiennes à l'aide de nos données d'entraînement.

- $\widehat{\pi_k} = \frac{n_k}{n}$
- $\widehat{\mu_k} = \sum_{i=k} \frac{x_i}{n_k}$
- $\widehat{\sum} = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i=k} (x_i - \widehat{\mu_k})(x_i - \widehat{\mu_k})^T$

En conclusion, la méthode de classification LDA résulte de la supposition que les observations dans chaque classe proviennent d'une distribution normale ou gaussienne avec un vecteur moyen spécifique à la classe et une variance (ou matrice de covariance) commune. Ces paramètres ainsi estimées seront intégrés dans la formule du classificateur de Bayes et une prédiction pourra alors être faite.

4.4 QDA : Analyse Discrimante Quadratique

La QDA est très proche de la LDA dans le sens où l'hypothèse sur la loi normale est la même et que les observations de chaque classe sont tirées d'une distribution gaussienne. Mais contrairement à la LDA, l'hypothèse sur le fait que les covariances de chaque classe sont les mêmes n'est pas vérifiée en QDA. Nous avons toujours des gaussiennes, mais cette fois ci, chaque classe a sa propre matrice de covariance (cas d'hétéroscédastique). Une observation de la kième classe est donc de la forme $X \sim \mathrm{N}(u_k, \sum_k)$ où $\sum_k$ est la matrice de covariance pour la kième classe. Avec cette hypothèse, le classificateur de Bayes assigne une observation X=x à la classe pour laquelle la fonction quadratique discriminante du groupe k est la plus grande.

$$\begin{aligned} \delta_k(x) &= -\frac{1}{2}(x - \mu_k)^T \sum_k^{-1} (x - \mu_k) + log\pi_k \\ &= -\tfrac{1}{2} x^T \textstyle\sum_k^{-1} x + x^T \sum_k^{-1} \mu_k - \tfrac{1}{2} \mu_k^T \sum_k^{-1} \mu_k + log\pi_k \end{aligned}$$

Commme les $\sum_k$ sont différents, les termes quadratiques apparaissent. Ainsi le classificateur de la QDA implique d'intégrer les estimations $\sum_k, \mu_k, \pi_k$ dans l'équation du dessus et d'affecter une observation $X = x$ à laquelle cette quantité est la plus grande. Contrairement au classificateur de Bayes dans la LDA, la

quantité x apparaît comme une fonction quadratique. La QDA suppose donc une limite de décision quadratique ce qui permettra de modéliser avec précision un plus grand nombre de problèmes que les modèles linéaires.

5 Synthèse : Comparaison Régression Logistique et LDA

Pourquoi utiliser l'analyse discriminante linéaire ?

1. Lorsque les classes sont bien séparées, l'estimation des paramètres de la régression logistique devient instable. La LDA ne souffre pas de ce problème.

2. Lorsque n est petit et la distribution de X est à peu près gaussienne dans chaque classe, la LDA est plus stable que la régression logistique.

3. La LDA est populaire lorsqu'il y a plus de deux conditions pour Y car elle permet de projeter les données dans des plans séparant les groupes. La régression logistique est populaire, quant à elle, pour la classification, surtout quand il n'y a que 2 classes.

L'analyse de la fonction discriminante reste néammoins très similaire à la régression logistique. En effet, les deux modèles peuvent être utilisés pour répondre aux mêmes questions de recherche.

Pour un problème à deux classes, la LDA a la même forme que la régression logistique. Considérons deux classes avec $p = 1$ prédicteur. soit $p_1(x)$ et $p_2(x) = 1 - p_1(x)$, les probabilités que l'observation $X = x$ appartienne aux classes 1 et 2. La LDA nous donne donc :

$$log(\frac{p_1(x)}{1 - p_1(x)}) = log(\frac{p_1(x)}{p_2(x)}) = c_o + c_1(x)$$

avec c_o et c_1 les fonctions de μ_1, μ_2 et σ^2

Pour la régression logistique, ce serait :

$$log(\frac{p_1}{1 - p_1}) = \beta_0 + \beta_1(x)$$

Ces deux équations ci-dessus se ressemblent et sont toutes deux des fonctions linéaires en x. Ces deux méthodes produisent des limites de décision linéaires.

La seule différence est dans la méthode de l'estimation des paramètres :

- la régression logistique utilise des vraisemblances conditionnelles et ne modélise que $\mathbb{P}(Y = k|X)$. On parle d'apprentissage discriminant.

- la LDA calcule la moyenne et la variance (ou matrice de covariance) estimées à partir d'une loi normale. On parle d'apprentissage génératif.

La régression logistique ne comporte pas autant d'hypothèses et de restrictions que l'analyse discriminante. Cependant, lorsque les hypothèses de l'analyse discriminante sont remplies, il est plus puissant que la régression logistique. Contrairement à la régression logistique, l'analyse discriminante peut être utilisée avec des échantillons de petite taille. Il a été démontré que lorsque la taille

des échantillons est égale et que l'homogénéité de la variance / covariance est respectée, l'analyse discriminante est plus précise.

Tout ceci étant pris en compte, la régression logistique est devenue le choix commun, car les hypothèses de l'analyse discriminante sont rarement satisfaites.

6 Représentation géométrique de la LDA

6.1 Introduction

L'analyse discriminante linéaire (LDA) est le plus souvent utilisée comme technique de réduction de la dimensionnalité lors de l'étape de prétraitement pour les applications de classification de modèles et d'apprentissage automatique. L'objectif est de projeter un jeu de données sur un espace de dimension inférieure avec une bonne séparabilité de classes afin d'éviter les surajustements («malédiction de la dimensionnalité») et de réduire les coûts de calculs.

6.2 Les étapes

1. Calculer les matrices de variances (matrice de variances intra-classe et inter-classe).
2. Calculer les vecteurs propres (e_1,e_2,... ,e_d) et les valeurs propres correspondantes (λ_1,λ_2,. . . ,λ_d) pour la matrice de variance.
3. Trier les vecteurs propres en diminuant les valeurs propres et choisir k-vecteurs propres avec les plus grandes valeurs propres pour former une $d \times k$ matrice dimensionnelle W (où chaque colonne représente un vecteur propre).
4. Utiliser ce $d \times k$ matrice de vecteurs propres pour transformer les échantillons dans le nouveau sous-espace. Ceci peut être résumé par la multiplication de la matrice : $Y = X \times W$

6.3 Calcul des matrices de dispersion

Nous cherchons à :

- Maximiser la distance entre les centres de gravité
  Maximiser la variance inter-groupe
- Minimiser la distance entre les points d'une classe et leur centre de gravité
  Minimiser la variance intra-groupe

$$S = S_W + S_B$$

S est la matrice de varaiance-covariance totale
S_B est la matrice de variance-covariance inter (ou between)
S_W est la matrice de varaince-covariance intra (ou Within)

Nous allons maintenant calculer les deux matrices de dimensions 10x10 : la matrice de dispersion intra-classe et la matrice de dispersion inter-classe.

6.3.1 Calcul de la matrice intra-classe

$$S_W = \sum_{i=1}^{c} S_i$$

$$S_i = \sum_{i=1}^{n} (x - m_i)(x - m_i)^T$$

x est notre base de données pour la classe et m_i est la moyenne des données pour chaque classe.

6.3.2 Calcul de la matrice inter-classe

$$S_B = \sum_{i=1}^{c} N_i (m_i - m)(m_i - m)^T$$

m est la moyenne globale, m_i et N_i sont les moyennes et la taille respective de chaque classe.

6.4 Résoudre le problème généralisé des valeurs propres pour la matrice

Si nous exécutons la LDA pour la réduction de la dimensionnalité, les vecteurs propres sont importants car ils formeront les nouveaux axes de notre nouveau sous-espace de fonctionnalités. Les valeurs propres associées présentent un intérêt particulier puisqu'elles nous indiqueront à quel point les nouveaux «axes» sont «informatifs».

$$S = S_W + S_B$$

Soit u un axe et $S(u)$ la variance totale. On peut montrer que :

$$S(u) = u^t S u$$
$$S_W(u) = u^t S_W u$$
$$S_B(u) = u^t S_B u$$

Après projection sur l'axe u, la décomposition de la variance est conservée :

$$S(u) = S_W(u) + S_B(u)$$

Un axe u est discriminant si :

- la variance inter-classe projetée sur u est grande : $S_B(u)$ grand.
- la variance intra-classe projetée sur u est petite : $S_W(u)$ petit.

D'après la loi de fisher nous pouvons dire :

Soit u la solution du problème de discrimination :

$$u = argmax(\frac{S_B(u)}{S_W(u)})$$

On démontre que résoudre l'équation ci-dessus revient à résoudre :

$$S_W^{-1} S_B u = \lambda u$$

avec u l'ensemble des vecteurs propres associés.

Pour la résoudre on utilisera une fonction de python.

6.5 Vérification

$$Av = \lambda v$$

$$A = S_W^{-1} S_B$$

$$\lambda = ValeurPropre$$

$$v = VecteurPropre$$

6.6 Tri des vecteurs propres en diminuant les valeurs propres

Rappelez-vous de l'introduction que nous ne souhaitons pas seulement projeter les données dans un sous-espace améliorant la séparabilité des classes, mais également réduire la dimensionnalité de notre espace de fonctions (où les vecteurs propres formeront les axes de ce nouveau sous-espace de fonctions).

Cependant, les vecteurs propres ne définissent que les directions du nouvel axe, car ils ont tous la même longueur unitaire 1.

Donc, afin de décider quel (s) vecteur (s) propre (s) nous voulons abandonner pour notre sous-espace de dimension inférieure, nous devons examiner les valeurs propres correspondantes des vecteurs propres. Les vecteurs propres avec les valeurs propres les plus basses portent le moins d'informations sur la distribution des données, et ce sont ceux que nous voulons supprimer. L'approche commune consiste à classer les vecteurs propres de la valeur propre la plus élevée à la plus faible et à choisir les k vecteurs propres. Après avoir trié les eigenpairsen, il est maintenant temps de construire notre $kd - dimensionnelle$ de vecteurs propres W (ici 10 1 : basé sur la paire propre la plus informative) et réduisant ainsi l'espace initial des caractéristiques dimensions en un sous-espace à 1 dimension.

6.7 Transformation des données dans le nouveau sous-espace

Dans la dernière étape, nous utilisons le $k - dimension$ matrice de W que nous venons de calculer afin de transformer nos données dans le nouvel sous-espace.

$$Y = x \times W$$

.

où x est un $n \times d - dimensions$ de matrice représentant les données et Y est la transformée de n x $k - dimensions$. Y répresente notre nouvel espace.

7 Applications à des bases de données sur Scikit-Learn

7.1 Introduction

D'après le journal CNBC, 1 million d'étudiants américains, soit près de 22 % des emprunteurs, font défaut pour leur prêt étudiant chaque année. Ces défauts ne sont pas sans coût pour les institutions financières et se chiffrent en milliards de dollars chaque année. Mais comment se fait-il qu'il y ait autant de défauts chaque année ? Les institutions financières ne sont-elles pas assez rigoureuses lors du choix de leurs clients ?

Pour prévenir le risque de défaillance, les banques analysent scrupuleusement le dossier de chaque demandeur. Cela peut s'avérer une tache couteuse et chronophage. Elles se basent généralement sur des critères comme le revenu, le nombre de personnes à charge ou encore le montant des autres prêts que peut avoir la personne.

Nous pensons qu'il est possible d'améliorer la fiabilité des prédictions en utilisant deux outils : le Big Data et l'intelligence artificielle. En effet le Big Data permet d'obtenir facilement et en grand nombre des données sur les clients qui permettront d'entrainer différents algorithmes de Machines Learning. Nous nous intéresserons ici principalement à la régression logistique et l'analyse discriminante linéaire (LDA) pour faire de la classification.

En appliquant des algorithmes de Machine Learning à ce problème nous souhaitons améliorer la situation des institutions financières et des particuliers. D'une part, réduire le taux de défaillance permettrait d'améliorer la rentabilité des banques en diminuant les pertes. D'autre part, cela permettra d'accélérer le processus de décision, de faire baisser le prix des prêts et d'éviter que des personnes insolvables n'aggravent leur situation. Ainsi une amélioration des prédictions de défaut d'emprunt serait bénéfique à chacun.

Nous allons travailler en Python sur deux bases de données distinctes afin d'implémenter nos modèles dans des situations différentes.

7.2 Méthodologie

Une description détaillée et commentée des codes est disponible sur GitHub (cf. lien en annexe). Pour résumer les points clés de notre méthodologie, nous avons travaillé comme suit :

1. Chargement des données et librairies
2. Analyse de la forme et d'échantillons de la base

3. Répartition de la classe ciblée
4. (Fusion des bases si nécessaire)
5. Vérification de la consistance de la classe
6. Graphique détaillé de chaque colonne
7. Suppression automatique des valeurs aberrantes
8. Feature engineering
9. Suppression de colonnes comprenant trop de valeurs manquantes
10. Suppression des colonnes inutiles
11. Encodage
12. Séparation des données en jeu de test et d'entrainement (80/20)
13. Entrainement des modèles
14. Affichage des résultats

7.3 Base de données : Give Me Some Credit

7.3.1 La base de données

Cette base donnée provient du site Kaggle. Nous avons décidé de commencer notre analyse avec celle-ci car elle était relativement simple à analyser comparativement à la seconde. Elle est composée de 250 000 lignes et 12 colonnes. L'objectif ici est de prédire si un individu aura connu un retard de paiement de 90 jours ou plus durant la durée de l'emprunt. Pour cela nous avons les 11 features suivants :

Revolving Utilization Of Unsecured Lines, age, Number Of Time 30-59 Days Past Due Not Worse, Debt Ratio, Monthly Income, Number Of Open Credit Lines And Loans, Number Of Times 90 Days Late, Number Real Estate Loans Or Lines, Number Of Time 60-89 Days Past Due Not Worse, Number Of Dependents.

Il s'agit d'informations élémentaires comme l'âge, le ratio de dette, ou encore si la personne a déjà fait défaut dans le passé. Toutes les valeurs sont numériques et le peu de valeurs manquantes ainsi que le nombre restreint de paramètres nous a permis d'étudier facilement la base de données à l'aide de matrices de corrélations ou de graphiques croisés.

7.3.2 Résultats et conclusions

A la suite de notre deuxième étude sur la seconde base de données et l'entrainement de nos modèles nous avons obtenu les résultats suivants :

Modèle	Score
Régression Logistique	93.08 %
LDA	93.20 %
Random Forest Classifier	93.48 %
Decision Tree Classifier	89.50 %

Pour obtenir ces résultats nous avons utilisé la fonction .score() de Scikit Learn.

La LDA et la régression logistique donne des taux de succès acceptables au vu de la distribution des données. Au cours de notre étude et de nos essais successifs, nous avons remarqué que la régression logistique était lourdement pénalisée par les valeurs aberrantes. La LDA, elle, est plus tolérante pour ce genre de valeurs. La LDA et la Régression Logistique donnent de meilleurs résultats lorsque les valeurs sont redimensionnées.

7.4 Home Credit Default Risk

7.4.1 La base de données

La base de données Home Credit Default Risk est beaucoup plus complexe que la précédente. Premièrement elle est beaucoup plus grande. Elle est composée de 11 fichiers CVS comptants entre plusieurs centaines de milliers et des millions de lignes chacunes. Chaque base a plusieurs dizaines de colonnes. De plus, elle possède beaucoup de valeurs manquantes ce qui complique son traitement. Enfin, une partie des variables est de type 'object' ce qui nous oblige à effectuer des conversions pour les rendre compréhensibles par nos modèles.

7.4.2 Résultats et conclusions

A la suite de notre deuxième étude sur la seconde base de données et l'entrainement de nos modèles nous avons obtenu les résultats suivants :

Modèle	Score
Régression Logistique	93.90 %
LDA	93.77 %
Random Forest Classifier	98.79 %
Decision Tree Classifier	98.14 %

Nous avons rencontré de nombreux problèmes lors de notre travail sur cette base. En autres, le traitement des relations entre les variables catégoriques et numériques (cf. traitement 'OWN_CAR_AGE' du code commenté sur GitHub).

La grande taille de la base et sa fiabilité nous a permis d'obtenir de bons résultats sur nos modèles. Grâce à la préparation adéquate de la base nous avons atteint un score quasiment parfait en utilisant un Random Forest Classifier. En ce qui concerne la Régression Logistique et la LDA elles etaient moins adaptées à cause de la distribution des données et de l'encodage des variables catégorielles.

7.5 Conclusions de l'étude

Nous pouvons tirer plusieurs conclusions de nos travaux. Tout d'abord, la régression logistique et la LDA obtiennent toujours des scores très proches pour une même base de données. Cela coïncide avec la théorie puisque les 2 méthodes sont proches dans leur manière de calculer des prédictions de manière linéaire. De plus la régression logistique est globalement plus efficace lorsque la taille des données à traiter augmente.

Également, nous avons remarqué que la LDA fonctionne significativement mieux que la Régression Logistique lorsque la base possède des valeurs aberrantes. Cela ne se traduit pas dans nos résultats puisque nous avons traité la base pour éviter ce problème.

Sur la deuxième base de données, Home Credit Default Risk, on atteint un score quasiment parfait en utilisant le Random Forest Classifier. Ce modèle est comparativement meilleur lorsque la complexité de la base de données augmente.

Si nous avions davantage de données, et de meilleures qualités, c'est-à-dire sans valeurs manquantes, imprécises ou aberrantes, nous pourrions probablement atteindre un score proche de 100 %. Cependant il convient de rappeler que le caractère intrinsèquement aléatoire de ce problème fait qu'il restera toujours une faible marge d'erreur.

Un problème auquel nous avons dû faire face est la puissance de nos ordinateurs, particulièrement pour la seconde base de données. Nous n'avons pas été en mesure d'exploiter cette base à son plein potentiel à cause de sa taille. Elle était composée de plusieurs fichiers CSV comprenant chacuns plusieurs millions de lignes et des centaines de colonnes. Si nous avions voulu l'exploiter dans sa totalité nous aurions dû gérer plusieurs centaines de milliards de valeurs à la fois. Nous n'avons pas non plus pû faire fonctionner les fonctions de la première base sur la seconde base à cause de sa taille.

8 Conclusion Générale

La Régression Logistique et la LDA sont deux modèles puissants pour la prédiction.

Lorsque les hypothèses à leurs utilisations sont avérées, ils permettent d'obtenir d'excellents scores. Cependant nous avons pu voir que lorsque cela n'est pas le cas, il peut être plus intéressant d'utiliser des modèles plus robustes comme le Random Forest Classifier par exemple.

En ce qui concerne leurs implémentations, ces deux modèles sont rapides à entrainer puisqu'ils réduisent significativement les temps de calculs. L'augmentation du nombre de paramètres améliore la précision mais rend plus complexe et plus difficile la validité des hypothèses de la LDA et de la Régression Logistique.

On peut donc imaginer que dans un futur proche, l'émergence du Big Data et l'accès simplifié à une grande puissance de calculs devrait permettre d'obtenir des modèles aux prédictions qui frôlent la perfection.

9 Annexes et Références

- Lien GitHub pour les codes : https ://github.com/Matthieu64/Projet-LDA-2F
- An Introduction of Statistical Learning de Gareth James, Daniela Witten, Trevor Hastie et Robert Tibshiran
- The Elements of Statistical Learning de Trevor Hastie, Robert Tibshirani, Jerome Friedman
- Le cours de Monsieur Marie sur les estimateurs de paramètres
- Hands-On Machine Learning with Scikit-Learn
- TensorFlow - Aurélien Géron