DÉVELOPPEUR DATA

SCRAPING GITHUB

Collecte des données via web Scraping

Réalisé par : OMAR DBAA

16 /06/2023

Introduction

Dans le cadre de ce projet, nous nous intéressons à l'obtention d'informations précieuses à partir des dépôts de code hébergés sur GitHub. Le contexte de notre projet repose sur la nécessité de disposer d'une connaissance approfondie des dépôts de code hébergés sur GitHub. Il s'agit d'une plateforme renommée qui regorge d'informations précieuses pour notre entreprise. En comprenant les tendances du développement, les langages les plus utilisés et les projets intéressants présents sur GitHub, nous pourrons prendre des décisions stratégiques éclairées et d'explorer de potentielles opportunités de collaboration.

Objectifs

Les objectifs de ce projet sont les suivants :

- Comprendre les tendances de développement : Nous souhaitons analyser les dépôts de code pour identifier les tendances actuelles du développement logiciel. Cela nous permettra de rester informés des évolutions technologiques et des bonnes pratiques.
- Identifier les langages de programmation les plus utilisés : Il est essentiel de connaître les langages de programmation les plus populaires sur GitHub. Cela nous permettra de mieux cibler nos ressources et d'adapter nos stratégies de développement.
- Trouver des projets intéressants : Nous voulons repérer des projets novateurs et prometteurs sur GitHub. Cela peut nous ouvrir de nouvelles opportunités de collaboration.

- Suivre les évolutions technologiques : GitHub est un reflet des avancées technologiques. En surveillant les nouveaux projets et les mises à jour, nous pourrons anticiper les changements et rester à la pointe de la technologie.

Les Étapes

1. Compréhension du Contexte : La première étape consiste à comprendre pleinement le contexte et les objectifs du projet. Cela implique une analyse approfondie des besoins de l'entreprise.
2. Recherche et Documentation : Une recherche approfondie est effectuée pour trouver les meilleures méthodes et bibliothèques permettant de collecter les données depuis GitHub. La documentation pertinente est rassemblée et analysée pour assurer une compréhension approfondie des mécanismes de collecte de données sur GitHub.
3. Définition de Méthodologie de Collecte de Données : Une méthodologie de collecte de données est mise en place en utilisant les outils et les approches identifiés lors de la phase de recherche. Des scripts Python peuvent être développés pour effectuer le scraping des données depuis GitHub de manière automatisée et efficace.
4. Surmonter les Défis : Tout au long du projet, des défis peuvent être rencontrés, tels que la gestion des taux limites imposés par l'API de GitHub. Des solutions sont adoptées pour surmonter ces défis, comme l'utilisation de stratégies de limitation de la fréquence des requêtes ou la mise en place de mécanismes de pagination pour récupérer toutes les données nécessaires.

Défis Rencontrés

Plusieurs défis ont été relevés lors de la réalisation de ce projet, notamment :

1. Gestion des taux limites de l'API de GitHub : L'API de GitHub impose des limites sur le nombre de requêtes pouvant être effectuées dans une certaine période. Il a donc été nécessaire de mettre en place des stratégies pour gérer ces taux limites afin d'éviter les restrictions d'accès et de garantir une collecte de données continue et efficace.
2. Traitement de volumes importants de données : GitHub héberge une quantité considérable de données, et il était essentiel de mettre en place des mécanismes pour gérer ces volumes importants de manière efficiente. Cela inclut l'utilisation de techniques de pagination, de parallélisation et d'optimisation des requêtes pour garantir une collecte rapide et efficace des données.
3. Sélection et extraction des informations pertinentes : Les dépôts de code sur GitHub contiennent de nombreuses informations, et il était important de sélectionner et extraire les données pertinentes pour répondre aux objectifs du projet. Cela a nécessité une analyse minutieuse des structures de données disponibles et le développement de techniques d'extraction spécifiques pour obtenir les informations souhaitées.
4. Garantie de la qualité et de la précision des données collectées : La collecte de données à partir de sources externes peut présenter des défis en termes de qualité et d'exactitude. Il a été nécessaire de mettre en place des processus de validation et de vérification des données pour s'assurer de leur fiabilité. Cela comprend la comparaison avec des sources de données connues et l'utilisation de méthodes de nettoyage et de normalisation des données.

5. Collecte de plus de données dans un délai limité : Le projet visait à collecter un large éventail de données pertinentes sur une période de temps définie. Cependant, le temps imparti était limité, ce qui a nécessité une optimisation du processus de collecte pour obtenir le maximum de données dans les délais impartis.
6. Un autre défi important rencontré lors de la réalisation de ce projet était le manque de familiarité avec les bibliothèques Python appropriées pour la collecte de données depuis GitHub.

Résultats

```
df= pd.read_csv('repositories.csv')
df.head()
```

full_name	url	stargazers_count	language	license	topics	forks	issues_count	year	watchers_count	created_at
mous/ComfyUI	https://github.com/comfyanonymous/ComfyUI	6157	Python	GNU General Public License v3.0	['stable-diffusion']	553	271	2023	6157	2023-01-17T03:15:56Z
shit06/typewind	https://github.com/Mokshit06/typewind	2070	TypeScript	MIT License	[]	24	12	2023	2070	2023-01-17T15:38:59Z
g-video-lecture	https://github.com/karpathy/ng-video-lecture	1999	Python	NaN	[]	468	15	2023	1999	2023-01-17T05:27:03Z
x-labs/readpilot	https://github.com/index-labs/readpilot	1081	TypeScript	MIT License	['gpt3', 'nextjs', 'openai', 'react', 'tailwin...	67	5	2023	1081	2023-01-17T17:24:08Z
n/CookieCloud	https://github.com/easychen/CookieCloud	719	JavaScript	GNU General Public License v3.0	[]	69	12	2023	719	2023-01-17T03:11:37Z

```
df.columns
```

```
Index(['id', 'full_name', 'url', 'stargazers_count', 'language', 'license',
       'topics', 'forks', 'issues_count', 'year', 'watchers_count',
       'created_at'],
      dtype='object')
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75096 entries, 0 to 75095
Data columns (total 12 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   id                 75096 non-null  int64 
 1   full_name          75096 non-null  object
 2   url                75096 non-null  object
 3   stargazers_count   75096 non-null  int64 
 4   language           64274 non-null  object
 5   license            36970 non-null  object
 6   topics             75096 non-null  object
 7   forks              75096 non-null  int64 
 8   issues_count       75096 non-null  int64 
 9   year               75096 non-null  int64 
 10  watchers_count     75096 non-null  int64 
 11  created_at         75096 non-null  object
dtypes: int64(6), object(6)
memory usage: 6.9+ MB
```