



**UNIVERSIDAD AUTÓNOMA
DEL ESTADO DE MÉXICO**



FACULTAD DE INGENIERÍA

**Análisis predictivo del tiempo de estancia
hospitalaria de pacientes con lesiones por
causa externa utilizando algoritmos de
minería de texto**

Tesis

**Para obtener el grado de
Doctor en Ciencias de la Ingeniería**

PRESENTA:

José Ramón Consuelo Estrada

ASESOR ACADÉMICO:

Dr. Otniel Portillo Rodríguez

ASESORES ADJUNTOS:

Dr. Rigoberto Martínez Méndez

Dr. Jorge Rodríguez Arce

Toluca México, Julio 2018

Resumen

El análisis del tiempo de estancia hospitalaria puede proporcionar información para la administración de recursos hospitalarios. En esta tesis se exploraron distintos métodos de minería de textos y algoritmos de clasificación para realizar un análisis predictivo del tiempo de estancia de pacientes en el servicio de urgencias de un hospital de tercer nivel.

Previa aprobación Ética, se analizaron los datos del servicio de urgencias del Centro Médico “Lic. Adolfo López Mateos” perteneciente al Instituto de Salud del Estado de México ubicado en Toluca, Estado de México. Se cuenta con variables estructuradas (filas y columnas) y de texto libre con un total de 119,220 casos para la comparación de modelos basados en ambos formatos.

El proyecto se dividió en tres etapas. La primera de ellas fue para el reconocimiento de los datos, en ésta se realizó estadística descriptiva y se analizó la integridad y el comportamiento de cada variable. Los resultados mostraron de que el problema conocido como *subregistro por códigos inespecíficos* afecta al 13.30% (n=15,859) del total de registros. En esta etapa también se realizó un análisis de clúster (Silhouette=0.3) encontrando grupos similares a los reportados en la literatura médica.

En la segunda etapa se abordó el problema de códigos inespecíficos y se presentó la clasificación de textos como alternativa de solución. Para validar la propuesta, se compararon cuatro predictores binarios: con datos de texto y representación *Frecuencia de Término - Frecuencia Inversa de Documento* (TFIDF) se construyó un modelo de regresión logística (LR). Para datos estructurados se generó una LR, un árbol de decisión (DT) y un boosting tree (BT). El basado en texto (exa = 0.9393, F_1 -score = 0.9392) superó a los basados en datos estructurados (exa = 0.8117, F_1 -score = 0.8184; exa = 0.8001, F_1 -score = 0.8006; exa = 0.8142, F_1 -score = 0.8181), respectivamente y estimó que los casos afectados por subregistro corresponden al 82.56% de los datos analizados.

En la tercer etapa se planteó la clasificación de textos para la construcción de un modelo predictivo del tiempo de estancia hospitalaria. La variable objetivo se codificó en dos clases *estancia normal* y *estancia prolongada*, resultando en una distribución desbalanceada. En esta sección se propuso texto sintético generado mediante una Red Neuronal Recurrente Long Short-Term Memory (RNN-LSTM) como herramienta de balanceo. Al ser un método inédito, se realizaron pruebas para evaluarlo con algoritmos como LR, DT, BT, Máquinas de Soporte Vectorial (SVM) y Random Forest (RF). Se comparó el F_1 -score contra otras técnicas de balanceo como *sobre muestreo aleatorio* y *Synthetic Minority Over-sampling Technique* (SMOTE) y se realizaron pruebas con dos conjuntos de datos distintos.

Se pudo corroborar que los algoritmos LR ($t=-7.196, 0.000$) y SVM ($t=-6.353, 0.000$) presentan una mejoría estadísticamente significativa cuando se emplea el método propuesto en contraste con los datos desbalanceados originales.

El mejor modelo encontrado para la predicción del tiempo de estancia es el que utiliza SVM, el método de balanceo por notas sintéticas y notas médicas originales. Este modelo cuenta con 107370 coeficientes (107369 variables), dos clases y logró un valor de exactitud de 0.86 y de F_1 -score de 0.68.

Palabras clave: Minería de Textos, Tiempo de Estancia, Subregistro por códigos inespecíficos, Balanceo de Clases, Redes Neuronales Recurrentes, Machine Learning.

Contenido

1. Introducción	1
1.1. Conceptos básicos	2
1.1.1. Tiempo de estancia hospitalaria	2
1.1.2. Lesiones por causa externa	2
1.1.3. Clasificación Internacional de Enfermedades (CIE-10)	3
1.2. Estado del arte	4
1.3. Planteamiento del problema	7
1.4. Hipótesis	7
1.5. Objetivos	7
1.5.1. Objetivo general	7
1.5.2. Objetivos específicos	7
1.6. Alcances y limitaciones	8
1.7. Contenido	8
2. Marco Teórico	11
2.1. Aprendizaje Automático	11
2.1.1. Elección de predictores	12
2.1.2. Sobre ajuste del modelo	12
2.1.3. Prueba y validación	14
2.1.4. Medidas de rendimiento	14
2.1.5. Descenso de gradiente	16
2.2. Análisis de textos	17
2.2.1. Representación del documento	19
2.3. Algoritmos de clasificación	19
2.3.1. Regresión	20
2.3.2. Árboles de decisión	21
2.3.3. Redes Neuronales Artificiales	23
2.3.4. Métodos combinados (Ensemble Methods)	26
2.4. Algoritmos de clusterización	27
2.4.1. Two-step clúster	28
2.4.2. k-Means	29
2.5. Técnicas de balanceo de clases	30
3. Material y métodos	31
3.1. Caracterización de lesiones por causa externa	31
3.2. Análisis de subregistro por códigos inespecíficos	32

3.3. Elección de modelo y balanceo de clases	34
3.3.1. Generación de texto sintético	35
3.3.2. Evaluación de los modelos	38
3.3.3. Elección del modelo	41
4. Resultados	43
4.1. Caracterización de Lesiones por Causa Externa	43
4.2. Subregistro por códigos inespecíficos	50
4.3. Elección de modelo y balanceo de clases	56
4.3.1. Generación de texto sintético	56
4.3.2. Comparación entre métodos de balanceo	57
4.3.3. Evaluación con otros conjuntos de datos	60
5. Discusión, conclusiones y trabajo futuro	65
5.1. Discusión de resultados	65
5.1.1. Caracterización de lesiones por causa externa	66
5.1.2. Análisis del subregistro por códigos inespecíficos	67
5.1.3. Análisis de balanceo mediante texto sintético	68
5.1.4. Elección y descripción del modelo	71
5.2. Conclusiones	71
5.3. Trabajo futuro	72
A. Artículos publicados	83
A.1. Epidemiología de las lesiones por accidentes de tránsito en el servicio de urgencias de un hospital de tercer nivel	83
A.2. Minería de textos para el análisis del subregistro de lesiones por causa externa en el servicio de urgencias de un hospital de tercer nivel	84
A.3. Lesiones por causa externa en el servicio de urgencias de un hospital en un periodo de cinco años	84
B. Herramientas de software	87
B.1. GraphLab Create	87

Índice de figuras

2.1. Matriz de confusión para un clasificador binario	15
2.2. Descenso de gradiente	17
2.3. Funciones de activación para el algoritmo <i>backpropagation</i>	25
2.4. Estructura típica de una Red Neuronal Recurrente [44]	26
3.1. Proceso para generar notas sintéticas a través de una RNN-LSTM	36
3.2. Procedimiento utilizado para obtener el valor F_1 -score promedio de los modelos generados con el algoritmo de regresión logística.	37
3.3. Proceso para entrenamiento y prueba.	40
4.1. Cantidad de pacientes hospitalizados por LCE	44
4.2. Consecuencia resultante de accidentes de tránsito, según el vehículo involucrado	47
4.3. Frecuencia subestimada de lesiones por causa externa.	48
4.4. Exactitud del modelo M1 con diferentes valores de regularización λ	51
4.5. Comportamiento de los coeficientes del modelo M1 a distintos valores de regularización λ	52
4.6. Exactitud del modelo M2 con diferentes valores de profundidad del árbol de decisión	52
4.7. Exactitud del modelo M3 para distintos valores de regularización	53
4.8. Exactitud del modelo M4 con diferentes valores de regularización λ	54
4.9. Resultados de la clasificación de dolores agudos	55
4.10. Frecuencia subestimada de las lesiones por causa externa según los modelos implementados.	56
4.11. Rendimiento de los algoritmos utilizando balanceo por notas sintéticas - notas médicas preprocesadas	58
4.12. Comparación de métodos de balanceo para diferentes algoritmos de clasificación - notas sintéticas preprocesadas.	60
4.13. Comparación de la estabilidad de los algoritmos para diferentes conjuntos de datos y subconjuntos aleatorios - sobremuestreo aleatorio.	62
4.14. Comparación de la estabilidad de los algoritmos para diferentes conjuntos de datos y subconjuntos aleatorios.	63
B.1. Representación de los datos en formato de texto libre y su correspondiente representación TFIDF	89

Índice de tablas

1.1. Clasificación de las lesiones por tipo de diagnóstico	3
3.1. Distribución de los datos según la etiqueta asignada para el análisis de subregistro por códigos inespecíficos.	33
3.2. Resumen de modelos generados para el análisis de códigos inespecíficos.	33
3.3. Parámetros utilizados en la RNN-LSTM para la generación de notas sintéticas.	37
4.1. Características sociodemográficas de los pacientes atendidos.	44
4.2. Pacientes que requirieron hospitalización después de su atención en urgencias.	46
4.3. Grupos encontrados mediante análisis de clúster.	49
4.4. Desempeño de los modelos contra los datos de prueba.	50
4.5. Variables con coeficientes más representativos del modelo M1 $\lambda = 100$	51
4.6. Palabras con coeficientes más representativos del modelo M4 con $\lambda = 1000$	54
4.7. Clasificaciones realizadas con datos reales de dolor agudo.	55
4.8. Comparación visual entre notas reales y las notas sintéticas.	57
4.9. Evaluación del rendimiento promedio para diferentes métodos de balanceo utilizando el conjunto de datos de notas médicas preprocesadas ($n = 33$ para todos los casos)	59
4.10. Evaluación del rendimiento promedio para diferentes conjuntos de datos con el método de balanceo por notas sintéticas ($n = 33$ para todos los casos)	61
4.11. Evaluación del rendimiento promedio para diferentes conjuntos de datos con el método de balanceo por sobremuestreo aleatorio ($n = 33$ para todos los casos)	62
4.12. Palabras con coeficientes más representativos para el modelo predictivo del tiempo de estancia hospitalaria $\lambda = 12000$	64

Capítulo 1

Introducción

La demanda de servicios de salud está en constante aumento alrededor del mundo; el crecimiento y envejecimiento de la población son factores que impactan en los costos y calidad de la atención. Sin embargo, la cantidad de recursos asignados no aumenta con la misma proporción, esto convierte a su gestión en un tema interés para las instituciones dedicadas al cuidado de la salud, paradójicamente la asignación de los insumos puede verse afectada tanto por la escasez como por el exceso de los mismos [1].

Uno de los principales indicadores utilizados en la asignación de recursos es el tiempo de estancia de un paciente en un servicio hospitalario. Dependiendo de dicho servicio, este parámetro puede medirse en horas o en días. El tiempo de estancia es una importante métrica tanto para las instituciones como para los pacientes ya que está relacionada, no solo con los costos de atención y la calidad del servicio, sino también con el estado de salud, el diagnóstico y el pronóstico del paciente. Es común relacionar una estancia corta con menor cantidad de recursos consumidos y bajos costos de atención [2, 3]. Debido a esto, es frecuente que se utilice como indicador de la efectividad y eficiencia de los servicios de salud [4, 5].

Para dotar a las autoridades hospitalarias con información veraz y oportuna se han desarrollado modelos que buscan detectar a pacientes con riesgo de estancias prolongadas. Debido a sus características, un modelo predictivo puede brindar información para coadyuvar en la toma de decisiones y el uso eficiente de recursos, detectando casos con características que requieran mayor tiempo de atención [1]. Los modelos predictivos tradicionales se basan en conjuntos de datos estructurados específicos para un área, diagnóstico, situación o incluso a un sistema de información hospitalario. Normalmente estos datos se encuentran en variables cualitativas o cuantitativas, sin embargo, este tipo de representación puede presentar resultados dudosos debido a problemas como los valores sesgados, datos nulos y códigos inespecíficos [6, 7]. Así mismo, se ha demostrado que gran parte de la información sobre atención hospitalaria se encuentra en formatos no estructurados proveniente de hojas escritas en formato libre, ya sea manual o electrónico [8, 9].

Se han propuesto modelos enfocados en características particulares de casos específicos. Se sabe, por ejemplo, que el diagnóstico del paciente es uno de los principales factores que impactan en el tiempo de estancia. Por lo que se han hecho predicciones con base a ésta

variable [5]. No obstante, aunque dos pacientes presenten la misma enfermedad o se les haya practicado el mismo procedimiento, su estancia tiende a variar de un sujeto a otro. Incluso se ha demostrado que puede haber diferencias cuando se trata del mismo padecimiento [10]. Esto puede ser debido a que cada caso presenta sus propias particularidades, tales como comorbilidades, eventos adversos (EAs) o situaciones administrativas que pueden afectar el tiempo de estancia en el servicio.

Se han realizado estudios que demuestran que variables tan diversas como la edad, el índice de masa corporal, el estado civil, la presencia de comorbilidades, consumo de drogas o la cobertura de la seguridad social están relacionadas con el tiempo de estancia en un servicio [11, 12, 13]. También, se han incluido las características del hospital así como sus políticas y la probabilidad de sufrir EAs [1, 2, 14]. Otros enfoques se centran en un área específica dentro de la atención hospitalaria o en un procedimiento quirúrgico [15]. Esto hace que la elección de variables sea una tarea que se repite en cada caso.

El uso de modelos basados en texto puede dar independencia sobre las variables, particularidades de una institución o datos de sistemas de información antagónicos reduciendo el tiempo y los esfuerzos empleados en la búsqueda de variables para cada modelo. En esta tesis se realizó un análisis de los datos en formato texto para la predicción de casos de estancia hospitalaria prolongada. Inicialmente, se estudian los casos de pacientes atendidos en el servicio de urgencias y que presentan Lesiones por Causa Externa (LCE).

1.1. Conceptos básicos

En esta sección se proporcionan las definiciones necesarias que se requieren para comprender el resto del documento.

1.1.1. Tiempo de estancia hospitalaria

Se refiere al periodo de tiempo que transcurre entre el ingreso de un paciente a un servicio hasta su egreso. En esta tesis, el tiempo de estancia se codificó en dos clases *estancia normal* para permanencias menores o iguales a ocho horas y *estancia prolongada* en otro caso, de acuerdo a las recomendaciones del *Manual de Procedimientos del Servicio de Urgencias para Hospitales Generales* del Instituto de Salud del Estado de México [16].

1.1.2. Lesiones por causa externa

Las LCE constituyen un problema de salud pública en el mundo debido a la cantidad de casos y a los problemas que derivan de ellas. Se ha reportado que representan hasta el 26.3% de la demanda de los servicios de urgencias y están consideradas como la cuarta causa de muerte en el continente Americano [17].

Estas lesiones son causadas por el tránsito, ahogamiento, envenenamiento, caídas, quemaduras, violencia, agresiones, autoinfligidas o traumatismos de cualquier tipo. Las lesiones se encuentran dentro de las diez principales causas de muerte, de las cuales aproximadamente 39.4% se deben a los accidentes de tránsito. Un problema importante en el análisis de las lesiones por causa externa es que existe un alto porcentaje de *subregistros* lo que resta confiabilidad a las cifras mencionadas [18].

Cada LCE puede generar múltiples diagnósticos cada uno con sus propias características, complicaciones y pronósticos. Una caída, por ejemplo, puede provocar traumatismos en la cabeza, en el abdomen, en extremidades o todas las anteriores de manera simultánea. En el trabajo realizado por Ávila-Burgos et al. [18] se realizó una agrupación de diagnósticos relacionados con LCE, esta clasificación se basa en las claves de la Clasificación Internacional de Enfermedades, versión 10 (CIE-10) y se dividen en ocho grupos (Tabla 1.1).

1.1.3. Clasificación Internacional de Enfermedades (CIE-10)

La CIE-10 se empezó a utilizar en los Estados Miembros de la Organización Mundial de la Salud (OMS) a partir de 1994. La primera edición tiene sus orígenes en el año 1850. La OMS asumió la responsabilidad de la CIE cuando se publicó la Sexta Revisión, en la cual se incluyeron por primera vez las *causas de morbilidad*.

La CIE-10 tiene el propósito de facilitar el registro sistemático, el análisis, la interpretación y la comparación de datos de mortalidad y morbilidad. También es uno de los estándares internacionales que más se utiliza para elaborar estadísticas de salud en el mundo. Su principal utilidad radica en convertir los términos diagnósticos y otros problemas de salud normalmente registrados en palabras, a códigos alfanuméricos para facilitar su manejo y el análisis estadístico de la información [19].

Tabla 1.1: Clasificación de las lesiones por tipo de diagnóstico. Modificado de [18].

Tipo de lesión	Código CIE-10
Traumatismos de cabeza y cuello	S00-S19
Traumatismos de tórax, abdomen y pelvis	S20-S39
Traumatismos de extremidad superior	S40-S79
Traumatismos de extremidad inferior	S80-S99
Lesiones múltiples	T00-T07
Quemaduras y corrosiones	T20-T32
Intoxicaciones	T36-T65
Otras lesiones	T08-T19, T33-T35, T66-T88, T90-T99

1.2. Estado del arte

Por su impacto en la administración de los recursos hospitalarios, la predicción del tiempo de estancia hospitalaria es un tema de mucho interés y ha sido estudiado desde distintos puntos de vista. En algunos casos se han construido modelos predictivos utilizando métodos estadísticos como regresiones lineales simples y análisis de correlación sobre conjuntos de datos estructurados normalmente recopilados especialmente para la investigación en curso. En el trabajo publicado por Awad et al. [3] se presenta un resumen de investigaciones relacionadas con la predicción del tiempo de estancia así como sus enfoques y los algoritmos utilizados.

En el trabajo de Azari et al. [1] los autores mostraron que utilizar clusterización para formar conjuntos de datos de entrenamiento, mejora las predicciones en comparación con los no clusterizados. Para sus análisis utilizaron una base de datos disponible en Internet [20]. Un resultado similar se describe en Rouzbahman et al. [21], aunque el objetivo del artículo es evitar la posibilidad de identificación de los pacientes a través de los datos, los autores concluyen que el uso previo de clusterización mejora las predicciones. En sus experimentos utilizaron un análisis de regresión lineal y hacen énfasis en la importancia de la adecuada elección de las variables predictoras.

Rhodes et al. [14] realizó un estudio del tiempo de estancia en el área de urgencias y la relación de éste con la ocurrencia de eventos adversos, el trabajo se efectuó con datos de pacientes mayores con trastornos mentales que requirieron atención del servicio de urgencias, para ello se diseñó un análisis de regresión lineal múltiple realizando análisis de Kappa, two sample t-test, ANOVA y correlación de Pearson, este último con objeto de elegir las variables más representativas. Concluyen que por cada 10 horas adicionales, el riesgo de sufrir un EA se incrementa 20%. Una limitación importante mencionada por los autores es que, debido a la particularidad del estudio es poco probable una generalización del modelo.

El trabajo realizado por Ramaraju et al. [13] utiliza un análisis de regresión logística para diseñar un modelo que predice el tiempo de estancia hospitalaria que está relacionada con la enfermedad pulmonar obstructiva crónica. En este caso se utiliza una variable objetivo con dos clases: estancia prolongada y estancia no prolongada donde más de 6 días se considera como una estancia prolongada. Las variables más relevantes en la predicción son la edad, el índice de masa corporal, la presencia de comorbiliades y la capacidad de ejercicio. El modelo presentó la limitación de variables que presentan valores nulos.

Gordon et al. [11] utilizan un método conocido como fase condicional discreta (DC-Ph) el cual consiste en dos componentes uno condicional y uno de proceso. Los autores proponen el uso de árboles de supervivencia como componente condicional con el que se utilizan covarianzas para formar grupos de pacientes basados en la distribución de su estancia. Finalmente, se utiliza la distribución de fase de Coxian para modelar el periodo de tiempo. Una limitante del trabajo es que únicamente utiliza tres variables independientes: edad, método de admisión y estado civil. El modelo fue validado comparándolo con la media empírica para cada grupo formado y presentó un 95% de confianza.

En un estudio realizado en Taiwan, Chuang et al. [15] dividieron las cirugías generales en

cirugías programadas y cirugías urgentes, con un total de 897 casos, utilizaron árboles de decisión (DT), máquinas de soporte vectorial (SVM) y *random forest* (RF) para calcular estancias prolongadas. El modelo más exacto y confiable fue el de RF con una exactitud de 0.87. Los autores sugieren la validación del modelo con otras técnicas de predicción y en diferentes hospitales.

Otro estudio que involucra algoritmos predictivos es el presentado por LaFaro et al. [25] donde se comparan las Redes Neuronales Artificiales (ANN) con DT y con RF para la predicción del periodo de estadía de pacientes ingresados a la unidad de cuidados intensivos de cirugías cardíacas. Se analizaron 185 casos con 8 variables que presentaron asociación estadísticamente significativa con respecto al tiempo de estancia. Se concluye que las ANNs presentan mayor exactitud (odds ratio de 9.8, $p < 0.0001$) que los clasificadores RF y DT.

Pei-Fang et al. [2] abordan el tema utilizando ANN, el estudio se enfoca en tres diagnósticos: aterosclerosis coronaria, falla cardíaca e infarto severo al miocardio, para ello utilizan dos ANNs independientes una para aterosclerosis y una para falla cardíaca e infartos severo, plantean un modelo con datos disponibles en la etapa anterior a la admisión y lo comparan con otro modelo en la etapa de cobro; sus resultados son comparados con los obtenidos mediante modelos de regresión logística. Encontraron que la regresión logística, en general, tiene mejor exactitud, pero las ANNs fueron mejores en la predicción de estadías de entre 8 y 11 días. Ninguno de los dos métodos fue capaz de predecir correctamente permanencias mayores a 18 días. La variable tiempo de estancia fue tomada como numérica de 0 a 35 días y se utilizó correlación de Pearson para definir las variables que se integraron el modelo.

Gholipour et al. [26] también proponen el uso de ANN, en este caso para pacientes con trauma en un hospital Iraní, el modelo obtiene una sensibilidad y especificidad del 95% y la validación se realiza mediante un análisis de coeficiente de correlación ($r=0.643$) entre lo predicho por la red neuronal y los datos reales registrados. Analizaron 125 datos en total.

Hachesu et al. [12] realizaron un estudio comparativo de algoritmos predictores que incluyó DT, SVM, una ANN y un algoritmo ensamblado que combina los tres anteriores. La SVM y el algoritmo ensamblado mostraron los mejores resultados con una exactitud del 0.96 y 0.98, respectivamente. También se encontró que las comorbilidades tales como enfermedades respiratorias y presión alta son factores que pueden afectar las estancias hospitalarias.

Otro estudio, donde se comparan algoritmos para la predicción del tiempo de estancia, es el descrito por Morton et al. [27] donde se utilizan datos de pacientes con diagnóstico de diabetes como caso de estudio. En el trabajo se evalúa el rendimiento de la regresión lineal múltiple, SVM, multi-task learning y RF. Para clasificar estancias largas contra estancias cortas. En este trabajo se dicotomizó la estadía del paciente en forma categórica donde menos de tres días se considera una estancia corta. Esta decisión se basa en en la distribución de los datos, sin embargo podría considerarse arbitraria. Las SVM obtuvieron los mejores resultados en cuestión de exactitud (0.68).

Abbi et al. [28] utilizaron un modelo gaussiano mixto para abordar el problema. Su enfoque se basó en la descomposición de la distribución de las estancias en grupos, tratando de eliminar el sesgo natural que dicha distribución presenta y planteando la hipótesis de que la

distribución de los tiempos está compuesta de varios grupos homogéneos y con una distribución normal. El enfoque práctico de la solución responde a dos planteamientos. Primero, dado un paciente que a permanecido, por ejemplo, 29 días en hospitalización, ¿cuál es la probabilidad de que pertenezca a un grupo de estancias en particular?. Segundo, dado que se conoce que un paciente pertenece a un grupo, ¿cuál es la probabilidad de que egrese dentro de cierto número de días?.

Con el objetivo de identificar pacientes, con probabilidad de tener estancias prolongadas, que excedan lo recomendado, Cheng et al. [4] desarrollaron un sistema de predicción automática que utiliza SVM para detectar pacientes con probabilidades de excederse más allá del tiempo recomendado para una apendicectomía. Su sistema utiliza la variable objetivo en formato categórico, donde una de las partes especifica los pacientes que están dentro del rango recomendado y la otra los que lo exceden. Debido a que la distribución de la muestra se presentó altamente desbalanceada, los autores incorporaron re-muestreo y métodos costo-sensitivos; así mismo, realizaron una comparación entre el método genérico y el modificado. Los resultados muestran valores de exactitud de 0.83.

Nouaouri et al. [29] propone un método denominado Evidential Length Of Stay prediction Algorithm (ELOSA), junto con el cual utilizan la denominada teoría de la evidencia para tratar con los problemas de preprocesado de datos. Para realizar sus predicciones, se apoya de 270 pacientes y toma en cuenta parámetros como edad, sexo y condiciones psicológicas de los mismos. Las variaciones realizadas al método propuesto presentan porcentajes distintos de *buenas predicciones* que van desde el 70 % al 90 % de exactitud.

En el estudio de los factores que afectan el análisis de datos y el rendimiento de algoritmos de clasificación o predicción se han identificado y analizado, al menos dos problemas recurrentes:

1. El subregistro por códigos inespecíficos.
2. Presencia de clases desbalanceadas.

Distintos autores han identificado, dentro de la clasificación CIE-10 un conjunto de códigos denominados *códigos inespecíficos* [6]. Estos tienen la particularidad de no proporcionar información contundente sobre los padecimientos diagnosticados y, como consecuencia minimizan su prevalencia. Sin embargo, esto no solo afecta a los diagnósticos, también se puede presentar al registrar opciones tales como *otros, se desconoce, no disponible, no respondió*, etc. El trabajo presentado por Pérez-Nuñez et al. [6] se estima que, debido a códigos inespecíficos, existe un subregistro del 18.85 % de muertes que se pueden atribuir a accidentes de tránsito. Otro estudio presentado por Híjar et al. [7] muestra un aumento del 18 al 45 % en las muertes causadas por accidentes de tránsito. Utilizan tres métodos para calcular el subregistro: proporcional, imputación simple y regresión. Variables como el diagnóstico, procedimientos o medicamentos han demostrado importancia para la estimación de la estancia hospitalaria y que pueden ser proclives al subregistro por códigos inespecíficos.

El desbalanceo en un clasificación binaria se presenta cuando una clase tiene un porcentaje mucho mayor de casos que la otra. Este es un problema de clasificación ampliamente estudiado y fue considerado entre los 10 problemas en minería de datos y reconocimiento de patrones

[30], sin embargo, no se puede establecer una técnica, algoritmo o método que pueda ser generalizable a cualquier situación. Distintos autores coinciden en que es una característica común en muchas situaciones de la vida real, tales como: detección de fraudes, diagnósticos de enfermedades extrañas o detección de intrusos en una red, solo por mencionar algunos. Así mismo, el problema del desbalanceo puede abordarse mediante la manipulación de los datos con técnicas como el *submuestreo* o el *sobremuestreo* o mediante la modificación de los algoritmos de clasificación.

1.3. Planteamiento del problema

Se han presentado varios trabajos que abordan el tema de la predicción del tiempo de estancia en distintos servicios de un hospital. Un problema que se presenta frecuentemente es la imposibilidad de trasladar un modelo predictivo, desde una situación a otra con circunstancias distintas. El problema radica en varios factores entre los que se pueden mencionar: la presencia de variables con valores nulos o sesgados, problemas de subregistro por códigos inespecíficos y la diversidad de las variables que componen al modelo. La elección de las variables predictoras es una etapa importante en el entrenamiento de algoritmos de clasificación y puede llegar a ser lenta y tediosa.

1.4. Hipótesis

Los modelos basados en texto libre tienen la media armónica necesaria para ser utilizados en la predicción del tiempo de estancia hospitalaria en pacientes con lesiones por causa externa.

1.5. Objetivos

1.5.1. Objetivo general

Construir un modelo predictivo del tiempo de estancia en un servicio de urgencias médicas para pacientes con lesiones por causa externa.

1.5.2. Objetivos específicos

1. Realizar una caracterización de las lesiones por causa externa de los pacientes atendidos en un servicio de urgencias.
2. Analizar el subregistro por códigos inespecíficos en pacientes con lesiones por causa externa en un servicio de urgencias.

3. Analizar el desbalanceo de clases para tareas de clasificación y predicción en los registros del servicio de urgencias médicas.

1.6. Alcances y limitaciones

De acuerdo con la literatura revisada, el tiempo de estancia esperado para un paciente, está en relación con varios factores, entre los que se pueden citar las características del paciente, las de su padecimiento, las del hospital y las administrativas, solo por mencionar algunas. Para el desarrollo de este trabajo se cuenta con datos de las características de la lesión, de su atención en urgencias y de las características sociodemográficas de los pacientes pero no se dispone de datos sobre estudios de laboratorio, estudios imagenológicos, sobre las características de la infraestructura hospitalaria o de sus políticas internas, por lo que estos datos no forman parte del análisis propuesto.

Como propuesta inicial, que nos permita validar el uso de notas médicas de LCE como predictores, se evalúa la exactitud de un modelo para la estimación de datos afectados por el subregistro por códigos inespecíficos y se realizan comparaciones contra modelos generados con base a datos estructurados. Para las pruebas se analizan variaciones de los modelos mediante el método de regularización ridge regression buscando el valor óptimo que maximice la confiabilidad. En esta evaluación inicial, los modelos no se someten a métodos tales como lasso regression, análisis de componentes principales o validación cruzada.

Posteriormente se aborda el tema del desbalanceo de clases con datos de texto. En esta fase del proyecto se presenta un método alternativo orientado específicamente al balanceo de datos de texto en el cual se generan notas sintéticas mediante RNN-LSTM. El ajuste de parámetros de la red que genera el texto sintético se realiza con base a las recomendaciones descritas por los autores de la misma. En este trabajo, no se llevan a cabo evaluaciones de la calidad del texto generado y se utilizan los rangos de valores recomendados.

1.7. Contenido

El contenido de este documento inicia con el presente capítulo, donde se mencionan los conceptos básicos y los antecedentes de los que surge esta tesis, se presenta una hipótesis y se describen los objetivos que se pretende alcanzar.

En el segundo capítulo, se proporciona una introducción a las herramientas de minería de datos y conceptos relacionados que serán utilizados en el resto del documento.

En el tercer capítulo, se describen los datos, métodos y procedimientos empleados para abordar cada uno de los objetivos planteados y la forma en que se realizan las evaluaciones y comparaciones, según sea el caso. Al final del capítulo se presentan los criterios para elegir el modelo predictivo final.

Los resultados obtenidos y la discusión de los mismos se presentan en los capítulos cuatro y cinco, respectivamente. Ambos capítulos se dividen en tres fases que corresponden a cada objetivo planteado. En el capítulo quinto también se aborda la elección del modelo, las conclusiones y el trabajo futuro.

Por último, en el apéndice A se presentan los resúmenes de las publicaciones que emergen de la presente investigación. Y en el apéndice B se explica el código empleado para la implementación de los modelos presentados.

Los objetivos planteados dividen al proyecto en tres fases, cada una tiene su propia sección en los capítulos tres, cuatro y cinco. Para facilitar la lectura, se pueden leer de manera individual. La fase dedicada a la caracterización de las lesiones se presenta en las secciones 3.1, 4.1 y 5.1.1. Para la fase relativa al subregistro por códigos inespecíficos es posible remitirse a las secciones 3.2, 4.2 y 5.1.2. Finalmente, para lo relativo al balanceo de clases y elección del modelo predictivo se pueden revisar las secciones 3.3, 4.3, 5.1.3 y 5.1.4.

Capítulo 2

Marco Teórico

2.1. Aprendizaje Automático

El aprendizaje automático (ML), cuenta con gran interés por investigadores de distintas áreas, hoy en día existen cientos de aplicaciones que utilizan sus técnicas para brindar soluciones en distintos ámbitos. En esta sección se presenta un resumen de conceptos básicos sobre el tema.

Existen distintas definiciones de ML. Puede ser definido como la ciencia (y el arte) de programar computadoras para que puedan aprender a partir de los datos [31].

Dependiendo del tipo de datos, la forma de tratarlos y las necesidades propias de la aplicación, los sistemas de ML pueden dividirse de la siguiente manera:

- Interacción con un humano.
 - Supervisado. Este tipo de aprendizaje se caracteriza porque el conjunto de entrenamiento que se utiliza para formar el modelo cuenta con la solución deseada, normalmente llamada *etiqueta*. Dependiendo del tipo de dato de la etiqueta (categórico o numérico), se le denomina clasificación o regresión, respectivamente.
 - No supervisado. Como se puede suponer, el aprendizaje no supervisado carece de una *etiqueta* por lo que se intenta aprender sin un *maestro*. Las principales tareas de este tipo de aprendizaje son: clustering, visualización, reducción de dimensiones y reglas de asociación.
 - Semi-supervisado. Algunos algoritmos tienen la capacidad de manejar datos parcialmente etiquetados. Normalmente con una mayoría no etiquetada y una minoría etiquetada.
 - Por refuerzo. En este tipo de aprendizaje el sistema aprende por retroalimentación. Primero selecciona y ejecuta una acción, recibe una retroalimentación positiva o negativa y actualiza sus políticas para *aprender* la mejor opción.

- Forma de aprendizaje.
 - En línea. En este tipo de aprendizaje el modelo aprende gradualmente ya sea con casos individuales o pequeños grupos (mini-batches). Cada paso en el entrenamiento tiene poco costo y suelen ser rápidos de ejecutar. Un parámetro importante es la tasa de aprendizaje (learning rate) que determina qué tan rápido el modelo es adaptado a los nuevos datos.
 - Batch. Cuando el aprendizaje toma mucho tiempo y utiliza gran cantidad de recursos, se realiza un aprendizaje en batch o fuera de línea. Primero se entrena al modelo y después se lanza a producción donde ya no sigue aprendiendo de los nuevos datos.
- Forma de procesamiento.
 - Basados en instancias. En este tipo de aprendizaje el sistema toma la instancia y la clasifica según su similitud con los datos pertenecientes a cada clase.
 - Basados en modelos. En este caso se construye un *modelo* que después se utiliza para realizar las predicciones. Normalmente los datos son divididos para entrenar, validar y probar la exactitud del modelo.

Estas clasificaciones no son mutuamente excluyentes y en la mayoría de las aplicaciones se combinan para dar solución a la problemática planteada.

2.1.1. Elección de predictores

Una tarea a considerar en el entrenamiento de modelos predictivos es la elección de características o variables predictoras (*feature engineering*). Los modelos solo serán tan buenos como lo sean los datos (*Garbage in - Garbage out*). Por lo tanto, se debe contar con la cantidad suficiente de variables pero solo deberían utilizar las relevantes para el modelo. Se realizan las siguientes tareas:

- Elección de características. Seleccionar las características más útiles en el contexto del problema planteado.
- Extracción de características. Es la combinación de las características existentes para producir nuevas más útiles.
- Crear nuevas características a partir de la disponibilidad de más datos.

2.1.2. Sobre ajuste del modelo

El sobre ajuste, mejor conocido como *overfitting*, consiste en que el modelo generado a partir del entrenamiento se comporta bien con los datos con los que fue entrenado pero no puede

ser generalizado a datos nuevos. Al aumentar la complejidad de un modelo se puede obtener mayor exactitud de predicciones en tiempo de entrenamiento, sin embargo, al momento de validarlo la exactitud puede degradarse.

Al proceso de restringir al modelo para que su complejidad permanezca simple y de esta manera reducir el riesgo de overfitting se le conoce como *regularización*. La cantidad de regularización que se aplica durante el entrenamiento puede ser controlado mediante parámetros, según el algoritmo utilizado [33]. El proceso de elección del mejor valor para este parámetro es una parte importante en la construcción de modelos de ML.

Ridge Regression

También llamada penalización L2, es un tipo de regularización de los modelos basados en regresión lineal. Funciona agregando el término $\lambda \sum_{i=1}^n \theta_i^2$ a la ecuación para reducir el error cuadrático medio (MSE) en la regresión lineal (ecuación 2.12). Esta regresión mantiene los pesos θ tan bajos como sea posible.

El parámetro λ controla la regularización del modelo. Si $\lambda = 0$ entonces se realiza la regresión lineal normal. La ecuación 2.1 representa la regularización Ridge Regression.

$$J(\theta) = MSE(\theta) + \lambda \frac{1}{2} \sum_{i=1}^n \theta_i^2 \quad (2.1)$$

Lasso Regression

Otra forma de regularización es la denominada Lasso Regression. De manera similar a Ridge Regression, también se agrega un término de regularización como se muestra en la ecuación 2.2. A diferencia de Ridge Regression, en este caso los pesos de las características menos importantes pueden disminuirse hasta ser eliminados con lo que se puede realizar una elección de características

$$J(\theta) = MSE(\theta) + \lambda \frac{1}{2} \sum_{i=1}^n |\theta_i| \quad (2.2)$$

Elastic Net

La regularización Elastic Net es un punto medio entre Ridge Regression y Lasso Regression. Se agrega un parámetro de regularización denominado *ratio* r . Cuando $r = 0$, Elastic Net es equivalente a la regularización Ridge Regression y cuando $r = 1$, es equivalente a Lasso Regression. En la ecuación 2.3 se muestra la regularización Elastic Net.

$$J(\theta) = MSE(\theta) + r\lambda \sum_{i=1}^n |\theta_i| + \frac{1-r}{2}\lambda + \sum_{i=1}^n \theta_i^2 \quad (2.3)$$

Elastic Net se comporta de mejor manera cuando el número de características es mayor que el número de registros de entrenamiento o cuando varias características se encuentran correlacionadas.

2.1.3. Prueba y validación

Para medir el nivel de exactitud del modelo entrenado se ejecutan procedimientos de validación y prueba. Normalmente, se realiza una división del conjunto de datos disponible en subconjuntos denominados de entrenamiento (training set), de validación (validation set) y prueba (test set)¹. La tasa de error del modelo frente a nuevos casos es conocida como *error de generalización*. Este error es una medida de que tan bueno es el modelo con datos desconocidos para él.

Un método común utilizado para la validación y prueba es conocido como validación cruzada (cross-validation) en el que conjunto de entrenamiento es dividido en subconjuntos y se entrena un modelo para cada combinación de estos subconjuntos. El método aleatorio divide K veces al conjunto de datos en subconjuntos de entrenamiento, validación y prueba; para cada subconjunto se entrena y valida un modelo clasificador. Se eligen los parámetros que mostraron el mejor rendimiento y se entrena el modelo final con estos parámetros y el conjunto de entrenamiento completo [34].

Si el error de entrenamiento es bajo pero el error de generalización es alto, entonces el modelo presenta overfitting y debe ser regularizado.

2.1.4. Medidas de rendimiento

La medida que se utilice para validar el modelo depende del tipo de modelo utilizado.

Regresión

Para algoritmos de *regresión* la medida típica es la raíz del error cuadrático medio (RMSE) el cual esta dado por la ecuación 2.4.

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x^{(i)}) - y^{(i)})^2} \quad (2.4)$$

¹Comúnmente se utiliza un 80% de los datos para el entrenamiento y el resto para validación y prueba. Sin embargo, esto puede variar dependiendo de las circunstancias.

Donde m es el número de registros en el conjunto de datos. $x^{(i)}$ es un vector con los valores de todas las variables predictivas del i^{th} registro en el conjunto de datos. y^i es el valor deseado del i^{th} registro del conjunto de datos. X es la matriz que contiene todos los valores de las variables predictivas de todos los registros en el conjunto de datos. h es la función de predicción.

En ciertas circunstancias se prefiere como medida de error el *Mean Absolute Error*, que se calcula mediante la ecuación 2.5

$$MAE(X, h) = \frac{1}{m} \sum_{i=1}^m |h(x^{(i)}) - y^{(i)}| \quad (2.5)$$

Ambos el RMSE y el MAE miden la distancia entre dos vectores, en este caso es la distancia entre el vector de predicciones y el vector con los valores esperados. El RMSE es más sensitivo a valores sesgados que el MAE. Cuando la distribución de los datos se aproxima a la curva de Gauss, se prefiere el RMSE [31].

Clasificación

Una de las medidas más populares para evaluar el rendimiento de un clasificador es mediante la exactitud (tasa de predicciones correctas). Sin embargo, cuando los datos se presentan desbalanceados, esta no es una medida adecuada del rendimiento del clasificador ya que puede sesgar los resultados en favor de la clase mayoritaria [32].

Una mejor manera de evaluar el rendimiento de un clasificador es mediante el valor de la media armónica o F_1 -score obtenido mediante la denominada *matriz de confusión*. Normalmente, la matriz de confusión representa los valores deseados en sus renglones, mientras que las columnas representan las predicciones realizadas por el clasificador. La Figura 2.1 muestra una matriz de confusión para un clasificador binario.

		Clasificación	
		Clase A	Clase B
Valor real	Clase A	Verdaderos Positivos	Falsos negativos
	Clase B	Falsos positivos	Verdaderos Negativos

Figura 2.1: Matriz de confusión para un clasificador binario

La matriz de confusión brinda información con la que es posible calcular distintas métricas para la evaluación del rendimiento de un clasificador. Por ejemplo, la *precisión* del clasificador esta dado por la ecuación 2.6.

$$P = \frac{VP}{VP + FP} \quad (2.6)$$

P es la precisión del modelo, VP son los verdaderos positivos y FP los falsos positivos.

Además de la precisión se suele hacer referencia a la métrica *sensibilidad* (recall o tasa de verdaderos positivos), esta es la tasa de los registros positivos correctamente detectados por el clasificador. La sensibilidad está dada por la ecuación 2.7.

$$S = \frac{VP}{VP + FN} \quad (2.7)$$

S es la sensibilidad del modelo, VP son los verdaderos positivos y FN los falsos negativos.

La precisión y la sensibilidad suelen combinarse en una sola medida denominada F_1 – score que representa la *media armónica* de la precisión y la sensibilidad, obteniendo buenos valores solo si ambos (precisión y sensibilidad) son altos. El F_1 – score se calcula mediante la ecuación 2.8 [32].

$$F_1 = \frac{VP}{VP + \frac{FN+FP}{2}} \quad (2.8)$$

VP son los verdaderos positivos, FN los falsos negativos y FP los falsos positivos.

Las curvas ROC o *receiver operating characteristic* son otra herramienta utilizada para evaluar clasificadores binarios. Son gráficas que contrastan la tasa de verdaderos positivos (sensibilidad o recall) contra la tasa de falsos positivos.

Una forma de comparar clasificadores es mediante el cálculo del área bajo la curva (AUC). Entre más cercano se encuentre de 1, el clasificador es mejor. Generalmente, cuando las clases no están balanceadas se prefieren las medidas de precisión y sensibilidad.

2.1.5. Descenso de gradiente

El descenso de gradiente (GD) es un algoritmo de optimización. La idea general consiste en encontrar el valor de los parámetros de manera iterativa que minimizan la función objetivo. El algoritmo inicia con un vector de valores θ que se pueden fijar de manera aleatoria, con

cada paso, el algoritmo actualiza el vector con valores que disminuyan el error de ajuste (por ejemplo, MSE para regresión lineal) hasta que se converja al mínimo².

Un parámetro importante a considerar en el algoritmo GD es el denominado *tamaño del paso* o *razón de aprendizaje* (ver Figura 2.2). Si dicho valor es demasiado pequeño el algoritmo requerirá de muchos pasos y más tiempo para lograr la convergencia, mientras que si es demasiado grande puede hacer que el algoritmo falle en encontrar la solución óptima.

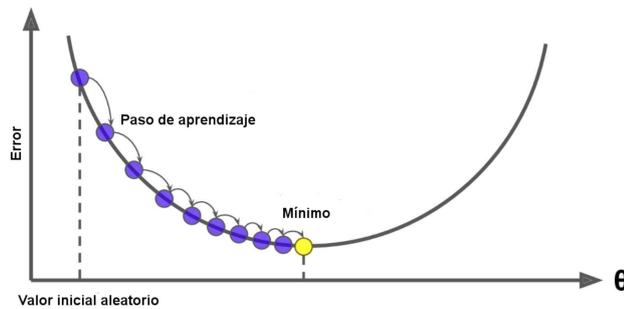


Figura 2.2: Descenso de gradiente. Modificado de [31]

Se han implementado distintas variaciones del algoritmo GD, tal como Batch-GD, Mini-batch GD o Stochastic GD que pueden ser de utilidad cuando se trabaja con cantidades de datos muy grandes y se busca mayor velocidad de convergencia [31].

2.2. Análisis de textos

Actualmente resulta simple generar y almacenar grandes cantidades de datos. Sin embargo, la mayoría de las veces estos datos permanecen acumulados por tiempo indefinido sin que se haga uso de ellos de ninguna forma. En años recientes la cantidad de datos se ha incrementado en varios ámbitos desde Internet, organizacional o incluso familiar o personal. Estos datos pueden presentarse en formato estructurado, semi-estructurado o no-estructurado. [35, 36].

La minería de textos utiliza técnicas de distintas áreas como la minería de datos, aprendizaje automático (ML), procesamiento de lenguaje natural (NLP), recuperación de información (IR) y manejo del conocimiento. De manera análoga a la minería de datos, la minería de textos busca extraer información útil contenida en las fuentes de datos a través de la identificación de patrones.

Sin embargo, mientras la minería de datos se enfoca en el análisis de datos estructurados, en la minería de textos se realiza un pre-procesamiento que se centra en la identificación y extracción de características representativas provenientes de documentos en lenguaje natural con la intención de transformar los datos no-estructurados a formato estructurado. La minería de textos tiene como objetivo descubrir patrones contenidos en grandes *colecciones*

²Antes de utilizar el algoritmo de GD, se deben poner las características en escala similar. De otra manera se incrementa el tiempo de convergencia.

de documentos³, estas colecciones pueden ser estáticas o dinámicas (se pueden agregar, eliminar o cambiar documentos). Es frecuente que cada documento dentro de una colección esté formado por un gran número de características lo que se denomina *dimensionalidad de características* y que se presente el fenómeno denominado *características dispersas* (feature sparsity)⁴. Por otro lado es importante considerar que un documento, puede pertenecer a distintas colecciones de documentos al mismo tiempo.

Aún cuando el texto de un documento es considerado como formato no estructurado, existen distintos elementos tipográficos como puntuación, letras mayúsculas, números y caracteres especiales que pueden dar indicios coadyuvantes a identificar componentes importantes como el título, fecha de publicación o autor del documento, solo por mencionar algunos.

Las principales características que se pueden extraer de un documento son:

- Caracteres. Esta representación puede incluir todos los caracteres de un documento o solo un grupo de ellos, sin embargo, suele estar muy limitada. Otras opciones, basadas en caracteres que incluyen información sobre su posición (bigrams o trigrams) son más comunes y útiles.
- Palabras. Estas son extraídas del cuerpo del documento, puede haber cientos o miles de palabras. A manera de optimización, se suele utilizar solo un subconjunto de ellas, eliminando las denominadas *stop words*⁵, caracteres simbólicos y números.
- Términos. Son simples palabras o frases seleccionadas directamente del documento. Existen distintas metodologías para convertir el texto crudo en una serie de términos normalizados que serán un subconjunto con mayor significado para el análisis.
- Conceptos. Son características generadas en base a procedimientos manuales, estadísticos, basados en reglas o métodos híbridos. Se identifican palabras, expresiones o unidades sintácticas largas. A diferencia de las caracterizaciones basada en palabras o términos, esta puede consistir de palabras no encontradas en el documento. En esta representación se suele utilizar la intervención de un experto en el tópico analizado y suele representar mayor complejidad.

Los formatos anteriores pueden ser mezclados para formar representaciones híbridas, según las características de los documentos analizados.

Puesto que la minería de textos se basa en el análisis basado en características extraídas del documento más que en el documento en sí, se tiene que realizar una representación y elección de características adecuado para el análisis y que no demande recursos en exceso.

³Grupos de documentos en formato texto

⁴Solo un pequeño porcentaje de todas las posibles características para una colección aparece en un documento determinado, entonces, cuando el documento es representado en forma de un vector binario, casi todos los valores de dicho vector son cero [35].

⁵Se refieren a las palabras más comunes en un idioma. Cualquier grupo de palabras puede ser elegida y eliminada con un propósito específico [37].

2.2.1. Representación del documento

Antes de poder utilizar un algoritmo de clasificación o regresión, es necesario convertir el documento original a una representación estructurada, normalmente basada en vectores de características. La representación más común es la denominada *bolsa de palabras* (bag of words) la cual utiliza todas las palabras en un documento como características, por lo tanto la dimensionalidad del vector de características será igual al número de palabras diferentes en el documento. Para asignar el valor a cada característica existen distintos métodos de los cuales el más simple es el binario en el cual el valor es uno si la palabra que representa la característica está en el documento o cero si la palabra no existe.

Otros métodos incluyen la frecuencia de palabra en el documento y la denominada *Frecuencia de Término - Frecuencia Inversa de Documento* (TFIDF). La ecuación 2.9 muestra como calcular la TFIDF.

$$TFIDF(p, d) = tf(p, d) \cdot \log\left(\frac{N}{1 + n}\right) \quad (2.9)$$

Donde $TFIDF(p, d)$ es la frecuencia inversa de la palabra p en el documento d , $tf(p, d)$ es la frecuencia de la palabra p en el documento d , N es el total de registros (documentos) y n es el total de registros que contienen la palabra p [38, 39].

Elección de características

El tamaño del diccionario de palabras puede ser grande. La dimensionalidad del vector de características es igual al número de palabras distintas y puede aumentar a cientos o miles, incrementando los recursos computacionales necesarios. Sin embargo, la mayoría de estas palabras resultarán irrelevantes para las tareas de clasificación o regresión, por lo tanto pueden ser eliminadas sin afectar el rendimiento de los algoritmos. A este proceso se le conoce como *elección de características* (feature selection). Es frecuente que se eliminen las *stop words* ya que generalmente no aportan información valiosa al análisis. Además se pueden utilizar medidas de relevancia a cada característica para determinar cuál es valiosa y cuál no. Existen métodos sofisticados como la *ganancia de información* o la chi-cuadrada X^2 para calcular la dependencia entre característica y encontrar las candidatas a eliminación. Una alternativa simple de realizar este proceso es mediante la frecuencia de palabras $tf(p, d)$. Utilizando solo el 10% de las palabras más frecuentes no reduce el rendimiento de los clasificadores [35].

2.3. Algoritmos de clasificación

En esta sección se describen los algoritmos de los clasificadores utilizados en este trabajo. Algunos de los temas, tal como el de optimización por GD son comunes a varios algoritmos, en cuyo caso se hará la aclaración pertinente.

2.3.1. Regresión

Existen múltiples tipos de regresiones. Para fines de este trabajo se analizarán la regresión lineal y la regresión logística. Ambas están estrechamente relacionadas en cuanto a su funcionamiento con la diferencia de que en la regresión lineal se tiene una variable objetivo de tipo numérico, mientras que en la regresión logística dicha variable es categórica (normalmente dicotómica), por lo tanto se utiliza en problemas de clasificación.

Regresión lineal

Existen dos métodos de entrenamiento de modelos de regresión lineal, el método cerrado y utilizando GD en cuyo caso se realiza una actualización gradual de los parámetros del modelo. En este trabajo se analizará el método por GD, para más información sobre ambos métodos consultar [31].

Un modelo de regresión simple utiliza la ecuación 2.10 para realizar predicciones.

$$\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_n x_n \quad (2.10)$$

Donde \hat{y} es la predicción realizada, n es el número de características, x_i es el valor de la i^{th} característica, θ_j es el valor del j^{th} parámetro del modelo (pesos del modelo) que incluye a θ_0 a la que se le denomina *bias* o *intersección*.

La ecuación 2.11 muestra la representación vectorial para el modelo de regresión lineal.

$$\hat{y} = h_{\theta}(\mathbf{x}) = \theta^T \cdot \mathbf{x} \quad (2.11)$$

Donde \hat{y} es la predicción realizada, θ es el vector de parámetros o vector de pesos; este vector contiene θ_0 y el peso de cada característica θ_1 a θ_n . \mathbf{x} es el vector de características de x_0 a x_n , con x_0 siempre igual a 1.

El entrenamiento del modelo consiste en encontrar el conjunto de parámetros θ que mejor se ajusten al conjunto de entrenamiento. Para esto se buscan los valores de θ que minimizan el RMSE de la ecuación 2.4. Sin embargo, por razones de simplicidad se suele utilizar el MSE que se calcula mediante la ecuación 2.12

$$MSE(X, h_{\theta}) = \frac{1}{m} \sum_{i=1}^m (\theta^T \cdot x^{(i)} - y^{(i)})^2 \quad (2.12)$$

Cuando los datos tienen un comportamiento que no se ajusta de manera adecuada a una línea recta, se puede hacer uso de la regresión polinomial. Para esto, se agregan potencias a cada característica y se procede de la misma manera que la regresión lineal. Utilizando este tipo de regresión es posible tener un mejor ajuste a los datos, sin embargo, también es más propenso al overfitting [46, 47].

Regresión logística

Este algoritmo es utilizado en problemas de clasificación binaria. Cuando la probabilidad de pertenencia a una clase rebasa cierto umbral, entonces el registro se etiqueta como perteneciente a esa clase; en otro caso, el registro se etiqueta en la otra clase.

$$\sigma(t) = \frac{1}{1 + \exp(-t)} \quad (2.13)$$

El procedimiento es similar al de regresión lineal, sin embargo, el resultado obtenido es sometido a una función logística. Normalmente se utiliza la función *sigmoid* (ecuación 2.13) que regresará un número entre 0 y 1.

$$\hat{\rho} = h_{\theta}(\mathbf{X}) = \sigma(\theta^T \cdot \mathbf{X}) \quad (2.14)$$

La ecuación 2.14 muestra la forma de calcular la probabilidad de pertenencia a una clase. Una vez que se ha calculado la probabilidad $\hat{\rho}$, la clasificación \hat{y} si $\sigma(t) < 0.5$ entonces $\hat{y} = 0$ y si $\sigma(t) \geq 0.5$ entonces $\hat{y} = 1$ (ecuación 2.15).

$$\hat{y} = \begin{cases} 0 & \text{si } \hat{\rho} < 0.5 \\ 1 & \text{si } \hat{\rho} \geq 0.5 \end{cases} \quad (2.15)$$

En la regresión logística, el objetivo es encontrar un conjunto de parámetros θ de modo que el modelo estime altas probabilidades para los registros de una clase ($y = 1$) y bajas probabilidades para los registros de la otra ($y = 0$).

Utilizando la ecuación 2.13 y la derivada parcial con respecto a θ de la ecuación 2.12, se obtiene la ecuación 2.16 para entrenar un modelo de regresión logística [46, 47].

$$\frac{\delta}{\delta \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (\sigma(\theta^T \cdot x^{(i)}) - y^{(i)}) x_j^{(i)} \quad (2.16)$$

2.3.2. Árboles de decisión

Los algoritmos de árboles de decisión se pueden utilizar para hacer las tareas tanto de clasificación como de regresión. Una característica importante de los árboles de decisión es

que requieren poco preprocesamiento de los datos, de hecho no requieren escalamiento de las variables. Así mismo, suelen ser bastante intuitivos y sus decisiones fáciles de interpretar.

En esta sección se describe el algoritmo *árbol de clasificación y regresión* (CART) para entrenamiento de árboles así como su regularización. El algoritmo CART modela solamente árboles binarios, a diferencia del algoritmo ID3 que puede contener nodos con más de dos nodos *hijos*. La decisión de utilizar uno u otro depende de la naturaleza del problema [31].

El algoritmo CART se basa en el concepto de *pureza* (gini). Se dice que un nodo es puro (gini = 0) si todos los registros en dicho nodo pertenecen a la misma clase. La ecuación 2.17 muestra como calcular la impureza G_i del i^{th} nodo.

$$G_i = 1 - \sum_{k=1}^n \rho_{i,k}^2 \quad (2.17)$$

Donde G_i es la impureza del i^{th} nodo y $\rho_{i,k}^2$ es la tasa de clase k entre los registros en el i^{th} nodo. Y n es el número de diferentes clasificaciones.

El algoritmo divide los datos de entrenamiento en dos subconjuntos (izquierdo/derecho) utilizando una característica k y un umbral t_k . La elección tanto de la característica como del umbral se hace de tal manera que se produzcan los subconjuntos *más puros*. La función costo que el algoritmo trata de minimizar esta dada por la ecuación 2.18.

$$J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right} \quad (2.18)$$

Donde $G_{left/right}$ mide la impureza del conjunto de datos (izquierdo/derecho) y $m_{left/right}$ es el número de registros en los subconjuntos (izquierda/derecha).

Una vez que el conjunto se ha dividido en dos, el proceso se repite recursivamente hasta que se alcanza la profundidad máxima (definida por el usuario)⁶ o cuando ya no se puede encontrar una división que reduzca la impureza de cada nodo.

El algoritmo CART es considerado como un algoritmo codicioso y por lo tanto no garantiza la convergencia a una solución óptima sino a una aproximación. Encontrar el árbol óptimo cae en el ámbito de lo que se conoce como problemas *NP-Complejos*.

Entropía

Otra medida utilizada para las divisiones de los datos en la construcción de un árbol es la entropía. Al igual que la medida de la impureza, la entropía es cero cuando todos los registros del sub-conjunto de datos pertenece a la misma clase. La ecuación 2.19 obtiene su valor.

⁶Existen varios parámetros para detener la construcción del árbol, para más información consultar [31]

$$H_i = - \sum_{k=1}^n \rho_{i,k} \log(\rho_{i,k}) \quad (2.19)$$

Donde $\rho_{i,k} \neq 0$. $\rho_{i,k}$ es la proporción de clase k entre los registros en el i^{th} nodo. Y n es el número de diferentes clasificaciones.

Ambas medidas (impureza y entropía) producen árboles similares, el cálculo de la impureza puede ser ligeramente más rápido, mientras que la entropía obtiene árboles mejor balanceados.

Regresión con árboles de decisión

Como se comentó al inicio de esta sección, los árboles de decisión también pueden realizar tareas de regresión (donde la variable objetivo es de tipo numérica). El algoritmo CART trabaja de forma similar para tareas de clasificación con la excepción de que en lugar de calcular la impureza (o entropía) se calcula el MSE. La ecuación 2.20 es la utilizada para entrenar árboles de regresión.

$$J(k, t_k) = \frac{m_{left}}{m} MSE_{left} + \frac{m_{right}}{m} MSE_{right} \quad (2.20)$$

Donde el $MSE_{nodo} = \sum_{i \in nodo} (\hat{y}_{nodo} - y^{(i)})^2$, y $\hat{y}_{nodo} = \frac{1}{m_{nodo}} \sum_{i \in nodo} y^{(i)}$. MSE_{left} y MSE_{right} representan el valor del nodo izquierdo y derecho respectivamente.

Regularización

Los parámetros necesarios para la regularización de árboles de decisión puede variar entre algoritmos, sin embargo, un parámetro general es el de la profundidad del árbol. Otra forma de realizar la regularización es mediante el método conocido como *poda* donde inicialmente se realiza el entrenamiento del árbol sin considerar ninguna restricción y posteriormente se eliminan los nodos considerados innecesarios. La decisión de si un nodo es o no importante suele hacerse en base a su significancia estadística utilizando pruebas como la X^2 .

2.3.3. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN), representan un modelo libre, adaptativo, tolerante a fallas, paralelo y con procesamiento distribuido. Las aplicaciones de las ANN, van desde funciones de aproximación, clasificación, mapeo no lineal, memoria asociativa, vectores de cuantización, optimización, extracción de características, clustering e inferencia aproximada [40].

Perceptron

El Perceptron es una de las arquitecturas ANN más simple, inventada en 1957 por Frank Rosenblatt. Se basa en una neurona artificial llamada *linear threshold unid* (LTU) donde cada entrada esta asociada con un peso, se realiza la suma de pesos y se aplica una *función escalón* (step function) para obtener el resultado según la ecuación 2.21.

$$z = w_1x_1 + w_2x_2 + \dots + w_nx_n = \mathbf{w}^T \cdot \mathbf{x} \quad (2.21)$$

Donde z es la suma de pesos \mathbf{w} multiplicados por las entradas \mathbf{x} .

La salida del perceptron esta dada por: $h_w(\mathbf{x}) = \text{step}(\mathbf{w}^t \cdot \mathbf{x})$. La función escalón más utilizada en perceptrones es la Heaviside step function (ecuación 2.22).

$$\text{heaviside}(z) = \begin{cases} 0 & \text{si } z < 0 \\ 1 & \text{si } z \geq 0 \end{cases} \quad (2.22)$$

Una LTU simple puede ser utilizada como un clasificador lineal binario. Calcula la combinación lineal de las entradas y, si el resultado sobrepasa cierto umbral, la salida es la clase positiva, de otro modo la salida es la clase negativa (de manera similar a como trabaja la regresión logística o las máquinas de soporte vectorial). Entrenar un red LTU significa encontrar los valores \mathbf{w} que obtengan los mejores resultados de clasificación. El entrenamiento del Perceptron se hace utilizando una variante de la regla de Hebb [41], donde se considera el error cometido por la red. Las entradas llegan al Perceptron y por cada registro, se realiza una predicción. Para cada neurona que produce una predicción errónea, se actualizan los pesos de las entradas que contribuyen a mejorar la predicción.

Perceptron multi-capa

El Perceptron multi-capa (MLP) se compone de una capa de entrada (input layer), una o más capas intermedias (hidden layers), y una capa de salida (output layer). A excepción de la capa de salida, el resto incluyen una neurona *bias* y están conectadas a todas las neuronas de la siguiente capa.

En 1985 Rumelhart et al. [42] presentan el algoritmo de entrenamiento para MLPs denominado *backpropagation*. En este algoritmo, para cada registro de entrada se calcula la salida en las neuronas y en cada capa consecutiva (forward pass). Una vez realizada la predicción, se calcula el error cometido por el MLP (la diferencia entre el valor esperado y el valor calculado por la red). Entonces se calcula cómo contribuye cada neurona, en cada capa intermedia anterior al error de salida, hasta llegar a la capa de entrada. El último paso del algoritmo backpropagation es un paso de GD en los pesos de todas las conexiones de la red.

Para que el algoritmo fuera posible, los autores cambiaron la función escalón de la ecuación 2.22 por la función logística de la ecuación 2.23. Esto es esencial debido a que la función escalón presenta discontinuidad y no es posible que el GD trabaje adecuadamente (ya que se basa en la derivada de la función).

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.23)$$

Sin embargo, el algoritmo backpropagation puede ser utilizado con otras funciones de activación dos de las más populares son la función tangente hiperbólica (ecuación 2.24) y la función ReLU (ecuación 2.25). En la Figura 2.3 se puede observar el comportamiento de las distintas funciones de activación.

$$\tanh(z) = 2\sigma(2z) - 1 \quad (2.24)$$

$$\text{ReLU}(z) = \max(0, z) \quad (2.25)$$

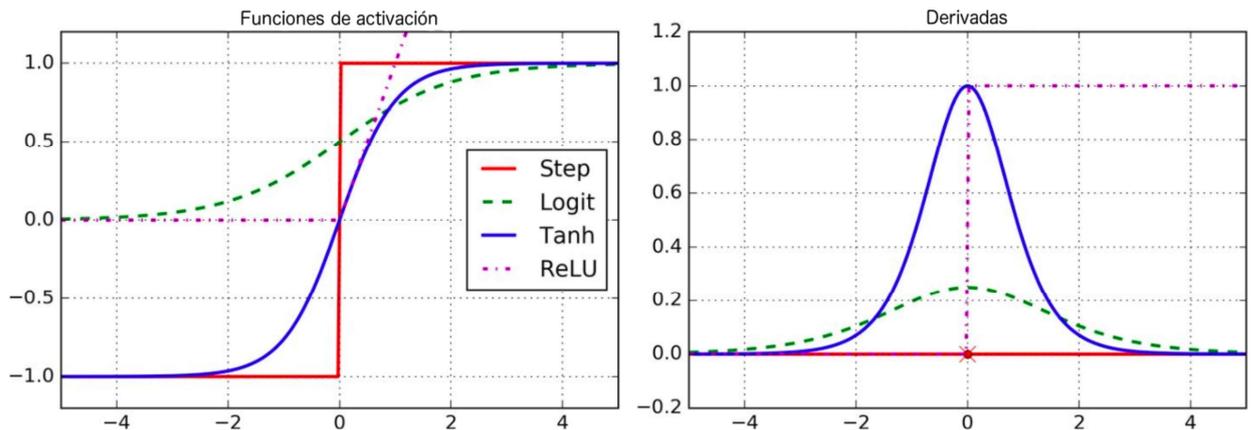


Figura 2.3: Funciones de activación. Modificado de [31]

Trabajar con ANN puede resultar complejo debido a la cantidad de parámetros que tienen que ser definidos, no obstante esto brinda gran flexibilidad a los algoritmos. Se pueden especificar el número de capas, el número de neuronas por capa, el tipo de función de activación la forma de inicialización de pesos, la tasa de aprendizaje o el número de *épocas* (ciclos) que se ejecutará el algoritmo. Para facilitar esta tarea se puede usar el método de validación cruzada (cross-validation) o métodos aleatorios, dependiendo del problema y sobre todo de la cantidad de datos de entrenamiento.

Redes Neuronales Recurrentes

Las Redes Neuronales Recurrente (RNN) están diseñadas para hacer predicciones tomando en cuenta estados previos en un sistema específico. Por ejemplo, analizar escenas de una

película. Son redes con ciclos internos que permiten la persistencia de la información. Una RNN evalúa una entrada x_t y obtiene una salida h_t ; la incorporación de un ciclo, permite la transferencia de la información de un ciclo de la red a otro. La Figura 2.4 muestra la estructura típica de una RNN. Se ha propuesto el uso de este tipo de redes en reconocimiento del habla, modelado de lenguaje, traducción y generación de subtítulos, solo por mencionar algunos. Conforme la brecha entre los estados pasados y el actual crece, se vuelve complicado para una RNN el enlace de la información [43]. Un caso especial de RNN, que resuelve este problema, es la denominada Long short-term memory (LSTM), estas pueden ser utilizadas, para predecir la palabra siguiente en un texto largo, basándose en las palabras anteriores [44, 45]. Este tipo de redes fueron diseñadas específicamente para evitar el problema de dependencias largas.

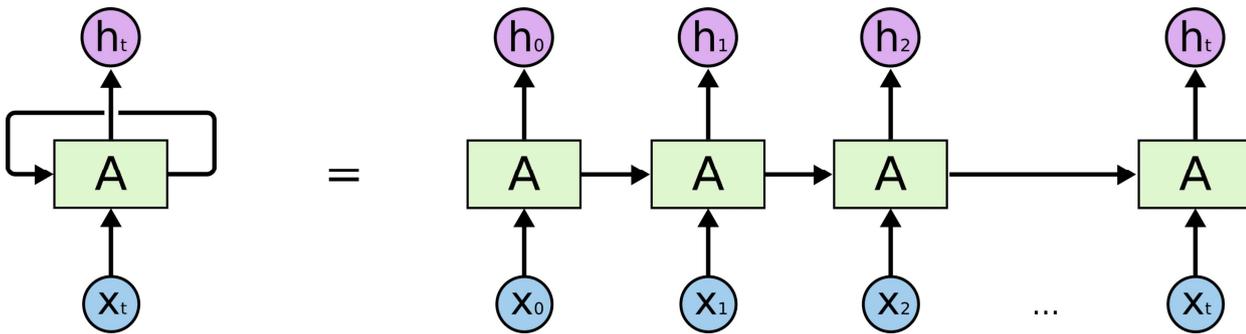


Figura 2.4: Estructura típica de una Red Neuronal Recurrente [44]

2.3.4. Métodos combinados (Ensemble Methods)

La idea de los métodos combinados es simple, varios predictores pequeños trabajando juntos pueden hacer mejores predicciones que un solo predictor más complejo. Un ejemplo es el método conocido como *Random Forest* que consiste en un grupo de árboles de decisión que se entrenan en subconjuntos de datos aleatorios a partir del conjunto de entrenamiento y para las predicciones se toma en cuenta la clase *más votada*. A pesar de su simplicidad, actualmente este método es uno de los más poderosos en el campo del ML [31]. Los métodos combinados trabajan mejor cuando los predictores son independientes. Una manera de lograr la independencia sería utilizar distintos algoritmos para cada clasificador.

Bagging

Este método consiste en utilizar el mismo algoritmo de entrenamiento para cada predictor y entrenar cada uno de ellos con subconjuntos aleatorios del conjunto de entrenamiento original. En este método se utiliza un muestreo con reemplazo⁷. Esto tiene la desventaja de que algunos registros podrían ser seleccionados varias veces por el mismo o distintos predictores, mientras que otros podrían no ser seleccionados nunca. Otro método similar a

⁷Cada elemento puede ser seleccionado más de una vez por lo que puede pertenecer a varios sub-conjuntos de entrenamiento.

bagging es el denominado *pasting* que utiliza muestreo sin reemplazo lo que minimiza el problema pues un mismo registro puede ser elegido por varios clasificadores pero nunca por el mismo [31]. Resumiendo, los dos métodos permiten que el mismo registro sea elegido varias veces a través de múltiples predictores pero solo bagging permite que el mismo registro sea elegido varias veces por el mismo predictor.

Una vez que el entrenamiento se ha realizado, el método combinado puede clasificar un nuevo registro aplicando una función de agregación. Para tareas de clasificación se suele utilizar la *moda estadística* y para tareas de regresión el *promedio*. Una ventaja del método bagging es que cada predictor puede ser entrenado en paralelo. El método RF es una combinación de árboles de decisión, generalmente entrenado mediante el método bagging. Cuando el algoritmo RF entrena los árboles de los que se compone, en lugar de buscar la mejor alternativa para dividir los nodos, busca entre un subconjunto aleatorio de características, esto da lugar a una gran diversidad de árboles que resulta en un mejor modelo combinado.

Boosting

La idea general de este método es entrenar varios predictores secuencialmente cada uno tratando de corregir a su predecesor. Los métodos boosting más conocidos son *AdaBoost* (Adaptive Boosting) y el Gradient Boosting.

En el método *AdaBoost* un nuevo predictor busca corregir a su antecesor en aquellos casos que fueron mal clasificados. De tal manera en que los nuevos predictores se enfocan más y más en los *casos duros* (hard cases). El primer clasificador o clasificador base es entrenado y utilizado para hacer predicciones en los datos de entrenamiento. El peso relativo de los registros mal clasificados se incrementa. El segundo clasificador es entrenado utilizando los pesos actualizados y también realiza predicciones en los datos de entrenamiento y los pesos son actualizados. Este proceso se repite hasta que se hayan entrenado todos los clasificadores. Debido a su naturaleza secuencial, este método no puede ser ejecutado en paralelo.

El método *Gradient Boosting* trabaja de manera similar al AdaBoost, de manera secuencial agrega predictores que intentan corregir a su predecesor. A diferencia del AdaBoost, en lugar de actualizar los pesos de los registros, Gradient Boosting intenta reducir el error residual obtenido del clasificador anterior. Al igual que un GD, este método requiere establecer el parámetro tasa de aprendizaje (learning rate).

2.4. Algoritmos de clusterización

El análisis de clúster es una herramienta exploratoria para organizar un conjunto de datos en grupos más manejables que su totalidad. Es un tipo de análisis no supervisado en el que no hay conocimiento previo sobre qué registros pertenecen a qué clúster. Cada registro miembro de un clúster es similar de algún modo al resto de registros en el mismo clúster y diferente a todos los registros miembros de otro clúster. La idea general de la clusterización es dividir el conjunto de datos en grupos específicos [46, 48].

2.4.1. Two-step clúster

El Two-step clúster de la International Business Machines (IBM), es un algoritmo diseñado para manejar grandes cantidades de datos. Una de sus principales ventajas es que puede analizar datos tanto continuos como categóricos al mismo tiempo. Este algoritmo también tiene la capacidad (opcional) de seleccionar automáticamente el número de clústers. Como su nombre lo indica se divide en dos pasos [49]:

1. Se realiza una preclusterización en el que los registros del conjunto de datos son asignados a pequeños subclústers. En este paso se recorren los registros del conjunto de datos uno por uno y se decide, en base al criterio de *distancia*, si el registro debe ser integrado con los clústers formados previamente o se debe iniciar uno nuevo.
2. En el segundo paso se toman los subclústers del paso anterior y se distribuyen en el número deseado de clústers. Debido a que el número de subclústers resulta mucho menor que la cantidad original de datos, se pueden utilizar métodos tradicionales de clusterización, en este caso se utiliza la *agrupación jerárquica aglomerante* (agglomerative hierarchical clustering).

En general, entre mayor sea el número de subclústers en el primer paso, el resultado final será más exacto, sin embargo, demasiados subclústers alentarán el segundo paso de clusterización.

Se debe tomar en cuenta que el algoritmo utiliza una versión modificada del árbol de características de clúster (CF) [51] que puede depender del orden de entrada del conjunto de datos y afectar el modelo construido y los resultados obtenidos, por lo que se recomienda ordenar los registros antes de construir el modelo. Además de los pasos comentados, el algoritmo cuenta con la opción de manejar los registros sesgados (outliers) de manera automática, se consideran datos sesgado si el número de registros en una hoja del árbol es menor a una fracción establecida (25 % por default).

Este algoritmo puede trabajar con dos tipos de distancia; la distancia euclidiana (solo variables continuas) y la distancia *log-likelihood*. Esta última tiene la capacidad de trabajar con variables continuas y categóricas y es una distancia basada en probabilidades. Utiliza la distribución normal para variables continuas y la multinomial para variables categóricas. Se asume que todas las características son independientes. La distancia entre el clúster i y j se define mediante las ecuaciones 2.26, 2.27 y 2.28.

$$d(i, j) = \xi_i + \xi_j - \xi_{\langle i, j \rangle} \quad (2.26)$$

$$\xi_v = -N_v \left(\sum_{k=1}^{k^A} \frac{1}{2} \log(\hat{\sigma}_k^2 + \hat{\sigma}_{vk}^2) + \sum_{k=1}^{K^B} \hat{E}_v k \right) \quad (2.27)$$

$$\hat{E}_v k = - \sum_{l=1}^{L_k} \frac{N_{vkl}}{N_v} \log \left(\frac{N_{vkl}}{N_v} \right) \quad (2.28)$$

Si $\hat{\sigma}_k^2$ se ignora en la ecuación 2.27, la distancia entre el clúster i con el clúster j es el decremento en log-likelihood cuando dos clústers son combinados. El término $\hat{\sigma}_k^2$ se agrega para resolver el problema de $\hat{\sigma}_{vk}^2 = 0$ que podría derivar en un logaritmo no definido (Por ejemplo cuando el clúster tiene solo un caso). K^A Es el total de características continuas. K^B es el total de características categóricas. L^K es el número de categorías para la k^{th} característica categórica. N es el total de registros. N_k es el total de registros en el clúster k . $\hat{\sigma}_k^2$ es la varianza estimada para la k^{th} característica continua en todo el conjunto de datos. $\hat{\sigma}_{vk}^2$ es la varianza estimada para la k^{th} característica continua en el clúster v . N_{vkl} es el número de registros en el clúster j donde la k^{th} característica toma la th categoría. N_{kl} es el número de registros en la k^{th} característica categórica que toma la th categoría.

Para poder calcular automáticamente el número de clústers el algoritmo se basa en dos criterios. El criterio conocido como Bayesian Information Criterion (BIC) y el Approximate Weight of Evidence (AWE) definidos en [52, 53], respectivamente.

Para evaluar la calidad de un modelo clusterizado podemos utilizar varios algoritmos, sin embargo el más extendido es el denominado coeficiente *Silhouette* que se basa en los conceptos de cohesión (qué tan compactos son los registros que pertenecen a un clúster) y separación (qué tan separados se encuentran los registros de clústers distintos) [49].

2.4.2. k-Means

K-means es probablemente el algoritmo de clusterización más utilizado. La idea principal del algoritmo k-means es que, a partir de un grupo de registros n , se defina un número de clústers k . Cada uno de los k clústers tiene un centro (algunas veces llamado *media*) que se tomará como referencia para calcular la distancia al resto de los n registros. Los clústers serán actualizados iterativamente en base a la distancia de los objetos. Posteriormente se recalcula el centro de cada clúster. El algoritmo se detiene cuando se ha llegado al punto donde el centro del clúster permanece fijo.

El número de clústers k se define antes de iniciar el algoritmo. La meta es minimizar las diferencias dentro de cada clúster y maximizar las diferencias entre clústers. El algoritmo inicia con un proceso heurístico en el que se asignan, de manera aleatoria o mediante expertos en el tema, los elementos de cada clúster⁸.

El cálculo de la similaridad se realiza bajo el concepto de *distancia*. El tipo de distancia más utilizado es la euclidiana (ecuación 2.29), pero también es posible utilizar la Manhattan o la Minkowski [46].

⁸Debido a la naturaleza heurística del algoritmo, es posible obtener resultados muy diferentes con un pequeño cambio a las condiciones iniciales. En [59] se propone el algoritmo denominado k-means++ con el que se propone una alternativa para disminuir el impacto de dichos cambios.

$$\text{dist}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (2.29)$$

Donde n es el número total de características y $\text{dist}(x, y)$ es la distancia entre el registro x y el registro y .

2.5. Técnicas de balanceo de clases

Existen dos problemas fundamentales que se presentan con conjuntos de datos desbalanceados [50]:

- El impacto de algunas factores puede permanecer oculto.
- Se pueden observar resultados sesgados o engañosos.

Los métodos para tratar con datos desbalanceados se pueden agrupar en cuatro categorías: submuestreo, sobremuestreo, combinación de submuestreo - sobremuestreo y métodos ensamblados de aprendizaje. Los métodos de submuestreo eliminan observaciones de la clase mayoritaria con el fin de igualar los tamaños de las clases. Los métodos de sobremuestreo utilizan distintas técnicas para generar nuevos casos a la clase minoritaria. Por su parte, los métodos ensamblados se pueden emplear para mitigar la pérdida de información cuando se utilizan métodos de submuestreo [54]. Un ejemplo de método de sobremuestreo sintético es el *Synthetic Minority Over-sampling Technique* (SMOTE), esta técnica es una de las más populares, en ella, se introducen casos ficticios a lo largo de los segmentos de línea que unen a los vecinos más cercanos a la clase minoritaria.[55, 56]. En el artículo publicado por Wang et al. [58] se presenta una modificación de esta técnica conocido como P-SMOTE que está orientada a clases desbalanceadas en formato texto.

Otra técnica de sobremuestreo es la conocida como *sobremuestreo aleatorio*, como su nombre lo indica, se eligen aleatoriamente casos de la clase minoritaria y se replican hasta que sean suficientes para minimizar el desbalanceo, con respecto a la clase mayoritaria.

En esta tesis se utilizan métodos de submuestreo y sobremuestreo en distintas fases del proyecto. En el caso de sobremuestreo se propone una nueva técnica dirigida específicamente a datos en formato texto. Esta técnica se basa en la generación de texto sintético a través de RNN-LSTM (Sección 2.3.3).

Capítulo 3

Material y métodos

Este capítulo, se divide en tres partes: la caracterización de las lesiones por causa externa, el análisis de subregistro por códigos inespecíficos y el balanceo de datos y elección de modelo para el tiempo de estancia del paciente en el servicio de urgencias. Cada fase esta relacionada con un objetivo específico (ver sección 1.5).

Se incluyen registros de pacientes con diagnóstico relacionado a lesiones por causa externa, ambos sexos, de 14 a 99 años de edad que asistieron al servicio de urgencias entre septiembre de 2010 y mayo de 2015, estos registros fueron capturados en cinco áreas del servicio: Triage, Admisión, Consultorios, Observación y Choque. Los casos se categorizaron con base a las claves del CIE-10 y según lo descrito en el trabajo de Ávila-Burgos et al. [18]. La base de datos original contiene 119,403 registro y se dividió de acuerdo a los grupos de edad definidos por el Instituto Nacional de Estadística y Geografía (INEGI) [60].

Para contar con el permiso de acceso a los datos, se realizó el trámite de registro, evaluación y aprobación del protocolo de investigación por parte del Comité de Ética en Investigación del Centro Médico “Lic. Adolfo López Mateos”.

3.1. Caracterización de lesiones por causa externa

Desde el punto de vista médico, esta fase pretende describir la epidemiología y los principales rasgos de las lesiones por causa externa. Se busca delinear las características sociodemográficas, requerimientos hospitalarios, tipos de causas que originan las lesiones, sus principales consecuencias y diagnósticos, lugar donde se originan, así como las áreas corporales afectadas. Así mismo, desde el punto de vista de la detección de pacientes con riesgo de estancia prolongada, esta sección nos permite analizar el conjunto de datos, tipos de variables y estadística descriptiva, posibles sesgos, valores nulos o poco verosímiles, también se realizó un análisis de clúster para hacer una descripción más completa de las variables involucradas y comenzar a evidenciar el tipo de herramientas que posibiliten el cumplimiento del objetivo principal.

Cuando fue necesario, los datos nulos fueron tratados con imputación por promedio para las variables numéricas y a las variables categóricas se les agregó la categoría denominada *dato no registrado* [56].

Para esta fase se consideraron 16,567 registros relacionados a LCE; se realizó estadística descriptiva, medidas de asociación entre características de las lesiones y pacientes (X^2 , ANOVA) y un análisis multivariable de clúster (TwoStep clúster de IBM [49]). Las variables consideradas incluyen diagnóstico, tipo de causa externa, edad, sexo, si el paciente requirió hospitalización, estado civil, escolaridad, lugar de residencia, sitio de ocurrencia, área afectada, procedimiento, medicamento administrado, entre otras.

3.2. Análisis de subregistro por códigos inespecíficos

Como se describe previamente, el problema de los códigos inespecíficos es común en varias áreas, no solo en el cuidado de la salud. El presente apartado, tuvo como finalidad determinar si la cantidad de lesiones por causa externa, pudiera estar afectada por el registro de códigos inespecíficos. Esta indagación es de trascendencia en el ámbito médico, ya que puede contribuir con información significativa a la hora de planificar el uso de recursos y en la creación de políticas públicas. En otro orden de ideas, es importante debido a que permite el uso de modelos de clasificación aplicados a los datos involucrados en la investigación, dando una visión de qué algoritmos y variables se pueden utilizar para solución del problema planteado. También permite intuir si es posible contar con más casos de los obtenidos anteriormente, esto es valioso porque algunos algoritmos de clasificación se comportan mejor cuando hay más datos disponibles. En esta etapa, se propone el uso de notas médicas redactadas en texto libre como predictores alternativos a los habitualmente utilizados basados en datos estructurados.

Partiendo de los resultados anteriores, en esta fase el énfasis se dirigió a dos vertientes: el subregistro del código inespecífico de *Dolor Agudo (R52.0)* y en la comparación de los modelos entrenados con datos estructurados contra los basados en datos de texto. Para realizar las pruebas, se contemplaron 119,220 registros que se dividieron en diagnósticos relacionados a *Lesiones por Causa Externa* (19,230, 16.13%), *Dolor Agudo* (15,859, 13.30%) y *Otros Diagnósticos* (84,131, 70.57%) (Tabla 3.1). Para estimar el subregistro se crearon cuatro modelos predictivos: 2 regresiones logísticas (LR), una para datos estructurados y una para datos de texto, un modelo de árbol de decisión (DT) y uno de Boosting Tree (BT), los dos últimos para datos estructurados.

Debido a que las clases formadas están desbalanceadas, fue necesario emplear un método de balanceo. El método elegido fue el submuestreo aleatorio que tiene la ventaja ser sencillo de implementar y utiliza solo datos reales a diferencia de los métodos de sobremuestreo, sin embargo, debido a la inherente pérdida de información, este método no se recomienda cuando se dispone de pocos datos [61, 62]. El parámetro de evaluación elegido fue la *exactitud* del modelo.

Cada modelo fue sujeto a un método de validación y ajuste de parámetros, dependiendo de

Tabla 3.1: Distribución de los datos según la etiqueta asignada para el análisis de subregistro por códigos inespecíficos.

Etiqueta	n	%
Lesión por Causa Externa (LCE)	19,230	16.13 %
Dolor Agudo (DA)	15,859	13.30 %
Otros diagnósticos (OTRO)	84,131	70.57 %
Total	119,220	100.00 %

su propia naturaleza. Para la LR se eligió el método ridge regression, descrito en la sección 2.1.2, en el caso del modelo basado en DT, el parámetro ajustado fue la profundidad del árbol y, finalmente, el número máximo de iteraciones (número de árboles) fue el utilizado para ajustar el modelo basado en BT.

Como es habitual para su análisis, el conjunto de datos se dividió en tres subconjuntos, uno de entrenamiento (80%), uno de validación (10%) y uno de prueba (10%). Así mismo, los registros identificados con el diagnóstico de dolor agudo fueron separados del resto de los datos para su posterior clasificación. En la Tabla 3.2 se describen los modelos generados para esta sección.

Tabla 3.2: Resumen de modelos generados para el análisis de códigos inespecíficos.

Modelo	Variabes	Método	Entrenamiento	Validación	Prueba	Clasificación
M1	EST	LR	28648	3666	3526	13297
M2	EST	DT	28648	3666	3526	13297
M3	EST	BT	28648	3666	3526	13297
M4	TFIDF	LR	30498	3880	3738	14826

EST.- Datos estructurados; LR.- Regresión Logística, DT.- Árbol de decisión, BT.- Boosting Tree; TFIDF.- Frecuencia inversa de palabras

Los predictores elegidos para los modelos estructurados se extrajeron de las variables analizadas previamente. El criterio para su inclusión se basó en la opinión de expertos en ciencias de la salud, especialistas en traumatología y en la integridad inherente a la propia variable. A su vez, las notas médicas disponibles se forman de varios campos redactados en texto libre que son: motivo de la consulta, antecedentes, padecimiento actual, exploración física, tratamiento y plan, observaciones, estudios realizados y pronóstico. Para el estudio propuesto, estas variables fueron concatenadas y utilizadas como un solo campo de texto. En esta sección la representación del texto para su clasificación fue TFIDF calculado mediante la ecuación 2.9.

Finalmente, coincidiendo con los datos estructurados, también hubo necesidad de preprocesamiento para las notas médicas. Para eliminar errores ortográficos, se generó un listado de palabras, tomando como base a un glosario de términos médicos y un diccionario de la

lengua española. El texto de cada nota fue filtrado, palabra por palabra, para determinar si existía en el listado. En caso de no estar presente, dicha palabra era descartada. También se eliminaron las palabras poco relevantes (con, que, el, de, para, etc.).

Para la programación, tratamiento de datos y entrenamiento de los modelos generados en esta sección se utilizó el lenguaje de programación Python 2.5 [63], el entorno de programación Jupyter 1.0.0 [64] instalado mediante Anaconda 4.3.17 [65] y la librería GraphLab Create v2.1 [66]. Las pruebas fueron realizadas en un servidor Dell con un procesador Intel Xeon con 4 núcleos y velocidad de reloj de 2.1 GHz, 8 GB de memoria RAM y sistema operativo Linux Fedora 23 [67].

3.3. Elección de modelo y balanceo de clases

Esta es la última etapa del proyecto. Aquí se propone el uso de notas médicas en formato de texto libre como predictores del tiempo de estancia en el servicio de urgencias. De igual manera que en la fase anterior, se presume que la predicción oportuna de pacientes con riesgo de estancia prolongada puede contribuir con información relevante para la asignación de recursos y creación de políticas públicas, además de coadyuvar en la administración del servicio y en la detección de posibles complicaciones en el tratamiento, proporcionando bases para mejorar la calidad en la atención brindada.

La variable objetivo en esta etapa es la duración de la estancia en el servicio de urgencias, ya sea normal o prolongada. Al no estar explícitamente capturada, es necesario derivarla de otras variables disponibles. Según las políticas hospitalarias, el procedimiento administrativo vigente en el periodo de atención considerado, indica que el primer contacto del paciente que ingresa al nosocomio vía el servicio de urgencias es en el área de Triage, donde un médico especialista lo examina, evaluando la gravedad de su padecimiento; por lo tanto se elige como *ingreso* la fecha y hora de registro en esta área. Por otro lado, entre los datos disponibles tampoco se encuentra el registro de la fecha y hora en la que un paciente egresa del servicio, motivo por el cual se elige como *egreso* el momento en que se registra la última nota médica realizada, que normalmente es la denominada *nota de egreso*.

La variable objetivo se construye obteniendo la diferencia entre la fecha y hora de ingreso y la fecha y hora de egreso, dadas las características de la atención en urgencias este valor está dado en horas. Siguiendo las recomendaciones del *Manual de Procedimientos del Servicio de Urgencias para Hospitales Generales* del Instituto de Salud del Estado de México [16], donde se recomienda que un paciente no exceda las ocho horas en el servicio, el conjunto de datos se divide en dos subconjuntos: los casos que presentan estancias normales (menor o igual a ocho horas) y aquellos con estancias prolongadas (mayores a ocho horas) de esta forma se obtiene una variable dicotómica utilizada en la clasificación binaria requerida para el cumplimiento del objetivo planteado.

Los resultados obtenidos en la segunda etapa del proyecto y presentados en la sección 4.2 muestran que las clasificaciones realizadas utilizando datos clínicos en formato texto son viables en la clasificación binaria y cuentan con la ventaja de no depender de la estructura de

la base de datos. Como consecuencia de lo anterior, se opta por el uso de datos en formato texto para esta sección. No obstante, siendo un problema de clasificación binaria desbalanceado es necesario, previo a la construcción del modelo final, abordar la solución a esta dificultad. El problema de los datos no balanceados está asociado con sesgos en la clasificación, donde la clase minoritaria tiende a ser mal clasificada en comparación con la clase mayoritaria [55].

En la segunda fase de este trabajo (sección 3.2), cuando se utilizó texto para detectar casos de subregistro por códigos inespecíficos, se llevó a cabo el entrenamiento de modelos para una clasificación binaria. En ese contexto el *submuestreo aleatorio* fue el método elegido para balancear las clases. La cantidad de registros disponibles para la clase minoritaria es el factor que marca la diferencia con el problema de esta fase. Previamente, había un total de 84,131 (81 %) y 19,230 (19 %) registros para las clases mayoritaria y minoritaria, respectivamente. Por su parte, en el conjunto de datos para esta etapa fueron incluidos 18,232 registros. La clase mayoritaria, que representa la estancia normal, consta de 14,760 registros (81 %), mientras que 3,472 casos componen la clase minoritaria, relativa a la estancia prolongada (19 %). Si bien, los porcentajes en ambos problemas coinciden, la cantidad de registros para la duración de la estancia es significativamente menor. Una de las mayores desventajas al utilizar submuestreo aleatorio es que se corre el riesgo de eliminar datos que pueden ser importantes durante el proceso de entrenamiento, por lo tanto esta técnica se recomienda cuando los conjuntos de datos cuentan con gran cantidad de registros [57].

Existen distintos métodos para el balanceo de clases, la mayor parte dirigidos a datos estructurados, algunos modificados para datos en formato texto [58], sin que exista un estándar para abordar el problema. En este apartado se propone el uso de Redes Neuronales Recurrentes de tipo Long short-term memory (RNN-LSTM) para la generación de *texto sintético* que puedan ser utilizados para el balanceo de clases.

3.3.1. Generación de texto sintético

La generación de texto a través de RNN-LSTM se presenta en el curso denominado *Deep Learning Nanodegree Program* de la plataforma Udacity [68]. Este método ha sido propuesto para la generación de texto nuevo a partir de texto disponible que se toma como base. El proyecto original utiliza diálogos de la serie de televisión *los Simpson*, para generar otras conversaciones *similares* entre los personajes¹. Basado en la misma metodología, Zack Thoutt propone la *escritura* del tomo seis de la serie de libros *Juego de Tronos* [69].

Por su forma de trabajo, la RNN-LSTM requiere de dos parámetros: la longitud del texto que será generado (medida en palabras por registro) y la palabra inicial de cada texto sintético.

En el diagrama de la Figura 3.1 se presenta un resumen de los pasos para la generación del texto². La parte derecha del diagrama corresponde a los pasos descritos en el curso. Por otro lado, su contra-parte izquierda conforma la adaptación realizada con propósitos de este

¹El código base para la construcción y entrenamiento de la red está disponible en [70].

²Al ser parte de un curso en línea, se motiva a los participantes a compartir *su* solución al ejercicio, por lo que se pueden encontrar diversas implementaciones en Internet (e. g. [71] y [72]). En este proyecto se realiza una implementación propia que, al momento de escribir este documento, no se encuentra en línea.

proyecto y se refiere a la obtención de los parámetros de entrada necesarios para la generación del texto sintético.

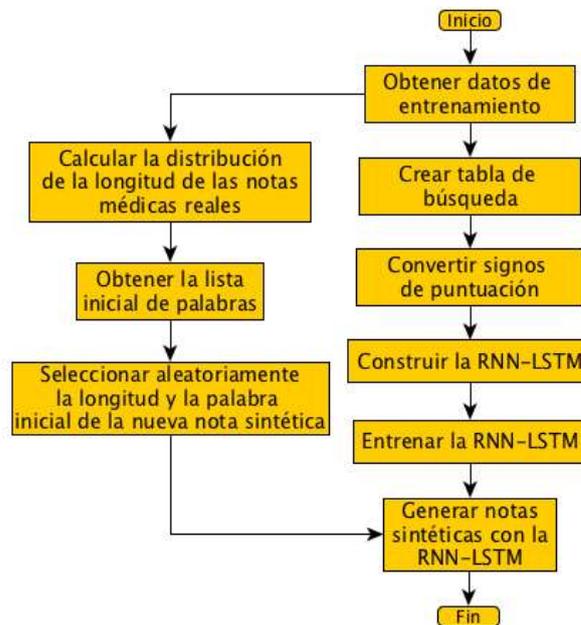


Figura 3.1: Proceso para generar notas sintéticas a través de una RNN-LSTM

El primer paso listado en el diagrama alude simplemente a la carga de los datos al entorno de programación. En la sección marcada como *Crear tabla de búsqueda* se definen dos diccionarios uno para convertir cada palabra a un identificador (*id*) y el segundo para la operación inversa. Posteriormente se procede a *Convertir signos de puntuación*. “Los signos de puntuación como puntos o signos de exclamación incrementan dificultad a la red neuronal, para distinguir entre elementos del texto como *bye* y *bye!*” [70]. Se escribe una función para convertir los signos tales como *!* en palabras de la forma `||Exclamation_Mark||` agregando su correspondiente espacio al inicio y fin de cada palabra.

La parte izquierda del diagrama describe la obtención de los parámetros para generar el texto sintético. Primeramente, para calcular la longitud del texto, se analizó la cantidad de palabras en cada registro del conjunto de datos original de la clase minoritaria (e.g. para el conjunto de notas médicas preprocesadas: $\bar{x} = 115.46$, $SD=46.48$). A partir de estas características, se generó un conjunto de números con una distribución normal. Finalmente, partiendo de este conjunto se eligió aleatoriamente la longitud de cada registro generado como texto sintético.

En el siguiente paso, para obtener la primer palabra para cada registro sintético, se formó un vector con las palabras de inicio de todos los casos disponibles en el conjunto de datos original de la clase minoritaria. Para la generación de cada nota se extrajo, de manera aleatoria, una palabra del vector y se utilizó como palabra inicial para el nuevo texto sintético.

Para este trabajo, los valores asignados a los parámetros de la red, son los mostrados en la Tabla 3.3, estos valores caen en el rango recomendado por los autores y alumnos del curso

mencionado, sin embargo, no existe en la literatura un parámetro que indique cuando un texto es mejor que otro, esto sería necesario en tareas futuras de optimización, buscando *mejorar* el texto con miras al balanceo y la clasificación.

Tabla 3.3: Parámetros utilizados en la RNN-LSTM para la generación de notas sintéticas.

Parámetro	Valor
Número de épocas (ciclos)	6
Tamaño de Batch	100
Tamaño de la red neuronal	256
Longitud de secuencia	10
Factor de aprendizaje	0.01

Se generó un conjunto de notas sintéticas tomando como base las clases minoritarias de los conjuntos de datos disponibles. La construcción y el entrenamiento de la RNN-LSTM se realizan utilizando la librería TensorFlow 1.0 [73] y el lenguaje de programación Python 3.5 [63]. Las notas sintéticas generadas se almacenaron en una base de datos PostgreSQL [74] para su posterior integración al conjunto de datos original. Finalmente, el total de nodos en capa de salida de la red neuronal es igual al número de palabras contenidas en *todas* las notas médicas que dependerá del conjunto de datos de entrenamiento.

El diagrama de la Figura 3.2 describe el algoritmo para entrenar, validar y probar cada modelo generado. Por simplicidad el proceso se describe para el algoritmo de LR. Para el resto de los algoritmos los pasos a seguir son similares, cambiando el parámetro de validación.

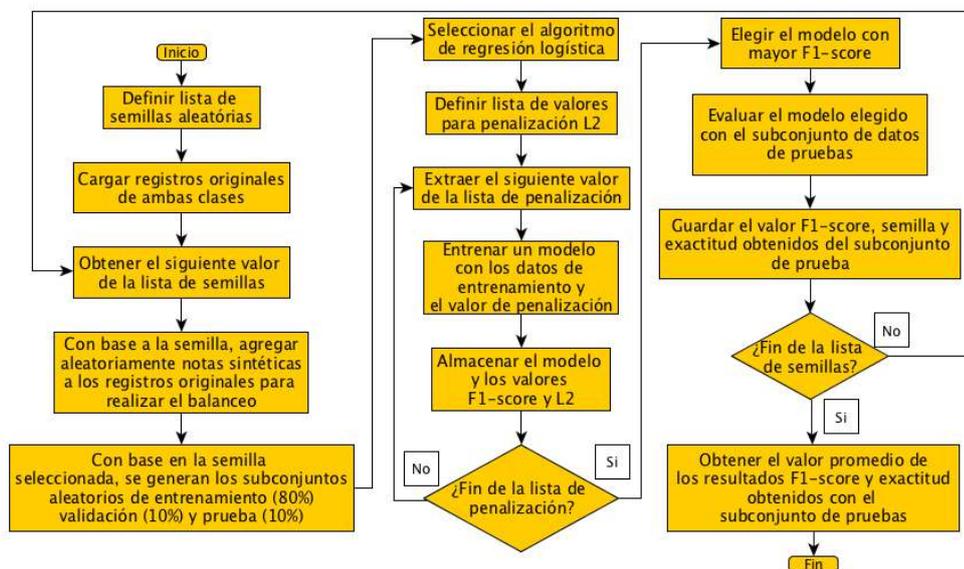


Figura 3.2: Procedimiento utilizado para obtener el valor F_1 -score promedio de los modelos generados con el algoritmo de regresión logística.

El proceso inicia definiendo un vector de semillas aleatorias, cada una de ellas fue utilizada

para dos tareas: la elección de notas sintéticas para la tarea de balanceo de clases y para la generación de los subconjuntos de entrenamiento (80 %), validación (10 %) y prueba (10 %). Con objeto de tener suficientes datos para evaluar estadísticamente el rendimiento de los modelos, se eligieron 33 semillas distintas. La siguiente tarea es la validación mediante la *penalización L2* o *ridge regression* descrito en la sección 2.1.2. Para los algoritmos SVM, BT, DT y RF el proceso es similar, cambiando el parámetro de validación según corresponda. En este proceso también se define un vector de valores para la validación. Con cada valor se entrenan y validan los modelos con su respectivo subconjunto aleatorio de datos. Se calcula el valor de la media armónica F_1 -score y se almacena. El proceso se repite hasta que se han utilizado todos los valores del vector de validación. En este momento se extrae el modelo que generó el F_1 -score más alto en el conjunto de validación y se utiliza para ser probado con el subconjunto separado para tal fin (subconjunto de prueba). Se calcula el valor F_1 -score, se almacena y se repite el proceso con la siguiente semilla de la lista. Cuando se han utilizado todos los valores aleatorios, se tienen los 33 mejores modelos con las diferentes representaciones aleatorias de los datos, todos debidamente validados y probados. Al calcular la media de estos valores se obtienen las tablas 4.9, 4.10, 4.11, y las gráficas de las Figuras 4.11, 4.12, 4.13, 4.14 mostradas en los resultados de la sección 4.3 haciendo las modificaciones pertinentes para variar los métodos de balanceo, conjuntos de datos y algoritmos de clasificación. En resume, los modelos generados siguiendo los pasos descritos anteriormente son combinaciones de los criterios listados a continuación.

- Semillas aleatorias (33).
- Conjuntos de datos.
 - Notas médicas originales
 - Notas médicas preprocesadas
 - Amazon
 - PubMed
- Algoritmos de clasificación.
 - Boosting Tree
 - Árboles de decisión
 - Regresión Logística
 - Random Forest
 - Máquinas de Soporte Vectorial

3.3.2. Evaluación de los modelos

Para comprobar el rendimiento del método de desbalanceo propuesto, se ejecutan distintas pruebas; en primera instancia, se contrastaron los resultados contra los obtenidos mediante modelos entrenados con los datos originales desbalanceados y, posteriormente, con otros

métodos de balanceo, tales como el sobremuestreo aleatorio y el SMOTE. También se realizan pruebas con cinco algoritmos de clasificación distintos: Regresión logística (LR), Árboles de decisión (DT), Boosting Tree (BT), Random Forest (RF) y Máquinas de Soporte Vectorial (SVM). Finalmente, buscando la generalización de los resultados, se realizan pruebas con las notas sin preprocesar y se incluyen además, dos conjuntos de datos de texto libre (en inglés). Estos conjuntos de datos se buscaron con fines de evaluación del método de balanceo por texto sintético. El único criterio para elegirlos fue la posibilidad de realizar clasificación de registros mediante el análisis de texto y que todos los registros de se encuentren en el mismo idioma (para ambos casos, el inglés).

El primer conjunto contiene registros acerca de comentarios de satisfacción sobre productos de Amazon [75], la calificación a cada producto varía en el rango entre 1 y 5, siendo uno la calificación más baja y cinco la mejor. Se crean dos clases, una para *sentimientos negativos* (calificaciones 1 y 2) y una para *sentimientos positivos* (calificaciones 4 y 5), los registros con calificación 3, que representa una calificación neutral, son descartados. En este caso, los sentimientos positivos corresponden a la clase mayoritaria ($n=29,309$, 89%), mientras que las sentimientos negativos representan la clase minoritaria ($n=3,588$, 11%).

El segundo conjunto de datos contiene resúmenes de artículos de investigación almacenados en la base de datos PubMed [76]. Este conjunto consta de textos redactados en distintos lenguajes, sin embargo, para este apartado solo se consideran los artículos escritos en inglés. Se plantea la identificación de los trabajos relacionados al área de odontología. La clase mayoritaria se etiqueta como *otros* ($n=15,285$, 99%), mientras que la minoritaria se etiqueta como *odontología* ($n=87$, 1%). Analizando la distribución de las clases, se puede apreciar que este caso de estudio es que tiene mayor grado de desbalanceo.

El método SMOTE no se considera adecuado cuando se dispone de conjuntos de datos con alta dimensionalidad [77]. Por otro lado, los datos en formato texto suelen generar conjuntos con estas características; en este trabajo, el único conjunto de datos que fue posible balancear utilizando esta metodología fue el de notas médicas preprocesadas. El tratamiento previo permitió una reducción inherente en las dimensiones del conjunto de datos original pasando de un promedio de $\bar{x} = 99730.33$ (SD=8,961.81) palabras a un promedio de $\bar{x} = 4,487.40$ (SD = 83.79). Aunque se intentó realizar pruebas con los otros conjuntos de datos, los programas no fueron capaces de concluir³.

El diagrama de la Figura 3.3 se describe el proceso general para la partición del conjunto de datos, el entrenamiento, validación y prueba de los modelos. El primer paso se refiere a la carga del conjunto de registros en formato texto al entorno de programación. El preprocesamiento de los datos de texto se describe anteriormente en la sección 3.2, y se refiere a descartar palabras mal escritas, no relevantes y números. Este mecanismo se realiza en los tres conjuntos de datos analizados (notas médicas, Amazon y PubMed). Puesto que el texto en Amazon y PubMed están redactados en inglés, se utilizan las librerías disponibles en el paquete Graphlab [66]; al no existir una versión para el idioma español, para las notas médicas este preprocesamiento se realiza de forma manual, como se ha explicado previamente en la sección referida. Para efectos de comparación, también se realizaron pruebas con las notas

³Para realizar el balanceo mediante SMOTE, se utilizó la librería scikit-learn [78]

médicas originales sin ser sometidas a este procedimiento a las que se ha nombrado *notas médicas originales*.

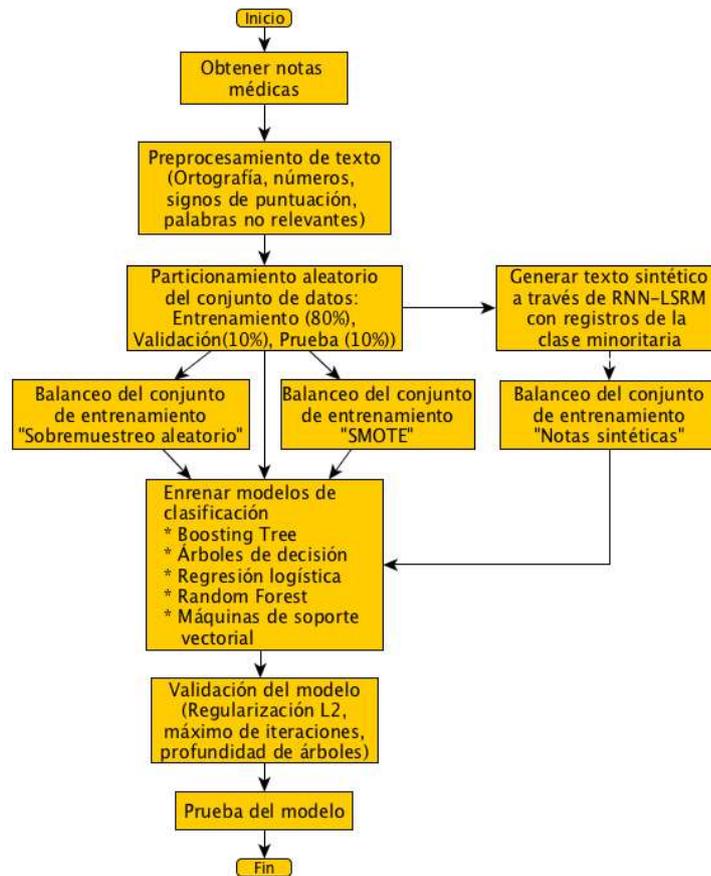


Figura 3.3: Proceso para entrenamiento y prueba.

El balanceo propiamente dicho es la siguiente fase del procedimiento. En el diagrama de la Figura 3.3, el cuadro que hace referencia a la generación de texto a través de RNN-LSTM fue descrito previamente cuando se explicó la Figura 3.1. Así mismo, el proceso de validación se describió en la explicación de la Figura 3.2. Para todos los casos, el balanceo se realiza sobre el conjunto de *entrenamiento*. Como se puede apreciar en el diagrama, se realiza una serie de pruebas utilizando los datos sin balancear con fines de comparación de los modelos.

El siguiente paso es encontrar los valores para los parámetros de cada algoritmo que proporcionen los mejores resultados en los conjuntos de entrenamiento y validación, para ello, se realiza el proceso de validación o penalización el cual dependerá del algoritmo elegido. Finalmente, la evaluación final de cada modelo se realiza con el conjunto de datos de prueba.

Como se ha comentado el proceso completo se realiza 33 veces para cada combinación de conjunto de datos, método de balanceo y algoritmo de clasificación. Para cada una de ellas, también se realizó una prueba de Kolmogorov-Smirnov para verificar la normalidad y comprobar que se cumplen los requisitos para la aplicación de las pruebas estadísticas que se describen a continuación.

Para comparar el rendimiento de los métodos de balanceo se realizó un análisis estadístico. El parámetro elegido para la comparación es el F_1 -score calculado mediante la ecuación 2.8. El primer objetivo es determinar si el método propuesto para balanceo de clases es una alternativa viable y mejora el valor F_1 -score con respecto a los modelos entrenados con clases desbalanceadas. La prueba *t-test* fue utilizada para confrontar los valores de rendimiento entre los modelos obtenidos con los datos originales (desbalanceados) y los tres métodos de balanceo analizados, uno a la vez. La prueba estadística ANOVA de un factor fue aplicada para contrastar la media de los valores F_1 -score de cada uno de los métodos de balanceo en conjunto. En caso de presentarse diferencias significativas en las comparaciones mediante la prueba ANOVA, se realizaron pruebas de *Bonferroni and Scheffe* para hacer el análisis uno a uno de los métodos comparados. Todos los análisis estadísticos realizados en esta sección se ejecutaron con el software de IBM Statistical Package for the Social Sciences (SPSS) [79].

Para explicar el uso de técnicas aleatorias utilizadas para elegir los registros provenientes de cada conjunto de datos, nuevamente se hace referencia al diagrama de la Figura 3.3. Como se muestra en dicho procedimiento, para cada modelo se realiza una partición aleatoria del conjunto de datos en base a porcentajes. Para asegurar que cada algoritmo cuente con los mismos datos seleccionados aleatoriamente y que estos sean distintos para cada modelo generado, se aplican distintos valores de *semillas*⁴. En este contexto, una semilla es utilizada para elegir subconjuntos aleatorios de entrenamiento que garanticen que, para cada serie de modelos entrenados con un mismo algoritmo, los registros elegidos aleatoriamente sean distintos, proceso conocido como método aleatorio de validación cruzada (para más información referirse a la sección 2.1.3). Por otro lado, también asegura que se utilice exactamente el mismo conjunto aleatorio para comparar los modelos predictivos, afianzando de esta manera, la validez de los experimentos. En total se generaron 60,000 textos sintéticos para cada uno de los cuatro conjuntos de datos analizados y solo se tomaron aleatoriamente las notas necesarias para el balanceo del modelo en turno.

3.3.3. Elección del modelo

Los resultados obtenidos a partir de estos experimentos determinaron el modelo predictivo elegido para detectar los pacientes con probabilidades de estancias prolongadas. Se consideraron tres criterios a elegir. El primero se refiere a la representación de los datos, se elige entre el texto original o el preprocesado (descartando faltas de ortografía, abreviaturas, números o palabras no relevantes). El segundo criterio es la elección del algoritmo de clasificación; se evalúa el rendimiento de cinco algoritmos, BT, DT, LR, RF y SVM. El último criterio es la elección del método de balanceo. Se evaluó si el método propuesto es viable para la situación planteada y se comparó con el sobremuestreo aleatorio y el método SMOTE. La combinación de los tres criterios descritos que presentó el valor promedio F_1 -score más alto, fue el modelo elegido.

⁴“La mayor parte de los generadores de números aleatorios son, en realidad, pseudoaleatorios; se calcula (o introduce internamente) un valor x_0 , que llamaremos semilla, y, a partir de él, se van generando x_1, x_2, x_3, \dots Siempre que se parta de la misma semilla, se obtendrá la misma secuencia de valores” [80].

Capítulo 4

Resultados

Esta sección se divide en tres partes. En ellas se describen los resultados obtenidos en función de cada uno de los objetivos propuestos.

La primer etapa engloba los análisis descriptivos de los datos utilizados. Ésto, además del interés médico inherente, también favorece la elección de variables y sustenta los análisis subsiguientes. La segunda etapa presenta los resultados del análisis de subregistro por códigos inespecíficos. Se presentan las predicciones de casos reales estimando la pérdida de información como consecuencia de este fenómeno y se ejecutan comparaciones entre los datos estructurados y los datos de texto comparando sus estimaciones y presentando nuevas opciones para abordar el problema. Finalmente, en la última etapa se elige el modelo para la estimación del tiempo de estancia en el servicio de urgencias basado en datos con formato de texto libre. Para tal efecto, se aborda el tema del balanceo de clases para clasificación binaria, se exponen los resultados obtenidos mediante el método original propuesto y se compara con métodos tradicionales. Se presentan también resultados de experimentos con otros conjuntos de datos, buscando la generalización de los mismos.

4.1. Caracterización de Lesiones por Causa Externa

En relación con los datos y métodos descritos en la sección 3.1, los resultados obtenidos indican lo siguiente: el total de registros incluidos fue de 16,567 con pacientes de edades de entre 14 y 99 años ($\bar{x} = 37.70$ y $SD=17.28$) con el 69.2 % ($n=11460$) de sexo masculino. Se encontró que el 0.60 % ($n=100$) presentaron algún tipo de discapacidad y que el 1.45 % ($n=74$) de las mujeres se encontraba en gestación. En la Tabla 4.1 se presentan las características sociodemográficas de los pacientes de la muestra analizada.

Los requerimientos de hospitalización en base a la edad y de acuerdo a la causa externa que generó la lesión se resumen en la gráfica de la Figura 4.1. Se encontraron diferencias significativas entre los pacientes que requirieron hospitalización y los que no, para las caídas ($F=97.254$ $p=0.000$), accidentes de tránsito ($F=7.257$ $p=0.007$), agresiones interpersonales dentro del hogar ($F=9.889$ $p=0.002$) y agresiones fuera del hogar ($F=4.702$ $p=0.030$).

Tabla 4.1: Características sociodemográficas de los pacientes atendidos.

		Femenino		Masculino		$X^2(p)$
		n	%	n	%	
Edad	5-14	6	0.12 %	16	0.14 %	857.36(0.00)
	15-24	958	18.76 %	3531	30.81 %	
	25-34	1122	21.97 %	3109	27.13 %	
	35-44	880	17.23 %	2103	18.35 %	
	45-64	1223	23.95 %	2049	17.88 %	
	≥65	918	17.98 %	652	5.69 %	
Estado civil	Casado	2203	43.69 %	5166	45.39 %	891.27 (0.00)
	Divorciado	3	0.06 %	2	0.02 %	
	Separado	166	3.29 %	137	1.20 %	
	Soltero	1691	33.54 %	4680	41.12 %	
	Unión libre	420	8.33 %	1229	10.80 %	
	Viudo	559	11.09 %	167	1.47 %	
Escolaridad	Ninguna	576	12.01 %	727	6.54 %	251.69 (0.00)
	Primaria	814	16.97 %	1672	15.03 %	
	Secundaria	620	12.92 %	2140	19.24 %	
	Bachillerato	360	7.50 %	883	7.94 %	
	Superior	346	7.21 %	531	4.77 %	
	Otra	2081	43.38 %	5168	46.47 %	

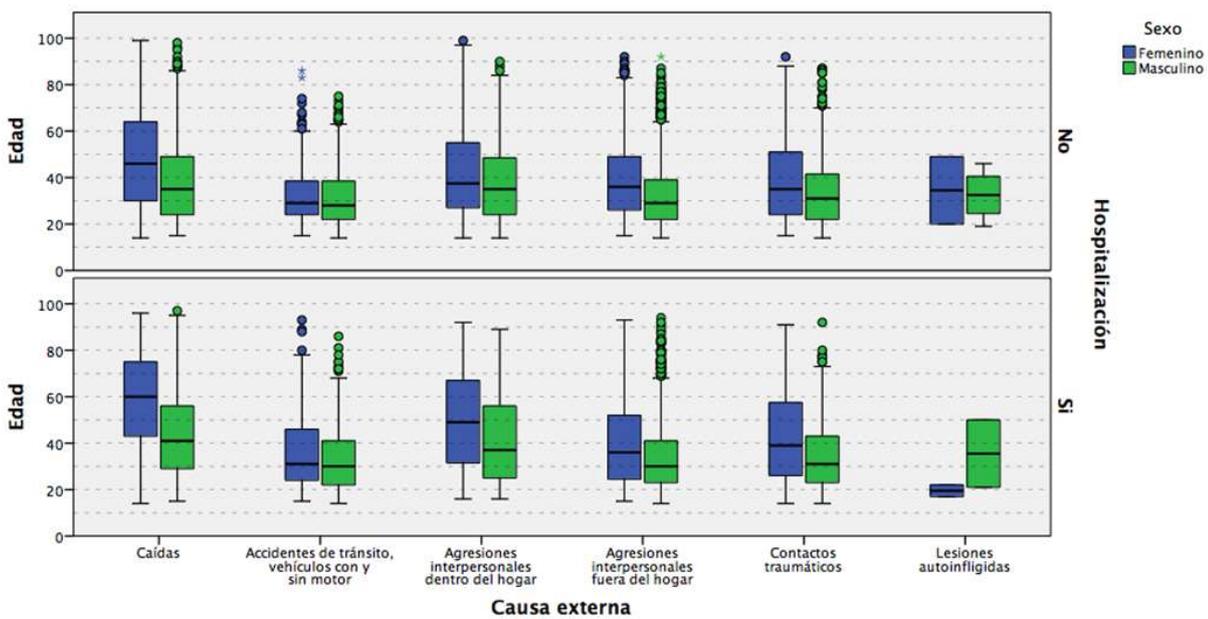


Figura 4.1: Cantidad de pacientes hospitalizados por LCE

El lugar de residencia de los sujetos atendidos se distribuye de la siguiente manera: el 98.12 % (n=16,256) de los registros fueron de pacientes con residencia dentro del Estado de México; 83.17 % (n=13,780) de zonas urbanas y 14.55 % (n=2,412) de zonas rurales; mientras que el 2.36 % (n=375) no fue registrado. El municipio de Toluca presentó la mayor frecuencia 45.67 % (n=7,567). En su zona urbana 96.99 % (n=7,339), la localidad con mayor prevalencia fue Toluca de Lerdo 83.73 % (n=6,145), seguido por San Pablo Autopan 4.03 % (n=296), San Mateo Otzacatipan 2.59 % (n=190) y San Andrés Cuexcontitlán 1.89 % (n=139). En la zona rural destacan Xicaltepec Tepaltitlán (Ejido San Lorenzo) 24.34 % (n=55), Ejido de Santa Cruz Atzacapozaltongo 12.39 % (n=28), San Diego Linares 7.96 % (n=18) y Tecaxic 6.64 % (n=15).

Los diagnósticos relacionados con las LCE fueron agrupados en ocho categorías con predominio de los traumatismos de cabeza y cuello 33.11 % (n=5,485), seguidos por traumatismos de extremidad superior 29.25 % (n=4,846) y traumatismos de extremidad inferior 15.87 % (n=2,630). La Tabla 4.2 muestra los requerimientos hospitalarios.

El grupo de causa externa más recurrente fue el de *agresiones interpersonales fuera del hogar* con el 32.72 % (n=5,420), seguidas por las *caídas* 24.98 % (n=4,139). Las causas externas que reportaron menores frecuencias son las *lesiones autoinfligidas* 0.06 % (n=10) y las *agresiones interpersonales dentro del hogar* con el 3.76 % (n=623). *Los accidentes de tránsito, vehículos con y sin motor* representan el 9.68 % (n=1,604) mientras que el grupo denominado otras causas externas de lesiones fue del 18.11 % (n=3,000). Los pacientes de sexo femenino tuvieron mayor prevalencia en el grupo de edad de 45-64 años, donde las caídas en el hogar fueron las más recurrentes.

El procedimiento que se registró en más ocasiones es la *consulta descrita como global* 95.13 % (n=15,760). Del 4.87 % (n=807) restante, el procedimiento *monitorización electrográfica* presentó la mayor frecuencia 28.87 % (n=233). También se halló asociación con las variables *consecuencia resultante* ($X^2=1,437.72$ p=0.000), *diagnóstico* ($X^2=976.22$ p=0.000), *destino del paciente* ($X^2=880.00$ p=0.000), *área de atención* ($X^2=447.48$ p=0.000) y con los grupos de *causas externas* ($X^2=942.26$ p=0.000).

En solo el 3.97 % (n=658) de los casos se registró el medicamento administrado. De éstos el *Omeprazol* fue el más utilizado 18.54 % (n=122) seguido del *Ketorolaco trometamina solución inyectable o ampollitas* con 12.77 % (n=84) y el *Midazolam solución inyectable 15 Mg.* con 12.61 % (n=83).

El principal sitio de ocurrencia de las LCE fue la vía pública que presentó un 26.33 % (n=4,362) seguida por el hogar con 23.67 % (n=3,922). Se encontró una asociación con respecto al sexo del paciente ($X^2=1,384.76$ p=0.000) donde las mujeres presentan mayor frecuencia en lesiones dentro del hogar 52.37 % (n=2,054). Sin embargo, los hombres las superan con más lesiones en la vía pública 75.26 % (n=3,283).

Se encontraron 1,604 accidentes de tránsito ($\bar{x}=33.11$, $SD=13.44$) de los cuales el 70 % (n=1,123) corresponde a pacientes de sexo masculino y el 63.9 % están dentro del grupo de edad de 15 a 34 años; el 22.25 % tenía educación básica (primaria y secundaria), el 44.07 % fueron sujetos casados y 44.70 % solteros.

Tabla 4.2: Pacientes que requirieron hospitalización después de su atención en urgencias.

		No hospitalizado		Nospitalizado		$X^2(p)$
		n	%	n	%	
Diagnóstico	Traumatismos de extremidad superior	3369	69.52 %	1477	30.48 %	479.40 (0.000)
	Traumatismos de cabeza y cuello	3291	60.00 %	2194	40.00 %	
	Traumatismos de extremidad inferior	1691	64.30 %	939	35.70 %	
	Traumatismos de tórax, abdomen y pelvis	954	46.58 %	1094	53.42 %	
	Otras lesiones	357	55.09 %	291	44.91 %	
	Quemaduras y corrosiones	311	46.84 %	353	53.16 %	
	Intoxicaciones	43	39.81 %	65	60.19 %	
	Lesiones múltiples	42	30.43 %	96	69.57 %	
Causa externa	Agresiones interpersonales fuera del hogar	3243	59.83 %	2177	40.17 %	221.08 (0.000)
	Caídas	2756	66.59 %	1383	33.41 %	
	Otras causas externas de lesiones	1695	56.50 %	1305	43.50 %	
	Contactos traumáticos	1149	64.88 %	622	35.12 %	
	Accidentes de tránsito, vehículos con y sin motor	775	48.32 %	829	51.68 %	
	Agresiones interpersonales dentro del hogar	434	69.66 %	189	30.34 %	
	Lesiones autoinfligidas	6	60.00 %	4	40.00 %	
Área afectada	Extremidades superiores	2133	72.06 %	827	27.94 %	1846.72 (0.000)
	Extremidades inferiores	1687	54.21 %	1425	45.79 %	
	Cabeza	1434	43.49 %	1863	56.51 %	
	Cara	1311	79.50 %	338	20.50 %	
	Mano	838	80.04 %	209	19.96 %	
	Cuello	549	81.82 %	122	18.18 %	
	Tórax	540	51.33 %	512	48.67 %	
	Columna vertebral	432	64.57 %	237	35.43 %	
	Otros	400	86.77 %	61	13.23 %	
	Pies	250	81.97 %	55	18.03 %	
	Múltiples	129	27.80 %	335	72.20 %	
	Abdomen	126	26.47 %	350	73.53 %	
	Pelvis	90	49.45 %	92	50.55 %	
	Espalda o glúteos	60	59.41 %	41	40.59 %	
	Se ignora	36	65.45 %	19	34.55 %	
	Región genital	24	60.00 %	16	40.00 %	
	Región ocular	19	73.08 %	7	26.92 %	

Del total de sujetos, el 60 % (n=963) recibieron algún tipo de atención prehospitalaria anterior a su ingreso al servicio de urgencias. El agente involucrado con mayor frecuencia en las lesiones de tránsito fue el automóvil (63.09 %; n=1012) seguido por bicicleta (15.84 %; n=254), motocicleta (11.35 %; n=182) y autobús (6.73 %; n=108). Sólo en el 12.78 % (n=205) se identificó el uso de cinturón de seguridad y de casco en el 0.81 % (n=13).

Las principales consecuencias resultantes de los accidentes de tránsito fueron: fractura (27.49 %; n=441), esguinces (20.07 %; n=322) y contusiones (16.27 % ; n=261). Mientras que las principales áreas afectadas correspondieron a la cabeza (26.93 %; n=432), extremidades inferiores (12.84 %; n=206) y cuello (12.47 %; n=200). La Figura 4.2 muestra la consecuencia resultante de acuerdo al tipo de vehículo involucrado en el accidente.

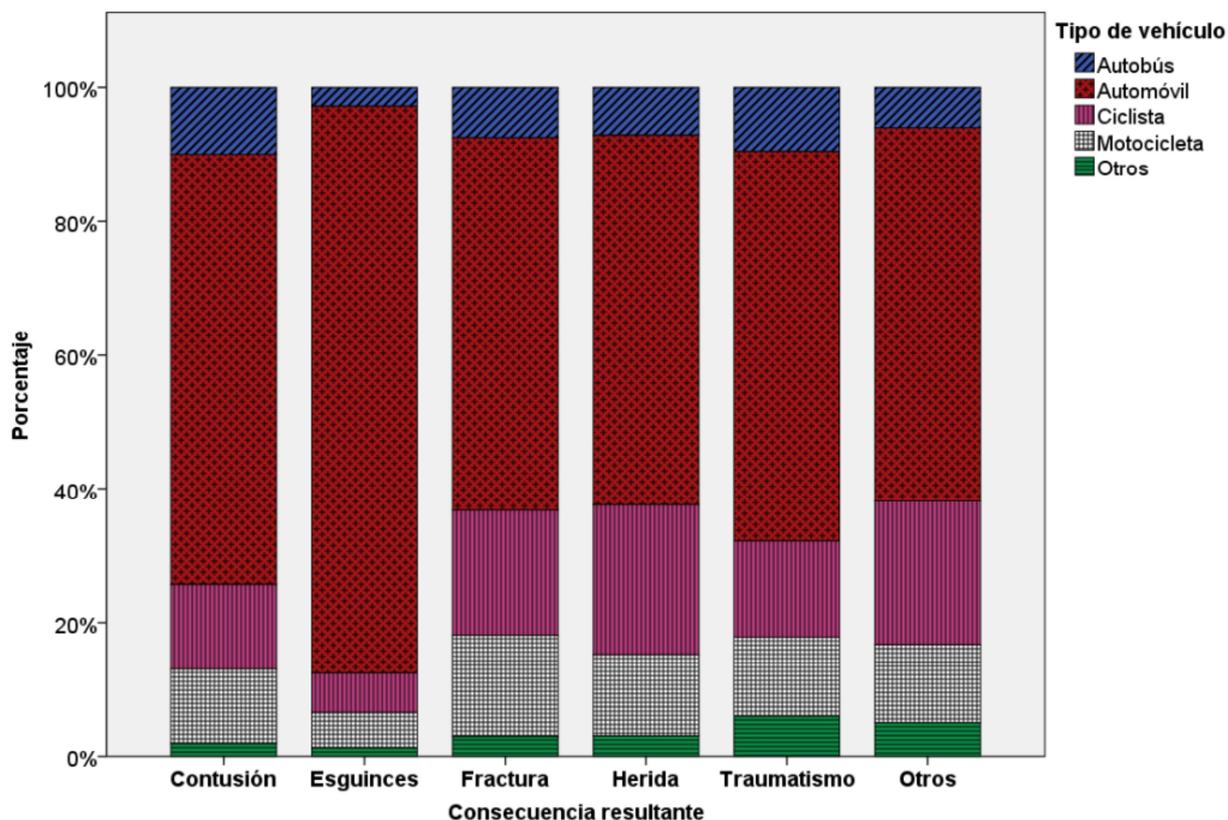


Figura 4.2: Consecuencia resultante de accidentes de tránsito, según el vehículo involucrado

El área de atención de los pacientes con LCE está asociada con el destino del paciente ($X^2=6,781.57$ $p=0.000$). El 83.67 % (n=1,680) de los casos ingresados en el área de choque y el 78.19 % (n=993) del área de observación requirieron posterior hospitalización. El 52.24 % (n=3,846) de los atendidos en consultorio fue enviado a su domicilio y el 43.99 % (n=2,607) de los atendidos en Traumatología fueron canalizados a consulta externa. El 6.01 % (n=24) de los sujetos atendidos en el área de choque se reportó como defunción.

En cuanto al día de ocurrencia, el 39.76 % (n=6,587) se registró en fin de semana. El domingo presentó la mayor prevalencia 21.02 % (n=3,483) y el martes la menor 11.12 % (n=1,843). El

3.00 % (n=497) de las lesiones se registró en día festivo¹.

Analizando incidencia de las LCE se encontró que la cantidad de registros refleja una tendencia a la baja en el periodo analizado, mientras que los registros del diagnóstico inespecífico *dolor agudo (R52.0)* aumentan en proporciones similares. La gráfica de la Figura 4.3 describe esta situación. En ella se aprecia la tendencia a través del tiempo de las LCE en comparación con el registro del código R52.0.

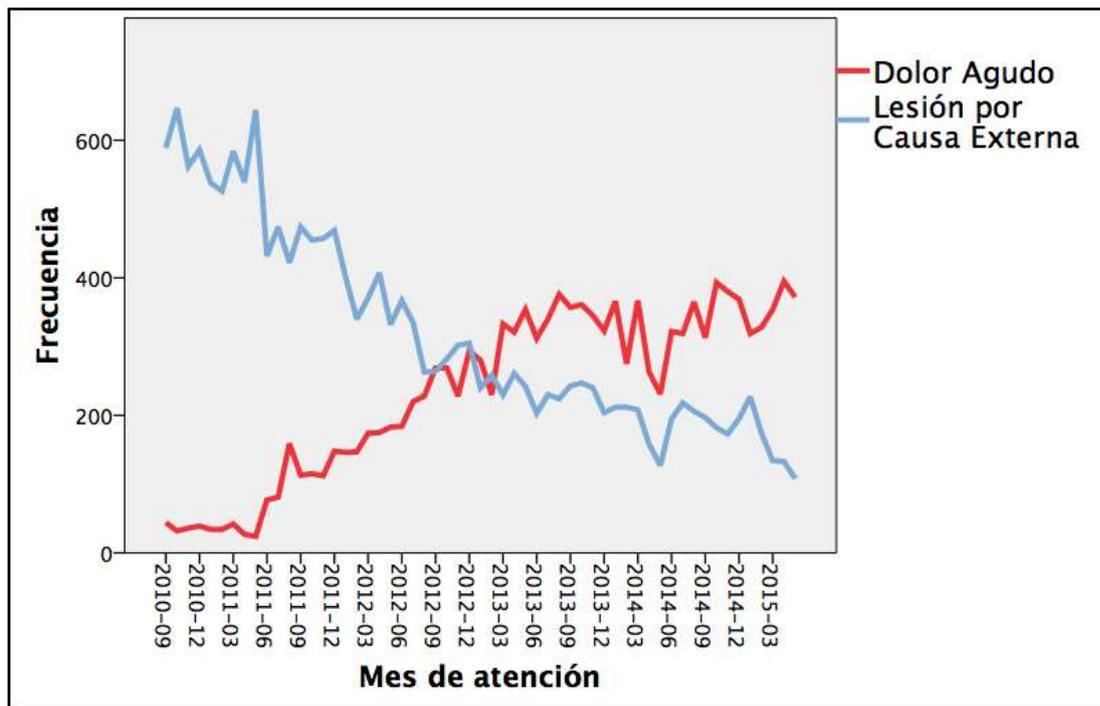


Figura 4.3: Frecuencia subestimada de lesiones por causa externa.

En el análisis de clúster ($Silhouette=0.3^2$) se incluyeron 22 variables cualitativas y 1 cuantitativa de esta manera se encontraron cuatro clústers (Tabla 4.3):

El clúster C1 32.70 % (n=5,417) mostró varones (78.40 %) jóvenes ($\bar{x}=34.13$ años) que acudieron por iniciativa propia (65.30 %) con lesiones en la cabeza (21.90 %) como resultado de agresiones interpersonales (27.70 %) en la vía pública (39.90 %). El 25.70 % se registró el domingo y su principal consecuencia son las fracturas (28.40 %), el área de atención fue el consultorio (49.60 %), y el 40.20 % requirieron hospitalización.

El clúster C2 26.30 % (4,363) mostró pacientes jóvenes ($\bar{x}=36.91$ años) de sexo masculino (73.40 %) que acudieron por iniciativa propia (69.30 %) con lesiones en la cabeza (18.70 %) como resultado de agresiones infligidas por otra persona (8.40 %) dentro del hogar (32.70 %).

¹Día de la independencia (16 de septiembre), revolución mexicana (20 de noviembre), constitución de 1917 (5 de febrero), día del trabajo (1 de mayo), natalicio de Benito Juárez (21 de marzo), año nuevo (1 de enero), Navidad (25 de diciembre).

²El **coeficiente Silhouette** es una medida de la cohesión y separación de los grupos encontrados. Los grupos deben estar separados entre sí mientras que sus elementos deben tener cohesión. $Silhouette=[0.2-0.4]$, se considera aceptable [49].

Tabla 4.3: Grupos encontrados mediante análisis de clúster.

Variable	Predictor	C1 32.7%(5417)	C2 26.3%(4363)	C3 15.9%(2640)	C4 25%(4147)	$x^2(p)$
AFE	1	Cabeza 21.90 %	Cabeza 18.70 %	Cabeza 30.20 %	Extremidades inferiores 35.80 %	3493.67 (0.000)
ATE	1	Consultorio 49.60 %	Consultorio 68.78 %	Consultorio 32.80 %	Traumatología 72.60 %	4063.24 (0.000)
CE	1	W504 27.70 %	W509 8.40 %	V499 26.90 %	W010 35.00 %	99402.00 (0.000)
CON	1	Fractura 28.40 %	Herida 34.90 %	Fractura 28.70 %	Fractura 51.60 %	4072.16 (0.000)
DIAG	1	S019 9.90 %	S098 7.70 %	S098 8.30 %	S934 6.30 %	11738.97 (0.000)
SOC	1	Peatón 39.90 %	Hogar 32.70 %	Vehículo 47.80 %	Hogar 47.10 %	7312.74 (0.000)
ATPR	0.98	35.30 %	31.10 %	63.60 %	20.00 %	1024.97 (0.000)
REF	0.93	IP 65.30 %	IP 69.30 %	UMSES 47.50 %	IP % 81.80 %	1061.74 (0.000)
DES	0.87	Hospitalización 40.20 %	Domicilio 44.00 %	Hospitalización 57.40 %	Consulta Externa 34.80 %	1099.21 (0.000)
Edad	0.87	34.13 F=4.93(p<0.00)	36.91 F=1.15(p<0.00)	34.16 F=2.28(p<0.00)	45.47 F=16.33(p<0.00)	-
Sexo	0.62	Masculino 78.40 %	Masculino 73.40 %	Masculino 71.40 %	Masculino 51.30 %	1009.55 (0.000)
Día	0.33	Domingo 25.70 %	Domingo 21.40 %	Domingo 19.70 %	Lunes 16.00 %	568.69 0.000

AFE=Área Afectada; ATE=Área de Atención; CE=Causa Externa; CON=Consecuencia; DIAG=Diagnóstico; SOC=Sitio de Ocurrencia; ATPR=Atención prehospitalaria; REF=Usuario Referido por; DES=Destino del paciente; W504=Aporreo, golpe, mordedura, patada, rasguño o torcedura infligidos por otra persona, calles y carreteras; W509=Aporreo, golpe, mordedura, patada, rasguño o torcedura infligidos por otra persona, lugar no especificado; W010= Caída en el mismo nivel por deslizamiento, tropezón y traspicé, vivienda; V499=Ocupante (cualquiera) de automóvil lesionado en accidente de tránsito no especificado; S019=Herida de la cabeza, parte no especificada; S934=Esguinces y torceduras del tobillo; S098=Otros traumatismos de la cabeza, especificados.

El 21.40 % se registraron en domingo y su principal consecuencia fueron heridas (34.90 %), el área de atención fue consultorio (68.78 %). El 44 % del grupo fue enviado a su domicilio.

El clúster C3 15.90 %(2,640) mostró pacientes jóvenes (\bar{x} = 34.16 años) de sexo masculino (71.40 %) que fueron referenciados por una unidad de servicios estatales de salud (47.50 %) con fractura en la cabeza (30.20 %) como resultado de accidentes con vehículos automotor (47.80 %). Este grupo, con la menor cantidad de casos, es el que demanda mayor cantidad de hospitalizaciones (57.40 %) y requiere atención prehospitalaria con mayor frecuencia (63.60 %).

El clúster C4 muestra pacientes adultos (\bar{x} = 45.47 años). Es en este grupo donde los pacientes de sexo femenino tienen mayor presencia (48.7 %). Son lesiones sufridas dentro del hogar (47.10 %) que presentaron fracturas en las extremidades inferiores (35.80 %). Principalmente fueron atendidas en Traumatología (72.60 %) con canalización a consulta externa (34.80 %).

4.2. Subregistro por códigos inespecíficos

En esta sección se plantea el uso de clasificación de textos para determinar el subregistro de lesiones como consecuencia del diagnóstico inespecífico *dolor agudo (R52.0)*. Para fines de este propósito se entrenaron y validaron cuatro modelos predictivos tal como fue descrito en la sección 3.2. En este apartado se muestran los resultados de los análisis realizados con este fin, se describe cada modelo en el orden presentado en la Tabla 3.2.

Para cada modelo se calculó la exactitud, F_1 -score, Precisión, Recall y el área bajo la curva (AUC). El rendimiento de todos los modelos se muestra en la Tabla 4.4 y fue calculado utilizando el conjunto de datos de prueba. Se encontró una exactitud promedio de 0.8413 siendo el modelo basado en notas médicas en formato texto (M4) el que mejores resultados mostró (exactitud = 0.9393). Así mismo, para el parámetro F_1 -score, dicho modelo superó al resto.

Tabla 4.4: Desempeño de los modelos contra los datos de prueba.

Modelo	Exactitud	F_1 -score	Precisión	Recall	AUC
M1	0.8117	0.8184	0.7941	0.8442	0.8758
M2	0.8001	0.8006	0.8026	0.7985	0.8702
M3	0.8142	0.8181	0.8054	0.8313	0.8878
M4	0.9393	0.9392	0.9538	0.9250	0.9729

El modelo M1 se basa en el conjunto de datos estructurados y se realiza una LR que incluye el uso de la ecuación 2.1 para la validación mediante una regularización *ridge regression*. La gráfica de la Figura 4.4 muestra la exactitud para cada valor de λ tanto de los valores de entrenamiento como de validación. En este proceso se busca que el valor de λ maximice la exactitud del conjunto de datos de validación.

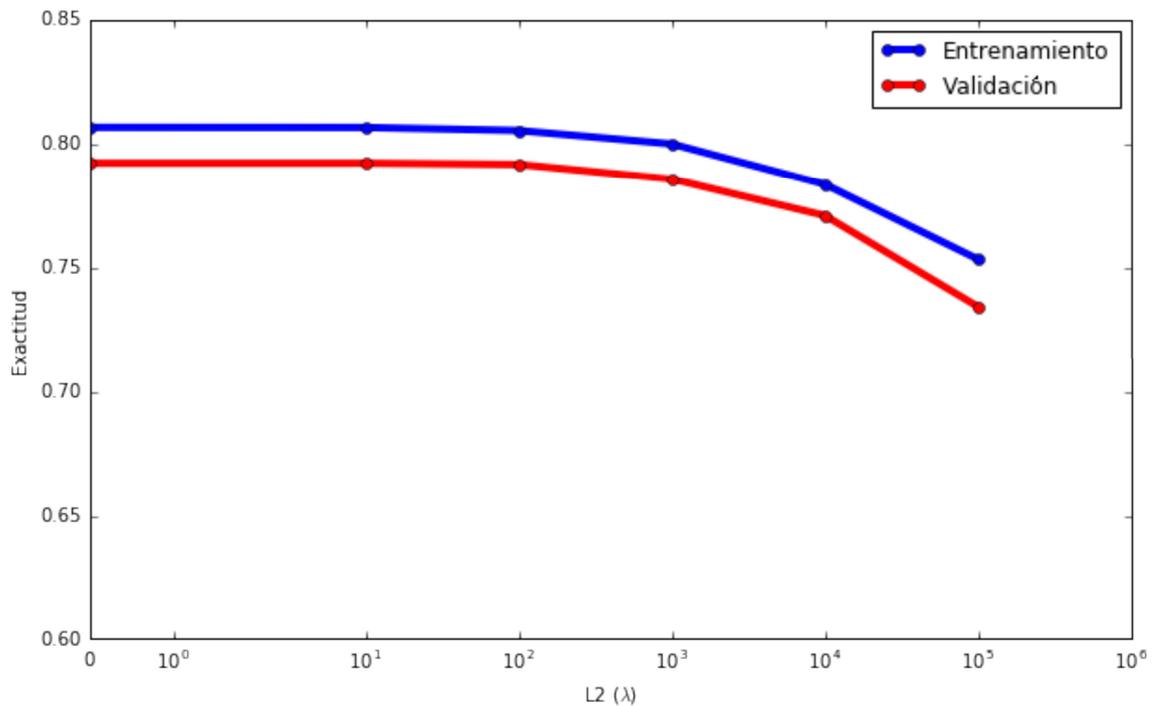


Figura 4.4: Exactitud del modelo M1 con diferentes valores de regularización λ

En este caso hay poca variabilidad en la exactitud del modelo (ver Figura 4.4). El valor de λ que minimiza el error y que por lo tanto aumenta la exactitud es 100 (10^2); este valor fue utilizado para entrenar la versión final del modelo M1. Los coeficientes del modelo resultante con pesos más representativos (positivos y negativos) son los mostrados en la Tabla 4.5.

Tabla 4.5: Variables con coeficientes más representativos del modelo M1 $\lambda = 100$

Variable	Coficiente	Descripción
proc.3809	10.0069	Incisión de venas, miembros inferiores
proc.8149	8.3752	Otra reparación de tobillo
med.02608	8.3734	Carbamazepina
med.00569	7.3166	Nitroprusiato de sodio
med.05506	7.2910	Celecoxib
med.03253	-6.2843	Haloperidol
proc.9393	-6.0935	Métodos de resucitación no mecánicos
proc.9908	-5.9974	Transfusión de expansor sanguíneo
med.03631	-5.9859	Solución de glucosa
med.01901	-5.7695	Sulfadiazina

proc.- Procedimiento médico, med.- Medicamento.

Los prefijos *proc* y *med*, hacen referencia al procedimiento aplicado al paciente y a los medicamentos prescritos, ambos valores basados en sus respectivos catálogos. Debido a que el

nombre de la variable es largo y para mejorar la lectura, se han acotado al prefijo *proc* y *med* más la clave correspondiente. El comportamiento de estos coeficientes a diferentes valores λ se muestra en la Figura 4.5.

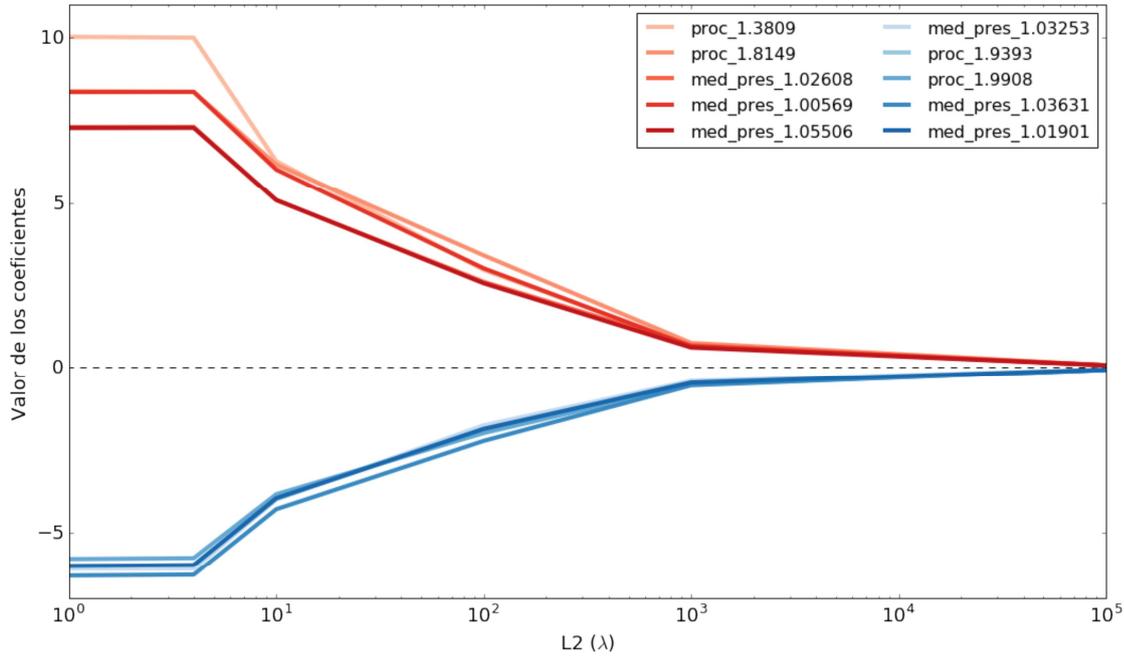


Figura 4.5: Comportamiento de los coeficientes del modelo M1 a distintos valores de regularización λ

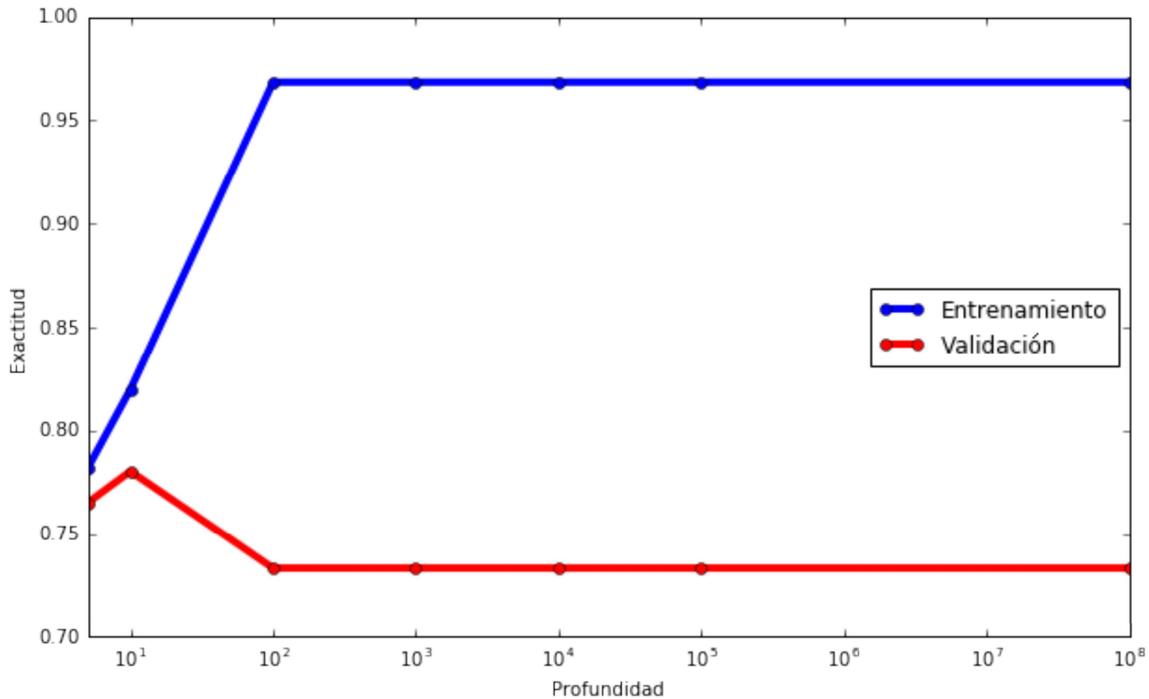


Figura 4.6: Exactitud del modelo M2 con diferentes valores de profundidad del árbol de decisión

El modelo M2 también utiliza datos estructurados para entrenar un DT. En este caso el parámetro a considerar es la profundidad del árbol. En la Figura 4.6 se puede apreciar que el mejor valor de profundidad es 10. Con este valor se realiza el entrenamiento final del modelo y posteriormente se realiza la prueba.

El modelo M3 utiliza un el algoritmo BT. En este modelo el parámetro a encontrar es la cantidad de árboles o *iteración* que conformarán el modelo. En la Figura 4.7 se muestra sus respectivos valores de exactitud a diferente número de árboles. Aunque se puede apreciar que al aumentar indefinidamente el número de árboles la exactitud en el conjunto de entrenamiento se hace casi perfecta, también se aprecia que la exactitud en el conjunto de datos de validación disminuye a partir del valor 50, por lo que se ocupa este último como parámetro del modelo final.

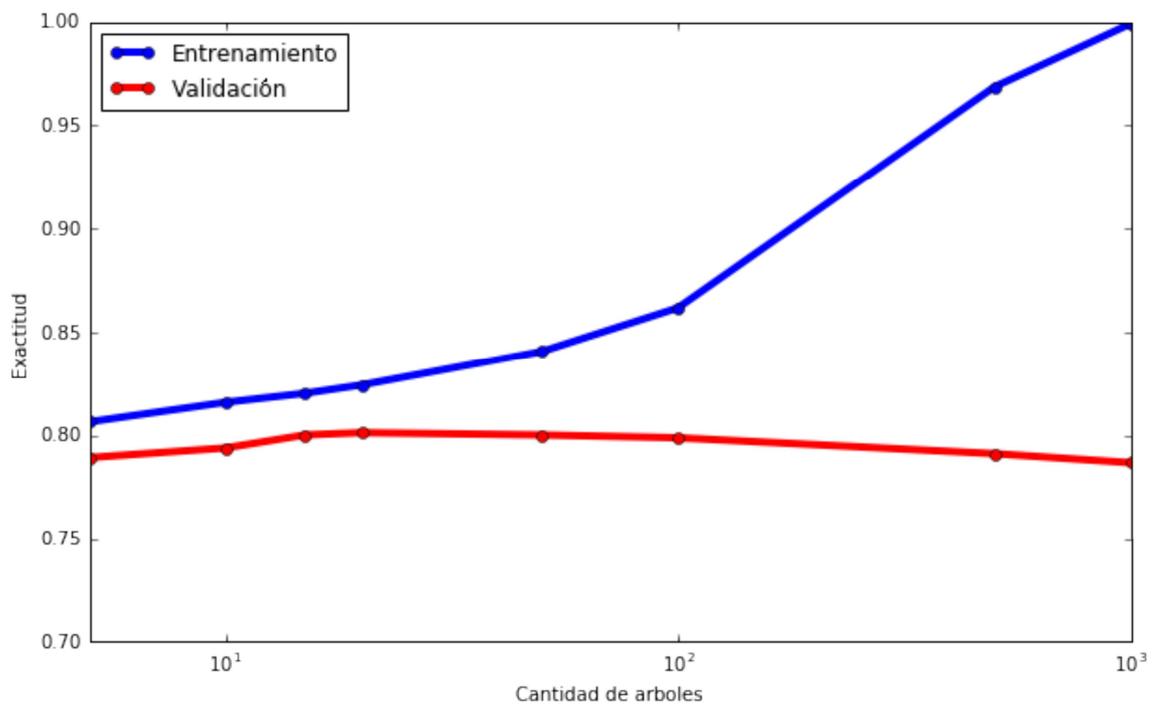


Figura 4.7: Exactitud del modelo M3 para distintos valores de regularización

Finalmente, para el modelo M4 una vez realizada la representación del texto con el método de TFIDF, que se calcula mediante la ecuación 2.9. Posteriormente se entrena un modelo LR donde el proceso es similar al del modelo M1. En la Figura 4.8 se muestran los valores de la exactitud según varía el valor del parámetro λ . En este caso el valor que presenta el mejor rendimiento en el conjunto de validación es con $\lambda = 1000(10^3)$.

A diferencia del modelo M1, el lugar de las variables es tomado por las palabras utilizadas en el análisis. Las que tienen coeficientes más representativos se muestran en la Tabla 4.6 (Las palabras fueron analizadas en mayúsculas y sin acentos).

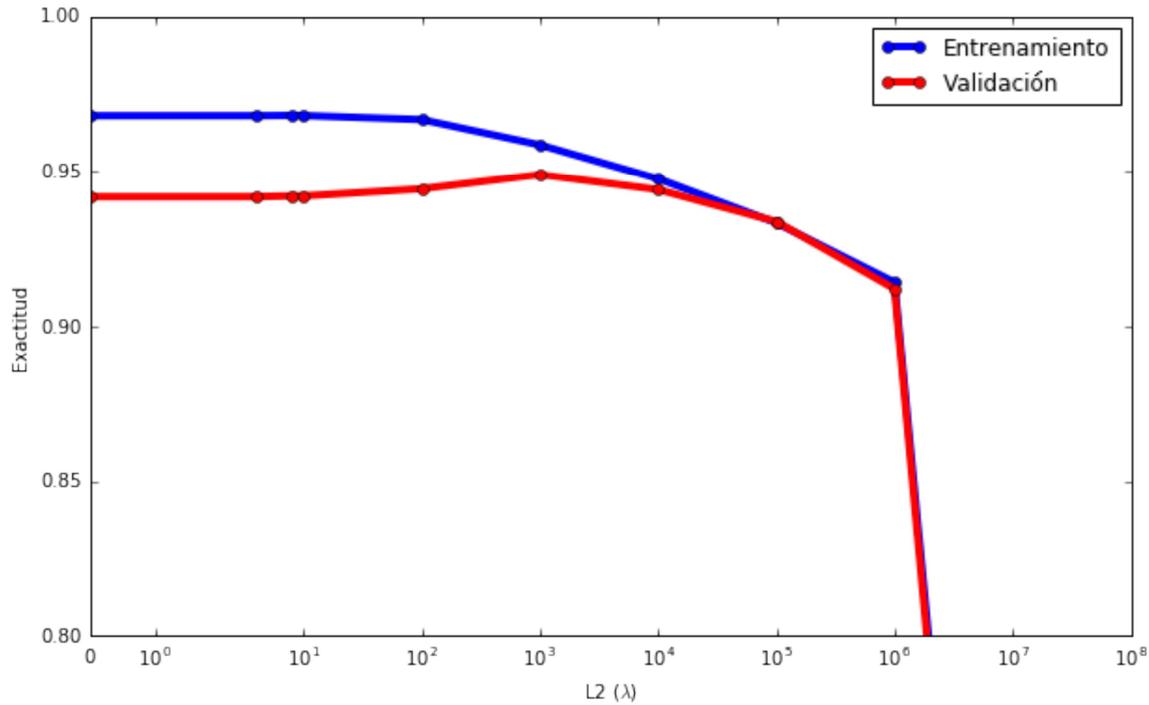


Figura 4.8: Exactitud del modelo M4 con diferentes valores de regularización λ

Tabla 4.6: Palabras con coeficientes más representativos del modelo M4 con $\lambda = 1000$

Palabra	Coficiente
Privación	0.5067
Minimizar	0.4201
Difteria	0.3803
Refuerza	0.3747
Arteriovenoso	0.3392
Alcali	-0.3454
Escotoma	-0.3140
Preparan	-0.3110
Leucemias	-0.2905
Purgante	-0.2692

La parte culminante en esta etapa del trabajo es utilizar los modelos implementados para clasificar los registros de atenciones médicas a pacientes que fueron diagnosticados con el código inespecífico de *dolor agudo (R52.0)*. En la columna *clasificación* del Tabla 3.2, se muestra la cantidad de datos estructurados y de texto a clasificar en este análisis. Estos datos no forman parte del conjunto original que se dividió para formar los subconjuntos de entrenamiento, validación y prueba; son casos reales con los que se pretende dar una estimación de la cantidad de información perdida a causa de este fenómeno. Los datos son sometidos

a cada uno de los modelos según su tipo de dato. Finalmente, se obtiene el porcentaje de registros que se estima pueden pertenecer a LCE. Los resultados de este análisis se muestran en la Tabla 4.7 y en la gráfica de la Figura 4.9.

Tabla 4.7: Clasificaciones realizadas con datos reales de dolor agudo.

Modelo	LCE(%)	OTRO(%)
M1	11172(84.02 %)	2125(15.98 %)
M2	10848(81.58 %)	2449(18.42 %)
M3	11199(84.22 %)	2098(15.78 %)
M4	12240(82.56 %)	2586(17.44 %)

LCE.- Cantidad de registros clasificados como Lesión por Causa Externa, OTRO.- Cantidad de registros clasificados como no Lesión por Causa Externa.

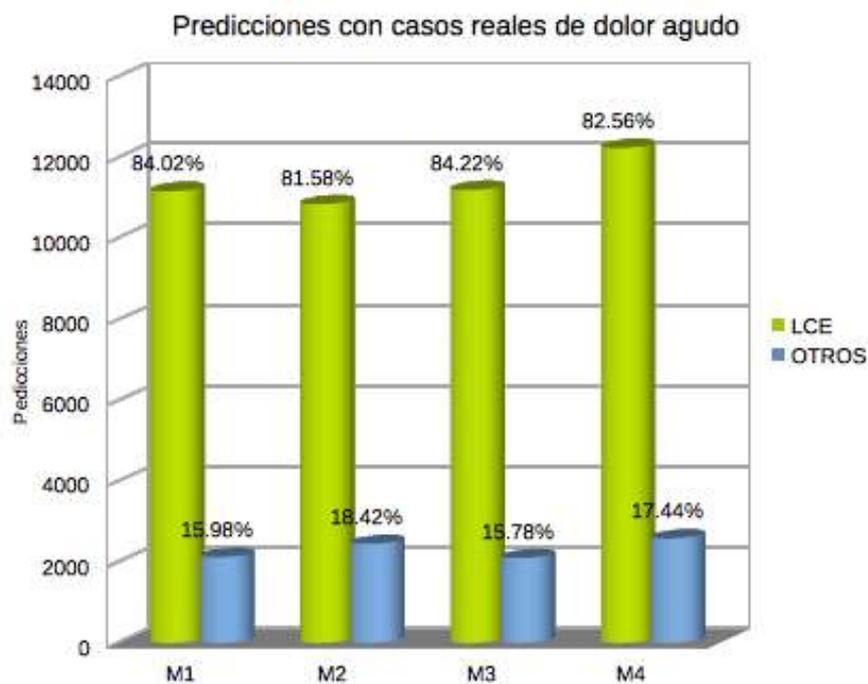


Figura 4.9: Resultados de la clasificación de dolores agudos

Con objeto de visualizar el impacto del subregistro de LCE como consecuencia del códigos inespecíficos de dolor agudo, el cual fue estimado mediante los modelos antes descritos, se suma la cantidad de registros originales a los clasificados mediante cada uno de los modelos. La gráfica de la Figura 4.10 muestra el incremento en la tendencia de la prevalencia de lesiones a través del tiempo.

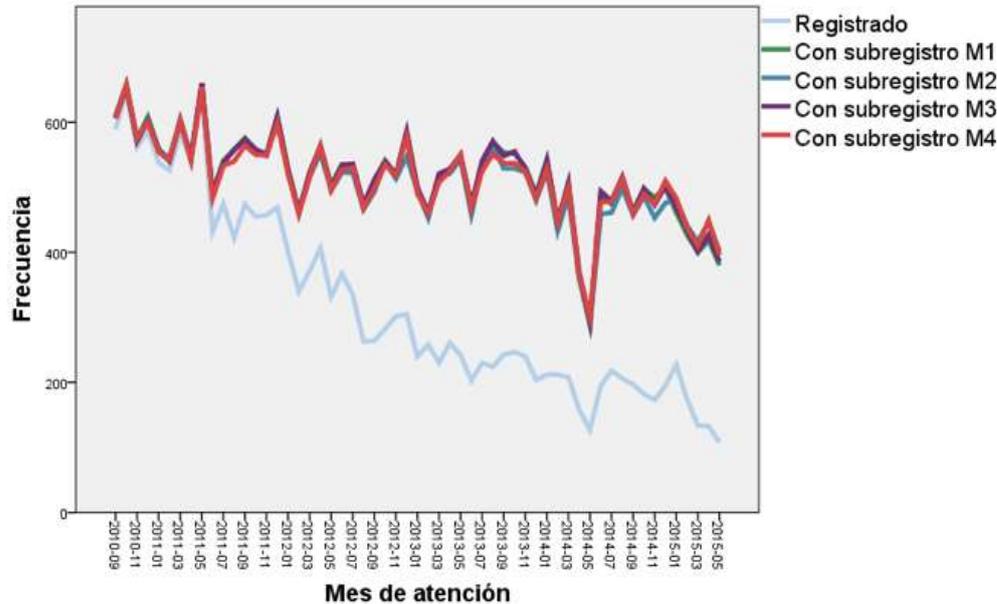


Figura 4.10: Frecuencia subestimada de las lesiones por causa externa según los modelos implementados.

4.3. Elección de modelo y balanceo de clases

La etapa final del trabajo consiste en inspeccionar el rendimiento de los algoritmos para detectar y elegir el que proporcione los valores más convenientes. Es necesario, sin embargo, determinar el método de balanceo que mejor se adapte a los datos disponibles, antes de elegir el modelo final para predecir el tiempo de estancia hospitalaria. Esta sección se divide en dos e inicia con los resultados correspondientes al balanceo de clases utilizando el método sugerido utilizando RNN-LSTM. Posteriormente se continúa con los resultados relativos al cotejo contra otros métodos y conjuntos de datos. La información aportada por estos análisis es indispensable para una elección adecuada del modelo final.

Entendiendo que el énfasis primordial del trabajo es el análisis de notas médicas y en concordancia con lo descrito en la sección 3.3.2 donde se analizan dos conjuntos de datos más: Amazon y PubMed. Se hace la aclaración de que, para evitar la explicación reiterativa relacionada con cada caso, se hará hincapié sólo en las notas médicas preprocedadas, asumiendo que el resto de los datos fue gestionado de manera análoga y sólo se hará la distinción pertinente cuando sea necesario aclarar algún punto específico.

4.3.1. Generación de texto sintético

En primer lugar, se aborda la parte referente a la generación de datos nuevos orientados al balanceo de clases para la clasificación binaria. Las salidas obtenidas a partir de la RNN-LSTM son textos sintéticos con características similares a los utilizados para entrenar la red neuronal. En la Tabla 4.8 se presentan ejemplos de texto original y texto sintético generado

Tabla 4.8: Comparación visual entre notas reales y las notas sintéticas.

Tipo	Estancia	Nota médica
Real(preprocesada)	Normal	NEGADOS ACUDE HABER PRESENTADO LESION TOBILLO DERECHO SECUNDARIO LESION ACTIVIDAD REFIERE FORZADA PRODUCE DOLOR INTENSO LIMITACION FUNCIONAL INCAPACIDAD DEAMBULACION ACUDE HABER PRESENTADO LESION TOBILLO DERECHO SECUNDARIO LESION ACTIVIDAD REFIERE FORZADA PRODUCE DOLOR INTENSO LIMITACION FUNCIONAL INCAPACIDAD DEAMBULACION INGRESA SERVICIO TRATAMIENTO QUIRURGICO CONSISTENTE REDUCCION ABIERTA FIJACION INTERNA COLOCACION PLACA TERCIO BAJO PRINCIPIO REPARACION
Sintética(preprocesada)	Normal	REFIERE INICIA DIA HOY UNA HR PREVIA INGRESO ER ENCONTRARSE VEHICULO TERCERAS PERSONAS ARMA FUEGO TORAX HERIDA FLANCO DERECHO CUENTA TIEMPO VALORACION PACIENTE DESCONTROL METABOLICO ANEMIA MODERADA FRACTURA TERCIO DISTAL CUBITO RADIO IZQUIERDO LESION DIAS INICIA ANALGESICOS SEDACION MIDAZOLAM DOBLE ESQUEMA ANTIBIOTICO ANALGESIA SEDACION AMV SEDACION DOBLE ESQUEMA ANTIMICROBIANO PARENTERALES GASOMETRIA ARTERIAL REGION
Real	Prolongada	NIEGA DE IMPORTANCIA PARA PA ALERGICOS QX O TRASNFSUSIOALES NEGADOS REFIERE LO INICIA EL DIA DE AYER APROX A LAS 15:00 HRS AL SER AGREDIDO POR TERCERAS PERSONAS CON ARMA DE FUEGO LARGA A UNA DISTANCIA DE APROX 20 MTS RECIBIENDO IMPACTO EN BRAZO I ZQUIERDO CON DOLOR Y LIMITACION FUNCIONAL POR LO QUE ACUDE HPPAF EN BRAZO IZQUIERDO DESPIERTO ORIENTADO BIEN HIDRATADO ADECUADA COLORACION CUELLO SIN ALTERACIOES CS PS LIMPIOS Y BIEN AEREAOS
Sintética	Prolongada	PACIENTE NO RECUERDA NADA DESPUES ENCONTRADO EN LA AZOTEA DE SU DOMICILIO POR ACCIDENTE AUTOMOVILISTICO Y DESCONOCIENDO MECANISMO CINEMATICA DE GOLPE EN MOLEDDORA DE ABORTO DE MANERA MISMO AUXILIADA POR PERSONAL PARAMEDICO POR LO QUE ACUDE DICLOFENAC SE COLOCA SEP AL ESTAR EN SU CASA SE DESCONOCE EL FAMILIAR AL ESTAR PARADO DEBAJO DE SU VEHICULO APARENTE PROYECCION FUERA PRESENTANDO PERDIDA DE ESTADO DE ALERTA REFIERE PORPIO

utilizando el conjunto de notas médicas preprocesado y con el conjunto de notas médicas original (sin preprocesar). Para generar el texto, se utilizaron los valores de parámetros señalados en la Tabla 3.3.

Una vez generado el texto sintético y balanceadas las clases respectivas a cada conjunto de datos, se entrenaron y validaron los modelos con base al procedimiento descrito en la Figura 3.1. Finalmente, se realizó la prueba de cada modelo con el subconjunto aleatorio de datos elegidos para tal efecto. Con los valores obtenidos en este último paso se procedió a evaluar y comparar los resultados de cada modelo, como se detalla a continuación.

4.3.2. Comparación entre métodos de balanceo

El primer punto a considerar, es la verificación de una hipotética mejoría en el rendimiento de los modelos entrenados utilizando el conjunto de datos desbalanceado en contraste con los balanceados, centrados en el método propuesto basado en notas sintéticas. Para este análisis inicial se utilizan los datos de notas médicas preprocesadas. Las gráficas de la Figura 4.11 muestran el comportamiento del parámetro F_1 -score para cada uno de los modelos valiéndose de distintos conjuntos de datos aleatorios obtenidos a partir de el valor de la semilla indicada. Una inspección visual sugiere que el los modelos que emplean el algoritmo BT no presenta una variación significativa en sus niveles de rendimiento. Por otro lado, los valores obtenidos de los algoritmos LR y SVM sí advierten una leve mejoría al aplicar el balanceo por notas

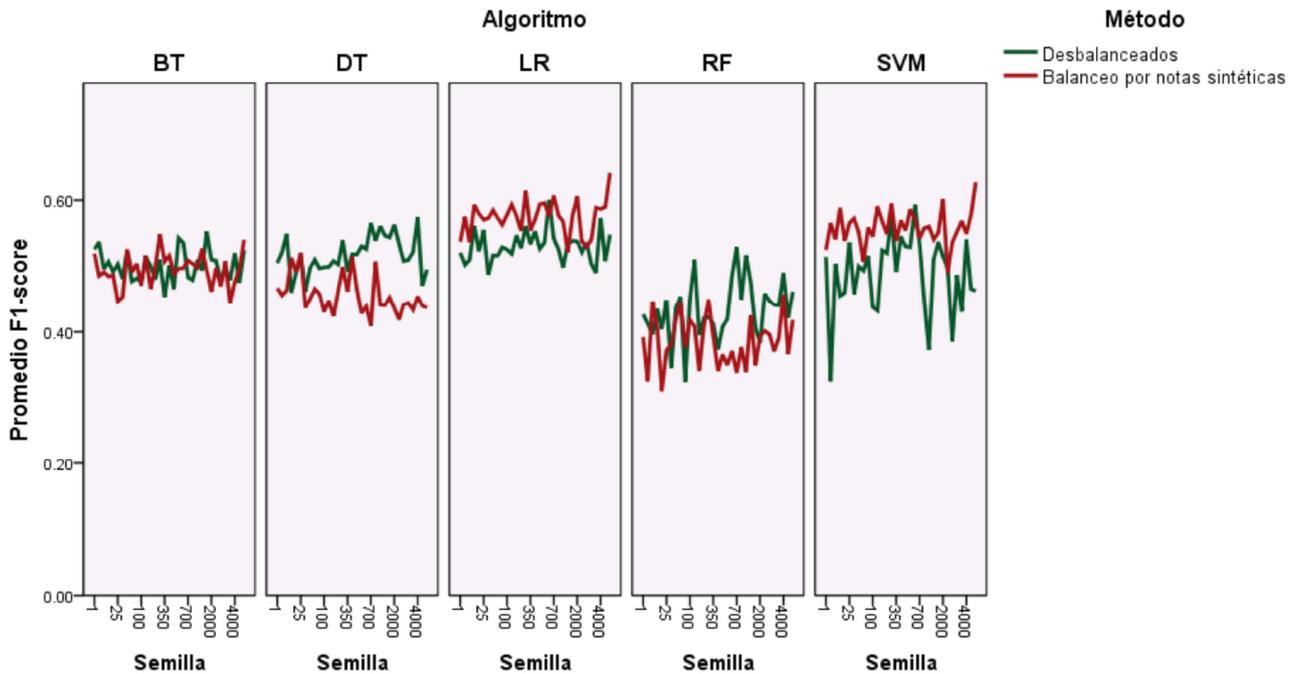


Figura 4.11: Rendimiento de los algoritmos utilizando balanceo por notas sintéticas - notas médicas preprocesadas

sintéticas con respecto a las desbalanceadas. No obstante, en los resultados mostrados a partir de los algoritmos DT y RF se percibe una degradación del rendimiento al utilizar el método en cuestión.

Al realizar el análisis estadístico para comprobar los resultados alusivos a la Figura 4.11, los valores obtenidos confirman que los algoritmos LR ($t=-7.196$, 0.000) y SVM ($t=-6.353$, 0.000) presentan una mejoría significativa cuando se emplea el método propuesto de balanceo con notas sintéticas en contraste con los datos desbalanceados originales. De la misma manera, para los algoritmos BT ($t=0.994$, 0.324), DT ($t=8.725$, 0.000) y RF ($t=4.626$, 0.000) se corrobora que no hubo mejoría, de hecho en algunos casos se presentaron mejores resultados con el conjunto desbalanceado original.

El siguiente punto en el análisis del rendimiento, es evaluar y comparar los resultados obtenidos utilizando otros métodos de balanceo tradicionales tales como SMOTE y el sobremuestreo aleatorio. En la Tabla 4.9 se presentan los valores promedio para los modelos evaluados, donde se pueden apreciar numéricamente los resultados descritos en el punto anterior. Así mismo, se evidencia que los valores alcanzados para el método SMOTE alusivos a los algoritmos LR y SVM son muy similares a los conseguidos mediante las notas sintéticas, antagónicamente se obtuvieron valores ligeramente mayores para el método de sobremuestreo aleatorio. Este último método presenta valores superiores a sus contrapartes. Finalmente, se aclara que, aunque el estudio del rendimiento se centra en los valores de la variable F_1 -score, con fines de completitud, en la Tabla 4.9 también se muestran los valores promedio obtenidos para la exactitud de los modelos.

Tabla 4.9: Evaluación del rendimiento promedio para diferentes métodos de balanceo utilizando el conjunto de datos de notas médicas preprocesadas ($n = 33$ para todos los casos)

	Desbalanceado		Sintéticas		SMOTE		Sobremuestreo	
	F_1	Exa.	F_1	Exa.	F_1	Exa.	F_1	Exa.
BT	0.50	0.83	0.49	0.83	0.51	0.83	0.60	0.81
DT	0.52	0.82	0.46	0.80	0.51	0.80	0.57	0.78
LR	0.53	0.83	0.57	0.82	0.56	0.80	0.58	0.80
RF	0.43	0.83	0.39	0.81	0.49	0.82	0.58	0.77
SVM	0.49	0.83	0.56	0.81	0.56	0.81	0.59	0.80

BT.- Boosting Tree, LR.- Regresión Logística, DT.- Árboles de Decisión, RF.- Random Forest, SVM.- Máquinas de Soporte Vectorial.

A continuación se presentan los resultados obtenidos a través de la prueba paramétrica *t-student* para muestras independientes, con objeto de comprobar la significancia estadística de las cantidades mostradas en la Tabla 4.9. Entre otras, se confirmó que para el método SMOTE hay una diferencia estadísticamente significativa para los algoritmos de LR ($t=-5.301$, 0.000), RF ($t=-6.113$, 0.000) y SVM ($t=-7.105$, 0.000) evidenciando una mejora significativa al balancear las clases empleando dicho método, de manera análoga a lo encontrado mediante el texto sintético para los algoritmos LR y SVM. Por otra parte, para el método de sobremuestreo aleatorio se encontró diferencia estadísticamente significativa para todos los algoritmos analizados BT ($t=-19.922$, 0.000), DT ($t=-9.022$, 0.000), LR ($t=-8.654$, 0.000), RF ($t=-17.735$, 0.000) y SVM ($t=-8.979$, 0.000), mejorando lo conseguido mediante los otros dos métodos. Con base a la gráfica de la Figura 4.12 se puede hacer una comparación visual del comportamiento de los algoritmos para diferentes métodos de balanceo. Es importante recordar que los cálculos se realizan con el conjunto de datos de notas médicas preprocesadas.

A partir del Tabla 4.9, de la Figura 4.12 y de los resultados descritos anteriormente parece evidente que el método de sobremuestreo aleatorio, además de ser el más sencillo de implementar en el entorno descrito, también presenta menos variaciones entre distintos algoritmos, sugiriendo ser más robusto que sus homólogos. Así mismo, obtiene los valores más altos para el F_1 -score.

Con objeto de complementar los resultados, separando los datos de acuerdo al algoritmo de entrenamiento en el que se basan, se realizó una comparación de medias utilizando una prueba ANOVA de un factor cuyos resultados se describen en seguida. Los análisis muestran que hay diferencia significativa para todos los métodos de balanceo analizados. Los valores F y su respectiva significancia para cada algoritmo son: BT ($F=266.049$, 0.000), DT ($F=115.328$, 0.000), LR ($F=4.892$, 0.009), RF ($F=410.665$, 0.000) y SVM ($F=12.201$, 0.000).

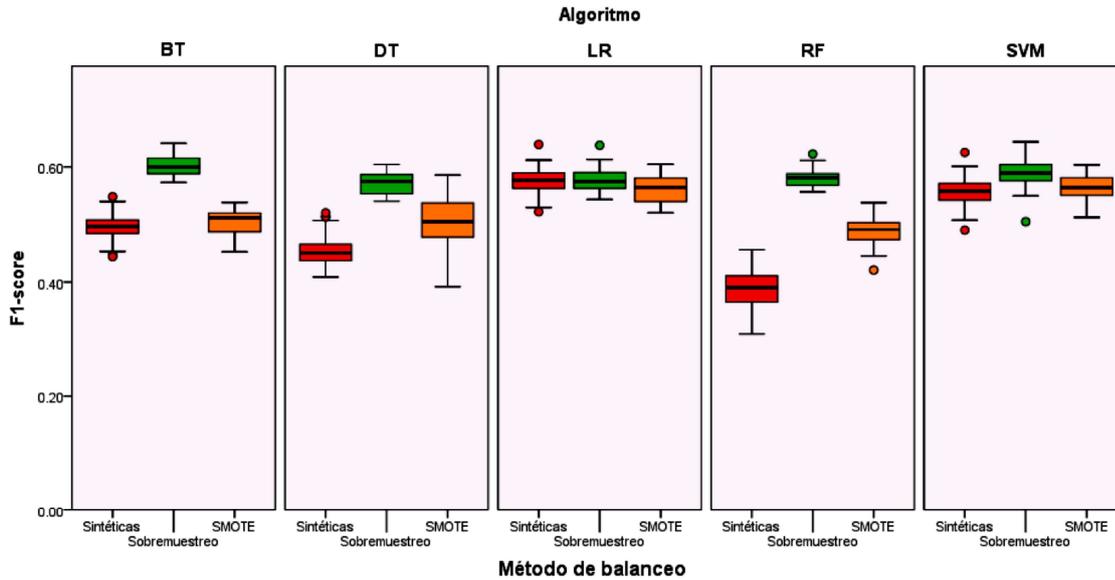


Figura 4.12: Comparación de métodos de balanceo para diferentes algoritmos de clasificación - notas sintéticas preprocesadas.

4.3.3. Evaluación con otros conjuntos de datos

Para continuar exponiendo los resultados obtenidos, se hace necesario recurrir nuevamente a la sección 3.3.2 en lo referente a la dificultad del método de balanceo SMOTE para trabajar con datos caracterizados por una alta dimensionalidad. Por lo anterior, en esta sección este método es excluido y solo se abordan las notas sintéticas y el sobremuestreo aleatorio como métodos de balanceo.

Dada la importancia de la estructura y formato de los datos para la tarea que nos ocupa, otro punto importante a considerar es la evaluación del comportamiento de los modelos y métodos de balanceo presentados anteriormente, cuando se hacen valer frente a conjuntos de datos distintos, ya que, como se discute en el siguiente capítulo, se ha encontrado que el éxito o fracaso de un método de balanceo y por lo tanto de un algoritmo de clasificación, puede estar supeditado a las características de los datos disponibles para su entrenamiento. Por tal motivo, buscando la generalización de los resultados y según lo explicado en la sección 3.3.2, se sometió a cada uno de los algoritmos y métodos al escrutinio de realizar más experimentos, para ello se hizo uso del conjunto de notas médicas originales, del conjunto de datos de opiniones de productos Amazon y del conjunto de datos de resúmenes de investigaciones de PubMed, los cuales son descritos en la sección aludida previamente.

Balanceo con notas sintéticas

Utilizando los otros conjuntos de datos y realizando el balanceo de los datos con el método de notas sintéticas se obtienen los promedios del valor F_1 -score y exactitud mostrados en la Tabla 4.10. Los valores relativos al conjunto de datos de notas médicas preprocesadas,

son los mismos que los mostrados en la Tabla 4.9, se repiten para facilitar su contraste con los resultados obtenidos con otros conjuntos de datos. Como se puede observar, el método de balanceo por notas sintéticas se comporta mejor en otros conjuntos de datos. Particularmente, el conjunto de notas médicas originales es el que tiene los mejores resultados para los algoritmos DT y LR. Mientras que, las notas médicas preprocesadas fueron las mejores al utilizar el algoritmo RF, sin embargo los resultados para este algoritmo son, en general, bajos.

Tabla 4.10: Evaluación del rendimiento promedio para diferentes conjuntos de datos con el método de balanceo por notas sintéticas ($n = 33$ para todos los casos)

	Amazon		Notas médicas		Notas preprocesadas		PubMed	
	F_1	Ex.	F_1	Ex.	F_1	Ex.	F_1	Ex.
BT	0.58	0.93	0.50	0.83	0.49	0.83	0.18	0.99
DT	0.30	0.86	0.52	0.81	0.46	0.80	0.19	0.99
LR	0.65	0.92	0.66	0.86	0.57	0.82	0.66	1.00
RF	0.17	0.80	0.38	0.82	0.39	0.81	0.07	0.99
SVM	0.60	0.91	0.68	0.86	0.56	0.81	0.84	0.95

Las tendencias anteriores se ven exhibidas en las gráficas de la Figura 4.13 y se dividen de acuerdo al algoritmo de clasificación utilizado. Por situaciones de espacio y claridad de las gráficas, no se muestran todos los valores de semilla utilizados ya que en total fueron 33 para cada algoritmo evaluado. La gráfica muestra que utilizando los algoritmos LR y SVM los resultados obtenidos fueron mejores para las notas médicas originales que para las preprocesadas (valor promedio F_1 -score). También se puede apreciar que el conjunto de datos PubMed, que es el conjunto con mayor tasa de desbalanceo (99:1), muestra mayor variación y dispersión en las tendencias obtenidas. Complementando estos resultados con los presentados anteriormente en la Tabla 4.10, se da una mejor idea del comportamiento de estos valores.

En lo que toca al análisis estadístico de los datos anteriores sobresalen los siguientes resultados. En primera instancia, se corrobora una diferencia estadísticamente significativa a favor de las notas médicas originales frente a las notas médicas preprocesadas, para los algoritmos de DT ($t=7.830$, 0.000), LR ($t=15.796$, 0.000) y SVM ($t=22.489$, 0.000) reforzando lo mostrado en la Figura 4.13. En contraste, los algoritmos de BT ($t=0.926$, 0.358) y RF ($t=-0.759$, 0.451) no presentaron tal diferencia.

Se utiliza un análisis con la prueba ANOVA de un factor para determinar si existe diferencia significativa entre los modelos entrenados con distintos conjuntos de datos. Se hace una división tomando como referencia el algoritmo utilizado. Los resultados muestran que existe diferencia significativa para todos los conjuntos de datos y específicamente para cada algoritmo utilizado. Los resultados para cada clasificador son: BT ($F=146.017$, 0.000), DT ($F=170.734$, 0.000), LR ($F=8.897$, 0.000), RF ($F=292.488$, 0.000) y SVM ($F=36.821$, 0.000).

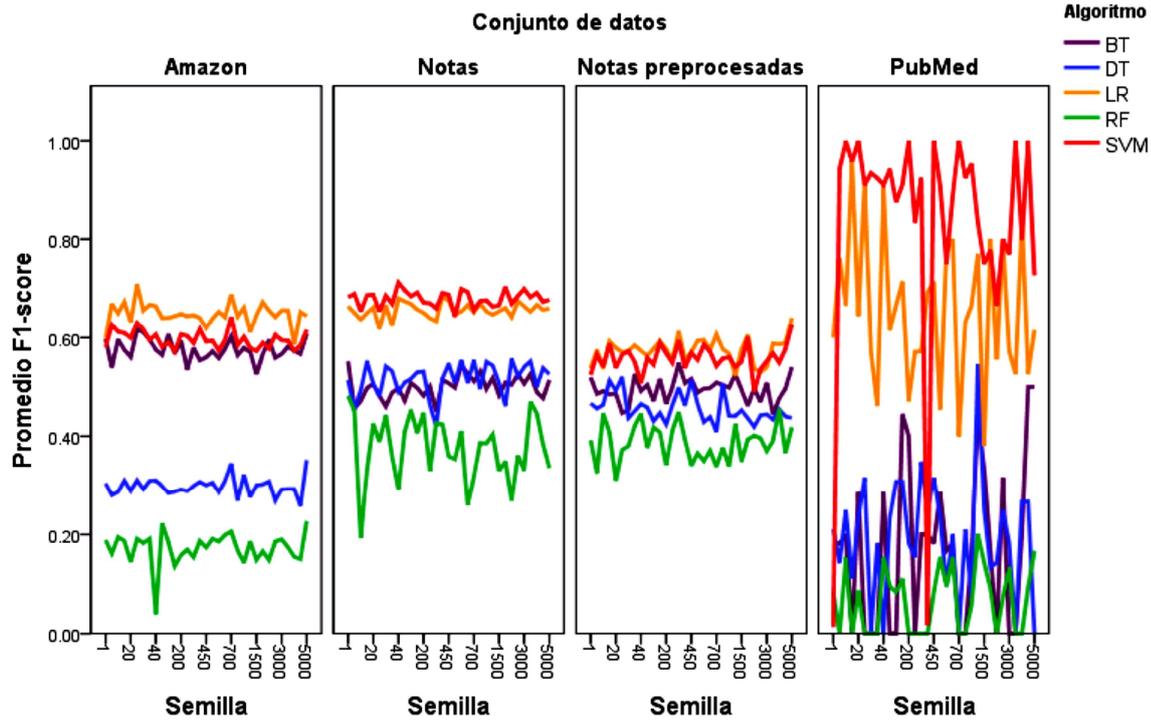


Figura 4.13: Comparación de la estabilidad de los algoritmos para diferentes conjuntos de datos y subconjuntos aleatorios - sobremuestreo aleatorio.

Balanceo con sobremuestreo aleatorio

Es momento de analizar los resultados del método de sobremuestreo aleatorio con distintos conjuntos de datos para compararlos con sus contrapartes obtenidos con el método de balanceo con texto sintético. Con esta meta en mente se presenta el Tabla 4.11 la cual muestra los resultados obtenidos para el sobremuestreo aleatorio, de manera análoga al trabajo realizado para las notas sintéticas.

Tabla 4.11: Evaluación del rendimiento promedio para diferentes conjuntos de datos con el método de balanceo por sobremuestreo aleatorio ($n = 33$ para todos los casos)

	Amazon		Notas médicas		Notas preprocesadas		PubMed	
	F_1	Ex.	F_1	Ex.	F_1	Ex.	F_1	Ex.
BT	0.62	0.92	0.60	0.81	0.60	0.81	0.34	0.99
DT	0.36	0.81	0.56	0.77	0.57	0.78	0.23	0.99
LR	0.60	0.92	0.52	0.82	0.58	0.80	0.13	0.81
RF	0.38	0.77	0.58	0.77	0.58	0.77	0.19	0.98
SVM	0.56	0.92	0.44	0.82	0.59	0.80	0.12	0.80

A diferencia del método de balanceo por texto sintético, en este caso las notas preprocesadas

tienen en promedio mejores valores del parámetro F_1 -score. Se puede apreciar también que los algoritmos basados en árboles (BT, DT y RF) tienen promedios más altos en comparación con los mismos algoritmos descritos anteriormente. Así mismo, se muestra que los resultados obtenidos con el conjunto de datos PubMed, son bastante pobres (lo que contrasta de manera evidente con la exactitud de los modelos que supera el 0.80 en todos los casos).

La representación gráfica de los valores resumidos en la Tabla 4.11 exhibe varios aspectos adicionales. En primer lugar, coincidiendo con el Tabla anterior, se aprecia un rendimiento superior de los algoritmos BT, DT y RF. En particular, cuando se trabaja con el conjunto de datos de Amazon, el algoritmo BT obtiene su mejor calificación. Otro punto a destacar es que los algoritmos presentan un comportamiento similar y homogéneo cuando se aplican a las notas preprocesadas.

Según los resultados mostrados y como se ilustra en la gráfica de la Figura 4.14 el conjunto de datos PubMed, presenta los resultados menos estables, la gráfica comentada revela una inestabilidad bastante marcada y superior a la sufrida cuando el balanceo se realiza por notas sintéticas.

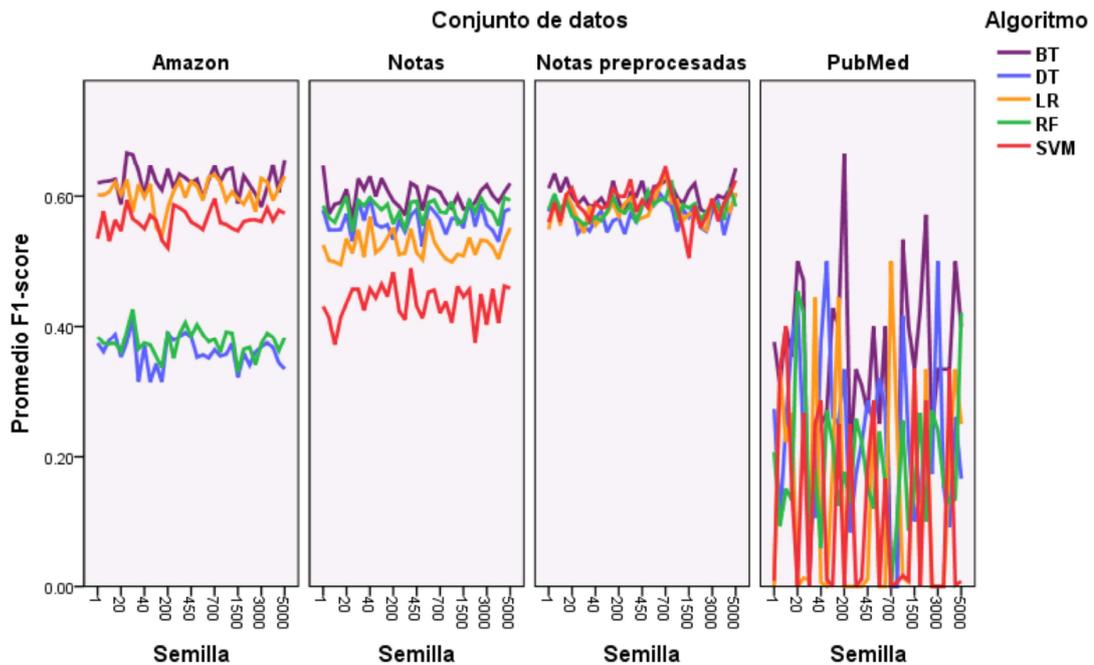


Figura 4.14: Comparación de la estabilidad de los algoritmos para diferentes conjuntos de datos y subconjuntos aleatorios.

Finalmente, en los promedios mostrados en la Tabla 4.11 y las gráficas de la Figura 4.14 parece evidente que los resultados obtenidos son dependientes de los datos con los que se trabaja, para corroborar esto, de la misma forma que se realizó para el método con texto sintético, se ejecuta un análisis estadístico mediante ANOVA de un factor (dejando fuera el conjunto de datos PubMed, por no cumplir con los supuestos de normalidad que la prueba exige) el cual confirmó que existe diferencias significativas para todos los algoritmos involucrados: BT ($F=14.870, 0.000$), DT ($F=1,043.386, 0.000$), LR ($F=131.026, 0.000$), RF ($F=1,477.731, 0.000$)

y SVM ($F=355.962$, 0.000).

Los conjuntos de datos de Amazon y PubMed se utilizaron buscando generalidad en los resultados de balanceo por notas sintéticas. No obstante, para elegir el modelo final se tomaron en cuenta únicamente las representaciones de las notas médicas originales y preprocesadas. Como se puede apreciar en la tabla 4.10 el conjunto de las notas médicas originales obtuvo el valor promedio F_1 -score más alto para los algoritmos DT(0.52), LR(0.66) y SVM(0.68) por lo que se elige esta representación para el modelo final.

Aunque el sobremuestreo aleatorio mostró mejor rendimiento que el balanceo por notas sintéticas cuando las notas médicas son preprocesadas, los resultados cambian al utilizar las notas médicas originales. De tal manera que se elige el balanceo por texto sintético.

El mejor modelo encontrado para la predicción del tiempo de estancia es el que utiliza el algoritmo de SVM, el método de balanceo por notas sintéticas y notas médicas originales. Este modelo cuenta con 107370 coeficientes (107369 variables), dos clases y logró un valor de exactitud del 0.86 y de F_1 -score de 0.68. Buscando la mejor combinación de estos señalamientos, se elige el modelo que cumpla con las siguientes características:

- Tipo de datos: Texto libre.
- Conjunto de datos: Notas médicas originales.
- Algoritmo de clasificación: Máquinas de Soporte Vectorial.
- Método de balanceo de clases: Notas sintéticas.

La tabla 4.12 muestra las palabras y sus coeficientes mas representativos para el modelo de predicción de tiempo de estancia hospitalario.

Tabla 4.12: Palabras con coeficientes más representativos para el modelo predictivo del tiempo de estancia hospitalaria $\lambda = 12000$.

Palabra	Coficiente
izquierda	2.1269
realizandop	1.9767
compleos	1.9767
aumetnao	1.9767
de	1.3874
aporoximadamente	-1.3703
nuaceoso	-1.2724
3hs	-1.2561
conq	-1.202
tosa	-1.202

Capítulo 5

Discusión, conclusiones y trabajo futuro

En este trabajo se aborda el tema de la predicción del tiempo de estancia hospitalaria, mediante el análisis de datos pertenecientes a sujetos que ingresan al servicio de urgencias con diagnóstico de Lesiones por Causa Externa. Una diferencia importante en relación a trabajos previos es el uso de notas médicas en formato texto como predictores, en lugar de los datos estructurados comúnmente utilizados. Para llegar a esta propuesta fue necesario un análisis previo de los datos disponibles y de sus características.

Este capítulo consta de tres secciones: discusión de resultados, conclusión y trabajo futuro. A su vez, en la primera de ellas se subdivide en tres partes, con objeto de presentar una discusión para cada etapa del trabajo, las cuales están ligadas a los objetivos planteados. Por otro lado, las conclusiones y el trabajo futuro se presentan de manera global.

5.1. Discusión de resultados

La discusión de resultados se aborda en función de los objetivos particulares planteados. La primer etapa, de carácter puramente descriptivo, permitió obtener un panorama general de los datos disponibles y la calidad de los mismos. Se describen los hallazgos encontrados que son de relevancia para el área médica y cómo éstos aportan al desarrollo general del trabajo.

La segunda etapa se enfoca en debatir los resultados relativos al subregistro por códigos inespecíficos. En esta parte se propone el uso de notas médicas en formato texto como alternativa para tareas de clasificación, en el contexto de los objetivos descritos. Así mismo, se realiza una serie de experimentos para comprobar la viabilidad de los modelos de clasificación, utilizando texto libre en comparación con los modelos entrenados con datos estructurados y en función del tema tratado.

La tercera y última etapa versa sobre la elección del modelo predictivo de estancia hospitalaria. Se alude al balanceo de clases y a la asociación que existe entre las características de los

datos y el comportamiento de los algoritmos utilizados para crear los modelos predictivos. En esta etapa, se propone un método de balanceo enfocado en la clasificación de textos y se realiza una serie de experimentos para comprobar su validez en comparación con otros métodos. En dichos experimentos se utilizan cuatro conjuntos de datos para corroborar los resultados.

5.1.1. Caracterización de lesiones por causa externa

La caracterización de las LCE permite tener una mejor perspectiva de la demanda en los servicios de urgencias. Los datos analizados en este estudio muestran que la mayor parte de los pacientes con LCE son de sexo masculino y que se encuentran en edades productivas de 14 a 34 años con una tendencia a la baja mientras la edad aumenta. Es importante aclarar que el hospital analizado está orientado a la atención de pacientes adultos, por tal motivo, solo se aceptan menores cuando se trata de lo que se denomina una *urgencia real* [97].

En general se encontraron resultados coincidentes con la literatura en cuanto a demanda de los servicios de salud como consecuencia de LCE. Los resultados más relevantes señalan que las agresiones interpersonales dentro del hogar y las lesiones autoinfligidas gozaron de poca prevalencia. Sin embargo, la OMS estima que un 35% de las mujeres a nivel mundial han sufrido violencia física o sexual, la mayor parte de las cuales son infligidas por la pareja la cual también es responsable del 38% de los asesinatos de mujeres que se producen en el mundo. Este organismo mundial también ha establecido que la violencia es un factor de riesgo de suicidio [81]. Por lo anterior, se considera prudente realizar estudios enfocados a detectar posibles casos que se pretendan ocultar o que estén siendo afectados por códigos inespecíficos. Así mismo, se pudo comprobar que, para algunos grupos de causas externas, como las caídas, accidentes de tránsito, y las agresiones interpersonales dentro y fuera del hogar, la edad es un factor que influye en la probabilidad de que un paciente sea hospitalizado o no. Por otro lado, como lo indican las cifras del Tabla 4.2 las áreas afectadas como abdomen, pelvis y áreas múltiples, aunque cuentan con baja prevalencia, tienen un alto porcentaje hospitalización.

Fuera del punto de vista médico, los resultados permiten observar el comportamiento de las variables involucradas, resaltando características no conocidas y que, de no analizarse y en su caso corregirse, pueden afectar el comportamiento de los algoritmos de clasificación y como consecuencia, introducir sesgos en los resultados.

En este trabajo se propone el uso de algoritmos de clusterización para agrupar las LCE de acuerdo a sus características. Se puede apreciar que los grupos formados a partir de esta herramienta son similares a los propuestos por Ávila-Burgos et al. [18] y que los resultados coinciden con lo reportado en otros estudios [17, 82, 83], este tipo de técnicas brindan información adicional sobre la estructura de los grupos y el comportamiento de las variables incluidas. No obstante, como se puede apreciar en la Figura 4.1, las LCE suelen dividirse en seis grupos, sin embargo, el análisis de clúster realizado muestra solamente cuatro agrupaciones (Tabla 4.3). Esto puede ser debido a que la frecuencia para cada una de las causas externas están claramente desbalanceadas y las clases minoritarias tienden a ser absorbidas por las mayoritarias. Un fenómeno interesante observado, aunque no reportado en el capítulo

dispuesto para la presentación de resultados, es que este tipo de clúster es muy sensible a pequeños cambios en las variables utilizadas. Ligeros cambios afectan sustancialmente los grupos generados.

Por otra parte, una revisión de los resultados presentados en el capítulo anterior revela que un buen número de registros en el área de urgencias pueden estar clasificados con un código inespecífico y subestimar el número real de LCE (Figura 4.3). Lo que a su vez puede ser causal de bajo rendimiento de los algoritmos de clasificación [84, 61].

En este trabajo se utilizan datos de sistemas de información hospitalaria, los cuales brindan ventajas en la atención médica y la administración [85], sin embargo, es importante hacer una reflexión en cuanto al diseño de los mismos. Los datos que no son obligatorios, normalmente quedan vacíos, lo que provoca un subregistro que impacta en el análisis posterior; cuando existen opciones como *Otros* sería conveniente que se especifique a que otro se refiere para que se proporcione información útil. Conviene además evitar el uso de opciones ambiguas tales como *Institución residencial* y *Hogar* de la variable sitio de ocurrencia, que pueden provocar confusión. Finalmente, es deseable que el registro no consuma demasiado tiempo del destinado a la atención del paciente [86]. Es necesario que los sistemas de información validen la calidad de los datos y que sean ágiles, rápidos y sencillos para que no interfieran en el tiempo y calidad de la atención al paciente.

Ejemplos de subregistro que pueden afectar tanto las estadísticas reportadas como los algoritmos de clasificación son las siguientes: La escolaridad del paciente se reportó con la opción *Otros* un 45.54%. La causa externa reportó 18.11% con *Otras causas externas de lesiones*. El 13.30% (15,859) del total de los datos corresponde al código inespecífico de *dolor agudo* (las LCE representaron el 16.13%, n=19,230).

5.1.2. Análisis del subregistro por códigos inespecíficos

El problema de los registros con valores inespecíficos es común a distintas áreas. En el sector salud se puede presentar en registros de mortalidad, de natalidad o de prevalencia o incidencia de enfermedades, solo por mencionar algunas. Este trabajo se enfoca en el subregistro de LCE como consecuencia del código CIE-10 que diagnóstica el *dolor agudo*, sin embargo, otros padecimientos pueden verse afectados por este o por algún otro código inespecífico [6]. Este problema puede afectar las cifras estadísticas reportadas, así como los algoritmos de clasificación debido a desbalanceo de clases o con variables predictoras sesgadas, según sea el caso.

Los datos disponibles para este trabajo muestran que en el periodo analizado, el 13.30% de los registros totales en el servicio de urgencias del hospital son diagnósticos de dolor agudo. Si se compara con el 16.13% que corresponde a LCE en el mismo periodo, se puede considerar como factor importante en las estadísticas que se reportan estatal y nacionalmente. Los resultados muestran que en promedio 83.10% de los registros de dolor agudo son LCE lo que aumentaría su prevalencia del 16.13% al 25.66% del total de atenciones en el servicio de urgencias del hospital analizado.

Por otro lado, gran parte de la información hospitalaria se encuentra en formato no estructurado y suele dejarse fuera del análisis de datos tradicional (con la pérdida de información que esto implica)¹, sin embargo, los resultados obtenidos en este trabajo sugieren que la minería de textos puede ser de utilidad en este tipo de situaciones ya sea como complemento a las técnicas tradicionales o como una alternativa confiable cuando no existen datos estructurados suficientes y de calidad.

Así mismo, al abordar el tema del subregistro por códigos inespecíficos se realizan comparaciones entre el rendimiento de modelos generados a partir de datos estructurados en contraste con datos en formato texto. Se puede apreciar en la Tabla 4.4 que la exactitud de los modelos basados en el texto ($\text{exa} = 0.9393$, $F_1\text{-score} = 0.9392$) supera al resto de los modelos evaluados. Así mismo, en las gráficas de las Figuras 4.9 y 4.10 se muestran los resultados de cada modelo y se puede apreciar que las predicciones realizadas con las notas médicas son compatibles con el resto de los modelos, mostrando la viabilidad del uso de texto libre en este contexto.

Al utilizar datos estructurados como predictores del tiempo de estancia de un paciente, se pueden identificar las variables más representativas para el modelo (ver Tabla 4.5 y Figura 4.5), esto puede ser útil para el área médica. Por otro lado, pese a que al utilizar notas médicas como predictores se puede obtener las *palabras* más representativas del modelo, esto en general, es menos útil para el área médica, según los comentarios de los especialistas. Esta particularidad se puede considerar como una desventaja inherente a la forma de trabajo del texto frente a datos estructurados.

Finalmente, es importante comentar que, los análisis que forman parte de esta sección, se realizaron con un conjunto de datos aleatorios pero sin recurrir a técnicas como la validación cruzada, para asegurar que los resultados no dependan de la partición elegida. Por este motivo, sería conveniente realizar los experimentos necesarios para mitigar las dudas que pudieran surgir.

5.1.3. Análisis de balanceo mediante texto sintético

En esta sección se discute cómo, el proceso de implementación de los experimentos realizados para la elección del modelo de clasificación, se fusionó con la indagación del método de balanceo de clases que busca maximizar el valor $F_1\text{-score}$ en la clasificación de los tiempos de estancia. Conforme se realizaron las pruebas para encontrar una combinación óptima entre el método de balanceo y el algoritmo de clasificación, se realizó también, implícitamente, la elección del modelo pertinente. Así mismo, se utilizaron distintos conjuntos de datos para generalizar los resultados obtenidos, sin embargo, como es de esperarse, el modelo final se elige entre los conjuntos de notas médicas ya sean preprocesadas o no, relegando al resto para tareas propias de la experimentación.

Al variar el método de balanceo y hacer la comparación entre el sobremuestreo aleatorio y

¹Se ha reportado que alrededor del 80 % de los datos dentro de las organizaciones se encuentra en formato texto [8, 9].

el método propuesto, se puede apreciar en las tablas 4.10 y 4.11 que el método de balanceo por texto sintético supera al método de sobremuestreo aleatorio cuando se utiliza el texto original y los algoritmos LR (F_1 -score = 0.66) y SVM (F_1 -score = 0.68). Trabajos similares como el presentado por Lucini et al. [87] donde también se realiza una clasificación binaria y representación del texto en TFIDF, reportan en promedio un F_1 -score de 0.77, cabe aclarar que el conjunto de datos que se presenta es un conjunto balanceado por lo que no requirió el balanceo previo. Otra diferencia importante es que no mencionan si se realiza algún método de validación con lo que se podría variar el resultado al variar la elección de los subconjuntos de entrenamiento, validación y prueba.

Por otro lado, Jingjing Wang et al. [58] proponen un método alternativo para balanceo de clases con datos de texto denominado P-SMOTE que, como su nombre lo indica es una modificación del SMOTE tradicional. Los autores se enfocaron en mejorar el rendimiento del algoritmo de clasificación SVM con resultados promedio ($n = 5$) de F_1 -score = 0.64 superando al SMOTE puro (F_1 -score = 0.63). Estos resultados son similares a los obtenidos en esta tesis para el algoritmo SVM y utilizando el método de balanceo por notas sintéticas con promedios $F_1 = 0.60, 0.68, 0.56$ y 0.84 para los conjuntos de datos Amazon, Notas médicas, Notas médicas preprocesadas y pubmed, respectivamente. El trabajo de Wang no menciona si realizaron algún tipo de validación. Tampoco se menciona si se realizan pruebas estadísticas para comprobar que las diferencias sean significativas.

Inicialmente, se aborda el tema de la generación de texto sintético a partir de una RNN-LSTM. En este contexto, el principal tópico a considerar, es la optimización de los parámetros de entrada, para mejorar la calidad del texto obtenido. Normalmente, el texto generado a través de este tipo de redes, describe un comportamiento y estructura parecido al texto original y exhibe cierta coherencia en las frases generadas. Se han utilizado técnicas basadas en el criterio humano, en otros contextos, tal es el caso de la evaluación de diálogos automáticos hombre-máquina o la generación de resúmenes multigénero [88, 89].

Las técnicas de evaluación de textos basadas en el criterio humano, aunque pueden considerarse válidas, cuentan con cierto grado de subjetividad, son vulnerables a la intervención de otros puntos de vista y son propensas a cambiar con el paso del tiempo. Además, también pueden ser lentas y tediosas. En este tenor, se han presentado trabajos enfocados a los métodos de evaluación propiamente dicho, tales como el Semantic Textual Similarity (STS) [90] el cuál mide el grado de equivalencia semántica entre dos textos, y es normalmente utilizado en tareas de traducción, lectura de máquinas, respuesta a preguntas etc. Por otro lado, en el artículo publicado por Tao et al. [91] se presentan las pruebas iniciales de un método de evaluación de conversaciones hombre-máquina, denominado: Referenced metric and Unreferenced metric Blended Evaluation Routine (RUBER). Los autores manifiestan una alta correlación con evaluaciones realizadas por humanos. Otro ejemplo es la publicación de Yang et al. [92] donde se presenta un modelo predictivo, en el contexto de validación de conversaciones. Realizan evaluaciones en cuanto a la similitud semántica de las oraciones involucradas y lo comparan con el STS frente al cual se argumenta buena competitividad. Aunque estas técnicas no están enfocadas al tema que nos ocupa, pudieran ser útiles si se realizan modificaciones para su adaptación. En cualquier caso, parece evidente la necesidad de un parámetro objetivo para la evaluación del texto generado que, posteriormente, mejore los resultados y la confiabilidad de la clasificación.

En este sentido, un punto a tener en cuenta es la capacidad de cómputo que se requiere para el entrenamiento de las RNN-LSTM. Dada la demanda de recursos que implica, normalmente es aconsejable utilizar cómputo de alto rendimiento, en particular se recomienda el uso de unidades de procesamiento de gráficos (GPU) que permiten acelerar la ejecución de este tipo de aplicaciones. Es común que las librerías especializadas en Deep Learning cuenten con compatibilidad para este tipo de implementaciones [93, 94].

En cuanto al rendimiento de los modelos de clasificación, se hará referencia, en primera instancia, a la confrontación realizada entre los datos desbalanceados y los datos balanceados mediante texto sintético. En este orden de ideas, se pudo comprobar que el balanceo mediante la generación de texto sintético, mejora significativamente los valores promedio de la variable F_1 -score de los modelos de clasificación, cuando se emplea para algoritmos de LR y SVM. No obstante, los resultados obtenidos a través de los algoritmos de BT, DT y RF están por debajo del 0.50 (ver Tabla 4.9). Esto puede deberse al funcionamiento interno de cada algoritmo. Mientras los primeros se basan en el algoritmo *gradient descent* para minimizar el error calculado durante la fase de entrenamiento, los últimos tres son basados en árboles (ver capítulo 2). Otro punto a considerar es la representación del texto. En este trabajo se optó por el TFIDF presentado en la sección 2.2.1, sin embargo, existen otras representaciones que pudieran cambiar el rendimiento obtenido por los modelos. Se recomienda ahondar en estas posibilidades realizando más experimentos con objeto de cotejar los resultados.

Una comparación visual de los métodos de balanceo, proporciona mayor información al respecto. En la gráfica de la Figura 4.12 se aprecia claramente que el método de sobremuestreo aleatorio tiene los rendimientos más altos, en comparación con el resto. Derivado del tamaño y características de las barras mostradas, y de los puntos respectivos a cada elemento, se puede argumentar que los valores tienen poca dispersión, lo que refuerza lo presentado en la Tabla 4.9. El sobremuestreo aleatorio también muestra valores similares de un algoritmo a otro, lo que sugiere mayor estabilidad que sus homólogos.

Por otro lado, la inclusión de otros conjuntos de datos permitió generalizar los resultados obtenidos. Coincidiendo con otras investigaciones, los resultados sugieren que el rendimiento de los algoritmos de clasificación es dependiente del conjunto de datos analizado [34]. En este caso, se encontraron diferencias que pudiera ser consecuencia de la tasa de desbalanceo de cada uno. El caso más evidente es el que deriva del conjunto PubMed. Estos datos, siendo los más desbalanceados, presentan los resultados más pobres. Esto se corrobora analizando la Figura 4.13 donde la parte correspondiente a PubMed se mostró claramente inestable. Aún así, se aprecia que los algoritmos LR y SVM tuvieron resultados superiores al resto.

Finalmente, haciendo una comparación entre LR y SVM, los mejores algoritmos para el método de balanceo con notas sintéticas, se aprecia que, en promedio, el rendimiento de la regresión logística ($\bar{x}=0.63$, $SD=0.04$) es ligeramente menor al de las máquinas de soporte vectorial ($\bar{x}=0.67$, $SD=0.12$), sin embargo, también tienen una menor dispersión. Esto es importante al elegir tanto un método de balanceo como un algoritmo de clasificación.

En el caso del sobremuestreo aleatorio, la tendencia cambia, se aprecia un mejor rendimiento para las notas médicas preprocesadas que para las notas sin preprocesar. También se puede apreciar que los resultados para el conjunto PubMed son bastante bajos, incluso en com-

paración con los obtenidos mediante el balanceo por notas sintéticas. Este comportamiento sugiere que este último método puede ser de utilidad cuando en conjuntos de datos altamente desbalanceados. Por otro lado, la gráfica de la Figura 4.13 ilustra una mejora notable en para el algoritmo BT cuando se utiliza para clasificar los conjuntos de datos de Amazon, y notas médicas con y sin preprocesar.

5.1.4. Elección y descripción del modelo

En la elección del modelo final, para detectar pacientes con probabilidad de estancia prolongada, se buscan cumplir varias expectativas. Por un lado que registre los mejores niveles en el parámetro F_1 -score, y que presente mejor estabilidad que sus contrapartes. Esta última característica la podemos dividir en dos vertientes: estabilidad de los algoritmos utilizando el mismo conjunto de datos y estabilidad en los valores promedio F_1 -score, variando los datos a clasificar. Para tomar esta decisión, se contemplan todos los resultados obtenidos en cada una de las etapas, considerando que tanto la clasificación de notas médicas en texto libre como el balanceo por notas sintéticas son alternativas viables para los objetivos de este proyecto.

Segun los criterios descritos el mejor modelo se construye con base al algoritmo SVM, con una representación del texto TFIDF con las notas médicas originales y el método de balanceo por notas sintéticas. Este modelo cuenta con 107370 coeficientes (107369 variables), dos clases y logró un valor de exactitud del 0.86 y de F_1 -score de 0.68. Una desventaja del modelo elegido son la poca utilidad de los coeficientes para el área médica, ya que en el texto original no se eliminan palabras mal escritas o poco relevantes, éstas pueden ser utilizadas en los coeficientes del modelo final pero al usuario no le proporcionan información útil.

5.2. Conclusiones

En el presente trabajo, buscando coadyuvar en la solución de un problema clásico de las Ciencias de la Salud, como es la predicción del tiempo de estancia hospitalario, y aplicando técnicas de una rama de la Ingeniería conocida como machine learning o aprendizaje automático, en este trabajo se conjuntan ambos campos. Se presenta un enfoque completo que abarca desde el análisis inicial de las características de los datos disponibles hasta la elección de un modelo predictivo que permita la detección de pacientes con estancias prolongadas.

Dos conclusiones importantes emergen a partir de este trabajo. El primero se refiere a la viabilidad del uso de datos en formato texto para abordar el tema del tiempo de estancia hospitalaria, el segundo es haber demostrado la factibilidad del texto sintético generado por medio de RNN-LSTM, para el balanceo de clases en tareas de clasificación, el cual es un tema ampliamente estudiado en el ámbito del Aprendizaje Automático e Inteligencia Artificial.

El tiempo de estancia hospitalaria es un parámetro importante en la administración y en la calidad de los servicios brindados por las instituciones de salud, ha sido relacionado con el

costo, demanda y calidad de la atención médica, especialmente en la asignación de recursos según las necesidades de la organización. Sin embargo, la escasez de datos confiables, valores perdidos o sesgados, códigos basura, etc. son aspectos importantes a considerar.

En el entorno del aprendizaje automático, los modelos predictivos se construyen a partir de una serie de casos disponibles. Estos casos, normalmente se encuentran en un formato estructurado constituido por renglones y columnas. Para colectarlos, se pueden aplicar tres procedimientos: el primero es mediante sistemas de información hospitalaria que se usan en la operación diaria de las instituciones de salud. También, se pueden obtener través de diseños de investigaciones específicas para ese propósito y finalmente, mediante la realización de encuestas las cuales pueden tener diferentes grados de alcance. Cada uno de estas técnicas tiene sus pros y sus contras. Por otro lado, en algunas publicaciones que aluden al tema, se dedican secciones completas al estudio de las variables que se deben contemplar como entradas para los modelos de clasificación [99]. En este trabajo, después de haber analizado la información disponible, se evaluó el uso de notas médicas redactadas en formato de texto libre como base para crear modelos de clasificación, comprobándose que la incorporación de análisis de textos evita algunos problemas de los datos estructurados, como la heterogeneidad, los códigos inespecíficos y datos nulos o incompletos.

La predicción del tiempo de estancia hospitalario se puede abordar a partir de diversas vertientes: de manera general a toda una institución, relativa a un servicio o enfocada a un padecimiento específico. Para cada una de estas se tendrían que buscar las variables adecuadas, significativas y relevantes. En diversas situaciones esto es, en el mejor de los casos, complicado lo que enclaustra a los modelos generados a situaciones específicas. La falta de los datos útiles, de variables indispensables, valores nulos o sesgados también son problemas detectados al tratar con datos estructurados. Esta situación se agrava si se involucran instituciones con sistemas de información heterogéneos que pueden almacenar los registros con distintos formatos, o codificaciones. Un punto adicional a considerar es que actualmente gran parte de la información registrada con respecto a la atención médica se realiza en texto libre, ya sea manual o electrónico y estos registros normalmente quedan sin analizar.

5.3. Trabajo futuro

Desde el punto de vista de las Ciencias de la Salud, un paso lógico a seguir es la experimentación utilizando el modelo propuesto para otras áreas y servicios con objeto de validar la facilidad de exportación a otros ámbitos. Así mismo, se motiva a incluir el modelo seleccionado al sistema de información del servicio analizado, con miras a tener información en tiempo real de los pacientes que arriban al hospital por esta vía, inicialmente como prueba piloto y posteriormente como implementación final. En el caso particular del hospital analizado, los sistemas de información con que cuenta son, en su mayoría desarrollados en el lenguaje de programación Java. En este caso, se pueden utilizar librerías especializadas para combinar Python con Java, una de las cuales es Jython [100].

Otro punto digno de analizar es en cuanto al uso de los métodos presentados para tratar los subregistros por códigos inespecíficos. En este trabajo se abordó el código relativo a *dolor*

agudo, sin embargo, existen otros códigos igualmente dañinos para la estadística reportada en los sistemas de salud y para la validez de los clasificadores entrenados. También es necesario aplicar técnicas de validación cruzada a los modelos presentados, pues para esta parte del trabajo no fueron realizados.

Un aspecto más profundo desde el punto de vista del aprendizaje automático es la evaluación de la calidad del texto sintético para tareas de balanceo de datos. Se requiere una medida que permita optimizar, de manera objetiva, el valor de los parámetros de la red que generan el texto sintético bajo la hipótesis de que, entre mejor sea la calidad del texto generado, el valor de la media armónica F_1 -score será mayor aumentando la confiabilidad de las clasificaciones. No obstante, no existe una herramienta cuantificable que permita tal comparación.

Otras experimentaciones sugeridas para mejorar la confiabilidad de las clasificaciones (elevar el valor F_1 -score), es el uso de algoritmos combinados, en particular se sugiere la clustereización de los datos previo a la clasificación final, se cree que al contar con datos más homogéneos entre sí, el resultado mejoraría. Así mismo, se pueden realizar experimentos con más fuentes de datos tales como resultados de laboratorios o estudios imagenológicos, incluso la combinación entre datos con distintos formatos

Finalmente, se sabe que el Deep Learning es un área actual que ha mostrado grandes avances en distintos ámbitos tales como la Inteligencia Artificial, Robótica y Procesamiento de Lenguaje Natural, entre otras, por lo que se recomienda realizar investigaciones similares a la presentada en este trabajo utilizando algunas de sus herramientas tales como: Redes Neuronales de Convolución (CNN), Redes Neuronales Recurrentes (RNN) o Generative Neural Networks (GAN).

Referencias

- [1] A. Azari, V. Janeja, A. Mohseni, “Predicting Hospital Length of Stay (PHLOS): A Multi-tiered Data Mining Approach”, 2012 IEEE 12th International Conference on Data Mining Workshops, 2012.
- [2] P. Tsai, P. Chen, Y. Chen, H. Song, H. Lin, F. Lin, Q. Huang, “Length of Hospital Stay Prediction at the Admission Stage for Cardiology Patients Using Artificial Neural Network”, Journal of Healthcare Engineering, vol. 2016, pp. 1-11, 2016.
- [3] A. Awad, M. Bader - El - Den, J. McNicholas, “Modeling and Predicting Patient Length of Stay: A Survey.”, International Journal of Advanced Scientific Research and Management, pp. 90-101, 2016.
- [4] Tsang-Hsiang Cheng, P. Hu, “A Data-Driven Approach to Manage the Length of Stay for Appendectomy Patients”, IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 39, no. 6, pp. 1339-1347, 2009.
- [5] T. Le, C. Kwok, E. Teo, K. Lee, “Analyzing Trends of Hospital Length of Stay Using Phase-Type Distributions”, 2011 IEEE 11th International Conference on Data Mining Workshops, 2011.
- [6] R. Pérez-Núñez, M. Mojarro-Íñiguez, M. Mendoza-García, S. Rosas-Osuna, M. Híjar, “Subestimación de la mortalidad causada por el tránsito en México: análisis subnacional”, Salud Pública de México, pp. 412-420, 2016.
- [7] M. Híjar, A. Chandran, R. Pérez-Núñez, J. Lunnen, J. Martín Rodríguez-Hernández, A. Hyder, “Quantifying the Underestimated Burden of Road Traffic Mortality in Mexico: A Comparison of Three Approaches”, Traffic Injury Prevention, vol. 13, no. 1, pp. 5-10, 2012.
- [8] M. Asim, M. Wasim, M. Ali, A. Rehman, “Comparison of feature selection methods in text classification on highly skewed datasets”, 2017 First International Conference on Latest trends in Electrical Engineering and Computing Technologies (INTELLECT), 2017.
- [9] R. Kumar, J. Novak, P. Raghavan, A. Tomkins, “Structure and evolution of blogspace”, Communications of the ACM, vol. 47, no. 12, p. 35, 2004.

- [10] A. Almashrafi, M. Elmontsri, P. Aylin, “Systematic review of factors influencing length of stay in ICU after adult cardiac surgery”, *BMC Health Services Research*, vol. 16, no. 1, 2016.
- [11] A. Gordon, A. Marshall, M. Zenga, “A Discrete Conditional Phase-Type Model Utilising a Survival Tree for the Identification of Elderly Patient Cohorts and Their Subsequent Prediction of Length of Stay in Hospital”, 2016 IEEE 29th International Symposium on Computer-Based Medical Systems (CBMS), 2016.
- [12] P. Hachesu, M. Ahmadi, S. Alizadeh, F. Sadoughi, “Use of Data Mining Techniques to Determine and Predict Length of Stay of Cardiac Patients”, *Healthcare Informatics Research*, vol. 19, no. 2, p. 121, 2013.
- [13] K. Ramaraju, “Predicting Healthcare Utilization by Patients Admitted for COPD Exacerbation”, *Journal of Clinical and Diagnostic Research*, 2016.
- [14] S. Rhodes, A. Patanwala, J. Cremer, E. Marshburn, M. Herman, F. Shirazi, P. Harrison-Monroe, C. Wendel, M. Fain, J. Mohler, A. Sanders, “Predictors of Prolonged Length of Stay and Adverse Events among Older Adults with Behavioral Health-Related Emergency Department Visits: A Systematic Medical Record Review”, *The Journal of Emergency Medicine*, vol. 50, no. 1, pp. 143-152, 2016.
- [15] M. Chuang, Y. Hu, C. Tsai, C. Lo, W. Lin, “The Identification of Prolonged Length of Stay for Surgery Patients”, 2015 IEEE International Conference on Systems, Man, and Cybernetics, 2015.
- [16] *Manual de Procedimientos del Servicio de Urgencias para Hospitales Generales*. Estado de México: Instituto de Salud del Estado de México, 2005.
- [17] L. Ávila-Burgos, C. Ventura-Alfaro, E. Hidalgo-Solórzano, M. Hajar-Medina, B. Aracena-Genao, A. Celis de la Rosa, “Atención de lesiones por tipo de causa externa en salas de urgencia en tres ciudades de México: Composición, frecuencia y gravedad”, *Rev Invest Clin*, vol. 4, no. 64, pp. 336-343, 2012.
- [18] L. Ávila-Burgos, C. Ventura-Alfaro, A. Barroso-Quiab, B. Aracena-Genao, “Las lesiones por causa externa en México. Lecciones aprendidas y desafíos para el Sistema Nacional de Salud”, Instituto Nacional de Salud Pública, Ciudad de México/Cuernavaca(MX), 2010.
- [19] “OPS OMS — Clasificación Internacional de Enfermedades”, Pan American Health Organization / World Health Organization, 2017. [En línea]. Disponible: http://www.paho.org/hq/index.php?option=com_content&view=article&id=3561%3A2010-clasificacion-internacional-enfermedades-cie&Itemid=2560&lang=es. [Accesado: 05- Jun- 2017].
- [20] “Heritage Health Prize — Kaggle”, [Heritagehealthprize.com](http://www.heritagehealthprize.com), 2017. [En línea]. Disponible: <http://www.heritagehealthprize.com>. [Accesado: 05- Jun- 2017].

- [21] M. Rouzbahman, A. Jovicic, M. Chignell, “Can Cluster-Boosted Regression Improve Prediction of Death and Length of Stay in the ICU?”, *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 3, pp. 851-858, 2017.
- [22] H. Dasenbrock, K. Liu, C. Devine, V. Chavakula, T. Smith, W. Gormley and I. Dunn, “Length of hospital stay after craniotomy for tumor: a National Surgical Quality Improvement Program analysis”, *Neurosurgical Focus*, vol. 39, no. 6, p. E12, 2015.
- [23] M. Durie, J. Darvall, D. Hadley and M. Tacey, “A - Code ICU - expedited review of critically ill patients is associated with reduced emergency department length of stay and duration of mechanical ventilation”, *Journal of Critical Care*, vol. 42, pp. 123-128, 2017.
- [24] A. Belderrar and A. Hazzab, “Hierarchical Genetic Algorithm and Fuzzy Radial Basis Function Networks for Factors Influencing Hospital Length of Stay Outliers”, *Healthcare Informatics Research*, vol. 23, no. 3, p. 226, 2017.
- [25] R. LaFaro, S. Pothula, K. Kubal, M. Inchiosa, V. Pothula, S. Yuan, D. Maerz, L. Montes, S. Oleszkiewicz, A. Yusupov, R. Perline, M. Inchiosa, “Neural Network Prediction of ICU Length of Stay Following Cardiac Surgery Based on Pre-Incision Variables”, *PLOS ONE*, vol. 10, no. 12, p. e0145395, 2015.
- [26] C. Gholipour, “Using an Artificial Neural Networks (ANNs) Model for Prediction of Intensive Care Unit (ICU) Outcome and Length of Stay at Hospital in Traumatic Patients”, *JOURNAL OF CLINICAL AND DIAGNOSTIC RESEARCH*, 2015.
- [27] A. Morton, E. Marzban, G. Giannoulis, A. Patel, R. Aparasu, I. Kakadiaris, “A Comparison of Supervised Machine Learning Techniques for Predicting Short-Term In-Hospital Length of Stay among Diabetic Patients”, *2014 13th International Conference on Machine Learning and Applications*, 2014.
- [28] R. Abbi, E. El-Darzi, C. Vasilakis, P. Millard, “A Gaussian Mixture Model Approach to Grouping Patients According to their Hospital Length of Stay”, *2008 21st IEEE International Symposium on Computer-Based Medical Systems*, 2008.
- [29] I. Nouaouri, A. Samet, H. Allaoui, “Evidential data mining for length of stay (LOS) prediction problem”, *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, 2015.
- [30] Q. Yang, X. Wu, “10 Challenging problems in data mining research”, *International Journal of Information Technology & Decision Making*, vol. 05, no. 04, pp. 597-604, 2006.
- [31] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow*, 1st ed. 2017.
- [32] M. Bekkar, H. Khelouane and T. Akrouf, “Evaluation Measures for Models Assessment over Imbalanced Data Sets”, *Journal of Information Engineering and Applications*, vol. 3, no. 10, pp. 27-39, 2013.
- [33] P. Cichosz, *Data mining algorithms*. Chichester: Wiley, 2015.

- [34] K. Stapor, “Evaluating and Comparing Classifiers: Review, Some Recommendations and Limitations”, *Advances in Intelligent Systems and Computing*, pp. 12-21, 2017.
- [35] R. Feldman, J. Sanger, *The text mining handbook*. New York: Cambridge University Press, 2013.
- [36] A. Srivastava, M. Sahami, *Text Mining Classification, Clustering, and Applications*, 1st ed. Chapman & Hall/CRC, 2009.
- [37] “Stop words”, [En.wikipedia.org](https://en.wikipedia.org/wiki/Stop_words), 2017. [En línea]. Disponible: https://en.wikipedia.org/wiki/Stop_words. [Accesado: 03-Jun-2017].
- [38] Aizawa A (2000) The Feature Quantity: An Information Theoretic Perspective of Tf-idf-like Measures. *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* 104-111.
- [39] Joachims T (1997) A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization. *Proceedings of ICML-97, 14th International Conference on Machine Learning* 143-151.
- [40] K. Du, M. Swamy, *Neural networks and statistical learning*, 1st ed. London: Springer, 2014.
- [41] C. Van Der Malsburg, “Frank Rosenblatt: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms”, *Brain Theory*, pp. 245-248, 1986.
- [42] D. Rumelhart, G. Hinton, R. Williams. “Learning internal representations by error propagation”. *California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.
- [43] Y. Bengio, P. Simard, P. Frasconi, “Learning long-term dependencies with gradient descent is difficult”, *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157-166, 1994.
- [44] “Understanding LSTM Networks – colah’s blog”, [Colah.github.io](https://colah.github.io/posts/2015-08-Understanding-LSTMs/), 2017. [En línea]. Disponible: <https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accesado: 01-Dec- 2017].
- [45] S. Hochreiter, J. Schmidhuber, “Long Short-Term Memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.
- [46] B. Lantz, *Machine learning with R*, 2nd ed. Packt Publishing, 2015.
- [47] M. Bowles, *Machine learning in Python*. Indianapolis, IN: John Wiley & Sons, 2015.
- [48] J. Bell, *Machine learning*, 1st ed. Indianapolis, Indiana: John Wiley & Sons, Inc., 2015.
- [49] “IBM Knowledge Center”, [Ibm.com](https://www.ibm.com/support/knowledgecenter/SSLVMB.22.0.0/com.ibm.spss.statistics.algorithms/alg_twostep.htm), 2017. [En línea]. Available: https://www.ibm.com/support/knowledgecenter/SSLVMB.22.0.0/com.ibm.spss.statistics.algorithms/alg_twostep.htm. [Accesado: 03- Jun- 2017].

- [50] B. Kitchenham, "A procedure for analyzing unbalanced datasets", *IEEE Transactions on Software Engineering*, vol. 24, no. 4, pp. 278-301, 1998.
- [51] T. Zhang, R. Ramakrishnan, M. Livny, "BIRCH", *Proceedings of the 1996 ACM SIGMOD international conference on Management of data - SIGMOD '96*, 1996.
- [52] C. Fraley, "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis", *The Computer Journal*, vol. 41, no. 8, pp. 578-588, 1998.
- [53] J. Banfield, A. Raftery, "Model-Based Gaussian and Non-Gaussian Clustering", *Biometrics*, vol. 49, no. 3, p. 803, 1993.
- [54] G. Lemaître, F. Nogueira, C. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning", *Jmlr.org*, 2017. [En línea]. Disponible: <http://jmlr.org/papers/v18/16-365.html>. [Accesado: 26- Nov- 2017].
- [55] B. Santoso, H. Wijayanto, K. Notodiputro, B. Sartono, "Synthetic Over Sampling Methods for Handling Class Imbalanced Problems : A Review", *IOP Conference Series: Earth and Environmental Science*, vol. 58, p. 012031, 2017.
- [56] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique", *J. Artificial Intelligence Research*, vol. 16, pp. 321-357, 2002.
- [57] M. Bach, A. Werner, J. ?ywiec and W. Pluskiewicz, "The study of under- and over-sampling methods? utility in analysis of highly imbalanced data on osteoporosis", *Information Sciences*, vol. 384, pp. 174-190, 2017.
- [58] J. Wang, W. Lu, H. Loh, "P-SMOTE: One Oversampling Technique for Class Imbalanced Text Classification", *Volume 2: 31st Computers and Information in Engineering Conference, Parts A and B*, 2011.
- [59] D. Arthur, S. Vassilvitskii, "K-means++: the advantages of careful seeding", In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007.
- [60] "Población. Número de habitantes", *Cuentame.inegi.org.mx*, 2017. [En línea]. Disponible: <http://cuentame.inegi.org.mx/poblacion/defunciones.aspx?tema=P>. [Accesado : 05-Jun- 2017].
- [61] W. Ng, J. Hu, D. Yeung, S. Yin, F. Roli, "Diversified Sensitivity-Based Undersampling for Imbalance Classification Problems", *IEEE Transactions on Cybernetics*, vol. 45, no. 11, pp. 2402-2412, 2015.
- [62] B. Yap, K. Rani, H. Rahman, S. Fong, Z. Khairudin, N. Abdullah, "An Application of Oversampling, Undersampling, Bagging and Boosting in Handling Imbalanced Datasets", *Lecture Notes in Electrical Engineering*, pp. 13-22, 2013.
- [63] "Welcome to Python.org", *Python.org*, 2018. [En línea]. Disponible: <https://www.python.org/>. [Accesado: 18- Abr- 2018].
- [64] "Project Jupyter", *Jupyter.org*, 2018. [En línea]. Disponible: <https://jupyter.org/>. [Accesado: 28- Abr- 2018].

- [65] “Python :: Anaconda Cloud”, Anaconda.org, 2018. [En línea]. Disponible: <https://anaconda.org/anaconda/python>. [Accesado: 28- Abr- 2018].
- [66] “Graphlab Create. Fast, Scalable Machine Learning Modeling in Python.”, Turi, 2018. [En línea]. Disponible: <https://turi.com/>. [Accesado: 28- Abr- 2018].
- [67] “Get Fedora”, Getfedora.org, 2018. [En línea]. Disponible: <https://getfedora.org/es/>. [Accesado: 28- Abr- 2018].
- [68] “Deep Learning Nanodegree — Udacity”, Udacity.com, 2018. [En línea]. Disponible: <https://www.udacity.com/course/deep-learning-nanodegree-nd101>. [Accesado: 07- Abr- 2018].
- [69] “zackthoutt/got-book-6”, GitHub, 2018. [En línea]. Disponible: <https://github.com/zackthoutt/got-book-6/blob/master/README.md>. [Accesado: 07- Abr- 2018].
- [70] “udacity/deep-learning”, GitHub, 2018. [En línea]. Disponible: <https://github.com/udacity/deep-learning/tree/master/tv-script-generation>. [Accesado: 27- Abr- 2018].
- [71] “RomelTorres/DLND-tv-script-generation”, GitHub, 2018. [En línea]. Disponible: https://github.com/RomelTorres/DLND-tv-script-generation/blob/master/dlnd_tv_script_generation.ipynb. [Accesado: 27- Abr- 2018].
- [72] “jg1141/tv-script-generation”, GitHub, 2018. [En línea]. Disponible: <https://github.com/jg1141/tv-script-generation>. [Accesado: 27- Abr- 2018].
- [73] “TensorFlow”, TensorFlow, 2017. [En línea]. Disponible: <https://www.tensorflow.org/>. [Accesado: 20- Nov- 2017].
- [74] “PostgreSQL: The world’s most advanced open source database”, Postgresql.org, 2018. [En línea]. Disponible: <https://www.postgresql.org/>. [Accesado: 18- Abr- 2018].
- [75] J. McAuley, “Amazon review data”, Jmcauley.ucsd.edu, 2018. [En línea]. Disponible: <http://jmcauley.ucsd.edu/data/amazon/links.html>. [Accesado: 17- Abr- 2018].
- [76] “Home - PubMed - NCBI”, Ncbi.nlm.nih.gov, 2018. [En línea]. Disponible: <https://www.ncbi.nlm.nih.gov/pubmed/>. [Accesado: 17- Abr- 2018].
- [77] R. Blagus, L. Lusa, “SMOTE for high-dimensional class-imbalanced data”, BMC Bioinformatics, vol. 14, no. 1, p. 106, 2013.
- [78] “imblearn.over_sampling.SMOTE - imbalanced-learn 0.3.0 documentation”, Contrib.scikit-learn.org, 2018. [En línea]. Disponible: http://contrib.scikit-learn.org/imbalanced-learn/stable/generated/imblearn.over_sampling.SMOTE.html. [Accesado: 01- May- 2018].
- [79] “IBM SPSS Statistics - Visión general - México”, Ibm.com, 2018. [En línea]. Disponible: <https://www.ibm.com/mx-es/marketplace/spss-statistics>. [Accesado: 02- May- 2018].

- [80] “Generador de números aleatorios”, [Es.wikipedia.org](https://es.wikipedia.org/wiki/Generador_de_n%C3%BAmeros_aleatorios), 2018. [En línea]. Disponible: https://es.wikipedia.org/wiki/Generador_de_n%C3%BAmeros_aleatorios. [Accesado: 04- May- 2018].
- [81] “Violencia”, Organización Mundial de la Salud, 2018. [En línea]. Disponible: <http://www.who.int/topics/violence/es/>. [Accesado: 24- Abr- 2018].
- [82] M. Bejarano, L. Rendón, “Lesiones de causa externa en menores y mayores de 18 años en un hospital colombiano”, *Revista Panamericana de Salud Pública*, vol. 25, no. 3, pp. 234-241, 2009.
- [83] B. Díaz-Apodaca, F. De Cosio, G. Moye-Elizalde, F. Fornelli-Laffon, “Egresos por lesiones externas en un hospital de Ciudad Juárez, México”, *Revista Panamericana de Salud Pública*, vol. 31, no. 5, pp. 443-446, 2012.
- [84] C. Huang, Y. Li, C. Loy, X. Tang, “Learning Deep Representation for Imbalanced Classification”, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [85] H. Lo, L. Newmark, C. Yoon, L. Volk, V. Carlson, A. Kittler, M. Lippincott, T. Wang, D. Bates, “Electronic Health Records in Specialty Care: A Time-Motion Study”, *Journal of the American Medical Informatics Association*, vol. 14, no. 5, pp. 609-615, 2007.
- [86] J. Cruz, J. Shabosky, M. Albrecht, T. Clark, J. Milbrandt, S. Markwell, J. Kegg, “Typed versus Voice Recognition for Data Entry in an Electronic Health Record: Emergency Department Physician Time Utilization and Interruptions”, *Western Journal of Emergency Medicine*, vol. 15, no. 4, 2014.
- [87] F. Lucini, F. S. Fogliatto, G. C. da Silveira, J. L. Neyeloff, M. Anzanello, R. de S. Kuchenbecker and B. D. Schaan, “Text mining approach to predict hospital admissions using early medical records from the emergency department”, *International Journal of Medical Informatics*, vol. 100, pp. 1-8, 2017.
- [88] I. Vlad Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, Y. Bengio, “A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues”, *CoRR*, vol. 160506069, no. 160506069, 2016.
- [89] A. Esteban, E. Lloret, “Propuesta y desarrollo de una aproximación de generación de resúmenes abstractivos multigénero”, [Dx.doi.org](https://dx.doi.org/10.26342/2017-58-5412), 2018. [En línea]. Disponible: <http://dx.doi.org/10.26342/2017-58-5412>. [Accesado: 08- May- 2018].
- [90] E. Agirre, D. Cer, M. Diab, A. Gonzalez-agirre, “SemEval-2012 Task 6: A Pilot on Semantic Textual Similarity”, *First Joint Conference on Lexical and Computational Semantics*, pp. 385-393, 2012.
- [91] C. Tao, L. Mou, D. Zhao, R. Yan, “An Unsupervised Method for Automatic Evaluation of Open-Domain Dialog Systems”, *CoRR*, vol. 170103079, no. 170103079, 2017.

- [92] Y. Yang, S. Yuan, D. Cer, S. Kong, N. Constant, P. Pilar, H. Ge, Y. Sung, B. Strope, R. Kurzweil, "Learning Semantic Textual Similarity from Conversations", Arxiv.org, 2018. [En línea]. Disponible: <https://arxiv.org/abs/1804.07754v1>. [Accesado: 08- May- 2018].
- [93] "NVIDIA sobre la computación de GPU y la diferencia entre GPU y CPU", La.nvidia.com, 2018. [En línea]. Disponible: <http://la.nvidia.com/object/what-is-gpu-computing-la.html>. [Accesado: 09- May- 2018].
- [94] "Using GPUs — TensorFlow", TensorFlow, 2018. [En línea]. Disponible: https://www.tensorflow.org/programmers_guide/using_gpu. [Accesado: 09- May- 2018].
- [95] J. Consuelo Estrada, O. Portillo Rodríguez, L. Gaona Valle, "Minería de textos para el análisis del subregistro de lesiones por causa externa en el servicio de urgencias de un hospital de tercer nivel", Research in Computing Science, no. 134, pp. 177-190, 2017.
- [96] J. Consuelo Estrada, L. Morales Díaz, C. González Castillo, L. Gaona Valle, O. Portillo Rodríguez, J. Rodríguez Arce, "Epidemiología de las lesiones por accidentes de tránsito en el servicio de urgencias de un hospital de tercer nivel", Inteligencia Epidemiológica, vol. 7 no. 1, pp. 11-16, 2017.
- [97] "Urgencias — Centro Médico Lic. Adolfo López Mateos", Salud.edomex.gob.mx, 2018. [En línea]. Disponible: <http://salud.edomex.gob.mx/cmalmateos/paginageneral.html>. [Accesado: 06- Jul- 2018].
- [98] "Congreso Mexicano de Inteligencia Artificial", Comia.org.mx, 2017. [En línea]. Disponible: <http://www.comia.org.mx/2017/>. [Accesado: 05- Jun- 2017].
- [99] M. Chuang, Y. Hu, C. Lo, "Predicting the prolonged length of stay of general surgery patients: a supervised learning approach", International Transactions in Operational Research, vol. 25, no. 1, pp. 75-90, 2016.
- [100] S. Chekanov, Scientific data analysis using Jython scripting and Java. London: Springer, 2010.
- [101] "Diccionarios en Python", DevCode Tutoriales, 2018. [En línea]. Disponible: <https://devcode.la/tutoriales/diccionarios-en-python/>. [Accesado: 07- Jun- 2018].

Apéndice A

Artículos publicados

Los resultados obtenidos en las distintas etapas del proyecto proporcionaron información para la publicación de tres artículos, en este apéndice se presenta el resumen de cada uno.

A.1. Epidemiología de las lesiones por accidentes de tránsito en el servicio de urgencias de un hospital de tercer nivel

Resumen

Introducción: Las Lesiones Causadas por Tránsito (LCT) representan un problema de salud pública. En los servicios de urgencias hospitalarias debe considerarse la asignación de recursos humanos y materiales para una atención médica adecuada. **Objetivo:** Describir la epidemiología de las LCT en el servicio de urgencias de un hospital de tercer nivel. **Material y métodos:** Estudio retrospectivo, observacional con datos del registro electrónico del servicio de urgencias de un hospital de tercer nivel, se consideraron casos con diagnóstico CIE-10 relacionado con LCT. El periodo estudiado abarca del 1 de septiembre de 2010 al 31 de mayo de 2015. **Resultados:** 1,604 pacientes de 14 a 93 años, el 70 % fueron hombres. El 20.45 % consumieron alcohol previo al accidente. Los vehículos involucrados con mayor frecuencia fueron automóvil, bicicleta, motocicleta y autobús. La consecuencia más frecuentemente fue la fractura con un 29.12 % y las regiones anatómicas más afectadas fueron cabeza y cuello con 52.61 % y el 51.68 % tuvo necesidad de hospitalización. **Conclusión:** La epidemiología de las LCT y las características de los traumas resultantes pueden coadyuvar a la administración de recursos hospitalarios y al diseño de programas sociales orientados a la prevención.

Palabras clave: Accidente de tránsito, lesiones, trauma.

Referencia

J. Consuelo-Estrada, L. Morales-Díaz, C. González-Castillo, L. Gaona-Valle, O. Portillo-

Rodríguez, J. Rodríguez Arce, “Epidemiología de las lesiones por accidentes de tránsito en el servicio de urgencias de un hospital de tercer nivel”, *Inteligencia Epidemiológica*, vol. 7 no. 1, pp. 11-16, 2017.

A.2. Minería de textos para el análisis del subregistro de lesiones por causa externa en el servicio de urgencias de un hospital de tercer nivel

Resumen

Las Lesiones por Causa Externa (LCE) representan un serio problema de salud, sin embargo el diagnóstico mediante códigos inespecíficos puede subestimar su gravedad. Esto es común en muchos hospitales, afectando la asignación de recursos y prioridades en salud. Analizando datos de un servicio de urgencias de un hospital de tercer nivel, se implementaron cuatro modelos predictivos: regresión logística con texto (TFIDF), regresión logística, árboles de decisión y boosting para determinar el porcentaje de diagnósticos de dolor agudo que pudieran ser LCE. El método más exacto fue el basado en texto y estima que, 12,240 (82.56 % n=14,826) de los dolores agudos son LCE. El porcentaje de LCE subestimado como resultado del uso códigos inespecíficos es alto y la minería de textos es una opción viable para su estimación.

Palabras clave: Minería de textos, machine learning, código inespecífico, lesiones por causa externa.

Referencia

J. Consuelo-Estrada, O. Portillo-Rodríguez, L. Gaona-Valle, “Minería de textos para el análisis del subregistro de lesiones por causa externa en el servicio de urgencias de un hospital de tercer nivel”, *Research in Computing Science*, no. 134, pp. 177-190, 2017.

A.3. Lesiones por causa externa en el servicio de urgencias de un hospital en un periodo de cinco años

Resumen

Introducción: En el Estado de México no existen investigaciones que proporcionen información para toma de decisiones y administración de recursos relacionados con la atención de las lesiones por causa externa (LCE). **Objetivo:** Describir las LCE en un servicio de urgencias durante un periodo de cinco años. **Método:** Se diseñó un estudio retrospectivo con pacientes que ingresaron al servicio de urgencias (2010-2015) por diagnóstico de LCE. Se

A.3. LESIONES POR CAUSA EXTERNA EN EL SERVICIO DE URGENCIAS DE UN HOSPITAL

realizó análisis descriptivo y de clúster. **Resultados:** En el servicio de urgencias, 16.59 % de las atenciones derivaron de LCE. Se incluyeron 16 567 pacientes de 14 a 99 años ($\bar{x} = 37.7$, $DE = 17.28$), 69.2 % fue del sexo masculino. Las LCE principalmente ocurrieron en la vía pública (26.3 %) y en el hogar (23.7 %). Las causas más frecuentes fueron agresiones fuera del hogar (32.7 %), en promedio a los 34 años; caídas (25 %) en promedio a los 45 años; accidentes ocasionados por vehículos de motor (9.7 %), en promedio a los 33 años. El análisis por clúster identificó cuatro grupos: agresiones fuera del hogar 32.7 % (5417), contactos traumáticos 26.30 % (4363), accidentes de tránsito 15.9 % (2,640) y caídas 25 % (4147). **Conclusión:** Las LCE relacionadas con vehículos de motor mostraron consecuencias más severas.

Palabras clave: Lesiones por causa externa. Servicios de urgencias. Accidentes de tránsito. Accidentes en el hogar.

J. Consuelo-Estrada, L. Gaona-Valle and O. Portillo-Rodríguez, “Lesiones por causa externa en el servicio de urgencias de un hospital en un periodo de cinco años”, *Gaceta Médica de México*, Para ser publicado.

Apéndice B

Herramientas de software

B.1. GraphLab Create

Graphlab Create es un paquete de análisis de datos para trabajar con el lenguaje de programación Python. En esta tesis se utilizó una licencia para uso académico. Entre las principales características del paquete se encuentran:

- Permite realizar análisis y procesamiento de datos, manejando terabytes, incluso con poca memoria RAM disponible.
- Exploración y visualización de datos de forma gráfica y estadística.
- Análisis de redes o teoría de grafos, utilizando *SFrames* para almacenar vértices y arcos.
- Creación de modelos predictivos con herramientas de machine learning.

En este anexo se describe brevemente el uso de métodos de machine learning para la creación de modelos predictivos. Este apartado es un extracto del manual de usuario presentado en el sitio web del paquete y se hace referencia específicamente a las herramientas utilizadas en esta tesis. Previo a su uso, se requiere una instalación que depende del sistema operativo anfitrión y cuya descripción detallada se encuentra en [66].

Una vez instalado el paquete, el primer paso, como en la mayoría de los códigos Python, es la importación del mismo al entorno de desarrollo ejecutando el siguiente código.

```
import graphlab as gl
```

La carga o importación del texto dependerá del formato y la fuente de los mismos. Para propósitos de este trabajo, los datos fueron almacenados previamente en archivos separados

por comas (*csv*) y que contiene dos variables, el texto de la nota medica y la etiqueta correspondiente a la estancia del paciente, codificada como *normal* o *prolongada*, según lo descrito en la sección 3.3. Al ser leídos por el programa, los datos fueron almacenados en una estructura de datos conocida como *SFrame* propia de la herramienta la cual cuenta con un método específico para dicha tarea (*read_csv*). Como se describe a continuación, el texto se almacena en la variable *csyn*.

```
csyn = gl.SFrame.read_csv(
    'cm_notas_sinteticas.csv',
    delimiter=',',
    header=True,
    verbose=False)
```

Observando el código anterior se nota que al parámetro *header* se le asigna el valor *true*, con esto se obtiene que los campos leídos desde el archivo conserven el nombre de la columna, también especificados en el archivo *csv*.

Graphlab cuenta con un conjunto de métodos para análisis de textos, uno de los utilizados para el desarrollo de este trabajo es la obtención de la representación TFIDF a partir de texto libre. El método aplica la formula presentada en la sección 2.2 y *convierte* el texto a una representación tabular con cada palabra representando una variable.

```
csyn['tfidf'] = gl.text_analytics.tf_idf(csyn['nota'])
```

La instrucción anterior agrega una columna a la variable *csyn* que contiene un *diccionario*¹ por cada registro de texto leído, con su respectiva representación TFIDF. La Figura B.1 muestra la estructura obtenida ejecutando el código anterior.

Uno de los aspectos más utilizados en el desarrollo de modelos predictivos con algoritmos de machine learning, es la división del conjunto de datos en los subconjuntos de entrenamiento, validación y prueba. En Graphlab este procedimiento se realiza empleando un porcentaje y, opcionalmente, una semilla aleatoria, para variar los registros que son elegidos en el particionamiento, como se explicó en las secciones 3.2 y 3.3. Anteriormente también se ha descrito, que para el análisis de balanceo de clases se utilizan distintas semillas aleatorias, 33 en total. En el siguiente segmento de código se inicia el vector denominado *seeds_list* con dichos valores (se muestra de forma abreviada). Para cada una de las semillas que contiene el vector, se realiza el proceso de validación.

¹En Python, un diccionario es una estructura de datos que contiene una etiqueta y un valor asociado [101]

etiqueta	nota	tfidf
normal	CASADO ESCOLARIDAD DIABETICO AÑOS EVOLUC ...	{'anafilactico': 16.402771910477217, ...}
normal	PREGUNTADOS NEGADOS INICIA HOY PRESENTANDO ...	{'paciente': 0.38884588456851826, ...}
prolongada	ORIGINARIA RESIDENTE ZINACANTEPEC RELIGION ...	{'normocefalo': 1.4955020922951414, ...}
normal	ALERGIAS NEGADO QX NEGADO FX NEGADO NIEGA ...	{'rodilla': 2.6751365206445645, ...}

Figura B.1: Representación de los datos en formato de texto libre y su correspondiente representación TFIDF

```
seeds_list = [1,5,10,...,5000]
for seed in seeds_list:
    train, rest_cm = csyn.random_split(.8, seed=seed)
    validation, test = rest_cm.random_split(.5, seed=seed)
    train = train.append(train_data_csyn)
```

En la primer línea del código dentro del ciclo *for*, el método *random_split* toma el 80% de los datos y los almacena en la primer variable de salida *train*. El restante 20% lo guarda en una variable auxiliar (*rest_cm*) que posteriormente es dividida al 50% para formar los conjuntos de validación y prueba.

El siguiente segmento de código corresponde a los pasos realizados en el proceso de validación de los modelos (en este caso para el clasificador LR, para el resto de los algoritmos el proceso es similar). Este código corresponde con el ciclo interno presentado en el diagrama de la Figura 3.2.

```
target = 'etiqueta'
l2_penalty_list = [0, 4,10, 100,200,300]
for L2 in l2_penalty_list:
    model = gl.logistic_classifier.create(train,
                                         target = target,
                                         features=['tfidf'],
                                         l2_penalty=L2,
                                         validation_set=None,
                                         verbose=False)
    train_resL2] = model.evaluate(train)['f1_score']
    validation_res[L2] = model.evaluate(validation)['f1_score']
```

```
L2 = max(validation_res , key=validation_res.get)
```

El método de evaluación del modelo *model.evaluate* recibe como parámetro el subconjunto de datos con el que se desea evaluar el modelo, en este caso se obtienen con los datos de entrenamiento y los datos de validación. Dependiendo del parámetro a evaluar, se puede variar entre *accuracy*, *f1-score*, *precision*, *recall* o *auc*. En este trabajo se utilizan solamente la *accuracy* y la media armónica *f1-score*. Por otro lado, los valores de los vectores *train_res* y *validation_res* pueden ser graficados para obtener una representación visual de los resultados. Este proceso se realizó en la fase 2 del proyecto y las gráficas se presentan en las Figuras 4.4, 4.6, 4.7, 4.8, variando el parámetro utilizado de acuerdo al algoritmo evaluado.

El siguiente paso consiste en utilizar el valor de la variable L2 del código anterior para obtener el modelo óptimo y evaluarlo con los datos del subconjunto de prueba almacenado previamente en la variable *test*. Para tal efecto se ejecuta el siguiente código.

```
model = gl.logistic_classifier.create(  
    train, target = target ,  
    features=['tfidf'],  
    l2_penalty=L2,  
    validation_set=None,  
    verbose=False)  
evaluation = model.evaluate(test)
```

Finalmente, los valores contenidos en la variable *evaluation* fueron almacenados en una base de datos PostgreSQL para su posterior análisis estadístico.