TAREA 2: Parte A
Regresión Logística

Profesor: Fernando Crema Garcia
Fecha: 27 de Noviembre 2023

La nota final será basada el promedio de cada uno de los modelos ponderados por un coeficiente **extra** dependiendo de la calidad del reporte asociado.

R. Logística	R. Lineal	Clasificación	Extra
RL	R	C	α

Fecha de entrega: 04/12/2023 23:59 pm Caracas.
Penalizaciones: 2 ptos / día. Máximo 10 ptos.
Nota final: $\text{mín}\left\{20\alpha\left(\frac{RL+R+C}{\text{puntos totales}}\right), 20\right\}$

I. Introducción

I-A. Objetivo

El objetivo de esta actividad es reforzar las habilidades adquiridas hasta el momento para aprendizaje supervisado específicamente para los casos:

1. Regresión Lineal
2. Regresión Logística
3. Clasificación con k-vecinos.

Siempre asumiendo que, para cada caso, aplicamos correctamente los conocimientos de preprocesamiento, selección de modelos y pipeline de proyectos en aprendizaje supervisado.

I-B. Reproducibilidad y evaluación

Cada una de las secciones tiene la misma ponderación y al final será la suma de todos los ejercicios que lograron resolver ponderada por un criterio del grupo docente (α).

Para lograr que el grupo docente reproduzca sus resultados, deben asignar al comienzo de su entregable la semilla referente a su **cédula de identidad**

I-C. Entregable

El entregable estará compuesto por un solo notebook que debe respetar las siguientes restricciones:

1. Un Jupyter Notebook con uso adecuado de los ítems de headings (#) separando cada nivel de acuerdo al tipo de modelo y nombre del mismo. Esta vez, se deja a juicio de ustedes la configuración del mismo.
2. Asuman que el Notebook lo va a leer una persona del equipo de toma de decisiones de una empresa pero que tiene conocimiento previo en Aprendizaje Automático. Sean tan detallados como sea posible justificando cada decisión. **No hay soluciones únicas**.
3. Incentivo la creatividad de ustedes y formará parte de la evaluación de α
4. **No se pueden usar soluciones que provee Kaggle para los casos de Regresión Logística y Clasificación**. Nunca es un problema revisar soluciones de terceros. Sin embargo, su solución debe ser producto de **su** trabajo. Plagios serán penalizados con nota mínima.
5. Incentivo la colaboración como descrito en el **código de honor** de la materia. Sin embargo, copias serán penalizadas con nota mínima.

email: fernando.cremagarcia@kuleuven.be

II. Regresión Logística

La popular empresa de desarrollo de videojuegos Riot Games recientemente introdujo un modelo que calcula la probabilidad de que un **equipo profesional** gane en base a situaciones similares de experiencias pasadas en su popular juego https://www.leagueoflegends.com/es-es. Una introducción sencilla al juego se puede ver en ¿Qué es League of Legends?

Citando su artículo sobre el tema dev: Probabilidad de Victoria con tecnología AWS: El porcentaje de probabilidad de victoria (PV) se obtiene a partir del número de equipos que se enfrentaron a una situación similar en el pasado y ganaron la partida. Nuestra estadística de PV se desarrolla utilizando un algoritmo de aprendizaje automático (AA) llamado xgboost, que tiene en cuenta muchos factores en su iteración actual.

II-A. Variables usadas en este modelo

En su artículo, el equipo de datos de Riot explica cuáles son las variables usadas para su modelo:

1. Tiempo de la partida (tiempo dentro de la partida)
2. Porcentajes de oro (oro del jugador / oro total en la partida)
3. XP total de equipo
4. Número de jugadores con vida
5. Derribos de torretas
6. Asesinatos de dragón (si un equipo tiene un alma de dragón o no)
7. Baratija de Heraldo en el inventario

8. Temporizadores de inhibidor (cuánto tiempo tarda en reaparecer un inhibidor) para cada inhibidor
9. Temporizadores de Barón (tiempo hasta que expira la mejora de Barón del equipo)
10. Temporizador de Dragón Ancestral (tiempo hasta que expira la mejora del Dragón Ancestral del equipo)
11. Número de jugadores con el Barón activo
12. Número de jugadores con el Dragón Ancestral activo

Sin embargo, se han dado cuenta que el modelo funciona bien solo a nivel profesional y es complicado implementar una funcionalidad para su API de desarrollo por lo que XGBoost ha dejado de funcionar [1]

II-B. Cómo se ve este modelo?

En la figura 1 podemos ver un ejemplo de la aplicación del modelo para explicar al usuario los cambios en probabilidad en el tiempo.

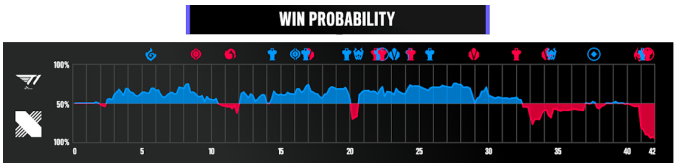

Figura 1. Obtenemos la probabilidad de victoria en el tiempo. Si la probabilidad de que el equipo azul es mayor que la del rojo, la gráfica tiene color azul y, en caso contrario, toma color rojo. En la parte superior de la gráfica hay íconos que se refieren a eventos de importancia que modifican radicalmente la probabilidad de victoria. Por ejemplo, asesinatos entre campeones, torres tiradas o dragones conseguidos.

II-C. Necesitamos tu ayuda

El equipo de datos de Riot nos ha pedido realizar un modelo cuyo output sea: **Probabilidad de que el equipo Azul Gane luego de 10 minutos**. Para hacer el problema más sencillo, usamos un conjunto de datos donde en el minuto 10 almacenamos variables similares a las que usa Riot.

Nuestro trabajo, además de conseguir el modelo, es explicar detalladamente cuáles variables seleccionamos y una breve explicación de la importancia en el modelo. Para ello, nos han pedido revisar la sección de Selección de variables y probar **dos** métodos que nos llamen la atención.

Además de ello, piden como entrega un Pipeline que involucre también el método selección de modelos justificado en su entrega.

Como tienen poco tiempo para revisar el informe, piden con ahínco que los resultados sean explicados con gráficas que puedan mostrar al equipo de negocios y decidir si seguir adelante con el proyecto (usando XGBoost) y datos nuevos [2].

[1] XGBoost sigue funcionando... pero lo usaremos luego =D
[2] Uno de los proyectos puede ser hacer esto!

II-D. Los datos disponibles

Usaremos el dataset disponible en Kaggle Diamond ranked games after 10 minutes. Los datos están divididos en 3 secciones:

1. gameId
2. Variables del equipo azul con prefijo blue.
3. Variables del equipo rojo con prefijo red.

Para cada tipo red y blue tenemos las variables:

1. **Wins**: Ganó el equipo o no.
2. **WardsPlaced**: Número de centinelas puestos por el equipo.
3. **WardsDestroyed**: Número de centinelas destruídos por el equipo.
4. **FirstBlood**: Tuvieron primer asesinato o no.
5. **Kills**: Número de asesinatos.
6. **Deaths**: Número de muertes.
7. **Assists**: Número de asistencias.
8. **EliteMonsters**: Número de monstruos élite asesinados.
9. **Dragons**: Número de dragones asesinados.
10. **Heralds**: Número de heraldos asesinados.
11. **TowersDestroyed**: Torres destruídas.
12. **TotalGold**: Oro total.
13. **AvgLevel**: Average del nivel de las leyendas.
14. **TotalExperience**: Experiencia total.
15. **TotalMinionsKilled**: Número total de minions asesinados.
16. **TotalJungleMinionsKilled**: Número total de minions de la jungla asesinados.
17. **GoldDiff**: Diferencia de oro.
18. **ExperienceDiff**: Diferencia de experiencia.
19. **CSPerMin**: Número de minions asesinados por minuto.
20. **GoldPerMin**: Oro por minuto.

Una sección de preprocesamiento de datos es necesaria y pueden usar las gráficas que deseen siempre que expliquen detalladamente lo que hacen.