Projet 31

Elèves : Antoine FONDEUR, Camille PRIGENT, Noémie SAUVAGE

Encadrant : Serge NEUMAN



**Collecte des demandes de concitoyens pour alimenter les systèmes de démocratie représentative**

# Contexte du projet

But du projet

Dans le cadre de la démocratie participative, qui vise à associer les citoyens au processus de décision politique, il est important de pouvoir entendre tous les citoyens et conserver leurs pensées, demandes et idées. Cette collecte doit se faire de manière simple pour les concitoyens, donc à travers quelque chose qu'ils ont l'habitude d'utiliser comme un site internet.

L'objectif avec notre projet est d'offrir un espace de recueil des doléances de nos concitoyens couplé à un outil de visualisation des données collectées avec analyse des thèmes prépondérants et des émotions. Ces données devront être présentées de manière élégante.

Enjeux

Dans un contexte de fracture politique et sociale en France, la nécessité de donner un moyen de communications entre élus / candidats et électeurs apparait comme cruciale. Notre projet s’inscrit comme un moyen donné aux politiciens de sonder les électeurs afin d’adapter leur offre aux réels besoins des citoyens en période électorale. Une fois élu, ils pourraient défendre des projets chers aux concitoyens en s’appuyant sur des données chiffrées.

# Architecture du projet

Voir la démonstration dans le diaporama PowerPoint. Ce schéma présente rapidement l’architecture du projet.

|  |  |
| --- | --- |
|  |  |

# Système de sondage

Cahier des charges

Les attendus pour le système de sondage étaient d’utiliser quelque chose de simple et de facilement accessible pour les utilisateurs. Il y avait donc deux options, passer par un système de messagerie tel WhatsApp en utilisant une API ou bien utiliser un site Web.

Nous avons envisagé l’option système de messagerie mais l’API de WhatsApp est désormais payante et nous n’avons pas identifié d’autres applications de messagerie aussi populaire pour être accessible pour le plus grand nombre.

Nous avons donc opté pour le site web qui avait pour autre avantage de pouvoir regrouper l’interface pour remplir le formulaire, la présentation des résultats et des informations pour les utilisateurs.

Le formulaire devra adopter le formalisme d’un User-story en 3 points :

* En tant que :
* Je souhaiterais :
* Parce que :

Le site web

Pour implémenter notre site web nous avons choisi d’utiliser le Framework (sorte de boite à outil prête à l’emploi) web Flask, nous avons fait ce choix car cela nous permet de coder en Python et que ce Framework nous avait été présenté en cours.

Pour gérer l’esthétique du site, nous avons décidé d’utiliser Tailwind, une librairie permettant d’incorporer les paramètres de design directement dans le doc HTML avec des classes déjà codés. Cela nous a permis d’obtenir un design satisfaisant sans avoir à rentrer dans la complexité du CSS.

Une image contenant texte, Appareils électroniques, capture d’écran, logiciel

Description générée automatiquement

Le site à l’esthétique ci-dessus et comporte quatre pages :

* Une page d’accueil avec un texte explicatif court de la démarche pour mettre en confiance l’utilisateur (« Démocratie participative »)
* Une page pour répondre au sondage (« Répondre »)
* Une page pour visualiser les nuages de mots (« Nuages de mots »)
* Une page pour consulter les résultats détaillés comportant différents graphiques (« Résultats détaillés »)

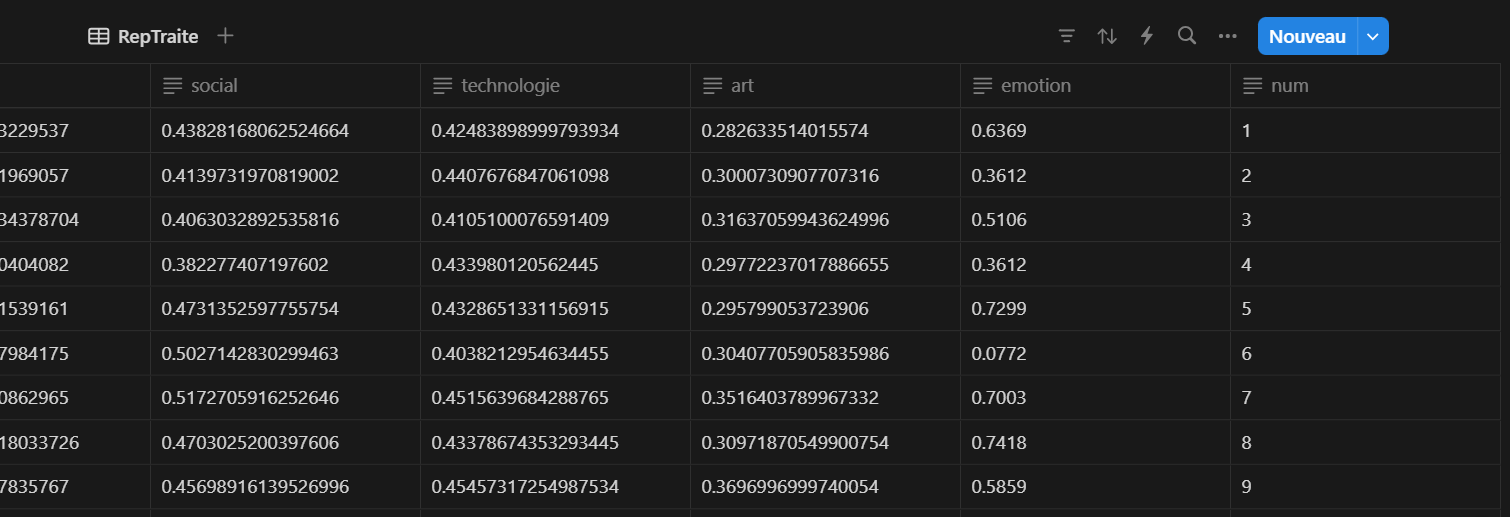
# Notion

Une fois collectées, les données arrivent dans une base de données Notion, une base de données accessible facilement en ligne. Nous avons choisi d’utiliser Notion car nous étions déjà familiarisés avec cet outil et car il est possible d’y ajouter des données très simplement et de facilement débuguer si besoin.

Nous avons deux tableaux :

* Le premier contient les réponses brutes. Il y a trois colonnes (une pour “en tant que”, une autre pour “je souhaite que” et une dernière pour “afin de”) + une colonne d’indexation. Cette dernière permet de s’y retrouver parmi les données brutes et les données traitées.
* Ces données traitées sont justement stockées dans le deuxième tableau. Celui-ci contient autant de colonnes que de catégories avec pour chaque réponse un score associé à chaque catégorie + une colonne pour l’indexation + une colonne pour l’émotion.

Voici à quoi ressemble le deuxième tableau :



Afin de permettre à notre code de communiquer avec Notion, nous avons utilisé la librairie Request sur Python et l’API de notre base Notion.

# Algorithme de traitement du langage

Le traitement d’une réponse se traduit par l’association d’une émotion (allant de –1 à 1) et celle d’un score pour chaque catégorie.

Pour l’analyse de sentiment, on utilise Vader. Nous avons constaté que ce module fonctionnait mieux sur les phrases écrites en anglais. Nous avons donc fait le choix de traduire les réponses en anglais (avec le module googletrans).

Pour la catégorisation des réponses l’algorithme suit les étapes suivantes :

* Lemmatisation et filtrage des réponses : on retire les mots vide de la phrase (déterminants par exemple) et on met les mots sous leur forme canonique (verbes à l’infinitif, noms communs au singulier...).
* On utilise un modèle Word2Vec pour associer à chacun de ses tokens un vecteur dans un espace vectoriel.
* On prend ensuite le barycentre des vecteurs obtenus afin d’avoir le vecteur associé à la réponse.
* On associe à chaque catégorie 1 moins la distance qui sépare l’intitulé de la catégorie avec le vecteur en question pour avoir la similarité.

Une fois la réponse traitée, on l’ajoute au tableau Notion.

# Présentation des résultats

Nuages de mots

Les nuages de mots sont une première présentation des résultats, obtenus avec peu de traitement. Nous proposons 3 nuages de mots : un qui correspond au champ « En tant que », un pour le champ « J’aimerais » et enfin un pour le champ « Parce que ».

Une image contenant texte, Police, capture d’écran, graphisme

Description générée automatiquement

Exemple de nuage de mots

Pour créer le nuage de mot, on traite grossièrement les données en les lemmatisant : on sépare tous les mots et on enlève les mots inutiles comme les prépositions, pronoms on retire également les verbes conjugués.

On utilise ensuite la bibliothèque Wordclouds pour générer le nuage de mots : un mot est d’autant plus grand qu’on le retrouve dans un grand nombre d’occurrence

Ce mode de présentation des données n’utilise pas le programme présenté dans la partie précédente (ou du moins qu’une partie restreinte de fonctions de la bibliothèque nltk ).

Graphiques

Les graphiques obtenus sur la dernière page de notre site web sont codés directement sur le doc app.py qui comprends tout le code du site web. Pour obtenir ces graphiques, le programme effectue un “fetch”, il récupère les données des deux tableaux Notion, les converties en tableau Panda et utilise la librairie Seaborn pour faire des graphiques “Bubble”.

Voici un exemple de graphique obtenu, à améliorer bien sûr :

A chart of different colored circles

Description automatically generated

# Améliorations et impacts

Fiabilité des données

Au cours de notre projet, nous n’avons pas eu le temps de nous plonger réellement dans les problématiques de fiabilité des données mais il pourrait être intéressant de protéger notre questionnaire contre les réponses multiples par des humains ou des robots.

Pour se protéger contre les robots, on pourrait penser à implémenter un système de CAPTCHA à notre formulaire.

Pour garantir qu’une même personne ne répond pas plusieurs fois la même chose, on pourrait :

* Collecter l’heure de réponse et utiliser un algorithme de détection de ressemblance pour tenter d’éliminer les réponses multiples d’une même personne : cela semble assez complexe à mettre en place
* Collecter l’adresse IP des répondants : beaucoup plus simple pour enlever les réponses multiples mais pas parfait car possibilité de contournement avec changement d’adresse IP et collecte de données personnelles

Protection des données

Le RGPD s'applique aux données personnelles, c'est-à-dire toute information se rapportant à une personne physique identifiée ou identifiable. Si les données collectées ne permettent pas d'identifier une personne physique, directement ou indirectement, ces données ne sont pas considérées comme des données personnelles.

La catégorie socio-professionnelle n’est pas une donnée personnelle, notre projet n’est donc dans son état actuel pas soumis aux exigences du RGPD. En revanche, l’adresse IP est une donnée personnelle, sa collecte entrainerait donc des problématiques de protection des données.

Améliorations

Nous pourrions pour améliorer le projet, proposer une interface plus complète d’analyse des résultats avec des filtres d’analyse par catégorie socio-professionnel ou par émotion ou thème prépondérant.

Une étape majeure pour finaliser le projet serait aussi la mise en ligne du site internet avec une gestion efficace de la base de données. Il faudrait pour cela acheter un nom de domaine et pour la gestion de la base de données passer par quelque chose de plus efficace que Notion qui devient vite limité si on augmente la quantité de données. On pourrait aussi imaginer implémenter un système automatique de traitement des données (tous les jours ou à chaque fois que x réponses sont ajoutés).