UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

ANÁLISIS ESTADÍSTICO EN EL ÁMBITO DEL FÚTBOL DESDE LA PERSPECTIVA DE BIG DATA

Autor: Diego Reyes Santana

Tutores: Luis Miguel Hernández Acosta

Titulación: Ingeniero de Telecomunicación

Fecha: Junio 2018

UNIVERSIDAD DE LAS PALMAS
DE GRAN CANARIA

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

ANÁLISIS ESTADÍSTICO EN EL ÁMBITO DEL FÚTBOL DESDE LA PERSPECTIVA DE BIG DATA

HOJA DE FIRMAS

Alumno:

Fdo.: Diego Reyes Santana

Tutor:

Fdo.: Luis Miguel Hernández Acosta

Titulación: Ingeniero de Telecomunicación

Fecha: Junio 2018

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA

PROYECTO FIN DE CARRERA

ANÁLISIS ESTADÍSTICO EN EL ÁMBITO DEL FÚTBOL DESDE LA PERSPECTIVA DE BIG DATA

CALIFICACIÓN:

Presidente:

Fdo.:

Vocal: **Secretario:**

Fdo.: Fdo.:

Titulación: Ingeniero de Telecomunicación

Fecha: ……

AGRADECIMIENTOS

Estos años hemos recorrido un largo camino que ha llegado a su fin, y digo hemos porque han sido muchas personas las que me han acompañado en este viaje, con los que he compartido y festejado los buenos momentos y los que me han tendido una mano tras cada tropiezo.

Es por ello que quiero dedicar y agradecer este proyecto a todos ellos:

- A mis compañeros de clase, grupo de luchadores que poco a poco nos curtimos en la batalla.
- A mi tutor Luis Hernández por la confianza depositada y el buen *feedback* alcanzado.
- A mi familia por su apoyo incondicional.
- Y a ti, siempre a mi lado.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Acrónimos

KDD – *Knwoledge Discovery in Databases.*

IoT – *Internet Of Things.*

M2M – *Machine To Machine.*

GFS – *Google File System*.

HDFS – *Hadoop Distributed File System.*

GGR – *Gross Gaming Revenue.*

YARN – *Yet Another Resourse Negociator.*

SGD – Descenso por gradiente estocástico.

OR – *Odd Ratio*

API – *Application Programming Interface*

RDD – *Resilient Distributed Dataset*

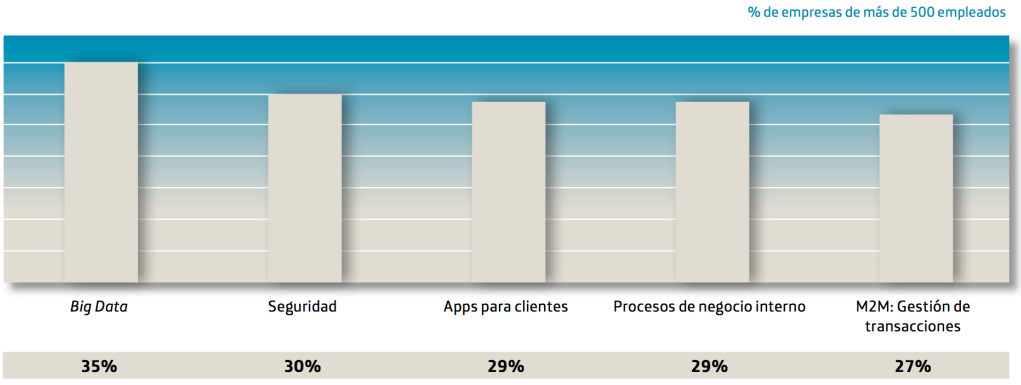
HTLM – *HyperText Markup Language*

PHP – *HyperText Preprocessor*

CAPÍTULO 1. INTRODUCCIÓN

1.1 Introducción

La toma de decisiones en cualquier ámbito dictará el camino del éxito o el fracaso, por lo tanto, en la actualidad, la tendencia es conseguir un valor añadido a las decisiones sustentadas en el análisis de datos. Esto unido al gran flujo de datos que se genera y almacena en cada instante, hace emerger soluciones para poder analizar grandes cantidades de datos para obtener conclusiones en períodos de tiempo cada vez más cortos, las cuales se engloban bajo el nombre de *"Big Data"*. Éste es el principal motivo por el que el análisis de datos está generando una gran expectación y ningún sector se quiere quedar al margen. Esto lleva a las tecnologías de la información y las comunicaciones a potenciar este sector, haciendo grandes inversiones y esfuerzos para ser pioneros. En la Figura 1 se ilustra la tendencia de la inversión que las empresas están realizando en los diferentes sectores, siendo Big Data la primera opción [1].



Fuente: Telefónica. Datos de julio de 2016.

Figura 1. Inversiones en soluciones digitales 2016

La rama de la minería de datos que tiene relación con la predicción de probabilidades y tendencias futuras se denomina análisis predictivo y permite

extraer conclusiones relativamente confiables sobre eventos futuros, a través de la aplicación de métodos estadísticos, matemáticos y de reconocimiento de patrones.

Existe una gran variedad de técnicas de modelización, aprendizaje automático y minería de datos para, analizando los datos actuales e históricos hacer predicciones futuras o prever acontecimientos no conocidos.

La tarea fundamental del análisis predictivo es relacionar las distintas variables de los datos históricos, con la intención de escalarlo a lo que puede ocurrir en el futuro. El nivel de fiabilidad de los resultados muchas veces está acotado o influido fuertemente por la calidad de los análisis realizados y las hipótesis adoptadas, lo que convierte en un factor importante la visión general del investigador y la interpretación de los resultados [2] [3].

Dentro del análisis predictivo, según el objetivo que se persiga, se suelen distinguir dos grandes grupos: modelos predictivos y modelos descriptivos.

El objetivo del modelo predictivo es la representación o descripción de una de las variables en relación con el resto, son conocidos como métodos asimétricos, supervisados o directos. Se llevan a cabo mediante la búsqueda de normas de clasificación o de predicción, ayudando a predecir o clasificar los resultados futuros.

Los modelos descriptivos o de aprendizaje no supervisados, permiten formar grupos de datos rápidamente, también son conocidos como métodos simétricos no supervisados o indirectos. Las observaciones son clasificadas en grupos que no son conocidos con anterioridad ya que las variables pueden estar conectadas entre sí de acuerdo a vínculos desconocidos.

Muchos expertos coinciden en que la forma de extraer conocimientos de los datos solo se puede conseguir mediante técnicas modeladas mediante procesos. Las técnicas de minería de datos antes mencionadas forman la parte central del procesado, pero no menos importante serán otros pasos preliminares así como la verificación. Al conjunto de estas tareas se le denomina Knowledge Discovery in Databases (KDD) [4].

1.1.1 Knwoledge Discovery in Databases (KDD)

La extracción de conocimiento está principalmente relacionada con el proceso de descubrimiento, conocido como KDD. Se refiere al proceso no trivial de descubrir información potencialmente útil dentro de los datos contenidos en algún repositorio de información. No es un proceso automático, es un proceso iterativo que exhaustivamente explora grandes volúmenes de datos para determinar relaciones. Es un proceso que extrae información de calidad que puede usarse, a su vez, para extraer conclusiones basadas en relaciones o modelos dentro de los datos.

En la Figura 2 se muestra un diagrama del proceso KDD que se divide en 5 etapas que se detallan a continuación:

1.Selección de datos. En esta etapa se escoge la fuente de datos y se fija qué tipo de información vamos a utilizar.

2.Preprocesado. Consiste en la preparación y limpieza de los datos extraídos de las fuentes para que sean manejables en fases posteriores. En esta etapa se llevarán a cabo diferentes estrategias para manejar la falta de datos, obteniéndose al final una estructura de datos de una forma concreta para la fase siguiente.

3.Transformación. Consiste en el tratamiento preliminar de los datos, generación de nuevas variables con una estructura adecuada. Además, se

debe tener en cuenta en esta etapa la normalización o cualquier requerimiento de la siguiente fase.

4.Minería de datos. Es la fase de modelado, donde métodos inteligentes son aplicados con el objetivo de extraer patrones previamente desconocidos, válidos, potencialmente útiles, comprensibles y que están contenidos en los datos.

5.Interpretación y evaluación. Se identifican y analizan los patrones encontrados que son realmente interesantes, basándose en alguna medida y también, una evaluación de los resultados obtenidos.

Se suele incluir una etapa preliminar de análisis de la tarea a desarrollar, donde se define el problema y se plantean los requerimientos, así como otra etapa final de interpretación de los resultados.

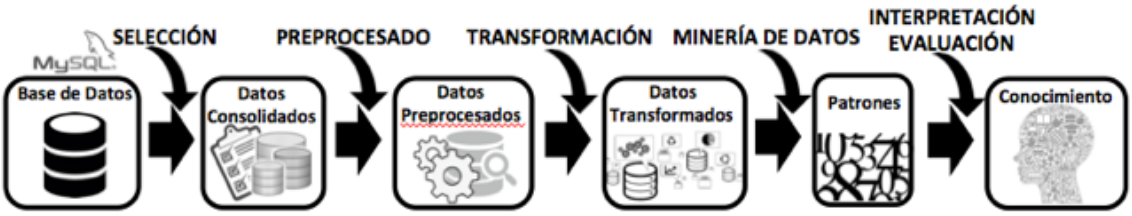


Figura 2.Diagrama proceso KDD

1.1.2 Métodos predictivos

A partir de ahora, nos centraremos solo en los métodos predictivos que son lo que realmente interesan en este el proyecto, haciendo solo una descripción superficial de los principales algoritmos pero, eso sí, poniendo en relieve sus ventajas y desventajas. Al final, esto nos dará una idea bastante concreta del problema que tenemos entre manos, lo que nos permitirá discernir la conveniencia de la elección que hagamos del algoritmo de aprendizaje a aplicar para alcanzar nuestros objetivos.

1.1.2.1 Redes neuronales.

Las redes neuronales tienen gran aceptación por parte de los investigadores por sus buenos resultados, pero requieren de más trabajo de exploración que otras técnicas. La combinación de los atributos de una observación es la clave de este método. El proceso de modelización consiste en entrenar la red neuronal para que aprenda a combinar los atributos con la estructura y pesos adecuados [5]. Las principales características son:

1. Sigue un proceso heurístico de entrenamiento que le permite ir ajustando los pesos para los atributos de entrada.
2. Las entradas deben normalizarse en rangos de 0 a 1 para facilitar la convergencia del algoritmo.
3. Entre más capas intermedias, más se ajustará el resultando pero también, mayor será el riesgo de sobreajuste *(overfitting)*. Para un correcto funcionémiento es importante tener una muestra de datos para entrenamiento y otra para test y que éstas no se solapen.
4. Se comporta como una "caja negra", dificultando su estudio e interpretación y suele ser éste, un argumento de peso para descartar su uso en muchos casos.

1.1.2.2 Regresiones logísticas.

La regresión logística es una técnica de modelización paramétrica, de las más usadas por su eficiencia y alta capacidad predictiva. La relación entre la variable explicativa y la transformada de la variable dependiente es lineal. Sus principales características son:

1. No hay limitaciones en cuanto a las variables independientes o explicativas, pueden ser categóricas o no categóricas.
2. Definida la variable dependiente (perteneciendo a una categoría u otra) el modelo de regresión logística la expresa, entonces, en términos de probabilidad.

3. Las regresiones logísticas requieren menos esfuerzo que las redes neuronales ya que no es necesario ni explorar diferentes estructuras ni comparar los diferentes sobreajustes.
4. En caso de tener que hacer uso de múltiples modelos resultan la mejor opción.

Existe cierta controversia en relación a qué modelos son más eficientes en la solución de problemas de predicción/clasificación de patrones. Por ello, muchos autores están estudiando las relaciones entre las técnicas estadísticas convencionales y los modelos conexionistas (ver Cherkassky, Friedman y Wechler, 1994; Flexer, 1995; Michie, Spiegelhalter y Taylor, 1994; Ripley, 1996; Sarle, 1994, 1998). De hecho las redes neuronales han sido conceptualizadas como técnicas estadísticas no paramétricas al estar libres del cumplimiento de los supuestos teóricos de la estadística paramétrica o como técnicas de regresión no lineal. El problema surge cuando encontramos resultados contradictorios a la hora de determinar qué modelos son más eficientes. Así, mientras algunos trabajos empíricos no encuentran diferencias entre los resultados de unos y otros (Croall y Mason, 1992; Michie et al, 1994; Ripley, 1993; Thrun, Mitchell y Cheng, 1991), otros resultados tienden a apoyar la idea de que existe una ligera superioridad de las redes neurales sobre las técnicas estadísticas (ver Garson, 1991; Huang y Lippman, 1987; White, 1994).

En concreto sobre esta discusión se ha encontrado un trabajo en el que se concluye que las redes neuronales tienen mejores resultados que las regresiones logísticas en técnicas de clasificación, no concluyendo en el pronunciamiento de desechar los métodos estadísticos convencionales a la hora de realizar clasificación, ya que las redes neuronales, pese a su mejor rendimiento, presentan una serie de inconvenientes que el investigador debe sopesar antes de decidirse. En primer lugar, el entrenamiento de redes neuronales es un proceso demasiado creativo que, normalmente, se solucionan con ensayo y error. Además, la calidad de las soluciones dadas por la red neuronal elegida no puede ser garantizada debido a su naturaleza de

caja negra. No hay que olvidar tampoco que una red neural no da información explícita sobre la importancia relativa de los distintos predictores. Por último, no hay que obviar el elevado coste computacional requerido en el entrenamiento de las redes neurales, muy superior al de los modelos estadísticos. En última instancia, deberá ser el investigador quien, sopesando tales limitaciones, decida si compensa la utilización de una arquitectura de uno u otro tipo [6].

1.1.3 Big Data

En la época actual, la generación de datos se estima en 2,5 quintillones por día. Estos datos provienen de diferentes fuentes. Por ejemplo, las industrias almacenan datos transaccionales, proveedores, clientes, etc. Las administraciones públicas también guardan grandes bases de datos, historiales clínicos, genoma humano, censo, impuestos, etc. Sin embargo, una de las fuentes más importantes son las generadas por cada individuo en su día a día, búsquedas en Internet, coordenadas GPS, interacciones con redes sociales, etc, las cuales son almacenadas por grandes servidores (Google y Amazon como principales empresas en el almacenamiento masivo). Se estima que se generan en las redes sociales, 12 Terabytes diarios en Tweets y 100 Petabytes en Facebook en forma de vídeos y fotos, por ejemplificar la magnitud de estas cifras.

Un estudio realizado por Cisco [7], entre 2011 y 2016 estima que la cantidad de tráfico por móviles crecerá a una tasa anual del 78%, así como el número de dispositivos móviles conectados a Internet superará al de habitantes mundiales. Este volumen de tráfico previsto a partir de 2016 podría llegar a 130 Exabytes anuales.

Si a todo esto añadimos las expectativas generadas por el Internet de las Cosas (IoT), donde el incremento de sensores digitales se espera que sea del 30% anual, contando en la actualidad aproximadamente con 30 millones de

sensores interconectados, esto creará una cantidad de datos M2M (machine-to-machine) difícilmente calculable.

Ante esta situación de disponer de un torrente de datos masivos generados en tiempo real, emerge Big Data como solución a la hora de poder analizar, describir y dar valor a ese sinfín de información, la cual no puede ser computada a través de las herramientas existentes, ya sea por la necesidad en la velocidad de obtener valor, por la imposibilidad de manejar este gran flujo de datos o disponer de bases de datos no estructuradas [8]. Con todo esto, lo que se pretende es obtener conclusiones y tomar decisiones, acerca de la nueva forma de actuar en este tipo de situaciones, cambiando así el escenario actual en muchos sectores que están apostando fuertemente por esta nueva tecnología.

Big Data, por lo tanto, no hace referencia solamente a la cantidad de datos sino también, a la velocidad y variedad en las fuentes que generan dichos datos [9]

1.1.3.1 Hadoop y su arquitectura

Apache Hadoop es un *framework* o *toolkit* software que soporta aplicaciones distribuidas bajo una licencia libre, y aparece como solución a uno de los grandes problemas de Big Data, que es analizar e interpretar datos de naturalezas dispares entre sí, permitiendo el procesamiento de grandes volúmenes de datos a través de clusters, usando un modelo de programación simple.

Hadoop está inspirado en el proyecto *Google File System* (GFS) y en el paradigma de programación MapReduce, el cual consiste en dividir en dos tareas (mapper-reducer) la manipulación de los datos distribuidos entre los nodos de un cluster, logrando como resultado un alto paralelismo en el

procesamiento [10]. Hadoop tiene una arquitectura maestro-esclavo, y su sistema de almacenamiento es *Hadoop Distributed File System* (HDFS) que permite desarrollar los cálculos usando con algoritmos del tipo MapReduce [11].

- ***Hadoop Common***

  Proporciona el acceso a los sistemas de archivos soportados por Hadoop y contiene el código necesario para poder ejecutar el *framework*.

- ***Hadoop Distributed File System (HDFS)***

  Los datos en el cluster de Hadoop son divididos en bloques distribuidos, así las funciones map y reduce pueden ser ejecutadas en pequeños subconjuntos y esto, provee de la escalabilidad necesaria para el procesamiento de grandes volúmenes de datos.

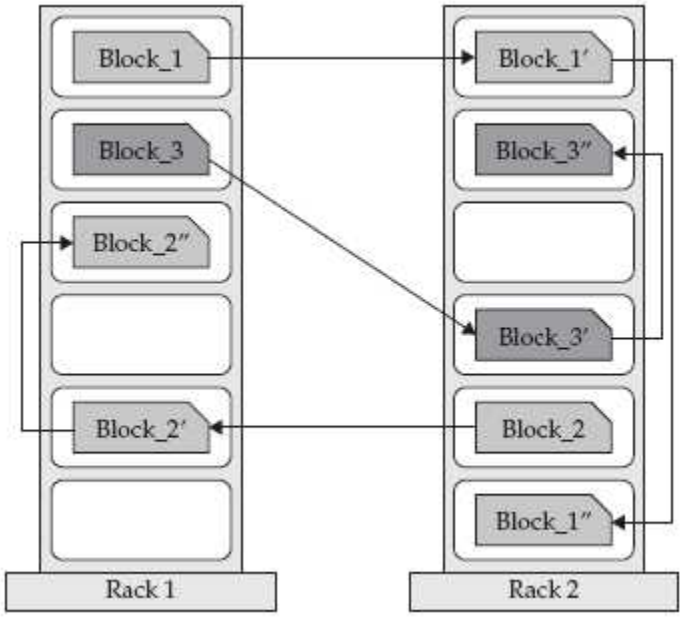


Figura 3.Sistema de almacenamiento HDFS

- ***Hadoop YARN (Yet Another Resourse Negociator)***

  YARN es una tecnología de gestión de clusters y se compone de un gestor de recursos central y otro que se ocupa de controlar un único nodo.

- ***Hadoop MapReduce***

  Fue desarrollado para hacer frente a problemas de manera distribuida, problemas que comparten ciertas similitudes pero requieren desarrollos completos desde el inicio. Se compone de dos fases: fase de mapeado (*map*) y fase de reducción (*reduce*). La fase de mapeado trabaja con datos sin procesar y produce valores intermedios que se pasan a la fase de reducción donde se recombinan y reordenan para producir la salida final.

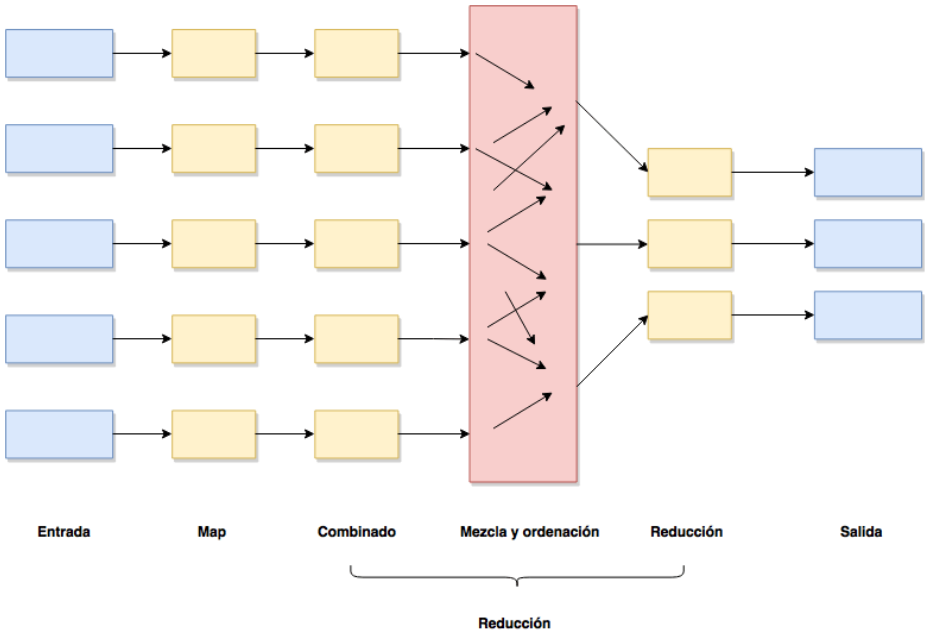


Figura 4 Diagrama MapReduce

1.1.4 Apuestas deportivas

El mundo del fútbol es uno de los negocios que no se resiente con el paso del tiempo, es más, cada año se analiza cualquier detalle que pueda mejorarse con la intención de hacer de éste un espectáculo aún mayor. Uno de los atractivos que ha surgido en esta última década es el de las apuestas deportivas. En España este tipo de juegos de azar no es nuevo, ya que La Quiniela data su existencia desde 1946, y desde entonces se ha profesionalizado; tanto es así, que se han ido creando peñas que obtienen asiduamente los premios de primera categoría y que no dejan acumular premios considerables de forma cotidiana. Estos botes acumulados es el principal atractivo de los juegos existentes en Loterías y Apuestas del Estado, perdiendo de esta forma popularidad con respecto a nuevas formas de juego, en el que el usuario es el que elige a qué eventos quiere jugar de los ofrecidos por las casas de apuestas.

Las apuestas son uno de los entretenimientos más antiguos de la historia. En lo que se refiere a apuestas en eventos deportivos, hay que remontarse a la antigua Grecia, donde se celebraban competiciones atléticas y competían por premios, sin embargo eran de gran expectación gracias a las fortunas que se jugaban en forma de apuestas desde las gradas. Fue en el siglo XVIII y XIX cuando se empezó a extender a gran escala. El Reino Unido fue la verdadera cuna con las carreras de caballos y galgos, luego con la aparición de casas físicas de apuestas comenzó la expansión a otros deportes.

Sin lugar a dudas la gran revolución del mercado llegaría gracias a Internet. El juego online proporciona la comodidad al usuario de participar en cualquier momento, incluso con apuestas en directo durante el transcurso de cualquier acontecimiento deportivo sin desplazarse a establecimientos especializados, un cóctel que ha conquistado a miles de usuarios en España. La creación de nuevas compañías en el sector y la lucha por captar nuevos usuarios, es el escenario actual del sector.

En 2012, entra en vigor la ley de ordenación de juego en España [12], como consecuencia de la irrupción de esta avalancha de nuevas empresas de juego online, las cuales ejercían sin licencias y libres de tributos en este ámbito.

Para hacernos una idea de la magnitud económica del sector, podemos observar en la Figura 5 y Figura 6 la evolución del mercado publicado en el informe del cuarto trimestre de 2016 [13]. Siendo GGR (*Gross Gaming Revenue*), el importe total de las cantidades dedicadas a la participación en el juego, netas de bonos y deducidos los premios satisfechos por el operador a los participantes, es decir, los beneficios obtenidos por las casas de apuestas.

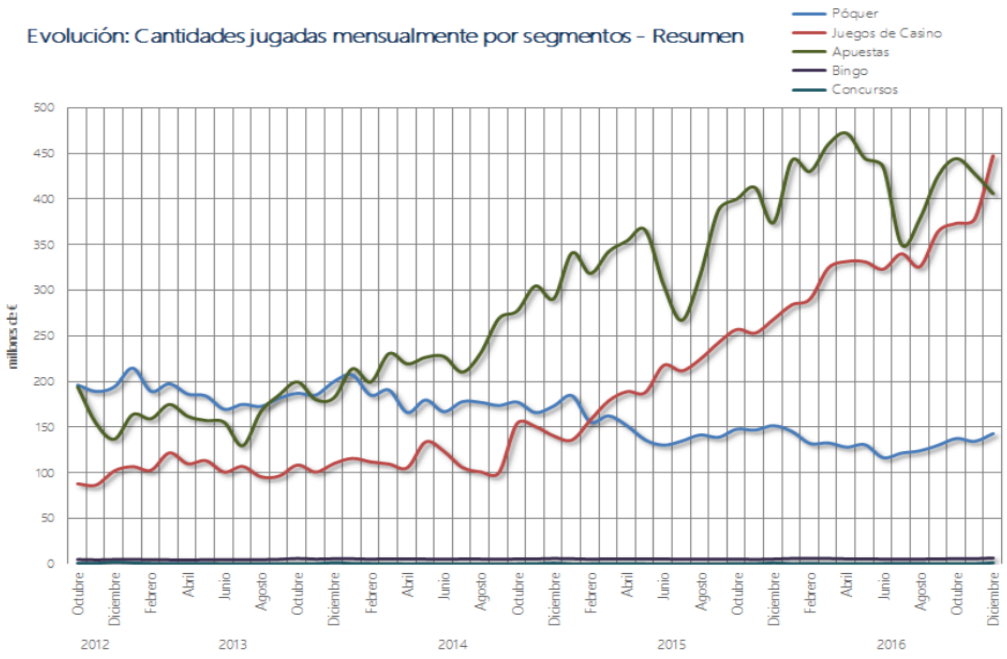


Figura 5.Cantidades jugadas por categorías

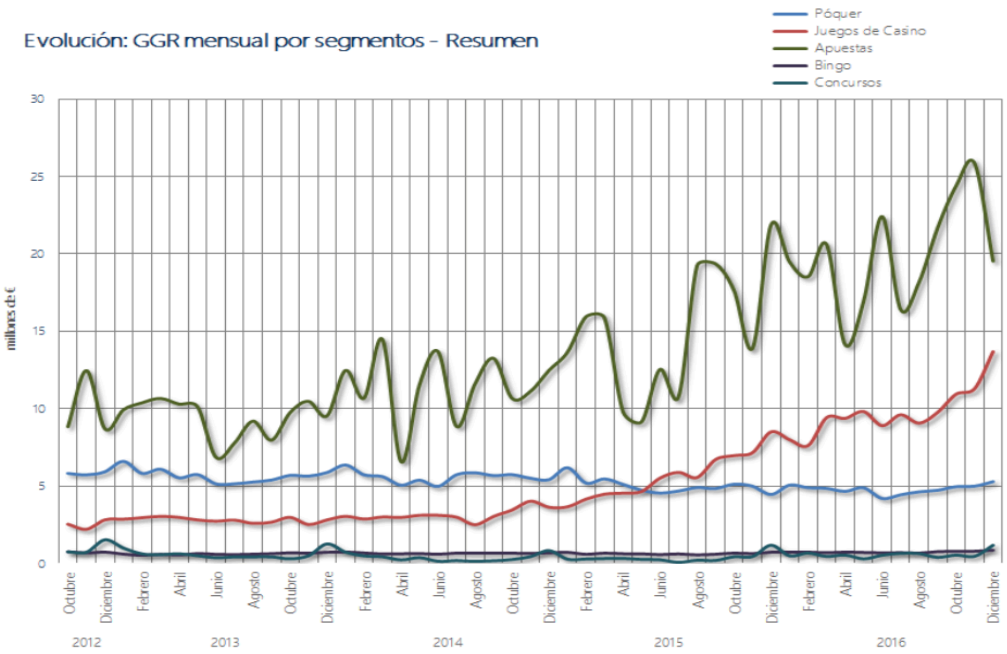


Figura 6.GGR, beneficios por categorías

1.1.4.1 Las cuotas deportivas

Comprender las cuotas deportivas y asimilar cómo las establecen las casas de apuestas será uno de los requisitos indispensables para poder desarrollar y comprender este proyecto.

Referirse a cuotas o probabilidades es lo mismo, ya que matemáticamente, una es proporcional a la otra. Éstas son usadas por las casas de apuestas, para facilitar el cálculo de las posibles ganancias en caso de acierto, siendo además un método muy práctico para los usuarios.

Las casas de apuestas asignarán, entonces, una probabilidad a cada suceso del encuentro. El método usado para hacerlo es desconocido, no hay que olvidar que el sector está profesionalizado desde hace años y, por tanto estarán un paso por delante, tanto en pronósticos como en las metodologías en que basan sus cuotas. Esto no significa que no tengamos margen para

aprovechar y sacar cierto beneficio de ello. Un ejemplo de las cuotas fijadas en un partido serían:

Victoria del equipo local	Empate	Victoria del equipo visitante
4/6	9/4	3/1
1,67	3,2	4

Tabla 1. Ejemplo de las cuotas fijadas en un partido

La primera fila expresa la cuota en sistema fraccional y la segunda en sistema decimal, que es el más usado en España. La interpretación del sistema fraccional en el ejemplo expuesto sería que en caso de acierto si apostamos a la victoria del equipo local (4/6), por cada seis unidades apostadas, se ganaran cuatro en caso de acierto. El sistema decimal nos indica que el retorno total, que incluye la apuesta inicial, será 1,67 por cada unidad apostada en caso de acierto.

El sistema fraccional también se le denomina odds, es un término muy usado en estadística ya que representa el cociente de ventaja de probabilidad; p/(1-p).

Además, si calculamos la inversa de las cuotas decimales obtenemos la distribución de probabilidad para ese partido:

Victoria del equipo local	Empate	Victoria visitante	Total
60%	31%	25%	116%

Tabla 2 Distribución de probabilidad

No puede haber una probabilidad superior al 100% de que algo suceda, por tanto, este superávit en el valor de la probabilidad, es el margen de beneficio de la casa de apuestas. Acortando de manera artificial las cuotas

para el resultado de un partido en concreto, o, dicho de otro modo, aumentando la probabilidad estimada para dicho resultado. Como consecuencia, las cuotas que la casa de apuestas ofrece para dicho partido, no son cuotas proporcionales a la probabilidad. Por ejemplo, no se puede apostar a los tres resultados de un partido en una misma casa de apuestas y no perder dinero. A la diferencia de precio entre las cuotas exactas y las de la casa de apuestas se le llama *over-round*. Es precisamente de este *over-round* de donde la casa de apuestas obtiene beneficios. En este ejemplo en concreto, el *over-round* es de 1,16 (o del 16%). Por lo general, los *over-rounds* de las casas de apuestas online suelen oscilar entre el 6% y 20% [14].

La casa de apuestas suele comenzar calculando la probabilidad real, para, posteriormente, cambiar las cuotas de acuerdo con sus márgenes de beneficio (*over-round*), siempre teniendo presentes los parámetros de apuestas de los jugadores online, aplican este margen de beneficio hacia uno de los resultados en los que esperan tener más apuestas. Las casas de apuestas no aplican ningún tipo de comisión cuando se hace una apuesta, por tanto, este margen es el único factor para obtener beneficios.

Sin embargo, los usuarios pueden optar masivamente por uno de los resultados en concreto, y en este caso, los operadores de apuestas con intención de garantizar beneficios deben equilibrar la distribución de las apuestas. El mecanismo usado por las casas es cambiar la distribución de probabilidad para hacer más atractivas las otras opciones.

A esta variación se le denomina desplazamiento de la línea de apuestas y quien lo provoca es la demanda de los apostantes por una opción. En este punto es cuando las apuestas actúan igual que el mercado de valores, cambiando los valores de las cuotas dependiendo de la demanda. Por esta razón, existen muchos jugadores que se dedican a aplicar las técnicas de *trading*, usadas en bolsa, en los mercados de intercambio de apuestas entre usuarios [15].

Este fenómeno es el principal factor por el que es posible encontrar partidos en los que la probabilidad real y la probabilidad ofrecida por el operador estén descompensadas, pudiendo entonces realizar una “apuesta de valor”, ya que la casa de apuestas ofrecería una cuota superior a la que debiera, superando en este caso la recompensa al riesgo.

En el ejemplo anterior, si la probabilidad real de que el equipo local fuera superior al 60% y se mantuviera la cuota ofrecida por el operador, tendríamos una posible “apuesta de valor”, lo que garantizaría una ganancia a largo plazo.

Para mejor comprensión y aplicar los conceptos expuestos, a continuación se evalúa un supuesto simple. Suponga que una casa de apuestas ofrece como evento el lanzamiento de una moneda equilibrada, la probabilidad real de que salga cara será del 50% y que salga cruz el 50% restante. Las cuotas en sistema decimal ofrecidas por la casa de apuestas son 1,9 en caso de que salga cara y 1,9 en caso de que salga cruz.

Haciendo la inversa de la cuota, obtendremos la distribución de probabilidad ofrecida que, en este ejemplo concreto, es del 52,63% en ambos casos.

En un principio las apuestas se colocan por partes iguales a cara y cruz, por lo que la casa de juegos ganaría veinte céntimos por cada par de euros que se apuesten de forma equilibrada. Si las apuestas se desequilibran y se deposita más capital a que sale una de las dos opciones (por ejemplo cara), entonces, la casa de apuestas estaría poniendo en riesgo el beneficio si sale cara. En este caso, difícilmente se equilibraría la línea de apuestas si mantienen las cuotas, entonces lo que hace la casa de juegos es cambiar la distribución de probabilidad y, por tanto, sus cuotas, quedando las nuevas cuotas como 1,75 cara y 1,95 cruz. Si el supuesto no fuera tan obvio, como es

el caso de un partido de fútbol, y se siguiera apostando a que sale la primera opción, la casa de apuestas tendría que volver a cambiar las cuotas dejando, por ejemplo, una cuota de 1,7 cara y 2,1 cruz. Si se calcula la inversa de estas cuotas se tendría en caso de cara una probabilidad de 58,82% y en caso de cruz de un 47,61%. En este momento el apostante que haga una apuesta a que salga cruz estaría haciendo una “apuesta de valor” ya que la probabilidad real del 50% es superior a la ofrecida por la casa de apuestas. Una vez se realicen suficientes apuestas para equilibrar el mercado, la línea de apuestas volverá a su origen.

La verdadera dificultad reside en el pronóstico de la probabilidad real, ya que, de la cuota, se ocupa la casa de apuestas, llegando a ser una tarea ardua, no solo calcular las probabilidades reales sino encontrar casas de apuestas que ofrezcan cuotas que se encuentren por encima de éstas, ya que incluyen el margen de beneficio. Solo haciendo un buen cálculo y unas buenas hipótesis se podría encontrar posibles “apuestas de valor”, sin embargo es bastante complicado ser certero en la cuantificación de las cuotas.

Existen muchos apostantes experimentados que, a través de la experiencia y del análisis de los datos de cada evento deportivo, asumen el riesgo de apostar con la hipótesis de que la probabilidad estimada no se ajusta a la realidad.

Esta hipótesis se basa en principios, como que la casa de apuestas no haya tenido en cuenta alguna circunstancia especial de algún equipo, (por ejemplo, que la situación en la tabla clasificatoria sea transitoria, o que se tenga en cuenta las tácticas deportivas de los equipos que se enfrentan), que conlleve un efecto en el partido distinto a lo esperado por los datos estadísticos.

Aunque las hipótesis sean ciertas, si no se hace un análisis estadístico, es difícil saber como afectaría a la probabilidad, y es aquí donde nace la idea de este Proyecto Final de Carrera de crear una herramienta *software* basada en un sistema de predicción que permita variar, o no tener en cuenta, el valor de alguna de las principales variables del modelo y, así, poder cuantificar en términos de probabilidad o cuotas, las hipótesis del apostante.

1.2 Estado del arte.

En este apartado nos centraremos, solo en un Proyecto Fin de Carrera de Ingeniería Informática de la Universidad Carlos III titulado; “Sistema de predicción de resultados deportivos y su aplicación en las apuestas” [16], cuyo objetivo es la creación de un sistema de predicción capaz de deducir el resultado más probable de un evento deportivo y medir su riesgo.

El proyecto se puede dividir en dos partes bien diferenciadas, el sistema de predicción y la generación de estrategias para la explotación del sistema.

El sistema de predicción desarrollado en el proyecto se ha basado en algoritmos de red bayesianas y árbol J48 que, tomando una pequeña muestra de partidos de cada competición y recogiendo una serie de variables como predictores, el sistema se entrena generando modelos de predicción para cada competición analizada; más concretamente lo que se obtiene son modelos clasificadores.

En cuanto a las estrategias desarrolladas, lo que propone el proyecto analizado es cuantificar, a través de la tasa de clasificaciones correctas, el porcentaje de acierto del modelo en una pequeña muestra para test. Con este porcentaje se calcula una cuota mínima a la que se debería apostar al resultado más probable arrojado por el sistema, para cualquier partido

analizado, es decir, si se obtuviera una tasa de clasificaciones correctas del 90%, se entiende, que apostando solo a cuotas superiores a 1/0,9 se obtendría beneficio a largo plazo, sin tener en cuenta cada partido en particular.

Además, para el cálculo de esta cuota, ha tomado la tasa de clasificaciones correctas, la cual incluye partidos a los que no apostaríamos según su criterio, al tener una cuota inferior a la propuesta, adulterando de este modo la cuota mínima. Esto es debido a que, si no se incluyen los partidos con probabilidades más altas para uno de los sucesos, la tasa de clasificaciones correctas descendería, aumentando el margen establecido. Se podrían exponer varios supuestos explicativos en el que el sistema fallaría en este sentido, pero no es objeto del proyecto.

A priori, puede parecer muy semejante a lo que se pretende desarrollar en nuestro proyecto (y es por ello que se ha elegido) pero, realmente, si se analiza más detalladamente se verá que no es así, en cuanto que el sistema de predicción de nuestro proyecto no se usará como un clasificador, aunque podría usarse como tal. Lo que se busca es un sistema de predicción para cuantificar la probabilidad de un partido y no que arroje el más probable. El motivo de los matices expuestos, es que las cuotas de cada partido deben ser analizadas individualmente, para poder tener una “apuesta de valor”, y no una cuota prefijada como margen indicativo de si la apuesta es segura.

Por otro lado, en cuanto a compañías que se dedican al análisis de datos deportivos es de interés nombrar a Opta [14], compañía que trabaja con una gran cantidad de medios de comunicación, casas de juego y clubs deportivos entre otros. Esta compañía ofrece datos analizados, herramientas para la retrasmisión o *widgets* para apuestas deportivas. Ofrecen modelos predictivos entre sus principales productos pero, al ser una compañía de pago, no se ha podido analizar su funcionamiento más detalladamente.

1.3 Objetivos

El objetivo principal de este Proyecto Fin de Carrera, es desarrollar e implementar una herramienta que ayude, en la toma de decisiones del usuario, a la hora de apostar a un partido de fútbol de Primera División Española, pudiendo discernir (a través de un modelo de predicción) si lo que ofrece la casa de apuestas es realmente justo o se ajusta a la probabilidad que el sistema propuesto dictamine.

Para desarrollar esta herramienta se usarán, inicialmente, técnicas de minería de datos, analizando datos históricos de partidos de la Liga de Fútbol Profesional, para luego, procesar estos datos y obtener variables que sean determinantes para el estudio estadístico y así, finalmente, crear un modelo de predicción a la medida del usuario que arroje una predicción de probabilidad para el evento seleccionado.

Para una mejor interacción con el usuario, se desarrrolará una interfaz de usuario como modo de presentar los resultados, pudiendo seleccionar tanto los diferentes eventos disponibles, como las variables a pasar al sistema de predicción.

Una vez desarrollada la herramienta, se estudiará la forma de adaptarla a un cluster implementado con Hadoop, que permita un cálculo paralelizado que haga esta herramienta escalable en el sentido que un aumento de la muestra y variables no afecte al coste computacional final.

1.4 Estructura del sistema

El objetivo de este apartado es detallar la arquitectura que se ha propuesto para la herramienta, así como exponer motivadamente las decisiones tomadas en la elección de las herramientas y técnicas usadas. También se describe la funcionalidad global y de cada uno de los componentes.

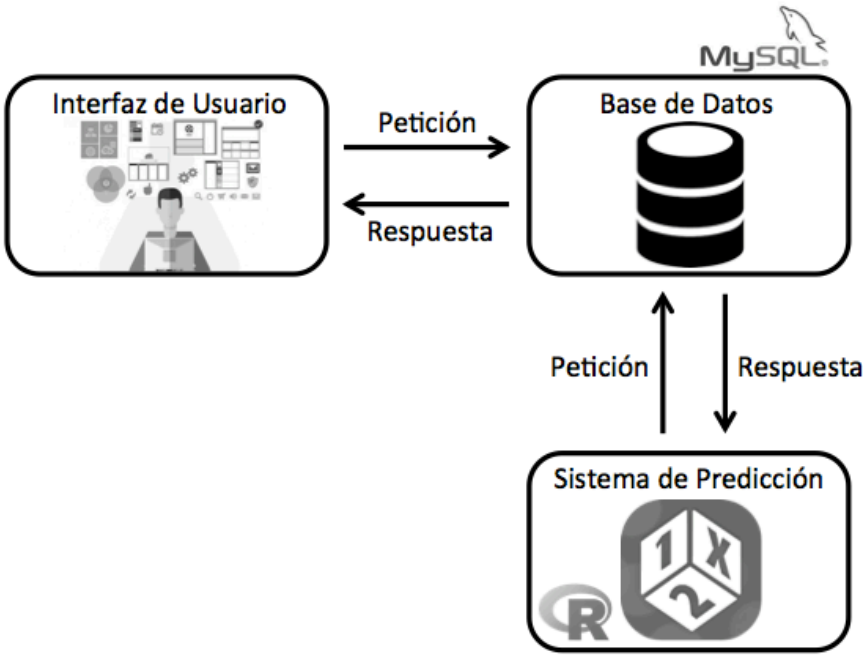


Figura 7. Esquema sistema global

La herramienta se compone de tres módulos principales como se puede observar en la Figura 7: la interfaz de usuario, la base de datos y el sistema de predicción.

La funcionalidad de la herramienta puede resumirse en la siguiente secuencia:

1- Seleccionar y visualizar un partido del histórico que se ha almacenado después del pre-procesado y transformación de los datos, así como el conjunto de variables que hemos fijado como explicativas.

2- Seleccionar de las posibles variables mostradas, las que se quieran incluir como entrada al sistema de predicción; pudiendo, además, modificar los valores de éstas. En estas posibles variaciones es donde el usuario puede plasmar sus hipótesis sobre dicha apuesta.

3- Una vez seleccionadas y modificadas las variables, se debe enviar una petición al sistema de predicción quedando, entonces, a la espera de los resultados.

4- Entonces, se generan los modelos de predicción, y se devuelve la respuesta, incluyendo las probabilidades calculadas y sus cuotas, tanto del modelo en el que se fuerce la inclusión de todas las variables seleccionadas, como las calculadas por el modelo con mínimas variables, denominada solución parsimoniosa.

5- Finalmente, se muestran los resultados en la interfaz de usuario

1.4.1 La interfaz de usuario

Como se ha descrito anteriormente, la herramienta necesita que el usuario seleccione un partido y sus variables, enviar una petición y mostrar los resultados.

Ya que esta herramienta, en principio, sería bastante útil para su uso en la red, como apoyo en el estudio de apuestas deportivas, se ha decidido desarrollar la interfaz de usuario en HTML y PHP. La Figura 8 muestra la vista del HTML resultante.

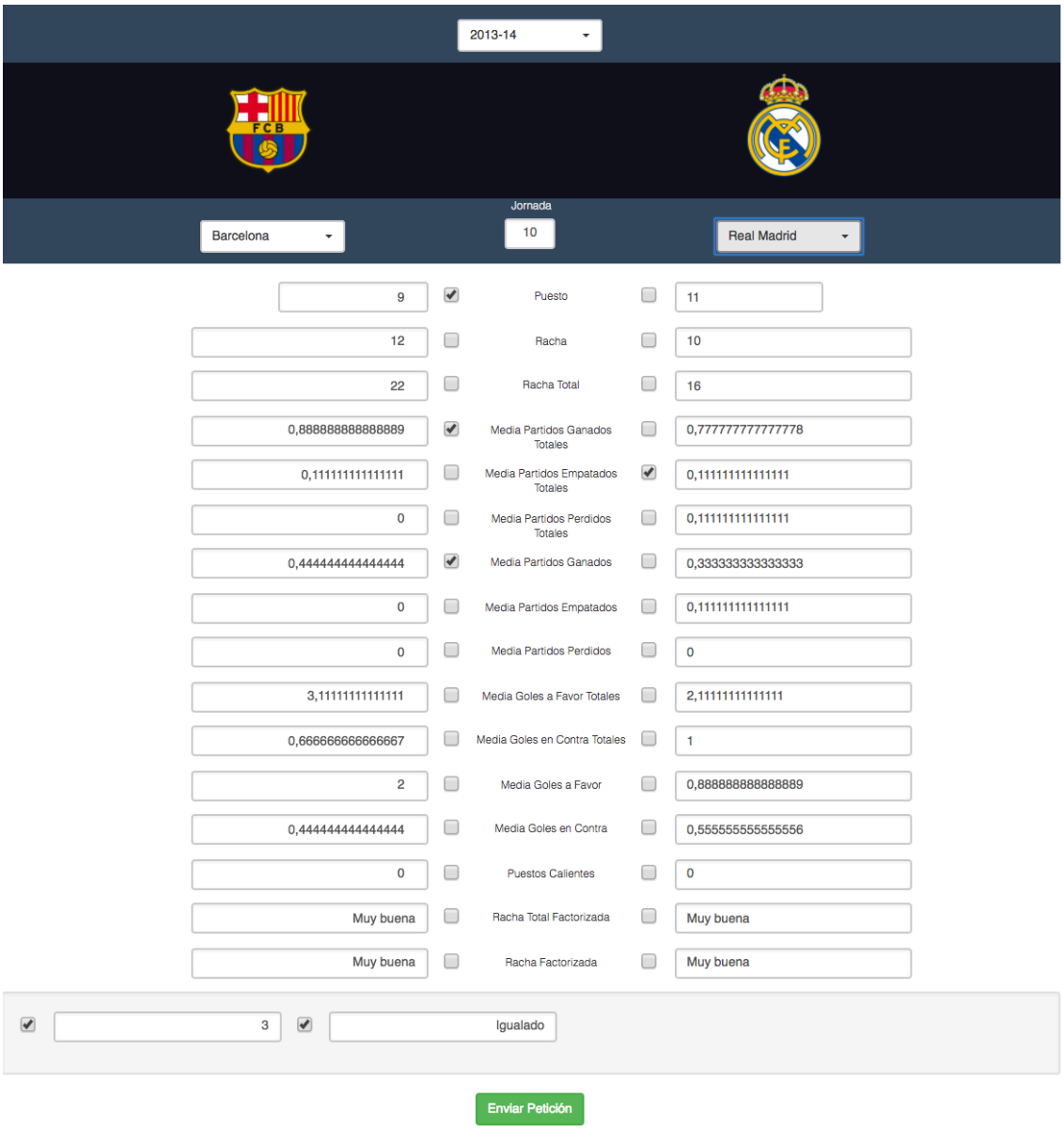


Figura 8. Interfaz de usuario

1.4.2 Base de datos.

La tabla que contiene el *dataset* con los partidos y las variables después del pre-procesado se almacenan en una base de datos para que, tanto la interfaz de usuario como el sistema de predicción, puedan acceder. En nuestro caso, se ha escogido un servidor MySQL por la alta versatilidad y fácil manejo que ofrece tanto para la interfaz como para las herramientas usadas en el sistema de predicción, concretamente hemos usado el software MAMP.

Además de almacenar el *dataset*, la base de datos se usa como "controlador" entre la interfaz de usuario y el sistema de predicción, almacenando las peticiones y respuestas.

1.4.3 Sistema de predicción.

Este sistema será el motor de la herramienta, y será el que realice las predicciones del partido seleccionado a través de un modelo de predicción. Este modelo se define tras realizar todas las fases de un estudio estadístico, automatizando el proceso a través del entorno y lenguaje de programación R.

R es uno de los lenguajes más utilizados por la comunidad estadística, siendo además muy popular en el campo de la minería de datos, la investigación biomédica, la bioinformática y las matemáticas financieras. A esto contribuye la posibilidad de cargar diferentes librerías o paquetes con funcionalidades de cálculo y gráficas.

De los modelos predictivos estudiados, los que mejor se adaptan a los requerimientos del proyecto son las redes neuronales y las regresiones logísticas, aunque como se expuso anteriormente; en general, las redes neuronales obtienen mejores resultados que las regresiones logísticas en estudios de clasificación. Sin embargo en nuestro caso, nos hemos decantado por las regresiones logísticas ya que estas no tienen la problemática de sobreentrenar el modelo como sí ocurre con las redes neuronales, en las cuales se requiere una gran experiencia. Además de esto y atendiendo al propósito del proyecto, una regresión logística devolverá en forma de coeficientes (o peso de cada variable del modelo), lo que será una utilidad añadida a la hora de analizar las variables propuestas.

Una vez se tenga una primera herramienta funcional con el entorno R, usando soluciones clásicas de minería de datos, se adaptará la misma implementando un nuevo sistema de predicción con las herramientas proporcionadas por Hadoop, donde existen paquetes específicos para poder usar el lenguaje de programación R.

La decisión de realizar este sistema con técnicas de minería de datos está apoyada en la existencia de mucha literatura, documentación y manuales sobre cómo aplicar estas técnicas en un entorno R, y así poder asimilar primero los conceptos básicos de estas técnicas antes de aplicarlo en un entorno de Big Data , el cual se encuentra en pleno desarrollo.

1.5 Estructura de la memoria

La memoria de este Proyecto Fin de Carrera está compuesta por 7 capítulos, 5 anexos , bibliografía, pliego de condiciones y presupuesto. A continuación se describe brevemente cada uno de ellos:

- **Capítulo 1. Introducción**. Introducción del tema tratado justificando su necesidad. Minería de datos. Apuestas deportivas. Se detalla el estado del arte actual. Se fijan objetivos y se especifica la estructura del sistema desarrollado.

- **Capítulo 2. Modelo de predicción**. Regresión logística multinomial. Descripción teórica de las regresiones logísticas. Se detallan todas las etapas de un estudio estadístico usando regresiones logísticas.

- **Capítulo 3. Base de datos y preprocesado**. Descripción de las características de la base de datos usada. Descripción de la etapa de

preprocesado con las diferentes técnicas aplicadas en cada momento. Análisis descriptivo.

- **Capítulo 4. Implementación del modelo de predicción**. Descripción de la metodología seguida para obtener un modelo de predicción, detallando cada apartado para su desarrollo.

- **Capítulo 5. Adaptación a BigData**. Se describe las diferentes herramientas que existenten dentro del ecosistema Hadoop. Puntualizando la selección escogida y desarrollo del sistema de predicción en el nuevo entorno.

- **Capítulo 6: Sistema general y caso de uso**. Se describe el funcionamiento de la herramienta, diagrama de flujos, comunicaciones entre módulos y se detalla un caso de uso.

- **Capítulo 7: Conclusiones**. Conclusiones obtenidas a partir del trabajo realizado y posibles líneas futuras.

- **ANEXO A: Contenido del CD.** Descripción del contenido del CD que se adjunta con esta memoria.

- **Bibliografía.** Se detalla toda la bibliografía utilizada en este trabajo.

- **Planos y Programas.** Se exponen los diferentes programas y funciones utilizados con sus respectivas descrpiciones.

- **Pliego de condiciones.** Se muestran las herramientas *software* y *hardware* utilizadas.

- **Presupuesto.** Se detalla el presupuesto necesario para la realización de este Proyecto Fin de Carrera.

CAPÍTULO 2: TEORÍA DE LOS MODELOS DE REGRESIÓN LOGÍSTICA MULTINOMIAL

2.1 Introducción

En este capítulo se explica la metodología que se utiliza para resolver el análisis estadístico planteado. Se utilizarán modelos de predicción, concretamente, el modelo de regresión logística multinomial. A continuación se describe este modelo explicando la formulación, los métodos de estimación, interpretación de parámetros, ajustes del modelo y validación.

Los modelos de regresión logística, son modelos estadísticos en los que se pretende conocer la relación entre una variable dependiente cualitativa, dicotómica (regresión logística binaria o binomial) o con más de dos categorías (regresión logística multinomial) y variables explicativas independientes, que pueden ser cualitativas o cuantitativas. Las variables cualitativas que sean dicotómicas, es aconsejable, que se codifiquen tomando valores 0, para una de las categorías (o para su ausencia) y 1 para la otra categoría (o para su presencia). Pero, si la variable cualitativa tuviera más de dos categorías, se realiza una transformación, para poderla incluir en el modelo. Esta transformación, consiste en crear varias variables cualitativas dicotómicas ficticias o de diseño, llamadas variables *dummies*, de forma que una de las variables se tomaría como categoría de referencia y cada una de las variables creadas entraría en el modelo de forma individual. En general, si la variable cualitativa posee n categorías, habrá que realizar n−1 variables ficticias [17] [18].

La regresión logística multinomial es utilizada en modelos con variable dependiente de tipo nominal con más de dos categorías y es una extensión multivariante de la regresión logística binaria clásica. Las variables independientes pueden ser tanto continuas como categóricas.

Estos modelos, se analizan eligiendo una categoría como referencia de la variable dependiente o de respuesta y se modelan varias ecuaciones simultáneamente, una para cada una de las restantes categorías respecto a la de referencia [19] [20].

2.2 Formulación e Interpretación del modelo

2.2.1 Formulación

Para los modelos de regresión logística binaria, si tenemos una variable dependiente Y, que toma valores Y=1 (presencia de una característica u otra categoría de la variable) e Y=0 (ausencia de la característica o la otra categoría de la variable), la ecuación de partida del modelo viene dada por:

$$\boldsymbol{P[Y=1|X]} = \frac{\boldsymbol{\exp\left(b_0 + \sum_{s=1}^{n} b_s x_s\right)}}{\boldsymbol{1 + \exp\left(b_0 + \sum_{s=1}^{n} b_s x_s\right)}} \qquad (1)$$

Donde $P[Y = 1|X]$ es la probabilidad de que Y tome el valor 1, en presencia de las variables X, que lo denotaremos por p(X).

X es un conjunto de n variables $\{x_1, x_2, x_3 \dots, x_n\}$ que forman parte del modelo; b_0 es la constante del modelo o término independiente y b_i los son los coeficientes de las variables.

Esta ecuación inicial del modelo es de tipo exponencial, pero se realiza su transformación logarítmica (*logit*), dada por:

$$\boldsymbol{Ln}\left[\frac{\boldsymbol{p(X)}}{\mathbf{1}-\mathbf{p(X)}}\right] = \boldsymbol{b_0} + \sum_{s=1}^{n} \boldsymbol{b_s x_s} \qquad (2)$$

De esta forma, se permite su uso como una función lineal más fácil de interpretar.

Para el caso de que la variable dependiente presente más de dos categorías, como es el caso que acontece, se utiliza el modelo de regresión logística multinomial que se modela, como se indicó anteriormente, mediante varios *logits* simultáneamente, uno para cada una de las restantes categorías respecto a la categoría de referencia que se haya considerado de la variable dependiente.

La variable resultado tiene tres categorías: Victoria local, empate o victoria visitante.

Se modelan dos *logits* simultáneamente:

- *logit* (empate / victoria local | z)= $a_1 + b_1 z$
- *logit* (victoria visitante / victoria local | z)= $a_2 + b_2 z$

La variable z es común en cada *logit*, pero se estiman coeficientes b_i diferentes para cada *logit* (incluso diferente constante, a_i).

Se considera una variable de respuesta politómica y con más de dos categorías de respuesta que detonaremos por $Y_1, Y_2 \ldots, Y_k$.

Se pretende explicar la probabilidad de cada categoría de respuesta en función de un conjunto de variables X = {x_1 , x_2 , ... , x_n } observadas. Es decir, ajustar un modelo de la forma:

$$\boldsymbol{p_j(x) = P[Y = Y_j | X = x] = f_j(x) \quad \forall j = 1, \ldots, k} \tag{3}$$

Para cada vector x de valores observados de las variables explicativas X.

En el caso de una variable de respuesta binaria, su distribución condicionada a cada combinación de valores observados de las covariables sigue una distribución de Bernouilli.

Cuando la variable de respuesta es politómica, la distribución de Bernouilli, se convierte en una distribución multinomial de parámetros, las probabilidades de cada una de las categorías de respuesta. Es decir, $(Y/X = x) \rightarrow M(1; p(x), \ldots, p(x))$, siendo $\sum_{j=1}^{k} p_j(x)$.

Así que para obtener un modelo lineal, se tiene $\binom{k}{2}$ transformaciones *logit* para comparar cada par de categorías de la variable respuesta, que sería de este tipo:

$$\boldsymbol{ln\left[\frac{\frac{p_i(x)}{p_i(x)+p_j(x)}}{\frac{p_j(x)}{p_i(x)+p_j(x)}}\right] = ln\left[\frac{p_i(x)}{p_j(x)}\right], \forall j = 1, \ldots, k \; (i \neq j)} \tag{4}$$

Que representan el logaritmo de la ventaja de respuesta Y_i frente a Y_j condicionado a las observaciones de las variables independientes que caen en uno de ambos niveles.

Pero para construir el modelo *logit* de respuesta multinomial bastaría con considerar $(k-1)$ transformaciones *logit* básicas, definidas con respecto a una categoría de referencia. Tomando como categoría de referencia la última Y_k . Así las transformaciones *logit* se definen como $L_j(x)$ el logaritmo de ventaja de respuesta $\mathrm{Y_j}$ dado que las observaciones de las variables independientes caen en la categoría $\mathrm{Y_j}$ o en la $\mathrm{Y_k}$.

El modelo lineal para cada una de las transformaciones logit generalizadas, para n variables explicativas, es de la siguiente forma:

$$\boldsymbol{L_j(x) = \sum_{s=0}^{n} b_{sj} x_s = x b_j \quad \forall j = 1, \dots, k \; (i \neq j)} \tag{5}$$

Para cada vector de valores observados de las variables explicativas x=($x_0, x_1, x_2 \dots x_n$) con x_0 =1 y b_j =($b_{0j}, b_{1j}, \dots, b_{nj}$) el vector de parámetros asociado a la categoría Y_j.

Para las probabilidades de respuesta, podemos escribir el modelo de la siguiente forma, o equivalentemente, podemos obtener de ambas expresiones, una expresión reducida del modelo:

$$\boldsymbol{P_j(x) = \frac{exp(\sum_{s=0}^{n} b_{sj} x_s)}{1 + \sum_{j=1}^{k-1} exp(\sum_{s=0}^{n} b_{sj} x_s)} \qquad \forall j = 1, \dots, k-1} \tag{6}$$

$$\boldsymbol{P_k(x) = \frac{1}{1 + \sum_{j=1}^{k-1} exp(\sum_{s=0}^{n} b_{sj} x_s)} \qquad \forall j = 1, \dots, k-1} \qquad (7)$$

2.2.1.1 Interpretación del modelo

A continuación mostramos la interpretación de los parámetros del modelo, distinguiendo los casos según sean las variables explicativas, cuantitativas o cualitativas.

- Una variable predictora cuantitativa X.

Si en el modelo solo tenemos una única covariable cuantitativa X, el modelo para cada valor observado x de la variable X viene dado por:

$$\boldsymbol{L_j(x) = a_j + b_j x, \ \forall j = 1, \dots, k-1} \qquad (8)$$

A continuación mostramos la exponencial de los parámetros b_j asociados a cada categoría de la variable dependiente, que se interpreta en términos de cocientes de ventajas (odds ratio):

$$\boldsymbol{\theta_j(\triangle X = 1) = \frac{\frac{p_j(x+1)}{p_k(x+1)}}{\frac{p_j(x)}{p_k(x)}} = \frac{exp(a_j + b_j(x+1))}{exp(a_j + b_j x)} \quad \forall j = 1, \dots, k-1} \qquad (9)$$

$\theta_j(\triangle X = 1)$ es el cociente de ventajas de respuesta Y_j frente a la última categoría, Y_k cuando aumenta en una unidad la variable X.

- Más de una variable predictora cuantitativa.

Para el modelo *logit* generalizado múltiple, los cocientes de ventajas se definen incrementando una de las variables y controlando fijas las demás.

$$\boldsymbol{\theta_j(\Delta X_r = 1 | X_s = x_s, s \neq r) = \frac{\frac{P[Y = Y_j \mid X_r = x_r + 1, X_s = x_s \neq r]}{P[Y = Y_k \mid X_r = x_r + 1, X_s = x_s \neq r]}}{\frac{P[Y = Y_j \mid X_r = x_r, X_s = x_s \neq r]}{P[Y = Y_k \mid X_r = x_r, X_s = x_s \neq r]}} = exp(b_{rj})} \quad (10)$$

$$\forall j = 1, \dots, k-1$$

siendo $\theta_j(\Delta X_r = 1 | X_s = x_s, s \neq r)$ el cociente de ventajas de respuesta Y_j frente a la última categoría, Y_K cuando aumenta en una unidad la variable X_r y las demás se controlan fijas.

- Variables predictoras categóricas.

Si se incluyen en el modelo variables independientes categóricas, se introducen mediante sus variables del diseño asociadas (variables *dummies*).

Supongamos que tenemos la variable categórica A con categorías $A_1, \dots, A_p$. Si de esta variable realizamos la transformación a variables de diseño mediante el método parcial que asigna un uno a la variable asociada a cada categoría y un cero al resto, y tomando como categoría de referencia la primera, obtenemos $p-1$ variables que las denotamos como $X_m^A (m = 2, \dots, p)$.

Así que, el modelo de regresión logística multinomial generalizado que obtenemos sigue siendo un modelo lineal, como en los casos anteriores, para cada *logit* generalizado en función de esas variables de diseño procedentes de la variable A y viene dado por:

$$\boldsymbol{L_{j/l} = ln\left[\frac{P_{j/l}}{P_{j/l}}\right] = b_{0j} + \sum \tau_{mj}^A \tau_{lm}^A, \quad l = 1, \dots, p; j = 1, \dots, k-1} \quad (11)$$

siendo $L_{j/l} = P[Y = Y_j \ / \ A = A_l]$. La probabilidad de respuesta Y_j en la categoría A_l.

Este modelo, en términos de cocientes de ventajas viene dado por:

$$\boldsymbol{\theta_{j/l1}} = \frac{\dfrac{\boldsymbol{p_{j/l}}}{\boldsymbol{p_{k/l}}}}{\dfrac{\boldsymbol{p_{j/1}}}{\boldsymbol{p_{k/1}}}} = \frac{\boldsymbol{exp(b_{0j} + \tau_{lj})}}{\boldsymbol{exp(b_{0j})}} \quad \forall \boldsymbol{j = 1, \dots, k-1}, \forall \boldsymbol{l = 2, \dots, p} \tag{12}$$

Que es cociente de ventaja de respuesta Y_j frente a la ultima Y_k para la categoría A_l de A respecto a la primera categoría A_1.

2.2.2 Otros aspectos a tener en cuenta sobre las variables

Para seleccionar el conjunto de variables predictoras que se incluyen en el modelo, se deben tener en cuenta todas aquellas variables que se consideren importantes para el modelo, independientemente de si se ha demostrado o no significación estadística en un análisis univariado previo, ya que puede conducir a dejar de incluir en el modelo variables con una débil asociación a la variable dependiente en solitario, pero que podrían demostrar ser fuertes predictores de la misma al tomarlas en conjunto con el resto de variables.

Con todo esto, debemos conseguir obtener un modelo que sea lo más reducido posible y que explique los datos (principio de parsimonia). Posiblemente un mayor número de variables en el modelo implicaría mayores errores estándar.

Cuando sean seleccionadas todas las covariables para ser incluidas en el modelo, se debe proceder a obtener el modelo más reducido que siga explicando los datos. Para ello se puede recurrir a métodos de selección.

Otro aspecto a tener en cuenta para elegir el número de covariables a incluir en un modelo de regresión logística, es el tamaño muestral. Ya que, modelos excesivamente grandes, para muestras con tamaños muestrales relativamente pequeños, podrían provocar errores estándar grandes o

coeficientes estimados falsamente muy elevados (sobreajuste). Por lo que se suele recomendar, que por cada covariable, se cuente con un mínimo entre diez y quince individuos por cada categoría de la variable dependiente con menor representación.

2.3 Métodos de estimación.

2.3.1 Estimación por máxima verosimilitud

Para la estimación, de los coeficientes del modelo y de sus errores estándar, se utiliza la estimación por máxima verosimilitud, es decir, estimaciones que hagan máxima la probabilidad de obtener los valores de la variable dependiente y proporcionados por los datos de la muestra. Al contrario de lo que ocurre con la estimación de los coeficientes de regresión lineal múltiple que se utiliza el método de los mínimos cuadrados, los cálculos para las estimaciones de los coeficientes de la regresión logística multinomial no son directos, hay que llevar a cabo métodos iterativos, como el método de *Newton–Raphson.*

Al aplicar estos métodos, además de obtener las estimaciones de los coeficientes de regresión, se obtienen sus errores estándar y las covarianzas entre las covariables del modelo.

A continuación, describimos el método de estimación de máxima verosimilitud para el cálculo de los coeficientes de nuestro modelo de regresión logística multinomial.

Supongamos, que disponemos de una muestra aleatoria de tamaño N con Q combinaciones diferentes de valores de las variables explicativas $X_1,...,X_k$. Denotemos a cada combinación de valores de las variables

explicativas por $x_q = (x_{q0}, x_{q0}, \dots, x_{qm})' \; con \; x_{q0}$=1 $\forall q$=1,...,Q . En cada una de estas combinaciones se tiene una muestra aleatoria de x_{q0} observaciones independientes de la variable de respuesta politómica Y, de entre las cuales denotamos por $y_{j/q}$ al número de observaciones que caen en la categoría de respuesta $Y_j \forall \; j = 1, \dots, k$. Así que, se verifica que :

$$\sum_{j=1}^{k} \boldsymbol{y_{j/q}} = \boldsymbol{d_q} \; \boldsymbol{y} \; \sum_{q=1}^{Q} \boldsymbol{d_q} = \boldsymbol{N} . \tag{13}$$

Los vectores $(\boldsymbol{y_{j/1}} \dots, \boldsymbol{y_{k/q}})' \; \forall \; q = 1, \dots, Q$ siguen una distribución de probabilidad multinomiales independientes, $M(\boldsymbol{d_q}; \boldsymbol{p_{1/q}} \dots, \boldsymbol{p_{k/q}})$, siendo $\boldsymbol{p_{j/q}} = \boldsymbol{P}[\boldsymbol{Y} = \boldsymbol{Y_j} \; / \; \boldsymbol{X} = \boldsymbol{x_q}]$ y verificando que $\sum_{j=1}^{k} \boldsymbol{P_{j/q}} = \boldsymbol{1}$

Por tanto, la función de verosimilitud de los datos viene dada por:

$$\boldsymbol{V} = \prod_{q=1}^{Q} \left(\frac{\boldsymbol{d_q}!}{\prod_{j=1}^{k} (\boldsymbol{y_{j/q}})} \prod_{j=1}^{k} \boldsymbol{p_{j/q}}^{y_{j/q}} \right) \tag{14}$$

Así que, el núcleo de la log-verosimilitud es: $K = \sum_{q=1}^{Q} \sum_{j=1}^{k} y_{j/q} \ln (p_{j/q})$

Normalmente, en vez de utilizar la función de verosimilitud se utiliza la siguiente función auxiliar:

$$\boldsymbol{\Lambda} = \boldsymbol{-2ln(V)} \tag{15}$$

Por lo que el problema de maximizar la verosimilitud equivale al de minimizar esta función auxiliar.

Teniendo en cuenta la ecuación del modelo *logit* generalizado multinomial, y sustituyendo en la expresión anterior, obtenemos la siguiente expresión del núcleo de la log-verosimilitud:

$$\mathbf{K} = \sum_{q=1}^{Q}\sum_{j=1}^{k} \boldsymbol{y_{j/q}}\left(\sum_{S=0}^{n} \boldsymbol{b_{sj}x_{qs}}\right) - \sum_{q=1}^{Q}\left(\sum_{j=1}^{k} \boldsymbol{y_{j/q}}\right)\boldsymbol{ln}\left(\sum_{j=1}^{k} \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj}x_{qs}}\right)\right) \tag{16}$$

$$= \sum_{q=1}^{Q}\sum_{j=1}^{k} \boldsymbol{y_{j/q}}\left(\sum_{S=0}^{n} \boldsymbol{b_{sj}x_{qs}}\right) - \sum_{q=1}^{Q} \boldsymbol{n_q ln}\left(\sum_{j=1}^{k} \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj}x_{qs}}\right)\right)$$

Derivando respecto de los parámetros:

$$\frac{\boldsymbol{\Delta K}}{\boldsymbol{b_{sj}}} = \sum_{q=1}^{Q} \boldsymbol{y_{jq}x_{qs}} - \sum_{q=1}^{Q} \boldsymbol{n_q x_{qs}} \frac{\boldsymbol{exp}(\sum_{s=0}^{n} \boldsymbol{b_{sj}x_{qs}})}{\sum_{j=1}^{k} \boldsymbol{exp}(\sum_{s=0}^{n} \boldsymbol{b_{sj}x_{qs}})} \tag{17}$$

Así, obtenemos las ecuaciones de verosimilitud con forma matricial:

$$\mathbf{X'}_{((n+1)xQ)}\boldsymbol{y_{j(Qx1)}} = \mathbf{X'}_{((n+1)xQ)}\boldsymbol{\hat{m}_{j(Qx1)}} \;\; \forall \boldsymbol{j} = \mathbf{1}, \dots, \boldsymbol{k-1} \tag{18}$$

Siendo $y_j = (y_{j/1}, \dots, y_{j/Q})'$ y $\hat{m}_j = (\hat{m}_{j/1}, \dots, \hat{m}_{j/Q})$ con $\hat{m}_{j/q}$ la frecuencia esperada de respuesta Y_j en la combinación x_q de valores observados de las variables predictoras, estimada bajo el modelo y definida como $\hat{m}_{j/Q} = d_q \hat{p}_{j/q}$.

Para obtener los estimadores de máxima verosimilitud hay que resolver $k-1$ sistemas de $n+1$ ecuaciones no lineales. Así que para resolverlo utilizamos el método iterativo de *Newton-Raphson*.

Con este método obtenemos el estimador de los parámetros b, que es una matriz de dimensión $(n+1)x(k-1)$ formado por las siguientes columnas: $\hat{b} = \left(\widehat{b'}_1, \widehat{b'}_2 \dots, \widehat{b'}_{k-1}\right)$ siendo $\widehat{b'}_j$ el estimador de máxima verosimilitud del vector de parámetros asociado a la categoría de la variable respuesta Y_j.

A continuación obtendremos la matriz de covarianzas de b, que es la inversa de la matriz de información de Fisher. Calculamos primero la matriz de covarianzas de cada vector de parámetros $\widehat{b'}_j$.

Para ello hay que calcular las derivadas segundas de K con r=s:

$$\frac{\boldsymbol{\Delta^2 K}}{\boldsymbol{\Delta b_{rj} \Delta b_{sj}}} = -\sum_{q=1}^{\boldsymbol{Q}} \boldsymbol{n_q x_{qs} x_{qr}} - \frac{\boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{qs}}\right)\left[\sum_{j=1}^{k} \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{qs}}\right) - \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{q}}\right.\right.}{\left[\sum_{j=1}^{k} \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{qs}}\right)\right]^2} \quad (19)$$

Así que la matriz de covarianzas viene dada por:

$$\boldsymbol{cov}\left(\widehat{b'}_j\right) = \left[-\boldsymbol{E}\left(\frac{\boldsymbol{\Delta^2 K}}{\boldsymbol{\Delta b_{rj} \Delta b_{sj}}}\right)\right]^{-1} = \left[\boldsymbol{X' Diag}\left[\boldsymbol{d_q p_{j/q}}(\boldsymbol{1 - p_{j/q}})\right]\boldsymbol{X}\right]^{-1} \quad (20)$$

Calculamos ahora las matrices de covarianzas cruzadas entre cada par de estimadores $\widehat{b'}_j$ y $\widehat{b'}_i (\, i \neq j)$. Para ello se calculan las siguientes derivadas segundas de K con r≠ $s\ y\ j \neq i$.

$$\frac{\boldsymbol{\Delta^2 K}}{\boldsymbol{\Delta b_{rj} \Delta b_{sj}}} = -\sum_{\boldsymbol{q=1}}^{\boldsymbol{Q}} \boldsymbol{n_q x_{qs} x_{qr}} \frac{-\boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{qs}}\right)\boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{si} x_{qs}}\right)}{\left[\sum_{j=1}^{k} \boldsymbol{exp}\left(\sum_{s=0}^{n} \boldsymbol{b_{sj} x_{qs}}\right)\right]^2} \quad (21)$$

Dando lugar a la siguiente expresión de la matriz de covarianzas:

$$\boldsymbol{cov}(\hat{b}_j, \hat{b}_i) = \left[-\boldsymbol{E}\left(\frac{\boldsymbol{\Delta^2 K}}{\boldsymbol{\Delta b_{ri} \Delta b_{sj}}}\right)\right]^{-1} = \left[-\boldsymbol{X' Diag}\left[\boldsymbol{d_q p_j p_i}\right)\right]\boldsymbol{X}\right]^{-1} \quad (22)$$

2.3.2 Descenso por gradiente estocástico (SGD)

El algortimo SGD se engloba dentro de los algoritmos online que a diferencia de los algoritmos bloque procesan un dato o subconjunto de datos en cada iteración del algoritmo. En cada iteración del algoritmo se obtiene una nueva estimación de los parámetros. Por tanto, la diferencia principal, es que la actualización se produce después de la llegada de cada nueva observación o subconjunto de observaciones mientras que en los algoritmos bloque se actualizan cuando se han procesado todas las observaciones [21], [22].

El algoritmo SGD es un método iterativo de optimización de funciones con la siguiente forma:

$$\boldsymbol{Q(\theta)} = \sum_{\boldsymbol{j=1}}^{\boldsymbol{j}} \boldsymbol{q_j(\theta)} \quad (23)$$

Donde cada función $q_j(\theta)$ es una función diferenciable y $Q(\theta)$ es la función objetivo o función de coste a optimizar. Para minimizar $Q(\theta)$ la iteración j-ésima del algoritmo es:

$$\boldsymbol{\hat{\theta}^{(j)} = \hat{\theta}^{(j-1)} - \gamma^{(j)} \nabla_\theta q_j\left(\hat{\theta}^{(j-1)}\right), \quad 0 < \gamma^{(j)} \leq 1} \quad (24)$$

Donde $\gamma^{(j)}$ es la constante de paso o constante de aprendizaje. En el caso de buscar la maximización de la función de coste, la expresión (24) se modificaría cambiando el signo negativo delante de la constante de paso por un signo positivo.

El algoritmo se inicializa partiendo un vector inicial de parámetros $\hat{\theta}^{(j)}$ y unos valores de $\gamma^{(j)}$ que deben ser determinados a priori.

2.3.2.1 Algoritmo SGD *mini-batch*

Este algoritmo es una variante del algoritmo SGD. La diferencia es que se procesa en cada iteración un bloque de b observaciones en lugar de una observación. El parámetro b es conocido como tamaño del batch [23], [24]. Se calculan los gradientes para todo todas las observaciones de un mismo bloque y se promedia el resultado obtenido. Con este resultado se procede a la actualización del parámetro a estimar de la siguiente forma

$$\widehat{\boldsymbol{\theta}}^{(k)} = \widehat{\boldsymbol{\theta}}^{(k-1)} + \frac{\mathbf{1}}{\boldsymbol{b}}\boldsymbol{\gamma}^{(k)} \sum_{\boldsymbol{j}=(\boldsymbol{k}-\mathbf{1})\boldsymbol{b}+\mathbf{1}}^{\boldsymbol{kb}} \boldsymbol{\nabla_{\theta} q_j}\left(\widehat{\boldsymbol{\theta}}^{(k-1)}\right) \tag{25}$$

El tamaño del bloque adecuado es un parámetro a determinar de forma experimental, típicamente $2 \leq b \leq 100$ [24]. Una ventaja a destacar del algoritmo SGD *mini-batch,* es que puede ser más rápido que el algoritmo SGD debido a que se puede implementar de forma vectorizada y utilizar procesadores en paralelo para realizar los cálculos. Nótese que para el problema se reduce al caso del algoritmo SGD convencional y para al Descenso por Gradiente [25].

2.4 Bondad de ajuste del modelo

2.4.1 Contrastes de bondad de ajuste del modelo

Uno de los primeros indicadores de importancia para apreciar el ajuste del modelo logístico multinomial, es el doble logaritmo del estadístico de verosimilitud (*likelihood*). Se trata de un estadístico que sigue una distribución similar a X^2 (31).

Sea $y_{j/q}$ el número de observaciones que caen en la categoría de respuesta $Y_j \; \forall \; j = 1, \ldots, k.$ y sean las d_q observaciones correspondientes a la q-ésima combinación de valores de las variables explicativas.

Denotamos por $\hat{m}_{j/q}$ la frecuencia esperada de respuesta Y_j en la combinación x_q de valores observados de las variables predictoras, estimada bajo el modelo y definida como $\hat{m}_{j/q} = d_q \hat{p}_{j/q}$

Así que, para contrastar la bondad del ajuste global del modelo cuando el número de observaciones en cada combinación de valores de las variables explicativas es grande se utiliza el estadístico *Chi-cuadrado de Pearson* y el estadístico de *Wilks de Razón de Verosimilitudes.*

El test global de bondad de ajuste del modelo de regresión logística multinomial múltiple contrasta el siguiente contraste de hipótesis:

$$\boldsymbol{H_0: p_{j/q} = \frac{exp\left(\sum_{s=0}^{n} b_{sj}\, x_{qs}\right)}{1 + exp\left(\sum_{s=0}^{n} b_{sj}\, x_{qs}\right)} \forall q = 1,..,Q; \quad \forall j = 1,..,k}$$

$$\boldsymbol{H_0: p_{j/q} \neq \frac{exp\left(\sum_{s=0}^{n} b_{sj} x_{qs}\right)}{1+exp\left(\sum_{s=0}^{n} b_{sj} x_{qs}\right)} \text{ para algún q y j}} \tag{26}$$

2.4.1.1 Test Chi-cuadrado de Pearson

El estadístico *Chi-cuadrado de Pearson* de bondad de ajuste a un modelo de regresión logística multinomial, M de la forma anterior viene dado por:

$$\boldsymbol{X^2(M) = \sum_{q=i}^{Q} \sum_{j=i}^{k} \frac{\left(y_{j/q} - d_q \hat{p}_{j/q}\right)^2}{d_q \hat{p}_{j/q}}} \tag{27}$$

Siendo $\hat{p}_{j/q}$la estimación por máxima verosimulitud de $p_{j/q}$.

Este estadístico tiene distribución asintótica Chi-cuadrado con grados de libertad obtenidos como la diferencia entre el número de parámentros $\hat{p}_{j/q}$ y el número de parámetros independientes en el modelo, $Q - (n+1)x(k-1)$. Es decir,

$$\boldsymbol{X^2(M) \xrightarrow{d} X_q^2 - (n+1)x(k-1), si\ d_q \to \infty} \tag{28}$$

Así que se rechaza la hipótesis nula con un nivel de significación α cuando $X^2(M)_{Obs} \geq X_Q^2 - (n+1)x(k-1); \propto$. O, equivalentemente, podemos definir el p-valor del contraste, como la probabilidad acumulada a la derecha del valor observado: $p\_valor = P[X^2(M) \geq X^2(M)_{Obs}]$ se rechaza la hipótesis

nula cuando p-valor ≤ α.

2.4.1.2 *Test Chi-cuadrado de Razón de Verosimilitudes. Estadístico de Wilks. Devianza*

El *estadístico de Wilks de Razón de Verosimilitudes*, para el contraste de bondad de ajuste del modelo de regresión logística multinomial M se obtiene como menos dos veces el logaritmo del cociente entre el supremo de la verosimilitud bajo la hipótesis nula y el supremo de la verosimilitud en la población. A partir de esta expresión operando se obtiene la expresión de este estadístico que viene dada por:

$$\boldsymbol{G^2(M) = 2\left[\sum_{q=1}^{Q}\sum_{j=1}^{k} y_{j/q} \ln\left(\frac{y_{j/q}}{\hat{m}_{j/q}}\right)\right]} \tag{29}$$

Este estadístico tiene distribución asintótica *Chi-cuadrado* con grados de libertad, la diferencia entre la dimensión del espacio paramétrico y la dimensión de este espacio bajo la hipótesis nula. Para un modelo de regresión logística multinomial los grados de libertad es la diferencia entre el número de parámetros $p_{j/q}$ y el número de parámetros b_{Sj} bajo el modelo, es decir,

$$\boldsymbol{Q - (n+1)x(k-1)\ grados\ de\ libertad}$$

$$\boldsymbol{G^2(M) \xrightarrow{d} X_q^2 - (n+1)x(k-1), si\ d_q \to \infty} \tag{30}$$

Así que se rechaza la hipótesis nula con un nivel de significación α cuando $G^2(M)_{Obs} \geq X_Q^2 - (n+1)x(k-1); \alpha$. O equivalentemente cuando $p\_valor = P[G^2(M)_{Obs} \geq G^2(M)_{Obs}] \leq \propto$. Al *estadístico de Wilk*, $G^2(M)$, se le

denomina *devianza*.

2.5 Inferencia en regresión logística multinomial

Con un modelo predictivo lo que se busca es que a través de los datos procedentes de una muestra, se extrapolen los resultados a la población general, algunos parámetros que mide la capacidad de adaptación a otra muestra son los intervalos de confianza.

Basándonos en la normalidad asintótica de los estimadores de máxima verosimilitud se pueden construir intervalos de confianza asintóticos para cada uno de los parámetros del modelo, utilizando la distribución normal, y mediante las transformaciones correspondientes, intervalos de confianza para las *odds ratio*.

- **Intervalos de confianza para los parámetros**: se construye un intervalo de confianza con nivel de confianza $1-\alpha$ para cada parámetro del modelo de regresión logística multinomial, b_{sj} con $j = 1\ldots,k$. La distribución asintótica de $\hat{b}_{sj}$ es $Nb\left(b_{sj}, \hat{\sigma}^2(\hat{b}_{sj})\right)$ donde $\hat{\sigma}^2(\hat{b}_{sj})$ es el valor correspondiente al error estándar del estimador del parámentro b_{sj}. Así tenemos que:

$$P - \left[-Z_{\alpha/2} \leq \frac{\hat{b}_{sj} - b_{sj}}{\hat{\sigma}(\hat{b}_{sj})} \leq Z_{\alpha/2}\right] = 1 - \alpha \qquad (31)$$

Por lo que obtenemos así el intervalo de confianza aproximado para b_{sj} al nivel $1-\alpha$:

$$\boldsymbol{IC}(b_{sj}) = \left(\hat{b}_{sj} \pm z_{\propto/2}\hat{\sigma}^2(\hat{b}_{sj})\right) \quad (32)$$

- **Intervalos de confianza para las odds ratio:** Sabemos que los cocientes de ventajas vienen dados por:

$$\theta_j(\Delta X_r = 1/X_s = x_{s'} s \neq r) = exp(b_{rj}) \;\; \forall r = 1, \ldots . n; \; \forall j = 1, \ldots, k-1 \quad (33)$$

Por lo tanto, el intervalo de confianza para los cocientes de ventajas se calcula tomando exponenciales en el intervalo de confianza obtenido anteriormente para cada uno de los parámetros b_{sj} . Así que el intervalo de confianza para $\exp(b_{sj})$ al nivel de confianza 1- α, viene dado por:

$$\boldsymbol{IC}\left(\boldsymbol{exp}(b_{rj})\right) = \boldsymbol{exp}\left(\left(\hat{b}_{sj} \pm z_{\propto/2}\hat{\sigma}(\hat{b}_{sj})\right)\right) \quad (34)$$

2.6 Métodos de selección del modelo

Una vez conocido el procedimiento de ajuste de modelos de regresión logística multinomial, el siguiente paso es el desarrollo de estrategias para seleccionar las variables que mejor explican a la variable de respuesta. Para ello se adoptará el principio de parsimonia, que consiste en seleccionar el modelo que con menor número de parámetros se ajuste bien a los datos y lleve a una interpretación sencilla en términos de cocientes de ventajas.

A continuación se describe el método de selección hacia delante para ajustar un modelo, aunque existen otros métodos no existen evidencias científicas de cúal es más eficiente.

- **Hacia adelante**
    - Se inicia con un modelo sin variables explicativas.
    - Se ajusta un modelo y se calcula el p-valor del contraste de razón de verosimilitud que resulta de incluir cada variable por separado.
    - Se selecciona el modelo con el criterio de selección, por ejemplo el p-valor más significativo.
    - Se ajusta de nuevo un modelo con las variables seleccionadas y se calcula el criterio selección de añadir cada variable no seleccionada anteriormente por separado.
    - Se selecciona el modelo con mejor significación.
    - Se repite los dos últimos pasos hasta que no queden variables significativas que incluir.

2.7 Validación del modelo

Existen dos componentes dentro de la validación de un modelo para las regresiones logísticas; la discriminación y la calibración, según la solución u objetivo que se persiga se medirá una de las dos o ambas. La calibración mide cómo de precisas son las probabilidades calculadas, mientras que en un análisis discrimintante se mide el porcentaje de acierto de las observaciones.

En este proyecto, al igual que en las apuestas, se pretende buscar un acierto a largo plazo y no individualmente, por lo tanto, se busca una buena calibración siendo la discriminación totalmente indiferente para le propósito del proyecto.

Además de la calibración existe otro componente importante para poder validar un modelo predictivo, la validez externa que está relacionada con el término reproducibilidad, que es la capacidad de realizar predicciones válidas a una muestra que no ha sido incluida en el cálculo del modelo.

Uno de los test más usuados para medir la calibración del modelo estimado, es a través de los estadisiticos de *Hosmer Lemeshow* [26], dependiendo de la muestra que se evalúe obtendremos la validez interna o externa.

2.7.1 Hosmer Lemeshow

Siendo Y una variable categórica con *c* posibles categorías, codificadas (0,,…,*c*-1). Y la categoría de referencia es Y=0, siendo x un vector de variables predictoras $X = (x_1, x_2, \dots x_p)$.

Tomando una muestra de n observaciones independientes, (x_i, y_i), $i = 1, \dots, n$. Recodificando y_i en una variable binaria siendo $\widetilde{y_{ij}}$, $\widetilde{y_{ij}} = 1$, cuando $y_i = j$ e $\widetilde{y_{ij}} = 0$ cuando toma otra categoría $(i = 1, \dots, n \; and \; j = 0, \dots, c - 1)$. Despues de tomar los valores de predichos, tomamos $\widehat{\pi_{ij}}$ como las probabilidades estimadas para cada observación $(i = 1, \dots, n)$ para cada posible categoría $(j = 0, \dots, c - 1)$.

El test se basa en la estrategia de ir clasificando las observaciones en g grupos de n/g observaciones. Para cada grupo calculamos la suma de observaciones y la frecuencia de estimaciones obtenidas para cada categoría,

$$O_{kj} = \sum_{l \in \Omega_k} \widetilde{y_{lj}}$$
$$E_{kj} = \sum_{l \in \Omega_k} \widehat{\pi_{lj}} \qquad (32)$$

donde $k = 1, \ldots, g$; $j = 0, \ldots, c-1$; y Ω_k son los índices de n/g observaciones del grupo k. Un resumen útil del test se obtiene colocando los valores O_{kj} y E_{kj}como en la Figura 9:

Group	$Y=0$		$Y=1$		$\cdots$	$Y=c-1$	
1	O_{10}	E_{10}	O_{11}	E_{11}	$\cdots$	$O_{1,c-1}$	$E_{1,c-1}$
2	O_{20}	E_{20}	O_{21}	E_{21}	$\cdots$	$O_{2,c-1}$	$E_{2,c-1}$
$\vdots$	$\vdots$		$\vdots$		$\ddots$	$\vdots$	
g	O_{g0}	E_{g0}	O_{g1}	E_{g1}	$\cdots$	$O_{g,c-1}$	$E_{g,c-1}$

Figura 9 Tabla de contingencia Observados y estimados

El test de bondad de ajuste es el *estadístico Chi-cuadrado de Pearson* de la tabla de frecuencias de observados y estimados.

$$C_g = \sum_{k=1}^{g} \sum_{j=0}^{c-1} (O_{kj} - E_{kj})^2 / E_{kj} \tag{33}$$

Bajo la hipótesis nula que el modelo estimado se ajusta suficientemente a los datos, la distribución C_g es *Chi-cuadrado* y tiene $(g-2)\times(c-1)$ grados de libertad.

CAPÍTULO 3: BASE DE DATOS Y PREPROCESADO

3.1 Introducción

El objetivo de este capítulo es detallar la base de datos usada para la herramienta, así como el preprocesado, creando variables que sean de interés para el pronóstico del resultado de un partido de fútbol. También se detallará un análisis descriptivo de las variables, paso previo que se realiza antes del inicio de un estudio estadístico para conocer las características básicas del caso de estudio.

3.2 Base de datos

Para la realización de este proyecto se adquirió una base de datos SQL de los partidos de la Liga de Fútbol Profesional, a través de una página especializada en estadísticas de fútbol [27]. Esta base de datos se compone de diez temporadas completas de la Liga (Primera División de fútbol español) desde la temporada 2005-06 hasta la 2014-15. Cada temporada de fútbol de la liga consta de 38 jornadas, en la que se enfrentan 20 equipos, por lo tanto 10 partidos por jornada, haciendo un total de 3.800 partidos compuestos por las variables y formato ilustrado en la Tabla 3:

Nombre	Tipo	Descripción
id	INT	Identificador único del partido
temporada	VARCHAR	Años que forman la temporada del partido disputado
categoría	VARCHAR	División a la que pertenece los equipos (todos 1ª)
jornada	VARCHAR	Número de jornada que se disputó el partido
Id_local	INT	Identificador del equipo local

Id_visitante	INT	Identificador del equipo visitante
nombre_local	VARCHAR	Nombre del equipo local
nombre_visitante	VARCHAR	Nombre del equipo visitante
goles_local	INT	Nº de goles marcado en el partido por el equipo local
goles_visitante	INT	Nº de goles marcado en el partido por el equipo visitante
árbitro	VARCHAR	Nombre del colegiado
estadio	VARCHAR	Nombre del estadio de fútbol
fecha	VARCHAR	Fecha en que se disputó el partido

Tabla 3 Campos base de datos inicial

La base de datos original es muy básica, conteniendo pocas variables susceptibles de ser incorporadas a un modelo predictivo. Esto hace que sea necesario preprocesar y trasformar la muestra, para crear, a partir de estos datos nuevas variables que añadir al *dataset.* Esta tarea será posible al tener todos los datos de los partidos de cada temporada; por ejemplo, una tabla de clasificación por jornadas con los datos estadísticos de la temporada hasta ese momento.

3.3 Preprocesado y transformación

Para el preprocesado, como se ha comentado anteriormente, se ha utilizado el lenguaje de programación R. Existe una gran cantidad de librerías para realizar cálculos, gráficas, modelados, entre otras tareas.

Para conectarse con el servidor de la base de datos se ha usado el paquete RMySQL [28].

Se ha considerado indispensable incluir las variables de la clasificación como variables explicativas del resultado, ya que estos datos proporcionan

información útil sobre el estado de cada equipo, por ejemplo, la posición del equipo en la tabla o la diferencia de puntos conseguidos en ese momento de la temporada, cabe esperar que influya en la probabilidad del resultado.

Esta clasificación se ha construido recorriendo la totalidad de partidos ordenadamente, calculando la diferencia entre “goles_local” y “goles_visitante” de la base de datos original, sumando los puntos según el resultado a la clasificación anterior de cada equipo, además de un conjunto de variables que se detallan en la Tabla 4.

Por otra parte, también se ha creando la variable resultado, que será la variable a predecir; guardando dicha variable en una nueva matriz de datos de partidos de la clase *data.frame* de R, se ha denominado “datasetpartidosR” que detallaremos posteriormente.

Nombre	Tipo	Descripción
"equipo"	CHAR	Nombre del equipo
"temporada"	CHAR	Temporada
"jornada"	NUMERIC	Jornada
"puesto"	NUMERIC	Puesto en la clasificación
"puntosT"	NUMERIC	Puntos Totales
"pGanT"	NUMERIC	Partidos Ganados Totales
"pEmpT"	NUMERIC	Partidos Empatados Totales
"pPerT"	NUMERIC	Partidos perdidos Totales
"gFavT"	NUMERIC	Goles marcados Total (a Favor)
"gContT"	NUMERIC	Goles encajados Total (contra)
"puntosL"	NUMERIC	Puntos conseguidos como Local
"pGanL"	NUMERIC	Partidos Ganados jugando como Local
"pEmpL"	NUMERIC	Partidos Empatados jugando como Local
"pPerL"	NUMERIC	Partidos Perdidos jugando como Local
"gFavL"	NUMERIC	Goles marcados jugando como local (a Favor)

"gContL"	NUMERIC	Goles encajados jugando como local (contra)
"puntosV"	NUMERIC	Puntos conseguidos como Visitante
"pGanV"	NUMERIC	Partidos Ganados jugando como Visitante
"pEmpV"	NUMERIC	Partidos Empatados jugando como Visitante
"pPerV"	NUMERIC	Partidos Perdidos jugando como Visitante
"gFavV"	NUMERIC	Goles marcados jugando como Visitante (a Favor)
"gContV"	NUMERIC	Goles encajados jugando como Visitante (contra)

Tabla 4. Variables de la clasifiacaión

Uno de los problemas que se ha tenido en cuenta a la hora de incluir estas variables para ser evaluadas en el modelo de predicción, es la falta de información o que actúen como variables de confusión.

Al principio de cada temporada, todos los equipos parten en igualdad de condiciones, es decir, con los mismo puntos; además, el resto de variables contempladas no son precisas en comparación con jornadas más avanzadas, por ejemplo; cualquier equipo puede meter cuatro goles en la primera jornada de liga, si se incluyera la media de goles por jornada, el valor que se tomaría en la segunda jornada estaría alterando la muestra global; o que un enfrentamiento entre dos equipos descompensados, los evalúe como si estos fueran igualados, o con una diferencia que no corresponde a la realidad, este efecto pasaría en todas las variables para estas primeras jornada.

Existen técnicas para detectar este tipo de observaciones, como son los análisis de residuos de los modelos, que se realiza una vez se tiene el modelo predictivo. Sin embargo sabiendo este dato a priori, es mejor tratar de solventarlo antes de analizar algún modelo.

Una posibilidad sería quitar de la muestra las primeras jornadas de cada temporada para que no influyera en el cálculo de los coeficientes. Esta restricción supondría la pérdida de datos de las jornadas iniciales, con la consiguiente incapacidad para analizar los datos resultantes en partidos que se disputaran en éstas.

La solución adoptada ha sido realizar una extrapolación de los datos de la última jornada de la temporada anterior, a las primeras cinco jornada de la siguiente temporada. Esta extrapolación es sustentada por la premisa de que los equipos en temporadas consecutivas siguen una línea de juego semejante, generalmente, mantienen la misma plantilla de jugadores y las variaciones estarían sujetas a los presupuestos que dispusiera cada club para los nuevos fichajes y a los objetivos ideales a cumplir en la siguiente temporada.

Esta solución no podría ser adoptada con los equipos recién ascendidos, tres en cada temporada. Siguiendo la misma línea anterior, se ha extrapolado los datos del equipo en una posición determinada en la tabla clasificatoria del año anterior. Esta posición será calculada como la media de la posición clasificatoria al final de la temporada, de todos los equipos recién ascendidos en todas las temporadas. Considerando con esta extrapolación, que todos los equipos recién ascendidos encuentran las mismas dificultades: adaptación a la nueva categoría, la gran diferencia de presupuestos con los equipos que llevan varios años en La Liga, entre otras.

Otra de las variables creadas son las rachas, cuantificando los resultados de los últimos cinco encuentros disputados, diferenciando cuando juegan como local, visitante o incluyendo ambos casos. Esta variable trata de cuantificar la proyección y el estado de ánimo del equipo. La forma de ponderar seguirá el siguiente criterio de puntuación: Victoria: +3 puntos, empate: +1 punto, derrota: -3 puntos.

Se ha considerado interesante evaluar, situaciones en las que una derrota o una victoria, suponga, alcanzar o no, uno de los objetivos marcados durante la temporada; alcanzar puestos que permitan disputar una competición europea, salir de puestos de descenso o ser líder de La Liga. Se presupone que el nivel de exigencia y por tanto la efectividad, será superior que otros equipos que tengan los objetivos conseguidos o cuando se encuentran en una posición cómoda durante la temporada. Estos puestos se han delimitado cuando se encuentren entre los seis primeros o los seis últimos de la clasificación, creando una variable categórica binomial (booleana).

Para las variables de la clasificación que indican; goles marcados y partidos ganados, empatados o perdidos, se ha calculado la media por jornadas para incluirlas en el juego de variables del modelo de predicción ("datasetpartidosR").

Una de las técnicas comúnmente usada en modelos de clasificación es la agrupación en variables cuantitativas, esto hace que la devianza disminuya, muchos autores consideran tener una devianza baja como buen indicador de ajuste del modelo a los datos [29]. Esta técnica se ha realizado para la diferencia de puntos y las rachas.

Para dividir en grupos se ha hecho uso de los quartiles de la distribución de la muestra en estas variables, formando cuatro grupos homogéneos, es decir, con el mismo número de observaciones (0-25%,25-50%,50-75%,75-100%).

Estas variables estratificadas fueron probadas y testeadas intentando averiguar cúales se comportarían mejor en nuestro modelo, aunque finalmente no han sido incluidas ya que la finalidad no es realizar un clasificador, sino una predicción de la probabilidad de la variable resultado. Además, otros autores apuntan como una técnica no recomendable en regresiones logísticas [29]. La

interpretación de esta recomendación sería, que para un clasificador, lo que hacemos es un “ajuste grueso” para intentar saber a qué grupo pertenece una variable, y con una regresión logística para saber qué probabilidad tiene de pertenecer a cada grupo. En este proyecto, el objetivo es buscar un ajuste lo más fino posible, por lo tanto descender la devianza del modelo no sería indicador de buen ajuste.

Las nuevas variables creadas para cada equipo: datos de la clasificación, rachas y puestos calientes, formarán el nuevo *dataset* de partidos, que será el punto de partida en el análisis predictivo. La Tabla 5 muestra la estructura final de los datos*:*

Nombre	Tipo	Descripción
"resultado"	FACTOR	Resultado del partido (1X2)
"diferencia"	NUMERIC	Diferencia de puntos entre el local y visitante
"racha_tl"	NUMERIC	Racha total del equipo local
"racha_tv"	NUMERIC	Racha total del equipo visitante
"racha_l"	NUMERIC	Racha total del equipo local jugando de local
"racha_v"	NUMERIC	Racha total del equipo visitante jugando como visitante
"puesto_local"	NUMERIC	Puesto en la clasificación del equipo local
"puesto_vis"	NUMERIC	Puesto en la clasificación del equipo visitante
"med_pGanTlocal"	NUMERIC	Media de partidos ganados total del equipo local
"med_pGanTvis"	NUMERIC	Media de partidos ganados total del equipo visitante
"med_pEmpTlocal"	NUMERIC	Media de partidos empatados total del equipo local
"med_pEmpTvis"	NUMERIC	Media de partidos empatados total del equipo visitante
"med_pPerTlocal"	NUMERIC	Media de partidos perdidos total del equipo local
"med_pPerTvis"	NUMERIC	Media de partidos perdidos total del equipo visitante
"med_pGanLlocal"	NUMERIC	Media de partidos ganados del equipo local como local
"med_pGanVvis"	NUMERIC	Media de partidos ganados total del equipo visitante jugando como visitante
"med_pEmpLlocal"	NUMERIC	Media de partidos empatados total del equipo local como local
"med_pEmpVvis"	NUMERIC	Media de partidos empatados total del equipo visitante jugando como visitante
"med_pPerLlocal"	NUMERIC	Media de partidos perdidos total del equipo local jugando como local

"med_pPerVvis"	NUMERIC	Media de partidos perdidos total del equipo visitante jugando como visitante
"med_gFTlocal"	NUMERIC	Media de goles marcados del equipo local
"med_gFTvis"	NUMERIC	Media de goles marcados del equipo visitante
"med_gCTlocal"	NUMERIC	Media de goles encajados o en contra del equipo local
"med_gCTvis"	NUMERIC	Media de goles encajados o en contra del equipo visitante
"med_gFlocal"	NUMERIC	Media de goles encajados o en contra del equipo local jugando como local
"med_gFvis"	NUMERIC	Media de goles encajados o en contra del equipo visitante jugando como visitante
"med_gClocal"	NUMERIC	Media de goles encajados o en contra del equipo local jugando como local
"med_gCvis"	NUMERIC	Media de goles encajados o en contra del equipo visitante jugando como visitante
"puestos_calientes_l"	BOOLEAN	Indica que el equipo local está entre los 6 primeros o últimos puestos de la clasificación
"puestos_calientes_v"	NUMERIC	Indica que el equipo visitante está entre los 6 primeros o últimos puestos de la clasificación
"diferencia_F"	NUMERIC	Diferencia de puntos factorizado
"racha_tlF"	NUMERIC	Racha total del equipo local factorizada
"racha_tvF"	NUMERIC	Racha total del equipo visitante factorizada
"racha_lF"	NUMERIC	Racha local del equipo local factorizada
"racha_vF"	NUMERIC	Racha visitante del equipo visitante factorizada
"resultado_1"	BOOLEAN	Variable de diseño para el resultado; cuando gana el equipo local es 1 y 0 otro suceso
"resultado_X"	BOOLEAN	Variable de diseño para el resultado; cuando empatan los 2 equipos es 1 y 0 otro suceso
"resultado_2"	BOOLEAN	Variable de diseño para el resultado; cuando gana el equipo visitante es 1 y 0 otro suceso

Tabla 5 Variables del dataset

3.4 Análisis unidimensional

Para tener un mejor conocimiento de las variables que se evalúan en el modelo predictivo, se realiza un análisis descriptivo de estas variables, dependiendo de su naturaleza se tratan con diferentes análisis estadísticos. La

matriz de datos está formada por variables cualitativas y variables cuantitativas, también se presentarán estos análisis por medio de distintas gráficas dependiendo de la clase a la que perten [30].

3.4.1 Variables cualitativas

Lo que interesa ver en este tipo de variables, es la frecuencia de cada categoría y para representarlas se usan diagramas de sectores o diagramas de barras.

La variable “resultado”, es la variable a predecir por el modelo, esta variable comúnmente se suele codificar con la terminología usada en La Quiniela, “1” para la victoria local, “X” cuando el encuentro termina en empate y “2” para la victoria visitante. Como se observa en la Figura 10, la proporción mayor es la victoria local con un 47,84% (1818/3800), seguida por la victoria visitante 28,42% (1080/3800) y un 23,74% (902/3800) en caso de empate.

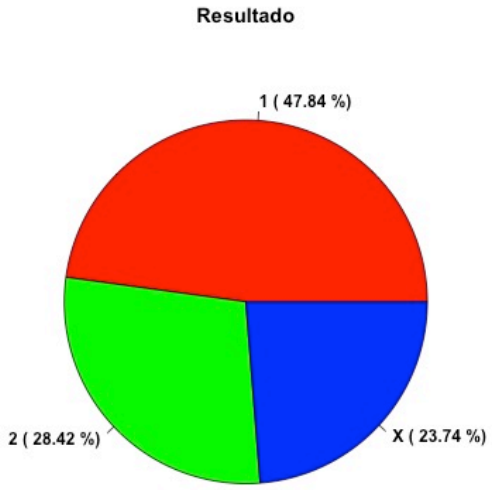


Figura 10 Diagrama de sectores de la frecuencia de resultados

Esta distribución, denota la ventaja que supone jugar en la ciudad del equipo local, ya sea, por el mayor número de aficionados animando al equipo local, conocimiento del estadio, dimensiones, estado del césped, o no tener que desplazarse grandes kilómetros para disputar el partido etc.

En las variables pertenecientes a las “rachas”, que han sido estratificadas como vimos anteriormente, existe una diferencia clara entre las rachas, en las que solo se han evaluado los partidos de los equipos cuando juegan como local y cuando juegan como visitante, reforzando la conclusión de que el equipo local parte con una clara ventaja. En cuanto a las variables de “rachas totales”, que se muestran en la Figura 11, en la que se incluyen cuando juegan como local y como visitante, son muy similares, como sería de esperar ya que los equipos alternan partidos jugando como local y visitante.

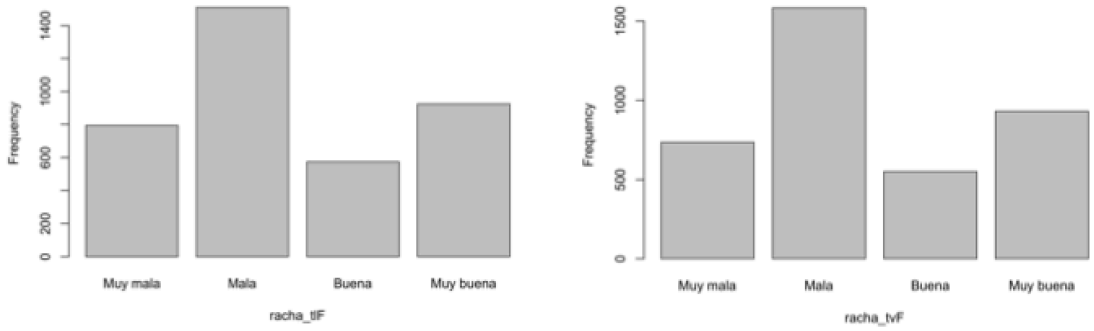


Figura 11.Rachas totales.

Existen otras variables cualitativas como son “puestos_calientes” y “diferencia_F”, que por sí solas no aportan ningún tipo de información y las trataremos en el análisis bidimensional, comparándolas con la variable de interés resultado.

3.4.2 Variables cuantitativas

Para las variables cuantitativas se hace uso de los parámetros estadísticos media, desviación estándar y los quartiles. Todos los resultados de las variables cuantitativas se representan en la Tabla 6.

Variable/Descripción	Descriptivo
Diferencia de puntos entre el equipo local y visitante.(“diferencia”).	Media(DE):-0.411(15.91) Min/Max:-64/60 Mediana (P25;P75): 0(-8;8)
Racha total del equipo local (“racha_tl”).	Media(DE):1.5(8.35) Min/Max: -24/24 Mediana (P25;P75): 1(-4;6)
Racha total del equipo visitante(“racha_tv”).	Media(DE):1.75(8.27) Min/Max:-24/24 Mediana (P25;P75): 61(46;74)
Racha total del equipo local jugando como local (“racha_l”).	Media(DE):2.86(5.21) Min/Max:-12/12 Mediana (P25;P75): 3(0;6)
Racha total del equipo visitante jugando como visitante ("racha_v").	Media(DE):-1.2(5.41) Min/Max: :-12/12 Mediana (P25;P75): -2(-6;2)
Puesto en la clasificación del equipo local ("puesto_local").	Media(DE):10.29(5.82) Min-Max:0-20 Mediana (P25;P75): 10(5;15)
Puesto en la clasificación del equipo visitante ("puesto_visitante”).	Media(DE):10.21(5.73) Min-Max:0-20 Mediana (P25;P75): 10(5;15)
Media de partidos ganados del equipo local ("med_pGanTlocal").	Media(DE):0.373(0.183) Min-Max: 0-1 Mediana (P25;P75): 0.33(0.25;0.47)
Diferencia de puntos entre el equipo local y visitante.(“diferencia”).	Media(DE):-0.411(15.91) Min/Max:-64/60 Mediana (P25;P75): 0(-8;8)
Racha total del equipo local (“racha_tl”).	Media(DE):1.5(8.35) Min/Max: -24/24 Mediana (P25;P75): 1(-4;6)
Racha total del equipo visitante(“racha_tv”).	Media(DE):1.75(8.27) Min/Max:-24/24 Mediana (P25;P75): 61(46;74)
Racha total del equipo local jugando como local (“racha_l”).	Media(DE):2.86(5.21) Min/Max:-12/12 Mediana (P25;P75): 3(0;6)
Racha total del equipo visitante jugando como visitante ("racha_v").	Media(DE):-1.2(5.41) Min/Max: :-12/12 Mediana (P25;P75): -2(-6;2)
Puesto en la clasificación del equipo local ("puesto_local").	Media(DE):10.29(5.82) Min-Max:0-20 Mediana (P25;P75): 10(5;15)
Puesto en la clasificación del equipo visitante ("puesto_visitante”).	Media(DE):10.21(5.73) Min-Max:0-20 Mediana (P25;P75): 10(5;15)
Media de partidos ganados del equipo local ("med_pGanTlocal").	Media(DE):0.373(0.183) Min-Max: 0-1 Mediana (P25;P75): 0.33(0.25;0.47)

Media de partidos ganados del equipo visitante ("med_pGanTvis").	Media(DE):0.378(0.185) Min-Max:0-1 Mediana (P25;P75): 0.34(0.25;0.47)
Media de partidos empatados del equipo local ("med_pEmpTlocal").	Media(DE):0.23(0.11) Min-Max:0-0.833 Mediana (P25;P75): 0.23(0.166;0.3)
Media de partidos empatados del equipo visitante ("med_pEmpTvis").	Media(DE):0.23(0.11) Min-Max:0-0.71 Mediana (P25;P75): 0.23(0.166;0.3)
Media de partidos perdidos del equipo local ("med_pPerTlocal").	Media(DE):0.375(0.15) Min-Max:0-0.90 Mediana (P25;P75): 0.4(0.28;0.48)
Media de partidos perdidos del equipo visitante ("med_pPerTvis").	Media(DE):0.37(0.15) Min-Max:0-0.88 Mediana (P25;P75): 0.39(0.28;0.47)
Media de partidos ganados del equipo local jugando como local ("med_pGanLlocal").	Media(DE):0.46(0.23) Min-Max:0-1 Mediana (P25;P75): 0.44(0.31;0.6)
Media de partidos ganados del equipo visitante jugando como visitante ("med_pGanVvis").	Media(DE):0.28(0.19) Min-Max:0-1 Mediana (P25;P75): 0.25(0.15;0.37)
Media de partidos empatados del equipo local jugando como local ("med_pEmpLlocal").	Media(DE):0.28(0.19) Min-Max:0-1 Mediana (P25;P75): 0.23(0.12;0.33)
Media de partidos empatados del equipo visitante jugando como visitante ("med_pEmpVvis").	Media(DE):0.23(0.15) Min-Max:0-1 Mediana (P25;P75): 0.26(0.15;0.37)
Media de partidos perdidos del equipo local jugando como local ("med_pPerLlocal").	Media(DE):0.23(0.15) Min-Max:0-1 Mediana (P25;P75): 0.23(0.12;0.33)
Media de partidos perdidos del equipo visitante jugando como visitante ("med_pPerVvis").	Media(DE):0.46(0.2) Min-Max:0-1 Mediana (P25;P75): 0.5(0.33;0.60)
Media de goles marcados por el equipo local ("med_gFTlocal").	Media(DE):1.3(0.56) Min-Max:0-3.86 Mediana (P25;P75): 1.2(0.95;1.5)
Media de goles marcados por el equipo visitante ("med_gFTvis").	Media(DE):1.3(0.57) Min-Max:0-4.33 Mediana (P25;P75): 1.2(0.96;1.52)
Media de goles encajados por el equipo local ("med_gCTlocal").	Media(DE):1.32(0.39) Min-Max:0-3.33 Mediana (P25;P75): 1.34(1.1;1.54)
Media de goles encajados por el equipo visitante ("med_gCTvis").	Media(DE):1.3(0.38) Min-Max:0-3.33 Mediana (P25;P75): 1.33(1.09;1.53)
Media de goles marcados por el equipo local jugando como local ("med_gFlocal").	Media(DE):1.53(0.74) Min-Max:0-6 Mediana (P25;P75): 1.36(1.05;1.83)
Media de goles marcados por el equipo visitante jugando como visitante("med_gFvis").	Media(DE):1.1 (0.5) Min-Max:0-4.25 Mediana (P25;P75): 1(0.78;1.33)
Media de goles encajado por el equipo local jugando como local ("med_gClocal").	Media(DE):1.09(0.43) Min-Max:0-3.66 Mediana (P25;P75): 1.09(0.82;1.33)
Media de goles encajados por el equipo visitante jugando como visitante (“med_gCvis").	Media(DE):1.53(0.54) Min-Max:0-4.66 Mediana (P25;P75): 1.53(1.2;1.85)

Media de partidos empatados del equipo visitante ("med_pEmpTvis").	Media(DE):0.23(0.11) Min-Max:0-0.71 Mediana (P25;P75): 0.23(0.166;0.3)
Media de partidos perdidos del equipo local ("med_pPerTlocal").	Media(DE):0.375(0.15) Min-Max:0-0.90 Mediana (P25;P75): 0.4(0.28;0.48)
Media de partidos perdidos del equipo visitante ("med_pPerTvis").	Media(DE):0.37(0.15) Min-Max:0-0.88 Mediana (P25;P75): 0.39(0.28;0.47)
Media de partidos ganados del equipo local jugando como local ("med_pGanLlocal").	Media(DE):0.46(0.23) Min-Max:0-1 Mediana (P25;P75): 0.44(0.31;0.6)
Media de partidos ganados del equipo visitante jugando como visitante ("med_pGanVvis").	Media(DE):0.28(0.19) Min-Max:0-1 Mediana (P25;P75): 0.25(0.15;0.37)
Media de partidos empatados del equipo local jugando como local ("med_pEmpLlocal").	Media(DE):0.28(0.19) Min-Max:0-1 Mediana (P25;P75): 0.23(0.12;0.33)
Media de partidos empatados del equipo visitante jugando como visitante ("med_pEmpVvis").	Media(DE):0.23(0.15) Min-Max:0-1 Mediana (P25;P75): 0.26(0.15;0.37)
Media de partidos perdidos del equipo local jugando como local ("med_pPerLlocal").	Media(DE):0.23(0.15) Min-Max:0-1 Mediana (P25;P75): 0.23(0.12;0.33)
Media de partidos perdidos del equipo visitante jugando como visitante ("med_pPerVvis").	Media(DE):0.46(0.2) Min-Max:0-1 Mediana (P25;P75): 0.5(0.33;0.60)
Media de goles marcados por el equipo local ("med_gFTlocal").	Media(DE):1.3(0.56) Min-Max:0-3.86 Mediana (P25;P75): 1.2(0.95;1.5)
Media de goles marcados por el equipo visitante ("med_gFTvis").	Media(DE):1.3(0.57) Min-Max:0-4.33 Mediana (P25;P75): 1.2(0.96;1.52)
Media de goles encajados por el equipo local ("med_gCTlocal").	Media(DE):1.32(0.39) Min-Max:0-3.33 Mediana (P25;P75): 1.34(1.1;1.54)
Media de goles encajados por el equipo visitante ("med_gCTvis").	Media(DE):1.3(0.38) Min-Max:0-3.33 Mediana (P25;P75): 1.33(1.09;1.53)
Media de goles marcados por el equipo local jugando como local ("med_gFlocal").	Media(DE):1.53(0.74) Min-Max:0-6 Mediana (P25;P75): 1.36(1.05;1.83)
Media de goles marcados por el equipo visitante jugando como visitante("med_gFvis").	Media(DE):1.1 (0.5) Min-Max:0-4.25 Mediana (P25;P75): 1(0.78;1.33)
Media de goles encajado por el equipo local jugando como local ("med_gClocal").	Media(DE):1.09(0.43) Min-Max:0-3.66 Mediana (P25;P75): 1.09(0.82;1.33)
Media de goles encajados por el equipo visitante jugando como visitante (“med_gCvis").	Media(DE):1.53(0.54) Min-Max:0-4.66 Mediana (P25;P75): 1.53(1.2;1.85)

Tabla 6.Resultados variables cuantitativas

Del análisis descriptivo de las variables cuantitativas, destaca una desviación en los valores entre las variables del equipo local y del equipo

visitante, reforzando de esta forma, la ventaja existente por el hecho de jugar en el campo del equipo local.

3.5 Análisis bidimensional

En este apartado, se busca la relación de una variable con respecto a otra, en este caso se hará con la variable dependiente de interés, “resultado”. Además se realizarán los test de significación estadística, cuyo objetivo es demostrar la asociación estadística entre dos variables. El test utilizado dependerá de la naturaleza de las variables a estudiar. Por lo tanto se dividirá en dos apartados; variables cualitativas categóricas y una variable cualitativa con una variable cuantitativa [31].

3.5.1 Variables cualitativas

Respecto a las variables explicativas cualitativas, finalmente, solo se incluirán en el juego de variables finales, las que indican el puesto de interés (“puesto_caliente”), es por ello, que se hará mención especifica a esta variable, en la Figura 12. Se presenta la comparación de estas variables cuando ambos equipos se encuentran en posiciones relevantes o no. Para una mejor compresión de la influencia, se han creado grupos distintos, dependiendo de ambas variables conjuntamente local y visitante.

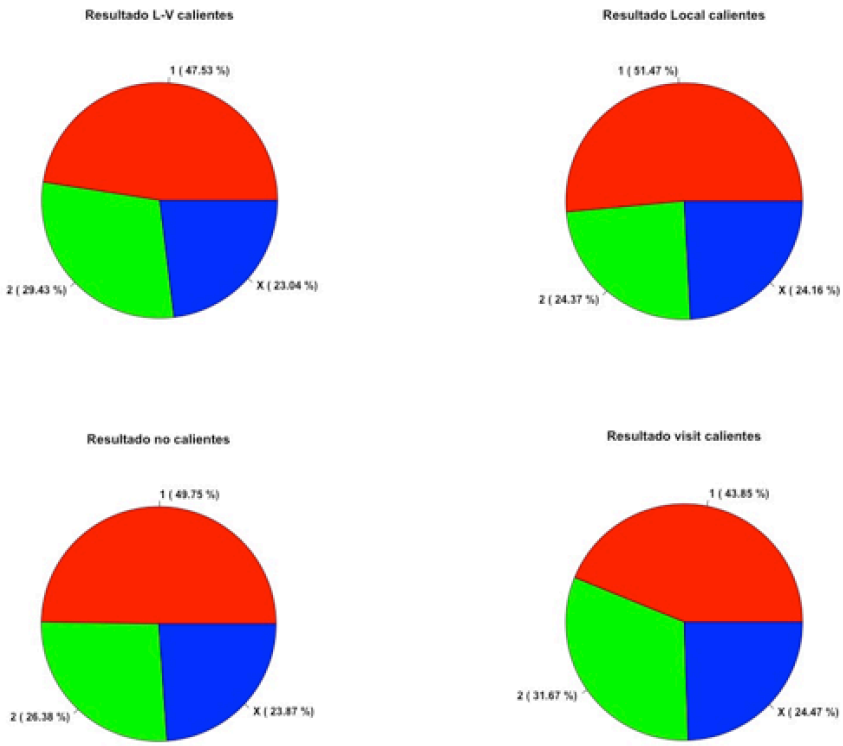


Figura 12 Variable racha local y visitante por grupos

Se observan diferencias notables, cuando un equipo visitante o local (segunda columna), no se encuentra en una situación clasificatoria delicada, también se puede observar en la tabla de contingencia de la Tabla 7.

Para las variables cualitativas, el test de significación usado es el test de independencia Chi-cuadrado, este test devolverá el valor del estadístico Chi-cuadrado de Pearson y su *p-valor* asociado, que es la probabilidad de que las diferencias observadas puedan ser explicadas por azar. Así que cuanto menor sea el valor del *p-valor*, o grado de significación estadística, mayor será la evidencia de que la variable estudiada sea explicativa. Por convenio, se suele considerar un valor menor de 0.05 como estadísticamente significativo.

En la Tabla 7 se muestra los resultados obtenidos de los test.

Nombre variables	Resultado			Chi-cuadrado	p-valor
	1	X	2		
puestos_caliente_l					
➢ 0	563	323	408	15.616	0.0004
➢ 1	1255	579	672		
puestos_caliente_v					
➢ 0	695	320	309	27.815	< 9.1e-07
➢ 1	1123	582	771		
diferencia_F					
Muy inferior	209	174	357	363.3	< 2.2e-16
Inferior	282	189	233		
Igualado	513	295	285		
Superior	363	143	130		
Muy superior	451	101	75		
Racha_tlF					
Muy mala	312	217	264	97.87	< 2.2e-16
Mala	677	378	454		
Buena	261	145	167		
Muy buena	568	162	195		
Racha_tvF					
Muy mala	312	217	264	97.87	< 2.2e-16
Mala	677	378	454		
Buena	261	145	167		
Muy buena	568	162	195		
Racha_lF					
Muy mala	106	66	84	57.693	1.3e-10
Mala	744	448	527		
Buena	478	214	448		
Muy buena	490	174	214		
Racha_vF					
Muy mala	530	225	239	61.76	1.97e-11
Mala	976	495	542		
Buena	221	129	182		
Muy buena	91	53	117		

Tabla 7 Resultado test Chi-cuadrado, Resultado – Variables cualitativas

Todas la variables estudiadas en este apartado, son estadísticamente significativas, y pueden ser incluidas como variables potencialmente explicativas en el modelo de predicción.

3.5.2 Variable cualitativa y variable cuantitativa continua

Existen diferentes test estadísticos de significación, los test paramétricos y no paramétricos. Para determinar cúal usar, se estudiará la normalidad de la distribución de la variable cuantitativa continua, haciendo uso del test de

normalidad de *Shapiro-wilk*. Si el resultado este test es no significativo, se cumple que la variable sigue una distribución normal, por lo que se usará un test paramétrico. En el estudio ninguna variable cumple con la condición de normalidad para usar los test paramétricos.

El test no paramétrico que se usa cuando la variable cuantitativa tiene más de dos categorías es el test de *Kruskal-Wallis*. La Tabla 8 expone los resultados de dicho test, los valores en rojo son las variables que han arrojado valores no significativos y por tanto no serían candidatas a entrar en el modelo de predicción, pero esta labor se hará a través de la selección de variables del sistema de predicción o el usuario según su criterio, ya que a priori no se sabe cómo actuará en conjunción con el resto de variables en el modelo.

Variables	p-valor	Variables	p-valor
diferencia	5.611118e-19	**racha_tl**	1.924814e-15
racha_tv	2.155936e-03	**racha_l**	1.350845e-11
racha_v	1.840432e-02	**puesto_local**	1.017374e-34
puesto_vis	4.442499e-12	**med_pGanTlocal**	1.000304e-08
med_pGanTvis	2.740201e-01	**med_pEmpTlocal**	3.001801e-01
med_pEmpTvis	5.907007e-01	**med_pPerTlocal**	9.324633e-05
med_pPerTvis	5.270655e-02	**med_pGanLlocal**	7.664733e-15
med_pGanVvis	8.837762e-03	**med_pEmpLlocal**	2.380526e-05
med_pEmpVvis	5.961664e-02	**med_pPerLlocal**	2.380526e-05
med_pPerVvis	3.582530e-03	**med_gFTlocal**	1.870185e-06
med_gFTvis	5.759285e-01	**med_gCTlocal**	5.165319e-02
med_gCTvis	4.414697e-01	**med_gFlocal**	2.033617e-07
med_gFvis	8.261697e-01	**med_gClocal**	1.377444e-02
med_gCvis	3.513008e-02		

Tabla 8 P-valor del test Khruskal Wallis, Resultado - Variables cuantitativa continua

Las variables que tienen mayor valor de significación son: el puesto en la clasificación del equipo local, diferencia de puntos entre local y visitante, media de partidos ganados del equipo local cuando juegan en casa (“med_pGanLlocal”), puesto visitante, racha total del equipo local (“racha_l”) y media de goles marcados por el equipo local (“med_gFlocal”).

A continuación representamos mediante gráficas algunas de las variables más significativas de cada grupo detallando la interpretación de éstas.

La Figura 13 representa, cómo varía la densidad de resultados en función de la variable diferencia de puntos entre el local y visitante. Como es de preveer, a mayor diferencia de puntos, la densidad de las victorias locales es superior a las del visitante, e inversamente cuando las diferencias son negativas, mientras que los empates se producen en mayor medida cuando las diferencias de puntos se acercan a cero.

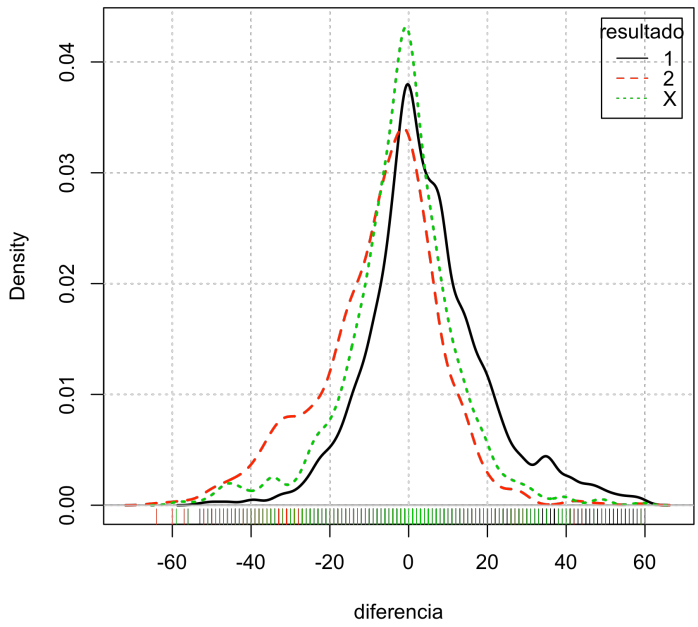


Figura 13. Densidad de la diferencia de puntos por grupos de resultados

En la Figura 14, se muestra, el histograma del puesto local, diferenciado por grupos de resultados. En éste se resalta, que hay mayor número de victorias locales que otros resultados, siendo el empate el que menos observaciones contiene. Además, comparando la victoria local con los otros resultados ,se ratifica la idea, de que cuanto mejor sea el puesto de clasificación del equipo local, mayor será la probabilidad de victoria local, descendiendo la frecuencia de victoria local, cuando el puesto del equipo local aumenta, quedando prácticamente igual la frecuencia entre victoria local y visitante cuando los puestos de clasificación del equipo local se encuentra en las últimas posiciones de la clasificación.

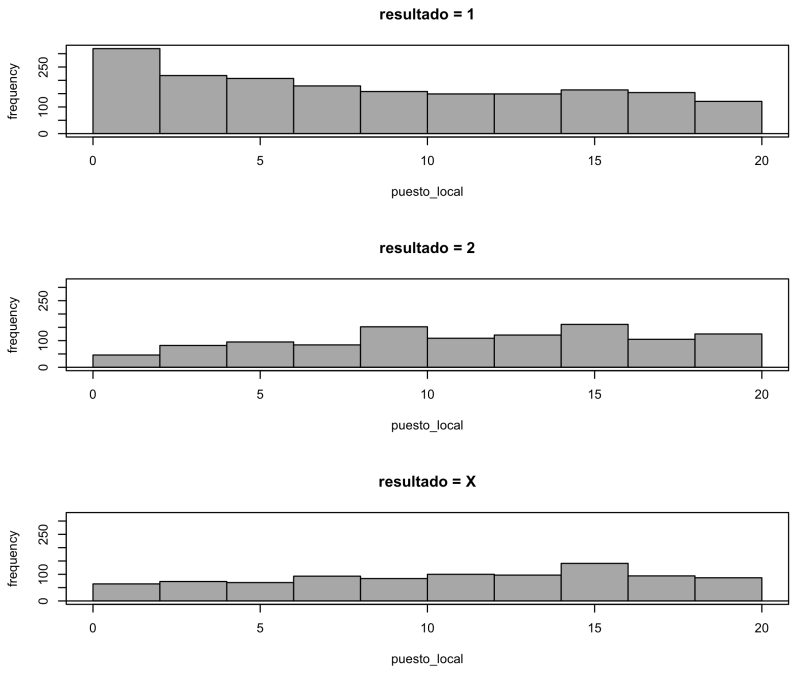


Figura 14 Histograma del puesto local por grupos de resultados

En contraposición, se encuentra el histograma del puesto visitante, Figura 15, siendo muy parecido el histograma del puesto local cuando gana el equipo local, lo que viene a resaltar que la probabilidad de victoria del equipo visitante es más alta entre mejor clasificado esté, pero con frecuencias menores que el del puesto local.

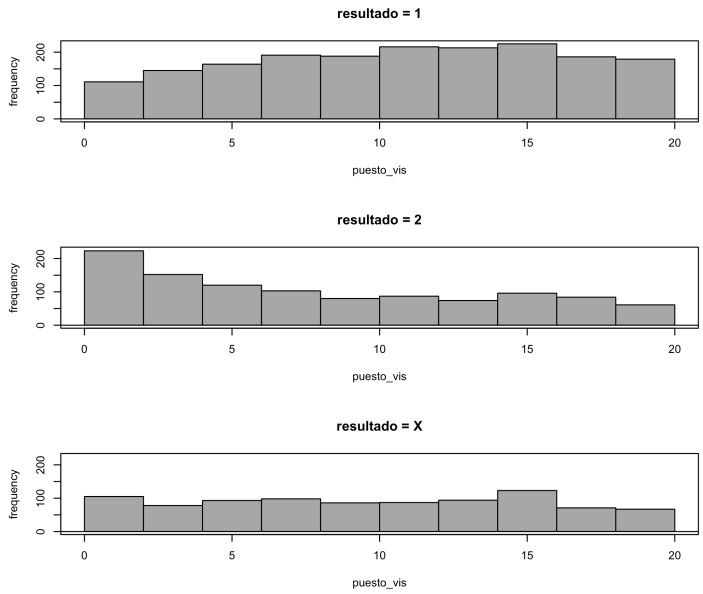


Figura 15 Histograma del puesto visitante por grupos de resultados

La Figura 16, representa la dispersión de la variable goles marcados por el equipo local, esta variable incide también en el resultado, como se puede ver, existe un alargamiento de la nube de observaciones, entre mayor sea la media, más probable será el éxito del equipo local. Para poder hacer la gráfica de dispersión, se codifica la variable resultado a numérica, siendo el tres el valor del empate.

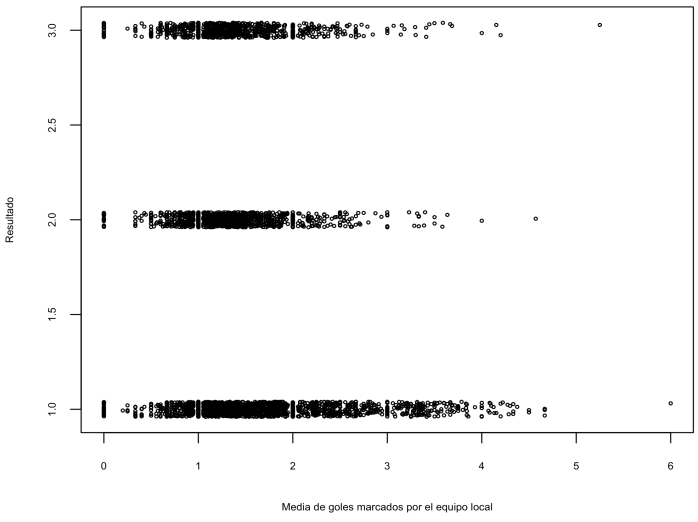


Figura 16 Dispersión de la media de goles marcados equipo local por grupos de resultado

En el diagrama de cajas de la Figura 17 se muestra la variable racha del equipo local, este diagrama de cajas compara las diferentes distribuciones según el resultado, la línea central de la caja que representa la mediana y al igual que sus respectivos primer y tercer quartil, laterales de la caja, está desplazada hacia valores más altos de la racha en el caso de la victoria local con respecto a los otros resultados. Esto significa que entre mejor puntuación en la variable racha mayor será la esperanza de victoria del equipo local.

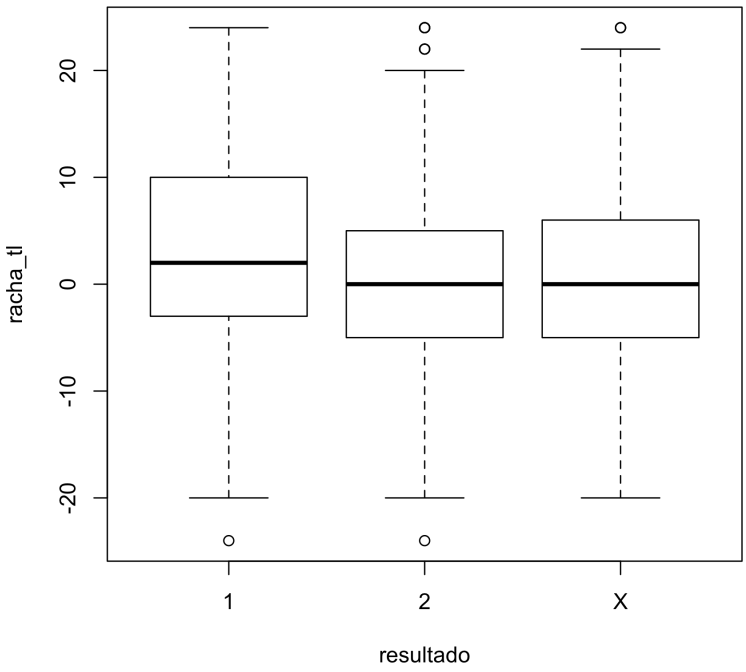


Figura 17 Diagrama de cajas de la variable racha local por grupos de resultados

CAPÍTULO 4: IMPLEMENTACIÓN DEL MODELO PREDICTIVO

4.1 Introducción

En este capítulo, se describe la metodología seguida para la obtención de un modelo predictivo a través de la regresión logística multinomial en el entorno R. El punto de partida, como se ha explicado anteriormente, es el juego de variables explicativas tras el preprocesado y trasformación, el objetivo es predecir la variable dependiente "resultado".

4.2 Formulación y parámetros del modelo.

Existen diferentes librerías que incluyen en su repertorio de modelos predictivos la regresión logística, que es una de las funciones con la que cuenta R, denominada *glm2* [32]. Esta función, crea un modelo logístico binomial, calculando los coeficientes de cada variable que se incluya. Combinando n-1 modelos, se obtendría un modelo logístico multinomial de n categorías de la variable dependiente, siendo esta tarea más laboriosa que si se tratara de una regresión binomial.

Para facilitar la tarea se hará uso de la función multinomial de la librería *nnet* [33], la cual realiza todas estas combinaciones de forma transparente.

En la función se deben incluir los parámetros de entrada necesarios para la construcción del modelo, que se almacenará para poder usarlo, por ejemplo, para predecir nuevas observaciones. Los principales atributos de la función son:

multinom(formula, data, ...)

- La fórmula se construye con la sintaxis respuesta (variable dependiente) ~ predictores (variables explicativas). La respuesta debe ser de tipo factor y las variables predictoras pueden ser variables cuantitativas o cualitativas, si son cualitativas deben ser binomiales y si tuviera más de dos categorías, transformadas con variables de diseño o *dummies*. Los predictores cuando se usan más de dos tienen que ir seguidos del símbolo "+" o "*" si se quiere hacer interacción entre las variables.
- Data, es el dataset o conjunto de variables, que incluyen las variables de la formula.

Además de estos argumentos, existen otros que no son obligatorios, pero con su uso se pueden conseguir diferentes configuraciones para modelar, estos se pueden consultar tanto en la ayuda de R como en el repositorio de *nnet*.

multinom(resultado~diferencia+puesto_local, data=datasetpartidos)

La respuesta de esta función sería el modelo ajustado, sus coeficientes para cada variable predictiva además de algunos parámetros y funciones de interés. En la figura 4.1 se expone un ejemplo de la respuesta.

```
> modelo_prueba<-multinom(resultado~diferencia + racha_l,data=datasetpartidos,trace=FALSE)
> modelo_prueba
Call:
multinom(formula = resultado ~ diferencia + racha_l, data = datasetpartidos,
    trace = FALSE)

Coefficients:
  (Intercept)  diferencia      racha_l
2  -0.5971949 -0.05492071  0.001563469
X  -0.6206768 -0.02999774 -0.016267997

Residual Deviance: 7562.177
AIC: 7574.177
```

Figura 18 Ejemplo de modelo multinomial en R

La *deviance,* es un término que compara el modelo saturado, en el que se incluyen todas las observaciones individuales, con el modelo estimado. Este valor se usará para saber si la inclusión de una nueva variable mejora el modelo, disminuyendo en caso afirmativo. Dicho de otro modo mide la calidad relativa de ajuste del modelo.

El termino AIC, o criterio de información de Akaike, también indica la *deviance,* pero teniendo en cuenta el número de variables incluidas, muy útil para buscar la solución parsimoniosa, que sería aquel modelo que explicara los datos con el mínimo de la variables posibles.

Además, del modelo estimado, se puede obtener: el error estándar de cada coeficiente estimado, los residuos, los valores predichos de todas las observaciones, etc.

4.3 Selección de variables.

Uno de los objetivos fijados, es que el usuario pueda escoger entre las diferentes variables y a través de un algoritmo escoja el mínimo número de variables en el modelo que explique los datos, devolviendo un resultado al usuario posteriormente.

Además de esta solución parsimoniosa, también se devolverá el resultado de otro modelo, que forzará la inclusión de todas las variables seleccionadas. En un principio, para esta segunda respuesta, podría parecer razonable incluir todas las variables sin más, en el modelo *multinom*, pero el orden en el que se incluyen estas variables influye en el cálculo de los coeficientes, por lo tanto, seguiremos el mismo criterio o método que en el modelo parsimonioso.

De los diferentes algoritmos de selección de variable, se ha escogido el método *forward* o hacia delante, en el que se parte de un modelo inicial sin variables predictoras, y se evalúa la inclusión de cada variable por separado; a través de un criterio se selecciona la que mejor se ajuste a los datos.

Los criterios de selección escogidos son: el criterio de información de Akaike y contrastes de razón de verosimilitudes.

Para realizar la función de selección del modelo, partimos de un vector con la selección de las variables que el usuario ha seleccionado y con el uso de dos bucles anidados se irán estimando una a una las variables, una vez seleccionada la variable con el criterio selecto, se vuelven a evaluar todas las variables menos las seleccionadas en las anteriores iteraciones, para mejor compresión y a modo visual del funcionamiento de este algoritmo se han creado dos matrices, donde se irán guardando los valores de los criterios.

En la Tabla 9, se presenta los resultados incluidos en la matriz con el criterio de Akaike con la selección de las diez primeras variables. Las columnas representan la iteración de la inclusión de una variable y las filas las diferentes variables evaluadas en cada iteración. La matriz, inicialmente ha sido inicializada con el número 9999, ya que lo que se busca, es el menor valor en cada iteración. Para ver la selección de la variable en la matriz es fácil, ya que cada variable seleccionada se queda con el valor inicializado en las siguientes iteraciones, porque no se volverá evaluar. La secuencia de variables seleccionadas en este caso sería: diferencia de puntos entre el equipo local y visitante (diferencia), media de goles marcados por el equipo local (med_gFtTlocal), media de goles marcados por el equipo visitante (med_gFTvis), media de partidos empatados por el equipo visitante jugando como visitante (med_pEmpVvis), media de goles encajados por el equipo visitante (med_gCTvis), el equipo visitante se encuentra en posiciones de riesgo (puestos_calientes_v), puesto visitante (puesto_vis), racha_v. A partir de la iteración nueve, el valor de AIC no desciende, por lo tanto, no se debería

incluir más variables para el modelo de mínimas variables, aunque se seguirán incluyendo hasta la última iteración para el modelo forzado por el usuario.

Variable	Valor de AIC en cada iteración									
	1	2	3	4	5	6	7	8	9	10
diferencia	7574.83	9999	9999	9999	9999	9999	9999	9999	9999	9999
racha_tl	7894.23	7569.89	7522.4	7462.69	7459.08	7458.05	7458.09	7453.91	7453.72	7455.07
racha_tv	7888.26	7571.99	7515.32	7463.27	7459.48	7458.33	7456.91	7453.86	7452.69	7454.03
racha_l	7939.3	7574.18	7521.15	7460.77	7457.14	7456.26	7456.67	7452.32	7452.13	7453.77
racha_v	7947.64	7575.96	7525.19	7461.28	7457.13	7455.87	7452.3	7449.92	9999	9999
puesto_local	7833.12	7567.95	7519.29	7461.75	7458.63	7457.85	7457.2	7453.85	7453.63	7455.13
puesto_vis	7846.24	7569.41	7521.96	7461.73	7458.52	7457.79	7450.06	9999	9999	9999
med_pGanTlocal	7775.66	7552.77	7518.96	7463.25	7459.59	7458.4	7457.88	7454.01	7453.86	7455.06
med_pGanTvis	7809.47	7556.99	7492.12	7461.79	7459.23	7458.17	7456.25	7453.98	7453.89	7455.13
med_pEmpTlocal	7962.08	7576.43	7526.29	7463.25	7459.44	7458.23	7457.67	7453.68	7453.54	7454.8
med_pEmpTvis	7956.93	7572.12	7520.16	7459.82	7459.29	7457.98	7457.42	7453.52	7453.41	7454.66
med_pPerTlocal	7853.91	7569.21	7523.09	7461.7	7458.75	7457.83	7456.72	7453.81	7453.67	7454.96
med_pPerTvis	7840.61	7572.1	7515.24	7462.68	7458.23	7457.38	7451.93	7452.36	7452.51	7453.79
med_pGanLlocal	7827.3	7560.27	7523.18	7463.26	7459.59	7458.42	7458.09	7454.03	7453.86	7455.14
med_pGanVvis	7871.32	7561.75	7512	7459.65	7457.87	7456.61	7453.88	7450.74	7453.01	7454.12
med_pEmpLlocal	7955.32	7576.8	7526.05	7462.44	7458.7	7457.68	7457.84	7453.28	7453.15	7454.34
med_pEmpVvis	7987.36	7571.42	7517.99	7455.59	9999	9999	9999	9999	9999	9999
med_pPerLlocal	7955.32	7576.8	7526.05	7462.44	7458.7	7457.68	7457.84	7453.28	7453.15	7454.34
med_pPerVvis	7912.07	7577.66	7523.45	7462.93	7458.93	7457.55	7450.27	7450.59	7452.89	7454.28
med_gFTlocal	7735.81	7522.42	9999	9999	9999	9999	9999	9999	9999	9999
med_gFTvis	7783.45	7549.5	7459.27	9999	9999	9999	9999	9999	9999	9999
med_gCTlocal	7940.15	7578.67	7526.03	7462.95	7458.93	7457.49	7454.11	7451.56	7451.54	7452.69
med_gCTvis	7904.94	7576.13	7521.52	7458.25	7454.67	7454.11	9999	9999	9999	9999
med_gFlocal	7780.1	7534.2	7525.65	7462.33	7458.74	7457.62	7457.36	7453.22	7453.07	7454.3
med_gFvis	7843.31	7552.89	7487.56	7459.96	7457.63	7456.43	7456.2	7452.1	7452.24	7453.51
med_gClocal	7965.42	7578.81	7526.41	7462.1	7458.27	7456.75	7453.97	7451.49	7451.53	7452.45
med_gCvis	7939.24	7577.27	7523.55	7461.34	7457.77	7457.15	7457.06	7452.78	7453.54	7454.79
puestos_calientes_l	7984.85	7571.28	7521.46	7460.33	7456.65	7455.86	7455.12	7451.28	7451.17	9999
puestos_calientes_v	7972.19	7567.78	7515.01	7457.92	7454.42	9999	9999	9999	9999	9999

Tabla 9. Valor AIC selección variables

Con el criterio de contraste de máxima verisimilitud, se compara el modelo anterior, con el modelo incluyendo una variable nueva, esta comparación se hace con el test de *Anova*, el cual nos devuelve un *p_valor* que indica si es significativa la inclusión de la nueva variable. En este caso como se

puede observar en la Tabla 10, escoge exactamente las mismas variables que con el criterio de *Akaike*; se inicializa la matriz con el valor *NULL* o vacío en el leguaje R, con la diferencia que deja de incluir variables desde la iteración cinco, al ser el p_valor mayor de 0.05 (no significación estadística), dejando fuera del modelo, puestos_calientes_v, puesto_vis y racha_v.

Como se ha comprobado, el criterio comparando modelos a través del p-valor es más restrictivo que el de AIC. Por lo tanto, se ha estimado usar el criterio de AIC, por el hecho, de que se busca que el usuario pueda escoger diferentes configuraciones de variables. Además, lo que se pretende con el sistema, es que funcione como una herramienta estadística de fácil uso y específica, pudiendo realizar de forma sencilla la labor de un investigador, probando diferentes configuraciones de variables y obteniendo resultados. Con un criterio muy restrictivo, podría tomar pocas variables o casi ninguna de las seleccionadas dejando poco margen de maniobra al usuario.

Variable	P-valor en cada iteración									
	1	2	3	4	5	6	7	8	9	10
diferencia	0,00E+00	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
racha_tl	0,00E+00	1,15E-02	1,34E-01	7,48E-01	7,73E-01	8,33E-01	9,93E-01	9,28E-01	9,02E-01	9,52E-01
racha_tv	0,00E+00	3,27E-02	3,90E-03	9,98E-01	9,45E-01	9,57E-01	5,50E-01	9,05E-01	5,41E-01	5,65E-01
racha_l	5,22E-14	9,76E-02	7,20E-02	2,87E-01	2,94E-01	3,40E-01	4,88E-01	4,19E-01	4,09E-01	4,97E-01
racha_v	3,38E-12	2,38E-01	5,42E-01	3,70E-01	2,92E-01	2,80E-01	5,49E-02	1,26E-01	NULL	NULL
puesto_local	0,00E+00	4,35E-03	2,83E-02	4,69E-01	6,19E-01	7,53E-01	6,37E-01	8,98E-01	8,65E-01	9,80E-01
puesto_vis	0,00E+00	9,00E-03	1,08E-01	4,64E-01	5,84E-01	7,28E-01	1,79E-02	NULL	NULL	NULL
med_pGanTlocal	0,00E+00	2,19E-06	2,40E-02	9,89E-01	9,96E-01	9,92E-01	8,91E-01	9,74E-01	9,67E-01	9,49E-01
med_pGanTvis	0,00E+00	1,81E-05	3,57E-08	4,78E-01	8,34E-01	8,83E-01	3,96E-01	9,61E-01	9,83E-01	9,81E-01
med_pEmpTlocal	4,61E-09	3,00E-01	9,39E-01	9,91E-01	9,25E-01	9,08E-01	8,06E-01	8,27E-01	8,25E-01	8,30E-01
med_pEmpTvis	3,51E-10	3,49E-02	4,37E-02	1,78E-01	8,57E-01	8,01E-01	7,11E-01	7,62E-01	7,75E-01	7,75E-01
med_pPerTlocal	0,00E+00	8,15E-03	1,90E-01	4,56E-01	6,56E-01	7,46E-01	5,00E-01	8,81E-01	8,84E-01	9,02E-01
med_pPerTvis	0,00E+00	3,46E-02	3,74E-03	7,43E-01	5,07E-01	5,95E-01	4,56E-02	4,27E-01	4,94E-01	5,02E-01
med_pGanLlocal	0,00E+00	9,33E-05	1,98E-01	9,97E-01	9,99E-01	9,99E-01	9,91E-01	9,84E-01	9,71E-01	9,86E-01
med_pGanVvis	0,00E+00	1,96E-04	7,41E-04	1,63E-01	4,23E-01	4,05E-01	1,21E-01	1,90E-01	6,33E-01	5,92E-01
med_pEmpLlocal	1,58E-10	3,63E-01	8,33E-01	6,60E-01	6,40E-01	6,91E-01	8,74E-01	6,76E-01	6,78E-01	6,60E-01
med_pEmpVvis	1,43E-03	2,46E-02	1,48E-02	2,15E-02	NULL	NULL	NULL	NULL	NULL	NULL
med_pPerLlocal	1,58E-10	3,63E-01	8,33E-01	6,60E-01	6,40E-01	6,91E-01	8,74E-01	6,76E-01	6,78E-01	6,60E-01

med_pPerVvis	0,00E+00	5,58E-01	2,27E-01	8,43E-01	7,16E-01	6,47E-01	1,98E-02	1,76E-01	5,98E-01	6,41E-01
med_gFTlocal	0,00E+00	5,62E-13	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
med_gFTvis	0,00E+00	4,27E-07	2,66E-15	NULL	NULL	NULL	NULL	NULL	NULL	NULL
med_gCTlocal	7,98E-14	9,22E-01	8,23E-01	8,54E-01	7,18E-01	6,29E-01	1,36E-01	2,86E-01	3,04E-01	2,90E-01
med_gCTvis	0,00E+00	2,59E-01	8,66E-02	8,12E-02	8,52E-02	1,16E-01	NULL	NULL	NULL	NULL
med_gFlocal	0,00E+00	2,03E-10	6,80E-01	6,25E-01	6,52E-01	6,71E-01	6,88E-01	6,55E-01	6,53E-01	6,48E-01
med_gFvis	0,00E+00	2,33E-06	3,64E-09	1,91E-01	3,75E-01	3,71E-01	3,86E-01	3,76E-01	4,32E-01	4,36E-01
med_gClocal	2,45E-08	9,88E-01	9,97E-01	5,56E-01	5,16E-01	4,34E-01	1,26E-01	2,76E-01	3,02E-01	2,57E-01
med_gCvis	5,06E-14	4,57E-01	2,38E-01	3,81E-01	4,01E-01	5,30E-01	5,93E-01	5,27E-01	8,26E-01	8,27E-01
puestos_calientes_l	4,06E-04	2,29E-02	8,38E-02	2,30E-01	2,30E-01	2,78E-01	2,24E-01	2,49E-01	2,53E-01	NULL
puestos_calientes_v	7,23E-07	3,99E-03	3,34E-03	6,89E-02	7,52E-02	NULL	NULL	NULL	NULL	NULL

Tabla 10. Selección de variables por Test Anova (p_valor)

El resultado de esta función es el cálculo de dos modelos diferentes, el modelo_AIC y el modelo 0 (incluye todas las variables). En la Figura 19, se presentan los diferentes parámetros que devuelve cada modelo; coeficientes, *deviance* y AIC.

```
> modelo_AIC
Call:
multinom(formula = as.formula(expresion), data = datasetpartidos2,
    trace = FALSE)

Coefficients:
  (Intercept)  diferencia med_gFTlocal med_gFTvis med_pEmpVvis puestos_calientes_v[T.1] med_gCTvis puesto_vis     racha_v
2  -0.3707954 -0.02019779   -0.9321807   1.017160   -0.1289728               0.13530498 -0.5778361 0.03338990 -0.01838224
X  -0.2001141 -0.01184820   -0.8335627   0.520416    0.6042533               0.02153301 -0.3971232 0.02818343 -0.01269695

Residual Deviance: 7413.922
AIC: 7449.922
> modelo0
Call:
multinom(formula = as.formula(expresion), data = datasetpartidos2,
    trace = FALSE)

Coefficients:
  (Intercept)  diferencia med_gFTlocal med_gFTvis med_pEmpVvis puestos_calientes_v[T.1] med_gCTvis puesto_vis     racha_v
2  -0.2707978 -0.02049672   -1.4672494  0.7979782    0.1369527               0.13854263 -0.4447380 0.03423569 -0.01791548
X  -0.1780509 -0.01232456   -0.6430008  0.7984043    0.7295100               0.04160219 -0.7082425 0.01299172 -0.01607619
  puestos_calientes_l[T.1] med_gClocal    racha_l  med_gFvis med_pGanVvis med_pPerTvis med_gCTlocal med_pPerTlocal med_pGanTvis  med_gCvis
2              -0.13569665  0.02509441 0.03265325  0.3106575   -0.4789894   -0.2752797    0.3675915      -1.156120    0.5719390 -0.1367272
X              -0.07154344  0.12088261 0.01421596 -0.2363441    0.3156855    1.4194110   -0.3846081       0.673775   -0.6821538  0.1654949
  med_gFlocal    racha_tv med_pEmpLlocal med_pPerVvis med_pGanLlocal     racha_tl med_pGanTlocal puesto_local med_pEmpTlocal med_pPerLlocal
2  0.29055228 0.005534681      -0.493235    0.7131941      -1.280959 -0.012226045     1.06915513 -0.002114975       0.4581228      -0.493235
X -0.07977634 0.009066674      -0.154774   -0.3496282      -0.055036 -0.003147336    -0.09975112  0.003993692       0.1215434      -0.154774
  med_pEmpTvis
2   0.07449813
X  -0.04168988

Residual Deviance: 7385.894
AIC: 7493.894
```

Figura 19 Parámetros de los modelos calculados

4.4 Parámetros e Intervalos de confianza

Otra forma de presentar los parámetros estimados, es a través de las *Odds ratio (OR)*, que son, la razón o cociente entre dos *Odds*, que permite comparar el pronóstico bajo condiciones distintas. La función que devuelve el modelo de regresión logística, también denominado *LOGIT*, es de la forma:

$$Ln\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \alpha + \beta_1 X_1 + \beta_2 X_{21} + \ldots + \beta_n X_n$$

Por lo tanto, aplicando la exponencial a los coeficientes, se obtienen las *Odds ratios* de cada parámetro.

	(Intercept)	diferencia	med_gFTlocal	med_gFTvis	med_pEmpVvis	puestos_calientes_v	med_gCTvis	puesto_vis	racha_v
OR/2	0,690	0,980	0,394	2,765	0,879	1,145	0,561	1,034	0,982
OR/X	0,819	0,988	0,434	1,683	1,830	1,022	0,672	1,029	0,987

Tabla 11. Odds Ratios de las variables modelo AIC

La interpretación de las *Odds ratio,* se debe entender como el aumento de una unidad en un parámetro y asumiendo que los demás parámetros son fijos, indica en qué factor aumenta la probabilidad de que se produzca el suceso, en contraposición del suceso de referencia. En este caso de estudio, el suceso de referencia, es que gane el equipo local, por lo tanto, el aumento en una unidad en la diferencia de puntos manteniendo el resto de variables constante, disminuirá la probabilidad de victoria del equipo visitante frente a la victoria local (odds) en 1,02 veces (1/0.98) y disminuirá la odds de empate en 1,01 (1/0.988) veces. La *OR* en la media de goles marcados por el equipo visitante es de 2,76, lo que significa que un aumento en una unidad en la media de goles marcados por el equipo visitante, aumentaría la *odds* de la victoria visitante 2,76 veces.

Lo interesante, es ver la fuerza de asociación entre las variables, entre más alejado a la unidad, mayor influencia tendrá en las *odds*. El valor de la unidad, sería la ausencia de influencia. Como es lógico, hay que tener en mente que no es una medida totalmente normalizada, ya que no es lo mismo

aumentar una unidad en una media de goles, que aumentar un punto en la diferencia, lo que supondría el aumento del número de goles por jornada al ser una media.

Una medida que siempre debe ir acompañada de las *OR* son sus intervalos de confianza, habitualmente a 2,5% y 97,5%. Lo que se puede interpretar con este valor es que el valor de la *odds ratio* estará entre esas acotaciones con un 95% de probabilidad. Si este intervalo incluye la unidad, límite de aumento o disminución de las *odds*, indica la poca fiabilidad de esta variable a la hora de trasportar el modelo a otra población.

En la Tabla 12, se presentan los valores de los intervalos de confianza resultantes. Algunas variables contienen en su intervalo la unidad, pero los márgenes están inclinados más hacia uno de los intervalos de separación; mayor o menor que uno.

En general, las *OR* no se alejan demasiado de la unidad, esto indica que las probabilidades estimadas no van a cambiar drásticamente por el cambio de una unidad en las variables, y da una idea de lo robusta que son las probabilidades en el fútbol, lo cual era previsible, ya que normalmente las cuotas no se suelen inclinar fuertemente hacia un suceso, oscilando en unos márgenes parecidos.

Variables	2		X	
	2,5 %,	97,5 %,	2,5 %,	97,5 %,
(Intercept)	0,429	1,111	0,509	1,316
diferencia	0,970	0,990	0,978	0,999
med_gFTlocal	0,312	0,497	0,343	0,550
med_gFTvis	2,164	3,534	1,293	2,190
med_pEmpVvis	0,501	1,541	1,042	3,214
puestos_calientes_v	0,957	1,369	0,855	1,222
med_gCTvis	0,407	0,773	0,484	0,933
puesto_vis	1,005	1,064	0,999	1,059
racha_v	0,964	1,000	0,969	1,006

Tabla 12. Odds Ratios Variables

4.5 Bondad de ajuste

La técnica de bondad de ajuste utilizada, se basa en los estadísticos propuestos por *Hosmer Lemeshow*, que estima las diferencias de los valores observados y esperados por intervalos de probabilidad.

Se distinguen dos tipos de validación con este método; validación interna y validación externa. Para la validación interna, se incluye toda la muestra contenida en el *dataset*, tanto para crear el modelo, como para su validación a través del test de *Hosmer Lemeshow*. Para la validación externa, lo que se pretende es comprobar cómo se comporta el modelo, usando datos en la validación que no han sido incluidos en el cálculo del modelo, lo que se hará es dividir la muestra en dos; una se usará para el cálculo del modelo y la otra para el test de validación.

4.5.1 Validacion interna

Los valores esperados, se calculan agrupando las probabilidades predichas de todo los partidos del *dataset* por el modelo, la agrupación se hará a través de los cuartiles, creando de esta forma grupos homogéneos. El número de cortes, habitualmente, se escoge arbitrariamente por encima de un valor mínimo. Este factor es determinante para la decisión de si el modelo se ajusta globalmente a los datos, ya que un número elevado de cortes hará más complicado que el valor esperado se ajuste al observado.

Se ha desarrollado un algoritmo, que busque el mayor número de cortes que expliquen los datos, ya que este valor será usado posteriormente como margen superior admisible para asumir el riesgo de una apuesta. Esto se ha

pensado, porque al agrupar, se estará evaluando de igual forma un partido que se encuentre en el margen superior como en el inferior del intervalo. Si se usase un número de corte pequeño, daría igual si estimáramos con un margen más amplio de error, ya que nuestro sistema los seguiría incluyendo en el mismo grupo a la hora de calibrar.

Lo primero será dividir la muestra en grupos de probabilidad. En la Figura 20, se presenta una división de los valores predichos en diez grupos homogéneos de 1.140 estimaciones. Esta división da una idea de cómo se distribuye la probabilidad predicha, advirtiendo que en los tramos superiores e inferiores incluyen un intervalo de probabilidades mucho mayor que en los grupos centrales. La formación de estos grupos en las colas, crea un problema para delimitar el margen superior de riesgo de las apuestas.

```
> prob.corte2<-cut(v.prob,breaks=quantile(v.prob, probs = seq(0, 1, 1/(9+ncorte))),include.lowest=TRUE)
>       table(prob.corte2)
prob.corte2
[0.0218,0.158]  (0.158,0.208]  (0.208,0.238]  (0.238,0.262]  (0.262,0.285]  (0.285,0.318]  (0.318,0.386]  (0.386,0.468]
          1140           1140           1140           1140           1140           1140           1140           1140
  (0.468,0.58]   (0.58,0.934]
          1140           1140
```

Figura 20 Division de la muestra en intervalos de probabilidad

El siguiente paso es crear una tabla de contingencia, con cada uno de los posibles sucesos, obteniendo de esta forma, los esperados para cada suceso así como en los observados como se muestra en la Figura 21.

```
> expect_1
prob.corte2[1:3800]
[0.0218,0.158]  (0.158,0.208]  (0.208,0.238]  (0.238,0.262]  (0.262,0.285]  (0.285,0.318]  (0.318,0.386]  (0.386,0.468]
      18.39371       15.30470       11.56942       15.23853       19.38926       37.14191      174.59333      370.81651
  (0.468,0.58]   (0.58,0.934]
     501.32225      654.25278
> obs_1
prob.corte2[1:3800]
[0.0218,0.158]  (0.158,0.208]  (0.208,0.238]  (0.238,0.262]  (0.262,0.285]  (0.285,0.318]  (0.318,0.386]  (0.386,0.468]
            18             15             12             21             20             39            180            368
  (0.468,0.58]   (0.58,0.934]
           493            652
```

Figura 21 Observados y esperados por intervalos de probabildad

El cálculo del estadístico de *Hosmer Lemeshow* calcula el estadístico Chi-cuadrado de Pearson de la tabla de observados y esperados.

$$C_g = \sum_{k=1}^{g} \sum_{j=1}^{c-1} \frac{\left(O_{kj} - E_{kj}\right)^2}{E_{kj}}$$

Una vez se tiene el estadístico, podemos evaluar la significatividad a través del *p-valor* teniendo en cuenta los grados de libertad de la muestra, (g-2)*(c-1) siendo g el número de grupos de probabilidades y c el número de sucesos posibles.

En el caso de estudio que hemos ido ejemplificando para un número de cortes igual a diez y tres posibles sucesos, se obtienen 16 grados de libertad, el resultado es 14,12 para C_g y un *p-valor* de 0.5898, estadísticamente no significativo. Aquí la Hipótesis nula es que el modelo se ajusta a la realidad. En un test de bondad de ajuste, siempre en la Hipótesis nula se afirma que el modelo propuesto se ajusta a lo observado. Por lo tanto, este *p-valor* implica que lo que observamos se ajusta suficientemente a lo esperado bajo el modelo.

El resultado del algoritmo para maximizar el número de cortes, fijando como mínimo/máximo 10 y 20 respectivamente, devuelve valores mayores a 0.05, por lo tanto se tiene total libertad a la hora de establecer dicho parámetro.

Este test, además de calcularlo para el modelo parsimonioso, se ha practicado con el modelo en el que se incluyen todas las variables, pudiendo verificar si realmente hay cambios significativos al incluir variables con menos peso dentro del modelo. Como se indica en la Tabla 13, el test afirma que los dos modelos están bien calibrados en cuanto a validación interna se refiere, aunque precisa destacar que el modelo en que se han incluido todas las variables, se ajusta ligeramente mejor a los datos, ya que los valores son más cercanos a la unidad.

Nº cortes	P-valor Forzado	P-valor AIC	Nº cortes	P-valor Forzado	P-valor AIC
10	0,876	0,589	**20**	0,978	0,494
11	0,636	0,339	**21**	0,920	0,396
12	0,730	0,365	**22**	0,426	0,661
13	0,707	0,334	**23**	0,862	0,355
14	0,838	0,416	**24**	0,691	0,252
15	0,553	0,426	**25**	0,816	0,607
16	0,881	0,943	**26**	0,636	0,299
17	0,892	0,674	**27**	0,750	0,499
18	0,739	0,464	**28**	0,753	0,378
19	0,678	0,616	**29**	0,265	0,221

Tabla 13 Resultados del Test Hosmer Lemeshow para diferente número de cortes

4.5.2 Validacion externa

El test de validación externa, sirve para estimar el grado de calibración del modelo con una muestra distinta que con la que se ha estimado el modelo. Esta herramienta se ha diseñado para ser usada calculando probabilidades de eventos futuros desconocidos, así que, esta prueba será la que dictamine si el modelo sirve para la objetivo marcado en el inicio del proyecto.

Siguiendo exactamente el mismo procedimiento que en la validación interna, pero calculando el modelo con la mitad de los datos, y la otra mitad como las nuevas observaciones, se obtiene los diferentes resultados de *p-valor* según el número de cortes.

Nº cortes	P-valor Forzado	P-valor AIC	Nº cortes	P-valor Forzado	P-valor AIC
10	<0,05	0,249	**20**	<0,05	0,432
11	<0,05	0,112	**21**	<0,05	0,417
12	<0,05	0,249	**22**	<0,05	0,361
13	<0,05	0,438	**23**	<0,05	0,351

14	<0,05	0,423	**24**	<0,05	0,575
15	<0,05	0,468	**25**	<0,05	0,070
16	<0,05	0,192	**26**	<0,05	0,675
17	<0,05	0,358	**27**	<0,05	0,602
18	<0,05	0,351	**28**	<0,05	0,484
19	<0,05	0,188	**29**	<0,05	0,391

Tabla 14 Resultados validacion externa Hosmer Lemeshow ,diferente número de cortes

En contraste con los *p-valor* calculados para la validación interna, se obtiene un ajuste bastante inferior en el modelo de mínimas variables. En cuanto al modelo forzado a incluir el resto de variables seleccionadas por el usuario, no se ajusta a los datos cuando se valora una muestra distinta, ya que todos los valores son inferiores a 0,05.

La validación externa, no ha sida incluida en el algoritmo de la herramienta y se ha calculado de forma particular para este apartado, la inclusión destinaría un coste computacional innecesario para la aplicación, éste se hace solo a efectos del estudio de la herramienta, y aunque en cada sesión de usuario se obtenga un nuevo modelo en el cual cabría realizar este test se realiza únicamente la validación interna.

4.6 Tiempo de ejecución

Se han medido los tiempos de ejecución del sistema en global, desde que se recibe una petición hasta que se calcula sus probabilidades, y el cálculo de un modelo ajustado en particular.

Para el medir el tiempo de ejecución se usa la función de R, *time.proc()*, el tiempo medido para el sistema global es de 73,269 segundos cuando se seleccionan todas las variables disponibles, este sistema global estima en cada iteración la inclusión de cada nueva variable en el modelo, por lo tanto, se

contemplan n*(n+1)/2 modelos distintos siendo n, el número de variables seleccionadas. En el caso de estudio, son 28 variables, por lo tanto, se han calculado 378 modelos diferentes, además de realizar tanto la selección a través del test de *Anova* y el cálculo de probabilidades. Además, se ha medido el tiempo de ejecución de cada modelo ajustado en particular.

- Modelo_AIC: 0,102 segundos
- Modelo Forzado: 0,207 segundos

CAPÍTULO 5: ADAPTACIÓN A UN SISTEMA ESCALABLE

5.1 Introducción

En este capítulo, se describe la metodología seguida para la adaptación a un sistema que soporte un aumento de flujo de datos, sin perder eficiencia del sistema predictivo desarrollado. Se detallan los problemas acontecidos a la hora de adaptar la herramienta desarrollada.

5.2 Herramientas disponibles

Para llevar a cabo esta tarea, se han necesitado algunas herramientas especificas, aunque *Hadoop* está disponible en diferentes plataformas, se ha usado *Linux*.

Todas las pruebas realizadas, se han realizado bajo un mismo ordenador personal, haciendo uso de máquinas virtuales, que se han creado con el *software VirutalBox*, con 4G de memoria RAM dedicada a cada máquina virtual. Para las pruebas realizadas, hemos usado el modo programador.

Antes de crear una arquitectura concreta, se han estudiado las diferentes posibilidades que ofrece el ecosistema *Hadoop*, para llevar a cabo tareas de análisis predictivos, concretamente, para regresiones logísticas. Entre las existentes debemos destacar las siguientes:

- ***Mahout*** es un software de *The Apache Software Foundation*, bajo licencia Apache versión 2.0, el objetivo del proyecto, es construir un entorno para crear de manera sencilla aplicaciones de aprendizaje

automático escalables. Entre los algoritmos que vienen integrados en este software, se encuentra la regresión logística, resuelta por gradiente estocástico descendiente (SGD). Aunque la forma de resolver la regresión logística difiere de la usada anteriormente en R, los resultados deben ser prácticamente igual, si se hace una buena configuración del algoritmo. Esta diferencia para calcular la regresión logística, se debe a la naturaleza del sistema distribuido y el cálculo paralelizado, ya que de esta forma se puede calcular de manera individual cada observación, resultando la aportación al cálculo global.

- ***DeltaRho:*** es un proyecto de código abierto con el objetivo de proporcionar métodos y herramientas para análisis de grandes datos, se compone de tres herramientas principales *datadr, Trelliscope, y RHipe*. *Datadr* es un paquete descrito para el entorno R, que facilita la tarea de dividir y recombinar; lo consigue gracias a *RHipe*, que es el motor que realiza las tareas en el entorno *Hadoop*. Entre los ejemplos de la funciones de *datadr,* se encuentra *drGML*, advirtiendo que actualmente solo es una prueba de concepto para realizar regresiones logísticas, usando divide y recombina, para ilustrar las ideas.

- **RHadoop**: es una colección de cinco paquetes de R destinados a que los usuarios puedan realizar operaciones con *Hadoop*. *Rhdfs* es el paquete destinado a la conectividad con el sistema de archivos *HDFS* de *Hadoop,* habilitando de esta forma la lectura, escribir y modificar datos desde una sesión de R. El paquete *Rhbase* proporciona conectividad a R con base de datos distribuidas. *Plyrmr,* permite la manipulación de datos comunes de R en conjunto de datos almacenados en Hadoop. *Rmr2* es un paquete que permite a los desarrolladores de R llevar a cabo análisis estadísticosa través de la funcionalidad *Hadoop MapReduce* en un *cluster*. *Ravro* paquete que permite leer y escribir archivos *avro* y *hdfs* además de añadir un formato *avro* de entrada para *rmr2*.

- **Apache Spark:** es un sistema de computación en *cluster* rápido y de uso general, proporciona API para diferentes entornos como *Java, Scala, Python* y R. Además incluye un conjunto de herramientas de alto nivel *Spark SQL* para *SQL*, procesamiento de datos estructurados *MLlib* para aprendizaje automatizado, *GraphX* para procesamiento de gráficos y *Spark Streaming*

Con cualquiera de las herramientas, se podría realizar un sistema de predicción a través de regresiones logísticas binomiales, pero para regresiones logísticas multinomiales, solamente *Mahout* y *Spark* ofrece la posibilidad de realizarlas directamente, *Mahout,* solo se puede usar en Java, lo que dejaría sin cabida el trabajo realizado anteriormente, lo que hace descartar esta opción.

En un principio, se hicieron múltiples pruebas con *RHadoop,* tanto de configuración, como para realizar la regresión logística multinomial, llegando a la conclusión, que esta herramienta no ha desarrollado las librerías necesarias para realizar un regresión multinomial como la que se requiere, teniendo que desarrollar complejas funciones matemáticas para su implementación, además, esta plataforma es mucho más engorrosa a la hora de instalar y usar que la propuesta por *Spark*, siendo ésta la elegida para la adaptación de la herramienta para su escalabilidad.

5.3 Spark

Apache Spark comenzó como un proyecto de investigación en UC Berkeley , enfocado en el análisis *Big Data*. [32]

El objetivo, fue desarrollar un modelo de programación compatible con una clase de aplicaciones mucho más amplia, que el ofrecido por *MapReduce*, manteniendo la tolerancia automática a fallos. En particular, *MapReduce,* es ineficiente para aplicaciones de múltiples pasos que requieren compartir datos de baja latencia en múltiples operaciones paralelas. Estas aplicaciones son bastante comunes en análisis de datos: algoritmos iterativos (algoritmos de aprendizaje automático o algoritmos gráficos), minería de datos interactiva, donde se desea cargar datos en RAM a través de un *cluster* y consultarlo repetidamente.

Los motores tradicionales *MapReduce* y *DAG,* no son óptimos para estas aplicaciones porque se basan en flujos de datos acíclicos. Una aplicación debe ejecutarse como una serie de trabajos distintos, cada uno de los cuales lee y escribe datos de un almacenamiento estable distribuido, con el coste que ello conlleva al cargar datos en cada paso y escribirlos en almacenamiento replicado.

Spark, ofrece los llamados *Resilient Distributed Datasets* (*RDD*), con los que puede abordar estas aplicaciones de manera eficiente. Los *RDD*s, pueden almacenarse en memoria entre consultas, sin requerir replicación. En caso de fallos, se reconstruyen los datos perdidos, ya que cada *RDD*s recuerda como se crearon a partir de otros conjuntos de datos. Los *RDD*s permiten, que *Spark* supere en rendimiento a los modelos existentes hasta 100 veces en el análisis de múltiples pasos.

Los *RDDs* se presentan como objetos, en los que se puede realizar operaciones a través de sus *APIs*. Transformaciones en la que puede ser una modificación del original o combinaciones con otros *RDD*s o acciones que consiste en aplicar un operación sobre el *RDD* y obtener un resultado.

Apache Spark está en constante desarrollo y se actualiza frecuentemente, además de estar muy bien documentado y estar adquiriendo protagonismo en la comunidad de científicos de datos.

5.3.1 SparkR

Spark proporciona diferentes *APIs* para diferentes entornos de desarrollo *Python, Java, Scala, y R.* *SparkR*. Nos centraremos en esta última para la adaptación de la herramienta, ya que se ha desarrollado en R.

SparkR esta construido con librerías de R, proporcionando un entorno para Apache Spark y así poder usar su sistema de computación. Estas librerías proporcionan soporte para consultas SQL, aprendizaje automático, analítica gráfica. También proporciona soporte para lectura desde diferentes tipos de sistemas de archivos como HDFS, HBase, Cassandra y otros tipos de formatos como JSON.

El principal componente de *SparkR* se centra en sus *data frame* distribuido, que habilita procesamiento de datos estructurado con una sintaxis familiar a R para grandes flujo de datos, a las que podemos aplicar funciones preestablecidas en las librerías MLlib.

En cuanto a la arquitectura, como se muestra en la Figura 22, se observa dos componentes principales, un driver que hace de puente entre R y JVM, este driver se nutre de una nueva clase en R "*jobj*", que referencia a objetos Java existentes, y las máquinas distribuidas que lanzan los procesos de R en *Spark*.

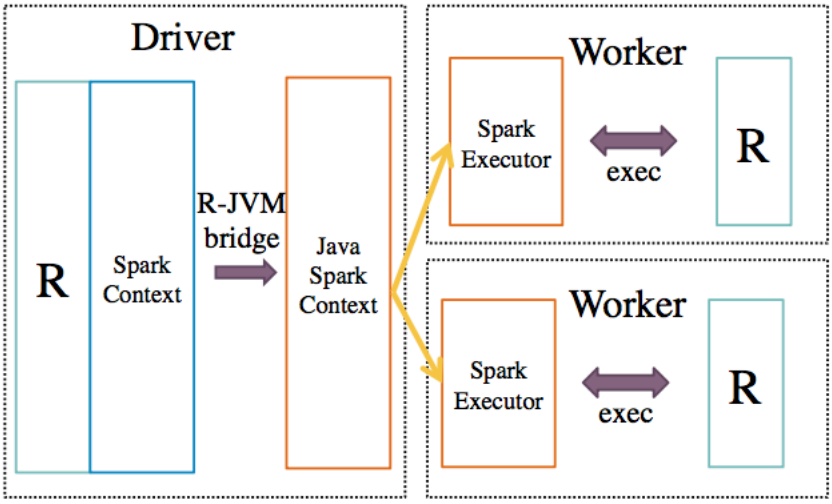


Figura 22 Arquitectura Spark

La librería MLlib de *Spark* incluye entre otros, algoritmos de clasificación y regresión [34] entre los que se encuentra la regresión logística multinomial.

5.4 Adaptación de la herramienta a SparkR

La pauta para la adaptación de la herramienta se compone de los siguientes pasos:

- Instalación y configuración de *Hadoop, Spark 2.3.1, Rstudio y JAVA* en la máquina virtual Ubuntu 16.0.0. [35]
- Guardar el dataset procesado realizado con R y transformar a los *DataFrame* de *Spark*.
- Realizar el modelo predictivo con los *DataFrame* de *Spark*, con los algoritmos de aprendizaje "*logit*" multinomial.
- Comparar los resultados con los obtenidos en el análisis predictivo en R.
- Remplazar el cálculo de los modelos predictivos realizados con R por los de *SparkR* en la herramienta desarrollada.

5.4.1 Modelo predictivo con *SparkR*

Para implementar el modelo predictivo se usa la función *logit* de *SparkR*, para ello, se usará las variables predictivas seleccionadas por el algoritmo desarrollada anteriormente con R, para luego comparar la bodad de ajuste de ambos modelos, para discernir si se ajusta a los datos y si mejora o empeora el desarrollado previamente.

5.4.2 Formulación y parámetros del modelo

Al igual que la función *multinom* usada en R, los parámetros de entrada son exactamente los mismos, incluyendo algunos parámetros propios para el cálculo de los coeficientes a través de la función gradiente descendente, usada por *Spark* [36].

Spark.logit(data,formula,…)

- Data, es el dataset y debe ser un objeto *SparkDataFrame*.
- Fórmula, al igual que en el anterior modelo se compone primero de la variable respuesta seguido

La respuesta de esta función sería el modelo ajustado y sus coeficientes.

Si comparamos el modelo obtenido con Spark, con el que obtenemos en R, es de destacar, que no devuelve términos importantes como *Residual Deviance* o el término AIC (criterio de Akaike), el cual, usábamos en el algoritmo de selección de la variables en el modelo, además de los errores estándar, residuos.

Esta ausencia, supone un punto crítico para poder reemplazar el anterior modelo predictivo, por el realizado con *SparkR*, ya que la herramienta desarrollada, estaba destinada a que el usuario seleccionara las variables, incluyendo el algoritmo las más significativas y descartando las que no mejoraran el modelo.

5.4.3 Bondad de ajuste

Al igual que en el modelo predictivo realizado con R, para realizar la bondad de ajuste, se han utilizado los estadísticos de *Hosmer Lemeshow*.

Como se expuso anteriormente, no podemos seleccionar las variables ni ordenarlas según el peso de éstas, por la ausencia del criterio de *Akaike,* por lo tanto, se ha optado para este apartado tomar las variables seleccionadas por el modelo realizado en R y crear un modelo en *Spark* con estas variables.

Tanto para la validación interna como externa, se necesitan los valores predichos por el modelo, que se obtienen con la función ***predict(model,dataset)*.**

La respuesta de esta función, devuelve la clasificación más probable (1,X,2), pero no los valores de la probabilidad para cada categoría, que es lo que interesa. Esto se debe, a que la respuesta que Spark devuelve a R es un *DataFrame*, en el cual no pueden incluir variables no serializables, en su lugar devuelve la dirección de memoria en el que están alojados los valores. Por lo que se tiene que acceder a cada dirección de memoria y transformar su contenido para su almacenamiento en R y poder realizar las tablas de contingencias requeridas.

Una vez obtenidas las probabilidades predichas, hallamos los *p-valor* para los distintos cortes de ambos modelos; con todas las variables seleccionadas y con el mínimo de variables. En la Tabla 15, se exponen los valores obtenidos para la validación interna, como podemos observar, todos los valores arrojan que el modelo se ajusta a los datos.

Nº cortes	P-valor Forzado	P-valor AIC	Nº cortes	P-valor Forzado	P-valor AIC
10	0,558	0,590	**20**	0,778	0,574
11	0,586	0,339	**21**	0,72	0,416
12	0,773	0,380	**22**	0,664	0,673
13	0,523	0,324	**23**	0,328	0,372
14	0,676	0,379	**24**	0,639	0,402
15	0,631	0,433	**25**	0,822	0,755
16	0,699	0,943	**26**	0,411	0,458
17	0,828	0,693	**27**	0,747	0,795
18	0,478	0,444	**28**	0,785	0,651
19	0,424	0,649	**29**	0,287	0,522

Tabla 15. Test Validacion Interna Hosmer Lemeshow modelo Spark

En comparación con el modelo realizado en R, se obtiene un ajuste bastante similar al modelo parsimonioso, siendo peor este ajuste cuando en el modelo se incluyen todas las variables, aunque teniendo en cuenta que los *p-valor* deben ser superiores a 0,05 para aceptar la hipótesis nula, es más que aceptable para cualquier número de cortes.

En cuanto a la validación externa, al igual que en el modelo predictivo realizado con R, se obtienen unos resultados bastante similares, como se expone en la Tabla 16, en la que cabe resaltar que el modelo que incluye todas las variables no se ajusta a los datos cuando se traslada a una muestra distinta a la usada para calcular el modelo.

Nº cortes	P-valor Forzado	P-valor AIC	Nº cortes	P-valor Forzado	P-valor AIC
11	>0,05	0,232	**21**	>0,05	0,447
12	>0,05	0,117	**22**	>0,05	0,420
13	>0,05	0,243	**23**	>0,05	0,363
14	>0,05	0,478	**24**	>0,05	0,350
15	>0,05	0,402	**25**	>0,05	0,708
16	>0,05	0,443	**26**	>0,05	0,060
17	>0,05	0,176	**27**	>0,05	0,690
18	>0,05	0,326	**28**	>0,05	0,615
19	>0,05	0.323	**29**	>0,05	0,513
20	>0,05	0,191	**30**	>0,05	0,348

Tabla 16. Validacion externa Hosmer Lemeshow

5.4.4 Tiempo de Ejecución

Se ha medido el tiempo de ejecución de las sentencias del modelo ajustado, ya que, al no poder hacer una selección de variables, no se puede medir para la herramienta en global. El tiempo de ejecución para el modelo parsimonioso, es 25,535 segundos y para el modelo con todas las variables, es de 34,447 segundos. Si queremos recoger los resultados de probabilidad, los tiempos medidos aumentarían, 65,535 segundos y 87,127 segundos respectivamente.

En principio, puede parecer que el tiempo de ejecución es demasiado alto, pero, hay que tener en cuenta, que usar este tipo de herramientas de cálculo paralelizado conlleva un coste computacional, que se va compensado entre más núcleos se destinen, además de conjuntos de datos más grandes. En nuestro caso, estamos usando un dataset de apenas 3 MBytes, que es un peso muy distante de las capacidades para las que fueron diseñados estos sistemas, capaces de abordar datasets de Gbytes.

5.4.5 Problemática en la adaptación a la herramienta

Como hemos expuesto anteriormente, hemos logrado un modelo predictivo, que al igual que el diseñado con R, supera tanto la validación interna como externa, pero con las funciones proporcionadas por SparkR, no devuelve los parámetros usados para poder escoger las variables, por lo tanto, no es posible añadir este nuevo módulo a la herramienta diseñada, ya que era uno de los requerimientos principales.

CAPÍTULO 6: SISTEMA GENERAL Y CASO DE USO.

6.1 Introducción

En este capítulo se describe el funcionamiento de la herramienta y, para ello, se detalla el digrama de flujo, las comunicaciones entre modulos, la composición de las tramas de petición/respuesta y el aspecto de la interfaz de usuario. Además se describirá un caso de uso para la herramienta.

6.2 Diagrama de flujo

Como se adelantó en la introducción, la herramienta desarrollada en este proyecto se compone de tres modulos, interfaz de usuario, base de datos y el sistema de predicción, que se comunican para realizar las peticiones que el usuario determine. En la Figura 25 se muestra el diagrama de flujo del sistema de predicción y la interfaz de usuario.

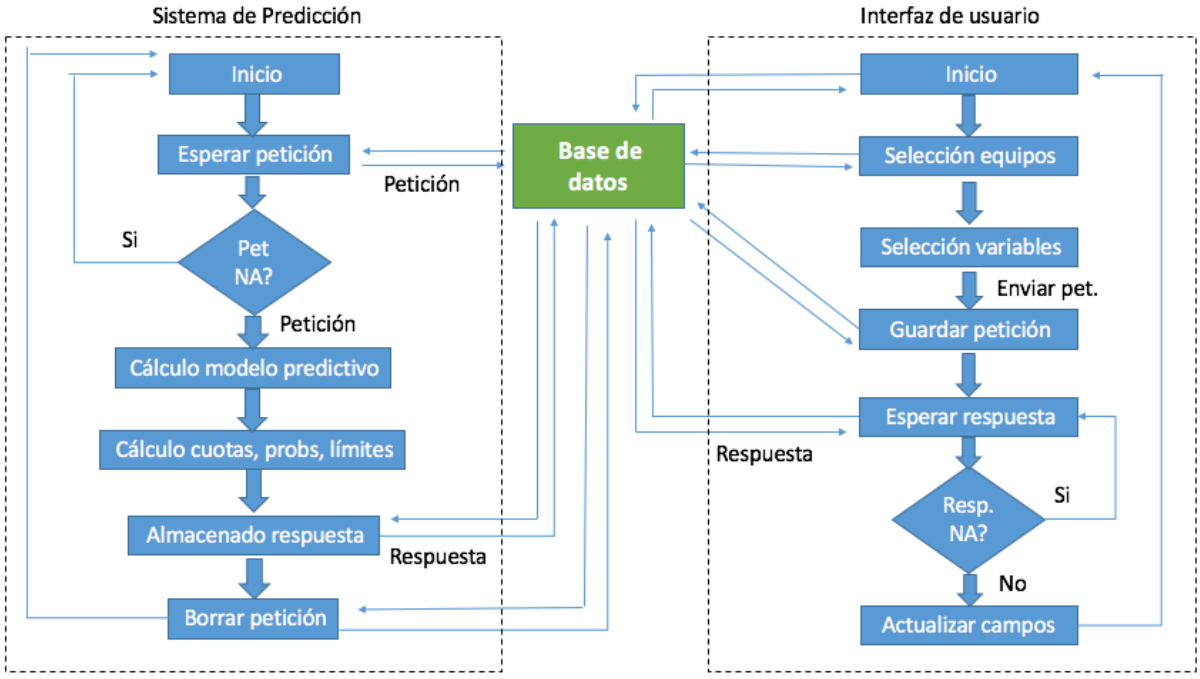


Figura 23. Esquema sistema global

- **Interfaz de Usuario**

En la Figura 26, se ilustra una captura de la interfaz de usuario, incluyendo una numeración de los difentes elementos que la componen (del 1 al 7), que nos sirve de ayuda para entender mejor la descripción siguiente.

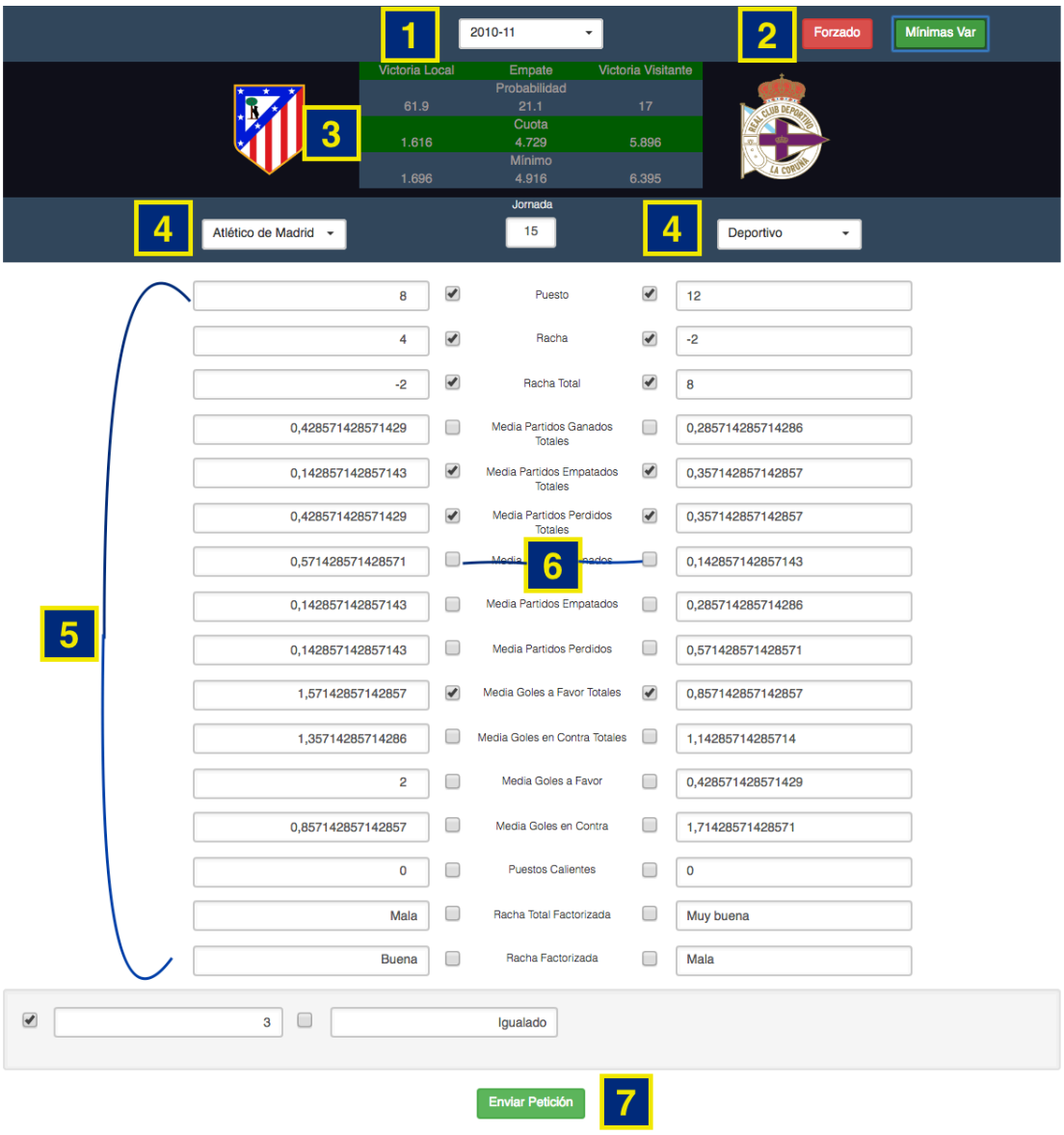


Figura 24.Interfaz de usuario, resaltando diferentes elementos

En un inicio, la interfaz de usuario consulta y guarda los valores de las diferentes temporadas existentes en la base de datos. Una vez se selecciona la temporada (1), se realiza una nueva consulta para cargar los equipos disponibles en esa temporada. Entonces, cuando son

seleccionados los equipos (4), se actualiza el valor de todas las variables disponibles para ese partido, en sus respectivos campos de textos editables (5).

Estos valores se pueden seleccionar y modificar a través de sus cajas de *ticks* (6) y, una vez se tienen modificadas y seleccionadas las variables adecuadamente, se puede ya iniciar el proceso de solicitud con el botón *"enviar petición"* (7).

Lo primero que se hace cuando se envía una petición, es generar un identificador *"id_pet"*, luego se almacenan los valores de las variables seleccionadas, dejando vacías (NULL) el resto de variables descartadas. Resaltar aquí, que la trama que se almacena en la base de datos como petición tiene el mismo formato que la información asociada a los partidos almacenados, además de el identificador de petición como se muestra en la Tabla 13.

Id_pet	id	temporada	nombre_local	nombre_visitante	jornada	resultado	diferencia	var_1	var_2	…	var_n
25	NULL	2010-11	Espanyol	Levante	8	1	NULL	NULL	9	…	val_n

Tabla 17. Estructura petición generada

Una vez almacenada la petición, la interfaz realiza consultas a la base de datos, esperando la respuesta almacenada por el sistema de predicción con el identificar generado. Una vez se obtiene la respuesta, se actualizan los campos de resultados (3), del modelo de minimas variables y del forzado con todas las variables, pudiendo variar la respuesta de cada modelo a través del botón correspondiente, en nuestro caso, de color rojo (2), volviendo finalmente al estado inicial.

- **Sistema de predicción**

En el estado inicial, el sistema hace consultas a la base de datos a la espera de recibir peticiones almacenadas y, una vez llega una petición, se

toma el valor de “Id_pet”, se llama a la función para calcular el modelo predictivo, pasando como variables de entrada la petición. El modelo predictivo recoge dicha petición y verifica qué campos están seleccionados, evaluando solo estas variables seleccionadas como predictores válidos para el cálculo del resultado.

Una vez realizado el análisis predictivo y calculado los dos modelos seleccionados, se hace la predicción del partido con los valores recogidos en la petición, obteniendo las probablidades para dicho partido, se calcula las cuotas y seguidamente se guarda la respuesta en el base de datos. En nuestro caso, la respuesta está formada por los valores de las cuotas, valores de probabilidad y límites calculados para los que se aceptaría realizar una apuesta. En la Tabla 18, se muestra el formato de la primera parte de la respuesta, seguido por las variables para el segundo modelo:

id_pet	couta_1	cuota_2	cuota_X	Probab_1	Probab_2	Probab_X	min_1	min_1	min_1	…
9	15	1.172	22.204	9.845	85.3	4.5	10.2	1.314	82.355	…

Tabla 18. Formato de la respuesta

6.3 Caso de uso

A continuación detallaremos un caso de uso de la herramienta, en el que se describe detalladamente desde el punto de vista del usuario, la secuencia de interacciones para el uso de la herramienta.

En primer lugar, se debe hacer una selección de la temporada. En principio, esta herramienta se ha desarrollado para realizar apuestas de eventos futuros. pero en este proyecto, se ha tomado como muestra de test la mitad de los partidos seleccionables, que corresponden a las cinco temporadas más recientes. Por lo tanto, seleccionaremos una de esas temporadas para mayor semejanza a un futuro uso de la herramienta.

La vista de la interfaz en un inicio, se muestra en la Figura 25 y lo primero que debe seleccionar es una temporada. Para el ejemplo, hemos seleccionado la temporada 2012-13.

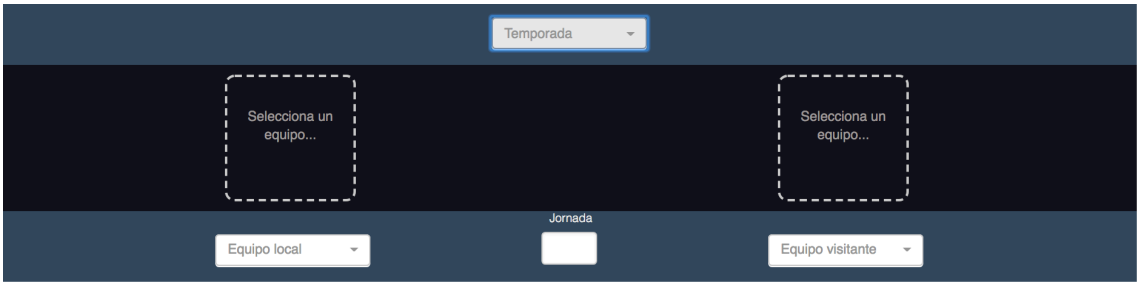


Figura 25. Interfaz inicial

Una vez seleccionada la temporada, se puede seleccionar los equipos disponibles para esa temporada. La primera selección para el primer equipo local es el Athletic de Bilbao, y como equipo visitante se selecciona la Real Sociedad, como muestra la Figura 26.

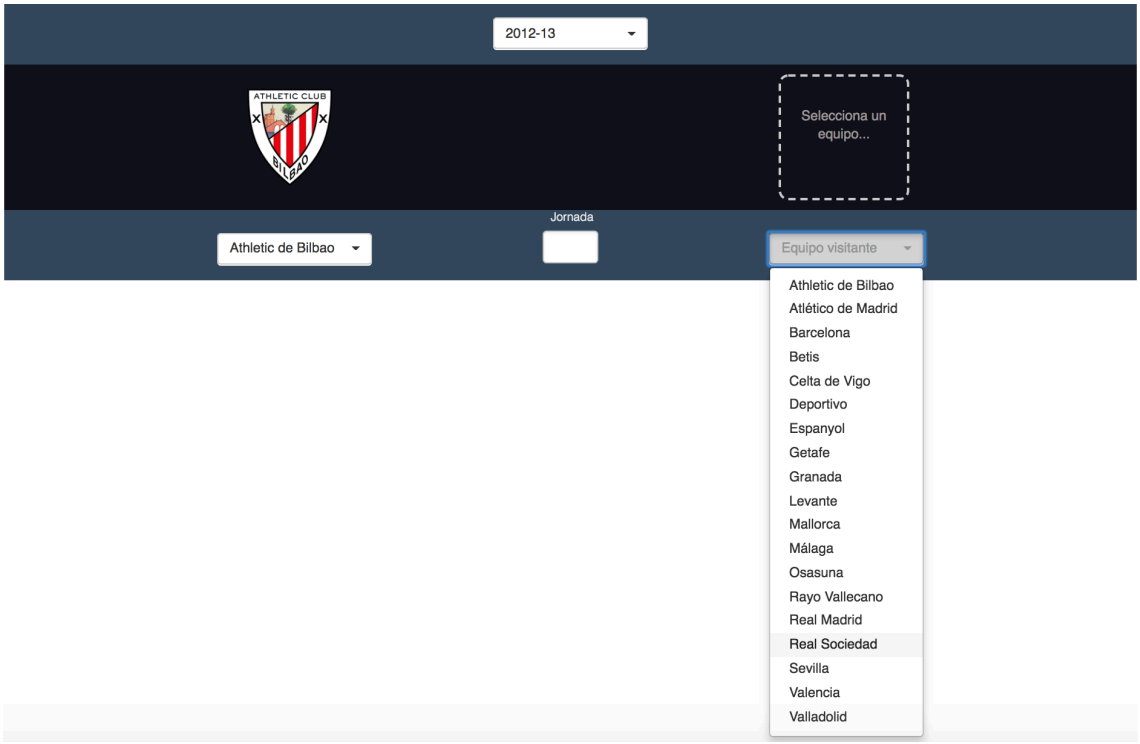


Figura 26. Interfaz de usuario, selección de equipos.

Como veremos en la siguiente Figura 27, una vez se seleccionan los equipos, se cargan los escudos de los respectivos equipos, además de todas las variables seleccionables para dicho partido.

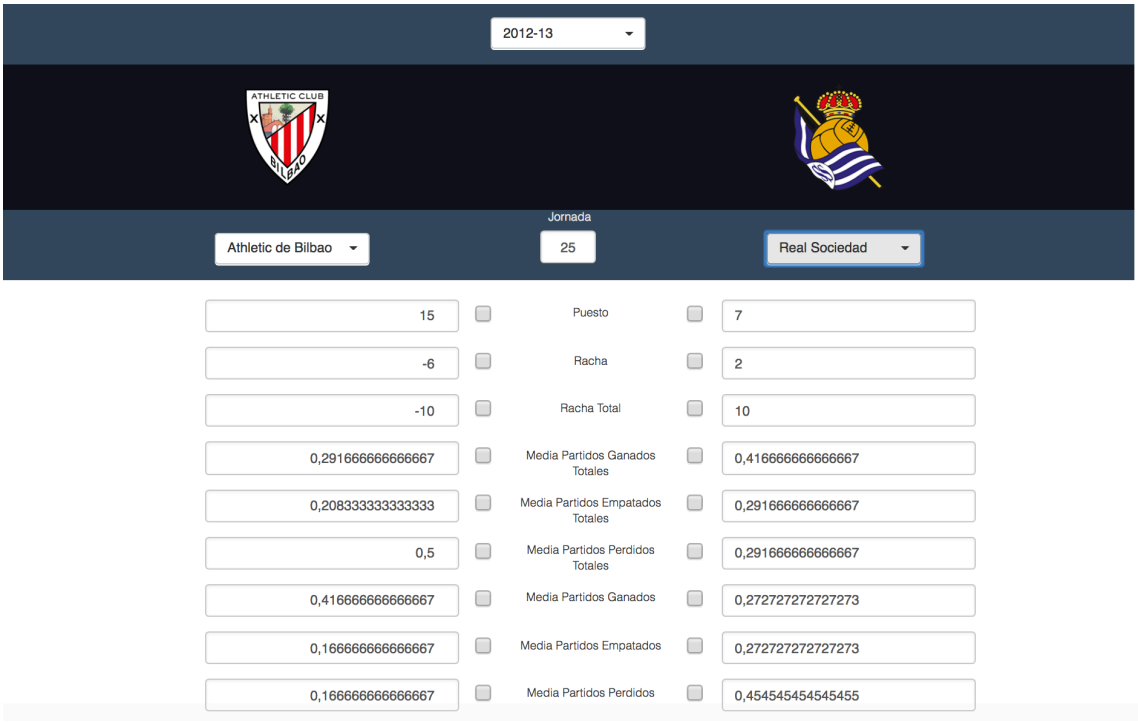


Figura 27. Interfaz de usuario, partido seleccionado.

Teniendo todas los valores de las variables cargadas en sus respectivos campos de textos editables, es el momento en que el usuario debe seleccionar las variables a incluir para que el sistema de predicción las evalúe. En un inicio se seleccionan todas las variables propuestas y realizaremos una petición con el botón “iniciar petición”, quedándose a la espera de la respuesta. Como se muestra en la Figura 28.

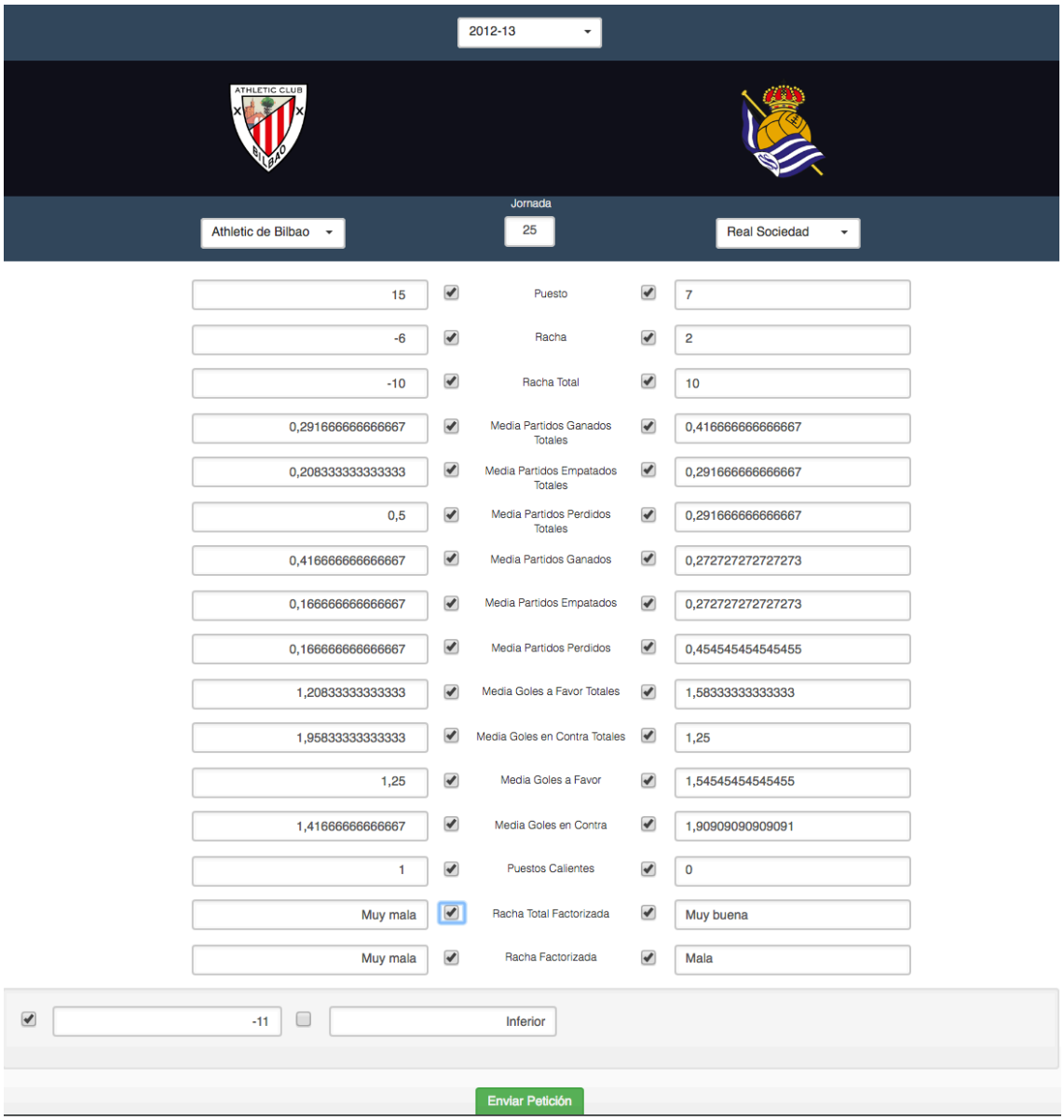


Figura 28. Interfaz de usuario, Todas las variables seleccionadas

Una vez el sistema de predicción haya realizado el cálculo y proporcionado la respuesta, se actualizan los campos relativos al resultado, pudiendo el usuario seleccionar los resultados de uno de los dos modelos. En la figura 29 podemos ver ambos cuadros de resultados.

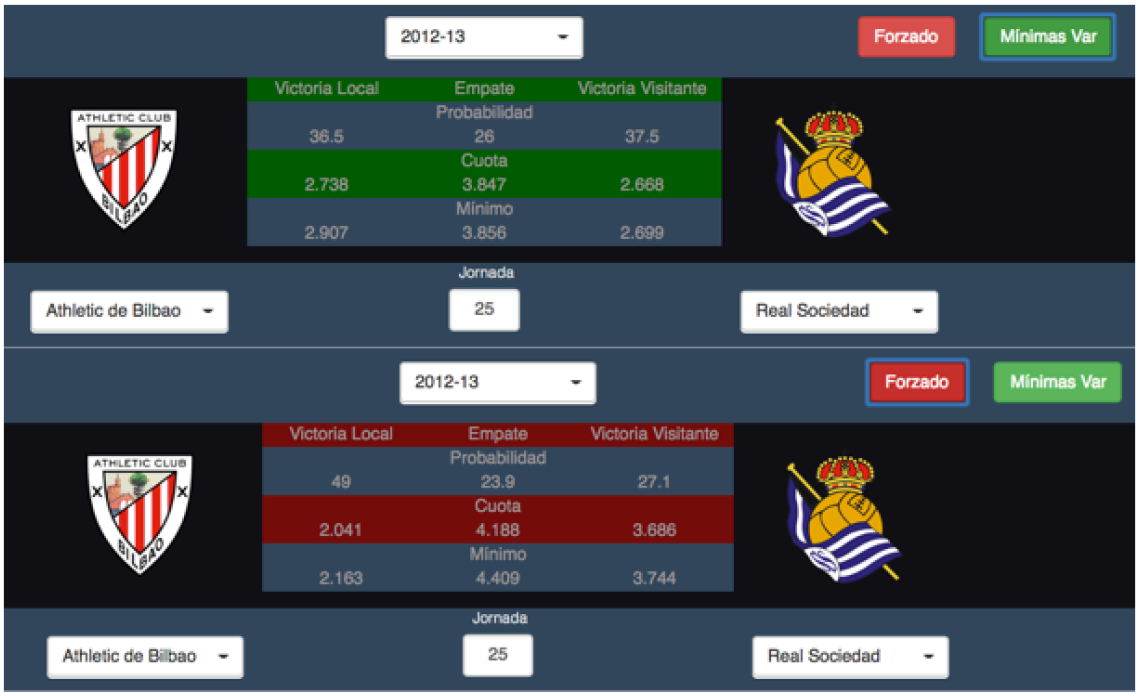


Figura 29.Interfaz de usuario, resultado de los modelos.

La probabilidad del modelo de mínimas variables es: 36,5%, 26% y 37,5% (1,X,2), cuotas: 2,738, 3,847 y 2,668, además de la cuota mínima recomendable: 2,907, 3,856 y 2,699. Para el modelo que incluye todas las variables, la probabilidad es: 49%, 23,9% y 27,1% , las cuotas: 2,041, 4,188 y 3,686, las cuotas mínimas recomendables: 2,163, 4,409 y 3,744. Podemos observar, que hay bastante diferencia entre las cuotas calculadas entre ambos modelos, siendo el de mínimas variables el único modelo que se ha validado externamente, por lo tanto, a priori, el más fiable.

Ahora haremos ciertas hipótesis en el partido elegido, para ver como varían las cuotas, con el propósito de medir la relevancia de estas hipótesis en las probabilidades.

El partido seleccionado, Athetic de Bilbao contra la Real Sociedad es un derbi (equipos de la misma región), en este tipo de partidos los equipos suelen hacer un sobreesfuerzo, dando más importancia a estos partidos, que al resto de partidos de la temporada, incluso a encuentros contra equipos rivales

directos en la clasificación. Es por ello, que las hipótesis en este caso de uso, irán dirigidas a no tener en cuenta las diferencias clasificatorias, no seleccionando el puesto de cada equipo, ni la variable diferencia, ni las rachas. Además variaremos el puesto calitente de la Real Sociedad a uno. Como el equipo visitante es el mejor clasificado, lo que se espera es que la probabilidad de victoria del equipo local aumentara.

En la Figura 30, se muestra los nuevos resultados con la nueva selección y variación de los valores.

Figura 30. Interfaz de usuario, nuevos resultados y selección de variables

Los nuevos resultados obtenidos para esta selección y los primeros resultados obtenidos se muestran en la Tabla 19. Como se observa, los resultados han cambiado, variando las probabilidades a favor del equipo local como habíamos previsto con las hipótesis planteadas. Con estos resultados (diferencias en las probabilidades), ya habriamos valorado la hipótesis planteada en términos de probabilidad, luego comparándolos con las cuotas ofrecidas por los operadores, podríamos discernir con mayor rigor la valoración del riesgo de la apuesta.

SIMULACIONES		RESULTADOS		
		1	X	2
PRIMERA SELECCIÓN	PROBABILIDADES	35,5%	26%	37,5%
	CUOTAS	2,907	3,856	2,699
	CUOTAS RECOMENDADA	2,163	4,409	3,744
SEGUNDA SELECCIÓN	PROBABILIDADES	40,2%	26,3%	33,5%
	CUOTAS	2,485	3,805	2,988
	CUOTAS RECOMENDADA	2,696	3,917	3,102

Tabla 19. Resultados de las simulaciones.

Se podrían seguir haciendo nuevas hipótesis sobre este partido, por ejemplo, variar la media de goles a favor/contra, o la media de partidos ganados, para valorarlas. Dependiendo de las hipótesis planteadas y variando las variables adecuadas, se obtendrían resultados diferentes para cada sesión de usuario.

CAPÍTULO 7: CONCLUSIONES Y LÍNEAS FUTURAS

7.1 Introducción

En este capítulo se describe las conclusiones y se resumen los objetivos alcanzados, además, se proponen posibles líneas futuras de trabajo a seguir en base al trabajo realizado.

7.2 Conclusiones

El objetivo principal de este Proyecto Fin de Carrera, es desarrollar e implementar una herramienta que ayude, en la toma de decisiones del usuario, a la hora de apostar a un partido de fútbol de Primera División Española, pudiendo discernir (a través de un modelo de predicción) si lo que ofrece la casa de apuestas es realmente justo o se ajusta a la probabilidad que el sistema propuesto dictamine.

Se han cumplido los principales objetivos previstos, ya que se ha conseguido desarrollar una herramienta capaz de predecir probabilidades. Se ha elaborado un sistema de predicción que se ajusta a los datos, validado externamente. A través de una interfaz de usuario, se pueden seleccionar diferentes variables para su inclusión y también variar cada valor de las variables explicativas para que el usuario pueda cuantificar en términos de probabilidad las hipótesis que quiera modelar.

Para alcanzar este objetivo, en primer lugar se ha realizado un preprocesado de la muestra de estudio, obteniendo, a partir de un conjunto de variables simples, una muestra con nuevas variables potencialmente

explicativas del resultado, realizando un estudio estadístico pormenorizado de cada variable, obteniendo como resultado si éstas son significativas para el estudio en cuestión.

Seguidamente se ha desarrollado un sistema de predicción basado en regresiones logísticas multinomiales en el lenguaje de programación R, en el que se ha tenido en cuenta las selecciones hechas por el usuario (y alamcenadas en la base de datos) a través de una interfaz de usuario también desarrollada. Este sistema de predicción realiza metódicamente la comparación de distintos modelos para la inclusión de las variables seleccionadas a través del criterio de Akaike, obteniendo de esta forma dos modelos, una con la inclusión de las mínimas variables y un segundo con todas las variables seleccionadas.

A cada modelo se le aplica una validación externa a través de los estadísticos de *Hosmer Lemeshow*, que servirá para acotar el límite de la cuota que sería aceptable asumir con un cierto riesgo para una apuesta concreta. Esta validación resultó, en nuestro ejemplo concreto descrito en la memoria, que el modelo predictivo parsimonioso estaba bien calibrado. Por el contrario, el modelo en que se incluyeron todas las variables resultó que no estaba bien calibrado, con lo que concluimos, que al incluir más variables en el modelo, se aumenta el error estándar o los factores de confusión, demostrando la importancia del criterio de selección. Esto no significa que realizando otras selecciones de variables esta validación resulte siempre negativa para el modelo forzado, es más, en la segunda simulación del caso de uso resulto positiva. Por lo tanto, se ha mantenido en el sistema la realización del modelo forzado, ya que cada sesión puede producir un modelo distinto, con resultados inciertos a priori en cuanto a validación externa se refiere.

También hay que destacar que, en el ejemplo ilustrado en la memoria, donde se seleccionaron todas las variables, además de la validación externa, también se realizó una validación interna que resultó que estaban bien calibrados para ambos modelos.

Por otro lado, se ha desarrollado un sistema para comunicar la interfaz de usuario con el sistema predictivo, usando un servidor SQL. A través de la interfaz de usuario se realizan peticiones que son almacenadas primero, quedando a la espera de la respuesta del sistema predictivo después.

Además, la interfaz de usuario desarrollada ha sido diseñada con el propósito de exponer el funcionamiento de la herramienta de forma local, y ha alcanzado los objetivos marcados.

También, se ha diseñado otro sistema predictivo en el entorno Apache Spark, con el fin de adaptar nuestra herramienta desarrollada a un posible escenario con una muestra de datos mucho mayor. Además aprovechando la nueva implementación, se podrían incorporar nuevas variables potencialmente explicativas, sin que esto conllevara un aumento del coste computacional desorbitado. Finalmente, se crearon los mismos modelos generados por el anterior sistema, es decir, se crearon con las funciones propias de SparkR, los modelos con la selección de variables realizada con el sistema desarrollado en R. Destacar que este se comportó, en cuanto a la validación interna y externa, de forma similar al primer sistema, aceptando por lo tanto que el cálculo paralelizado consigue los mismos resultados válidos para el propósito del proyecto.

En cuanto a la adaptación de la herramienta a este nuevo entorno, decir que no se ha logrado integrar el sistema predictivo, debido a que en los objetivos fijados, el usuario podía hacer una selección de variables diferente en cada sesión y no unas variables prefijadas. En los paquetes de funciones incluidos en *SparkR* y, más concretamente, en la función para la regresión logística multinomial (“*logit”* incluído en librería MLib), se da el caso que no devuelve todos los parámetros necesarios para poder realizar una selección de variables, como la necesaria para realizar dicho objetivo. Hay que destacar que *SparkR* está en pleno desarrollo con constantes actualizaciones de sus

paquetes, la última de estas actualizaciones es de febrero de 2018, y sería de esperar que se incorporen estas funcionalidades en futuros reajustes.

7.3 Líneas futuras de trabajo

Como consecuencia de la experiencia adquirida durante el desarrollo del proyecto, se presentan algunas líneas de trabajo que se podrían realizar en el futuro prosiguiendo el desarrollo iniciado en este trabajo:

- **Desarrollo de un sistema de predicción que funcione en tiempo real para apuestas en directo**. Para el desarrollo de este sistema se debe ampliar el juego de variables de los datos recogidos, de forma que puedan evaluarse nuevas variables explicativas, todas dependientes del tiempo consumido de partido y resultado del mismo en cada momento. Las variables indispensables para este caso podrían ser en qué minuto se produjeron los goles de cada partido, además de otras variables de interés como los tiros a puerta o la posesión de balón.

- **Desarrollo del sistema de predicción con otros algoritmos de aprendizaje**. Por ejemplo, las redes neuronales es otra herramienta oportuna para evaluar variables categóricas y, así, poder comparar su validación, frente al sistema expuesto en este trabajo.

- **Desarrollar un sistema totalmente adaptado a Big Data.** Se propone desarrollar un sistema de predicción en un entorno escalable y, para salvaguardar la problemática encontrada en la adaptación, se propone dejar variables fijas transformando la herramienta descrita aquí en un sistema de predicción final, es decir, como recomendador de apuestas y no

como lo planteado, que es una herramienta para familiarizarse con las probabilidades y las hipótesis de un partido concreto.

- **Mejoras en los módulos propuestos.** Se propone mejorar cada una de las partes del sistema según se indica a continuación:

  - **Mejora de la interfaz de usuario.** Se pueden incluir (a modo de visualización) la tabla clasificatoria, gráficas de cada variable, cuotas ofrecidas por casas de apuestas, resaltado de las variables seleccionadas por el algoritmo, la validación del modelo, etc.

  - **Mejora en el servidor de peticiones.** Desarrollar un sistema capaz de recibir múltiples peticiones de diferentes usuarios para su puesta en funcionamiento en la red.

  - **Estudio y análisis de nuevas variables de predicción**. Se propone para ello un nuevo preprocesado de datos y la creación, a partir del juego de variables existentes, otras nuevas , interacciones entre ellas y la medición de su influencia sobre el resultado.

  - **Cambio de variable a predecir.** En este proyecto solo se ha tenido en cuenta la variable resultado a la hora de predecir, pero existen multitud de apuestas diferentes, por ejemplo, el número de goles totales de un partido o de los equipos individualmente.

Anexo A: Contenido del CD

A.1 Introducción

Junto con esta memoria, se adjunta un CD que recopila el trabajo realizado a lo largo de este proyecto. El contenido de este CD, que es tratado en este anexo, es el siguiente: memoria en formato PDF (*Portable Document Format*), programas y funciones implementadas en R, base de datos usada y la interfaz de usuario desarrollada.

A.2 Descripción del contenido

Al introducir el CD en el ordenador y entrar en su contenido se observarán una serie de carpetas:

- Memoria
- Programas y funciones
- Bases de datos
- Interfaz de usuario

En la carpeta llamada *Memoria*, se encuentra un archivo PDF, el cual, se corresponde con la redacción de la memoria. En cuanto a la carpeta *Programas,* decir que contiene cada una de las funciones utilizadas en este proyecto y realizadas en R. La carpeta *Bases de datos*, contiene la base de datos en formato SQL. Por último, la carpeta interfaz contiene la carpeta con el proyecto HTML para su importación al software MAMP

BIBLIOGRAFÍA

[1] La Sociedad de la Información en España 2016 [Internet]. Recuperado a partir de: https://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/itempubli/558/.

[2] Extending the Value of Your Data Warehousing Investment 2007. Eckerson, Wayne [Internet] a partir de: https://tdwi.org/articles/2007/05/10/predictive-analytics.aspx?sc_lang=en.

[3] Predictive Analytics White Paper, American Institute for Chartered Property Casualty 2007 Nyce, Charles [Internet] a partir de: https://www.the-digital-insurer.com/wp-content/uploads/2013/12/78-Predictive-Modeling-White-Paper.pdf.

[4] "From Data Mining to Knowledge Discovery in Databases" 2008. Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth [Internet] a partir de: https://www.aaai.org/ojs/index.php/aimagazine/article/viewFile/1230/1131.

[5] Hilera, J.R. y Martínez, V.J. "Redes neuronales artificiales: Fundamentos, modelos y aplicaciones". [Internet]. Madrid 1995. Recuperado a partir de: https://www.researchgate.net/publication/31763291_Redes_neuronales_artificiales_fundamentos_modelos_y_aplicaciones_JR_Hilera_Gonzalez_VJ_Martinez_Hernando.

[6] Alfonso Pitarque, Juan Francisco Roy y Juan Carlos Ruiz "Redes neurales vs modelos estadísticos: Simulaciones sobre tareas de predicción y clasificación". [Internet]. 1998. Universitat de València. Recuperado a partir de: https://www.uv.es/revispsi/articulos3.98/pitarque.pdf.

[7] Cisco, Internet será cuatro veces más grande en 2016, Artículo web. [Consulta: 5 noviembre 2015] Disponible en: http://www.cisco.com/web/ES/about/press/2012/2012-05-30-internet-sera-cuatro-veces-mas-grande-en-2016--informe-vini-de-cisco.html.

[8] Hadoop. The Definitive Guide (2012) Tom White. Ed O'Reilly.

[9] Big Data Application Architecture Q&A (2013) Nitin Sawant and Himanshu Shah. Ed Apress.

[10] Aprenda más acerca de Apache Hadoop. [Consulta: 5 noviembre 2015] Disponible en: http://hadoop.apache.org/.

[11] Hadoop in Action (2011) Chuck Lam. Ed Manning.

[12] «BOE» núm. 127, de 28 de mayo de 2011, páginas 52976 a 53022 (47 págs.).

[13] Ordenación de juego online, Informe trimestral. [Consulta: 5 noviembre 2015] Disponible en: https://www.ordenacionjuego.es/es/informes-trimestrales.

[14] Joseph Buchdahl, How To Find A Black Cat In A Coal Cellar, The truth abaut sports tipers, High Stakes Publishing, 2012.

[15] Ismael Chanclón, Así se gana en las apuestas deportivas, Plataforma Editorial, ISBN: 978-84-16429-08-0, 2015.

[16] Fernando Valera Guardiola, David Griol Barres, Sistema de predicción de resultados de eventos deportivos y su aplicación en las apuestas, Universidad Carlos III de Madrid, PFC, 2013.

[17] Hosmer DW LS. Applied logistic regression. Second edition ed. New York: Wiley; 2000.

[18] Pando Fernández V, San Martín Fernández R. Regresión logística multinomial. Cuad Soc Esp Cien For 2004;18.

[19] Agresti A. Categorical Data Analysis. Second Edition ed. New York: Wiley; 2002.

[20] Aguilera del Pino, A. M. Modelos de Respuesta Discreta. Granada: Copias Coca, Dep. Legal GR-11554-02; 2002.

[21] S. Shalev-Shwartz, Online learning: Theory, algorithms, and applications. Technical report, The Hebrew University, PhD thesis, 2007.

[22] Bottou, Léon, Online Algorithms and Stochastic Approximations, Online Learning and Neural Networks. Cambridge University Press, ISBN 978-0-521-65263-6, 1998.

[23] A. Cotter, O. Shamir, N. Srebro, and K. Sridharan. Better mini-batch algorithms via accelerated gradient methods. In NIPS, volume 24, pages 1647–1655, 2011.

[24] Li, Mu, et al. Efficient mini-batch training for stochastic optimization. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2014.

[25] Frederic P. Miller, Agnes F. Vandome, John Mc Brewster, Gradient Descent, Alphascript Publishing, 2010.

[26] Fagerland, M. W., D. W. Hosmer, and A. M. Bofin. 2008. Multinomial goodness-of-fit tests for logistic regression models. Statistics in Medicine 27: 4238–4253.

[27] DBFutbol, [Consulta en: Mayo 2018] Disponible en: https://www.bdfutbol.com/es/index.html.

[28] Database Interface and 'MySQL' Driver for R, [Consulta en: Mayo 2018] Disponible en: https://cran.r-project.org/web/packages/RMySQL/RMySQL.pdf.

[29] Estrategias para la elaboración de modelos estadísticos de regresión Eduardo Nuñez , Ewout W. Steyerberg y Julio Nuñez, Articulo Rev Esp Cardiol. 2011.

[30] Estadistica básica para estudiantes de ciencias, Javier Gorgas García, Nicolás Cardiel López, Jaime Zamorano Calvo. Departamento de Astrofísica y Ciencias de la Atmósfera Facultad de Ciencias Físicas Universidad Complutense de Madrid, 2011.

[31] Test estadísticos [Consulta en: Mayo 2018] Disponible en: https://www.scientific-european-federation-osteopaths.org/los-tests-estadisticos/.

[32] Funcion glm2() [Consulta en: Mayo 2018] Disponible en: https://cran.r-project.org/web/packages/glm2/glm2.pdf.

[33] Librería *nnet* [Consulta en: Mayo 2018] https://cran.r-project.org/web/packages/nnet/nnet.pdf.

[34] Spark 2.3.1 [Consulta en: Mayo 2018] Disponible en: http://spark.apache.org/docs/latest/.

[35] Spark guía instalación [Consulta en: Mayo 2018] Disponible en: https://spark.apache.org/docs/latest/ml-guide.html.

[36] SparkR logit [Consulta en: Mayo 2018] Disponible en: https://spark.apache.org/docs/latest/api/R/spark.logit.html

PLANOS Y PROGRAMAS

PP.1 Introducción

En este apartado se presentan los algoritmos utilizados en el desarrollo de este trabajo.

En la explicación de cada una de las programas o funciones, aparecerán los parámetros de entrada y de salida de las mismas, así como una breve descripción de lo que hace. Los parámetros de entrada, son aquellos que se le deben pasar a la función para obtener un resultado. Los parámetros de salida, son los parámetros que calcula la función a partir de los parámetros de entrada.

Los programas y funciones presentados en este capítulo es el utilizado para realizar los casos expuestos tanto en el caso de uso, como en el ejemplo seguido en este proyecto. Dicho código se encuentra en el CD adjunto con este trabajo.

PP.2 Programa para la creación del *dataset*.

En el archivo *preprocesado.R,* se encuentra el código necesario para realizar el preprocesado de la muestra inicial, para el funcionamiento de este programa es necesario que el servidor MySQL esté iniciado con la base de datos *dbfutbol* con la tabla *partidossql* original. Además, se crean las tablas necesarias en la base de datos (respuesta y petición) necesarias para el sistema de predicción.

Este programa crea las siguientes tablas en la base de datos *dbfutbol*:

- **clasificacionesR**: Clasificación de cada una de las jornadas de todas las temporadas que incluye la base de datos original. Esta clasificación incluye todas las variables expuestas en el Capítulo 3.

- **datasetpartidosR**: Dataframe que incluye las variables creadas para la *ClasifiacionR* de la jornada anterior al encuentro, además de otras variables que se pueden consultar en el Capitulo 3.

PP.3 Programa para el análisis de la variables.

El programa está formado por el código en R necesario para realizar el análisis descriptivo de las variables, tanto unidemensional como relacionando cada variable, con la variable de estudio resultado. También se expone, el código para las diferentes gráficas usadas en el proyecto.

Este programa necesita que se cargue el *datasetpartidosR* en el *data.frame* llamado *datasetpartidos*.

PP.4 Funciones del sistema de predicción.

En el archivo *Sistemapredictivo.R* se encuentra el código en R de las funciones necesarias para realizar el modelo predictivo y las predicciones de las peticiones almacenadas en la base de datos. El programa necesita que se cargue el datasetpartidosR en el Data.Frame llamado *datasetpartidos*. Las funciones que componen este archivo son:

-calcuota()

-calcmodel()

-***escucha()***

La función ***calcuota()*** incluye el Test de *Hosmer Lemeshow* para la validación externa del sistema que es usado para calcular las cuotas límites de riesgo, además de la predicción del partido.

Parámetros de entrada:

- *model* : modelo predictivo realizado con R.
- *peti* petición realizada por el usuario.
- *varselect*: las variables seleccionas por el usuario.

Parámetros de salida:

- *respuesta*: vector formado por la predicción en valor de probabilidad de cada suceso, sus respectivas cuotas y el límite superior del intervalo calculado por el Test de *Hosmer Lemeshow*.

La función ***calcmodel()*** realiza la selección de variables dependiendo de las variables seleccionadas por la petición; calcula dos modelos de predicción para mínimas variables y con todas las seleccionadas, calcula las probabilidades con la función ***calcuota()*** y guarda los resultados en la base de datos.

Parámetros de entrada:

- *peti*: petición realizada por el usuario.

La función ***escucha()*** realiza consultas a la base de datos a la espera de recibir peticiones, una vez obtiene alguna petición llama a la función ***calcmodel()*** con la petición recibida. Además calcula el tiempo de ejecución en responder la petición. Esta función no tiene parámetros ni de entrada ni de salida.

PP.5 Programa del sistema de predicción con *SparkR*.

En el archivo *SparkRprediccion.R* se encuentra el código en R de las sentencias, para realizar el modelo predictivo en este nuevo entorno. El programa necesita que se cargue el *datasetpartidosR* en el *Data Frame* llamado *datasetpartidostotal*.

Este programa realiza un modelo predictivo específico, con las variables que nos resultó seleccionadas con las funciones de R anteriores, además de guardar las predicciones para poder evaluarlas.

PP.6 Interfaz de Usuario

La carpeta *dbfubol* contine todos los archivos que deben ser guardados como nuevo proyecto en el MAMP, dentro de la carpeta *htdocs*.

Es necesario revisar la configuración de la base de datos y el puerto para que funcione todo correctamente.

Una vez importado el proyecto de la interfaz y comprobado su configuración, la url en la que se podria acceder:

http://localhost:8888/dbfutbol/Peticions/add

PP.7 Interfaz de Usuario

Se incluye la base de datos con todas las tablas creadas denominada ***dbfutbol.sql***.

PLIEGO DE CONDICIONES

PL.1 Introducción

En la realización de este proyecto se ha hecho uso de un conjunto de herramientas *software* y equipos *hardware* cuyas características y versiones se listan en los siguientes apartados.

PL.2 Equipos Hardware:

Como elementos hardware se empleó:

- Un Ordenador portátil con procesador Intel® Core™ 2 Duo a 2,7 GHz, 8GB de memoria RAM y 250 GB de disco duro

El sistema mínimo necesario para el funcionamiento de este sistema de reconocimiento facial será un ordenador con las siguientes características: CPU de 1,5 GHz, 1 GB de memoria RAM y espacio suficiente para poder instalarl R, Spark, Hadoop, MAMP, Virtual Box y guardar cada una de las funciones realizadas. Además será necesario disponer de la base de datos.

PL.3 Herramientas software

También se utilizaron las siguientes herramientas software:

- **R 3.3.1 GUI 1.68 Mavericks build (7238):** programa empleado para implementar las funciones que forman el sistema de predicción y para realizar el preprocesado de la muestra.

- **Microsoft Office® 2007:** es el paquete de herramientas, en las que se incluye Microsoft Word y Microsoft PowerPoint, que se han utilizado para la elaboración de la memoria y presentación del proyecto.

- **Ref Works:** Aplicación multi-lingue online diseñada para la gestión de bibliografías y utilizada para tal propósito en la redacción de esta memoria.

- **Mac OSX Yosemite:** es el sistema operativo bajo el que se trabajó en este proyecto.

- **MAMP:** Software que incluye los servidores necesarios para comunicar y los distintos elementos de la herramienta (Interefaz de usuario y Base de Datos). Incluye servidor MySQL y servidor Apache**.**

- **Virtual Box:** Programa para generara máquinas virtuales sobre la que se hicieron las pruebas con Hadoop.

- **UBUNTU 14.04:** Es el sistema operativo bajo el que se trabajo en las máquinas virtuales.

PL.4 Garantía

El autor del sistema lo presenta “AS IS” (tal cual), sin garantía implícita de ningún tipo. Tampoco se responsabiliza de los daños que puedan causar el código presentado o sus archivos derivados a cualquier equipo o del uso que hagan de éstos terceras personas.

PRESUPUESTO

Presupuesto

Don Diego Reyes Santana, autor del presente Proyecto Fin de Carrera, declara que:

El Proyecto Fin de Carrera con título “Sistema de Identificación Biométrica Basado en la Fusión de Rasgos Faciales”, desarrollado en la Escuela de Ingeniería de Telecomunicación y Electrónica, de la Universidad de Las Palmas de Gran Canaria, en el período de un año, tiene un coste de desarrollo total de 47.753,22 € correspondiente a la suma de las cantidades consignadas a los apartados considerados a continuación.

El autor del proyecto
Diego Reyes Santana
Junio de 2018

PR.1 Desglose del Presupuesto

Para la realización del presupuesto se han seguido las recomendaciones del Colegio Oficial de Ingenieros de Telecomunicación (COIT) sobre los baremos orientativos mínimos para trabajos profesionales en 2009. Actualmente, los colegios profesionales ya no pueden establecer baremos de honorarios orientativos ni de ningún otro tipo, según se estableció en el artículo 5 de la Ley 25/2009, de 22 de diciembre, de modificación de diversas leyes para su adaptación a la ley sobre el libre acceso a las actividades de servicio y su ejercicio.

El presupuesto se ha desglosado en varias secciones en las que se han separado los distintos costes asociados al desarrollo del proyecto. Estos costes se dividen en:

- Recursos materiales.
- Trabajo tarifado por tiempo empleado.
- Costes de redacción del proyecto.
- Material fungible.
- Derechos de visado del COIT.
- Gastos de tramitación y envío.
- Aplicación de impuestos.

PR.2 Recursos Materiales

Entre los recursos materiales utilizados para la realización de este proyecto, se incluyen las herramientas software de desarrollo de los algoritmos del sistema, los paquetes software usados para la redacción de la memoria, y el sistema operativo bajo el que se ejecutó el trabajo. Asimismo, se incluyen los equipos hardware usados para dar soporte a estas herramientas.

Se estipula el coste de amortización para un período de 3 años. Para ello, se utilizará un sistema de amortización lineal o constante, en el que se supone que el inmovilizado material se deprecia de forma constante a lo largo de su vida útil. La cuota de amortización anual, se calcula usando la siguiente fórmula:

$$Cuota\ Anual = \frac{Valor\ de\ adquisición - Valor\ residual}{Número\ de\ años\ de\ vida\ útil}$$

El valor residual es el valor teórico que se supone que tendrá el elemento después de su vida útil.

PR.2.1 Recursos software

Las herramientas software utilizadas en presente proyecto fueron:

- R 3.3.1 GUI 1.68 Mavericks build (7238)
- Microsoft Office® 2007
- OSX Yosemite
- Ubuntu 14.04
- Hadoop 3.1.0
- Spark 2.3.0
- R 3.2.2
- MAMP
- Virtual Box

Teniendo en cuenta que la duración del proyecto es de un año y el cálculo del coste de amortización se establece en un período de 3 años, los costes de amortización se calcularán para el primer año. Estos costes se pueden ver en la tabla de costes de las herramientas software.

Descripción	Coste Total	Valor Residual (3 años)	Valor Amortización (1 año)
R 3.3.1	0 €	0 €	0 €
Microsoft Office® 2007	900 €	0 €	116,67 €
OSX YOSEMITE	0 €	0 €	0 €
Ubuntu 14.04	0€	0€	0€
Hadoop 3.1.0	0€	0€	0€
Apache Spark	0€	0€	0€
R 3.2.2	0€	0€	0€
MAMP	0€	0€	0€
Virtual Box	0€	0€	0€
Total			**116,67 €**

Tabla 20. Costes de los recursos software.

Por lo tanto, el coste total del material software libre de impuestos asciende a *ciento dieciséis euros con sesenta y siete céntimos* (**116,67 €**).

PR.2.2 Recursos hardware

Las herramientas hardware en la que se apoya el presente proyecto son:

- Un ordenador portátil con procesador Intel Core 2 Duo a 2,7GHz, 8GB de memoria RAM y 250GB de disco duro.

Aplicando la regla de costes anterior y teniendo en cuenta que, se han utilizado durante todo el año de duración del proyecto, se tiene:

Descripción	Coste Total	Valor Residual (3 años)	Valor Amortización
Ordenador portátil	1.250 €	0 €	450 € (1 año)
Total			**450 €**

Tabla 21. Costes de los recursos hardware.

Finalmente, el coste total del material hardware libre de impuestos asciende a *cuatrocientos cincuenta euros* (**450 €**).

PR.3 Trabajo tarifado por tiempo empleado

Se han invertido 12 meses en las tareas de documentación, diseño y desarrollo necesarias para la elaboración del presente Proyecto Fin de Carrera.

Siguiendo las recomendaciones del COIT, se obtiene una aproximación del importe de las horas empleadas en la realización del proyecto. Los honorarios totales se calculan en base a la siguiente expresión:

$$H = C_t * 74{,}88 * H_n + C_t * 96{,}72 * H_e \text{ €}$$

donde:

H son los honorarios totales por el tiempo dedicado.
H_n son las horas normales trabajadas (dentro de la jornada laboral)
H_e son las horas especiales.
C_t es un factor de corrección función del número de horas trabajadas.

Para la realización de este proyecto han sido necesarias 1440 horas dentro del horario normal:

(6 horas/día · 5 días/semana · 4 semanas/mes ·12 meses)

Según el COIT, el coeficiente C_t tiene un valor variable en función del número de horas empleadas de acuerdo con la siguiente tabla:

Horas empleadas	Factor de corrección C_t
Hasta 36 horas	**1,00**
36 a 72 horas	**0,90**
72 a 108 horas	**0,80**
108 a 144 horas	**0,70**
144 a 180 horas	**0,65**
180 a 360 horas	**0,60**
360 a 540 horas	**0,55**
540 a 720 horas	**0,50**
720 a 1080 horas	**0,45**
Más de 1080 horas	**0,40**

Tabla 22. Factor de corrección en función del número de horas invertidas.

Como se puede observar, el número de horas es superior a 1080, según la tabla P.3, C_t = 0.40 por lo que según la ecuación del importe de horas de trabajo se obtiene una tarifa total por el tiempo empleado de 43.130,88 €.

$$H = 0{,}4 * 74{,}88 * 1440 + 0{,}4 * 96{,}72 * 0 = 43.130{,}88\ €$$

En la siguiente tabla, se desglosa el tiempo de trabajo invertido.

Descripción	Tiempo	Coste/mes	Importe
Formación	1 mes	3.594,24 €	3.594,24 €
Documentación	1 mes	3.594,24 €	3.594,24 €
Desarrollo	9 meses	3.594,24 €	32.348,16 €
Redacción	1 mes	3.594,24 €	3.594,24 €
Total de Costes			**43.130,88 €**

Tabla 23. Desglose del coste por tiempo empleado.

Por lo tanto, los honorarios totales por tiempo dedicado ascienden a cuarenta y tres mil ciento treinta euros con ochenta y ocho céntimos (**43.130,88 €**).

PR.4 Costes de redacción del proyecto

El importe de la redacción del proyecto se calcula de acuerdo a la siguiente expresión:

$$R = 0{,}07 \cdot P \cdot C_h$$

donde:

P es el presupuesto del proyecto.

C_h es el coeficiente de ponderación en función del presupuesto.

En la siguiente Tabla 24, se muestra el presupuesto calculado hasta el momento:

Descripción	Importe
Recursos software	116,67 €
Recursos hardware	450 €
Trabajo Tarifado por tiempo empleado	43.130,88 €
Total	**43.697,55 €**

Tabla 24. Presupuesto de ejecución material.

El presupuesto P calculado hasta el momento asciende a 43.697,55 €. Como el coeficiente de ponderación para presupuestos de más de 42.070,70 €, y menos de 63.106,05 € viene definido por el COIT con un valor de 0,45, el coste derivado de la redacción del proyecto es de:

$$R = 0{,}07 \cdot 45.397{,}55 \cdot 0{,}45 = 1.376{,}47\ €$$

Por tanto, el coste libre de impuestos derivado de la redacción del proyecto es de *mil trescientos setenta y seis euros con cuarenta y siete céntimos* (**1.376,47 €**).

PR.5 Material fungible

Este apartado, engloba los gastos derivados de los materiales utilizados en la realización del proyecto, como son: papel, tóner de impresora, encuadernación y demás material de papelería.

Se detalla en la siguiente tabla:

Descripción	Importe
Papel	20 €
Tóner	95 €
Encuadernación	40 €
Total	155 €

Tabla 25. Costes del material fungible.

Por lo tanto, el coste del material fungible asciende a ciento cincuenta y cinco euros (**155 €**).

PR.6 Derechos de visado del COIT

Los gastos de visado del COIT se tarifan mediante la siguiente expresión:

$$V = 0{,}006 \cdot P \cdot C_v$$

donde:

P es el presuluesto del proyecto.

C_v es el coeficiente reductor en función del presupuesto del proyecto.

El presupuesto P calculado hasta el momento asciende a la suma de los costes de ejecución material, de redacción y de material fungible.

$$P = 43.697{,}55 + 1.376{,}47 + 155 = 45.229{,}02\ €$$

Como el coeficiente C_v para presupuestos de más de 30.050 € y menos de 60.101 €, viene definido por el COIT con un valor de 0,90, el coste de los derechos de visado del proyecto asciende a la cantidad de:

$$V = 0{,}006 \cdot 46{,}982{,}57 \cdot 0{,}90 = 244{,}23\ €$$

Por tanto el coste de los derechos de visado del proyecto asciende a *doscientos cuarenta y cuatro con veintitrés céntimos* (**244,23 €**).

PR.7 Gastos de tramitación y envío

Los gastos de tramitación y envío están fijados en 6,01 €.

PR.8 Aplicación de impuestos

El coste total del proyecto, antes de aplicarle los correspondientes impuestos, asciende 47.242,29 €, a lo que hay que sumarle el 5% de IGIC, con lo que el coste definitivo del proyecto es:

	Coste parcial	Total
Recursos Materiales		566,67 €
Software	116,67 €	
Hardware	450 €	
Coste de Ingeniería		43.130,88 €
Coste de Redacción		1.376,47 €
Material Fungible		155 €
Derechos de Visado		244,23 €
Tramitación y Envío		6,01 €
Subtotal		**45.479,26 €**
Aplicación de impuestos(5% I.G.I.C.)		2.273,96 €
TOTAL		**47.753,22 €**

Tabla 26. Costes totales del proyecto.

El presupuesto total asciende a la cantidad de ***cuarenta y siete mil setecientos cincuenta y tres euros con veintidós céntimos*** (**47.753,22 €**).

El autor del Proyecto.

Diego Reyes Santana