Universidad de Costa Rica
Facultad de Ingeniería
Escuela de Ingeniería Eléctrica

TRABAJO FINAL DE GRADUACIÓN
LICENCIATURA

Desarrollo de plataforma de algoritmos de Análisis de Datos que permitan hacer correlaciones entre las variables generadas por los dispositivos WIMU en partidos de fútbol

Por:

Daniel Méndez Zeledón

Ciudad Universitaria Rodrigo Facio, Costa Rica

Junio de 2021

Desarrollo de plataforma de algoritmos de Análisis de Datos que permitan hacer correlaciones entre las variables generadas por los dispositivos WIMU en partidos de fútbol

Por:

Daniel Méndez Zeledón

Sometido a la Escuela de Ingeniería Eléctrica de la Facultad de Ingeniería de la Universidad de Costa Rica como requisito parcial para optar por el grado de:

LICENCIATURA EN INGENIERÍA ELÉCTRICA

Aprobado por el Tribunal:

Dr. Marvin Coto Jiménez
Representante del Director

Dr. rer. nat. Francisco Siles Canales
Director, Comité Asesor

M.Sc. Marco Villalta Fallas, M.Sc.
Miembro, Comité Asesor

Dr. Jose Moncada Jiménez
Miembro, Comité Asesor

M.Sc. Osvaldo Fernández Cascante
Miembro del Tribunal

Agradecimientos y Dedicatoria

Agradezco enormemente a mi mejor amiga y compañera de vida, Ximena Abarca Solís, por todo el apoyo que ella me ha dado de manera incondicional durante esta etapa de mi vida. Por todas esas palabras y empujones que me ha dado, en su afán de apoyarme y recordarme que siempre hay una mejor versión de uno mismo y que vale la pena siempre el esfuerzo para lograr esa meta. Por todo el cariño y el afecto que me ha dado durante momentos en los que he caído y los hermosos momentos que hemos formado juntos. Por todos los pasos que hemos dado juntos y por todas esas veces que nos hemos jalado las orejas para mejorar, GRACIAS! Eres una persona impresionante y me siento muy orgulloso de tenerte en mi vida. Gracias por darte a los demás y al mundo, para que todos mejoremos de manera integral. Gracias por recordarme la importancia de nuestra naturaleza y nuestro mundo. Te amo por eso y mucho más.

A mi madre, Hellen Méndez, porque con su trabajo hizo que yo tuviese siempre la mejor educación que pude tener y por tantas batallas que hemos tenido juntos y hemos logrado sobrellevar. Gracias por todo el esfuerzo para que yo tuviese un muy buen futuro. Que, como en toda familia, siempre hay discusiones, hemos logrado sobrellevarlas y tu amor incondicional siempre lo he recibido y lo aprecio cada día. Gracias por todas las oportunidades que me has abierto que me han hecho ser parte de quien soy hoy.

A mi abuelita, Jeffie Zeledón, que mientras yo estuve días y noches estudiando, ella siempre tenía un espacio para estar conmigo, vacilar y, como buena abuelita, darme de comer, se preocupaba más por mi panza que por mis notas jaja. Gracias por todas las canas que te saqué cuando no quería comer pero igual insistías, como toda clásica abuelita. Por todos esos chistes y esas canciones que me cantabas que van a ser mis recuerdos por el resto de mi vida, haciéndome entender también que la música es parte importantísima de la vida.

A mi tía, Maureen Méndez, que siempre ha ayudado a mi abuelita y se ha preocupado por mi bienestar, a veces hasta más que mi madre y mi abuelita jutnas. Que aunque me jodías de vez en cuando, siempre buscabas ver que yo estuviese bien, siempre preocupada que su chito estuviese contento y lleno de

alegría. Muchas gracias por todo.

A mi profesor guía y quizá futuro colega, el profesor Francisco Siles Canales, que fomentó en mi esas ideas de seguir adelante investigando mi mente, de no conformarme con simples actividades, sino que esperaba de mi mucho y me motivó a lograrlo. Sin él, ni éste ni mi anterior proyecto se hubiesen materializado para poder convertirme en el ingeniero que soy. Gracias por ser un profesor diferente, lleno de energía para los estudiantes que, aunque a no todos se les impregna, muchos lo seguimos con el fin de sacar lo mejor de nosotros en esta ingeniería tan chiva.

La verdad, aunque suene un poco ególatra, estoy muy orgulloso de mí mismo, ya que nunca me visualicé como ingeniero, ni logrando tantos cambios y cosas buenas por mí y los que me rodean, no porque no quisiera, sino porque a veces me he estancado pero, con el apoyo de mis amigos y familia, he logrado seguir adelanta para ir alcanzando, poco a poco, las metas que me he propuesto. Buen trabajo Danny!

Resumen

Con el presente proyecto se pretende abrir una puerta para Costa Rica en el análisis por computador de datos de un partido de fútbol, a partir de la captura de movimiento de los jugadores en un partido a tiempo real usando una tecnología llamada WIMU de RealTrack Systems y analizando las variables que éste genera.

Usando algoritmos de Machine Learning, se crea una plataforma la cual permite el análisis de las diferentes variables obtenidas por el dispositivo WIMU. Actualmente, este sistema obtiene 174 diferentes variables, las cuales van desde la aceleración del usuario, así como las distancias entre jugadores o las áreas de juego de cada uno. Sin embargo, aunque las mismas, por sí solas, representan cierta información de gran uso para los equipos, se quieren buscar resultados más profundos a través de la interrelación entre ellas y cómo pueden afectar el desempeño de un juego colectivo.

En la plataforma se implementan 12 algoritmos de Machine Learning, los cuales permiten dar una correlación entre subgrupos de variables y el resultado del partido, con el fin de identificar patrones en las variables que determinen un resultado de partido esperado (Ganar, Perder o Empatar) y así, mejorar el planeamiento de entrenamientos, estrategias de juego y resultados de los partidos.

Abstract

The purpose of the following project is to open the door for Costa Rica on the computer data analysis for soccer matches, by capturing the player's movement on a real time match using the device called WIMU by RealTrack Systems and analyzing the data and variable generated by it.

With the use of Machine Learning algorithms, a platform was created to analize all the different variables obtaind by the WIMU device. Currently, this systems collects 174 different variables, which have different areas like: player's acceleration, distance between players or their playing areas. Eventhough these variables by themselves represent a very usefull information for the teams, deeper results can be descovered by their interrelation which can affect the team's performance.

In this platform implements 12 Machine Learning Algorithms which allow to correlate the different variable subgroups with the match results, with the purpose of identifying patterns on these variables that could determine the expected match result (Win, Draw or Lose) and then, enhance the training plans, game strategies and game results.

Índice general

Índice de figuras

Índice de cuadros

Acrónimos

$WIMU$	Wireless Inertial Measurement Unit
IMU	Inertial Measurement Unit
$GNSS$	Global Navigation Satellite System
UWB	Ultra-Wide Band
BLE	Bluetooth Low Energy
EMG	Electromiografía
AQ	Algorithm Quasioptimal
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
$2D$	Dos Dimensiones
$3D$	Tres Dimensiones
nD	n-Dimensiones
SVM	Support Vector Machine
MCM	Margen Clasificador Máximo
MS	Margen Suave
CVS	Clasificador de Vector de Soporte
GMM	Gaussian Mixture Model

1 Introducción

1.1. Antecedentes

Alrededor del mundo, se han ido desarrollando poco a poco técnicas y formas nuevas de obtención de datos deportivos, ya sea para fútbol, atletismo, natación o rugby, los cuales, en su mayoría, se utilizan sistemas de adquisición de datos como GPS, acelerómetros, giroscopios y, más actuales, los WIMU, los cuales son un conjunto de los tres anteriores(Gaffney et al. (2009)). Sin embargo, a nivel nacional, el desarrollo de investigaciones alrededor de estas tecnologías para obtención y análisis de datos es muy escaso, siendo la UCR en el laboratorio PRISLAB los únicos desarrollándolas. A pesar de esto, es de gran utilidad hacer uso de las investigaciones anteriores para determinar y centralizar el tema que se quiere tratar a continuación.

Inicialmente se tenía la idea de desarrollar estrategias de entrenamiento y jugadas, aprovechando el uso de los datos que se tienen a partir de los WIMU. Sin embargo, diversos artículos han ayudado a entender que la propuesta original abarcaba muchos detalles que quizá eleve la complejidad del proyecto a un punto que no se pueda dar solución a alguna particularidad.

Varias de los esfuerzos y propuestas planteadas en otros proyectos se han centrado en el rugby, especialmente el de Australia(Gabbett et al. (2012), Brian et al. (2009), Wisbey et al. (2018)), donde se enfocan en el desgaste de los jugadores y la movilidad que tienen ellos. A partir de WIMUs y sensores de palpitaciones, se pudo analizar y determinar que los jugadores se desgastan más rápido de lo que se había estimado en un principio. También, se han hecho estudios donde se intenta caracterizar los movimientos de corredores y nadadores de una manera fisiológica y biométrica, capturando lugares específicos del cuerpo y haciendo uso de WIMUs y medidores cardíacos (Silva et al. (2011)). Se notó a partir de estos, que los análisis se hacían alrededor de un limitado número de jugadores, para poder tener datos tangibles y poder trabajar con ellos con mayor facilidad. Además que el WIMU permite la obtención de una gran cantidad de variables y que no se sabe aún cómo interrelacionarlas. Muchos se han encargado de armar los sistemas y capturar valores relevantes pero fáciles de obtener, como distancia del ejercicio, pero análisis profundos para determinar la interrelación entre estas variables no se ha encontrado, solo llegan a mencionar los posibles alcances (Wang et al. (2018)).

La influencia del tamaño del sistema (sensores) que se utilice para la obtención de datos, juega un papel sumamente importante en el desarrollo de los mismos, ya que, evidentemente, estos equipos deben procurar ser lo menos invasivos posibles para que quienes estén practicando el deporte, no vean una imposibilidad de movimiento al llevarlos encima. Dado esto, existen desarrollos para la creación de equipos de obtención de datos que cada vez sean más pequeños, con más autonomía y que tengan más posibilidades de conexión para diferentes servicios, como internet (Kos y Kramberger (2017), Nieminen et al. (2005)). Se han encontrado casos donde un proyecto no se puede escalar ya que el equipo que se utiliza requiere mucho cableado y un ambiente controlado(Hughes et al. (2019), Schuldhaus et al. (2015)), pero muchos análisis se requieren en un campo de juego. Por esto, es de suma importancia identificar cual va a ser nuestro enfoque para poder identificar bien cual equipo es el que se va a utilizar para la obtención de los datos.

Como parte de el desarrollo de investigaciones, también se han encontrado propuestas que determinaban las facilidades y dificultades entre las diferentes formas de obtención de datos de un deporte, específicamente en el fútbol (Rein y Memmert (2016)). Se determina que, a pesar de que los WIMU y GPS facilitan mucho la determinación de los datos, la cantidad de datos que se obtienen es tanta que se vuelve casi que inmanejable. A su vez, describe que el análisis de imagen a través de una computadora es bastante complicado porque siempre existe la superposición de jugadores que imposibilita un seguimiento efectivo de los jugadores(McMillan (2015)). Se concluye que la forma manual por medio de la vista sigue siendo la mejor manera de obtener datos, a pesar de que no es 100 % certera, ya que siempre existe el error humano. A partir de estos artículos se logró entender que, el hacer uso de los datos de los WIMU puede volverse sumamente complicado cuando se analizan los 22 jugadores que hay en el campo, por lo que se debe limitar mucho la cantidad de jugadores que se van a utilizar para el análisis.

Dado que el tema a investigar es algoritmos de machine learning, también se encontraron datos muy relevantes alrededor de planteamiento y análisis de estrategias en un encuentro de fútbol, específicamente en el mundial 2006, un grupo de personas buscaban hacer un análisis de un encuentro en su totalidad, haciendo uso de análisis por imágenes automatizadas, donde una computadora determinaba, a partir de machine learning, las jugadas que se iban a dar (Grunz et al. (2012)). Durante la elaboración del mismo, se dieron cuenta de que, como ya se dijo con los otros artículos, la cantidad de datos era tan amplia que se volvió inmanejable, por lo que decidieron analizar únicamente 8 jugadores, 4 en ataque y 4 en defensa, cuando alguno de los dos equipos

atacaba el área del contrincante. Finalmente, lo que lograron fue hacer que la computadora entendiese los datos y que realizase con éxito ciertas estrategias, que eran corroboradas con el análisis visual de las imágenes originales. Se entiende que la cantidad de datos que se obtienen con los WIMU es enorme y que tratar de analizarlo todo, es algo que va a llevar demasiado tiempo. Esto, para un único proyecto, se sale de los parámetros reales del mismo.

Varias investigaciones leídas se centralizaban en el análisis de las posibles lesiones que podrían tener los atletas al correr, saltar, hacer un pase y tirar. Se hizo uso de los WIMU y de medidores cardíacos para ello(Lee et al. (2015), Bonomi et al. (2009)). Se hizo correr en una cinta a varios jugadores de fútbol de élite y hacer varios ejercicios de pases y tiros, así como ejercicios con saltos para forzar los músculos al desgaste. Al final se logró determinar que sí se puede valorar con este equipo si el jugador puede lesionarse con hacer movimientos muy fuertes. Sin embargo, una limitante grande de este método es el hecho que muchos de los equipos usados se tenían que colocar en las piernas y zapatos de los jugadores, por lo que su uso a nivel de campo era imposible, concluyendo tipo de análisis se puede hacer, por ahora, en un ambiente controlado de un laboratorio.

Un último tema que se vio al leer los artículos fue que, todo este tipo de equipo de análisis es de muy alto costo. Hacer uso de unas instalaciones de laboratorio para hacer las toma de medidas, todo el software computacional eleva el precio cuando los interesados en los resultados son equipos no populares o de divisiones inferiores o inclusive jugadores o deportistas que quieren entender su cuerpo a través de los datos(Ahmadi et al. (2014)). Un grupo de personas logró hacer uso de un pulsómetro de muñeca para poder hacer un análisis de las rutinas deportivas de un jugador de fútbol junto con un programa que iba ser de acceso libre(Hossain et al. (2017)). Estos son inicios de una de las cosas que se quieren hacer con el proyecto que se va a realizar.

Queda en evidencia varias cosas con la bibliografía analizada. Tomar una muestra de jugadores grande va a comprometer el buen análisis de los datos, por lo que encontrar una buena muestra es indispensable para el desarrollo del proyecto. Luego para poder hacer análisis más específicos de posibles lesiones, puede que sea necesario el uso de más equipo, como pulsómetros que ayuden a determinar el cansancio de los jugadores. Luego, el costo para muchos equipos o jugadores de estos análisis es muy alto, por lo que nuestra investigación va bien encaminada en hacer algo de bajo costo.

Finalmente, en Costa Rica no existe este tipo de análisis ni desarrollos y sólo el laboratorio para el que se está trabajando es el que se conoce con

proyectos de esta índole, es por eso que se quiere, a partir de esta investigación, sentar bases para los futuros análisis y trabajos alrededor de este tema, ya que se planea hacer una base de algoritmos en un cluster, que a partir del machine learning, pueda aprender las estrategias planteadas en los datos y desarrollar listas con datos relevantes para en análisis.

1.2. Justificación

Con la presente investigación se pretende entrar en un tema el cual, en Costa Rica, no se tiene mucho conocimiento. A nivel comercial, el análisis de partidos de carácter colectivo como fútbol o básquetbol se está convirtiendo en algo de interés para los clubes, ya que permite hacer un análisis estratégico más exacto y con mayor velocidad, lo que favorece con el desempeño de jugadores y su efectividad de juego.

El desarrollo de la plataforma de implementación de algoritmos que se plantea hacer, será de gran importancia para los equipos interesados en mejorar los resultados positivos de sus equipos, así como la salud de los jugadores debido a la mejora en el desempeño de los entrenamientos y estrategias que éste análisis puede llegar.

Actualmente, muchos de los análisis de deportes colectivos se hacen por medio de análisis de imágenes haciendo uso de algoritmos de detección de patrones a partir de videos, así como el análisis directo hecho por un humano. Sin embargo, el análisis de videos tiene limitantes, como la sobre posición de jugadores en el mismo cuadro en análisis, las repeticiones en caso de que se analice directamente desde una transmisión de televisión o la cobertura de área de las cámaras, ya que se pierde exactitud al no tener el área completa de juego y además, se va a depender de la resolución de las mismas para que la imagen sea más fácil de analizar. Igualmente, el análisis hecho directamente a través de la observación por un ser humano, implica un resultado lento de datos y se requieren muchas personas al mismo tiempo para poder hacer una localización de cada jugador en tiempo real. Es por esto que se han desarrollado sistemas de adquisición de datos más automatizados cuyo análisis a través de una computadora, van a facilitar, agilizar y, con esto, mejorar el resultado final obtenido.

El desarrollo de estas tecnologías ha hecho que la cantidad de datos obtenidos sea muy grande, inmanejable para un ser humano, por lo que la automatización del análisis es de gran relevancia. Actualmente, empresas como Real Track Systems, haciendo uso de su dispositivo WIMU, han logrado la

obtención de datos en tiempo real para deportes colectivos, como fútbol, fútbol sala, balonmano y básquetbol. Han logrado depurar su sistema a tal punto que se pueden obtener 174 variables en tiempo real de cada uno de los jugadores. Estas variables dan resultados inmediatos relevantes como la velocidad del jugador, distancias recorridas, aceleraciones o altura, pero no existe una forma de determinar una correlación entre ellas para identificar algún patrón relevante, como por ejemplo, si existe una relación entre la distancia recorrida con el hecho de que el partido se haya ganado y poder predecirlo. Aquí es donde esta investigación tiene relevancia. Se pretende implementar una serie de algoritmos existentes para poder identificar correlaciones entre las 174 variables, con el fin de identificar cuales son relevantes para realizar una predicción del resultado y evaluar la efectividad del mismo, así como crear reglas que nos permitan determinar las variables más relevantes para su entrenamiento, a partir de un resultado querido.

1.3. Planteamiento del problema

Existe un sistema de captación de datos llamado WIMU, creado por la empresa RealTrack Systems, el cual tiene la facilidad de registrar 174 variables diferentes para cada jugador. Las variables van desde las velocidades, distancias, impactos, hasta la relación entre ellas. Esto implica que su análisis es sumamente complicado, siendo imposible determinar las posibles relaciones que hayan entre ellas sin herramientas de extracción de datos. Por tanto, haciendo uso de Machine Learning, se pretende realizar una implementación de al menos 9 algoritmos de análisis de datos capaces de identificar las posibles interrelaciones entre las 174 variables que este sistema permite extraer de cada uno de los jugadores que lleven el dispositivo puesto. El objetivo primordial es predecir un resultado a partir de datos de juegos anteriores e identificar cuales son las variables más importantes para enfocar los entrenamientos alrededor de dichas variables. También se intentará explorar mejoras a los algoritmos en inclusive determinar si es posible agregar más con el fin de tener mejores comparaciones. Además, se pretende crear reglas que indiquen cuales de las variables son determinantes para identificar el resultado final del equipo, esto con el fin de hacer recomendaciones de entrenamiento de dichas variables. Es de gran relevancia el resultado obtenido al final para darle aún mas peso al desarrollo de estas tecnologías de captación de datos, ya que se están volviendo cada vez más populares y se puede inclusive comercializar para obtener beneficios económicos para futuras investigaciones.

1.4. Objetivos

1.4.1 Objetivo General

Implementar algoritmos de análisis de datos que permita extraer resultados relevantes para el equipo de fútbol en estudio, a partir de la combinación de las 174 variables extraídas con sistema WIMU de Real Track Systems.

1.4.2 Objetivos específicos

- Determinar cuales algoritmos predicen mejor los resultados entre ganar, perder o empatar, a partir de la exactitud de cada uno.
- Determinar cuales variables permiten que el equipo gane, pierda o empate, haciendo uso de las reglas de asociación.
- Identificar cuales variables son más relevantes según cada uno de los algoritmos.
- Generar reglas que permitan al equipo reforzar entrenamientos alrededor de las variables determinadas como importantes.
- Identificar si la combinación de varios algoritmos generan un mejor resultado de predicción.
- Identificar cuales partidos se relacionan naturalmente entre sí, dadas sus características parecidas, a través del uso de clustering.

1.5. Alcance

Con este proyecto se pretende implementar algoritmos de análisis de datos que permita crear varios subconjuntos de variables que van a ser relevantes para determinar un resultado esperado. No se pretende crear una interfaz gráfica para seleccionar cual algoritmo se va a utilizar. Se pretende usar como base 9 algoritmos, pero en un futuro, se pueden utilizar más. Los datos utilizados serán, en su mayoría, con partidos del campeonato nacional de fútbol de Primera División. El resultado de la investigación será hacer una primera aproximación para identificar cuales variables son las más relevantes y cuales permiten que el equipo gane, pierda o empate, así como buscar predecir eficazmente estos resultados y dar recomendaciones de qué particular variable entrenar para mejorar la posibilidad de ganar o el resultado que se quiera obtener.

1.6. Metodología

1.6.1 Descripción de los datos originales

Al comenzar con este proyecto, gracias a la colaboración con el Centro de Investigación del Movimiento Humano CIMOHU un equipo de fútbol de Primera División facilitó los datos de dos temporadas completas (torneo corto de 6 meses cada uno) de sus partidos, específicamente del torneo de invierno de 2018 y el de verano de 2019, incluyendo los partidos internacionales que el equipo enfrentó en este período de tiempo para un total de 53 partidos. Cada partido correspondía a una fila y las 174 variables determinadas por el WIMU en cada uno de ellos correspondían a las columnas. Éstas ultimas eran el promedio del desempeño de todos los jugadores que participaron en el partido, tanto en el primer tiempo, como en el segundo tiempo, los cuales se sumaban para dar el total de cada variable por jugador a lo largo del partido, y luego se promediaban para dar el resultado de todo el equipo por variable.

Cabe destacar que estas 174 variables son las variables predeterminadas por el sistema WIMU, no hubo ninguna modificación en la elección original de estas variables de parte del preparador físico encargado del uso y recolección de datos del sistema. Estas variables se subdividían en 6 grandes grupos de variables:

- Relacionadas con **Distancia:** por ejemplo:
  - *Vel Rel[A-B]:* Velocidad Relativa en la zona de A-B.
  - *HSR Abs Dist:* Abreviatura de "High Speed Running", es el HSR absoludo por minuto a velocidades superiores a HIA (por defecto 75.5 % de la máxima del jugador).
- Relacionadas con **Aceleración:** por ejemplo:
  - *Dec/min:* Media de desaceleraciones por minuto.
  - *Max Dec (m/s2):* Desaceleración máxima alcanzada en la sesión.
- Relacionadas con **Sprints:** por ejemplo:
  - *MAX Speed(km/h):* Velocidad máxima alcanzada.
- Relacionadas con **Impactos Horizontales:** por ejemplo:
  - *Impacts [A-B]:* Número de impactos en la zona de A-B.
- Relacionadas con **Cantidad de Pasos:** por ejemplo:

  - *Takeoff[A-B]:* Número de despegues en la zona de A-B.

- Relacionadas con **Cargas:** por ejemplo:

  - *DSL:* Abreviatura de "Dynamic Stress Load", es el total de los impactos ponderados, los cuales se basan en los valores del acelerómetro por encima de 2G.

1.6.2 Preanálisis de los datos y algoritmos

A partir de los datos brindados por el CIMOHU, se hizo un análisis previo para entender los mismos y así identificar qué de toda la información que se tenga a disposición, es la relevante para usar en el proyecto. Como se mencionó anteriormente, se tenía un resumen de cada partido, con 174 variables de cada jugador, subdivididos en primer tiempo, segundo tiempo y totalidad del encuentro.

Una vez sabiendo la cantidad de datos, se procedió a hacer una búsqueda en la web, donde se escogieron, inicialmente, 9 algoritmos (3 de Clasificación, 3 de Clustering y 3 de Reglas de Asociación) los cuales se adaptaban mejor a los datos que se tenían, esto según la teoría. Se utilizaron algoritmos que son conocidos para la minería de datos y Machine Learning y, junto con la recomendación del profesor Francisco Siles, se tomaron, preliminarmente, los siguientes algoritmos: K-Nearest Neighbors, Linear Discriminant Analysis y Suport Vector Machine en la seccion de algoritmos Clasificadores, X-Means, Gaussian Mixture Model y DBSCAN para los algoritmos de Clustering, AprioriTID, AprioriHybrid y FP-Growth para los algoritmos de reglas de asociación. Se tenía entendido que la escogencia podría variar según a como se fuese desarrollando el proyecto.

Una vez hecho esto, se procedió a buscar librerías las cuales contuviesen ya los algoritmos listos para su uso, pues el enfoque del proyecto no era crear algoritmos desde cero, sino hacer uso de ellos. Se encontró un curso donde se explica el uso de la librería SKLearn la cual contenía la mayor parte de los algoritmos base de Machine Learning por lo que se escogió esta como base, la cual hacía uso del lenguaje de programación Python para su ejecución.

1.6.3 Preprocesamiento, implementación, y análisis de los algoritmos

Seguidamente, se comenzó a implementar los algoritmos usando la base de datos descrita y se identificó que a los datos originales se le debían hacer modificaciones para poder ser consumidos por la librería. Los datos estaban

distribuidos originalmente en tablas resumen en Excel, con los datos de cada jugador separados en primer tiempo, segundo tiempo y totalidad del encuentro. Eran varios archivos, por lo que se determinó que es mejor utilizar una única hoja de datos para trabajar.

Para esto fue necesario separar cada uno de los partidos en hojas de cálculo separadas para hacerles un análisis general y luego resumirlas en un solo documento. En cada una de las hojas, se procedió a identificar las columnas y filas más relevantes de la siguiente forma: el análisis original se debía hacer por partido y no por cada jugador, por lo que se tomó únicamente las filas que contenían los promedios de las columnas, quedando 3 específicamente, promedio del primer tiempo, promedio del segundo tiempo y promedio del total del partido; a nivel de columnas, existían varias cuyos valores eran siempre 0, por lo cual se eliminaron para el análisis de los datos.

Para la extracción de estas filas y columnas, se elaboró un script el cual toma las hojas de cada partido anteriormente descritas, les extrae únicamente las 3 filas necesarias y elimina las columnas cuyos valores sean 0 o no contengan datos. A su vez les extrae el equipo contra el que se jugó, la locación del encuentro (casa o visita) y marcador. Estos datos, se separaron en 3 documentos, uno con los datos de todos los promedios del primer tiempo de cada partido, otro con todos los promedios del segundo tiempo de cada partido y un tercero con todos los promedios del partido completo de cada partido, por lo que cada archivo final consta de 52 filas, que son los 52 partidos disputados y 142 columnas correspondientes a las variables promediadas.

1.6.3.1 Modificación de Variables Cualitativas

En Machine Learning es indispensable hacer uso de datos numéricos, por lo que los datos cualitativos son algo que hay que modificar. Todos los datos cualitativos fueron codificados para que funcionaran como datos cuantitativos haciendo uso del método **One Hot**.

Los datos cualitativos que existen en la tabla son: el equipo contrario y la localidad del partido (casa o visita), esto generó 14 nuevas columnas agregando más variables a los datos originales, sin embargo, estas 2 columnas originales fueron agregadas para poder tener control de los datos, no son parte de las 174 determinadas por los WIMU.

1.6.3.2 Ajuste de Variables Cuantitativas

Fue necesario hacer un procesamiento de variables cuantitativas para ajustar la escala en la que éstas se encuentran. Este cambio se suele hacer debido a que algunos modelos de Machine Learning, por el tamaño de escala en la cada una de las clases se encuentra, hacen que algunas sean más dominantes que las demás, haciendo que estas variables no sean consideradas del todo por el modelo.

El modelo de escalación que se utilizó es el de **Estandarización**. Se elige éste método sobre el método llamado Normalización ya que el método Normalización es funcional cuando los datos de cada clase se muestran como una distribución normal, mientras que el de Estandarización se utiliza para cualquier grupo de datos. Otra detalle importante es que este método no se debe aplicar al las variables cualitativas convertidas a cuantitativas (Eremenko et al. (2019)).

1.6.3.3 Preprocesamiento en los Algoritmos Clasificadores

Una vez completada la primera fase, se comenzó la ejecución de los algoritmos, empezando con los Clasificadores. Cada uno de los algoritmos ejecuta varios preprocesamientos de los datos, primero se transforma las variables cualitativas en cuantitativas con el método One Hot previamente mencionado, ya que se necesita que la tabla de datos contenga únicamente números, luego se separaron los datos para que parte de ellos sean de uso exclusivo para el entrenamiento de los algoritmos y otro para las pruebas de su funcionamiento, y finalmente se normalizaron los valores de cada columna para que todos tengan un mismo rango, haciendo uso de la Estandarización.

Ya con los datos preprocesados, se entrenó cada algoritmo con el grupo de datos preparado para ello y se procedió a probar la efectividad del algoritmo haciendo uso de la otra sección de datos, tratando de predecir si el partido que se encuentra en estudio se Gana, se Pierde o se Empata. Como resultado, la cantidad de veces que la predicción fue la correcta, es la que determinaba la precisión de cada algoritmo.

Al llegar a este punto, se notó la facilidad con que se podían manejar los algoritmos de clasificación y se decidió agregar 3 algoritmos más al set, con el fin de tener más comparaciones y así determinar cuál algoritmo se desempeña mejor. De igual forma, se procedió a ejecutar la parte de preprocesamiento de datos, entrenamiento y pruebas, dando como resultado el porcentaje de efectividad de los mismos. Al final, se utilizaron los siguientes algoritmos de

Clasificación:

- K-Nearest Neighbors
- Linear Discriminant Analysis
- Suport Vector Machine
- Naive Bayes
- Decision Trees
- Random Forests

1.6.3.4 Preprocesamiento en los Algoritmos de Clustering

Seguidamente implementaron los algoritmos de Clustering, donde se identificó la dificultad de ponerlos en marcha. Como se había mencionado, el objetivo del proyecto era elaborar una batería de algoritmos que pudiesen servir como base para la comparación de los mismos y crear un primer paso en el análisis de los datos, y no crear un algoritmo desde cero. Por tanto, el algoritmo de X-means fue reemplazado por el K-means ya que la librería SKLearn sólo contaba con este tipo de Cluster, a su vez que el algoritmo DBSCAN mostraba resultados muy erráticos, por lo que cambió por el algoritmo Hierarchical Clustering. Estos nuevos algoritmos de Clustering tenían una gran cantidad de documentación de implementación lo cual facilitaba más el ponerlos en funcionamiento, todo esto con el fin de reducir el tiempo total de programación para enfocarse más en la ejecución y el análisis de los resultados. Al final, a manera de resumen, se utilizaron los siguientes algoritmos:

- K-Means
- Gaussian Mixture Model
- Hierarchical Clustering

A nivel de ejecución, en los algoritmos de Clustering se hizo un preprocesamiento para eliminar las variables cualitativas, ya que, si se convierten en cuantitativas, al ser valores de 1 y 0, van a tener mucho peso entre sí a la hora de toma de decisión de cómo agrupar los valores, por lo que se eliminaron. Además, se partió la base de datos, de igual manera que con los Clasificadores, para poder tener una parte para entrenar el algoritmo y otra para probarlo.

Para saber cuántos Clusters eran necesarios para cada algoritmo, se hizo uso del *"Método del Codo"*, el cual nos indica cual es la cantidad de ciclos más óptima que le toma al algoritmo en crear los Clusters, y esto va a depender de qué tan agrupados estén los Clusters al final de la ejecución. Una vez hecho esto, el algoritmo toma cada partido y lo agrupa en un Cluster según las características propias de cada uno. El resultado de la ejecución es una lista donde indica a cual cluster pertenece cada uno de los partidos que se están analizando.

1.6.3.5 Preprocesamiento en las Reglas de Asociación

Al igual que con los Clusters, obtener librerías con los algoritmos previamente escogidos fue tarea difícil, por lo que se cambió la escogencia de los mismos debido a la facilidad de implementación con una librería llamada MLXtend. Los algoritmos de AprioriTID y AprioriHybrid se eliminaron y se escogió el Apriori y el FPMax. El algoritmo FPGrowth no se modificó ya que también viene en esta librería mencionada. Resumiendo, se probaron los siguientes algoritmos:

- Apriori
- FP-Growth
- FP-Max

Para la implementación de estos algoritmos se creó un script el cual hizo un preprocesamiento de los datos muy diferente a los anteriores, ya que las reglas de asociación suelen hacerse con datos cualitativos y no cuantitativos, por lo tanto se tiene que modificar todos los valores numéricos a literales. Para esto, se deben discretizar los valores para tenerlos en rangos. Se encontró que una buena manera de discretizar valores para transferir variables cuantitativas a cualitativas es haciendo uso del Clustering. Cada una de las columnas se le aplicó el método K-Means, el cual indicaba una cantidad de Clusters óptima para el agrupamiento de los números, que en este caso, se convirtieron en los rangos. Cada columna se analizó por aparte para determinar la cantidad óptima de rangos para dicha columna, y luego se creó una sola tabla con todos los valores discretizados, que luego fueron usados por los algoritmos.

Una vez hecho esta modificación de los datos, se procedió a ejecutar los 3 algoritmos por aparte. Luego, haciendo uso de la misma librería, se crearon las reglas de asociación según cada uno de los algoritmos, la cual genera una cantidad amplia de reglas. Se filtraron las mismas para obtener los resultados

que más se quieran.

1.6.4 Optimización y extracción de datos

A nivel teórico, muchos de estos algoritmos, con el fin de ser aprendidos, utilizaban grados de libertad predeterminados que facilitan su entendimiento, sin embargo, a nivel práctico, se debieron buscar los valores óptimos para que el algoritmo se desempeñase de la mejor manera.

Las librerías a utilizar para los Clasificadores, tenían bastantes grados de libertad que podían ser modificados y así obtener el mejor resultado de los algoritmos, por lo que, una vez implementados los mismos, se buscó optimizar la mayoría de ellos. Para ello, se creó un script por cada uno, que identificó la mejor combinación de grados de libertad que haga que el porcentaje de éxito del algoritmo fuese el mejor posible.

Para el proyecto era muy importante identificar cuales columnas son las más influyentes, por lo que se agregó a cada algoritmo de clasificación, una funcionalidad llamada *FeatureImportance*, que permitía extraer la importancia de cada columna para la determinación de los resultados de las pruebas, pudiéndose ver, en un gráfico de barras y además en una tabla con el porcentaje de importancia.

A nivel de Clustering, también se hicieron mejoras con el fin de optimizar los resultados iniciales, ya que, de igual forma, existían varios grados de libertad en ellos que al ser modificados, se obtendría el mejor resultado posible. Por ello se creó un script que determinaba cual era la mejor combinación de grados de libertad a partir de la desviación estándar que generaba cada una de las posibles combinaciones, ya que entre menos desviación, mejor es la agrupación de los elementos del cluster.

Además, es muy importante entender cuales columnas influyeron en la decisión de cómo agrupar los diferentes partidos, pero la librería no da estos datos. Sin embargo, se aprovecharon las características del algoritmo LDA, que puedo usar tanto de Clasificador como de reductor de dimensiones, haciéndolo parecido a un algoritmo de Clustering. Por lo que una vez ejecutado cada algoritmo de Cluster, se creó una nueva tabla con las mismas filas y columnas pero agregando una nueva columna de Cluster asociado determinada por el algoritmo de clustering en cuestión. Luego, en vez de identificar si el equipo pierde, gana o empata como en los clasificadores originales, se ejecutó el algoritmo LDA para que determine en cual Cluster debería estar el partido

que se está analizando y se identificó el porcentaje de precisión con que este análisis se hace, obteniendo así un porcentaje de efectividad del Cluster y a la vez, al haber implementado la obtención de columnas importantes para LDA, se obtienen las columnas importantes para el Cluster.

Para la extracción de datos de las reglas de asociación, debido a que la combinación de reglas y datos es tan amplia, se determinó que se iba a obtener todas las combinaciones de 3 a 5 columnas que determinaran una aparición de al menos 30 % de las veces en las columnas, para poder limitar la cantidad de reglas, ya que son en el rango de los miles.

Finalmente, se hizo una última optimización de los algoritmos de clasificación reduciendo la cantidad de columnas por medio de PCA. El PCA proyecta la dimensión real de una base de datos en una de menor valor, facilitando su análisis, por lo que se buscó reducir la cantidad de columnas a únicamente 3 para poder visualizarla en 3D. Sin embargo, igual que con el resto de algoritmos, el PCA tenía una cantidad de grados de libertad que se podían modificar, por lo que se hizo además un script que optimizase la mejor cantidad de columnas a las cuales la tabla original se debía reducir para obtener el mejor porcentaje de acierto al aplicar los algoritmos.

1.6.5 Caracterización de los algoritmos de Clasificación

Una vez implementados todos los algoritmos anteriores, se determinó que, a pesar de que un clasificador en específico dio como resultado un 100 % de precisión, ésto iba a ocurrir sólo para un grupo específico de datos y no para todos los posibles casos. Esto se da porque se escogió que la variable *RandomState*, la cual hace la separación pseudo aleatoria de los datos para la base de entrenamiento y la base de prueba fuese la mejor para el set de datos, esto no garantiza que cuando se tengan más datos nuevo, se pueda garantizar ese 100 %.

Para solucionar lo anterior, se puso a prueba cada algoritmo, creando un nuevo script que caracteriza todos los posibles valores de esta variable *RandomState*, lo cual genera un valor de precisión para cada uno de ellos, usando 4 métricas distintas (*Accuracy, Precision, Recall y F1-Score*). Luego, se determinó la mediana y la desviación estándar del conjunto de precisiones de cada métrica para decidir cual es el mejor algoritmo de clasificador a nivel general.

Sin embargo, para facilitar la caracterización y debido a que la cantidad de partidos que se gana, pierde y empata no es la misma (50 % Gana, 25 % Pierde

y 25 % Empata), se decidió partir la base de datos en dos tablas diferentes, una con todos los partidos separadas entre Ganar y No Ganar, y otra con los partidos que no se gana, separados en Empatar y Perder, así se podía tener siempre una cantidad igual entre cada una de las categorías.

Ya con las tablas separadas, se procedió a hacer la caracterización de cada uno de los algoritmos, tomando en cuenta el valor de mediana más alto como el algoritmo que mejor se acopla a todas las situaciones. Se desarrolló una tabla comparativa con el fin de visualizar mejor los valores determinados.

1.6.6 Nuevas Reglas de Asociación

Como base del trabajo, se tenía la pregunta: "*Cuál de todas las variables del WIMU son determinantes para que el equipo Gane, Pierda o Empate?*". Una vez implementados todos los algoritmos de Reglas de Asociación, se descubrió que como éstos buscan analizar todas las posibles relaciones que haya entre las variables del WIMU, computacionalmente se vuelve muy pesado el trabajo para obtener las relaciones en donde se muestra la variable de "*Resultado*"(indicativa de si el equipo Gana, Pierde o Empata) como relevante para las reglas, por lo que no se le podía dar respuesta a la pregunta original del trabajo.

Para poder entonces contestar la pregunta, se decidió reducir la cantidad de columnas variable de la tabla original, a únicamente aquellas que los algoritmos de clasificación determinaran como relevantes. Para facilitar aún más el trabajo, se utilizó el mejor algoritmo de Clasificación obtenido a partir de la caracterización hecha anteriormente. Los algoritmos de clasificación, haciendo uso de la funcion *FeatureImportance*, entregaban una lista con cada una de las columnas variables ordenadas de mayor importancia a menor importancia, por lo que solamente se le colocó un umbral mínimo para escoger una cierta cantidad.

A partir de lo anterior, se creó una nueva base de datos con solamente una parte de las columnas variables y se vuelven a correr los algoritmos de Reglas de Asociación, obteniendo así un mejor set de reglas en donde la variable "*Resultado*"(Columna que indica el resultado final del partido) pudiese aparecer entre las reglas y poder filtrar cuando se Ganase, Perdiese o Empatase.

1.7. Procedimiento de evaluación

Primero, se hizo una búsqueda de algoritmos cuya implementación fuese rápida, con el fin de agilizar el proceso de programación de las mismas para enfocarse en la ejecución y el análisis de los resultados. Se le pidieron recomendaciones al profesor guía con el fin de que la escogencia fuese acorde con los datos que se estaban usando, dada la experiencia de él en el ámbito de Machine Learning.

Seguidamente, cada uno de los algoritmos de Clasificación tuvo una corrida con los grados de libertad predeterminados por la librería como comunes, con el fin de tener un primer resultado, luego se hicieron las optimizaciones respectivas a cada uno con el fin de obtener el mejor porcentaje de aciertos en la predicción. Luego se comparó entre todos esos mejores resultados para determinar un primer algoritmo. Luego se caracterizarían cada uno de los algoritmos de Clasificación, con el fin de determinar cual de ellos se comporta más estable a lo largo de diferentes datos de entrada.

Con el fin de determinar que los resultados sean robustos, se separó una parte de los datos para poder hacer una última validación, haciendo que los algoritmos no hayan usado nunca estos datos como parte de su entrenamiento, solo para determinar que efectivamente, el mejor algoritmo fuese el escogido.

Para los algoritmos de Clustering, haría una corrida que determinará el funcionamiento de cada uno, determinando el cluster al cual cada partido pertenece. Con este resultado, se creó una nueva base de datos que, haciendo uso del algoritmo LDA, determinó la efectividad de cada uno de los algoritmos y se supor cuales son las columnas más relevantes de ellos. De igual forma, para tener una mejor elección de los algoritmos, se separó la base de datos para tener datos de validación, que luego se harían pasar por el proceso de clasificación LDA de los clusters para comparar los resultados de validación con los obtenidos en las pruebas.

Para los algoritmos de Reglas de Asociación, se ajustarían poco a poco los grados de libertad de ellos, lo cual identificaría un umbral mínimo que nos brinde una buena cantidad de reglas en un tiempo corto. Si el algoritmo tardase muchas horas ejecutando, se subirían los límites para que se obtengan resultados a una mayor brevedad. Seguidamente, se intentaría encontrar entre las reglas, todas aquellas que den como resultado si el equipo Gana, Pierde o Empata, todas las demás se debían obviar. Si el algoritmo no generaba alguna de las reglas, se procedería a ejecutar la reducción de columnas con el fin de eliminar posibilidades que no son relevantes y así poder obtener cuales

columnas implicarían que el equipo entre en alguna de las categorías indicadas.

No se pretendía que los algoritmos den resultados exactos o que siempre sean correctos, simplemente se planeaba implementar los algoritmos y que éstos den un resultado esperado.

2 Marco Terórico

2.1. Wireless Inertial Measurement Unit (WIMU)

Una Unidad de Medición Inercial (Inertial Measurement Unit IMU, en inglés), es un sistema cerrado que se utiliza para detectar los cambios en el movimiento angular y la velocidad. Es utilizado ampliamente para diferentes actividades, como para la detección de altitud y cambios en el movimiento en aeronaves, también en la geolocalización en sensores para soldados a campo traviesa, o como estabilizadores en cámaras (Tanenhaus (2012)). En el caso del WIMU, es pues una unidad de medición inercial pero de carácter inalámbrico.

Para el cálculo de cualquier movimiento inercial, es necesario es necesario hacer uso de una instrumentación muy precisa el cual debe detectar el movimiento rotacional (momento angular) y el movimiento linear relativo a una referencia estable. Estos instrumentos son los que se conocen como giroscopios y acelerómetros (Morrison (1985)). Un WIMU, para entonces detectar los movimientos, requiere pues giroscopios y acelerómetros para estas mediciones.

2.1.1 Giroscopios

Los giroscopios son equipos que normalmente se usan para detectar el movimiento angular. En su forma más simple, un giroscopio se muestra como un rotor que gira muy rápido o un aro montado en un eje que le permite inclinarse sobre su eje de rotación relativa a la base. La velocidad de rotación del aro da una resistencia al cambio de movimiento angular del mismo, por tanto, el giroscopio tiende a mantener su orientación original el cual, con un un buen alineamiento inicial se convierte en el marco de referencia de estabilidad del objeto en movimiento que lo contiene (Morrison (1985)).

Normalmente los giroscopios son implementados en con un sistema servo controlado para estabilizar el marco de referencia en el cual los acelerómetros van a estar montados para medir el movimiento linear. Este marco se va colocar en algún lugar como parte del objeto donde se va a tomar la medición (Morrison (1985)).

Un giroscopio de un grado de libertad es aquel en el cual su eje de rotación puede moverse en una única dirección (sobre eje X, Y o Z) pero restringida en las otras. Uno de dos grados de libertad es aquel que se le permite al eje de rotación moverse en dos direcciones y restringida una y, similarmente, un giroscopio de tres grados de libertad es cuando el giroscopio es capaz de moverse en los tres ejes de rotación. A estos dos últimos se les denominan giroscopios libres (Morrison (1985)). Un ejemplo de un giroscopio de dos grados de libertad se puede observar en la Figura 2.1.

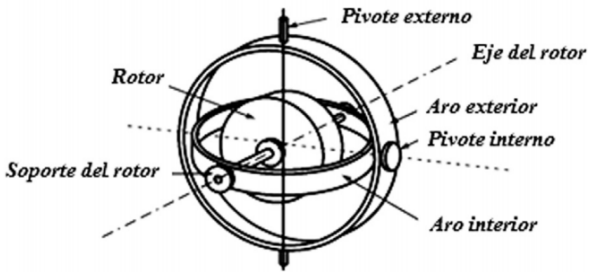


Figura 2.1: Partes de un giroscopio de dos grados de libertad. Nota: tomado de Reif-Acherman (2014), p.194.

Debido a que tres ejes de estabilización son requeridos para montar un marco de referencia, van a ser necesarios tres giroscopios de un grado de libertad, dos de dos grados de libertad o uno de tres grados de libertad. En los inicios del uso de estos artefactos, se prefería la opción de tres giroscopios de un grado de libertad hasta que se desarrollaron técnicas y modelos para poder eliminar el acoplamiento cruzado entre los ejes en los giroscopios de múltiples grados de libertad (Morrison (1985)).

2.1.2 Acelerómetros

Un acelerómetro es un dispositivo capaz de medir la aceleración de un vehículo en una dirección específica relativa a un marco de referencia estabilizado. Los acelerómetros mecánicos, siendo estos los más básicos, se componen de un resorte o bisagra unidos a una masa, que al aplicar una fuerza en ellos, se desplaza, este desplazamiento va a indicar la aceleración del vehículo. El desplazamiento de esta masa es detectada por un sensor mecánico, óptico, inductivo o capacitivo (Morrison (1985)).

Los acelerómetros mecánicos tienen el defecto de que, debido a su construcción y los requerimientos del cliente, factores como la vibración, los golpes, la variación de la temperatura y el paso del tiempo hacen que los acelerómetros

tengan errores en el cálculo final de la aceleración, es por esto que requieren mucho más mantenimiento y constante calibración a diferencia de los giroscopios. Agregando a esto, la manufactura de estos acelerómetros es muy difícil debido a que cada cliente tiene un requerimiento distinto para su uso. Es por todo lo anteriormente dicho que los acelerómetros tienen costos de producción altos (Morrison (1985)).

Haciendo uso de los acelerómetros y giroscopios que los IMU tienen, uno de éstos detecta entonces la aceleración y la posición angular de los tres ejes y mide, a través del tiempo, estos parámetros para medir el cambio respecto a la posición inicial. Sin embargo, esto no quiere decir que estos sirven como un sistema de navegación inercial, ya que éstos no conocen su propia posición, pero, un IMU con capacidades extendidas puede proveer de datos como velocidad, posición, posición angular, aceleración con detección de error y todo esto con una gran precisión (Tanenhaus (2012)).

Los IMUs convencionales no detectan la velocidad lineal ni la posición directamente, si no que utilizan la integración y la doble integración de la aceleración para determinarlos, lo que implica que un error pequeño en la aceleración, al ser integrados, van a generar grandes errores en la velocidad y más aún en la posición. Es por esto, que a su vez, los IMUs se les implementa sistemas de posicionamiento global GPS para complementar las mediciones de los mismos (Tanenhaus (2012)).

En el caso de los WIMU de RealTrack Systems(2.2), los IMUs implementados son de carácter inalámbricos, el cual consta de muchos equipos internos que le permiten la obtención de múltiples variables. Dentro de los dispositivos que tienen estos artefactos se encuentran un Procesador Intel Atom de 2 Cores a 1 GHz., un sistema híbrido de GPS, GNSS y UWB (sistema capaz de proveer capacidades de posicionamiento preciso de un objeto utilizando una red de rango corto (Sahinoglu et al. (2005))), acelerómetro de 3 dimensiones 16G a 1000 Hz usado para recoger datos de inclinación, caída libre y tiempo en el aire, acelerómetro de 3 dimensiones de 400G a 1000 Hz para recoger datos de aceleración, impacto, tiempo de reacción, saltos y golpes, un giroscopio de tres grados de libertad a 1000 Hz y 8000^{o}/seg para medir velocidad angular, rodamientos, inclinación y posicionamiento sobre el plano, un magnetómetro de tres dimensiones 100 Hz usado como compás y evaluación de dirección de movimientos, un barómetro a 120 kPa para determinar datos como presión atmosférica o altura barométrica, todo esto siendo monitorizado en tiempo real a través de 4 modos de seguimiento: WIFI 802.11 B/G/N para la emisión de datos, Bluetooth 4.1 para intercambio de datos, ANT+ para interactuar con ostros dispositivos (pulsómetros, podómetros, medidores de oxígeno en

músculo, presión sanguínea, sensor de actividad eléctrica EMG y temperatura timpánica) y BLE (RealTrack Systems (2017)).

Figura 2.2: Dispositivo WIMU Pro. Nota: tomado de la página oficial de Real-Track Systems (2017).

2.2. Métodos de normalización y métricas

2.2.1 One Hot

One Hot es una método que transforma columnas con características cualitativas a cuantitativas. Según Eremenko et al. (2019), por cada valor cualitativo que exista, se crea una columna nueva con el nombre de ese valor y agrega un 1 o un 0 según pertenece a la nueva clase.

Color
Red
Blue
Yellow
Yellow
Red

Red	Blue	Yellow
1	0	0
0	1	0
0	0	1
0	0	1
1	0	0

Figura 2.3: Método One Hot para transformación de variables cualitativas.

Como se puede nota en la Figura 2.3, por cada variable cualitativa, se va a crear una columna nueva, creando una matriz con tantas columnas como variables, seguidamente el método coloca un 1 en la columna correspondiente

a la variable original y agrega un 0 en las demás columnas. De esta manera, algoritmos como los Clasificadores pueden analizar las relaciones entre estas variables cualitativas con las cuantitativas.

2.2.2 Estandarización

En Machine Learning es indispensable normalizar los datos, ya que éstos pueden venir muy segregados por cada variable, además que esa segregación produce que los algoritmos consideren variables más importantes por la dimensión de los datos, cuando en la realidad, sus valores normalizados pueden ser tan representativos como otras variables.

$$X_{stand} = \frac{X - Mean(X)}{StdDeviation(X)} \tag{2.1}$$

El modelo de escalación llamado **Estandarización**, se puede describir en la Ecuación 2.1. Como se puede notar, se toma cada valor X de la columna o variable y se le resta el valor de la Mediana del conjunto de datos de dicha columna, luego se divide entre la Desviación Estándard de la misma. Una vez implementada, los datos de una columna o variable se transforman a un rango de -3 a 3. Ésto se repite para todas las posibles columnas o variables que hayan haciendo que todos los datos sean igualmente representativos a nivel numérico (Eremenko et al. (2019)).

Es importante destacar que éste método se puede utilizar para cualquier tipo de datos, no hace falta que se acoplen a algún tipo de distribución, como es el caso del método de Normalización, donde es necesario que los datos se muestren como una distribución normal, por lo que su uso es muy versátil (Eremenko et al. (2019)).

2.2.3 Accuracy, Precision, Recall y F1-Score

- **Accuracy:** El accuracy es la división del numero de Verdaderos Positivos (T_p) entre el total de muestras (N). Se puede observar descrita en la Ecuación 2.2.

$$A = \frac{T_p}{N} \tag{2.2}$$

- **Precision:** La precisión se define como al número de Verdaderos Positivos (T_p) entre la suma del número de Verdaderos Positivos y el numero de Falsos positivos (F_p). Se puede observar descrita en la Ecuación 2.3. Se puede decir que Precision es la habilidad de un clasificador de no identificar como positivo una muestra que es negativa en realidad.

$$P = \frac{T_p}{T_p + F_p} \tag{2.3}$$

- **Recall:** El recall se define como al número de Verdaderos Positivos (T_p) entre la suma del número de Verdaderos Positivos y el numero de Falsos negativos (F_n). Se puede observar descrita en la Ecuación 2.4. Se puede decir que Recall es la habilidad de un clasificador de encontrar todas las muestras que sean positivas.

$$P = \frac{T_p}{T_p + F_n} \tag{2.4}$$

- **F1-Score:** El F1-Score se define como la media armónica entre la Precision y el Recall. Se puede observar descrita en la Ecuación 2.5.

$$P = 2\frac{PR}{P + R} \tag{2.5}$$

A partir de la información anterior, se puede decir que un sistema con un Recall alto pero una Precision baja va a retornar muchos resultados, pero la mayoría de ellos habrán sido clasificados incorrectamente al compararlos con los datos de entrenamiento. Caso contrario, si en un sistema con una Precision alta pero un Recall bajo, va a devolver muy pocos resultado, pero la mayoría de los datos predichos fueron los correctos. Se puede decir que en un sistema ideal, se espera que haya un alto Recall y un alto Precision, para que se puedan tener muchos datos y la mayoría sean los correctos (SciKit-Learn-Lib (2020)).

2.3. Librerías y métodos

2.3.1 Librería Scikit-Learn

La librería Scikit-Learn es un módulo de Python el cual integra una gran variedad de algoritmos de Machine Learning considerados básicos, para la solución de problemas supervisados y no supervisados de mediana escala. Este paquete de funcionalidades permite llevar el Machine Learning a personas no especializadas en el campo, haciendo uso de un nivel de lenguaje de alto nivel y sencillo para todos.

Se enfatiza en la facilidad de su uso, desempeño, documentación y consistencia. Las dependencias que ella muestra son mínimas y es de libre acceso, bajo una licencia BSD simplificada, alentando su uso para actividades tanto

académicas como comerciales.

Todos los códigos fuente, archivos binarios y documentacion pueden ser descargados desde `http://scikit-learn.sourceforge.net` y también a través de supágina web `https://scikit-learn.org/stable/index.html` (Pedregosa et al. (2011)).

2.3.2 Grado de Libertad: Random State

Para todos los algoritmos implementados de la librería Scikit-Learn, existe una variable interna llamada **Random State**. Ésta controla la aleatoriedad de del estimador usado. Las columnas o variables de los datos, llamadas *Features*, son siempre permutadas de manera aleatoria en cada separación de la data. El algoritmo tiene una opción para definir el total de columnas a usar, con el fin de obtener la mejor separación de la data. Por tanto, siempre que se elijan menos columnas que el total de columnas disponibles, el algoritmo va a seleccionar de manera aleatoria cuales de esas columnas son las importantes hasta encontrar la mejor forma de separar los datos. Sin embargo, la mejor elección va a variar entre cada corrida. Por tanto, para obtener un resultado determinístico durante la separación de los datos y tener resultados reproducibles, se debe utilizar un dato numérico fijo (SciKit-Learn-DecisionTree (2020), SciKit-Learn-Glosary (2020)).

2.3.3 Método: FeatureImportance

El método **FeatureImportance** es una funcionalidad que permite identificar, a través de porcentajes, la importancia de las columnas de un set de datos. La librería que la contiene es Yellowbrick la cual es una extensión de Scikit-Learn y se enfoca en el análisis visual y herramientas de diagnóstico, lo cual ayuda a evaluar de mejor manera el rendimiento, estabilidad y predictivilidad de los modelos de machine learning y ayuda al diagnóstico de los problemas a través del flujo de trabajo (Bengfort et al. (2018)).

A pesar de que no todos los algoritmos pueden utilizar éste método, la escogencia de las columnas o *Features* más importantes implica seleccionar un mínimo de columnas requeridas para producir un modelo válido porque, entre más columnas contenga un modelo, más complejo se vuelve (y más dispersión de datos contiene), y por ende, el modelo se vuelve más sensible a errores debido a la varianza. Una manera común de eliminación de columnas es mediante la descripción de su importancia relativa al modelo, luego eliminar las columnas más débiles o combinaciones de columnas y reevaluar para verificar

si el modelo se comporta de mejor manera a través de una validación cruzada (Bengfort et al. (2018)).

2.4. Machine Learning por Clasificación

Es una técnica utilizada para identificar a qué clase o grupo de variables independientes, una variable dependiente pertenece. Los algoritmos que se usan para la Clasificación son de aprendizaje supervisado (Supervised Learning), en el cual, los algoritmos aprenden a partir de datos con grupos bien identificados. El dato nuevo, una vez que los algoritmos determinan a qué grupo pertenece, es asignado a éste de acuerdo con los patrones que lo asemejan con dicho grupo (Shetty, B. (2019)).

2.4.1 Algoritmo k-Nearest Neighbors (KNN)

El algoritmo k-Nearest Neighbour kNN se basa en que las diferentes instancias de una hoja de datos suelen estar bastante cerca a otras instancias que mantienen propiedades similares (Maglogiannis (2007)). Este algoritmo utiliza un método no parámetrico, lo cual significa que no hace ninguna suposición acerca de los datos, lo que lo hace muy efectivo ya que le permite manejar datos más realistas. También se considera un algoritmo vago, o sea, que memoriza la hoja de datos de entrenamiento en lugar de aprender de la hoja de datos alguna función que le ayude a discriminar. A su vez, este algoritmo puede ser usado para resolver tanto problemas de clasificación como regresiones (Lateef (2020)).

Dado un dato llamado q_i sin clasificar, y dados k valores vecinos previamente clasificados, podemos determinar que la clasificación de q_i según la k cantidad de vecinos con la misma característica que éste tenga. El método kNN localiza entonces a las k instancias más cercanas a nuestro valor sin clasificar y determina su clase al identificar la característica más común dentro de los vecinos (Maglogiannis (2007)). Como se puede observar en el ejemplo de la figura 2.4, habiendo 2 clases, el nuevo elemento que tiene forma de estrella, buscan en un radio determinado a las k instancias más cercanas, dado que los datos de la clase B son mayoría, este dato nuevo también se considera clase B.

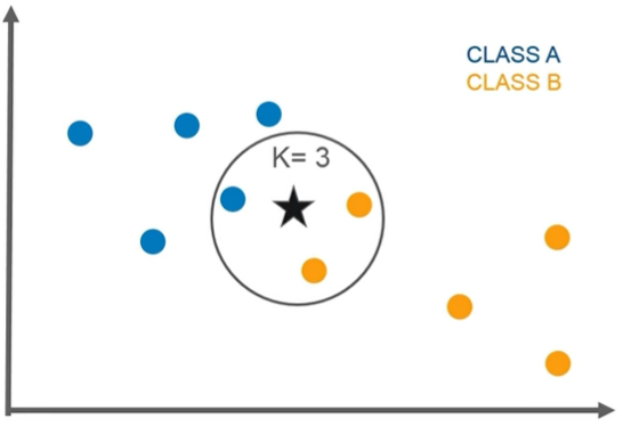


Figura 2.4: Ejemplo del Algoritmo KNN para un k=3. Nota: tomado de Waseem (2020)

Podemos decir que, en general, cada instancia es considerada como puntos en un espacio d-dimensional donde cada una de las d-dimensiones corresponde a una de las d-características que describen a dicha instancia. La posición absoluta de dichas instancias dentro de este espacio no es tan significativo como la distancia relativa entre las instancias. Esta distancia relativa se mide utilizando una valor medible, el cual debe ser el valor mínimo de distancia entre dos instancias cuya clasificación es similar(Maglogiannis (2007)).

Cada valor en ese espacio d-dimensional se puede expresar como un d-vector de coordenadas $p = (p_1, p_2,...,p_n)$ y la distancia entre dos puntos suele minimizarse haciendo uso de la distancia Euclideana (Ec. 2.6)(Kuang y Zhao (2009)):

$$dist(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{2.6}$$

El poder determinar el valor de k es de suma importancia ya que éste decide cuantos vecinos van a ser utilizados para determinar la clasificación de nuestros datos en el algoritmo kNN. La escogencia del mismo tiene un impacto muy significativo en el desempeño del algoritmo. Un valor de k grande, reduce el impacto de la varianza causada por algún valor de error, pero corre con el riesgo de ignorar patrones pequeños pero de suma importancia. Debe haber siempre un balance entre sobre-compensación y sub-compensación. Se puede sugerir que el valor de k sea igual a la raíz cuadrada del total de ob-

servaciones de la hoja de datos con que se va a entrenar el sistema (Zhang (2007)) y además, ser impar para evitar un empate entre clases (Mora (2018)).

A pesar de ser un algoritmo simple de usar, existen varias desventajas en el desempeño que se muestran a continuación (Kuang y Zhao (2009) y Maglogiannis (2007)):

- Según la dimensión de la data, esto va a extender la complejidad del algoritmo proporcionalmente, por lo que la complejidad computacional y el tiempo de operación van a crecer demasiado.
- El punto anterior también implica que guardar los resultados del algoritmo van a implicar mucho uso de memoria.
- La escogencia del valor de k no es algo fijo y estudiado, sino que se debe probar a partir de la data que se tiene.

2.4.2 Linear Discriminant Analysis (LDA)

A grandes rasgos, es un algoritmo en el cual, busca la reducción de las dimensiones de la data que se están utilizando (figura 2.5) pero se concentra también en que la separación de las categorías que se crean sea la máxima posible, con el fin de obtener la mejor información de las mismas (StatQuest (2016)).

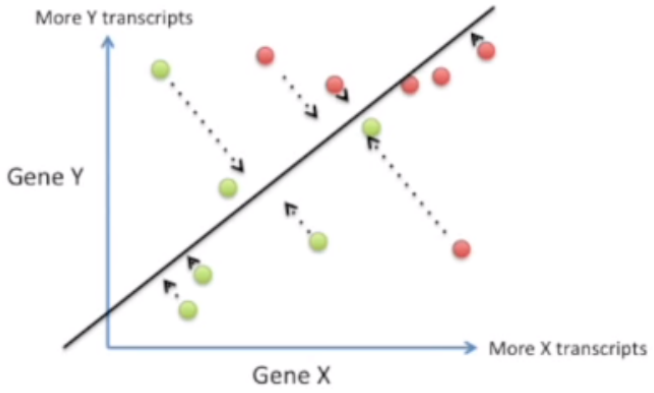


Figura 2.5: Cambio de dimensión de 2D a 1D usando LDA. Nota: tomado de StatQuest (2016).

Para esto, toma los datos de dos categorías, por ejemplo, distribuidos en los ejes o dimensiones que tenga (2D, 3D ... nD) y los proyecta en nuevos ejes de menor dimensión, por ejemplo de 2D a 1D, tomando en cuenta que

la separación entre la mediana de las categorías sea la máxima posible y a su vez, que la dispersión entre los datos (*"scatter"*) de una misma categoría sea la menor posible (StatQuest (2016)) (Figura 2.6).

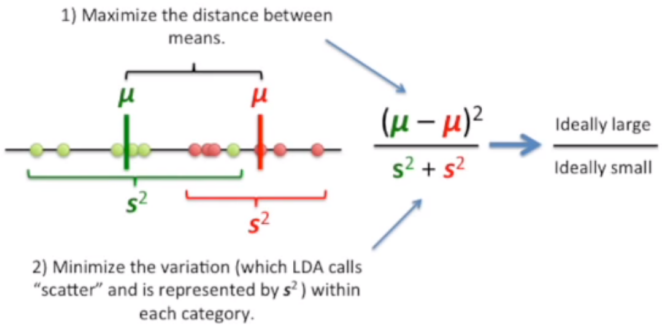


Figura 2.6: Criterios a tomar en cuenta al hacer un LDA. Nota: tomado de StatQuest (2016).

Al hacerse una correcta separación entre las clases, se logra una reducción alta en el costo computacional y a su vez evitar la llamada "maldición de dimensionalidad"(representación de dimensiones que en el mundo 3D no se pueden visualizar)(Raschka (2014)).

Matemáticamente hablando, el LDA busa reducir la dimensión de una serie de datos de d-dimensiones a través de una transformación lineal a un subespacio de dimensión k, donde $k<d$ y a su vez $k<c$-1, donde c corresponde a la cantidad de clases o datos que hay (Mora (2018)). En el ejemplo de la figura 2.5, k sería 1, d correspondería a 2, por la cantidad de ejes, y c sería 2 porque hay 2 categorías de datos.

Para saber cuál valor de k es el más óptimo para usar, se calculan los vectores propios de la data original. Cada uno de estos vectores propios va a estar directamente asociado a un valor propio, el cual indica la magnitud del vector. Si todos los valores propios calculados tienen un valor similar, esto es un muy buen indicador de que la data ya se encuentra en un estado espacial óptimo, caso contrario, una diferencia entre valores propios alta nos sugiere que los valores propios de menor valor podrían ser eliminados y utilizar solamente los valores propios de magnitud alta para crear un nuevo subespacio (Mora (2018)).

2.4.3 Support Vector Machine (SVM)

Este algoritmo busca la categorizar un grupo de datos sin clasificar, construyendo, a partir de datos ya conocidos o de entrenamiento, un o varios hiperplanos que mejor separen dichos datos. Estos hiperplanos se van a convertir en los límites de decisión y la intersección entre ellos indican a cual categoría va a pertenecer un nuevo dato que se quiera categorizar. Para hacerlo, aumenta la dimensión de los datos a partir de una función, facilitando la visualización del límite que separa los datos (Mora (2018)).

Primero se va a explicar varios conceptos que nos van a ayudar a entender la forma de buscar los hiperplanos (Figura 2.7):

- **Margen:** es la distancia más corta que hay entre un dato y un límite.
- **Vector Soporte:** es el dato más próximo al límite que existe.

Esto quiere decir que el mejor hiperplano que se puede escoger es aquel que maximiza los márgenes entre las categorías(Mora (2018)). Entiéndase por hiperplano a todo aquél límite que separe varias categorías presentes en un espacio n-dimensional.

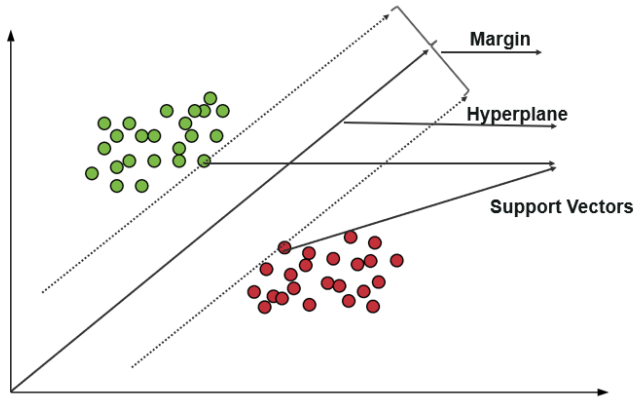
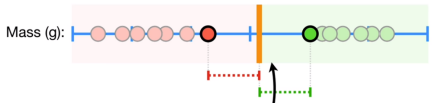


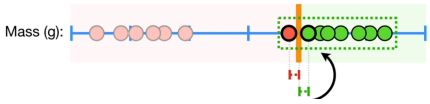
Figura 2.7: Conceptos básicos para SVM. Nota: tomado de Waseem (2020)

Es importante entender que cuando el límite entre dos categorías se coloca exactamente a la mitad de la distancia total entre dos vectores soporte, se va a crear un margen el cual es el más grande que puede haber, y a este se le llama **Margen Clasificador Máximo (MCM)**(Figura 2.8a). Este tipo de clasificador es muy sensitivo ya que si en los datos de entrenamiento, un elemento de una clase está muy cerca de otro, el margen entre las clases será muy pequeño,

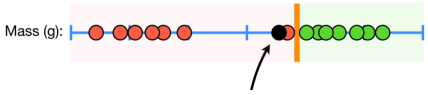
haciendo que la clasificación sea muy pobre ya que la varianza va a aumentar (Figura 2.8b) y los datos nuevos podrían quedar mal clasificados, a pesar de que las características de este dato sean más similares a la clase que se encuentre más cerca de ella (Figura 2.8c). Es posible que este vector soporte tan cercano a la otra clase sea inclusive un dato mal clasificado dentro de los datos de entrenamiento, por lo que usar este tipo de clasificadores no es lo más óptimo, ya que no permite datos mal clasificados(StatQuest (2019)).



(a) Margen Clasificador Máximo



(b) MCM con datos muy juntos



(c) Mala clasificación de un dato debido al uso del MCM

Figura 2.8: Uso del MCM como clasificador. Nota: tomado de StatQuest (2019)

Para dar solución a malas clasificaciones de los datos de entrenamiento, es preferible utilizar el llamado **Margen Suave (MS)**. Este clasificador es un clasificador que permite malas clasificaciones en los datos de entrenamiento y se calcula utilizando una validación cruzada, o sea, se copara uno a uno los posibles MS que haya hasta que exista uno donde las clasificaciones malas sea la menor. Al usar un clasificador MS, podemos decir que estamos usando un **Clasificador de Vector de Soporte (CVS)** (StatQuest (2019)). En el ejemplo de la figura 2.9, podemos apreciar que a partir la validación cruzada, se determinó un MS el cual permitió 2 nuevos datos bien clasificados y permitió 1 dato mal clasificado.

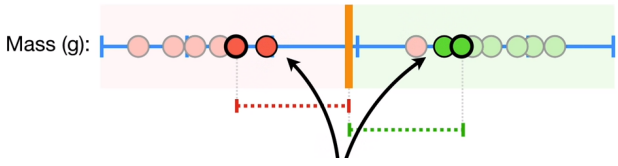


Figura 2.9: Margen Suave determinado a partir de la validación cruzada. Nota: tomado de StatQuest (2019)

Ahora bien, SVM se destaca cuando los datos de varias categorías se en-

cuentran mezclados y no es posible separarlos con una simple línea (hiperplano), ya que sea donde sea que se coloque el CVS, muchos datos van a quedar mal clasificados (Figura 2.10).

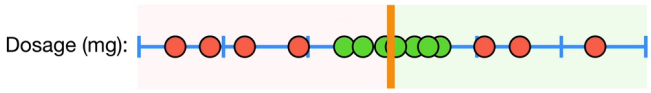
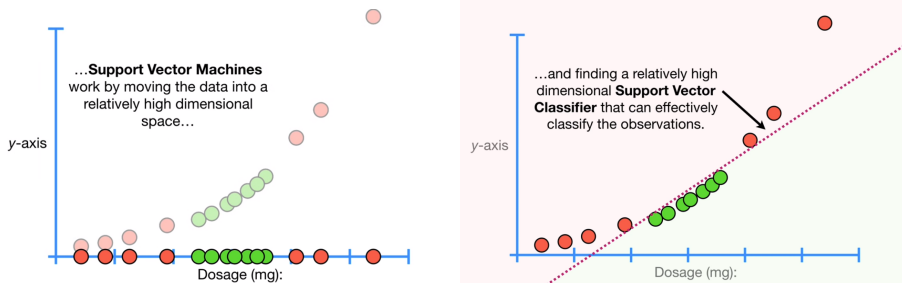


Figura 2.10: Datos mezclados dificultando determinar un MS adecuado. Nota: tomado de StatQuest (2019)

SVM busca aumentar la dimensión de los datos de entrenamiento, introduciendo un nuevo eje. Para hacer esto, proyecta los datos a este nuevo eje a partir de una **función Kernel**. Con ella, busca la relación de cada par de observaciones y determina cual es el mejor CVS para ese set de datos(Mora (2018) y Maglogiannis (2007)). Por ejemplo, haciendo uso de la función Kernel Polinomial, podemos determinar que la mejor forma de separar los datos de la figura 2.10 es a partir de x^2, como se muestra en la figura 2.11a. A partir de esta función, se proyectan los datos a un nuevo eje, facilitando así la determinación del hiperplano que mejor separa los datos, tal y como se muestra en la figura 2.11b (StatQuest (2019) y Maglogiannis (2007)).



(a) Datos proyectados usando la función x^2 (b) Hiperplano nuevo que separa los datos

Figura 2.11: Uso de la función Kernel. Nota: tomado de StatQuest (2019)

Dentro de las ventajas que tiene este algoritmo, podemos destaca:

- La cantidad de vectores soporte es poca en relación con la cantidad de datos que hay y por esta razón podemos decir que las SVM son excelentes para manejar rutinas de aprendizaje cuando la cantidad de categorías es grande con respecto a los datos de entrenamiento que se usaron (Maglogiannis (2007)).

- Es efectivo cuando se trabaja con datos con dimensiones altas (Mora (2018)).

- El uso de funciones Kernel permite creación de límites complejos (Mora (2018)).

Así mismo, existen desventajas como:

- Debido a que no existe una forma de determinar cual función Kernel es la mejor para cuales tipos de datos, el entrenamiento del algoritmo tiende a ser lento, ya que se deben probar varias funciones hasta dar con la que mejor se adecúe a nuestro problema (Maglogiannis (2007)).

- No brinda estimaciones de probabilidad, por lo que simplemente sirve para clasificar, no estimar (Waseem (2020) y Mora (2018)).

2.4.4 Naive Bayes

Este algoritmo se basa en el Teorema de Bayes. Para explicar este teorema, podemos imaginar dos maquinas dos máquinas que producen llaves inglesas, este teorema nos puede ayudar a determinar la posibilidad de, dadas ciertas llaves defectuosas, éstas provengan de una máquina en común. Dependiendo de la pregunta que busquemos, probabilidad de que las llaves buenas sean de la maquina 1, o probabilidad de que las llaves malas sean de la maquina dos, el Teorema de Bayes nos ayuda a resolver este tipo de preguntas(Eremenko et al. (2019)).

Para determinar esta probabilidad se usa la siguiente ecuación:

$$\mathrm{P}(A|B) = \frac{\mathrm{P}(B|A) * \mathrm{A}(Y)}{\mathrm{P}(B)} \tag{2.7}$$

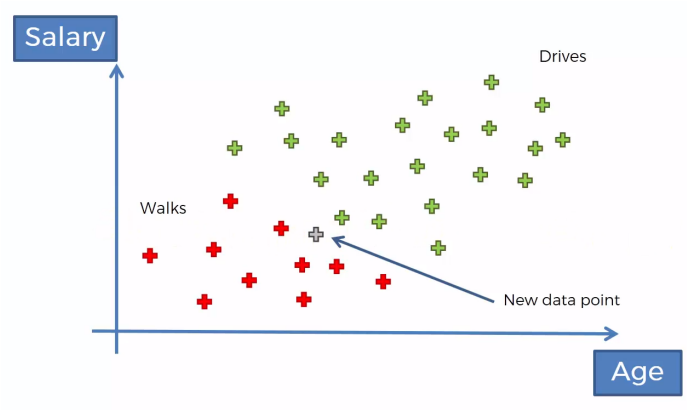


Figura 2.12: Algoritmo Naive Bayes en un grupo de datos . Nota: tomado de Eremenko et al. (2019)

Dado un nuevo punto en un grupo de datos (Figura 2.12), el algoritmo hace uso del Teorema de Bayes y calcula cual es la probabilidad de que dado ese punto, este vaya a estar en el grupo donde los datos son rojos, luego va a calcular cual es la probabilidad de que dado ese punto, vaya a estar en el grupo donde los datos son verdes y luego se comparan ambas probabilidades para identificar cual es la mayor, para así asignarle a este nuevo dato, el grupo al que pertenece (Eremenko et al. (2019)).

2.4.5 Decision Tree

Este algoritmo nos va a ayudar a predecir datos categóricos, por ejemplo, si un partido se gana o se pierde, no nos va a brindar valores reales numéricos (como lo hacen los Regression Trees). Para lograr esta predicción, dado un grupo de datos, el algoritmo comienza por separar las diferentes categorías de tal forma que cada una de ellas, tenga la mayor cantidad de datos de cierta categoría en alguna de las separaciones, minimizando la entropía entre ellas (Eremenko et al. (2019)). Podemos observar esto en la Figura 2.13.

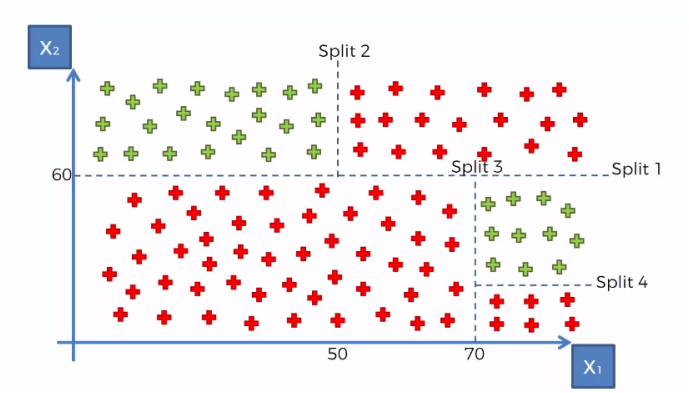


Figura 2.13: Separación hecha por el algoritmo Decision Tree . Nota: tomado de Eremenko et al. (2019)

Luego, el algoritmo crea un árbol (2.15) el cual nos caracteriza cada uno de las separaciones que hizo, para que, al llegar un nuevo dato, podamos ir categorizándolo según los valores o características por las cuales se hicieron las separaciones. De tal manera que, en el ejemplo,si un nuevo dato se encuentra a menor valor que X_2 y menor valor que X_1, podemos categorizarlo como rojo (Eremenko et al. (2019)).

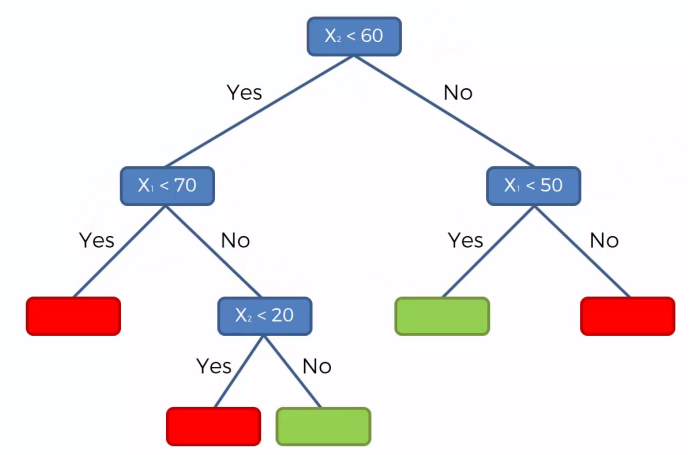


Figura 2.14: Árbol creado por el algoritmo Decision Tree . Nota: tomado de Eremenko et al. (2019)

2.4.6 Random Forest

Este algoritmo se basa en los Decision Trees, donde, en lugar de usar un solo árbol de decisión para determinar la clasificación, se construyen varios árboles por el cual los datos van a pasar para determinar estas nuevas predicciones. A esto se le llama aprendizaje de ensamble, cuando se crea un nuevo algoritmo a partir de varios otros (Eremenko et al. (2019)).

Para crear los árboles, se toma cierta parte de los datos que tengo y se crea un árbol de decisión. Luego, determino cuantos arboles quiero construir en total y repito lo que hice anteriormente para con el resto de las partes de los datos. Finalmente, para ejecutar la predicción de un dato nuevo, hago que el dato pase por todos los arboles de decisión y la predicción final será la cual la mayoría de los arboles me predijo como posible (Eremenko et al. (2019)).

Este algoritmo es el utilizado en el hardware Kinnect, desarrollado por Microsoft para la consola XBox 360, el cual, por medio del algoritmo Random Forest, permite determinar en tiempo real, la posición y la profundidad de un cuerpo frente a las cámaras de profundidad presentes en el aparato(Eremenko et al. (2019) y Shotton et al. (2011)).

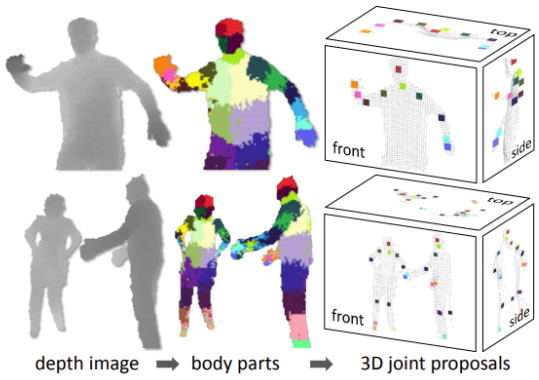


Figura 2.15: Captación de profunidad y posición del Kinnect. Nota: tomado de Shotton et al. (2011)

2.5. Machine Learning por Clustering

Según Fung (2001) y Abu Abbas (2008), la técnica de clustering se basa en agrupar todos aquellos datos cuyas características sean similares y dividirlos en grupos o clusters, por ente, estos datos entre sí van a ser similares en un cluster y a su vez diferentes de los pertenecientes a otros clusters. Podríamos

por ende decir, que los elementos de un cluster son aquellos que naturalmente se relacionan entre sí.

2.5.1 K-Means Clustering

Como se mencionó, el clustering ayuda a identificar diferentes grupos de datos según sus características, a partir de las particularidades entre ellos (Eremenko et al. (2019)). En la Figura 2.16, podemos observar un set de datos antes y después de procesar los datos a través de K-Means clustering.

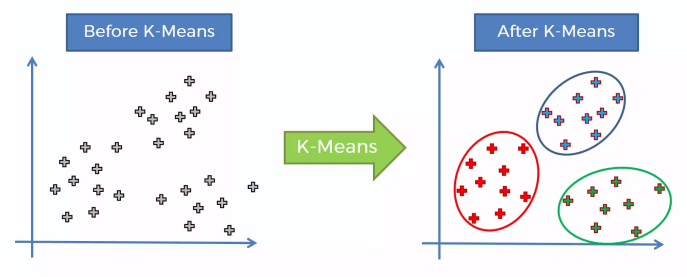


Figura 2.16: Antes y después de ejecutar K-Means Clustering. Nota: tomado de Eremenko et al. (2019)

Para determinar los clusters, se toma un numero K de clusters, donde K se calcula utilizando el método llamado **Suma de Cuadrados de Cluster** (*Within Cluster Sum of Squares* en inglés), junto con el **Método del Codo**, de la siguiente manera: para hacer la Suma de Cuadrados de cada Cluster, se toman los puntos dentro de cada cluster y suma la distancia al centro de cada uno al cuadrado y luego se suman todos los clusters (Eremenko et al. (2019)). El resultado es la robustez de todos los cluster, que entre más pequeño, menos distancia del centro a los puntos de cada cluster y con ésto más robusto. Su representación matemática se puede ver en la Ecuación 2.8.

$$WCSS = \sum_{P_iC1} (distance(P_i, C1)^2) + \sum_{P_iC2} (distance(P_i, C2)^2) + ... + \sum_{P_iCn} (distance(P_i, Cn)^2) \tag{2.8}$$

Este valor WCSS va a variar según la cantidad de clusters que haya. Por tanto, se hace un barrido en la cantidad de clusters y se grafican. La gráfica se llama Método del Codo, ya que la variación entre las distancias de los puntos a los clusters, cada vez es más pequeña y la gráfica va a mostrar el cambio abrupto de estos valores entre mayor cantidad de clusters se tengan. Un ejemplo se puede observar en la Figura 2.17:

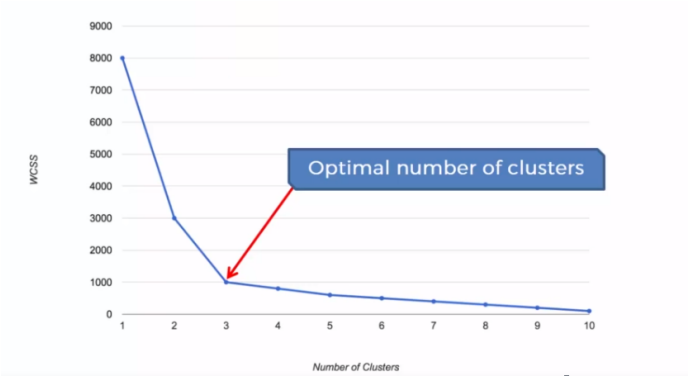
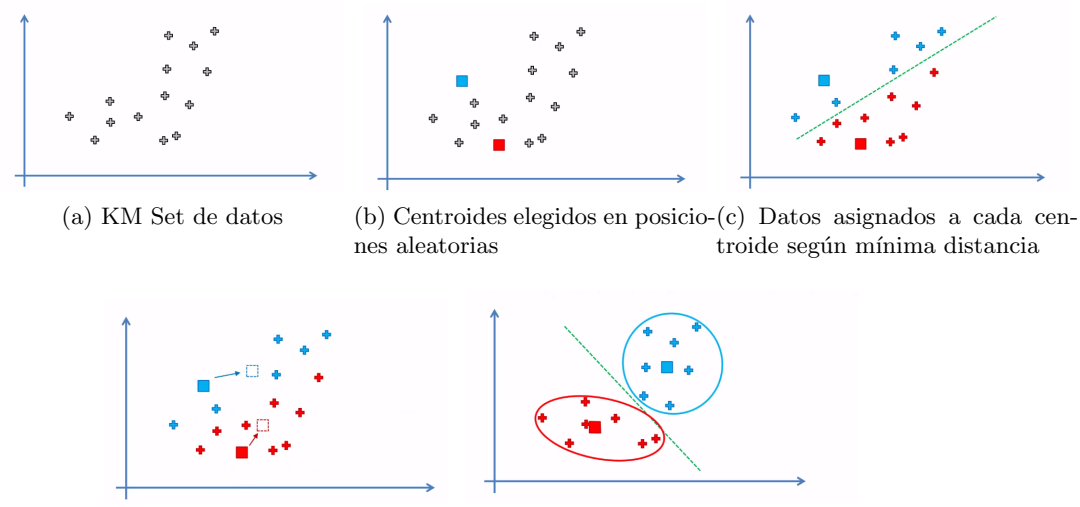


Figura 2.17: Método del Codo. Nota: tomado de Eremenko et al. (2019)

Seguidamente, se declaran una cantidad K de centroides, que representan un primer centroide para cada grupo de datos. La posición inicial de estos K centroides es al azar (Figura 2.18a y 2.18b). Se procede a medir la distancia Euclideana (linea recta) entre el centroide y cada uno de los puntos de nuestro set de datos para así asignar dichos puntos al cluster que esté mas cerca (2.18c) (StatQuest (2018) y Eremenko et al. (2019)).



(a) KM Set de datos

(b) Centroides elegidos en posiciones aleatorias

(c) Datos asignados a cada centroide según mínima distancia

(d) Calculo y modificación de nuevos centroides

(e) Cluster final

Figura 2.18: Algoritmo detallado K-Means Clustering. Nota: tomado de Eremenko et al. (2019)

Una vez determinado esto, se procede a crear un centroide nuevo determi-

nado a partir de la media de todos los puntos de cada cluster. Nuevamente se procede a medir la distancia entre cada punto y los nuevos centroides calculados (2.18d). Se repite el proceso hasta que el centroide no se mueva (2.18e) (StatQuest (2018) y Eremenko et al. (2019)).

Dado que el centroide se elije al azar, los clusters pueden variar si elijo una posición de centroide inicial diferente cada vez que hago el cluster. Para evitar esto, se calcula la suma de la variación entre los centroides creados, se guardan y se ejecuta un nuevo calculo de clusters desde cero. Esto se repite por varios ciclos. Finalmente se comparan los resultados de variabilidad entre cada ciclo hasta que haya uno donde la variabilidad de cada cluster sea equitativa (Figura 2.19)(StatQuest (2018) y Eremenko et al. (2019)). Este método se ve limitado debido a que solo calcula clusters circulares, ya que la distancia se calcula como la distancia Euclideana (StatQuest (2018)).

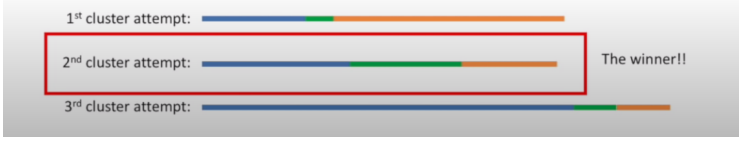
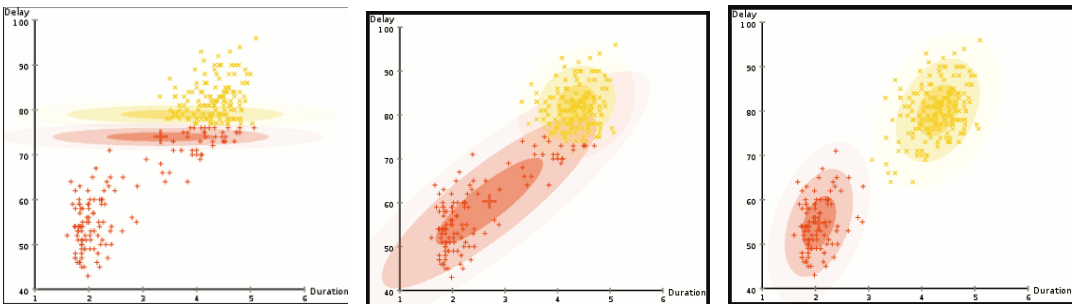


Figura 2.19: Calculo de la variabilidad entre cada ciclo de K-Means Clustering. Nota: tomado de StatQuest (2018)

2.5.2 Gaussian Mixture Model (GMM)

Dado que el clustering a través del K-Means tiene como limitante el hecho de que sólamente encuentra clusters redondos, el clustering a partir de Gaussian Mixture Model se hacea travéz de una distribución Gausseana, esto quiere decir que el cluster, aparte de tener una media (Means), se le va a asignar una una varianza. Esto permite obtener más información de la distribución que solamente con la media y me va a facilitar la creación de clusters con diferentes formas. La varianza que se le da al cluster puede ser en varias dimensiones, además permite determinar la probabilidad de la data que esté en un cluster y esto facilita la escogencia de la variable K. Ahora, en vez de calcular la distancia entre el centroide (media) y el punto, ahora calculamos la probabilidad de ese punto dado ese centroide(Boyd-Graber (2018)).



(a) Inicialización de distribuciones (b) Datos asignados a nuevo cluster dada su probabilidad (c) Clusters finales

Figura 2.20: Algoritmo detallado GMM Clustering. Nota: tomado de Seif (2020)

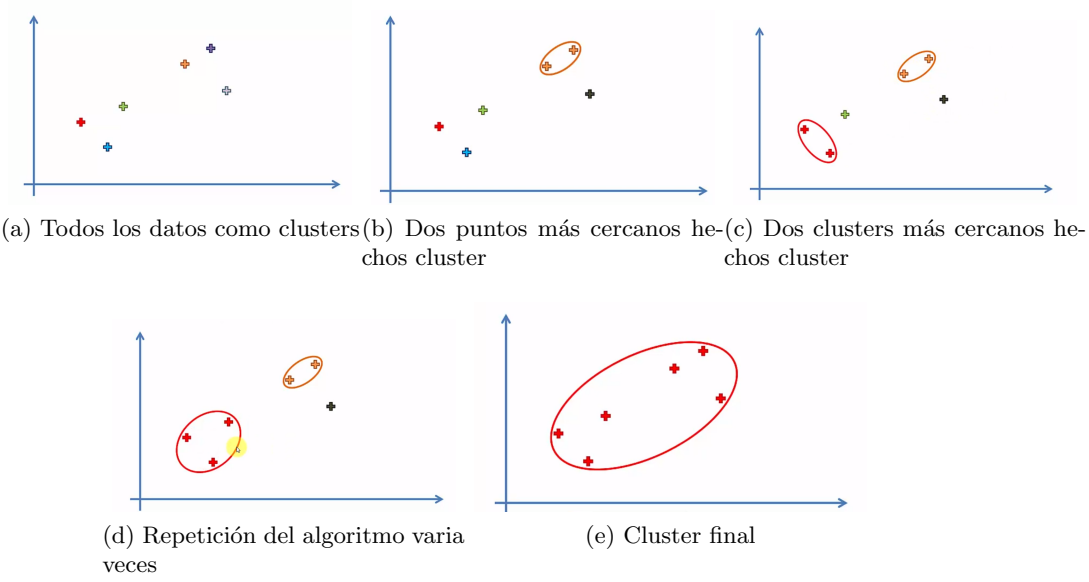
Para la ejecución del algoritmo, primero se toma una cantidad de clusters K (de igual manera que se elige en K-Means) y se inicializa las distribuciones gausseanas de manera aleatoria para cada cluster (Figura 2.20a). Seguidamente se calcula la probabilidad de que cada punto pertenezca a ese específico cluster. Entre más cerca esté el punto al centro del cluster, más probable va a ser que pertenezca a éste (esto es evidente ya que así se comporta una distribución gausseana). A partir de estas probabilidades, se crean un nuevo set de parámetros para la distribución gauseana, de tal forma que las probabilidades se maximicen para todos los puntos dentro de un cluster (Figura 2.20b). Estos parámetros se calculan usando la suma de los pesos de la posición de cada punto, donde este peso es la probabilidad de que ese punto pertenezca a ese cluster. Esto se repite hasta que la distribución casi no cambie (Figura 2.20c) (Seif (2020)).

2.5.3 Hierarchical Clustering

Al igual que con los métodos anteriores, el Hierarchical Clustering pretende buscar la mejor manera de agrupar un set de datos de mamera que los grupos que se obtienen presentan similitudes entre ellos en sus características. Cabe destacar tambien, que el tipo de Hierarchical Cluster que se va a utilizar el el tipo aglomerativo, el cual va tomando cada punto como un cluster y va agregando poco a poco más puntos al cluster (Eremenko et al. (2019)).

Algo muy importante a destacar en este tipo de clustering es la distancia entre clusters. Parte del algoritmo necesita una medida de distancia para determinar si un punto o no pertenece a un cluster, es por eso que se debe

definir bien qué medida se quiere. Por ejemplo, la distancia entre dos clusters puede ser la distancia entre los dos puntos más cercanos de esos clusters, tambien pude ser la distancia entre los puntos más alejados de esos clusters, otra forma puede ser el promedio de las distancias de todos los puntos de un cluster a otro, o finalmente la distancia entre centroides. Sea cual sea la que se utilice, se tiene que tener claro cual se está usando, ya que dependiendo del tipo de datos que se esté usando, algunas medidas van a mejorar el resultado del clustering (Eremenko et al. (2019)).

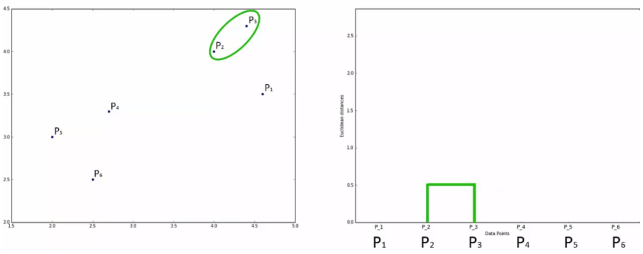


(a) Todos los datos como clusters (b) Dos puntos más cercanos hechos cluster (c) Dos clusters más cercanos hechos cluster

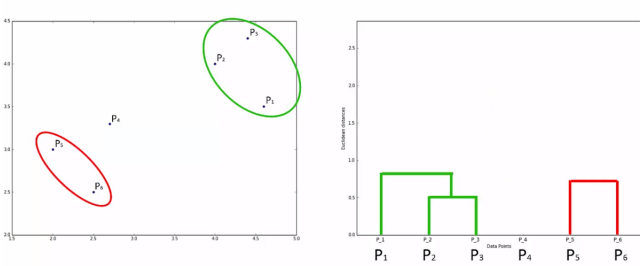
(d) Repetición del algoritmo varia veces (e) Cluster final

Figura 2.21: Algoritmo detallado Hierarchical Clustering. Nota: tomado de Eremenko et al. (2019)

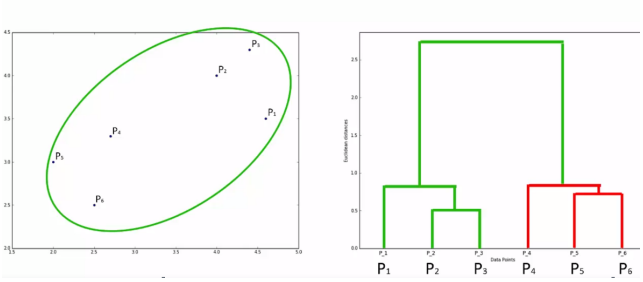
Para usar Hierarchical Clustering, se comienza por hacer que cada uno de los puntos en la gráfica, sea un cluster (Figura 2.21a). Seguidamente, se mide la distancia entre cada uno de los clusters y aquellos cuya medida sea la menor, se unen para formar un nuevo cluster (Figura 2.21b). Seguidamente, se toman las medidas entre todos los clusters de nuevo, y aquellos dos cuya distancia sea la menor, se unen para formar un nuevo cluster (Figura 2.21c. Este proceso se repite hasta que se tiene un único cluster con todos los puntos (Figuras 2.21d y 2.21e).



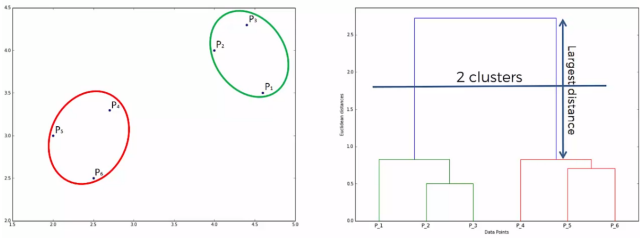
(a) Primer Cluster formado



(b) Clusters formados en una tercera iteración



(c) Dendograma completo



(d) Obtención de Clusters a partir del Dendograma

Figura 2.22: Algoritmo detallado Hierarchical Clustering. Nota: tomado de Eremenko et al. (2019)

Este algoritmo tiene la cualidad de que guarda cada una de las iteraciones hechas por el algoritmo en un gráfico llamado **Dendograma**, el cual nos va a

ayudar a determinar cuánta es la cantidad óptima de clusters para el set de datos. Trabaja como la memoria del algoritmo ya que cada combinación nueva que haga el algoritmo, se va a representar en forma de esquema de la manera mostrada en la Figura 2.22c. La línea horizontal que conecta los dos puntos representa la unión de esos puntos (o clusters) para formar un nuevo cluster y la línea vertical que los toca representa la distancia entre los puntos (o clusters según sea el caso), la cual significa qué tan similares están entre sí, entre más baja, más cerca están, por tanto, más relacionados (Figuras 2.22a y 2.22b). Una vez obtenido el dendograma, se mide la distancia más larga posible entre todas las líneas verticales, y a partir de ahí, se traza un umbral, donde la cantidad de veces que la linea creada corte a las lineas verticales, esa será la cantidad óptima de clusters para ese set de datos (Figura 2.22d)(Eremenko et al. (2019)).

2.6. Machine Learning por Reglas de Asociación

Según Mata et al. (2002) y Khurana y Sharma (2013), la búsqueda de Reglas de Asociación es una de las técnicas más importantes para la Minería de Datos y Machine Learning, la cual se basa en descubrir relaciones entre los atributos, extraer correlaciones interesantes y patrones que tengan cierta frecuencia en una grupo o base de datos. Estas reglas no van a extraer una preferencia individual específica, sino, como ya se mencionó, se van a obtener las diferentes relaciones que haya entre los elementos conjuntos de cada una de las transacciones hechas (Garg (2020)). En nuestro caso, estas reglas nos van a indicar la posible relación entre los Clasificadores y los Clusters.

Para la obtención de dichas reglas, es necesario que los datos sean discretos (Mata et al. (2002), Born y Schmidt-Thieme (2004) y Ludl y Widmer (2000)), o sea, solo trabajan con datos categóricos, por lo que es necesario de técnicas de discretización para todas aquellas categorías numéricas dentro de nuestra base de datos. Dos de las técnicas más conocidas y comunes de usar, según Moreno et al. (2007) son la discretización en grupos de igual tamaño y discetización en grupos de igual frecuencia. El primero se basa en tomar todo el posible rango de valores y dividirlos en rangos de igual tamaño cada uno, mientras que el otro se basa en dividir los rangos de tal manera que la frecuencia de los datos, en esos rangos, sea aproximadamente la misma. Sin embargo, Moreno et al. (2007) describe y utiliza una forma la cual permite que los datos se muevan en un rango más acorde con su naturaleza y es por medio del Clustering, tomando cada categoría y aplicando el algoritmo de K-Means para la generación de la cantidad de rangos más óptima posible.

Cabe destacar que estas reglas son muy utilizadas en los análisis de compras de mercados, en los cuales se busca obtener que ciertos productos estén localizados en lugares estratégicos con el fin de, no solo reducir el tiempo de compra de los consumidores sino también recordarle al consumidor cuales productos podrían ser relevantes para él en su compra, lo cual ayuda a mejorar las ganancias del local (Garg (2020)).

Para todas las reglas de asociación, se puede pensar como una relación del tipo SI-ENTONCES, donde los *antecedentes* (SI) corresponden a los items que se encuentran frecuentemente comprados o presentes en los datos y los *consecuentes* (ENTONCES) son los itemes que se adquieren o vienen como consecuencia de los antecedentes (Keshari (2020)). En la regla de ejemplo que se observa en la Figura 2.23, podemos ver que el pan y la leche son dos items comúnmente comprados y que SI estos se encuentran juntos, ENTONCES como consecuencia, voy a comprar leche.

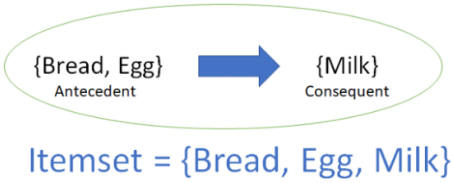


Figura 2.23: Ejemplo de una Regla de Asociacion, con sus Antecedentes y Consecuentes. Nota: tomado de Garg (2020)

Sin embargo, los algoritmos que generan estas reglas van a determinar un número enorme de ellas ya que encuentran todas las posibles relaciones entre cada uno de los items que tengamos en nuestra base de datos, por lo cual es necesario determinar una forma de medir cuales de todas las reglas son las más importantes o llamativas (Keshari (2020)). Para esto, se tiene 3 formas de medir las reglas de asociación ((Keshari (2020)) y Garg (2020)):

- **Support (Soporte)**: Esta medida nos indica la relación entre las transacciones que contienen uno o varios elementos y el total de transacciones hechas. Por ejemplo: si tengo dos transacciones, una con Pan y Mantequilla, la otra con Pan y Shampoo, podríamos asegurar es más frecuente comprar la primera opción que la segunda, por lo que la primera tiene más soporte que la otra.**AGREGAR LA FUNCION MATEMATICA**
- **Confidence (Confianza)**: Ésto nos indica que tan frecuente ocurre algún consecuente, dado que existe un antecedente en nuestro carrito de com-

pras. Por ejemplo, esto ayuda a responder, de todas las transacciones que contienen Pan, cuantas también contienen Mantequilla? podríamos decir que muchas, pero, de todas las transacciones que contienen Pan, cuantas también contienen Shampoo? posiblemente menos, por lo que el consecuente Mantequilla tiene una mayor confianza de ocurrencia frente al Shampoo.

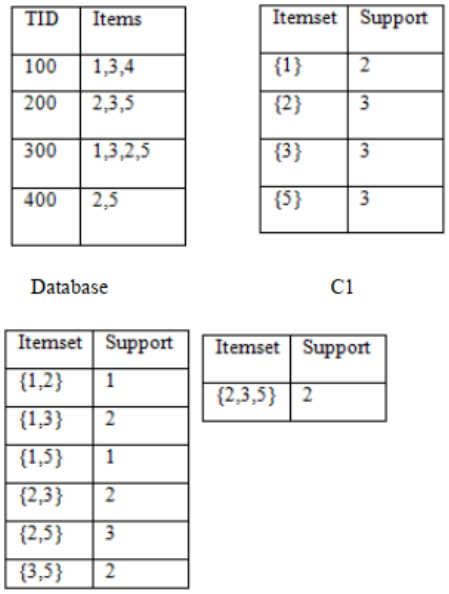
- **Lift (Levantamiento)**: El Lift es el incremento en la probabilidad de tener un consecuente en nuestro carrito con el conocimiento de que existe un antecedente sobre la probabilidad de tener un consecuente sin ningún antecedente. Se puede describir también como la fuerza de una regla sobre la ocurrencia aleatoria de un antecedente y un consecuente. Es qué tan fuerte es una regla. Por ejemplo, sabiendo que la probabilidad de tener Pan en nuestro carrito es de un 0.8, sin necesidad de que haya otro producto, pero la probabilidad de que compremos leche teniendo Shampoo en nuestro carrito es de un 0.7 nos indica que a nivel de relación, el hecho de que haya Shampoo en nuestro carrito reduce la probabilidad de tener Pan, por lo que el Lift es de 0.7/0.8=0.87. El hecho de que el Lift sea menor a 1 nos indica que tener un Shampoo en nuestro carrito, no incrementa las probabilidades de tener Pan en nuestro carrito, por lo que los valores de Lift mayores a 1 son los que se buscan, ya que implica una mayor asociación entre los productos analizados.

2.6.1 Algoritmo Apriori

Segun Keshari (2020), Garg (2020) y Eremenko et al. (2019), este algoritmo utiliza los sets de items frecuentes para genera las reglas de asociación, el cual se basa en el concepto que si un grupo de items son frecuentes, los subgrupos de ese mismo set de items tiene que también ser frecuente. La obtención de estas reglas se basa en los parametros de Support y Confidence, de acuerdo con Khurana y Sharma (2013), donde las reglas aceptadas son únicamente aquellas cuyo valor de Support y Confidence sean mayor a un límite dado por el usuario.

Este algoritmo está compuesto por dos grandes pasos. El primero es la generación de todos los sets de items que son más frecuentes en el cual simplemente cuenta la ocurrencia de cada item por separado, luego va determinando todas las combinaciones de 2 items y refleja su frecuencia, luego 3 y así sucesivamente hasta lograr todas las posibles combinaciones de los items siempre y cuando aparezcan un mínimo de veces dado por el usuario (Figura 2.24). El segundo paso es la generación de las reglas a partir del Confidence y Support del paso anterior (Khurana y Sharma (2013), Kumbhare y Chobe (2014) y

Garg (2020)).



TID	Items
100	1,3,4
200	2,3,5
300	1,3,2,5
400	2,5

Database

Itemset	Support
{1}	2
{2}	3
{3}	3
{5}	3

C1

Itemset	Support
{1,2}	1
{1,3}	2
{1,5}	1
{2,3}	2
{2,5}	3
{3,5}	2

Itemset	Support
{2,3,5}	2

Figura 2.24: Obtención de las frecuencias de todas las posibles combinaciones de los items y sub grupos. Nota: tomado de Khurana y Sharma (2013)

Se puede resumir los pasos de este algoritmo de la siguiente manera (Eremenko et al. (2019)):

- Se determina un minimo Support y Confidence.
- Se crean todos las posibles combinaciones de los items las cuales tengan un mayor Support (frecuencia) que el minimo Support dado.
- Tomar todas las reglas creadas de estos sub grupos hechos los cuales tengan un mayor nivel de Confidence al escogido.
- Ordenar todas las reglas con el lift de mayor a menor.

2.6.2 Algoritmo Frequent Pattern Growth (FP-Growth)

Para entender este algoritmo, hay que entender de manera correcta el algoritmo Apriori, ya que muchas de las técnicas se utilizan en ella. Éstos métodos vienen a solventar los dos problemas que tiene el algoritmo Apriori: la primera es que en cada ciclo que se ejecuta el algoritmo, las posibles reglas se tienen que volver a generar desde cero, lo que implica que en tiempo, dura mucho, y

la segunda es que se requiere constantemente estar revisando la base de datos original para su ejecución (Kumbhare y Chobe (2014) y Gupta (2020)).

Para dar solución, este algoritmo crea lo que se llama un FP-Tree (Árbol de Patrones Frecuentes), el cual revisa la base de datos únicamente dos veces. Primero, a partir de la base de datos original (2.25a) toma todos los items por separado y genera una lista de frecuencias de cada uno (2.25b), según las veces que aparezca en las diferentes transacciones, ordenadas de mayor a menor, descartando los items que no cumplan con la condición mínima de Support (2.25c). Luego vuelve a pasar por la base donde, a partir de las transacciones hechas, haciendo uso del orden previamente creado y eliminando los items que no cumplan con el Support (2.25d), crea el Árbol de Patrones Frecuentes (2.25e) (Kumbhare y Chobe (2014)).

Transaction ID	Items
T1	{E,K,M,N,O,Y}
T2	{D,E,K,N,O,Y}
T3	{A,E,K,M}
T4	{C,K,M,U,Y}
T5	{C,E,I,K,O,O}

(a) Base de datos con transacciones iniciales

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

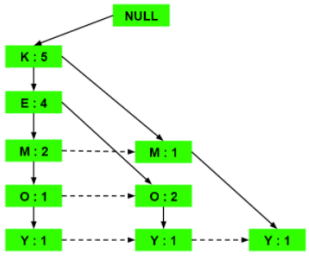
(b) Determinación de frecuencias por cada item

$$L = \{K : 5, E : 4, M : 3, O : 3, Y : 3\}$$

(c) Items ordenados de mayor a menor que cumplen con Support

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}

(d) Transacciones con items ordenados y filtrados



(e) Árbol de Patrones Frecuentes

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

(f) Base de Patrones Condicionales y Árbol de Patrones Frecuentes Condicionales

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 3>}
K	

(g) Patrones Frecuentes Generados

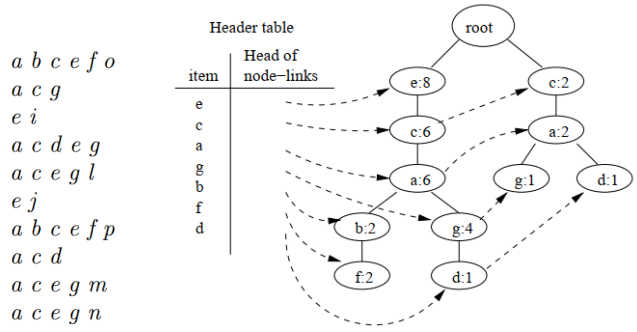
Figura 2.25: Algoritmo detallado FP-Growth. Nota: tomado de Gupta (2020)

Una vez creado el Árbol, se crea una Base de Patrones Condicionales la cual muestra los diferentes caminos por los que el Árbol se debe mover para llegar al item final. Estos ítems están ordenados de menor a mayor (2.25f). Luego, a partir de esta Base, se calculan un Árbol de Patrones Frecuentes Condicionales, el cual simplemente es la suma de los elementos comunes de todos los subgrupos de la Base de Patrones Condicionales y la cantidad de veces que aparece(2.25f). Finalmente, se generan los Patrones Frecuentes a partir de la combinancion del item en revisión junto con los grupos creados a partir del Árbol de Patrones Frecuentes Condicionales (2.25g).

2.6.3 Algoritmo Maximum Frequent Patterns (FP-Max)

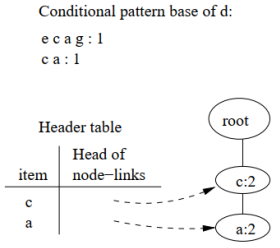
Este algoritmo es una variación del FP-Growth, en el cual, para crear el Árbol de Patrones Frecuentes, se basa en el concepto de Maximal Frequent Itemsets (Grupos de Items con Frecuencia Máxima). Se dice que un grupo de items es máximo si éste no pertenece a un grupo más grande que sea frecuente (Raschka (2020) y Grahne y Zhu (2014)). En otras palabras, si tenemos un grupo de items X el cual es frecuente, para que sea máximo, éste no debe ser un sub grupo de uno más grande y frecuente (Raschka (2020)). Esto quiere decir, que al construir el Árbol, el FP-Max únicamente creará una nueva rama del árbol si se determina que este patrón frecuente no está contenido en un patrón mucho más grande.

De igual forma que con el FP-Growth, FP-Max toma todas las transacciones (Figura 2.26a), y crea una lista de frecuencias de cada item, ordenadas de mayor a menor y omitiendo todas aquellas que no cumplan con el Support, luego crea un Árbol de Patrones Frecuente tal y como lo hace el FP-Growth (Figura 2.25b). Como se puede ver en la 2.26c, se muestra la forma de crear este árbol para el item d (Grahne y Zhu (2014)).

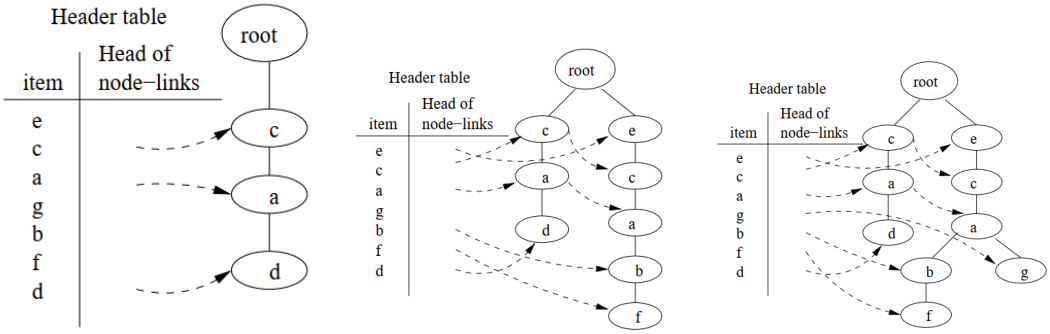


(a) Base de datos con transacciones iniciales

(b) Árbol de Patrones Frecuentes



(c) Determinación de la rama para el item d



(d) Rama de Árbol de Patrones de Frecuencias Máximas para d

(e) Rama de Árbol de Patrones de Frecuencias Máximas para f

(f) Árbol de Patrones de Frecuencias Máximas final

Figura 2.26: Algoritmo detallado FP-Max. Nota: tomado de Grahne y Zhu (2014)

Una vez creado este primer árbol, a partir de éste se crea el Árbol de Patrones de Frecuencias Máximas. Observando la Figura 2.26b, podemos observar que hay múltiples ramificaciones para llegar a varios items iguales, por lo que se comienza por leer el árbol de abajo hacia arriba. Por ejemplo, para el item d, existe una única rama para llegar a él ($\{c,\ a,\ d\}$), por tanto es máximo, por lo que se crea la primera rama del árbol (Figura 2.26d). Seguidamente se hace lo mismo para el item f, que al también ser único, es máximo y se agrega al nuevo árbol (Figura 2.26e). Si quisiéramos revisar para el item b, la única forma de llegar a él es por medio de los items $\{e,\ c,\ a,\ b\}$, y como notamos, ya esta rama existe como un subgrupo de una rama más grande, por lo que no se agrega ya que no es máximo. Finalmente se hace lo mismo para el item g (Figura 2.26f).Una vez determinado el Árbol de Patrones de

Frecuencias Máximas, cada rama va a ser los Patrones Frecuentes Máximos del algoritmo(Grahne y Zhu (2014)).

3 Resultados

A continuación, se presentan los resultados obtenidos de todos algoritmos, haciendo comparaciones entre ellos, gracias a los porcentajes de efectividad de cada uno, tanto para los Clasificadores como para los Clusters. En el caso de las Reglas de Asociación, los resultados van a buscar una comparación entre las reglas obtenidas por cada uno de los algoritmos y la relevancia real de cada una, ya que el algoritmo saca todas la relaciones entre las columnas y puede que hayan relaciones que sean obvias que existan.

3.1. Clasificadores

Una vez hecho todas las optimizaciones de los algoritmos, se pueden dividir los resultados de cada uno de ellos de la siguiente manera:

- **Datos sin optimizar:** Resultados con los grados de libertad predeterminados.
- **Datos optimizados:** Resultados con los grados de libertad optimizados.
- **Datos sin optimizar y con PCA sin optimizar:** Resultados con los grados de libertad predeterminados tanto para el algoritmo como para el PCA.
- **Datos sin optimizar y con PCA optimizado:** Resultados con los grados de libertad predeterminados pero con el PCA optimizado.
- **Datos optimizados y PCA sin optimizar:** Resultados con los grados de libertad optimizados pero el PCA sin optimizar.
- **Datos y PCA optimizados:** Resultados con los grados de libertad y el PCA optimizados.

En primera instancia, cada uno de los algoritmos fue probado sin hacer ninguna reducción de columnas, ajustando los grados de libertad de cada uno a los valores que recomienda la librería. Seguidamente, se implementó el Optimizador y se buscó cuales valores para esos grados de libertad generaban el mejor porcentaje de éxito.

En segundo lugar, se corrieron los algoritmos haciendo uso del PCA (Principal Component Analysis), el cual, como se mencionó con anterioridad, reduce la dimensionalidad de los datos, que en este caso, se traduce a una reducción de columnas. Esto facilita el análisis de los datos al punto que puede mejorar la precisión de los algoritmos. Además, facilita la visualización de los datos ya que se puede llegar a una dimensión de 3, la cual se puede graficar.

Accuracy of every algorithm							
Feat.Reduction	**Algorithm**	**KNN**	**LDA**	**SVM**	**NaiveBayes**	**DesisionTree**	**Rand.Forest**
No PCA	Not Optim.	36 %	45 %	55 %	36 %	64 %	45 %
	Optimized	**82 %**	73 %	**82 %**	**55 %**	**91 %**	**100 %**
PCA Not Optim.	Not Optim.	18 %	27 %	36 %	36 %	55 %	64 %
	Optimized	63 %	55 %	36 %	36 %	55 %	55 %
PCA Optimized	Not Optim.	45 %	45 %	45 %	36 %	45 %	36 %
	Optimized	**82 %**	**82 %**	73 %	36 %	82 %	82 %

Cuadro 3.1: Tabla comparativa de todos los algoritmos de Clasificación

Una vez hecho cada análisis para los diferentes valores de variables en cada uno de los algoritmos, se obtienen los resultados observados en la Tabla 3.1, en donde se puede apreciar claramente que el algoritmo que mejor se comporta es el **Random Forest**, con una precisión de un 100 %. Cabe destacar que lo que se determinó para cada algoritmo fue el mejor caso posible para una combinación de todas las variables, ya que a partir de las optimizaciones, se determinan las variables que dan éste resultado. Ésto no quiere decir que, con un set nuevo de datos, se va a obtener una precisión de 100 %, ya que, como se mencionó en la Sección 1.6.5, este resultado es para este set con que se está trabajando. Sin embargo, esto fue lo que impulsó los experimentos que se mencionaron con anterioridad en la Sección 1.6.5 donde se obtiene una mejor forma de escoger el algoritmo más óptimo para esta tarea de clasificar.

Como parte de los resultados obtenidos para el algoritmo Random Forest, se obtiene una tabla con las columnas más importantes para la determinación de la precisión del algoritmo, lo cual sugiere cuales son las posibles columnas a las que hay que darles una mayor énfasis para los análisis. En este caso, podemos obtener de la Tabla 3.2 que variables como la Aceleración Máxima o la Aceleración Absoluta son de mucha importancia para las conclusiones del algoritmo. Cabe mencionar que todos los demás algoritmos, exceptuando KNN y NaiveBayes, generan estas mismas tablas y se pueden observar, junto con los resultados totales en el Apéndice A.

Feature Importance RF Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
MAX Acc(m/s^2)	55.52906071
Acc Abs [3 10]m	54.46971513
Takeoff [8 - 100]G	52.12255712
Energy Expenditure	45.35642029
DSL	40.07258677
Vel Rel [75 - 85] %	38.34124415
Landing [8 - 100]G	35.21820303
[0,5-1] hz(%)	33.89600908
heredia	31.70347561
Vel Rel [75 - 85](m)	30.30608627
HIBD (m/min)	30.01920903
[1,5-100] hz(%)	29.77468961
MAX Speed(km/h)	28.26094478
Vel Rel [85 - 95](m)	28.14046849
Impacts [0-3] G	27.84264266
Vel Abs [24 - 50] %	27.51718357
Dec/min	26.48976775
Acc Abs [-10 -3] %	26.03127353
Vel Rel [75 - 85](m/min)	25.1977825

Cuadro 3.2: Tabla de importancia de columnas para RF optimizado

3.2. Clusters

Al igual que con los Clasificadores, los Clusters se van a ver divididos según la forma en la que se aplicaron las optimizaciones, por lo que se pueden dividir los resultados de la siguiente forma:

- **Datos sin optimizar y con LDA sin optimizar:** Resultados con los grados de libertad predeterminados tanto para el algoritmo como para el LDA.
- **Datos sin optimizar y con LDA optimizado:** Resultados con los grados de libertad predeterminados pero con el LDA optimizado.
- **Datos optimizados y LDA sin optimizar:** Resultados con los grados de libertad optimizados pero el LDA sin optimizar.
- **Datos y LDA optimizados:** Resultados con los grados de libertad y el LDA optimizados.

Hay que recordar que para estos algoritmos, es necesario determinar un número óptimo de clusters por medio del *WCSS* o *"Método del Codo"*, por tanto ésto no se considera una optimización del algorimto. Sin embargo, el resto de grados de libertad se ajustaron a los valores recomendados por la librería. Recordando que las salidas del Cluster son únicamente el numero del cluster al cual el partido pertenece, se usa el algoritmo LDA para generar un porcentaje de precisión de cada cluster y la lista con las columnas más importantes. Por ende, las versiones sin optimizar van a ser los grados de libertad del algoritmo en sí y los del LDA predeterminados por la librería.

Para la optimización del cluster, se ideó un script el cual toma los grados de libertad y el rango del numero de clusters, ejecuta el *WCSS* y determina la desviación estándar de cada una de las posibilidades, para luego elegir la opción cuya desviación estándar sea la menor, ya que esto implica una mayor cohesión entre los datos, lo cual nos indica que el cluster está más optimizado. La optimización del LDA es la misma que se utiliza en la parte de Clasificadores para el algoritmo. Cabe destacar que el mismo optimizador se usó para los 3 métodos de clustering y no uno por cada uno.

Accuracy				
Feat.Reduction	**Algorithm**	**KMeans**	**GMM**	**Hierarchical C.**
LDA Not Optim	Not Optim.	81 %	81 %	**91 %**
	Optimized	81 %	**100 %**	72 %
LDA Optimized	Not Optim.	72 %	**100 %**	**91 %**
	Optimized	**100 %**	81 %	**91 %**

Cuadro 3.3: Tabla comparativa de todos los algoritmos de Clustering

Completada la ejecución para los tres algoritmos, se genera la Tabla 3.3 donde se puede observar que tanto el algoritmo de Kmeans como el GMM, para diferentes configuraciones, pueden obtener una precisión de un 100 % usando LDA al tratar de determinar a cual cluster pertenece cada uno de los partidos. De igual manera, esto sólo nos indica que para ciertos valores específicos de clustering, existe la posibilidad de tener un máximo de precisión, no quiere decir que siempre se vaya a comportar de la misma manera para diferentes sets de valores.

De igual manera, parte de los resultados obtenidos corresponde a las columnas más relevantes, las cuales se generan para cada uno de los algoritmos. Sin embargo, ya que el algoritmo Kmeans se usó para las Reglas de Asociación, entonces éste fue el que se eligió como representativo. Podemos apreciar en la

Tabla 3.4 que muchos de los clusters dependen de los equipos contra los que se hayan jugado, pero las variables de HSR Absoluta y Sprints son bastante relevantes para la determinación de los clusters. El resto de datos se puede apreciar en el Apéndice B.

Feature Importance KM/LDA Optimized	
Column Name	*Importance (%)*
lda	79.30762405
grecia	65.20539046
casa	56.22154174
HSR Abs Dist (m)	52.43552797
santos	42.33961243
Sprint Abs(m)	42.17739716
Vel Rel [95 - 100](m/min)	40.89293377
Acc Abs [-10 -3] %	40.84245855
Takeoff [8 - 100]G	40.54150187
Power met	40.03546309
MAX Acc(m/s^2)	39.09847084
sc	37.39637376
HSR Rel Dist (m)	36.56580581
Energy Expenditure	36.19535355
Vel Rel [0 - 45](m/min)	35.81002968
Takeoff [5 - 8]G	34.8233967
Step Balance(%)	33.8326272
Sprints REL	31.78371614
Vel Abs [12 - 19](m/min)	31.56941736
Acc Abs [3 10] %	30.08245714

Cuadro 3.4: Tabla de importancia de columnas para KMeans y LDA optimizado

3.3. Reglas de Asociación

Luego de hacer el preprocesamiento de los datos para discretizar todos los datos, se ejecutaron los diferentes algoritmos de Reglas de Asociación para obtener sets de reglas que dan a conocer diferentes relaciones entre los atributos (variables del WIMU). Es importante entender que todos estos algoritmos desglosan todas las posibles combinaciones de reglas que cumplan con un límite dado, en el caso de éstos algoritmos, se hizo un filtro a través del Support y el Lift, por lo que al poner un límite muy alto, la cantidad de reglas se va a reducir, mientras que al reducir los límites, se van a aumentar exponencialmente la cantidad de reglas.

El total de reglas obtenidas está en la cantidad de los cientos de miles, por lo que tratar de ver todas las reglas es imposible. Además, la pregunta más importante que trata de responder este proyecto es si "*¿existe cierto grupo de variables del WIMU que puedan indicar si el equipo gana, pierde o empata?*", por lo que el objetivo es ver cuales reglas tienen como consecuente la variable **Resultado** en ella, ya que podremos filtrar si el resultado es Ganar (Resultado "1"), Empatar (Resultado "0") o Perder (Resultado "−1"). Para esto, simplemente se toma el archivo de salida del algoritmo, se abre en el programa Excel y se hace un filtrado por la variable Resultado.

Al llegar acá, nos dimos cuenta que, debido a que el equipo gana mucho, los filtros que colocamos al inicio, no eran suficientes para extraer las reglas que nos indicaran si el equipo estaba en alguna de estas 3 condiciones, por lo que los límites se disminuyeron. Sin embargo, solo se logró obtener el resultado de cuando el equipo Ganaba (Tabla 3.5), e intentar disminuir más los límites hacía que el costo computacional fuese mucho más alto, haciendo que el algoritmo tomase muchísimo más tiempo en ejecutar, por lo que en esta sección solo se observaron las reglas que implicaban que el equipo Ganase. Esto dio idea a nuevos experimentos al usar los algoritmos de Clasificación como apoyo para reducir la cantidad de columnas, ya que como vimos con anterioridad, éstos nos indicaban las columnas más relevantes para el análisis. Más adelante se describirá este proceso.

Respecto a cual de los algoritmos es mejor, se tuvo varios problemas. Los algoritmos Apriori y FP-Growth tardan aproximadamente lo mismo en ejecutar bajo las mismas condiciones. Ambos generan un set de reglas el cual, tratar de hacer una comparación una a una se hace imposible. Se escogió el Algoritmo Apriori para mostrar los resultados en esta sección pero el resto de resultados se muestran en el Apendice C. Luego, el algoritmo FP-Max nunca brindó ningún set de reglas, por lo que se determinó que, al tener tantas columnas relacionadas entre sí, el algoritmo no logra obtener los Árboles de Patrones Frecuentes Máximos, por ende, no crea ningún tipo de relación entre las columnas.

Es importante indicar que, debido al preprocesamiento previamente descrito, los antecedentes que muestran las reglas son las columnas con los rangos específicos de cada una, lo que indica una mayor precisión en la variable que hay que hacer énfasis a la hora de hacer los análisis, ya que no solo indica una variable, sino un específico rango de valores a los cuales esta variable se tiene que dar para que por consecuencia se Gane, Pierda o Empate.

A partir de la Tabla 3.5, podemos determinar que la columna del **Marcador**

Association Rules created by Apriori Algorithm				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
Takeoff [3 - 5]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.396226	0.75	1.528846
Takeoff [3 - 5]/min[0.01 0.01]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.396226	0.75	1.528846
Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.339623	0.818182	1.667832
Vel Rel [95 - 100](m/min)[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.339623	0.75	1.528846
Takeoff [8 - 100]/min[0. 0.]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.339623	0.818182	1.667832
Vel Rel [95 - 100](m/min)[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.339623	0.75	1.528846
Vel Rel [75 - 85] %.1[0.49 0.62]	Ganar	0.320755	0.73913	1.506689
Vel Rel [75 - 85] %[0.49 0.62]	Ganar	0.320755	0.73913	1.506689
Vel Rel [95 - 100] %.1[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [95 - 100] %[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Vel Rel [75 - 85] %[0.49 0.62]	Ganar	0.320755	0.73913	1.506689
Edi (/min)[0.01 0.01]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.809524	1.650183
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.73913	1.506689
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [95 - 100] %.1[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Vel Rel [75 - 85] %[0.49 0.62]	Ganar	0.320755	0.73913	1.506689
Edi (/min)[0.01 0.01]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.809524	1.650183
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.73913	1.506689
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Ganar	0.320755	0.772727	1.575175

Cuadro 3.5: Tabla de Reglas Apriori que determinan cuando el equipo Gana

Contrario, **Takeoff**, **Landing** o **Velocidad Relativa** son importantes indicadores que el equipo va a Ganar, destacando que el Marcador Contrario, según el rango mostrado, debe ser de 0 o 1 gol, el Landing tiene que ser de 0.01 o la Velocidad Relativa de entre un 75 % a un 85 % debe estar en un rango de 0.49 a un 0.62. Como se muestra, existe una combinación entre una o varias de las columnas y éstas van a ser más o menos importantes según el Support que se muestra, así como el Confidence y el Lift. La Tabla 3.5 viene ordenada de mayor a menor por el Support que tenga la regla, lo que indica, como se vio previamente, el total de veces que aparece esta relación entre el total de relaciones posibles que hay.

3.4. Caracterizaciones de los Clasificadores

Como se vio en la Sección 3.1, los resultados obtenidos de cada uno de los Clasificadores eran únicamente para un grupo específico de datos, o sea, estábamos determinando el mejor de los casos, pero esto no nos da el mejor resultado posible, ya que se espera poder tener resultado más llamativos con sets de datos diferentes.

Como ya se discutió en la Sección 1.6.5, se hizo una caracterización para cada uno de los algoritmos para obtener un porcentaje de efectividad más constante, haciendo un barrido de los grados de libertad y, por medio de la Mediana y la Desviación Estándar, tomar una mejor decisión de cuál era el mejor algoritmo. También, en esta Sección se discutió que los datos se subdividieron en dos categorías, una con datos divididos entre Ganar y No Ganar, y otra con Empatar y Perder, con el fin de tener una mejor distribución de los datos a la hora de clasificarlos.

Caract. for RandomState Variable on Win/NoWin Data						
Algorithm	KNN		LDA		SVM	
Metric	Med	SD	Med	SD	Med	SD
Accuracy	54.5 %	16.1	63.6 %	13.2	**72.7 %**	12.7
Precision	53.6 %	16.3	63.3 %	13.8	**75.2 %**	14
Recall	53.5 %	15	63.3 %	15.1	**76 %**	13.9
F1Score	53 %	16.2	60.7 %	13.9	**71.8 %**	14.4
Algorithm	NaiveBayes		Decis.Tree		Rand.Forest	
Metric	Med	SD	Med	SD	Med	SD
Accuracy	45.5 %	14.2	**72.7 %**	16.7	54.4 %	14.9
Precision	45 %	14	**76 %**	15	61.1 %	15
Recall	45 %	15.7	**75 %**	16.3	60.7 %	14.9
F1Score	45 %	14.6	**71.8 %**	17	54.5 %	15.3

Cuadro 3.6: Tabla comparativa con la caracterización de cada algoritmo para los datos de Ganar/NoGanar

Caract. for Random. State Variable on Draw/Lose Data						
Algorithm	KNN		LDA		SVM	
Metric	Med	SD	Med	SD	Med	SD
Accuracy	33.3 %	20.3	33.3 %	18	41.7 %	22.7
Precision	33.3 %	23.7	33.3 %	23.8	**50 %**	25.9
Recall	37.5 %	22	43.8 %	20.3	**50 %**	24.8
F1Score	33.3 %	20	33.3 %	17.1	33.3 %	23.1
Algorithm	NaiveBayes		Decis.Tree		Rand.Forest	
Metric	Med	SD	Med	SD	Med	SD
Accuracy	**50 %**	17.5	33.3 %	20.6	33.3 %	16
Precision	**50 %**	22.5	37.5 %	22.7	30 %	20.4
Recall	**50 %**	19.6	38.8 %	22.9	37.5 %	18.4
F1Score	36.7 %	18.3	33.3 %	20.6	25 %	15.7

Cuadro 3.7: Tabla comparativa con la caracterización de cada algoritmo para los datos de Empatar/Perder

Los resultados se pueden observar en la Tabla 3.6, donde los algoritmos **Support Vector Machine** y **Decision Tree** son los que tienen un mejor respuesta dan a la hora de clasificar los partidos entre Ganar y No ganar, mientras que en la Tabla 3.7, vemos que para la mayoría de medidas de exactitud, **Naive Bayes** es el mejor para clasificar los partidos entre Empatar y Perder, aunque **Support Vector Machine** también se podría considerar bueno para ésto. Sin

embargo, debido a que en la Tabla 3.1, en la Sección 3.1, el algoritmo Decision Tree fue el segundo mejor de todos, se decidió que ése es el mejor algoritmo de Clasificación sobre todos los demás.

Además, como parte de los resultados, se muestran las caracterizaciones a nivel gráfico del Decision Tree (Figura 3.1), donde se observa que la mayoría de los valores para el grado de libertad *Random State* dan un muy buen resultado, inclusive dando varios un resultado de 100 % en la predicción de Ganar y No Ganar, caso contrario en la Figura 3.2 donde se muestra que este algoritmo no es tan bueno para clasificar partidos entre Empates y Pérdidas. Esto último es debido a que, aunque se tiene un 50 % de partidos con Empate y un 50 % de partidos con Pérdida, el total de partidos es muy poco, por lo que el algoritmo no tiene suficientes datos para poder entrenarse de una manera mucho más apropiada y por ende, falla más.

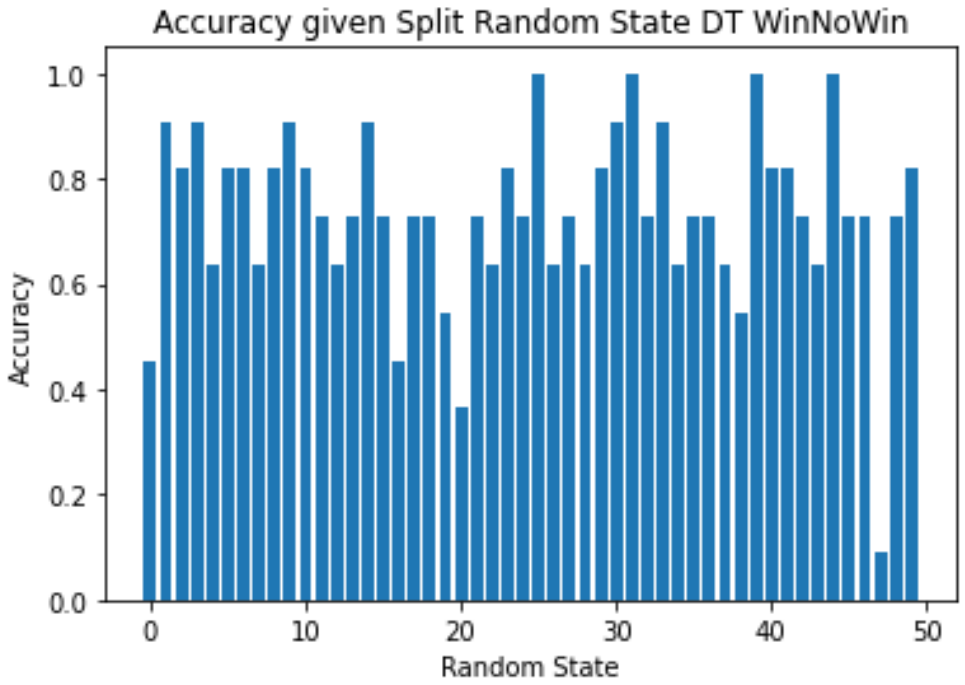


Figura 3.1: Caracterización del algoritmo Decision Tree para clasificar Ganar o No Ganar

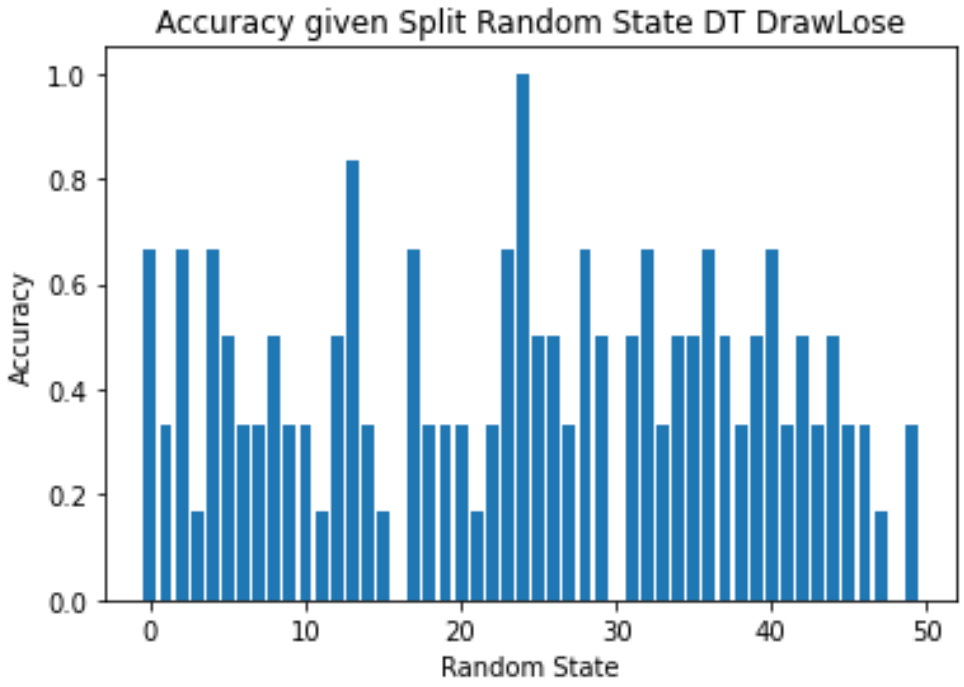


Figura 3.2: Caracterización del algoritmo Decision Tree para clasificar Empatar o Perder

Finalmente, con los datos obtenidos de la Sección 3.1, se volvió a correr el algoritmo de Decision Tree para generar la Tabla 3.8 con las columnas más relevantes para dicho algoritmo, que, como se mencionó en la Sección 1.6.5, serán usados para generar las nuevas Reglas de Asociación.

Feature Importance
Column Name
Marcador Contr
Marcador Casa
HSR Abs (m/min)
HSR Rel Count
Hz*G MAX
Impacts [0-3] G
MAX Dec(m/s^2)
Vel Rel [45 - 65] %

Cuadro 3.8: Columnas más importantes determinadas por el Decision Tree

3.5. Nuevas Reglas de Asociación

Como se describió en la Sección 1.6.6 y 3.3, dado a que hay demasiadas columnas para poder identificar con éxito si el equipo Empata o Pierde, solo se puede determinar si Gana, se decidió hacer un nuevo script el cual, filtra todas las columnas y deja únicamente aquellas que determinó el mejor algoritmo de clasificación, que como ya vimos en la Sección 3.4, fue el Decision Tree en la Tabla 3.8. Luego, se preprocesa los datos para que sean únicamente estas columnas las cuales se discreticen y se usen como entrada para los algoritmos de Reglas de Asociación.

Como resultado, se obtuvo una cantidad de reglas muchísimo más manejable que dan excelentes resultados y relaciones entre las columnas analizadas. De igual forma, ya que no se puede determinar con exactitud cual algoritmo es mejor, se trabajó con ambos, pero se presenta únicamente las reglas dadas por el algoritmo Apriori. No se volvió a experimentar con el algoritmo FP-Max por cuestiones de tiempo pero se podría, en un futuro, volver a tratar de implementar.

Podemos observar en la Tabla 3.9 el total de reglas que se determinaron para que el equipo Ganase. De esta tabla podemos notar que el **Marcador Contrario**, **Marcador Casa**, **Impactos**, **HSR Absoluto** y **HSR Relativo** son las columnas más importantes que determinan que el equipo Gana. Se pueden hacer conclusiones como que el equipo gana, con un soporte de 0.45, cuando el Marcador Contrario es 0 o 1, también, que el equipo suele ganar cuando tiene un marcador a favor de 2. Algo que se discutió con profesionales en el área de fútbol es que, a pesar de que los Impactos son importantes, es difícil de medir realmente a qué se refiere esta variable, ya que puede implicar muchas acciones que no se pueden entrenar.

En la Tabla 3.10 podemos notar que para que el equipo Empate, las columnas más importantes vuelve a ser el **HSR Absoluto**, **HSR Relativo**, **Marcador Contrario** y **Marcador Casa**, sin embargo, los rangos de estas variables ahora son diferentes, por eso aparecen por acá. Por ejemplo, se tiene que el equipo empata cuando el Marcador Contrario es 0 o 1, ó 2 o 3, lo que nos lleva a decir que si el equipo contrario llevase la delantera por 2 o 3 goles, el equipo Casa lo que puede llegar es a empatar.

Finalmente, la Tabla 3.11 nos indica las reglas en las que el equipo va a Perder. **HSR Absoluto**, **HSR Relativo**, **Marcador Contrario** y **Marcador Casa** vuelven a ser relevantes para esto, con rangos diferentes. Por ejemplo, podemos decir que el equipo Casa suele perder cuando el equipo contrario

consigue anotar 2 o 3 goles, pero ellos logran 0 o 1. Esto es algo obvio, pero nos ayuda a determinar las razónes por las cuales el equipo puede perder, con el fin de evitar goles que, segun los datos, los pueden llevar a perder el partido.

Reglas que determinan si el equipo GANA				
Antecedents	**Concequents**	**Support**	**Confidence**	**Lift**
Marcador Contr[0. 1.]'	Ganar	0.45283	0.705882	1.438914
Impacts [0-3] G[10093.23 12253.58]'	Ganar	0.301887	0.5	1.019231
Impacts [0-3] G[10093.23 12253.58]', 'Marcador Contr[0. 1.]'	Ganar	0.301887	0.727273	1.482517
HSR Abs (m/min)[3.75 4.55]'	Ganar	0.283019	0.576923	1.176036
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]'	Ganar	0.283019	0.882353	1.798643
HSR Rel Count[3.77 4.54]'	Ganar	0.226415	0.545455	1.111888
Marcador Casa[2. 2.]'	Ganar	0.226415	0.666667	1.358974
Vel Rel [45 - 65] %[9.11 10.09]'	Ganar	0.226415	0.571429	1.164835
Marcador Casa[2. 2.]', 'Marcador Contr[0. 1.]'	Ganar	0.226415	1	2.038462
Vel Rel [45 - 65] %[9.11 10.09]', 'Marcador Contr[0. 1.]'	Ganar	0.226415	0.923077	1.881657
Hz*G MAX[4.04 4.29]'	Ganar	0.207547	0.6875	1.401442
Marcador Contr[0. 1.]', 'HSR Rel Count[3.77 4.54]'	Ganar	0.207547	0.733333	1.494872
Impacts [0-3] G[12489.85 14153.08]'	Ganar	0.188679	0.526316	1.072874
Marcador Casa[3. 5.]'	Ganar	0.188679	0.769231	1.568047
Vel Rel [45 - 65] %[8.05 8.81]'	Ganar	0.188679	0.5	1.019231
HSR Abs (m/min)[3.75 4.55]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Ganar	0.188679	0.625	1.274038
Hz*G MAX[4.04 4.29]', 'Marcador Contr[0. 1.]'	Ganar	0.188679	0.714286	1.456044
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Ganar	0.188679	0.909091	1.853147
HSR Abs (m/min)[2.7 3.64]'	Ganar	0.169811	0.428571	0.873626
HSR Rel Count[4.77 5.54]'	Ganar	0.169811	0.75	1.528846

Cuadro 3.9: Tabla Reglas Apriori después de Caracterización que indican si el equipo Gana

Reglas que determinan si el equipo Empata				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
HSR Abs (m/min)[2.7 3.64]'	Empatar	0.150943	0.380952	1.346032
HSR Rel Count[2.08 3.69]'	Empatar	0.113208	0.4	1.413333
Impacts [0-3] G[10093.23 12253.58]'	Empatar	0.169811	0.28125	0.99375
Marcador Contr[0. 1.]'	Empatar	0.150943	0.235294	0.831373
Marcador Contr[2. 3.]'	Empatar	0.132075	0.411765	1.454902
Marcador Casa[0. 1.]'	Empatar	0.150943	0.363636	1.284848
MAX Dec(m/sÂ²) [-6.38 -6.06]'	Empatar	0.113208	0.428571	1.514286
HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]'	Empatar	0.113208	0.428571	1.514286
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]'	Empatar	0.113208	0.545455	1.927273
Marcador Casa[0. 1.]', 'Marcador Contr[0. 1.]'	Empatar	0.150943	0.571429	2.019048
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]'	Empatar	0.113208	0.666667	2.355556

Cuadro 3.10: Tabla Reglas Apriori después de Caracterización que indican si el equipo Empata

Reglas que determinan si el equipo Pierde				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
HSR Abs (m/min)[3.75 4.55]'	Perder	0.132075	0.269231	1.189103
HSR Rel Count[2.08 3.69]'	Perder	0.113208	0.4	1.766667
Impacts [0-3] G[10093.23 12253.58]'	Perder	0.132075	0.21875	0.966146
Marcador Contr[2. 3.]'	Perder	0.150943	0.470588	2.078431
Marcador Casa[0. 1.]'	Perder	0.188679	0.454545	2.007576
Vel Rel [45 - 65] %[8.05 8.81]'	Perder	0.113208	0.3	1.325
HSR Abs (m/min)[3.75 4.55]', 'Marcador Casa[0. 1.]'	Perder	0.132075	0.7	3.091667
Impacts [0-3] G[10093.23 12253.58]', 'Marcador Casa[0. 1.]'	Perder	0.113208	0.545455	2.409091
Marcador Casa[0. 1.]', 'Marcador Contr[2. 3.]'	Perder	0.132075	1	4.416667

Cuadro 3.11: Tabla Reglas Apriori después de Caracterización que indican si el equipo Pierde

4 Discusión y Conclusiones

Luego de un amplio recuento de resultados, podemos comenzar a juntar todas las conclusiones que se lograron obtener de toda la experimentación hecha. Como se mencionó al inicio, ésta es una primera aproximación al análisis de estos datos, los cuales dan unos resultados bastante prometedores, pero no 100 % concluyentes, ya que primero se está haciendo una exploración de los datos y una manipulación por medio de Machine Learning, lo cual nos ayuda a tener un mejor horizonte para seguir.

Se puede empezar por la base de datos. Los datos que se tenían originalmente eran la de cada partido, separados en primer tiempo, segundo tiempo y total del encuentro, esto por jugador. Sin embargo, el análisis que se quería hacer era por partido, por lo que se hizo un promedio de los resultados de todos los jugadores para aproximar los datos de cada partido, esto quiere decir que se trabajó con un resumen del partido, y no con todo el partido completo. Además, al ser un resumen, no se pueden hacer conclusiones temporales del partido, o mejor dicho, no se puede hacer un análisis en tiempo real de las situaciones del partido que podrían llevar a determinar mejores resultados, mejores entrenamientos e identificar situaciones más específicas las cuales pueden ser entrenables.

Para más profundidad en los análisis, en un futuro se puede trabajar ya no con los promedios, sino con cada uno de los jugadores, para poder tener una mayor cantidad de resultados y agregar el aspecto temporal en ellos, para así descubrir más detalles situacionales, lo cual ayudaría a los entrenadores a practicar más este tipo de eventos, que ayudarían al equipo a ganar.

Otro detalle a destacar es que los datos no necesariamente estaban completos. Puede que durante algún partido, al entrenador se le haya olvidado accionar el inicio de la recolección de datos, por lo que en varios de ellos se mostraban valores parciales, como únicamente las primeras partes, esto hace que el promedio del equipo se disminuyese y esto alterase un poco el total para esos partidos con fallas.

Cabe destacar que los WIMUs utilizados tienen los valores predeterminados por el vendedor del producto. Esto lo que quiere decir que muchos de las variables vienen con rangos quizá mal ajustados para el equipo en sí. Es

importante que para futuros análisis, primero se haga un ajuste correcto de cada una de las variables del WIMU para así poder entender mucho mejor las Reglas de Asociación obtenidas y las Columnas relevantes para cada algoritmo, ya que actualmente podrían ser, muchas de ellas, irrelevantes. Por ejemplo, en algunas de las columnas de variables, los datos mostrados eran binarios, debido a que ningún jugador llegaba a esos rangos, mientras que en otras variables, los rangos eran bastante dinámicos por lo que facilitaba su entendimiento. Esto a su vez es importante ya que las variables binarias pueden llegar a tener más peso a la hora de ejecutar los algoritmos de Clasificación, Clustering o Reglas de Asociación, haciendo quizá que columnas que si son relevantes para el entrenador, quedasen por fuera.

Es importante hacer notar que los datos que se tienen, a pesar de tener muchas variables, son pocos, pues son solamente 52 partidos y estos sólo contienen los promedios de todo el equipo, por lo que los algoritmos, al necesitar de una base de datos para aprender el comportamiento de los partidos, no se obtienen resultados tan concluyentes como se quieren, aunque no quiere decir que con los que se tengan no se pueda ya tener buenos resultados. Se recomienda aumentar la cantidad de análisis de los partidos para generar una mayor base de datos con el cual entrenar los algoritmos y seguir analizando los diferentes resultados obtenidos.

Otro aspecto importante es que, aunque para los aficionados al equipo no sea algo malo, el Equipo Casa gana muchos partidos, lo que hace que el total de partidos Ganados, Perdidos y Empatados no tengan un balance entre sí. Esto lleva a que algunos algoritmos a determinar más fácilmente si el equipo Gana, pero a la hora de determinar si el equipo Empata o Pierde, no son tan buenos.

Durante la primera parte del desarrollo del proyecto, se tenía la idea de que solamente era importante determinar cual era el mejor de los casos para la determinación de una buena precisión. Sin embargo, se notó que era necesario hacer ajustes en los algoritmos para que éstos pudiesen ser utilizados no solo en un caso específico, sino más bien con cualquier set de datos que se le agreguen. Razón por la cual, en primera instancia, el algoritmo Random Forest parecía como el mejor algoritmo para predecir los resultados, pero ésto era solo para un caso específico. Se tuvo que hacer una caracterización para poder determinar cual, de todos los algoritmos, iba a funcionar con cualquier set de datos, razón por la cual se determinó que el Decison Tree era el mejor opción para determinar resultados.

Hacer uso del PCA para reducción de columnas en los algoritmos de Cla-

sificación y Clustering no hizo ningún aprote de mejora en los algoritmos, sino más bien, empeoraron la efectividad con que éstos respondían. Con esto se determinó que la combinación de algoritmos, al menos para este set de datos en específico, no produjo ningún beneficio, por lo que no se tomó en cuenta para la presentación de resultados, sin embargo, los mismos se pueden apreciar en los apéndices.

El porcentaje de precisión visto en los algoritmos no indica el porcentaje de cuando el equipo Gana, Empata o Pierde, si no, más bien indica la facilidad con que se determina el resultado, sea cual sea, dado un partido con ciertas características, por lo que se usa las Reglas de Asociación para determinar cuales son las variables que me van a llevar a Ganar, Perder o Empatar.

Se descubrió también que los cluster nos iban a ayudar a poder categorizar las variables, para poder hacer uso de las Reglas de Asociación y se reforzó la decisión de tomar el algoritmo KMeans por medio de las referencias utilizadas, ya que demostraba que el uso de Clustering para discretizar las variables era la mejor opción.

Las primeras Reglas de Asociación obtenidas no eran tan concluyentes como se esperaban, ya que solamente nos indicaba cuando el equipo ganaba. Gracias a los clasificadores y la facilidad de identificar las columnas más importantes, se logró reformular la forma de tratar los datos y se lograron obtener unas reglas más concluyentes. Sin embargo, cabe decir que estos algoritmos determinan todas las posibilidades que haya, por lo que muchas veces podemos encontrar conclusiones obvias como parte de los resultados.

Las Reglas de Asociación no solo van a indicar cual columna es relevante, sino también nos van a indicar cual es el rango al cual esa variable debe estar para que el equipo Gane, Pierda o Empate. Esto es importante ya que nos va a determinar qué ejercicio debe entrenarse y bajo qué condiciones.

Cabe destacar también que las Reglas de Asociación donde muestra que el equipo Empata o Pierte tienen un Support bajo, debido a que, como se ha venido mencionando, se necesitó reducir el umbral de los grados de libertad para poder descubrir estas variables, por lo que a pesar de existir las reglas, no se pueden tomar como muy determinantes a la hora de dar conclusiones.

Como parte de las conclusiones dadas por estas reglas, se determinaron las siguientes:

- El Marcador Contrario, es muy determinante para identificar si el equipo Gana, Pierde o Empata. De alguna forma esto es obvio pero nos lleva a

entender cosas como que el equipo suele ganar bajo un marcador de 2 a 0 ó 2 a 1 a favor. Esto no quiere decir que si el equipo toma la delantera al minuto 10 con 2 goles a favor, ya se sabe que se ganó el partido. Hay que recordar que lo que se usó fueron datos de todo el partido en forma de resumen y que no existe ningún factor de tiempo agregado a ellos, por tanto, lo que podemos manifestar es lo que suele suceder al final del partido respecto al Marcador.

- A pesar de que Marcador Contrario y Marcador Casa son relevantes, son variables que no se pueden entrenar, por lo tanto, de alguna forma son irrelevantes para generar acciones futuras, más que mostrar datos estadísticos.
- Los Impactos de 0 a 3 G con concluyentes a la hora de determinar si el equipo Gana, pero es difícil de saber qué podemos entrenar con esta variable, ya que los impactos pueden ir desde barridas, a golpes de balón y roces entre jugadores. Lo que se puede decir es que una alta cantidad de impactos fuertes implica mucho roce entre los jugadores, lo cual indica que la intensidad del partido debe ser fuerte.
- La columna de Distancia total recorrida no aparece como concluyente. Ésta, según los profesionales consultados, es importante para los entrenadores, ya que les interesa que sus jugadores corran. Sin embargo, dados los resultados, se puede determinar que, no porque el equipo corra, implica que se gana un partido.
- La Velocidad Relativa de entre un 45 % y un 65 % aparece como influyente para que el equipo Gane, lo que refuerza la conclusión anterior, ya que el jugador no debería preocuparse tanto por correr sino más bien en el juego intenso.
- Reforzando las conclusiones anteriores, se tiene que el HSR Absoluto por minuto es importante para que el equipo gane. El HSR lo que indica es la distancia recorrida a velocidades superiores al umbral dado por el entrenador. El rango obtenido es de 3.75 a 4.55 metros por minuto, mostrando que lo importante es la intensidad del juego, no correr largas distancias.
- Respecto a los Empates, el HSR Absoluto por minuto con un rango de 2.7 a 3.64 es importante para cuando el equipo Empate, aunque, el Support es bajo, de un 0.15. Sin embargo, esto apoya la conclusión de que si el equipo reduce la intensidad de juego, no va a lograr Ganar sino más bien Empatar.

- Se puede decir también que el equipo suele empatar bajo un Marcador de 0 a 0 ó 1 a 1, según las reglas.
- Otra importante conclusión respecto al Marcador Contrario es que, según las reglas, el equipo Empata o Pierde cuando el contrincante hace 2 o 3 goles, lo que nos lleva a decir que si el equipo contrario va ganando por 2 o 3 goles, el Equipo Casa tiende, a lo mucho, a Empatar.
- El HSR Absoluto por minuto con un rango de 2.7 a 3.64 también aparece en la lista de reglas para cuando el equipo Pierde, lo que nos lleva a agregar a la conclusión anterior que, sin intensidad, el equipo puede Empatar o Perder.
- Se puede decir también que, el Equipo Casa tiende a perder sus partidos bajo Marcadores de 2 a 1 ó 3 a 1.

A nivel de peso computacional, aunque se tenía presupuestado hacer uso de clusters de computadoras para poder implementar estos algoritmos, se determinó que una computadora con un procesador relativamente actual puede sobrellevar el peso de los algoritmos y sus caracterizaciones. Sin embargo, en un futuro, se planea tener más datos y poner a caracterizar más grados de libertad, lo que haría que existiesen muchos más ciclos de análisis en cada uno de los algoritmos, viendo ahora sí, la necesidad de clusters de computadoras para que el procesamiento de toda esta información sea más rápida.

A nivel de mejoras, varios de los algoritmos requieren que manualmente se hagan configuraciones para su ejecución, como es el caso de la discretización de las columnas, por lo que en un futuro se espera poder trabajar en la automatización de las mismas. A su vez, los algoritmos se ejecutan uno a uno por separado, se espera poder hacer una batería el cual una todos los algoritmos en uno solo y por medio del teclado, colocar pasos para la ejecución de algún algoritmo que se requiera.

Con el análisis hecho y con los datos que se tenían, se determinó que no es posible determinar si un jugador tiene la posibilidad de lesionarse. Esto debido a que, como se mencionó ya varias veces, los datos son promedios del equipo y no valores a lo largo de un tiempo por jugador, por lo que es imposible filtrar la posibilidad de lesión de un jugador. Para poder lograrlo, se debe hacer análisis en el tiempo por jugador en cada partido, pero esto se va a hacer en futuros trabajos de investigación.

Otra mejora que se recomienda hacer a los algoritmos actuales, debido a que muy complicado tener un balance entre todas las categorías (ganar,

perder o empatar) es hacer un muestreo de aquella categoría que sea la más grande, para así de alguna forma reducir la cantidad de datos a una que sea comparable con el resto de las categorías, lo que mejoraría la exactitud de los algoritmos para determinar las categorías como empatar y perder, que en nuestro caso, eran las que tenían menos datos.

Respecto a la escogencia de los algoritmos de Clasificación, se tomaron los más comunes de acuerdo a la literatura de Machine Learning, sin embargo, se puede profundizar más para encontrar algoritmos más robustos para obtener resultados diferentes y quizá hasta mejores.

A nivel de clustering, los resultados obtenidos solamente me indican la relación que tienen varios partidos entre sí, debido a las características que sus variables describen, pero no me facilitan decisiones respecto a si el equipo gana, pierde o empata, o si el equipo debe entrenar alguna variable en específico. Entonces, para trabajos futuros se espera seguir trabajando con clustering pero con barridos de datos, con el fin de obtener resultados, por ejemplo, a medio tiempo y poder tomar decisiones de si el equipo necesita hacer algún tipo de cambio táctico ya que, debido a la naturaleza de los datos, se está tendiendo a un cluster en específico que implica que el equipo pierda o empate.

Varias de las combinaciones de grados de libertad nos llevan a una exactitud tope de 91 %, como en el caso del Decision Tree, y cada una de estas combinaciones trae consigo una serie de columnas importantes, no necesariamente iguales entre cada set de grados de libertad, por lo que en un futuro, vale la pena hacer un barrido con los mejores resultados para éste algoritmo y determinar más columnas importantes, que van a ampliar la cantidad de reglas posibles que se pueden determinar, haciendo más robustas las conclusiones, ya que en el actual trabajo, solamente se usó una de las combinaciones.

Bibliografía

Abu Abbas, O. (2008). Comparisons between data clustering algorithms. *The International Arab Journal of Information Technology*, 5(3):320–325.

Ahmadi, A., Mitchell, E., Destelle, F., Gowing, M., OConnor, N., Richter, C., y Moran, K. (2014). Automatic Activity Classification and Movement Assessment During a Sports Training Session Using Wearable Inertial Sensors. *IEEE*.

Bengfort, B., Bilbro, R., Danielsen, N., Gray, L., McIntyre, K., Roman, P., Poh, Z., et al. (2018). Yellowbrick.

Bonomi, A., Goris, A., Yin, B., y Westerterp, K. (2009). Detection of Type, Duration, and Intensity of Physical Activity Using an Accelerometer. *American College of Sports Medicine*, páginas 1770–1777.

Born, S. y Schmidt-Thieme, L. (2004). Optimal discretization of quantitative attributes for association rules. *Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS)*, páginas 287–296.

Boyd-Graber, J. (2018). Clustering: Gaussian mixture models (12c). `https://www.youtube.com/watch?v=0NMC2NfJGqo`. [Video].

Brian, C., Wayne, P., Julien, B., y Bruce, D. (2009). An Evaluation of the Physiological Demands of Elite Rugby Union Using Global Positioning System Tracking Software. *Journal of Strenght and Conditioning Research*, 27:1195–1203.

Eremenko, K., de Ponteves, H., y SuperDataScience (2019). Machine learning a-z™: Hands-on python r in data science. `https://www.udemy.com/course/machinelearning/`. [Udemy Course].

Fung, G. (2001). A comprehensive overview of basic clustering algorithms.

Gabbett, T., Jenkins, D., y Abernethy, B. (2012). Physical demands of professional rugby league training and competition using microtechnology. *Journal of Science and Medicine in Sport*, 15(1):80–86.

Gaffney, M., O'Flynn, B., Mathewson, A., Buckley, J., Barton, J., Angove, P., Vcelak, J., Ó Conaire, C., Healy, G., Moran, K., O'Connor, N., Coyle, S., Kelly, P., Caulfield, B., y Conroy, L. (2009). Wearable wireless inertial measurement for sports applications. *Cork Open Research Archive, Institute of Electrical and Electronics Engineers (IEEE)*, páginas 21–24.

Garg, A. (2020). Complete guide to association rules. `https://towardsdatascience.com/association-rules-2-aa9a77241654`.

Grahne, G. y Zhu, J. (2014). High performance mining of maximal frequent itemsets. *6th International workshop on high performance data mining*, 16):34.

Grunz, A., Memmert, D., y Perl, J. (2012). Tactical pattern recognition in soccer games by means of special self-organizing maps. *Human Movement Science*, 31:334–343.

Gupta, A. (2020). Ml | frequent pattern growth algorithm. `https://www.geeksforgeeks.org/ml-frequent-pattern-growth-algorithm/?ref=lbp`.

Hossain, H., Khan, M., y Roy, N. (2017). SoccerMate: A personal soccer attribute profiler using wearables. *IEEE Xplore*.

Hughes, T., Jones, R., Starbuck, C., Sergeant, J., y Callaghan, M. (2019). The value of tibial mounted inertial measurement units to quantify running kinetics in elite football (soccer) players. A reliability and agreement study using a research orientated and a clinically orientated system. *Journal of Electromyography and Kinesiology*, 44:156–164.

Keshari, K. (2020). Apriori algorithm : Know how to find frequent itemsets. `https://www.edureka.co/blog/apriori-algorithm/#:~:text=Apriori%20algorithm%20uses%20frequent%20itemsets,a%20threshold%20value(support)`.

Khurana, K. y Sharma, S. (2013). A comparative analysis of association rules mining algorithms. *International Journal of Scientific and Research Publications*, 3(5).

Kos, M. y Kramberger, I. (2017). A Wearable Device and System for Movement and Biometric Data Acquisition for Sports Applications. *IEEE Xplore*, 5.

Kuang, Q. y Zhao, L. (2009). A practical GPU based kNN algorithm. *The 2009 International Symposium on Computer Science and Computational Technology (ISCSCI 2009)*, página 151.

Kumbhare, T. A. y Chobe, S. V. (2014). An overview of association rule mining algorithms. *International Journal of Computer Science and Information Technologies*, 5(1):927–930.

Lateef, Z. (2020). Knn algorithm: A practical implementation of knn algorithm in r. `https://www.edureka.co/blog/knn-algorithm-in-r/#What%20Is%20KNN%20Algorithm`.

Lee, Y., Ho, C., Shih, Y., Chang, S., Robert, F., y Shiang, T. (2015). Assessment of walking, running, and jumping movement features by using the inertial measurement unit. *Gait Psoture*, 41(4):877–881.

Ludl, M.-C. y Widmer, G. (2000). Relative unsupervised discretization for association rule mining. *European Conference on Principles of Data Mining and Knowledge Discovery*, páginas 148–158.

Maglogiannis, I. (2007). Emerging artificial intelligence applications in computer engineering: Real word ai systems with applications in ehealth, hci, information retrieval and pervasive technologies. *Frontiers in artificial intelligence and applications*, páginas 3–24.

Mata, J., Alvarez, J. L., y Riquelme, J. C. (2002). Discovering numeric association rules bia evolutionary algorithm. *Pacific-Asia conference on knowledge discovery and data mining*, páginas 44–51.

McMillan, P. (2015). The validity, reliability sensitivity of global positioning system inertial sensors to monitor training readiness in soccer. *Tesis de Maestría. Universidad de Glasgow.*

Mora, A. (2018). Cell phenotype classification using m-phase features in live-cell bright field time-lapse microscopy (tesis de maestría). *Universidad de Costa Rica, San José, San Pedro, Montes de Oca.*

Moreno, María N.and Segrera, S., López, V. F., y Polo, M. J. (2007). Improving the quality of association rules by preprocessing numerical data. *II Congreso Español de Informática*, páginas 223–230.

Morrison, M. (1985). Inertial Measurement Unit. *United States Patent*, 4,711,125.

Nieminen, H., Vermola, L., y Hyyppa, H. (2005). Sport movement analyzer and training device. *IFI CLAIMS Patent Services.*

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., y Duchesnay, E.

(2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Raschka, S. (2014). Linear discriminant analysis. https://sebastianraschka.com/Articles/2014_python_lda.html#step-1-computing-the-d-dimensional-mean-vectors.

Raschka, S. (2020). Maximal itemsets via the fp-max algorithm. http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/fpmax/#:~:text=In%20contrast%20to%20Apriori%2C%20FP,focuses%20on%20obtaining%20maximal%20itemsets.

RealTrack Systems (2017). WIMU Pro. http://www.realtracksystems.com/wp-content/uploads/2017/10/wimupro_dossier.pdf. Online; accesado el 16 de Noviembre de 2019.

Reif-Acherman, S. (2014). Juguetes como instrumentos de enseñanza en ingeniería: los casos del péndulo de Newton y el giroscopio. *Ingeniería y Competitividad*, 16(2):189–198.

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

Sahinoglu, Z., Gezici, S., y Suvec, I. (2005). Ultra-wideband positioning systems. *Cambridge*, New York.

Schuldhaus, D., Zwick, C., Korger, H., Dorschky, E., Kirk, R., y Eskofier, B. (2015). Inertial Sensor-Based Approach for Shot/Pass Classification During a Soccer Match. *KDD Workshop on Large-Scale Sports Analytics 2015, Sydney, Australia*.

SciKit-Learn-DecisionTree (2020). sklearn.tree.decisiontreeclasifier. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html?highlight=decision%20tree#sklearn.tree.DecisionTreeClassifier. [SKLearn Library].

SciKit-Learn-Glosary (2020). Glossary of common terms and api elements. https://scikit-learn.org/stable/glossary.html#term-random_state. [SKLearn Library].

SciKit-Learn-Lib (2020). sklearn.metrics.precision_score. https://scikit-learn.org/stable/auto_examples/model_selection/plot_precision_recall.html?highlight=precision%20recall. [SKLearn Library].

Seif, G. (2020). The 5 clustering algorithms data scientists need to know. https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68.

Shetty, B. (2019). An in-depth guide to Supervised Machine Learning Classification. An exhaustive understanding of classification algorithms in machine learning. https://builtin.com/data-science/supervised-machine-learning-classification. Online; accesado el 8 de Agosto de 2020.

Shotton, J., Fitzgibbon, A., Blake, A., Kipman, A., Finocchio, M., Moore, B., y Sharp, T. (2011). Real-time human pose recognition in parts from a single depth image. En *CVPR*. IEEE. Best Paper Award.

Silva, A., Salazae, A., y Correia, M. (2011). WIMU: WEARABLE INERTIAL MONITORING UNIT. A MEMS-based Device for Swimming Performance Analysis. *BIODEVICES*.

StatQuest (2016). Statquest: Linear discriminant analysis (lda) clearly explained. https://www.youtube.com/watch?v=azXCzI57Yfc. [Video].

StatQuest (2018). Statquest: K-means clustering. https://www.youtube.com/watch?v=4b5d3muPQmA. [Video].

StatQuest (2019). Support vector machines, clearly explained!!! https://www.youtube.com/watch?v=efR1C6CvhmE. [Video].

Tanenhaus, M. (2012). Miniaturized Inertial Measurement Unit and Associated Methods. *United States Patent*, US 8,239,162 B2.

Wang, Y., Li, H., Wan, B., Z. X., y Shan, G. (2018). Obtaining Vital Distances Using Wearable Inertial Measurement Unit for Real-Time, Biomechanical Feedback Training in Hammer-Throw. *Applied Sciences*, 8(12):2470.

Waseem, M. (2020). How to implement classification in machine learning? https://www.edureka.co/blog/classification-in-machine-learning/#svm.

Wisbey, B., Montgomery, P., Pyne, D., y Rattray, B. (2018). Quantifying movement demands of AFL football using GPS tracking. *Journal of Science and Medicine in Sport*, 13(5):531–536.

Zhang, Z. (2007). Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11):218.

5 Cronograma de actividades

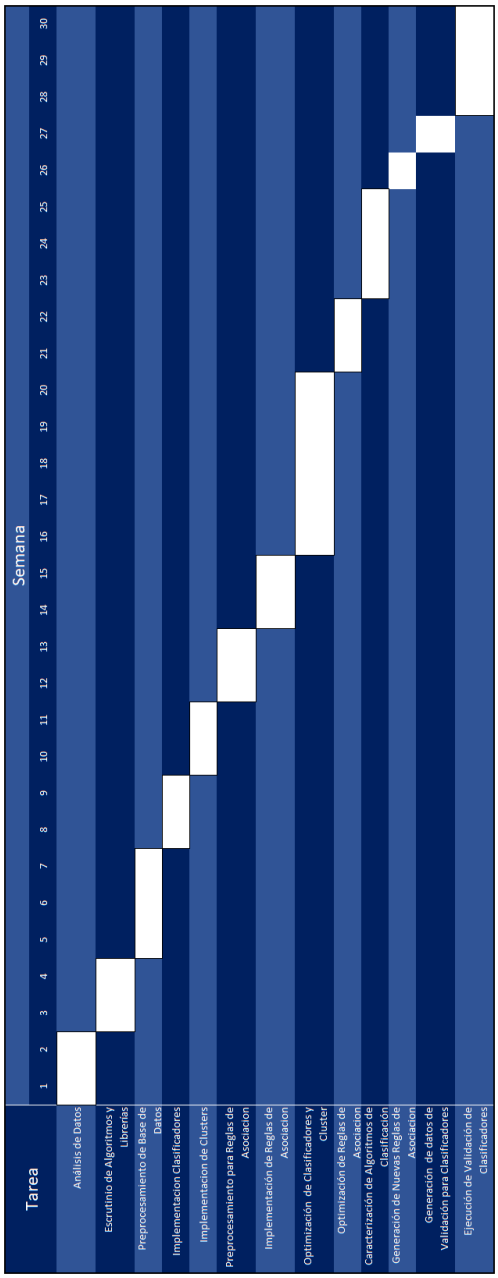


Figura 5.1: Diagrama de Gantt con Cronograma de Actividades

A Resultados de cada uno de los Algoritmos de Clasificación

Como parte de los resultados, se muestra lo que se denomina *"Matriz de Confusión"*. En ella se presenta una comparación entre los resultados que se esperan del algoritmo y el valor obtenido real. Al analizar la información, se espera que todas las comparaciones queden en la diagonal, indicando que el algoritmo fue preciso en su predicción.

Para todos los algoritmos, como parte de los grados de libertad, existe un valor llamado *Random State*, el cual es un valor que se usa como controlador del generador de números aleatorios cada vez que existe algún algoritmo que los necesite. Este dato va a afectar qué tanto se pueden reproducir los resultados según la función que se use. Si el valor que se le introduce es un integer, la aleatoriedad de la función va a ser la misma, sin embargo, es recomendable variar este valor que se escoja para que la estabilidad de los resultados se mantenga. Los datos que más se utilizan suelen ser 0 y 42, por lo que, en nuestro caso, se probaron ambos y se visualizó qué tan buenos resultados daba cada uno. para determinar el mejor.

En todos los algoritmos se va a mostrar entonces primero su comportamiento sin hacer ningún ajuste en los grados de libertad, usando los predeterminados por el algoritmo, y comparando el resultado con el ajuste hecho por medio de la optimización inicial al algoritmo, visualizando los grados de libertad usados, su Matriz de Confusión y luego el porcentaje de precisión de cada uno de ellos. Seguidamente se muestra la comparación entre las 4 posibilidades que hay de optimización al hacer uso del PCA para reducción de columnas. Se muestra exactamente el mismo orden mencionado anteriormente observando al final los diferentes porcentajes de precisión para cada algoritmo.

La librería SKLearn permite también, como se mencionó anteriormente, una función que permite extraer las Columnas (Features) más importantes para el algoritmo. Sin embargo, no todos los algoritmos tienen esta funcionalidad. Se puede observar que, justo luego del primer análisis del algoritmo, cuando no se ha agregado el PCA, las dos tablas que se generan con las columnas, uno para cuando el algoritmo no está optimizado y otra para cuando lo está. Cabe destacar que los algoritmos KNN y NaiveBayes no presentan estos

resultados ya que la librería no extraría las columnas para ellos.

A.1. KNN: K-NEAREST NEIGHBORS

KNN Not Optimized		**KNN Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Neighbors	5	No. of Neighbors	7

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	2	0	1
D	1	0	2
W	3	0	2

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	1	0	0
D	0	1	0
W	0	2	7

Accuracy	**36 %**	*Accuracy*	**82 %**

Cuadro A.1: Resultados del algoritmo KNN

KNN Not Optimized PCA Not Optimized		**KNN Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Neighbors	5	No. of Neighbors	7
No. of Components	3	No. of Components	3

Confusion Matrix

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	0	2	1	L	0	0	1
D	1	0	2	D	0	1	0
W	2	1	2	W	0	3	6

Accuracy	**18 %**	*Accuracy*	**63 %**

KNN Not Optimized PCA Optimized		**KNN Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Neighbors	5	No. of Neighbors	7
No. of Components	31	No. of Components	31

Confusion Matrix

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	2	0	1	L	1	0	0
D	0	0	3	D	0	1	0
W	2	0	3	W	0	2	7

Accuracy	**45 %**	*Accuracy*	**82 %**

Cuadro A.2: Resultados del algoritmo KNN aplicando PCA

A.2. LDA: LINEAR DISCRIMINANT ANALYSIS

LDA Not Optimized		**LDA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	49
Confusion Matrix		*Confusion Matrix*	

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	1	2	0	L	2	1	0
D	0	1	2	D	0	1	0
W	0	2	3	W	0	2	5

Accuracy	**45 %**	*Accuracy*	**73 %**

Cuadro A.3: Resultados del algoritmo LDA

Feature Importance LDA Not Optimized	
Column Name	*Importance (%)*
tigres	100
santos	74.02237
pz	21.56884
Marcador Contr	15.96136
ucr	14.05782
Takeoff [5 - 8]G	12.90918
Jumps AVG Take off(g)	12.84818
Jumps/min	12.33977
Takeoff [8 - 100]G	10.81635
sc	10.53176
Acc Abs [-10 -3]m	9.74238
Vel Rel [95 - 100](m/min)	7.987125
Power met (/min)	7.461599
Acc/min	7.333992
Acc Rel [0 50]/min	7.296411
Landing [3 - 5]G	6.826216
Vel Rel [95 - 100](m)	6.532096
MAX Speed(km/h)	6.474691
Hz*G MAX	6.206101
Distance Dec(m)	5.804419

Cuadro A.4: Tabla de importancia de columnas para LDA sin optimizar

Feature Importance LDA Optimized	
Column Name	*Importance (%)*
santos	100
ucr	38.97173
Marcador Contr	32.20111
heredia	30.72751
tigres	29.92636
visita	28.53675
Jumps AVG Take off(g)	27.36564
Takeoff [8 - 100]G	26.50166
Power met (/min)	17.3678
Acc Abs [-10 -3]m	16.57726
Acc/min	16.32428
Jumps/min	16.24549
Acc Rel [0 50]/min	16.11091
[0-0,5] hz(%)	15.0433
Vel Rel [95 - 100](m)	14.25487
Edi (/min)	13.19631
Vel Abs [19 - 24](m)	12.6464
Vel Abs [19 - 24](m/min)	12.51446
HML (m/min)	10.76198
Vel Rel [45 - 65](m/min)	9.874012

Cuadro A.5: Tabla de importancia de columnas para LDA optimizado

LDA Not Optimized PCA Not Optimized		**LDA Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	49
No. of Components	3	No. of Components	3
Confusion Matrix		*Confusion Matrix*	

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	0	2	1	L	0	1	2
D	0	0	3	D	1	0	0
W	0	2	3	W	0	1	6

Accuracy	**27 %**	*Accuracy*	**55 %**

LDA Not Optimized PCA Optimized		**LDA Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	49
No. of Components	31	No. of Components	31
Confusion Matrix		*Confusion Matrix*	

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	1	2	0	L	2	1	0
D	0	1	2	D	0	1	0
W	0	2	3	W	0	1	6

Accuracy	**45 %**	*Accuracy*	**82 %**

Cuadro A.6: Resultados del algoritmo LDA aplicando PCA

A.3. SVM: SUPPORT VECTOR MACHINE

SVM Not Optimized

Adjustable Values	
Variable	Value
Split Random State	0
SVM Random State	0

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	3	0	0
D	0	1	2
W	1	2	2

Accuracy	**55 %**

SVM Optimized

Adjustable Values	
Variable	Value
Split Random State	9
SVM Random State	0

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	2	0	0
D	1	0	0
W	0	1	7

Accuracy	**82 %**

Cuadro A.7: Resultados del algoritmo SVM

Feature Importance SVM Not Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
Landing [3 - 5]/min	23.4380137
heredia	19.58791446
Takeoff [8 - 100]G	17.96796836
Landing [3 - 5]G	17.00664619
HIBD (m/min)	16.99476756
Sprints ABS	15.59691065
Vel Rel [85 - 95] %.1	13.7825243
Vel Rel [85 - 95] %	13.7825243
Jumps AVG Take off(g)	13.17829793
Dif. ACC/DEC	13.11069778
HIBD (m)	13.03616165
Distance Acc(m)	12.48045123
Acc Abs [3 10]m	12.47795042
MAX Acc(m/s^2)	11.9033541
Vel Rel [0 - 45](m/min)	11.84245926
Sprint Abs(m)	11.36806534
Vel Abs [19 - 24](m/min)	10.7864134
Hz*G AVG	9.920446331
HSR Abs (m/min)	9.767462657

Cuadro A.8: Tabla de importancia de columnas para SVM sin optimizar

Feature Importance SVM Optimized	
Column Name	*Importance (%)*
Marcador Contr	98.74287269
Takeoff [8 - 100]G	47.58951263
Landing [3 - 5]G	28.9028569
Landing [3 - 5]/min	26.81404342
heredia	24.91507309
Acc Rel [50 60]m	24.65355758
Landing [5 - 8]/min	24.51420659
Vel Abs [19 - 24] %	23.8466962
Vel Abs [19 - 24] %.1	23.8466962
visita	22.97247868
Vel Abs [19 - 24](m/min)	21.84731998
AVG Dec(m/s^2)	21.8048912
Vel Abs [19 - 24](m)	21.0690601
Distance Acc(m)	17.21526464
Acc Abs [3 10]m	17.20958312
HSR Abs (m/min)	16.85566112
HSR Abs Dist (m)	16.1807805
% HSR Abs	13.44580568
HIBD (m/min)	11.88501299
Acc Abs [3 10]	11.48220818

Cuadro A.9: Tabla de importancia de columnas para SVM optimizado

SVM Not Optimized PCA Not Optimized		**SVM Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	9
SVM Random State	0	SVM Random State	0
No. of Components	3	No. of Components	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	2	1
D	0	0	3
W	0	1	4

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	1	1
D	0	1	0
W	0	5	3

Accuracy	**36 %**	*Accuracy*	**36 %**

SVM Not Optimized PCA Optimized		**SVM Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	9
SVM Random State	0	SVM Random State	0
No. of Components	32	No. of Components	32

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	1	2	0
D	0	1	2
W	0	2	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	2	0	0
D	1	0	0
W	0	2	6

Accuracy	**45 %**	*Accuracy*	**73 %**

Cuadro A.10: Resultados del algoritmo SVM aplicando PCA

A.4. NB: NAIVE BAYES

NB Not Optimized		**NB Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	4
Confusion Matrix		*Confusion Matrix*	

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	0	1	2	L	2	1	3
D	0	2	1	D	0	1	0
W	2	1	2	W	0	1	3

Accuracy	**36 %**	*Accuracy*	**55 %**

Cuadro A.11: Resultados del algoritmo NB

NB Not Optimized PCA Not Optimized		**NB Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	4
No. of Components	3	No. of Components	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	1	0	2
D	0	1	2
W	1	2	2

Accuracy **36 %**

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	1	5
D	0	1	0
W	0	1	3

Accuracy **36 %**

NB Not Optimized PCA Optimized		**NB Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	42	Split Random State	4
No. of Components	2	No. of Components	2

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	1	0	2
D	0	0	3
W	0	2	3

Accuracy **36 %**

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	0	6
D	0	1	0
W	0	1	3

Accuracy **36 %**

Cuadro A.12: Resultados del algoritmo NB aplicando PCA

A.5. DT: DESICION TREE

DT Not Optimized		**DT Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	8
DT Random State	0	DT Random State	16

Confusion Matrix				*Confusion Matrix*			
	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	1	2	0	L	2	0	0
D	0	3	0	D	0	1	0
W	0	2	3	W	0	1	7

Accuracy	**64 %**	*Accuracy*	**91 %**

Cuadro A.13: Resultados del algoritmo DT

Feature Importance DT Not Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
Hz*G MAX	54.35256401
Vel Abs [19 - 24] %.1	51.25455845
Marcador Casa	47.67414571
HSR Rel Count	32.08101645
Srint ABS AVG Dur (s)	31.42228688
Frec AVG (hz)	24.11536749
Acc Rel [0 50] %	24.11536749

Cuadro A.14: Tabla de importancia de columnas para DT sin optimizar

Feature Importance DT Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
Marcador Casa	90.00272
Vel Abs [12 - 19](m/min)	88.71435
HSR Abs Dist (m)	61.85352
Vel Abs [24 - 50](m)	61.14471
Takeoff [8 - 100]G	54.96987
Dec/min	40.9462
DSL (/min)	35.07879
Impacts [8-100] G	16.93928

Cuadro A.15: Tabla de importancia de columnas para DT optimizado

DT Not Optimized PCA Not Optimized		**DT Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	8
DT Random State	0	DT Random State	16
No. of Components	3	No. of Components	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	2	0	1
D	0	1	2
W	1	1	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	2	0
D	0	1	0
W	1	2	5

Accuracy	**55 %**	*Accuracy*	**55 %**

DT Not Optimized PCA Optimized		**DT Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	8
DT Random State	0	DT Random State	16
No. of Components	4	No. of Components	4

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	2	1
D	0	1	2
W	0	1	4

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	2	0	0
D	0	1	0
W	1	1	6

Accuracy	**45 %**	*Accuracy*	**82 %**

Cuadro A.16: Resultados del algoritmo DT aplicando PCA

A.6. RF: RANDOM FOREST

RF Not Optimized		**RF Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Estimators	100	No. of Estimators	24
RF Random State	42	RF Random State	7
Confusion Matrix		*Confusion Matrix*	

	Expected Val.				Expected Val.		
Obtained	L	D	W	Obtained	L	D	W
L	0	2	1	L	1	0	0
D	0	0	3	D	0	1	0
W	0	0	5	W	0	0	9

Accuracy	**45 %**	*Accuracy*	**100 %**

Cuadro A.17: Resultados del algoritmo RF

Feature Importance RF Not Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
Frec MAX (hz)	47.95888463
Vel Rel [75 - 85](m)	45.9776815
Marcador Casa	44.02861079
DSL	33.14985016
HSR Rel Count	26.75380146
Vel Rel [75 - 85] %.1	26.06342803
Vel Rel [75 - 85] %	24.13314951
HSR Rel (m/min)	23.559381
HSR Abs Count	22.24511409
[1,5-100] hz(%)	22.09402368
Edi max	21.37921552
HSR Abs (m/min)	20.71655284
Vel Abs [19 - 24](m)	20.29750371
HMLD (m)	20.26993592
Jumps AVG Take off(g)	19.91832817
Jumps AVG Landing(g)	19.76534236
Steps count	19.3712307
Impacts [0-3] G	19.19800697
Impacts [8-100] G	18.90917874

Cuadro A.18: Tabla de importancia de columnas para RF sin optimizar

Feature Importance RF Optimized	
Column Name	*Importance (%)*
Marcador Contr	100
MAX Acc(m/s²)	55.52906071
Acc Abs [3 10]m	54.46971513
Takeoff [8 - 100]G	52.12255712
Energy Expenditure	45.35642029
DSL	40.07258677
Vel Rel [75 - 85] %	38.34124415
Landing [8 - 100]G	35.21820303
[0,5-1] hz(%)	33.89600908
heredia	31.70347561
Vel Rel [75 - 85](m)	30.30608627
HIBD (m/min)	30.01920903
[1,5-100] hz(%)	29.77468961
MAX Speed(km/h)	28.26094478
Vel Rel [85 - 95](m)	28.14046849
Impacts [0-3] G	27.84264266
Vel Abs [24 - 50] %	27.51718357
Dec/min	26.48976775
Acc Abs [-10 -3] %	26.03127353
Vel Rel [75 - 85](m/min)	25.1977825

Cuadro A.19: Tabla de importancia de columnas para RF optimizado

RF Not Optimized PCA Not Optimized		**RF Optimized PCA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Estimators	100	No. of Estimators	24
RF Random State	42	RF Random State	7
No. of Components	3	No. of Components	3

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	1	0	2
D	0	1	2
W	0	0	5

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	0	1
D	1	0	0
W	1	2	6

Accuracy	**64 %**	*Accuracy*	**55 %**

RF Not Optimized PCA Optimized		**RF Optimized PCA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Split Random State	0	Split Random State	48
No. of Estimators	100	No. of Estimators	24
RF Random State	42	RF Random State	7
No. of Components	39	No. of Components	39

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	0	3
D	0	0	3
W	0	1	4

Confusion Matrix

	Expected Val.		
Obtained	L	D	W
L	0	1	0
D	0	1	0
W	0	1	8

Accuracy	**36 %**	*Accuracy*	**82 %**

Cuadro A.20: Resultados del algoritmo RF aplicando PCA

B Resultados de cada uno de los Algoritmos de Clustering

En esta sección se muestran todos los resultados para los algoritmos de clustering. De igual manera que para los Clasificadores, los Cluster tienen grados de libertad que pueden ser ajustados para lograr una mayor precisión y de igual forma, la variable **Random State** está presente, por lo que se hizo una optimización para cada uno de los algoritmos para tener una comparación entre cuando se optimizaba y cuando no, sin embargo, esto sólo se puede mostrar inicialmente como dos tablas con el número de cluster de cada partido, ya que , como se mencionó anteriormente, ellos dan como resultado únicamente el valor del cluster al cual cada partido pertenece.

Para poder tener una mejor visualización de las comparaciones, como se mencionó en secciones anteriores, se utilizó el algoritmo LDA para predecir a cuales cluster pertenecía cada partido a partir de los datos previamente obtenidos con los algoritmos de clustering. Se agregó la columna creada por los algoritmos de clustering como parte de las variables de la base de datos y se ejecutó el algoritmo LDA para determinar estas precisiones. Por ende, luego se muestran las 4 diferentes combinaciones entre el algoritmo optimizado y no optimizado, y el LDA optimizado y no optimizado, con los grados de libertad utilizados y la precisión de cada uno, seguidamente de las Matrices de Confusión de cada uno. Finalmente, como el algoritmo LDA puede mostrar las columnas relevantes, se muestra las columnas que fueron relevantes para cada una de las combinaciones en cada uno de los algoritmos.

B.1. KMeans Clustering

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
carmelita	visita	1	0	0
guadalupe	casa	1	3	0
limon	casa	2	2	0
limon	visita	1	1	0
limon	visita	1	2	0
pz	visita	1	1	0
santos	visita	3	5	0
carmelita	visita	2	2	1
cartago	casa	4	1	1
cartago	visita	1	0	1
guadalupe	visita	2	0	1
heredia	visita	0	2	1
limon	casa	4	0	1
pz	casa	2	2	1
pz	visita	2	0	1
santos	visita	3	3	1
sc	casa	3	3	1
ucr	casa	4	0	1
ucr	visita	4	0	1
heredia	casa	2	3	2
carmelita	casa	3	0	3
cartago	casa	1	2	3
grecia	casa	1	2	3
grecia	visita	3	1	3
guadalupe	casa	2	0	3
guadalupe	visita	2	0	3
heredia	casa	1	2	3
heredia	visita	0	1	3
heredia	visita	2	2	3
lda	casa	2	1	3
santos	casa	2	1	3
sc	casa	0	0	3
sc	visita	2	2	3
heredia	casa	1	0	4
sc	visita	2	0	4
ucr	casa	4	0	4
cartago	casa	2	0	5
grecia	casa	4	2	5
grecia	casa	5	0	5
heredia	casa	2	1	5
heredia	visita	0	2	5
lda	visita	0	1	5
lda	visita	1	1	5
pz	casa	2	1	5
pz	visita	1	1	5
pz	visita	3	2	5
sc	visita	1	1	5
carmelita	casa	2	1	6
sc	casa	1	1	6
tigres	casa	1	0	6
tigres	visita	1	5	6
ucr	visita	2	0	6
lda	casa	0	0	7

Cuadro B.1: Clusters determinados con KMeans sin optimizar

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
carmelita	visita	1	0	0
guadalupe	casa	1	3	0
limon	casa	2	2	0
limon	visita	1	1	0
limon	visita	1	2	0
pz	visita	1	1	0
santos	visita	3	5	0
carmelita	visita	2	2	1
cartago	casa	4	1	1
cartago	visita	1	0	1
guadalupe	visita	2	0	1
heredia	visita	0	2	1
limon	casa	4	0	1
pz	casa	2	2	1
pz	visita	2	0	1
santos	visita	3	3	1
sc	casa	3	3	1
ucr	casa	4	0	1
ucr	visita	4	0	1
heredia	casa	2	3	2
carmelita	casa	3	0	3
cartago	casa	1	2	3
grecia	casa	1	2	3
grecia	visita	3	1	3
guadalupe	casa	2	0	3
guadalupe	visita	2	0	3
heredia	casa	1	2	3
heredia	visita	0	1	3
heredia	visita	2	2	3
lda	casa	2	1	3
santos	casa	2	1	3
sc	casa	0	0	3
sc	visita	2	2	3
heredia	casa	1	0	4
sc	visita	2	0	4
ucr	casa	4	0	4
cartago	casa	2	0	5
grecia	casa	4	2	5
grecia	casa	5	0	5
heredia	casa	2	1	5
heredia	visita	0	2	5
lda	visita	0	1	5
lda	visita	1	1	5
pz	casa	2	1	5
pz	visita	1	1	5
pz	visita	3	2	5
sc	visita	1	1	5
carmelita	casa	2	1	6
sc	casa	1	1	6
tigres	casa	1	0	6
tigres	visita	1	5	6
ucr	visita	2	0	6
lda	casa	0	0	7

Cuadro B.2: Clusters determinados con KMeans optimizado

KMeans Not Optimized LDA Not Optimized		**KMeans Optimized LDA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
Kmeans Random State	42	Kmeans Random State	24
No. of Clusters	8	No. of Clusters	8
LDA Random State	42	LDA Random State	42
Accuracy	**81 %**	*Accuracy*	**81 %**

KMeans Not Optimized LDA Optimized		**KMeans Optimized LDA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
KMeans Random State	42	KMeans Random State	24
No. of Clusters	8	No. of Clusters	8
LDA Random State	5	LDA Random State	5
Accuracy	**72 %**	*Accuracy*	**100 %**

Cuadro B.3: Resultados del algoritmo KMeans aplicando PCA

		Cluster Esperado						
		C0	C1	C2	C3	C4	C5	C6
Cluster Obtenido	C0	1	0	0	0	0	0	0
	C1	0	2	0	0	0	0	0
	C2	0	0	0	1	0	0	0
	C3	0	1	0	2	0	0	0
	C4	0	0	0	0	1	0	0
	C5	0	0	0	0	0	2	0
	C6	0	0	0	0	0	0	1

Cuadro B.4: Matriz de Confusion KMeans/LDA no optimizados

		Cluster Esperado						
		C0	C1	C2	C3	C4	C5	C6
Cluster Obtenido	C0	1	0	0	0	0	0	0
	C1	0	1	0	0	0	0	0
	C2	0	0	1	0	0	0	0
	C3	0	0	0	0	0	0	1
	C4	1	0	0	0	0	0	0
	C5	0	0	0	0	0	2	0
	C6	0	0	0	0	0	0	4

Cuadro B.5: Matriz de Confusion KMeans optimizado / LDA no optimizado

		Cluster Esperado					
		C0	C1	C3	C4	C5	C6
Cluster Obtenido	C0	2	0	0	0	0	0
	C1	0	2	2	0	0	0
	C3	0	0	2	0	0	0
	C4	0	0	0	1	0	0
	C5	0	0	0	0	1	1
	C6	0	0	0	0	0	0

Cuadro B.6: Matriz de Confusion KMeans no optimizado / LDA optimizado

		Cluster Esperado				
		C0	C1	C2	C5	C6
Cluster Obtenido	C0	2	0	0	0	0
	C1	0	2	0	0	0
	C2	0	0	1	0	0
	C5	0	0	0	2	0
	C6	0	0	0	0	4

Cuadro B.7: Matriz de Confusion KMeans / LDA optimizado

Feature Importance KM/LDA Not Optimized	
Column Name	*Importance (%)*
cartago	100
sc	71.41943723
lda	37.55556379
casa	30.01360235
HSR Abs (m/min)	29.67858057
Sprint Abs(m)	25.24695142
Sprints ABS	23.68903636
Sprints REL	22.77876135
Vel Rel [95 - 100](m/min)	21.45612637
Steps/min	21.24325942
heredia	20.56581425
Sprint REL AVG Dur (s)	19.16919008
Landing [5 - 8]G	17.56584253
Acc Abs [-10 -3] %	17.33134674
Takeoff [8 - 100]/min	16.68402538
Vel Rel [75 - 85](m/min)	15.83599703
Acc Rel [50 60]/min	15.43682287
Impacts [0-3] G	14.76144664
Power met Avg	14.6360122
Vel Rel [95 - 100](m)	14.4938135

Cuadro B.8: Tabla de importancia de columnas para KMeans/LDA no optimizado

Feature Importance KM Opt./LDA Not Opt.	
Column Name	*Importance (%)*
cartago	100
sc	68.58163462
lda	46.22312242
grecia	37.81557844
Sprints REL	30.26693145
Sprint Abs(m)	27.54801291
casa	25.3775284
Sprint REL AVG Dur (s)	25.35673264
HSR Abs (m/min)	23.66905735
Takeoff [8 - 100]/min	21.51114708
Vel Rel [95 - 100](m/min)	18.87945369
MAX Acc(m/s^2)	18.83733657
Impacts [0-3]/min	18.31101352
Takeoff [0 - 3]/min	17.14394126
HSR Rel Dist (m)	16.51569317
Acc Rel [50 60]/min	16.35396857
Vel Rel [75 - 85](m/min)	14.6132522
Impacts [0-3] G	13.58825418
Distance Acc(m)	13.38539223
Vel Rel [0 - 45](m/min)	13.29948059

Cuadro B.9: Tabla de importancia de columnas para KMeans optimizado y LDA no optimizado

Feature Importance KM Not Opt./LDA Opt.	
Column Name	*Importance (%)*
limon	100
Energy Expenditure	88.1858298
ucr	76.08504134
Vel Rel [95 - 100](m/min)	67.05867007
Power met	49.69413527
HIBD (m)	44.72957886
HSR Abs Dist (m)	41.51822735
Explosive Dist(m)	40.27185503
MAX Speed(km/h)	40.17558477
Vel Rel [0 - 45](m/min)	40.05372314
Vel Abs [24 - 50] %	36.60291359
[0-0,5] hz(%)	35.25422477
Vel Rel [95 - 100](m)	33.12883868
pz	32.15203935
Dist (m/min)	29.9990772
MAX Acc(m/s^2)	29.72977431
Jumps AVG Take off(g)	28.44032045
Takeoff [8 - 100]G	27.20712846
Vel Rel [75 - 85](m/min)	26.82785712
HML (m/min)	26.31631833

Cuadro B.10: Tabla de importancia de columnas para KMeans no optimizado y LDA optimizado

Feature Importance KM/LDA Optimized	
Column Name	*Importance (%)*
lda	79.30762405
grecia	65.20539046
casa	56.22154174
HSR Abs Dist (m)	52.43552797
santos	42.33961243
Sprint Abs(m)	42.17739716
Vel Rel [95 - 100](m/min)	40.89293377
Acc Abs [-10 -3] %	40.84245855
Takeoff [8 - 100]G	40.54150187
Power met	40.03546309
MAX Acc(m/s^2)	39.09847084
sc	37.39637376
HSR Rel Dist (m)	36.56580581
Energy Expenditure	36.19535355
Vel Rel [0 - 45](m/min)	35.81002968
Takeoff [5 - 8]G	34.8233967
Step Balance(%)	33.8326272
Sprints REL	31.78371614
Vel Abs [12 - 19](m/min)	31.56941736
Acc Abs [3 10] %	30.08245714

Cuadro B.11: Tabla de importancia de columnas para KMeans y LDA optimizado

B.2. Gaussian Mixture Model Clustering GMM

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
sc	visita	2	0	0
heredia	casa	1	0	0
ucr	casa	4	0	0
carmelita	casa	3	0	1
carmelita	visita	2	2	1
grecia	casa	1	2	1
grecia	visita	3	1	1
guadalupe	casa	2	0	1
guadalupe	visita	2	0	1
heredia	casa	1	2	1
heredia	visita	0	1	1
heredia	visita	0	2	1
heredia	visita	2	2	1
lda	casa	2	1	1
santos	casa	2	1	1
sc	casa	0	0	1
sc	visita	2	2	1
ucr	casa	4	0	1
ucr	visita	4	0	1
cartago	casa	2	0	2
grecia	casa	4	2	2
grecia	casa	5	0	2
guadalupe	visita	2	0	2
heredia	casa	2	1	2
heredia	visita	0	2	2
lda	visita	0	1	2
lda	visita	1	1	2
pz	casa	2	1	2
pz	visita	1	1	2
pz	visita	3	2	2
santos	visita	3	5	2
sc	visita	1	1	2
heredia	casa	2	3	3
carmelita	casa	2	1	4
carmelita	visita	1	0	4
cartago	casa	1	2	4
guadalupe	casa	1	3	4
limon	casa	2	2	4
limon	visita	1	2	4
pz	visita	1	1	4
sc	casa	1	1	4
tigres	casa	1	0	4
tigres	visita	1	5	4
ucr	visita	2	0	4
lda	casa	0	0	5
cartago	casa	4	1	6
cartago	visita	1	0	6
limon	casa	4	0	6
pz	casa	2	2	6
pz	visita	2	0	6
santos	visita	3	3	6
sc	casa	3	3	6
limon	visita	1	1	7

Cuadro B.12: Clusters determinados con GMM sin optimizar

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
carmelita	visita	2	2	0
cartago	casa	4	1	0
cartago	visita	1	0	0
grecia	casa	1	2	0
heredia	visita	0	1	0
heredia	visita	0	2	0
heredia	visita	2	2	0
lda	casa	0	0	0
limon	casa	4	0	0
pz	casa	2	2	0
pz	visita	2	0	0
santos	visita	3	3	0
sc	casa	3	3	0
cartago	casa	2	0	1
grecia	casa	4	2	1
grecia	casa	5	0	1
guadalupe	visita	2	0	1
heredia	casa	2	1	1
heredia	visita	0	2	1
lda	casa	2	1	1
lda	visita	0	1	1
lda	visita	1	1	1
pz	casa	2	1	1
pz	visita	1	1	1
pz	visita	3	2	1
sc	visita	1	1	1
heredia	casa	2	3	2
carmelita	visita	1	0	3
guadalupe	casa	1	3	3
limon	casa	2	2	3
limon	visita	1	1	3
limon	visita	1	2	3
santos	visita	3	5	3
carmelita	casa	3	0	4
grecia	visita	3	1	4
guadalupe	casa	2	0	4
guadalupe	visita	2	0	4
heredia	casa	1	2	4
santos	casa	2	1	4
sc	casa	0	0	4
sc	visita	2	2	4
ucr	casa	4	0	4
ucr	visita	4	0	4
carmelita	casa	2	1	5
ucr	visita	2	0	5
heredia	casa	1	0	6
sc	visita	2	0	6
ucr	casa	4	0	6
cartago	casa	1	2	7
pz	visita	1	1	7
sc	casa	1	1	7
tigres	casa	1	0	7
tigres	visita	1	5	7

Cuadro B.13: Clusters determinados con GMM optimizado

GMM Not Optimized LDA Not Optimized		**GMM Optimized LDA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
GMM Random State	0	GMM Random State	23
No. of Clusters	8	No. of Clusters	8
LDA Random State	0	LDA Random State	0
Accuracy	**81 %**	*Accuracy*	**100 %**

GMM Not Optimized LDA Optimized		**GMM Optimized LDA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
GMM Random State	0	GMM Random State	23
No. of Clusters	8	No. of Clusters	8
LDA Random State	5	LDA Random State	5
Accuracy	**100 %**	*Accuracy*	**81 %**

Cuadro B.14: Resultados del algoritmo GMM aplicando PCA

		Cluster Esperado			
		C1	C2	C4	C6
Cluster Obtenido	C1	3	0	0	0
	C2	0	2	0	0
	C4	0	1	2	0
	C6	1	0	0	2

Cuadro B.15: Matriz de Confusion GMM/LDA no optimizados

		Cluster Esperado				
		C0	C1	C3	C4	C7
Cluster Obtenido	C0	3	0	0	0	0
	C1	0	2	0	0	0
	C3	0	0	2	0	0
	C4	0	0	0	3	0
	C7	0	0	0	0	1

Cuadro B.16: Matriz de Confusion GMM optimizado / LDA no optimizado

		Cluster Esperado				
		C0	C1	C4	C5	C6
Cluster Obtenido	C0	1	0	0	0	0
	C1	0	4	0	0	0
	C3	0	0	2	0	0
	C5	0	0	0	2	0
	C6	0	0	0	0	2

Cuadro B.17: Matriz de Confusion GMM no optimizado / LDA optimizado

		Cluster Esperado				
		C0	C1	C3	C4	C6
Cluster Obtenido	C0	2	0	0	0	0
	C1	0	2	0	0	0
	C3	0	0	2	0	0
	C4	1	0	0	3	0
	C6	0	1	0	0	0

Cuadro B.18: Matriz de Confusion GMM / LDA optimizado

Feature Importance GMM/LDA Not Optimized	
Column Name	*Importance (%)*
limon	100
heredia	29.18800102
grecia	21.01628098
Vel Abs [24 - 50] %	20.57095927
Impacts [5-8] G	19.2158671
Dist (m/min)	19.07701194
Impacts [5-8]/min	18.28312643
Acc Abs [-10 -3] %	16.15093501
Vel Rel [85 - 95] %	14.53350779
Vel Rel [85 - 95] %.1	14.53350779
Edi max	14.33823882
Acc Abs [3 10]/min	13.99165163
Acc/min	13.99165163
Vel Rel [95 - 100](m/min)	13.91094094
Power met	12.90099847
MAX Speed(km/h)	12.60013366
Steps/min	12.37103491
MAX Acc(m/s^2)	11.46942423
HIBD (m)	11.4683906
Takeoff [0 - 3]/min	10.15299306

Cuadro B.19: Tabla de importancia de columnas para GMM/LDA no optimizado

Feature Importance GMM Opt./LDA Not Opt.	
Column Name	*Importance (%)*
santos	100
Vel Rel [95 - 100](m/min)	69.87526164
pz	64.80196304
Hz*G MAX	49.19008235
Vel Abs [12 - 19](m/min)	45.57097307
MAX Speed(km/h)	43.23262649
Power met	42.74804856
casa	42.35633468
Vel Abs [24 - 50] %	41.75469306
Vel Rel [0 - 45](m/min)	38.70213109
lda	36.84882498
Acc Abs [3 10]/min	35.82398987
Acc/min	35.82398987
Impacts [0-3]/min	34.8196865
Dist (m/min)	33.89253511
Acc Abs [-10 -3] %	33.33104007
Acc Rel [50 60]	27.78710798
Energy Expenditure	27.51588698
Vel Rel [45 - 65](m/min)	27.13837397
HSR Abs Dist (m)	25.85481269

Cuadro B.20: Tabla de importancia de columnas para GMM optimizado y LDA no optimizado

Feature Importance GMM Not Opt./LDA Opt.	
Column Name	*Importance (%)*
limon	100
sc	38.90190064
HSR Rel Count	22.00744159
Energy Expenditure	21.31232035
Vel Abs [24 - 50] %	20.80121632
Vel Rel [95 - 100](m/min)	19.87183884
Power met	19.77960277
Impacts [5-8] G	19.64316597
Impacts [5-8]/min	18.70548536
Dec/min	18.53866038
Landing [5 - 8]G	18.52449362
MAX Speed(km/h)	17.87623409
Dist (m/min)	17.80009829
grecia	17.24751329
Acc Abs [-10 -3] %	16.69896007
MAX Acc(m/s^2)	16.68824621
santos	16.16670377
[0,5-1] hz(%)	15.40316599
Vel Rel [65 - 75] %	14.57193017
Vel Rel [65 - 75] %.1	14.57193017

Cuadro B.21: Tabla de importancia de columnas para GMM no optimizado y LDA optimizado

Feature Importance GMM/LDA Optimized	
Column Name	*Importance (%)*
santos	94.42745455
Vel Rel [95 - 100](m/min)	87.27217216
casa	52.05801949
carmelita	50.85689195
grecia	48.84921516
Sprint Abs(m)	46.94275503
Vel Rel [0 - 45](m/min)	44.22171711
HSR Abs Dist (m)	43.11592238
Step Balance(%)	42.39563052
Vel Abs [12 - 19](m/min)	41.2336533
Takeoff [8 - 100]G	41.14752508
Takeoff [5 - 8]G	37.94455225
Sprints REL	35.78685297
lda	34.11768153
Vel Abs [24 - 50] %	32.76679433
Energy Expenditure	31.6747608
Acc Abs [-10 -3]m	29.05484355
tigres	27.86911882
Landing [5 - 8]G	27.43747494
Vel Rel [95 - 100] %.1	27.39837282

Cuadro B.22: Tabla de importancia de columnas para GMM y LDA optimizado

B.3. Hierarchical Clustering

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
carmelita	casa	3	0	0
carmelita	visita	2	2	0
cartago	visita	1	0	0
grecia	casa	1	2	0
grecia	visita	3	1	0
guadalupe	casa	2	0	0
guadalupe	visita	2	0	0
heredia	casa	1	2	0
heredia	visita	0	1	0
heredia	visita	0	2	0
heredia	visita	2	2	0
lda	casa	2	1	0
limon	casa	4	0	0
santos	casa	2	1	0
santos	visita	3	3	0
sc	casa	0	0	0
sc	visita	2	2	0
ucr	casa	4	0	0
ucr	visita	4	0	0
carmelita	visita	1	0	1
cartago	casa	2	0	1
grecia	casa	4	2	1
guadalupe	casa	1	3	1
heredia	casa	2	1	1
heredia	visita	0	2	1
lda	visita	0	1	1
lda	visita	1	1	1
limon	casa	2	2	1
pz	casa	2	1	1
pz	visita	1	1	1
pz	visita	3	2	1
santos	visita	3	5	1
sc	visita	1	1	1
cartago	casa	4	1	2
lda	casa	0	0	2
pz	casa	2	2	2
pz	visita	2	0	2
sc	casa	3	3	2
limon	visita	1	1	3
pz	visita	1	1	3
cartago	casa	1	2	4
grecia	casa	5	0	4
guadalupe	visita	2	0	4
heredia	casa	1	0	4
sc	visita	2	0	4
ucr	casa	4	0	4
heredia	casa	2	3	5
limon	visita	1	2	6
carmelita	casa	2	1	7
sc	casa	1	1	7
tigres	casa	1	0	7
tigres	visita	1	5	7
ucr	visita	2	0	7

Cuadro B.23: Clusters determinados con HC sin optimizar

Partido	Ubicación	Gol.Sap.	Gol.Cont.	Cluster
carmelita	casa	2	1	0
limon	visita	1	2	0
sc	casa	1	1	0
tigres	casa	1	0	0
tigres	visita	1	5	0
ucr	visita	2	0	0
carmelita	visita	1	0	1
cartago	casa	2	0	1
grecia	casa	4	2	1
guadalupe	casa	1	3	1
heredia	casa	2	1	1
heredia	visita	0	2	1
lda	visita	0	1	1
lda	visita	1	1	1
limon	casa	2	2	1
limon	visita	1	1	1
pz	casa	2	1	1
pz	visita	1	1	1
pz	visita	1	1	1
pz	visita	3	2	1
santos	visita	3	5	1
sc	visita	1	1	1
cartago	casa	4	1	2
lda	casa	0	0	2
pz	casa	2	2	2
pz	visita	2	0	2
sc	casa	3	3	2
carmelita	casa	3	0	3
carmelita	visita	2	2	3
cartago	visita	1	0	3
grecia	casa	1	2	3
grecia	visita	3	1	3
guadalupe	casa	2	0	3
guadalupe	visita	2	0	3
heredia	casa	1	2	3
heredia	visita	0	1	3
heredia	visita	0	2	3
heredia	visita	2	2	3
lda	casa	2	1	3
limon	casa	4	0	3
santos	casa	2	1	3
santos	visita	3	3	3
sc	casa	0	0	3
sc	visita	2	2	3
ucr	casa	4	0	3
ucr	visita	4	0	3
cartago	casa	1	2	4
grecia	casa	5	0	4
guadalupe	visita	2	0	4
heredia	casa	1	0	4
sc	visita	2	0	4
ucr	casa	4	0	4
heredia	casa	2	3	5

Cuadro B.24: Clusters determinados con HC optimizado

HC Not Optimized LDA Not Optimized		**HC Optimized LDA Not Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
No. of Clusters	8	No. of Clusters	6
LDA Random State	0	LDA Random State	0
Accuracy	**91 %**	*Accuracy*	**72 %**

HC Not Optimized LDA Optimized		**HC Optimized LDA Optimized**	
Adjustable Values		*Adjustable Values*	
Variable	Value	Variable	Value
No. of Clusters	8	No. of Clusters	6
LDA Random State	19	LDA Random State	19
Accuracy	**91 %**	*Accuracy*	**91 %**

Cuadro B.25: Resultados del algoritmo HC aplicando PCA

		Cluster Esperado					
		C0	C1	C2	C4	C5	C7
Cluster Obtenido	C0	4	0	0	0	0	0
	C1	0	3	0	0	0	0
	C2	0	0	1	0	0	0
	C4	0	0	0	1	0	0
	C5	1	0	0	0	0	0
	C7	0	0	0	0	0	1

Cuadro B.26: Matriz de Confusion HC/LDA no optimizados

		Cluster Esperado				
		C0	C1	C2	C3	C4
Cluster Obtenido	C0	0	1	0	0	0
	C1	1	2	0	0	0
	C2	0	0	1	1	0
	C3	0	0	0	4	0
	C4	0	0	0	0	1

Cuadro B.27: Matriz de Confusion HC optimizado / LDA no optimizado

		Cluster Esperado			
		C0	C1	C2	C7
Cluster Obtenido	C0	6	0	1	0
	C1	0	1	0	0
	C2	0	0	2	0
	C7	0	0	0	1

Cuadro B.28: Matriz de Confusion HC no optimizado / LDA optimizado

		Cluster Esperado			
		C0	C1	C2	C3
Cluster Obtenido	C0	1	0	0	0
	C1	0	1	0	0
	C2	0	0	1	1
	C3	0	0	0	7

Cuadro B.29: Matriz de Confusion HC / LDA optimizado

Feature Importance HC/LDA Not Optimized	
Column Name	*Importance (%)*
Explosive Dist(m/min)	58.66618
HIBD (m)	52.46508
guadalupe	43.84473
pz	42.43665
ucr	34.65585
Power met (/min)	30.6335
Acc Abs [3 10]m	30.57887
Vel Abs [24 - 50] %.1	23.49782
Acc Abs [-10 -3]	23.46081
Locacion	22.72925
visita	21.10531
Distance Acc(m)	20.00125
Takeoff [0 - 3]/min	19.16408
limon	19.05571
Power met	18.93938
MAX Speed(km/h)	18.45793
Landing [5 - 8]G	16.60854
Takeoff [5 - 8]G	16.56237
Decelerations	15.86589
Acc Abs [-10 -3]/min	15.74944

Cuadro B.30: Tabla de importancia de columnas para HC/LDA no optimizado

Feature Importance HC Opt./LDA Not Opt.	
Column Name	*Importance (%)*
tigres	100
Power met	51.1727726
MAX Speed(km/h)	49.27713938
HSR Abs Dist (m)	46.11195771
Vel Abs [12 - 19](m/min)	44.44958771
santos	40.1615175
Acc/min	38.10945603
Acc Abs [3 10]/min	38.10945603
HIBD (m)	35.02730149
Hz*G MAX	30.15609515
Vel Rel [0 - 45](m/min)	28.72028008
carmelita	27.29712736
Vel Rel [95 - 100](m/min)	26.94722552
Dec/min	26.19820697
ucr	25.57119209
Energy Expenditure	24.05848747
% HSR Rel	23.45761644
casa	22.99212986
grecia	21.64699179
Impacts [5-8] G	21.17678699

Cuadro B.31: Tabla de importancia de columnas para HC optimizado y LDA no optimizado

Feature Importance HC Not Opt./LDA Opt.	
Column Name	*Importance (%)*
HIBD (m)	88.04592402
pz	73.36561124
Explosive Dist(m/min)	53.77740864
guadalupe	49.17778498
santos	37.23136803
Power met (/min)	34.43412262
Acc Abs [3 10]m	29.23961474
Energy Expenditure	25.81602193
casa	25.71271386
Acc Abs [3 10] %	25.11509419
Vel Abs [24 - 50] %	23.39493907
Explosive Dist(m)	23.16429266
DSL	21.24570058
[1-1,5] hz(%)	20.08218498
Sprint Abs(m)	19.48812298
MAX Speed(km/h)	19.15173445
Acc Rel [0 50] %	18.67970714
heredia	18.45058756
Dist (m/min)	17.89121023
Srint ABS AVG Dur (s)	17.09614862

Cuadro B.32: Tabla de importancia de columnas para HC no optimizado y LDA optimizado

Feature Importance HC/LDA Optimized	
Column Name	*Importance (%)*
guadalupe	100
santos	83.64043
Explosive Dist(m)	52.08685
casa	47.87356
tigres	38.18012
Energy Expenditure	37.89672
DSL	35.3476
Vel Abs [24 - 50] %	33.47727
HML (ms)	33.20898
pz	32.11833
sc	30.72294
HIBD (m)	28.59718
Dist (m/min)	27.02806
Acc Abs [3 10] %	26.85626
Power met Avg	24.07125
Steps count	22.34456
Vel Rel [0 - 45](m/min)	22.0718
[1-1,5] hz(%)	21.94397
Landing [8 - 100]G	21.52843
Marcador Casa	21.38068

Cuadro B.33: Tabla de importancia de columnas para HC y LDA optimizado

C Resultados de cada uno de los Algoritmos de Reglas de Asociación

En esta sección se muestra los resultados generados por los algoritmos de Reglas de Asociación. Se muestran únicamente las primeras reglas que determinan que el equipo gana, ya que el total de reglas son en el orden de los cientos. Esto es un filtrado del total de reglas que hay para mostrar únicamente las que están relacionadas con Ganar, Perder o Empatar, por lo que a este documento se adjuntan las dos tablas originales en Excel para poder hacer más filtrados según los requerimientos que haya. También, sólo muestra las reglas de cuando el equipo gana ya que, al tener todas las columnas, habría que ejecutar los algoritmos con las variables ajustadas a un punto muy bajo para obtener las reglas de cuando se Empata o Pierde, lo que implicaría un gasto computacional enorme y muchísimo tiempo de ejecución, por ende, se limitó a obtener únicamente las reglas de cuando se ganase.

C.1. Reglas de Asociación dadas por el Algoritmo Apriori

Se puede observar en la Tabla C.1

C.2. Reglas de Asociación dadas por el Algoritmo FP-Growth

Se puede observar en la Tabla C.2

Association Rules created by Apriori Algorithm				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
Takeoff [3 - 5]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.396226	0.75	1.528846
Takeoff [3 - 5]/min[0.01 0.01]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.396226	0.75	1.528846
Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.818182	1.667832
Vel Rel [95 - 100](m/min)[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.75	1.528846
Takeoff [8 - 100]/min[0. 0.]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.818182	1.667832
Vel Rel [95 - 100](m/min)[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.75	1.528846
Vel Rel [75 - 85] %.1[0.49 0.62]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Vel Rel [75 - 85] %[0.49 0.62]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Vel Rel [95 - 100] %.1[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100] %[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Vel Rel [75 - 85] %[0.49 0.62]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Edi (/min)[0.01 0.01]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.809524	1.650183
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100] %.1[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %[0.49 0.62]', 'Takeoff [8 - 100]/min[0. 0.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [75 - 85] %.1[0.49 0.62]', 'Vel Rel [75 - 85] %[0.49 0.62]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Edi (/min)[0.01 0.01]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.809524	1.650183
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175

Cuadro C.1: Tabla de Reglas Apriori que determinan cuando el equipo Gana

Association Rules created by FP-Growth Algorithm				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
Takeoff [3 - 5]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.396226	0.75	1.528846
Takeoff [3 - 5]/min[0.01 0.01]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.396226	0.75	1.528846
Vel Rel [95 - 100](m/min)[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.75	1.528846
Vel Rel [95 - 100](m/min)[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.75	1.528846
Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.818182	1.667832
Takeoff [8 - 100]/min[0. 0.]', 'Landing [5 - 8]/min[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.339623	0.818182	1.667832
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Takeoff [3 - 5]/min[0.01 0.01]', 'Sprint Rel(m)[0. 0.92]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Vel Rel [95 - 100](m/min)[0. 0.]', 'Sprint Rel(m)[0. 0.92]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Edi (/min)[0.01 0.01]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.73913	1.506689
Vel Rel [95 - 100]%.1[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%.1[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Vel Rel [95 - 100]%.1[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Vel Rel [95 - 100]%.1[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%[0. 0.]', 'Vel Rel [95 - 100](m/min)[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%[0. 0.]', 'Vel Rel [95 - 100]%.1[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100](m/min)[0. 0.]', 'Vel Rel [95 - 100]%[0. 0.]', 'Takeoff [8 - 100]/min[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175
Vel Rel [95 - 100]%[0. 0.]', 'Vel Rel [95 - 100]%.1[0. 0.]', 'Vel Rel [95 - 100](m/min)[0. 0.]', 'Marcador Contr[0. 1.]	Resultado[1. 1.]	0.320755	0.772727	1.575175

Cuadro C.2: Tabla de Reglas FP-Growth que determinan cuando el equipo Gana

D Reglas de Asociación de sólo Columnas Importantes

Una vez hecho el filtrado de las columnas, haciendo uso del algoritmo clasificador ganador y subdividiendo la base de datos entre Ganar-NoGanar y Empatar-Perder, se obtuvieron un set de reglas más manejables las cuales se muestran en la siguiente sección. Se muestra primero todas las reglas generadas que indican cuando el equipo Gana (en Verde), luego todas las reglas para cuando el equipo Empata (en Amarillo) y finalmente para cuando el equipo Pierde (en Rojo). Las tablas vienen ordenadas por Support de mayor a menor.

D.1. Reglas de Asociación determinadas con Algoritmo Apriori

- Cuando el equipo **Gana**: Tabla D.1
- Cuando el equipo **Empata**: Tabla D.2
- Cuando el equipo **Pierde**: Tabla D.3

D.2. Reglas de Asociación determinadas con Algoritmo FP-Growth

- Cuando el equipo **Gana**: Tabla D.4
- Cuando el equipo **Empata**: Tabla D.5
- Cuando el equipo **Pierde**: Tabla D.6

Reglas que determinan si el equipo GANA				
Antecedents	**Concequents**	**Support**	**Confidence**	**Lift**
Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.45283	0.705882	1.438914
Impacts [0-3] G[10093.23 12253.58]'	Resultado[1. 1.]	0.301887	0.5	1.019231
Impacts [0-3] G[10093.23 12253.58]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.301887	0.727273	1.482517
HSR Abs (m/min)[3.75 4.55]'	Resultado[1. 1.]	0.283019	0.576923	1.176036
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.283019	0.882353	1.798643
HSR Rel Count[3.77 4.54]'	Resultado[1. 1.]	0.226415	0.545455	1.111888
Marcador Casa[2. 2.]'	Resultado[1. 1.]	0.226415	0.666667	1.358974
Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.226415	0.571429	1.164835
Marcador Casa[2. 2.]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.226415	1	2.038462
Vel Rel [45 - 65] %[9.11 10.09]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.226415	0.923077	1.881657
Hz*G MAX[4.04 4.29]'	Resultado[1. 1.]	0.207547	0.6875	1.401442
Marcador Contr[0. 1.]', 'HSR Rel Count[3.77 4.54]'	Resultado[1. 1.]	0.207547	0.733333	1.494872
Impacts [0-3] G[12489.85 14153.08]'	Resultado[1. 1.]	0.188679	0.526316	1.072874
Marcador Casa[3. 5.]'	Resultado[1. 1.]	0.188679	0.769231	1.568047
Vel Rel [45 - 65] %[8.05 8.81]'	Resultado[1. 1.]	0.188679	0.5	1.019231
HSR Abs (m/min)[3.75 4.55]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.188679	0.625	1.274038
Hz*G MAX[4.04 4.29]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.188679	0.714286	1.456044
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.188679	0.909091	1.853147
HSR Abs (m/min)[2.7 3.64]'	Resultado[1. 1.]	0.169811	0.428571	0.873626
HSR Rel Count[4.77 5.54]'	Resultado[1. 1.]	0.169811	0.75	1.528846

Cuadro D.1: Tabla Reglas Apriori después de Caracterización que indican si el equipo Gana

Reglas que determinan si el equipo Empata				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
HSR Abs (m/min)[2.7 3.64]'	Resultado[0. 0.]	0.150943	0.380952	1.346032
HSR Rel Count[2.08 3.69]'	Resultado[0. 0.]	0.113208	0.4	1.413333
Impacts [0-3] G[10093.23 12253.58]'	Resultado[0. 0.]	0.169811	0.28125	0.99375
Marcador Contr[0. 1.]'	Resultado[0. 0.]	0.150943	0.235294	0.831373
Marcador Contr[2. 3.]'	Resultado[0. 0.]	0.132075	0.411765	1.454902
Marcador Casa[0. 1.]'	Resultado[0. 0.]	0.150943	0.363636	1.284848
MAX Dec(m/sÂ²) [-6.38 -6.06]'	Resultado[0. 0.]	0.113208	0.428571	1.514286
HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]'	Resultado[0. 0.]	0.113208	0.428571	1.514286
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]'	Resultado[0. 0.]	0.113208	0.545455	1.927273
Marcador Casa[0. 1.]', 'Marcador Contr[0. 1.]'	Resultado[0. 0.]	0.150943	0.571429	2.019048
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]'	Resultado[0. 0.]	0.113208	0.666667	2.355556

Cuadro D.2: Tabla Reglas Apriori después de Caracterización que indican si el equipo Empata

Reglas que determinan si el equipo Pierde				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
HSR Abs (m/min)[3.75 4.55]’	Resultado[-1. -1.]	0.132075	0.269231	1.189103
HSR Rel Count[2.08 3.69]’	Resultado[-1. -1.]	0.113208	0.4	1.766667
Impacts [0-3] G[10093.23 12253.58]’	Resultado[-1. -1.]	0.132075	0.21875	0.966146
Marcador Contr[2. 3.]’	Resultado[-1. -1.]	0.150943	0.470588	2.078431
Marcador Casa[0. 1.]’	Resultado[-1. -1.]	0.188679	0.454545	2.007576
Vel Rel [45 - 65]%[8.05 8.81]’	Resultado[-1. -1.]	0.113208	0.3	1.325
HSR Abs (m/min)[3.75 4.55]’, ’Marcador Casa[0. 1.]’	Resultado[-1. -1.]	0.132075	0.7	3.091667
Impacts [0-3] G[10093.23 12253.58]’, ’Marcador Casa[0. 1.]’	Resultado[-1. -1.]	0.113208	0.545455	2.409091
Marcador Casa[0. 1.]’, ’Marcador Contr[2. 3.]’	Resultado[-1. -1.]	0.132075	1	4.416667

Cuadro D.3: Tabla Reglas Apriori después de Caracterización que indican si el equipo Pierde

Reglas que indican si el equipo Gana				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.45283	0.705882	1.438914
Impacts [0-3] G[10093.23 12253.58]'	Resultado[1. 1.]	0.301887	0.5	1.019231
Impacts [0-3] G[10093.23 12253.58]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.301887	0.727273	1.482517
HSR Abs (m/min)[3.75 4.55]'	Resultado[1. 1.]	0.283019	0.576923	1.176036
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.283019	0.882353	1.798643
HSR Rel Count[3.77 4.54]'	Resultado[1. 1.]	0.226415	0.545455	1.111888
Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.226415	0.571429	1.164835
Vel Rel [45 - 65] %[9.11 10.09]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.226415	0.923077	1.881657
Marcador Casa[2. 2.]'	Resultado[1. 1.]	0.226415	0.666667	1.358974
Marcador Casa[2. 2.]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.226415	1	2.038462
Marcador Contr[0. 1.]', 'HSR Rel Count[3.77 4.54]'	Resultado[1. 1.]	0.207547	0.733333	1.494872
Hz*G MAX[4.04 4.29]'	Resultado[1. 1.]	0.207547	0.6875	1.401442
HSR Abs (m/min)[3.75 4.55]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.188679	0.625	1.274038
HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]', 'Vel Rel [45 - 65] %[9.11 10.09]'	Resultado[1. 1.]	0.188679	0.909091	1.853147
Impacts [0-3] G[12489.85 14153.08]'	Resultado[1. 1.]	0.188679	0.526316	1.072874
Hz*G MAX[4.04 4.29]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.188679	0.714286	1.456044
Marcador Casa[3. 5.]'	Resultado[1. 1.]	0.188679	0.769231	1.568047
Vel Rel [45 - 65] %[8.05 8.81]'	Resultado[1. 1.]	0.188679	0.5	1.019231
Impacts [0-3] G[10093.23 12253.58]', 'HSR Abs (m/min)[3.75 4.55]'	Resultado[1. 1.]	0.169811	0.529412	1.079186
Impacts [0-3] G[10093.23 12253.58]', 'HSR Abs (m/min)[3.75 4.55]', 'Marcador Contr[0. 1.]'	Resultado[1. 1.]	0.169811	0.818182	1.667832

Cuadro D.4: Tabla Reglas FP-Growth después de Caracterización que indican si el equipo Gana

Reglas que indican si el equipo Empata				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
MAX Dec(m/sÂ²) [-6.38 -6.06]	Resultado[0. 0.]	0.113208	0.428571	1.514286
Marcador Contr[2. 3.]	Resultado[0. 0.]	0.132075	0.411765	1.454902
Impacts [0-3] G[10093.23 12253.58]	Resultado[0. 0.]	0.169811	0.28125	0.99375
Marcador Casa[0. 1.]	Resultado[0. 0.]	0.150943	0.363636	1.284848
Marcador Contr[0. 1.]	Resultado[0. 0.]	0.150943	0.235294	0.831373
HSR Abs (m/min)[2.7 3.64]	Resultado[0. 0.]	0.150943	0.380952	1.346032
HSR Rel Count[2.08 3.69]	Resultado[0. 0.]	0.113208	0.4	1.413333
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]	Resultado[0. 0.]	0.113208	0.545455	1.927273
Marcador Casa[0. 1.]', 'Marcador Contr[0. 1.]	Resultado[0. 0.]	0.150943	0.571429	2.019048
HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]	Resultado[0. 0.]	0.113208	0.428571	1.514286
Marcador Casa[0. 1.]', 'HSR Abs (m/min)[2.7 3.64]', 'Marcador Contr[0. 1.]	Resultado[0. 0.]	0.113208	0.666667	2.355556

Cuadro D.5: Tabla Reglas FP-Growth después de Caracterización que indican si el equipo Empata

Reglas que indican si el equipo Pierde				
Antecedents	**Consequents**	**Support**	**Confidence**	**Lift**
Marcador Casa[0. 1.]	Resultado[-1. -1.]	0.188679	0.454545	2.007576
Marcador Contr[2. 3.]	Resultado[-1. -1.]	0.150943	0.470588	2.078431
HSR Abs (m/min)[3.75 4.55]	Resultado[-1. -1.]	0.132075	0.269231	1.189103
Vel Rel [45 - 65] %[8.05 8.81]	Resultado[-1. -1.]	0.113208	0.3	1.325
Impacts [0-3] G[10093.23 12253.58]	Resultado[-1. -1.]	0.132075	0.21875	0.966146
HSR Rel Count[2.08 3.69]	Resultado[-1. -1.]	0.113208	0.4	1.766667
Marcador Casa[0. 1.]', 'Marcador Contr[2. 3.]	Resultado[-1. -1.]	0.132075	1	4.416667
HSR Abs (m/min)[3.75 4.55]', 'Marcador Casa[0. 1.]	Resultado[-1. -1.]	0.132075	0.7	3.091667
Impacts [0-3] G[10093.23 12253.58]', 'Marcador Casa[0. 1.]	Resultado[-1. -1.]	0.113208	0.545455	2.409091

Cuadro D.6: Tabla Reglas FP-Growth después de Caracterización que indican si el equipo Pierde