Proyecto final Data Analytics AR Oct24

Presentación del proyecto

1. **Descripción del Caso de Negocio**

En el mundo del fútbol profesional la toma de decisiones basada en datos es crucial para el éxito. Los clubes de fútbol manejan grandes volúmenes de datos relacionados, tanto con el rendimiento individual de los futbolistas, como con el juego del equipo y de los rivales. Visualizar estos datos de manera efectiva y eficiente es un factor diferencial para mejorar la toma de decisiones en áreas clave como:

- **Rendimiento del equipo:** Optimización de tácticas y estrategias basadas en el análisis detallado del rendimiento en partidos.
- **Desarrollo de jugadores:** Evaluación precisa del rendimiento individual de los jugadores para identificar áreas de mejora y optimizar su desarrollo.
- **Análisis de oponentes:** Preparación de estrategias efectivas mediante el análisis profundo de los equipos rivales.
- **Fichajes:** Identificación de posibles fichajes que se ajusten a las necesidades del equipo mediante el análisis comparativo de jugadores.

En este caso se va a presentar un proyecto de una herramienta de análisis para un club de fútbol.

2. Objetivos del Proyecto y su Impacto Esperado

El objetivo principal de este proyecto es desarrollar un dashboard interactivo, alimentado por una base de datos, que permita, tanto a cuerpo técnico como directiva, explorar y analizar los datos de rendimiento del equipo de manera eficiente.

- **Objetivos específicos:**
  - **Centralizar los datos:** Crear una base de datos SQL robusta que integre datos de diversas fuentes, garantizando la integridad y disponibilidad de la información.
  - **Visualización interactiva:** Diseñar un dashboard en Power BI que ofrezca visualizaciones claras e interactivas de los datos.
  - **Análisis del rendimiento:** Facilitar el análisis detallado del rendimiento individual de los jugadores y del rendimiento colectivo del equipo.
  - **Soporte para la toma de decisiones:** Proporcionar información valiosa y accionable que respalde la toma de decisiones en áreas como la táctica, la gestión de jugadores y la estrategia de fichajes.
- **Impacto esperado:**
  - **Mejora en la toma de decisiones:** El acceso a datos claros y concisos permitirá a los entrenadores, analistas y directivos tomar mejores decisiones.

· **Optimización del rendimiento del equipo y jugadores:** El análisis detallado del rendimiento permitirá identificar áreas de mejora y optimizar tanto las tácticas y estrategias del equipo como el desarrollo de los jugadores.

· **Ventaja competitiva en fichajes:** El análisis comparativo de jugadores permitirá identificar posibles fichajes que se ajusten a las necesidades del equipo, obteniendo una ventaja en el mercado de fichajes.

3. Tecnologías y herramientas que se utilizarán.

- PowerBi: Para la creación del dashboard
- Python: Proceso ETL
- SQL: Creación base de datos

4. Identificación y justificación de las fuentes de datos.

- Datos extraídos de la temporada 2015/2016
- Datos extraídos de:Statsbomb (https://statsbomb.com/what-we-do/hub/free-data/)
- Datos abiertos que recogen estadísticas individuales y eventos de todos los partidos de la liga. Se eligen estos datos ya que son los únicos abiertos que contienen información detallada que permita un análisis profundo.
- API Transfermarkt (https://transfermarket.p.rapidapi.com/=)
- API Fbref (https://fbrapi.com/)
    - Esta API se va a usar para completar datos individuales que no se puedan extraer de Statsbomb.
- Wyscout (https://wyscout.hudl.com/app/)
    - Descarga de csv que pueden completar información a los datos iniciales.

5. Pasos a seguir

- Proceso de ETL
- Proceso de EDA
- Dashboard interactivo
- Conclusiones finales
- Next steps

ETL (Extracción, Transformación y Carga)

Esta fase se encarga de obtener los datos de sus fuentes originales, limpiarlos, estructurarlos y prepararlos para el análisis posterior.

1.1. Extracción (Extract)

Se obtienen datos de dos fuentes principales:

- **Datos de eventos de partidos (StatsBomb):**
    - Se utiliza la librería statsbombpy para acceder a la API de StatsBomb.
    - Se identifican las competiciones disponibles y se filtra específicamente por "La Liga" (temporada 2015/2016).
    - Se obtiene el match_id de todos los partidos de La Liga en esa temporada.
    - Se itera a través de los match_id para extraer los eventos detallados de cada partido (ej. pases, tiros, duelos, etc.).
    - Se almacenan los eventos en una lista de diccionarios que luego se convierte en un DataFrame de Pandas.
    - Se obtienen los datos de los equipos que participaron en cada partido,

- **Extracción de Datos de Jugadores desde Transfermarkt (RapidAPI)**

Este bloque de código se encarga de recopilar información de jugadores de fútbol de la plataforma Transfermarkt a través de API.

- **Extracción de IDs de Equipos de La Liga (2015):**
- Se realiza una solicitud GET a la API para obtener la tabla de clasificación de la liga española (La Liga, ID: ES1) de la temporada 2015.
- La respuesta JSON de la API se parsea para extraer los IDs de todos los equipos participantes en esa temporada y se almacenan en una lista.
- **Extracción de Plantillas de Equipos:**
- Se itera a través de cada `id` de equipo obtenido en el paso anterior.
- Para cada equipo, se realiza una nueva solicitud GET a la API para obtener la plantilla completa del equipo para la temporada 2015.
- Las respuestas JSON se almacenan en un diccionario, donde la clave es el ID del equipo.
- **Extracción de IDs de Jugadores:**
- Se recorre la estructura de datos anidada de las plantillas de los equipos para extraer todos los IDs de los jugadores de cada equipo y se añaden a una lista.
- **Extracción de Perfiles de Jugadores:**
- Se itera sobre cada id de jugador.

- Para cada jugador, se hace una solicitud GET a la API para obtener su perfil detallado.
- Los perfiles se almacenan en un diccionario.
- **Manejo de Errores/Reintentos:** Si algunas solicitudes de perfil de jugador fallan (indicado por un mensaje de error en la respuesta JSON), se identifican esos IDs (no_añadidos) y se reintentan las solicitudes. Si hay muchos es necesario generar una nueva API-Key ya que la API tiene un límite de peticiones en la versión gratuita.
- **Consolidación y Almacenamiento:**
- Se extrae la sección playerProfile de la información de cada jugador y se compila en una lista de diccionarios.
- Esta lista se convierte en un DataFrame de Pandas .
- Finalmente, el DataFrame se guarda como un archivo CSV.

- **Extracción de Datos de Equipos y Jugadores desde FBref (Web Scraping con Selenium y API)**

Este bloque se centra en obtener datos de equipos y estadísticas de jugadores de la plataforma FBref, utilizando tanto web scraping como su API.

- **Importación y Configuración de Selenium:** Se importan librerías para web scraping (`selenium`, `BeautifulSoup`)
- **Obtención de IDs de Equipos de FBref:**
  - Se navega a la URL de la tabla de la liga española 2015-2016 en FBref.
  - Se localizan todos los enlaces de los equipos dentro de la tabla y se extrae el ID de cada equipo de sus URLs.
- **Generación de API Key para FBref:** Se realiza una solicitud POST a la API de FBref para generar una clave de API que se utilizará para futuras solicitudes.
- **Extracción de Información de Equipos (Plantillas):**
  - Se itera sobre cada team_id y season_id obtenidos.
  - Se realiza una solicitud GET a la API de FBref (https://fbrapi.com/) para obtener la plantilla de cada equipo.
  - Las respuestas se almacenan en el diccionario jugadores.
- **Consolidación de Datos de Plantillas:**
  - Se recorre la estructura anidada de los datos de plantillas para extraer la información individual de cada jugador.
  - Se añade el season_id y team_id a cada registro de jugador.
  - Estos datos se consolidan en una lista de diccionarios y se convierten en un DataFrame de Pandas.
- **Extracción de Estadísticas de Temporada de Jugadores:**
  - Se realiza una nueva serie de solicitudes GET a la API de FBref para obtener estadísticas detalladas por temporada para cada jugador y equipo de la liga.

.

- **Normalización y Consolidación de Estadísticas de Jugadores:**
    - Se itera sobre los datos para extraer la información de cada jugador.
    - Se separan los datos en columnas individuales en un nuevo diccionario para cada jugador.
    - Estos diccionarios se añaden a una lista, que luego se convierte en un DataFrame de Pandas.
- **Almacenamiento:** El DataFrame se guarda como un archivo CSV.

- **Extracción de Clasificación de Liga y Lesiones desde Ceroacero.es (Web Scraping con Selenium)**

Este bloque se encarga de extraer la clasificación de la liga jornada a jornada y la información de lesiones de los equipos de la web Ceroacero.es.

- **Importación y Configuración de Selenium:** Se reimportan librerías y se configura la automatización del navegador.
- **Generación de URLs por Jornada:**
    - Se crea un diccionario que contiene una URL específica para cada una de las 38 jornadas de la liga española 2015-2016 en Ceroacero.es.
- **Extracción de la Clasificación por Jornada:**
    - Se itera a través de cada URL de jornada.
    - El navegador accede a cada URL. Se implementa lógica para aceptar cookies si aparecen.
    - Se localiza la tabla de clasificación.
    - Se extraen los encabezados de la tabla, con lógica para limpiar y asegurar nombres de columna apropiados.
    - Se extraen los datos de cada fila de la tabla (posición, equipo, puntos, partidos jugados, etc.). Se añade el número de jornada a cada registro.
    - Los datos de cada jornada se consolidan en un DataFrame.
- **Limpieza y Normalización de la Clasificación:**
    - Se elimina la columna Logo del DataFrame por no ser necesaria.
    - **Normalización de Nombres de Equipos:** Se comparan los nombres de los equipos de Ceroacero con los nombres de StatsBomb (cargados de Tabla_Partidos.csv) utilizando una función de normalización (src.normalizacion_nombres) para crear un mapeo uniforme.
    - Se utiliza este mapeo para estandarizar los nombres de los equipos en la columna Equipo de df_jornadas.
    - Se fusiona el DataFrame de jornadas con los IDs de clubes de df_wyscout (de Tabla_Partidos.csv) para añadir el clubid a la clasificación.
    - Se renombra la columna id_club_local_L a clubid.

- **Almacenamiento de Clasificación:** El DataFrame combinado se guarda como CSV.
- **Extracción de URLs y Nombres de Equipos para Lesiones:**
    - Se vuelve a iniciar Selenium para navegar a la página principal de la liga en Ceroacero.es.
    - Se extraen los identificadores únicos (segmentos de URL) y los nombres de todos los equipos presentes en la tabla.
    - Se genera una URL específica para la sección de "indisponibles" (lesionados/sancionados) de cada equipo.
- **Extracción de Datos de Lesiones por Equipo:**
    - Se itera sobre las URLs de "indisponibles" de cada equipo.
    - El navegador accede a cada URL, aceptando cookies si es necesario.
    - Se localizan las tablas de lesiones (si existen).
    - Se extraen los encabezados y los datos de cada fila (jugador, tipo de lesión, etc.).
    - Estos datos se concatenan en un DataFrame.
- **Extracción de Nombres Completos de Jugadores (para Lesiones):**
    - A partir de los href de los jugadores lesionados, se vuelve a utilizar Selenium para visitar la página de perfil de cada jugador en Ceroacero.es.
    - Desde el perfil, se extrae el nombre completo del jugador para asegurar consistencia.
- **Mapeo y Normalización Final de Lesiones:**
    - Se mapean los nombres completos de los jugadores y los nombres estandarizados de los equipos al DataFrame.
- **Almacenamiento de Lesiones:** El DataFrame final se guarda como CSV.

- **CSV Wyscout:**
    - Desde una web con acceso privado se descargan y concatenan en excel varios archivos con estadísticas de todos los equipos en cada partido de la liga.

1.2. Transformación (Transform)

Una vez extraídos los datos, se realizan diversas operaciones para limpiarlos, enriquecerlos y adaptarlos para el análisis:

- **Transformación de datos jugadores:**
    - Los datos extraídos con la información de cada jugador se analizan y modifican para una correcta carga a la BBDD.
    - Se modifican los nombres que aparecían duplicados como “Sergio Álvarez” ya que corresponden a dos jugadores diferentes con ID diferente.
    - Se normalizan nombres e IDs de equipos y jugadores para que coincidan con la tabla de Eventos.
    - Se eliminan columnas innecesarias.
- **Transformación de datos lesiones:**

    - Los datos extraídos con la información de todas las lesiones registradas se modifican para una correcta carga a la BBDD.
    - Se crea un ID único por lesión y tipo.
    - Se normalizan nombres e IDs de equipos y jugadores para que coincidan con la tabla de Eventos.
    - Se eliminan columnas innecesarias.
- **Transformación de datos equipos:**
    - Los datos extraídos con la información de cada jugador se analizan y modifican para una correcta carga a la BBDD.
    - Se normalizan nombres e IDs de equipos y jugadores para que coincidan con la tabla de Eventos.
    - Se eliminan columnas innecesarias.

1.3. Carga (Load)

Tras la transformación de los datos se carga cada archivo en una tabla de la BBDD creada en PostgreSQL.

EDA (Análisis Exploratorio de Datos)

Esta fase se centra en entender las características principales de los datos, identificar patrones, detectar anomalías y probar hipótesis iniciales, a menudo con la ayuda de visualizaciones.

Para realizar este proceso se realizado el análisis en Python y en la fase posterior se realizará un análisis a través de las visualizaciones en PowerBI que se reflejan en el Dashboard final. Todo el análisis está basado en el Athletic Club ya que es el club objetivo de este proyecto y no es necesario analizar de forma individual el resto de equipos de la liga.

En Python se realiza un análisis de las estadísticas descriptivas de los diferentes dataframes generados en fases anteriores.

Este análisis se divide en estudio de los datos globales como disparos, goles, pases, etc., un análisis de las lesiones y un análisis individual de los componentes de la plantilla.

Dentro del análisis de los datos globales se encuentran datos relevantes como;

- La edad media es más alta que la media de la liga y el ATH tiene menos internacionales que la media.
- Dominio local: En la mayoría de estadísticas consideradas como de ataque el ATH destaca por encima de la media cuando juega como local. Supera la media en remates, ataques posicionales, centros y corners entre otros. Por los datos se puede inferir que el equipo prefiere atacar por bandas ya que el número de centros es casi 1 punto superior a la media de la liga y casi 8 duelos aéreos más. Los datos también sugieren un ritmo de juego alto como muestran el alto nº de pérdidas de balón, una menor duración de las acciones y más de 7 pases hacia delante respecto a la media de la liga.
- Repliegue jugando de visitante: Los datos como visitante también muestran un equipo dominador ya que es superior a la media en muchas estadísticas pero los datos reflejan un cambio de estilo respecto a ser local. El análisis de los datos sugiere que, jugando como visitante, el ATH trata de hacer un juego algo más directo que se refleja en la diferencia en ataques posicionales de 3 respecto a su media como local. Otro aspecto destacable es que como visitante recupera más balones en el campo rival y completa menos pases.

En el análisis de lesiones se puede observar que el número total de lesiones fue de 14, teniendo una media inferior a la liga en todos los tipos de lesión y concentrado en un total de 8 jugadores, siendo Iñaki Williams el jugador con más lesiones del equipo.

El análisis individual de cada futbolista se realizará en powerBI ya que para analizar mapas de calor, tiros, posicionamiento, etc. es más útil desde esta herramienta.

Informe final

Tras haber concluido el análisis de los datos extraídos, y siguiendo con el esquema de presentación del proyecto, las conclusiones finales se dividen en tres grandes bloques;

- **Composición de la plantilla:** La edad media de la plantilla es ligeramente superior a la media de edad de la liga, esto no es per se un problema pero puede derivar en un aumento del riesgo de lesiones y un menor rendimiento físico.

- **Estilo de Juego y Rendimiento:**
    - **Dominio Ofensivo:** El análisis demuestra que el ATH es uno de los equipos con mejores registros ofensivos, siendo el 4º equipo más goleador, y teniendo mejores números que la media de la liga en tiros, ataques posicionales, corners y centros entre otros registros ofensivos.
    - **Juego por banda:** El mapa de calor muestra una clara tendencia al juego por banda. Este modelo de juego se ve respaldado por los números ya que el ATH es el equipo que más duelos aéreos gana.
    - **Ritmo de Juego Local:** Se propone un ritmo alto de partido lo que se respalda viendo la alta cantidad de pérdidas de balón, una menor duración de acciones y un mayor número de pases hacia adelante respecto al número de pases totales.
    - **Adaptación y Repliegue:** Al jugar como visitante, el ATH muestra una ligera adaptación táctica hacia un juego más directo y un repliegue más efectivo, evidenciado por menos ataques posicionales pero más recuperaciones en campo rival.
    - **Fortaleza en duelos:** Tanto en duelos aéreos, como en duelos totales (ofensivos y defensivos) el equipo está entre los mejores registros. Esta fortaleza refuerza la idea de un fútbol directo y con predominancia de las bandas.

- **Hallazgos Clave de Lesiones:**
    - **Incidencia General:** A pesar de un total de 14 lesiones, el Athletic Club se mantuvo por debajo de la media en todos los tipos de lesión, lo que sugiere una gestión de carga efectiva o menor propensión a lesiones graves entre la plantilla.
    - **Concentración en Jugadores:** Las lesiones se concentraron en solo 8 jugadores, siendo Iñaki Williams el más afectado.

- **Hallazgos Individuales de Jugadores:**
    - A nivel estadístico destacan Aritz Aduriz (máximo goleador) y Ager Aketxe (mayor numero de regates, % de duelos ganados y mejor % de pases completados)
    - El 11 inicial lo componen: Gorka Iraizoz, Oscar De Marcos, Xabi Exeita, Aymeric Laporte, Mikel Balenziaga, Beñat, Mikel San José, Iker Muniain, Ibai Gomez, Markel Susaeta y Aritz Aduriz.
    - Varios de los jugadores se encuentran superando el percentil 90 en varias estadísticas lo que sugiere una temporada extraordinaria a nivel individual.

Next steps

Como siguientes pasos para continuar el análisis se propondrían;

- Añadir filtro en cuadro de ojeo cumpliendo la filosofía del ATH (jugadores vascos o formados en EuskalHerria) que se ha inciado en la carpeta de extras.
- Eficientar proceso de ETL y automatizar la descarga de partidos desde statsbomb
- Conseguir datos actualizados
- Creación de pestañas y análisis detallado por fase del juego