Machine Learning
Premiers apprentissages

Jean-François COUCHOT
couchot[arobase]femto-st[point]fr

12 novembre 2024

1 Apprentissages supervisés

1.1 Classification bayésienne naïve

Exercice 1.1 (Mise en place du TP). *Tout le TP repose sur le dataset sur les personnes d'origine PIMA atteintes ou pas de diabètes vu en cours. Ce dataset est accessible sur la page de Pima Indians Diabetes Database de kaggle.*

1. *Récupérer le dataset et l'enregistrer sous le nom de* `diabetes.csv`.
2. *Comprendre le prétraitement sur les données (T11 du cours). L'exécuter.*

Exercice 1.2 (Apprentissage bayésien naïf gaussien).

1. *Lorsqu'on écrit* `gnb = GaussianNB()` *au T13, que suppose-t-on sur les distributions des valeurs des attributs ?*
2. *Est-ce cohérent avec les histogrammes donnés au T12 ?*
3. *Exécuter cependant le code donné en T13. Noter le score moyen de précision.*
4. *Constater que ce score ne change pas malgré un découpage aléatoire des données selon l'instruction rappelé ci-dessous. Pourquoi ?*

```
cv = StratifiedShuffleSplit(n_splits=5, test_size=0.2, random_state=0)
```

5. *Pourquoi apprend-on sur tout le dataset à la ligne contenant l'instruction* `gnb.fit(X_train.values, y_train)` *?*
6. *En utilisant la méthode en* `predict_proba` *de* `GaussianNB`*, expliquer pourquoi la méthode prédit que la personne ayant les caractéristiques* `[1, 184, 84, 33, 82, 35.5, 0.355, 41]` *a le diabète à la dernière ligne.*

Exercice 1.3 (Combinaison de distributions (plus difficile)). *L'exercice précédent a montré qu'on avait approximé des distributions non normales par des distributions qui le sont. Cela peut paraître perfectible.*

1. *Repérer les distributions qui sont gaussiennes et celles qui ne le sont pas en exploitant le T12 du CM.*
2. *Pour les valeurs de* `Age`*, et* `Pregnancies`*,...on pourrait exploiter une approche à base de multinomial naïve Bayes, qui construit les probabilités selon les fréquences. Comprendre cette méthode en étudiant la section correspondante sur le site de Wikipedia.*
3. *Proposer un estimateur répondant à cette problématique.*

1.2 Arbres de décision

Exercice 1.4 (Premier arbre).

1. *Rejouer le code donné au T22.*
2. *La construction des noeuds est réalisée en choisissant ceux qui réduisent le plus l'entropie. Constater ceci dans le code.*
3. *Changer ce critère pour une autre méthode nommée l'indice de Giny. Cela a-t-il une influence significative sur la précision de la classification ?*
4. *Afficher l'arbre construit. Interpréter les nœuds qui ont permis de découper le plus efficacement l'espace.*
5. *Avec cet arbre, comprendre la réponse donnée à la classification d'une personne avec les caractéristiques suivantes. [1, 184, 84, 33, 82, 35.5, 0.355, 41].*

1.3 Réseau de neurones

Exercice 1.5 (Premier réseau).

1. *Rejouer le code donné sur l'apprentissage à base de MLP donné dans le cours. Comprendre particulièrement le paramètre* `hidden_layer_sizes=(12, 8)`.
2. *Modifier le code pour permettre de varier le nombre de couches entre 2 et 5 et à chaque fois, de varier le nombre neurones entre 5 et 10. Attention, cela implique d'évaluer 9324 configurations différentes (il faudrait justifier ceci). On pourra essayer de paralléliser les évaluations en exploitant l'exécuteur* `ThreadPoolExecutor`.
3. *Quelle est la configuration qui obtient le meilleur score moyen de validation croisée pour l'accuracy ?*
4. *Cette métrique ne tient compte que de la moyenne sans tenir compte de l'écart type. On préférera comme critère la maximisation du score $m - 2\sigma$ où m est la moyenne et σ est la déviation standard. En lisant l'article* [1]*, justifier le choix de ce critère et le retenir pour choisir la meilleure configuration.*
5. *Avec ce réseau interpréter la réponse donnée pour une entrée dont les valeurs sont*

   ```
   [1, 184, 84, 33, 82, 35.5, 0.355, 41]
   ```

   en analysant les valeurs de la couche de sortie. On pourra pour cela exploiter la méthode `predict_proba` *de la classe* `MLPClassifier`.

2 Apprentissages non supervisés

Exercice 2.1 (Détection d'anomalies). *kmeans permet de regrouper les données en clusters. Les éléments très éloignés de ces clusters peuvent être considérés comme des éléments atypiques. On va se servir de ceci pour détecter les jours où des événements non ordinaires se sont produits dans une liste de relevés de constituants de l'air.*

1. *Charger le jeu de données Air Quality Dataset de kaggle.*
2. *Supprimer les lignes et les colonnes inutiles, et transformer les séparateurs décimaux en point.*
3. *Comme indicateur de qualité de l'air, retenir tous les attributs sauf la date et l'heure. Appliquer une mise à l'échelle standard des données (*`StandardScaler()`*) pour obtenir* `X_scaled`.
4. *On va choisir 3 clusters : ce choix sera détaillé au chapitre 4. En exploitant la méthode* `fit_predict` *de la classe* `KMeans`*, construire les libelles* `y_kmeans` *du tableau* `X_scaled` *et les centroïdes associés.*
5. *Pour afficher le nuage de points en 2D, on peut penser à exploiter une réduction par analyse par composants principaux (PCA). Un exemple de code réalisant ceci est accessible à* `https://www.kaggle.com/code/rodrigofragoso/explained-k-means-pca-visualization`.
6. *Construire alors la liste des points à une distance au centroïde respectif supérieure à 99.9% des autres distances.*
7. *Il resterait à interpréter ceci !*

1. https://fr.wikipedia.org/wiki/R%C3%A8gle_68-95-99,7