

Prédiction Car Pollution

Introduction

Nous allons dans ce projet construire une API avec Fast API qui va permettre d'interroger les deux modèles : RandomForestRegressor et RandomForestClassifier entraînés afin de prédire l'émission de CO2

Problématique

La pollution au CO2 étant l'une des principales causes du réchauffement climatique, la question à laquelle cette API devra répondre est la suivante :
Estimation de rejet de CO2 d'un véhicule à partir de ses caractéristiques techniques.

Jeu de données utiliser :

- eeu.europa

Variables explicatives les plus pertinentes :

W (mm), At1 (mm), At2 (mm), Ft, ec (cm3), ep (KW), year, Enedc (g/km)

Variable cible:

Enedc (g/km)

Modèles de Machine Learning:

Utilisation de Modèle RandomForestRegressor et RandomForestClassifier

Entraînement des modèles et sauvegarde avec pickle :
exemple :

```
import pickle
pickle_out=open("modelRandomForestRegressor.pkl","wb")
pickle.dump(modelRandomForestRegressor,pickle_out)
```

Pour le modèle classifieur :

Nous allons construire différents modèles de classification. La variable cible étant à la base une variable continue, nous allons la compartimenter en classes pour pouvoir utiliser des modèles de classification. Ces classes correspondent aux classes énergétiques officielles, de manière à répondre à la problématique métier consistant à prédire la classe d'émission d'un véhicule en fonction de ses caractéristiques.

L'API

utilisation de Fast API avec les deux modèles de machine learning

Les utilisateurs autorisés sont les suivants :

username	password
alice	wonderland
bob	builder
clementine	mandarine

Les Tests

1- Test Authentication :

Trois tests sont lancés: deux tests avec pour les utilisateurs alice et clementine avec des faux mots de passe (échec) et un test pour bob:builder avec succès
résultat

```
my_authentication_from_compose |
my_authentication_from_compose | =====
my_authentication_from_compose | Authentication test
my_authentication_from_compose | =====
my_authentication_from_compose |
my_authentication_from_compose | request done at "/login
my_authentication_from_compose | | username=alice
my_authentication_from_compose | | password=data
my_authentication_from_compose |
my_authentication_from_compose | ==> FAILURE
my_authentication_from_compose |
my_authentication_from_compose |
my_authentication_from_compose |
my_authentication_from_compose | =====
my_authentication_from_compose | Authentication test
my_authentication_from_compose | =====
my_authentication_from_compose |
```

```

my_authentication_from_compose | request done at "/login
my_authentication_from_compose | | username=bob
my_authentication_from_compose | | password=builder
my_authentication_from_compose |
my_authentication_from_compose | ==> SUCCESS
my_authentication_from_compose |
my_authentication_from_compose |
my_authentication_from_compose |
my_authentication_from_compose | =====
my_authentication_from_compose | Authentication test
my_authentication_from_compose | =====
my_authentication_from_compose |
my_authentication_from_compose | request done at "/login
my_authentication_from_compose | | username=clementine
my_authentication_from_compose | | password=fauxpassword
my_authentication_from_compose |
my_authentication_from_compose | ==> FAILURE
my_authentication_from_compose |
my_authentication_from_compose |

```

2- Test Prédiction Regressor : prédire la valeur de la consommation de co2

```

/predictionmodelRandomForestRegressor HTTP/1.1" 200 OK
my_prediction_from_compose | {"Prediction Enedc (g/km)":[523.9783333333332]}
my_prediction_from_compose |
my_prediction_from_compose | =====
my_prediction_from_compose | Content test
my_prediction_from_compose | =====
my_prediction_from_compose |
my_prediction_from_compose | request done at "/predictionmodelRandomForestRegressor"
my_prediction_from_compose | | username=alice
my_prediction_from_compose | | password=wonderland
my_prediction_from_compose | |data={'W (mm)': 2456, 'At1 (mm)': 1441, 'At2 (mm)': 1434, 'Ft':
2, 'ec (cm3)': 999, 'ep (KW)': 55, 'year': 2015}
my_prediction_from_compose |
my_prediction_from_compose |
my_prediction_from_compose | ==> SUCCESS
my_prediction_from_compose |
my_prediction_from_compose |

```

3- Test prédiction Classifier : prédire la classe d'émission d'un véhicule en fonction de ses caractéristiques.

```

predictionmodelRandomForestClassifier HTTP/1.1" 200 OK
my_content_from_compose | {"Class prediction":6}
my_content_from_compose |
my_content_from_compose | =====
my_content_from_compose |         Content test
my_content_from_compose | =====
my_content_from_compose |
my_content_from_compose | request done at "/predictionmodelRandomForestClassifier"
my_content_from_compose | | username=alice
my_content_from_compose | | password=wonderland
my_content_from_compose | | data={'W (mm)': 2456, 'At1 (mm)': 1441, 'At2 (mm)': 1434, 'Ft':
2, 'ec (cm3)': 999, 'ep (KW)': 55, 'year': 2015}

```

Les tests sont builders dans des images séparés et un fichier docker-compose.yml a été créé pour faire communiquer les 4 images (api + 3 tests) pour cela il suffit de lancer le fichier setup.sh:

```
bash setup.sh
```

Déploiement Kubernetes

Le fichier my-deployment.yml contient la déclaration du Deployment

```
kubectl create -f my-deployment.yml
```

Le fichier my-service.yml contient la déclaration du Service

```
kubectl create -f my-service.yml
```

Le fichier my-ingress.yml contient la déclaration de l'Ingress

```
kubectl create -f my-ingress.yml
```

Pods									
Name	Images	Labels	Node	Status	Restarts	CPU Usage (cores)	Memory Usage (bytes)	Created	
my-project-fastapi-648655c67-9qp9v	khedidja/my_project_fastapi:latest	app: my-project-fastapi pod-template-hash: 648655c67	minikube	Running	0	-	-	49 minutes ago	
my-project-fastapi-648655c67-rdmm5	khedidja/my_project_fastapi:latest	app: my-project-fastapi pod-template-hash: 648655c67	minikube	Running	0	-	-	49 minutes ago	
my-project-fastapi-648655c67-x6fq5	khedidja/my_project_fastapi:latest	app: my-project-fastapi pod-template-hash: 648655c67	minikube	Running	1	-	-	49 minutes ago	

Interagir avec l'api en local :

```
kubectl get ingress
```

Cette commande renvoie l'adresse IP du Ingress.

```
ssh -i "data_engineering_machine.pem" ubuntu@addressmachineV -fNL 8000:192.168.49.2:80
```

Prediction Car Pollution 1.0.0 OAS3

/openapi.json

My project API to predict car pollution with modelRandomForestRegressor / modelRandomForestClassifier

Authorize



default



GET

/ Index



GET

/status Return Status "Ok" Quand Api Marche



GET

/user Current User



POST

/predictionmodelRandomForestRegressor Post Prediction



POST

/predictionmodelRandomForestClassifier Predict Car Pollution



Schemas



CarPyllution >

HTTPValidationError >

Améliorations envisagées :

- Ajouter un container pour une BDD MYSQL pour le username et password
- Ajouter le chargement de modèle pickle (.pkl) à partir de l'API