Machine Learning - TP1

Notions de Base sur les Données, Prétraitement des Données, Introduction à l'Apprentissage Automatique avec Python et scikit-learn.

Avant de quitter la classe, veuillez enregistrer votre notebook sous le nom de `TP1 <Prénom><Nom>.ipynb' et le soumettre via la plateforme Moodle.

Installations et Tutoriel

Pour nos TPs, nous utiliserons Python 3, la bibliothèque d'apprentissage automatique Scikit-learn et Jupyter Notebook. Scikit-learn est choisi pour être le logiciel d'apprentissage automatique leader en Python. Il est facile à utiliser et à étendre, et il est open-source. Jupyter Notebook favorise une nouvelle forme de programmation appelée programmation littéraire. Il est donc facile à utiliser pour des objectifs de démonstration, de recherche et d'enseignement, notamment pour les sciences.

Pour une installation facile de notre environnement de travail, nous vous suggérons de télécharger la plateforme Anaconda, pour Python 3.7. Cela installera également Jupyter Notebook.

Pour télécharger et installer Anaconda, suivez le lien : https://www.anaconda.com/distribution/.

Pour plus d'informations sur l'installation de Scikit-learn, suivez le lien : http://scikit-learn.org/stable/install.html.

Remarque : Scikit-learn est construit sur Python, ainsi que sur les bibliothèques NumPy et SciPy. Il est donc nécessaire de les avoir également installés sur votre machine. Mais ne vous inquiétez pas ! Si vous avez installé Anaconda avec succès, ces packages sont également installés !

Familiarisation avec Python et NumPy

Ouvrez Anaconda Navigator et lancez Jupyter Notebook. Créez un nouveau Notebook (Python3) avec le nom `TP1' dans le répertoire de votre choix. Dans ce notebook, vous allez écrire votre code ainsi que tout texte/commentaire que vous souhaitez inclure pour décrire votre code ou le résultat.

Pour vous aider à commencer avec Python et NumPy, il existe un excellent tutoriel en ligne, créé à Stanford. Il peut être téléchargé sous forme de notebook à l'adresse https://github.com/kuleshov/cs228-material/find/master (sous le nom : `cs228-python-tutorial-3 7.ipynb').

Téléchargez le tutoriel dans votre bibliothèque Jupyter (bouton Télécharger). Prenez un peu de temps et parcourez les sections suivantes :

Python (optionnel - si vous n'êtes pas très familier avec Python) :

- Types de données de base
- Containers

NumPy :

- Array : très couramment utilisés dans les applications Scikit-learn.
- Datatypes
- Array math

Enfin, nous utiliserons **Matplotlib** pour tracer des diagrammes. Donc, jetez un œil à cette section également.

Exercices - Préparation des Données

Exercice 1: Acquisition de Données

- Nous allons travailler avec un ensemble de données réelles sur les prix immobiliers (vente de maisons, appartements, etc.) de 2016 à 2020 en France, disponibles auprès du gouvernement français sur https://www.data.gouv.fr/fr/datasets/demandes-de-valeurs-foncieres/.
- Téléchargez les données pour 2019 et les données disponibles pour 2020. Téléchargez également la description des données (`Notice descriptive des fichiers de valeurs foncières').
- Prenez un moment pour comprendre le schéma des données et pour voir quels problèmes vous pouvez identifier dans les données qui doivent être résolus avant de les utiliser dans un pipeline ML.
    a) Listez les problèmes que vous observez et écrivez quelle solution vous pourriez appliquer pour chacun.
    b) Chargez les fichiers txt dans votre notebook en utilisant la bibliothèque pandas (import pandas as pd) et la fonction read_csv(). La sortie sera des objets DataFrame. Pour plus de détails, voir http://pandas.pydata.org/pandas-docs/stable/io.html#io-read-csv-table.
    c) Convertissez l'objet DataFrame en un tableau NumPy en utilisant DataFrame.to_numpy(). Intégrez les deux ensembles de données en un seul.

Exercice 2: Nettoyage des Données - Valeurs Manquantes

1. Écrivez une fonction qui met à jour les valeurs manquantes d'une caractéristique d'un ensemble de données, en utilisant la moyenne ou le mode, en fonction de ce que l'utilisateur demande.
2. Pour quelles caractéristiques de votre ensemble de données jugeriez-vous pertinent d'utiliser cette technique ? Pour quelles caractéristiques est-il plus approprié de supprimer la caractéristique complètement ?
3. Traitez vos données pour qu'aucune valeur manquante n'apparaisse. Remarque : si vous optez pour l'appel de la fonction que vous avez créée, réfléchissez si regrouper les données par rue ou par ville vous donnera de meilleurs résultats.

Exercice 3: Analyse Statistique - Variance - Covariance

1. Calculez la matrice de covariance pour les attributs `Valeur foncière', Nombre pièces principales', Surface terrain' et Surface réelle bâtie'` en utilisant la fonction numpy.cov(array, rowvar=False). En voyant la matrice de covariance, croyez-vous qu'il y ait

une relation entre les attributs ? Vérifiez la force de la dépendance (s'il en existe une), en calculant également la matrice de corrélation en utilisant numpy.corrcoef(array, rowvar=False). Remarque : définir le paramètre (rowvar) sur False est important ! Sinon, chaque ligne sera considérée comme une caractéristique et chaque colonne comme un point de données.

2. Tracez chacune des paires de caractéristiques de la question précédente, par exemple, (`valeur foncière, Nombre pièces principales`), (`valeur foncière, Surface réelle bâtie`), etc., en utilisant matplotlib.pyplot.scatter(vector, vector). Observez-vous graphiquement les conclusions que vous avez tirées précédemment ?

Exercice 4: Nettoyage des Données - Détection des Valeurs Aberrantes

- Trouvez et supprimez les valeurs aberrantes. Une valeur aberrante dans ce contexte peut être une maison/appartement dont la valeur par mètre carré a une grande distance (c'est-à-dire au-dessus d'un seuil que vous définirez) par rapport à la moyenne si les données sont biaisées.

Exercice 5: Reformatage des Données - Encodage One Hot

1. Créez une fonction qui, étant donné un ensemble de données, une caractéristique catégorielle et une liste de valeurs correspondant aux valeurs acceptées de la caractéristique catégorielle, transforme les données en encodage one hot.
2. Appelez la fonction dans votre code pour la tester sur la caractéristique 'Type de local'.

Exercice 6: Réduction de la Numérosité - Échantillonnage

- Effectuez un échantillonnage aléatoire avec remplacement (sklearn.utils.resample(x, n_samples=NoOfsamples, random_state=0)) après avoir mélangé les données (np.random.shuffle(x)) pour conserver 60 échantillons.

Exercice 7: Transformation des Données - Normalisation

- Maintenant, normalisez les attributs numériques dans la plage 0-1 en utilisant le MinMaxScaler de sklearn.preprocessing.

Ces exercices couvrent divers aspects du prétraitement des données, y compris la gestion des valeurs manquantes, des valeurs aberrantes, des caractéristiques catégorielles et la transformation des données, essentiels pour préparer les données aux pipelines d'apprentissage automatique.