La détection de l’anxiété : une intelligence artificielle pour des résultats plus fiables

Présenté par : Saad KIOUEH
Numéro d’inscription: MK199M

Sommaire :

Introduction

L’anxiété: Maladie de l’époque

L’intelligence artificielle au service des maladies mentales:

Algorithmes d'apprentissage automatique :

Supervisé:	Non supervisé:	Semi-supervisé:	Par renforcement:
• La régression linéaire • La régression logistique	• Les algorithmes Apriori • K-means	• Classificateur bayésien naïf • Réseaux antagonistes génératifs	• Q-Learning • Algorithme basé sur un modèle

La régression logistique:

Définitions:

- Définition: Inventé par 'JOSEPH BERKSON'

- Définition mathématique: $P(Y=yi/X)$ où $X=(X1,X2,\ldots\ldots..,Xj)$

Définitions:

Régression logistique binaire (Y=0/1)

Régression logistique multinomiale (Y=yi)

Domaines d'application:

- **En médecine** : Prédiction des maladies
- **Domaine bancaire** : Les groupes à risque lors de la souscription d'un crédit
- **Domaines des assurances** : Cibler une fraction clientèle sensible à une police d'assurance sur un tel ou tel risque

Pourquoi la régression logistique?

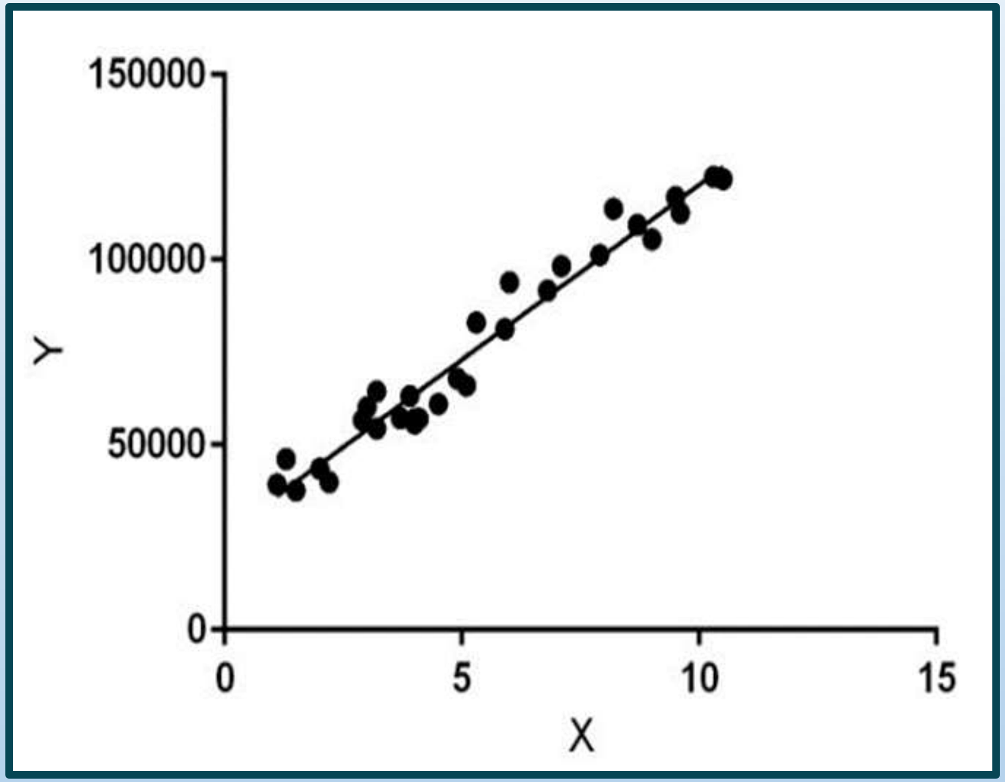


Figure 1 : Exemple d'un sujet de régression linéaire

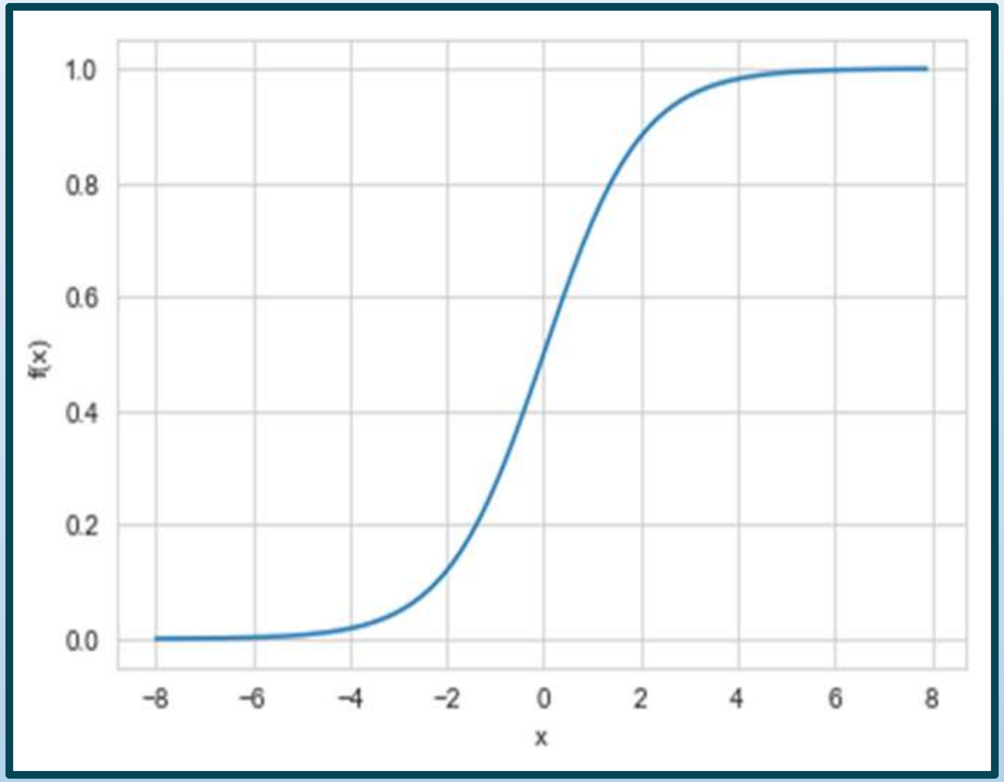


Figure 2: Exemple d'un sujet de régression logistique

Calculer la probabilité d'avoir l'anxiété (hypothèses de calcul):

- On utilise la régression logistique binaire
- Soit Y une variable qualitative prenant 2 valeurs $0\ et\ 1$ où : 1 si l'individu est atteint de l'anxiété, 0 sinon. Donc: $Y \hookrightarrow Be(n)$
- Soient X_k une variable qualitative ou quantitative où $k \in [\![1, n]\!]$ et n désigne le nombre de variables explicatives
- On pose $X = (X_1, X_2, \ldots \ldots \ldots, X_n)$
- On cherche à déterminer $P(Y = 1/X)$ et $P(Y = 0/X)$ en utilisant 2 modèles différents

Le modèle Logit:

- On cherche par exemple à déterminer $P(Y=1/X)$ en fonction de X

- La régression logistique fait partie de la classe la plus large des modèles linéaires, donc on cherche à identifier une fonction $\boldsymbol{g}$ tel que:

$$\boldsymbol{g}\big(\,\boldsymbol{P(Y=1/X)}\big)=\boldsymbol{a_0}+\boldsymbol{a_1X_1}+\ldots\ldots+\boldsymbol{a_nX_n}$$

- On definit donc la fonction logit:

$$\boldsymbol{logit(p)}=\mathbf{ln}\left(\frac{\boldsymbol{p}}{\mathbf{1}-\boldsymbol{p}}\right);\boldsymbol{o}\text{ù}\,\boldsymbol{p}\in\ ]\mathbf{0},\mathbf{1}[$$

Le modèle Logit:

- Donc, on choisit: $g(P(Y=1/X)) = logit(P(Y=1/X))$

$$\overset{1}{\Rightarrow} \ln\left(\frac{P(Y=1/X)}{1-P(Y=1/X)}\right) = a_0 + a_1X_1 + \ldots\ldots + a_nX_n$$

$$\overset{2}{\Rightarrow} \frac{P(Y=1/X)}{1-P(Y=1/X)} = e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}$$

$$\overset{3}{\Rightarrow} P(Y=1/X) = \left(e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}\right).(1-P(Y=1/X))$$

$$\overset{4}{\Rightarrow} P(Y=1/X).(1+e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}) = e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}$$

$$\overset{5}{\Rightarrow} P(Y=1/X) = \frac{e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}}{1+e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}}$$

$$et \quad P(Y=0/X) = 1 - P(Y=1/X) = \frac{1}{1+e^{a_0+a_1X_1+\ldots\ldots+a_nX_n}}$$

Le modèle semi-paramétrique:

- Enoncé du théorème de Bayes: $P(Y=1/X) = \frac{P(X/Y=1)P(Y=1)}{P(X)}$

- On calcule le rapport : $\frac{P(Y=1/X)}{P(Y=0/X)} = \frac{P(X/Y=1)P(Y=1)}{P(X)} . \frac{P(X)}{P(X)P(X/Y=0)P(Y=0)} = \frac{P(Y=1)}{P(Y=0)} . \frac{P(X/Y=1)}{P(X/Y=0)}$

- L'utilisation du terme semi-paramétrique

- Le rapport $\frac{P(Y=1)}{P(Y=0)}$ est facile à estimer, l'enjeu est donc l'estimation de $\frac{P(X/Y=1)}{P(X/Y=0}$

- La régression logistique introduit l'hypothèse fondamentale suivante
  - $\ln(\frac{P(X/Y=1)}{P(X/Y=0)}) = b_0 + b_1X_1 + ... + b_nX_n$

Equivalence des 2 approches:

$$\ln\left(\frac{P(Y=1/X)}{1-P(Y=1/X)}\right) = \ln\left(\frac{P(Y=1/X)}{P(Y=0/X)}\right)$$

$$= a_0 + a_1X_1 + \ldots\ldots.. + a_nX_n$$

$$= \ln\left(\frac{P(Y=1)}{P(Y=0)} \cdot \frac{P(X/Y=1)}{P(X/Y=0)}\right)$$

$$= \ln\left(\frac{P(Y=1)}{P(Y=0)}\right) + \ln\left(\frac{P(X/Y=1)}{P(X/Y=0)}\right)$$

$$= b_0 + b_1X_1 + \ldots\ldots.. + b_nX_n$$

$$\Rightarrow \begin{cases} a_0 = b_0 + \ln\left(\dfrac{P(Y=1)}{P(Y=0)}\right) \\ a_j = b_j\ ;\quad j \geq 1 \end{cases}$$

Modélisation informatique:

Etapes de la création d’un modèle de prédiction:

1. Importer les données
2. Nettoyer les données
3. Diviser les données entre 2 ensembles: entrainement/test
4. Créer un modèle
5. Entraîner le modèle
6. Faire des prédictions
7. Evaluer et améliorer le modèle

Exemple d'une base de données:

Id	Anxiety	Education	Married	Race	Religion	Sexual orientation	Age	Gender	Birth place	Urban	Pregnant	Co
1	1	high school	yes	black	muslim	heterosexual	17	M	usa	urban	unavailable	
2	0	high school	no	arab	christian	homosexual	16	M	usa	urban	unavailable	
3	0	university d	no	asian	christian	homsexual	19	M	usa	rural	unavailable	
4	0	less than hig	yes	black	atheist	heterosexual	24	F	France	suburba	yes	
5	1	less than hig	yes	white	atheist	bisexual	22	F	belgium	rural	no	
6	1	university d	yes	other	muslim	heterosexual	36	M	india	rural	unavailable	
7	0	university d	no	other	jewish	heterosexual	45	M	morocco	urban	unavailable	
8	0	university d	no	other	atheist	bisexual	23	F	usa	suburba	no	
9	1	less than hi	yes	black	other	homosexual	17	M	usa	urban	unavailable	
10	1	high school	yes	white	other	homosexual	28	F	morocco	urban	yes	

Programme informatique:

Exemple de résultats obtenus:

Python

```
>>> model.predict_proba(x)
array([[0.74002157, 0.25997843],
       [0.62975524, 0.37024476],
       [0.5040632 , 0.4959368 ],
       [0.37785549, 0.62214451],
       [0.26628093, 0.73371907],
       [0.17821501, 0.82178499],
       [0.11472079, 0.88527921],
       [0.07186982, 0.92813018],
       [0.04422513, 0.95577487],
       [0.02690569, 0.97309431]])
```

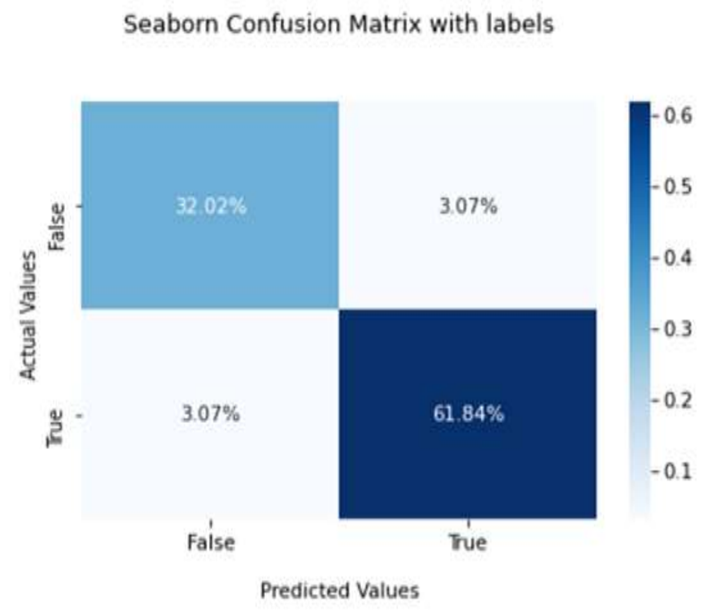
Python

```
>>> model.predict(x)
array([0, 0, 0, 1, 1, 1, 1, 1, 1, 1])
```

Exemples de matrices de confusion:

Output

```
[[ 73   7]
 [  7 141]]
```



Seaborn Confusion Matrix with Labels

Conclusion:

Merci pour votre attention!

Annexes :

Annexe 1:

```
Entrée [ ]: #importer les bibliothèques,les modules et les méthodes nécessaires
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import jaccard_similarity_score
from sklearn.metrics import confusion_matrix
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt

#importer les données
df=pd.read_csv('data.csv')

#informations sur la base de données
df.describe()
df.head()
df.values()
df.shape()

#nettoyer les données
#traitement des données manquantes
#supprimer les colonnes indésirables
#supprimer les valeurs répétées
#opérations qui dépendent de la base de données
X=df.drop(['anxiety','id'], axis = 'columns')
Y=df['anxiety']
```

Annexe 1:

```
#diviser les données entre 2 ensembles: entrainement/test
X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size = 0.2)

#création du modèle
model = LogisticRegression()

#entrainement du modèle
model.fit(X_train, Y_train)

#tester l'algorithme
P_pred=model.predict_proba(X_test)
Y_pred=model.predict(X_test)

#demander les données à l'utilisateur
Education=input("indiquez votre niveau scolaire")
Married=input("indiquez si vous êtes marié")
Race=input("indiquez votre ethnicité")
Religion=input("indiquez votre religion")
Sexual_orientation=input("indiquez votre orientation sexuelle")
Age=int(input("indiquez votre âge"))
Gender=input("indiquez votre sexe")
Birth_place=input("indiquez votre lieu de naissance")
Urban=input("indiquez si vous habitez dans un milieu urbain")
Pregnant=input("indiquez si vous êtes enceinte")

#faire des prédictions
P_pred1=model.predict_proba([[Education,Married,Race,Religion,Sexual_orientation,Age,Gender,Birth_place,Urban,Pregnant]])
Y_pred1=model.predict([[Education,Married,Race,Religion,Sexual_orientation,Age,Gender,Birth_place,Urban,Pregnant]])
```

Annexe 1:

```
#évaluer le modèle
score=jaccard_similarity_score(Y_test, Y_pred)

#matrice de confusion
cf_matrix = confusion_matrix(Y_test, Y_pred)

#matrice de confusion pour les classes binaires avec pourcentage
ax = sns.heatmap(cf_matrix/np.sum(cf_matrix), annot=True, fmt='.2%', cmap='Blues')
ax.set_title('Seaborn Confusion Matrix with labels\n\n');
ax.set_xlabel('\nPredicted Values')
ax.set_ylabel('Actual Values ')
ax.xaxis.set_ticklabels(['False','True'])
ax.yaxis.set_ticklabels(['False','True'])
plt.show()
```

Annexe 2:

La règle de décision pour le modèle logit:

La règle d'affectation peut être basée sur $P(Y = 1/X)$ de différents manières:

- Si $\frac{P(Y=1/X)}{1-P(Y=1/X)} > 1$ alors $Y = 1$
- Si $P(Y = 1/X) > 0.5$ alors $Y = 1$

Elle peut aussi être basée simplement sur le terme $A(X) = a_0 + a_1X_1 + \ldots\ldots\ldots + a_nX_n$ tel que:

- Si $\mathrm{A}(X) > 0$ alors $Y = 1$

La règle de décision pour le modèle semi-paramétrique:

- Si $\frac{P(Y=1/X)}{P(Y=0/X)} > 1$ alors $Y = 1$

Annexe 3:

Interprétation des coefficients de la régression logistique $(\mathbf{a_0}, \mathbf{a_1}, \dots \dots \dots, \mathbf{a_n})$:

- On définit d'abord l'$\boldsymbol{odds}$ (ou « cote »):

  → Soit P une probabilité, son $odds$ est défini par: $odds_P = \frac{P}{1-P}$

- Interprétation de a_0: si $(X_1, X_2, \dots \dots \dots, X_n) = (0,0, \dots \dots \dots, 0)$, a_0 est égal au ln de l'$odds$ de l'événement d'intérêt, et donc e^{a_0} est égal au $odds$.

Annexe 3:

- On définit ensuite l'$odds$ ratio (ou « rapport des cotes »):

  - $OR = 1$: maladie indépendante du symptôme .
  - $OR > 1$: maladie plus fréquente pour les individus qui ont le symptôme .
  - $OR < 1$: maladie plus fréquente pour les individus qui n'ont pas le symptôme .

- Interprétation de a_1:

  - on pose $odds_x$ pour l'événement d'intérêt quand $X_1 = x$, et $odds_{x+1}$quand $X_1 = x + 1$, donc $a_1 = \ln(odds_{x+1}) - \ln(odds_x)$ et $e^{a_1} = RC = \frac{odds_{x+1}}{odds_x}$

Annexe 4:

- **Modèle Probit:**

  - Identiquement au modèle logit, on cherche à identifier une fonction g tel que:

$$g(\,P(Y=1/X)\,) = a_0 + a_1X_1 + \ldots\ldots.. a_nX_n$$

  - On prend donc: $g\big(P(Y=1/X)\big) = \Phi^{-1}(P(Y=1/X))$; où: $\Phi^{-1}:]0,1[\rightarrow \mathbb{R}$ est la fonction probit, qui est définie comme la réciproque de la fonction de répartition de la loi normale centrée réduite

    Donc: $P(Y=1/X) = \Phi(a_0 + a_1X_1 + \ldots\ldots.. a_nX_n)$; où:

$$\Phi(z) = \int_{-\infty}^{z} \left(\frac{1}{2\pi}\right)^{\frac{1}{2}} \exp\left(\frac{-u^2}{2}\right) du$$