Predicción de victoria en league of legends usando clasificadores

Reconocimiento de variables dependientes en un dataset

Alvarado Ramos José Antonio[1], Arreguin Perez Marcos Manuel[2],
Rodriguez Vazquez Jerson Jesus[3], Saúl Efrén Velazquez Cruz[4], Zapata Correa Jossie Ivana[5]

[1] FES Acatlán, Matemáticas Aplicadas y Computación
e-mail: antonio98cchn@gmail.com
[2] FES Acatlán, Matemáticas Aplicadas y Computación
e-mail: marreguinp@gmail.com
[3] FES Acatlán, Matemáticas Aplicadas y Computación
e-mail: jerson_rv@outlook.es
[4] FES Acatlán, Matemáticas Aplicadas y Computación
e-mail: saulvelazquez178@gmail.com
[5] FES Acatlán, Matemáticas Aplicadas y Computación
e-mail: jossieivana@gmail.com

May 23, 2020

ABSTRACT

Context. League of legends es un juego competitivo Multiplayer Online Battle Arena (MOBA) para pc que se ha convertido en uno de los mas competitivos en los últimos años, por eso es que hicimos este proyecto estudiando los primeros 10 minutos de aproximadamente 10,000 para determinar sí hay alguna estadística especial que influya más que otras o que es lo más decisivo en una partida sin contar la habilidad del jugador. Tratar de encontrar que objetivos dentro del juego ponen más ventaja en un equipo.

Aims. Deseamos encontrar ciertas gráficas que nos marquen tendencias y relaciones entre las variables y regresiones de clasificación para determinar si a los 10 minutos se puede predecir el final de una partida.
Methods.
Para desarrollar este estudio aplicaremos modelos estadísticos para el estudio de variables categóricas tratando de encontrar variables dependientes a otras y las mas importantes en los modelos. Nos centraremos principalmente en:

1. Regresión logística: Es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores.
2. Random Forest: Es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging que construye una larga colección de árboles no correlacionados y luego los promedia.
3. Clasificador bayesiano ingenuo: Un clasificador Bayesiano ingenuo es un clasificador probabilístico fundamentado en el teorema de Bayes y algunas hipótesis simplificadoras adicionales. Es a causa de estas simplificaciones, que se suelen resumir en la hipótesis de independencia entre las variables predictoras.

Results. Podemos observar que hay muchas variables que afectan a nuestra variable target, que nos indica la victoria o derrota del equipo azul, sin embargo a los 10 minutos no es concluyente el resultado final de la partida. Seria conveniente hacer un estudio similar con las partidas completas y de esta manera complementar los resultados del articulo para buscar unos mas concluyentes y verificar si son correctos nuestros resultados de las partidas parciales.

Key words. Regresión logística – Predicción – Clasificador– League of legends

1. Introducción

League of legends es un juego multijugador MOBA (Multiplayer Online Battle Arena) para pc que se ha convertido en uno de los juegos mas jugados y con mayor nivel competitivo en los últimos años, a causa de esto hemos desarrollado este proyecto. El juego consiste en 5 jugadores que tratan de destruir la base enemiga, para lograr eso deben atravesar el mapa y destruir las torretas enemigas y enemigos para lograrlo, las partidas suelen durar un tiempo promedio de 35 minutos, por eso decidimos tomar este dataset que incluye los primeros 10 minutos de partidas para ver si es posible determinar el resultado de las partidas en los primeros 10 minutos de juego o si se debe de tomar mas precauciones los primeros 10 minutos para asegurar la victoria o prevenir una derrota. el data set incluye los primeros 10 minutos de un aproximado de 10,000 partidas, incluye datos como:

1. blueWins que nos indica si el equipo azul ganó o perdió
2. blueFirstBlood nos indica si el equipo azul logro el primer asesinato
3. blueKills el número de asesinatos hechos por el equipo azul
4. blueDeaths el número de muertes del equipo azul

A priori creemos que esas son las principales variables que pueden afectar a una victoria o derrota, entonces comenzaremos con hacer un análisis para determinar si podemos determinar la victoria del equipo y procederemos a ver las variables con mas peso durante la partida.

2. Estado del arte

En este articulo de Identification of human vital functions directly relevant to the respiratory system based on the cardiac and acoustic parameters and Random Forest se desarrolló un enfoque analítico conveniente sin participación humana para la evaluación de la calidad del té con gran importancia. En este estudio, la imagen hiperespectral del infrarrojo cercano (HSI) combinada con múltiples métodos de árbol de decisión se utilizó como una herramienta de análisis objetivo para delinear la calidad y el rango. La fusión de datos que integraba características de textura basadas en la matriz de coincidencia de nivel y las características espectrales de infrarrojo cercano de onda corta eran la información característica objetivo para el modelado. Los resultados indicaron que el rendimiento de los modelos fue mejorado por la fusión de características de percepción múltiple. El modelo de árbol fino basado en la fusión de datos obtuvo el mejor rendimiento predictivo, y la tasa de clasificación correcta.

En este articulo de Using near-infrared hyperspectral imaging with multiple decision tree methods to delineate black tea quality del sueño se incorporan ondas cerebrales, el nivel de oxígeno en la sangre, frecuencia cardíaca y respiración, y grabaciones de movimientos de piernas. Se ha estudiado una técnica alternativa para los trastornos respiratorios relacionados con el sueño basada en parámetros cardíacos y acústicos seleccionados y el bosque aleatorio. Se propone un sistema dedicado a la detección de ECG adquirido simultáneamente y señales acústicas, que se recogen durante el sueño en el entorno doméstico. Los resultados obtenidos indican que los modelos de árbol de clasificación y regresión como el bosque aleatorio son apropiados para la evaluación de trastornos del sueño. Por lo tanto, los modelos predictivos estadísticos permiten la identificación de eventos respiratorios con altos niveles de sensibilidad y especificidad, proporcionando un diagnóstico económico y preciso.

En esté articulo de Modelo de regresión logistica que citamos, se busca estimar mediante regresión logística la asociación de dependencia según escala de Lawton y Brody en mayores de 65 años y variables sociodemográficas. El cual incluye sexo, grupos de edad, convivencia y clase social, asociadas con la dependencia. Y aquí sé considero útil la escala en mayores de 65 años para detectar precozmente la dependencia.

3. Metodología

Para realizar este estudio y tratar de predecir si en realidad se puede predecir la victoria de un equipo trascurridos 10 minutos de partida utilizamos diferentes clasificadores logísticos para buscar una predicción certera. Comenzamos con una regresión logística

1. Regresión logística:
   la regresión logística es un tipo de análisis de regresión utilizado para predecir el resultado de una variable categórica (una variable que puede adoptar un número limitado de categorías) en función de las variables independientes o predictoras. Es útil para modelar la probabilidad de un evento ocurriendo como función de otros factores. El análisis de regresión logística se enmarca en el conjunto de GLM.

```
[44]
y = df['blueWins'].values
X = df.drop(['blueWins'],axis=1).values
scaler = MinMaxScaler()
scaler.fit(X)
X_scaled = scaler.transform(X)
X_train,X_test,y_train,y_test = train_test_split(X_scaled,y,shuffle=True)

[45]
clf = LogisticRegression(random_state=0).fit(X_train,y_train)
accuracy_score(clf.predict(X_test),y_test)

0.7267206477732794
```

   Primero marcamos nuestra variable target como nuestro eje 'y' y posteriormente la sacamos del dataset en nuestra variable x para no tener los repetidos. Después solo corremos el modelos que sacamos de la librería:
   from sklearn.linear_model import LogisticRegression
   y obtenemos un score en el train de los datos de:
   0.7267206477732794
   que es bastante bueno para comenzar.

2. Naive Bayes:
   Los clasificadores ingenuos de Bayes son una familia de "clasificadores probabilísticos" simples basados en la aplicación del teorema de Bayes con fuertes supuestos de independencia (ingenuos) entre las características. Se encuentran entre los modelos de red bayesianos más simples. Pero podrían combinarse con la estimación de densidad de Kernel y lograr niveles de precisión más altos.

```
[52]
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score
clf_nb = GaussianNB()
clf_nb.fit(X_train, y_train)
pred_nb = clf_nb.predict(X_test)
acc_nb = accuracy_score(pred_nb, y_test)
print(acc_nb)

0.7267206477732794
```

   El modelo de Bayes de igual manera se importa de las librerias de sklearn:
   from sklearn.naive_bayes import GaussianNB
   Este modelo podemos observar que nos da el mismo score que nuestro modelo de regresión logística:
   0.7267206477732794
   que es bastante bueno, pero no es determinante.

3. Decision Tree:
   utiliza un árbol de decisión como un modelo predictivo que mapea observaciones sobre un artículo a conclusiones sobre el valor objetivo del artículo. Es uno de los enfoques de modelado predictivo utilizadas en estadísticas, minería de datos y aprendizaje automático. Los modelos de árbol, donde la variable de destino puede tomar un conjunto finito de valores se denominan árboles de clasificación. En estas estructuras de árbol, las hojas representan etiquetas de clase y las ramas representan las conjunciones de características que conducen a esas etiquetas de clase. Los árboles de decisión, donde la variable de destino puede tomar valores continuos (por lo general números reales) se llaman árboles de regresión.

```
from sklearn import tree
from sklearn.model_selection import GridSearchCV
tree = tree.DecisionTreeClassifier()
grid = {'min_samples_split': [5, 10, 20, 50, 100]},
clf_tree = GridSearchCV(tree, grid, cv=5)
clf_tree.fit(X_train, y_train)
pred_tree = clf_tree.predict(X_test)
acc_tree = accuracy_score(pred_tree, y_test)
print(acc_tree)
```

```
0.6829959514170041
```

De igual manera el modelo de el árbol de decisión fue importado de las librerías de sklearn:
from sklearn import tree
Este por el momento es nuestro peor score obtenido en los modelos con una puntuación de:
0.6829959514170041
Que es considerablemente mala si lo comparamos con nuestros modelos anteriores.

4. Random forest:
los bosques de decisión aleatorios son un método de aprendizaje conjunto para la clasificación , la regresión y otras tareas que operan construyendo una multitud de árboles de decisión en el momento del entrenamiento y generando la clase que es el modo de las clases (clasificación) o predicción media (regresión) de los árboles individuales. Los bosques de decisión aleatorios corrigen el hábito de los árboles de decisión de sobreajustarse a su conjunto de entrenamiento.

```
[54]
from sklearn.ensemble import RandomForestClassifier
rf = RandomForestClassifier()
grid = {'n_estimators':[100,200,300,400,500], 'max_depth': [2, 5, 10]}
clf_rf = GridSearchCV(rf, grid, cv=5)
clf_rf.fit(X_train, y_train)
pred_rf = clf_rf.predict(X_test)
acc_rf = accuracy_score(pred_rf, y_test)
print(acc_rf)
```

```
0.7323886639676114
```

El modelo de random forest de igual manera es importado de las librerias de sklearn:
from sklearn.ensemble import RandomForestClassifier
Como podemos leer en la descripción del random forest, este supone un mejor score de entrenamiento sobre nuestro score que obtuvimos en el árbol de decisión y como se menciona anteriormente esto es debido a que nuestro modelo corrige el hábito de sobreajustarse a su modelo de entrenamiento. El score obtenido para este modelo es el mejor que obtuvimos con un puntaje de:
0.7323886639676114.

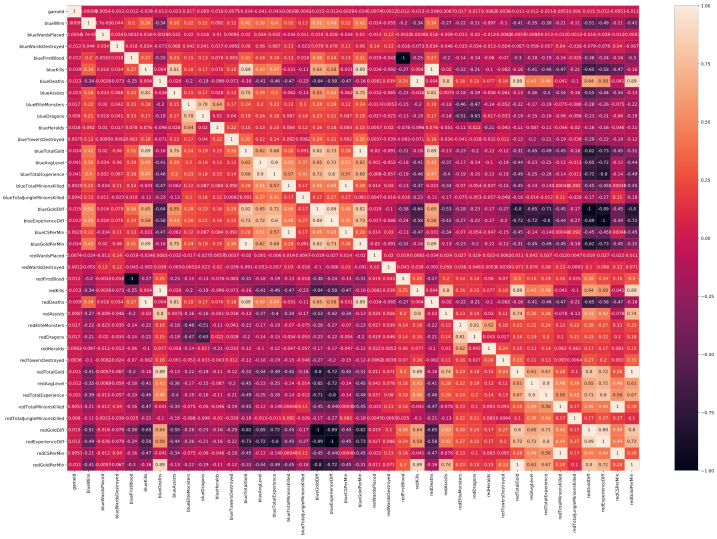
Para corroborar la veracidad de nuestros modelos aplicamos un kfold e hicimos 10 subconjuntos que se muestran en la tabla posterior:

Modelo	Score	Train	Test
Regresion logística	0.726721	0.728437	0.74251
Naive Bayes	0.720648	0.724253	0.731579
Arbol de decisión	0.70081	0.677957	0.693927
Random Forest	0.729555	0.728033	0.735628

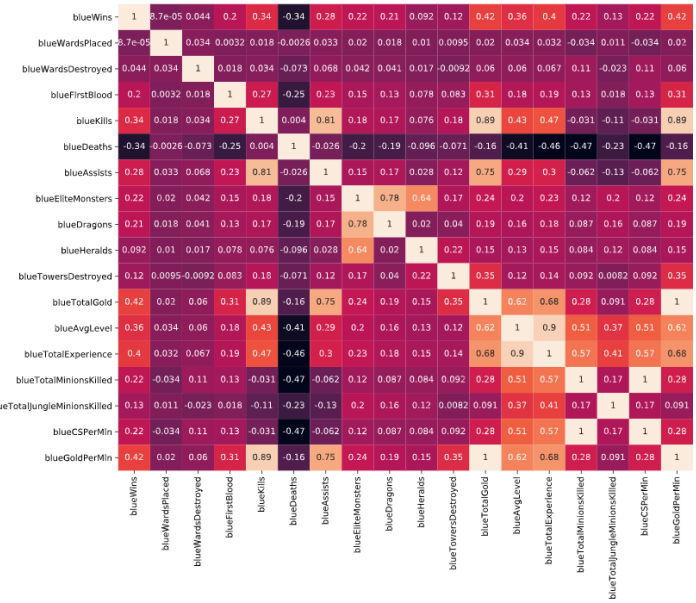
Como podemos ver los resultados varían un poco a los anteriores, esto debido a que volvimos a correr los modelos para corroborar que no tuviéramos un overfittin o underfitting, como podemos observar los Random Forest sigue siendo nuestro mejor modelo.

Con los resultados obtenidos en los modelos de clasificación podemos concluir 2 cosas, para empezar notamos que el modelo de random forest es el mejor modelo de los que utilizamos y por otra parte podemos concluir que no nos es posible determinar con una precisión del 100% el resultado de una partida en los primeros 10 minutos, aunque como obtuvimos un puntaje alto es preciso observar mas de cerca las variables para mirar que es lo que afecta la condición de victoria en una partida.

Empezamos por hacer un heat map de muestro data set para observar las variables que mas se relacionan entre ellas.



Para ver la imagen completa puede ingresar al proyecto en github para verlo mejor. de igual manera podemos notar patrones en los 4 cuadrantes de nuestro heat map, por lo que las variables que se relacionan en el primer cuadrante son iguales a las del 4, ya que es el comportamiento de ambos equipos durante la partida y en los cuadrantes 2 y 3 no vemos relación así que los omitiremos por ahora:

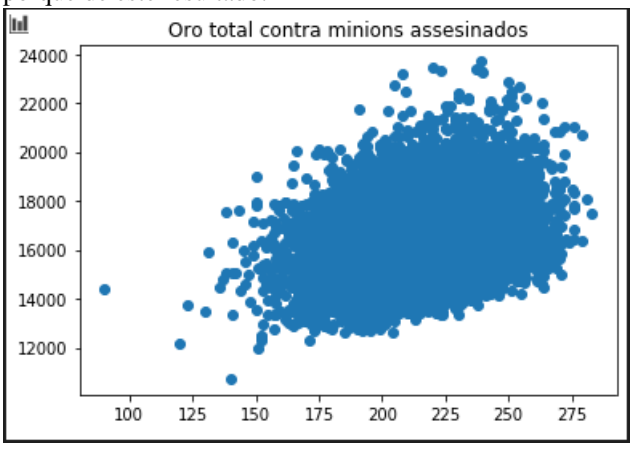


Para considerar una variable dependiente de otra buscaremos un puntaje superior a los 0.4 puntos.

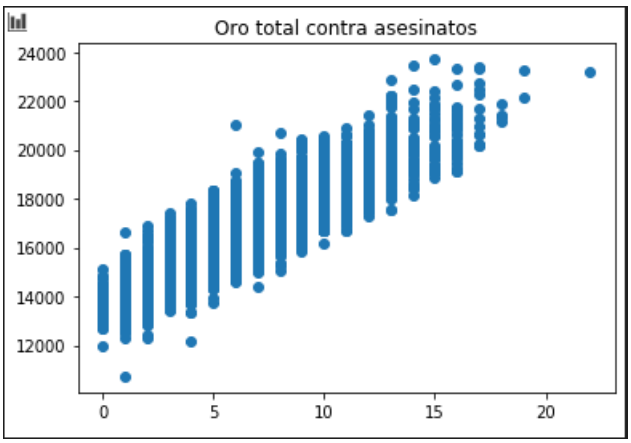
1. blueWins:
Podemos ver que la victoria de un equipo tiene mucho peso el oro total, la cantidad de experiencia que reciben y el oro por minuto que es capaz de generar el equipo.

2. blueKills:
   Para generar asesinatos en las primeras instancias de las partidas observamos que tiene mucho peso la ventaja de oro que se puede tener sobre el rival y las asistencias, lo que implica mucha presencia de los compañeros para poder lograr los asesinatos, también podemos ver que la ventaja de nivel afecta pero no es tan determinante.
3. blueAssists:
   Creo es un poco obvio que las kills estarían presentes como variables relacionadas, de igual manera la cantidad de oro ya que la asistencia y el oro aportan a la suma total de oro.
4. blueEliteMonsters:
   Aquí excluimos los resultados dado que lo que afecta a esta variable es directamente eliminar los objetivos de elite en el mapa.
5. blueTotalGold:
   Como ya lo habiamos visto anteriormente el oro total afecta nuestras condiciones de victoria, también vemos que como fuente principal de oro durante la partida tenemos los asesinatos y asistencias generados por el equipo y queda en un segundo plano la capacidad de matar a los subditos.
6. blueAvgLevel
   Lo que mas afecta nuestra variable de nivel es la capacidad de conseguir oro por parte del equipo, lo que hace mucho sentido porque todas las maneras de conseguir oro nos dan de igual manera experiencia.
7. blueGoldPerMin
   En esta variable podemos ver que es lo que nos aporta mas oro y podemos observar que las muertes de los campeones enemigos y la participación de los compañeros es lo que mas nos aporta experiencia de manera porcentual, de igual manera nos muestra mayor influencia al momento de la victoria del equipo.

De manera reciproca tenemos la importancia de estas variables para el equipo rojo.
El resultado de la relevancia del oro contra los minions asesinados parece que esta menos relacionada con lo que debería estar, por lo que miraremos un poco mas de cerca para observar el porque de este resultado:

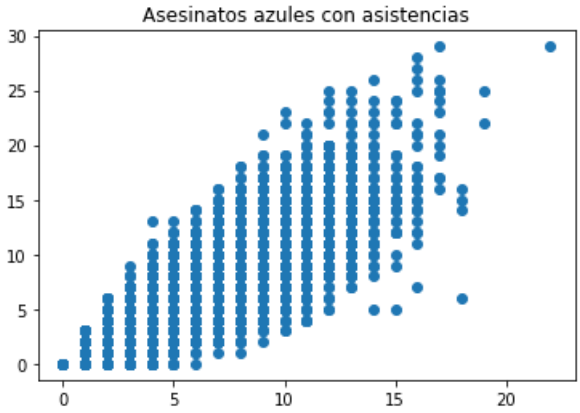


Como podemos observar en la gráfica, el heatmap estaba correcto, si existe una relación entre la cantidad de minions asesinados con la cantidad de oro generada, sin embargo vemos que nuestros datos están muy dispersos por lo que asumimos que es correcto que hay variables que afectan mas a la cantidad de oro generado, como lo podrían ser los asesinatos.



En esta gráfica se puede notar de una manera mas clara la tendencia que hay con la cantidad de asesinatos al incremento de oro del equipo, podríamos incluso pasar una recta que se ajustara a estos datos.

De igual manera buscamos la relación de los asesinatos con las asistencias y podemos ver cierta tendencia aunque hay mayor dispersión en nuestros datos, ya que no es indispensable un compañero para lograr asesinatos.



4. Resultados

Modelo	Resultados
Regresión logística	0.7267206477732794
Naive Bayes	0.7267206477732794
Decision Tree	0.6829959514170041
Random forest	0.7323886639676114

Table 1. Resultados del entrenamiento.

Como podemos observar el modelo que nos da mas precisión es Random Fores, aunque este no se encuentra tan alejado de la regresión logística y naive bayes que tienen la misma precisión, por otro lado Decision Tree si nos arroja un resultado malo en general, nos da un score muy bajo, por lo que debemos apegarnos al resultado en Random Forest, este resultado en términos del juego nos indica que si podemos ver cierta tendencia en los primeros 10 minutos de partida para determinar un resultado en la partida, pero no podemos determinar con certeza el resultado del juego, hay maneras de remontar las partidas con habilidad o con errores cometidos por los enemigos.
Como las variables mas importantes para determinar el resultado

de una partida tenemos:

Variable	Relación
blueWins	Oro total, Experiencia recibida, Oro por minuto
blueKills	Ventaja de oro, Asistencias, Diferencia de nivel
blueAssists	Asesinatos aliados, Oro del equipo
blueTotalGold	Asesinatos, Asistencias, Subditos
blueAvgLevel	Asesinatos, Asistencias, Subditos,Oro
blueGoldPerMin	Asesinatos, Asistencias, Subditos,Oro

Table 2. Variables dependientes.

Como podemos observar de manera general las kills se repiten mayormente en relación a todas las demás variables por lo tanto si queremos generar una gran ventaja en los primeros 10 minutos de juego esa seria la variable en la que mas atención deberíamos tener y generar la mayor cantidad posible para ganar la partida.

5. Conclusión

Los primeros 10 minutos de una partida de league of legends no es suficiente para determinar quien será el ganador, aunque si hay variables que pueden impactar la victoria de una manera muy importante. Los jugadores profesionales se centran mucho en el asesinato correcto de súbditos y de conseguir objetivos mas que de generar muertes para su equipo, se podría decir que dejan las peleas como último recurso pero nuestro estudio no es de jugadores profesionales, si no de jugadores con un alto nivel pero que no están en completa comunicación ni sintonía que los equipos profesionales, por lo que este trabajo seria bueno contrastarlo con partidas de equipos profesionales para ver las diferencias que tienen los jugadores normales a los profesionales, seria interesante ver cuales son las condiciones de victoria en los 10 minutos de los profesionales ya que esta casi eliminado el factor de que puedan ser remontados por errores individuales.

References

[1] Huang, G.-B. Bin et al. (2006) 'Extreme learning Mach', Neurocomputing, 70, pp. 489–501. doi: 10.1016/j.neucom.2005.12.126.

[2] Ren, G., Wang, Y., Ning, J., Zhang, Z. (2020). Using near-infrared hyperspectral imaging with multiple decision tree methods to delineate black tea quality. Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy, 118407.

[3] Proniewska, K., Pregowska, A., Malinowski, K. P. (2020). Identification of Human Vital Functions Directly Relevant to the Respiratory System Based on the Cardiac and Acoustic Parameters and Random Forest. IRBM.

[4] Pérez, R. G., Pino, G. G., Ballester, D. G., Moreno, R. G. (2010). Modelo de regresión logística para estimar la dependencia según la escala de Lawton y Brody. SEMERGEN-Medicina de Familia, 36(7), 365-371.