Université Paris 8 - Vincennes à Saint-Denis

Le Projet Gutenberg

Anis Ait Yacoub

Date de rendu : le 31/05/2023

Table des matières

Chapitre 1 **Introduction** *ii*

1.1 Le Projet Gutenberg ii

1.2 Les Modules ii

1.2.1 Le Scraper de Livres ii

1.2.2 L'Analyseur de Données iii

1.2.3 Le Module de Clustering de Données iii

1.3 L'Exécution iii

Chapitre 2 **Fonctionnalités** *iv*

2.1 Technologies Utilisées v

Chapitre 3 **Analyseur de Données** *vi*

3.0.1 Chargement des Données vi

3.0.2 Analyses vi

Chapitre 4 **Clustering de Données** *viii*

4.0.1 Lecture et Prétraitement des Données viii

4.0.2 Clustering K-Means viii

4.0.3 Clustering Mean Shift viii

4.0.4 Évaluation des Clusters ix

4.0.5 Visualisation des Clusters ix

Chapitre 5 **Résultats** *x*

5.0.1 Téléchargement des Livres x

5.0.2 Clustering de Données x

Chapitre 6 **Conclusion** *xii*

Chapitre 1

Introduction

Ce rapport décrit en détail un système de traitement de données destiné à interagir avec le site web du Projet Gutenberg, une ressource en ligne considérable d'œuvres littéraires disponibles au public. Le programme, écrit en Python, est constitué de trois composants principaux : un scraper de livres, un analyseur de données et un module de clustering de données. Chacun de ces modules joue un rôle essentiel dans la fonctionnalité globale du système et a été intégré dans un script principal qui coordonne l'exécution séquentielle des modules.

1.1 Le Projet Gutenberg

L'initiative de ce projet est née d'une proposition de projet tuteuré visant à simplifier l'accès aux informations sur les livres les plus populaires disponibles sur le Projet Gutenberg. Le programme est spécifiquement conçu pour cibler les 100 livres les plus téléchargés au cours des 30 derniers jours. Cette focalisation sur les livres populaires vise à identifier les tendances de lecture, faciliter les recommandations de livres et fournir une analyse détaillée des caractéristiques communes des livres fréquemment téléchargés.

1.2 Les Modules

1.2.1 Le Scraper de Livres

Le premier module est un scraper de livres qui navigue sur le site web du Projet Gutenberg pour extraire des informations précieuses sur les livres. Ces informations incluent le titre du livre, l'auteur, l'URL de téléchargement du livre, le sujet ou le genre du livre, la classe LoC (Library of Congress Classification), et le nombre de téléchargements pour chaque livre. Ces données constituent une base solide pour une analyse ultérieure visant à obtenir des insights pertinents.

1.2.2 L'Analyseur de Données

Le deuxième module est un analyseur de données. Ce module utilise les données extraites par le scraper de livres pour réaliser une série d'analyses destinées à déduire des tendances et des modèles. Ces analyses peuvent comprendre l'identification des genres les plus populaires, des auteurs les plus lus, et toute corrélation entre différents aspects des livres.

1.2.3 Le Module de Clustering de Données

Le dernier module du système est le module de clustering de données. Ce module utilise des techniques d'apprentissage automatique pour regrouper les livres en fonction de leurs caractéristiques communes. Les groupes, ou clusters, peuvent être basés sur n'importe quelle combinaison de caractéristiques, telles que le genre ou la classe LoC. Ce regroupement peut aider à identifier les sous-groupes parmi les livres les plus populaires et à fournir des recommandations de livres en fonction de ces sous-groupes.

1.3 L'Exécution

Tous ces modules sont orchestrés par un script principal qui les exécute en séquence. Le processus commence par le scraping des données sur le site web du Projet Gutenberg. Les données collectées sont ensuite analysées pour extraire des insights. Enfin, le module de clustering de données est utilisé pour regrouper les livres en fonction de leurs attributs communs. Ce rapport détaille le fonctionnement de chaque module, ainsi que les résultats obtenus par leur exécution. Il présentera également les avantages potentiels de l'utilisation de ce système et les perspectives.

Chapitre 2

Fonctionnalités

Le scraper de livres comprend les fonctionnalités suivantes :

— **Extraction des titres et auteurs des livres :** La fonction `get_book_titles_and_authors(url)` est utilisée pour récupérer les titres, les auteurs et les URL de chaque livre à partir de la page des 100 livres les plus téléchargés du Project Gutenberg. Elle envoie une requête HTTP à la page en utilisant le module `requests` et utilise `BeautifulSoup` pour analyser la réponse et extraire les informations nécessaires. Les informations de chaque livre sont stockées dans un dictionnaire, qui est ensuite ajouté à une liste.

— **Récupération d'informations supplémentaires sur chaque livre :** La fonction `get_book_info(book)` est utilisée pour récupérer des informations supplémentaires sur chaque livre, telles que le sujet, la classe LoC (Library of Congress) et le nombre de téléchargements. Elle utilise également les modules `requests` et `BeautifulSoup` pour obtenir et analyser les informations de chaque page de livre.

— **Téléchargement du contenu de chaque livre et conversion en format PDF :** La fonction `download_book(book)` est utilisée pour télécharger le contenu de chaque livre au format texte, puis le convertir en PDF en utilisant la fonction `convert_to_pdf(text_file, pdf_file)`. Le module `requests` est utilisé pour télécharger le contenu du livre, qui est ensuite sauvegardé en tant que fichier texte. Le contenu du fichier texte est ensuite converti en PDF en utilisant le module `reportlab`.

— **Sauvegarde des informations extraites dans un fichier CSV pour une analyse ultérieure :** La fonction `save_book_info_to_csv(book_list)` est utilisée pour sauvegarder les informations de chaque livre dans un fichier CSV. Le module `csv` de Python est utilisé pour écrire les informations dans le fichier CSV.

De plus, le script utilise la fonction `main()` pour exécuter l'ensemble du processus. La fonction `scrape_and_download_books()` effectue toutes les opérations décrites ci-dessus en séquence. Le module `concurrent.futures` est utilisé pour paralléliser les téléchargements de livres et les requêtes d'informations sur les livres afin d'améliorer les performances. Un système de journalisation (`logging`) est également mis en place pour suivre les erreurs et les informations importantes.

2.1 Technologies Utilisées

Le scraper de livres utilise les technologies suivantes :

— **BeautifulSoup :** Utilisé pour analyser les pages HTML et extraire les informations nécessaires.

— **requests :** Utilisé pour envoyer des requêtes HTTP aux pages web et récupérer les réponses.

— **reportlab :** Utilisé pour convertir le contenu des livres en format PDF.

— **concurrent.futures :** Utilisé pour gérer les téléchargements simultanés de livres et les requêtes d'informations sur les livres afin d'améliorer les performances.

Ces technologies offrent des fonctionnalités puissantes pour le scraping de livres à partir du site Project Gutenberg et la manipulation des données extraites.

Dans la section suivante, nous détaillerons chaque fonctionnalité du scraper de livres et présenterons des exemples concrets de son utilisation.

Chapitre 3

Analyseur de Données

L'analyseur de données est un module conçu pour effectuer des analyses statistiques et graphiques sur les données collectées à partir du scraper de livres. Cet outil utilise plusieurs bibliothèques Python, dont pandas pour la manipulation des données, matplotlib et seaborn pour la visualisation des données. Le code pour ce module est structuré de manière à charger les données à partir d'un fichier CSV, puis à effectuer trois analyses distinctes. Voici une description détaillée de ces analyses :

3.0.1 Chargement des Données

Le module commence par charger les données à partir d'un fichier CSV en utilisant la fonction `load_data_from_csv(file_path)`. Cette fonction utilise la bibliothèque pandas pour lire le fichier CSV et convertit ensuite le champ `'Downloads'` en un nombre entier pour faciliter les analyses ultérieures.

3.0.2 Analyses

Une fois les données chargées, trois types d'analyses sont effectués :

1. **Distribution des Téléchargements :** Une analyse de la distribution des téléchargements est effectuée. Pour visualiser cette distribution, un histogramme est créé en utilisant la bibliothèque seaborn. L'histogramme affiche le nombre de téléchargements sur l'axe des x et la fréquence sur l'axe des y. Le graphique est sauvegardé sous le nom `'distribution_downloads.png'`.
2. **Nombre de Livres par Auteur :** L'analyseur calcule ensuite le nombre de livres écrits par chaque auteur. Pour ce faire, il compte le nombre de fois que chaque auteur apparaît dans les données et affiche les dix auteurs avec le plus grand nombre de livres dans un graphique à barres. Le graphique est sauvegardé sous le nom `'books_per_author.png'`.
3. **Nombre de Livres par Sujet :** Enfin, le nombre de livres par sujet est analysé. Pour ce faire, le module compte le nombre de fois que chaque sujet apparaît dans les données et affiche les dix sujets les plus courants dans un graphique à barres. Le graphique est sauvegardé sous le nom `'books_per_subject.png'`.

Toutes les analyses sont intégrées dans une seule fonction, `data_analysis_main(data)`, qui est appelée dans la fonction `main()`. Cette structure facilite l'exécution de l'analyse en une seule étape.

En résumé, l'analyseur de données est un outil essentiel pour comprendre les tendances et les caractéristiques des livres téléchargés à partir du Projet Gutenberg.

Chapitre 4

Clustering de Données

Le module de clustering de données est responsable de l'agrégation des livres selon certaines caractéristiques, telles que le sujet ou la classe LoC. Il utilise des techniques de clustering non supervisé, comme K-means et MeanShift. Le code pour ce module est présenté et détaillé ci-dessous :

4.0.1 Lecture et Prétraitement des Données

La fonction `read_data` est utilisée pour lire les données du fichier CSV et les prétraiter pour le clustering.

— **One-hot encoding :** Pour que les données catégorielles soient correctement interprétées par les algorithmes de clustering, elles sont encodées en utilisant une méthode appelée "One-hot encoding". Cela crée une nouvelle colonne pour chaque catégorie unique dans les colonnes 'Auteur', 'Sujet' et 'LoC Class'.

— **Nettoyage des données :** Les colonnes qui ne sont plus nécessaires après le codage sont supprimées du dataframe. Ce sont 'Title', 'URL', 'Author', 'Subject' et 'LoC Class'.

4.0.2 Clustering K-Means

La méthode K-Means est une technique populaire de clustering non supervisé. Elle vise à diviser les données en un nombre prédéfini de clusters (dans ce cas, 3), chaque cluster étant identifié par son centre (ou "centroïde"). La fonction `kmeans_clustering` effectue le clustering K-Means sur les données et renvoie le modèle formé.

4.0.3 Clustering Mean Shift

La méthode Mean Shift est une autre technique de clustering non supervisé qui ne nécessite pas que le nombre de clusters soit défini à l'avance. Elle repose sur l'estimation de la densité de probabilité des données. La fonction `mean_shift_clustering` effectue le clustering Mean Shift sur les données. L'argument `bandwidth` dans l'implémentation de sklearn représente le rayon de la fenêtre du noyau et est estimé à l'aide de la méthode `estimate_bandwidth`.

4.0.4 Évaluation des Clusters

Pour évaluer la qualité des clusters formés par les deux méthodes, la mesure de silhouette est utilisée. Elle mesure la proximité de chaque point de données dans un cluster par rapport aux points de données dans les clusters voisins. Les scores de silhouette varient de -1 à 1, où un score plus élevé indique que les points de données sont bien regroupés.

4.0.5 Visualisation des Clusters

Enfin, la fonction `plot_clusters` est utilisée pour visualiser les clusters formés par les deux méthodes. Les graphiques de clusters sont sauvegardés sous forme d'images dans le répertoire `graphs`.

En résumé, ce module lit et prépare les données, effectue le clustering à l'aide de deux méthodes différentes, évalue la qualité des clusters et visualise les résultats. Le code est bien organisé et modulaire, ce qui le rend facile à comprendre et à modifier si nécessaire.

Chapitre 5

Résultats

Le script a été exécuté avec succès, accomplissant l'ensemble des tâches qu'il avait été programmé à réaliser. Les résultats sont détaillés ci-dessous :

5.0.1 Téléchargement des Livres

Le script a réussi à télécharger les 100 livres les plus téléchargés du dernier mois à partir du site du Projet Gutenberg. Ceci est confirmé par la barre de progression affichée, qui indique que 100% (100 sur 100) des livres ont été téléchargés. L'ensemble du processus de téléchargement a pris 31 secondes.

5.0.2 Clustering de Données

Une fois que les livres ont été téléchargés et que leurs informations ont été sauvegardées, le script a entamé le processus de clustering de données. Ce processus comprend plusieurs étapes :

1. **Chargement des Données :** Les données sur les livres, qui avaient été sauvegardées dans un fichier CSV, ont été chargées avec succès dans le script.
2. **Encodage des Caractéristiques Catégorielles :** Les caractéristiques catégorielles, telles que le sujet et la classe LoC, ont été encodées avec succès. Cela a transformé ces caractéristiques en un format que les algorithmes de clustering peuvent comprendre et utiliser.
3. **Clustering KMeans :** L'algorithme KMeans a été appliqué aux données, créant ainsi des clusters de livres basés sur leurs caractéristiques. Le score silhouette pour KMeans était de 0.7243, ce qui indique une bonne séparation des clusters. En d'autres termes, les livres au sein de chaque cluster étaient assez similaires les uns aux autres, mais différents des livres dans les autres clusters.
4. **Clustering MeanShift :** L'algorithme MeanShift a également été appliqué aux données. Le score silhouette pour MeanShift était de 0.7531, ce qui indique également une bonne séparation des clusters.
5. **Tracé des Clusters :** Enfin, les clusters créés par les deux algorithmes ont été tracés et les graphiques correspondants ont été sauvegardés avec succès.

Les résultats affichés dans le terminal lors de l'exécution du script sont les suivants :

```
Starting data clustering...
Loading data...
Data loaded successfully.
Encoding categorical features...
Categorical features encoded successfully.
Performing KMeans clustering...
Silhouette score for KMeans: 0.7242743155783912
Performing MeanShift clustering...
Silhouette score for MeanShift: 0.7530932001697799
Plotting clusters...
Clusters plotted and saved successfully.
Data clustering completed successfully.
```

En conclusion, le script a été capable de télécharger les livres, de regrouper les données, et de générer des clusters avec une bonne séparation. Le score silhouette pour les deux algorithmes de clustering indique que les clusters ont été bien formés, avec les livres similaires regroupés ensemble. Ces résultats démontrent la réussite de l'implémentation du script.

Chapitre 6

Conclusion

Ce projet représente une application remarquable de plusieurs domaines importants de l'informatique, à savoir le web scraping, l'analyse de données et le clustering. À travers l'exploration et l'implémentation de ces techniques, nous avons pu construire un programme efficace pour extraire, analyser et regrouper les informations relatives aux livres les plus populaires provenant d'une source en ligne spécifique, le site web du Projet Gutenberg.

La première étape de ce processus consistait à utiliser le web scraping pour extraire les informations essentielles des livres. En naviguant sur le site web et en extrayant des informations clés sur chaque livre, nous avons créé une base de données significative pour l'analyse ultérieure.

Par la suite, ces données ont été soumises à un processus d'analyse approfondie. Grâce à l'analyse de données, nous avons pu obtenir des aperçus intéressants sur les tendances actuelles en matière de lecture, par exemple, les sujets les plus populaires ou les auteurs les plus lus.

Enfin, nous avons utilisé des techniques de clustering pour regrouper les livres en fonction de leurs caractéristiques. Les algorithmes de clustering nous ont permis de regrouper les livres en fonction de différentes variables, offrant ainsi une autre dimension à notre analyse.

Cependant, bien que nous ayons réalisé un progrès significatif, il y a encore place pour l'expansion et l'amélioration de ce projet. Par exemple, nous pourrions envisager d'intégrer une interface utilisateur pour rendre le programme plus accessible et plus facile à utiliser. De plus, nous pourrions également envisager d'ajouter d'autres sources de livres afin d'enrichir notre base de données et de rendre notre analyse plus complète.

En somme, ce projet a démontré l'efficacité de l'application des techniques de scraping, d'analyse de données et de clustering dans un contexte réel. Il a montré comment ces techniques peuvent être combinées pour obtenir des résultats précis et informatifs. Il a également mis en évidence les possibilités d'amélioration et d'expansion, offrant un potentiel d'exploration future.