

PROJET GAMING

PRÉDICTION DE VICTOIRE SUR LEAGUE OF LEGENDS

Clément Auclin
Félix Doublet
Vincent Gallot
Paul Trincklin

École Centrale Nantes
Option INFO-IA
2023

SOMMAIRE

- 1. Contexte du projet**
- 2. Collecte des données**
- 3. Étude des données**
- 4. Comparaison des modèles**
- 5. Reconnaissance d'images**
- 6. Recommandation de champions**



A dark silhouette of a knight in full armor, including a helmet with a plume, stands in a dynamic pose. He is holding a large shield in his left arm and a sword in his right hand, which is drawn back as if ready to strike. The knight is positioned on the left side of the slide, set against a background of stylized blue waves.

10

CONTEXTE DU PROJET

Objectifs du projet

- Apprendre à utiliser une API pour récolter des données.
 - Définir un problème de machine learning à résoudre sur un jeu multijoueur.
 - Réaliser du feature engineering, manipuler et transformer des jeux de données.
 - Étudier et comparer des modèles de machine learning et deep learning sur un problème concret.
 - Apprendre à utiliser différentes bibliothèques python autour de la data science.
 - Définir par nous-même des objectifs intéressants en fonction de nos avancées.
- ⇒ Objectif : développer un prototype simple ou une preuve de concept d'outil d'aide aux joueurs. (exemple : recommandation d'un champion pour maximiser les chances de victoire dans une partie)

Outils utilisés

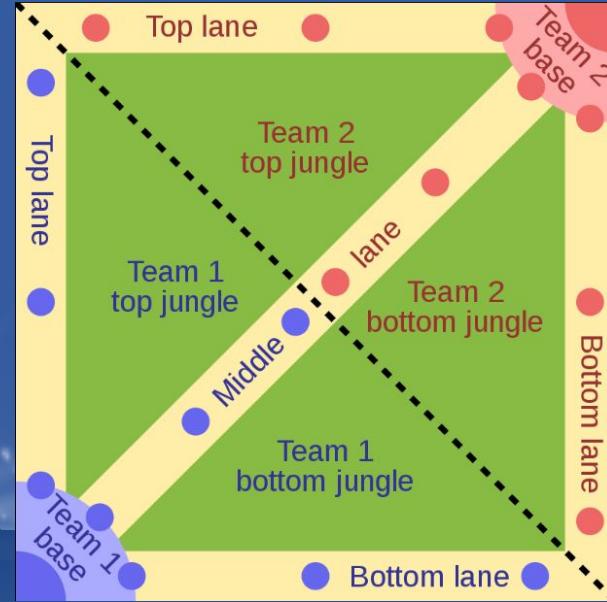
- Utilisation d'une API Rest: exemple de celle de riot games.
- Utilisation des bibliothèques Python de manipulation des données : pandas, numpy, requests, html, json, et de visualisation : matplotlib, seaborn.
- Utilisation de bibliothèques de machine learning: sklearn, keras.
- Utilisation d'outils collaboratifs : git et github, jupyter notebooks.



Présentation rapide de LoL : sélection des champions



Présentation rapide de LoL: carte et partie



Présentation rapide de LoL: elo et ranks



Ranked Solo

Challenger 1,057 LP 76W 53L 59% Win Rate

Match History

All Matches ▾ Search Champion or Played With... Q

60% WR 4.18 KDA 100% (IW 0L) 78% (7W 2L) 50% (4W 4L)

Last 20 games 7.8 3.7 7.7 6.00 KDA 6.56 KDA 3.71 KDA

Ranked Flex Unranked

Ranked Solo 16 hours ago 7 / 5 / 6 2.60 KDA 179 CS (6.3) 39 vision A Yhwach where my ... Boldog Cs... Agurin elmanic ? NORD Wo... FNC Hylo... pdfile janna

▼ 14 LP LOSS 28:26

Ranked Solo 17 hours ago 7 / 1 / 7 14.00 KDA 127 CS (5.6) 39 vision IHG Polski... Skenz Sabez Lucifer Pandä

▲ 23 LP WIN 22:41

Ranked Solo 17 hours ago 5 / 8 / 9 1.75 KDA 120 CS (4.8) 19 vision Have a nill... spoiled fruit Albetrayber Agurin Treasure ... Don Noway KHK TLamp cale cycle ... g4444mln... Azapp

▼ 13 LP LOSS 25:04

Ranked Solo 18 hours ago 7 / 0 / 5 Perfect KDA 128 CS (6.8) 31 vision Albetrayber FaithXD PMP DAD... AL KAIDA ... 5 PDFs ON... Lathyrus

▲ 22 LP WIN 18:47

Ranked Solo 18 hours ago 14 / 5 / 17 6.20 KDA 172 CS (5.2) 52 vision Darth Vader Jig Gap Mb many11 in ... Fozyfoster Go toucou... yourtch1 2023 SOL... Batuu

▼ 13 LP WIN 32:58

Recently Played With Last 20 games



02

COLLECTE DES DONNÉES



Bref état de l'art

Ce problème a déjà été testé avec différentes méthodes:

- Using Machine Learning to Predict Game Outcomes Based on Player-Champion Experience in League of Legends (75.1%)
- Predicting Outcome of League of Legend Ranked games in ChampSelect via Machine Learning (74.2%)
- Win Prediction in Multiplayer Esports: Live Professional Match Prediction (77.5%) (Dota 2)
- Predicting League of Legends Ranked Games Outcome (~79%)

Ces différents résultats nous ont permis d'établir un objectif pour le projet (70%) et de pouvoir voir quelles features pouvaient être intéressantes, ainsi que les modèles les plus efficaces (XGBoost et Random Forest).

Récolte de données brutes: API riot

LEAGUE-V4

GET /lol/league/v4/challengerleagues/by-queue/{queue}

GET /lol/league/v4/entries/by-summoner/{encryptedSummonerId}

GET /lol/league/v4/entries/{queue}/{tier}/{division}

GET /lol/league/v4/grandmasterleagues/by-queue/{queue}

GET /lol/league/v4/leagues/{leagueId}

GET /lol/league/v4/masterleagues/by-queue/{queue}

Données récupérées : données de partie (gagnant, champions, rôles), données de joueur (niveau, elo, maîtrise), et historique des joueurs sur les 5 dernières parties (nombre de victoires, KDA moyen)

LEAGUE-V4

GET /lol/league/v4/challengerleagues/by-queue/{queue}

[Jump to Inputs](#)

RESPONSE CLASSES

Return value: LeagueListDTO

LeagueListDTO

NAME	DATA TYPE
leagueId	string
entries	List[LeagueItemDTO]
tier	string
name	string
queue	string

Récolte de données brutes: web scraping

Name	Winrate	Best partner	Counters ...	Is countered by ...
Aatrox Top	47.3%	Anivia +6.7%	Akali +13.0%	Irelia -7.9%
Ahri Mid	48.6%	Rakan +6.5%	Vladimir +4.9%	Akshan -6.8%
Akali Mid, Top	47.5%	Maokai +7.2%	Syndra +4.7%	Diana -8.1%
Akshan Mid, Top	52.2%	Zac +8.4%	Katarina +7.0%	Anivia -6.9%
Alistar Support	49.7%	Syndra +5.8%	Yuumi +7.9%	Janna -8.0%

Les données sont abondantes sur les sites d'aide aux joueurs : on peut les récupérer à l'aide de la librairie *requests* de python, sur des sites comme u.gg ou mobachampion.com.

Rank	Champion	Win Rate	KDA	LP Gain	Max Kills	Max Deaths	CS	Damage	Gold	...	▲	▼	◆	◆
1	Elise	70% / 32W 14L	3.88 7.5 / 3.9 / 7.7	+ 607LP	14	10	149	20,047	11,570	33	1	-	-	-
2	Wukong	59% / 20W 14L	3.26 6.1 / 4.1 / 7.2	+ 239LP	18	9	157.4	15,973	11,154	28	2	1	-	-
3	Kha'Zix	53% / 8W 7L	2.76 5.5 / 4.7 / 7.3	+ 130LP	14	9	158.9	16,710	10,411	3	-	-	-	-
4	Evelynn	64% / 7W 4L	3.55 8.5 / 4.0 / 5.6	+ 75LP	19	14	132.4	16,428	9,815	11	2	-	-	-
5	Jarvan IV	38% / 3W 5L	3.21 6.8 / 6.5 / 14.1	+ 26LP	12	9	179	18,999	12,797	3	1	-	-	-

On a ainsi pu récupérer des données sur les taux de victoire des champions, les performances des joueurs sur les champions et leur historique de victoire.

Données mises en forme

Les données ont été mises en forme de manière pour pouvoir être utilisées dans des tâches de machine learning. On dispose, pour chaque partie, d'une valeur Y indiquant le résultat de la partie que l'on cherche à prédire à partir de 18 variables, dupliquées pour chacun des 10 joueurs de la partie, formant un tableau X de 180 features. Les données sont de multiples types qu'il faut transformer en données numériques.

Par ailleurs, nous avons récupéré 2887 parties, limité par l'API de riot (nous pouvions récupérer environ une partie par minute).

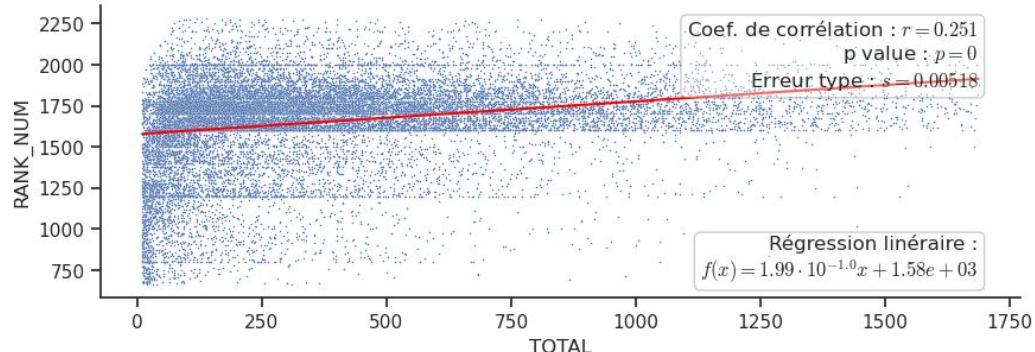
Features	signification	Source	Type
Y	Résultat de la partie	API (direct)	bool
CHAMP	Champion joué sur la partie	API (direct)	string
LVL	Niveau d'invocateur	API (direct)	int
TOTAL	Nombre de parties jouées sur la saison	API (direct)	int
GWR	Winrate en classé sur la saison	API (direct)	float
VET	Attribut vétéran du joueur	API (direct)	bool
RANK	Rang en classé solo/duo du joueur (elo)	API (direct)	list
HOT	Attribut "série de victoires du joueur"	API (direct)	bool
KDAG	KDA moyen sur les 5 dernières parties (tous champions)	API (5 games)	list
KDA	KDA moyen sur les 5 dernières parties (champion de la partie)	API (5 games)	list
WR	Winrate moyen sur les 5 dernières parties (champion de la partie)	API (5 games)	float
NB	Nombre de partie jouées sur le champion sélectionné parmi les 5 dernières	API (5 games)	int
FILL	Le joueur joue-t-il sur le même poste que ses 5 dernières parties (autofill)	API (5 games)	bool
VS	Winrate moyen sur le matchup entre les 2 champions d'un même poste	mobachampion	float
MAS	Niveau de maîtrise sur le champion joué	API (direct)	int
WRCH	Winrate de la saison sur le champion de la partie	scraping u.gg	float
WCH	Victoires de la saison sur le champion de la partie	scraping u.gg	int
LCH	Défaites de la saison sur le champion de la partie	scraping u.gg	int
TOTCH	Total de parties jouées dans la saison sur le champion de la partie	scraping u.gg	int



03

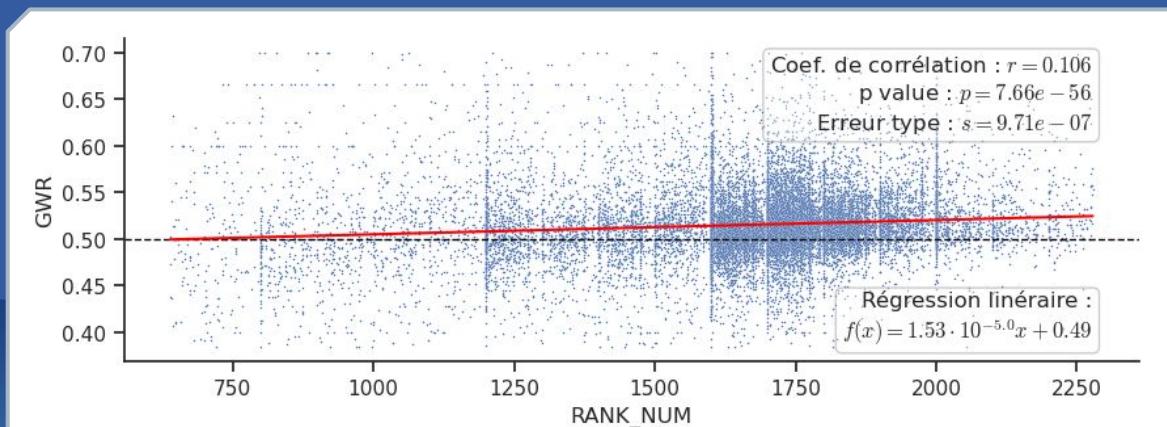
ÉTUDE DES DONNÉES

ANALYSE DE CORRÉLATION

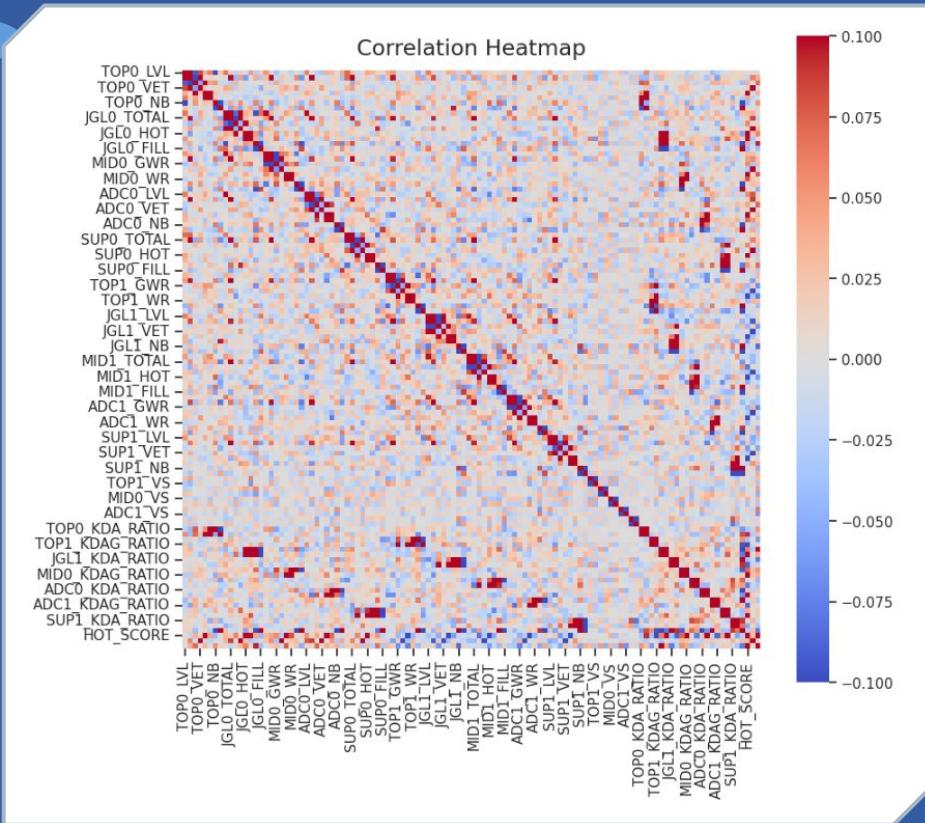


“Les joueurs qui font plus de parties ont tendance à être mieux classés”

“Les joueurs mieux classés ont tendance à gagner plus souvent”



MATRICE DE CORRÉLATION



RÉGRESSION LINÉAIRE MULTIPLE

Objectif :

Permet d'identifier les relations entre les variables et la donnée à prédire.
Les valeurs les plus importantes sont susceptibles de faire de bonnes features.

RÉGRESSION LINÉAIRE MULTIPLE

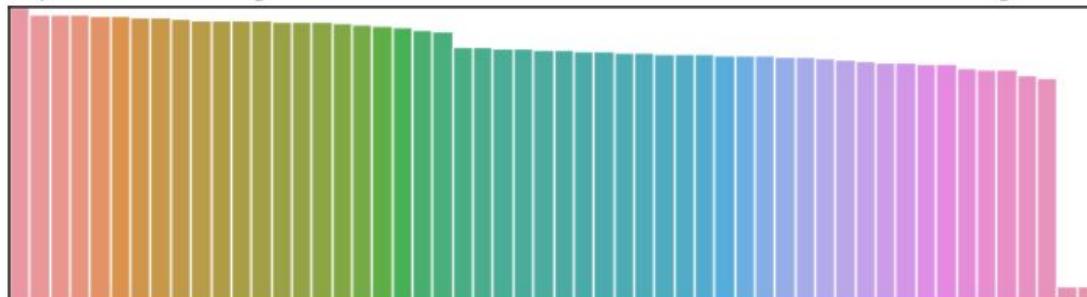
MODÈLE LASSO

HOT_SCORE	:	0.0383
JGL0_KDA_RATIO	:	0.0148
SUP0_NB	:	0.014
JGL1_NB	:	-0.0137
JGL0_NB	:	0.0121
ADC0_VS	:	0.0115
TOP1_KDA_RATIO	:	-0.00954
TOP0_NB	:	0.00909
ADC1_NB	:	-0.00888
MID0_KDA_RATIO	:	0.00735
SUP1_NB	:	-0.00715
JGL0_VS	:	0.00642
JGL1_KDA_RATIO	:	-0.00627
FILL_SCORE	:	-0.00615
TOP0_VS	:	0.00594
MID1_KDA_RATIO	:	-0.00518
TOP0_KDA_RATIO	:	0.00483
MID0_VS	:	0.00376
MID1_NB	:	-0.00324
ADC0_NB	:	0.00257
TOP1_NB	:	-0.0017
SUP0_VS	:	0.00156
MID1_RANK_NUM	:	-0.000175
ADC1_RANK_NUM	:	-0.000172
JGL0_RANK_NUM	:	0.00014
SUP1_RANK_NUM	:	-0.000134
TOP0_RANK_NUM	:	0.000127

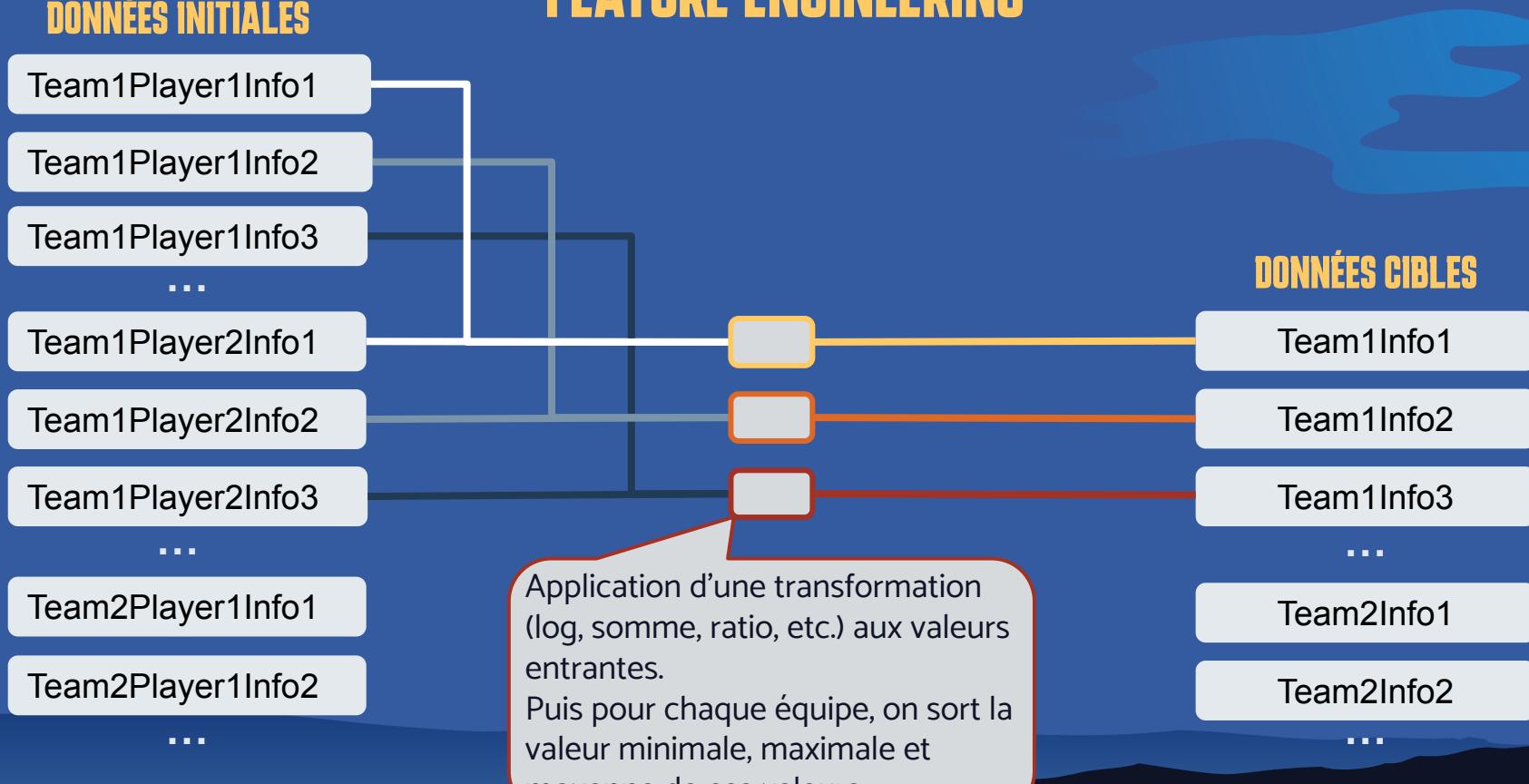
Représentation de la valeur absolue des coefficients non nuls de la régression



Représentation du logarithme de la valeur absolue des coefficients non nuls de la régression



FEATURE ENGINEERING

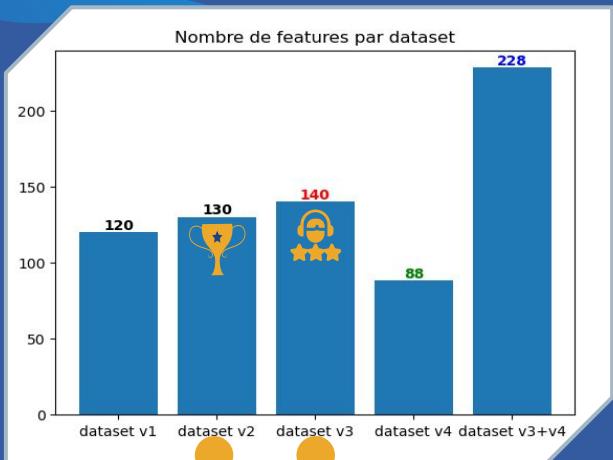


04

COMPARAISON DES MODÈLES

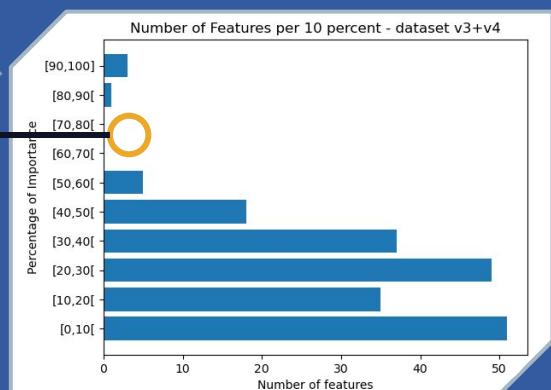
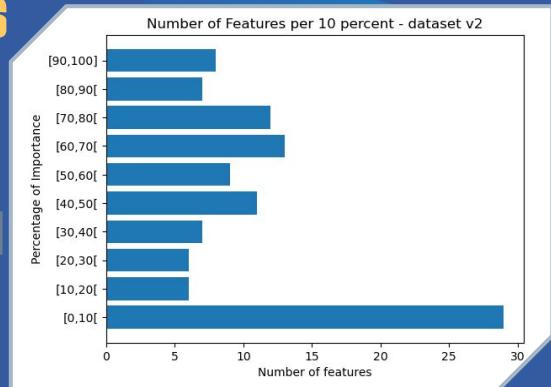


Statistiques sur les features



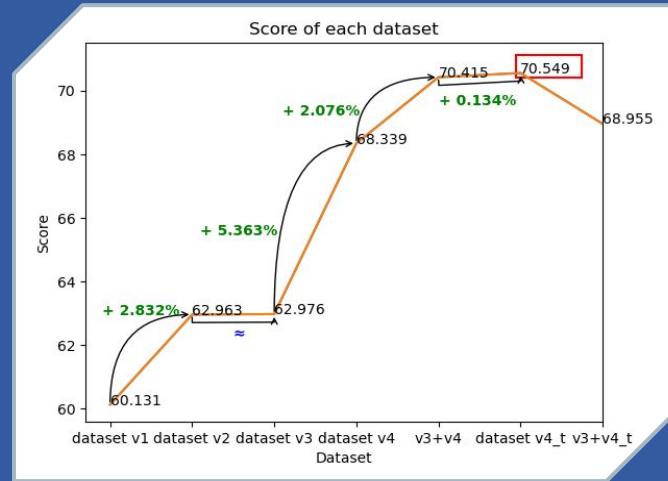
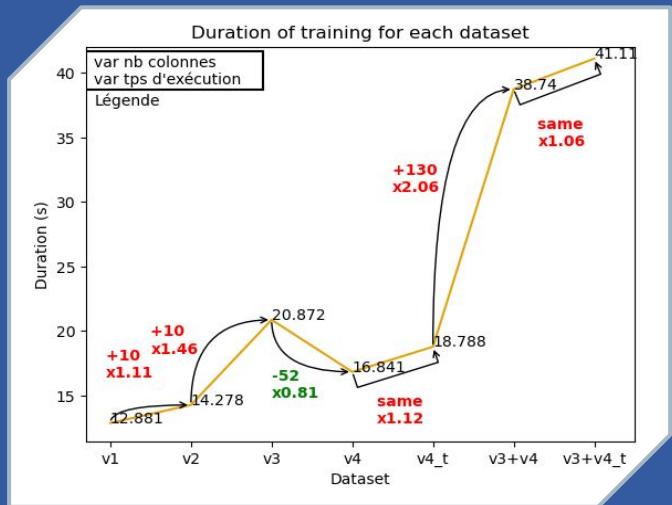
- ajout des "MAS" : points de maîtrise sur le champion pour chaque joueur
- ajout des "VS" : winrate moyen du champion par rapport à l'ensemble des champions de l'autre équipe

Delta important entre les différentes features



Comparaison des scores et des temps de train

- 1
- 2
- 3



Pour les calculs :

- Utilisation de GridSearchCV
- 15 folds
- Modèle GradientBoostingClassifier

Modèle final retenu

1 2 3

XGBOOST

PRÉDICTIONS



70.5%

CORRECTES



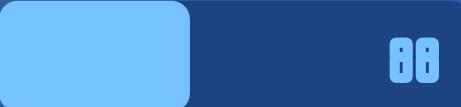
29.5%

FAUSSES

CARACTÉRISTIQUES

Caractéristiques principales du modèle

Nb de Features



88

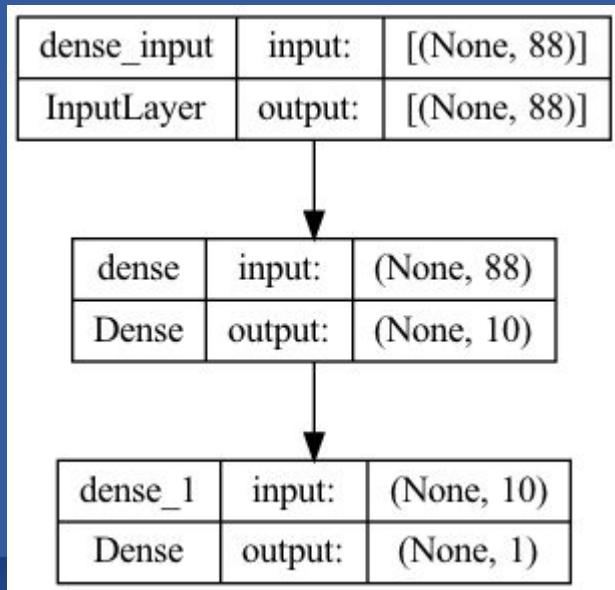
Temps de train sur le dataset



16.8s

COMPARAISON : DEEP LEARNING

Une première approche simple



Couche d'entrée :
88 features

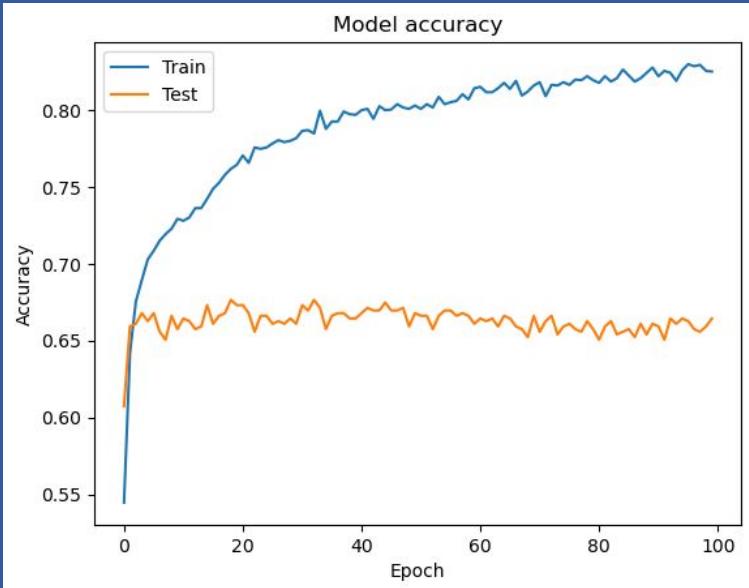
Couche cachée

Couche de sortie :
prédiction binaire

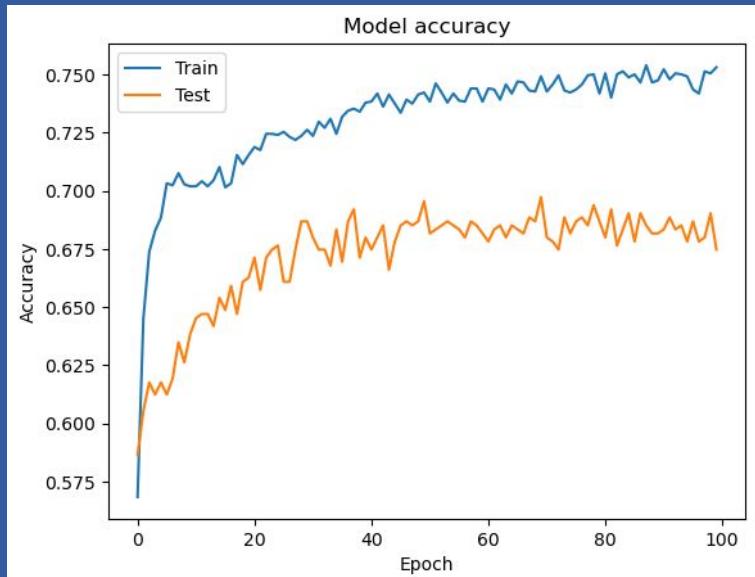


Comparaison des scores

Sans régularisation

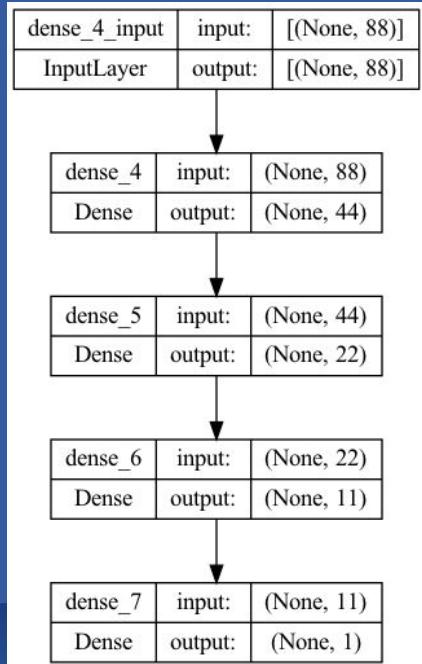


Avec régularisation L1



COMPARAISON : DEEP LEARNING

Seconde approche



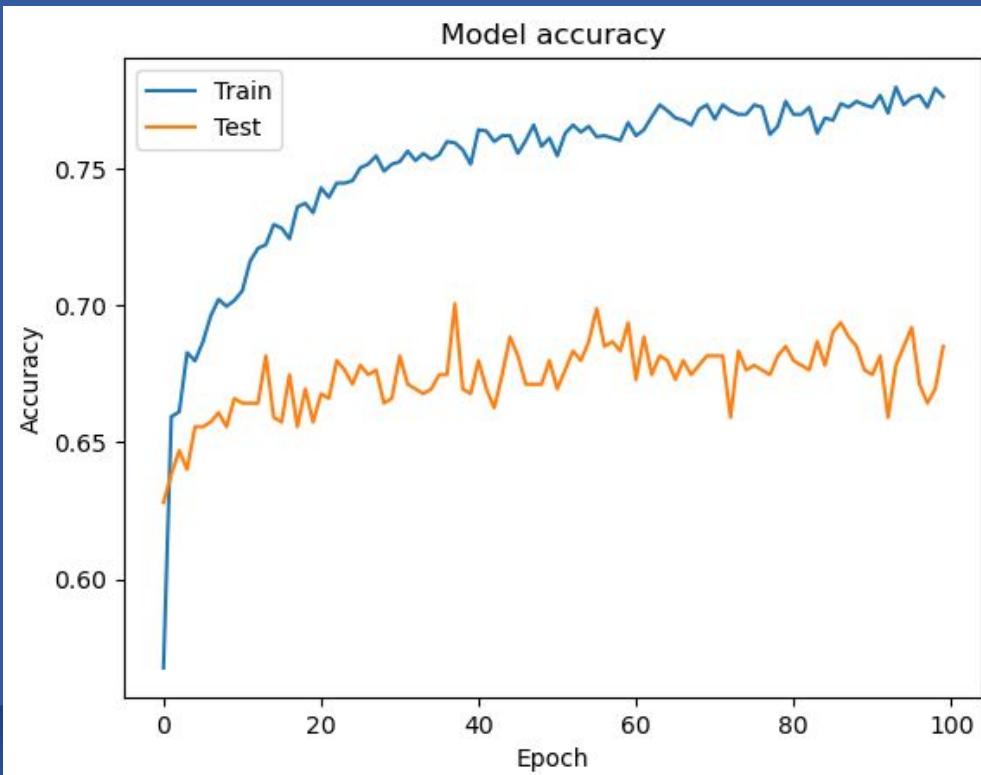
Couche d'entrée :
88 features



Couche de sortie :
prédiction binaire



Résultats obtenus



CONCLUSION DEEP LEARNING

→ La précision peut varier de manière importante entre deux epochs (+/- 2 %) :

- La régularisation L1 permet de contrer en partie l'overfitting
- Un modèle plus complexe n'améliore pas vraiment les résultats





05

RECONNAISSANCE D'IMAGES

Détection des images



Récupération des bannissements

Zone de récupération des picks des joueurs

→ Utilisation des proportions du client de League of Legends

Création du dataset



→ **Data Augmentation sur les images originales**

- Variation de qualité
- Rognage
- Variation de luminosité
- Effet de flou

162 images ➤ 28 188 images

Modèle utilisé et résultats

→ Utilisation d'un FCN (Fully Convolutional Network)

- Permet d'utiliser des images de différentes tailles
- 23 couches
- +1 600 000 paramètres pouvant être entraînés



Résultats obtenus

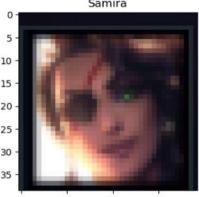
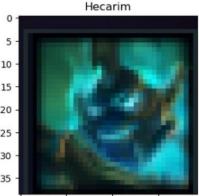
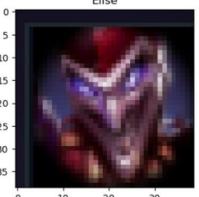
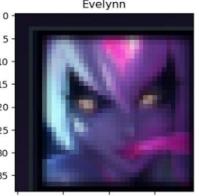
top 5 **73.08%** Picks

top 1 **46.15%** Picks

top 5 **72.41%** Bans

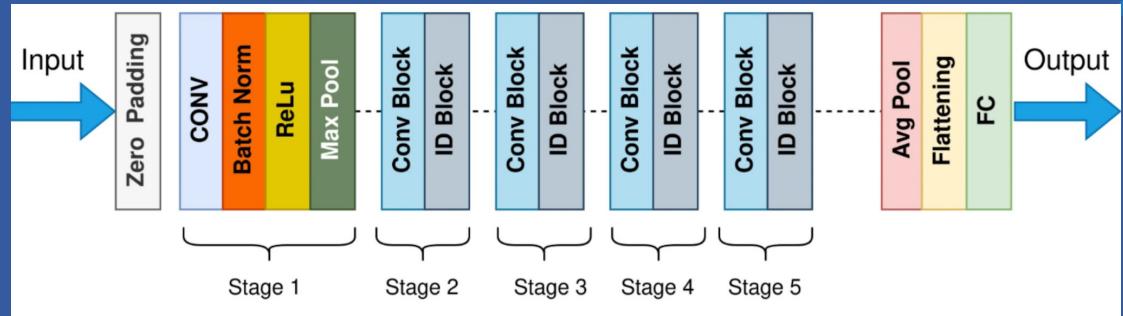
top 1 **51.72%** Bans

Exemple de prédictions :



TRANSFER LEARNING : RESNET50

CNN à 50 couches cachées :
- Entraîné sur 14 millions d'images



Transfer Learning : Dégeler les deux dernières couches et rajouter une couche de 162 neurones + softmax pour prédire le champion

Résultats assez décevant -> 11% de bonnes prédictions



06

Recommandation de champions

Application : système de recommandation en temps réel

Objectif :

Pouvoir recommander un champion à un joueur durant la phase de sélection pour maximiser ses chances de victoire



INFORMATIONS DISPONIBLES



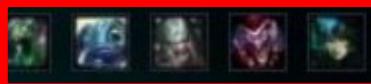
INFORMATIONS DISPONIBLES



ÉQUIPE ROUGE

INFORMATIONS DISPONIBLES

ÉQUIPE BLEUE



Champions bannis par les alliés

		Jhin	BOTTOM	Tot Swerve
		Viktor	MID	OneeChen
		Gangplank	TOP	Rase

Champions déjà verrouillés par les alliés

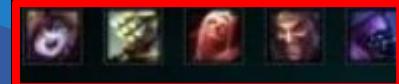
Joueur

		Picking...	JUNGLE	Leckoo
--	--	------------	---------------	--------

Champions “pré-verrouillés” par les alliés

		SUPPORT	Get Triggered	
--	--	----------------	---------------	--

Champions bannis par les adversaires



Champions déjà verrouillés par les adversaires

		AKALI	Summoner 1	
		IRELIA	Summoner 2	
		SHEN	Summoner 3	

Champions pas encore verrouillés par les ennemis

		PICKING NEXT	Summoner 4	
		PICKING NEXT	Summoner 5	

ÉQUIPE ROUGE

INFORMATIONS DISPONIBLES

ÉQUIPE BLEUE

Yuumi	Zed	Leona	LeBlanc	Swain	Elise
	BOTTOM	Summoner 1			
	iot Swerve				
	MID	Viktor	Summoner 2		
		OneeChen			
	TOP	Gangplank	Summoner 3		
		Rase			
	JUNGLE	Picking...	PICKING NEXT	Summoner 4	
		Leckoo			
	SUPPORT	Get Triggered	PICKING NEXT	Summoner 5	

Champion

Poste

Pseudo

Champion

AKALI
Summoner 1

IRELIA
Summoner 2

SHEN
Summoner 3

PICKING NEXT
Summoner 4

PICKING NEXT
Summoner 5

ÉQUIPE ROUGE

PIPELINE DE L'APPLICATION

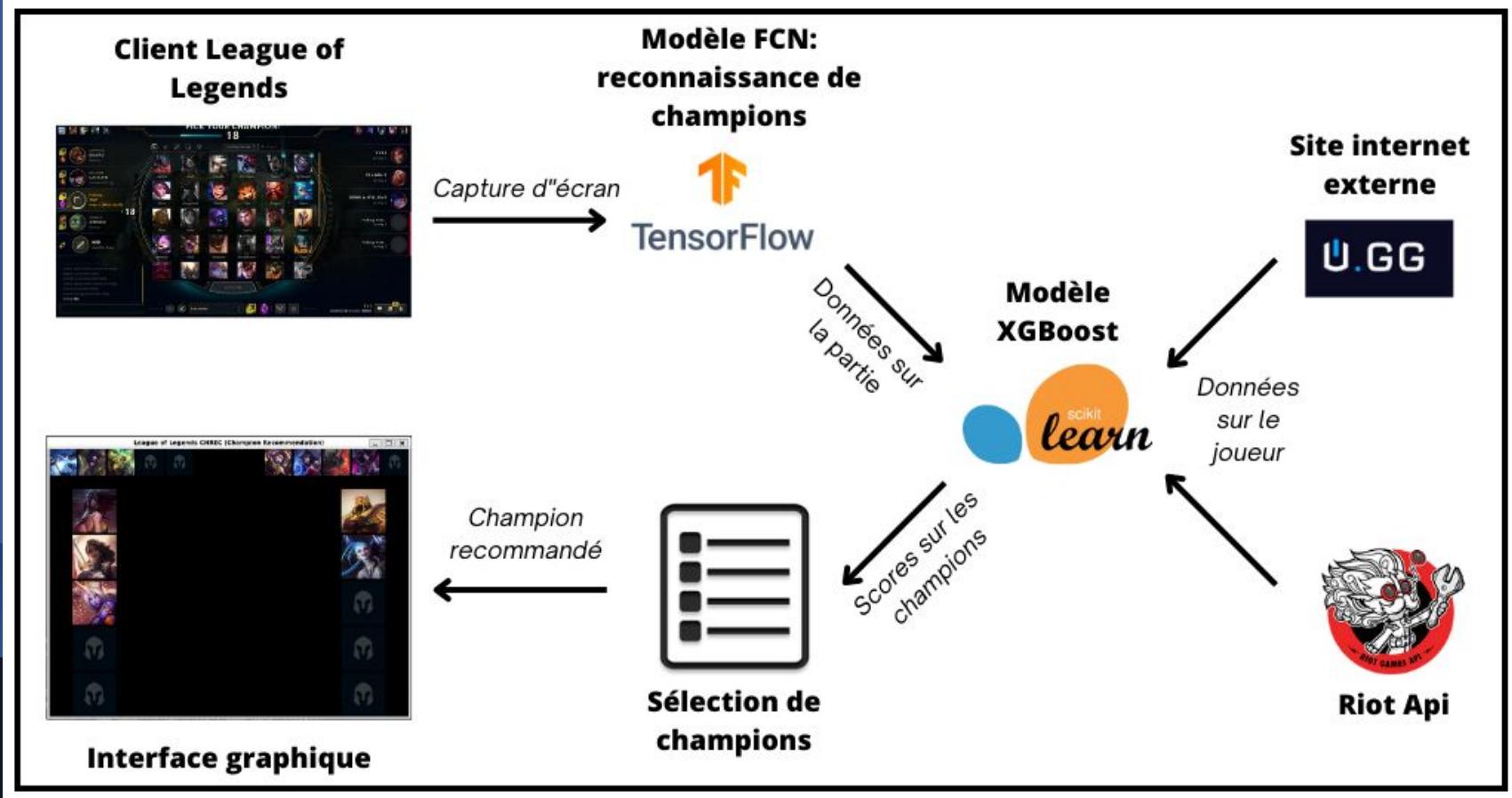
ÉQUIPE BLEUE

		Jhin BOTTOM Tot Swerve		
		Viktor MID OneeChen		
		Gangplank TOP Rase		
		Picking... JUNGLE Leckoo		
		SUPPORT Get Triggered		



		AKALI Summoner 1		
		IRELIA Summoner 2		
		SHEN Summoner 3		
		PICKING NEXT Summoner 4		
		PICKING NEXT Summoner 5		

Architecture de notre application





Merci de votre attention !

