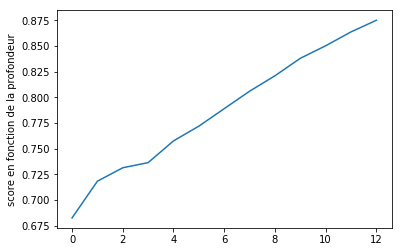
Compte-rendu TME

TME 1 : Arbres de décision et sélection de modèles (Fichier associés : Dans TME 1 : exercice2.py et exo1(3).py)

Q 1.3) L’indice du meilleur attribut est 17 (Drama). Si le gain est de 0, cela veut dire que l’entropie est égale à l’entropie conditionnelle. Le test sur cet attribut ne servirait à rien. Si le gain est de 1, alors on peut totalement classifier l’exemple selon la valeur de cet attribut.

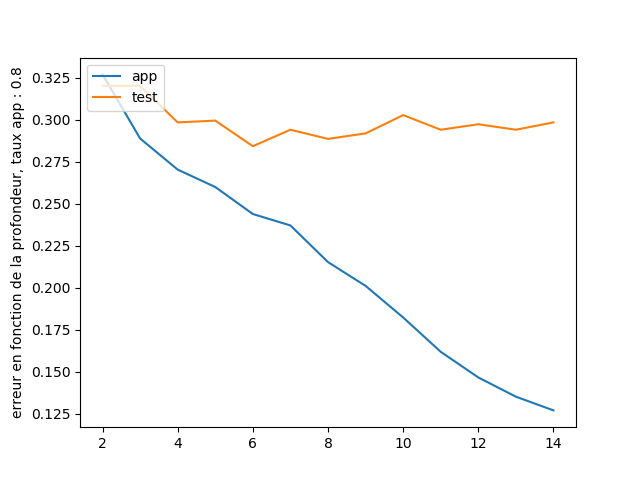
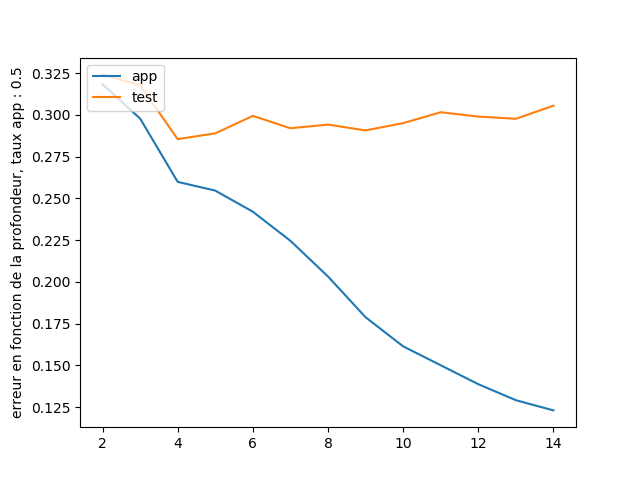
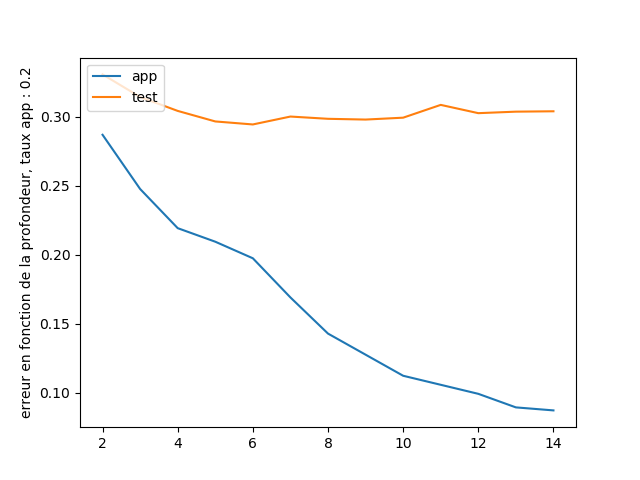
Q 1.4) Plus on descend, moins il y a de films qui sont séparés à chaque nœud car il y a moins de films totaux dans chaque nœud. Cela est normal car cela permet de classifier de plus en plus précisément les films. Cependant, cela peut mener à du surapprentissage.

Q 1.5) Plus on augmente la profondeur, plus le score augmente. Cela est normal car plus la profondeur augmente, plus l’arbre est précis pour classifier.



Q 1.6) Non, pas forcément car ici on utilise le même ensemble de données pour l’apprentissage est le test. Il y a donc un risque que l’arbre de décision apprenne par cœur les données. Il n’aura donc pas défini des règles suffisamment générales pour classifier des bons ou des mauvais films qui ne sont pas dans la base de données d’entrainement

Q 1.7)

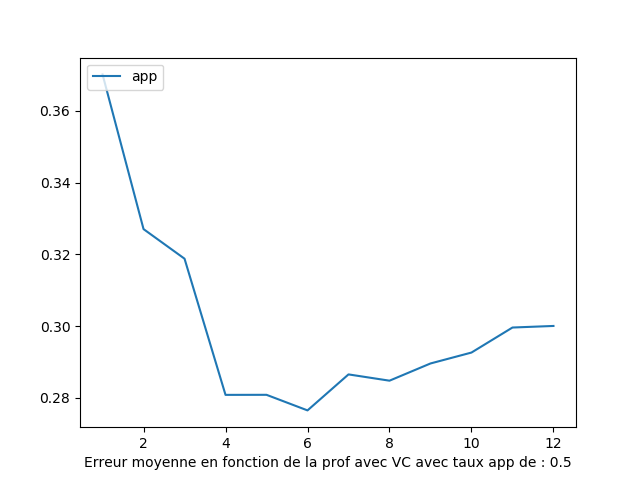


Q 1.8) L’erreur en apprentissage descend un peu plus rapidement et un peu plus quand on a moins d’exemples pour l’apprentissage.

L’erreur en test semble suivre la même tendance. Elle diminue rapidement quand peu d’exemples de test et est plus basse. Elle diminue moins rapidement quand on a beaucoup d’exemples de test.

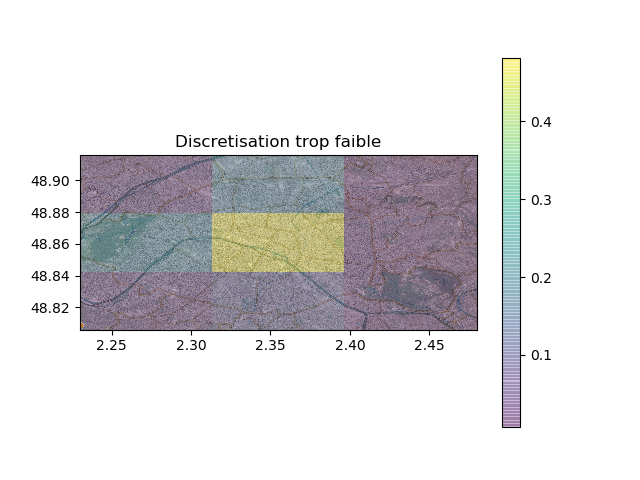
Q 1.9) Il peut exister un biais lors de la sélection des exemples qui vont dans l’apprentissage ou dans l’ensemble de test. Donc on ne pas assurer qu’ils soient totalement fiables et stables.

Validation croisée :

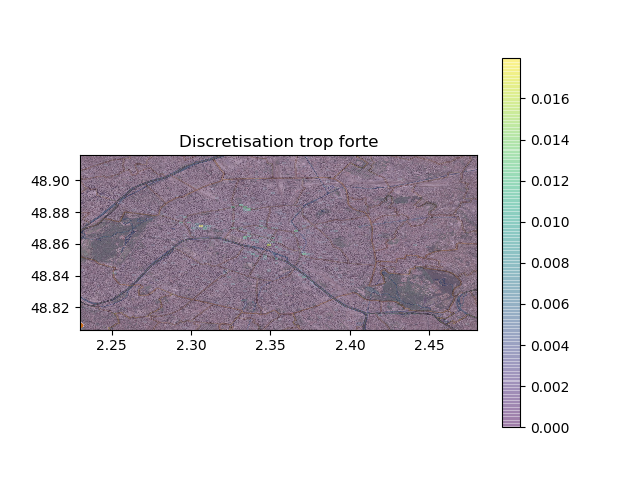


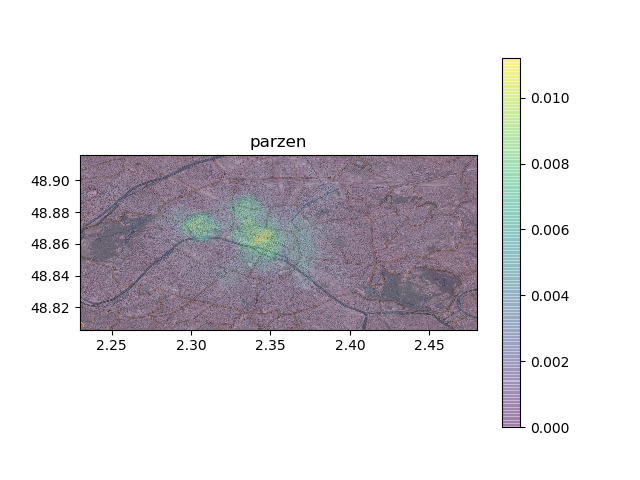
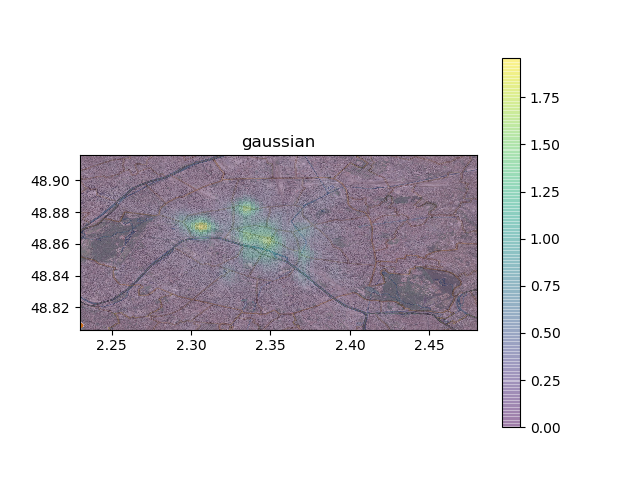
TME 2 : Estimation de densité (Dans TME 2 : fichier tme2-poi.py)

Si on choisit une discrétisation faible, l’histogramme ne va pas être précis du tout.



Si on en choisit une forte, cela ne sera trop localisé avec des cases avec des densités très élevées et juste à côté très basses. Cela n’est donc pas vraiment représentatif.

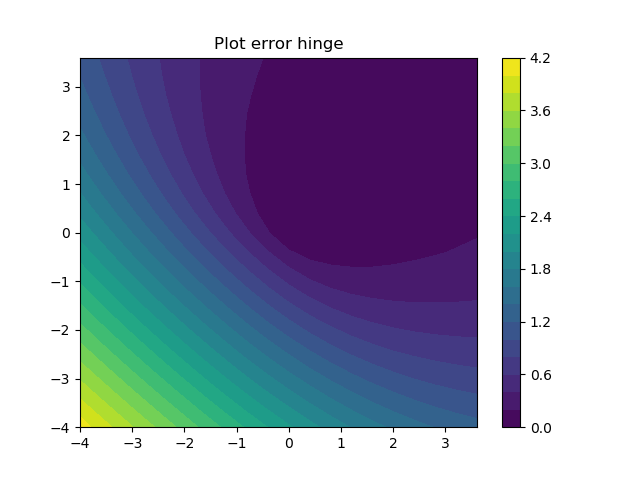
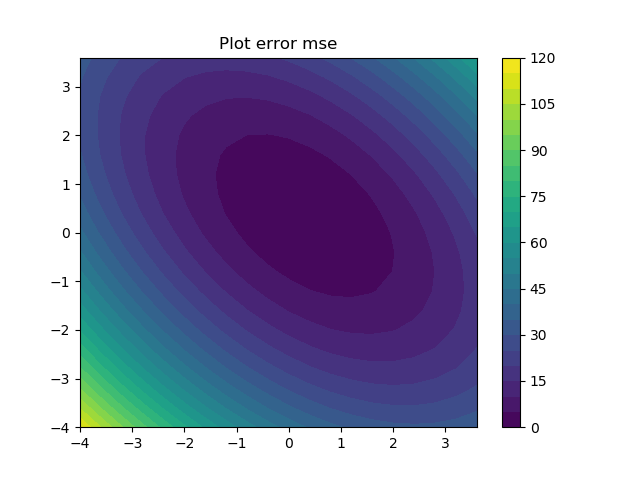


Le paramètre à définir est la « taille » du noyau (par exemple largeur de la fenêtre pour parzen). Si elle est trop faible on efface des détails, et on déforme la réalité si elle est trop forte. Cela est similaire à la discrétisation pour la méthode des histogrammes. 

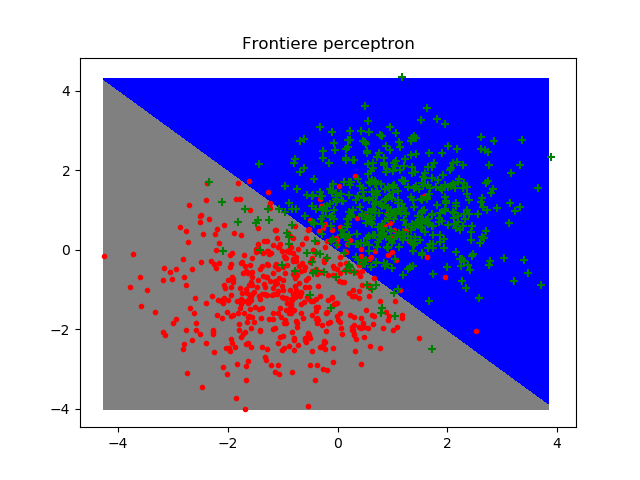
Pour déterminer les meilleurs paramètres, on peut utiliser de la validation croisée.

TME 3 : Descente de gradient et perceptron

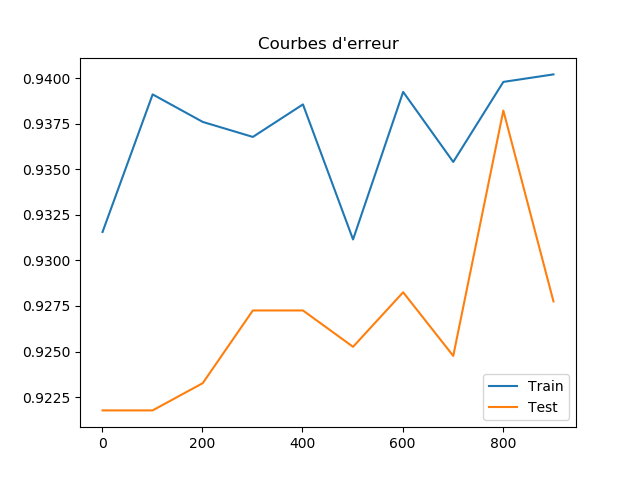
Graphiques des surfaces de l’erreur :



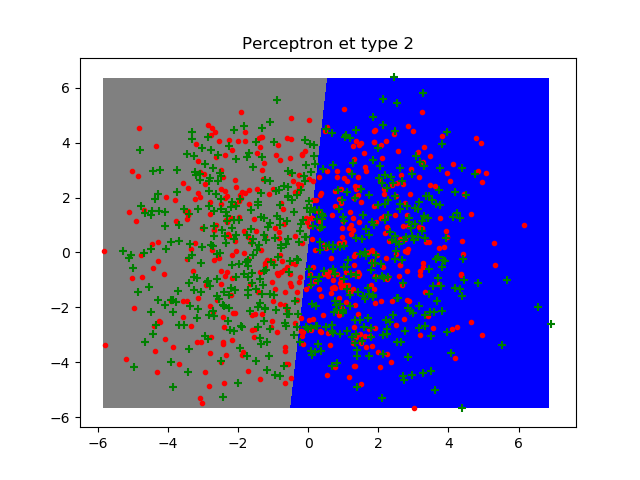
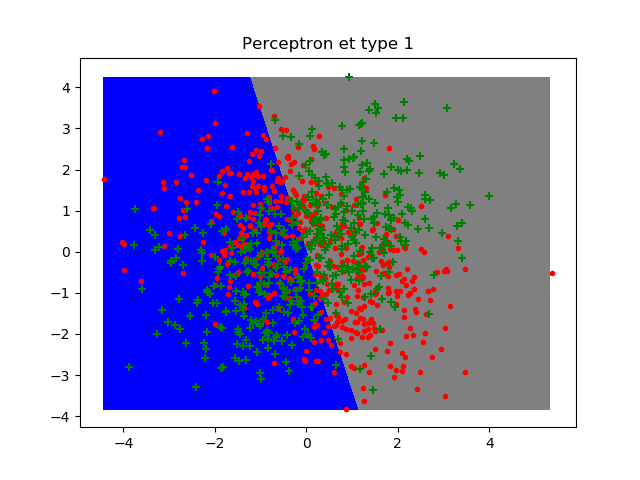
Graphique des frontières obtenues :



Pour prendre en compte le biais, il faut qu’on ajoute une valeur aux poids et une colonne correspondante dans les données.



Il semblerait qu’il n’y a pas de surapprentissage étant donné que le score reste élevé même sur les données de test.



En testant le perceptron sur les autres données artificielles, on peut observer que le perceptron a un score beaucoup plus faible. Cela est normal étant donné qu’il s’agit d’un problème non linéairement séparable. Il faut donc augmenter la dimension en effectuant une projection.

TME 4 : SVM (TME 4 : Fichier tme4.py)

3) SVM et Grid Search

En explorant les différents noyaux, on peut observer que :

* Pour les données de type mélange de 2 gaussiennes, les trois noyaux sont efficaces
* Pour les données de type mélange de 4 gaussiennes, linear n’est plus du tout efficace et poly l’est un peu moins. RBF reste bon.
* Pour les données de type échiquier, le noyau rbf est le plus efficace

Pour le nombre de vecteurs supports on obtient cela : (A mettre sous forme de tableau

|  |  |  |  |
| --- | --- | --- | --- |
| Nombre de vecteurs supports pour le noyau et les données | Noyau linéaire | Noyau poly | Noyau RBF |
| Données type 0 / Param par défaut | 207 | 282 | 222 |
| Données type 0 / C élevé ( C = 500) | 204 | 285 | 190 |
| Données type 1 / Param par défaut | 984 | 979 | 54 |
| Données type 1 / C élevé | 984 | 977 | 11 |
| Données type 2 / Param par défaut | 975 | 974 | 927 |
| Données type 2 / C élevé | 975 | 975 | 499 |
|  |  |  |  |

Il semblerait que dans le cas linéaire l’impact du paramètre C n’ait pas d’impact sur le nombre de vecteurs supports.

Pour le noyau poly, augmenter le nombre de degré et le paramètre gamma semble augmenter aussi le nombre de vecteurs supports.

Grid Search :

Pour l’apprentissage multi-classe, nous avons obtenu des meilleurs résultats avec le one-versus-one.