

b. Évaluation

Pour l'évaluation, il faudra réaliser une API portant sur l'analyse de sentiment. Il suffira alors de renseigner la phrase dont nous voulons étudier le sentiment associé par des requêtes HTTP. L'API sera accessible via un système de pseudo/mot de passe fourni par le fichier `credentials.csv` récupérable à l'aide de la commande suivante :

```
1 | wget https://dst-de.s3.eu-west-3.amazonaws.com/flask_fr/credentials.csv
```

De ce fichier, il est visible que l'API possède deux modèles d'analyse de sentiment et que les utilisateurs n'ont pas tous accès aux mêmes



`/v2/sentiment` devra renvoyer le `compound` d'un `VaderSentiment.SentimentIntensityAnalyzer()`. Pour plus d'informations sur ce modèle, vous pouvez vous diriger vers ce lien.

Voici un récapitulatif des routes à réaliser avec des spécificités dans les méthodes à utiliser.

- `GET /status`: renvoie 1 si l'API fonctionne.
- `GET /welcome`: renvoie un message de salutation avec le `username` indiqué par chaîne de requête.
- `POST /permissions`: renvoie la liste des permissions d'un utilisateur authentifié par son username et son password, rajoutez un en-tête à la réponse renseignant comportant le `username`, la valeur prise par `v1` et `v2`.
- `POST /v1/sentiment`: renvoie le score de sentiment par le modèle `v1` de la phrase proposée par l'argument `sentence` envoyé par `POST` si l'utilisateur s'est bien identifié par la paire (`username,password`). Cette paire devra être transmise par l'en-tête suivant `{Authorization: username=password}`, la paire devrait être encodée mais pour simplifier l'exercice, vous pouvez laisser tel quel.
- `POST /v2/sentiment`: Même fonctionnement que la précédente route, mais nous voudrions avoir le score retourné par le modèle `v2`

Une gestion des erreurs est attendue. Si un utilisateur s'est mal authentifié avec le code d'état adéquat, de même si l'utilisateur essaie d'utiliser un modèle dont il n'a pas les droits.

Rendus

Un fichier python `app.py` qui contient le programme de l'API développée avec Flask, un fichier `requirements.txt` qui représente l'environnement (Python) minimal qui permet de faire tourner cette API, ainsi qu'une documentation fournie sous forme de fichier texte ou de type pdf de votre API. Dans cette documentation, vous devrez fournir des exemples de requêtes pour pouvoir tester les différentes routes et aussi leurs résultats.

Validated