



DATA SCIENTEST

# MODÈLE DE PRÉDICTION DES RECORDS D'ATHLÉTISME

JEUX OLYMPIQUES

2024



Auteurs :

Armen ASATRYAN

Yannick OREAL

Yves-Marie SALAUN

Xavier GERVAIS

Managers :

Jérémy ROBERT

Raja AARAB

## RAPPORT FINAL

Projet réalisé dans le cadre de la formation  
DATA SCIENTIST BOOTCAMP d'avril 2024

ATHLÉTISME DU 01 AU 11 AOÛT 2024



# Sommaire

Le Projet en rétrospective	<u>01</u>
Les données	<u>02</u>
Hypothèses de travail	<u>03</u>
Preprocessing	<u>04</u>
Visualisations	<u>05</u>
Qualité des données	<u>07</u>
Modélisations Baseline	<u>08</u>
Classifications	<u>09</u>
Régressions	<u>10</u>
Keras & bilan préliminaire	<u>12</u>
Enrichissements	<u>13</u>
Target & Conclusion	<u>18</u>
Annexe 1 : Rapport de Preprocessing	
Annexe 2 : Rapport de Visualisation	
Annexe 3 : Rapport de Modélisation	

## Le GitHub des sources utilisées

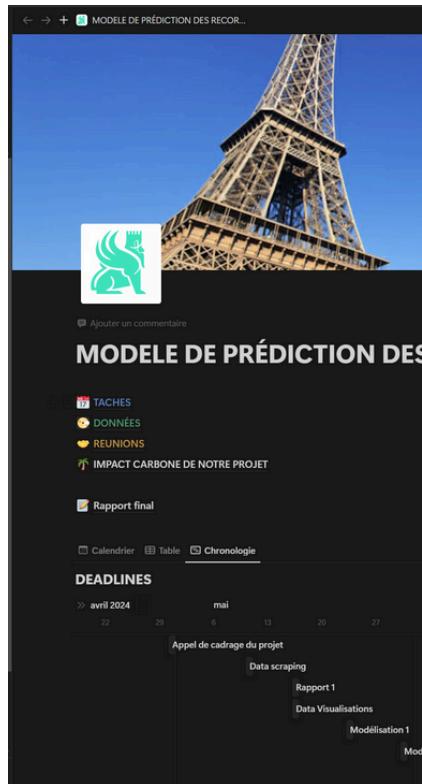
[GitHub sources Permalink](#)



# avr24\_bds\_olympic\_games

# Le projet en rétrospective

## Introduction



## Un peu de contexte

Le choix d'un projet prédictif lié aux JO s'est imposé à nous par l'arrivée des jeux cet été 2024 à Paris. C'était l'opportunité parfaite pour y appliquer des modèles de Machine Learning.

Nous avons vite restreint le sujet à un objectif atteignable: le domaine choisi étant l'athlétisme individuel pour éviter une complexité supplémentaire à un projet déjà ambitieux sur le plan prédictif.

La cible : **prédir les records battus lors de ces épreuves.**

La réunion d'une équipe, la mise en place d'un Notion pour s'organiser, d'un github pour les données et le projet avait officiellement débuté.

Il restait à trouver des données exploitables, ce qui fut finalement la clé de tout le projet.

### Matrice des attributions de tâches

Qui ?	Aa fait quoi ?	sur quoi ?	Pour quand ?	statut	URL
Yannick	Scraping	Wikipedia	3 mai 2024	Fini	
Armen	Conditions météo	fichier	30 avril 2024	Fini	
Xavier	fini 105 pour méthodologie	Qualité	23 avril 2024	Fini	
Xavier	Repo projet	Github	22 avril 2024	Fini	github.com/gex...t-axyx
Armen	Initiation Rapport Projet OLYMPIC GAMES	Word document	5 mai 2024	Fini	notion.so/DON...e48956
Armen	Rapport exploration données - meteo	Excel (template d'exploration)	5 mai 2024	Fini	notion.so/Rap...0fe99a
Yves-Marie	Scraping	World Athletics	3 mai 2024	En cours	
Yves-Marie	Scraping	Lequipe.fr	3 mai 2024	En cours	
Xavier	Récupération data	Autres fichiers Kaggle	3 mai 2024	En cours	
Xavier	Récupération	Statista	3 mai 2024	En cours	

### Scraping Wikipedia + Matching Kaggle

```
Par Année

dataset_wiki_updated_10052024.zip (4.6 MB)
dataset_kaggle_updated_10052024.zip (7.0 MB)

# Importer les bibliothèques
import pandas as pd
import numpy as np

# Importer les données utilisées
df_wiki_dtypes = pd.read_csv("og_events_distance.csv")
df_wiki_timegap = pd.read_csv("og_events_time.csv")
df_kaggle = pd.read_csv("athlete_events_athletics.csv")

# Clean / modify dataset kaggle
df_kaggle['Sport'].unique()
df_kaggle[df_kaggle['Sport']=='Athletics']
df_kaggle.drop(['Name', 'Season', 'Sport', 'Team', 'City'], axis=1)
```

# avr24\_bds\_olympic\_games

# Le projet en rétrospective

## Les données

The screenshot displays a complex web environment for data analysis. On the left, a 'Calendar / Results' interface shows a list of competitions with columns for Date, Name, Venue, Country, Category, Discipline, and Competition Group. One entry is for the 'Belgian 10.000m Championships' on May 8th. Below this is a 'Data Explorer' for 'athletes\_events.csv' containing 134732 unique values across various columns like Name, Sex, Age, Height, and Weight. To the right, a developer's browser window shows the network tab with numerous requests to 'worldathletics.org' for various resources like CSS files and JSON APIs. Other tabs include 'WorldAthletics', 'Wikipedia', 'Kaggle', and a direct view of the 'athletes\_events.csv' file itself. A status bar at the bottom indicates the project is in 'main-4434c'.

Les sources ne manquaient à priori pas :

- Kaggle : 120 years of Olympics
- wikipedia : page des records
- statista
- lequipe
- worldathletics : toutes les compétitions.

Et avec elles les problématiques sur ces données :

- très disparates.
- qui se recoupaient mal.
- avec des doutes sur les corrélations des variables disponibles avec notre cible.

Elles se sont avérées extrêmement déséquilibrées, en cohérence et en répartition dans le temps.

## Evènements

15 400

Des compétitions mondiales, JO et Mondiaux, jusqu'aux compétitions nationales.

sur WorldAthletics

# Hypothèses de travail

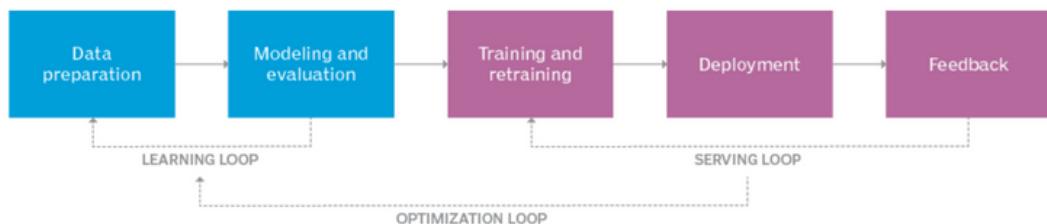
## Approches du projet

Au cours du projet, plusieurs approches ont été envisagées afin de prédire si des records pouvaient être battus lors des prochains jeux olympiques.

**La première approche** : l'objectif étant de déterminer si un record pouvait ou non être battu, les modèles de classification (avec une classe 1 ou 0 si un record était battu ou non) ont été explorés.

**La seconde approche** : nous avons considéré que prédire la performance brute d'un athlète permettrait à postériori de prédire s'il battrait ou non un record également. Avec la probabilité associée.

Cette prédiction de performance a été testée à partir de séries temporelles et à partir de modèles de régression linéaire.



## Stratégie adoptée

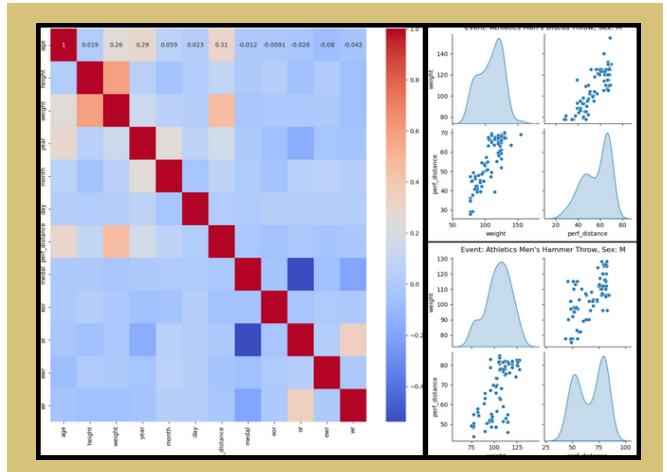
La stratégie retenue sur les variables tout au long du projet : se focaliser sur le dataset de World Athletics et l'enrichir au mieux.

- ajout de la taille et du poids de l'athlète (et extrapolation des NaN par moyenne en fonction du type d'épreuve)
- ajout de données météo
- ajout du record olympique en vigueur lors de l'épreuve et de la target "or\_flag" pour la classification
- calcul de l'âge à partir de la date de naissance et la date de la course.
- calcul du nombre de performances , de la moyenne des performances, et de la meilleure performance sur les 365 jours qui précèdent.

# avr24\_bds\_olympic\_games Dataset

## Preprocessing et Visualisations

WA_dist_enc_11062024.csv								
Origine du fichier	Délimiteur	Détection du type de données						
1252: Europe de l'Ouest (Windows)	Virgule	Selon les 200 premières lignes						
date	competition_name_anc	event_anc	rtype	name_anc	nationality_anc	gender	age	height
12/01/2006	4071.0	10.0	1.0	38330.0	142.0	0.0	24.032854209445585	178.0
18/09/2004	4044.0	10.0	1.0	38330.0	142.0	0.0	22.55415178493425	178.0
02/09/2007	2774.0	10.0	1.0	38330.0	142.0	0.0	25.508555783709788	178.0
31/08/2007	2774.0	10.0	6.0	38330.0	142.0	0.0	25.5038008213552	178.0
07/03/2004	1238.0	10.0	1.0	38330.0	142.0	0.0	22.02053388093049	178.0
06/03/2004	1238.0	10.0	6.0	38330.0	142.0	0.0	22.017963011636	178.0
17/09/2006	864.0	10.0	1.0	38330.0	142.0	0.0	24.550308008213552	178.0
28/08/2004	5461.0	10.0	1.0	38330.0	142.0	0.0	22.4969199178644	178.0
26/08/2004	5461.0	10.0	6.0	38330.0	142.0	0.0	22.4914442169202	178.0
22/09/2007	5286.0	10.0	1.0	38330.0	142.0	0.0	25.56313279945243	178.0
09/09/2006	5286.0	10.0	1.0	38330.0	142.0	0.0	24.52840520191649	178.0
02/03/2002	2355.0	10.0	1.0	38330.0	142.0	0.0	20.00547570174262	178.0
01/03/2002	2355.0	10.0	6.0	38330.0	142.0	0.0	20.00273785078713	178.0
11/08/2006	2615.0	10.0	1.0	38330.0	142.0	0.0	24.44900752908963	178.0
08/08/2006	2615.0	10.0	6.0	38330.0	142.0	0.0	24.44079379672827	178.0
09/03/2009	5853.0	10.0	1.0	38330.0	142.0	0.0	26.02600958477755	178.0
08/03/2008	5853.0	10.0	6.0	38330.0	142.0	0.0	26.023171731690624	178.0
18/09/2004	4044.0	10.0	1.0	13971.0	141.0	0.0	26.064394939761	185.0
13/09/2003	4044.0	10.0	1.0	13971.0	141.0	0.0	25.0485968514716	185.0
28/08/2004	5461.0	10.0	1.0	13971.0	141.0	0.0	26.00684462696783	185.0



Le preprocessing a été une phase complexe et itérative.

Le scrapping de WorldAthletics s'est avéré difficile via API GraphQL (annexe 1) et celui de Wikipedia fastidieux.

WorldAthletics a été notre mirage: une base de données extrêmement complète sur le plan des compétitions recensées (Nationales, juniors, séniors...)

Mais :

- plus de 50% concernent les années 2023 et 2024
- pas de données taille / poids des athlètes

Au final, nous nous sommes concentrés sur les athlètes ayant participés aux JO, tirés de Wikipedia (avec leurs tailles et poids), avons complété ces données avec les compétitions auxquelles ils ont participé sur WorldAthletics.

Après traitement des NaN :

- Les tailles et poids manquants sont remplacés par les moyennes respectives, en prenant en compte le sexe et l'épreuve.
- les vents définis à 0 sans mesures

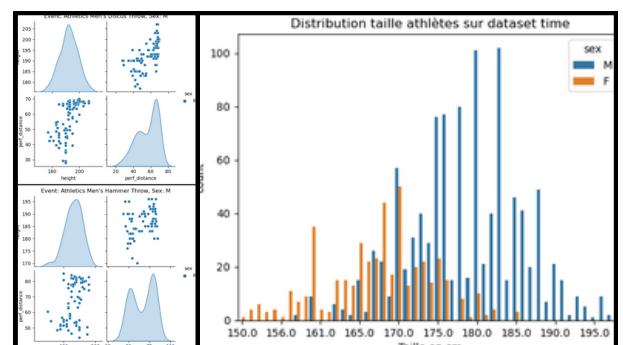
Calcul de l'age = Race\_Date - Birth\_Date

Encodages avec LeaveOutEncoder et LabelEncoder.

Nos visualisations :

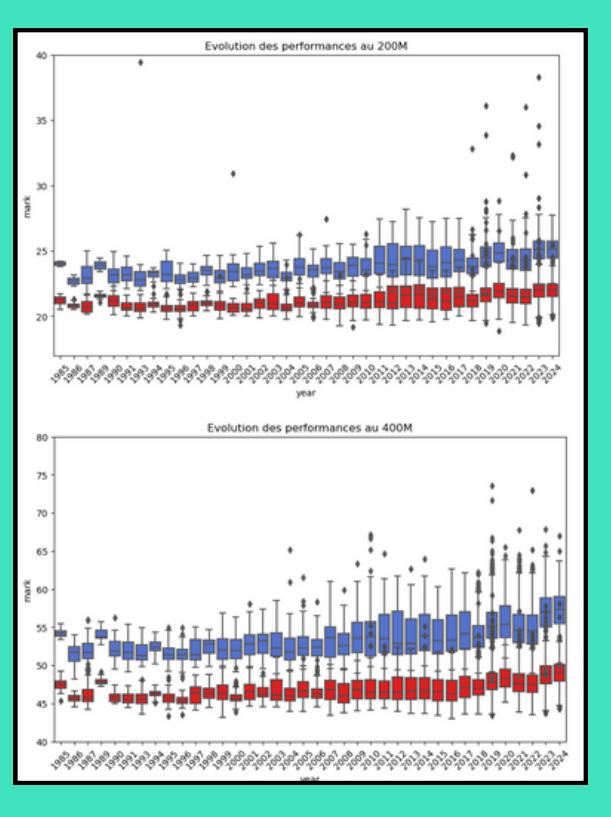
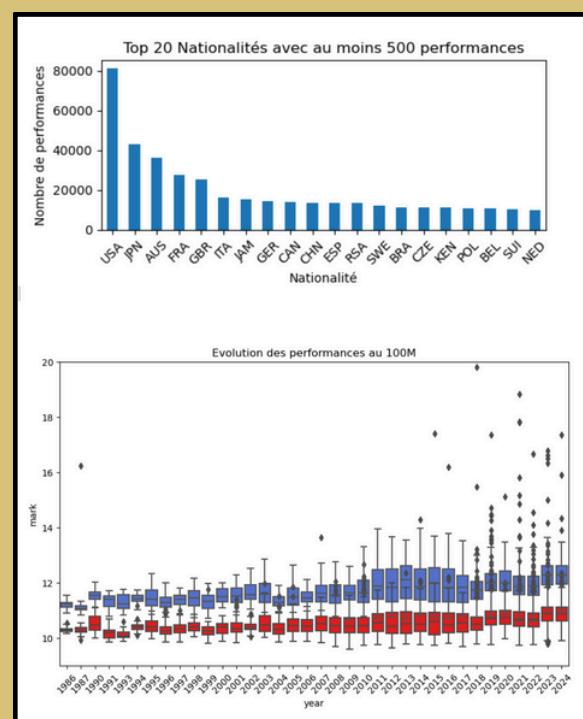
- heatmap
- pairplot
- Distributions

Ces visualisations confirment une corrélation avec la taille, le poids et l'âge, mais les valeurs de corrélations s'avèrent très faibles.



# avr24\_bds\_olympic\_games Dataset TEMPS

## Visualisations

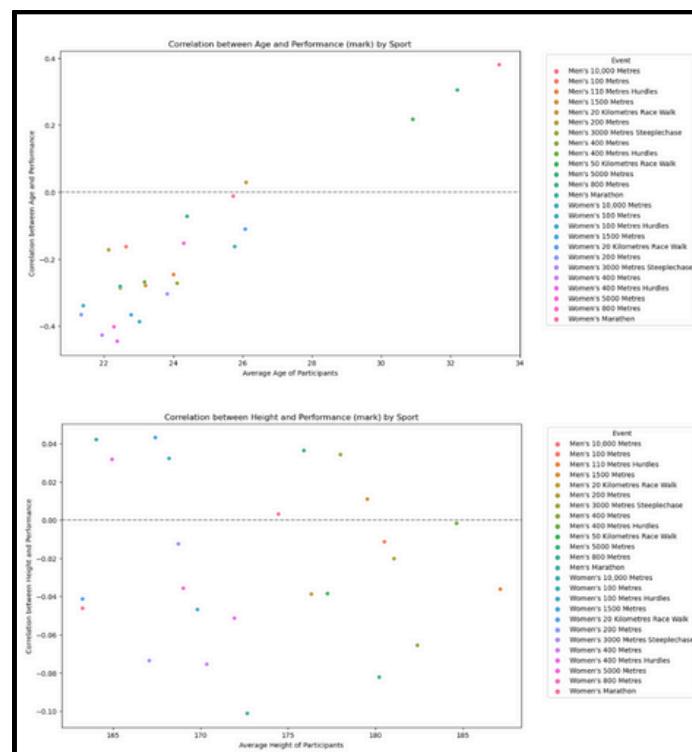


Les athlètes performants sur les épreuves de temps courtes (100M, 200M, 400M, etc.) ont tendance à être jeunes tandis que les athlètes d'endurance (marches, marathon) sont plus âgés.

La **corrélation négative** indique bien que les temps « diminuent » avec l'âge plus jeune sur courtes distances.

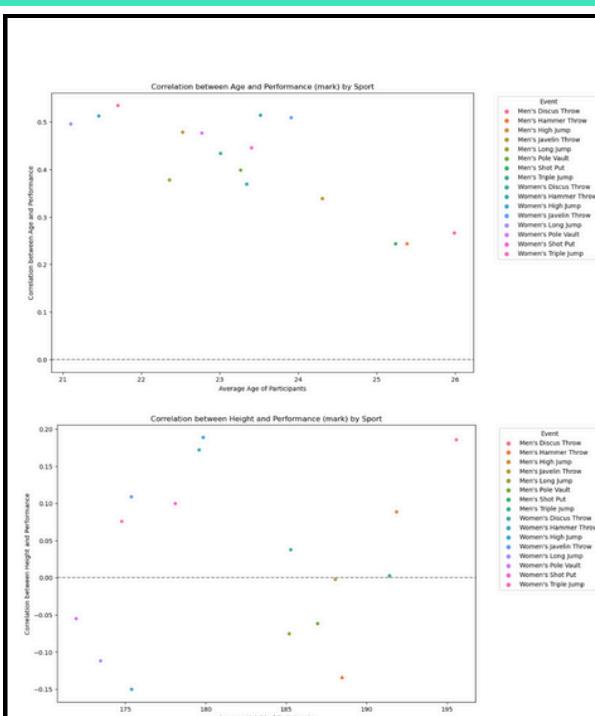
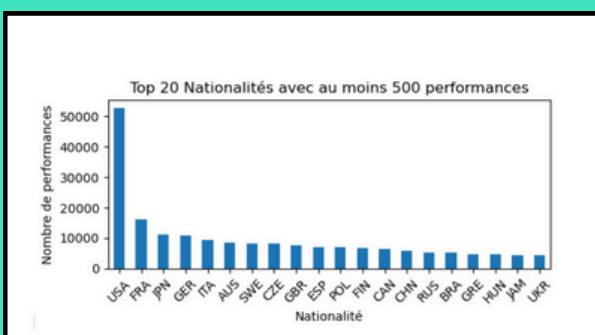
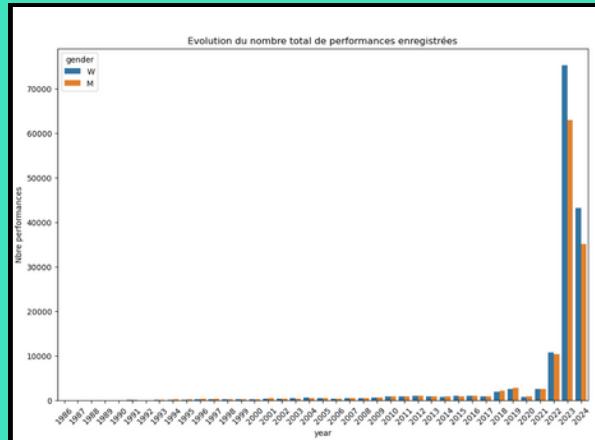
Sur le graphique de corrélation de la taille, nous observons qu'aucune corrélation n'est particulièrement évidente. Les scores obtenus sont très faibles et indiquent donc qu'il n'y a pas de corrélation.

Les nationalités affichent aussi une corrélation étonnamment faible avec la performance.



# avr24\_bds\_olympic\_games Dataset DISTANCE

## Visualisations



Le gap de performance entre hommes et femmes reste constant dans le temps.

Nous avons cependant un visuel direct sur l'effet de « dilution » des résultats sur les années 2020 compte tenu du nombre très élevés d'enregistrements sur cette période par rapport aux années précédentes.

Les **scores de correlations** obtenus sont une nouvelle fois assez bas. Nous observons qu'ils oscillent globalement entre 0.35 et 0.5 pour l'âge avec une tendance à être plus élevé pour les athlètes ayant entre 21 et 24 ans.

Cette non-corrélation est bien plus visible sur le graphique de la taille.

Les scores obtenus varient globalement de -0.15 à 0.20, et nous n'observons pas de tendance précise.

Nous pouvons constater que les athlètes de "distance" ont tendance à avoir une taille plus grande que les athlètes de "temps".

La taille moyenne masculine pour les athlètes de distance est de 1m88 tandis que cette moyenne est de 1m80 pour les athlètes de temps. Ce constat est identique chez les femmes : leur taille moyenne est de 1m75 pour les athlètes distance tandis qu'elle est de 1m68 pour les athlètes de temps.

# Qualité des données

## Evaluation des csv agrégés



### Validité

Pas d'erreurs, fautes de frappe ou de syntaxe



### Complétude

Pas d'informations manquantes



### Précision

Données correctes



### Disponibilité

Données accessibles facilement et rapidement



### Actualité

Données mises à jour régulièrement



### Cohérence

Chaque variable stockée dans un seul format

## Analyse

- 01. Validité**
- 02. Complétude**
- 03. Précision**
- 04. Disponibilité**
- 05. Actualité**
- 06. Cohérence**

## Les données agrégées sont :

fiables et sans erreurs.

incomplètes (NaN sur JO très anciens)  
Très déséquilibrées dans le temps

Précises pour les plus récentes.

disponibles en consultation libre sur les sites.

collectées à un instant t et sauvegardées intégralement.

présentes en format unique.

# avr24\_bds\_olympic\_games

# Modélisations Baseline

## Premiers modèles

### Séries temporelles

Nous avons exploré la piste des séries temporelles : la récursion des évènements tous les 4 ans nous a amené à explorer cette possibilité.

Un espoir vite douché puisque les modèles SARIMAX déployés s'avérèrent peu probants, avec des valeurs aberrantes (JB), sans doute dûs au manque de régularité temporelle et au manque d'observations par athlète.

Pas de saisonnalité avérée.



The screenshot shows the MLflow UI for a classification experiment. The search bar filters by 'metrics.rmse < 1 and params.model = "tree"'. The table view lists several models: keras\_mark2, keras\_mark, keras\_batch10-epochs50, svm1linear, svm001sigmoid, RFF 20 20 42, knn2, knn20, knn5, RFF 3 5 42, and RFF 10 10 42. Each entry includes a 'Run Name' column with colored circles and a 'Created' column showing the time since run creation.

### Classifications

Nous avons ensuite lancé les classifications, avec pour target si oui ou non l'athlète battait le record : or\_flag 1/0

or\_flag = mark - or\_perf

Un serveur MLFlow installé sur une VM permanente mise à disposition par DataScientest nous a permis de comparer plusieurs modèles : Random Forest, KNN, SVM...

Aucun ne donnait de résultat vraiment satisfaisant :

- Le score toujours très bon s'expliquait par la distribution déséquilibrée.
- Quand les mse, mae, rmse étaient correctes, le r2 ne l'était pas et inversement.

Seul Isolation Forest nous donna un résultat passable après une première erreur qui affichait des résultats trop bons pour être vrais (précision 95%...) dûs, après l'analyse, à des variables biaisées laissées par erreur (mark).

# avr24\_bds\_olympic\_games

# Classifications

## Classifications

classif\_dist\_JO2024\_1106 [Provide Feedback](#) [Share](#)

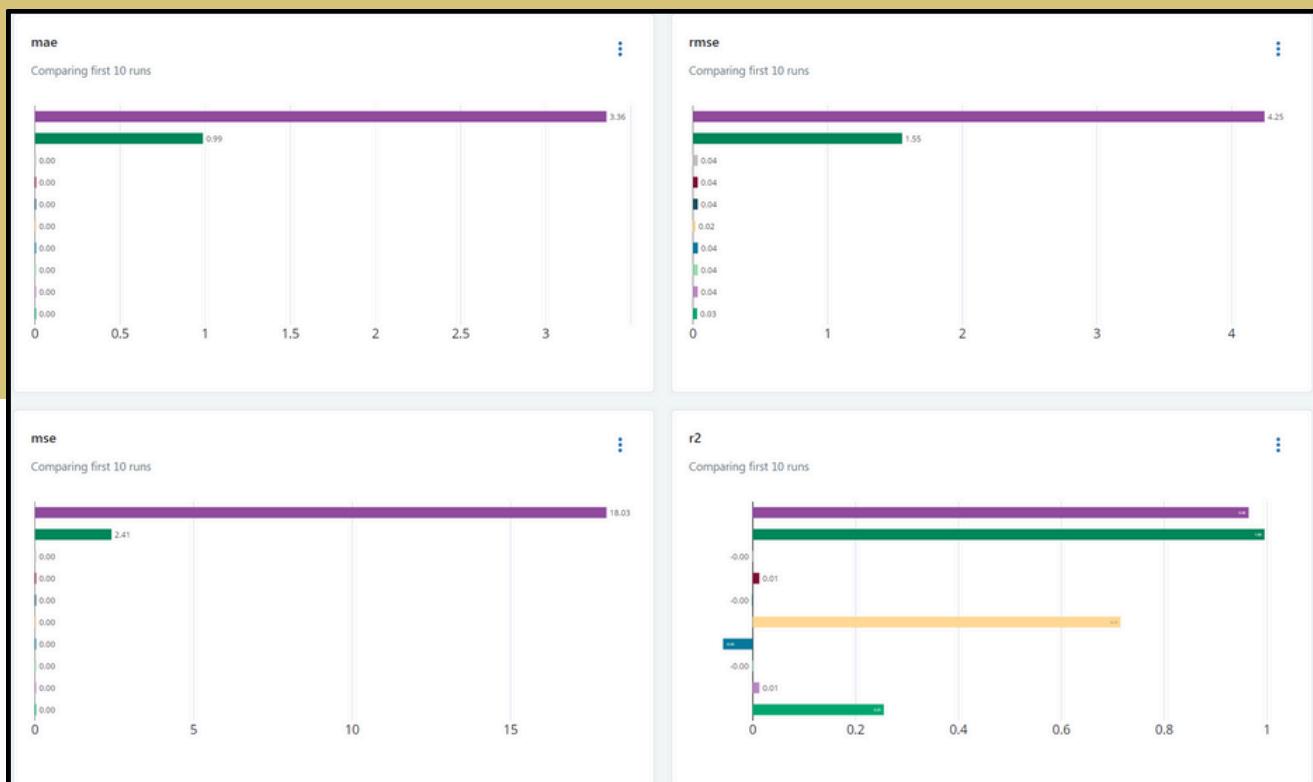
Experiment ID: 373134642549619295 Artifact Location: file:///home/ubuntu/MLflow/mlruns/373134642549619295

Description Edit

Metrics: rmse < 1 and params.model = "tree"

Time created: Sort: Created Columns: + New run

Run Name	Created	Duration	Source	Models	mae	mse	r2	rmse	C	batch_size
keras_mark2	7 days ago	47ms	experiment	-	3.356772028...	18.02510013...	0.964260683...	4.245597735...	-	10
keras_mark	8 days ago	44ms	experiment	-	0.986938641...	2.413630946...	0.99521436...	1.553586478...	-	10
keras_batch10-epochs50	8 days ago	23ms	experiment	-	0.001244435...	0.001244435...	-0.00124598...	0.035276551...	-	10
svmlinear	8 days ago	7.6s	experiment	sklearn	0.001226907...	0.001226907...	0.012856070...	0.035027244...	1	-
svm001sigmoid	8 days ago	9.4s	experiment	sklearn	0.001244435...	0.001244435...	-0.00124598...	0.035276551...	0.01	-
RFF 20 20 42	9 days ago	6.5s	experiment	sklearn	0.000656421...	0.000354149...	0.715059111...	0.018818851...	-	-
knn2	9 days ago	13.0s	experiment	sklearn	0.001314544...	0.001314544...	-0.05765421...	0.036256642...	-	-
knn20	9 days ago	13.9s	experiment	sklearn	0.001244435...	0.001244435...	-0.00124598...	0.035276551...	-	-
knn5	9 days ago	13.8s	experiment	sklearn	0.001226907...	0.001226907...	0.012856070...	0.035027244...	-	-
RFF 3 5 42	9 days ago	6.8s	experiment	sklearn	0.001835616...	0.000926080...	0.254895299...	0.030431571...	-	-
RFF 10 10 42	9 days ago	6.6s	experiment	sklearn	0.000856224...	0.000423546...	0.659223263...	0.020580252...	-	-



*r2 excellents quand mse, mae, rmse mauvais et inversement...*

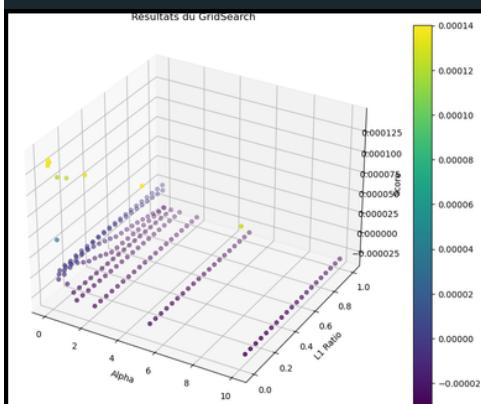
# avr24\_bds\_olympic\_games

## Régressions

### Analyse

N'ayant pas de corrélations fortes entre les variables descriptives et notre valeur à prédire ("mark" = performance de l'athlète), un premier essai avec le modèle "Linear Regression" de Sklearn nous surprend. Nous obtenons en effet un très bon coefficient de détermination R2 à 0,99, équivalent sur le jeu d'entraînement et le jeu de test.

En étudiant les coefficients nous remarquons que la variable sur laquelle le modèle s'est appuyé est l'event (le type d'épreuve), ce qui est normal : nous avons un jeu de données hétérogène avec de multiples types d'épreuves et des performances qui ont un rapport d'échelle très marqué suivant cette variable (exemple : un 100 mètres sera entre 9s et 11s, un marathon sera supérieur à 8000s).



Cette hétérogénéité entraîne une moyenne d'erreur très élevée quelle que soit la métrique MAE/MSE/RMSE.

```
coef_lr=pd.DataFrame(lr.coef_)

coefs_lr=pd.concat([X_variables,coef_lr],axis=1)
coefs_lr.columns=[('variables','coef_lr')]
print(coefs_lr)

variables coef_lr
0 competition_name 358.488649
1 event 14278.365755
2 rtype -2.038834
3 gender 18.967878
4 nationality -353.495151
5 wind -7.496532
6 height -40.505457
7 weight -46.224309
8 age 92.848381

list_scores_lr=pd.Series(score(y_test,y_pr
list_scores_label=pd.Series(['Coef determination
scores_md1=pd.concat([list_scores_label,list
print(scores_md1)

0 1
0 Coef determination 0.994861
1 MeanAbsoluteError 41.911753
2 MeanSquaredError 18794.654150
3 RootMeanSquaredError 137.093596
4 MedianAbsoluteError 12.547914
```

regress\_dist\_JO2024 [Provide Feedback](#)

Experiment ID: 827141048806461476 Artifact Location: file:///home/ubuntu/Mlflow/mlruns/827141048806461476

Description Edit

metrics.rmse < 1 and params.model == "tree"

Time created State: Active Sort: Created Columns

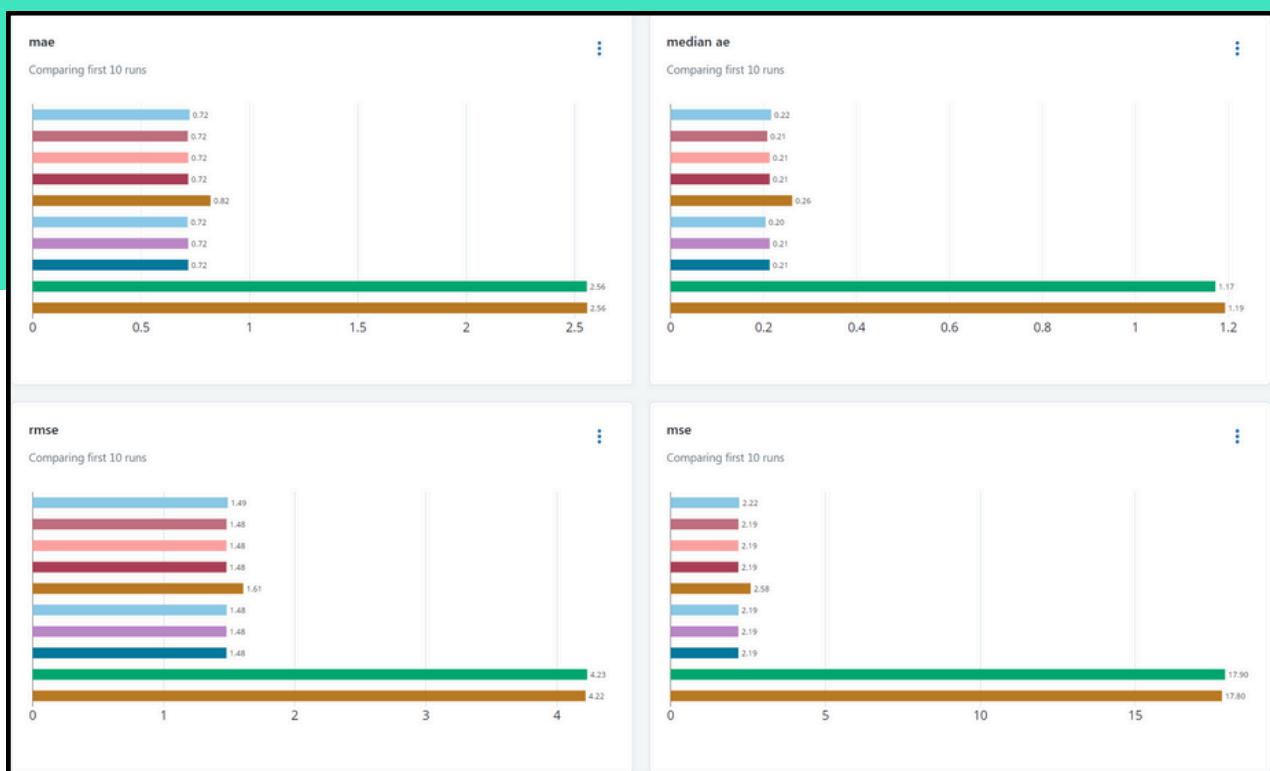
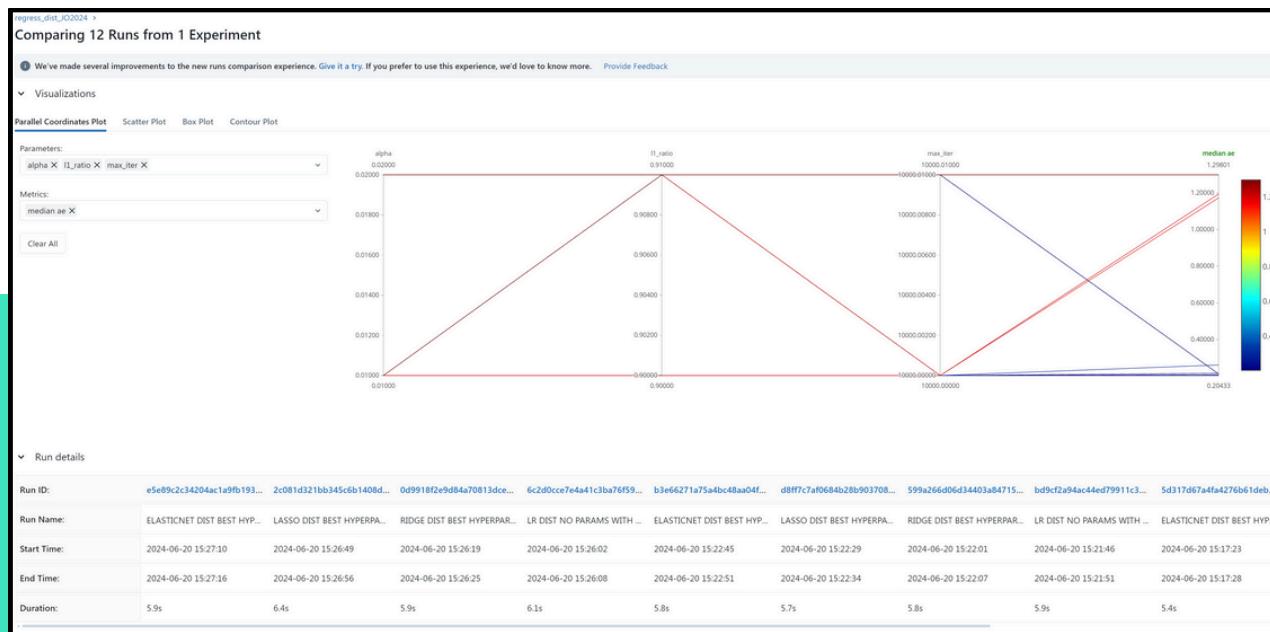
Table Chart Evaluation Experimental

	Run Name	Created	Duration	Source	Models	mae	median ae	mse	r2	rmse	alpha
0	ELASTICNET DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.9s	experiment...	sklearn	0.724264819...	0.216224975...	2.216238845...	0.995931925...	1.488703746...	0.01
1	LASSO DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 6.4s	experiment...	sklearn	0.715647784...	0.20781502...	2.189282928...	0.995981404...	1.479622562...	0.01
2	RIDGE DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.9s	experiment...	sklearn	0.717815471...	0.213155247...	2.187470242...	0.995984732...	1.479009885...	0.01
3	LR DIST NO PARAMS WITH ENRICHMENT STD	20 hours ago	~ 6.1s	experiment...	sklearn	0.717815681...	0.213184498...	2.187471608...	0.995984729...	1.479010347...	-
4	ELASTICNET DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.8s	experiment...	sklearn	0.821041747...	0.261617119...	2.584113766...	0.995256662...	1.607517890...	0.01
5	LASSO DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.7s	experiment...	sklearn	0.715271351...	0.204333219...	2.192887348...	0.995974788...	1.480840082...	0.01
6	RIDGE DIST BEST HYPERPARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.8s	experiment...	sklearn	0.717812718...	0.213219779...	2.187452024...	0.995984765...	1.479003727...	0.01
7	LR DIST NO PARAMS WITH ENRICHMENT MMS	20 hours ago	~ 5.9s	experiment...	sklearn	0.717815681...	0.213184498...	2.187471608...	0.995984729...	1.479010347...	-
8	ELASTICNET DIST BEST HYPERPARAMS NO ENRICHMENT MMS	20 hours ago	~ 5.4s	experiment...	sklearn	2.556974677...	1.172216506...	17.90383130...	0.967136157...	4.231291918...	0.01
9	LASSO DIST BEST HYPERPARAMS NO ENRICHMENT MMS	20 hours ago	~ 5.5s	experiment...	sklearn	2.558434058...	1.193170222...	17.80401916...	0.967319370...	4.219480911...	0.01
10	RIDGE DIST BEST HYPERPARAMS NO ENRICHMENT MMS	20 hours ago	~ 5.5s	experiment...	sklearn	2.607338440...	1.298000234...	17.76374066...	0.967393304...	4.214705287...	0.01
11	LR DIST NO PARAMS NO ENRICHMENT MMS	20 hours ago	~ 5.5s	experiment...	sklearn	2.607340167...	1.298009673...	17.76374051...	0.967393305...	4.214705270...	-

# avr24\_bds\_olympic\_games

# Régressions

## MLFLOW



# KERAS & bilan préliminaire

Nous avons mis en place du modèle Deep Learning via **Keras** sur de la régression avec pour cible la performance "mark".

La structure neuronale choisie était :

- 2 neurones 64, 32, 1
- activation relu
- sortie sigmoid
- epoch = 50
- batch size = 64
- loss = mean squared error

Les résultats sont très décevants.

```
son > ⚡ experiment_keras.py ...
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Input, Dropout
from sklearn.model_selection import train_test_split
from sklearn (function) def read_csv(
import pandas as pd
filepath_or_buffer: FilePath | ReadCsvBuffer[bytes] |
import numpy as np
ReadCsvBuffer[str],
+
# tracking
client = M
sep: str | None = ...,
delimiter: str | None = ...,
header: int | Sequence[int] | Literal['infer'] | None = ...,
names: List[LikeMutable | None = ...,
index_col: int | str | Sequence[str | int] | Literal[False] | None =
run_name = ...,
artifact_p
usecols: UsecolsArgType = ...,
dtype: DtypeArg | defaultdict | None = ...,
# Dataset
data = pd.read_csv("IA_distr_eco_14062024.csv")
X = data.dropna(inplace=True, axis=1, how='any')
y = data['mark']
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)

# Keras neurons
model = Sequential()
model.add(Input(shape=(X_train.shape[1],)))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))
```

Table	Chart	Evaluation	Experimental								
				Metrics							
				mae	mse	r2	rmse				
	Run Name	Created									
	Pin run	s_mark	32 minutes ago	-	23ms	experiment_keras.py	-	87.0020697...	92375.20383...	0.973184542...	303.9328936...

## En chiffres

### Preprocessing

#### Données brutes

Des NaN dans nos variables significatives

### Classification

#### Isolation Forest

est celui qui s'en est le mieux sorti

### Régression

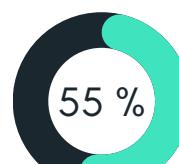
#### Lasso

est celui qui s'en est le mieux sorti



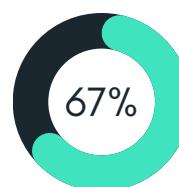
#### Accuracy

Le taux était biaisé par un dataset trop déséquilibré



#### MSE, MAE

Leurs valeurs moyennes quel que soit le modèle choisi



#### R2

Valeur inversément liée aux résultats des MSE, MAE

## Conclusion préliminaire

Les premiers modèles sous performant, nous avons poursuivis 2 pistes d'amélioration :

- Modifier le train/test manuellement : au lieu de le prendre arbitrairement en appliquant un test\_size de 0.2 ou 0.1, nous avons sélectionné le test\_set sur l'année 2024. Les résultats ont été peu impactés.
- Enrichir les données, ce que nous exposons ci-après.

# avr24\_bds\_olympic\_games

## Améliorations

### Enrichissements

```
## DECLARING FUNCTION
def annual_perf_stat(X):
    max=len(X)
    ## LOOP ON EACH PERF
    for idx,el in enumerate(X.name):
        moyperf=0
        nbsperf=0
        print("last annual perf bonus ",idx,"/",max," ",end="\r")
        # Get Race Date
        rdate=X.iloc[idx]['rdate']
        # Calculate Race Date - 365 days
        rdatem1=rdate-pd.Timedelta(days=365)
        # Get event
        event=X.iloc[idx]['event']
        # I want to have all athlete name and Race Date = 365 days
        sampleddfX[(X.event==event)&(X.rdate<rdate)&(X.name==el)&(X.rdate>rdatem1)]
        # Calculate Mean perf
        moyperf=sampled['mark'].mean()
        # Calculate Max perf
        maxperf=sampled['mark'].max()
        # Calculate Number of perf
        nbsperf=len(sampled)
        X.at[idx,'annual_perf_nb']=nbsperf
        X.at[idx,'annual_perf_moy']=moyperf
        X.at[idx,'annual_perf_max']=maxperf
    return None

## ADDING LAST ANNUAL PERF BONUS FEATURE (-1 year date < perf < race date )
dfd['annual_perf_nb']=0
dfd['annual_perf_nb']=0
dfd['annual_perf_max']=0
dfd['annual_perf_moy']=0
dfd['annual_perf_nb']=0
dfd['annual_perf_moy']=0
dfd['annual_perf_max']=0
dfd['annual_perf_nb']=0
dfd['annual_perf_moy']=0
annual_perf_stat(dfd)
dfd.to_csv("data/_/MA_temps.csv",index=False)
annual_perf_stat(dfd)
dfd.to_csv("data/_/MA_dist.csv",index=False)
```

L'ajout de variables d'enrichissement doit permettre de refléter le niveau de forme des athlètes à partir de leurs performances passées. Ce qui devrait aussi limiter le poids de la variable "Event".

Les 3 nouvelles variables ajoutées :

- le nombre d'épreuves dans les 365 derniers jours.
- la moyenne des performances dans ce laps de temps.
- la meilleure performance de l'année.

Nous évaluons avec un GridCV plusieurs modèles de régression, le Ridge, le Lasso, ainsi que l'Elastic Net puis nous comparons les résultats.

Compte tenu de la grande amplitude des variables enrichies (performance moyenne et maximale), l'encodage des variables qualitatives est décliné de deux façons différentes: en utilisant le MinMaxScaler (normalisation) ou le StandardScaler (standardisation).

### Métriques enrichies

	Run Name	Created	Dat	Duration	Source	Models	mae	mse	r2	rmse
	LR NO PARAMS MMS 1	4 days ago	-	6.0s	experience	sklearn	0.718056802...	2.187599512...	0.995984494...	1.479053586...

Ayant un bon coefficient de détermination nous nous focalisons sur le **RMSE** (Root Mean squared Error) comme métrique d'évaluation afin d'obtenir les modèles qui délivrent l'erreur moyenne la plus faible. 10 cross-validations nous donnent les meilleurs hyper paramètres pour chaque modèle.

Les résultats comparatifs entre les modèles permettent, selon le jeu de données (temps / distance) et la métrique observée, de constater qu'il n'est pas simple de départager les modèles sur la base des métriques MAE/MSE/RMSE

On remarque que le StandardScaler entraîne une réduction notable de l'amplitude des erreurs dans de nombreux cas, sauf sur le jeu de données "Distance" et paradoxalement sur le modèle Lasso qui donne les meilleurs résultats.

# avr24\_bds\_olympic\_games

## Performances

StandardScaler

Jeu de données Temps (performances en secondes) / Standard Scaler				
Score type / Modèle	Linear Regression	Ridge	Lasso	ElasticNet
Coef determination	0.998328484	0.99832814	0.998286014	0.998306379
MeanAbsoluteError	16.99970093	17.00048197	16.06035463	17.60308541
MeanSquaredError	6113.088748	6114.346422	6268.412879	6193.934347
RMeanSquaredError	78.18624398	78.19428638	79.17330913	78.70155238
MedianAbsoluteError	2.53820497	2.536903724	1.321199917	2.893781551

MinMaxScaler

Jeu de données Temps (performances en secondes) / MinMaxScaler				
Score type / Modèle	Linear Regression	Ridge	Lasso	ElasticNet
Coef determination	0.998328484	0.998186271	0.99707548	0.991372008
MeanAbsoluteError	16.99970093	19.815374	23.68428914	73.85770379
MeanSquaredError	6113.088748	6633.191392	10695.59194	31554.40429
RMeanSquaredError	78.18624398	81.44440676	103.419495	177.6355941
MedianAbsoluteError	2.53820497	4.092283916	2.544704253	38.28603672

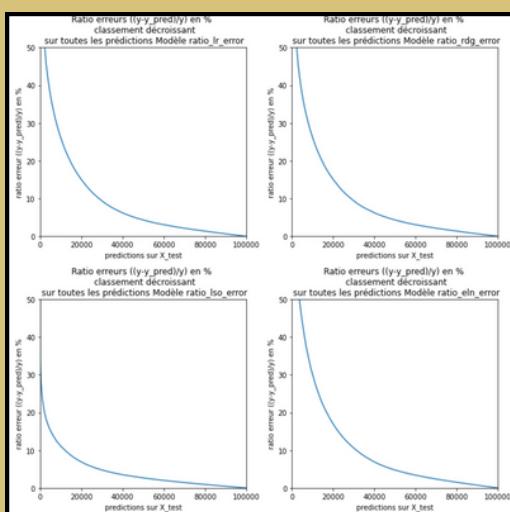
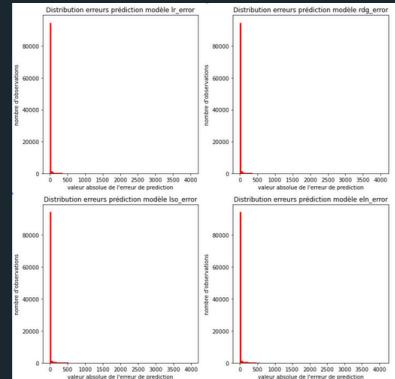
Jeu de données Distance (performances en mètres) / Standard Scaler				
Score type	Linear Regression	Ridge	Lasso	ElasticNet
Coef determination	0.99598473	0.995984732	0.995981406	0.995931925
MeanAbsoluteError	0.71781709	0.71781688	0.715647521	0.724264819
MeanSquaredError	2.187471473	2.187470107	2.189282349	2.216238843
RMeanSquaredError	1.479010302	1.47900984	1.479622367	1.488703746
MedianAbsoluteError	0.213170789	0.213141035	0.207873498	0.216224974

Jeu de données Distance (performances en mètres) / MinMaxScaler				
Score type	Linear Regression	Ridge	Lasso	ElasticNet
Coef determination	0.99598473	0.995984766	0.995974789	0.995256663
MeanAbsoluteError	0.71781709	0.717814122	0.715271351	0.821041746
MeanSquaredError	2.187471473	2.187451888	2.192887349	2.584113763
RMeanSquaredError	1.479010302	1.479003681	1.480840082	1.607517889
MedianAbsoluteError	0.213170789	0.213200427	0.204333218	0.261617119

L'amplitude moyenne des erreurs est beaucoup plus grande sur le jeu de données Temps car l'amplitude des performances par épreuve est elle aussi très grande.

Pour analyser les erreurs, nous affichons la distribution de la valeur absolue de  $y-y_{\text{pred}}$  (à droite)

On observe qu'une très large part des prédictions ont une erreur modérée et que c'est un petit nombre de prédictions, dont l'erreur est importante, qui impacte la moyenne des erreurs. Sur ce critère, les 4 modèles semblent assez proches.



En mesurant le ratio de l'erreur par rapport à la valeur cible  $((y-y_{\text{pred}})/y)$ , et en triant de manière décroissante, on remarque que le modèle Lasso est bien meilleur sur le jeu de données "Temps".

En effet 20% des prédictions ont un ratio d'erreur inférieur à 8/9% alors que les 3 autres modèles affichent entre 30 et 40 %.

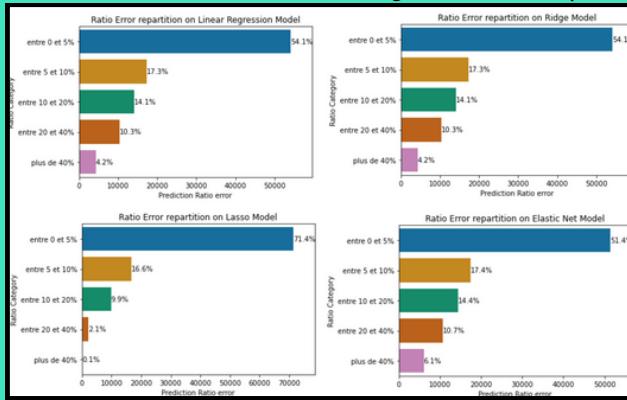
Même résultat en utilisant la métrique **Median Absolute Error** (et qui est constaté également sur le jeu de données "Distances"). Cette métrique étant moins sensible aux valeurs extrêmes puisqu'elle se base sur la médiane des erreurs.

Globalement, cette tendance est la même sur les deux jeux de données, mais l'échelle et le volume des erreurs sont beaucoup plus importants sur le jeu de données "Temps".

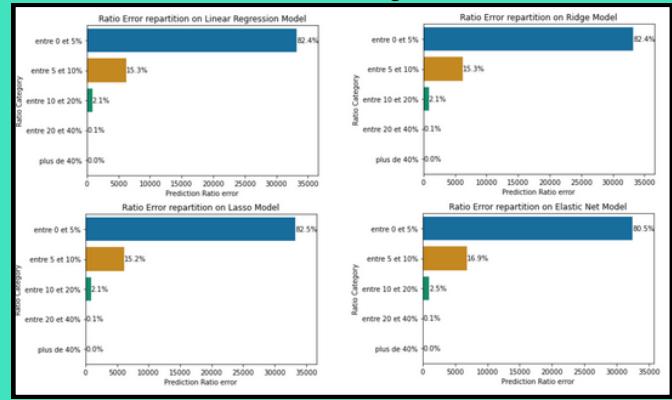
# avr24\_bds\_olympic\_games

## Ratio error

*Ratio error Regression Temps*



*Ratio error Regression Distance*



Jeu de données Temps (performances mesurées en secondes)				
variables / Modèles	Coefficients LR	Coefficients Ridge	Coefficients Lasso	Coefficients ElasticNet
competition_name	-15.54477645	-15.36021698	-2.657286484	-6.882072599
event	357.7602682	360.972178	369.5244595	449.3940971
rtype	-0.739369251	-0.731893881	0	-0.466691715
gender	1.331795886	1.358894079	0	2.668792531
nationality	-2.839745469	-2.845489139	-1.234545263	-3.994318268
wind	-0.189816128	-0.189363085	0	-0.183395495
height	-1.898975071	-1.899233948	0	-2.344131962
weight	1.053421486	1.011295416	0	-0.306192285
age	1.985289584	1.983656942	0.793595627	2.518536203
annual_perf_nb	0.078282881	0.115707272	0	-0.17615039
annual_perf_max	551.8703621	600.48353901	1416.45402	739.9997653
annual_perf_moy	1002.60607	950.5948861	113.6181174	713.2749573

L'analyse des coefficients de chaque modèle permet d'éclaircir les résultats finaux, et nous pouvons faire plusieurs constats :

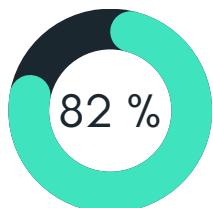
- les coefficients sont très différents selon le jeu de données (les hyper paramètres ne sont pas les mêmes). Mais c'est surtout l'hétérogénéité des données qui impacte le poids accordé à chaque variable (exemple : sur le jeu de données "Temps", un poids très important est toujours donné au type d'épreuve "event").
- le modèle **Lasso** a pénalisé beaucoup plus de variables sur le jeu de données "Temps", et a accordé un poids majeur à la **meilleure performance annuelle**.

Jeu de données Distance (performances mesurées en mètres)				
variables / Modèles	Coefficients LR	Coefficients Ridge	Coefficients Lasso	Coefficients ElasticNet
competition_name	0.05332729	0.053327125	0.045187437	0.060240493
event	1.047601154	1.047650636	1.106052205	1.494629964
rtype	0.058126851	0.058125165	0.045424654	0.035174153
gender	-0.020203903	-0.020200992	-1.97678E-05	0
nationality	0.015495517	0.015496751	0.009299047	0.019020057
wind	3.01981E-14	0	0	0
height	-0.003769984	-0.003772513	0	0
weight	0.023703756	0.023705307	0.000556116	0.008149722
age	-0.017091104	-0.017086553	0	0.005621178
annual_perf_nb	-0.009300137	-0.009309993	-0.016365784	-0.042569641
annual_perf_max	3.069955755	3.071045257	4.849851271	8.870905214
annual_perf_moy	19.25414305	19.25300213	17.418272121	12.99277879

# Conclusion Regression enrichie

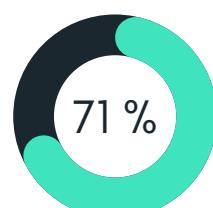
La disparité dans les échelles des observations de la performance pénalise les prédictions des modèles.

Nous avons envisagé d'isoler chaque type d'épreuve dans un dataset distinct. Malheureusement les tests effectués sur l'épreuve du 100 mètres (hommes+femmes) avec les mêmes modèles révèlent sur le jeu de test un r2 beaucoup moins bon (0.82) et une différence notable avec le jeu d'entraînement (0.62) ; sans doute du fait d'un nombre insuffisant d'observations.



**Prédictions Distance**

à un ratio d'erreur inférieur à 5%



**Prédictions Temps**

à un ratio d'erreur inférieur à 5%

Une prédiction dont le ratio d'erreur est inférieure à 5% est exploitable. Or 82% des prédictions sur le jeu de données "Distance" et 71% sur le jeu de données "Temps" obtiennent un ratio en dessous de 5%.

A partir de la prédiction de la performance de l'athlète, nous pouvons examiner la probabilité que celui-ci dépasse ou non le record olympique. Les variables enrichies ont permis de fournir une prédiction assez proche d'une performance réelle.

En ajoutant un indice de confiance nous pourrions tenter d'établir une pondération de la probabilité qu'un athlète batte ou pas un record olympique.

```
dftgrp=dft.groupby(['name','event']).agg({'coef_perf':[ 'count','median','min','max','std']}).reset_index()
dftgrpb=dft.groupby(['name','event'])['coef_perf'].apply(lambda x: (x*8).sum()).reset_index(name='orperf_count')
dftgrp.columns = ['name', 'event', 'perf_number', 'coef_perf_median', 'coef_best_perf', 'coef_worst_perf', 'coef_perf_et']
dftgrp['orperf_count']=dftgrpb['orperf_count']
dftgrp['orperf_probab']=dftgrpb['orperf_count']/dftgrp['perf_number']
dftgrp['orperf_proba2']=np.where((dftgrp['perf_number']/5)<=1,dftgrp['orperf_probab']*(dftgrp['perf_number']/5),dftgrp['orperf_probab'])
dftgrp['orperf_proba']=np.where((dftgrp['coef_perf_median'])<0,dftgrp['orperf_proba2']*(1+abs(dftgrp['coef_perf_median']/100)),dftgrp['orperf_proba2'])
dftgrp=dftgrp.drop(columns=['orperf_probab','orperf_proba2'])
dftgrp=dftgrp.sort_values(by=[ 'coef_perf_median','coef_perf_et'],ascending=[True,True])
dftgrp.to_excel("debug/temps_synthese.xlsx")
```

# Fichiers Target et Résultats

## Les athlètes aux JO 2024

athlete_events.csv (41.5 MB)		
Detail	Compact	Column
ID	Name	S
1	134732 unique values	M F
1	A Dijiang	M
2	A Lamusi	M
3	Gunnar Nielsen Aaby	M
4	Edgar Lindenaau Aabye	M
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
5	Christine Jacoba Aaftink	F
6	Dan Kwart Aaftink	M

## Préparation de notre set validation

Comment prédire les records des JO 2024 en athlétisme individuel ?

De base notre set d'athlètes qui sont qualifiés pour ces JO est incomplet. Certaines fédérations, dont plusieurs majeures (USA, Kenya,...), n'annonceront pas de liste définitive avant le 9 juillet !

Nous sommes partis sur un dataset projeté à partir des annonces partielles, en le complétant avec les athlètes ayant réalisé les meilleures performances de l'année.

## Quelques prédictions :

competition_name	event	rtype	rdate	name	gender	nation	age	annual_perf_nb	annual_perf_max	annual_perf_noy	height	weight	or_perf	y_pred_lasso	or_fla
0	The XXXIII Olympic Games	Men's Long Jump	Final	2024-05-1 MATTIA FURLANI	M	ITA	18,987	14	8,44	8,032142857	185	75	8,9	8,063930223	
1	The XXXIII Olympic Games	Men's Triple Jump	Final	2024-05-1 JAYDON HIBBERT	M	JAM	19,27721	6	17,7	17,45666667	185	77	18,09	17,3626672	
2	The XXXIII Olympic Games	Men's Javelin Throw	Final	2024-05-1 MAX DEHNING	M	GER	19,46664	13	90,2	76,68384615	188	95	90,57	78,95970569	
3	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 NATALIA UNARES	W	COL	21,05955	10	6,86	6,493	173	60	7,4	6,534610571	
4	The XXXIII Olympic Games	Women's Hammer Throw	Final	2024-05-1 SIUA KOSONEN	W	FIN	21,23203	19	74,19	72,36894737	175	82	82,29	71,74735886	
5	The XXXIII Olympic Games	Women's High Jump	Final	2024-05-1 LARISSA SPIRIDONOVA	W	RUS	21,44285	19	2	1,896315789	180	61	2,06	1,903201683	
6	The XXXIII Olympic Games	Women's High Jump	Final	2024-05-1 NATALYA SPIRIDONOVA	W	RUS	21,44285	19	2	1,896315789	180	61	2,06	1,903201683	
7	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 IRENEA IAPICHINO	W	ITA	21,52772	15	6,95	6,76	173	60	7,4	6,729721372	
8	The XXXIII Olympic Games	Men's Discus Throw	Final	2024-05-1 MYKOLAS ALEKNA	M	LTU	21,54415	10	74,35	67,342	197	120	69,89	67,98241787	
9	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 ACKELLA SMITH	W	JAM	21,99042	10	7,1	6,71	175	61	7,4	6,737507116	
10	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 ACKELLA SMITH	W	JAM	21,99042	10	7,1	6,71	175	61	7,4	6,737507116	
11	The XXXIII Olympic Games	Women's Triple Jump	Final	2024-05-1 LEYANIS PÉREZ HERNANDEZ	W	CUB	22,06434	15	14,98	14,67466667	175	61	15,67	14,57458963	
12	The XXXIII Olympic Games	Women's Triple Jump	Final	2024-05-1 LEYANIS PÉREZ HERNANDEZ	W	CUB	22,06434	15	14,98	14,67466667	175	61	15,67	14,57458963	
13	The XXXIII Olympic Games	Women's High Jump	Final	2024-05-1 YAROSLAVA MAHUCHIKH	W	UKR	23,3655	12	2,04	1,989166667	180	61	2,06	1,993429484	
14	The XXXIII Olympic Games	Women's High Jump	Final	2024-05-1 YAROSLAVA MAHUCHIKH	W	UKR	23,3655	12	2,04	1,989166667	180	61	2,06	1,993429484	
15	The XXXIII Olympic Games	Men's Hammer Throw	Final	2024-05-1 MIKHAYLO KOKHAN	M	UKR	23,03354	11	79,59	77,55181818	188	111	84,8	77,09006924	
16	The XXXIII Olympic Games	Men's Long Jump	Final	2024-05-1 WAYNE PINNOCK	M	JAM	23,27447	9	8,54	8,335555556	185	75	8,9	8,309391834	
17	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 AGATE DE SOUSA	W	STP	23,60849	12	7,03	6,668333333	173	60	7,4	6,674026335	
18	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 AGATE DE SOUSA	W	STP	23,60849	12	7,03	6,668333333	173	60	7,4	6,674026335	
19	The XXXIII Olympic Games	Women's Pole Vault	Final	2024-05-1 MOLLY CAUDERT	W	GBR	23,86585	16	4,86	4,643125	172	60	5,05	4,637895062	
20	The XXXIII Olympic Games	Women's Pole Vault	Final	2024-05-1 MOLLY CAUDERT	W	GBR	23,86585	16	4,86	4,643125	172	60	5,05	4,637895062	
21	The XXXIII Olympic Games	Men's Long Jump	Final	2024-05-1 YU TANG LIN	M	TPE	23,96167	7	8,4	7,737142857	185	75	8,9	7,849764802	
22	The XXXIII Olympic Games	Women's Discus Throw	Final	2024-05-1 ORINDE VAN KLINKEN	W	NED	23,98631	13	67,2	64,04538462	186	88	72,3	63,7510427	
23	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 KC LIGHTFOOT	M	USA	24,16975	17	6,07	5,823529421	187	80	6,03	5,81575816	
24	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 ARMAND DUPLANTIS	M	SWE	24,21355	19	6,24	5,958947368	187	80	6,03	5,946209384	
25	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 EMMANOUIL KARALAS	M	GRC	24,25188	24	5,85	5,626666667	187	80	6,03	5,607453023	
26	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 ERISU SASMA	M	TUR	24,30664	20	5,9	5,629	187	80	6,03	5,634139088	
27	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 THIBAUT COLLET	M	FRA	24,57221	21	5,92	5,736190476	187	80	6,03	5,706523289	
28	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 TARA DAVIS WOODHALL	W	USA	24,6872	14	7,18	6,915	173	60	7,4	6,887986148	
29	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 TARA DAVIS WOODHALL	W	USA	24,6872	14	7,18	6,915	173	60	7,4	6,887986148	
30	The XXXIII Olympic Games	Men's High Jump	Final	2024-05-1 JUVAUGHN HARRISON	M	USA	24,7666	6	2,36	2,318333333	192	77	2,39	2,31675833	
31	The XXXIII Olympic Games	Men's Pole Vault	Final	2024-05-1 BO KANDA LITA BAEHRE	M	GER	24,78576	11	5,82	5,602727273	187	80	6,03	5,61760161	
32	The XXXIII Olympic Games	Women's Shot Put	Final	2024-05-1 JESSICA SCHILDER	W	NED	24,86242	17	20,31	18,9517647	178	95	22,41	18,96061256	
33	The XXXIII Olympic Games	Women's Shot Put	Final	2024-05-1 JESSICA SCHILDER	W	NED	24,86242	17	20,31	18,9517647	178	95	22,41	18,96061256	
34	The XXXIII Olympic Games	Women's Hammer Throw	Final	2024-05-1 CAMRYN ROGERS	W	CAN	24,87064	11	78,62	76,75727273	175	82	82,29	75,90742155	
35	The XXXIII Olympic Games	Women's Hammer Throw	Final	2024-05-1 CAMRYN ROGERS	W	CAN	24,87064	11	78,62	76,75727273	175	82	82,29	75,90742155	
36	The XXXIII Olympic Games	Women's Shot Put	Final	2024-05-1 ADELAIDE AQUILLA	W	USA	24,90623	17	19,38	18,19941176	178	95	22,41	18,21188185	
37	The XXXIII Olympic Games	Men's Discus Throw	Final	2024-05-1 KRISTJAN CEH	M	SLO	25,01574	17	71,86	68,68823529	197	120	69,89	68,39663061	
38	The XXXIII Olympic Games	Men's Triple Jump	Final	2024-05-1 CRISTIAN NAPOLES	M	CUB	25,16906	11	17,4	16,76727273	185	77	18,09	16,77591864	
39	The XXXIII Olympic Games	Women's Long Jump	Final	2024-05-1 MILICA GARDASEVIC	W	SRB	25,31143	19	6,91	6,648421053	173	60	7,4	6,619190593	
40	The XXXIII Olympic Games	Women's Pole Vault	Final	2024-05-1 WILMA MURTO	W	FIN	25,57153	17	4,81	4,647058824	181	68	5,05	4,629299831	
41	The XXXIII Olympic Games	Men's Long Jump	Final	2024-05-1 CAREY MCLEOD	M	JAM	25,80972	9	8,27	8,073333333	185	75	8,9	8,053977406	
42	The XXXIII Olympic Games	Men's Long Jump	Final	2024-05-1 Miltiadis Tentoglou	M	GRC	25,86095	15	8,52	8,234666667	185	75	8,9	8,210948176	

Fichier excel prédictif obtenu par application du Lasso entraîné sur notre dataset validation Distance. **Pas de records battus sur les Distances.**

# Au final : direction les bookmakers ?

## Conclusions

competition_name	event	name	gender	nationality	height	weight	age	annual_perf_nb	annual_perf_max	annual_perf_moy	or_perf	y_pred_lasso	or_flag
The XXXIII Olympic Games	Women's 1500 Metres	DIRIBE WELTEJI	W	ETH	169	56	21,93839836	9	235,08	238,1822222	233,11	232,71	
The XXXIII Olympic Games	Men's 110 Metres Hurdles	RACHID MURATAKE	M	JPN	188	80	22,2587269	10	13,04	13,417	12,91	11,71	
The XXXIII Olympic Games	Women's 800 Metres	WORKNESH MESELE	W	ETH	171	58	22,85831622	5	118,75	119,572	113,43	111,95	
The XXXIII Olympic Games	Women's 400 Metres	MARY MORAA	W	KEN	169	59	23,61396304	10	50,38	51,312	48,25	47,26	
The XXXIII Olympic Games	Men's 110 Metres Hurdles	SHUNSUKE IZUMIYA	M	JPN	188	80	24,33271732	12	13,04	13,2375	12,91	12,13	
The XXXIII Olympic Games	Men's 100 Metres	ABDUL HAKIM SANI BROWN	M	JPN	181	76	25,04859685	16	9,97	10,195625	9,63	9,15	
The XXXIII Olympic Games	Women's 100 Metres Hurdles	YANNI WU	W	CHN	170	62	26,7963723	11	12,76	13,02363636	12,26	11,19	
The XXXIII Olympic Games	Women's 1500 Metres	GUDAF TSEGAY	W	ETH	167	52	27,3887748	2	230,3	232,165	233,11	229,85	
The XXXIII Olympic Games	Women's Marathon	TIGST ASSEFA	W	ETH	168	53	27,38124572	2	7913	8048	8587	8301,93	
The XXXIII Olympic Games	Men's 100 Metres	FERDINAND OMANYALA	M	KEN	180	75	28,6009582	22	9,78	10,01272727	9,63	6,65	
The XXXIII Olympic Games	Women's Marathon	RUTH CEPNGETICH	W	KEN	163	48	29,70294319	3	8137	8367	8587	8496,43	
The XXXIII Olympic Games	Men's Marathon	SISAY LEMMA	M	ETH	172	60	33,34154689	3	7308	7490,333333	7502	7547,54	

Sur le **dataset Temps**, plusieurs records battus sont annoncés.

Avec des erreurs flagrantes du modèle : records du 100m en 9,15s ou 6,67s (sic) pour Abdul Hakim Sani Brown. Par contre les prédictions de records battus sur le Marathon, homme et femme sont crédibles pour Tigst ASSEFA et RUTH CEPNGETICH par exemple.

Le modèle est peu fiable pour les courses rapides.

name	er	tio	y_pred_lasso
DIRIBE WELTEJI	W	ETH	232,71
RACHID MURATAKE	M	JPN	11,71
WORKNESH MESELE	W	ETH	111,95
MARY MORAA	W	KEN	47,26
SHUNSUKE IZUMIYA	M	JPN	12,13
ABDUL HAKIM SANI BROWN	M	JPN	9,15
YANNI WU	W	CHN	11,19
GUDAF TSEGAY	W	ETH	229,85
TIGST ASSEFA	W	ETH	8301,93
FERDINAND OMANYALA	M	KEN	6,65
RUTH CEPNGETICH	W	KEN	8496,43
SISAY LEMMA	M	ETH	7547,54

## En perspective



Mieux gérer certaines tâches (avec les cours à posteriori) :

- Meilleure gestion de la répartition du temps (preproc/model).
- Contrôle qualité : erreurs de preprocessing = perte de temps.
- mieux gérer les API (cours API) pour le scrapping.
- Mieux identifier nos target et resserrer le dataset autour.
- Keras : paramétrage.

Prochaines étapes d'amélioration :

- Standardisation des Noms Prénoms / clé unique d'ID.
- Données prélevées à l'entraînement des athlètes.
- Coefficient de confiance à affiner.
- Pipelines de préprocessing pour gagner du temps CPU.
- Davantage de Deep Learning.
- Création d'une API pour récupérer les données et nourrir le modèle.
- Conteneurisation pour production.