Cas Kaggle: Predicció de victòria a

Daniel García Nilo - 1633613

Continguts de la presentació

1. Introducció del projecte - Context del joc

League of Legends és un videojoc d'estratègia competitiva.

Dos equips de **5 jugadors contra 5.**

Cada jugador controla un **campió** amb habilitats diferents (de +150 disponibles).

L'objectiu final és **destruir la base rival.**

1. Introducció del projecte - Per què és un bon problema de dades?

Cada partida genera molta informació.

Decisions estratègiques mesurables:
- objectius aconseguits
- moments clau de la partida
- composició dels equips

El resultat final (qui guanya) és clar.

Això és ideal per entrenar un model predictiu, perquè tenim entrades clares (què passa durant la partida) i una sortida clara (quin equip guanya).

01 **Dades** → **02** **Model** → **03** **Predicció**

Objectiu del projecte

Construir un model predictiu capaç d'estimar quin equip guanyarà una partida de League of Legends utilitzant informació estratègica i disponible abans que la partida estigui decidida.

Entendre en quins casos el model falla i per què.

2. Origen i elecció del Dataset – Estructura original

7 arxius .csv amb +180.000 partides.

Gran volum de partides amb informació detallada de cada partida.

Variables tant estratègiques com temporals.

Permet crear noves variables a partir del coneixement del joc.

- ***matches.csv*** *: Informació general de cada partida, com la durada, la versió del joc i els identificadors.*
- ***participants.csv:*** *cada jugador de cada partida, amb el campió seleccionat i l'equip al qual pertany.*
- ***champs.csv:*** *Informació dels campions els +150 campions.*
- ***teamstats.csv:*** *Estadístiques per equip, com kills, morts, assistències, or i minions.*
- ***stats1.csv i stats2.csv:*** *Informació detallada d'objectius i esdeveniments* ***de la partida****, com torres, dracs, barons i inhibidors.*
- ***Dataset final (lol_final.csv):*** *Dataset construït a partir de la unió i neteja de tots els anteriors, amb una fila per partida i les variables utilitzades pel model.*

3. Data Exploration - Elements d'una partida i el seu impacte

3. Data Exploration - Elements d'una partida i el seu impacte

01

Objectius generals

02

Estructures pròpies

03

Estadístiques de partida

3. Data Exploration - Com es reflecteix una partida de LoL en dades

Tenim 60 columnes a lol_final.csv, les podem dividir en aquests grups de dades:

01

Identificadors de partida, equip, durada de partida i winner (guanyador).

02

Esdeveniments d'objectius inicials (el primer de la partida).

03

Objectius totals d'una partida per equip.

04

Estadístiques per equip (Or, kills, assistències, minions...).

05

Qualitat de la composició d'equip (Target Encoding).

4. Data Analysis – Que volem entendra a partir de les dades?

L'anàlisi de dades té tres objectius principals:

- *Entendre quines variables estan relacionades amb la victòria.*
- *Detectar patrons generals en les partides.*
- *Decidir quines variables tenen sentit utilitzar en un model predictiu realista.*

- **Variables binàries (0/1):** Indiquen si un equip ha aconseguit una primera acció. Ex: blue_firstblood, red_firsttower, blue_firstbaron.
- **Variables contínues:** Representen quantitats o valors numèrics reals. Ex: blue_goldearned, phase_te_mean_blue, duration.
- **Variables discretes (comptadors):** Compten el nombre d'accions realitzades. Ex: dragonkills, baronkills, towerkills
- **Variables de composició d'equip:** Resumeixen la força dels campions escollits. Ex: blue_te_mean, phase_te_mean_blue_weighted.
- **Variables de diferència entre equips:** Comparen directament l'avantatge blau vs vermell. Ex: te_mean_diff, phase_te_mean_diff.

5. Feature Engineering: Aportació principal del projecte

Fort a partides de +30 minuts

+ Duració + Força
- Duració - Força

≠

Fort als 15 primers minuts de la partida

+ Duració - Força
- Duració + Força

5. Feature Engineering: Mesurar força de composición segons la fase

Classifiquem cada partida segons la seva duració real:

- **Early game** → fins a 15 minuts.
- **Mid game** → de 15 a 30 minuts.
- **Late game** → més de 30 minuts.

Aquesta informació s'utilitza per entendre en quin moment s'ha jugat la partida i analitzar el rendiment dels campions segons fase.

1. Target Encoding per fase

Per a cada campió i fase (early / mid / late), calculem:
amb quina freqüència guanya històricament
Això dona un valor numèric que representa la seva **força estimada**.
Convertim "quin campió és" en "com de sovint guanya".

2. Agregació per equip

Sumem la informació dels 5 campions:
mitjana,
mínim,
màxim,
variabilitat.
Així obtenim la **força global de la composició**.

3. Pes extra per al late game

• Si la partida és llarga la composició de late game pesa més. Ho fem amb un **multiplicador suau**, sense salts artificials.

6. Data Preparation – Variables descartades

01

Variables descartades per manca d'informació predictiva

S'han eliminat variables que no aporten informació útil.

Identificadors

Versió del joc

Resultats duplicats

02

Variables eliminades per evitar prediccions "amb trampes"

S'han descartat variables que **descriuen directament el resultat final**, com ara:

Estadístiques totals

Rendiment final

03

Variables de primeres accions amb massa pes

Les primeres accions aporten informació valuosa, però algunes tenien **massa pes**.

first inhibitor.

6. Data Preparation – Neteja del dataset i tractament de NaNs

Durant la preparació s'han aplicat diversos filtres:

Eliminació de partides *remake*
Partides amb durada < 300 segons.
No representen partides reals (empats tècnics).

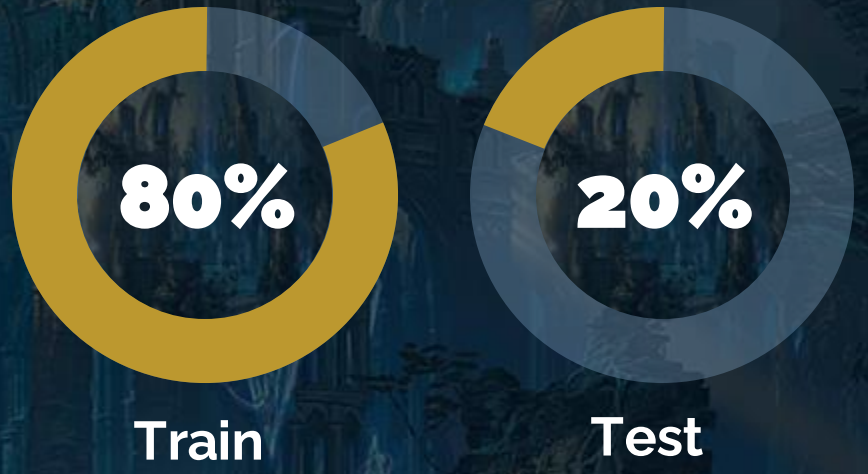
Tractament de valors nuls
Només unes poques columnes presentaven NaNs.
Cas concret: desviació estàndard del target encoding per fase.
S'ha aplicat imputació controlada.

```
Total de NaNs després del tractament: 0
Total de files després d'eliminar remakes: 179267

blue_firstblood     0
red_firstblood      0
blue_firsttower     0
red_firsttower      0
blue_firstdragon    0
dtype: int64
```

7. Comparació de models - Validació i partició de les dades

Les dades es divideixen les dades en **train i test**,
- s'avaluen diferents models de classificació,
- es comparen amb mètriques objectives,
- i es tria el model final.

L'objectiu no és només encertar, sinó garantir que el model:
- generalitza bé,
- no memoritza les dades,
- i és estadísticament robust.




```
Mides del conjunt de dades:
  - X_train: (143413, 38)
  - X_test : (35854, 38)
  - y_train: (143413,)
  - y_test : (35854,)
```

7. Comparació de models - Distribució de classes i mètriques d'avaluació

Abans de comparar models, és important analitzar la **distribució de la variable objectiu (winner)**.

Això indica que el dataset està **pràcticament equilibrat**, fet que permet utilitzar mètriques globals sense risc de biaix cap a una classe concreta.

Mètriques utilitzades:

Per avaluar i comparar els models s'han utilitzat diverses mètriques:

Accuracy
Percentatge total de prediccions correctes.

F1-score
Combina precisió i recall.
És útil per assegurar que el model funciona bé per a **ambdues classes**.

ROC-AUC *(mètrica principal)*
Mesura la capacitat del model per **distingir entre victòria blau i vermell**, independentment del llindar de decisió.

Matriu de confusió
Permet visualitzar els errors de classificació per cada classe.

7. Comparació de models - models simples (sense tuning)

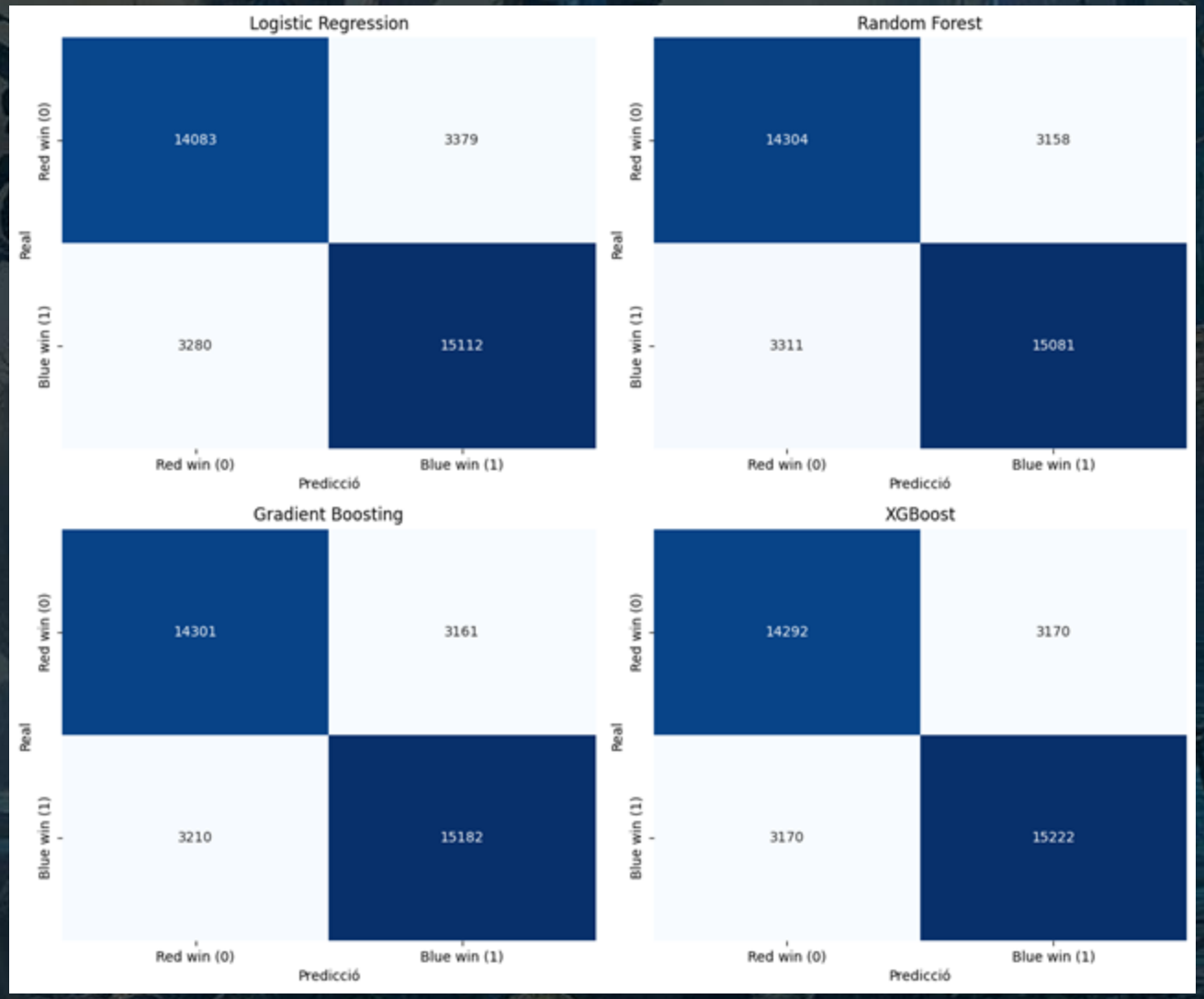
S'han entrenat i comparat diversos models **sense optimització d'hiperparàmetres**, utilitzant exactament el mateix dataset i el mateix *train/test split*:

- **Els models lineals** tenen un rendiment correcte, però limitat.

- **Els models basats en arbres i boosting**: capturen millor les relacions no lineals,aprofiten millor les variables creades (composició, fases, interaccions).

	Accuracy	F1-score	ROC-AUC
Logistic Regression	0.8143	0.8195	0.8770
Random Forest	0.8196	0.8234	0.9050
Gradient Boosting	0.8223	0.8266	0.9094
XGBoost	0.8232	0.8276	0.9107

7. Comparació de models - models simples (sense tuning)



Model triat: XGBoost

8. Optimització del model triat – cerca d'hiperparàmetres XGBoost

Grid Search

Prova totes les combinacions possibles.

Avantatges :

Adequat quan hi ha pocs hiperparàmetres
i valors molt ben definits.

Inconvenient:

cost computacional molt elevat.

Vs

Randomized Search

Prova combinacions aleatòries dins d'un rang.

Avantatges:

molt més eficient en models complexos, permet explorar millor l'espai de solucions, aconsegueix bons resultats amb menys temps.

XGBoost té molts hiperparàmetres rellevants,per això Randomized Search és l'opció més adequada

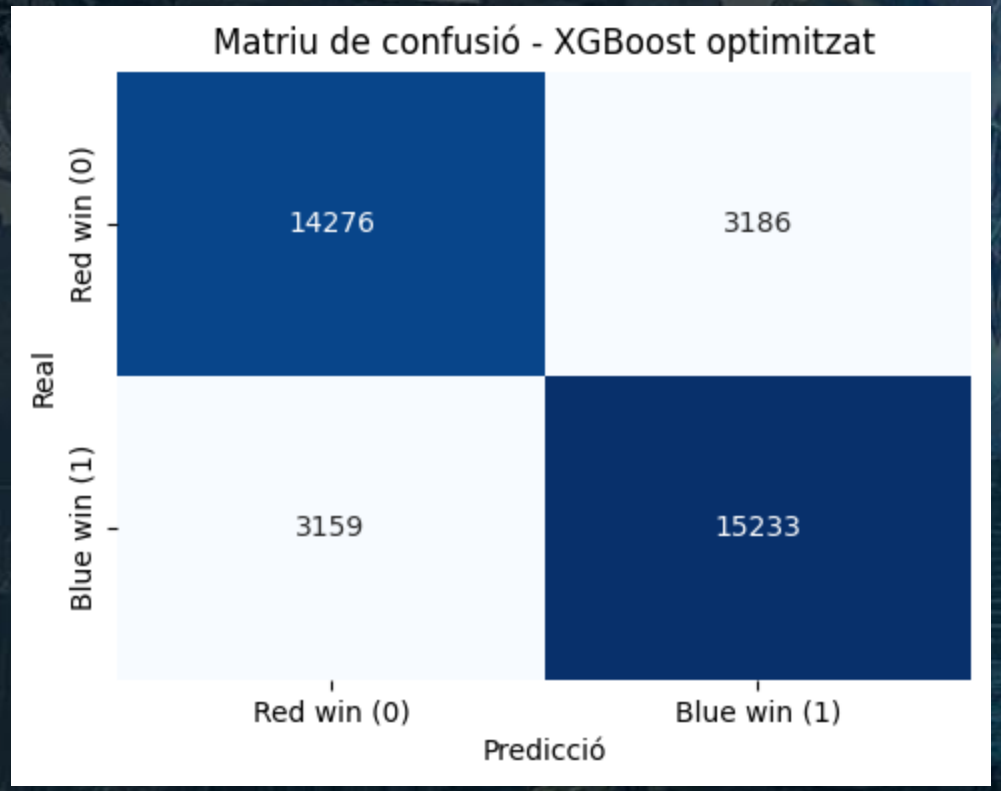
8. Optimització del model triat – Model XGBoost optimitzat: resultats finals

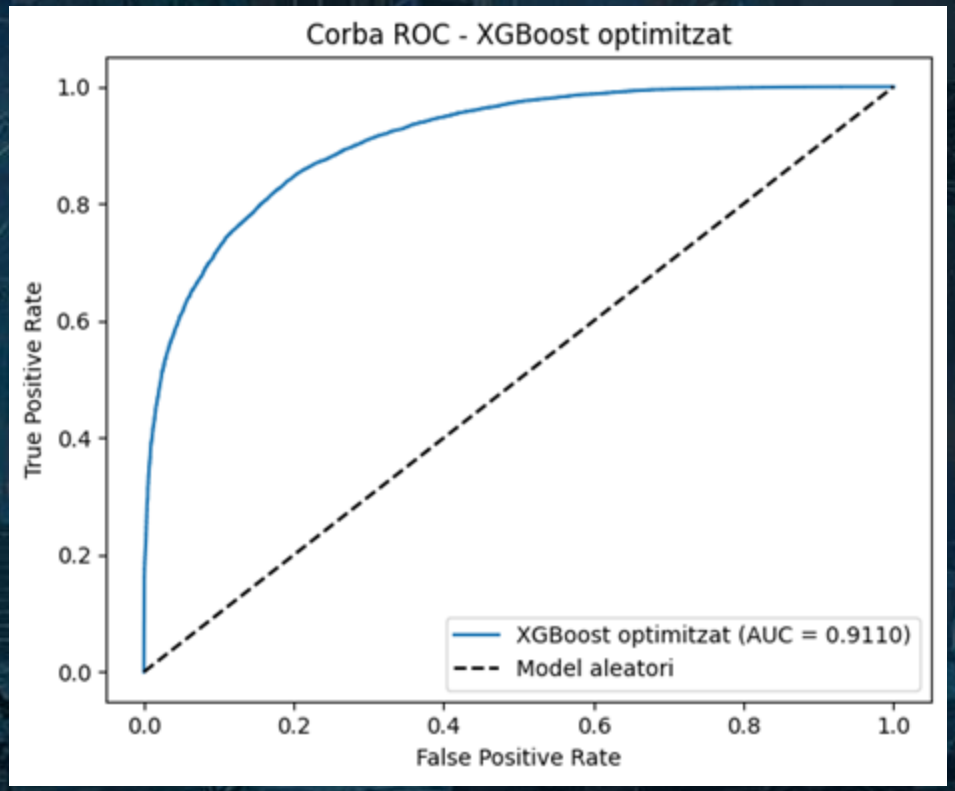
Validació creuada
Cada combinació s'avalua amb **validació creuada (CV = 3)**.
El criteri d'optimització és **ROC-AUC**.

```
Fitting 3 folds for each of 25 candidates, totalling 75 fits

Millors hiperparàmetres trobats:

{'subsample': 1.0,
 'n_estimators': 100,
 'min_child_weight': 3,
 'max_depth': 5,
 'learning_rate': 0.1,
 'gamma': 0.2,
 'colsample_bytree': 0.8}
```

```
Accuracy (XGBoost optimitzat): 0.8230
F1-score (XGBoost optimitzat): 0.8276
ROC-AUC (XGBoost optimitzat): 0.9110

Informe de classificació (XGBoost optimitzat):
              precision    recall  f1-score   support

           0       0.82      0.82      0.82     17462
           1       0.83      0.83      0.83     18392

    accuracy                           0.82     35854
   macro avg       0.82      0.82      0.82     35854
weighted avg       0.82      0.82      0.82     35854
```

8. Optimització del model triat – Model XGBoost optimitzat: resultats finals



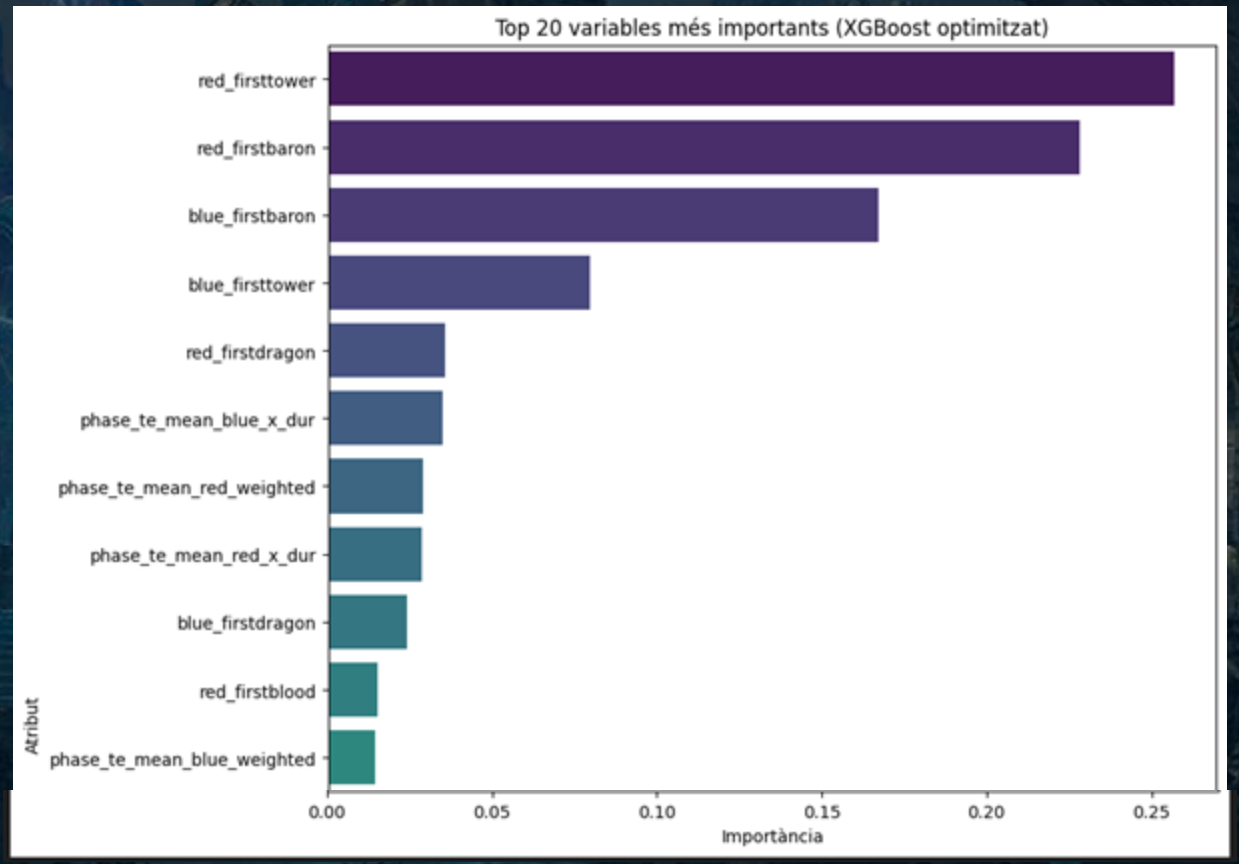


9. Interpretació del model - què ha après XGBoost

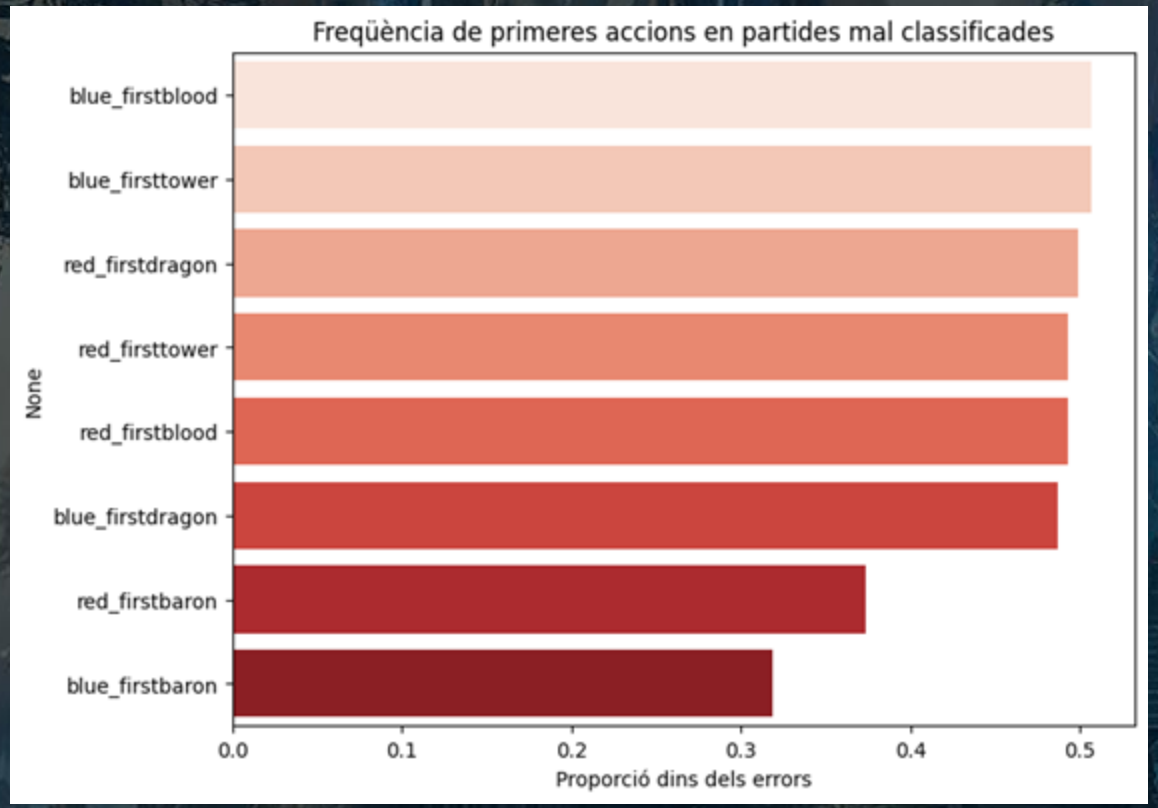
Primeres accions del joc
Primera torre, primer baró, primer drac, first blood
indiquen control inicial del mapa i pressió estratègica.

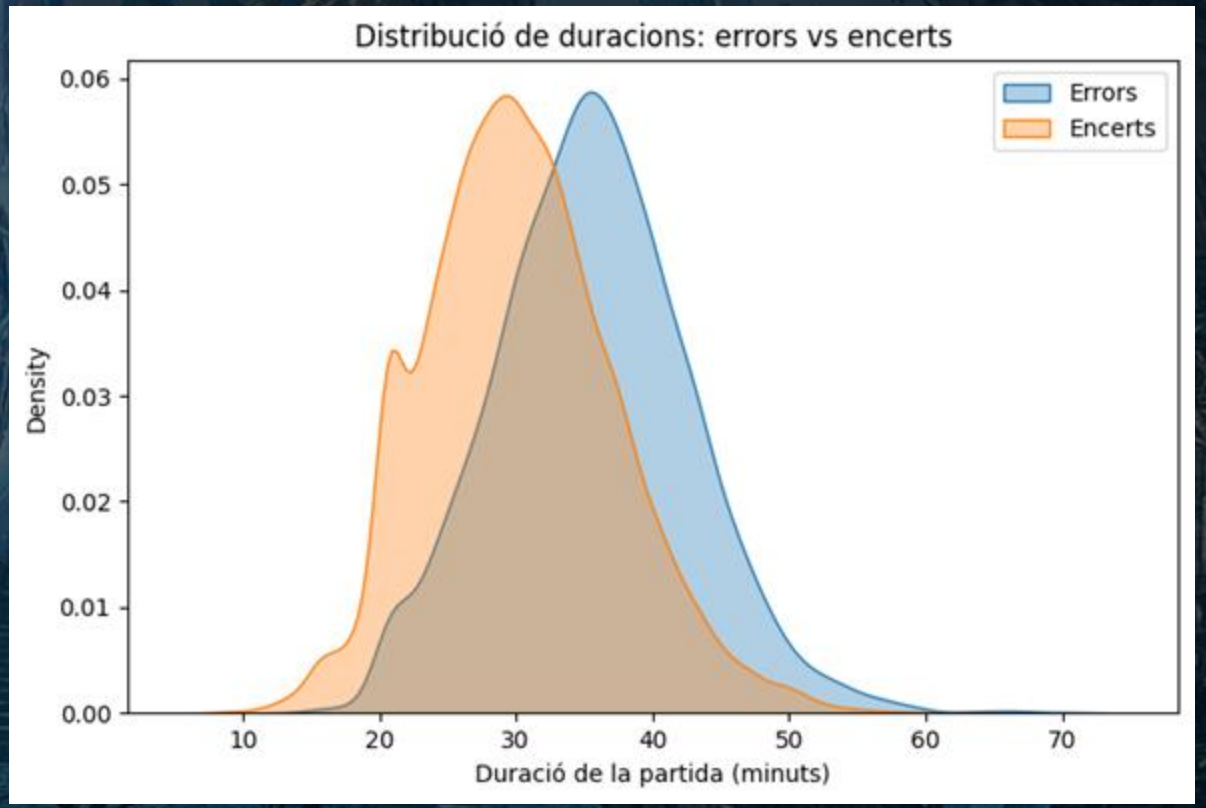
Variables de composició d'equip
Força mitjana dels campions
diferències entre equips.

Força segons la fase del joc
Composicions que escalen millor en partides llargues
variables ponderades pel *late game*.



10. Anàlisi d'errors - Quan falla el model





11. Conclusions

S'ha construït un model predictiu sòlid i realista per a League of Legends.

El projecte aplica correctament: preparació de dades, validació creuada, comparació de models i optimització d'hiperparàmetres.

S'ha prioritzat un enfocament: estratègic i explicable, evitant variables que revelen directament el resultat.

Les variables creades (composició, fases i late game) aporten valor real. XGBoost ofereix el millor equilibri entre rendiment i generalització.


Gràcies