

CUÁN IMPORTANTE SON LOS PRIMEROS 10 MINUTOS DE PARTIDA EN LEAGUE OF LEGENDS

1. Introducción

League Of Legends (*LoL*) es el MOBA (Multiplayer Online Battle Arena) más popular actualmente, el juego consiste en dos equipos de cinco jugadores luchan en un combate jugador contra jugador, cada equipo ocupando y defendiendo su mitad del mapa. Cada uno de los diez jugadores controla un personaje, conocido como "campeón", con habilidades únicas y diferentes estilos de juego. Durante una partida, los campeones se vuelven más poderosos acumulando puntos de experiencia y ganando oro que se emplea para la compra de artículos para derrotar al equipo contrario. En el modo principal de juego, "La Grieta del Invocador", un equipo gana al abrirse paso hasta la base enemiga y destruir su "nexo", una gran estructura ubicada dentro de la cual se emanan tres líneas de súbditos o "minions", personajes no jugadores que al ser eliminados por el equipo enemigo otorgan oro y experiencia. Por lo general una partida dura entre 25 minutos y 40 minutos.

Ya sabiendo de qué trata el juego, podemos hablar sobre la forma de clasificación que existe dentro del juego, las partidas clasificatorias se producen en "La Grieta del Invocador" y otorgan o restan puntos de liga, los cuales sirven para identificar el rango de un jugador, sin complicar mucho la explicación existen los diferentes rangos, de menor a mayor, hierro, bronce, plata, oro, platino, diamante, maestro, gran maestro y retador. Nuestro análisis se dará en el rango de diamante, ya que es un buen punto antes de llegar a las ligas más altas y no tan bajo para que los datos sean muy caóticos.

2. Objetivos

- Determinar cuánto influyen los primeros 10 minutos de juego en el resto de la partida.

- Analizar el dataset para encontrar la mejor forma de ganar una partida clasificatoria.

3. Conceptos teóricos

- **Sklearn**

Scikit-learn es uno de los open-source y bibliotecas de aprendizaje automático más populares en Python. Es una librería muy completa para trabajar con inteligencia artificial, de aquí haremos uso del árbol de decisión y la búsqueda exhaustiva.

- **Árbol de decisión**

Un árbol de decisión es un mapa de los posibles resultados de una serie de decisiones relacionadas. Permite que un individuo o una organización comparen posibles acciones entre sí según sus costos, probabilidades y beneficios. Se pueden usar para dirigir un intercambio de ideas informal o trazar un algoritmo que anticipe matemáticamente la mejor opción.

- **Búsqueda exhaustiva**

Es una técnica trivial pero a menudo usada, que consiste en enumerar sistemáticamente todos los posibles candidatos para la solución de un problema, con el fin de chequear si dicho candidato satisface la solución al mismo.

- **Dataset**

Un conjunto de datos o dataset corresponde a los contenidos de una única tabla de base de datos o una única matriz de datos de estadística, donde cada columna de la tabla representa una variable en particular, y cada fila representa a un miembro determinado del conjunto de datos que estamos tratando.

4. Datos

El dataset cuenta con las siguientes columnas:

gameId

El id del juego.

blueWins

Si el equipo azul consiguió la victoria.

blueWardsPlaced

El número de centinelas puestos por el equipo azul.

blueWardsDestroyed

El número de centinelas destruidos por el equipo azul.

blueFirstBlood

Si el equipo azul consiguió la primera sangre.

blueKills

El número total de asesinatos por el equipo azul.

blueDeaths

El número total de muertes por el equipo azul.

blueAssists

El número total de asistencias por el equipo azul.

blueEliteMonsters

El número total de monstruos elite conseguidos por el equipo azul.

blueDragons

Si el equipo azul consiguió el primer dragon.

blueHeralds

Si el equipo azul consiguió el primer heraldo.

blueTowersDestroyed

La cantidad de torres que destruyó el equipo azul.

blueTotalGold

El oro total del equipo azul.

blueAvgLevel

Nivel promedio de todos los campeones del equipo azul.

blueTotalExperience

Experiencia total del equipo azul.

blueTotalMinionsKilled

La cantidad total de minions conseguidos por el equipo azul.

blueTotalJungleMinionsKilled

El total de minions de la jungla conseguidos por el equipo azul.

blueGoldDiff

La diferencia de oro que tiene el equipo azul con el equipo rojo.

blueExperienceDiff

La diferencia de experiencia que tiene el equipo azul con el equipo rojo.

blueCSPerMin

La cantidad de minions por minuto que consiguió el equipo azul.

blueGoldPerMin

La diferencia de oro que tiene el equipo azul con el equipo rojo por minuto.

redWardsPlaced

El número de centinelas puestos por el equipo rojo.

redWardsDestroyed

El número de centinelas destruidos por el equipo rojo.

redFirstBlood

Si el equipo rojo consiguió la primera sangre.

redKills

El número total de asesinatos por el equipo rojo.

redDeaths

El número total de muertes por el equipo rojo.

redAssists

El número total de asistencias por el equipo rojo.

redEliteMonsters

El número total de monstruos elite conseguidos por el equipo rojo.

redDragons

Si el equipo rojo consiguió el primer dragon.

redHeralds

Si el equipo rojo consiguió el primer heraldo.

redTowersDestroyed

La cantidad de torres que destruyó el equipo rojo.

redTotalGold

El oro total del equipo rojo.

redAvgLevel

Nivel promedio de todos los campeones del equipo rojo.

redTotalExperience

Experiencia total del equipo rojo.

redTotalMinionsKilled

La cantidad total de minions conseguidos por el equipo rojo.

redTotalJungleMinionsKilled

El total de minions de la jungla conseguidos por el equipo rojo.

redGoldDiff

La diferencia de oro que tiene el equipo rojo con el equipo rojo.

redExperienceDiff

La diferencia de experiencia que tiene el equipo rojo con el equipo rojo.

redCSPerMin

La cantidad de minions por minuto que consiguió el equipo rojo.

redGoldPerMin

La diferencia de oro que tiene el equipo rojo con el equipo rojo por minuto.

5. Preprocesamiento

Para el procesamiento de datos se está haciendo uso de la librería `MinMaxScaler` de `sklearn`, la razón de porque se usa esta librería es porque nos devuelve los datos entre 0 y 1, así tenemos menos dispersión en los datos, además tenemos que tener en cuenta que el 0 si es un dato relevante en nuestro dataset, por tal motivo se decidió hacer elección de este forma de preprocesamiento de los datos. Tampoco contamos con datos del tipo `NaN`, ya que todos los datos se encuentran presentes, por tal motivo no hace falta rellenar campos vacíos, ya que no se tiene ninguno.

El código que haremos uso será:

```
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
X = df_clean
y = df['blueWins']
scaler = MinMaxScaler()
scaler.fit(X)
X = scaler.transform(X)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

En el cual el parámetro `test_size` primeramente estará en un 20%, para luego pasar a 50%.

6. Modelo

Para analizar los datos hacemos uso de un árbol de decisión, con búsqueda exhaustiva.

La búsqueda exhaustiva se realiza sobre valores de parámetros especificados por un estimador. Entre los parámetros que necesita se encuentran, un

clasificador, que para esta ocasión será un árbol de decisión, un diccionario con los hiperparametros que haremos uso y por último la estrategia para realizar la validación cruzada.

El árbol de decisión nos va dando una serie de caminos, entre si y no, de acuerdo a su predicción. Esto nos servirá como clasificador, el cual haremos uso en la búsqueda exhaustiva.

7. Resultados

Entre los resultados obtenidos tendríamos, que el 71,15% de las partidas se lograron predecir, eso nos da como resultado que la mayoría de las partidas se logran decidir entre los primeros 10 minutos, dando mucha relevancia a las siguientes ventajas:

Diferencia de oro, mientras mas grande sea la diferencia de oro entre ambos equipos es más difícil que exista una victoria por parte del equipo que va por debajo en oro.

El primer dragón, esto puede llegar a significar una diferencia muy grande, ya que acerca más al equipo que lo consiguió a tener los 4 dragones, significante esto una diferencia notoria con el otro equipo.

La diferencia de experiencia entre equipos, esto provoca que el equipo que vaya por debajo en experiencia no tenga libertad de toma de decisiones y se vea condicionado por lo que hace el otro equipo, que mientras mas grande sea la diferencia pocas opciones le queda para ganar al equipo que va por atrás.

Las asistencias totales de un equipo, esto nos indica en cuántas muertes participo cada integrante del equipo, cada asistencia significa un poco de oro y experiencia para el que la tiene, que aunque signifique menos oro que un asesinato, con las suficientes asistencias se puede equiparar.

Cantidad de centinelas puestos, todas las ventajas anteriores nos dan un mejor control del juego, eso implica que podemos tener un mejor control de visión con los centinelas, de esta forma asegurando la ventaja del equipo que va por delante.

El modelo nos puede mostrar mas opciones, pero estan llegan a ser las más relevantes, además es difícil pensar en mas variables mientras se encuentra en una partida, además cada una va seguida de la otra, me explico, si se tiene ventaja de oro, facilita a conseguir el primer dragón, estos dos a su vez dan una diferencia de experiencia entre ambos equipos, lo cual da mas asesinatos al equipo que tiene ventaja, lo que significa mas asistencias para el mismo equipo y gracias a toda esa ventaja pueden poner mas centinelas y tener un mejor control de visión del mapa.