



NATACHA BATMANABANE
NICOLAS BERLIOZ
LANA BONHOMME



PRÉSENTATION DU JEU

Avec trois races, quatre modes et une infinité de façons de jouer, *StarCraft II* représente l'expérience ultime de stratégie en temps réel.



1

CONSTRUISEZ VOTRE BASE



2

BÂTISSEZ VOTRE ARMÉE



3

À L'ASSAUT

NOTRE OBJECTIF

PRÉDIRE VOTRE LEAGUE



Bronze



Argent



Or



Platine



Diamant



Maitre



Grand Maitre

Comment déterminer la League d'un joueur à partir de ses statistiques ?

COMMENT ?



← LADDERS

Sparklepony 5600

Race Report

Map Reports

Mode: 1V1 Season: Current Season

Map Name	Games Played	Wins	Losses	Win %
Blackwater Gulch	25	12	13	48%
Sharkunus Plateau	10	5	5	50%
Nerazim Crypt	15	10	5	66%
Abyssal Caverns	88	44	44	50%
Scorched Crater	8	8	0	100%
Blackwater Gulch	3	1	2	33%
Tal'darah Aerie LE	22	11	11	50%
Kel'Naga Caverns	11	6	5	57%
Typhon Peaks	55	28	27	50%
Antiga Shipyard	23	12	11	55%

GRÂCE À VOS STATISTIQUES DE JEUX

COMMENT ÇA FONCTIONNE ?

1

Présentation du dataset

4

Modeling

2

Data pre-processing

5

API

3

Data visualization

6

Conclusion



PRÉSENTATION DU DATASET

La base de données contient 20 colonnes et 3395 lignes.

Les colonnes sont les suivantes :

GameID : Identifiant joueur

LeagueIndex

Age

HoursPerWeek

TotalHours

APM : Actions par minute

SelectByHotkeys : Nombre de sélections avec un raccourci clavier

AssignToHotkeys : Nombre de raccourcis clavier

UniqueHotkeys : Nombre de raccourcis clavier uniques

MinimapAttacks : Nombre d'attaques sur la minimap

MinimapRightClicks : Nombres de clics droits sur la minimap

NumberOfPACs : Nombre de PAC

GapBetweenPACs : durée moyennes (ms) entre les PAC

ActionLatency : Temps de réaction (ms)

ActionsInPAC : Nombre moyen d'actions au sein de chaque PAC

TotalMapExplored : Nombre de grilles de jeu visualisées

WorkersMade : Nombre d'ouvrier générés et entraînés

UniqueUnitsMade : Nombre d'unités uniques créées

ComplexUnitsMade : Nombres d'unités complexes créées

ComplexAbilitiesUsed : Nombre de capacités spécifiques utilisées

DATA PRE-PROCESSING

Aucune valeur NaN dans notre dataset mais présence de « ? »
Remplacement de ces valeurs par « NaN »

Age	HoursPerWeek	TotalHours
?	?	?
?	?	?
?	?	?
?	?	?
?	?	?

Nombre de valeurs nulles :

Age	55
HoursPerWeek	56
TotalHours	57

Age	HoursPerWeek	TotalHours
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN
NaN	NaN	NaN

Nous remarquons également que les valeurs NaN sont seulement sur la League professionnelle (= 8)

LeagueIndex
5
5
4
3
3

Création d'une nouvelle colonne 'LeagueIndex'
A partir de la colonne 'League'
Pour que l'utilisation des données soit plus simple par la suite

```
1:"Bronze"  
2:"Silver"  
3:"Gold"  
4:"Platinum"  
5:"Diamond"  
6:"Master"  
7:"GrandMaster"  
8:"Professional"
```

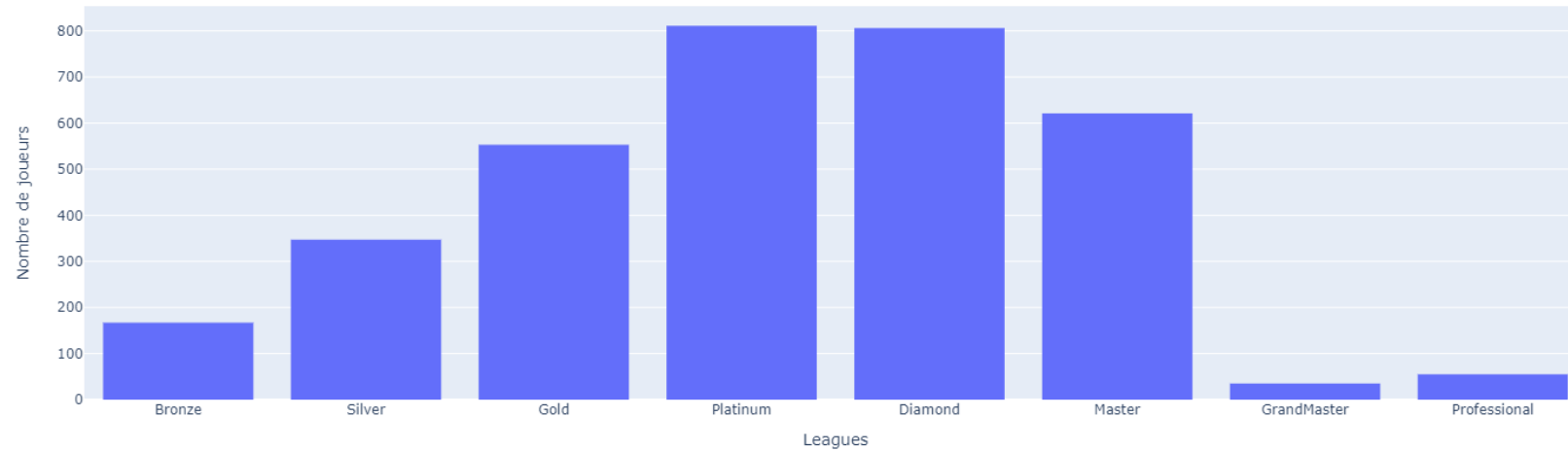
Conversion des colonnes 'Age', 'HoursPerWeek' et 'TotalHours'
De Object en Int

Age	object
HoursPerWeek	object
TotalHours	object

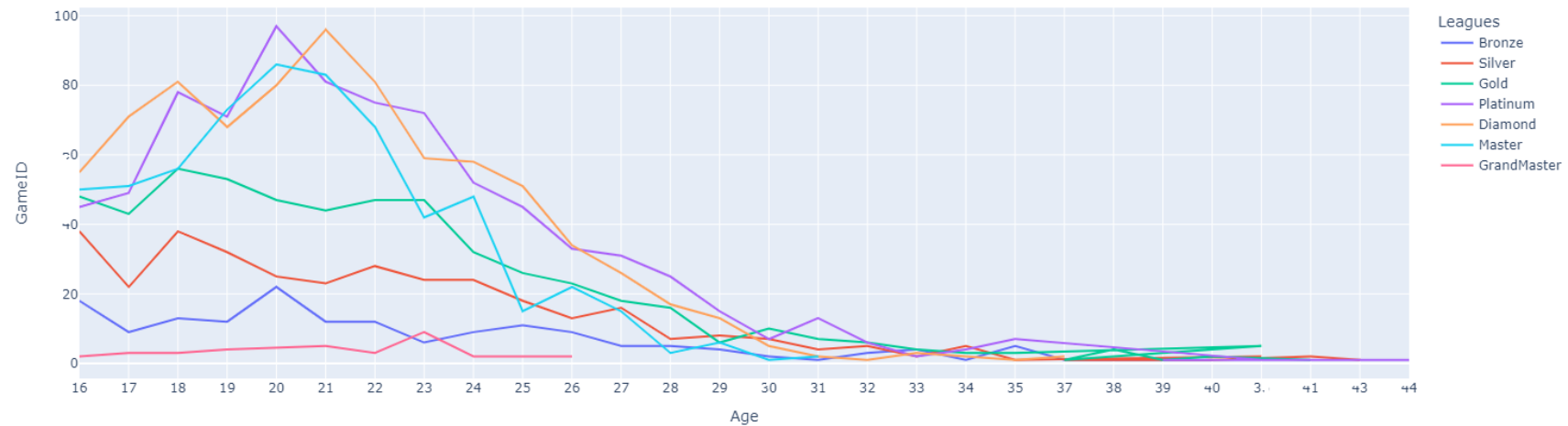
Age	int32
HoursPerWeek	int32
TotalHours	int32

DATA VISUALIZATION

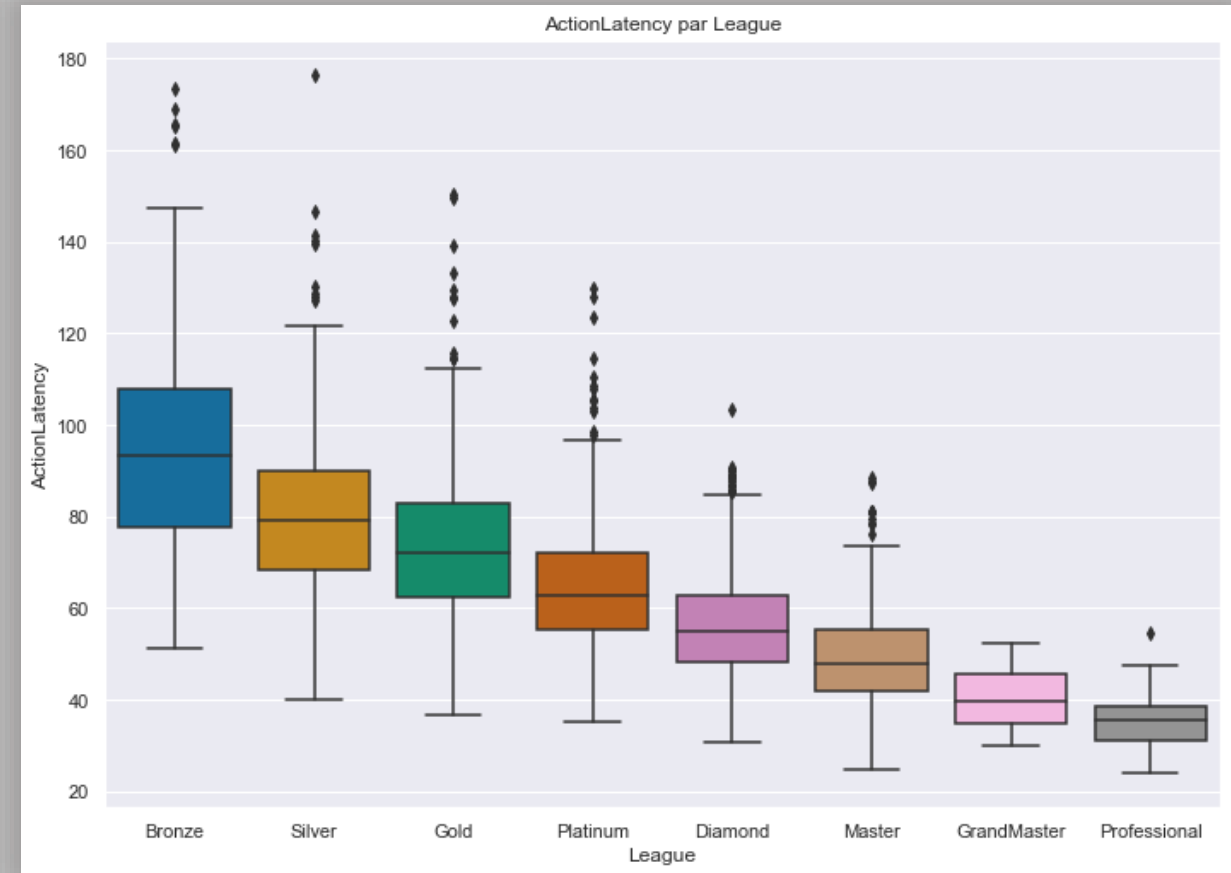
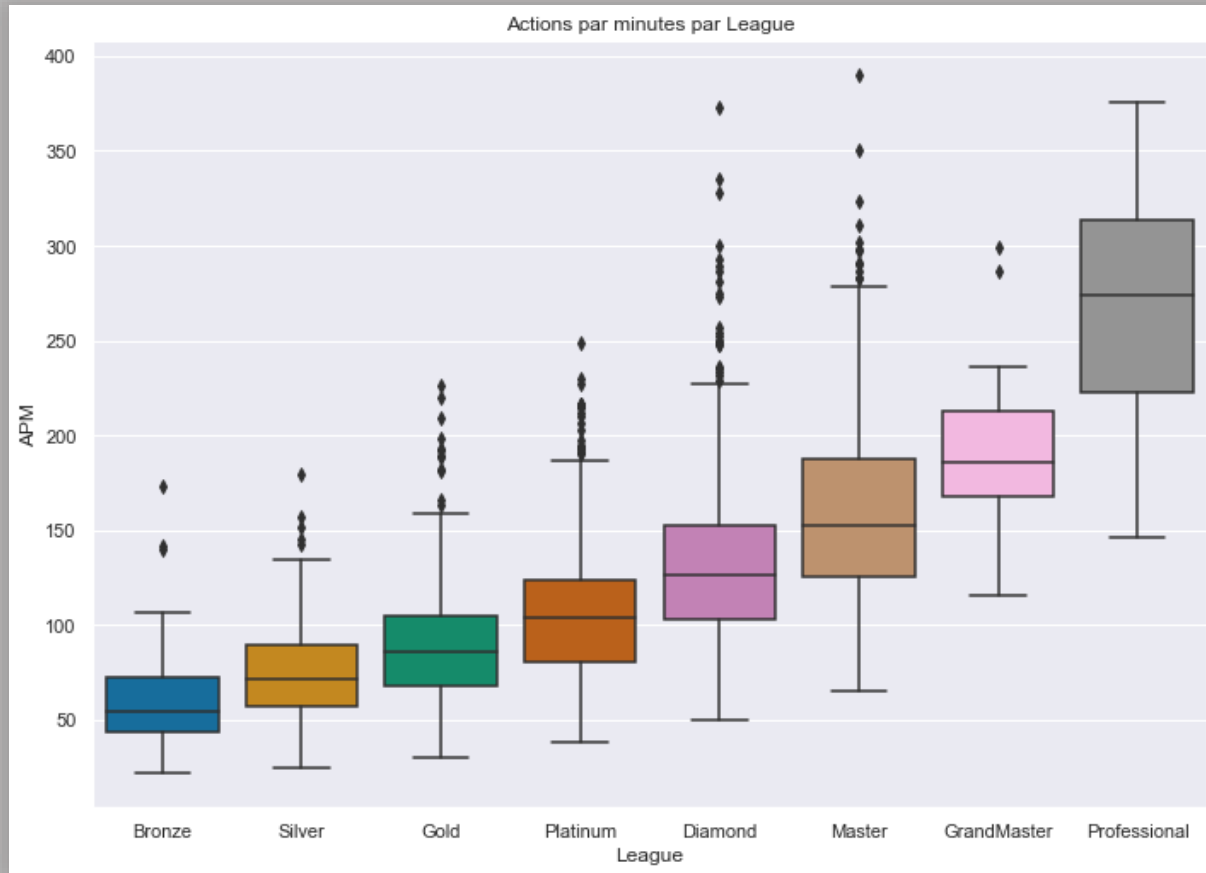
Nombre de joueurs par Leagues



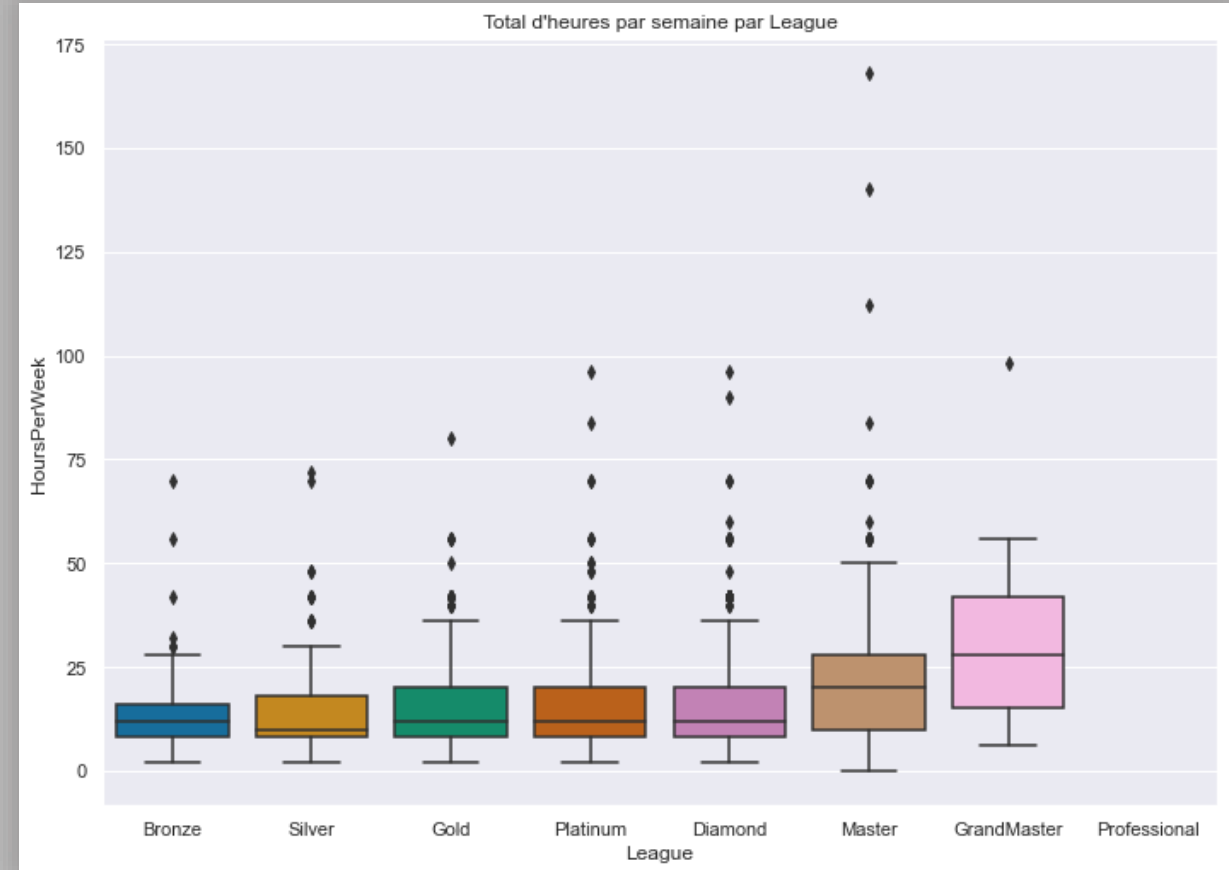
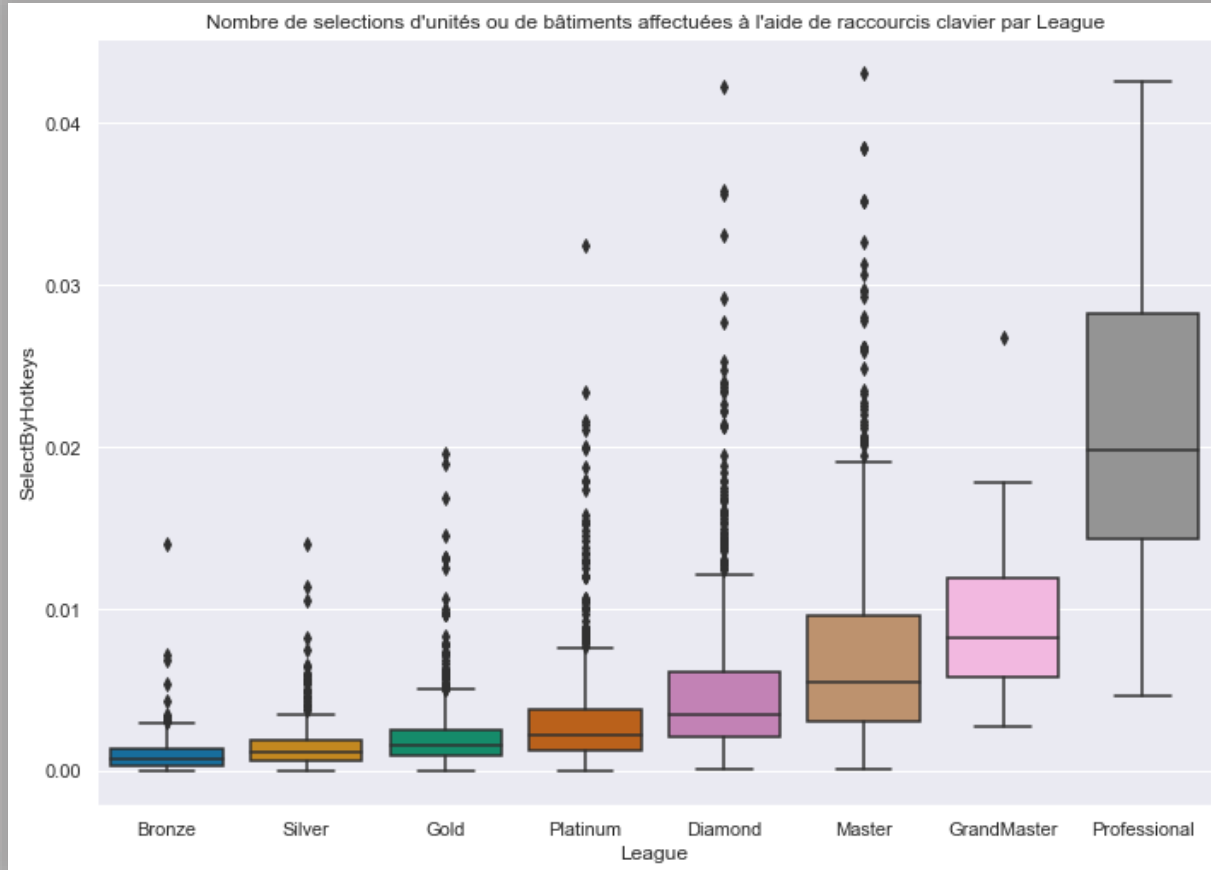
Age des joueurs par Leagues



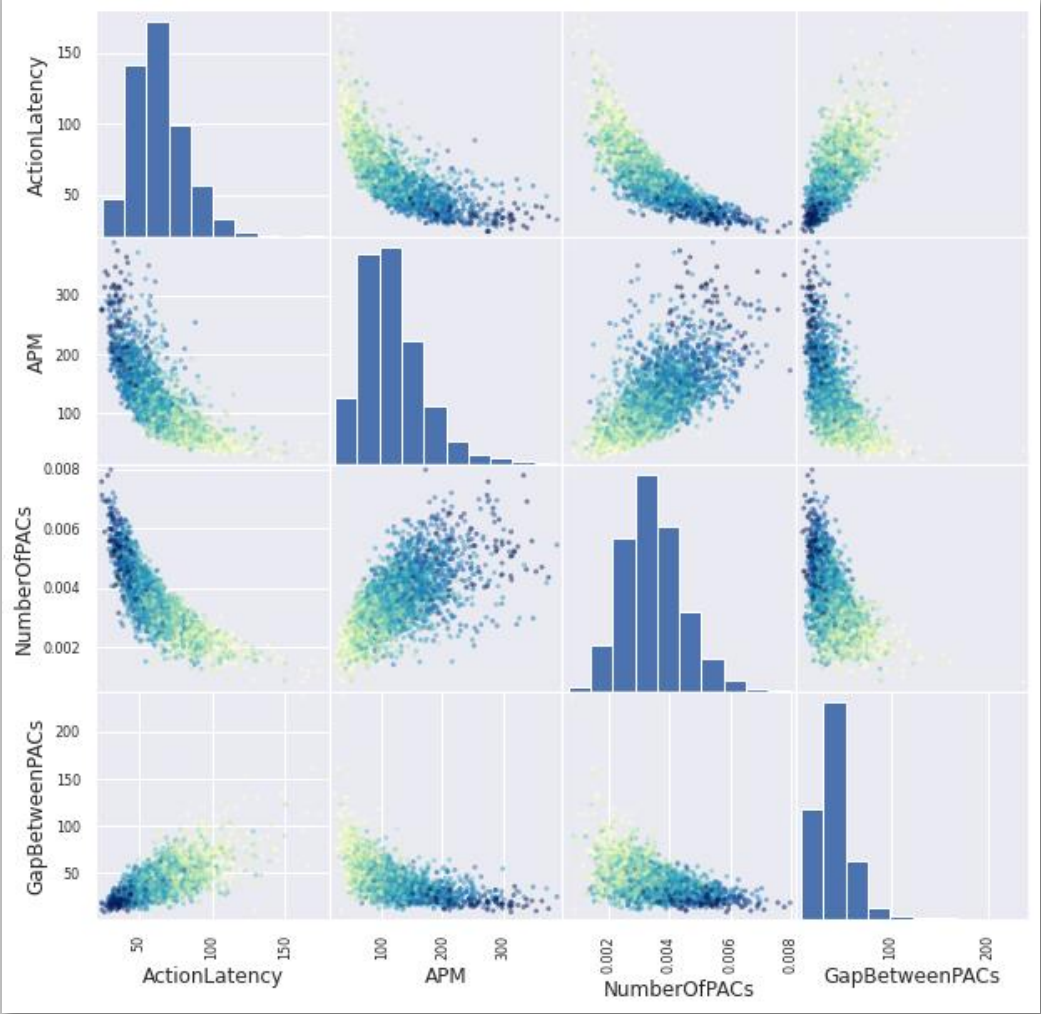
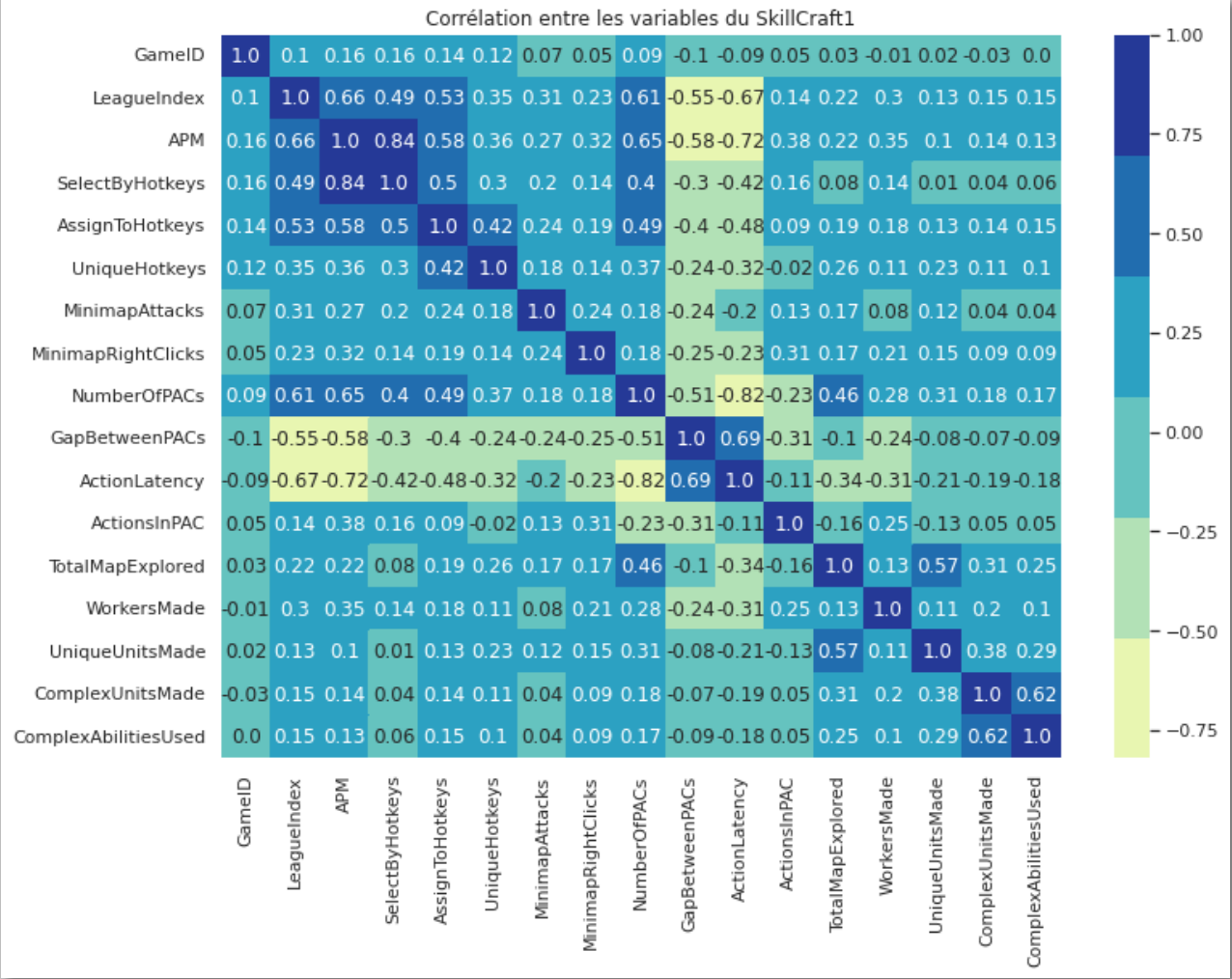
DATA VISUALIZATION



DATA VISUALIZATION



DATA VISUALIZATION



MODELING

1

Suppression des lignes comportant des valeurs « NaN » c'est-à-dire la League Professionnelle. Nous nous retrouvons avec un dataset de 3338 lignes (3395 à l'origine). Puis nous avons converti les colonnes 'Age', 'HoursPerWeek', 'TotalHours' en entier

```
SC_sansNull = SkillCraft  
SC_sansNull.dropna(inplace = True)
```

```
SC_sansNull.shape  
(3338, 21)
```

Age	int32
HoursPerWeek	int32
TotalHours	int32

2

Séparation des données d'apprentissage pour notre modèle

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2)
```

Nous avons donc 2670 lignes pour les X et 668 lignes pour les Y

MODELING

3

Les différents Modèles

RÉGRESSION LINÉAIRE MULTIPLE

Modèle sur 11 colonnes :

HoursPerWeek
APM
SelectByHotkeys
AssignToHotkeys
UniqueHotkeys
MinimapAttacks
NumberOfPACs
GapBetweenPACs
ActionLatency
ActionsInPAC
WorkersMade

RÉSULTAT

Accuracy = 0.4041916167664671

Confusion matrix =

[[2	10	17	3	0	0	0]
[1	8	36	23	1	0	0]
[0	6	50	42	6	1	0]
[0	4	36	105	37	2	0]
[0	1	7	56	73	12	0]
[0	0	0	21	70	29	2]
[0	0	0	0	1	3	3]]

MODELING

3

Les différents Modèles

GRADIENT BOOSTING CLASSIFIER

Modèle sur 11 colonnes :

HoursPerWeek
APM
SelectByHotkeys
AssignToHotkeys
UniqueHotkeys
MinimapAttacks
NumberOfPACs
GapBetweenPACs
ActionLatency
ActionsInPAC
WorkersMade

RÉSULTAT

Best accuracy = 0.3962546816479401

Confusion matrix :

```
[[13  9  4  1  0  0  0]
 [22 17 18  7  3  0  0]
 [ 6 17 26 22 12  1  0]
 [ 2 19 42 73 39 14  0]
 [ 0  2 12 56 70 59  0]
 [ 0  0  4  8 21 57  9]
 [ 0  0  0  0  1  2  0]]
```

MODELING

3

Les différents Modèles

RANDOM FOREST CLASSIFIER

+
SMOTE

Modèle sur 11 colonnes :

HoursPerWeek
APM
SelectByHotkeys
AssignToHotkeys
UniqueHotkeys
MinimapAttacks
NumberOfPACs
GapBetweenPACs
ActionLatency
ActionsInPAC
WorkersMade

SMOTE

```
sm = SMOTE(k_neighbors = 3 ,random_state=42)  
X_res, Y_res = sm.fit_resample(X, Y)
```

4	811	5	811
5	804	4	811
6	621	3	811
3	553	2	811
2	347	1	811
1	167	7	811
7	35	6	811

RESULTAT

Random Forest :

Best accuracy = 0.6315821229700062

Best parameters : {'max_depth': 50, 'n_estimators': 350}

Confusion matrix :

```
[[ 0  0  0  0  0  0  0  0]  
[ 0  0  0  0  0  0  0  0]  
[ 24 17 15 14  6  0  0]  
[ 14 39 55 71 97 82 90]  
[ 0  0  0  0  0  0  0]  
[138 89 78 96 77 67 67]  
[ 0  0  0  0  0  0  0]]
```


API

Utilisation du framework Streamlit

Affichage selon l'élément sélectionné

Réglages

☐ Afficher le dataframe

I. Les leagues

☒ Etudes sur les leagues

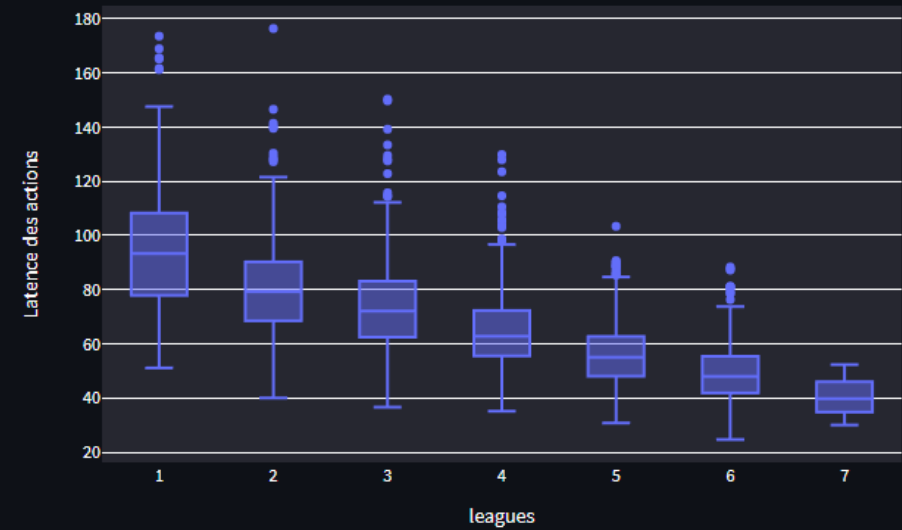
Selectionnez une donnée

Latence des actions

☐ Afficher les corrélations avec le rang



Etude sur les leagues



La colonne ActionLatency semble un peu plus intéressante. Apparemment, la plupart des joueurs de la league GrandMaster ont une faible latence et réagissent donc rapidement aux événements (ce qui serait logique)

API

Utilisation du framework Streamlit

Affichage sa league selon les statistiques

Nombre de cycles Perception/Attaque
(NumberOfPACs)

0,00 - +

Intervalle des cycles Perception/Attaque
(GapBetweenPACs)

45,17 - +

Temps de réaction (ActionLatency)

64,75 - +

Actions durant les cycles Perception/Attaque
(ActionsInPAC)

4,53 - +

Ouvriers générés (WorkersMade)

0,00 - +

Prédire



CONCLUSION

Le meilleur modèle est le random Forest avec un score accuracy = 0.63.
C'est celui qu'on utilise dans notre API pour déterminer la league des joueurs.

Nous voulions savoir comment déterminer la league d'un joueur à partir de ces statistiques.

Les caractéristiques pour déterminer la league d'un joueur sur nous avons utilisées sont :
'HoursPerWeek', 'Actions par minute', 'SelectByHotkeys', 'AssignToHotkeys',
'UniqueHotkeys', 'MinimapAttacks', 'NumberOfPACs', 'GapBetweenPACs',
'ActionLatency', 'ActionsInPAC', 'WorkersMade'

