Predicción de Resultados de la NFL Utilizando Métodos de Aprendizaje Supervisado

Gonzalo Peralta

02/10/2022

1 Título

Predicción de Resultados de la NFL Utilizando Métodos de Aprendizaje Supervisado

2 Introducción

Para estructurar un proyecto por primera vez es recomendable seguir alguna estructura particular, estandarizada.

La aplicación Coockiecutter provee una estructura estándar que parece adecuada para esto.

Para algunos ejemplos de reportes técnicos de proyectos, revisa los proyectos finales del curso CS229: Machine Learning de Standford.

- 2021.
- 2019.
- 2014.

2.1 Formulación del Problema

Se plantea crear un clasificador capaz de predecir las probabilidades de que un determinado equipo de fútbol americano de la liga profesional NFL gane un determinado partido antes de que ocurra, esto con mayor precisión que los métodos utilizados actualmente por apostadores tales como el modelo desarrollado por la plataforma FiveThirtyEight.

En un sentido técnico se puede expresar de la siguiente manera:

Sea $X = x_1, x_2, ..., x_n$ un vector de datos correspondiente a un enfrentamiento entre dos equipos de la liga profesional de fútbol americano NFL (Local y Visitante), y C_k tal que:

$$C_0 = Derrotaequipolocal$$

$$C_1 = Victoriaequipolocal$$

C_k es la variable que de pretende predecir a partir de X, así mismo, a cada C_k le corresponde una probabilidad $P(C_k = C_1)$ donde $C_1 = 1 - C_0$

Por lo tanto, el objetivo es el de minimizar las asignaciones de C_k a la clase incorrecta por medio de un modelo que permita también determinar la probabilidad de que tal asignación se cumpla. Para eso, X se partirá en dos regiones de decisión utilizando un modelo de clasificación, tal que los puntos x en una determinada región R_k son asignados a su clase correspondiente C_k. Se puede decir que un error ocurrirá cuando un valor de X que pertenece a C_k es asignado a la clase contraria, matemáticamente la probabilidad de un error es:

$$P(E) = P(x \in R_1, C_2) + P(x \in R_2, C_1)$$

Finalmente, el objetivo general matemáticamente se expresa:

$$\arg\min_x P(E)$$

El sistema debe devolver un porcentaje de probabilidad que tiene un equipo de ganar el juego que tiene más próximo, el problema es supervisado ya que se comparará el resultado del modelo para juegos anteriores y se medirá la precisión para realizar la clasificación, así mismo, se trata de un problema online ya que se pretende insertar datos nuevos a medida que la temporada 2022 avanza, con la esperanza de aumentar la precisión según aumente la información disponible.

En cuanto a la evaluación del modelo, se utilizarán las métricas "Accuracy", "F1 Score" y la curva ROC, esto debido a que no existe una ponderación de los errores Tipo I y Tipo II.

El objetivo mínimo es el de superar la precisión alcanzada por el algoritmo de FiveThirtyEight, que el año pasado registró un 62% de juegos clasificados correctamente (2021).

3 Datos

Dado el objetivo, se puede deducir que es importante disponer de una fuente de información vasta y que contenga variables significativas para la correcta clasificación.

La base de datos que se utilizará proviene de un repositorio creado por entusiastas del análisis estadístico en el fútbol americano, y contiene información recopilada "Play By Play", es decir que cada linea de la matriz corresponde a una sola jugada en un determinado juego, esto

implica que la base de datos contiene alrededor de 50,000 lineas por cada temporada, así como 372 variables de decisión. Se puede acceder al repositorio a través de una librería de Python llamada “nfl-data-py”.

Una vez realizado el análisis de selección de variables se seleccionaron las siguientes:

Variable	Descripción
game_id	Identificador de diez dígitos para el juego de la NFL.
home_team	Abreviatura para el equipo local.
away_team	Abreviatura para el equipo de visitante.
season_type	Reg ‘o’ Post ’que indica si el juego pertenece a la temporada regular o playoffs. (Dummy)
season	Número de 4 dígitos que indica a qué temporada pertenece el juego.
week	Semana de la temporada.
game_date	Fecha del juego.
home_score	Puntos totales anotados por el equipo local.
away_score	Puntos totales anotados por el equipo visitante.
result	La diferencia entre Home_Score y Away_score, es el resultado del juego desde la perspectiva del equipo local.
div_game	Indicador binario para si el juego dado era un juego entre rivales de división.
location	Indica si el equipo local jugaba en su estadio o en un sitio neutral (Como fuera de los EEUU o en un estadio prestado). (Dummy)
roof	Indica el tipo del techo del estadio en el que se jugó el juego. (Dummy)
surface	En qué tipo de terreno se jugó el juego. (Dummy)
sp	Indicador binario de si se produjo o no una puntuación en la jugada.

Variable	Descripción
yards_gained	Yardas numéricas ganadas (o perdidas) por el equipo con posesión, excluyendo los yardas obtenidas a través de recuperaciones de balón suelto y laterales.
yards_after_catch	Valor numérico para la distancia en yardas perpendiculares desde la línea donde el receptor hizo la recepción hasta donde terminó la jugada.
epa	Puntos esperados agregados (EPA) por el equipo con posesión para la jugada dada.
air_epa	EPA solo de las yardas aéreos. Esto representa el valor real proporcionado a través del aire. Para incompletos, esto representa el valor hipotético que podría haberse agregado a través del aire si se hubiera completado el pase.
yac_epa	EPA de los yardas después de la recepción. Para incompletos, esto representa la diferencia entre la Air_EPA hipotética y la EPA real de la jugada.
comp_air_epa	EPA solo desde las yardas aéreas solo para pases completos.
comp_yac_epa	EPA de las yardas después de la captura solo para pases completos.
wpa	Probabilidad de ganar agregado (WPA) para el equipo con posesión.
vegas_wpa	Probabilidad de ganar agregado (WPA) para el equipo con posesión ajustado para los momios de Las Vegas.
air_wpa	WPA con la misma lógica que Air_EPA.

Variable	Descripción
yac_wpa	WPA con la misma lógica que YAC_EPA.
comp_air_wpa	El AIR_WPA solo de pases completos.
comp_yac_wpa	El YAC_WPA solo de pases completos.
third_down_converted	Indicador binario para si el primer down se convirtió en el tercer intento.
fourth_down_converted	Indicador binario para si el primer down se convirtió en el cuarto intento.
interception	Indicador binario para si el pase fue interceptado.
fumble_lost	Indicador binario para si el balón suelto se perdió.
qb_hit	Indicador binario si el QB fue golpeado en la jugada.
rush_attempt	Indicador binario para si la jugada fue una carrera por tierra.
pass_attempt	Indicador binario para si la jugada fue un intento de pase (incluye capturas).
sack	Indicador binario para si la jugada terminó en una captura de QB.
touchdown	Indicador binario para si la jugada dio como resultado un TD.
success	Indicador binario para si el EPA > 0 en la jugada.
penalty	Indicador binario de si ocurrió o no un castigo.
penalty_yards	Yardas ganadas (o perdidas) por el equipo con posesión gracias al castigo.

Aquí describes los datos que usaste, sus fuentes, variables, etc. Deberá contener la siguiente información:

1. Justificación concisa de por qué el conjunto de datos elegido es relevanta para el problema elegido.

2. Describir las fuentes de los datos.
3. Describir qué procesamiento se hizo para dejarlos en estado usable.
4. Describir las variables que contienen los datos (e.g., codebook, en caso de que se usen abreviaturas para las variables).

4 Métodos y Análisis

La figura Figura 1 muestra el proceso que estaremos siguiendo en esta fase.

IDI-III tratará principamente de estudiar el problema, limpiar y transformar los datos y seleccionar las variables, pero también comenzaremos a escribir y familiarizarnos con las herramientas de publicación (en este caso, Quarto).

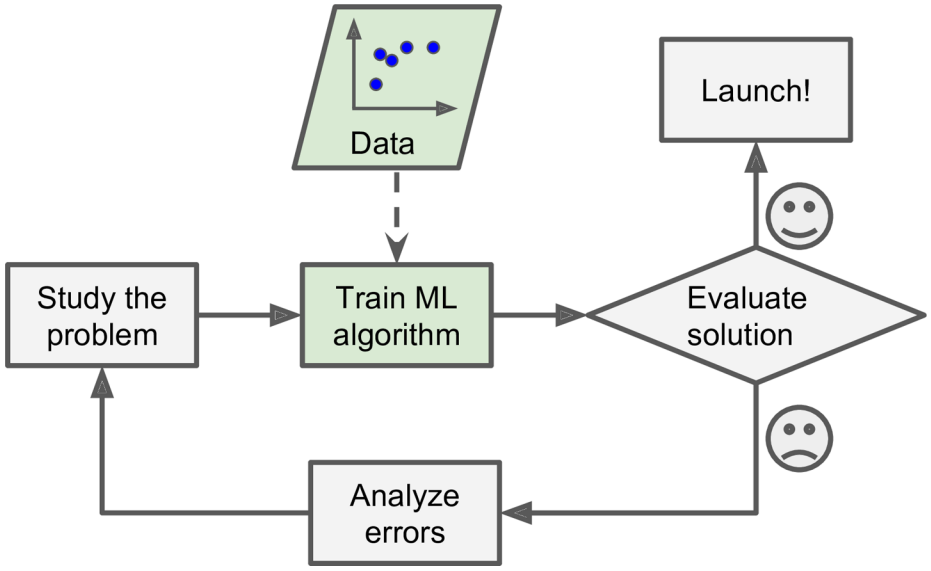

Figura 1: Proceso típico de proyecto de DS. Tomado de Géron (2019).

1. Análisis exploratorio. Añadir tablas, gráficos exploratorios, etc. Esto no es propiamente un resultado, sino un análisis que se realiza para justificar otras decisiones.
2. Si se realizan transformaciones en una variable (e.g., se log-transformó, se exponenció, se escaló, se normalizó, etc) o cualquier ingeniería de características, extracción, etc., a partir de los datos exploratorios. Justificar la decisión.
3. Descripción de los métodos, como algoritmos, benchmarks, métricas de comparación (e.g., $RMSE$) etc. *No se colocan esos resultados aquí*, solo se menciona qu+e se utilizó.

Quarto soporta renderización de ecuaciones usando la sintaxis de LaTeX. Ver este artículo y este para ver cómo escribir matemáticas.

5 Resultados

En esta sección se colocan los resultados principales, como tablas de comparación, gráficos, etc. Por ejemplo, si se probó un algoritmo con respecto a otros tres, se coloca la ejecución en diversas métricas.

Las figuras y tablas deben ir acompañadas de una etiqueta y una breve descripción. Se pueden referenciar usando `@{label}` en donde `label` en este caso específico es `fig-polar`. La referencia renderizada se verá así **?@fig-polar**.

Las tablas pueden hacerse con Markdown

Tabla 2: Leyenda de tabla

Col1	Col2	Col3
A	B	C
E	F	G
A	G	G

Ver la tabla Tabla 2.

6 Referencias

Para citar, usar `(@alcala2021statistical)` que se renderiza como (López-Cárdenas et al. (2021)). La entrada `@alcala2021statistical` debe estar tal cual en el archivo `referencias.bib`. Las referencias en formato de bibtex se pueden obtener desde Google Scholar.

Para imprimir las referencias hay que colocar

```
::: {#refs}
:::
```


Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

López-Cárdenas, P. G., Alcalá, E., Sánchez-Torres, J. D., & Araujo, E. (2021). A Resampling Approach for the Data-Based Optimization of Nanosensors. *2021 18th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, 1-4. https://doi.org/10.1109/CCE53527.2021.9633114