

Groupe 4

Documentation du Projet : Analyse des Performances des Équipes de Hockey

Scraping et Analyse des données avec Python

Table des matières

1. Introduction	2
1.1 Contexte du projet.....	2
1.2 Objectifs.....	2
2. Technologies Utilisées	2
3. Architecture du Projet	3
3.1 Vue d'ensemble	3
3.2 Structure des fichiers et dossiers.....	3
4. Extraction et Stockage des Données	4
4.1 Web Scraping avec BeautifulSoup	4
4.2 Structure du Dataset	4
5. Analyse et Visualisation des Données	4
5.1 Analyse des performances	4
5.2 Visualisation des données	5
6. Interface Utilisateur.....	5
6.1 Fonctionnalités principales.....	5
7. Conclusion et Perspectives	5

1. Introduction

1.1 Contexte du projet

L'analyse des performances des équipes de hockey est essentielle pour comprendre leur évolution, identifier les tendances et comparer les performances sur plusieurs saisons. Ce projet vise à développer une application web interactive permettant d'exploiter des données historiques sur les équipes de hockey à travers une approche combinant extraction, analyse et visualisation des données.

Le web scraping est utilisé pour collecter dynamiquement les données d'un site spécialisé (Scrape This Site). Une fois les données extraites, elles sont nettoyées, transformées et analysées à l'aide de bibliothèques Python dédiées à la manipulation et à l'analyse statistique des données.

Enfin, une interface web intuitive permet aux utilisateurs de consulter les statistiques avancées et de visualiser les résultats sous forme de graphiques interactifs.

1.2 Objectifs

Ce projet couvre plusieurs compétences techniques et méthodologiques :

1. **Mise en œuvre du web scraping** : Extraction des données depuis un site dynamique avec BeautifulSoup et requests.
2. **Manipulation et analyse des données** : Utilisation de Pandas et NumPy pour le traitement et l'analyse des données tabulaires.
3. **Application des concepts statistiques** : Calcul des indicateurs de performance tels que la moyenne, la médiane et l'écart-type.
4. **Visualisation interactive des données** : Génération de graphiques avec Matplotlib et Seaborn.
5. **Développement d'une interface web** : Construction d'une API avec Flask permettant aux utilisateurs de consulter et d'explorer les données.

2. Technologies Utilisées

Catégorie	Outils
Web Scraping	BeautifulSoup, requests
Analyse des données	Pandas, NumPy
Visualisation	Matplotlib, Seaborn
Backend	Flask
Frontend	HTML, CSS, JavaScript
Gestion de fichiers	CSV, Excel, PDF (via reportlab)

3. Architecture du Projet

3.1 Vue d'ensemble

Le projet est structuré en plusieurs modules :

1. **Scraping des données** : Extraction des performances des équipes de hockey depuis le site cible.
2. **Stockage et analyse des données** : Manipulation, nettoyage et analyse statistique des données scrappées.
3. **Visualisation des données** : Génération de graphiques pour représenter les tendances et les corrélations.
4. **Interface web** : Développement d'une application web interactive pour afficher les résultats.

3.2 Structure des fichiers et dossiers

```
└─ fops4-tp_scraping/
    └─ README.md          # Documentation du projet
    └─ LICENSE.chromedriver # Licence associée à ChromeDriver
    └─ THIRD_PARTY_NOTICES.chromedriver # Notices tierces pour ChromeDriver
    └─ analyzer.py        # Analyse et traitement des données
    └─ app.py             # Application Flask ou FastAPI pour l'interface web
    └─ dataset.csv        # Données scrappées stockées en CSV
    └─ scrape_all_teams.py # Script pour récupérer les données de toutes les équipes
    └─ scraper.py         # Module principal pour le scraping des données
    └─ t.py               # Fichier temporaire/test
    └─ visualizer.py      # Génération et affichage des graphiques
    └─ static/            # Ressources statiques
    |   └─ hockey_stats_new_york_islanders.csv # Exemple de dataset par équipe
    |   └─ script.js      # JavaScript pour interactivité de l'interface web
    |   └─ style.css       # Feuille de style pour l'interface web
    |   └─ graphs/        # Dossier contenant les graphiques générés
```

```
└─ templates/           # Fichiers HTML pour l'interface utilisateur
|  └─ index.html        # Page principale de l'application web
└─ test/                # Dossier contenant des fichiers de test et de sauvegarde
    └─ LICENSE.chromedriver
    └─ THIRD_PARTY_NOTICES.chromedriver
    └─ backup_dataset.csv # Copie de sauvegarde des données
    └─ dataset.csv       # Dataset utilisé pour les tests
    └─ scrape_all_teams.py # Script de test pour le scraping
```

4. Extraction et Stockage des Données

4.1 Web Scraping avec BeautifulSoup

Le projet utilise BeautifulSoup et requests pour extraire les données du site cible. Les étapes sont les suivantes :

1. Envoi d'une requête HTTP au site cible avec requests.
2. Analyse du HTML avec BeautifulSoup pour extraire les informations pertinentes.
3. Stockage des données extraites dans un fichier CSV (dataset.csv).

4.2 Structure du Dataset

Les données extraites contiennent les champs suivants :

- Nom de l'équipe
- Année
- Nombre de victoires, défaites et défaites en prolongation
- Pourcentage de victoires
- Buts marqués et encaissés

5. Analyse et Visualisation des Données

5.1 Analyse des performances

L'analyse repose sur les métriques suivantes :

- **Tendance des victoires** : évolution des performances au fil des saisons.

- **Corrélations** : relation entre les victoires et d'autres facteurs comme les buts marqués.
- **Comparaison des équipes** : classement basé sur le pourcentage de victoires et la solidité défensive.

5.2 Visualisation des données

Les données sont illustrées à l'aide de différents types de graphiques :

- **Matplotlib** : courbes, histogrammes, barres.
- **Seaborn** : boxplots, heatmaps et scatter plots.

6. Interface Utilisateur

L'application web permet de rechercher une équipe et d'afficher ses performances sous forme de tableaux et graphiques.

6.1 Fonctionnalités principales

1. **Recherche d'une équipe** : affichage des statistiques dynamiques.
2. **Analyse des performances** : statistiques descriptives et corrélations.
3. **Comparaison des équipes** : graphiques de comparaison.
4. **Export des données** : possibilité de télécharger les données analysées sous format CSV, Excel ou PDF.

7. Conclusion et Perspectives

Ce projet illustre l'utilisation du web scraping, de l'analyse de données et de la visualisation interactive pour extraire des insights sur les performances des équipes de hockey.

Perspectives d'amélioration :

- Automatisation du scraping pour des mises à jour en temps réel.
- Ajout d'un modèle prédictif basé sur l'apprentissage machine.
- Amélioration de l'interface utilisateur avec des tableaux interactifs et des filtres avancés.