

Projet de Machine Learning - Learning from the Crowd

CORVISIER Jean-Christophe - DELORO Yonatan - KHELDOUNI Mohammed Amine

1 juin 2017

École des Ponts ParisTech

Encadrés par :

Objectifs du projet

- Apprentissage supervisé à partir d'annotations de qualités diverses.
Idée : "combiner le savoir de sources multiples" quand la vérité terrain n'existe pas ou est difficilement accessible.
 - Déterminer la présence ou non d'une maladie en fonction de divers pronostics donnés par plusieurs médecins lorsque seule une opération permettrait de connaître la vérité ;
 - Estimer la 'vraisemblance' d'une information en croisant les sources d'actualité (Wikipedia)
- Application choisie : Établir des tendances socio-économiques sans posséder de vérité terrain.

Problème et Notations

- On dispose de N données (X_i) vecteurs de *features*
- Pour ce jeu de données, on détient des labels distribués par T "arbitres". On note alors $Y_i^{(t)}$ le label délivrée par l'arbitre t pour la donnée X_i .
- Nous noterons également dans la suite Z_i le vrai label correspondant à cette donnée.

Dans ce projet, on s'intéresse exclusivement à la classification binaire ($Y_i, Z_i \in \{+1, -1\}$). Nous avons ainsi cherché, à partir d'un jeu d'entraînement $X_i, Y_i^{(t)}$, à prédire le vrai label Z associé à un vecteur de *features* X .

Majority Voting

Pour un modèle à plusieurs annotateurs, une stratégie classique est de choisir le label voté par la majorité d'entre eux.

Pour un problème à classification binaire :

$$Z_i = \begin{cases} 1 & \text{si } \frac{1}{T} \sum_{t=1}^T Y_i^t > 0.5 \\ 0 & \text{si } \frac{1}{T} \sum_{t=1}^T Y_i^t < 0.5 \end{cases}$$

On pourra utiliser alors cette stratégie pour trouver une première estimation de la probabilité du label en se basant sur les annotations.

$$\mathbb{P}(Z_i = 1 | Y_i^1, \dots, Y_i^T) = \frac{1}{T} \sum_{t=1}^T Y_i^t$$

Description des modèles

Première modèle

Modèle

- Indépendance des annotateurs
- Chaque annotateur t possède une probabilité α^t d'énoncer à raison le label 1 et une probabilité β^t d'énoncer à raison le label 0.
 α est la sensibilité et β est la spécificité.

Classification Nous utilisons des fonctions linéaires $f_w(X) = w^T X$. Etant donné un seuil γ , le label vaut

$$\begin{cases} 1 & \text{si } f_w(X) > \gamma \\ 0 & \text{sinon} \end{cases}$$

Apprentissage Etant donné les paramètres $\theta = \{\alpha, \beta, w\}$, on peut écrire la vraisemblance du modèle comme suit :

$$\begin{aligned} \ln(\mathbb{P}(Y, Z | \theta)) &= \ln\left(\prod_i p((Y_i^1, \dots, Y_i^T) | X_i; \theta)\right) \\ &= \sum_{i=1}^N Z_i \ln(p_i a_i) + (1 - Z_i) \ln(1 - p_i) b_i \end{aligned}$$

- $p_i = \sigma(w^T X_i)$, avec σ la fonction sigmoïde.
- $a_i = \prod_{t=1}^T \alpha_t^{Y_i^t} (1 - \alpha_t)^{(1-Y_i^t)}$
- $b_i = \prod_{t=1}^T \beta_t^{(1-Y_i^t)} (1 - \beta_t)^{Y_i^t}$

On maximise la log-vraisemblance en fonction de θ à l'aide d'un algorithme EM (*Expected Maximisation*).

- E-step : Les valeurs des Z_i étant inconnues, on les approxime par des variables μ_i avec $\mu_i = \mathbb{P}(Z_i = 1 | Y_i^1, \dots, Y_i^T, X_i, \theta)$
- M-step : On maximise par rapport à w à l'aide d'un algorithme d'optimisation type Newton-Raphson ou BFGS. On optimise les α et β par formules closes.

Initialisation de l'EM par *Majority Voting*

Spécialisation des annotateurs

Modèle

Les annotateurs sont supposés indépendants et produisent dans ce modèle des labels suivant des lois de Bernoulli de paramètre $\eta^t(X)$ dépendant de la donnée.

$$p(Y_i^t | Z_i, X_i) = (1 - \eta^t(X_i))^{Y_i^t - Z_i} [\eta^t(X_i)]^{1 - Y_i^t - Z_i}$$

où $\eta^t(X_i) = \sigma(\alpha_t^T X_i + \beta_t)$

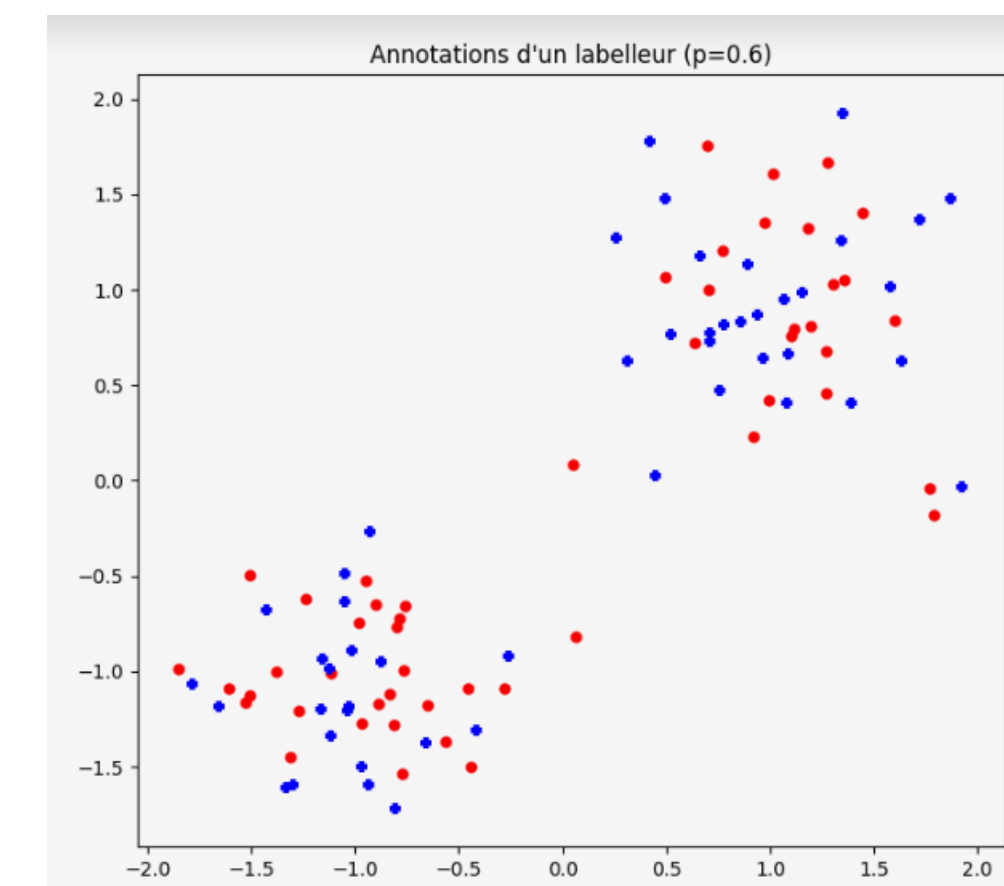
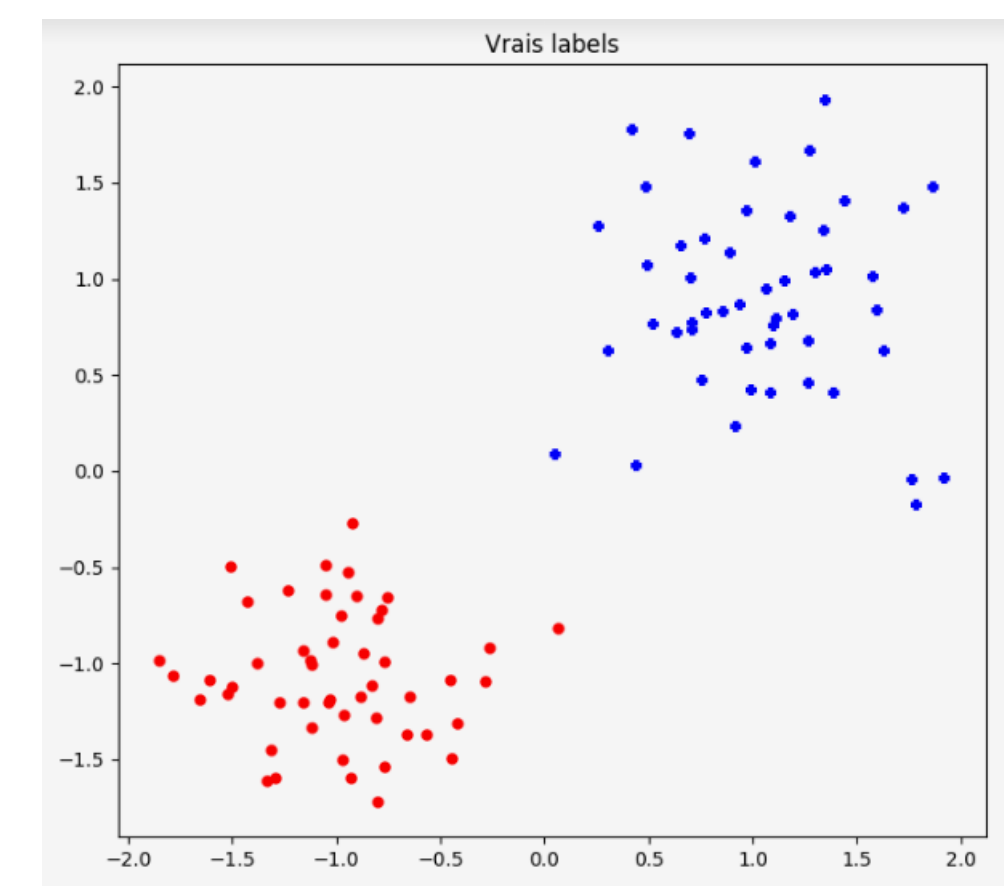
Classification Pour ce modèle, la classification se fait toujours à l'aide de fonctions linéaires $f_{w, \gamma}(X) = \sigma(w^T X + \gamma)$.

Apprentissage La formule générale de la vraisemblance est la même que dans le premier modèle. On cherche à maximiser cette fonction par rapport aux paramètres $\theta = \{\alpha_t, \beta_t, \gamma, w\}$ à l'aide d'un algorithme EM :

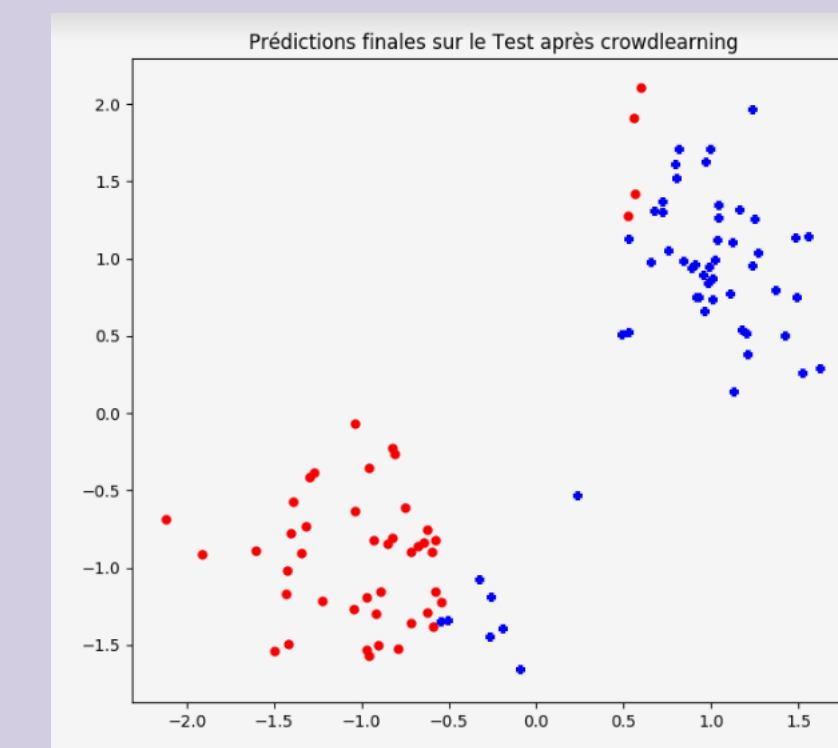
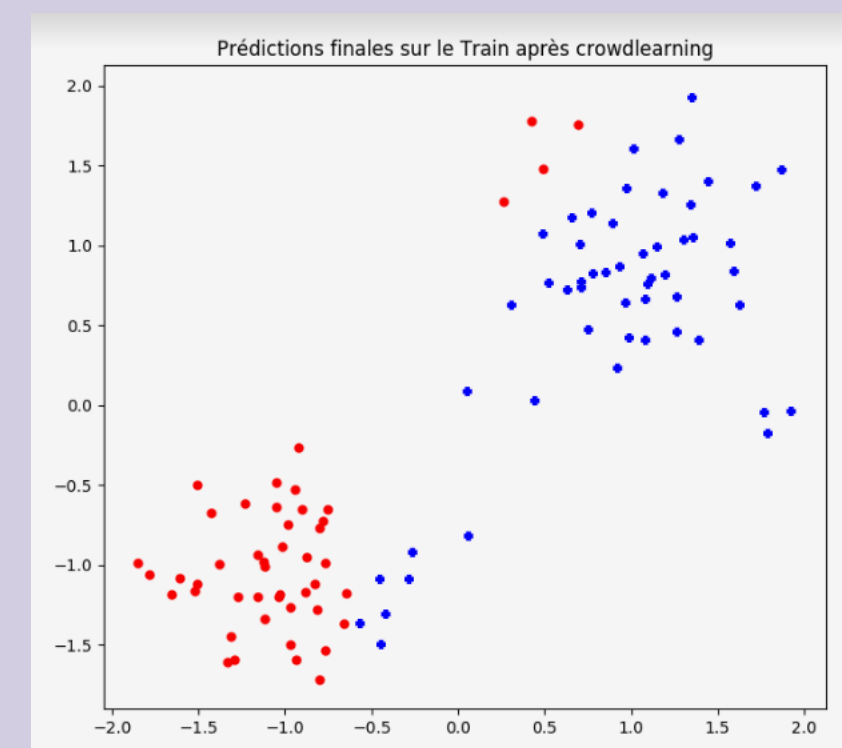
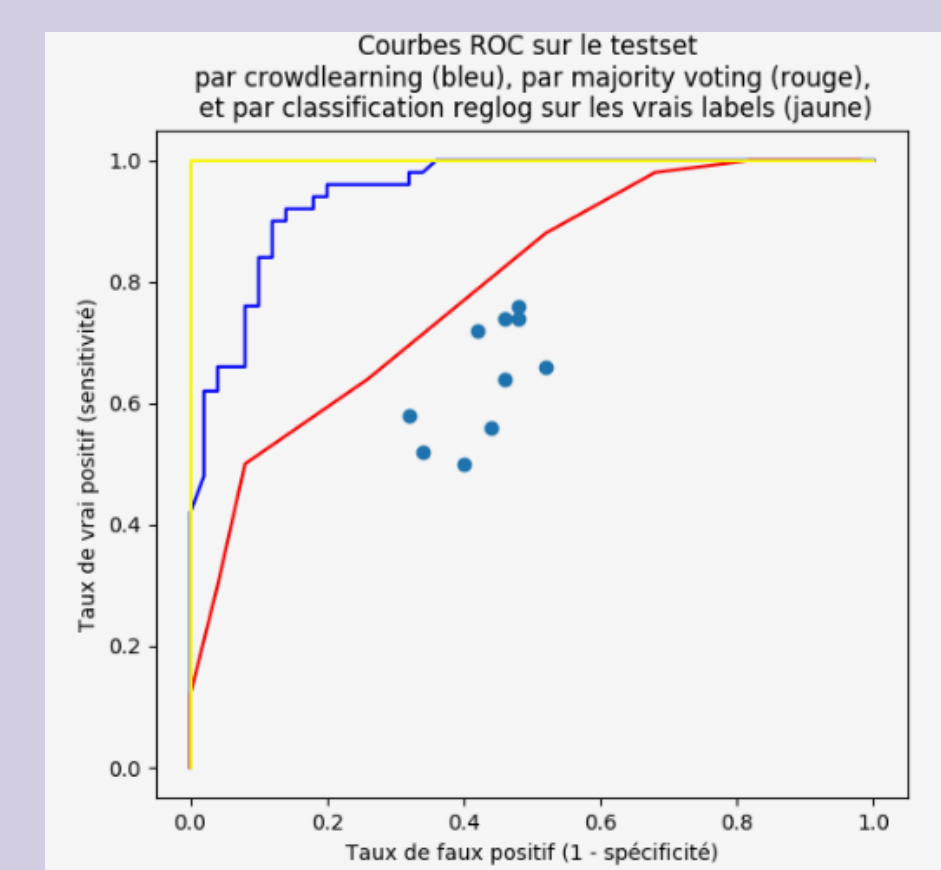
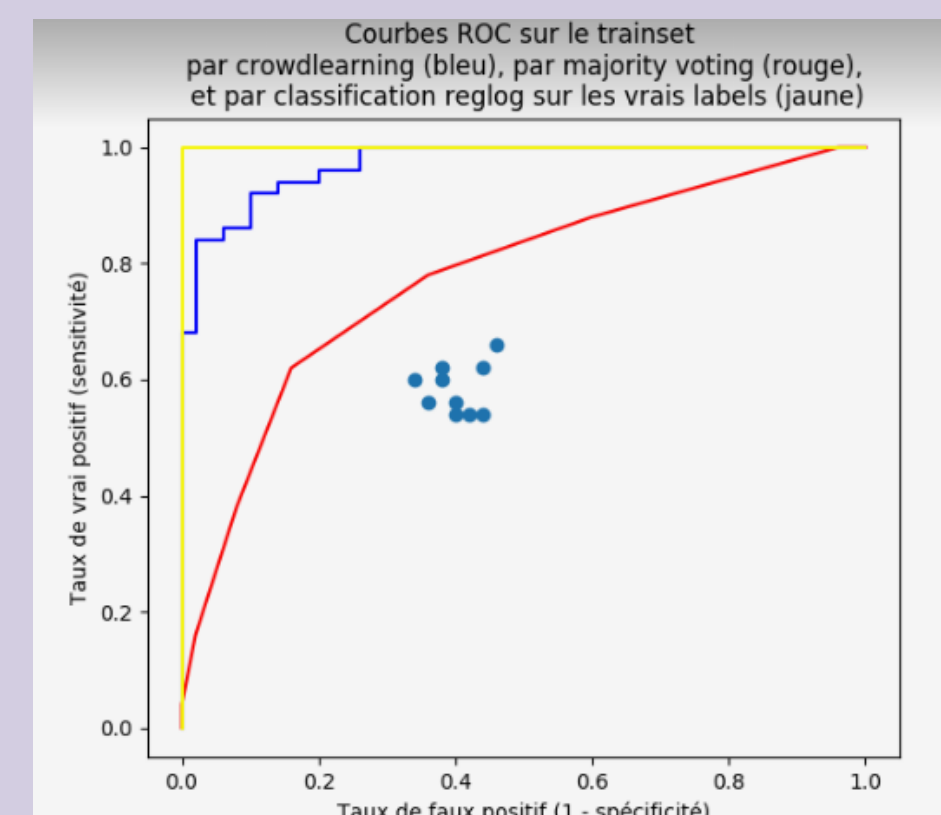
- E-step : Estimation des $\tilde{p}(Z_i) = \prod_t p(Y_i^t | X_i, Z_i) p(Z_i | X_i)$
- M-step :

$$\max_{\theta} \sum_{i,t} \mathbb{E}_{\tilde{p}(Z_i)} [\ln(p(Y_i^t | X_i, Z_i) + \ln(p(Z_i | X_i))]$$

On utilise un algorithme BFGS (quasi-Newton) pour l'optimisation. Problème de robustesse dans notre implémentation vis-à-vis de l'initialisation de θ



Résultats (données artificielles)



Génération des données : 2 gaussiennes bruitées (0.3)
10 annotateurs de probabilité de succès 0.6 ($\alpha = \beta = 0.6$)

Comparaisons :

- du Crowdlearning (première approche, apprentissage avec X, Y , prédiction avec X)
- du Majority Voting (prédiction avec Y)
- d'une classification linéaire par régression logistique directement à partir des vrais labels (apprentissage avec X, Z , prédiction avec X)

Dépendance des annotateurs (Ébauche)

Après implémentation des modèles, nous avons pensé à modéliser des relations de dépendance entre ces annotateurs en supposant une répartition par groupe selon leur sensibilité à suivre une consigne de vote. Formellement, l'expression de $\mathbb{P}(y_i^t | z_i, x_i)$ dépend de la propension des annotateurs à suivre la consigne ($\sigma(\nu_t) \sigma(s_t)$), l'exclure ($\sigma(\nu_t)(1 - \sigma(s_t))$), ou annoter selon les données ($(1 - \sigma(\nu_t))$).

$$\mathbb{P}(y_i^t | z_i, x_i) = (1 - \sigma(\nu_t)) \eta_t(x_i)^{|y_i^t - z_i|} (1 - \eta_t(x_i))^{1 - |y_i^t - z_i|} + \sigma(\nu_t) \sigma(s_t)^{y_t} (1 - \sigma(s_t))^{1 - y_t}$$

Résultats (données réelles)

- On considère dans ce projet des données socio-économiques décrivant des individus, le but étant de prédire s'ils ont un capital annuel de plus de 50k€.
- On considère dans les expériences ci-dessous la partition *train/(train+test)* égale à 80%.
- On obtient les scores suivants :

$$score_{train} = 77.13\%$$

$$score_{test} = 73.1\%$$

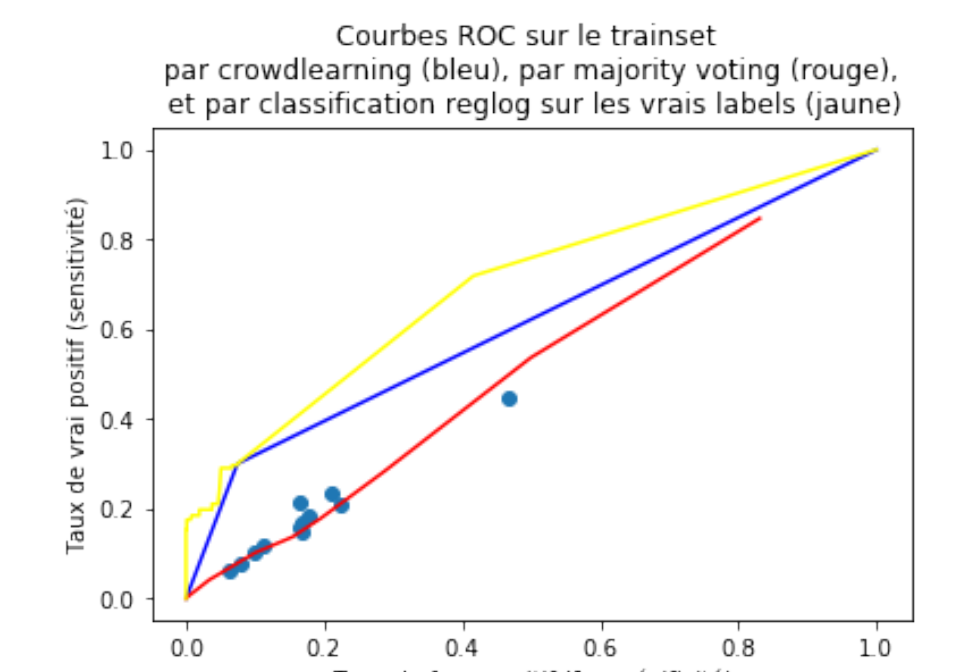


FIGURE – Courbes ROC sur les données réelles en entraînement

Comme le montre les courbes, notre modèle est plus performant en entraînement que le *Majority Voting*, et est assez proche d'un classifieur linéaire appris avec les vrais labels pour les petites valeurs de seuil, ce qui montre l'intérêt de la méthode.

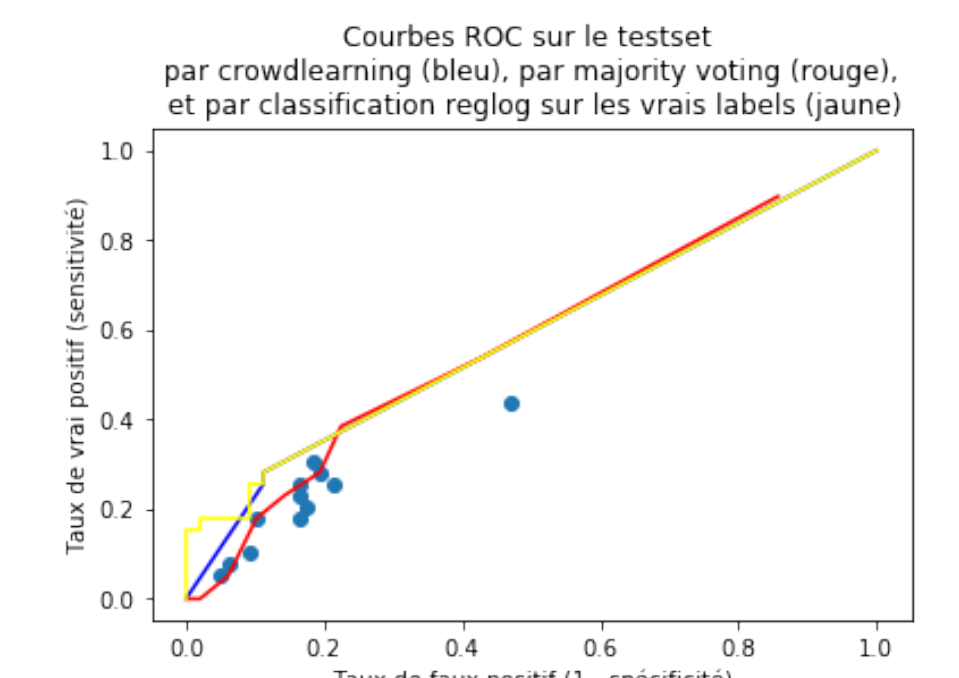


FIGURE – Courbes ROC sur les données réelles en test

Sur les données de tests on constate que le *crowdlearning* donne quasiment les mêmes résultats que le classifieur linéaire appris avec les vrais labels ou le *Majority Voting*.

- Tentative de régularisation de l'EM (ajout d'un terme $\lambda \|w\|^2$)
- Analyse des scores par le taux de *slicing* ($train/(train+test)$).

Bibliographie

- Yan Yan, Gerardo Hermosillo - *Modeling annotator expertise : Learning when everybody know a bit of something*
- Vikas C. Raykar, Shipeng Yu - *Learning from the crowd*
- UCI Machine Learning - *"Adult" datasets*