Sport Data Campus

Análisis interactivo de la NFL

Trabajo de Fin de Máster - Big Data Deportivo

Juan Marcos Díaz
2025

Índice

Resumen

Este proyecto se plantea como una solución frente a la gran diferencia que se presenta entre la generación de datos en la NFL y la dificultad para su análisis por parte de aficionados y expertos. Para ello, se ha creado "**NFL Analytics Hub**", una aplicación web interactiva desarrollada en Python con Streamlit, que analiza el rendimiento de equipos y jugadores de la liga durante las últimas temporadas. Utilizando la librería *nfl-data-py* como fuente de datos, que incluye métricas avanzadas como el EPA, se aplicó un proceso ETL para estructurar la información.

La plataforma ofrece dashboards dinámicos para el análisis de equipos y jugadores, comparativas "Head-to-Head" mediante gráficos de radar basados en percentiles y visualización de tendencias temporales mediante gráficos de líneas. El núcleo del proyecto integra modelos de Machine Learning no supervisado: Clustering K-Means para identificar arquetipos de jugadores según su estilo de juego y un sistema basado en Análisis de Componentes Principales (PCA) para construir un buscador de perfiles estadísticos similares.

El resultado es una herramienta que centraliza y democratiza el acceso a la analítica deportiva avanzada, transformando datos brutos en conocimiento visual e intuitivo.

Palabras Clave: NFL Analytics, Big Data, Visualización de Datos, Streamlit, Python, Machine Learning, Aprendizaje no Supervisado, Clustering K-Means, Análisis de Componentes Principales (PCA), Analítica Deportiva, EPA.

Capítulo 1 - Planteamiento del proyecto

1.1 Introducción: La revolución analítica en el deporte

En los últimos años, el deporte profesional, y en menor medida el deporte amateur, ha experimentado una transformación silenciosa pero profunda, liderada por la importante aparición del análisis de datos. Actualmente, en todos los sectores incluido el deporte, se genera y se almacena constantemente un gran volumen de datos de todo tipo en lo que se conoce como la era del Big Data.

Por lo tanto, el Big Data puede entenderse como un concepto que engloba conjuntos de datos de tal magnitud, complejidad y velocidad de crecimiento que las herramientas de software convencionales resultan insuficientes para su gestión, procesamiento o análisis. Su principal característica se define a través de las "5 Vs" (Figura 1.1):

- **Volumen**: la enorme cantidad de datos generados, que pueden ir desde terabytes hasta zettabytes.
- **Velocidad**: la alta velocidad a la que se generan, reciben y procesan los datos, a menudo en tiempo real.
- **Variedad**: los diferentes tipos de datos, que incluyen datos estructurados (como bases de datos SQL), semi-estructurados (como ficheros XML, JSON) y no estructurados (como texto, imágenes, vídeos).
- **Veracidad**: la calidad, fiabilidad y exactitud de los datos, es decir, que los datos son correctos y dignos de confianza.
- **Valor**: la capacidad de transformar los datos en información útil que genere un valor tangible, ya sea a través de mejores decisiones de negocio, optimización de procesos o nuevos productos.

El análisis de datos surge como respuesta a esta situación, permitiendo mediante técnicas avanzadas transformar dichos datos crudos en información de gran utilidad y valor para la toma de decisiones de cualquier entidad, empresa o individuo.

En el mundo del deporte, la toma de decisiones, antes dominada por la intuición y la experiencia acumulada de entrenadores, ojeadores y cuerpos técnicos, se ha visto progresivamente complementada y, en muchos casos, redefinida por la evidencia estadística extraída de grandes volúmenes de información. Este fenómeno, popularizado masivamente por la historia de los Oakland Athletics en el béisbol, narrada en el libro y posterior película "Moneyball", se ha consolidado como una base fundamental para la gestión de las franquicias deportivas (Figura 1.2).

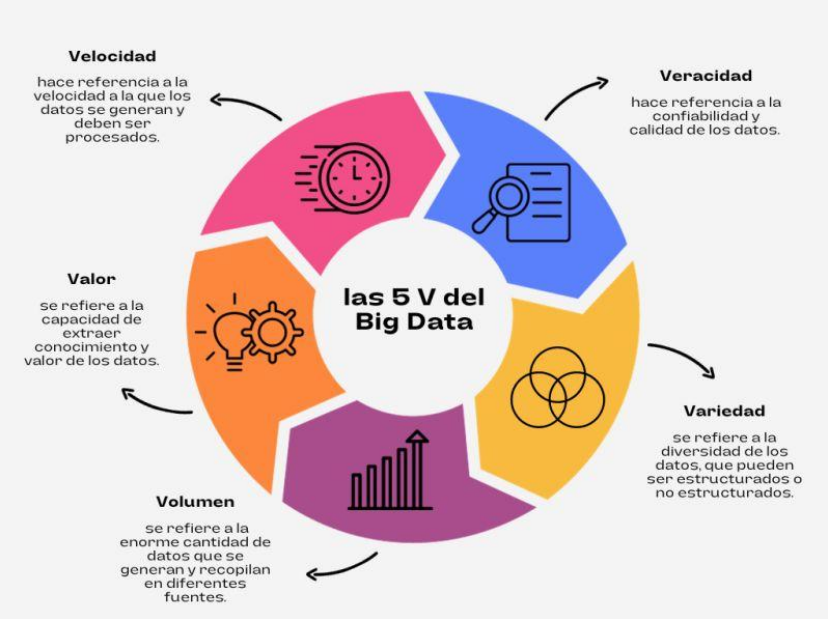


Figura 1.1: Representación gráfica de las 5 Vs del Big Data. Fuente: Google Images

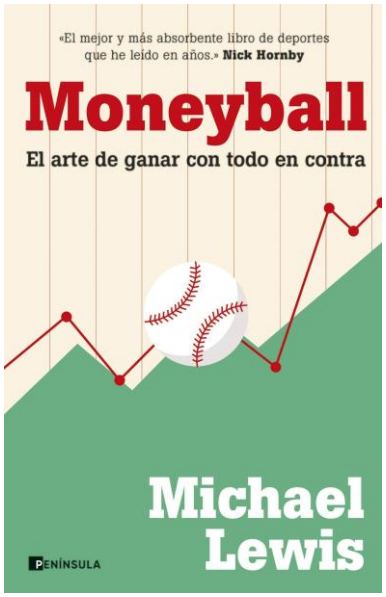


Figura 1.2: El fenómeno "Moneyball" popularizó el uso de la analítica avanzada en el deporte (Michael Lewis, 2003). Fuente: Google Images

El fútbol americano no es menos y en particular, la National Football League (NFL), siendo la liga que mayor volumen de ingresos genera a nivel mundial, así como uno de los ecosistemas deportivos más complejos y estratégicos, no ha sido ajena a esta revolución.

Figura 1.3: Logo de la NFL. Fuente: Google Images

Las estadísticas tradicionales, como las yardas de pase o las intercepciones, si bien siguen siendo de gran relevancia, han demostrado ser insuficientes para capturar el verdadero valor y la eficiencia de un jugador o una jugada. Por este motivo, durante los últimos años han surgido métricas avanzadas como los Puntos Añadidos Esperados (Expected Points Added - EPA), que cuantifican el impacto de cada acción en la probabilidad de anotar, o el Porcentaje de Pases Completados sobre lo Esperado (Completion Percentage Over Expected - CPOE), que mide la precisión de un quarterback ajustada a la dificultad de sus lanzamientos.

Este cambio a la hora de interpretar el juego ha propiciado un nuevo clima competitivo. Los equipos que logran integrar eficazmente estos análisis en su estrategia de fichajes, preparación de partidos y toma de decisiones en tiempo real, obtienen una ventaja significativa. Sin embargo, este volumen de datos presenta un desafío considerable: su volumen, complejidad y la necesidad de herramientas especializadas para su correcta interpretación.

1.2 Definición del problema y justificación

El principal problema que este proyecto busca abordar es la brecha existente entre la inmensa cantidad de datos estadísticos, tanto simples como avanzados generados por la NFL y la capacidad de los analistas, entrenadores, o aficionados para acceder a ellos, visualizarlos, interpretarlos y extraer conclusiones de manera sencilla y eficiente.

Aunque mediante este estudio no se llegue a realizar un análisis detallado y exhaustivo de equipos y jugadores (como se realizaría en un departamento de analytics), sí que se busca trasladar los datos y estadísticas brutas en gráficos y KPI’s de fácil interpretación para cualquier usuario, desde el más amateur al analista experto.

Así pues, analizando el problema a tratar, podemos observar que el análisis profundo de los datos generados por la NFL se enfrenta a distintas barreras:

- **Dispersión de la información:** los datos se encuentran en diversas plataformas, a menudo requiriendo conocimientos de programación para su extracción y procesamiento. Además, algunas de las métricas más avanzadas se encuentran sólo en portales y páginas de pago como Pro Football Focus (PFF), para los que se requiere una suscripción.

- **Complejidad de las métricas:** las estadísticas avanzadas, aunque muy potentes, no son intuitivas y requieren un contexto adecuado para su correcta interpretación.

- **Herramientas estáticas:** la mayoría de los informes y análisis disponibles son estáticos, impidiendo al usuario explorar los datos de forma interactiva, aplicar filtros personalizados o realizar comparaciones directas entre diferentes entidades.

Este proyecto se justifica por la necesidad de crear una solución centralizada e interactiva que democratice el acceso al análisis de datos de la NFL. Al desarrollar una aplicación web, "NFL Analytics Hub", se busca motivar a sus usuarios, permitiéndoles pasar de ser consumidores pasivos del juego a ser exploradores activos del dato y partícipes de la toma de decisiones. Una herramienta de estas características no solo sirve como un recurso de consulta, sino que también fomenta una comprensión más profunda y matizada del juego, permitiendo identificar patrones, evaluar el rendimiento de forma objetiva y fundamentar opiniones con evidencia estadística sólida.

1.3 Objetivos del proyecto

Para dar solución al problema planteado, se han establecido los siguientes objetivos:

Objetivo general:

Crear "NFL Analytics Hub", una aplicación web interactiva destinada al análisis estadístico avanzado del desempeño de equipos y jugadores de la NFL, con un enfoque en las temporadas de 2020 a 2024. Para el desarrollo de la aplicación web, se utilizará la librería Streamlit de Python, reconocida por su facilidad de integración, sus amplios recursos interactivos y su capacidad de visualización.

Objetivos específicos:

- Crear dashboards dinámicos para visualizar el rendimiento ofensivo y defensivo de los 32 equipos de la liga, permitiendo al usuario aplicar filtros por año, conferencia y/o división.

- Implementar una herramienta de comparación directa ("Head-to-Head") que permita contrastar el perfil estadístico de dos equipos o dos jugadores de forma visual e intuitiva.

- Aplicar técnicas de Machine Learning no supervisado (Clustering y PCA) para identificar arquetipos de jugadores según su estilo de juego y construir un sistema de búsqueda de perfiles estadísticamente similares.

- Desarrollar visualizaciones de la evolución temporal del rendimiento, permitiendo analizar la progresión, consistencia o declive de equipos y jugadores a lo largo de las cinco temporadas estudiadas.

Capítulo 2 - Búsqueda y descripción de los datos

2.1 Fuentes de datos en la NFL

El ecosistema de datos de la NFL es muy amplio y diverso, presentando distintas opciones según el nivel de acceso, granularidad y coste. A grandes rasgos, las fuentes de datos se pueden clasificar en tres categorías principales:

- **Fuentes oficiales (NFL Next Gen Stats):** la propia liga, en colaboración con AWS y a través de su iniciativa "Next Gen Stats", recopila datos de posicionamiento de jugadores en tiempo real mediante sensores en las hombreras (RFID). Estos datos son de una riqueza inigualable, ya que permiten calcular velocidades, aceleraciones y distancias. Sin embargo, el acceso público a estos datos crudos es extremadamente limitado y se reserva principalmente para las propias franquicias, socios de la liga y eventos específicos como el "Big Data Bowl".

- **Proveedores comerciales:** empresas como Sportradar o Stats Perform son proveedores oficiales de datos de la NFL para casas de apuestas, medios de comunicación y las propias franquicias. Ofrecen APIs robustas y datos muy fiables, pero operan bajo un modelo de suscripción con costes elevados, lo que los hace inviables para proyectos académicos.

- **Proyectos de código abierto (Open Source):** en los últimos años, la comunidad de analistas de datos ha desarrollado proyectos extraordinarios para recopilar y distribuir datos de la NFL de forma gratuita. El paquete nflscrapR, si bien fue pionero, operaba mediante un proceso de *web scraping* individualizado, un método que resultaba lento y frágil.
  Para solucionar estas deficiencias, en 2020, los analistas Ben Baldwin y Sebastian Carl lanzaron nflfastR. Este nuevo paquete para el lenguaje R revolucionó el proceso al cambiar el paradigma: en lugar de que cada usuario realice scraping, el equipo de nflfastR procesa los datos una única vez, los enriquece con métricas avanzadas cruciales como el EPA (Expected Points Added) y los aloja en un repositorio público de acceso rápido.
  Finalmente, para hacer accesible este valioso ecosistema al entorno de Python, se desarrolló la librería **nfl-data-py**. Esta librería, utilizada en el presente proyecto, actúa como un puente directo al repositorio de datos mantenido por nflfastR, permitiendo beneficiarse de la velocidad, fiabilidad y riqueza de la fuente de datos que se ha convertido en el estándar de la industria.

2.2 Fuente seleccionada: nfl-data-py

Para la realización de este proyecto, la fuente de datos seleccionada ha sido la librería de Python **nfl-data-py**. Esta elección se fundamenta en una serie de ventajas clave que la convierten en la mejor opción:

- **Acceso gratuito y programático:** al ser una librería de Python, permite la extracción y manipulación de los datos directamente en el entorno de trabajo del proyecto, facilitando la automatización del proceso ETL (Extract, Transform, Load).

- **Riqueza y granularidad de los datos:** nfl-data-py proporciona acceso a los datos de Play-by-Play, que es el nivel de detalle más alto disponible públicamente. Cada fila de este dataset representa una única jugada, con más de 300 variables que describen todo lo que ocurrió.

- **Calidad y fiabilidad:** la librería se nutre directamente de los datos procesados por el proyecto nflfastR, ampliamente reconocido y validado por la comunidad analítica por su precisión y su metodología de limpieza y enriquecimiento de datos.

- **Inclusión de métricas avanzadas:** los datos de nfl-data-py no solo incluyen estadísticas tradicionales, sino que ya vienen enriquecidos con métricas avanzadas fundamentales para este proyecto, como el EPA (Expected Points Added) ahorrando una cantidad significativa de tiempo en el preprocesamiento.

- **Comunidad activa:** al ser un proyecto de código abierto popular, cuenta con una comunidad activa y una buena documentación, lo que facilita la resolución de dudas y problemas.

2.3 Descripción de los datasets finales

Tras el proceso de extracción, transformación y carga, se generaron tres ficheros de datos principales en formato CSV, que constituyen la base sobre la que se construye toda la aplicación "NFL Analytics Hub". A través de nfl-data-py, importamos los datos y construimos tres datasets distintos para su posterior análisis:

- **offensive_team_stats_advanced_2020-2024.csv**: datos ofensivos resumidos por equipo y temporada.
- **defensive_team_stats_advanced_2020-2024.csv**: datos defensivos resumidos por equipo y temporada.
- **detailed_player_stats_advanced_2020-2024.csv**: estadísticas simples y avanzadas por

jugador y temporada.

Para simplificar el análisis de los equipos, se ha focalizado el estudio en las jugadas de pase y de carrera, tanto ofensivamente como defensivamente por lo que, aunque sea un aspecto importante del juego, no se tendrán en cuenta los equipos especiales ni los puntos logrados por la unidad defensiva.

A continuación, se describen algunas de las variables más relevantes de cada dataset.

Categoría	Variable	Descripción	Dataset(s)
Generales	team	Abreviatura del equipo (ej. 'KC').	Todos
	year	Temporada (año).	Todos
	conference	Conferencia del equipo (AFC o NFC).	Equipos
	division	División del equipo (ej. 'AFC West').	Equipos
	player_name	Nombre completo del jugador.	Jugadores
	position	Posición principal del jugador (QB, WR, RB, etc.).	Jugadores
Ofensiva (Equipo)	total_yards	Suma de yardas de pase y carrera generadas.	Ofensivo
	total_plays	Suma de jugadas de pase y carrera.	Ofensivo
	yards_per_play	Eficiencia de la ofensiva (Yardas/Jugada).	Ofensivo
	cmp_percentage	Porcentaje de pases completados.	Ofensivo
	net_yards_per_pass	Yardas de pase por intento (ajustado por sacks).	Ofensivo
	yards_per_rush	Promedio de yardas por intento de carrera.	Ofensivo
	total_turnovers	Suma de intercepciones y fumbles perdidos.	Ofensivo
	passing_tds	Touchdowns de pase	Ofensivo

	rushing_tds	Touchdowns de carrera	Ofensivo
	offensive_tds	Suma de los touchdowns de pase y de carrera	Ofensivo
Defensiva (Equipo)	total_yards_allowed	Suma de yardas de pase y carrera permitidas.	Defensivo
	yards_per_play_allowed	Eficiencia de la defensa (Yardas/Jugada permitidas).	Defensivo
	opponent_cmp_percentage	% de pases completados permitido al rival.	Defensivo
	yards_per_pass_allowed	Yardas por intento de pase permitidas.	Defensivo
	yards_per_rush_allowed	Yardas por intento de carrera permitidas.	Defensivo
	turnovers_forced	Suma de intercepciones y fumbles forzados.	Defensivo
	sack_rate	Porcentaje de pases rivales que acaban en sack.	Defensivo
	passing_tds_allowed	Touchdowns de pase permitidos	Ofensivo
	rushing_tds_allowed	Touchdowns de carrera permitidos	Ofensivo
	offensive_tds_allowed	Suma de los touchdowns de pase y de carrera permitidos	Ofensivo
Jugador (Avanzadas)	passing_epa	Puntos Añadidos Esperados en jugadas de pase.	Jugadores
	rushing_epa	Puntos Añadidos Esperados en jugadas de carrera.	Jugadores
	receiving_epa	Puntos Añadidos Esperados en jugadas de recepción.	Jugadores
	wopr	Weighted Opportunity Rating (Cuota de Oportunidad).	Jugadores

	target_share	% de los pases del equipo dirigidos al jugador.	Jugadores
	air_yards_share	% de las yardas aéreas del equipo para un jugador.	Jugadores

Tabla 2.1: Descripción de principales variables seleccionadas en los datasets finales del proyecto. El resto de las variables del dataset de jugadores se puede encontrar en el siguiente enlace de la librería original [nflfastr]

Capítulo 3 - Limpieza y tratamiento de datos

Una vez identificada la fuente de datos, el siguiente paso fundamental en cualquier proyecto de análisis deportivo (o de cualquier otro caso) es el proceso de **Extracción, Transformación y Carga (ETL)**. En este capítulo se detalla dicho proceso de conversión de los datos crudos proporcionados por la librería nfl-data-py en los tres *datasets* estructurados, limpios y ampliados que constituyen la base de la aplicación mencionados en el anterior capítulo. Todo este proceso se ha centralizado en un único script de Python, Data_extraction.py, garantizando la reproducibilidad y consistencia del tratamiento.

3.1 Proceso ETL (Extract, Transform, Load)

El proceso ETL es una secuencia de tres fases que asegura la obtención de los datos, su preparado (pre-procesado, limpieza y *feature engenieering*) y posterior almacenaje óptimo para su análisis y visualización. A continuación, se describe cada una de estas fases en el contexto del proyecto.

1. Extracción (Extract): la primera fase consistió en la extracción de los datos necesarios desde la librería nfl-data-py. Se cargaron tres tipos de datasets distintos para las temporadas 2020-2024.

- **Datos Play-by-Play (pbp_data):** el conjunto de datos más detallado, donde cada fila representa una única jugada de un determinado partido. Es la fuente principal para calcular las estadísticas de equipo de manera agrupada.

- **Datos de temporada por jugador (seasonal_player_data):** un dataset ya pre-agregado que contiene las estadísticas acumuladas de cada jugador para cada temporada. Para obtener los datos de los jugadores apenas hace falta pre-procesado.

- **Datos de plantillas (seasonal_roster_data):** contiene información de los jugadores de cada equipo por temporada, como su nombre, posición y equipo, fundamental para enriquecer el dataset de estadísticas de jugador.

Se decidió trabajar sólo con las últimas 5 temporadas para simplificar el uso de la aplicación, pero el fichero de Python permite la extracción de los datos de las temporadas que se quiera en la propia definición de la función.

```python
def create_nfl_stats_report_advanced(start_year=2020, end_year=2024):
    """
    Genera un reporte completo con estadísticas avanzadas para equipos y jugadores,
    utilizando únicamente datos de la TEMPORADA REGULAR.
    """
    years = list(range(start_year, end_year + 1))
    print(f"Iniciando la generación de reportes para las temporadas: {years}...")

    try:
        pbp_data_full = nfl.import_pbp_data(years=years)
        roster_data = nfl.import_seasonal_rosters(years=years)
        seasonal_player_data_full = nfl.import_seasonal_data(years=years)
        print("Datos cargados exitosamente.\n")

        # --- FILTRADO POR TEMPORADA REGULAR ---
        print("Filtrando datos para mantener solo la Temporada Regular ('REG')...")
        pbp_data = pbp_data_full[pbp_data_full['season_type'] == 'REG'].copy()
        seasonal_player_data = seasonal_player_data_full[seasonal_player_data_full['season_type'] == 'REG'].copy()
        print("Filtrado completado.\n")

    except Exception as e:
        print(f"Error al cargar los datos: {e}")
        return
```

Figura 3.1: Fragmento del código de Data_extraction.py para la extracción de los datasets utilizados. Podemos ver la opción de definir el rango de temporadas y el filtrado por temporada regular.

2. Transformación (Transform): esta es la fase más compleja, en la que los datos crudos se limpian, se reestructuran y se enriquecen con la creación de nuevas variables (*feature engenieering*). Las transformaciones clave realizadas fueron:

- **Filtrado por temporada regular:** para asegurar la consistencia y comparabilidad de los datos, se descartaron todas las jugadas y estadísticas pertenecientes a la postemporada (playoffs), quedándonos únicamente con los datos de la temporada regular (season_type == 'REG'), al trabajar con métricas de volúmenes totales (pases, yardas, TDs, etc.)

- **Agregación de datos:** se agruparon los datos de Play-by-Play por equipo y por temporada para calcular las estadísticas acumuladas de ofensiva y defensivas tanto de pase como de carrera mediante sumatorios con los comandos groupby/agg.

```python
# --- TABLA 1: OFENSIVA AVANZADA POR EQUIPO Y AÑO ---
print("--- Procesando Tabla Ofensiva Avanzada ---")
pass_plays = pbp_data.loc[pbp_data['pass_attempt'] == 1]
rush_plays = pbp_data.loc[pbp_data['rush_attempt'] == 1]

team_pass_offense = pass_plays.groupby(['posteam', 'season']).agg(
    pass_completions=('complete_pass', 'sum'), pass_attempts=('pass_attempt', 'sum'),
    passing_yards=('passing_yards', 'sum'), passing_tds=('pass_touchdown', 'sum'),
    interceptions=('interception', 'sum'), sacks_taken=('sack', 'sum')
).reset_index()
```

Figura 3.2: Agrupación (sumatorio) de las variables por equipo y temporada para las jugadas ofensivas de pase.

- **Creación de métricas:** se calcularon nuevas variables (*features*) a partir de los datos existentes. Esto incluye tanto métricas de totales (cómo total_yards) como métricas avanzadas de eficiencia (cómo net_yards_per_pass).

```python
# CÁLCULO DE MÉTRICAS
offensive_df['total_plays'] = offensive_df['pass_attempts'] + offensive_df['rush_attempts']
offensive_df['total_yards'] = offensive_df['passing_yards'] + offensive_df['rushing_yards']
```

Figura 3.3: Creación de métricas totales sumando jugadas de pase y de carrera.

- **Unión de datos:** se combinaron diferentes dataframes para consolidar la información. Por ejemplo, se unieron las estadísticas de pase y de carrera de los equipos en un único dataset ofensivo. Además, se añadió un diccionario con la Conferencia y División de cada equipo utilizados en posteriores filtros.

```python
offensive_df = pd.merge(team_pass_offense, team_rush_offense, on=['posteam', 'season'], how='outer').fillna(0)
```

Figura 3.4: Unión de los dataframes ofensivos agrupados de pase y carrera.

3. Carga (Load): La fase final del proceso consistió en guardar los tres dataframes transformados (offensive_df, defensive_df, y player_df) en ficheros de formato CSV. Este formato fue elegido por su simplicidad, portabilidad, poco peso y tamaño y fácil integración con la librería Pandas, que cargará estos archivos para su posterior uso en la aplicación web de Streamnlit.

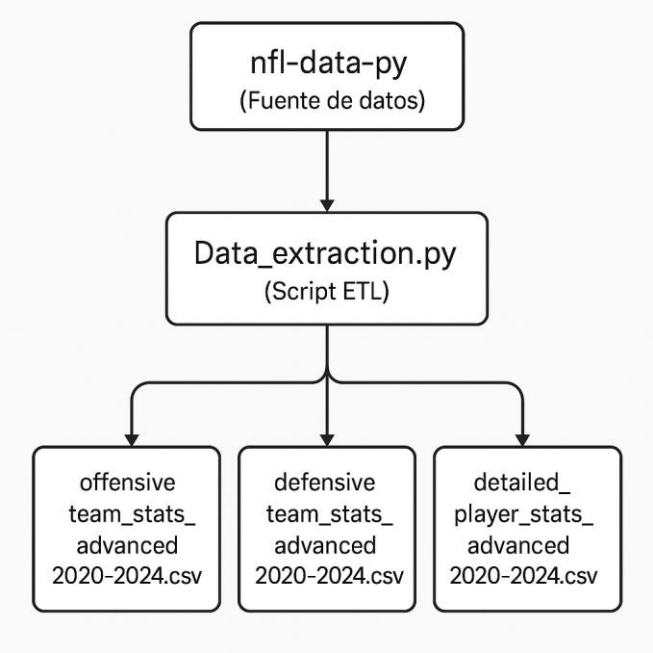


Figura 3.5: Diagrama de flujo del proceso ETL del proyecto. Fuente: elaboración propia en Canva.

3.2 Agregación y creación de métricas

El paso de datos granulares de "jugada a jugada" a estadísticas agregadas (sumatorios) por equipo es la parte más importante de la fase de transformación. Este proceso se realiza mediante la operación groupby() de la librería Pandas, que permite agrupar todas las jugadas pertenecientes a un mismo equipo en una misma temporada. Una vez agrupados los datos, se utiliza la función agg() para aplicar operaciones de agregación (como la suma sum) a las columnas. Por ejemplo, para calcular las estadísticas de pase de cada equipo, se agrupan todas sus jugadas de pase y se suman las yardas, los touchdowns, las intercepciones, etc. (Figura 3.2)

Tras realizar este proceso tanto para las jugadas de pase como para las de carrera, se obtienen dos dataframes separados. Para consolidar toda la información ofensiva en una única tabla, se utiliza la función pd.merge(), que une los dos dataframes utilizando el equipo y la temporada como variables claves. (Figura 3.3)

Este mismo proceso se replica de manera análoga para las estadísticas defensivas, pero agrupando en este caso por la columna defteam (equipo defensor).

3.3 Feature engenieering: creación de métricas avanzadas

El feature enginieering se define como el proceso de crear nuevas variables más completas a partir de las ya existentes. En este proyecto, este paso es muy importante para ir más allá de las estadísticas básicas y generar métricas de eficiencia para poder comparar equipos o jugadores en la aplicación.

Un ejemplo clave es el cálculo de **Net Yards per Pass Attempt (NY/A)** o Yardas Netas por Intento de Pase. Las yardas por intento de pase tradicionales (passing_yards / pass_attempts) son una buena medida, pero incompleta, ya que ignoran un aspecto fundamental del juego aéreo: los derribos al quarterback (sacks). Un sack es una jugada de pase fallida que resulta en una pérdida de yardas. La métrica NY/A ajusta la fórmula para tener en cuenta tanto los intentos de pase como los sacks en el denominador, y las yardas perdidas por sacks en el numerador (aunque en el caso de nfl-data-py, estas ya vienen restadas de passing_yards).

$$\mathbf{NY/A} = \frac{(\boldsymbol{Yardas\ de\ Pase - Yardas\ perdidas\ en\ Sacks})}{(\boldsymbol{Intentos\ de\ Pase + Sacks})}$$

Esta métrica es mucho más robusta, ya que evalúa no solo al quarterback, sino también la capacidad de la línea ofensiva para protegerlo. Un NY/A alto indica una ofensiva aérea verdaderamente eficiente y explosiva.

```
offensive_df['net_yards_per_pass'] = offensive_df['passing_yards'] / (offensive_df['pass_attempts'] + offensive_df['sacks_taken'])
```

Figura 3.5: Creación de la métrica avanzadas Net Yards per Pass Attempt

Además de las métricas calculadas en el script de extracción, en la propia aplicación web se realiza la creación de variables adicionales para evaluar de forma más completa a los quarterbacks modernos, que suponen una doble amenaza (pase y carrera). Para ello, se crean métricas mixtas que combinan su producción en ambas facetas del juego, como los touchdowns totales o los primeros downs totales.

```
player_df['total_tds'] = player_df['passing_tds'] + player_df['rushing_tds']
player_df['total_first_downs'] = player_df['passing_first_downs'] + player_df['rushing_first_downs']
player_df['total_turnovers'] = player_df['interceptions'] + player_df['rushing_fumbles_lost'] + player_df['sack_fumbles_lost']
```

Figura 3.5: Creación de métricas mixtas para QBs en la aplicación de Streamlit.

Capítulo 4 - Modelo analítico

Este capítulo forma la parte más valiosa del proyecto, en la que se aplican técnicas de **Machine Learning no supervisado** para trascender el análisis descriptivo tradicional, que constituye la mayor parte del proyecto, permitiendo descubrir patrones y estructuras latentes en los datos de los jugadores. A diferencia del aprendizaje supervisado, que busca predecir una etiqueta conocida, el aprendizaje no supervisado nos permite explorar los datos sin una meta predefinida, encontrando agrupaciones y relaciones que no son evidentes a simple vista.

En el contexto deportivo, estas técnicas son de inmenso valor para tareas como la **segmentación de jugadores** en perfiles o arquetipos según su estilo de juego, y la **búsqueda de perfiles similares** (comparables), tareas fundamentales en el scouting, formación y la construcción de plantillas.

Para este proyecto, se han implementado dos de los algoritmos más robustos y reconocidos en este campo: el **Clustering K-Means** para la identificación de arquetipos y el **Análisis de Componentes Principales (PCA)** como base para un sistema de búsqueda de similitudes.

4.1 Metodología y preparación de los datos

Antes de aplicar los modelos, es necesario realizar un preprocesamiento de los datos para asegurar que los algoritmos funcionen de manera óptima y que los resultados sean significativos. Dentro de la propia aplicación, este proceso se ha realizado en la última página "Modelado Analítico" mediante de los siguientes pasos:

1. **Selección de características (Features):** no todas las estadísticas son igualmente relevantes para definir el estilo de un jugador. Se ha realizado una selección manual de las métricas más descriptivas para cada una de las tres posiciones analizadas (QB, RB y Receptor), combinando estadísticas de volumen, eficiencia y métricas avanzadas como el EPA.

2. **Filtrado por participación mínima:** para evitar que jugadores con un volumen de juego muy bajo distorsionen el análisis, se ha implementado un filtro dinámico. El usuario puede establecer un umbral mínimo de participación (intentos de pase para QBs, acarreos para RBs y targets para Receptores) para asegurar que solo se incluyan en el modelo jugadores con una muestra de datos estadísticamente representativa.

3. **Escalado de datos:** las características seleccionadas tienen escalas muy diferentes (ej. las yardas de pase se miden en miles, mientras que el EPA se mueve en un rango mucho más pequeño). Si no se normalizan, las variables con magnitudes más grandes dominarían el modelo. Para solucionar esto, se aplica un **Escalado Estándar** (StandardScaler de la librería Scikit-learn), que transforma cada característica para que tenga una media de 0 y una desviación estándar de 1, asegurando que todas contribuyan de forma equitativa al análisis. El escalado estándar es uno de los métodos más utilizados y útiles y supone que los datos siguen una distribución normal. Consiste en restar a cada observación la media de la variable μ y dividir por la desviación típica σ de tal modo que las nuevas variables tendrán $\mu = 0$ y $\sigma = 1$. Para i variables (columnas) y j individuos (filas) las fórmulas serían:

$$\mu_i := \frac{1}{n}\sum_{j=1}^{n} x_i^{(j)} \qquad \sigma_i := \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(x_i^{(j)} - \mu_i\right)^2}$$

4.2 Modelo de clustering (K-Means) para identificación de arquetipos

El objetivo de este modelo es agrupar a los jugadores de una misma posición en un número predefinido de clústeres o "arquetipos", donde cada clúster representa un estilo de juego distinto. Para realizar esta tarea, al igual que a la hora de comparar jugadores o mostrar sus estadísticas, nos hemos centrado en las posiciones de Quarterback, Running Back y Receptor (incluyendo Wide Receiver y Tight End), ya que por construcción de los datasets, estas son las posiciones para las que contamos con más estadísticas y variables relevantes.

4.2.1 Algoritmo K-Means

Se ha utilizado el algoritmo **K-Means**, uno de los métodos de clustering más populares por su simplicidad y eficacia, permitiendo agrupar las observaciones en k clusters, siendo k un número predeterminado. La idea general del algoritmo es asignar observación al cluster con el centro, denominado centroide, más cercano. Este centroide es la media de todos los puntos pertenecientes al cluster, es decir, sus coordenadas se corresponden con la media aritmética para cada variable por separado sobre todos los puntos del cluster. El desarrollo general del algoritmo sería el siguiente:

1. Elegir el número k de clusters.

2. Seleccionar aleatoriamente k observaciones del conjunto de datos como los centroides iniciales. Algunas variaciones del algoritmo seleccionan puntos aleatorios del espacio de las variables como centroides, sin necesidad de que estos sean observaciones.

3. Asignar cada observación al cluster cuyo centroide sea el más cercano.

4. Calcular los nuevos centroides para cada cluster como la media de todas las observaciones pertenecientes a cada cluster.

5. Repetir los pasos 3 y 4 hasta la convergencia, es decir, hasta que los centroides no cambien de una iteración a otra o se cumpla algún criterio de tolerancia.

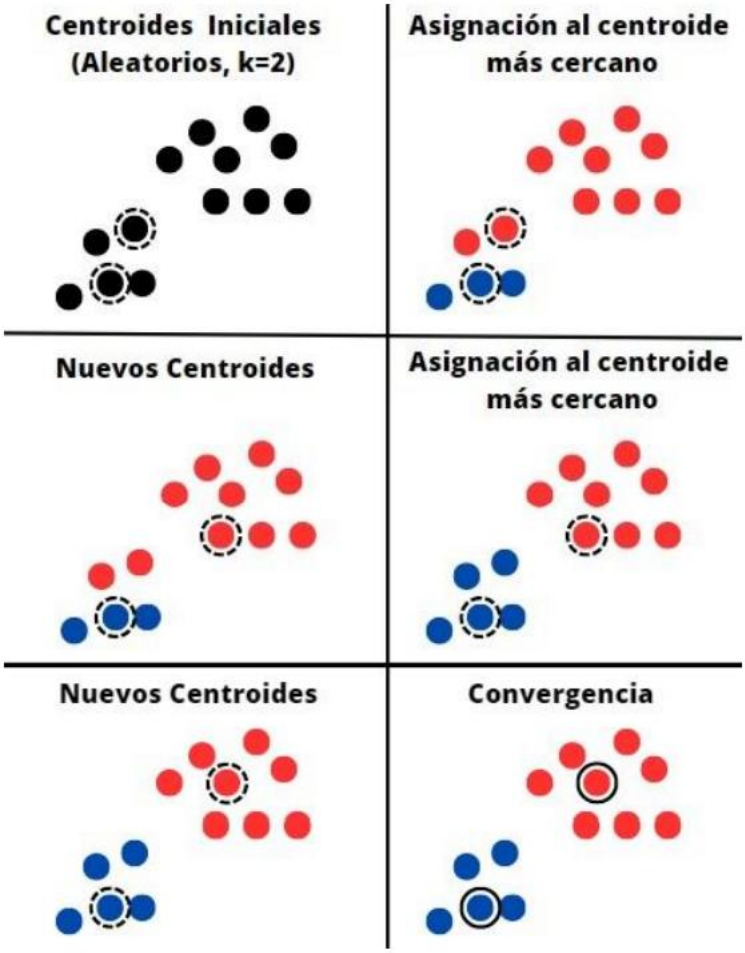


Figura 4.1: K-Means para k=2. Fuente: elaboración propia en Canva

4.2.2 Determinación del número óptimo de clústeres (k)

La pregunta más relevante a la hora de realizar el K-Means es: ¿cuántos grupos o clusters existen realmente? Para responder a esta pregunta de forma objetiva, la aplicación calcula y visualiza dos métricas para un rango de valores de *k* (de 2 a 8):

- **Método del Codo (Inertia SSE):** la inercia es la suma de las distancias al cuadrado de cada jugador a su centroide más cercano. Un valor de inercia bajo significa que los clústeres son densos y compactos. Elegida una medida de distancia (d) y siendo k el número de clusters, su fórmula es la siguiente:

$$SSE = \sum_{i=1}^{k} \sum_{x \in C_i} d^2(m_i, x)$$

  Donde x es un unto perteneciente al cluster Ci cuyo punto central es mi con $i = 1, \cdots, k$. El método del codo representa gráficamente el número de clusters k en el eje X y la inercia en el eje Y de tal forma que el "codo" de la gráfica, es decir, el punto en el que aunque aumentemos k, la inercia disminuye poco, será un buen indicador del número de clusters.

- **Coeficiente de Silueta (Silhouette Score):** esta métrica mide cómo o cuanto de similar es un jugador a los de su propio clúster en comparación con los de otros clústeres. El valor varía de -1 a 1, donde un valor alto indica que los clústeres están bien definidos y separados entre sí. En este proyecto, se utiliza el valor de *k* que maximiza el Coeficiente de Silueta como la recomendación principal para el usuario. La fórmula de esta métrica, para un punto x es la siguiente:

$$s(x) = \frac{d_{ext}(x) - d_{int}(x)}{max(d_{int}, d_{ext})}$$

  donde $dint(x)$ sería la distancia media entre el punto x y el resto de los puntos de su cluster, mientras que $dext(x)$ es la distancia media entre el punto x y los puntos del cluster más cercano al que no pertenezca.

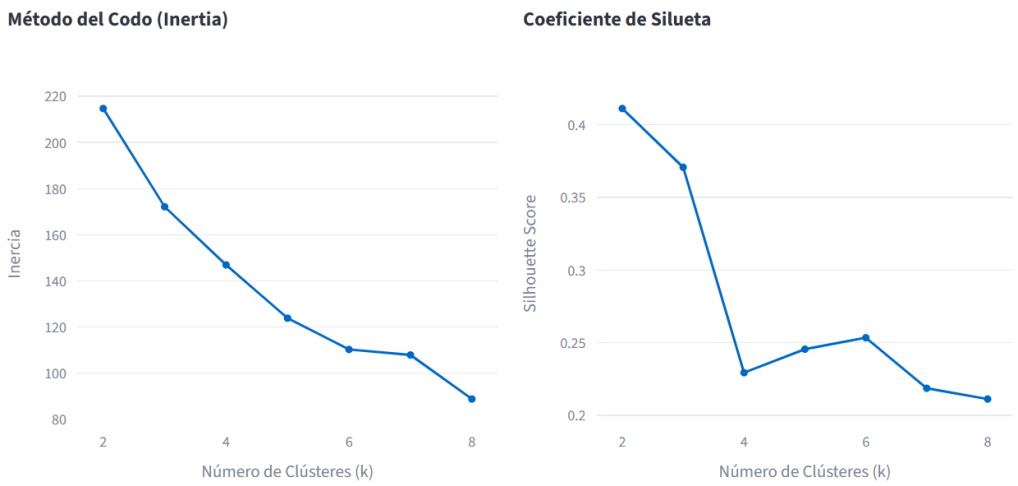


Figura 4.2: Ejemplo de gráfico del codo (Inercia) y de Silhouette score en función del número de clusters. Atendiendo al ejemplo y ambas métricas, en este caso 3 podría ser un número adecuado de clusters. Fuente: captura de la aplicación de Streamlit.

4.2.3 Interpretación y caracterización de los clusters

Una vez que los jugadores han sido asignados a un cluster, el paso final es interpretar qué significa cada uno. Para ello, la aplicación genera una tabla de caracterización que muestra el valor medio de cada una de las características para los jugadores pertenecientes a cada grupo. Analizando esta tabla, se puede asignar una etiqueta descriptiva a cada arquetipo.

Por ejemplo, un clúster de receptores con un target_share alto pero un air_yards_share bajo podría definirse como "Receptores de Posesión". En la siguiente imagen, podemos ver cómo podemos ver cómo por ejemplo para los receptores de la última temporada hace 3 grupos: grupo 2, malos receptores con peores estadísticas; grupo 1, buenos receptores con mejores estadísticas; y grupo 0, donde están los jugadores con un RACR (Receiver Air Conversion Ratio = Receiving Yards / Air Yards) muy alto, es decir, aquellos que ganan más yardas tras recibir el pase y también tienen más fumbles o pérdidas. Este último grupo por tanto se corresponde principalmente con los Tight Ends.

	0	1	2
receiving_epa	22.49	47.30	15.90
receiving_tds	4.11	7.65	4.08
racr	31.41	16.28	13.10
receiving_yards_after_catch	354.00	396.55	199.44
receiving_first_downs	32.61	53.39	30.33
receiving_fumbles	1.56	0.39	0.25
receiving_yards	698.78	1101.19	610.65

Tabla 4.1: Características medias de los 3 clusters de receptores en 2024. Fuente: captura propia de la aplicación de Streamlit.

4.3 Modelo de reducción de dimensionalidad (PCA) para la búsqueda de jugadores similares

El segundo modelo utilizado tiene como objetivo construir un sistema de similitud de jugadores, de tal forma que dado un jugador en las citadas posiciones ofensivas (QB, RB o receptor), pueda encontrar a los 10 jugadores con el perfil estadístico más similar en la liga para una temporada determinada en función de las estadísticas que se han considerado más relevantes para cada posición.

4.3.1 Análisis de Componentes Principales (PCA)

Los jugadores son definidos por un gran número de variables estadísticas distintas, (dimensiones). Trabajar directamente en este espacio multidimensional resulta complejo y muy difícil de visualizar. El **Análisis de Componentes Principales (PCA)** es una técnica de reducción de dimensionalidad que nos permite proyectar estos datos en un nuevo espacio de menos dimensiones (en este caso hemos seleccionado 2) con la mínima pérdida de información posible.

El PCA crea nuevos ejes, llamados **Componentes Principales**, que son combinaciones lineales de las estadísticas originales. El primer componente principal (PC1) captura la mayor varianza posible en los datos, y el segundo (PC2) captura la mayor parte de la varianza restante (entendiendo máxima varianza como máxima información retenida). El proceso sería análogo e iterativo si pretendiésemos obtener más componentes principales.

Estas dos componentes nos permiten visualizar a todos los jugadores en un gráfico de dispersión 2D, donde la posición de cada jugador resume su perfil estadístico global. En este mismo gráfico podremos ver los clusters formados en el anterior paso, donde cada color representará un cluster distinto.

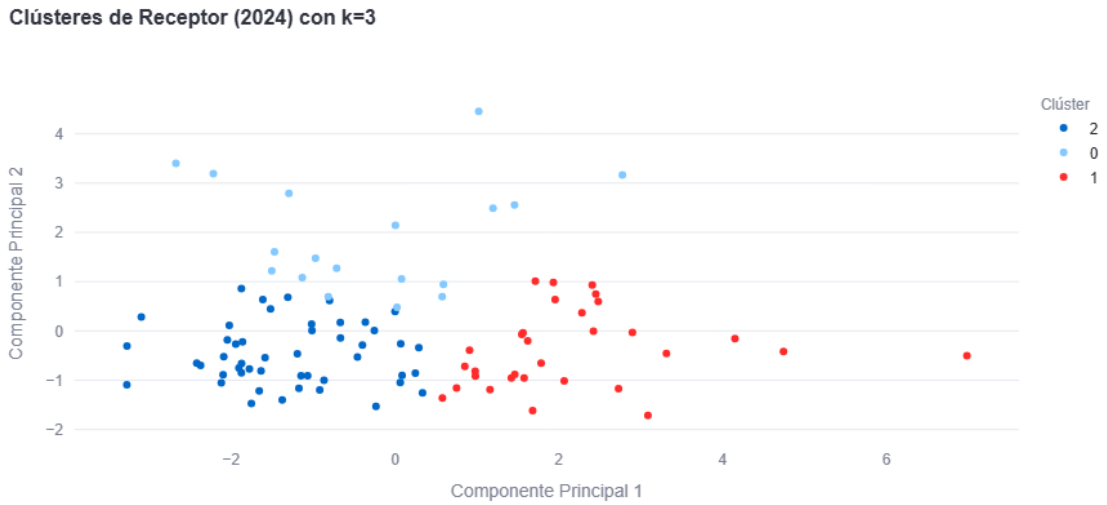


Figura 4.3: Representación de los 3 clusters de receptores en un espacio de 2 dimensiones formado por las 2 primeras componentes principales. Fuente: captura propia de la aplicación de Streamlit.

4.3.2 Cálculo de similitud

Una vez que cada jugador está representado por sus coordenadas en el espacio de los componentes principales, encontrar perfiles similares se convierte en un problema geométrico. Para ello, se utiliza la **distancia euclidia**, que se define cómo la distancia más corta entre dos puntos en línea recta. a. Para dos puntos A y B, con coordenadas Xi e Yi respectivamente donde i = 1, … , n indica la dimensión, la distancia es:

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(X_i - Y_i)^2}$$

Cuando un usuario selecciona un determinado jugador (después de haber filtrado por posición y participación), el sistema calcula la distancia euclidia desde ese jugador a todos los demás. Los jugadores con las distancias más bajas son los que tienen los perfiles estadísticos más parecidos.

Para que el resultado sea más fácil de interpretar, esta distancia se convierte en un **"Índice de Similitud"** (de 0 a 100) mediante la fórmula

$$Índice\ Similitud = \frac{1}{1 + d(A, B)} * 100$$

donde una puntuación más alta indica una mayor similitud. Los resultados se presentan al usuario en una tabla ordenada, mostrando los 10 jugadores más similares al seleccionado.

Top 10 Jugadores más similares a A.J. Brown:

	player_name	team	similarity_score	receiving_epa	receiving_tds	racr	receiving_yards_after_catch	receiving_first_
49	Nico Collins	HOU	51.5%	42.83	7.00	11.82	365.00	
43	Darnell Mooney	ATL	44.4%	43.91	5.00	14.25	275.00	
54	DeVonta Smith	PHI	44.1%	57.75	8.00	10.94	299.00	
85	Zay Flowers	BAL	39.3%	45.60	4.00	17.66	463.00	
13	George Kittle	SF	38.9%	67.94	8.00	22.37	522.00	
46	Jerry Jeudy	CLE	37.6%	37.64	4.00	12.37	387.00	
89	Malik Nabers	NYG	37.0%	45.59	7.00	12.97	462.00	
11	Tyreek Hill	MIA	37.0%	28.03	6.00	12.51	292.00	
60	Jameson Williams	DET	36.6%	37.62	7.00	17.58	497.00	
29	Jakobi Meyers	LV	36.2%	31.35	4.00	13.33	293.00	

Tabla 4.2: Tabla de similitud de los 10 receptores más similares a A.J. Brown en 2024. Fuente: captura propia de la aplicación de Streamlit.

Capítulo 5 - Visualización

Una de las fases finales de cualquier proyecto de análisis de datos es la comunicación clara y efectiva de los resultados obtenidos. Los datos, por muy limpios y bien modelados que estén, solo adquieren valor real cuando se presentan de una forma que sea comprensible, intuitiva, atractiva y que permita al usuario final extraer conclusiones significativas. En este proyecto, la visualización es el núcleo central del trabajo, ya que la idea original es, a través de una aplicación web interactiva, presentar los datos crudos y tabulares extraídos de una Api en gráficos atractivos y fácilmente interpretables sobre los que poder extraer conclusiones de valor.

Este capítulo se divide en dos secciones. En la primera, se justifica la elección de la herramienta tecnológica utilizada para construir la aplicación, la librería de Python **Streamlit**. En la segunda, se realiza un recorrido visual y funcional a través de los distintos cuadros de mando y dashboards que componen la web "NFL Analytics Hub", explicando el propósito y el diseño de cada uno.

5.1 Herramienta de visualización: Streamlit

Para el desarrollo de la aplicación web se ha seleccionado **Streamlit**, un framework de código abierto (OpenSource) en Python diseñado específicamente para que los científicos de datos y analistas puedan construir y compartir aplicaciones de datos interactivas con un esfuerzo mínimo en el desarrollo de la interfaz de usuario (frontend). Es decir, el usuario se puede centrar en elegir qué contenido mostrar, ya que el diseño de la web correrá principalmente por parte de Streamlit facilitando así su implementación.

La elección de Streamlit frente a otras posibles herramientas, softwares, librerías y/o lenguajes (HTML, JavaScript, Wordpress, Django, Flask…) se basa en una serie de características relacionadas con el proyecto:

1. **Fácil de integrar en Python y con librerías de de Data Science:** Streamlit opera íntegramente en Python. Esto elimina la necesidad de aprender otros lenguajes de desarrollo web (como JavaScript) y permite una integración directa las librerías fundamentales del análisis de datos utilizadas en este proyecto, como **Pandas** para la manipulación de datos, **Plotly** para la creación de gráficos interactivos y **Scikit-learn** para la implementación de los modelos de Machine Learning.

2. **Rapidez de desarrollo:** el paradigma de Streamlit es "scripting". La aplicación se construye como un script de Python simple y legible. Cada vez que se guarda el código, la aplicación se actualiza en tiempo real, permitiendo un ciclo de desarrollo de la web muy rápido y facilitando la depuración de errores. Esto resulta fundamental para añadir nuevas páginas y funciones de forma ágil.

3. **Componentes interactivos integrados:** Streamlit ofrece una amplia gama de widgets interactivos (selectores, sliders, menús desplegables, etc.) que se pueden implementar con una sola línea de código, importando el widget correspondiente. Esta característica es fundamental para proporcionar a la aplicación la forma interactiva buscada, permitiendo al usuario final filtrar los datos y personalizar las visualizaciones según sus intereses.

4. **Interfaz de usuario profesional por defecto:** sin necesidad de conocimientos avanzados en diseño web o CSS, Streamlit genera aplicaciones con una apariencia limpia, bonita y profesional. Su sistema de layout (columnas, expanders, pestañas) ha permitido organizar una gran cantidad de información de forma estructurada y sin sobrecargar al usuario.

En resumen, Streamlit ha permitido centrar los esfuerzos en lo verdaderamente importante (el análisis de datos y la calidad de las visualizaciones), eliminando gran parte de la complejidad asociada al desarrollo de una web tradicional.

5.2 Recorrido visual por la aplicación "NFL Analytics Hub"

La aplicación se ha estructurado en un formato multipágina, utilizando la barra de navegación lateral de Streamlit para facilitar el acceso a las diferentes herramientas de análisis. A continuación, se detalla la funcionalidad de cada una de estas páginas.

5.2.1 Página principal o Inicio

La página de inicio sirve como portal de bienvenida y punto de partida para el usuario. Su diseño busca ser claro y conciso, presentando la misión del proyecto y guiando al usuario a través de las distintas secciones disponibles. Se incluyen, además, KPIs agregados de todo el dataset (temporadas analizadas, touchdowns totales, etc.) para ofrecer una primera impresión del volumen de datos manejado.

Figura 5.1: Captura de la página de Inicio donde podemos observar el índice lateral con el resto de páginas o secciones y una breve descripción de cada una.

5.2.2 Análisis de Equipos

Esta página ofrece una visión panorámica del rendimiento de todos los equipos de la liga. La interfaz se ha diseñado mediante menús desplegables para cada categoría de estadísticas (Totales, Pase y Carrera), evitando así la sobrecarga de información. Dentro de cada menú, dos columnas enfrentan el rendimiento ofensivo y defensivo. El usuario puede utilizar los filtros de la barra lateral (temporada, conferencia y división) y los selectores de métricas para generar gráficos de barras horizontales que clasifican a los equipos. Estos gráficos utilizan una escala de color divergente (rojo-amarillo-verde) para facilitar la identificación inmediata de los equipos con mejor y peor rendimiento en la métrica seleccionada. Además, se tiene en cuenta para qué estadísticas un valor menor es mejor (intercepciones, sacks, fumbles…). Por último, también se incluye una sección desplegable comparando el juego de pase y de carrera.

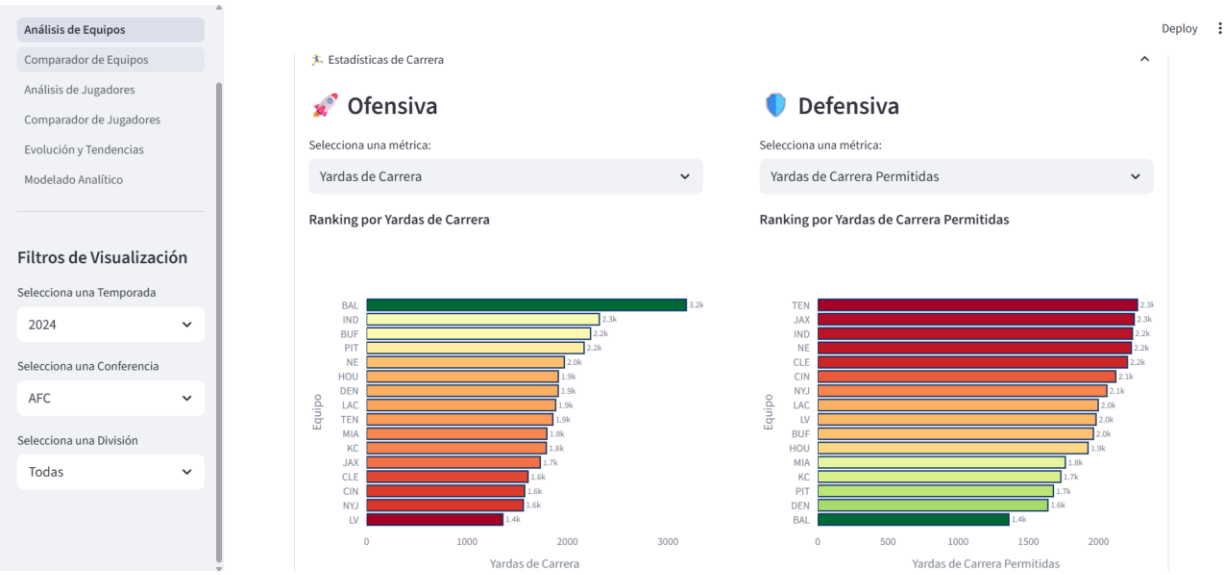


Figura 5.2: Análisis de Equipos. Nos hemos centrado en el juego de carrera de los equipos de la AFC durante la temporada pasada (2024). Podemos observar como en las dos métricas seleccionadas el juego de carrera de Baltimore destaca tanto en ofensiva como en defensiva.

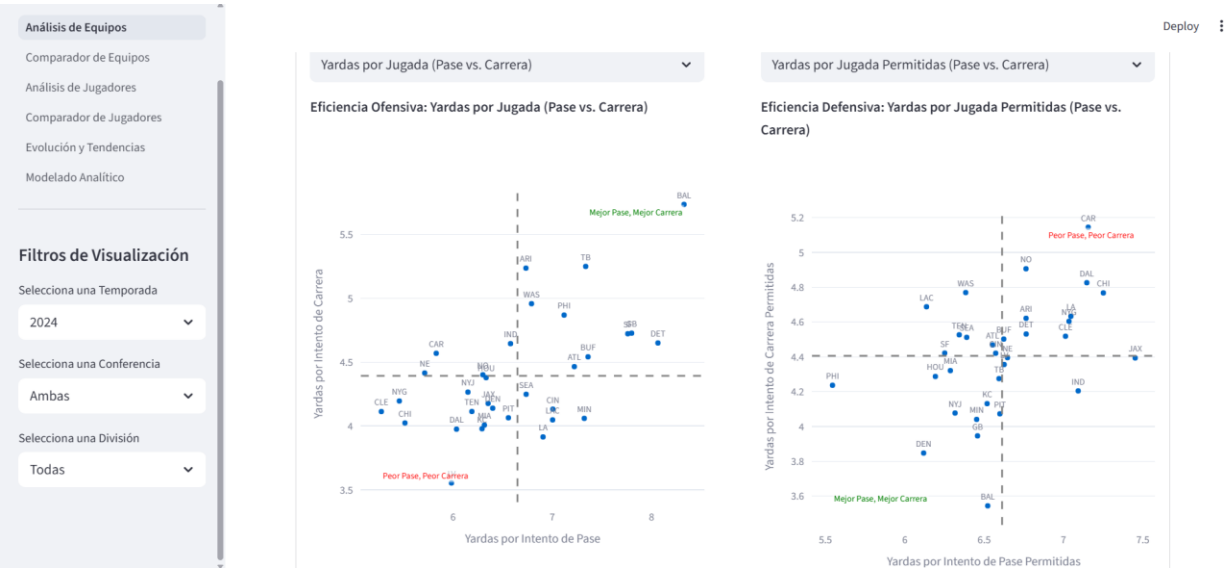


Figura 5.3: Análisis de Equipos. Comparativa para todos los equipos del juego de carrera frente al de pase durante la temporada pasada.

5.2.3 Comparador de Equipos

Esta página permite un análisis "Head-to-Head" entre dos equipos para una temporada específica. La visualización principal es un **gráfico de radar**, una elección deliberada por su capacidad para mostrar fortalezas y debilidades relativas en múltiples variables de forma simultánea. Para asegurar una comparación justa, el gráfico no utiliza los valores brutos, sino el **percentil de rendimiento** de cada equipo en la liga para cada métrica. De esta forma, un valor más alto (más cerca del borde exterior) siempre indica un mejor rendimiento. La página se complementa con los logos de los equipos y tablas comparativas que muestran los datos brutos para dar contexto a los percentiles. Se seleccionaron sólo las 6 métricas (3 ofensivas y 3 defensivas) que se consideraron más representativas para dar un resultado homogéneo y fácil de interpretar.

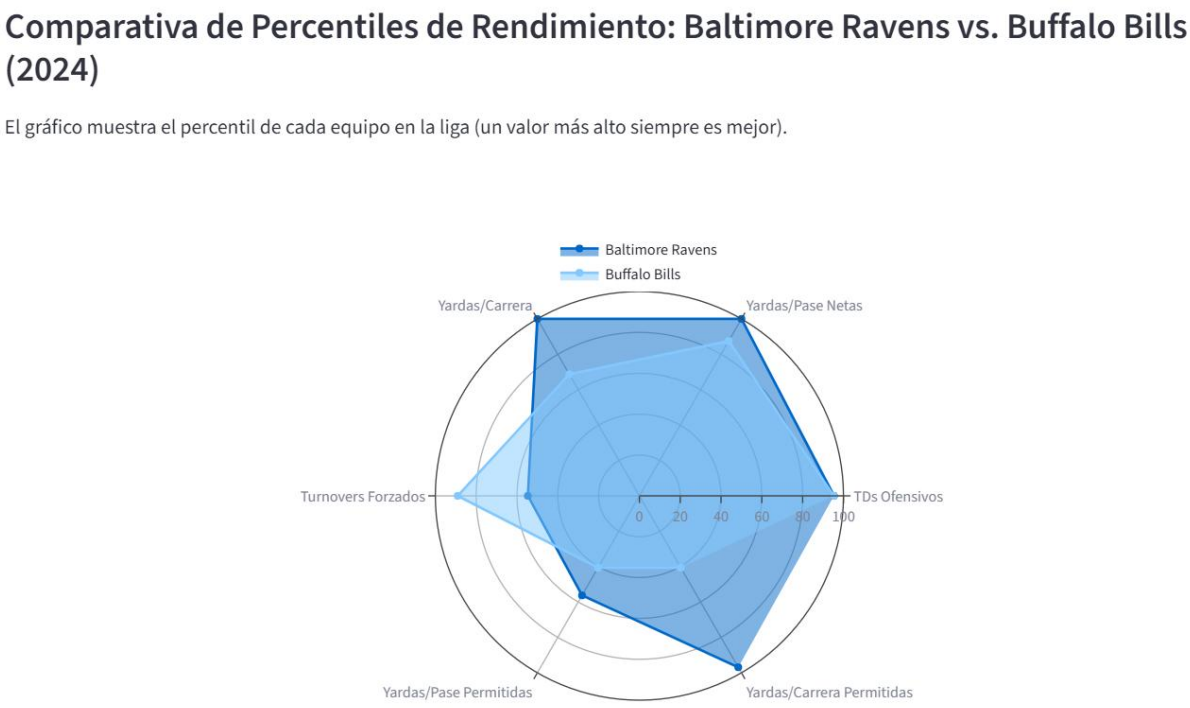
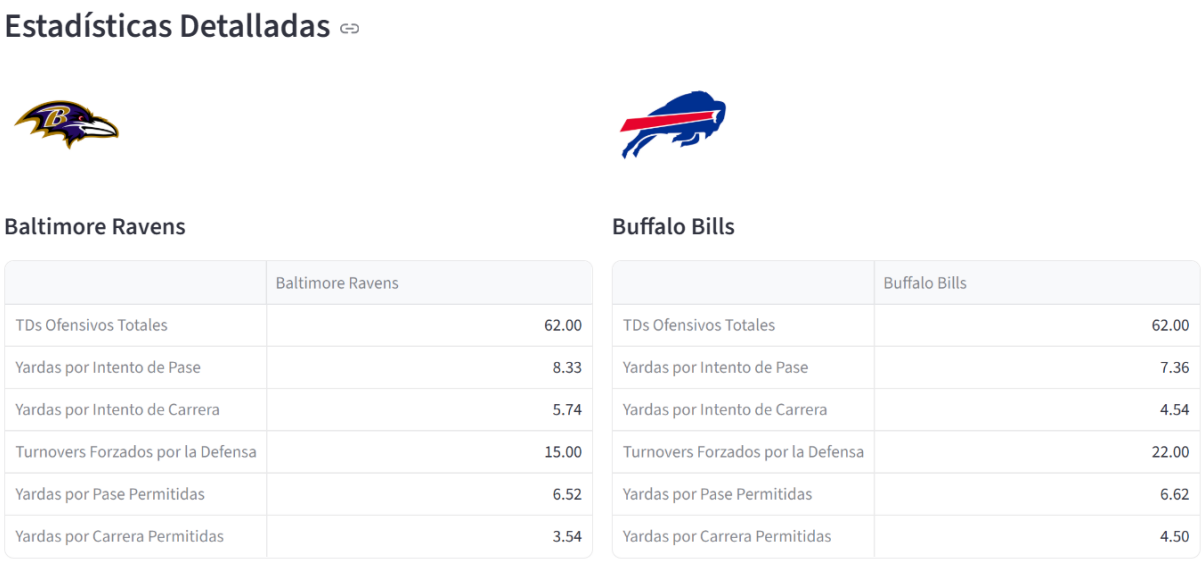


Figura 5.4: Comparador de Equipos, gráfico de radar.



	Baltimore Ravens
TDs Ofensivos Totales	62.00
Yardas por Intento de Pase	8.33
Yardas por Intento de Carrera	5.74
Turnovers Forzados por la Defensa	15.00
Yardas por Pase Permitidas	6.52
Yardas por Carrera Permitidas	3.54

	Buffalo Bills
TDs Ofensivos Totales	62.00
Yardas por Intento de Pase	7.36
Yardas por Intento de Carrera	4.54
Turnovers Forzados por la Defensa	22.00
Yardas por Pase Permitidas	6.62
Yardas por Carrera Permitidas	4.50

Tabla 5.1: Comparador de Equipos, tabla con los datos absolutos. Comparando dos de los mejores equipos de la temporada pasada, vemos que Buffalo sólo supera a Baltimore en menos turnovers.

5.2.4 Análisis de Jugadores

Esta página se centra en el rendimiento individual. Permite al usuario filtrar por posición (QB, RB o Receptor, ya que son las posiciones para las que contamos con más datos), temporada, conferencia y división, así como por participación mínima para asegurar la relevancia estadística. La página se adapta dinámicamente a la posición seleccionada, ofreciendo las métricas correspondientes para cada posición. La visualización principal es un gráfico de barras del Top 20 de jugadores para la métrica elegida, utilizando también una escala de color global (basada en todos los jugadores filtrados, no solo el Top 20) para contextualizar el rendimiento.

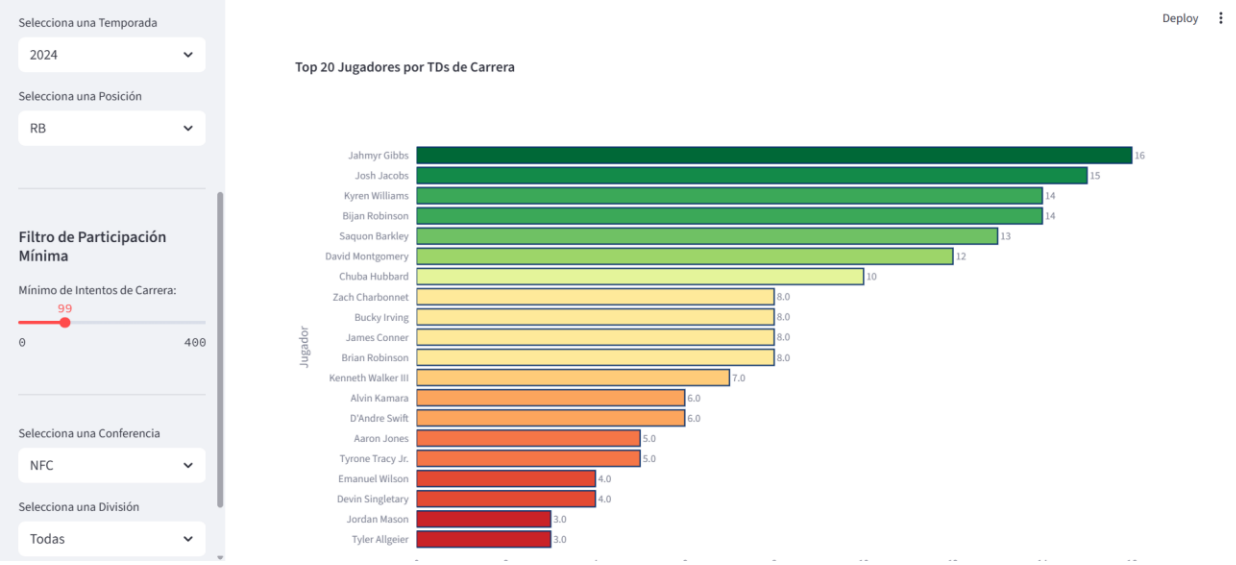


Figura 5.5: Análisis de Jugadores. Análisis de los TDs de carrera para Running Backs con al menos 99 intentos de carrera de la NFC durante la temporada pasada (2024).

5.2.5 Comparador de Jugadores Ofensivos

De manera análoga al Comparador de Equipos, en esta página una vez elegida la posición (QB, RB o Receptor) y la temporada, se puede comparar cara a cara a dos jugadores mediante un atractivo y fácil de interpretar gráfico de radar. Para cada posición se han elegido las métricas que se han considerado más relevantes, pero la comparativa se vuelve a realizar sobre los percentiles. Además, también se incluye el filtro de participación mínima y la tabla de métricas absolutas.

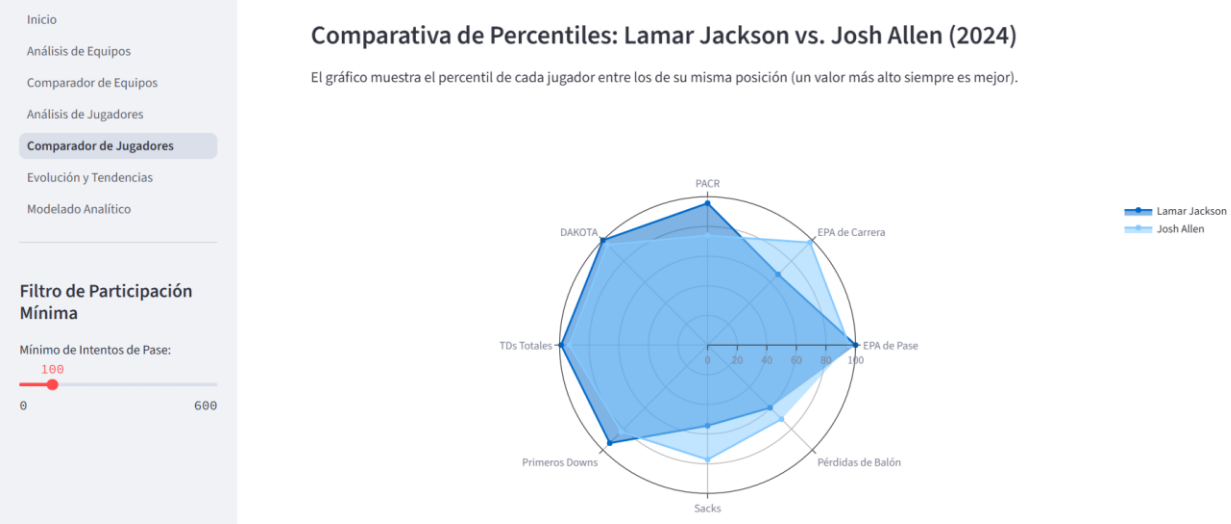


Figura 5.6: Comparador de Jugadores Ofensivos. Comparación entre los 2 principales candidatos a MVP de la temporada pasada (QB, 2024)

5.2.6 Evolución y Tendencias

Esta página está dedicada al análisis de series temporales. El usuario puede seleccionar si desea analizar equipos o jugadores, y posteriormente elegir dos entidades para comparar su evolución a lo largo de las cinco temporadas (2020-2024) en una métrica específica. La visualización principal es un **gráfico de líneas**, la herramienta estándar y más efectiva para mostrar tendencias, picos de rendimiento y declives a lo largo del tiempo.

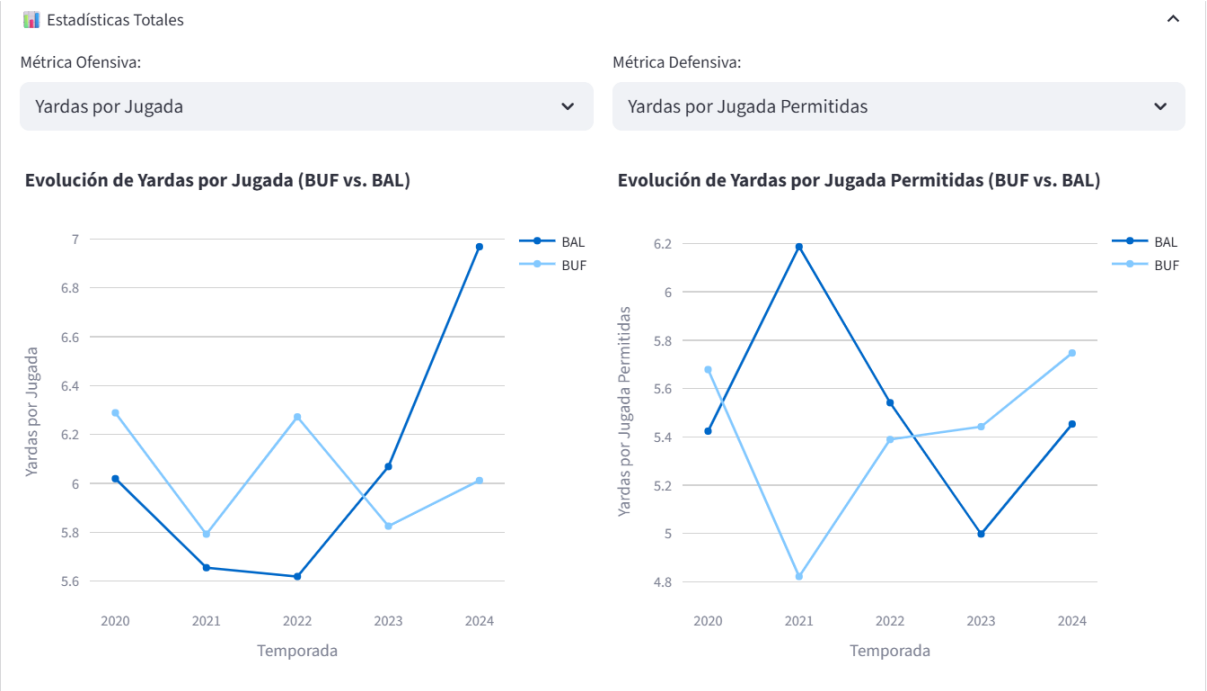


Figura 5.7: Evolución y tendencias. Evolución de Equipos durante las últimas 5 temporadas entre los citados equipos de Baltimore y Buffalo

5.2.7 Modelado Analítico

Esta es la sección más avanzada de la aplicación y presenta los resultados de los modelos de Machine Learning comentados en el anterior capítulo en dos pestañas:

- **Clustering de jugadores:** esta pestaña permite al usuario explorar los arquetipos de jugadores. Incluye un análisis para determinar el número óptimo de clústeres y un gráfico de dispersión interactivo que visualiza a los jugadores en el espacio de las dos primeros componentes principales, coloreados por el clúster al que pertenecen. Se complementa con una tabla de caracterización que define cada arquetipo mostrando el valor medio de sus estadísticas.

- **Buscador de jugadores similares:** esta herramienta permite al usuario seleccionar un jugador y obtener una lista de los 10 jugadores con el perfil estadístico más parecido, basado en la distancia euclidia en el espacio de componentes principales.

Capítulo 6 - Conclusiones

Este último capítulo tiene como objetivo resumir los resultados del proyecto, evaluar el grado de cumplimiento de los objetivos propuestos y reflexionar sobre los hallazgos, las limitaciones y las posibles vías de continuación del trabajo realizado.

6.1 Resumen de la solución al problema planteado

El problema inicial que motivó este proyecto fue la gran diferencia que existe entre la masiva generación de datos en la NFL y la dificultad para que analistas, entrenadores y aficionados puedan acceder a ellos y extraer conocimiento de forma fácil, interactiva y comprensible. La web *"NFL Analytics Hub"* se ha diseñado y construido como una solución directa a este desafío.

El proyecto ha logrado solucionar el problema inicial a través de los siguientes puntos clave:

1. **Centralización y democratización del acceso:** la aplicación unifica la información al integrar en una única plataforma opensource datos ofensivos, defensivos y de jugadores para cinco temporadas completas. El usuario final no necesita ningún conocimiento de programación para acceder a los datos.

2. **Simplificación y traducción de los datos:** se han transformado tablas de datos con cientos de columnas en visualizaciones interactivas. Los gráficos de barras con escalas de color, los gráficos de dispersión con cuadrantes de eficiencia y los radares de percentiles traducen métricas complejas en *insights* visuales e inmediatos.

3. **Interactividad y personalización:** a diferencia de los informes estáticos, la aplicación dota al usuario de control total sobre el análisis. Mediante el uso de filtros dinámicos, el usuario puede personalizar su consulta por temporada, posición, equipo o umbrales de participación, adaptando la herramienta a sus necesidades específicas de análisis.

4. **Incorporación de análisis avanzado:** el proyecto no se ha limitado al análisis descriptivo mediante la visualización de estadísticas, sino que ha integrado modelos de Machine Learning no supervisado. Esto permite al usuario adentrarse aún más, pudiendo descubrir arquetipos de jugadores y encontrar perfiles estadísticamente similares, funcionalidades que no suelen estar disponibles en herramientas de acceso público.

En resumen, el *"NFL Analytics Hub"* cumple con su objetivo de actuar como un puente entre el Big Data y la toma de decisiones, convirtiendo datos crudos en una herramienta estratégica para la comprensión del rendimiento en la NFL.

6.2 Principales hallazgos e "insights" obtenidos

A lo largo del desarrollo y uso de la aplicación, se han podido extraer diversos hallazgos que demuestran el valor de la herramienta:

- **Identificación de perfiles de equipo y evolución de estos:** los gráficos de dispersión que comparan el juego de pase y de carrera han demostrado ser muy útiles para clasificar rápidamente el "ADN" de las ofensivas y defensivas. Permiten identificar visualmente a los equipos balanceados, a las potencias unidimensionales y a las unidades con debilidades claras en una de las dos facetas. Además, una vez analizados cuales son los puntos fuertes y débiles de los equipos, podemos observar cómo ha sido su evolución en estos campos en las últimas temporadas y analizar si ha existido un punto de inflexión en su juego.

- **Descubrimiento de arquetipos de Jugadores:** la aplicación del clustering K-Means ha validado la existencia de distintos perfiles de jugadores dentro de una misma posición. Por ejemplo, para los receptores, el modelo ha sido capaz de diferenciar entre grandes receptores, receptores de perfil más bajo y receptores den corto con un mayor porcentaje de yardas ganadas tras la recepción (principalmente Tight Ends). Aún así, esta herramienta se ve bastante caracterizada por el volumen.

- **Contextualización del rendimiento:** el uso de percentiles en los gráficos de radar ha resultado ser una metodología muy eficaz para comparar jugadores o equipos de forma justa. Un jugador puede acumular muchas yardas, pero si su percentil de eficiencia (como el EPA) es bajo, la herramienta permite identificarlo como un jugador de alto volumen, pero baja eficiencia, un matiz crucial para la evaluación de talento.

6.3 Limitaciones y futuras líneas de trabajo

A pesar de haber cumplido los objetivos propuestos, es importante reconocer las limitaciones del proyecto y las posibles vías para su expansión y mejora en el futuro.

Limitaciones:

- **Alcance de los datos:** el análisis se basa exclusivamente en datos estadísticos de jugadas. No se incluyen datos contextuales de gran importancia como información de contratos, lesiones, calificaciones de ojeadores (scouting) o datos de posicionamiento de jugadores (Next Gen Stats), cuyo acceso es restringido. Además, respecto a las estadísticas de jugadores, se ve muy limitado al ser sólo las 3 posiciones analizadas las que cuentan con suficientes estadísticas para realizar un estudio significativo.

- **Naturaleza descriptiva:** el proyecto se centra en el análisis descriptivo y no en el predictivo. La herramienta explica lo que ha ocurrido, pero no intenta predecir resultados de partidos o el rendimiento futuro de los jugadores.

Futuras líneas de trabajo:

- **Integración de nuevas fuentes de datos:** una nueva versión del proyecto podría buscar integrar datos públicos sobre contratos de jugadores para realizar análisis de "valor por dinero", comparando el rendimiento estadístico con el coste salarial.

- **Desarrollo de modelos predictivos:** la base de datos creada es un buen punto de partida para construir modelos de Machine Learning supervisado que puedan predecir los resultados de los partidos o la progresión estadística de los jugadores.

- **Análisis de equipos especiales:** El proyecto se ha centrado en la ofensiva y la defensiva. Una ampliación natural sería incluir un análisis detallado del rendimiento de los equipos especiales, una faceta crucial del juego.

Bibliografía

Baldwin, B., & Carl, S. (2020). *nflfastR*. R package. https://www.nflfastr.com/

Burke, B. (2019). DeepQB: A deep learning approach to passing analytics in the NFL. *Journal of Sports Analytics, 5*(2), 99-118.

Gandomi, A., & Haider, M. (2015). Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management, 35*(2), 137-144.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651-666.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer Series in Statistics.

Laney, D. (2001). *3D Data Management: Controlling Data Volume, Velocity, and Variety*. META Group.

Lewis, M. (2003). *Moneyball: The Art of Winning an Unfair Game*. W. W. Norton & Company.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, 1*(14), 281-297.

Pearson, K. (1901). On Lines and Planes of Closest Fit to Systems of Points in Space. *Philosophical Magazine, 2*(11), 559-572.

Rein, R., & Memmert, D. (2016). Big data and tactical analysis in elite soccer: future challenges and opportunities for sports science. *SpringerPlus, 5*(1), 1410.

Sarlis, V., & Tjortjis, C. (2020). Sports analytics—A survey: On the journey to the golden era. *Computers, 9*(2), 46.

Streamlit Inc. (2023). *Streamlit: The fastest way to build and share data apps*. https://streamlit.io

The nflverse team. (2021). *nfl-data-py*. Python package. https://github.com/nflverse/nfl-data-py

Yurko, R., Horowitz, M., & Ventura, S. L. (2019). nflWAR: A Reproducible Method for Offensive Player Evaluation in Football. *Journal of Quantitative Analysis in Sports, 15*(3), 173-193.