

Rapport : Analyse des Avis des Employés

Belghalmi Thamer

8 février 2025

1 Description du Problème

Ce projet consiste à développer un programme en Python pour analyser les feedbacks des employés envers leurs collègues. Ces feedbacks sont stockés sous forme de paragraphes dans un fichier Excel. Le fichier contient les colonnes suivantes :

- **ID de l'employé** : L'identifiant de l'employé qui reçoit le feedback.
- **Avis** : Le texte du feedback donné par un collègue.
- **Relation** : La relation entre le donneur d'avis et l'employé évalué. Cette colonne contient uniquement trois catégories : *manager*, *superviseur*, ou *ami*.

Les feedbacks peuvent être rédigés en anglais ou en français. L'objectif est d'analyser ces paragraphes en identifiant les critères positifs et négatifs mentionnés. Le classement des employés sera déterminé par la différence entre le nombre total de critères positifs et négatifs. Cette différence constituera le score de chaque employé.

2 Fonctionnalités du Programme

2.1 Analyse des Feedbacks

- Identifier et compter les critères positifs et négatifs dans chaque paragraphe.
- Calculer le score de chaque employé en soustrayant le nombre total de critères négatifs du nombre total de critères positifs.

2.2 Affichage des Résultats

2.2.1 Cas général (aucun employé sélectionné)

- Afficher un résumé global des critères positifs et négatifs présents dans l'ensemble du fichier Excel.
- Représenter ces données sous forme de deux diagrammes : un diagramme en camembert (pie chart) et un diagramme en barres.
- Afficher le nombre total de feedbacks donnés.

2.2.2 Cas spécifique (un employé sélectionné)

- Afficher les statistiques spécifiques à l'employé sélectionné (critères positifs et négatifs, score).
- Mettre à jour les diagrammes pour refléter les données de l'employé sélectionné.
- Afficher les feedbacks remarquables pour cet employé.

2.3 Sélection des Avis Remarquables

- Les avis remarquables seront sélectionnés selon un ordre de priorité : d'abord les avis des *managers*, puis des *superviseurs*, et enfin des *amis*.
- Si plusieurs avis ont la même priorité (par exemple, deux avis de managers), celui qui contient le plus de critères (positifs ou négatifs) sera priorisé.

2.4 Affichage Supplémentaire

- Dans tous les cas, l'ID de l'employé ayant le meilleur score sera affiché en haut de la page et ne sera pas modifié.

3 Exemple de Sortie

3.1 Résumé Global

- Diagramme en camembert : Répartition des critères positifs et négatifs.
- Diagramme en barres : Comparaison des critères positifs et négatifs.
- Nombre total de feedbacks : X.

3.2 Résumé par Employé

- Diagrammes mis à jour pour l'employé sélectionné.
- Avis remarquables : Liste des feedbacks prioritaires (manager > superviseur > ami).
- Score de l'employé : X (positifs - négatifs).

3.3 Meilleur Employé

- ID du meilleur employé : [ID].

4 Clarifications Supplémentaires

- **Traitement des Langues** : Le programme doit être capable de traiter les feedbacks en anglais et en français sans distinction.
- **Priorisation des Avis** : En cas d'égalité dans le nombre de critères, l'avis le plus long ou le plus détaillé sera priorisé.
- **Interface Utilisateur** : L'affichage des résultats doit être clair et intuitif, avec une séparation visuelle entre les différentes sections (résumé global, résumé par employé, meilleur employé).

5 Étape 1 : Compréhension du Problème et Planification

1. **Définir les objectifs** :
 - Analyser les feedbacks pour identifier les critères positifs et négatifs.
 - Calculer un score pour chaque employé basé sur ces critères.
 - Afficher les résultats sous forme de diagrammes et de statistiques.
2. **Identifier les entrées et sorties** :
 - **Entrées** : Fichier Excel contenant les colonnes ID, Avis, et Relation.
 - **Sorties** :
 - Diagrammes (camembert et barres) pour les critères positifs et négatifs.
 - Score des employés.
 - Avis remarquables triés par priorité.
 - ID du meilleur employé.
3. **Choisir les outils et technologies** :
 - **Langage** : Python.
 - **Bibliothèques** :
 - `pandas` pour la manipulation des données Excel.
 - `matplotlib` ou `seaborn` pour les visualisations.
 - `nltk` ou `spaCy` pour le traitement du langage naturel (NLP).
 - `openpyxl` ou `xlrd` pour lire les fichiers Excel.

6 Étape 2 : Préparation des Données

1. **Importer les données** :
 - Lire le fichier Excel avec `pandas` pour obtenir un `DataFrame`.
 - Vérifier la structure des données (colonnes, types de données, valeurs manquantes).
2. **Nettoyer les données** :
 - Supprimer les lignes avec des valeurs manquantes ou invalides.
 - Standardiser les textes (minuscules, suppression de la ponctuation, etc.).
 - Traduire les feedbacks en anglais (si nécessaire) pour une analyse uniforme.

7 Étape 3 : Analyse des Feedbacks

1. **Identifier les critères positifs et négatifs :**
 - Créer des listes de mots-clés associés aux critères positifs et négatifs.
 - Utiliser des techniques NLP pour compter les occurrences de ces mots-clés.
2. **Calculer le score des employés :**
 - $\text{Score} = \text{total des critères positifs} - \text{total des critères négatifs}$.

8 Étape 4 : Sélection des Avis Remarquables

- Trier les avis par priorité (manager > superviseur > ami).
- Sélectionner l'avis contenant le plus de critères en cas d'égalité.

9 Étape 5 : Visualisation des Résultats

- Générer des diagrammes (camembert, barres) pour représenter les critères.
- Mettre à jour l'affichage selon l'employé sélectionné.
- Afficher l'ID du meilleur employé.

10 Étape 6 : Interface Utilisateur

- Développer une interface avec `tkinter` ou `streamlit`.
- Tester l'interface pour garantir une bonne expérience utilisateur.

11 Étape 7 : Tests et Validation

- Vérifier l'exactitude de l'analyse des critères.
- Tester le calcul des scores et la priorisation des avis.

12 Étape 8 : Documentation et Déploiement

- Documenter le code et ajouter un fichier README.
- Préparer le programme pour un déploiement facile.