Option : Ml OIVM
Module : TP ATDN

TP5 : Approches avancées pour la classification

Objectif :
Ce TP explore des techniques avancées pour la classification binaire, incluant des approches avec SVM non linéaires, ensembles d'apprentissage (Random Forest et Gradient Boosting), et modèles probabilistes avancés comme Hidden Markov Models (HMM). Vous travaillerez également sur l'analyse approfondie des résultats à l'aide de visualisations et d'interprétations.

Exercice 1 : Classification avec SVM non linéaires (noyau RBF)

Expérimenter avec des SVM utilisant des noyaux non linéaires pour améliorer les performances.

1. Chargez et prétraitez le dataset SMS Spam Collection.
2. Divisez les données en ensembles d'entraînement et de test (70/30). Pourquoi le choix du ratio est-il important ?
3. Entraînez un modèle SVM avec un noyau RBF (kernel='rbf'). Ajustez les paramètres C et gamma via GridSearchCV.
4. Évaluez les performances (précision, rappel, F1-score) et comparez-les à un SVM linéaire.
5. Tracez la matrice de confusion et interprétez les résultats.

Exercice 2 : Ensembles d'apprentissage (Random Forest et Gradient Boosting)

Comprendre comment les algorithmes d'ensemble améliorent la robustesse des modèles.

1. Entraînez un modèle *Random Forest* sur le dataset. Quels sont les hyperparamètres importants à ajuster ?
2. Entraînez un modèle *Gradient Boosting* (par exemple, avec *XGBoost*). Comparez sa complexité avec celle du Random Forest.
3. Évaluez les deux modèles en utilisant les métriques classiques (précision, rappel, F1-score).
4. Tracez la courbe ROC-AUC pour les deux modèles. Lequel est le plus performant ?
5. Quels sont les avantages des modèles d'ensemble pour des données déséquilibrées comme SMS Spam Collection ?

Exercice 3 : Modèles probabilistes avancés (Hidden Markov Models)

Appliquer les HMM pour la classification et analyser leurs limites.

1. Expliquez brièvement le fonctionnement des HMM. Dans quel contexte sont-ils adaptés ?
2. Vectorisez les données de manière séquentielle (basée sur la fréquence des mots ou les séquences).
3. Implémentez un modèle HMM pour différencier les classes "spam" et "ham".

4. Évaluez les performances du HMM en utilisant les métriques classiques et comparez-les aux autres modèles explorés dans ce TP.
5. Discutez des défis liés à l'utilisation des HMM pour ce type de données.

Exercice 4 : Analyse des modèles avec SHAP (SHapley Additive exPlanations)

Interpréter les décisions des modèles en identifiant les caractéristiques influentes.

1. Installez la bibliothèque SHAP et utilisez-la pour expliquer les décisions prises par un modèle *Random Forest* sur le dataset.
2. Identifiez les mots les plus influents pour prédire le spam ou le ham.
3. Visualisez les valeurs SHAP pour un échantillon de données (diagramme de force, résumé des caractéristiques).
4. Discutez des limites de l'interprétabilité pour des modèles complexes.
5. Que pouvez-vous conclure sur la pertinence des caractéristiques utilisées pour la classification ?

Exercice 5 : Comparaison globale des modèles et pratique avancée

Comparer les performances des différents modèles explorés et explorer la combinaison de modèles.

1. Comparez tous les modèles testés (SVM linéaire, SVM RBF, Random Forest, Gradient Boosting, HMM) en termes de précision, rappel, F1-score, ROC-AUC, temps d'entraînement et de prédiction.
2. Entraînez un *Stacking Classifier* combinant les modèles les plus performants (par exemple, Gradient Boosting et SVM).
3. Testez différentes combinaisons de modèles pour améliorer les performances globales.
4. Discutez des cas où un modèle peut surpasser les autres (par exemple, données déséquilibrées ou limites computationnelles).
5. Proposez une approche optimisée et argumentez votre choix.

Exercice 6 : Classification avec Réseaux de Neurones Artificiels (ANN):

Explorer les réseaux de neurones artificiels (ANN) pour la classification binaire, en se concentrant sur l'architecture d'un réseau de neurones simple et son optimisation via des techniques modernes.

1- Chargez le dataset SMS Spam Collection et effectuez la vectorisation des données textuelles en utilisant un TfidfVectorizer.

2- Divisez les données en ensembles d'entraînement et de test (70/30).

3- Créez un réseau de neurones artificiels (ANN) à l'aide de Keras avec une architecture de base : une couche d'entrée, une ou deux couches cachées, et une couche de sortie avec une fonction d'activation sigmoïde.

4- Quel est le rôle de chaque composant du réseau ? Pourquoi utilisez-vous une fonction d'activation sigmoïde dans la couche de sortie ?

5- Entraînez votre modèle avec les données d'entraînement.

6- Utilisez une fonction de perte binary_crossentropy et un optimiseur comme adam.

7- Quelles sont les raisons d'utiliser binary_crossentropy et adam dans ce cas ?

8- Évaluez les performances du modèle sur l'ensemble de test à l'aide des métriques classiques (précision, rappel, F1-score, AUC-ROC).

9- Affichez la courbe ROC-AUC et discutez de la performance du modèle.

10- Optimisez les hyperparamètres du réseau (nombre de couches, nombre de neurones par couche, taux d'apprentissage) à l'aide de GridSearchCV ou RandomizedSearchCV.

11- Quelle influence ont ces hyperparamètres sur les performances du modèle ?

Discussion :

IMPORTANT : Dans le rapport faites une analyse **S.W.O.T** et un synoptique de bilan de TOUS les modèles vu lors des TPs

Remarques :

- URL de l'ensemble de données SMS Spam Collection :

  https: archive.ics.uci.edu/ml/machine-leaming-databases/00228 smsspamcollection.zip

- Utiliser les bibliothèques scikit-leam, matplotlib, numpy et autres au besoin.
- Documenter le code de manière claire.