



STNP

solution IA

Documentation technique

Nom du projet	STNP Solution IA
Type de document	Documentation technique
Date	24/03/2022
Version	1.3

Table des matières

Résumé du document	3
Rappel sur le fonctionnement de l'application	4
Description de l'application	4
Décomposition du projet	4
Architecture globale	5
Le frontend	6
Technologies utilisées	6
Le backend	7
Technologies utilisées	7
L'intelligence artificielle	8
Technologies utilisées	8
L'application en exploitation	9
Démarrage du frontend	9
Surveillance des données	10
L'API REST	11
L'intelligence artificielle	12

Résumé du document

Ce document technique est la présentation des différentes parties du projets avec ses fonctionnalités, elle est divisé en 3 parties :

- Le Frontend - L'interface d'utilisateur
- Le Backend - La configuration et les fonctionnalités
- L'Intelligence Artificielle - Classement du texte

Rappel sur le fonctionnement de l'application

Description de l'application

Notre projet consiste à créer une application web, qui prendra les commentaires des utilisateurs, puis déterminera si celui-ci est un commentaire positif ou négatif.

Nous aurons une partie utilisateur, où il y aura une zone de texte pour saisir une phrase en anglais. Et, une partie administrateur où l'on verra le nombre de commentaires positifs et négatifs qui auront été saisis.

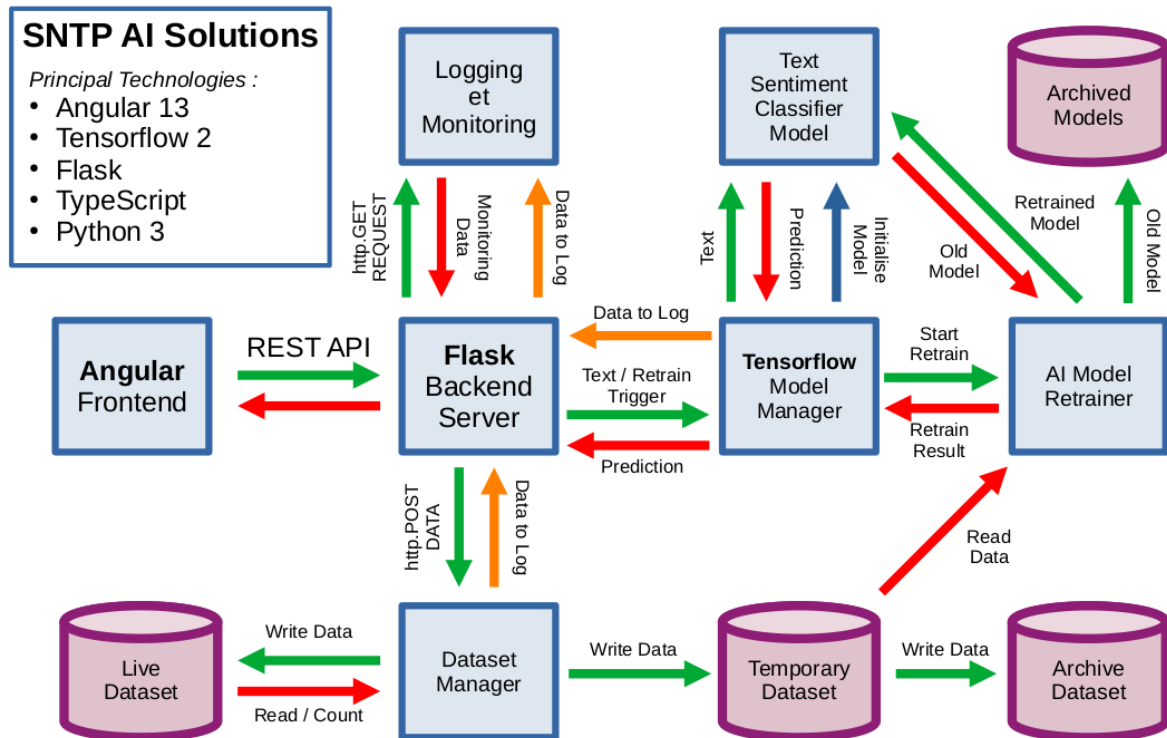
Pour savoir le sentiment du commentaire, nous avons un jeu de données où nous avons effectué un apprentissage. Tous les 1 000 commentaires, un nouvel apprentissage est fait.

Décomposition du projet

Notre projet se décompose en 3 grandes parties :

- Le frontend, qui sera l'interface graphique pour les utilisateurs et l'administrateur.
- Le backend comportera toutes les fonctionnalités reliées à l'interface.
- L'architecture de l'intelligence artificielle et modèle d'inférence d'IA.

Architecture globale



Architecture de l'application

Le frontend

Technologies utilisées

L'interface graphique a été effectuée avec le framework Angular open-source avec TypeScript. Angular et TypeScript ont été choisis par notre équipe de développement pour plusieurs raisons :

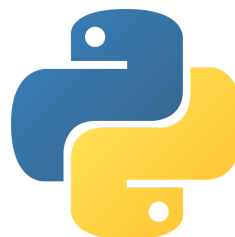
- Angular permet de créer des applications web modernes et dynamiques.
- La structure de projet appliquée par Angular et TypeScript est bien adaptée au développement en équipe.
- Son utilisation dans les produits Google (et de nombreuses autres applications Web) donne l'assurance que le cadre est à la fois fiable et de qualité professionnelle.



Le backend

Technologies utilisées

Le backend est codé en Python 3, un langage de programmation interprété. Ce langage a été choisi pour sa compatibilité croisée avec Tensorflow (utilisé pour développer notre modèle d'IA) et Flask, le framework backend de notre choix. Flask a été choisi comme framework backend car il est à la fois léger et efficace, tout en conservant une qualité de performance qui a été démontrée dans une variété d'applications professionnelles (Netflix, Uber et Airbnb). Flask et Python fonctionnent comme l'interface entre le frontend et notre modèle de classification de texte basé sur l'IA. Flask a permis à notre équipe de développement d'implémenter l'API REST standard de l'industrie pour contrôler le flux de messages entre le frontend et le backend. L'API REST garantit que notre solution fonctionnera sur tous les principaux navigateurs Web de manière sécurisée et fiable.



L'intelligence artificielle

Technologies utilisées

Le modèle d'intelligence artificielle a été développé en utilisant Python 3 et Tensorflow 2. Tensorflow a été choisi en raison de sa large utilisation dans l'industrie. Développé par des chercheurs de Google, Tensorflow offre une grande flexibilité dans le développement de modèles d'intelligence artificielle robustes. Les commandes de niveau supérieur qu'il fournit à l'aide de l'API Keras permettent à nos développeurs de tester rapidement une variété d'architectures sans avoir à recommencer à zéro à chaque fois. Cela nous permet de concentrer davantage notre énergie sur la qualité des ensembles de données et le réglage fin de nos modèles pour la performance.



L'application en exploitation

Enfin, si l'on veut retrouver le projet, il faut se rendre sur le lien github suivant : <https://github.com/Mahadia/SNTP>

Tout d'abord, il faut configurer sa machine en installant les logiciels pour veiller au bon fonctionnement du projet.

Démarrage du Frontend

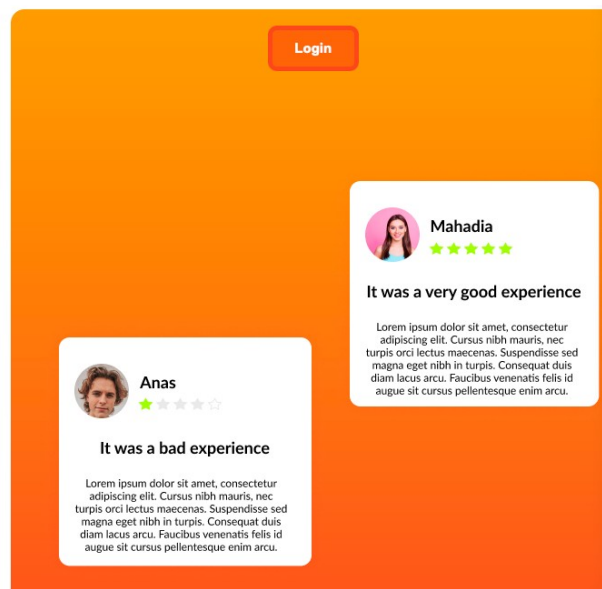
1. Pour lancer le frontend, il faut d'abord installer NodeJS, le Node Package Manager et Angular CLI.
2. Une fois cette opération terminée, accédez simplement au dossier du projet `/sntp/sntp_angular_frontend`
3. Lancer la commande **npm install**
4. Lancer la commande **npm start**
5. C'est tout, vous avez fini !

Dans un environnement de test, l'interface sera accessible à `http://localhost:4200`

Naviguer dans le frontend

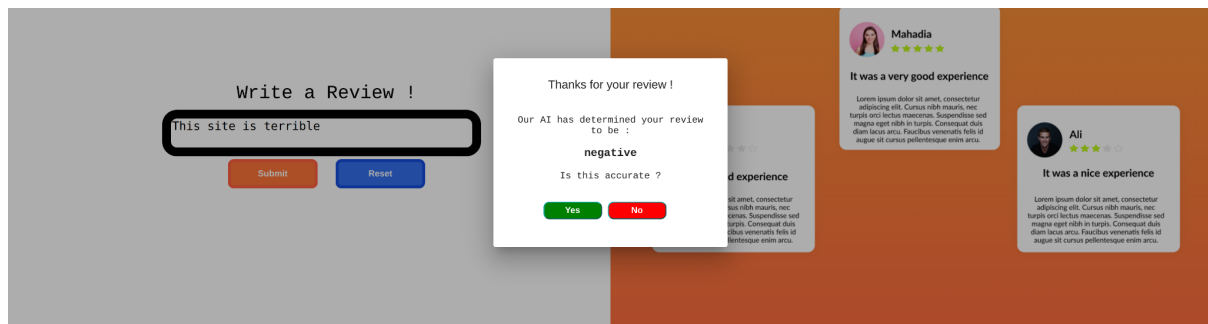


Write a Review !



la page d'accueil

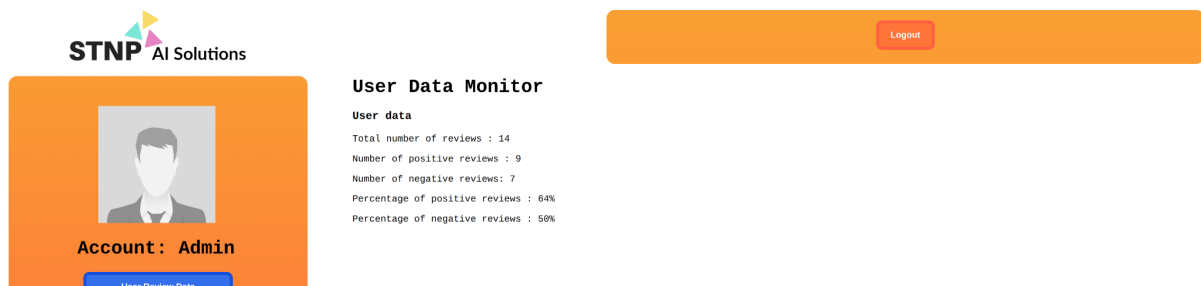
La page d'accueil permet aux utilisateurs de saisir du texte et de recevoir une prédiction.



une prédiction de sentiment a été faite

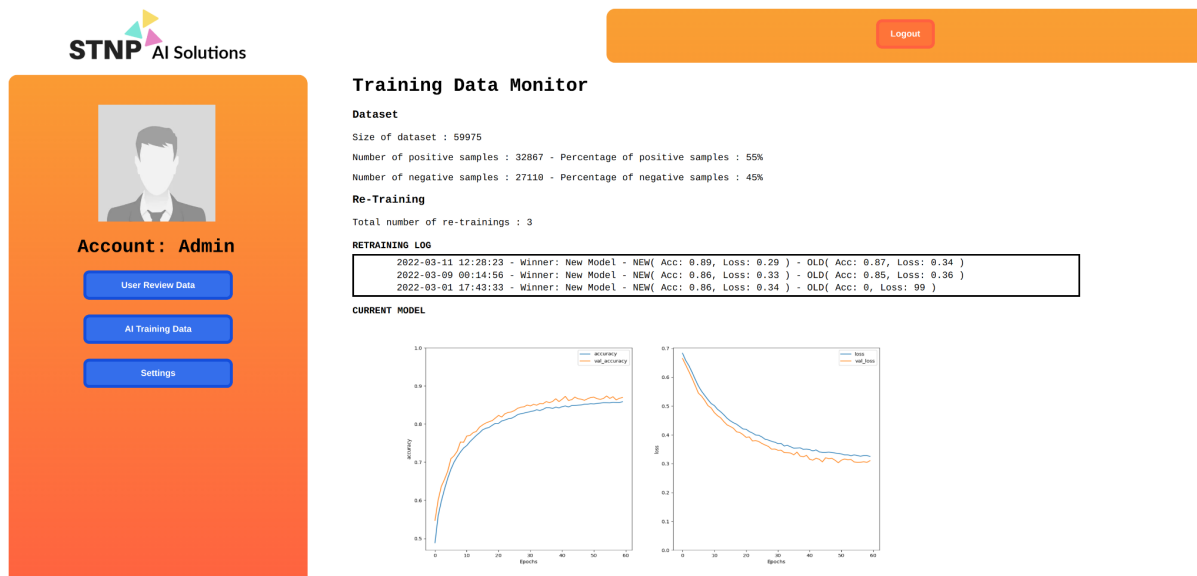
Après avoir cliqué sur le bouton soumettre, le texte de l'utilisateur est envoyé au backend et traité par notre modèle d'inférence de classificateur de texte AI. L'utilisateur peut alors corriger la prédiction si notre modèle d'IA n'a pas correctement prédit le sentiment de l'utilisateur. Cela offre plus d'engagement avec l'application tout en garantissant que la qualité de notre ensemble de données reste cohérente avec la valeur du texte réel.

Surveillance des données



Moniteur de données utilisateur

L'application fournit également une section, accessible par mot de passe, qui permet aux administrateurs de surveiller l'activité du site, la qualité des prédictions de l'IA et la qualité de l'ensemble de données au fur et à mesure de sa croissance. Il s'agit d'un outil important pour identifier les biais qui peuvent apparaître dans l'ensemble de données si, par exemple, nos utilisateurs préfèrent le positif. S'il n'est pas contrôlé, ce type de biais peut dégrader la qualité de notre modèle d'IA au fil du temps.



Surveiller les données liées à l'IA

L'API REST

Faire une prédiction

Dans la zone de texte, l'utilisateur va entrer la phrase suivante : " I'm lovin it " . Après avoir appuyé sur le bouton **Submit**, la réponse affichée sera positive, ce qui représente la prédiction, qui est bien en accord avec le sentiment du texte.

- POST : {action: "predict", text: "I'm lovin it"}
- RESPONSE : {prediction: "positive", text: "I'm lovin it"}

Enregistrer les données dans un ensemble de données

Si la réponse affichée est en accord avec le sentiment du texte, alors on devra appuyer sur le bouton **Yes** pour signifier que l'algorithme a détecté le type de commentaire et **No**, pour signifier que l'algorithme s'est trompé. Dans le cas de "I'm lovin it", le modèle a bien détecté que c'était un commentaire positif donc c'est un succès.

- POST : {action: "record", prediction: "positive", text: "I'm lovin it"}
- RESPONSE : {status: "success"}

Récupérer les données du moniteur

Nous savons que tous les 1 000 commentaires, une phase de réapprentissage est réalisée. Un nouveau modèle est alors créé et les performances des modèles précédents sont stockées et affichées.

- GET : {params: {target: "monitor_data"}}
- RESPONSE : {logs: [{ timestamp: "2022-03-11 12:28:23", winner: "new", old_model: {loss: 0.32, accuracy: 0.87}, new_model: {loss: 0.29, accuracy: 0.89} }, ...] }

Récupérer le graphique d'entraînement le plus récent

On pourra observer deux images de graphes :

- l'accuracy
- loss

Ces deux graphes donnent les performances du modèle actuel.

- GET : {params: {target: "monitor_graph"}}
- RESPONSE : < PNG IMAGE > blob binaire

L'intelligence artificielle

TF (TensorFlow) modèle

Machine utilisée :

Processeur	Intel i7-7700k
Carte graphique	NVIDIA GTX1060
RAM	32 Go

Le modèle utilisé:

RNN avec LSTM bidirectionnel

