Master SID
Année universitaire 2021-2022

TP5 Apprentissage Automatique 2

Étude comparative

simon.bernard@univ-rouen.fr

L'objectif de ce TP est de comparer plusieurs méthodes d'apprentissage sur plusieurs bases de données de classification. Pour cela nous proposons 5 bases de données, toutes disponibles en téléchargement au format CSV sur la plateforme UniversiTICE et dont voici une description succincte :

- *heart* : 270 instances, 13 caractéristiques, 2 classes
- *diabetes* : 768 instances, 8 caractéristiques, 2 classes
- *vehicle* : 846 instances, 19 caractéristiques, 4 classes
- *segment* : 2310 instances, 20 caractéristiques, 7 classes
- *spambase* : 4601 instances, 58 caractéristiques, 2 classes

Nous proposons également de comparer 3 méthodes que vous avez vu en cours :

- Régression logistique
- SVM linéaire
- SVM avec noyau RBF

Enfin, nous fournissons en fin d'énoncé le squelette du script python qui implémente le protocole expérimental de cette comparaison. Pour ce TP, il vous est demandé de compléter ce script pour :

1. Charger les données et proposer un pré-traitement si nécessaire
2. Proposer une procédure de découpage des bases de données respectant les principes vus en cours
3. Proposer pour chaque méthode un protocole de sélection d'hyper-paramètres pour que chacune d'elle soit comparée avec son plein potentiel en généralisation
4. Choisir une ou plusieurs mesures de performances pour estimer les capacités de généralisation des trois méthodes sur chaque base
5. Présenter, analyser et commenter les résultats obtenus

Quelques conseils pour vous guider :

- Les temps d'exécution pourraient devenir long en fonction de votre protocole expérimental, il est conseillé de faire les premiers tests sur la base de données la plus petite, avant de lancer tout le protocole sur toutes les bases.
- Il est également conseillé de sauvegarder les résultats intermédiaires au fil de l'exécution de votre protocole, pour éviter d'avoir à relancer l'ensemble des traitements en cas de problème.
- Il est important de soigner la présentation des résultats pour mieux appuyer l'analyse. Chaque tableau ou graphique proposé doit être accompagné d'un commentaire textuel et d'une légende pour permettre au lecteur de comprendre ce qui est représenté et ce que cela illustre.

```
import time


def load_CSV_dataset(name):
   # TODO
   pass

def split_dataset(X, y):
   # TODO
   pass

def run_logreg(data):
   # TODO
   pass

def run_linsvm(data):
   # TODO
   pass

def run_rbfsvm(data):
   # TODO
   pass

def process_results(res_logreg, res_linsvm, res_rbfsvm):
   # TODO
   pass


if __name__ == '__main__':
   ds_names = ["heart", "diabetes", "vehicle", "segment", "spambase"]
   res_logreg = []
   res_linsvm = []
   res_rbfsvm = []

   for name in ds_names:
      start = time.time()
      name += ".csv"
      print(name)
      X, y = load_CSV_dataset(name)
      data = split_dataset(X, y)
      res_logreg.append(run_logreg(data))
      res_linsvm.append(run_linsvm(data))
      res_rbfsvm.append(run_rbfsvm(data))
      end = time.time()
      print(end - start)

   process_results(res_logreg, res_linsvm, res_rbfsvm)
```