Maîtriser l'Apprentissage Automatique

1 Description

Dans le monde actuel axé sur les données, la capacité à exploiter la puissance du big data pour obtenir des informations et prendre des décisions est une compétence précieuse. Ce projet offre une opportunité passionnante aux étudiants de plonger dans le monde de l'analyse de données en utilisant Python et des ensembles de données du monde réel, tels que ceux disponibles dans le répertoire de l'apprentissage automatique de l'UCI. Dans ce projet, les étudiants exploreront le monde multifacette de la science des données en travaillant avec deux types distincts jeux de données : tabulaires ou séries temporelles et images.

2 Objectifs du projet

Ce projet est conçu pour fournir une compréhension des techniques de science des données et d'apprentissage automatique, ainsi que de leur applicabilité à différentes modalités de données. Plus de détails :

- **Exploration des données** : Vous apprendrez à naviguer et à comprendre des jeux de données complexes, en acquérant des informations sur les structures, les motifs et les anomalies des données.
- **Prétraitement des données** : Vous découvrirez des techniques pour nettoyer, prétraiter et transformer les données brutes/images en un format utilisable, garantissant la qualité des données.
- **Modélisation d'apprentissage automatique** : Vous appliquerez des algorithmes d'apprentissage automatique pour résoudre deux types de tâches différentes, telles que la classification/régression et le regroupement.
- **Visualisation** : Vous créerez des visualisations informatives des données pour présenter efficacement vos résultats.
- **Interprétation** : Vous développerez les compétences nécessaires pour interpréter les résultats du modèle et tirer des informations significatives des données.

3 Jeu de données

Ce projet concerne l'apprentissage automatique avec Python en utilisant des jeux de données du répertoire de l'apprentissage automatique de l'UCI (ou de Kaggle). Voici une liste d'ensembles de données volumineux possibles qui conviennent à diverses tâches d'apprentissage automatique. Dans ce projet, chaque étudiant travaillera avec deux ensembles de données distincts et de grande taille : le premier pour un problème de **classification** et le deuxième pour un problème de **régression**.

Les deux jeux de données doivent être tabulaires ou sous forme de séries temporelles, ce que l'on rencontre couramment dans divers domaines, notamment la finance, les soins de santé et le marketing. Vous explorerez une gamme diversifiée de tâches d'apprentissage automatique supervisées et non supervisées applicables aux données tabulaires ou séries temporelles. Ces tâches peuvent englober la régression, la classification, le regroupement (clustering) et la détection d'anomalies. En travaillant avec des données tabulaires ou séries temporelles, vous développerez une compréhension approfondie de la manipulation/visualisation de données 1D, de l'ingénierie des caractéristiques (features), de la sélection de modèles et de l'ajustement

(tuning) des hyperparamètres, autant d'aspects cruciaux des sciences des données et de l'apprentissage automatique dans des scénarios réels.

3.1 Jeux de données

Ci-dessous se trouvent quelques ensembles de données de type tabulaires , séries temporelles. Vous avez la liberté de choisir n'importe lequel, à condition qu'il n'ait pas déjà été choisi par un autre collègue.

1. Wind Turbine SCADA Data For Early Fault Detection.
2. Performance of an Electrical Distribution Network with Soft Open Point during a Grid Side Fault.
3. GPVS-Faults: Experimental Data for fault scenarios in grid-connected PV systems under MPPT and IPPT modes.
4. PaSTS An Operational Dataset for Domestic Solar Thermal Systems.
5. A semi-labelled dataset for fault detection in air handling units from a large-scale office. Nous avons également trouvé un autre ensemble de données pour le même problème..
6. Industrial Internet of Things embedded devices fault detection and classification. A case study.
7. Desert Knowledge Australia Solar Centre Fault Data.
8. Load Fault Arc Detection Data.
9. A dataset of cyber-induced mechanical faults on buildings with network and buildings data.
10. Data underlying the research of Innovative control model and strategy development and applications to MSFR.
11. Development of Anomaly Detectors for HVAC Systems using Machine Learning.
12. The vibration sensor on railway lines.
13. Residential Power and Battery Data.
14. Prediction of energy consumption in Mexico by integrating environmental, economic, and energy data using artificial neural network models.
15. Rice-irrigation automation using a fuzzy controller and weather forecast.
16. LoadSignatures_CR.
17. Multilayer Cyberattacks Identification and Classification Using Machine Learning in Internet of Blockchain (IoBC)-Based Energy Networks.
18. Theft detection in smart grid environment.
19. Smart Meter Data-Driven Evaluation of Operational Demand Response Potential of Residential Air Conditioning Loads.
20. Load-driven interactions between energy efficiency and demand response on regional grid scales: Data inputs.

21. Cyberattacks Patterns in Blockchain-Based Communication Networks for Distributed Renewable Energy Systems: A study on datasets.
22. Black-box Dynamic Modeling of Smart Inverters : Dataset and Models.
23. Predicting Failures of Autoscaling Distributed Applications.
24. fault detection and diagnosis of an air handling unit from a real industrial facility.
25. Automatic Machine Condition Monitoring and Maintenance System in Limited Resource Situations.
26. Data Anomaly Detection in Cyber-Physical Energy Systems.
27. PSML: A Multi-scale Time-series Dataset for Machine Learning in Decarbonized Energy Grids.
28. Transmission line data of different fault instances retrieved through Phasor Measurement Unit (PMU).
29. VSB Power Line Fault Detection

Vous pouvez également consulter les jeux de données existants sur le site du CIC ou sur mendeley.

En effet, la majorité de ces jeux de données sont connus pour leur taille et leur complexité, ce qui en fait d'excellents choix pour les projets impliquant l'analyse de big data et l'apprentissage automatique. D'autre part, l'une des principales distinctions de ce projet réside dans le contraste entre les deux ensembles de données. Alors que les ensembles de données d'images reposent fortement sur les modèles d'apprentissage en profondeur, les données tabulaires et les séries temporelles nécessitent l'utilisation d'un éventail plus large de techniques d'apprentissage automatique, y compris des algorithmes traditionnels tels que les arbres de décision, les forêts aléatoires (Random Forest) et les machines à vecteurs de support (SVM), entre autres. Cette approche double vous permettra d'apprécier les forces et les limites des différentes méthodes d'apprentissage automatique et souligne l'importance de choisir les meilleures méthodes en fonction des types de données et des tâches spécifiques.

4 Tâches à accomplir

Ce qui suit est une série de questions liées à divers aspects du prétraitement des données, de l'apprentissage automatique et du post-traitement pour les deux ensembles de données.

4.1 Data Loading: Tabular/time series data set

1. Chargez le jeu de données sélectionné (par exemple, dans un DataFrame Pandas) en utilisant des fonctions appropriées telles que ***pd.read_csv()*** dans le cas d'un jeu de données tabulaire ou de séries temporelles enregistré au format CSV.
2. Combinez les données provenant de plusieurs sources ou fichiers, si possible, afin d'enrichir l'ensemble de données.
3. Effectuez des tâches de pré-traitement des données telles que la gestion des valeurs manquantes, les conversions de types de données et le nettoyage des données.
    - Vérifiez les valeurs manquantes en utilisant ***df.isna()*** et traitez-les en les imputant ou en les supprimant. Assurez-vous d'expliquer les raisons derrière votre décision d'utiliser des techniques d'imputation ou de suppression pour traiter les données manquantes.

- Convertissez les types de données au besoin en utilisant ***df.astype()***.
- Nettoyez les données manquantes en supprimant les doublons à l'aide de ***df.drop_duplicates()*** et en corrigeant les valeurs incohérentes. Vous devez indiquer quelle technique a été appliquée. Si vous choisissez l'imputation, précisez quelle méthode spécifique vous trouvez la plus appropriée pour votre jeu de données et pourquoi.
- Créez de nouvelles fonctionnalités ou transformez celles existantes pour améliorer la qualité et la pertinence des données (si possible).

4.2 Analyse exploratoire des données (EDA)

Profilage des données pour obtenir des informations sur leur distribution, leurs relations, leurs statistiques sommaires et les éventuels problèmes de qualité des données (données aberrantes).

Dans cette partie, vous utiliserez Matplotlib, Seaborn ou Plotly pour créer une variété de graphiques, notamment :

- Des graphiques linéaires pour visualiser les tendances au fil du temps.
- Des graphiques de dispersion (scattering) pour identifier les relations entre les variables numériques.
- Des graphiques à barres pour comparer les données catégorielles.
- Des cartes thermiques pour montrer les corrélations entre les variables.
- Créez des visualisations interactives à l'aide de bibliothèques comme Plotly pour améliorer l'expérience utilisateur.

Ensuite, vous devez :

1. Calculer des statistiques sommaires (à l'aide de fonctions comme "***df.describe()***") pour comprendre les tendances centrales de jeu de données et les distributions en utilisant des histogrammes, des tracés de densité (KDE) et d'autres visualisations pour visualiser la distribution des données.
2. Les valeurs aberrantes peuvent avoir un impact significatif sur l'analyse et les performances. Déterminez si les valeurs aberrantes sont des points de données valides ou des erreurs, et gérez-les en conséquence. Vous pouvez gérer les valeurs aberrantes en les visualisant à l'aide de box plots et en décidant de les conserver ou de les supprimer. Les box plot permet de visualiser les quartiles, la médiane et les valeurs aberrantes dans les données.
3. Calculez les matrices de corrélation et tracez-les pour identifier les relations entre les variables numériques. Réalisez également un graphique entre les variables et la variable cible (dans le cas de la régression/classification).
4. Utilisez des techniques appropriées de détection d'anomalies (non-supervisées) telles que Isolation Forest ou One-Class SVM pour identifier les valeurs aberrantes/anomalies dans l'ensemble de données. Visualisez et analysez les anomalies détectées. Quelles conclusions pouvez-vous en tirer?

4.3 Manipulation des données

1. Appliquez la regroupement (grouping) et l'agrégation pour calculer des statistiques sommaires pour des catégories spécifiques. Vous pouvez utiliser la fonction ***groupby()*** et des fonctions d'agrégation telles que ***sum()***, ***mean()***, et ***count()*** pour créer des tables de synthèse.

2. Effectuez des opérations de filtrage (à l'aide de l'indexation booléenne basée sur des conditions spécifiques) et des opérations de tri pour extraire des sous-ensembles de données (afin d'obtenir des informations).

3. Appliquez des techniques d'analyse de séries temporelles pour découvrir des tendances temporelles dans le cas d'un jeu de données de séries temporelles. Utilisez des moyennes mobiles (rolling averages) ou d'autres fonctions de séries temporelles pour lisser le bruit dans les données.

4.4 Dérivation d'Informations (IA)

1. Assurez-vous de traiter toutes les valeurs manquantes ou les anomalies identifiées lors de l'examen du jeu de données.

2. Effectuez une analyse pour identifier les corrélations au sein du jeu de données sélectionné. Dans le cas d'un ensemble de données de séries temporelles, une analyse plus approfondie est nécessaire pour identifier la saisonnalité et les tendances au fil du temps.

3. Explorez des techniques de réduction de la dimensionnalité telles que l'Analyse en Composantes Principales (PCA) ou t-SNE pour visualiser des données de grande dimension.

4. Interprétez les résultats pour en tirer des conclusions significatives.

4.5 Tâches d'apprentissage automatique supervisées : ensembles de données tabulaires/séries temporelles/images

Dans cette partie, vous construirez et évaluerez des modèles d'apprentissage automatique supervisées (régression et/ou classification) pour vous assurer que les modèles entraînés sont efficaces et robustes.

1. Normalisez ou mettez à l'échelle les caractéristiques si nécessaire en utilisant des techniques telles que MinMaxScaler, StandardScaler ou RobustScaler

2. Prétraitez les données en encodant les variables catégorielles (le cas échéant) et en mettant à l'échelle les données.

3. Divisez l'ensemble de données en ensembles d'entraînement et de test.

4. Entraînez au moins six modèles de classification différents sur l'ensemble d'entraînement (par exemple, Decision Tree, Random Forest, Extra-Trees, CatBoost, LightGBM, XGBoost, Multi-Layer Perceptron) en utilisant la bibliothèque Python Scikit-Learn et/ou des modèles d'apprentissage profond tels que Keras ou PyTorch.

5. Évaluez les performances de chaque modèle sur l'ensemble de test en calculant des métriques pertinentes telles que l'exactitude, la précision, le rappel et le score F1 dans le cas de la tâche de classification. D'autres métriques sont utilisées pour la tâche de régression, telles que l'erreur quadratique moyenne (MSE), l'erreur absolue moyenne (MAE), l'erreur absolue pourcentage (MAPE), et le coefficient de détermination R^2.

6. Sélectionnez le meilleur modèle en fonction des métriques d'évaluation et expliquez pourquoi il a performé mieux que les autres.

7. Visualisez les performances du modèle sélectionné en utilisant les techniques de visualisation appropriées.

8. Appliquez différentes méthodes d'ajustement des hyperparamètres telles que la recherche en grille, la recherche aléatoire et l'optimisation bayésienne. Quelles conclusions pouvez-vous en tirer ?

9. Évaluez les performances de chaque modèle ajusté et non ajusté sur le jeu de test en calculant les métriques correspondants.

4.6 Tâches avancées d'apprentissage automatique supervisé

1. Comprenez les concepts et les avantages de chaque technique d'ensemble, tels que le vote, le bagging et le stacking.

2. Appliquez ces techniques d'ensemble avec et sans ajustement des hyperparamètres.

3. Fournissez une comparaison des modèles entre les modèles individuels ajustés et les modèles d'ensemble ajustés en évaluant et en comparant leurs performances en utilisant l'exactitude, la précision, le rappel et le score F1. Quelles conclusions pouvez-vous en tirer ?

4. Fournissez une comparaison des modèles entre les modèles d'ensemble ajustés et non ajustés en évaluant et en comparant leurs performances en utilisant l'exactitude, la précision, le rappel et le score F1. Quelles conclusions pouvez-vous en tirer ?

5. Calculez les scores d'importance des caractéristiques pour les modèles individuels et les modèles d'ensemble. De plus, vous devriez visualiser et comparer la contribution des caractéristiques au processus de classification/régression pour les meilleurs modèles individuels et d'ensemble.

6. Explorez les valeurs SHAP (SHapley Additive exPlanations) ou d'autres techniques d'interprétabilité. Mettez en œuvre ces techniques pour expliquer les décisions prises par les modèles individuels et d'ensemble ajustés. Interprétez les résultats des analyses et traduisez-les en informations exploitables.

4.7 Tâches d'apprentissage automatique non supervisées : Clustering

Dans cette partie, nous considérerons que le jeu de données n'est pas étiqueté (éliminez la variable cible [label ou target]). Vous effectuerez des analyses de regroupement avancées (pour regrouper des points de données similaires) afin d'extraire des informations plus approfondies. À cette étape, vous devrez tester l'ensemble de données sélectionné avec des techniques de réduction de la dimensionnalité telles que PCA , TSNE ou avec/sans auto-encodeur pour visualiser des données de grande dimension.

1. Appliquez plusieurs algorithmes de regroupement (par exemple, K-Means, modèles de mélange gaussien (GMM), DBSCAN, clustering hiérarchique) sur les données transformées.

    - Énumérez les métriques existantes utilisées pour détecter le nombre optimal de clusters. Ensuite, utilisez ces techniques pour déterminer le nombre optimal de clusters sur l'ensemble de données sélectionné. Pouvez-vous les classifier ?
    - Regroupez l'ensemble de données en utilisant le nombre optimal de clusters détecté.
    - Visualisez les clusters et interprétez les résultats.
    - Évaluez la qualité du regroupement à l'aide de métriques appropriées (par exemple, score de silhouette, indice Davies-Bouldin).

- Implémentez le clustering hiérarchique en utilisant la liaison simple (single linkage) et la liaison complète (complete linkage). Visualisez ensuite le dendrogramme hiérarchique. En outre, quelles sont les différences entre la liaison simple et la liaison complète ? Que pouvez-vous conclure des résultats obtenus ?
- Adaptez un modèle de mélange gaussien (GMM) aux données. Visualisez les composants du GMM et leurs paramètres. Ensuite, énumérez les avantages du GMM par rapport à K-Means.
- Appliquez le clustering basé sur la densité (DBSCAN) pour découvrir des clusters dans l'ensemble de données sélectionné. Ensuite, déterminez les hyperparamètres optimaux (par exemple, epsilon et min_samples). Ensuite, visualisez les clusters DBSCAN et les points de bruit. De plus, énumérez les forces et les faiblesses de DBSCAN.

2. Examinez et évaluez les performances de regroupement de différentes méthodes d'apprentissage non supervisé en utilisant à la fois des métriques de validation internes et externes, telles que l'indice Rand ajusté. Explorez également les caractéristiques de l'ensemble de données qui influencent le choix de la technique à utiliser.

4.8 Documentation et Présentation

- Créez un fichier Jupyter notebook qui documente toutes les étapes du projet. Vous devriez documenter l'ensemble du processus, y compris la pré-traitement des données, la mise en œuvre du code et les résultats.
- Maintenez une documentation complète pour assurer la transparence et la reproductibilité. Cette documentation peut vous aider, ainsi que d'autres, à comprendre le jeu de données et les étapes pour en améliorer la qualité des données.
- Préparez une présentation claire et concise qui résume et met en évidence les principales solutions, améliorations réalisées et conclusions.

5 Résultats Attendus:

- Maîtrise de Python pour l'analyse de données et l'apprentissage automatique.
- Expérience pratique de travail avec des ensembles de données du monde réel.
- Compétences en visualisation de données pour transmettre des informations complexes.
- Capacité à appliquer des techniques évolutives pour l'analyse de big data.
- Amélioration des compétences en résolution de problèmes et en pensée critique.

6 conclusion

À la fin de ce projet, vous disposerez des compétences et de la confiance nécessaires pour relever des défis du monde réel en matière d'apprentissage automatique.

Bon Travail!