



Analyse Big Data UFC avec Kafka et Elasticsearch

Rapport de Projet - UFC Analysis



Big Data Technologies - Semestre 8

International University of Rabat (UIR)

Elabore par :

Fadi BAHTAT (117970)

Ilyas HIMIT (117054)

Supervisé par :

Mr. Hamza GAMOUH

Mme. Nada SBIHI

Table des matières

1. Introduction	3
2. Architecture et fonctionnement de Kafka	3
3. Implémentation du projet.....	4
1.1. Production des données UFC.....	4
1.1. Configuration du connecteur Elasticsearch Sink	4
3. Création de l'index pattern dans Kibana	4
4. Construction des visualisations et dashboards	4
4. Fonctionnalités avancées	4
5. Visualisations clés créées	5
6. Conclusion	6

1. Introduction

- **Contexte du projet** : Analyse des données UFC pour extraire des insights sur les performances des combattants.
- **Objectif** : Construire un pipeline Big Data utilisant Apache Kafka pour ingérer, traiter et visualiser ces données en temps quasi réel.

2. Architecture et fonctionnement de Kafka

Apache Kafka

- Système de streaming distribué et décentralisé, utilisé pour le traitement de flux de données en temps réel.
- Composé de plusieurs composants clés :
 - Producer : application qui envoie des messages dans un topic Kafka.
 - Broker : serveur Kafka stockant les données des topics.
 - Consumer : application qui lit les messages depuis un topic.
 - Connectors : plugins pour intégrer Kafka avec d'autres systèmes (bases de données, Elasticsearch, etc.).

Architecture de notre projet

- **Producer** : script Python qui récupère les données des combattants UFC via une API, et envoie ces données dans un topic Kafka nommé 'ufc-events'.
- **Kafka Broker** : stocke et distribue les événements aux consommateurs.
- **Kafka Connect Elasticsearch Sink** : connecteur qui consomme les événements du topic 'ufc-events' et indexe les données dans Elasticsearch.
- **Elasticsearch** : moteur de recherche et stockage pour l'analyse et les visualisations.
- **Kibana** : outil de visualisation pour créer des dashboards interactifs basés sur les données UFC.

3. Implémentation du projet

Étapes clés

3.1. Production des données UFC

- Script Python (producer.py) récupérant les informations des combattants (ex: nom, poids, victoires, défaites, méthodes de victoire).
- Envoi de chaque événement dans le topic Kafka.

3.2. Configuration du connecteur Elasticsearch Sink

- Création d'un fichier JSON 'es-ufc-sink.json' pour configurer le connecteur.
- Envoi de la configuration via Kafka Connect.
- Les données sont indexées dans Elasticsearch dans l'index 'ufc-events'.

3.3. Création de l'index pattern dans Kibana

- Définition de l'index pattern 'ufc-events*' pour permettre la recherche et visualisation.
- Mapping des champs (ex : 'Wins' en entier, 'WeightClass.keyword' en mot-clé).

3.4. Construction des visualisations et dashboards

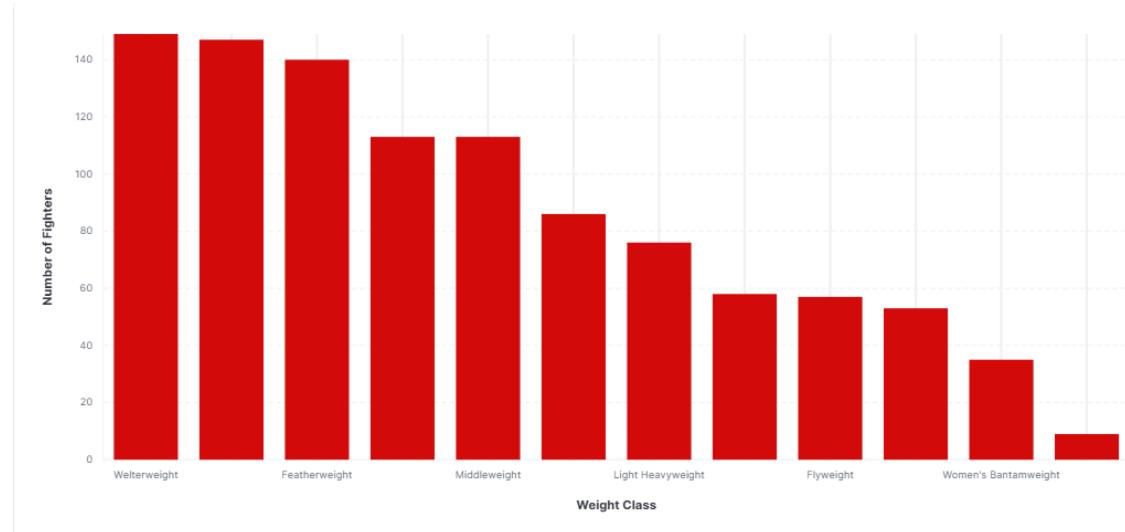
- Création de plusieurs visualisations pertinentes (exemples dans la section suivante).
- Assemblage de ces visualisations dans un dashboard interactif.

4. Fonctionnalités avancées

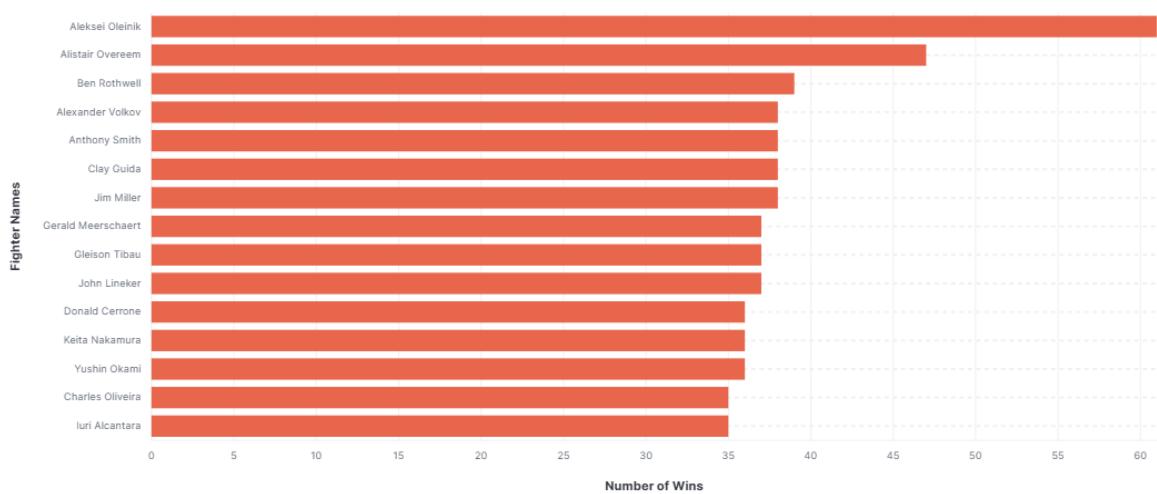
- Calculs personnalisés dans Elasticsearch pour déterminer la méthode de victoire principale d'un combattant (script Painless).
- Filtres dynamiques dans Kibana pour explorer les données par catégorie de poids, méthode de victoire, années, etc.
- Actualisation continue grâce à Kafka pour intégrer les mises à jour des données UFC en temps réel.

5. Visualisations clés créées

■ Nombre de victoires par catégorie de poids : Top 15 des combattants par nombre de victoires



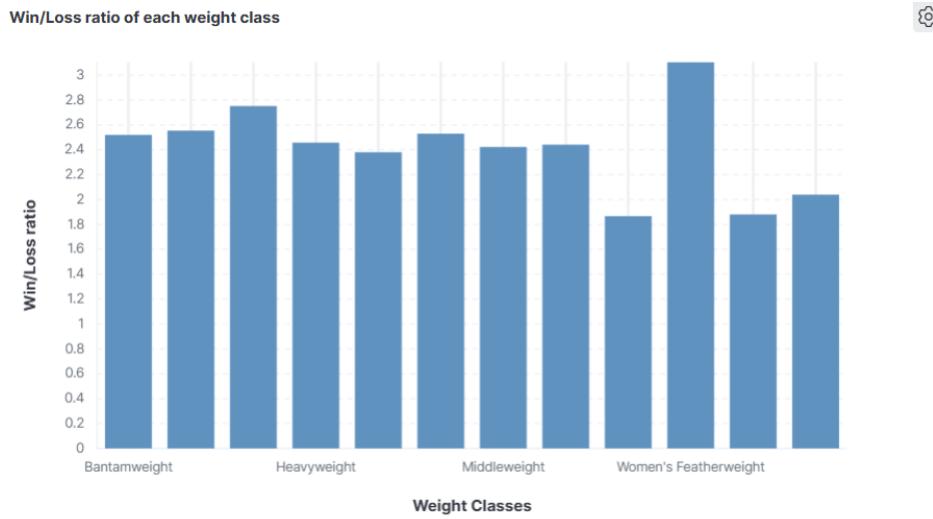
■ Top 15 des combattants par nombre de victoires



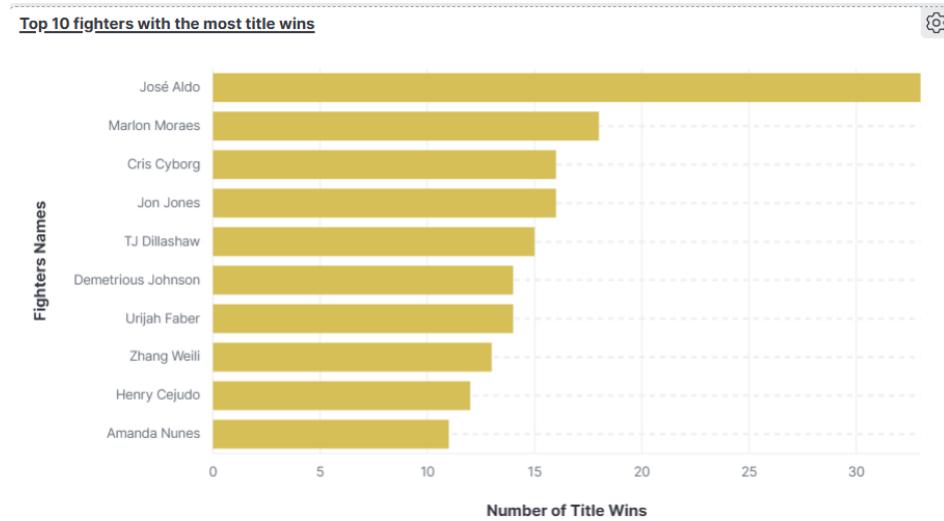
■ Tableau des combattants invaincus

Fighter	Weight Class	Wins	Losses
Khabib Nurmagomedov	Lightweight	29	0
Movsar Evloev	Featherweight	19	0
Shavkat Rakhmonov	Welterweight	19	0
Lerone Murphy	Featherweight	16	0
Aliaskhab Khizriev	Middleweight	14	0
Khamzat Chimaev	Middleweight	14	0
Ramazan Kuramagomedov	Middleweight	13	0
Dmitry Sosnovskiy	Heavyweight	11	0
Taylor Callens	Lightweight	6	0

■ Ratio Victoires/Défaites de chaque catégorie de poids

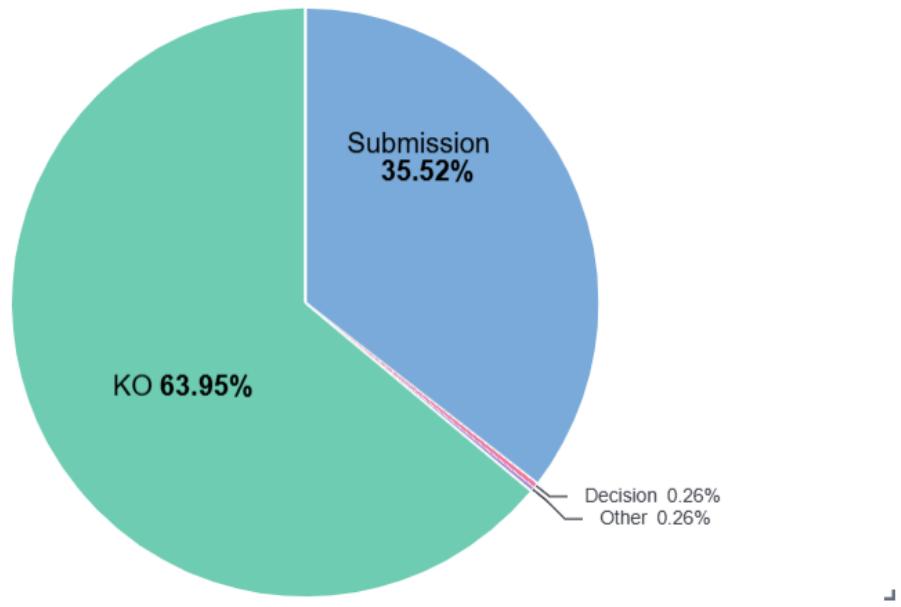


■ Top 10 combattants avec le plus de victoires en titre



■ % de victoires par méthode de victoire

% of wins by Method

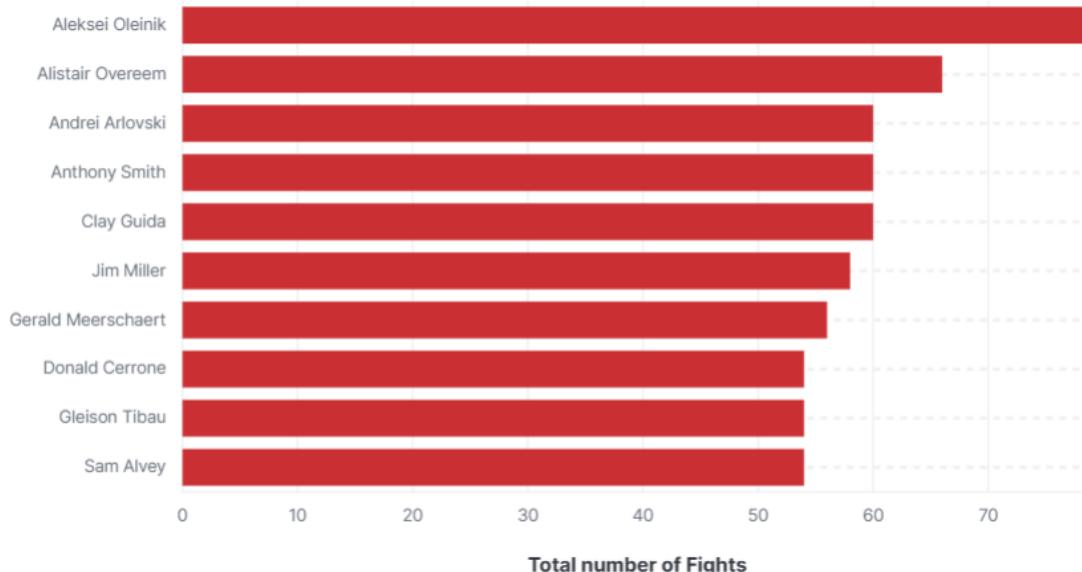


■ Top 10 des combattants les plus expérimentés (le plus de combats)

Top 10 Most Experienced Fighters (Most fights)

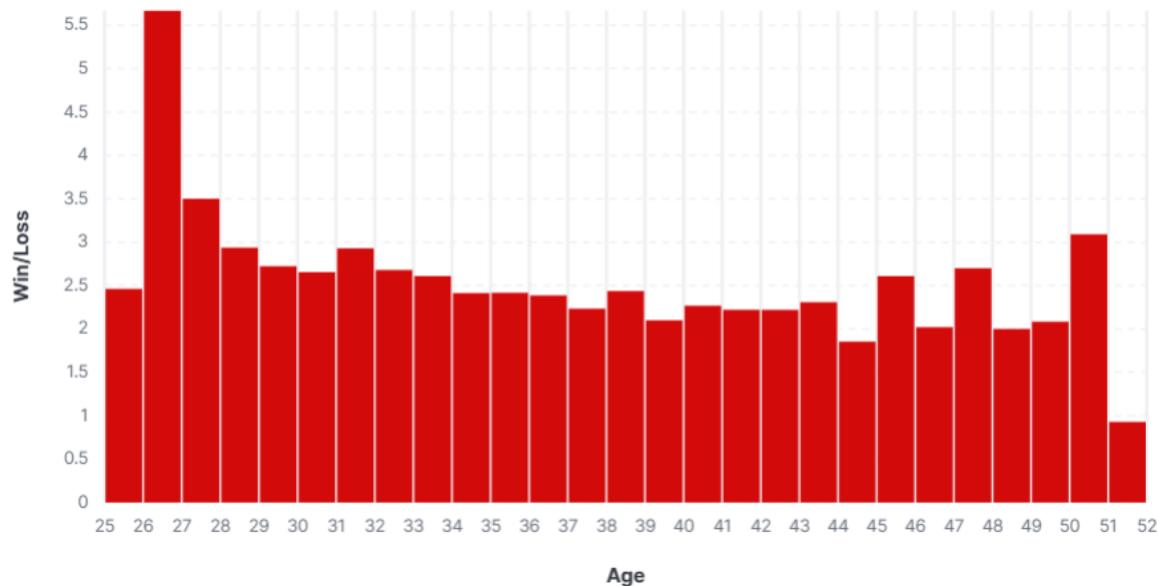


Full Name



■ Age du combattant vs performance (victoires/défaites)

Fighter Age vs Performance (Win/Loss)

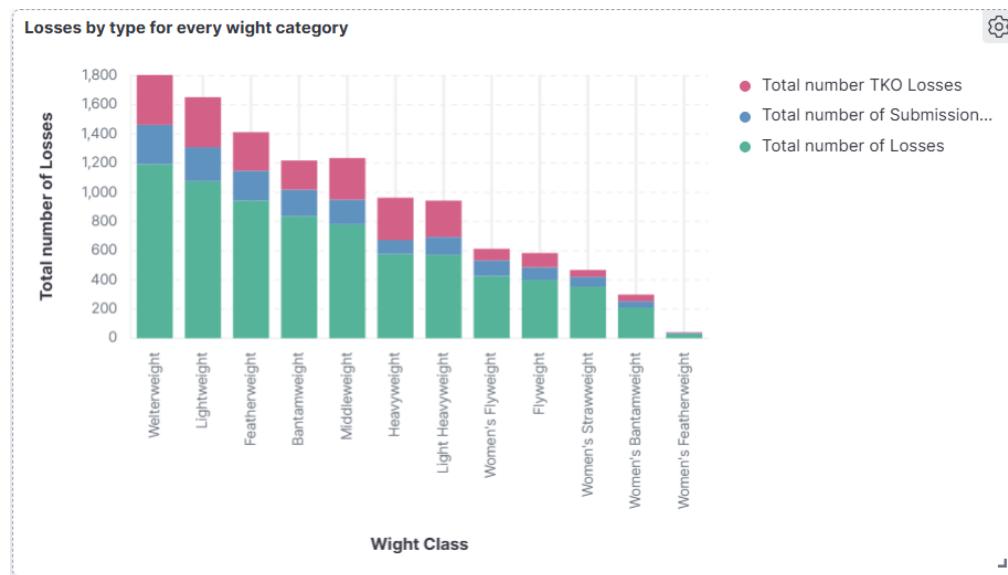


■ Distribution des types de victoires par catégorie de poids

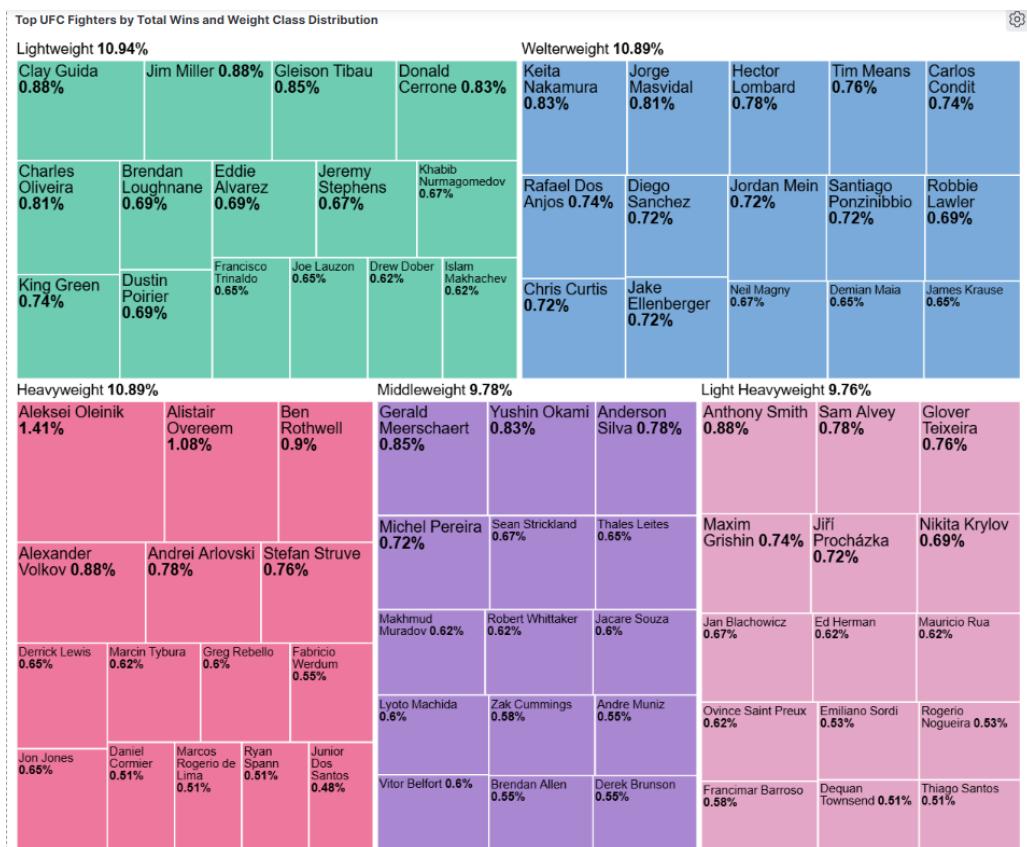
[No Title]



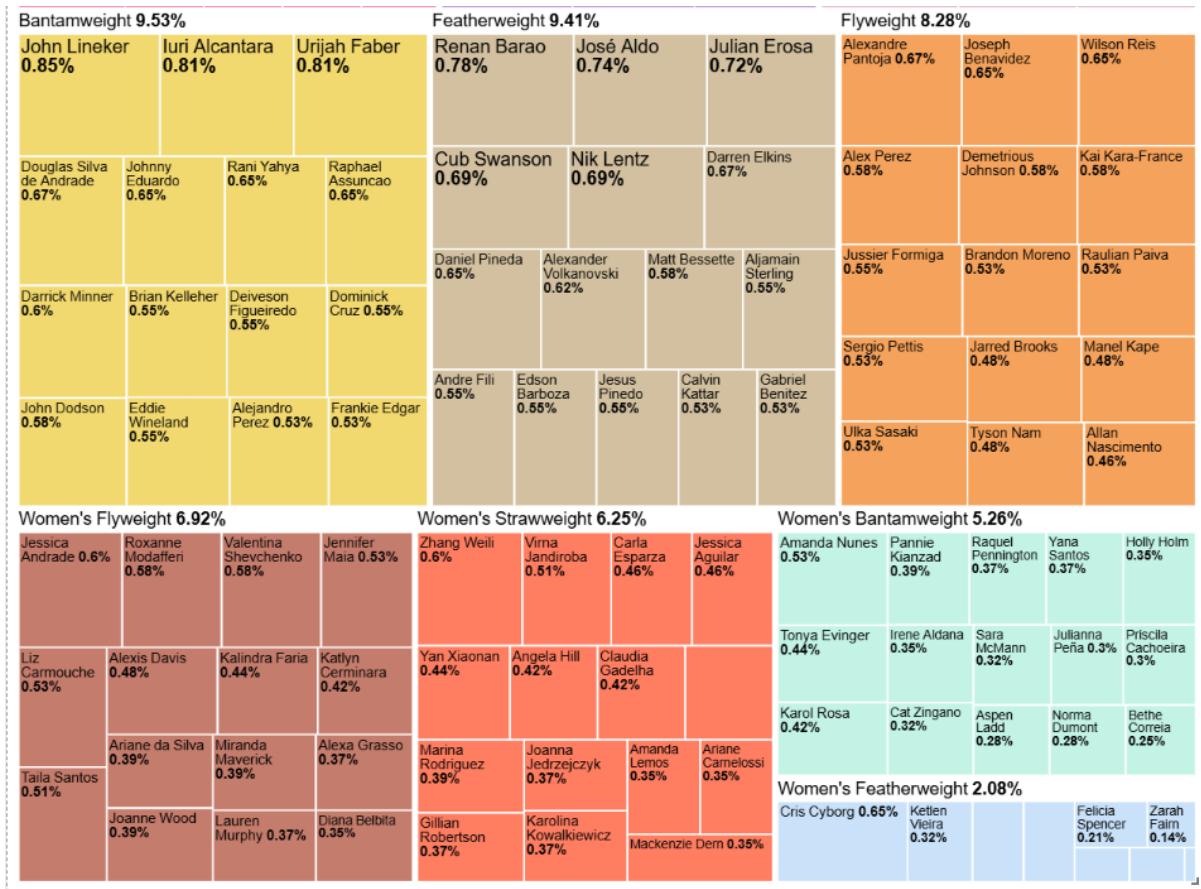
Distribution des types de défaites pour chaque Catégorie de poids



Meilleurs combattants UFC selon le nombre total de victoires et répartition par catégorie de poids (1/2)



(2/2)



6. Conclusion

Ce projet a permis de mettre en œuvre une architecture complète de traitement et de visualisation de données en temps réel, en exploitant les technologies Apache Kafka, Kafka Connect et Elasticsearch, tout en s'appuyant sur Kibana pour une exploration visuelle efficace.

Grâce à l'intégration de ces outils, nous avons pu :

- Collecter, transmettre et indexer dynamiquement des données complexes liées aux combattants de l'UFC.
- Concevoir des visualisations pertinentes et interactives mettant en évidence les tendances clés, comme les méthodes de victoire par catégorie de poids, les ratios victoire/défaite, ou encore les combattants invaincus.
- Proposer une solution extensible, facilement adaptable à d'autres types de données sportives ou à d'autres cas d'usage.

Cette approche démontre la puissance des systèmes distribués de streaming de données combinés à des moteurs de recherche performants pour générer des insights clairs à partir de données volumineuses et variées. Elle peut servir de base à des systèmes de recommandation, de prédiction ou de monitoring avancé dans des contextes variés allant du sport à l'industrie.