Rapport de Netflix Project

Réalisé par : - Aymen Benbani
- Maryam Moustaghfir

Table des matières

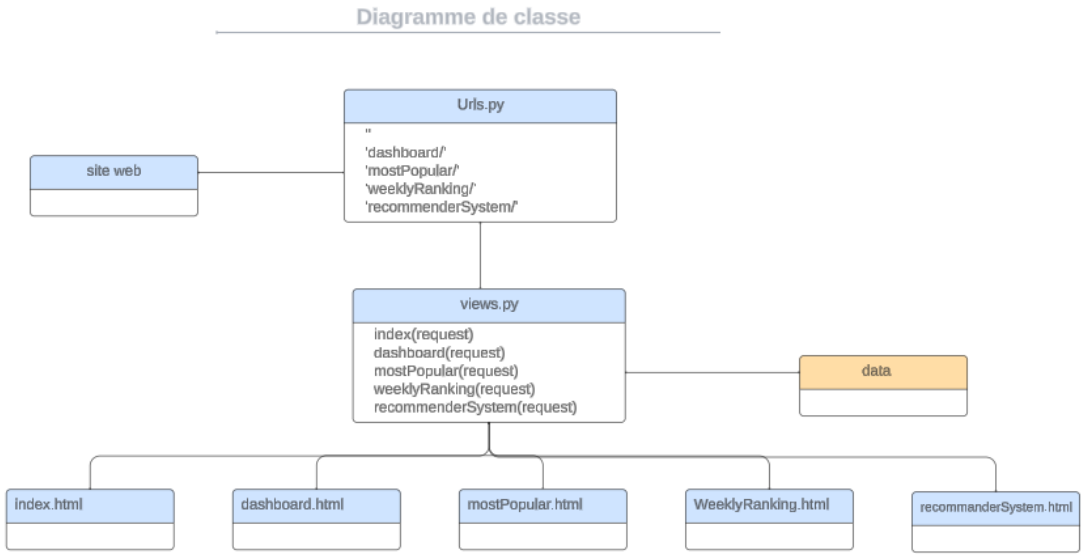
1. Introduction :

Notre projet est **une application Web** axée sur la visualisation de données liées aux films et émissions de télévision de Netflix. L'objectif principal est de créer une interface conviviale permettant aux utilisateurs d'explorer des informations telles que les classements par pays, les films les plus populaires, et de recevoir des recommandations personnalisées.

La Business Intelligence (BI) est au cœur du projet, avec des fonctionnalités telles que la création de graphiques interactifs dans un **Dasboard**, la génération de recommandations personnalisées, et l'affichage des classements hebdomadaires. **La Data Science** intervient dans la préparation des données, la génération de statistiques, et la mise en œuvre d'un système de recommandation avancé.

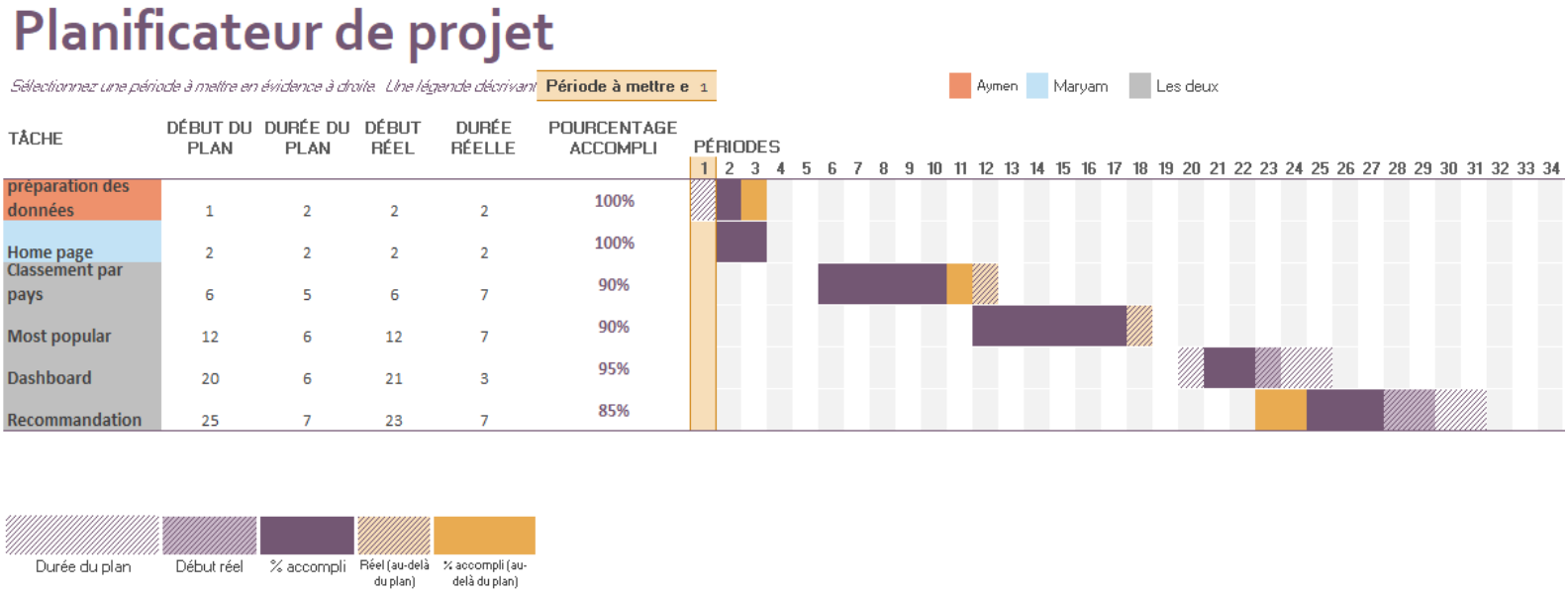
En termes **d'architecture logicielle**, le projet repose sur le framework web Django pour organiser le code de manière claire et maintenable. Pour l'interface utilisateur, nous utilisons Bootstrap pour assurer une mise en page réactive et attrayante. Les graphiques interactifs sont générés à l'aide de Plotly Express, offrant une expérience visuelle immersive.

2. Diagramme de classe :



3. Diagramme de Gantt :

Nous avons réalisé un diagramme de Gantt réel pour examiner comment les tâches ont été réparties entre les membres du projet et comparer avec la planification initiale. Ce fichier est fourni avec les autres documents pour plus de visibilité.



4. Analyse des écarts entre le prévisionnel et le réel :

1. Préparation des données :

- **Planifié** (Début: Jour 1, Durée: 2h) : La tâche de préparation des données a été exécutée conformément au plan initial, démarrant le jour 1 et se terminant comme prévu en 2 heures.

- **Réel** (Début: Jour 2, Durée: 2h) : Bien que la tâche ait commencé un jour plus tard que prévu, elle a été achevée dans le délai initial de 2 jours, atteignant ainsi un taux d'achèvement de 100%. Les décalages dans le calendrier n'ont pas eu d'impact significatif sur la réalisation de cette tâche.

2. Home Page :

- **Planifié** (Début: Jour 2, Durée: 2h) : La création de la page d'accueil a suivi le calendrier initial, commençant le jour 2 et se terminant dans les 2 jours prévus.

- **Réel** (Début: Jour 2, Durée: 2h) : La tâche a été exécutée conformément au plan, atteignant un taux d'achèvement de 100%. Aucun retard n'a été enregistré.

3. Classement par pays :

- **Planifié** (Début: Jour 6, Durée: 5h) : Bien que la tâche ait commencé comme prévu le jour 6, elle a pris 2 heures de plus que prévu pour être complétée.

- **Réel** (Début: Jour 6, Durée: 7h) : La tâche a demandé un temps supplémentaire pour son achèvement, atteignant un taux d'achèvement de 90%. Ce léger retard peut être attribué à la lenteur du processus de mise à jour de la page. Initialement, l'intention était d'afficher le classement par pays sur une base hebdomadaire depuis 2021, étant donné le volume important de données à traiter. Cependant, une décision ultérieure a été prise de filtrer les données pour présenter uniquement le classement pour une semaine, ce qui a influencé le déroulement temporel de la tâche.

4. Most Popular :

- **Planifié** (Début: Jour 12, Durée: 6h) : La tâche a commencé comme prévu le jour 12, mais elle a dépassé le délai initial de 1 heure.

- **Réel** (Début: Jour 12, Durée: 7h) : Bien que la tâche ait accusé un léger retard, elle a néanmoins abouti à un taux d'achèvement de 90%. Cette légère prolongation s'explique par la complexité de la recherche de l'API qui va permettre d'afficher le poster des images selon leur id dans la base de données et selon le choix entre TV ou films.

5. Dashboard :

- **Planifié** (Début: Jour 20, Durée: 6h) : Bien que la tâche ait commencé avec un jour de retard (jour 21), elle a été complétée dans le délai prévu.

- **Réel** (Début: Jour 21, Durée: 3h) : La tâche a été menée à bien avec succès, mais elle a été exécutée plus rapidement que prévu, atteignant un taux d'achèvement de 95%. Cette célérité inattendue s'explique par la préparation, le filtrage et le nettoyage préalables des données, facilitant ainsi l'utilisation des graphes interactifs avec **Plotly**.

6. Recommandation :

- **Planifié** (Début: Jour 25, Durée: 7h) : La tâche a débuté deux jours avant la date prévue et a nécessité 7 jours pour être complétée.

- **Réel** (Début: Jour 23, Durée: 7h) : Malgré le démarrage anticipé de la tâche, elle a nécessité la durée initialement estimée, atteignant un taux d'achèvement de 85%. La complexité de la recherche du modèle de recommandation a contribué à cette durée. Le système de recommandation utilisé par Netflix est très avancé, s'appuyant sur l'historique de l'utilisateur et ses préférences. Cependant, la difficulté a résidé dans l'absence de bases de données adaptées pour cette spécificité, entraînant des défis supplémentaires dans la mise en œuvre du modèle de recommandation.

Bilan :

La gestion du projet a présenté un certain nombre de succès et de défis, illustrant la dynamique inhérente à tout développement logiciel. Certains aspects du projet ont été exécutés avec aisance, tandis que d'autres ont nécessité des ajustements et des efforts supplémentaires :

Facilités rencontrées :

Préparation des données et Home Page : Ces phases ont été menées à bien sans heurts majeurs, respectant les délais et atteignant un taux d'achèvement de 100%. La planification initiale s'est avérée efficace pour ces tâches.

Most Popular et Dashboard : Malgré des légères variations par rapport au plan initial, ces tâches ont été complétées avec succès. La préparation proactive des données pour le tableau de bord a contribué à une exécution plus rapide que prévu.

Difficultés rencontrées :

Classement par pays : La lenteur du processus de mise à jour de la page a posé des défis, entraînant une extension de la durée. Les ajustements dans la portée du projet ont été nécessaires pour garantir une expérience utilisateur optimale.

Recommandation : La recherche du modèle de recommandation a présenté des difficultés en raison de la complexité du système utilisé par Netflix, basé sur l'historique de l'utilisateur. Les contraintes liées à la disponibilité de bases de données appropriées ont ajouté des défis supplémentaires.

5. Outils utilisés :

Chaque membre du binôme a choisi un outil du génie logiciel pour approfondir ses compétences. Nous avons utilisé Git comme gestionnaire de version, Django pour le backend, Bootstrap pour l'interface utilisateur. En plus **Aymen** a utilisé pandas pour la préparation des données, et **Maryam** a utilisé plotly pour la création des graphes interactifs.

Django (Python Web Framework) :

- ★ Gestion des vues : Django facilite la gestion des vues en suivant le modèle MVC (Modèle-Vue-Contrôleur) pour organiser le code de manière claire et maintenable.
- ★ Routage des URL : Le framework Django offre un système de routage simple et puissant, permettant de mapper les URL aux vues de manière élégante.
- ★ Intégration des données dans les pages HTML : Django facilite l'intégration des données dynamiques dans les pages HTML en utilisant son moteur de templates.

Bootstrap (CSS Framework) :

- ★ Mise en page des pages HTML : Bootstrap simplifie le processus de conception de l'interface utilisateur en fournissant des composants préconçus et un système de grille flexible pour assurer une mise en page réactive et attrayante.

Plotly.js et plotly.express:

- ★ Création de graphiques interactifs pour le Dashboard : Plotly.js et Plotly.express sont des bibliothèques de visualisation de données en JavaScript, permettant de générer des graphiques interactifs, des diagrammes à barres, des nuages de points, etc.
- ★ Inclusion à partir du CDN dans les balises HTML : L'utilisation d'un CDN (Content Delivery Network) pour Plotly.js garantit une performance optimale en permettant le chargement des bibliothèques directement depuis un serveur externe.

CSS :

- ★ Stylisation des graphiques : Les feuilles de style en cascade (CSS) ont été utilisées pour personnaliser l'apparence des graphiques générés par Plotly.js, assurant une présentation visuelle cohérente avec le reste de l'interface utilisateur.

HTML/JS (Script tags) :

- ★ Inclusion de balises <script> pour charger les bibliothèques Plotly.js : L'inclusion de balises <script> dans le code HTML permet le chargement asynchrone des bibliothèques Plotly.js, garantissant une intégration fluide des graphiques interactifs.

Pandas :

- ★ Lecture de données depuis un fichier Excel ou CSV dans un DataFrame : Pandas a été utilisé pour lire et manipuler les données provenant de fichiers Excel ou CSV, les convertissant en structures de données adaptées à l'analyse.
- ★ Préparation, nettoyage des données inutiles et dupliquées, traitement des valeurs manquantes.

WordCloud :

- ★ Génération de nuages de mots à partir du texte : L'outil WordCloud a été employé pour créer des visualisations graphiques représentant la fréquence des mots les plus utilisés dans les titres de Netflix à partir de données textuelles.

Requests :

- ★ Description : La bibliothèque requests est utilisée pour effectuer des requêtes HTTP afin d'interagir avec l'API The Movie Database (TMDb). Elle simplifie la gestion des requêtes et des réponses HTTP.

The Movie Database (TMDb) API :

- ★ Description : L'API TMDb est une ressource externe utilisée pour obtenir des informations détaillées sur les films, y compris les chemins des affiches. L'API fournit des données structurées au format JSON.

Github :

- ★ Suivi des changements dans le code source : Git a servi de gestionnaire de version pour suivre les modifications du code source, faciliter la collaboration et permettre le retour en arrière en cas de besoin.

6. Documentation :

- ★ **templates :**

  - base.html: Il sert de modèle de base pour les pages de votre site. Il définit la structure commune, telle que l'en-tête, le pied de page, et d'autres éléments qu'on souhaite partager entre les différentes pages.

  - Header.html: Il se concentre sur la création de la partie en-tête commune à plusieurs pages du site. Il peut inclure des éléments tels que le logo du site, la barre de navigation, et d'autres informations importantes pour la navigation.

  - index.html: Il s'agit de la page d'accueil de l'application. Elle contient des informations de bienvenue et le menu.

  - dashboard.html: Cette page est dédiée à afficher les graphiques interactifs et visualisations générés par les fonctions dans views.py. On trouve dedans des conteneurs HTML pour intégrer les graphiques créés avec Plotly Express, des sections pour afficher les statistiques totales, etc.

  - recommenderSystem.html: Cette page comporte un formulaire pour que l'utilisateur puisse choisir un film, et elle affichera ensuite les recommandations générées par la fonction recommend.

  - weeklyRankings.html: Il s'agit de la page permettant à l'utilisateur de sélectionner une semaine, un pays et une catégorie pour afficher les classements hebdomadaires correspondants pour le mois 12. Elle contient des éléments interactifs pour sélectionner ces filtres.

  - mostPopular.html: Cette page comporte une liste des films les plus populaires, avec la possibilité pour l'utilisateur de filtrer par langue d'origine. Elle utilisera également les fonctions de views.py pour afficher les posters des films.

★ views.py :

- index(request)

Cette fonction rend simplement la page d'accueil (index.html).

- create_bar_chart(data, x_col, y_col, y_label, title) :

Cette fonction génère un diagramme à bandes à partir des données fournies. Les paramètres spécifient les colonnes x et y, l'étiquette y, et le titre du graphique.

- stacked_line_chart(data) :

Cette fonction génère un graphique en courbes empilées montrant le nombre de titres par année de sortie, séparés par type (Movie/TV Show).

- plot_movies(data) :

Cette fonction crée un diagramme à barres représentant les dix premiers pays avec le plus grand nombre de films.

- plot_tv_shows(data) :

Cette fonction crée un diagramme à barres représentant les dix premiers pays avec le plus grand nombre d'émissions de télévision.

- title_pie(data) :

Cette fonction génère un diagramme circulaire représentant le pourcentage de films et d'émissions de télévision.

- wordcloud(data) :

Cette fonction génère un nuage de mots basé sur les titres des films et émissions de télévision.

- rating_pie(data) :

Cette fonction génère un diagramme circulaire représentant la distribution des classifications d'âge.

- dashboard(request) :

Cette fonction rend la page du tableau de bord (dashboard.html) en utilisant les fonctions ci-dessus pour générer des graphiques et des visualisations.

- fetch_poster(movie_id) :

Cette fonction utilise l'API The Movie Database (TMDb) pour obtenir le chemin de l'affiche d'un film à partir de son ID.

- recommend(movie) :

Cette fonction recommande les cinq films les plus similaires à un film donné en utilisant une matrice de similarité précalculée.

- recommenderSystem(request) :

Cette fonction gère la logique de la page de recommandation (recommenderSystem.html) en utilisant la fonction recommend.

- weeklyRankings(request) :

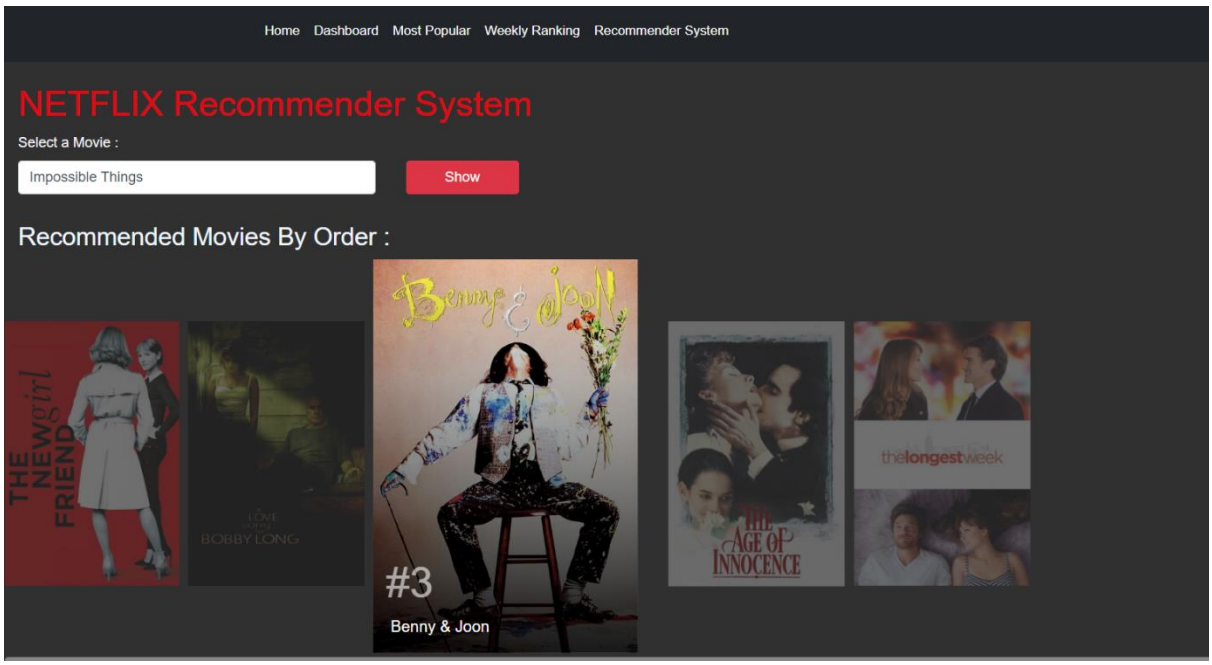
Cette fonction rend la page des classements hebdomadaires (weeklyRankings.html), permettant la sélection de la semaine, du pays et de la catégorie.

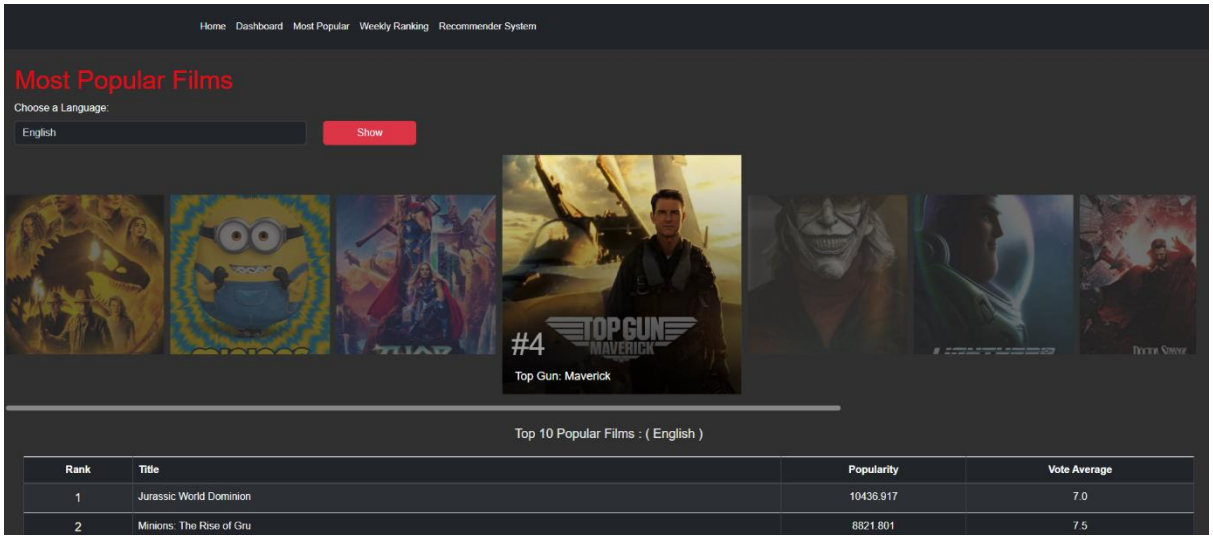
- mostPopular(request) :

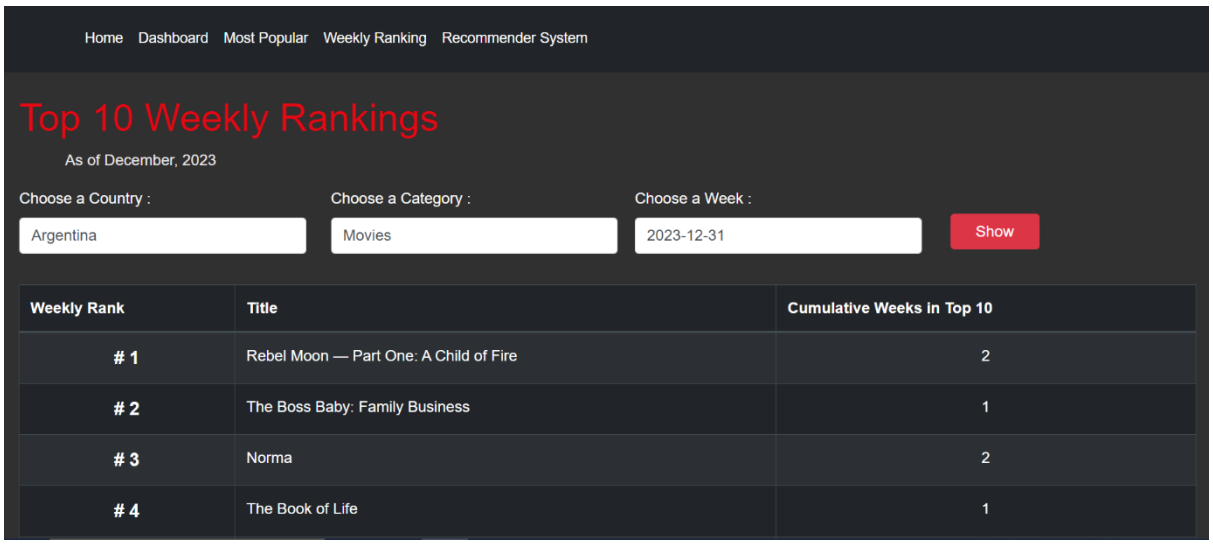
Cette fonction rend la page des films les plus populaires (mostPopular.html), permettant la sélection de la langue d'origine. Elle utilise également la fonction fetch_poster pour obtenir les affiches des films.ws.py.

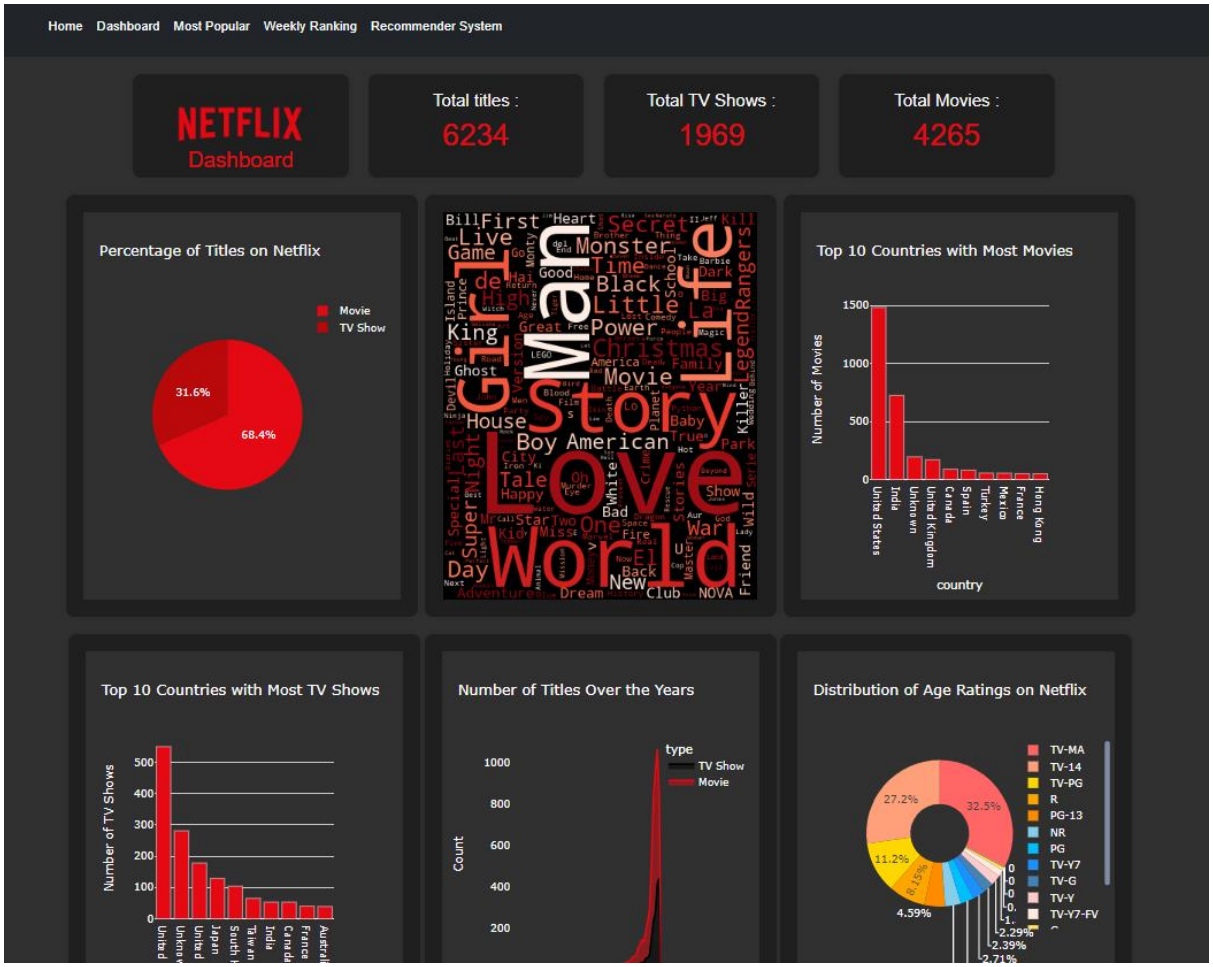
★ Affichage des templates :

- Ecran large

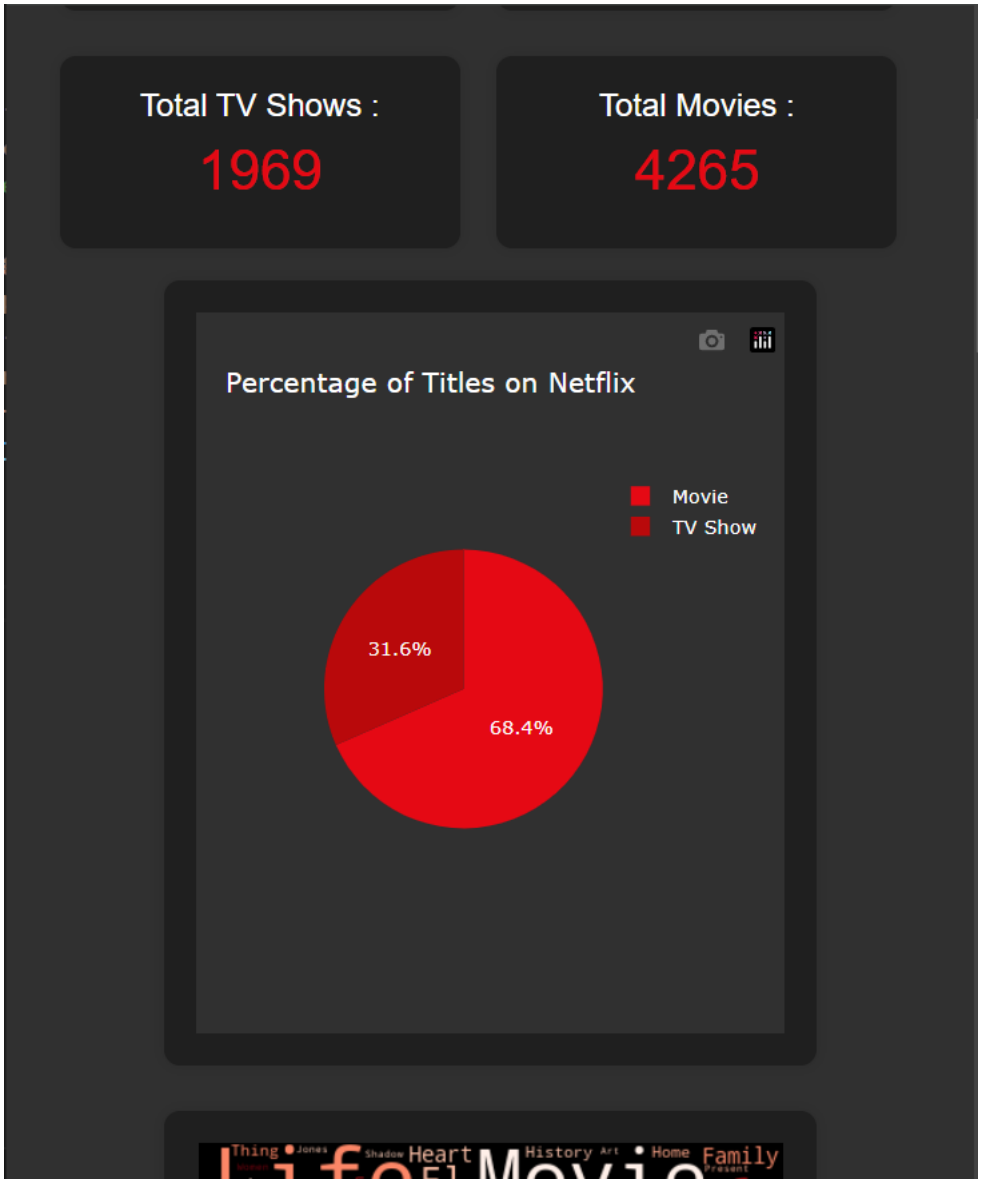


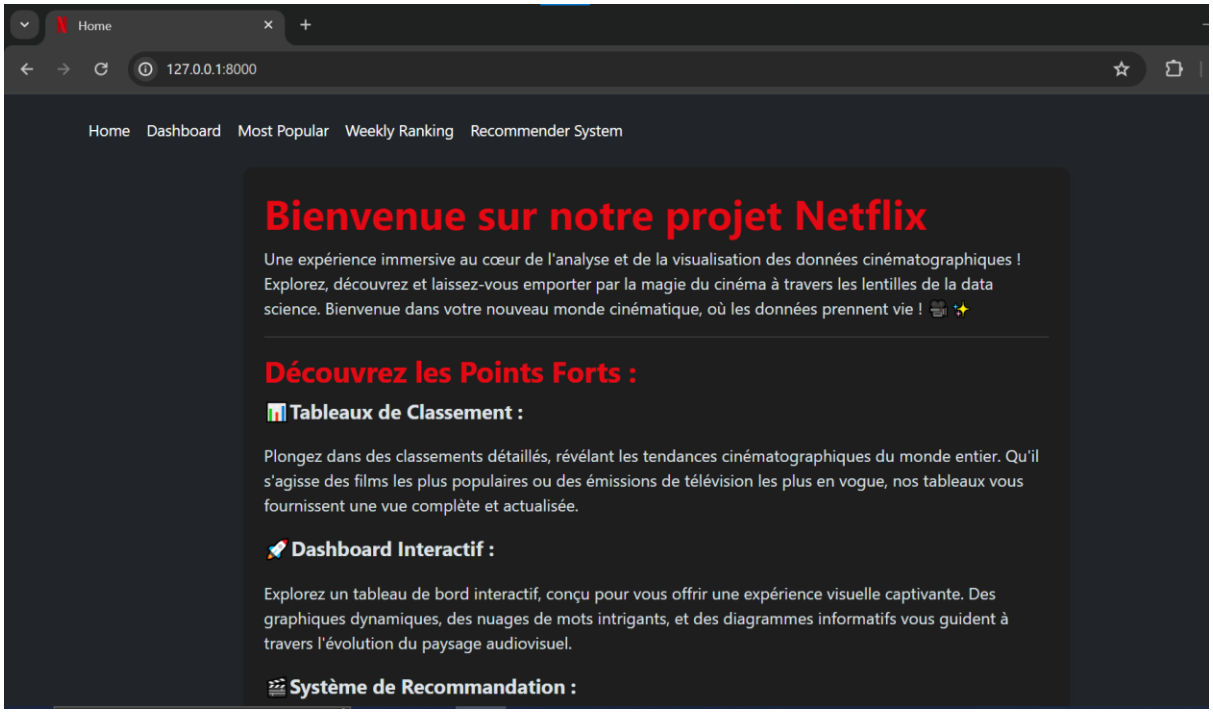

Home Dashboard Most Popular Weekly Ranking Recommender System
Most Popular Films
Choose a Language:
English
Show
#4
Top Gun: Maverick
Top 10 Popular Films : (English)
Rank
Title
Popularity
Vote Average
1
Jurassic World Dominion
10436.917
7.0
2
Minions: The Rise of Gru
8821.801
7.5


Home Dashboard Most Popular Weekly Ranking Recommender System
Top 10 Weekly Rankings
As of December, 2023
Choose a Country :
Argentina
Choose a Category :
Movies
Choose a Week :
2023-12-31
Show
Weekly Rank
Title
Cumulative Weeks in Top 10
1
Rebel Moon — Part One: A Child of Fire
2
2
The Boss Baby: Family Business
1
3
Norma
2
4
The Book of Life
1


Home Dashboard Most Popular Weekly Ranking Recommender System
NETFLIX
Dashboard
Total titles :
6234
Total TV Shows :
1969
Total Movies :
4265
Percentage of Titles on Netflix
Movie
TV Show
31.6%
68.4%
Top 10 Countries with Most Movies
Number of Movies
1500
1000
500
0
United States
India
Unknown
United Kingdom
Canada
Spain
Turkey
Mexico
France
Hong Kong
country
Top 10 Countries with Most TV Shows
Number of TV Shows
500
400
300
200
100
0
Number of Titles Over the Years
type
TV Show
Movie
Count
1000
800
600
400
200
Distribution of Age Ratings on Netflix
TV-MA
TV-14
TV-PG
R
PG-13
NR
PG
TV-Y7
TV-G
TV-Y
TV-Y7-FV
32.5%
27.2%
11.2%
4.59%
2.39%

- Ecran Smartphone :





7. Références :

https://codepen.io/joshhunt/pen/LVQZRa
https://www.kaggle.com/
https://www.netflix.com/
https://getbootstrap.com/
https://scikit-learn.org/

Construire l'URL de l'API TMDb en utilisant l'ID du film : https://api.themoviedb.org/3/movie/12?api_key=c7ec19ffdd3279641fb606d19ceb9bb1&language=en-US".