

SAE S4: Module de Machine Learning (ML)

Il vous est demandé d'incorporer un module de **Machine Learning (ML)** à votre application web. Ce projet comporte trois parties décrites ci-dessous.

- La SAÉ est à rendre pour le **17 avril 2023**.
- En plus de votre **code**, prière de joindre un **rapport pdf** ou un **fichier md** qui décrit votre approche du problème, démarche, modèle et solution s'intégrant.
- Prière de déposer vos rendus sur **e-Campus**.

Partie 1: Modèle ML

Concevoir et entraîner un **modèle de machine learning** pour un problème d'**apprentissage supervisé** de type **régression** ou **classification**. Si vous désirez vous lancer dans un problème plus compliqué, par exemple d'**apprentissage par renforcement**, sentez-vous libre.

Le modèle devra être implémenté en `scikit-learn` (bibliothèques utilisées en cours S4.04) et entraîné sur un dataset public (par exemple issu du site Kaggle) ou un dataset que vous aurez créé vous-même (ce qui demande plus de travail). Par la suite, le modèle sera utilisé pour obtenir des prédictions à partir de données que vous aurez récupérées (scrapées) sur le web (cf. partie 2).

<https://scikit-learn.org/stable/>

<https://www.kaggle.com/>

***Exemple:** Créer un modèle qui prend comme inputs des textes à contenus financiers et qui prédit leur sentiment. Ce modèle sera entraîné sur un dataset prévu à cet effet (cf. Kaggle, sentiment datasets):*

<https://www.kaggle.com/datasets?search=sentiment+analysis>

Pour cette partie, il vous est demandé de fournir un **joli notebook jupyter** et un **fichier de data** contenant l'implémentation, l'entraînement et la sauvegarde de votre modèle ML.

Partie 2: Web scraping

Créer un **script d'extraction de données (data)** qui va chercher des données qui vous intéressent sur un ou plusieurs sites web. Ces data seront ensuite fournies à votre modèle afin qu'il puisse en tirer des prédictions.

Il y a des bibliothèques Python pour cela (`Beautiful Soup`, `Scrapy`). Une fois extraites, il est recommandé de nettoyer et de mettre en forme les data de manière à ce qu'elles puissent être traitées par votre modèle (cf. partie 1).

<https://beautiful-soup-4.readthedocs.io/en/latest/>

<https://scrapy.org/>

***Exemple:** Créer un script qui va chercher des news financières sur le site du Financial Times.*

Pour cette partie, il vous est demandé de fournir un **notebook jupyter** ou un **fichier python** qui implémente l'acquisition de vos data.

Partie 3: Déploiement du modèle

Création et déploiement d'une **application web** (endpoint ou point de terminaison en ligne). Votre application devra être capable d'aller "scraper" des données (partie 2), de les passer à votre modèle, et retourner les prédictions associées (partie 1).

Il y a différentes méthodes pour cela que je vous laisse investiguer... Entre autre, vous pouvez utiliser `Flask` ou `Scikit.js`. Les librairies et les tutoriel ci-dessous vous donnent plus d'informations à ce sujet. N'hésitez pas à faire vos propres recherches.

<https://flask.palletsprojects.com/en/2.2.x/>

<https://towardsdatascience.com/a-flask-api-for-serving-scikit-learn-models-c8bcdaa41daa>

<https://developer.ibm.com/tutorials/deploy-a-python-machine-learning-model-as-a-web-service/>

<https://scikitjs.org/>

Exemple: Créer une petite application qui, lorsqu'on clique sur un bouton, va chercher les dernières news financière sur le site du Financial Times, passe ces new à votre modèles de sentiment, récupère les prédictions associées, puis affiche ces news et leurs sentiments sur votre site web.