ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE UN MODELO PREDICTIVO PARA EL ANÁLISIS DE ANOMALÍAS DE DESEMPEÑO EN FUTBOLISTAS PROFESIONALES UTILIZANDO MINERÍA DE OPINIONES

TESIS PREVIA A LA OBTENCIÓN DEL GRADO DE MÁSTER (MSc)
EN SISTEMAS DE INFORMACIÓN
MENCIÓN EN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS

DANNY CRISTÓBAL ALBUJA GONZÁLEZ
danny.albuja@gmail.com

DIRECTORES:
Ph.D. Edison Loza Aguirre
edison.loza@epn.edu.ec

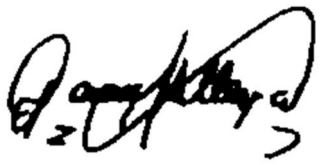
Ph.D. Lorena Recalde
lorena.recalde@epn.edu.ec

Quito, diciembre 2021

DECLARACIÓN

Yo, Danny Cristóbal Albuja González, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

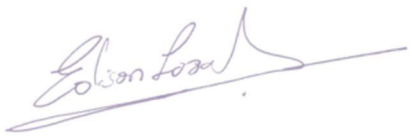
A través de la presente declaración cedo mis derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.



Danny Cristóbal Albuja González

CERTIFICACIÓN

Como director del trabajo de titulación DESARROLLO DE UN MODELO PREDICTIVO PARA EL ANÁLISIS DE ANOMALÍAS DE DESEMPEÑO EN FUTBOLISTAS PROFESIONALES UTILIZANDO MINERÍA DE OPINIONES desarrollado por el señor Danny Cristóbal Albuja González con cédula de ciudadanía 1715789960, estudiante de la Maestría en Sistemas de Información Mención en Inteligencia de Negocios y Analítica de Datos Masivos, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, puedo certificar la finalización del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la Defensa Oral.



Ph.D. Edison Loza Aguirre

Director del Proyecto

Ph.D. Lorena Recalde

Codirectora del Proyecto

DEDICATORIA

Esta tesis va dedicada a mi papá y mamá, siempre he podido contar con su apoyo, comprensión y empuje para poder seguirme convirtiendo en una mejor versión de mi persona.

De igual forma va para mis sobrinos, que sea un buen ejemplo de todo lo que pueden alcanzar con esfuerzo y a sus papás que los apoyen siempre en sus estudios, que los impulsen a alcanzar su mejor potencial, pero sin olvidar el seguir siendo buenas personas.

Espero poder hacer grandes aportes y así empezar a ayudar al mundo desde este campo de conocimiento, aunque sea una persona a la vez.

Índice de Contenido

LISTA DE FIGURAS

LISTA DE TABLAS

RESUMEN

En la actualidad el fútbol requiere herramientas que confirmen lo que se percibe de forma empírica, para corroborar mediante estadísticas, cuándo un jugador está bajando su desempeño y puedan tomarse los correctivos necesarios.

En el presente proyecto se pretende cumplir con dos metas. Primero, establecer un mecanismo que permita identificar anomalías en el rendimiento deportivo de los futbolistas profesionales, entendiéndose como anomalía, el desarrollo (en términos de rendimiento y valoración financiera) de un deportista que no haya cumplido su proyección como futura estrella. A partir de la identificación de estos rendimientos anómalos, en segundo lugar, se pretende desarrollar un modelo que permita explicar estos comportamientos mediante minería de opiniones tomando como fuente de dato lo que se ha dicho del deportista en la red social Twitter.

Para implementar estas tareas se seguirá la metodología CRISP-DM, además de procesos de extracción de información, ejecución de librerías de Python para procesamiento de texto, entre otras herramientas; implementando técnicas de Inteligencia Artificial y Aprendizaje de Máquina.

Al final, se pudo relacionar las opiniones de las personas con el desempeño del jugador, de forma que se pronostica si este jugador es riesgoso y sirva de apoyo en la toma de decisiones sobre el futuro del jugador dentro del equipo de fútbol.

Palabras Clave: Fútbol, Desempeño, Twitter, Python, Análisis de sentimientos, Word Embeddings

ABSTRACT

Nowadays, soccer requires tools to confirm through statistics when a player's performance is dropping, and consequently, propose corrective. The present project aims to accomplish two goals. The first one focuses on establishing a mechanism to identify anomalies in the performance of professional soccer players. We understand as an anomaly, the development (in terms of performance and financial valuation) of an athlete who has not fulfilled his projection as a future star. Secondly, based on the identification of these anomalous performances, we intend to develop a model that allows explaining these behaviours through opinion mining from what has been said about the athlete in Twitter. The relevant information will be extracted from people's tweets and analysed through Sentiment Analysis, in addition to word cleaning and vectorization processes.

To do so, the CRISP-DM methodology will be followed, in addition to information extraction processes, execution of Python libraries for text processing, among other tools; all for implementing methods of Artificial Intelligence and Machine Learning.

At the end, it was possible to relate people's opinions with players' performance to predict if a player is risky and to support the decision-making process about the player's future within the soccer team.

Keywords: Soccer, Performance, Twitter, Python, Sentiment analysis, Word Embeddings

1 CAPÍTULO I: INTRODUCCIÓN

1.1 Contexto

La minería de datos se ha vuelto clave en la toma de decisiones en diferentes ámbitos ya que vivimos en un mundo donde grandes cantidades de datos se recolectan diariamente. Analizar esos datos es un requerimiento importante, que se debe satisfacer mediante el uso de herramientas que permitan extraer conocimiento de esa información en diferentes campos [1]. Así, el campo deportivo es una de las temáticas donde se ha explotado el uso de la minería de datos. En el futbol la analítica de datos ha jugado un rol importante en la consecución de títulos por parte del Liverpool de Inglaterra, por ejemplo [2]; o en un caso más cercano, en nuestro país, los logros del Independiente del Valle descritos en el artículo de la Revista Gestión como “El éxito de las estadísticas aplicadas al fútbol” [3].

En la actualidad el negocio del fútbol mueve billones de dólares al año, de acuerdo con un informe publicado en 2019 por la empresa Deloitte [4]. En este negocio, las ganancias de los equipos están estrechamente relacionadas con el rendimiento deportivo; lo cual depende a su vez de contratar jugadores que mantengan un alto desempeño y que aporten con presencia en la cancha, siendo regulares en sus carreras. Es por esto que, en varios equipos, la labor de reclutamiento de nuevos talentos es altamente importante.

A pesar de la relevancia que se le da a la búsqueda de jugadores, hasta donde tenemos conocimiento, no se ha establecido un modelo que permita prever el rendimiento de un jugador en contraste con las opiniones que se registran en las cuentas de la red social Twitter, sean estas cuentas pertenecientes a prensa especializada o a usuarios que mencionen a los jugadores que van a ser evaluados.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un modelo predictivo para el análisis de anomalías de desempeño en futbolistas profesionales utilizando minería de opiniones usando herramientas de código abierto.

1.2.2 Objetivos Específicos

- Realizar una revisión de la literatura sobre los trabajos existentes relacionados con la predicción de rendimiento deportivo.
- Diseñar un método de clasificación de futbolistas profesionales en las diferentes ligas basado en sus características deportivas y su valor de mercado.

- Aplicar métodos de minería de opiniones para recabar los comentarios que existen en Twitter sobre jugadores profesionales de fútbol y clasificarlos como positivos o negativos.
- Mapear las opiniones negativas en Twitter sobre jugadores profesionales para establecer una relación con las anomalías en el desempeño deportivo de tal manera que sirvan como apoyo para la labor de los buscadores de talentos.

1.3 Alcance

Dentro de los objetivos de esta tesis se plantea el crear un modelo que sirva de soporte para toma de decisiones en la industria del futbol, pero no se pretende desarrollar el front-end de interacción con el usuario final. El modelo resultante de este trabajo puede ser embebido en una plataforma donde se busque clasificar a un jugador como alto riesgo o bajo riesgo basado en el modelo que sea obtenido con valores de la temporada actual y de información recolectada previamente de Twitter. En este sentido, el sistema no se actualizará de forma automática con nuevos datos de cada año ni tampoco se reentrenará por sí mismo con las opiniones de las personas expresadas en redes sociales en tiempo real.

1.4 Marco Teórico

1.4.1 Inteligencia Artificial

En palabras simples, lo que busca la Inteligencia Artificial (IA) es que los sistemas o máquinas lleguen a imitar la inteligencia humana para poder realizar tareas y mejorar continuamente a partir de la información que se va obteniendo de conocimientos previos [5]. En la actualidad vemos varios ejemplos de esta rama de la informática como son, entre otros:

- Los chatbots que buscan dar soluciones a los problemas de los clientes y proporcionar respuestas más eficientes.
- Los asistentes inteligentes que mediante el uso y procesamiento de grandes conjuntos de datos dan respuestas adecuadas a lo que un usuario espera obtener.
- Los motores de recomendación que dan sugerencias para programas de TV, películas, etc. según los hábitos que los usuarios tienen.

Aunque Hollywood muestra imágenes de robots de aspecto humano de alto funcionamiento que se apoderan del mundo, la IA no pretende reemplazar a los humanos. Su objetivo es mejorar significativamente las capacidades y contribuciones humanas. Eso la convierte en un activo empresarial muy valioso [6].

1.4.2 Aprendizaje de Máquina

Conocido en inglés como Machine Learning, es una rama de la IA que consiste en extraer conocimientos de los datos. Es un campo de investigación en la intersección de la estadística, la IA y la informática, que también es conocido como análisis predictivo o aprendizaje estadístico [7].

Como se observa en la Figura 1, dentro del aprendizaje de máquina tenemos tres tipos [8]:

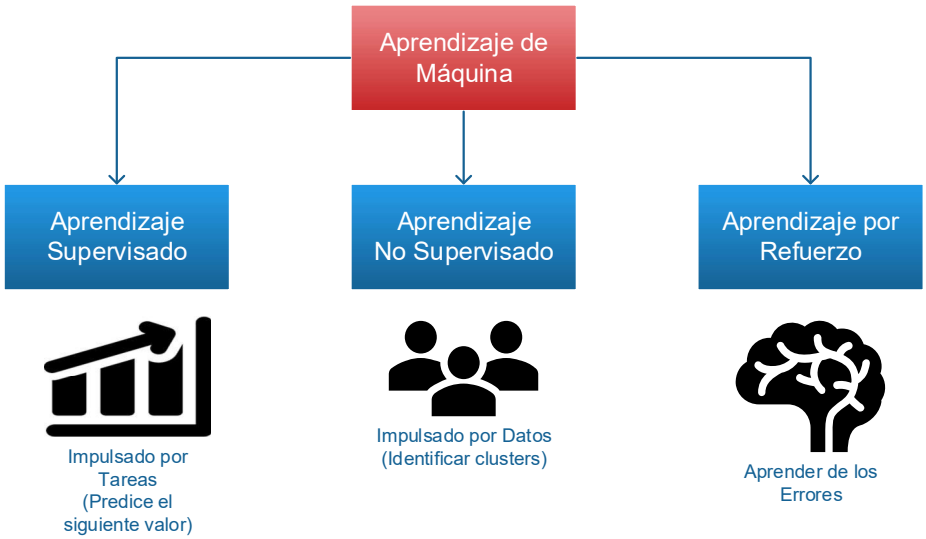


Figura 1. Tipos de aprendizaje de máquina [9]

1. Aprendizaje supervisado: Algoritmos que aprenden a partir de un conjunto de entrenamiento de ejemplos que han sido etiquetados para caracterizar al conjunto de todas las entradas posibles. Algunos ejemplos de estas técnicas son: regresión logística, máquinas de vectores de apoyo, árboles de decisión, bosques aleatorios, etc.
2. Aprendizaje no supervisado: Algoritmos que aprenden a partir de un conjunto de entrenamiento de ejemplos que no han sido etiquetados. Se utiliza para explorar los datos según algún criterio estadístico, geométrico o de similitud. Algunos ejemplos de técnicas son: clustering de k-means y la estimación de la densidad del kernel.
3. Aprendizaje por refuerzo: Algoritmos que aprenden a partir de críticas recibidas sobre la calidad de una solución.

1.4.3 Herramientas utilizadas en el proyecto

Para este trabajo de titulación se tiene previsto usar herramientas de software libre como:

- **Pentaho Community Edition**

  Es una suite completa con todas las funcionalidades necesarias para el correcto desarrollo de proyectos de Inteligencia de Negocios. Es una versión comunitaria, sin costos de licencia ni servicios de soporte asociados. Dentro del proyecto de tesis será utilizado para la tarea inicial de cargar archivos CSV dentro de las tablas de una base de datos.

- **MySQL**

  Es un gestor de bases de datos relacional considerado como la base de datos de código abierto más popular del mundo. Este gestor tiene una versión de tipo Community, distribuida bajo Licencia pública. Esta base de datos será la utilizada para almacenar los registros que se van a insertar en la carga inicial de información de los jugadores, los mismos que son extraídos de archivos planos de tipo CSV.

- **Anaconda – Python**

  Es una plataforma de programación para lenguajes Python y R, utilizada en ciencia de datos, y aprendizaje automático. De igual forma es una distribución libre. Esta plataforma será usada en gran parte del proyecto de titulación, al ser un lenguaje muy extendido para la minería de datos. Además, Anaconda provee de librerías y documentación suficiente que puede ayudar durante las tareas que se tienen previstas para obtener el modelo de análisis de jugadores.

- **Twitter – Tweepy**

  Para el caso de recolección de opiniones de las personas, se va a utilizar Tweepy que es una librería de Python que sirve para la extracción de información de la red social Twitter mediante un API creado por la misma red social, a la cual se puede acceder mediante un registro en su plataforma de desarrollo. Se decidió usar Twitter ya que es una red social gratuita y de acceso mundial, donde los usuarios pueden compartir sus opiniones, gustos e intereses sobre diferentes temas de forma textual.

  Esta red social servirá para recolectar las opiniones de sus usuarios sobre un grupo de jugadores, de forma que se pueda extraer toda esta información de los tweets de los usuarios para luego evaluarlas con análisis de sentimientos, y para que, en los pasos finales del proyecto, sea procesada para obtener el modelo que permita clasificar a estos jugadores.

- **TextBlob**

  Esta librería de Python permite evaluar un texto de forma que se pueda caracterizar el sentimiento que tenía el autor de ese texto. Es decir, si el autor quería expresar odio

o amor, la librería evaluará las palabras y hará esta clasificación de forma automática, clasificando el texto de acuerdo a una polaridad positiva, negativa o neutral [10].

- **Word2Vec**

  Es un algoritmo de aprendizaje no supervisado desarrollado por Google en el 2013. En el sitio web donde se encuentra alojado el proyecto Word2Vec la documentación lo define como: “una implementación eficiente que crea una arquitectura de bolsa de palabras que permite calcular representaciones vectoriales de palabras” [11]. Este algoritmo servirá para poder buscar relaciones entre un conjunto de palabras que tengan cierta similitud como cuando el cerebro humano relaciona conceptos en diferentes ámbitos. En la Figura 2 podemos observar en forma simple como este algoritmo trata de emular las relaciones que existen entre las palabras dentro del lenguaje humano. Para esto, Word2Vec asigna vectores a cada una de las palabras y si existen palabras que se relacionen entre sí, las ubica de forma cercana dentro del plano vectorial.

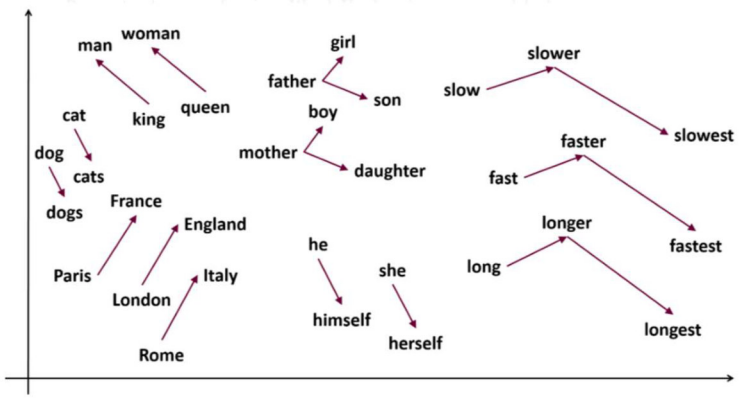


Figura 2. Ejemplo de relaciones entre palabras [12].

2 CAPÍTULO II: METODOLOGÍA

La metodología que se usará en este proyecto de titulación será CRISP-DM, cuyas siglas en inglés se refieren a un Proceso Estándar Interindustrial para Minería de Datos. Es una metodología ampliamente usada en proyectos de minería de datos cuya popularidad radica en tener bien definidas las fases a seguirse para poder estructurar los datos obtenidos y poder relacionar cada uno de los resultados con las fases posteriores. Su popularidad se ha medido en varias ocasiones [13], tal como se puede ver en la Figura 3 donde se ven los resultados de una encuesta realizada en varios años por el sitio https://www.kdnuggets.com/ y otra encuesta realizada en el 2020 por el sitio https://www.datascience-pm.com/.

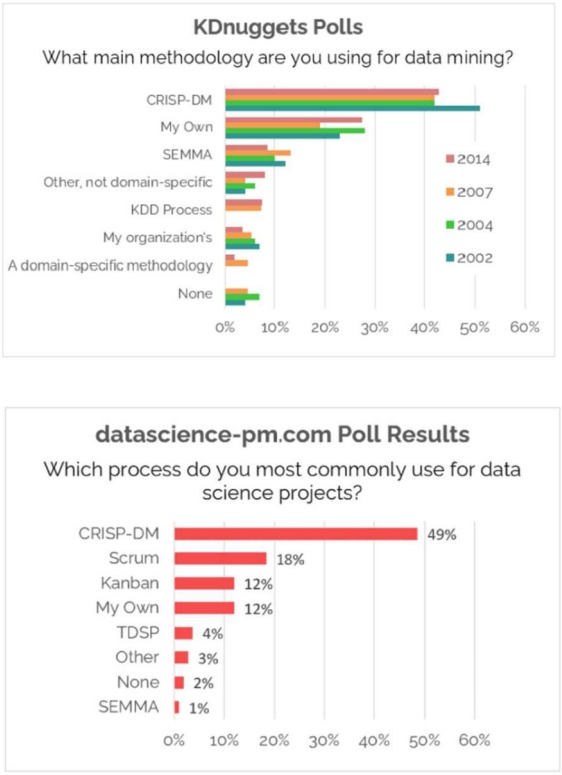


Figura 3. Resultados de encuestas sobre el uso de CRISP-DM [13].

Dentro de esta metodología contamos con las siguientes fases (Figura 4):

- Conocimiento del negocio, etapa en la cual se busca entender el objetivo de las metas y los datos con los que cuenta un negocio en particular.
- Conocimiento de los datos, en donde determinamos características de los datos recolectados al igual que la calidad de estos.
- Preparación de los datos, en esta fase preparamos los datos, se realiza limpieza, estandarización, categorizaciones, etc.

- Modelado, en esta fase se desarrolla el modelo cuyos resultados deberán satisfacer lo que estamos esperando obtener del estudio.
- Evaluación, en este paso validamos los resultados obtenidos y se busca dar explicación de la información resultante.
- Implementación, en esta última fase el objetivo es usar el modelo obtenido de la metodología y aplicarlo dentro del negocio para obtener un beneficio del estudio realizado.

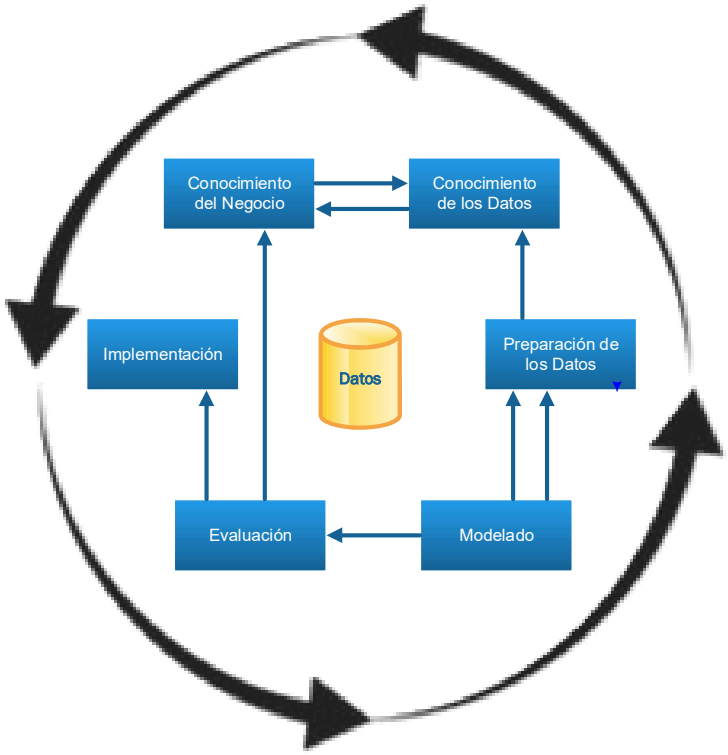


Figura 4. Metodología CRISP-DM [13].

2.1 Comprensión del Negocio

2.1.1 Determinar los Objetivos del Negocio

Lo primero a realizar será obtener información deportiva de diferentes archivos de datos, procesar esa información y estructurarla de tal forma que se pueda segmentar a los jugadores de acuerdo con diferentes características. Esto permitirá clasificar entre jugadores que estén rindiendo de buena forma en contraste con los jugadores que no estén respondiendo de la forma que se tenía proyectado.

El objetivo final es mapear las opiniones negativas en Twitter sobre jugadores profesionales para establecer si existe una relación con las anomalías en el desempeño deportivo de tal manera que sirvan como apoyo para la labor de los buscadores de talentos al identificar cómo evoluciona un jugador y tratar de prevenir que su desempeño deportivo vaya en declive.

2.1.2 Evaluación de la Situación

Actualmente los estudios sobre análisis de sentimientos han tomado relevancia sobre temas diversos, sin embargo en el área del deporte, la investigación significativa se ha destinado al análisis de sentimientos para *i)* evaluar la percepción de los fans frente a la copa mundial del 2014 [14], *ii)* medir la participación de las personas en sitios web deportivos [15], y *iii)* determinar los sentimientos previos a un partido de tenis de 31 jugadores de esta disciplina [16]. En contraste, en términos de estudios que incluyan minería de opiniones desde una perspectiva más general, los trabajos existentes se han enfocado en la medición de la percepción de políticos de EEUU [17]; por lo tanto, en el contexto de deportes, o particularmente del fútbol, no se han investigado anomalías en el desempeño de los jugadores contrastado con minería de opiniones.

2.1.3 Determinar los Objetivos de la Minería de Datos

Lo primero que se buscará será establecer un método de clasificación de futbolistas profesionales basado en sus características deportivas y su valor de mercado. Para esto se tomará en cuenta su desempeño durante cada una de las temporadas al evaluar el valor general que es asignado para cada jugador dentro del set de datos de FIFA. El cálculo se basa en una clasificación general que se le asigna a cada jugador dentro del juego, así como seis puntuaciones para las estadísticas clave: ritmo, disparo, pase, regate, defensa y físico. Estas estadísticas se combinan con el reconocimiento internacional de un jugador para calcular su puntuación general [18].

Lo siguiente que se buscará hacer es recolectar información de Twitter para contrastar los datos que se obtuvieron en el paso anterior, para esto se aplicará minería de opiniones para recabar los comentarios que existen en Twitter sobre jugadores profesionales de fútbol y clasificarlos como positivos o negativos.

2.1.4 Plan del Proyecto

El plan de proyecto detallado en la Tabla 1 abarca todos los pasos que se van a realizar en cada una de las 7 fases de CRISP-DM y la duración aproximada de cada una de las actividades que se va a realizar. Al final de la tabla tendremos el total de horas que se tiene estimado para completar todo el proyecto.

Tabla 1.Descripción del plan de proyecto

Actividad		Duración (horas)
Conocimiento del negocio	Revisar literatura sobre detección de anomalías en el contexto deportivo.	20
	Evaluar los repositorios de datos de EA Sports	10
	Revisar la información financiera de Transfermrkt.	10
	Revisar pruebas de minería de opiniones en Twitter.	10
Conocimiento de los datos	Recolectar y clasificar los datos	20
	Obtener información financiera de jugadores por temporada.	30
Preparación de los datos	Limpieza y preprocesamiento de datos	20
	Integración de datos	10
	Crear conjuntos de datos para entrenamiento y validación.	10
Modelado	Identificación de anomalías	20
	Selección de anomalías para estudio	10
	Modelamiento de texto para el mineo de opiniones de Twitter	10
Evaluación	Comparación de datos de anomalías	10
	Comparación de resultados de algoritmos de predicción	10
Implementación	Validación del artefacto	5
Generación del reporte	Redacción del documento de titulación	5
Total		210

2.2 Conocimiento de los Datos

2.2.1 Recolección de los Datos Iniciales

Dentro del proyecto de tesis se ha decidido utilizar 2 fuentes de datos: SOFIFA y Twitter. SOFIFA[1] es un sitio web que contiene los datos que se han recopilado por parte de la compañía EA Sports dedicada al desarrollo de videojuegos y que centra sus intereses en grandes ligas deportivas, no solo en el ámbito del fútbol ya que su alcance ha incursionado en otras disciplinas como básquet, fútbol americano, béisbol, etc. FIFA dedica un gran esfuerzo a recolectar información sobre el desempeño y características deportivas de los jugadores durante cada temporada para poder utilizarla en sus videojuegos deportivos.

Para poder acceder a toda esta información de SOFIFA de forma más sencilla y estructurada se usó un repositorio dentro del sitio de datos masivos Kaggle el cual ha sido obtenido con varias técnicas de extracción de datos. El autor de estos archivos es Stefano Leone[2], quien

[1] https://sofifa.com/
[2] https://www.kaggle.com/stefanoleone992

lleva más de 3 años como colaborador de este sitio web y es calificado dentro de Kaggle como un Experto en Datasets.

Se consideró otra fuente de datos, el repositorio Transfermarkt[3]. El objetivo principal de este sitio es mantener datos actualizados de los valores de mercado de los jugadores de futbol para su explotación comercial. Sin embargo, durante el desarrollo de este proyecto se liberó una nueva versión de SOFIFA en Kaggle el 08 de octubre de 2020 que contenía los datos del valor de mercado de los jugadores, por lo tanto, se decidió usar los datos de valores de mercado esta misma fuente.

En resumen, los datos de EA Sports fueron desplegados en el sitio web de SOFIFA y posteriormente convertidos en archivos CSV publicados en Kaggle; estas relaciones se ven en la Figura 5.

Figura 5. Relación entre fuentes de datos.

La segunda fuente de datos usada en el presente trabajo de titulación consiste en extraer información de Twitter. Para esto se usó la librería de Python llamada Tweepy que permite interactuar con el API de esta red social para la extracción de datos. Para el uso de Tweepy es necesario registrarse en el sitio https://developer.twitter.com/ donde se crea una cuenta para análisis de datos que será utilizada en pasos posteriores. El objetivo principal es poder extraer los comentarios hechos sobre jugadores en la red social. Con estos datos se hará un análisis de sentimientos para poder determinar si el desempeño de los jugadores tiene relación con la percepción de estos por parte de los usuarios de esta red social. La relación entre las herramientas de esta segunda fuente de datos usada se puede ver en la Figura 6.

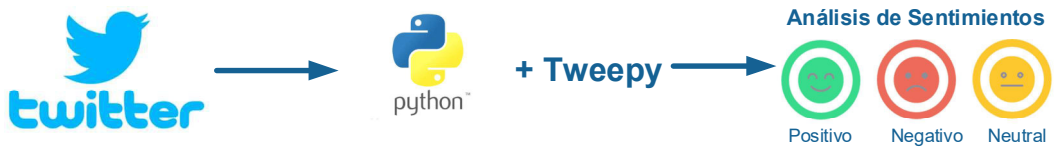


Figura 6. Manejo de datos de Twitter.

[3] https://www.transfermarkt.com/

2.2.2 Descripción de los Datos

Los datos de SOFIFA obtenidos de Kaggle consisten en siete archivos de tipo CSV que contienen registros de las temporadas 2015 al 2021. Cada archivo está compuesto de un total de 106 columnas. Para el presente trabajo se ha procedido a cargar estos datos en una base de datos de tipo MySQL. Sin embargo, luego de un estudio de los datos, se determinó que solo se va a tomar en cuenta la información contenida en las columnas listadas en la Tabla 2.

Tabla 2. Columnas seleccionadas para el estudio.

Campo	Descripción del Contenido
sofifa_id	Identificador único de cada jugador
long_name	Nombre del jugador
club_name	Club al que pertenece en esa temporada
age	Edad
overall	Consiste en un valor nominal del desempeño general de un jugador
potential	Es el valor de potencial que tiene un jugador, es decir, la expectativa que se tiene sobre el rendimiento de un jugador
value_eur	Valor de mercado del jugador en euros durante esa temporada

2.2.3 Exploración de los Datos

Cada uno de los archivos CSV recolectados corresponden a una temporada. Entre cada temporada, existen jugadores que se añaden o se eliminan de acuerdo a como se mueve el mercado deportivo del fútbol. Por lo tanto, en cada temporada se tendrá un diferente número de registros. La cantidad de jugadores en cada archivo se detalla en la Tabla 3.

Tabla 3. Cantidad de registros por cada archivo CSV.

Año	Número de Jugadores
2015	16155
2016	15623
2017	17597
2018	16800
2019	18085
2020	18483
2021	18944

2.2.4 Calidad de los Datos

Los datos de EA Sports disponibles en Kaggle corresponden a aquellos extraídos de la página SOFIFA. Los datos, que tienen una alta tasa de descargas dentro del sitio Kaggle, son calificados por los usuarios como datos con alta usabilidad y según la información de la misma

página web, el dataset utilizado ha sido descargado 5943 ocasiones a fecha de 21 de octubre de 2021.

Como se puede ver en la Figura 7, el sitio web permite visualizar los datos contenidos en los archivos CSV. Subrayado en amarillo tenemos una de las columnas del dataset que contiene la página exacta de donde se sacó los datos del jugador, en este caso la URL pertenece a los datos de Lionel Messi.

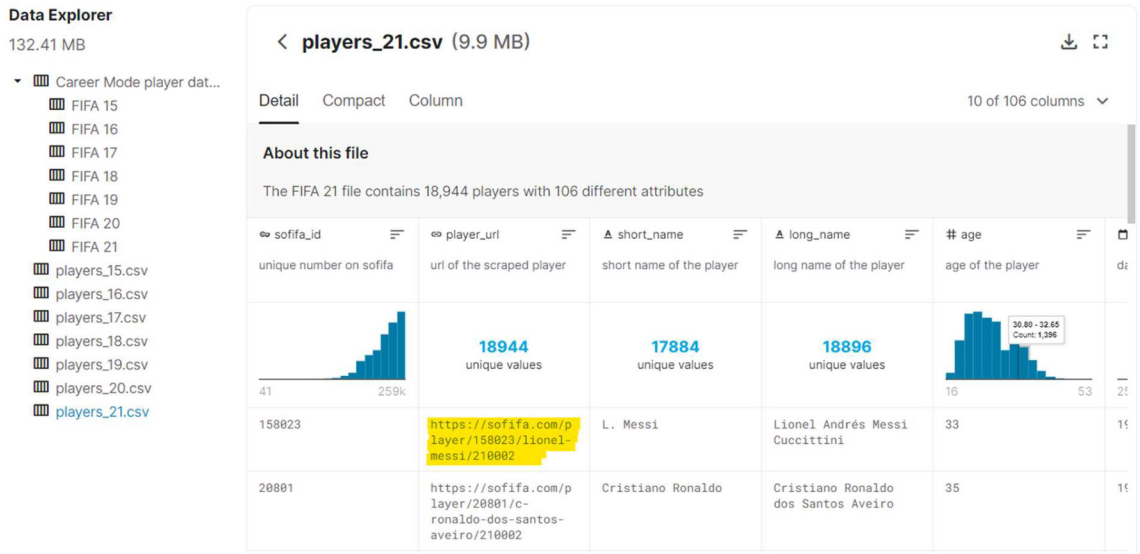


Figura 7. Dataset desplegado dentro del sitio Kaggle.

Si se utiliza esa URL en cualquier navegador, en la página desplegada (Figura 8) se podrán ver los datos generales del jugador, y a su vez se puede comparar con los datos que se visualizaron en la figura anterior donde se tiene el detalle de los archivos CSV en el sitio Kaggle. Se realizaron verificaciones aleatorias para comprobar la exactitud de los datos de los archivos CSV con los del sitio.

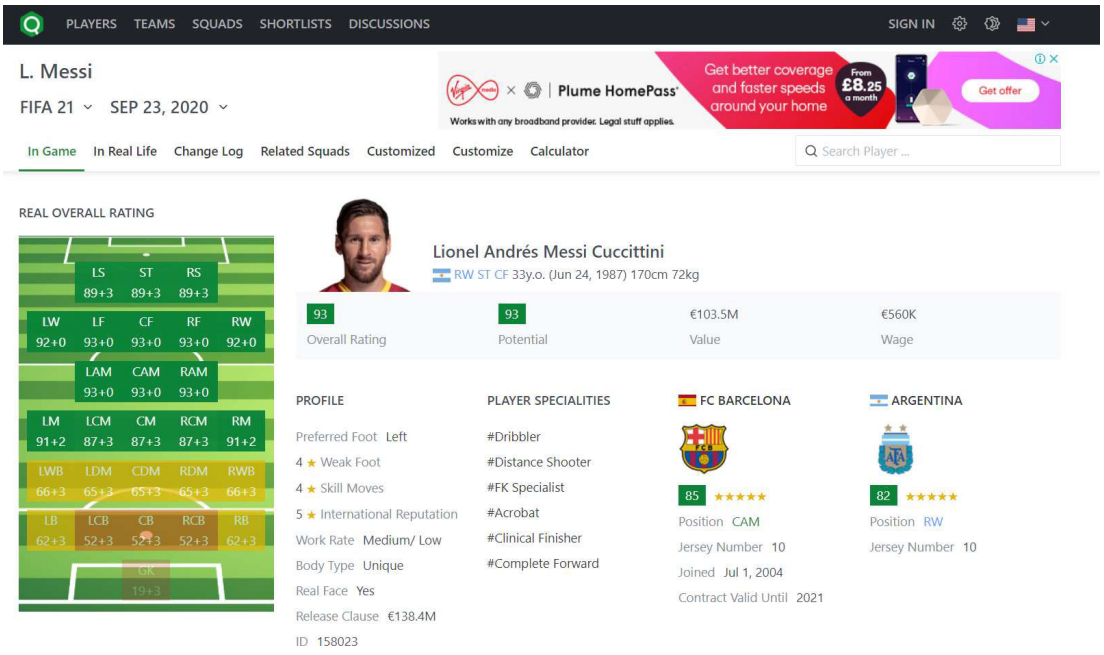


Figura 8. Información general del jugador Lionel Messi en el sitio SOFIFA.

En las columnas seleccionadas para el estudio, no se encontraron datos faltantes o con problemas de formato que impidan su explotación. Cabe señalar que las valoraciones asignadas siguen siendo cifras subjetivas de la calidad de un deportista, las cuales corresponden a evaluaciones realizadas por expertos de EA Sports. Así, aunque no son exactas, si corresponden al resultado de una evaluación sistemática y coherente.

2.3 Preparación de los Datos

2.3.1 Seleccionar los Datos

Como tarea inicial se cargó la información obtenida del repositorio de datos Kaggle referente a la página SOFIFA. La información está constituida en varios archivos CSV que almacenan los valores del 2015 al 2021 de los jugadores de fútbol. Se procedió a cargar estos registros dentro de una base de datos MySQL utilizando el programa Spoon que es parte de la suite Pentaho Data Integration. Se decidió utilizar esta herramienta ya que es de tipo código abierto y compatible con la base de datos que se quiere utilizar. Esta herramienta permite diagramar ETLs (Extract, Transform, Load) que tienen la funcionalidad de procesar diferentes fuentes de datos, que en nuestro caso eran los archivos CSV.

Por ejemplo, se toma el archivo de jugadores del año 2021, se lo ordena de acuerdo con un identificador único que se le asigna a cada jugador, y se procede a crear el script de la nueva tabla (Anexo 1). Al final, se puede ya empezar a cargar los registros extraídos del archivo CSV. Los pasos se pueden ver en la Figura 9.

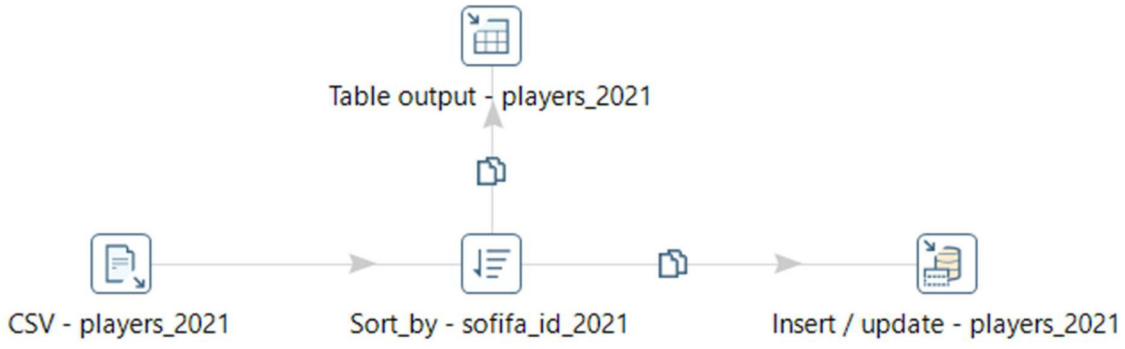


Figura 9. Extracción de registros de un archivo CSV a una tabla de MySQL.

El mismo proceso se repitió con cada uno de los 7 archivos CSV. El tiempo de carga fue de aproximadamente media hora ya que se tenía un total de 121.687 registros con 106 columnas cada uno.

2.3.2 Limpieza de los Datos

En Anaconda, mediante el uso del lenguaje Python, se extrajeron los datos cargados en MySQL. Todo el código de esta fase se lo puede ver en el Anexo 2. Como se dijo anteriormente, este set de datos contiene información que va desde el año 2015 al 2021. Debido a que muchos jugadores tan solo aparecen durante el 2015, se debió filtrar esos datos mediante un identificador único entre todos los archivos para evaluar cuantos jugadores se mantuvieron vigentes durante los 7 años. Aplicado ese filtro inicial, nuestro grupo de datos final es de tan solo 4031 jugadores.

Con el grupo de 4031 registros, lo primero que se calculó es el valor del Máximo Desempeño del jugador durante estos 7 años. Para esto se recopiló el valor de Overall de cada temporada y se sacó el valor más alto que será almacenado en la columna MaxOverall, El siguiente paso fue usar la función Describe de Python para determinar diferentes estadísticas de esa columna, entre ellas el valor medio, tal como se ve en la Figura 10. El valor medio es de 73.90 aproximadamente y con una desviación estándar de 5.75

```
dfComplete['maxOverall'].describe()

count    4031.000000
mean       73.895559
std         5.748644
min        57.000000
25%        70.000000
50%        74.000000
75%        78.000000
max        94.000000
Name: maxOverall, dtype: float64
```

Figura 10. Descripción matemática del valor de máximo desempeño.

Con el valor del máximo desempeño vs la edad de los jugadores, al proceder a usarlo dentro un gráfico de tipo de barras, se determinó que el valor donde se inicia el mayor desempeño es a los 21 años. Esto se lo puede observar fácilmente en la curva de la Figura 11.

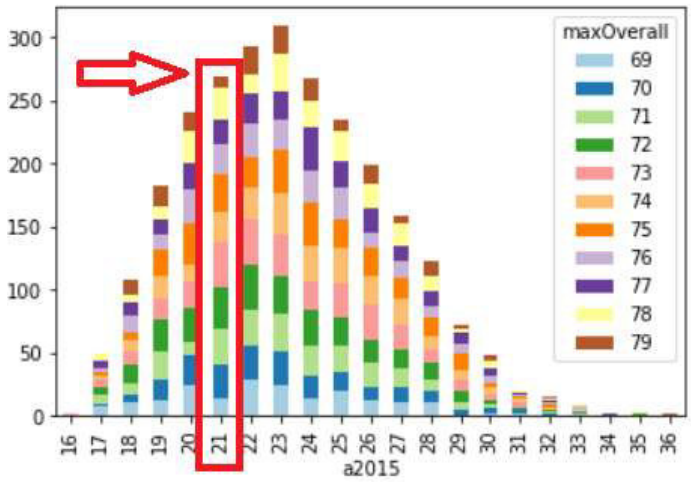


Figura 11. Distribución del desempeño máximo comparado con la edad.

Ya escogida la edad de 21 años como el inicio del máximo desempeño de un jugador, se pudo tomar la decisión de solo usar el grupo de jugadores menores a 28 años para el año 2021, ya que se cuenta con información de 7 temporadas. Al hacer este filtrado se llegó a una cantidad de 1394 registros con los cuales se continuará el resto del proceso.

2.3.3 Construcción de Datos

Primer Cálculo: Potencial vs Máximo Desempeño

El primer cálculo que se determinó fue el resultante entre el máximo valor de desempeño de un jugador restado del potencial promedio que se tenía proyectado para el jugador durante ese rango de 7 años.

Operacion1 = avgPotential - maxOverall

(Promedio del Potencial - Máximo desempeño)

Con este dato se desea encontrar cuáles jugadores fueron los que menos se acercaron a su rendimiento esperado. En la Figura 12 se refleja que la mayoría de los 1394 jugadores está agrupada dentro del intercuartil que va entre 0.86 y 3.82.

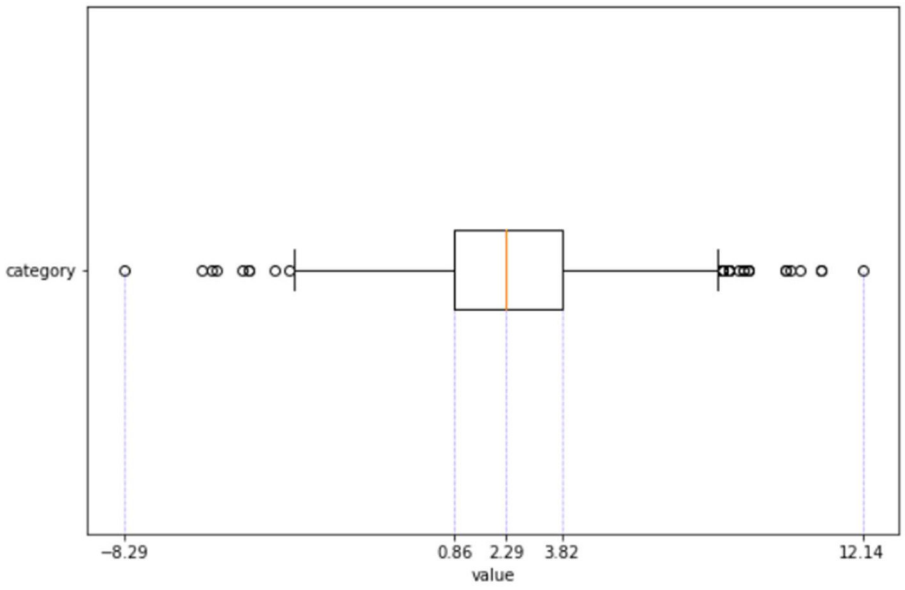


Figura 12. Cuartiles de potencial vs máximo desempeño.

De igual forma, en la Figura 13 se pueden observar los valores atípicos o outliers que serán los de peor desempeño, cuyo valor en la Operación1 fue mayor a 8.43.

```
dfCalculos.quantile([0.01, 0.25, 0.5, 0.75, 0.99])
```

	sofifa_id	a2015	avgPotential	maxOverall	operacion1
0.01	192011.79	17.0	63.275714	61.0	-3.428571
0.25	205753.00	19.0	71.714286	68.0	0.857143
0.50	212197.00	20.0	75.142857	73.0	2.285714
0.75	219981.75	21.0	79.142857	77.0	3.821429
0.99	225368.49	21.0	87.438571	87.0	8.428571

Figura 13. Valores de cada uno de los cuartiles.

Con ese valor, al filtrar los resultados del set de datos, se determinaron 17 jugadores que son los que peor se han desempeñado durante estos 7 años basados en los resultados de la Operación1 que se puede observar encerrado en color rojo en la Figura 14.

	sofifa_id	long_name	a2015	c2021	avgPotential	maxOverall	operacion1
3143	212782	Hiram Boateng	18	Milton Keynes Dons	72.000000	63	9.000000
3205	213519	Brad Walker	18	Shrewsbury	72.000000	61	11.000000
3225	213729	Sondre Løvseth Rossbach	18	Odds BK	77.857143	69	8.857143
3381	215425	Filip Sachpekidis	16	Kalmar FF	70.857143	62	8.857143
3409	215818	Emerson Hyndman	18	Atlanta United	77.000000	67	10.000000
3594	219995	Vittorio Parigini	18	Genoa	76.428571	68	8.428571
3610	220140	Frankie Kent	18	Peterborough United	73.428571	65	8.428571
3753	222079	Josh Onomah	17	Fulham	81.000000	72	9.000000
3754	222095	Callum Burton	17	Cambridge United	70.428571	60	10.428571
3799	222836	Ryan Ledson	16	Preston North End	79.142857	67	12.142857
3802	222878	Tyler Walker	17	Coventry City	74.714286	66	8.714286
3804	222922	Jake Hesketh	18	Southampton	76.000000	67	9.000000
3894	223756	Marvin Schulz	19	FC Luzern	75.428571	67	8.428571
3912	224021	Oluwaseyi Babajide Ojo	17	Cardiff City	77.142857	67	10.142857
3919	224103	Erick Germain Aguirre Tafolla	17	Pachuca	79.428571	71	8.428571
3981	224925	Gianluca Gaudino	17	BSC Young Boys	80.000000	69	11.000000
4010	225279	Conor McGrandles	18	Lincoln City	72.000000	62	10.000000

Figura 14. Jugadores que son atípicos al tener un valor de desempeño mayor a 8.43.

Segundo Cálculo: Incremento del valor de mercado entre el 2015 al 2021

El segundo cálculo está enfocado en determinar el incremento del valor de mercado de los jugadores, para esto se resta el valor de mercado del 2021 contra el del 2015 cuando iniciaban sus carreras y luego se divide para el valor del año 2015.

Operación 2 = v2021-2015 / v2015

(Valor de mercado en el 2021 menos el 2015, dividido para el valor del 2015)

Para tener una idea de cómo ha fluctuado el valor de mercado para uno de estos jugadores atípicos o outliers, en la Figura 15 se puede ver que para el jugador Hiram Boateng se tiene un incremento de aproximadamente 5 veces de su valor de mercado inicial al comparar los datos de la temporada 2015 en contraste con la del 2021.

long_name	v2015	v2021	operacion2
Hiram Boateng	80000	475000	5.0

Figura 15. Incremento en el valor de mercador del jugador Hiram Boateng.

Si se procede a graficar esos valores de mercado de los 17 jugadores con peor desempeño, se obtiene la Figura 16 que da a entender que tienen una tendencia similar entre ellos, a excepción de un solo jugador (Josh Onomah) que tiene un incremento y decremento atípico.

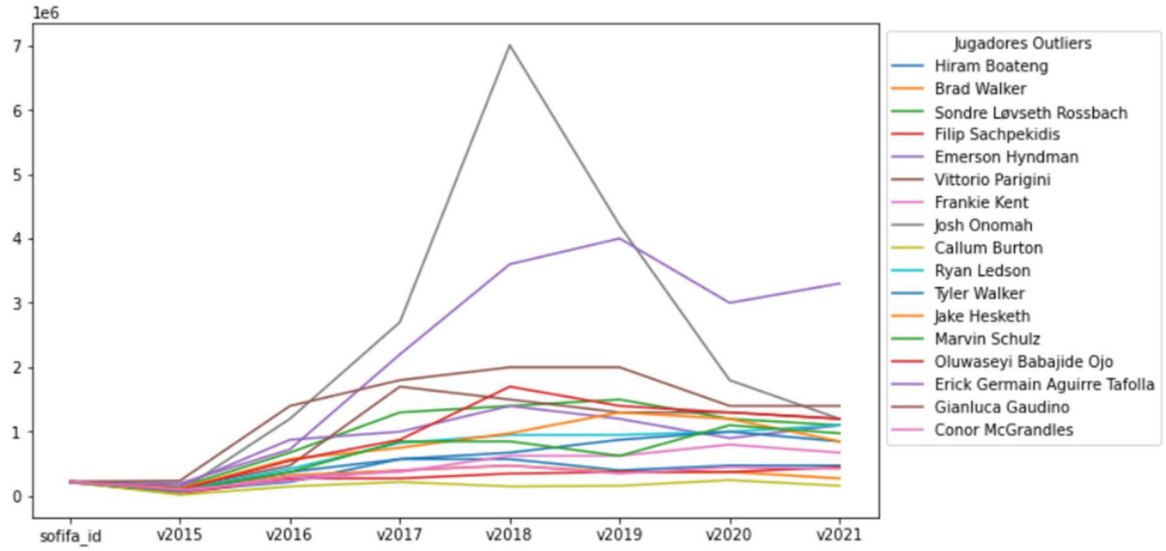


Figura 16. Incremento de los sueldos de los jugadores atípicos malos.

Pero, si se toma esa misma información y se la compara con el resto de los cálculos de valor de mercado de los 1377 jugadores con desempeño general, se pueden observar dos líneas claramente distanciadas en cuanto a este incremento de su valor de mercado año a año. Tenemos un incremento de mercado mucho mayor y sostenido para todos los jugadores de desempeño típico que en contraste con los jugadores atípicos de bajo desempeño que tienen una línea muy por debajo del valor general, tal como se puede ver en la Figura 17.

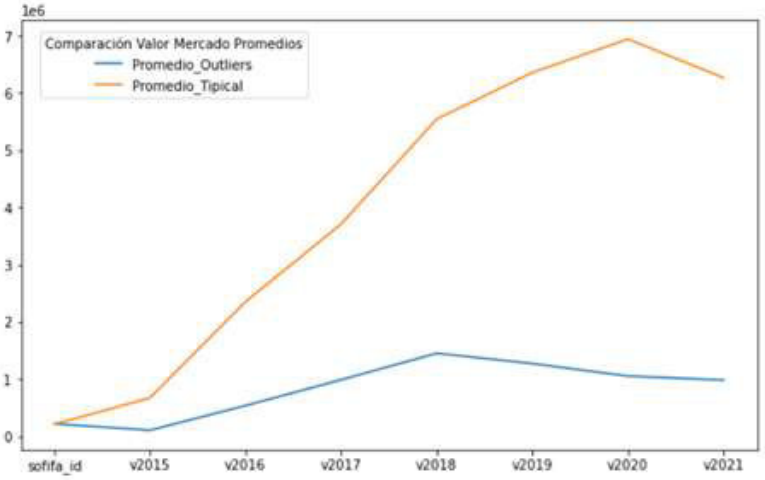


Figura 17. Comparativa del valor de mercado entre jugadores típicos y atípicos.

2.3.4 Integración de Datos

Ya establecido el grupo de 1394 jugadores a usarse para este análisis, en esta fase se procedió a usar la librería de Python denominada Tweepy, la cual sirve para conectarse al API de desarrollo de Twitter y acceder a las funciones que provee esta red social. En este caso particular, el objetivo era obtener tweets referentes a los jugadores activos y almacenar el texto que posteriormente se procederá a procesar, limpiar y evaluar. Todo el código con este proceso está en el Anexo 3.

Cada recolección demoró alrededor de 3 días de ejecución del proceso. Esto incluyó caídas de la señal y tiempos de espera ya que se usa una cuenta gratuita del API de Twitter que tiene varias limitaciones, entre ellas el que se pueda extraer un máximo de 300 tweets cada 15 minutos [19].

Se hizo la recolección en dos fechas diferentes, la primera ocasión fue en mayo del 2021 obteniendo un total de 148.318 tweets y en una segunda ocasión, un mes después, se recolectó 141.605, dando un total de 289.923 tweets. De este grupo de Tweets tan solo 709 jugadores fueron mencionados durante la recolección de información, donde la mediana de ese valor de recolección es de tan solo 16 tweets por jugador, tal como se ve en la Figura 18.

```
dfTotalConteos.TotalTweets.describe()

count     709.000000
mean      209.193230
std       489.774306
min         1.000000
25%         3.000000
50%        16.000000
75%       118.000000
max      2000.000000
Name: TotalTweets, dtype: float64
```

Figura 18. Promedio de tweets extraídos por cada jugador.

Los tweets recolectados equivalen a un total de 289.923 registros, los cuales se buscaron usando los nombres de los jugadores filtrados de SOFIFA. De cada jugador se buscó recolectar un máximo de 2000 tweets para tener una cantidad significativa de información. Además, se decidió usar un solo lenguaje, en este caso inglés, para poder usar las librerías existentes de las cuales existe suficiente documentación y ejemplos en ese idioma. Para preparar y limpiar estos 289.923 Tweets se siguieron los siguientes pasos:

1. Remover URLs
2. Remover menciones a otros usuarios de Twitter

3. Remover el signo de hashtag #
4. Remover el signo de guión bajo
5. Remover el signo de arroba
6. Remover los dos puntos
7. Remover las letras RT, esto para eliminar los retweets
8. Estandarizar caracteres usando unidecode
9. Remover stop words[4]
10. Aplicar lematización[5]

Completada esta limpieza, el siguiente paso fue remover los Tweets duplicados. Al final de este proceso, se obtuvo un total de 100272 tweets únicos, como se ve en la Figura 19.

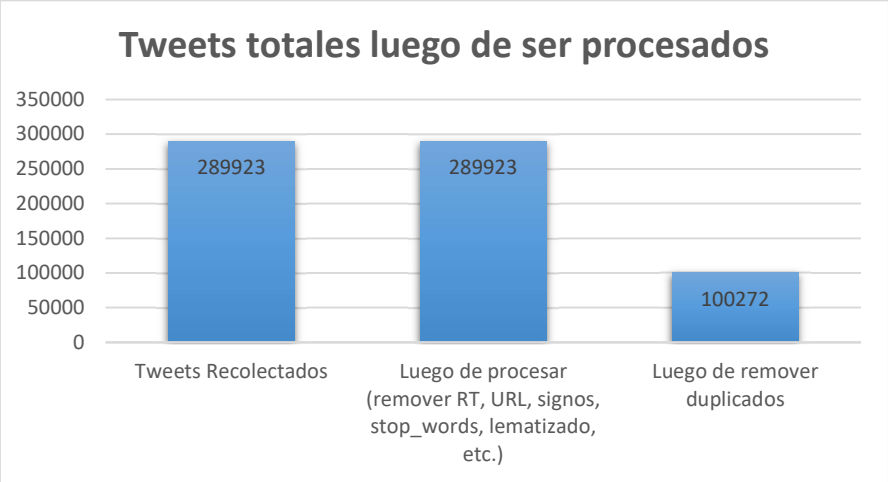


Figura 19. Cantidad de tweets luego de procesarlos.

El siguiente cuadro representa el número de tweets recolectados para los jugadores outliers de mal desempeño representado (Figura 20).

[4] En español, se consideran stop words a las preposiciones, conjunciones, artículos, adverbios, pronombres y algunos verbos.
[5] Relaciona una palabra flexionada o derivada con su forma canónica o lema. Ejemplo: Las palabras canto, cantas, cantamos, cantan son distintas conjugaciones del mismo verbo cantar.

	sofifa_id	long_name	TotalTweets
407	212782	Hiram Boateng	1
437	213519	Brad Walker	454
480	215818	Emerson Hyndman	1
546	220140	Frankie Kent	4
601	222079	Josh Onomah	112
602	222095	Callum Burton	75
624	222836	Ryan Ledson	10
627	222878	Tyler Walker	453
629	222922	Jake Hesketh	2

Figura 20. Cantidad de Tweets recolectados de los outliers de mal desempeño.

De igual forma, la Figura 21, nos muestra la cantidad de Tweets referentes a los outliers de buen desempeño.

	sofifa_id	long_name	TotalTweets
40	199904	Jesse Joronen	2
64	200855	George Baldock	70
92	202048	Conor Coady	767
582	221479	Dominic Calvert-Lewin	1894
584	221531	Marko Maroši	155
586	221587	Joe Lolley	15
648	223597	Ruben Aguilar	6
651	223689	Wout Weghorst	561
652	223697	Robin Gosens	2000
653	223724	Stefan Lainer	67
702	225375	Konrad Laimer	242

Figura 21. Cantidad de Tweets recolectados de los outliers de buen desempeño.

Para la integración se procede a usar los 100.272 tweets retenidos para aplicar Análisis de Sentimientos, para lo cual se usó la librería TextBlob, que se encarga de tomar un tweet y clasificarlo de acuerdo a si el texto habla positiva o negativamente sobre el jugador; en caso de no tener una tendencia a alguno de estos dos polos, se lo clasifica como un tweet neutral [10]. En base a esta librería, obtuvimos 38.576 tweets calificados como positivos, 16.678 negativos y el resto son neutrales (45.018 tweets). Los valores clasificados se pueden ver en la Figura 22.

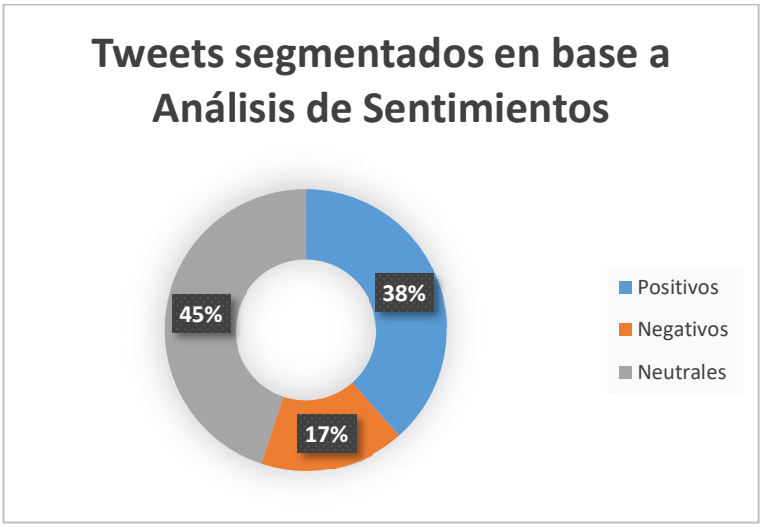


Figura 22. Porcentajes de tweets agrupados de acuerdo con el análisis de sentimientos.

2.3.5 Formateo de los Datos

A continuación, se agruparon las palabras de los tweets clasificados como positivos y de igual forma las palabras de los tweets clasificados como negativos mediante un proceso de tokenización que consiste en segmentar el tweet en palabras para poder evaluarlos de forma unitaria. En total, de los 100.272 tweets, se obtuvieron 32.733 palabras únicas clasificadas de forma positiva y negativa luego de haber usado la librería de TextBlob. Este resultado se puede ver en la Figura 23.

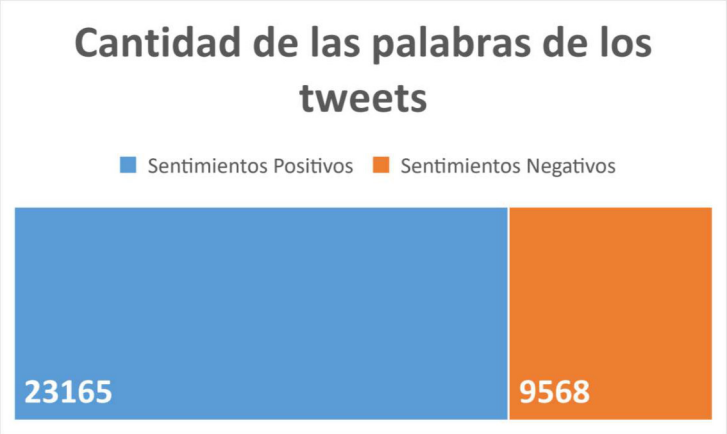


Figura 23. Cantidad de palabras de los tweets clasificados como positivos o negativos.

Ya teniendo este grupo de palabras definido, lo que se procedió a hacer es remover las palabras comunes dentro de estos dos grupos, es decir, si una palabra aparece tanto en el grupo de palabras positivas o negativas se la remueve, de forma que tengamos palabras que no aparezcan en ambos grupos, similar a lo que se puede observar en la Figura 24.

Figura 24. Operación para remover palabras comunes.

La cantidad final de palabras por grupo, luego de agrupar las palabras de acuerdo con el tipo de sentimiento y restar las palabras comunes entre ambos grupos se puede observar en la Figura 25.

Figura 25. Cantidad de palabras sin valores comunes entre sentimientos positivos y negativos.

Lo que quiere decir que se obtuvieron 11.274 palabras positivas y 4.234 negativas.

2.4 Modelado

2.4.1 Técnica de Modelado

Para el modelado de los datos se usó la librería Word2Vec para convertir las palabras de los Tweets en vectores y de esta forma poder realizar operaciones sobre ellos. Esta sección de código se puede ver en el Anexo 4. Lo que se procedió a hacer es:

1. Usar los tweets previamente clasificados basados en análisis de sentimientos:
    a. Tweets positivos con un total de 38.576
    b. Tweets negativos con un total de 16.678
    c. Tweets neutrales con un total de 45.018
2. Crear los 3 espacios vectoriales de Tweets en base al análisis de sentimientos del paso anterior usando Word2Vec y un modelo final que incluya todos los 100.272 Tweets.
3. Calcular los centroides vectoriales de cada uno de los espacios vectoriales positivos, negativos y neutrales para validar que no se superpongan entre sí.
4. Validar las distancias de los jugadores respecto a los centroides.

2.4.2 Plan de Prueba

En este punto se verificó cómo se clasifican los jugadores inicialmente segmentados basados en su desempeño deportivo. Para esto se usaron los jugadores atípicos con peor desempeño y se procedió a calcular si los tweets asociados a estos jugadores se ubican más cercanos a los centroides de sentimientos negativos. De igual forma se hizo con los jugadores atípicos con un alto desempeño deportivo y ver si existe proximidad al centroide de sentimientos positivos.

Como paso final para validar el proceso con jugadores que tengan un desempeño regular, se buscó los más próximos al centroide negativo para hacer una validación sobre si su desempeño tiene opiniones negativas. De forma similar se identificó los que estén más cercanos al centroide positivo para ver si las opiniones van de acuerdo con esa tendencia. Lo esperado es que un jugador de bajo desempeño sea visto de forma negativa en las opiniones de Twitter y de forma opuesta, un jugador de alto desempeño se asociado a opiniones positivas.

2.4.3 Construcción del Modelo

La librería Word2Vec tiene varios parámetros que se pueden ver en la Tabla 4, con los que se pueden configurar los resultados (Figura 26). Para este caso se han manipulado los siguientes parámetros [11]:

Tabla 4. Parámetros de la librería Word2Vec.

Parámetro	Descripción
Archivo texto	Es el archivo que contiene las oraciones con las que se construye el modelo, en este caso serán la totalidad de los Tweets recolectados.
Archivo bin	Es el nombre con el que se va a grabar el modelo.
size	Se refiere al número de dimensiones que va a tener cada vector resultante.
window	Es el valor que define el número de palabras que van a considerarse cercanas, es decir, un rango antes o después de la palabra que está siendo evaluada.
min_count	Con este parámetro podemos definir si queremos excluir palabras que aparezcan un mínimo número de ocasiones.
sample	Configuración del muestreo descendente para las palabras frecuentes. Por defecto es 1e-3, el rango útil es (0, 1e-5).

```
word2vec.word2vec("listadfRecoleccionFiltrado.txt", 'listadfRecoleccionFiltrado.bin',
                  size=300, window=5, min_count=4, sample=1e-3, verbose=True)
```

```
Starting training using file listadfRecoleccionFiltrado.txt
Vocab size: 18389
Words in train file: 947726
Alpha: 0.002792  Progress: 89.90%  Words/thread/sec: 174.13k
```

Figura 26. Uso de la librería Word2Vec y los parámetros que serán configurados.

2.4.4 Evaluación del Modelo

Debido a que Word2Vec es un algoritmo no supervisado, se tienen que usar métodos de evaluación que sean aproximados para poder determinar si el modelo funciona.

En esta creación de espacios vectoriales se probaron varias configuraciones dentro de Word2Vec que no dieron resultados positivos, por ejemplo:

- size=400, window=10, min_count=3, sample=1e-3
- size=200, window=10, min_count=3, sample=1e-3
- size=200, window=5, min_count=5, sample=1e-3

Estas 3 configuraciones no segmentaron correctamente a los jugadores en las diferentes evaluaciones realizadas. En el caso del uso de analogías tuvimos resultados como el que se ve en la Figura 27, donde se puede ver que las palabras no guardan ninguna relación lógica entre sí para un humano.

```
def analogy(worda, wordb, wordc):
    indexes, metrics = modelTweetsProcesadosLematizado.analogy(pos=[worda, wordc], neg=[wordb], n=10)
    return modelTweetsProcesadosLematizado.generate_response(indexes, metrics).tolist()
```

```
analogy('cristiano', 'ronaldo', 'messi')
```

```
[('krafth', 0.3319901312963207),
 ('ger', 0.3319263067243812),
 ('match', 0.33005522101270657),
 ('tomas', 0.3297304365217541),
 ('injury', 0.3291951149768647),
 ('kalas', 0.32849512901110234),
 ('france', 0.3281955003383211),
 ('ferran', 0.3278068664637326),
 ('fcb', 0.3275685825441602),
 ('world', 0.3271598886200597)]
```

```
analogy('juventus', 'italy', 'barcelona')
```

```
[('former', 0.37314639695088503),
 ('set', 0.367759496382995),
 ('rejecting', 0.3612984666752034),
 ('clarke-harris', 0.3612196694284273),
 ('tiemoue', 0.3605052334746915),
 ('inter', 0.3599315594816568),
 ('birmingham', 0.35884261686963176),
 ('attracts', 0.35679040556238173),
 ('plea', 0.35644174581485255),
 ('milan', 0.35598492963592776)]
```

Figura 27. Uso de analogías que no da ningún resultado lógico

Al final, la configuración con mejores resultados a nivel de distancias de centroides y evaluación de analogías fue:

- size=300, window=5, min_count=4, sample=1e-3

Al aplicar a este modelo el uso de analogías, tenemos resultados más lógicos como los que se observan en la Figura 28.

```
analogy('cristiano', 'ronaldo', 'messi')
```

```
[('lionel', 0.33592509731247),
 ('dishes', 0.31312242063297113),
 ('wurz', 0.2986414530404642),
 ('lorenzo', 0.29094054305058026),
 ('8', 0.29064860429668915),
 ('1.63', 0.28991414291775264),
 ('7th', 0.2893181316820266),
 ('14/15', 0.28872105718758867),
 ('lucas', 0.2865914101671013),
 ('torreira', 0.2854183504086939)]
```

```
analogy('juventus', 'italy', 'barcelona')
```

```
[('nfts', 0.25023108860029725),
 ('memphis', 0.24982950002576687),
 ('aguero', 0.2481902977294183),
 ('signings', 0.24800048559029286),
 ('complement', 0.2453320330081637),
 ('depay', 0.24343216422737268),
 ('profiles', 0.24247588336612372),
 ('...', 0.2411770405919101),
 ('florenzi', 0.24049575708851118),
 ('join', 0.23729502240306025)]
```

Figura 28. Analogías retornan un resultado más lógico que relaciona las palabras entre sí.

Otro método que se usó fue el de reducir los vectores de cada palabra de 300 a solo 2 dimensiones. Si se grafican varias palabras dentro de este modelo, se podrán ver en un plano cartesiano cómo están relacionadas o cuánta cercanía tienen. Por ejemplo, en la Figura 29 podemos ver fácilmente un grupo de palabras que se relacionan entre sí como son Instagram, Youtube, photo, videos. Igual si nos vamos a los apellidos de jugadores, Messi y Ronaldo se los tiene con alta cercanía ya que continuamente se los comparaba y asociaba mutuamente, en contraste con otro jugador de nombre Onomah que no es muy conocido y por lo tanto se lo ve distante al borde del plano cartesiano.

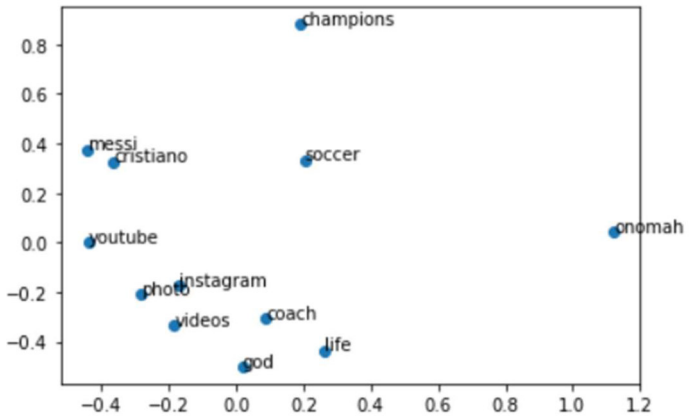


Figura 29. Proximidad de palabras dentro de un modelo reducido a 2 dimensiones.

Otra forma de evaluación que se aplicó es el de ver las palabras de los espacios vectoriales mediante una nube que muestra las palabras más repetidas dentro de cada uno de los espacios vectoriales. En la Figura 30 se muestran, a ambos lados, el modelo de palabras positivas y negativas, en donde la nube de la parte izquierda de tweets positivos tiene palabras como great, good, love, etc. En contraste, dentro de la nube de palabras de los tweets negativos tendremos bad, poor, wrong, etc. incluso insultos. No es un método muy preciso de comparación, pero nos da una idea de cómo están siendo usadas las palabras dentro de cada modelo.

Figura 30. Nubes de palabras de tweets positivos vs negativos.

Un método que se evaluó, pero no dio resultados buenos, fue el clusterizar los jugadores. El objetivo es que la librería encuentre un patrón para clasificar los jugadores en varios clústeres y determinar si los jugadores outliers de buen desempeño se ubican en un mismo clúster; y a su vez, si es diferente al clúster donde están ubicados los jugadores outliers de mal desempeño.

Por ejemplo, en la Figura 31, en color verde están encerrados los outliers de buen desempeño y en rojo los de mal desempeño. Pero en cuanto a los resultados, vemos que la mayoría de los jugadores se clasificaron en el clúster 0 independientemente de su desempeño. Después de varios ensayos, no se pudo clasificar a los jugadores de forma en clústers que sean coherentes con el desempeño de los jugadores.

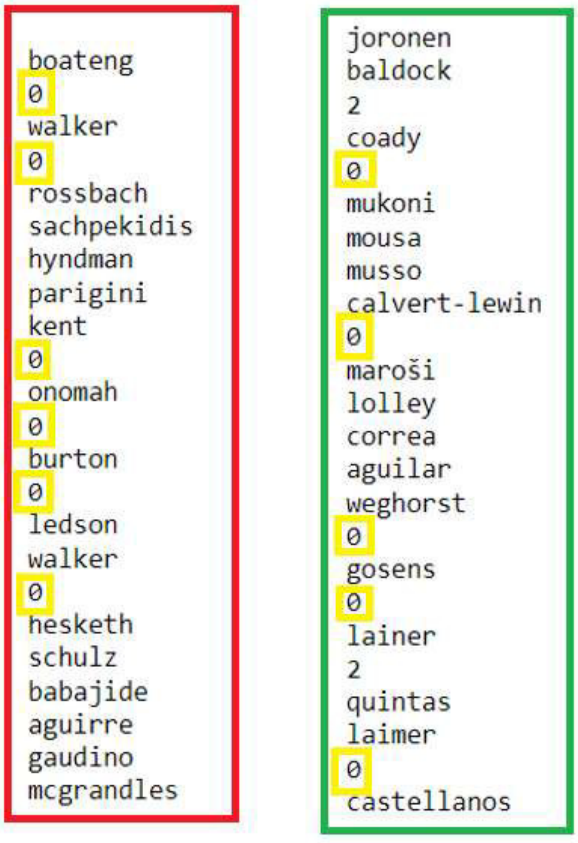


Figura 31. Uso de clústeres para clasificar a jugadores sin ningún resultado concluyente

3 CAPÍTULO III: RESULTADOS

3.1 Evaluación de Resultados

Tal como se estableció en el plan de pruebas, se buscó clasificar los tweets de forma que tengamos sentimientos positivos, negativos o neutrales. De estos espacios vectoriales se calculó las distancias de cada uno de los jugadores que en un inicio fueron clasificados de acuerdo con su desempeño, de forma que se pueda confirmar si las anomalías que hemos encontrado guardan relación con el desempeño deportivo y financiero de los deportistas.

Por lo tanto, el primer paso realizado fue calcular un centroide, o media estadística entre todos los valores de los vectores, de cada uno de los espacios vectoriales: positivo, negativo y neutral. Esto se realizó mediante una función que obtiene este valor al sumar todas las 300 dimensiones de cada palabra vectorizada y dividirla para el número de elementos, la función aplicada a los 3 espacios vectoriales se lo puede ver en la Figura 32.

```
centroideNegativo = np.mean(modelTweetsProcesadosLematizadoNegativo.vectors[:,:], axis=0)
centroidePositivo = np.mean(modelTweetsProcesadosLematizadoPositivo.vectors[:,:], axis=0)
centroideNeutral = np.mean(modelTweetsProcesadosLematizadoNeutral.vectors[:,:], axis=0)
```

Figura 32. Función para el cálculo de la media estadística o centroide de un grupo de vectores.

Por ejemplo, si se despliega el valor del centroide negativo, se observará un vector con 300 dimensiones que será el valor medio de todas las palabras contenidas en el modelo de tweets de sentimientos negativos. Este vector con sus dimensiones se lo ve parcialmente en la Figura 33.

```
print(centroideNegativo)
```

```
[-1.21720431e-02  5.53375049e-02  6.85056793e-02  4.01614854e-02
  1.09377807e-01 -1.50870682e-02  5.06481758e-03 -2.70780091e-02
 -6.29978012e-03  3.24952993e-02  8.65173275e-02 -2.93739079e-02
 -9.52127038e-02 -6.16930705e-02  6.26155410e-02 -3.58526846e-02
  6.81564688e-02 -2.68072999e-02 -7.48219152e-02  4.41426924e-02
 -5.07070393e-02 -4.81018138e-02  5.14707054e-02 -3.63545312e-02
 -2.52217150e-02  3.11831830e-04  8.50692756e-03 -3.99309548e-02
 -3.14459358e-05 -9.80715875e-02  9.26288231e-03 -7.75193260e-03
 -1.96853247e-02 -1.68852521e-02 -5.20391253e-02  1.45391044e-01
 -4.06074143e-02  9.86268595e-02 -1.59266888e-02 -6.56441257e-03
```

Figura 33. Centroide de los tweets con sentimientos negativos.

Ya calculados los centroides de cada modelo basado en sentimientos, se hizo una primera verificación que consistió en evaluar la distancia entre cada uno de esos centroides para

verificar que no se superponen entre sí. Para este cálculo se usó la función de similitud Cosine para obtener un valor de similitud. Coseno es una medida que existe entre dos vectores al calcular el ángulo coseno que existe entre los mismos. Si el valor es igual a 1, significaría que ambos vectores apuntan al mismo lugar, pero si su valor se aproxima más al -1 esto da a entender que los dos vectores apuntan a lugares totalmente diferentes entre sí [20].

En la Figura 34 se muestra el resultado del cálculo de función de similitud Coseno entre todos los centroides. El que más nos interesó es la diferencia entre el centroide de Tweets positivos versus el centroide de Tweets negativos, en este caso su valor es de -0.057 que corresponde a una relación ortogonal entre ambos vectores, por lo tanto, los centroides de estos espacios vectoriales se encuentran separados entre sí.

```
print(float("{0:.3f}".format(1 - spatial.distance.cosine(centroideNegativo, centroidePositivo))))
print(float("{0:.3f}".format(1 - spatial.distance.cosine(centroideNegativo, centroideNeutral))))
print(float("{0:.3f}".format(1 - spatial.distance.cosine(centroideNeutral, centroidePositivo))))
```

```
-0.057
0.032
-0.062
```

Figura 34. Cálculo de la Similitud Coseno entre los centroides de los espacios vectoriales.

Para verificaciones individuales, se procedió a filtrar los datos de uno de los outliers en cada fase, en este caso se decidió usar al primero de la lista de jugadores outliers. En la Figura 35 podemos ver que el primer nombre que aparece en este filtrado es Hiram Boateng.

```
dfRecoleccionFiltrado.loc[dfRecoleccionFiltrado['OutliersMalos'] == 1]
```

	long_name	sofifa_id	text	created_at	Subjetividad	Polaridad	text_limpio	Analisis	long_name_procesado	text_limpio_procesado	Ou
33201	Hiram Boateng	212782	@cravopafc In all seriousness I genuinely miss...	Sun Jun 27 13:44:38 +0000 2021	0.500000	0.400000	seriousness genuinely miss matthew matty kenne...	Positivo	['hiram', 'boateng']	['seriousness', 'genuinely', 'miss', 'matthew'...	
34818	Brad Walker	213519	Congrats to 2021 4ball Open Champs Justin Ahle...	Mon Jun 28 16:03:20 +0000 2021	0.500000	0.000000	congrats 2021 4ball open champ justin ahlers b...	Neutral	['brad', 'walker']	['congrats', '2021', '4ball', 'open', 'champ',...	
34819	Brad Walker	213519	RT @BobbyAllen1a: Yes very glad for Shark raci...	Mon Jun 28 15:01:42 +0000 2021	0.625000	0.662500	yes glad shark racing good win jackson nationa...	Positivo	['brad', 'walker']	['yes', 'glad', 'shark', 'racing', 'good', 'wi...	

Figura 35. Listado de Tweets recolectados de los outliers de bajo desempeño.

Luego, con la referencia de este jugador, lo que se hizo es tomar la información almacenada de un Tweet recolectado en el cual se menciona al jugador y se enfocó en el texto que se encuentra en el campo text, porque es en este campo en específico en donde se graba el texto que el usuario ha escrito dentro de la red social Twitter, tal como se ve en la Figura 36.

```
dfRecoleccionFiltrado.loc[dfRecoleccionFiltrado['OutliersMalos'] == 1].iloc[0, [0, 2, 9, 14]]

long_name                                                                   Hiram Boateng
text                   @cravopafc In all seriousness I genuinely miss Matthews, Matty Kennedy and Hiram Boateng
```

Figura 36. Tweet recolectado donde se menciona al jugador Hiram Boateng.

Al texto que se tiene originalmente, se le aplicó el procesamiento mencionado previamente dentro de la sección de 2.3.4 Integración de Datos. Este proceso consiste en varios pasos, entre ellos el de remover URLs, stop words, etiquetas, etc. El texto procesado se lo almacenó en otra fila que hemos denominado text_limpio_procesado (Figura 37).

```
dfRecoleccionFiltrado.loc[dfRecoleccionFiltrado['OutliersMalos'] == 1].iloc[0, [0, 2, 9, 14]]

long_name                                                                   Hiram Boateng
text                   @cravopafc In all seriousness I genuinely miss Matthews, Matty Kennedy and Hiram Boateng
text_limpio_procesado          [seriousness, genuinely, miss, matthew, matty, kennedy, hiram, boateng]
```

Figura 37. Texto del tweet que ha sido procesado.

Y como último paso, se eliminaron las palabras comunes que existen entre los espacios vectoriales de Tweets Negativos y Tweets Positivos, tal como se mencionó en la sección 2.3.5 Formateo de los Datos. Tras remover palabras comunes nos quedamos con lo que se puede ver en la fila text_limpio_sinComunes_procesado de la Figura 38.

```
dfRecoleccionFiltrado.loc[dfRecoleccionFiltrado['OutliersMalos'] == 1].iloc[0, [0, 2, 9, 14]]

long_name                                                                        Hiram Boateng
text                            @cravopafc In all seriousness I genuinely miss Matthews, Matty Kennedy and Hiram Boateng
text_limpio_procesado                   [seriousness, genuinely, miss, matthew, matty, kennedy, hiram, boateng]
text_limpio_sinComunes_procesado                              [seriousness, matthew, kennedy, hiram, boateng]
```

Figura 38. Texto del tweet del cual se ha removido palabras en común entre espacios vectoriales.

Por lo tanto, del tweet original que tenía 13 palabras, nos quedamos tan solo con 5 palabras luego de aplicar todas las operaciones antes mencionadas.

Ya con este resultado, el siguiente paso fue tomar esas 5 palabras, buscarlas dentro del modelo y sacar el valor promedio de ese tweet, de esta forma cada Tweet tendrá un solo vector característico que lo identifique. Ya teniendo la función que devuelve el valor medio de las palabras de un Tweet, se procedió a calcular la distancia de ese valor promedio del Tweet en comparación con los centroides de los espacios vectoriales negativo y positivo.

Como se mencionó inicialmente, se trabajó con un total de 100.272 Tweets que ya fueron formateados de la forma que se ha descrito en los pasos anteriores. Al separar los Tweets de acuerdo con su pertenencia a jugadores outliers buenos y malos, tenemos las siguientes

cantidades: 2267 Tweets de outliers buenos y 860 Tweets de outliers malos; el resto de los Tweets son los que pertenecen a los jugadores de desempeño típico.

Si graficamos estos datos de forma porcentual, vemos que el 97% de los tweets recolectados son de jugadores típicos y solo 3% pertenecen a los outliers, siendo un 2% outliers buenos y 1% outliers malos. Esto se lo puede ver en la Figura 39.

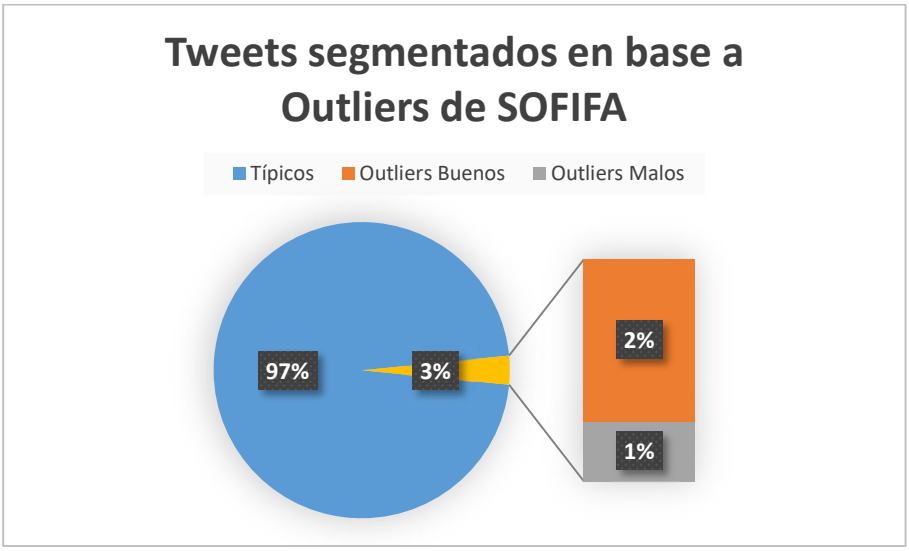


Figura 39. Tweets clasificados de acuerdo con jugadores de desempeño típico y outliers.

Todo lo mencionando anteriormente (cálculo de funciones de vectores promedio, centroides, textos procesados, etc.) se implementaron en una función que se definió en Python encargada de usar todas estas piezas en una sola ejecución.

En la Figura 40 tenemos una representación de lo realizado por la función que se detalla en el Anexo 5, de la cual los principales pasos son:

1. Identificar el jugador a analizar.
2. Extraer información de Twitter.
3. Obtener tweets del jugador.
4. Procesar el Tweet para limpiarlo
5. Obtener un conjunto de palabras para pasarlas por el modelo.
6. Buscar las palabras en el modelo y su posición vectorial.
7. Obtener un vector promedio que identifique al jugador.
8. Calcular la distancia a los centroides mediante la similitud coseno.
9. Identificar a cuál centroide tiene tendencia el jugador.
10. Evaluar los resultados.

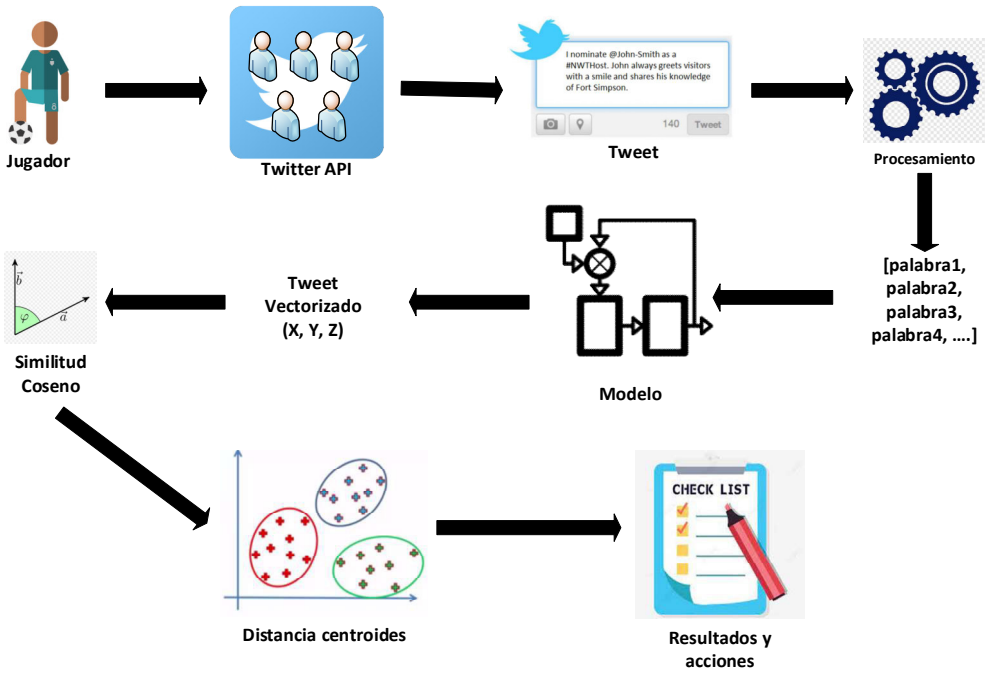
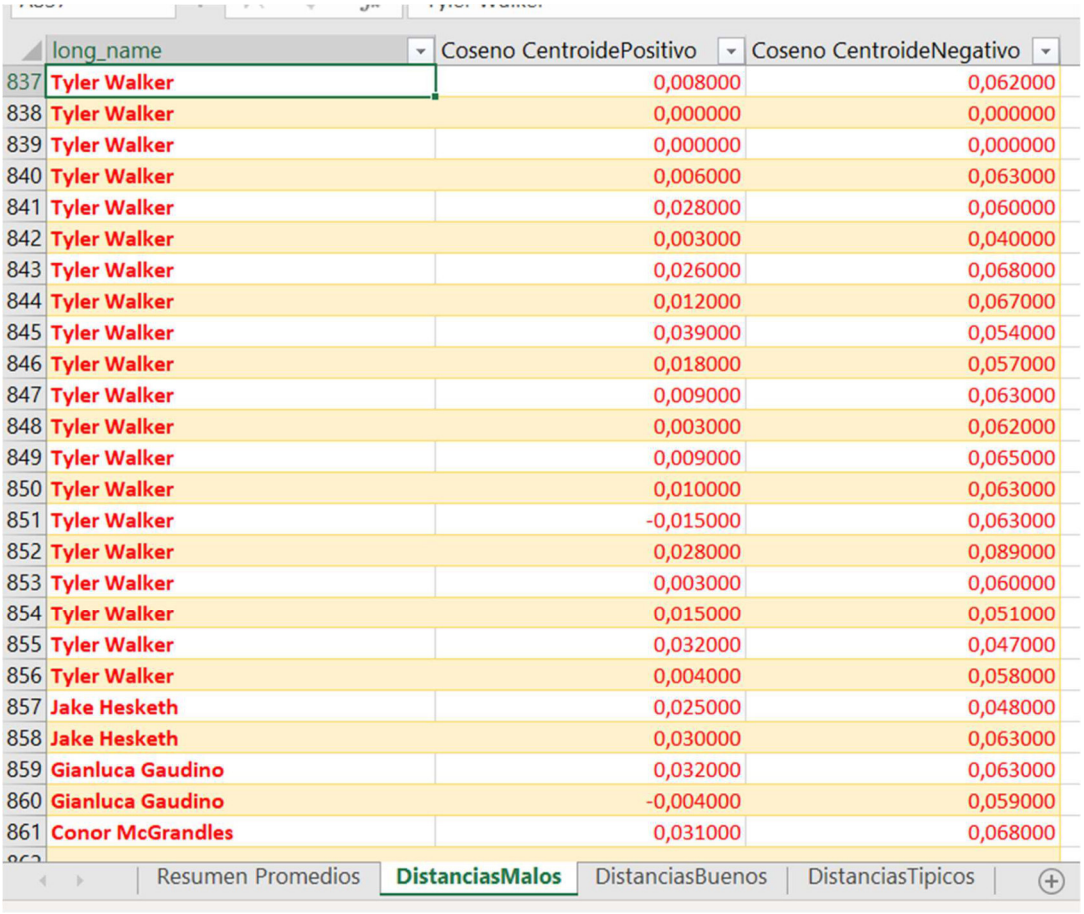


Figura 40. Pasos de la función que calcula la distancia del texto de un Tweet a cada centroide.

Cada uno de los resultados de las distancias calculadas con la función anteriormente detallada se almacenó en 3 archivos CSV que integraremos en un solo archivo de Excel. En cada una de las hojas tendremos primero el nombre del jugador, en la segunda y tercera columna el valor de la Similitud Coseno del Tweet de ese jugador a los centroides de sentimientos positivos y negativos respectivamente. Cada hoja será separada de acuerdo con los jugadores outliers buenos, jugadores outliers malos y los jugadores de desempeño típicos. Un ejemplo de esto se puede ver en la Figura 41.



	long_name	Coseno CentroidePositivo	Coseno CentroideNegativo
837	Tyler Walker	0,008000	0,062000
838	Tyler Walker	0,000000	0,000000
839	Tyler Walker	0,000000	0,000000
840	Tyler Walker	0,006000	0,063000
841	Tyler Walker	0,028000	0,060000
842	Tyler Walker	0,003000	0,040000
843	Tyler Walker	0,026000	0,068000
844	Tyler Walker	0,012000	0,067000
845	Tyler Walker	0,039000	0,054000
846	Tyler Walker	0,018000	0,057000
847	Tyler Walker	0,009000	0,063000
848	Tyler Walker	0,003000	0,062000
849	Tyler Walker	0,009000	0,065000
850	Tyler Walker	0,010000	0,063000
851	Tyler Walker	-0,015000	0,063000
852	Tyler Walker	0,028000	0,089000
853	Tyler Walker	0,003000	0,060000
854	Tyler Walker	0,015000	0,051000
855	Tyler Walker	0,032000	0,047000
856	Tyler Walker	0,004000	0,058000
857	Jake Hesketh	0,025000	0,048000
858	Jake Hesketh	0,030000	0,063000
859	Gianluca Gaudino	0,032000	0,063000
860	Gianluca Gaudino	-0,004000	0,059000
861	Conor McGrandles	0,031000	0,068000

Resumen Promedios | DistanciasMalos | DistanciasBuenos | DistanciasTipicos

Figura 41. Valor de la Similitud Coseno de los vectores de cada Tweet hacia los centroides positivo y negativo.

Dentro del mismo archivo de Excel, se procedió a aplicar la función PROMEDIO de los valores de la segunda y tercera columna por jugador. Se obtiene un promedio de todos los Tweets de los jugadores outliers buenos y de los malos respecto a los centroides positivo y negativo, y esas cifras se las refleja en la Tabla 5.

Tabla 5. Promedio del valor de la Similitud Coseno de los Tweets de los jugadores outliers respecto a los centroides.

Tipos	**Coseno Centroide Positivo**	**Coseno Centroide Negativo**
Outliers Buenos	0,040430	0,036786
Outliers Malos	0,025419	0,057631

Como se puede ver en la Tabla 5, los jugadores outliers buenos tienen un valor de mayor

cercanía con referencia al centroide de sentimientos positivos. En contraste, los jugadores outliers malos tienen mayor cercanía al centroide negativo.

A pesar de no ser un valor muy diferenciado entre ambas distancias, se confirma lo que inicialmente se había esperado: que un jugador de desempeño bajo se lo empieza a percibir también de forma negativa dentro de la red social Twitter y, por otro lado, esto sucede de forma similar con los jugadores de desempeño bueno, a los cuales se los percibe de forma positiva dentro de Twitter.

3.2 Revisión del Proceso

Como ejemplo del uso de la función y su distancia al centroide, vamos a mostrar el mismo resultado, pero solo del jugador Hiram Boateng que se lo ha mencionado en pasos anteriores. Se puede ver en la Tabla 6 que este jugador con desempeño malo tiene una similitud próxima al Centroide Negativo de análisis de sentimientos.

Tabla 6. Distancia del jugador outlier malo llamado Hiram Boateng.

long_name	Coseno Centroide Positivo	Coseno Centroide Negativo
Hiram Boateng	0,009000	0,041000

De igual forma ocurre con un jugador de desempeño bueno Jesse Joronen, sus tweets tienen mayor similitud al Centroide Positivo tal como se ve en la Tabla 7, por lo que se puede decir que es de bajo riesgo.

Tabla 7. Distancia del jugador outlier bueno llamado Jesse Joronen.

long_name	Coseno Centroide Positivo	Coseno Centroide Negativo
Jesse Joronen	0,048000	0,036000

Si aplicamos el mismo concepto con un jugador típico que esté más aproximado al centroide negativo, deberíamos ver opiniones negativas sobre su persona, como es el caso de Tom Nichols que en la Tabla 8 podemos ver que tiene más tendencia a un centroide negativo, por lo tanto, es un jugador de alto riesgo.

Tabla 8. Distancia del jugador llamado Jesse Tom Nichols hacia los centroides

long_name	Coseno Centroide Positivo	Coseno Centroide Negativo
Tom Nichols	-0,034000	0,122000

Si buscamos información en Google sobre este jugador, podemos ver que hinchas de su antiguo equipo lo reprochaban por su desempeño, tal como se menciona en el periódico de Bristol “Los abucheos y las críticas en las redes sociales no han impedido que Tom Nichols recuerde con cariño su paso por el Bristol Rover” [21].

De igual forma, si vemos sus datos dentro de Transfermarkt, en la Figura 42 se puede ver que su costo de transferencia no ha variado en 3 temporadas.

TRANSFER HISTORY

Season	Date	Left	Joined	MV	Fee
20/21	Sep 7, 2020	Bristol Rovers	Crawley Town	€300Th.	free transfer
19/20	May 31, 2020	Cheltenham	Bristol Rovers	€300Th.	End of loan
19/20	Jan 31, 2020	Bristol Rovers	Cheltenham	€300Th.	loan transfer
17/18	Jul 17, 2017	Peterborough	Bristol Rovers	€300Th.	?
15/16	Feb 1, 2016	Exeter City	Peterborough	€50Th.	€330Th.
12/13	Apr 18, 2013	Dorchester	Exeter City	€50Th.	End of loan
12/13	Mar 18, 2013	Exeter City	Dorchester	€50Th.	loan transfer
12/13	Sep 1, 2012	Hereford Utd.	Exeter City	€50Th.	End of loan
12/13	Aug 1, 2012	Exeter City	Hereford Utd.	€50Th.	loan transfer
11/12	Jul 1, 2011	Exeter U18	Exeter City	-	-

Figura 42. Historial futbolístico del jugador Tom Nichols.

Como un paso final, se procedió a aplicar el proceso especificado en la Figura 40 sobre un solo jugador, recolectando Tweets actualizados y viendo cómo se aproxima a cada centroide. El código referente q este proceso se lo puede ver en el Anexo 5. Para este ejemplo se usó al jugador Erling Haaland que es uno de los jugadores del fútbol alemán, del cual se obtuvo al final del proceso un total de 176 tweets con una proximidad al Centroide Positivo con un valor de 0.037 en color amarillo, 0.045 al Centroide Negativo en verde y en azul al Centroide Neutral con un valor de -.0.058, tal como se ve en la Figura 43.

	Subjetividad	Polaridad	Coseno_CentroidePositivo_Tweets	Coseno_CentroideNegativo_Tweets	Coseno_CentroideNeutral_Tweets
count	176.000000	176.000000	176.000000	176.000000	176.000000
mean	0.331794	0.098938	0.036722	0.045051	-0.057909

Figura 43. Cercanía al centroide positivo del jugador Erling Haaland.

Si se aplica el mismo proceso usando al jugador ecuatoriano Gonzalo Plata que tuvo un accidente en a inicios del mes de diciembre del 2021, se puede ver en la Figura 44 un comportamiento de mayor proximidad al Centroide Negativo con un valor de 0.057 igual en color verde, dando a entender que es de mayor riesgo que el caso mencionado en la Figura 43.

	Subjetividad	Polaridad	Coseno_CentroidePositivo_Tweets	Coseno_CentroideNegativo_Tweets	Coseno_CentroideNeutral_Tweets
count	27.000000	27.000000	27.000000	27.000000	27.000000
mean	0.425428	0.102167	0.021407	0.057000	-0.069407

Figura 44. Cercanía al centroide positivo del jugador Gonzalo Plata.

3.3 Determinar los Próximos Pasos

Entre los siguientes pasos para continuar implementando y mejorando el modelo están:

- Recolectar información de Twitter de forma periódica o establecer un plan en el cual se recoja más Tweets sobre jugadores para ir refinando más el modelo.
- Incluir tweets en español, para que se pueda hacer un análisis más local, ya que actualmente el modelo solo tiene texto en inglés.
- Establecer mejoras a nivel de la implementación para que sea un proceso más mantenible y automatizado.
- Crear un proceso de categorización, de forma que se pueda dar más peso a opiniones que vengan de prensa especializada.

3.4 Implementación

En el caso de esta tesis, en el alcance se definió que tan solo se llegaba a la creación del modelo y a realizar pruebas sobre el mismo, ninguna interfaz o aplicación fue definida como parte del proyecto. De igual forma, las actualizaciones no son automáticas ni tampoco se puede monitorear cambios sobre la misma.

Sin embargo, se ha procedido a especificar fases de implementación que pueda extender lo expuesto anteriormente para que las funciones creadas puedan ser utilizadas de forma que el usuario final pueda ver el resultado con una interfaz más amigable y de fácil uso.

3.4.1 Planear la Implementación

Para aplicaciones creadas en Python existe un framework liviano denominado Flask[6], el mismo que permite la creación de sitios web livianos y de fácil implementación. El tipo de licencia de Flask es de uso libre y existe suficiente documentación que explica la forma en que se deben hacer llamadas a las funciones creadas durante el presente proyecto de tesis.

3.4.2 Planear la Monitorización y Mantenimiento

Lo principal en este punto es establecer un plan que permita la actualización del modelo, de forma que se siga alimentando con información y tweets de forma continua. Para esto lo primero debe ser el contar con una licencia de Twitter comercial, para poder recolectar una mayor cantidad de información. Lo segundo es establecer la frecuencia en que se van a realizar las recolecciones, de forma que coincidan con fechas en que haya un mayor flujo de información, en el caso del fútbol de Europa existen las competiciones locales que son complementadas con las competiciones entre países, por lo tanto, existe un flujo constante de datos; pero se debería evitar hacer recolección en las épocas de vacaciones o cierres de temporada, en donde generalmente solo existe información que se basa más en rumores. El objetivo final de todo esto es hacer que el modelo siga mejorando sus predicciones y pueda ser evaluado de forma constante.

3.4.3 Producir el Informe Final

En base a la aplicación que se pueda desarrollar usando Flask, se debería crear una interfaz que sea lo suficientemente sencilla como para dar una respuesta fácil de interpretar y que a su vez pueda igual mostrar información más detallada en caso de que sea requerida. Por ejemplo, usar un sistema de semaforización en conjunto con una nube de palabras, para que se pueda ver como se percibe al jugador y con que términos se lo asocia de forma más común.

Esto creará reportes o dashboards fáciles de interpretar y visualizar para las personas que vayan a interactuar con el modelo creado.

3.4.4 Revisar el Proyecto

Cada cierto tiempo se debería evaluar a ambos grupos de jugadores, tanto los de buen desempeño como los de bajo desempeño, para poder corroborar que el modelo y su comportamiento siguen acordes a lo ya analizado. Si esto no se produce, empezar a evaluar de forma unitaria las anomalías para poder refinar el modelo hasta mejorarlo nuevamente y que exista una veracidad de sus resultados.

[6] https://flask.palletsprojects.com/en/2.0.x/

4 CAPÍTULO IV: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

- El procesamiento inicial de los datos de EA Sports se facilitó al poder usar un repositorio de Kaggle que ya los tenía recolectados y estructurados, en este punto el reto fue entender los datos, interpretarlos y saber que columnas eran las relevantes para el estudio. Luego de esa etapa de identificación, fue más sencillo crear un ETL para cada año y de esa forma tenemos una tabla para cada archivo de cada año. El poder manipular estos datos en Python después de importarlos, hizo que podamos trabajar de mejor forma los cálculos que se vieron en las fases iniciales de esta tesis y poder separarlos en diferentes sets de datos que incluso favorecieron en el desempeño del código.
- En el caso de la fase de recolección de Twitter, se deben tomar en cuenta las restricciones del API de desarrollo. Pero ya solventado esto con el uso de diferentes librerías de Python, lo siguiente fue usar correctamente los datos recolectados, ya que es un JSON que tiene mucha meta data que no es relevante para el estudio.
- Se debió hacer varias pruebas hasta tener un modelo funcional que responda de manera similar durante las fases de implementación, como se vio en la etapa de modelado tenemos varias consideraciones que afectaron directamente en la cantidad de información que si se podía utilizar y que daban los resultados esperados.
- Uno de los puntos clave fue el remover las palabras comunes entre espacios vectoriales positivos y negativos, esto debido a que existían muchas palabras que aparecían en ambos contextos. La distancia entre centroides mejoró sustancialmente al quitar estas palabras duplicadas y tener palabras únicas dentro de cada uno de los espacios vectoriales.
- Al ser el fútbol el deporte más conocido a nivel mundial, el proceso realizado en esta tesis permite que sea un tema de fácil entendimiento para la gran mayoría de personas. De esta forma, se pueden extrapolar los procesos realizados en cada fase a diferentes ámbitos de negocios. Por ejemplo, si una compañía de un producto X quiere saber cómo éste se viene desempeñando, podremos ver el número de ventas de ese producto año a año y compararlo con la percepción en redes sociales, así se pueden tomar decisiones en caso de ver un estancamiento en la percepción del producto.

4.2 Recomendaciones

- El modelo se puede refinar mucho más si se recoge información de todos los jugadores en diferentes épocas del año, ya que siempre afectará las opiniones de un

jugador a inicio de temporada versus las opiniones a final del año, en especial si se contrasta con el desempeño del equipo en general. Por lo tanto, es importante establecer un plan en el cual se recoja más Tweets sobre los jugadores para ir refinando más el modelo y que este proceso sea de forma periódica.

- Otro punto que influirá directamente en la recolección de Tweets es el poder contar con una versión paga de Twitter, para que las limitaciones sean las menores posibles al momento de recoger información dentro de esta red social. Para este estudio se usó una versión gratuita y esto afectaba en el tiempo y cantidad de información que se podía recopilar.
- Actualmente el modelo se enfoca en el idioma inglés, por lo que es muy difícil el poder usarlo para el ámbito local. Que el modelo sea multilenguaje ayudaría a que se pueda hacer un análisis local.
- Se deberían establecer los rangos en que las distancias de un jugador a un centroide negativo o positivo sea realmente relevante, esto debido a que los rangos tan cercanos entre sí no son fáciles de interpretar y para un usuario común serían difíciles de identificar. Es por lo que una próxima fase debería incluir una interfaz que entregue de forma más amigable y entendible los resultados.
- La información recolectada de Twitter es evaluada de la misma forma al momento de crear los espacios vectoriales, es decir, no existe un ranking de pesos a las opiniones entregadas. Si una persona X opina algo negativo sobre un jugador, dentro del modelo, tendrá el mismo peso que la opinión de un periodista de prensa especializada. Por todo se recomienda realizar una categorización que permita dar más peso a opiniones que vengan de prensa especializada por encima de las emitidas por cualquier persona.
- De igual forma se debería excluir los Tweets de personas que tan solo retuitean el mismo texto cambiado un par de palabras. Estos casos se vieron de forma manual y no se pudieron corregir ya que no existe en patrón que permita identificarlos.

5 BIBLIOGRAFÍA

[1] J. Han y M. Kamber, *Data mining: concepts and techniques*, 3rd ed. Burlington, MA: Elsevier, 2012.

[2] B. Schoenfeld, «How Data (and Some Breathtaking Soccer) Brought Liverpool to the Cusp of Glory», *The New York Times*, may 22, 2019. Accedido: oct. 10, 2021. [En línea]. Disponible en: https://www.nytimes.com/2019/05/22/magazine/soccer-data-liverpool.html

[3] «El éxito de las estadísticas aplicadas al fútbol | Gestión». https://www.revistagestion.ec/estrategia-analisis/el-exito-de-las-estadisticas-aplicadas-al-futbol (accedido oct. 10, 2021).

[4] «Football Money League 2019». https://www2.deloitte.com/ec/es/pages/consumer-business/articles/football-money-league-2019.html (accedido sep. 22, 2020).

[5] «¿Qué es la inteligencia artificial (IA)? | Oracle México». https://www.oracle.com/mx/artificial-intelligence/what-is-ai/ (accedido nov. 20, 2021).

[6] «¿Qué es la inteligencia artificial (IA)?» https://www.oracle.com/mx/artificial-intelligence/what-is-ai/ (accedido oct. 19, 2021).

[7] A. C. Müller y S. Guido, *Introduction to machine learning with Python: a guide for data scientists*, First edition. Sebastopol, CA: O'Reilly Media, Inc, 2016.

[8] L. Igual y S. Seguí, *Introduction to Data Science*. Cham: Springer International Publishing, 2017. doi: 10.1007/978-3-319-50017-1.

[9] «Aprendizaje Supervisado y No Supervisado - Fernando Sancho Caparrini». http://www.cs.us.es/~fsancho/?e=77 (accedido oct. 19, 2021).

[10] P. Shah, «My Absolute Go-To for Sentiment Analysis — TextBlob.», *Medium*, nov. 06, 2020. https://towardsdatascience.com/my-absolute-go-to-for-sentiment-analysis-textblob-3ac3a11d524 (accedido oct. 26, 2021).

[11] Y. Goldberg y O. Levy, «word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method», *arXiv:1402.3722 [cs, stat]*, feb. 2014, Accedido: dic. 30, 2021. [En línea]. Disponible en: http://arxiv.org/abs/1402.3722

[12] L. Ma y Y. Zhang, «Using Word2Vec to process big text data», en *2015 IEEE International Conference on Big Data (Big Data)*, Santa Clara, CA, USA, oct. 2015, pp. 2895-2897. doi: 10.1109/BigData.2015.7364114.

[13] «CRISP-DM», *Data Science Process Alliance*. https://www.datascience-pm.com/crisp-dm-2/ (accedido oct. 16, 2021).

[14] Y. Yu y X. Wang, «World Cup 2014 in the Twitter World: A big data analysis of sentiments in U.S. sports fans' tweets», *Computers in Human Behavior*, vol. 48, pp. 392-400, jul. 2015, doi: 10.1016/j.chb.2015.01.075.

[15] M. Bagić Babac y V. Podobnik, «A sentiment analysis of who participates, how and why, at social media sport websites: How differently men and women write about football», *Online Information Review*, vol. 40, n.º 6, pp. 814-833, ene. 2016, doi: 10.1108/OIR-02-2016-0050.

[16] A. Gruettner, M. Vitisvorakarn, T. Wambsganss, R. Rietsche, y A. Back, «The New Window to Athletes' Soul – What Social Media Tells Us About Athletes' Performances», ene. 2020. doi: 10.24251/HICSS.2020.303.

[17] «Measuring Political Sentiment on Twitter: Factor Optimal Design for Multinomial Inverse Regression», *ResearchGate*. https://www.researchgate.net/publication/227172192_Measuring_Political_Sentiment_on_Twitter_Factor_Optimal_Design_for_Multinomial_Inverse_Regression (accedido sep. 22, 2020).

[18] F. Staff, «FIFA 22: Player ratings explained», *fourfourtwo.com*, oct. 07, 2021. https://www.fourfourtwo.com/features/fifa-22-player-ratings-explained-chemistry-fut-ultimate-team-pace-passing-speed-card (accedido oct. 11, 2021).

[19] «Rate limits». https://developer.twitter.com/en/docs/twitter-api/rate-limits (accedido oct. 25, 2021).

[20] «Similitud coseno», *Wikipedia, la enciclopedia libre*. sep. 10, 2019. Accedido: oct. 31, 2021. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Similitud_coseno&oldid=119153799
[21] S. Frost, «Nichols reflects on mental struggles, Clarke-Harris' kindness and Crawley move», *BristolLive*, dic. 22, 2020. https://www.bristolpost.co.uk/sport/football/football-news/nichols-reflects-mental-struggles-bristol-4820887 (accedido nov. 01, 2021).

ANEXO 1. ESTRUCTURA DE LA TABLA DENTRO DE MYSQL

```
CREATE TABLE `sofifa_2015` (
  `sofifa_id` bigint(20) NOT NULL,
  `player_url` varchar(200) DEFAULT NULL,
  `short_name` varchar(50) DEFAULT NULL,
  `long_name` varchar(200) DEFAULT NULL,
  `age` bigint(20) DEFAULT NULL,
  `dob` datetime DEFAULT NULL,
  `height_cm` bigint(20) DEFAULT NULL,
  `weight_kg` bigint(20) DEFAULT NULL,
  `nationality` varchar(100) DEFAULT NULL,
  `club_name` varchar(100) DEFAULT NULL,
  `league_name` varchar(100) DEFAULT NULL,
  `league_rank` bigint(50) DEFAULT NULL,
  `overall` bigint(20) DEFAULT NULL,
  `potential` bigint(20) DEFAULT NULL,
  `value_eur` bigint(20) DEFAULT NULL,
  `wage_eur` bigint(20) DEFAULT NULL,
  `player_positions` varchar(30) DEFAULT NULL,
  `preferred_foot` varchar(5) DEFAULT NULL,
  `international_reputation` bigint(20) DEFAULT NULL,
  `weak_foot` bigint(20) DEFAULT NULL,
  `skill_moves` bigint(20) DEFAULT NULL,
  `work_rate` varchar(30) DEFAULT NULL,
  `body_type` varchar(30) DEFAULT NULL,
  `real_face` varchar(3) DEFAULT NULL,
  `release_clause_eur` bigint(50) DEFAULT NULL,
  `player_tags` varchar(250) DEFAULT NULL,
  `team_position` varchar(3) DEFAULT NULL,
  `team_jersey_number` bigint(20) DEFAULT NULL,
  `loaned_from` varchar(50) DEFAULT NULL,
  `joined` datetime DEFAULT NULL,
  `contract_valid_until` bigint(20) DEFAULT NULL,
  `nation_position` varchar(3) DEFAULT NULL,
  `nation_jersey_number` bigint(20) DEFAULT NULL,
  `pace` bigint(20) DEFAULT NULL,
  `shooting` bigint(20) DEFAULT NULL,
  `passing` bigint(20) DEFAULT NULL,
  `dribbling` bigint(20) DEFAULT NULL,
  `defending` bigint(20) DEFAULT NULL,
  `physic` bigint(20) DEFAULT NULL,
  `gk_diving` bigint(20) DEFAULT NULL,
  `gk_handling` bigint(20) DEFAULT NULL,
  `gk_kicking` bigint(20) DEFAULT NULL,
  `gk_reflexes` bigint(20) DEFAULT NULL,
  `gk_speed` bigint(20) DEFAULT NULL,
  `gk_positioning` bigint(20) DEFAULT NULL,
  `player_traits` varchar(200) DEFAULT NULL,
  `attacking_crossing` varchar(30) DEFAULT NULL,
  `attacking_finishing` varchar(30) DEFAULT NULL,
  `attacking_heading_accuracy` varchar(30) DEFAULT NULL,
  `attacking_short_passing` varchar(30) DEFAULT NULL,
  `attacking_volleys` varchar(30) DEFAULT NULL,
  `skill_dribbling` varchar(30) DEFAULT NULL,
  `skill_curve` varchar(30) DEFAULT NULL,
  `skill_fk_accuracy` varchar(30) DEFAULT NULL,
  `skill_long_passing` varchar(30) DEFAULT NULL,
  `skill_ball_control` varchar(30) DEFAULT NULL,
  `movement_acceleration` varchar(30) DEFAULT NULL,
```

```
  `movement_sprint_speed` varchar(30) DEFAULT NULL,
  `movement_agility` varchar(30) DEFAULT NULL,
  `movement_reactions` varchar(30) DEFAULT NULL,
  `movement_balance` varchar(30) DEFAULT NULL,
  `power_shot_power` varchar(30) DEFAULT NULL,
  `power_jumping` varchar(30) DEFAULT NULL,
  `power_stamina` varchar(30) DEFAULT NULL,
  `power_strength` varchar(30) DEFAULT NULL,
  `power_long_shots` varchar(30) DEFAULT NULL,
  `mentality_aggression` varchar(30) DEFAULT NULL,
  `mentality_interceptions` varchar(30) DEFAULT NULL,
  `mentality_positioning` varchar(30) DEFAULT NULL,
  `mentality_vision` varchar(30) DEFAULT NULL,
  `mentality_penalties` varchar(30) DEFAULT NULL,
  `mentality_composure` varchar(30) DEFAULT NULL,
  `defending_marking` varchar(30) DEFAULT NULL,
  `defending_standing_tackle` varchar(30) DEFAULT NULL,
  `defending_sliding_tackle` varchar(30) DEFAULT NULL,
  `goalkeeping_diving` varchar(30) DEFAULT NULL,
  `goalkeeping_handling` varchar(30) DEFAULT NULL,
  `goalkeeping_kicking` varchar(30) DEFAULT NULL,
  `goalkeeping_positioning` varchar(30) DEFAULT NULL,
  `goalkeeping_reflexes` varchar(30) DEFAULT NULL,
  `ls` varchar(4) DEFAULT NULL,
  `st` varchar(4) DEFAULT NULL,
  `rs` varchar(4) DEFAULT NULL,
  `lw` varchar(4) DEFAULT NULL,
  `lf` varchar(4) DEFAULT NULL,
  `cf` varchar(4) DEFAULT NULL,
  `rf` varchar(4) DEFAULT NULL,
  `rw` varchar(4) DEFAULT NULL,
  `lam` varchar(4) DEFAULT NULL,
  `cam` varchar(4) DEFAULT NULL,
  `ram` varchar(4) DEFAULT NULL,
  `lm` varchar(4) DEFAULT NULL,
  `lcm` varchar(4) DEFAULT NULL,
  `cm` varchar(4) DEFAULT NULL,
  `rcm` varchar(4) DEFAULT NULL,
  `rm` varchar(4) DEFAULT NULL,
  `lwb` varchar(4) DEFAULT NULL,
  `ldm` varchar(4) DEFAULT NULL,
  `cdm` varchar(4) DEFAULT NULL,
  `rdm` varchar(4) DEFAULT NULL,
  `rwb` varchar(4) DEFAULT NULL,
  `lb` varchar(4) DEFAULT NULL,
  `lcb` varchar(4) DEFAULT NULL,
  `cb` varchar(4) DEFAULT NULL,
  `rcb` varchar(4) DEFAULT NULL,
  `rb` varchar(4) DEFAULT NULL,
  PRIMARY KEY (`sofifa_id`),
  UNIQUE KEY `sofifa_id_UNIQUE` (`sofifa_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

ANEXO 2. CÓDIGO DE PYTHON QUE PROCESA LOS DATOS DE SOFIFA

Refiérase al archivo: SOFIFA desempeño - Jupyter Notebook.pdf

ANEXO 3. CÓDIGO DE PYTHON QUE SE ENCARGA DE LA EXTRACCIÓN DE INFORMACIÓN DE TWITTER

Refiérase al archivo: Twitter extracción - Jupyter Notebook.pdf

ANEXO 4. CÓDIGO DE PYTHON QUE CREA EL MODELO Y HACE EL ANÁLISIS DE LOS JUGADORES

Refiérase al archivo: Twitter análisis - Jupyter Notebook.pdf

ANEXO 5. CÓDIGO DE PYTHON PARA LA EVALUACIÓN DE UN SOLO JUGADOR CONTRA EL MODELO CREADO

Refiérase al archivo: Twitter evaluación unitaria - Jupyter Notebook.pdf