ESSECT
Ecole Supérieure des Sciences Economiques et Commerciales de Tunis

Rapport de Projet BI

Analyse des données de basketball

Mission : La Création d' un tableau de bord interactif

Elaboré par : Rouahi Hajer & Aouinet Hamis & Chedli Haroun

Enseignants : Mattoussi Helmi et Zouhaier Lamia

Année universitaire : 2024-2025

Introduction

Contexte du projet

Le projet consiste à analyser et visualiser les données de basketball à partir de la plateforme **Basketball Reference** (https://www.basketball-reference.com/), une référence incontournable pour les statistiques des joueurs et des équipes de basketball. Cette plateforme fournit une multitude de données détaillées sur les performances des joueurs, les résultats des équipes, les saisons passées et actuelles, ainsi que d'autres statistiques pertinentes pour les amateurs et les professionnels du basket.

L'objectif du projet est de créer un **tableau de bord interactif** permettant de visualiser ces données de manière claire et dynamique. Ce tableau de bord aura pour but d'afficher des statistiques sur les joueurs et les équipes, en permettant aux utilisateurs d'explorer différentes métriques (par exemple, points marqués, rebonds, victoires, etc.) au fil des saisons. Ces visualisations permettront d'obtenir des insights pertinents et d'améliorer la compréhension des performances dans le domaine du basketball.

Outils et technologies utilisées

Pour mener à bien ce projet, plusieurs outils et bibliothèques ont été utilisés pour l'extraction des données, la manipulation des informations, la création des visualisations et l'interactivité du tableau de bord. Voici les principales technologies utilisées :

- **Dash** : Dash est un Framework Python qui permet de créer des applications web interactives. Il est utilisé pour concevoir l'interface du tableau de bord et gérer l'interactivité entre les utilisateurs et les visualisations. Grâce à Dash, nous avons pu créer une interface fluide et réactive pour afficher et interagir avec les graphiques.
- **Dash Core Components (dcc)** et **Dash HTML Components (html)** : Ces modules de Dash sont utilisés pour ajouter des éléments interactifs et structurer l'interface utilisateur, comme les graphiques, les menus déroulants, et les boutons permettant de filtrer ou d'explorer les données.
- **Plotly Express** : Plotly Express est une bibliothèque puissante pour la visualisation interactive de données. Elle est utilisée pour créer des graphiques interactifs, tels que des graphiques en barres, en lignes, et en nuages de points, permettant aux utilisateurs d'explorer les données en temps réel.
- **Pandas** : Pandas est une bibliothèque Python essentielle pour le traitement des données. Elle est utilisée pour organiser et manipuler les données extraites sous forme de **DataFrame**, ce qui permet de nettoyer, filtrer et structurer les informations avant de les exploiter pour la création des graphiques.
- **Requests** : La bibliothèque **requests** est utilisée pour effectuer des requêtes HTTP et récupérer les pages HTML du site Basketball Reference. Elle est utilisée pour télécharger les données nécessaires pour l'analyse.
- **BeautifulSoup** : BeautifulSoup est une bibliothèque de **scraping** utilisée pour extraire les données spécifiques des pages HTML récupérées. Elle permet de parcourir le contenu des pages et d'en extraire les informations pertinentes sur les joueurs, les équipes et les saisons.

- **Datetime** : Le module **datetime** est utilisé pour travailler avec les dates et les heures, notamment pour gérer les dates des saisons de basketball et formater les informations de manière adéquate.
- **Re (expressions régulières)** : Le module **re** est utilisé pour effectuer des recherches et manipuler des chaînes de caractères, particulièrement utile pour extraire des données spécifiques de textes complexes, comme les statistiques ou les dates, à l'aide d'expressions régulières.
- **URLJoin** : Cette bibliothèque permet de gérer les liens relatifs et absolus, garantissant que les URLs extraites des pages sont correctement formatées pour être utilisées dans le processus de scraping.

Étape 1 : Extraction des données (Scraping)

Objectif de l'extraction des données

L'objectif de cette étape est d'extraire les données pertinentes à partir de la plateforme **Basketball Reference** pour analyser et visualiser les performances des joueurs, des équipes et des saisons de basketball. Le scraping des données permet de collecter des informations directement à partir du site web, en utilisant des techniques d'extraction automatique. Cela permet d'obtenir un ensemble de données structuré, prêt à être analysé et visualisé dans un tableau de bord interactif.

Nous avons ciblé des informations spécifiques liées aux **joueurs**, **équipes**, et **saisons** afin de fournir une vue d'ensemble complète des performances dans le domaine du basketball. Ces données sont essentielles pour les visualisations interactives que nous avons créées, permettant aux utilisateurs d'explorer les statistiques de manière dynamique.

Méthode d'extraction

Le processus de scraping a été effectué en utilisant la bibliothèque **BeautifulSoup** pour analyser le HTML du site Basketball Reference. Nous avons ciblé des pages spécifiques contenant des tableaux de statistiques, tels que les pages de chaque saison ou les pages de chaque joueur ou équipe. Les données ont été extraites et nettoyées avant d'être structurées sous forme de **DataFrame** avec l'aide de **Pandas** pour faciliter leur manipulation et leur analyse ultérieure.

Code :

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

url = "https://www.basketball-reference.com/teams/"
response = requests.get(url)

if response.status_code == 200:
    # Parser le contenu HTML avec BeautifulSoup
    soup = BeautifulSoup(response.text, 'html.parser')
```

```
    table = soup.find('table', {'id': 'teams_active'})
    if table:
        team_rows = table.find_all('tr', {'class': 'full_table'})
        all_teams_data = []

        for row in team_rows:
            team_data = {
                "Name": row.find('th', {'data-stat':
'franch_name'}).text.strip(),
                "League": row.find('td', {'data-stat':
'lg_id'}).text.strip(),
                "YearMin": row.find('td', {'data-stat':
'year_min'}).text.strip(),
                "YearMax": row.find('td', {'data-stat':
'year_max'}).text.strip(),
                "Years": int(row.find('td', {'data-stat':
'years'}).text.strip()),
                "Games": int(row.find('td', {'data-stat':
'g'}).text.strip()),
                "Wins": int(row.find('td', {'data-stat':
'wins'}).text.strip()),
                "Losses": int(row.find('td', {'data-stat':
'losses'}).text.strip()),
                "Win-Loss%": float(row.find('td', {'data-stat':
'win_loss_pct'}).text.strip()),
                "PlayoffYears": int(row.find('td', {'data-stat':
'years_playoffs'}).text.strip()),
                "DivisionChampionships": int(row.find('td', {'data-
stat': 'years_division_champion'}).text.strip()),
                "ConferenceChampionships": int(row.find('td', {'data-
stat': 'years_conference_champion'}).text.strip() ),
                "LeagueChampionships": int(row.find('td', {'data-stat':
'years_league_champion'}).text.strip() )
            }

            all_teams_data.append(team_data)

        df = pd.DataFrame(all_teams_data)
        print(df)

        # exportation de   DataFrame dans un fichier CSV
        df.to_csv('teams_data.csv', index=False, encoding='utf-8')
        print("Les données ont été exportées dans 'teams_data.csv'.")
    else:
        print("Le tableau 'teams_active' n'a pas été trouvé.")
else:
    print(f"Erreur lors de la requête : {response.status_code}")
```

Étape 2 : Nettoyage des données

Objectif du nettoyage des données

L'objectif de cette étape est de préparer les données extraites du site **Basketball Reference** pour leur utilisation dans les visualisations et analyses. Le nettoyage des données consiste à traiter les incohérences, les valeurs manquantes, les doublons, et à formater les données pour qu'elles soient cohérentes et prêtes à être exploitées dans des graphiques interactifs. Un nettoyage minutieux garantit des visualisations fiables et précises qui fournissent une interprétation correcte des données.

Traitement des données manquantes

Le traitement des données manquantes est une étape cruciale pour assurer la qualité et la fiabilité des analyses. Lors de l'extraction des données, il est fréquent de rencontrer des **valeurs manquantes** (nulles ou NaN), surtout dans des statistiques moins courantes ou pour certaines saisons où les joueurs ou équipes ne sont pas toujours présents. Voici les méthodes que nous avons utilisées pour traiter ces données manquantes :

- **Identification des données manquantes** :
    - Nous avons d'abord utilisé **Pandas** pour identifier les valeurs manquantes dans le **DataFrame**. Cela a été réalisé avec des méthodes comme `df.isnull()` et `df.isna()` pour repérer les colonnes ou lignes qui contiennent des données absentes.
    - Après l'identification, un rapport a été généré pour évaluer l'étendue des données manquantes.

      Code :

```
#Traiter les valeurs manquantes

print(df.isnull().sum())
```

- **Suppression des lignes ou des colonnes vides** :
    - Si certaines **colonnes** comportaient trop de valeurs manquantes (plus de 40-50 % de valeurs absentes), elles ont été complètement **supprimées** du **DataFrame**, car leur manque d'informations les rendait inutilisables pour une analyse significative.
    - Si une **ligne** comportait un joueur ou une équipe dont les informations étaient incomplètes et qu'elle ne pouvait pas être complétée de manière logique, nous avons aussi choisi de **supprimer** ces lignes.

- **Remplacement des valeurs manquantes** :
    - Dans le cas des **valeurs manquantes non critiques**, telles que les statistiques de rebonds ou de passes pour certains joueurs, nous avons appliqué des méthodes de

remplacement. Les valeurs manquantes ont été remplacées par des valeurs **par défaut** comme des zéros, ou dans certains cas, par des **moyennes** calculées sur les autres valeurs disponibles pour ces statistiques. Par exemple, si un joueur avait des valeurs manquantes pour les rebonds, la moyenne des rebonds des autres joueurs de la même saison pouvait être utilisée pour remplacer ces valeurs.
- Pour certaines statistiques temporelles, comme les **dates**, des valeurs manquantes ont été remplacées par la **date de début** de la saison en cours, si l'année de la saison était présente.

- **Correction des incohérences de format** :

  - Certaines colonnes, telles que les **noms des joueurs**, les **équipes**, ou les **dates**, peuvent comporter des incohérences de format (espaces supplémentaires, erreurs de saisie, etc.). Nous avons utilisé des fonctions comme `str.strip()` pour enlever les espaces superflus et `str.title()` pour uniformiser les majuscules dans les noms de joueurs ou d'équipes.
  - Les **valeurs numériques** (par exemple, points par match, rebonds par match) ont été converties en types numériques, afin d'éviter les erreurs de type lors de l'analyse ou de la création des graphiques.

- **Suppression des doublons** :

  - Après avoir nettoyé les données, nous avons vérifié la présence de **doublons** dans les données extraites. Par exemple, si un même joueur apparaissait plusieurs fois pour une même saison, ces doublons ont été éliminés à l'aide de la méthode `df.drop_duplicates()`, pour éviter des biais dans les visualisations et analyses.

Code :

```
#Suppression   les doublons :
df = df.drop_duplicates()
print(df)
```

Résultat du nettoyage

À la fin de cette étape, les données extraites étaient prêtes pour une analyse plus approfondie et pour la création des visualisations interactives. Les valeurs manquantes ont été traitées de manière appropriée, les incohérences corrigées, et les doublons éliminés, assurant ainsi la qualité des données. Cela a permis de construire un **jeu de données propre** qui reflète fidèlement les performances des joueurs et des équipes de basketball.

Étape 3 : Visualisation des données

Objectif de la visualisation des données

L'objectif de cette étape est de créer des visualisations interactives et informatives à partir des données nettoyées, permettant ainsi une analyse approfondie des performances des joueurs et des équipes de basketball. Ces graphiques visent à rendre les données facilement compréhensibles et accessibles, tout en offrant des outils interactifs permettant à l'utilisateur d'explorer différentes facettes des données.

Pour cela, nous avons utilisé des outils de visualisation comme **Plotly** et **Dash**, qui permettent de créer des graphiques dynamiques et interactifs. Ces outils offrent également la possibilité de filtrer, zoomer, et interagir avec les données de manière intuitive.

Code :

```
import dash
from dash import dcc, html, Input, Output
import pandas as pd
import plotly.express as px

# Charger le fichier CSV
file = 'teams_dataset.csv'
df_teams = pd.read_csv(file)


# Initialiser l'application Dash
app = dash.Dash(__name__)
app.title = "Tableau de Bord des équipes de basketball"

# Initialiser l'application Dash
app.layout = html.Div([

    html.H1("Analyse des performances des équipes ",
style={'textAlign': 'center'}),

    # Dropdown pour sélectionner une équipe
    html.Div([
        html.Label("Choisir une équipe :"),
        dcc.Dropdown(
            id='teamName',
            options=[{'label': team, 'value': team} for team in
df_teams['Name'].unique()],
            value=df_teams['Name'].unique()[0],
            clearable=False,
            style={'width': '50%'}
        )
    ], style={'marginBottom': '30px', 'textAlign': 'center'}),
```

```
    # Graphique 1 : Victoires et défaites par équipe
    dcc.Graph(id='wins-losses-bar-chart'),

    # Graphique 2 : Corrélation entre Win% et PlayoffYears
    dcc.Graph(id='win-playoff-scatter', style={'marginTop': '30px'}),

    # Graphique 3 : Top 10 des équipes par Win%
    dcc.Graph(id='top-teams-bar-chart', style={'marginTop': '30px'}),

    # Graphique 4 : nombre de victoires et de défaites, et nombre de
games par ligue
    dcc.Graph(id='league-win-losses-games', style={'marginTop':
'30px'}),

])
```

Étape 4 : Tableau de bord interactif

Objectif

L'objectif de cette étape est de regrouper toutes les visualisations créées précédemment dans un tableau de bord interactif. Ce tableau de bord permettra à l'utilisateur d'explorer les données sur les joueurs et les équipes de manière dynamique et intuitive. Le tableau de bord combinera plusieurs graphiques interactifs avec des composants de filtrage pour permettre à l'utilisateur de personnaliser et affiner son analyse des statistiques de basketball.

Composants du tableau de bord

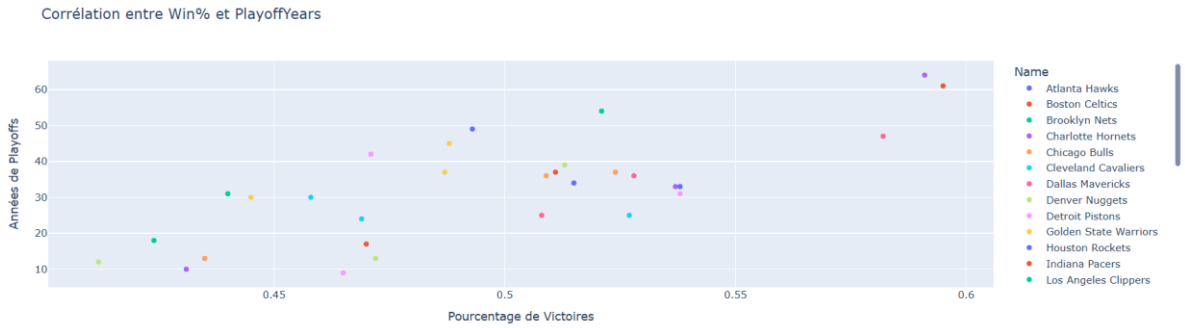
Le tableau de bord interactif comprend plusieurs composants essentiels :

1. **Graphiques interactifs** :
    - **Graphique sur les joueurs** : Un graphique en barres représentant les meilleurs marqueurs ou d'autres statistiques pertinentes des joueurs.
    - **Graphique sur les équipes** : Un graphique en nuage de points pour comparer les performances des équipes en termes de victoires et de défaites.
    - **Graphique sur les saisons** : : Un graphique en barres représentant la comparaison des leaders par saison.
2. **Filtres et sélections** :
    - **Sélection de la saison** : Un filtre pour que l'utilisateur puisse sélectionner une saison spécifique pour voir les données correspondantes.
    - **Sélection des joueurs ou équipes** : Permet à l'utilisateur de filtrer les données en fonction des joueurs ou des équipes spécifiques.
    - **Filtres supplémentaires** : D'autres filtres interactifs peuvent être ajoutés pour sélectionner des statistiques spécifiques ou trier les joueurs par position, performances, etc.
3. **Tableau récapitulatif** :
    - Un tableau interactif qui affiche des informations détaillées sur les joueurs ou les équipes en fonction des filtres appliqués.

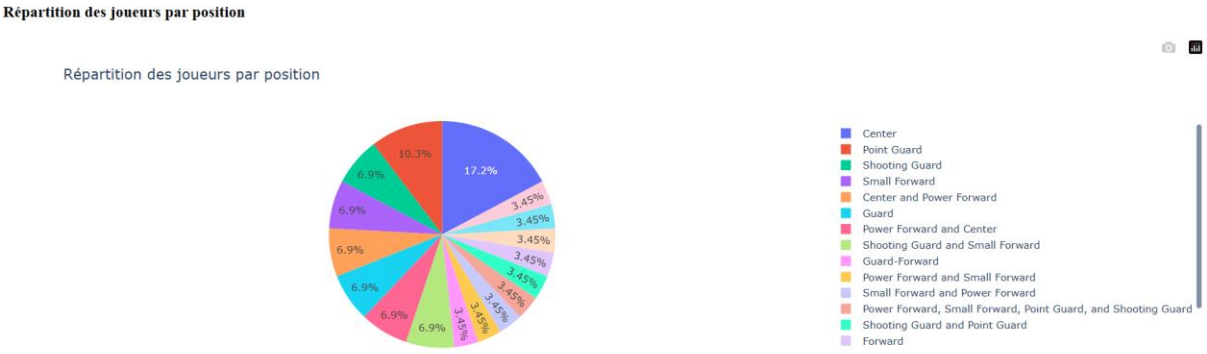
Étape 5 : Interprétation des visuels

Objectif : L'interprétation des visualisations est essentielle pour fournir une compréhension plus approfondie des données présentées. Chaque graphique et composant interactif du tableau de bord a été choisi en fonction de l'objectif du projet : rendre les données du basketball accessibles, compréhensibles et dynamiques.

- **Nuages de points** : Pour visualiser les relations entre différentes variables, comme les performances des équipes, tout en offrant une vue dynamique et interactive.



- **Filtres interactifs** : Pour offrir aux utilisateurs la possibilité de personnaliser l'affichage des données selon leurs préférences et besoins spécifiques, améliorant ainsi l'expérience d'analyse.



⇨ Ces visuels ont été choisis pour permettre une exploration détaillée et personnalisée des performances des joueurs et des équipes.

Conclusion

Résumé des étapes du projet

Ce projet a suivi plusieurs étapes clés :

1. **Extraction des données** : Nous avons extrait les données pertinentes depuis le site Basketball Reference, incluant des informations sur les joueurs, les équipes et les saisons.
2. **Nettoyage des données** : Les données ont été nettoyées pour supprimer les valeurs manquantes et structurer les informations de manière appropriée.
3. **Création des visualisations** : Des graphiques interactifs ont été générés pour visualiser les performances des joueurs et des équipes.
4. **Construction du tableau de bord** : Un tableau de bord interactif a été créé pour regrouper et afficher les visualisations de manière interactive.