Projet de Fin d'Étude (PFE)

Analyse et prédictions de matchs de la National Basketball Association (NBA) via un dashboard interactif

COYLA Timoteï

Sommaire

I. Introduction

➔ Pourquoi avoir choisi un dataset de la NBA ?

Figure 1: Logo NBA

II.A Fonctionnement de la NBA

Figure 2: Logos des équipes NBA

- ➔ 82 matchs de saison régulière
- ➔ Deux conférences : Est et Ouest
- ➔ 30 équipes réparties sur les deux conférences
- ➔ Les Playoffs

II.B Le jeu de données

Le jeu de données est composé de 4 fichiers au format csv

games

- ➔ Une ligne par match joué
- ➔ Contient le nombre de points, rebonds, passes, etc. pour l'équipe à domicile et à l'extérieur
- ➔ Contient une colonne indiquant quelle équipe a gagné (domicile ou extérieur)

games_details

- ➔ Une ligne par joueur et par match
- ➔ Contient les statistiques de chaque joueur pour chaque match

teams

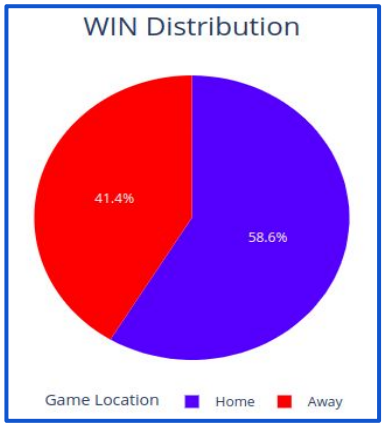
- ➔ Contient toutes les informations de chaque équipe, leur nom, leur capacité de salle, etc.

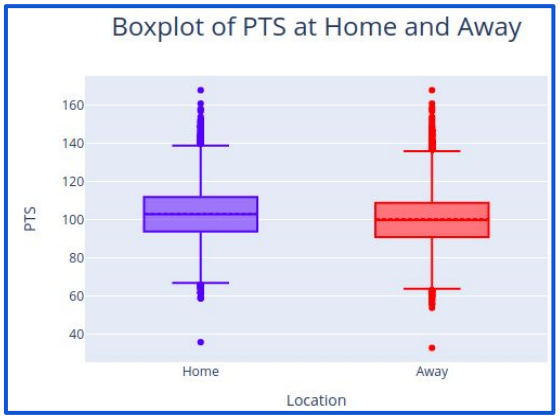
ranking

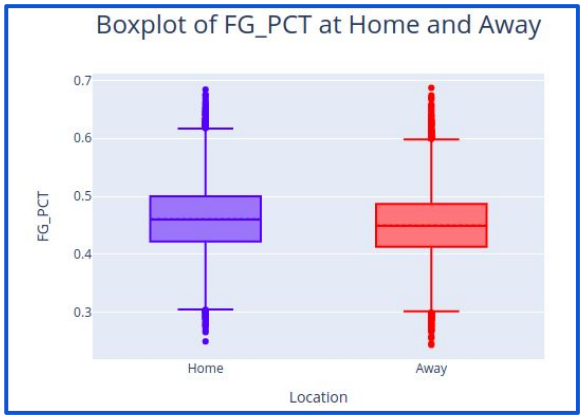
- ➔ Contient les informations de victoire pour chaque équipe et chaque saison
- ➔ Contient une colonne indiquant la conférence de l'équipe

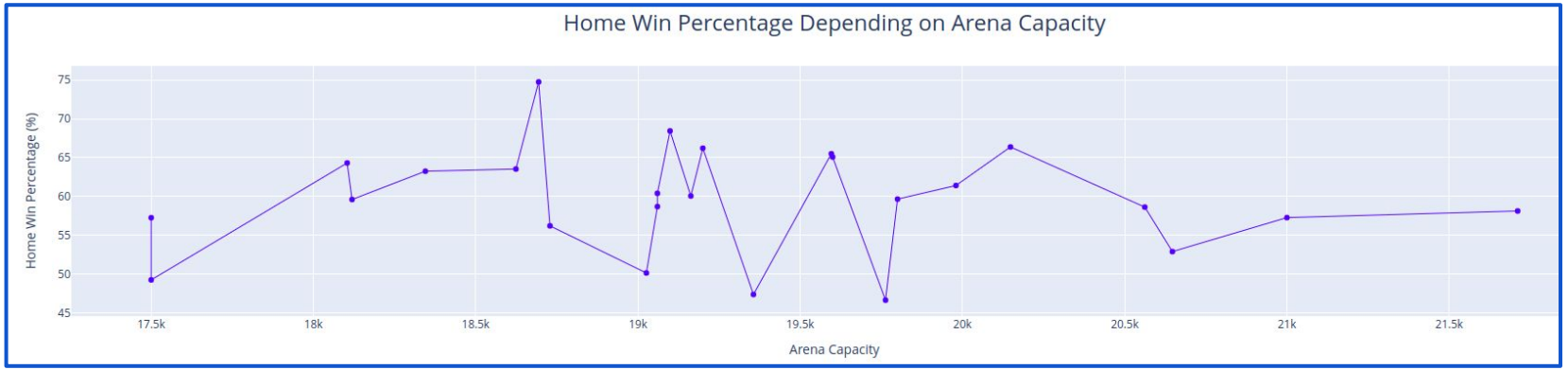
II.C Analyse des variables

Analyse de l'influence de la localisation du match (domicile / extérieur) sur les performances des équipes



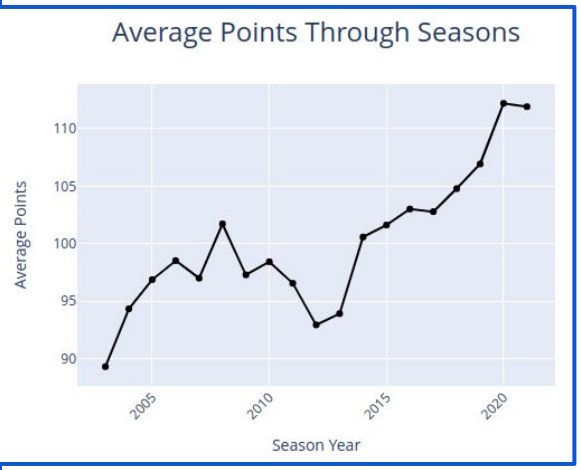


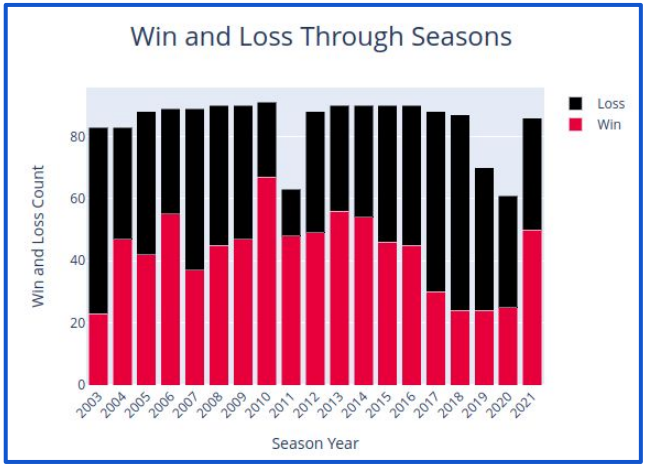


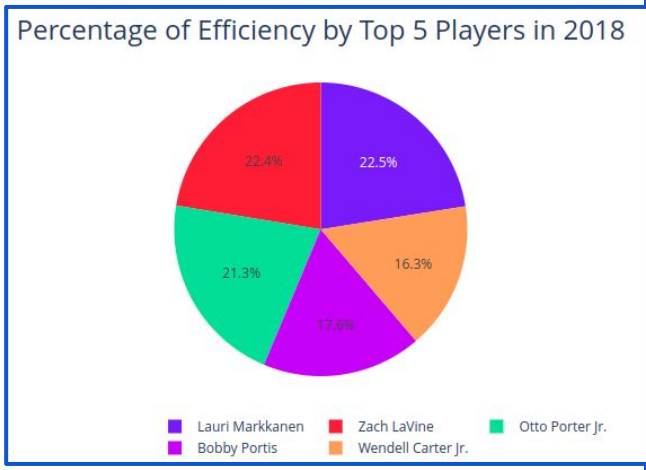


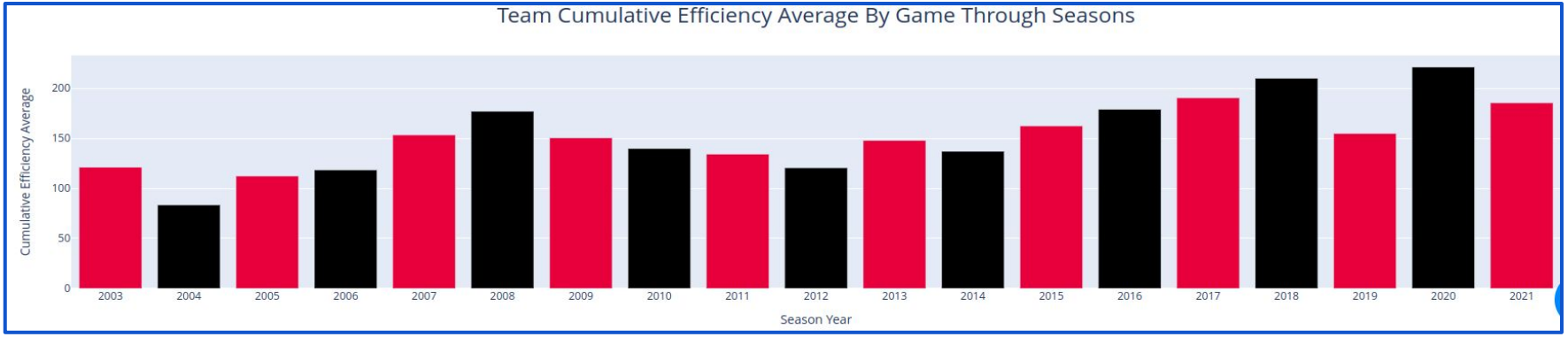
II.C Analyse des variables

Analyse des statistiques des équipes et des joueurs qui les constituent









III.A Choix des variables d'entrées

- ➔ Inspiration de travaux réalisés et mis à disposition sur kaggle (lien dans bibliographie), l'auteur a choisi comme variables d'entrées les statistiques des équipes (points, rebonds, pourcentage au tir, etc.)

- ➔ Le but du modèle de machine learning est de prédire le nombre de victoire de chaque équipe pour une saison régulière donnée en entraînant le modèle sur la saison précédente

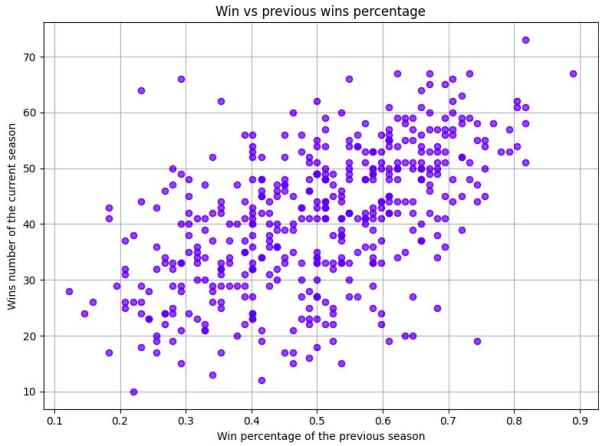
- ➔ Première approche "naïve" en choisissant une seule variable d'entrée : le pourcentage de victoire de l'année précédente

- ➔ Deuxième approche en ajoutant l'efficacité de l'équipe calculée et le plus minus de la saison précédente

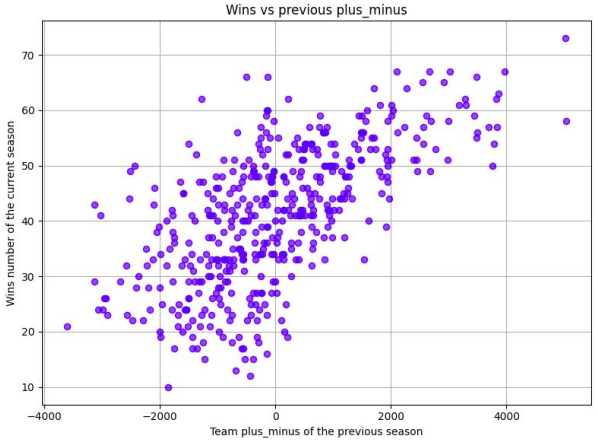
III.B Création des modèles de prédiction

➔ 5 modèles différents ont été créés afin d'en comparer les résultats en se basant sur la Root Mean Squared Error (RMSE)

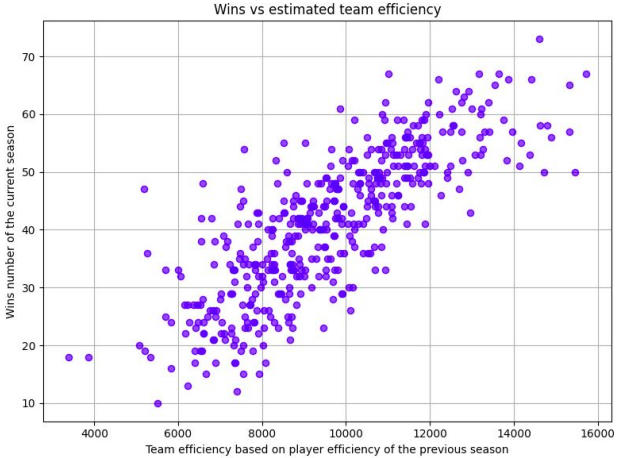
➔ Étude de corrélation entre les variables d'entrées et celle de sortie :



Corrélation : 0.545



Corrélation : 0.648

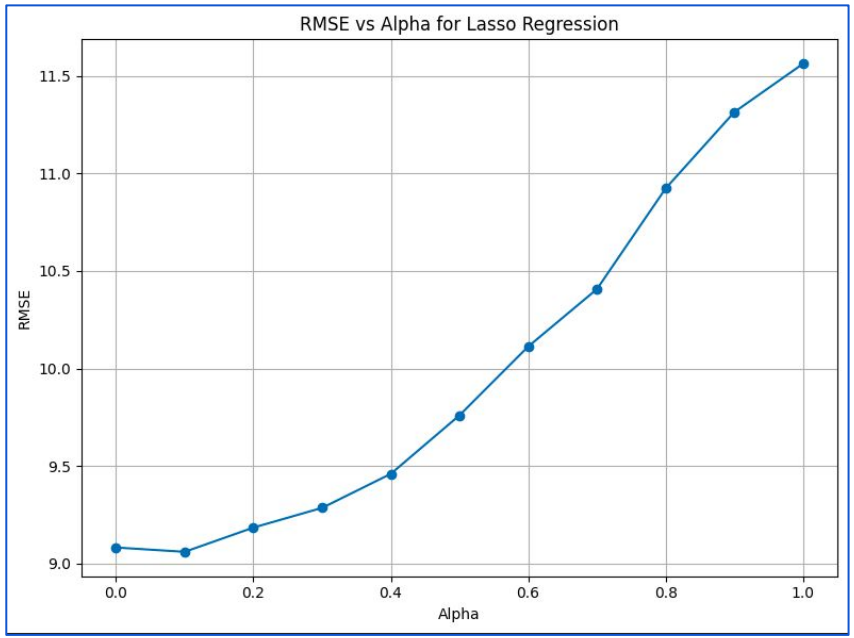


Corrélation : 0.804

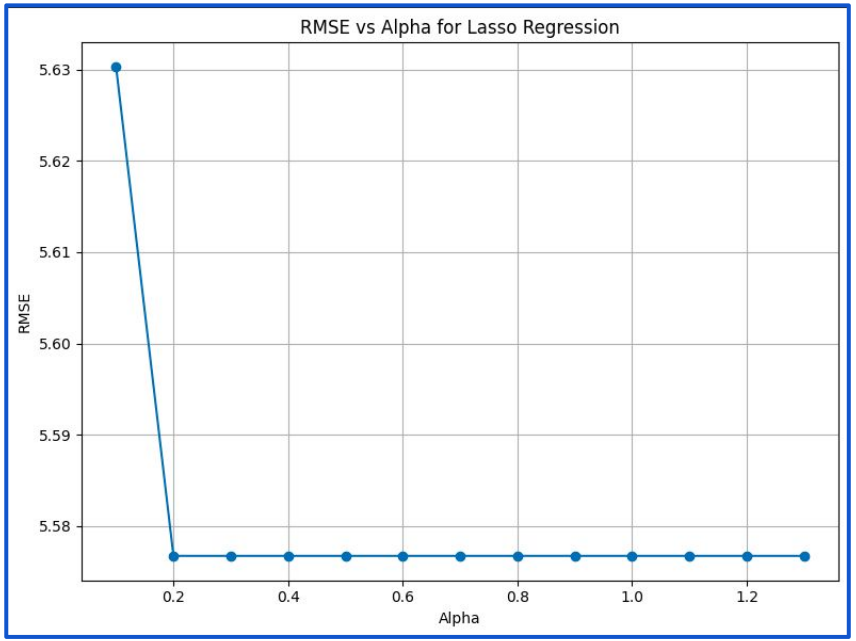
III.B Création des modèles de prédiction

➔ Régression linéaire

➔ Régression lasso avec le facteur de “pénalité” alpha



1ère approche

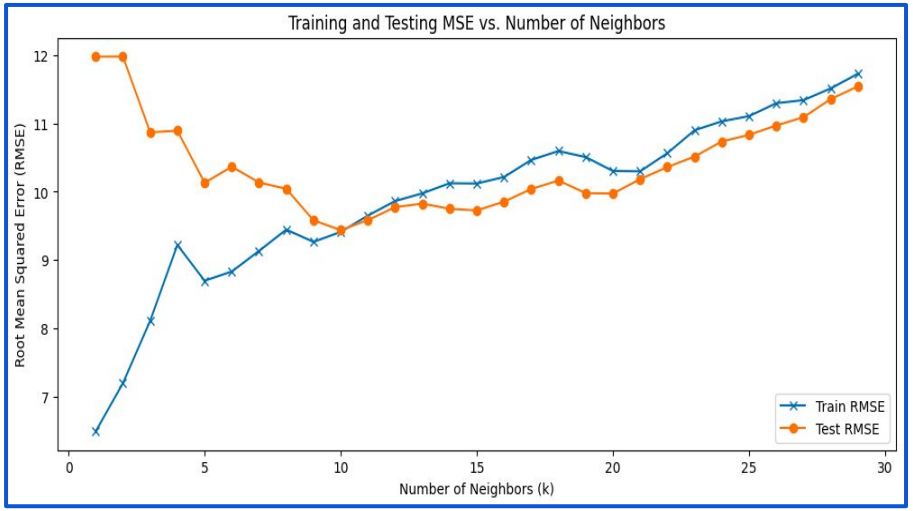


2nde approche

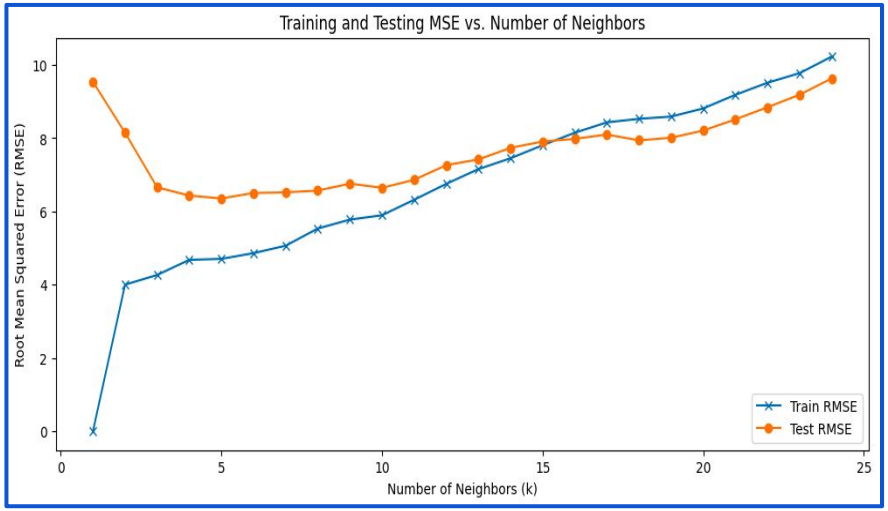
III.B Création des modèles de prédiction

➔ Forêt aléatoire : ajustement des hyperparamètres avec GridSearch (le nombre d'arbres, leur profondeur, le nombre d'éléments par noeud et par feuille)

➔ Algorithme des k plus proches voisins (KNN) :



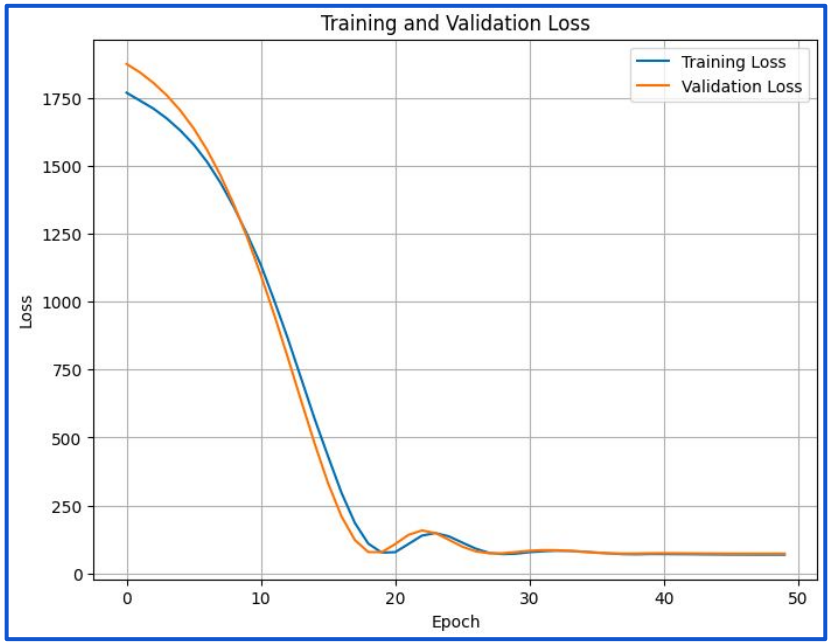
1ère approche



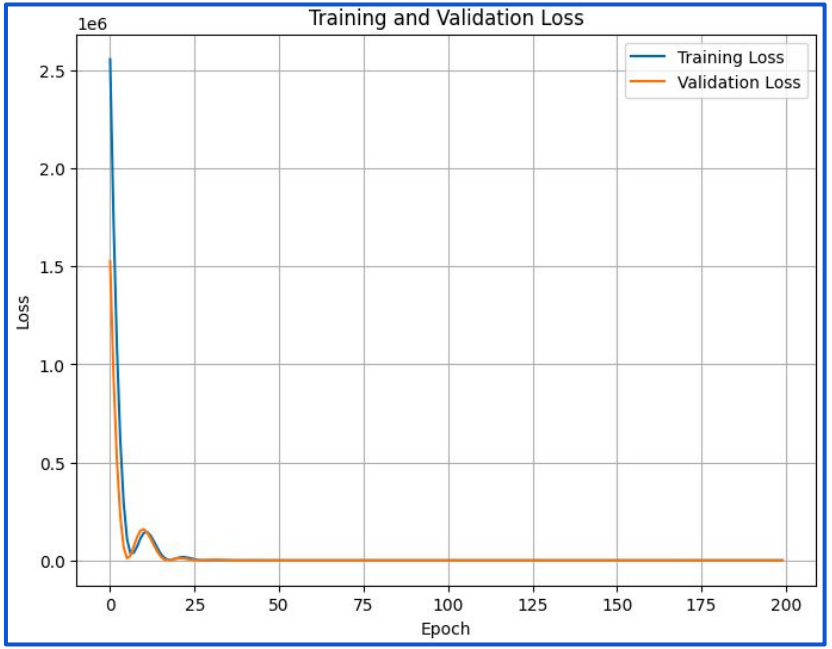
2nde approche

III.B Création des modèles de prédiction

➔ Réseau de neurones (entraînement sur les deux dernières années) : deux couches cachées de 32 neurones avec fonction d'activation ReLu



1ère approche :
Loss ~= 70



2nde approche :
Loss ~= 50

III.C Analyse et comparaison des résultats

➔ Avec la première approche

	Régression linéaire	**Régression lasso**	**Forêt aléatoire**	**KNN**	**Réseau de neurones**
RMSE	9.082	9.061	9.856	9.439	9.365
Kaggle **RMSE**	9.891	9.833	10.362	Not Done	Not Done

Figure 3: Tableau comparatifs des résultats

➔ Avec la seconde approche

	Régression linéaire	**Régression lasso**	**Forêt aléatoire**	**KNN**	**Réseau de neurones**
RMSE	6.234	5.187	6.083	6.355	5.219
Kaggle **RMSE**	9.891	9.833	10.362	Not Done	Not Done

Figure 4: Tableau comparatifs des résultats

IV.A Les librairies Plotly et Dash

➔ Plotly : Librairie graphique Python open source pour créer des graphiques interactifs

Figure 5: Logo Plotly

➔ Dash : Framework Python permettant de créer des applications web interactives

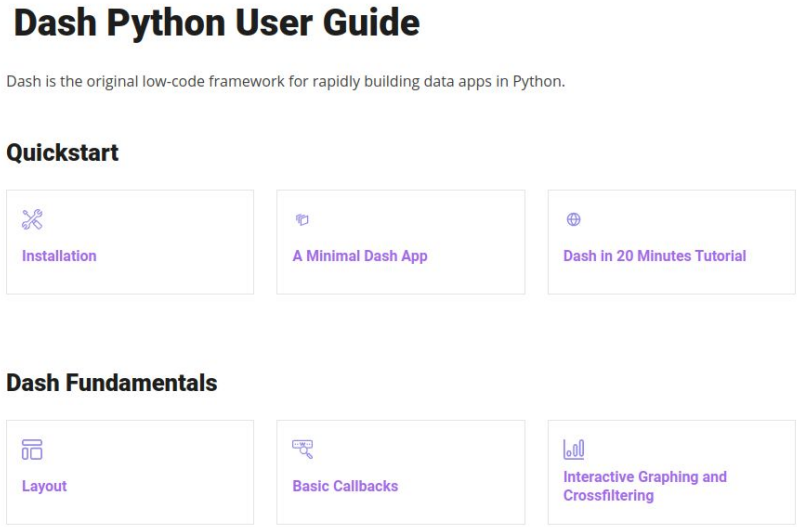
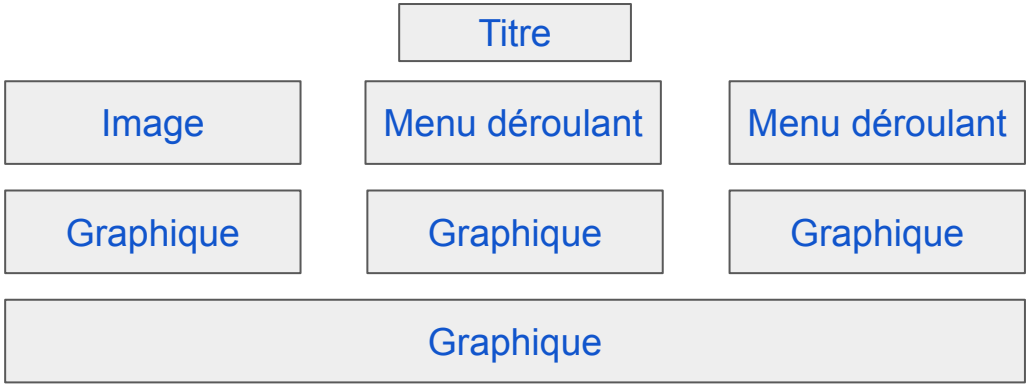


Figure 6: Guide d'utilisateur de dash en Python

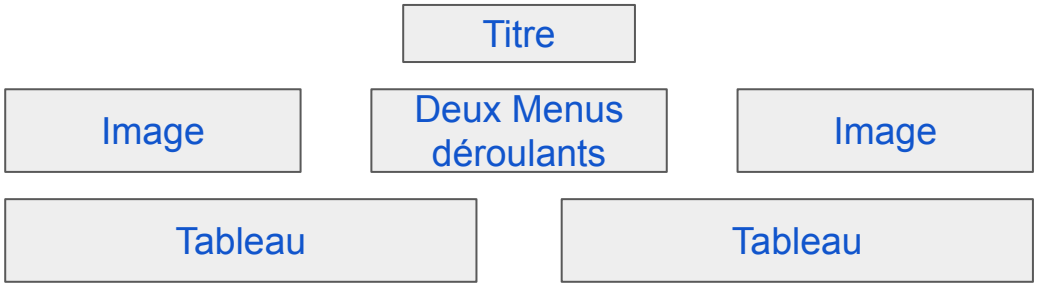
IV.B La structure du dashboard

➔ Deux pages : Visualization et Prediction

➔ Visualization Page :



➔ Prediction Page :



IV.C Démonstration

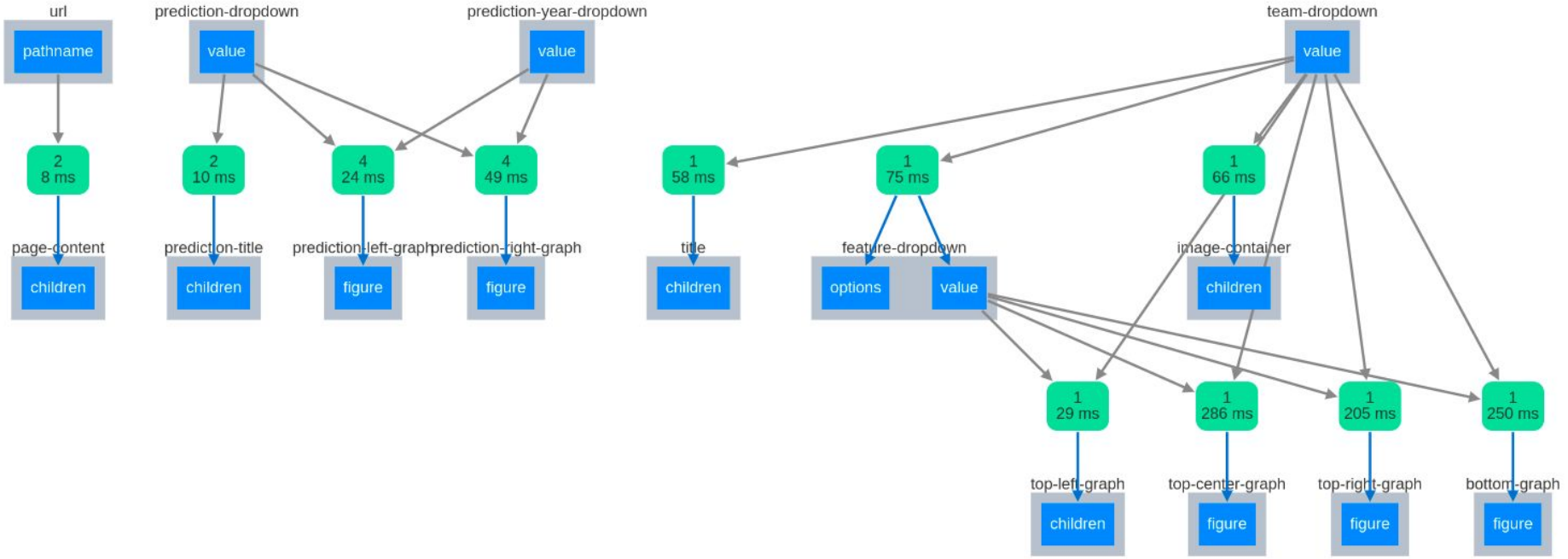


Figure 7: Graphique des Callbacks

V.A Améliorations

➔ Prendre en compte les retours de blessures

➔ Prendre en compte les joueurs “rookies”

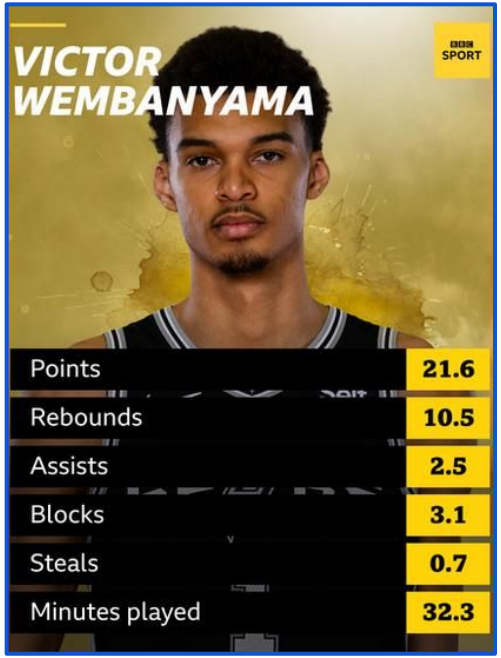


Figure 8: Image du rookie Français Victor Wembanyama

V.B Travaux futurs

➔ Intégration du dashboard et des modèles de prédiction à la Google Cloud Platform (GCP)

Figure 9: Image de déploiement de dash sur la GCP

VI. Conclusion

Merci pour votre écoute !

VII. Références

- Figure 1 : https://www.andyedge.com/articles/nba-logo
- Figure 2 : https://fr.dreamstime.com/illustration/nba-logos.html
- Figure 3 : Réalisé dans Google Slides
- Figure 4 : Réalisé dans Google Slides
- Figure 5 : https://dash.plotly.com/
- Figure 6 : https://dash.plotly.com/
- Figure 7: Réalisé avec dash
- Figure 8 : https://www.bbc.com/sport/africa/67156801
- Figure 9 : https://datasciencecampus.github.io/deploy-dash-with-gcp/
- Jeux de données : https://www.kaggle.com/datasets/nathanlauga/nba-games
- Travaux déjà réalisés sur le dataset : https://www.kaggle.com/code/hqfang/drp-nba-game-wins-prediction
- Autres images et graphiques : Réalisés dans le notebook ou le dashboard