# 

**TFM**

**ANÁLISIS CARACTERÍSTICAS DE LOS JUGADORES DEL FIFA 2019**

**APRENDIZAJE SUPERVISADO - REGRESIÓN**

**JULIÁN MARCOS VÁZQUEZ**

**INDICE:**

[1. INTRODUCCIÓN: 3](#_Toc25701615)

[2. DESCRIPCIÓN DE LOS DATOS: 4](#_Toc25701616)

[3. FASES 4](#_Toc25701617)

[4. METODOLOGÍA UTILIZADA: 5](#_Toc25701618)

[5. RESULTADOS 6](#_Toc25701619)

[6. DIFICULTADES ENCONTRADAS 8](#_Toc25701620)

[7. CONCLUSIONES 8](#_Toc25701621)

# INTRODUCCIÓN:

En los últimos años, la obtención de datos tanto de productos como de clientes, se han transformado en el petróleo del siglo XXI para las empresas. Poder analizarlos en profundidad y en tiempo real para sacar conclusiones orientadas a mejorar el modelo de negocio y aumentar los ingresos se ha convertido en el objetivo fundamental.

Aunque el concepto de Data Science no es nuevo, ha aumentado el volumen y la velocidad con la que se generan los datos y nuestra capacidad para analizarlos, teniendo en cuenta que el 90% de los datos del mundo han sido creados durante los últimos dos años.

Por eso, Aena, decidió crear un nuevo departamento de Análisis de Datos para conocer cuál es el perfil del pasajero y de ahí surgió la necesidad de realizar este máster. A su vez, está llevando a cabo un proceso de transformación digital que permitirá alimentar las bases de datos con información relativa a los datos de los usuarios y experiencia de pasajero, los cuales, posteriormente, hay que explotar.

Desde la división de marketing, tenemos la intención de recopilar los datos en bruto, estudiarlos, elaborar dashboards, pasando por machine learning y aplicar métodos estadísticos en la resolución de problemas habituales del Data Science. Este conocimiento permitirá potenciar las actividades comerciales que se traducirán en un incremento de las ventas en los aeropuertos de la red.

Para eso, se necesitarán estos perfiles especializados que combinen la analítica y la estrategia con la parte técnica.

Por todo esto, decidí elegir un dataset que pudiera asemejarse al que en un futuro podamos obtener en Aena, un conjunto de individuos con distintas características al cual podamos darle un valor que nos permita diferenciarle del resto de pasajeros y clientes.

# DESCRIPCIÓN DE LOS DATOS:

El dataset escogido es “*FIFA 19 complete player dataset*“, del site “*Kaggle.com*”, que cuenta con 18207 filas, correspondientes cada una a un jugador diferente del prestigioso juego.

<https://www.kaggle.com/karangadiya/fifa19>

A su vez, el fichero tiene 89 columnas, todas ellas correspondientes a las distintas características del jugador, aunque no utilizaré todas ellas para mi trabajo fin de máster, sino solo las más relevantes para la metodología que quiero utilizar.

# FASES

Para la realización de este proyecto, para poder utilizar más recursos de los estudiados en clase, he seguido los siguientes pasos:

1. Limpieza de datos: Dado el dataset original, decidí hacer una limpieza de los jugadores que tuvieran alguno de los campos vacíos (nan). Además, dado que el peso de los jugadores venía en libras y la altura en pies y pulgadas, decidí pasar estas columnas a Kilos y centímetros, aunque luego, no me hicieron falta en la resolución por machine learning.
2. Visualización de datos: Decidí representar algunos datos interesantes como la Distribución de jugadores por países, mostrando un diagrama de barras con los países top ten más numerosos.

Además, dibujé un boxplot con los países cuya puntuación de los jugadores era mayor.

1. Utilización de machine learning para que, a partir de unas variables de entrada, tanto numéricas como categóricas, se haga la predicción de la puntuación total del jugador en el juego.
2. Utilización de machine learning para que, a partir de un algoritmo clasificador, pueda obtener la posición que debería tener cada jugador a partir del resto de variables.

NOTA: Realizaré una comparativa de los datos obtenidos la primera vez que presenté el TFM, donde sólo incluía variables numéricas, con esta segunda vez que incluye 3 variables categóricas.

# METODOLOGÍA UTILIZADA:

Para realizar mi proyecto, decidí utilizar **machine learning**. Dado que mi objetivo consistía en obtener la puntuación global de cada jugador, dependiendo del valor obtenido en el resto de variables, decidí utilizar un **método de aprendizaje supervisado**, ya que mi intención es encontrar una función que dadas las variables de entrada (input data), les asigne un valor de salida adecuada. El algoritmo entrena con un conjunto de datos ya etiquetados y así “aprende” a asignar la etiqueta de salida adecuada a un nuevo valor, prediciendo el valor de salida.



Además, dentro del aprendizaje supervisado, dado que mi variable objetivo es de tipo numérica y continua, opté por usar **una regresión**, en vez de un algoritmo de clasificación.

***ALGORITMOS:***

Una vez me decidí por realizar una regresión dentro del aprendizaje supervisado, realicé los cálculos de los 3 algoritmos vistos en clase para comprobar cuál de todos me ofrecía mejores resultados:

1. **Regresión lineal**: Permite determinar el grado de dependencia de las series de valores X e Y, prediciendo el valor y estimado que se obtendría para un valor x que no esté en la distribución.,
2. **K-vecinos**: predice cogiendo los k vecinos más cercanos. Parte de la idea de que una nueva muestra será clasificada en base a la cual pertenezca la mayor cantidad de vecinos más cercanos del conjunto de entrenamiento.
3. **Árboles de decisión:** Un árbol de decisión está formado por un conjunto de nodos de decisión (interiores) y de nodos-respuesta (hojas):

Un nodo de decisión está asociado a uno de los atributos y tiene 2 o más ramas que salen de él, cada una de ellas representando los posibles valores que puede tomar el atributo asociado.

Una hoja o leaf, está asociado a la clasificación que se quiere proporcionar, y devuelve la decisión del árbol con respecto al ejemplo de entrada.

Además, utilicé el método aprendido en clase (Grid SearchCV) y que indica cuál es el valor óptimo de los parámetros tanto para K-vecinos como para Árboles de decisión.

***MÉTRICAS:***

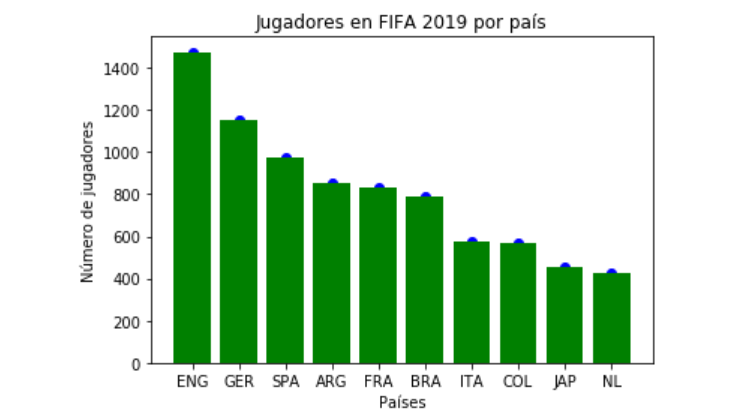
Dentro de cada algoritmo, he usado las siguientes métricas para comprobar la eficiencia de cada uno de ellos utilicé las métricas utilizadas en clase:

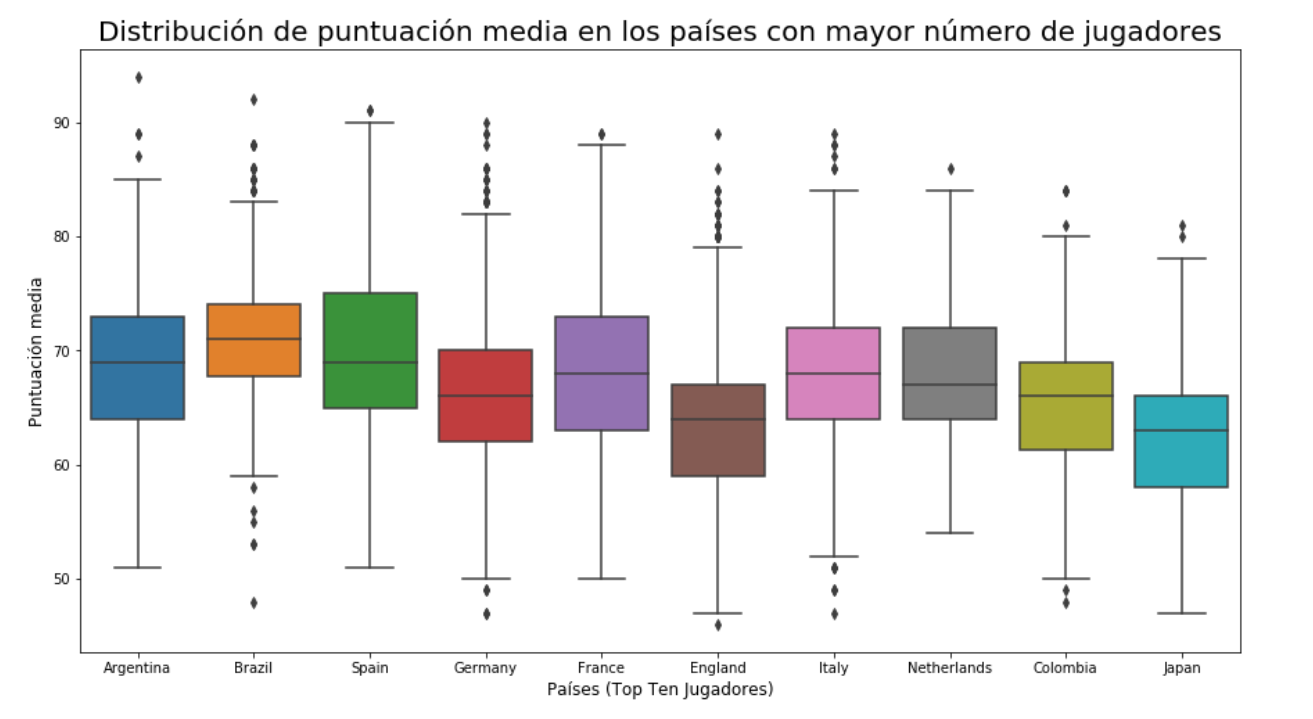
1. **MAE:** Error absoluto medio: es la suma de los valores absolutos del error
2. **MSE:** Error cuadrático medio, es la media de la diferencia entre los puntos reales de datos y la salida predicha, al cuadrado
3. **RMSE:** Raíz del error cuadrático medio, es la raíz de la media de los errores elevados al cuadrado.

# RESULTADOS

**FASE DE VISUALIZACIÓN:**

A la hora de visualizar resultados, viendo el dataset, decidí mostrar los ratos más relevantes a mi parecer:





**FASE DE APLICACIÓN DE MACHINE LEARNING:**

Después de aplicar los correspondientes algoritmos a los datos “limpios” de la fase 1, los resultados obtenidos fueron los siguientes:

**SOLO VARIABLES NUMÉRICAS (V1- diciembre 2019):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Linear Regression** | **K-Neighbors (k=4)** | **GridSearchCV (k=25-óptimo)** | **Decission Tree (Depth=3)** | **GridSearchCV (depth=6-óptima)** |
| **MAE** | 3,483 | 3,492 | **3,193** | 5,257 | 5,064 |
| **MSE** | 21,55 | **19,057** | 19,562 | 42,590 | 41,827 |
| **RMSE** | 4,642 | **4,365** | 4,422 | 5,977 | 5,874 |

**CON VARIABLES NUMÉRICAS Y CATEGÓRICAS (V2 – febrero 2020):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Linear Regression** | **K-Neighbors (k=4)** | **GridSearchCV (k=8-óptimo)** | **Decission Tree (Depth=3)** | **GridSearchCV (depth=19-óptima)** |
| **MAE** | 2,06 | 1,815 | **1,754** | 5,473 | 3,619 |
| **MSE** | 6,847 | 6,481 | **5,191** | 39,9 | 20,579 |
| **RMSE** | 2,616 | 2,545 | **2,278** | 6,012 | 4,279 |

Dado que MSE es la métrica que menos aporta, me centraré en los valores de MAE y RMSE. Respecto a éstos, MAE es más robusto, aunque menos sensible a los valores atípicos que RMSE, pero esto no significa que siempre sea mejor usar el MAE.

Como puede observarse, hay una diferencia significativa entre los datos que obtuve en diciembre (con una variable como combinación de 3 características de los jugadores que tenían mayor coeficiente) y los que he obtenido ahora introduciendo todas las variables numéricas y algunas categóricas. De hecho, los valores obtenidos ahora son incluso mejores. Aún así, los mejores resultados los sigue dando el algoritmo k-neighbors, pero en esta ocasión los mejores datos los da siempre usando el valor óptimo que nos aporta el GridSearchCV, ya que la MAE y RMSE son los más bajos de toda la comparativa.

A continuación, como pasó la vez anterior, el algoritmo que mejores valores ha obtenido ha sido la regresión lineal, y por último, el que peores resultados ha calculado ha sido el árbol de decisión.

# DIFICULTADES ENCONTRADAS

Dado que antes de comenzar el máster mis conocimientos en Python eran nulos y los de estadística escasos, la realización del TFM me ha resultado bastante laboriosa, ya que para cualquier línea de código que se saliera del estándar, tenía que estar consultando en Internet y tardaba mucho tiempo en tener una sentencia sin errores. Lo mismo ha ocurrido con la valoración de los resultados. Mi escasa experiencia en este campo hacía que dudara si cualquier resultado era correcto o si me había equivocado al aplicar las fórmulas o las funciones estadísticas.

En este caso, a diferencia con lo presentado en diciembre, todo el código se encuentra en el fichero llamado “ML\_Regresion\_Multiple.ipynb”. La diferencia es que cuando comencé realizando machine learning utilizando como variable X el conjunto de todas las características que pueda tener un jugador y pasé varios días intentando poder dibujar la recta de regresión lineal, hasta que Tony Almagro me dijo que no se podía dibujar el plot, ya que estaba utilizando muchas variables en xgrid y solo una en la predicción. Para hacer un plot normal solo podía tener una variable de dimensión 1 en x, y una variable de dimensión 1 en la y, por lo que, en su lugar, decidí hacer varios plot, uno para cada una de las variables contenidas en xgrid. A raíz de esto, decidí terminar el ejercicio calculando las métricas y realicé uno nuevo notebook utilizando una variable X que fuera la media de 3 de las características de los jugadores, tal y como habíamos hecho en el ejercicio de clase con Tony Almagro, pudiendo así obtener la recta de regresión para los 3 algoritmos. Sin embargo, en esta segunda entrega no he dibujado la recta de regresión, y he decidido poner la variable X como el conjunto de variables numéricas y categóricas.

En definitiva, realizando el TFM me he dado cuenta de que hay muchas cosas que obviamos durante las clases y que luego nos ralentizan el trabajo cuando nos enfrentamos a un problema real.

# CONCLUSIONES

A pesar de que no creo que pueda aplicar el código desarrollado para mi proyecto de fidelización en la empresa, su realización me ha ido dando muchas ideas que podré aplicar, tanto para este proyecto, como para el nuevo departamento de análisis de datos que se va a implantar en Aena.

Además, he podido comprobar que, aunque GridSearchCV suele dar los mejores resultados tanto para k-vecinos como para Árboles de decisión, en este caso, no se ha cumplido. Ya que visualmente dibujé el MAE para un gran espectro de vecinos y el k con menor MAE no coincidía con el que me dio GridSearchCV. Dada mi reducida experiencia en este campo, esto me hizo pensar que el desarrollo que estaba realizando estaba bien.

Otra gran duda que tuve fue saber qué variables utilizar en mi modelo simple, ya que podía haber utilizado cualquiera de ellas. Al final me decidí por las 3 que me daban el mayor valor de coeficiente.