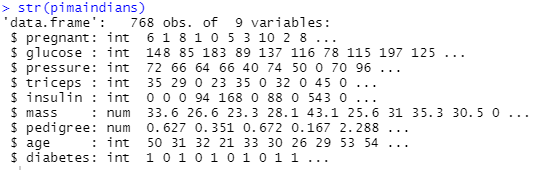
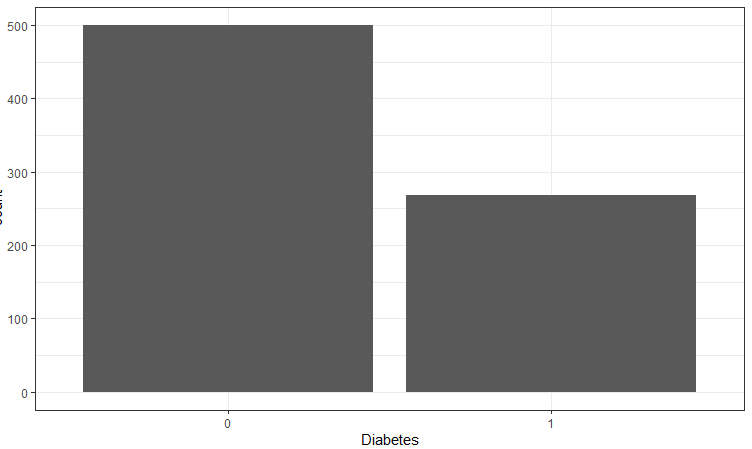
**Compte rendu TP noté**

Dans le cadre de ce TP, il nous a été proposé d’effectuer une analyse sur un des deux jeux de données qui nous a été fourni. Dans le premier jeu de données, nous avons différentes mesures physiologiques et médicales chez des femmes amérindiennes du peuple Pima et l’objectif est de savoir si on peut utiliser ces mesures pour modéliser voire prédire la présence/absence de diabètes chez ces femmes. Dans le deuxième jeu de données, nous avons différentes mesures et caractéristiques concernant des matériaux supraconducteurs et le but est de savoir si on peut modéliser la température critique en fonction de ces différentes mesures.

Dans les consignes, il nous a été demandé de répondre à la problématique d’un seul jeu de données et j’ai choisi d’étudier la présence ou l’absence de diabète chez les femmes du peuple Pima.

1. **Etude du jeu de données**

Le jeu de données sur le peuple Pima contient 768 observations et 9 colonnes y compris la colonne à prédire et elles sont toutes numériques.



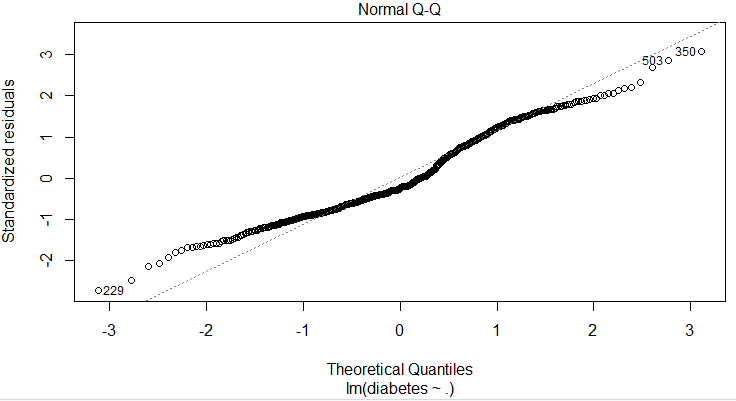
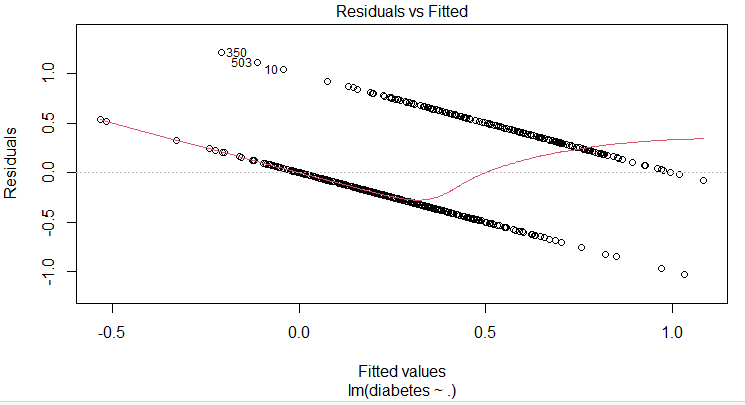
Le diagramme ci-contre montre que le jeu de données n’est pas équilibré car on dénombre bien plus de femmes saines que de femmes diabétiques. Cela pose problème car le modèle aura tendance à mieux prédire les cas sains que les cas de diabètes. L’idéal serait d’équilibré le jeu de données en y insérant autant de cas sains que de cas de diabètes.

Les valeurs observées sont relativement petites (< 500). Dans les cas où on aurait des valeurs assez grandes (de > 103), il aurait été judicieux d’appliquer une fonction logarithmique à la colonne correspondante (log10) pour diminuer l’ordre de grandeur des valeurs et ainsi facilité les calculs. Mais ce n’est pas notre cas donc on utilisera les valeurs tel qu’elles sont.

Dans la suite du TP, nous allons diviser le jeu de données en 2 parties : Une partie pour entrainer le modèle (70%) et une autre pour tester le modèle (30%). Cela permet d’augmenter les performances du modèle en le testant sur de nouvelles données qu’il n’a pas connues lors de l’apprentissage.

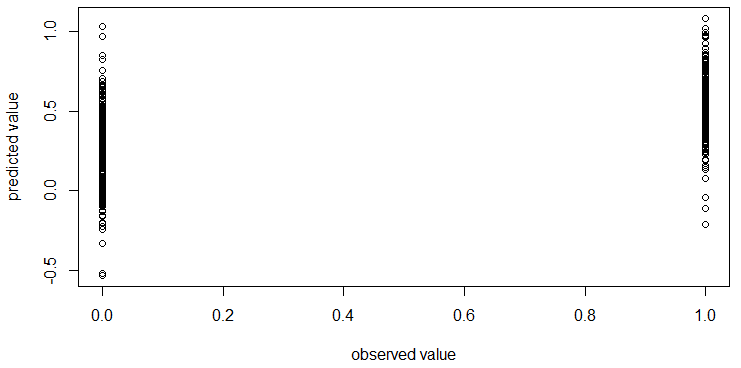
1. **Régression linéaire ou logistique ?**

Dans cette partie nous allons déterminer le type d’analyse à effectuer sur ce jeu de données pour répondre à la problématique. Supposons que la régression linéaire est adapté dans ce cas. Ainsi, nous avons les résultats suivant :

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG Le R2 est très faible (**R2 = 0.3015**).

La moyenne des résidus n’est pas nul et la variance n’est pas constante car elle dépend fortement de la valeur prédite (**pas d’homoscédasticité**)

Les résidus **ne sont pas Gaussien** car les quantiles ne sont pas alignés sur la droite **y=x**



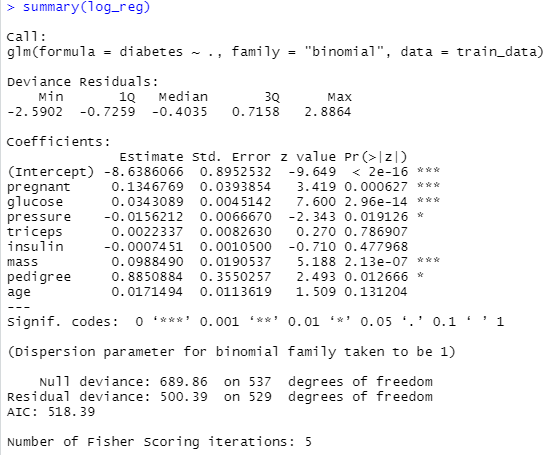
Les valeurs à prédire sont soit 0 ou 1 tandis que le modèle prédit des valeurs entre 0 et 1.

Au vu des résultats ci-dessus, on peut conclure que le modèle de régression linéaire n’est pas adapté à ce problème car le fait d’avoir une **R2** très faible signifie que l’écart entre les valeurs prédites et les valeurs réelles est assez grand et de plus le modèle doit prédire soit 0 ou 1 et non des valeurs comprises entre 0 et 1.

Dans la suite de ce TP, nous allons donc appliquer le modèle de régression logistique (modèle de classification) pour répondre à la problématique de ce jeu de données.

1. **Analyse des données**
2. **Application du modèle de classification**

* **Détermination du modèle**

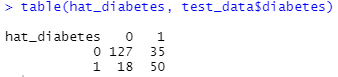
****En appliquant le modèle de régression logistique, on obtient les résultats suivants :

* **Deviance Residuals** : Les quantiles de la déviance résiduelle
* **Estimate**: estimations des coefficients **βj**de chaque variable de prédiction
* **Std. Error** : racine carrée de la variance empirique des coefficients **βj**
* **z value :** valeur de la statistique de test pour le test **βj** **= 0** qui suit une loi normal centré réduite
* **Pr(>|z|)** : p-value associée au test **βj** **= 0**
* **Signif. codes** : correspondances entre symboles et niveau de signiﬁcativité, c’est-à-dire jusqu’à quel niveau **α** (entre 0.001 et 0.1) peut-on rejeter l’hypothèse **H0**.
* **Null deviance**: valeur de la déviance nulle. Plus elle est faible, plus le modèle nul (modèle sans aucun prédicteur) explique bien la réponse et les prédicteurs n’apportent pas d’information supplémentaires pour expliquer la réponse.
* **Residual deviance :** valeur de la déviance résiduelle. Plus elle est faible, plus le modèle proposé explique bien la réponse et les prédicteurs sont utiles pour mieux expliquer la réponse.
* **AIC** : valeur de l’AIC du modèle. Plus c’est faible, meilleur est le modèle.
* **Number of Fisher Scoring iterations** : nombre d’itérations dans l’algorithme d’optimisation de la vraisemblance (pour trouver les estimations des coeﬃcients **βj**).

**Interprétation** :

* On remarque que les prédicteurs, **pregnant, glucose** et **mass** sont significatifs pour le modèle au niveau **α = 0,001** et les prédicteurs **pressure** et **pedigree** le sont aussi mais au niveau **α = 0,05.** Donc une première amélioration du modèle serait d’éliminer les prédicteurs qui ne sont pas significatif pour le modèle (**triceps, insulin et age**).
* La déviance résiduelle est inférieure à la déviance nulle ce qui veut dire que les prédicteurs dont utiles pour le modèle.
* **Test du modèle**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGEn appliquant le modèle ci-dessus aux données de test (en considérant un **threshold de 50%**), on obtient les résultats suivant :

**C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG****Taux d’erreur =>**

**Précision =>**

**Matrice de confusion**

**C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG**

**<= Sensibilité Spécificité =>**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

**<= taux de faux négatif taux de faux positif =>**

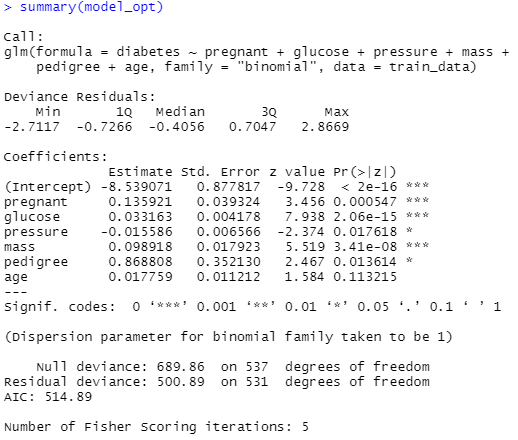
**Interprétation :**

Le modèle avec tous les prédicteurs prédit correctement la présence ou l’absence de diabètes avec une précision de **76,9%.** Par contre, il prédit mieux l’absence de diabète (à **87,5%**) que sa présence (à **58,8%**). Ce qui signifie qu’on sera plus confiant lorsque le modèle prédit l’absence de diabète que sa présence.

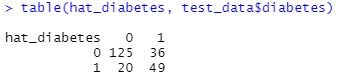
1. **Optimisation du modèle**

Pour améliorer notre modèle, on procèdera tout d’abord à une sélection des prédicteurs afin d’éliminer ceux qui ne sont pas significatifs et ensuite on déterminera le meilleur threshold.

* **Sélection des meilleurs prédicteurs**

Pour ce TP, on effectuera une sélection **backward**. Ainsi, on obtient le modèle suivant :

**Test du modèle :**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG On applique ici le modèle obtenu sur les données de test et on obtient les résultats suivants :

**C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGTaux d’erreur =>**

**Précision =>**

**Matrice de confusion**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

**<= Sensibilité Spécificité =>**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

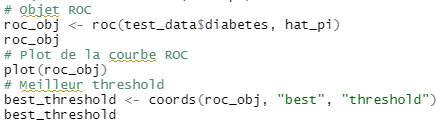
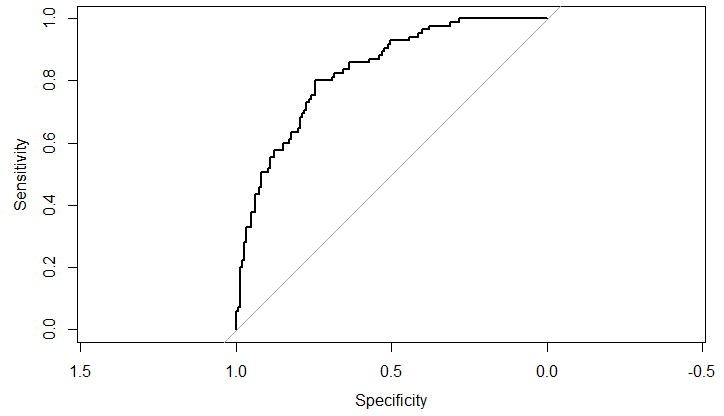
**<= taux de faux négatif taux de faux positif =>**

**Interprétation :**

On constate, que les prédicteurs **triceps** et **insulin** ont été éliminés comme on avait supposé lors de la détermination du modèle. Ces prédicteurs ne sont pas significatifs pour le modèle d’autant plus que les valeurs obtenues en les éliminant sont restées quasiment les mêmes.

On a la déviance résiduelle qui est passé de **500,39** à **500,89** en améliorant le critère **AIC** de **518,39** à **514,89.** La précision est passé de **76,9%** à **75,6%** tout comme la sensibilité et la spécificité qui ont légèrement chuté. Par contre on a toujours une bonne précision à la prédiction et le modèle aura toujours tendance à bien prédire l’absence du diabète que sa présence. Ceci est dû au fait que le jeu de données n’est équilibré comme on l’a mentionné dans l’étude du jeu de données.

* **Détermination du meilleur threshold**

Pour déterminer le meilleur threshold, il faut tracer la courbe ROC, ensuite déterminer le point de coordonnées **(x = spécificité, y = sensibilité)** appartenant à la courbetel que **x y** (**la différence entre x et y est la plus faible**) et enfin retenir la valeur du threshold qui a permis d’obtenir ce résultat. Dans RStudio, on obtient les résultats suivant :

**Code**

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG**Courbe ROC**

**Valeur du meilleur threshold**

**Interprétation :**

Pour obtenir un meilleur modèle que les précédents, il faut fixer le threshold à **0,298.** Ainsi, on aura une spécificité de **74,48%** et une sensibilité de **80%.** De plus, l’aire en dessous de la courbe ROC est de **0,8296,** ce qui est proche de **1.** Par conséquent, le modèle sera apte à bien prédire la présence tout comme l’absence de diabètes chez les patients et de ce fait réduire l’incertitude et le taux d’erreur des diagnostiques.

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGEn appliquant ce threshold, on obtient les résultats suivant :

C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

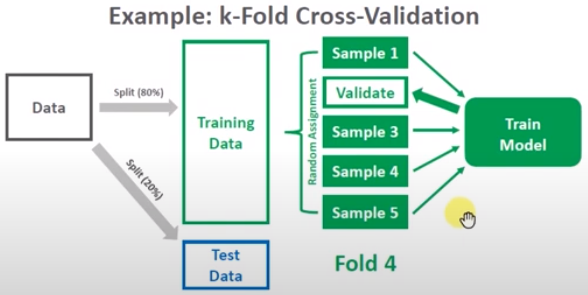
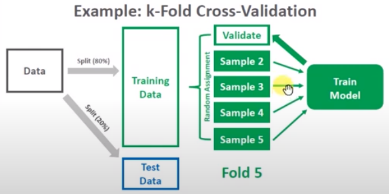
C:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNGC:\Users\mazou\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Capture.PNG

**Conclusion :**

Pour ce TP, il nous a été donné un jeu de données dans lequel on cherchait à savoir s’il était possible d’utiliser les mesures fournis pour modéliser voire prédire la présence/absence de diabètes chez les femmes du peuples Pima. Pour répondre à cette problématique, il est judicieux d’appliquer un modèle de régression logistique sur ces données en fixant un seuil de classification (**threshold**) à **29,8%.** Ainsi, on obtient une précision de prédiction de **76,52%,** une sensibilité de **80%,** une spécificité à **74,48%** et un taux d’erreur de **23,35%.**

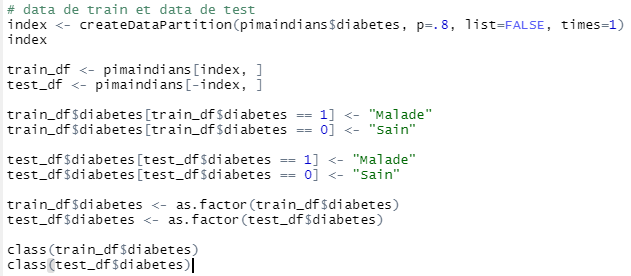
1. **Autre analyse : Application de la Cross Validation à la régression logistique**

**La Cross Validation** est une méthode d’analyse de données dans laquelle on split les données de train en plusieurs partition dont une pour la **validation.** Cette méthode permet d’améliorer le modèle car on effectuer le même nombre d’apprentissage que de partition en changeant chaque la partition qui sert à la validation.

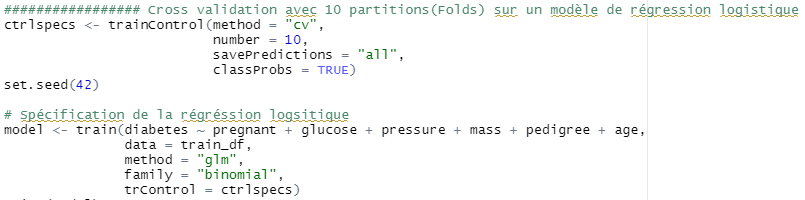
****

**=>**

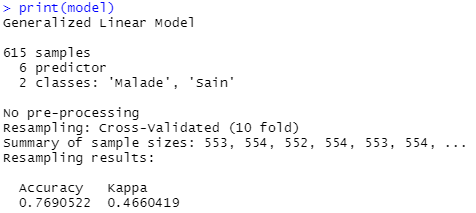
Pour mettre en place la **Cross Validation,** on commence par divisé les données de départ en données de train et de test et on transforme la variable de prédiction en facteur (Il est judicieux de les remplacer par des valeurs plus significative).

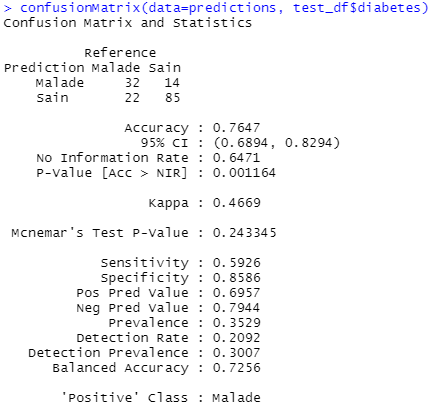


Ensuite, on divise les données de train en partitions (Dans ce TP j’ai créé 10 partitions dont une de validation) et on applique la régression logistique pour apprendre le modèle.



Dans cette analyse j’ai retenu, le même modèle que celui de la régression logistique simple (le modèle sans les variables **triceps** et **insulin** car ils ne sont pas significatif pour le modèle, de plus on obtient les quasiment les mêmes résultats avec ou sans ces variables).

Ainsi, on obtient une précision(**accuracy**) de **77%** sur les données de train. Voir ci-dessous :

En appliquant ce modèle pour prédire les données de test, on obtient les résultats suivants :

**Interprétation :**

On a une précision de **76,47%.** Ce qui signifie que le modèle prédit correctement la présence ou l’absence de diabètes à **76,47%.** Par contre, le modèle aura tendance à bien prédire correctement l’absence du diabète (**Spécificité = 85,86%**) que sa présence (**Sensibilité = 59,26%**).

On obtient quasiment les mêmes résultats que la régression logistique sans cross validation.

**Conclusion :**

Pour prédire l’absence ou la présence du diabète chez le peuple Pima, on applique la régression logistique avec cross validation. Ainsi on aura de meilleur résultat à savoir : **Précision = 76,47%, Sensibilité = 85,86%** et une **Spécificité = 59,26%.** Pour améliorer ces résultats, il faudrait fournir plus de données ayant autant de cas malade que de cas sain.

Lien du GitHub : <https://github.com/GariraMOUNDI/TP_ML>