MSE | MASTER OF SCIENCE IN ENGINEERING

Big data analysis project
Football analysis

Groupe D -
Daniel Ribeiro Cabral
Killian Ruffieux
Ruben Terceiro

Yverdon, le 7 juin 2024

Dataset

- Données sur les matchs de football
- 12 GB
- Grande quantité de caractéristiques :
  - Compétitions
  - Résultats matchs
  - Lineups
  - Événements dans les matchs
  - 360: champ de vision de la caméra

https://github.com/statsbomb/open-data

Structure du projet & Technologies

- EDA : Analyses statistiques et visuelles
  - Python, Jupyter Notebook, PySpark & Plotly
- Model 1 : Prédiction du vainqueur d’un match
  - Scala, Zeppelin, Spark & ML Spark
- Model 2 : Prédiction des Events
  - Scala, Zeppelin, Spark & ML Spark

EDA

Exploratory Data Analysis : Technologies (1)

	Avantage	Désavantage
Python	**Syntaxe simple**	Pas compatible multithreading
Scala	Performant pour traitement de données parallèles et distribuées	Syntaxe complexe
Jupyter Notebook	**Compatible avec toutes les librairies Python** (notamment visualisation comme **Plotly**, **Matplotlib**,etc..)	Peut-être lent avec gros datasets et compatible avec seulement Python et R
Zeppelin	Compatible plusieurs langages (Scala, SQL,..)	Moins populaire donc moins de lib
PySpark	**Rapide pour grande quantité de données**	Courbe d'apprentissage
Pandas	Facile à prendre en mains	Lent avec grandes quantité de données

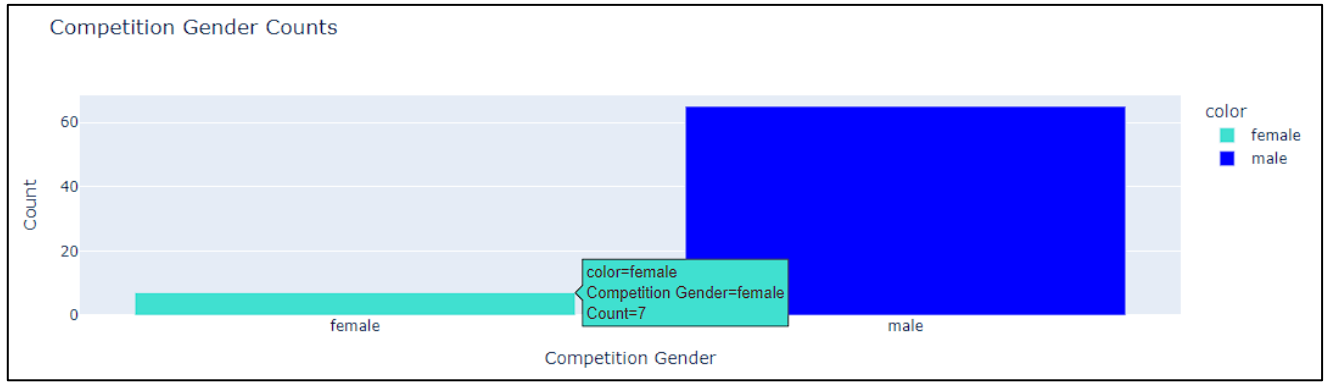
Exploratory Data Analysis : Competitions (2)

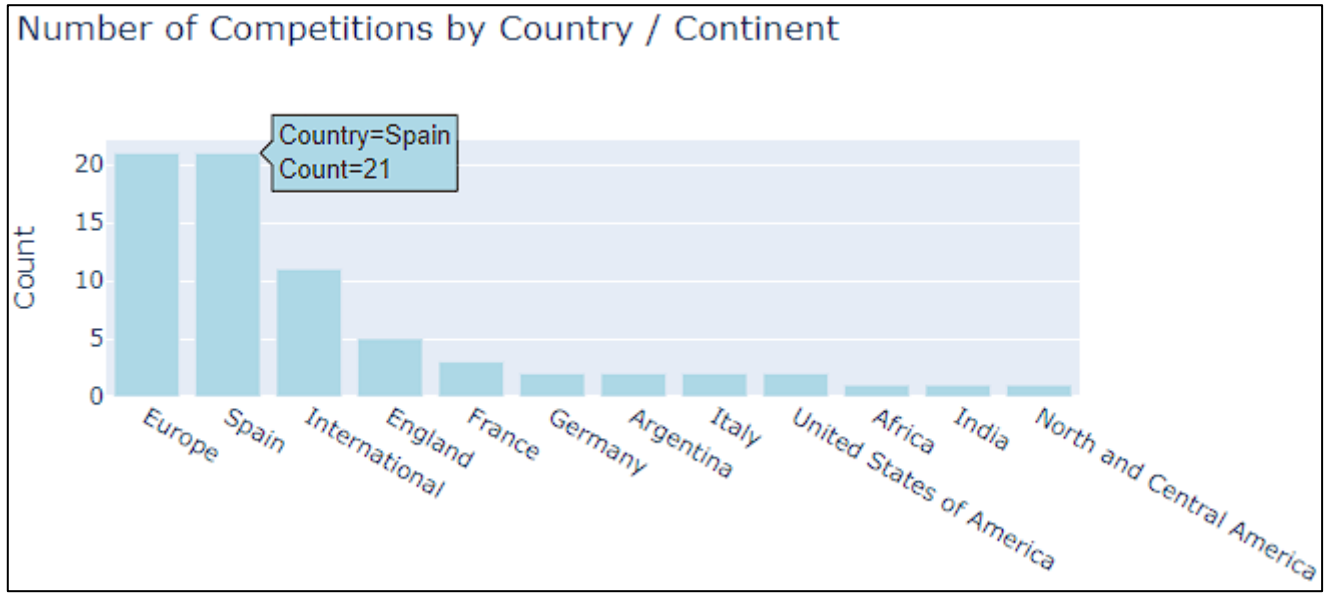
1 fichier json

31,5 Ko

Level 0 : 12 features

Compétitions,
Saison, Dates MAJs





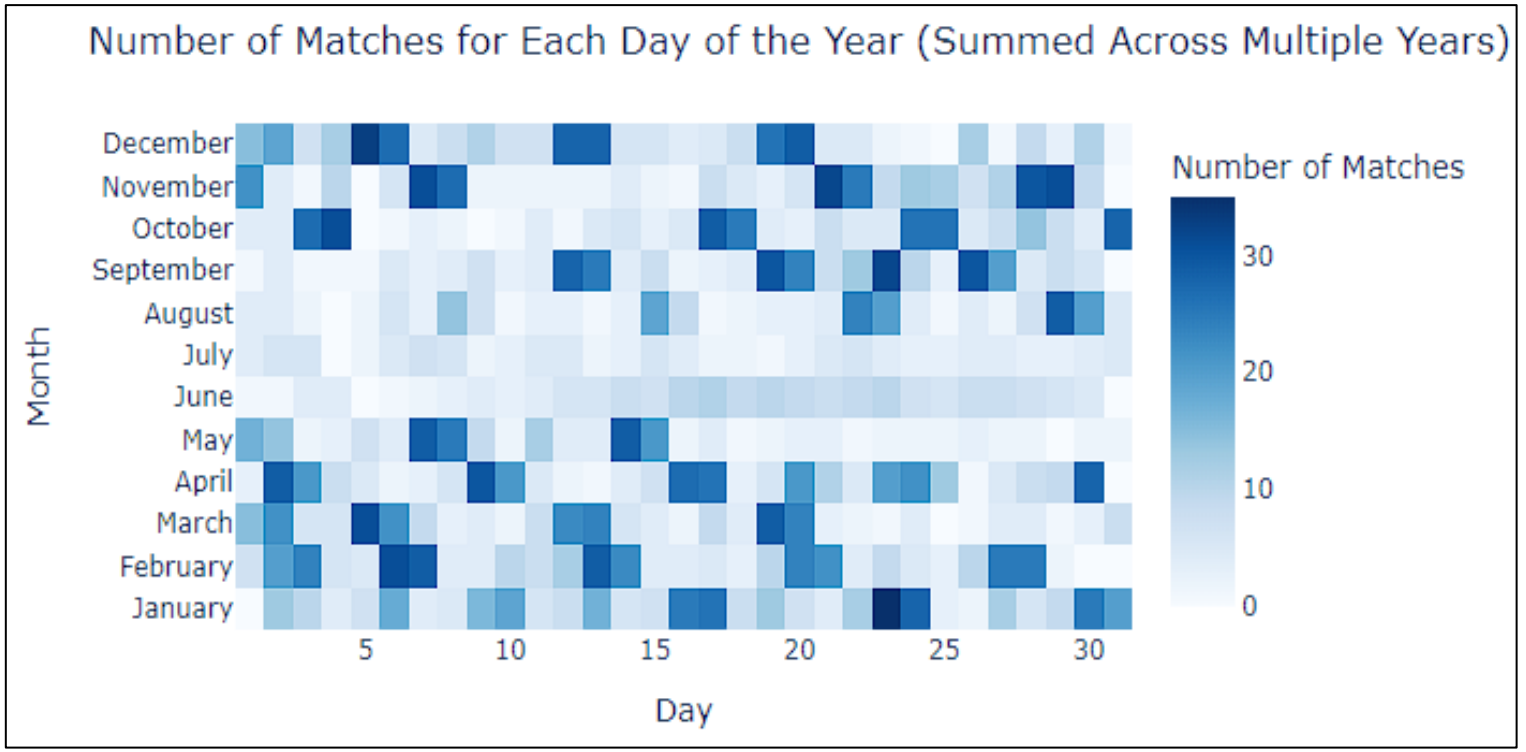
Exploratory Data Analysis : Matches (3)

72 fichiers json

6,23 Mo

Level 0 : 4 features
Level 1 :18 features
Level 2: 28 features
Level 3 : 18 features

Dates, Scores, Équipes, Compétition



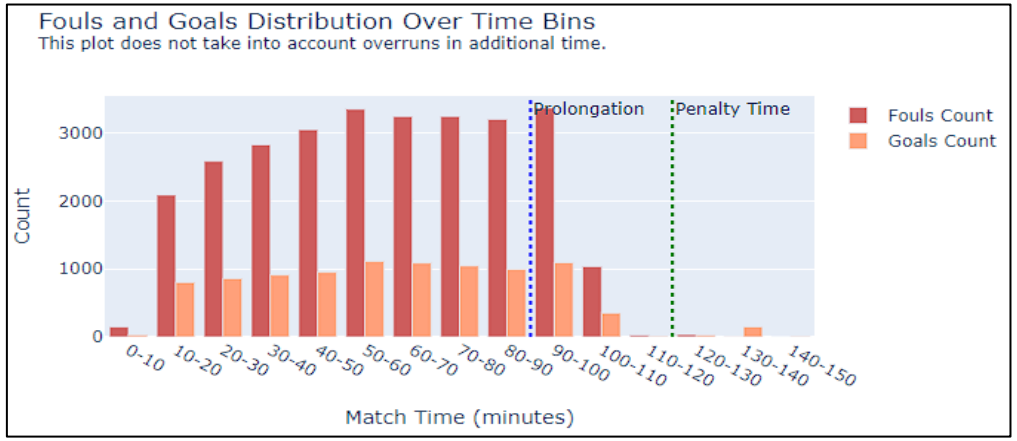
Exploratory Data Analysis : Events (4)

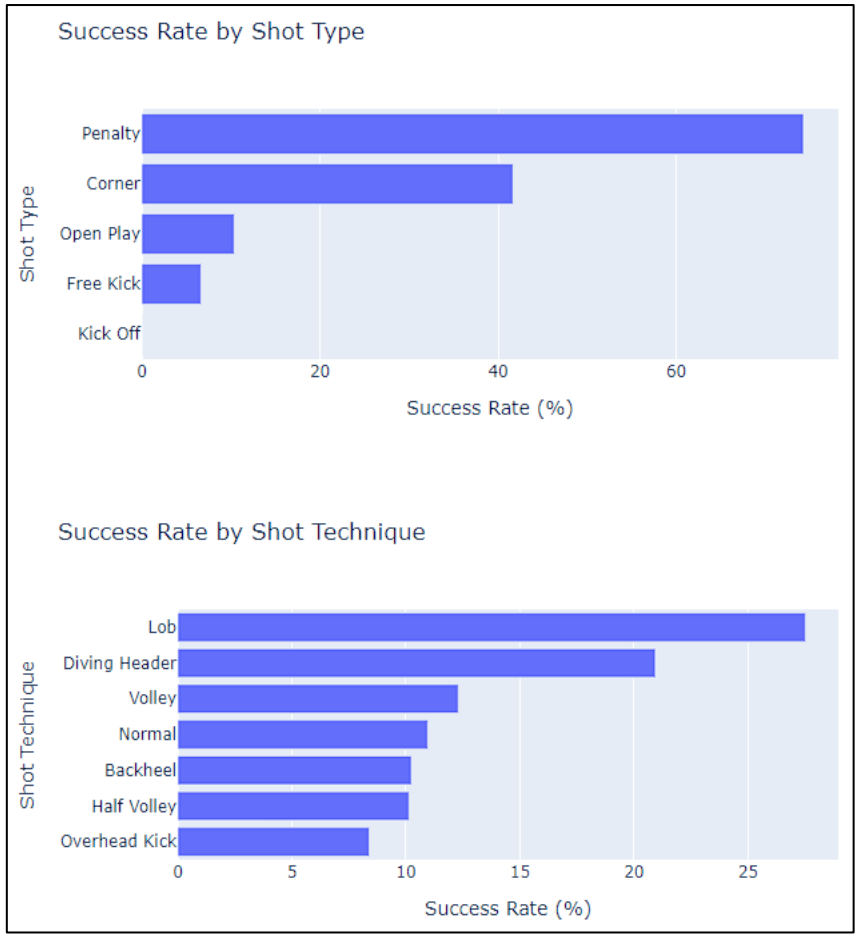
3350 fichiers json

9.92 GB

Événements, Actions, Joueurs

Level 0 : 2 features
Level 1 : 44 features
Level 2 : 108 features
Level 3 : 61 features
Level 4 : 8 features





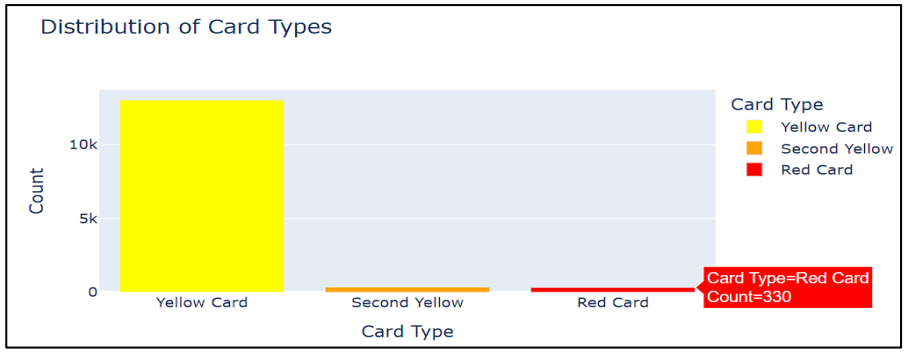
Exploratory Data Analysis : Lineups (5)

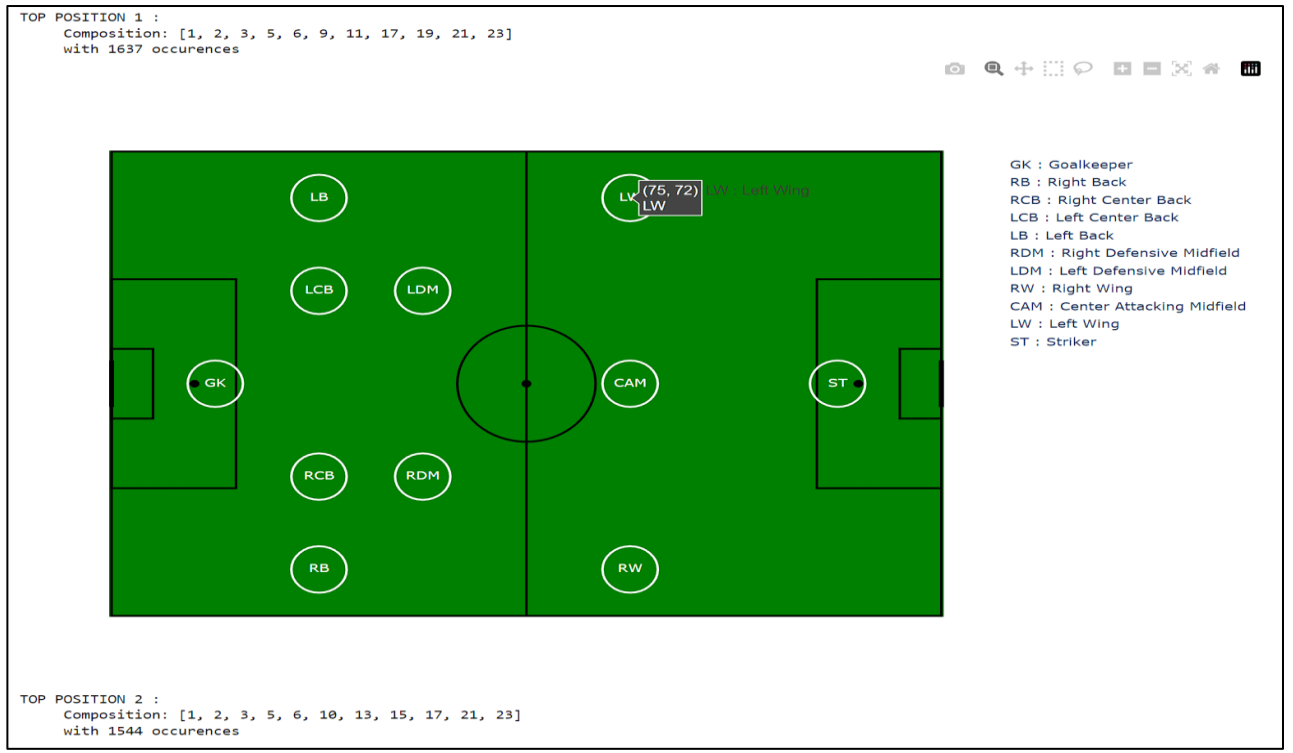
3350 fichiers json

70,0 Mo

Level 0 : 2 features
Level 1 : 5 features
Level 2 : 7 features
Level 3 : 14 features

Alignements, Joueurs, Équipes





https://fcpython.com/visualisation/drawing-pitchmap-adding-lines-circles-matplotlib

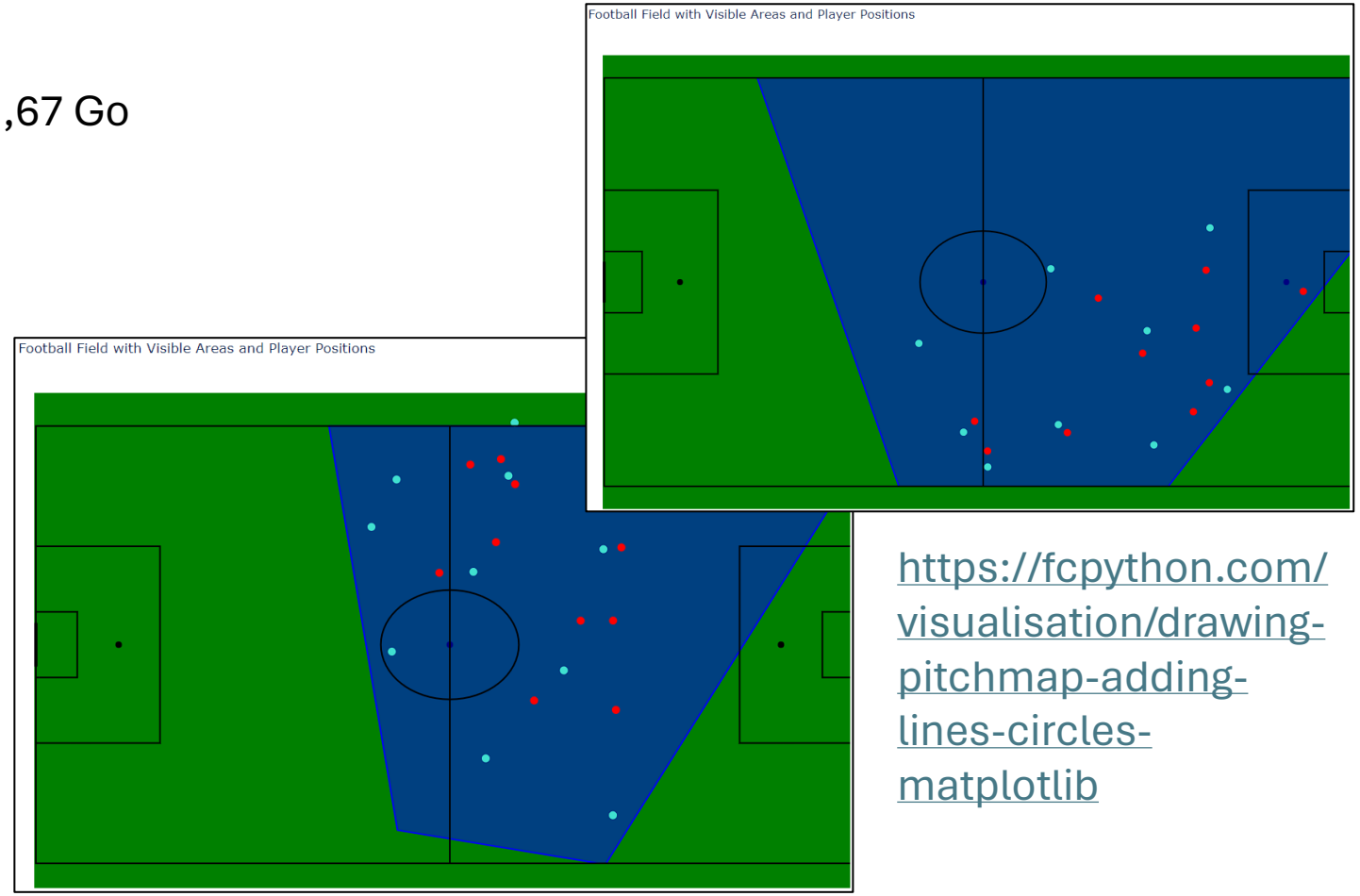
Exploratory Data Analysis : Three-Sixty (6)

244 fichiers
json

1,67 Go

Level 0 : 4 features
Level 1 : 4 features

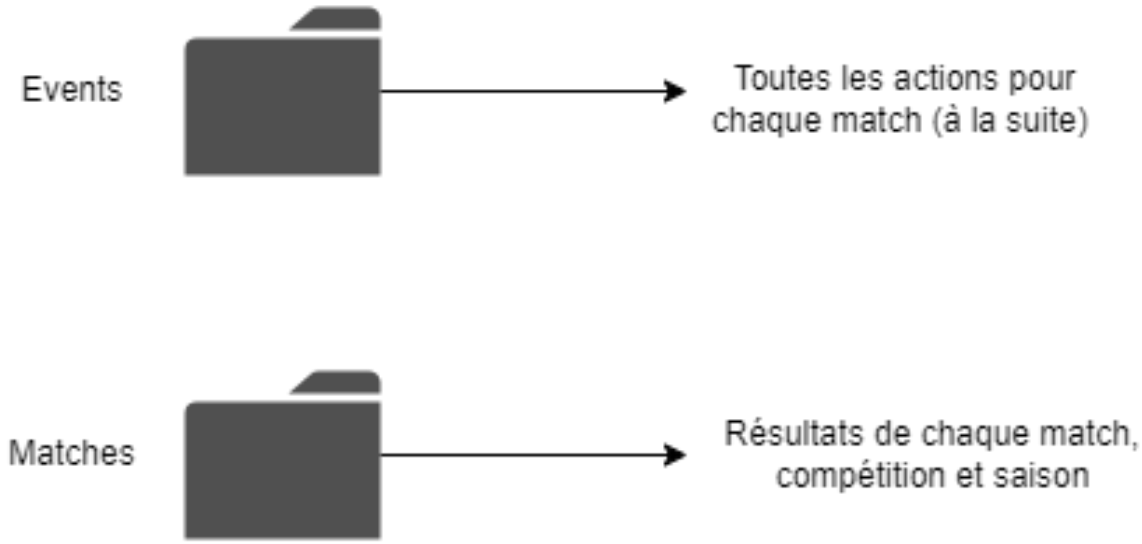
Visuel, Alignements,
Lineups, Localisation



https://fcpython.com/visualisation/drawing-pitchmap-adding-lines-circles-matplotlib

Model 1 : Prédiction vainqueur d’un match

Données utilisées



Données utilisées

printschema()

```
|  |  |  |  |  |-- id: long (nullable = true)
|  |  |  |  |  |-- name: string (nullable = true)
|  |  |  |  |-- position: struct (nullable = true)
|  |  |  |  |  |-- id: long (nullable = true)
|  |  |  |  |  |-- name: string (nullable = true)
|  |-- team: struct (nullable = true)
|  |  |-- id: long (nullable = true)
|  |  |-- name: string (nullable = true)
|  |-- timestamp: string (nullable = true)
|  |-- type: struct (nullable = true)
|  |  |-- id: long (nullable = true)
|  |  |-- name: string (nullable = true)
|  |-- under_pressure: boolean (nullable = true)
|  |-- file_name: string (nullable = false)
|  |-- match_id: string (nullable = false)
|-- match_id: string (nullable = false)
```

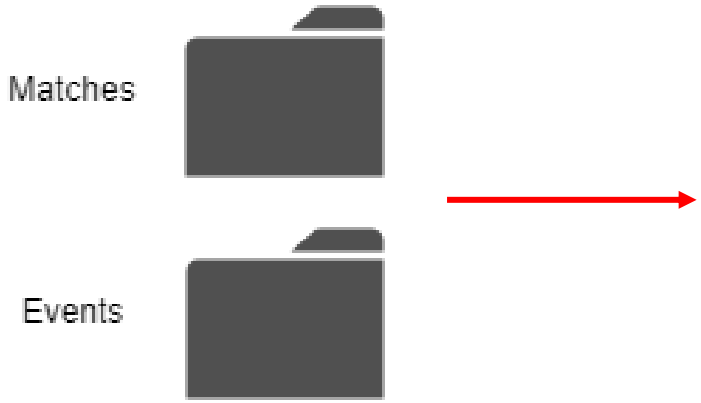
- Chaque object correspond à une action, spécifier par un type, dans un match
- On a notre disposition toutes les informations des matchs (dont nombres de passes, tirs, carton jaunes etc...)

Données utilisées


```
{
  "id" : "549567bd-36de-4ac8-b8dc-6b5d3f1e4be8",
  "index" : 5,
  "period" : 1,
  "timestamp" : "00:00:00.575",
  "minute" : 0,
  "second" : 0,
  "type" : {
    "id" : 30,
    "name" : "Pass"
  },
  "possession" : 2,
  "possession_team" : {
    "id" : 206,
    "name" : "Deportivo Alavés"
  },
  "play_pattern" : {
    "id" : 9,
    "name" : "From Kick Off"
  },
  "team" : {
    "id" : 206,
    "name" : "Deportivo Alavés"
  },
  "player" : {
    "id" : 6581,
    "name" : "Jonathan Rodríguez Menéndez"
  },
  "position" : {
    "id" : 16,
    "name" : "Left Midfield"
```


Exemple ici d’action durant un match à un certain moment

Données utilisées

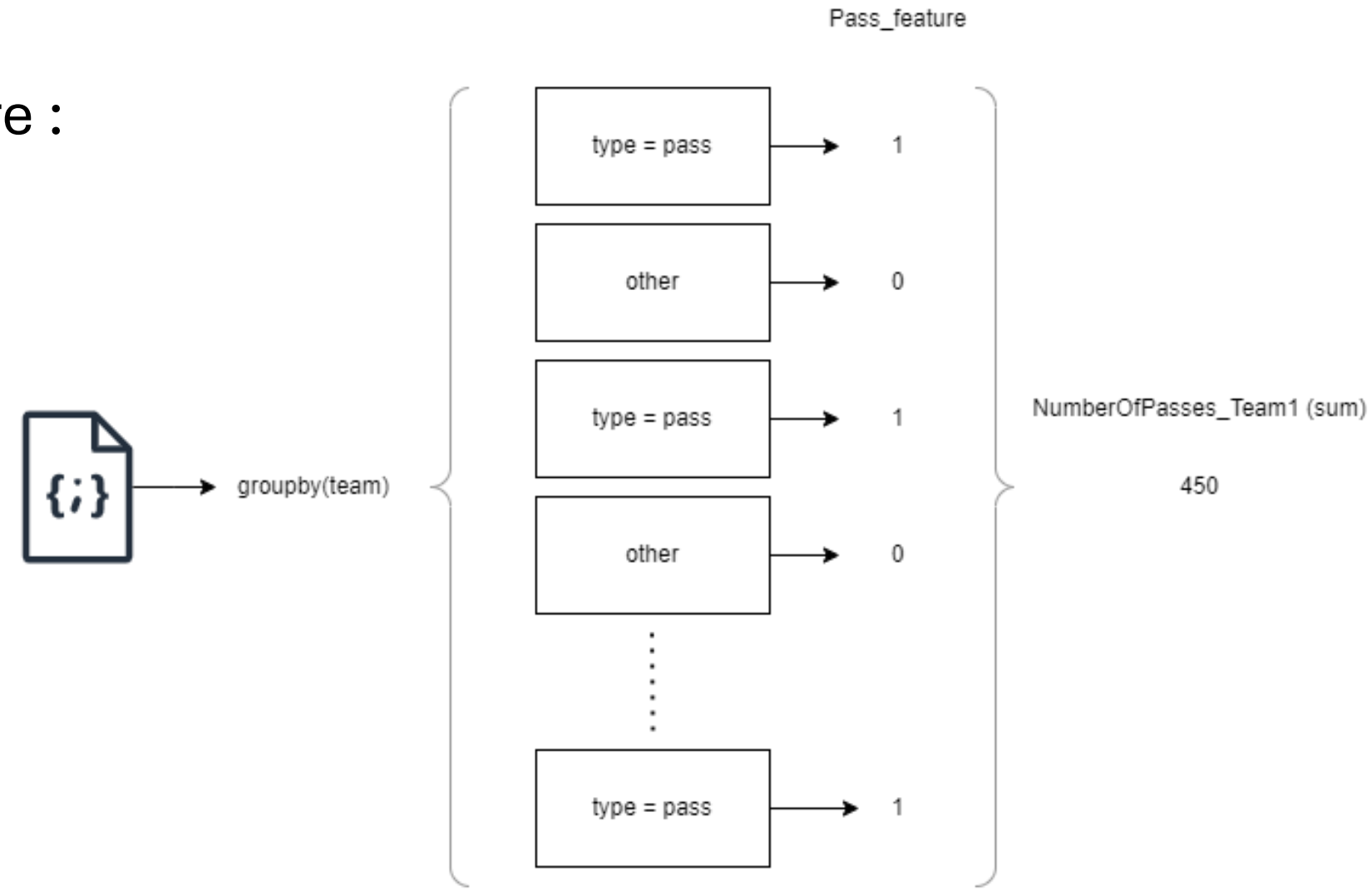


Définition de 16 features
- Certaines sont rassemblés dans une seule (goalkeeper actions)

```
root
 |-- match_id: string (non utilisé)
 |-- match_teams: string (non utilisé)
 |-- Pass_team1: long (utilisé)
 |-- Pass_team2: long (utilisé)
 |-- Shot_team1: long (utilisé)
 |-- Shot_team2: long (utilisé)
 |-- Foul_won_team1: long (utilisé)
 |-- Foul_won_team2: long (utilisé)
 |-- Foul_committed_team1: long (utilisé)
 |-- Foul_committed_team2: long (utilisé)
 |-- Bad_Behaviour_Yellow_Card_team1: long (utilisé)
 |-- Bad_Behaviour_Yellow_Card_team2: long (utilisé)
 |-- total_red_cards_team1: long (utilisé)
 |-- total_red_cards_team2: long (utilisé)
 |-- total_actions_team1: long (utilisé)
 |-- total_actions_team2: long (utilisé)
 |-- match_date: string (utilisé)
 |-- home_score: long (utilisé)
 |-- away_score: long (utilisé)
 |-- home_team_id: long (utilisé)
 |-- team1_results: double (utilisé)
 |-- away_team_id: long (utilisé)
 |-- team2_results: double (utilisé)
 |-- winning_team: string (non utilisé) --> utilisé pour la ground truth (valeur y)
```

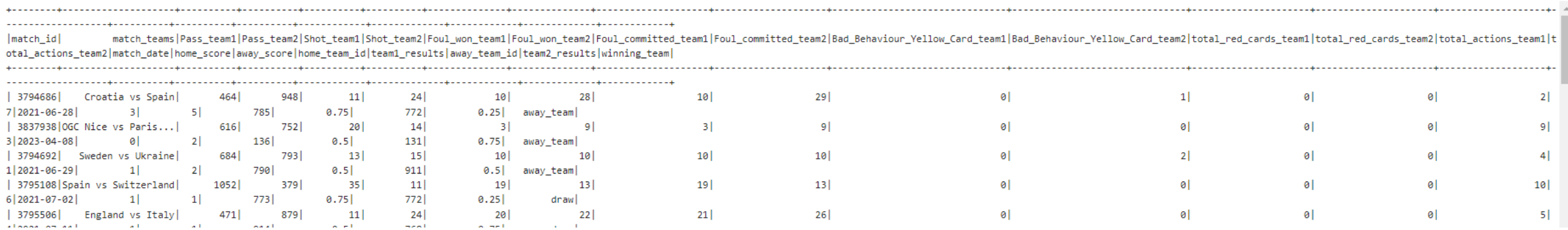
Préparations des données - Spark

- Exemple de feature :
  - Pass



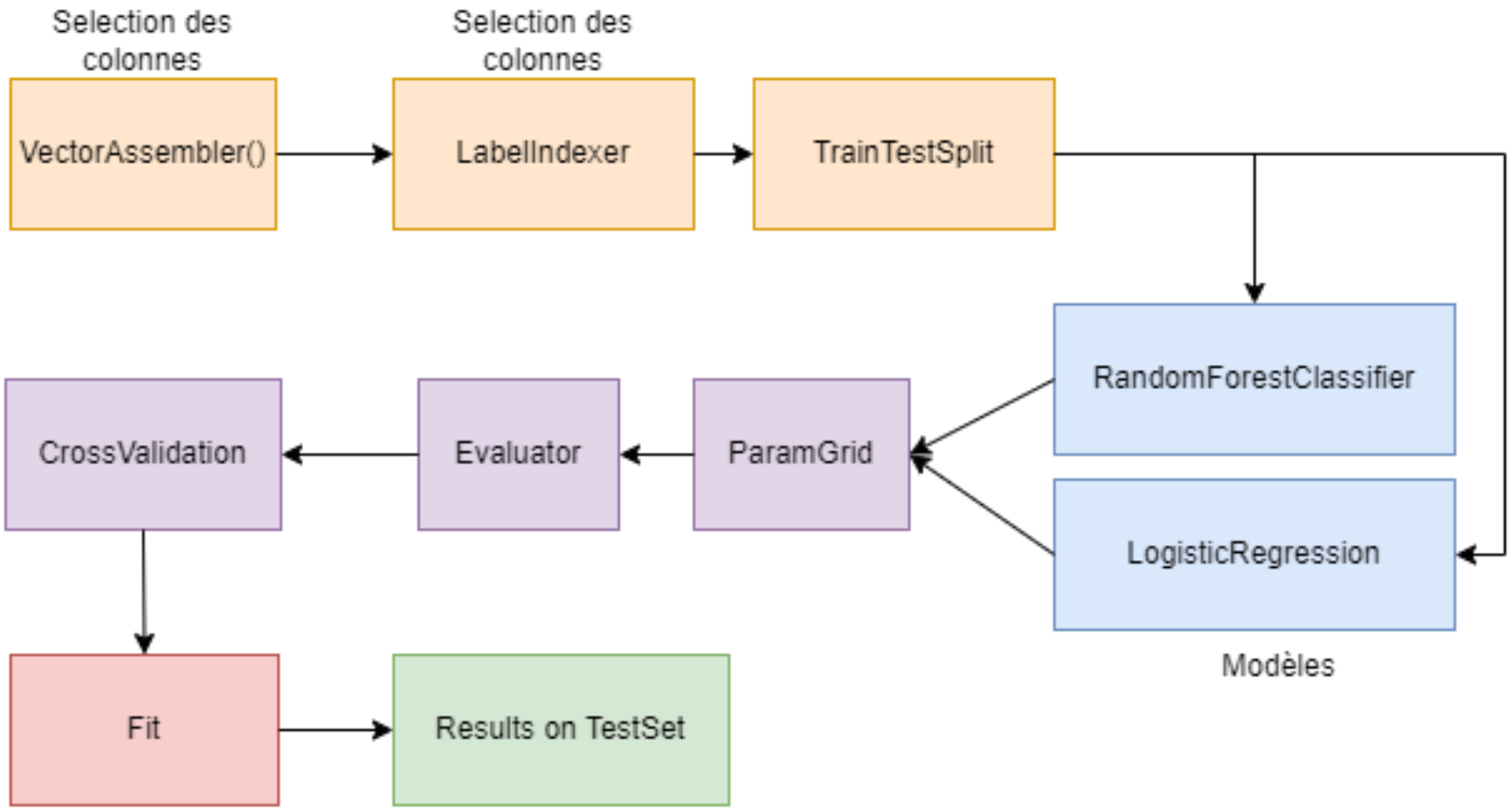
Préparations des données - Spark

- Résultat final :
  - 3350 données



```
+--------+--------------------+----------+----------+----------+----------+-------------+-------------+--------------------+--------------------+----------------------------+----------------------------+--------------------+--------------------+--------------------+
|match_id|        match_teams|Pass_team1|Pass_team2|Shot_team1|Shot_team2|Foul_won_team1|Foul_won_team2|Foul_committed_team1|Foul_committed_team2|Bad_Behaviour_Yellow_Card_team1|Bad_Behaviour_Yellow_Card_team2|total_red_cards_team1|total_red_cards_team2|total_actions_team1|total_actions_team2|match_date|home_score|away_score|home_team_id|team1_results|away_team_id|team2_results|winning_team|
+--------+--------------------+----------+----------+----------+----------+-------------+-------------+--------------------+--------------------+----------------------------+----------------------------+--------------------+--------------------+--------------------+
| 3794686|    Croatia vs Spain|       464|       948|        11|        24|           10|           28|                10|                29|                           0|                           1|                   0|                   0|                   2|                   7|2021-06-28|         3|         5|         785|         0.75|         772|         0.25|   away_team|
| 3837938|OGC Nice vs Paris...|       616|       752|        20|        14|            3|            9|                 3|                 9|                           0|                           0|                   0|                   0|                   9|                   3|2023-04-08|         0|         2|         136|          0.5|         131|         0.75|   away_team|
| 3794692|   Sweden vs Ukraine|       684|       793|        13|        15|           10|           10|                10|                10|                           0|                           2|                   0|                   0|                   4|                   1|2021-06-29|         1|         2|         790|          0.5|         911|          0.5|   away_team|
| 3795108|Spain vs Switzerland|      1052|       379|        35|        11|           19|           13|                19|                13|                           0|                           0|                   0|                   0|                  10|                   6|2021-07-02|         1|         1|         773|         0.75|         772|         0.25|        draw|
| 3795506|    England vs Italy|       471|       879|        11|        24|           20|           22|                21|                26|                           0|                           0|                   0|                   0|                   5|
```

Préparation du modèle - Pipeline



Résultats obtenus

- **Durée de l'entrainement** : 17min sur Google Cloud

- **Logistic Regression :**
  - Accuracy : 70.4 %

- **Random Forest :**
  - Accuracy : 68.24 %

```
//rfPredictions.select("match_id", "winning_team", "prediction").s

Logistic Regression Test set accuracy = 0.703862660944206
Random Forest Test set accuracy = 0.6824034334763949
+--------+------------+----------+
|match_id|winning_team|prediction|
+--------+------------+----------+
|   15978|   away_team|       1.0|
|   16205|   home_team|       0.0|
|   18241|   home_team|       0.0|
|   18242|   away_team|       1.0|
|   19739|   away_team|       1.0|
|   19742|   away_team|       0.0|
|   19746|   away_team|       0.0|
|   19749|   home_team|       0.0|
|   19758|   away_team|       0.0|
|   19760|   home_team|       0.0|
+--------+------------+----------+
only showing top 10 rows
```

Model 2 : Prédiction des types d’events

But

- Prédire la prochaine action
- 32 actions différentes

Par exemple:

- [Pass, Duel, Foul, Shot, Pass] --> [Pass]
- [Duel, Foul, Shot, Pass, Pass] --> [Pass]
- [Foul, Shot, Pass, Pass, Pass] --> [Shot]

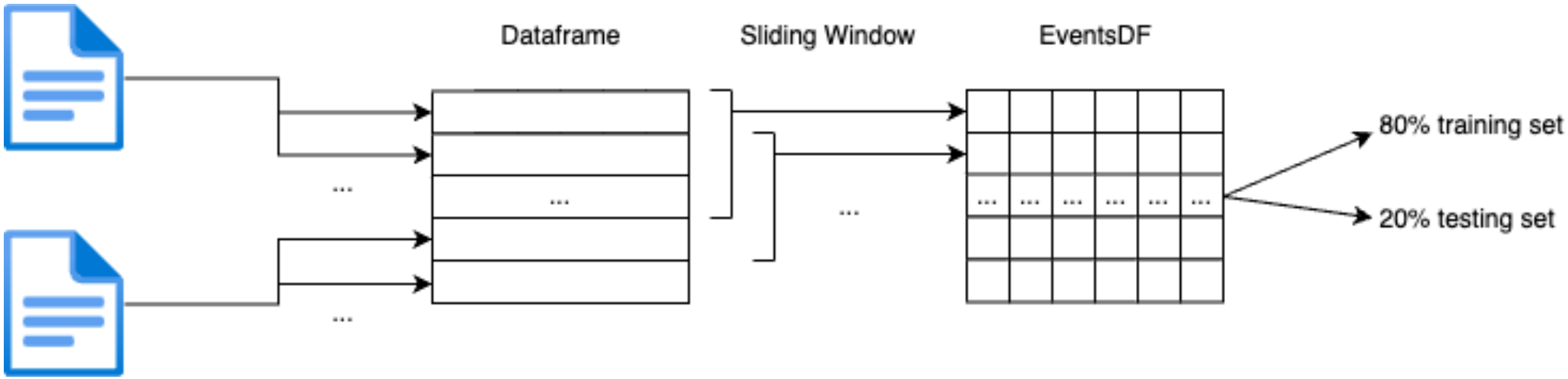
Données utilisées


```
{
  "id" : "549567bd-36de-4ac8-b8dc-6b5d3f1e4be8",
  "index" : 5,
  "period" : 1,
  "timestamp" : "00:00:00.575",
  "minute" : 0,
  "second" : 0,
  "type" : {
    "id" : 30,
    "name" : "Pass"
  },
  "possession" : 2,
  "possession_team" : {
    "id" : 206,
    "name" : "Deportivo Alavés"
  },
  "play_pattern" : {
    "id" : 9,
    "name" : "From Kick Off"
  },
  "team" : {
    "id" : 206,
    "name" : "Deportivo Alavés"
  },
  "player" : {
    "id" : 6581,
    "name" : "Jonathan Rodríguez Menéndez"
  },
  "position" : {
    "id" : 16,
    "name" : "Left Midfield"
```


Exemple ici d’action durant un match à un certain moment

Procédé – Version 1

- Version naive
- Considère tous les events comme une suite
- Sépare aléatoirement en train / test



Taille des données

Dataframe des events:

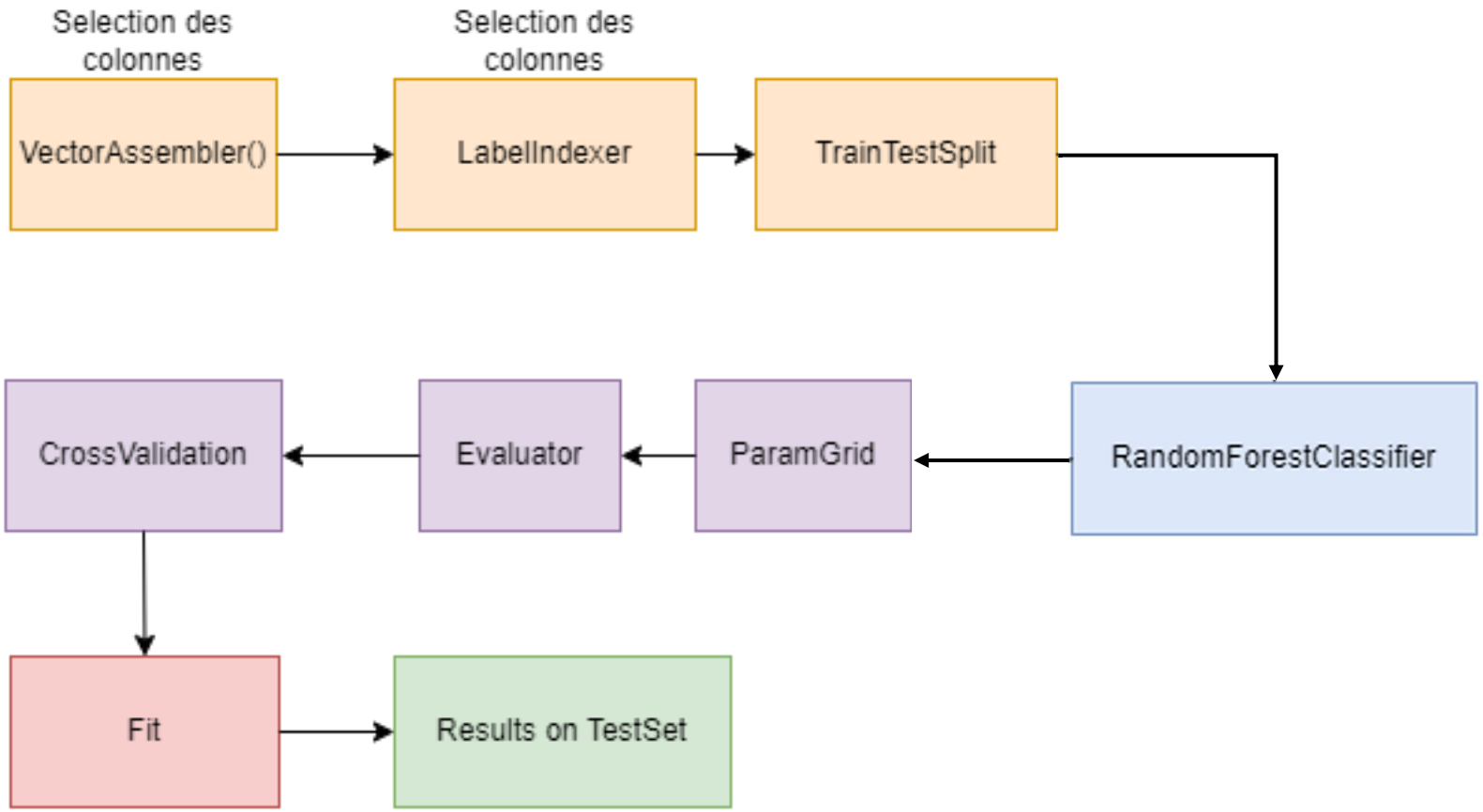
- 11’491’082 entrées
- 1.7Gb

=> Nécessaire d’appliquer des techniques de Big Data

```
// Repartition the DataFrame to distribute data across the cluster
val repartitionedDF = df.repartition(200)
```

```
// Cache the DataFrame to avoid recomputation
repartitionedDF.cache()
```

Préparation du modèle - Pipeline



Résultat – Version 1

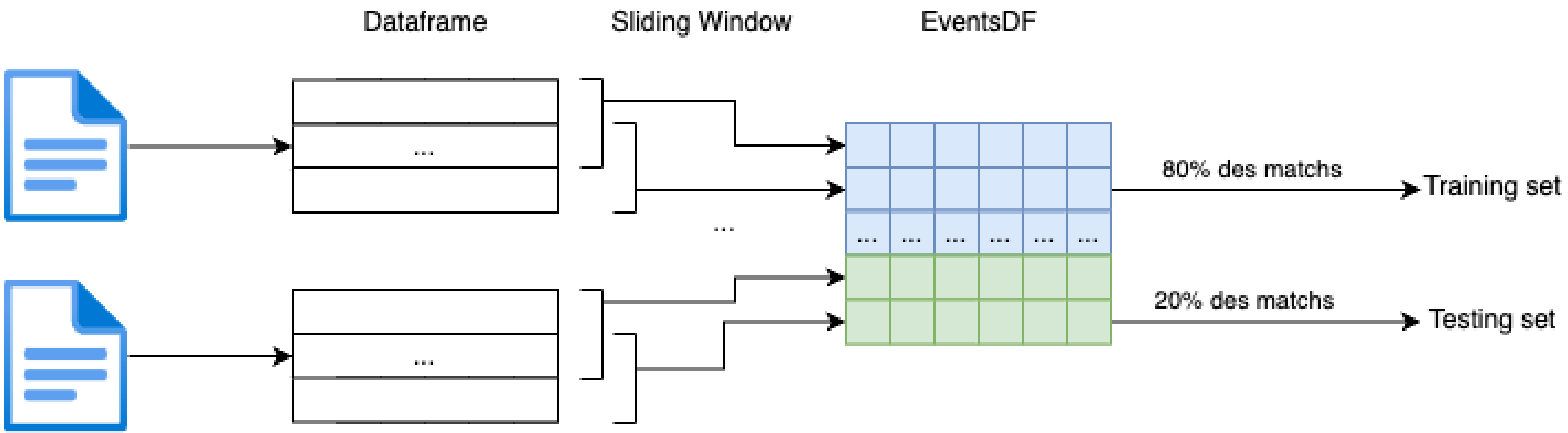
- 72.7%
- Temps d’entrainement: 2h

- Mais comment interpréter ?

=> Version 2

Procédé – Version 2

- Traitement comme des vrais Time Series
- Considère les matchs séparément
- Test ne se retrouve pas dans le train



Résultat – Version 2

- 3%...
- Temps d’entrainement: 2h

Mais, notre modèle a prédit :

- Passe au lieu de "Block"
- Passe au lieu de "Ball Recovery"
- Duel au lieu de Dribble
- Passe au lieu de "Dispossessed"
- "Goal Keeper" au lieu de "Clearance"

Conclusion et futures améliorations

- Agrandir le dataset avec d’autres matchs
  - Soit payant
  - Soit compliqué (scraping)

- Modèle 1:
  - Test d’autres modèles différents
  - Sélection des features plus importantes et création d’autres
- Modèle 2:
  - Difficile de prédire les events
  - Test en utilisant qu’une seule équipe

Questions ?

Merci de votre attention