

Evaluación de modelos y estado del arte del análisis de sentimientos con opiniones en español

Daniel Santa Rendón, Daniel Torres González, Santiago Quintero Hincapié

*Ingeniería de sistemas, Universidad de Antioquia
Medellín, Colombia*

santiago.quintero@udea.edu.co
daniel.santar@udea.edu.co
daniel.torresg@udea.edu.co

Resumen

El análisis de sentimientos es una herramienta que está ganando cada vez más popularidad por sus grandes ventajas y beneficios de uso que consiste en clasificar un texto como positivo, negativo o neutro. Este se ha convertido en una herramienta muy útil para analizar la percepción acerca de un tema, producto o servicio, plasmada en un texto escrito. Para muchas entidades, tanto del sector privado como del sector público, contar con una herramienta que pueda analizar y clasificar de forma automática como positivo, negativo o neutro, la gran cantidad de información que se genera en torno a sus marcas, servicios o productos, se ha transformado en un componente indispensable al momento de tomar decisiones corporativas. En este artículo se van a exponer experimentos realizados con diferentes modelos de machine learning, la transformación que se le hizo a las opiniones y los resultados obtenidos. Es importante aclarar que este trabajo quedó en un punto en el cual se puede seguir investigando y probando otro tipo de representaciones y modelos, por tal razón promovemos el uso de los resultados obtenidos en este trabajo.

Introducción

Debido a que no existe la suficiente investigación de los métodos de análisis de sentimientos enfocados a las opiniones de hoteles en español, específicamente de la ciudad de Medellín, en este trabajo se busca hacer un repaso de las investigaciones sobre el análisis de sentimientos que se han realizado hasta el momento, con el fin de conocer los métodos usados para la solución de un problema, los contextos abordados y las bases de datos utilizadas en estas investigaciones. Y así, conocer qué han encontrado los demás autores y que sirve para este proyecto, para posteriormente evaluar los métodos hallados y elegir o modificar uno de estos para conseguir los mejores resultados con opiniones de hoteles en español de Medellín.

Para esto realizará una revisión sistemática de literatura en bases de datos confiables como la IEEE, Springer link, entre otras.

En la sección 1 se hablará de los antecedentes, en donde queda plasmado, de forma cronológica, la investigación sobre análisis de sentimientos y se exponen los métodos que se han aplicado a este campo pero en el contexto de las opiniones de hoteles, en la sección 2 se toman los métodos encontrados y se identifican los pros y contras de cada método, y para finalizar, en la sección 3 se formulan las conclusiones de la investigación realizada.

Palabras clave

Análisis de sentimientos, características lingüísticas, matriz de términos en documentos, vectores semánticos, machine learning, deep learning.

1. Antecedentes

El análisis de sentimientos, también denominado minería de opiniones, es el campo de estudio que analiza las opiniones, sentimientos, evaluaciones, valoraciones, actitudes y emociones de las personas hacia entidades como productos, servicios, organizaciones, individuos, temas, eventos y sus atributos (Bing Liu. 2012).

El análisis de sentimientos ya ha sido abordado en diferentes niveles tales como: a nivel de documentos, a nivel de oraciones y a nivel de entidades y aspectos, este último llamado también nivel de características (Hu and Liu. 2004).

Los documentos de opinión no solo existen en la web (a menudo llamado datos externos); muchas organizaciones tienen datos internos, como por ejemplo, *feedback* de clientes recolectados de e-mails y call centers y encuestas conducidas por organizaciones (Bing Liu. 2015), es de vital importancia analizar estos tipos de datos para así poder resumir la opinión de los usuarios.

Es importante aclarar que el nivel de documento se refiere a un conjunto de opiniones que están dentro de un mismo contexto, es decir, se van a analizar oraciones que opinan sobre un mismo producto o servicio, en este caso sobre hoteles de Medellín, mientras que a nivel de oraciones no se dispone de un conjunto de datos (opiniones) relacionadas entre sí, es decir, son opiniones que no están dentro de un mismo contexto.

En este proyecto nos centraremos en el análisis de sentimientos a nivel de documento, específicamente en opiniones de hoteles en español de Medellín. Sabiendo que el análisis de sentimientos a este nivel es quizás el más extenso tema de estudio en el campo del análisis de sentimientos, especialmente en sus primeros días (Surveys by Pang and Lee, 2008; Liu, 2012). Y sabiendo también que un documento está compuesto por párrafos y oraciones, inicialmente la tarea es determinar si estos expresan una opinión o sensación positiva, negativa o neutra, aunque también existen otros dos tipos de opiniones, opiniones regulares y opiniones de comparación, lo cual pone las cosas aún más interesantes y retadoras (Jindal and Liu. 2006b). Este nivel de análisis está altamente relacionado con la clasificación de subjetividad (Wiebe et al. 1999). Se discutirá n los principales temas de investigaciones del análisis de sentimientos y su estado del arte en lo concerniente a algoritmos.

Otras investigaciones también han analizado las frases (Wilson et al. 2004), pero el nivel de las oraciones aún no es suficiente, por ejemplo, "*Apple* lo está haciendo muy bien en esta pésima economía." en este nivel de análisis implícitamente se asume que cada párrafo expresa una opinión en una única entidad como por ejemplo un único producto o servicio, así que no es aplicable a documentos que evalúan o comparen múltiples entidades, para los cuales es necesario un análisis más detallado (Bing Liu. 2015).

Por otra parte se tiene el concepto de subjetividad que ha sido ampliamente usado en análisis de sentimientos. Este concepto ha causado algunas confusiones entre investigadores. En muchos artículos, ser subjetivo y ser poseedor de sentimientos son considerados como equivalentes, pero no son lo mismo. Una oración objetiva establece alguna información objetiva, mientras una frase subjetiva expresa algún sentimiento personal, punto de vista, juicio, o creencias. Un ejemplo de una frase objetiva es "*The iphone is an Apple product*", un ejemplo de una frase subjetiva es "*I like the iphone*" la tarea es determinar ya sea si una oración es subjetiva u objetiva, esto es llamado clasificación de subjetividad (Wiebe Janyce and Ellen Riloff. 1999).

La manera correcta de abordar este análisis de sentimientos es identificando palabras y modismos que demuestran una opinión o sensación dentro de una oración, esta lista de palabras y modismos es llamada léxico de sentimientos, sin embargo, aunque ésta es necesaria para realizar un correcto análisis de sentimientos, no es suficiente (Bing Liu. 2012).

En general, dado un grupo de opiniones, el análisis de sentimientos consistirá en lo siguiente: extracción y

categorización de entidades, aspectos, opinión del titular, fecha de la opinión dada y sentimiento de la opinión (Liu. 2006, 2011). Esta clasificación se puede llevar a cabo con algoritmos de *machine learning*, debido a que la clasificación de sentimientos se reduce a un problema de clasificación de texto, algunos métodos existentes de aprendizaje supervisado pueden ser directamente aplicados, tal como clasificación de *naïve Bayes* o *support vector machines* (SVM) (Joachims, 1999; Shawe-Taylor and Cristianini, 2000).

El análisis de sentimientos en opiniones de hoteles ha sido abordado en otros idiomas con resultados interesantes, como por ejemplo en el caso de Lisa Medrouk y Anna Pappa quienes trabajaron particularmente en reseñas de hoteles y restaurantes que están escritas en diferentes idiomas tales como francés, inglés y griego. En este caso, ellas para el enfoque de aprendizaje automático, utilizaron dos redes neuronales profundas diferentes, Redes neuronales convolucionales (ConvNets) y Redes neuronales recurrentes (RNN).

El modelo de aprendizaje que ellas aplicaron explota la información a nivel de n-gramas y logra una alta precisión para la polaridad de los sentimientos y la clasificación de temas de acuerdo con las pruebas y resultados experimentales. teniendo en cuenta sus resultados, ellas tienden a promover el inicio de un enfoque simple pero poderoso para alimentar redes profundas en un contexto multilingüe. (Medrouk y Pappa, 2018).

Otra solución es la planteada por Xiaobo Zhang y Qingsong Yu, en la cual plantean el uso de un vector de características reducido por medio de la herramienta Word2Vec y el algoritmo de agrupación ISODATA. Emplear esta herramienta y este algoritmo trae ventajas destacables como la posibilidad de usar palabras únicas y de evitar la necesidad de conocer mucho de la base de datos. Por otro lado, implementar Word2Vec podría provocar que el sentido de las palabras no se capture de manera independiente, por ejemplo, dos palabras escritas iguales con más de un sentido, serían clasificadas como iguales, así mismo, al combinarse con el algoritmo ISODATA, podrá llevar mucho tiempo si la base de datos es muy desestructurada, ya que por sí solo, tiende a tardar. (Zhang, X., & Yu, Q. 2017)

Por otra parte, Qianjun Shuai *et al.* llevaron a cabo un análisis de opinión sobre las revisiones de hoteles sacadas de Doc2vec y algoritmos de machine learning como Support Vector Machine (SVM), Logistic Regression y Naïve Bayes. Evaluaron el rendimiento de estos algoritmos usando índices de precisión, tasa de recuperación y F-measure. En este caso el que mejor dio resultado fue el SVM. Proponen para futuras investigaciones, mejorar los resultados usando más datos o mezclar el vector de características y el diccionario de sentimientos. Además, también consideran el uso de algoritmos de *deep learning* como las redes neuronales convolucionales. (Shuai, Q., Huang, Y., Jin, L., & Pang, L. 2018)

Aunque estos métodos dan resultados positivos, hay que tener en cuenta, aparte de las contras que tiene cada método, que no han sido entrenados y validados con

opiniones de hoteles en español, y mucho menos, de hoteles de Medellín. Es importante aclarar que cada idioma maneja unos modismos particulares los cuales hacen variar drásticamente el significado de una opinión y por lo tanto, el resultado del algoritmo.

2. Discusión

Habiendo investigado sobre las aplicaciones que se han hecho usando el análisis de sentimientos como herramienta fundamental para mejorar productos y servicios y el excelente resultado que se ha tenido, es interesante y lógico ver hasta dónde puede llegar a ayudar su aplicación en la ciudad de Medellín específicamente dentro del contexto de hoteles.

En el siguiente diagrama se puede visualizar de manera general la metodología que se llevó a cabo para el desarrollo de este trabajo.



Figura 1: Pasos para evaluar el análisis de sentimiento

2.1. Revisión de Datasets

El Datasets obtenido está compuesto de opiniones hechas por usuarios que utilizaron servicios de hoteles, las cuales reflejan la satisfacción del usuario con el uso de este servicio.

Las 514 opiniones sobre hoteles ubicados en la ciudad de Medellín, fueron obtenidas de booking.com, TripAdvisor.co y Expedia.es y su calificación fue dada por los usuarios de los hoteles, los cuales fueron seleccionados al azar. Las opiniones clasificadas como positivas son aquellas calificadas como good, very good, excellent y exceptional, y negativas las demás.

2.2. Pre-procesamiento

Debido a que los datos fueron obtenidos directamente de las plataformas booking.com, TripAdvisor.co y Expedia.es, no fue necesario hacer un pre-procesamiento ya que cada opinión fue extraída y leída por un humano que confirmaba la falta de caracteres especiales, como emojis, @ (arrobos), # (hashtags), etc. Para su correcto procesamiento.

2.3. Transformación

Inicialmente se importaron las siguientes librerías para desarrollar el proceso de Transformación.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
```

Figura 2: Librerías necesarias para Tokenizar

Después se importó la base de datos usando Pandas, esta librería fue utilizada para organizar la base de datos en una matriz y de esta manera darle un mejor tratamiento a los datos.

Una vez importada la base de datos, se procedió a separar las entradas y salidas (Opiniones y sentimiento), para luego usar CountVectorizer y poder organizar las opiniones en un vector, en este caso agrupados en conjuntos de 1 y 2 palabras.

Los datos en el vector resultante tomarían la siguiente estructura:

```
'10',
'10 argentina',
```

Figura 3: Pasos para evaluar el análisis de sentimiento

Una vez se tiene el vector con todo el vocabulario, se pasa a crear la matriz Bag Of Words, en la siguiente figura se puede ver cómo es su estructura:

```
[[1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 3 0]]
```

Figura 4: BagOfWords

Se puede notar que la matriz resultante tiene muchos ceros, esto es debido a que, en cada posición de la matriz está la frecuencia que tiene cada conjunto de palabras (columna) en cada opinión (fila) y como cada opinión tiene un número reducido de palabras, comparado con el tamaño del vocabulario resultante, este queda mayormente lleno de ceros.

2.2. Clasificación y evaluación

Continuando con la clasificación y evaluación de los textos se importaron las siguientes librerías:

```
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import KFold
```

Figura 5: Librerías para entrenamiento y testeo de modelos

La librería **KFold** se utilizó para implementar la metodología de validación cross-validation, para hacer el entrenamiento y validación de los diferentes modelos probados, dividiendo las muestras en 10 grupos (folds), y la librería **confusion_matrix** se utilizó para hallar las medidas de error para cada uno de los modelos, para la comparación de los resultados se utilizó la exactitud,

sensibilidad, especificidad, el error en la etapa de validación y el tiempo de ejecución.

La exactitud o eficiencia mide la tasa general de aciertos del sistema, la sensibilidad es la tasa de aceptación de las muestras positivas, es decir, la medida de muestras clasificadas como 1 que realmente son de la clase 1, la especificidad, por otro lado, mide la tasa de aceptación de las muestras negativas, es decir, la medida de muestras clasificadas como 0 que realmente son de la clase 0 y, por último, el error, que es el total de muestras bien clasificadas sobre el total de muestras clasificadas, es decir, 1 menos la eficiencia. Para entender mejor estas métricas se usa la matriz de confusión:

	Real		
Predicho		1	0
	1	TP	FP
	0	FN	TN

Figura 6. Matriz de confusión.

Entonces las ecuaciones serían:

$$\text{Exactitud} = \frac{TP + FN}{TP + TN + FP + FN}$$

$$\text{Sensibilidad} = \frac{TP}{TP + FN}$$

$$\text{Especificidad} = \frac{TN}{TN + FP}$$

$$\text{Error} = 1 - \text{Exactitud}$$

Una vez se tienen los datos organizados, pasamos a desarrollar los códigos para los diferentes modelos clasificadores:

Los modelos que se usaron para este problema fueron: Regresión logística, k-vecinos más cercanos, árboles de decisión, random forest, máquinas de soporte vectorial y redes neuronales de tipo perceptrón multicapa.

Cada uno de los algoritmos planteados fueron ejecutados en la misma máquina, con la misma base de datos y con el mismo programa, con el fin de tener unos resultados menos sesgados por factores externos a los métodos como el hardware y el software usados.

- *Regresión logística:*

Este modelo busca una función que sea capaz de dividir el espacio de características y dejar los conjuntos de muestras de cada clase separados.

Eta (tasa de aprendizaje) fue el parámetro que varió para las diferentes pruebas en el intervalo [1.0, 0.9, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1, 0.09, 0.08, 0.07, 0.06, 0.05, 0.04, 0.03, 0.02, 0.01]. Para observar los resultados de todas las pruebas, ver apéndice 1.

- *K-Vecinos más cercanos (KNN):*

Es un algoritmo simple que almacena todas las muestras disponibles y predice un valor objetivo con base en alguna medida de similaridad. La suposición fundamental de este tipo de métodos es que los cambios de la función (modelo) que describe el comportamiento

son suaves. Por consiguiente, se asume que el comportamiento de un punto puede ser estimado como el promedio del comportamiento de sus k-vecinos.

En este caso el parámetro a variar fue el número de vecinos, así: 1, 2, 3, 4, 5, 6, 7, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Para observar los resultados de todas las pruebas, ver apéndice 2.

- *Árbol de decisión:*

Es un método para la predicción de variables a partir de funciones discretas, en los cuales la función aprendida es representada a través de un árbol de decisión. Son en general métodos de aprendizaje inductivo muy empleados, pueden representarse en forma de conjuntos de reglas if-then.

Para este algoritmo, la máxima profundidad del árbol es la encargada de variar los resultados, por esto, se cambia el valor de este parámetro en cada ejecución, los valores dados fueron: [0 (None), 5, 10, 20, 30, 50, 60, 70, 80, 100, 200]. Para observar los resultados de todas las pruebas, ver apéndice 3.

- *Random Forest:*

consiste no sólo en crear un grupo de árboles (bosque), sino también en incluir un componente aleatorio en la partición que se realiza en cada nodo. En un Random Forest el conjunto de variables que se evalúan en cada nodo se escoge de manera aleatoria del conjunto de variables originales, es decir, antes de analizar cuál variable usar para hacer la partición, se escogen de manera aleatoria m variables y la escogencia de la partición se realiza únicamente usando dicho subconjunto.

En este caso hay dos parámetros importantes para encontrar la mejor solución, el número de árboles del bosque y la cantidad de variables analizadas por nodo, los valores dados a estos parámetros fueron para el número de árboles: [5, 10, 20, 50, 100, 200] y para la cantidad de variables por nodo: [50, 100, 150, 200, 250, 300]. Para observar los resultados de todas las pruebas, ver apéndice 4

- *Máquinas de soporte vectorial:*

Son un tipo de modelos de aprendizaje que permiten encontrar la mejor solución utilizando como criterio de ajuste la maximización del margen, entendiendo margen como la distancia más corta entre la frontera de decisión y cualquiera de las muestras.

El tipo de kernel, el valor C para darle peso al error y gamma que es el coeficiente del kernel, son los 3 parámetros importantes para este algoritmo, en este caso los valores asignados fueron: kernel tipo lineal o rbf, C con valores [0.001, 0.01, 0.1, 1, 10, 100, 200, 300] y gamma si es tipo lineal, es 0, si es rbf toma estos valores [0.001, 0.01, 0.1, 1, 0.001, 0.01, 0.1, 1]. Para observar los resultados de todas las pruebas, ver apéndice 5.

- *Red neuronal tipo perceptrón multicapa:*

Son por definición redes neuronales de propagación hacia adelante ya que la activación de las neuronas se hace desde la entrada (lugar donde se conectan las

variables) hacia las neuronas de salida las cuales entregan la predicción deseada.

Para este modelo, hay que variar el número de capas ocultas y el total de neuronas por capa, los valores asignados fueron: [1, 2] para las capas ocultas y [20,24,28,32,36] para el total de neuronas por capa. Para observar los resultados de todas las pruebas, ver apéndice 5.

3. Resultados y análisis

Los mejores resultados de cada modelo fueron recopilados en la tabla 1, en la cual se evidencia cada uno de los índices propuestos.

Usando únicamente la representación Bag of Words y usando el modelo de red neuronal tipo perceptrón multicapa con 2 capas ocultas, de 32 neuronas cada una.

Estos resultados, que se pueden considerar buenos comparados con lo reportado en la literatura para este tipo de dataset, se considera que se deben al ambiente y contexto del cual fueron extraídas estas opiniones, ya que un simple conteo de palabras puede diferenciar bien las opiniones negativas de las positivas cuando se trata de calificar los servicios hoteleros.

Para ver el código y la aplicación puede seguir el enlace al siguientes [repositorio](#).

4. Conclusiones

Los resultados del presente trabajo indican que cuando se trata de análisis de sentimientos, los modelos que tienen como base la Regresión Logística tiene mejores resultados, tales como las redes neuronales.

También hay que decir que se pudo observar que el lenguaje usado en las revisiones de los hoteles fue más estándar y permitió una clasificación con muy buenas medidas de error solo teniendo en cuenta la representación básica de Bag of Words.

Es importante dejar en claro que debido al reto representado por el proyecto y su tiempo estipulado de dedicación, quedó pendiente probar modelos de última generación como Redes Neuronales Recurrentes, LSTM y GRU para observar si se mejoran los resultados usando la misma representación probada con los modelos de

machine learning tradicionales, esto queda como propuesta para futuras investigaciones.

5. Referencias

Bing Liu. Sentiment Analysis. Mining Opinions, sentiments and Emotions. 2015

Bing Liu. *Sentiment Analysis and Opinion Mining*. 2012

Chen, Yubo and Jinhong Xie. *Online Consumer Review: Word-of-Mouth as a New Element of Marketing Communication Mix*. 2008

Hu, Minqing and Bing Liu. *Mining and summarizing customer reviews*. In *Proceedings of ACM SIGKDD International Conference on knowledge Discovery and Data Mining (KDD-2004)*. 2004.

Jaramillo, A. *Requirements Elicitation Approaches: A Systematic Review*. 2015

Jindal, Nitin and Bing Liu. *Mining comparative sentences and relations*. In *Proceedings of National Conf. on Artificial Intelligence (AAAI-2006)*. 2006b.

Liu, Bing. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. 2006 and 2011: Springer.

Nello Cristianini, John Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. 1999

surveys by Pang and Lee, Liu. *Sentiment Analysis and Opinion Mining*. 2012

Wiebe, Janyce, Rebecca F. Bruce, and Thomas P. O'Hara. *Development and use of a gold-standard data set for subjectivity classifications*. In *Proceedings of the Association for Computational Linguistics (ACL-1999)*. 1999.

Wiebe Janyce, Ellen Riloff. *Learning Extraction Patterns for Subjective Expressions*. 1999

Wilson, Theresa, Janyce Wiebe, and Rebecca Hwa. *Just how mad are you? Finding strong and weak opinion clauses*. In *Proceedings of National Conference on Artificial Intelligence (AAAI-2004)*. 2004.

Tabla 1. Mejores resultados para cada modelo.

	Índice							
Algoritmo	Exactitud	IntExactitud	Sensibilidad	IntSensibilidad	Especificidad	IntEspecificidad	Error de validación	Tiempo de ejecución
Regresión Logística	0.850566	0.0581382	NaN	NaN	NaN	NaN	0.149434	0.0880797
KNN	0.698992	0.0458225	0.635596	0.116344	0.765994	0.0831358	0.301008	1.18008
Árbol de Decisión	0.805891	0.0319229	0.787105	0.0516668	0.825625	0.0520855	0.194109	2.13894
Random Forest	0.875194	0.0288493	0.89055	0.0434278	0.862124	0.0608779	0.124806	11.7295
SVM	0.882481	0.02561	0.889462	0.0362354	0.87642	0.0480377	0.117519	3.55753
Red Neuronal	0.905581	0.0235225	0.927005	0.0316055	0.884653	0.0422383	0.0944186	279.846

Apéndices

Apéndice 1. Resultados generados por el algoritmo regresión logística

Tasa de aprendizaje	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
1.0	0.848605	0.0593352	NaN	NaN	NaN	NaN	0.151395	0.108099
0.9	0.850566	0.0581382	NaN	NaN	NaN	NaN	0.149434	0.0920832
0.7	0.844721	0.0574067	NaN	NaN	NaN	NaN	0.155279	0.0890806
0.6	0.844759	0.059818	NaN	NaN	NaN	NaN	0.155241	0.0860779
0.5	0.844759	0.059818	NaN	NaN	NaN	NaN	0.155241	0.091083
0.4	0.842798	0.0568743	NaN	NaN	NaN	NaN	0.157202	0.0960877
0.3	0.840875	0.060207	NaN	NaN	NaN	NaN	0.159125	0.087079
0.2	0.84276	0.0649014	NaN	NaN	NaN	NaN	0.15724	0.0780711
0.1	0.840799	0.0663257	NaN	NaN	NaN	NaN	0.159201	0.0720651
0.09	0.842722	0.0644758	NaN	NaN	NaN	NaN	0.157278	0.0690637
0.08	0.836878	0.063701	NaN	NaN	NaN	NaN	0.163122	0.0710642
0.07	0.834917	0.0637799	NaN	NaN	NaN	NaN	0.165083	0.0700638
0.06	0.831033	0.0676705	NaN	NaN	NaN	NaN	0.168967	0.0630574
0.05	0.829072	0.0652599	NaN	NaN	NaN	NaN	0.170928	0.0620558
0.04	0.823228	0.0667495	NaN	NaN	NaN	NaN	0.176772	0.0620575
0.03	0.819344	0.0635768	NaN	NaN	NaN	NaN	0.180656	0.0620563
0.02	0.803695	0.0675255	NaN	NaN	NaN	NaN	0.196305	0.0600543
0.01	0.772511	0.0730267	NaN	NaN	NaN	NaN	0.227489	0.0600541

Apéndice 2. Resultados generados por el algoritmo K-Vecinos más cercanos

Número de vecinos	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
1	0.679767	0.0436826	0.783296	0.0988136	0.580226	0.120041	0.320233	1.07897
2	0.697752	0.0380792	0.559769	0.125513	0.836971	0.0928052	0.302248	1.097
3	0.699302	0.0463083	0.768644	0.121609	0.6317	0.152797	0.300698	1.14804
4	0.699845	0.0418185	0.636781	0.0938485	0.766628	0.0844821	0.300155	1.21711
5	0.692171	0.0453511	0.788497	0.103814	0.597684	0.125278	0.307829	1.17006
6	0.686202	0.0484292	0.659669	0.13989	0.71608	0.117169	0.313798	1.17706
7	0.688295	0.0525681	0.815924	0.114142	0.56473	0.146021	0.311705	1.16906
10	0.696822	0.0459121	0.716135	0.144771	0.680783	0.136308	0.303178	1.18307
20	0.677054	0.068658	0.908955	0.0890012	0.452257	0.170904	0.322946	1.2001
30	0.668682	0.0725593	0.90773	0.0952577	0.436433	0.19071	0.331318	1.2131
40	0.636744	0.0674465	0.92471	0.0909997	0.35282	0.182982	0.363256	1.23313
50	0.600698	0.0751138	0.932444	0.0986818	0.277818	0.187448	0.399302	1.24213
60	0.568527	0.0602582	0.96339	0.0773044	0.181204	0.153894	0.431473	1.28416
70	0.547132	0.0565645	0.990336	0.0357067	0.100346	0.0961127	0.452868	1.3162
80	0.542093	0.0465666	0.998816	0.00449984	0.0699319	0.0540571	0.457907	1.3182
90	0.519612	0.0431673	0.999704	0.00207572	0.0411601	0.030536	0.480388	1.34422
100	0.519767	0.0421377	1	0	0.0415287	0.034084	0.480233	1.32521

Apéndice 3. Resultados generados por el algoritmo árbol de decisión

Maxima profundidad	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
0	0.801473	0.0359759	0.783527	0.0537222	0.820673	0.0555066	0.198527	1.8667
5	0.798527	0.0350212	0.739353	0.0628442	0.858683	0.0754876	0.201473	1.13102
10	0.799767	0.0348933	0.754525	0.0509587	0.84702	0.070236	0.200233	1.58645
20	0.802403	0.0280251	0.776758	0.0530684	0.82912	0.0535419	0.197597	1.82765
30	0.802093	0.0343097	0.786057	0.0582778	0.81924	0.062677	0.197907	1.88271
50	0.802713	0.030448	0.78039	0.0527565	0.826249	0.0481187	0.197287	1.89372
60	0.800543	0.0395099	0.776159	0.0504729	0.825706	0.0694893	0.199457	1.8747
70	0.806667	0.0321087	0.796299	0.0525613	0.819506	0.0613129	0.193333	1.89873
80	0.803721	0.0329595	0.787713	0.0607321	0.821595	0.054803	0.196279	1.89873
100	0.796589	0.0328726	0.777944	0.0553386	0.816913	0.0614805	0.203411	1.91674
200	0.802713	0.0363825	0.786662	0.0571554	0.82044	0.0606788	0.197287	1.90802

Apéndice 4. Resultados generados por el algoritmo random forest

N arboles	Variables x nodo	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
5	50	0.78186	0.0422466	0.837142	0.0649389	0.731011	0.0905796	0.21814	1.36224
	100	0.797132	0.0375478	0.826036	0.0573304	0.770025	0.0802323	0.202868	1.44024
	150	0.80186	0.0324346	0.821537	0.0618153	0.78601	0.0736598	0.19814	1.50669
	200	0.806899	0.0377559	0.825284	0.0715546	0.792427	0.0768466	0.193101	1.60603
	250	0.811938	0.0358877	0.814093	0.0616129	0.811556	0.0709526	0.188062	1.71556
	300	0.809612	0.0321368	0.827997	0.0651991	0.794081	0.0726004	0.190388	1.78662
10	50	0.820698	0.0388163	0.814478	0.0604297	0.830771	0.0779301	0.179302	2.5075
	100	0.834031	0.0329452	0.807452	0.0657875	0.86191	0.0623601	0.165969	2.60547
	150	0.835194	0.0376048	0.801636	0.068807	0.871209	0.0599687	0.164806	2.85492
	200	0.831628	0.0379937	0.799966	0.0627805	0.865885	0.0616352	0.168372	2.97971
	250	0.836589	0.0304145	0.804192	0.061182	0.873246	0.0601494	0.163411	3.11237
	300	0.842016	0.0327932	0.814137	0.0669705	0.873682	0.0650473	0.157984	3.25096
20	50	0.845271	0.0349941	0.865965	0.0506045	0.82927	0.0777168	0.154729	4.5571
	100	0.853023	0.0282051	0.843985	0.0490086	0.864161	0.0570133	0.146977	4.94884
	150	0.856744	0.0276795	0.845225	0.0525723	0.870721	0.0577237	0.143256	5.30882
	200	0.849535	0.0323748	0.832309	0.0500048	0.870263	0.0673333	0.150465	5.55744
	250	0.852713	0.0296607	0.840153	0.0514993	0.868471	0.059435	0.147287	5.87089
	300	0.852093	0.0291638	0.841256	0.052174	0.86671	0.0616726	0.147907	6.16471
50	50	0.870853	0.0317019	0.894775	0.0447238	0.848952	0.069191	0.129147	10.9801
	100	0.865194	0.0354204	0.870109	0.0552967	0.863908	0.0661019	0.134806	11.8607
	150	0.867907	0.0261027	0.871804	0.0430187	0.865742	0.0555864	0.132093	12.6432
	200	0.861938	0.0298363	0.865208	0.0414467	0.861022	0.0583198	0.138062	13.4538
	250	0.860775	0.0329785	0.862836	0.0433322	0.861084	0.0587778	0.139225	14.3867
	300	0.862713	0.0305252	0.858682	0.0551898	0.870511	0.0632037	0.137287	15.0743
100	50	0.875349	0.0302361	0.907638	0.0436105	0.845808	0.059347	0.124651	21.9849
	100	0.870465	0.0302682	0.892584	0.0432534	0.851197	0.0585225	0.129535	23.088

	150	0.86814	0.0314598	0.873417	0.0510123	0.866483	0.0631349	0.13186	24.7595
	200	0.860388	0.03047	0.871096	0.0521907	0.853881	0.0613928	0.139612	26.392
	250	0.863101	0.0320225	0.870309	0.0517237	0.859218	0.0631445	0.136899	27.7713
	300	0.860465	0.0286926	0.861856	0.0488818	0.861797	0.0600565	0.139535	29.2686
200	50	0.875116	0.0286275	0.906383	0.0398056	0.847272	0.0596952	0.124884	42.5687
	100	0.87031	0.031063	0.891885	0.0346532	0.850894	0.0611451	0.12969	45.9838
	150	0.870853	0.029523	0.878897	0.0487623	0.865876	0.0587337	0.129147	49.4299
	200	0.868372	0.0305977	0.878063	0.0417524	0.862324	0.0576529	0.131628	52.3376
	250	0.867364	0.0310445	0.873298	0.0476311	0.864026	0.0590861	0.132636	55.3543
	300	0.866977	0.0300575	0.871452	0.0423218	0.865316	0.0520271	0.133023	58.4471

Apéndice 5. Resultados generados por el algoritmo máquinas de soporte vectorial

Kernel	C	gamma	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
linear	0.001	0.000	0.613178	0.0907308	0.910885	0.203575	0.342276	0.285394	0.386822	4.57816
	0.010	0.000	0.801705	0.0343749	0.939009	0.0263683	0.667171	0.0563749	0.198295	3.50519
	0.100	0.000	0.876899	0.0284461	0.909747	0.0319434	0.843444	0.0471681	0.123101	3.10582
	1.000	0.000	0.879457	0.0282999	0.886978	0.0454565	0.872824	0.0461595	0.120543	3.22394
	10.000	0.000	0.881705	0.0252532	0.889757	0.0379881	0.874197	0.0446603	0.118295	3.21092
	100.000	0.000	0.88031	0.0247169	0.887845	0.0401755	0.873168	0.0445101	0.11969	3.19991
	200.000	0.000	0.88155	0.0252404	0.887546	0.0346238	0.87531	0.0431118	0.11845	3.20391
	300.000	0.000	0.881008	0.0268619	0.891851	0.0375008	0.870668	0.0409014	0.118992	3.1979
rbf	0.001	0.001	0.472093	0.0228781	0.61	0.48775	0.39	0.48775	0.527907	4.61219
		0.010	0.467984	0.0276399	0.65	0.47697	0.35	0.47697	0.532016	4.65224
		0.100	0.471628	0.023119	0.58	0.493559	0.42	0.493559	0.528372	4.66925
		1.000	0.472791	0.0214126	0.57	0.495076	0.43	0.495076	0.527209	4.66724
	0.010	0.001	0.466744	0.0249823	0.56	0.496387	0.44	0.496387	0.533256	4.60919
		0.010	0.47	0.0251587	0.59	0.491833	0.41	0.491833	0.53	4.67625
		0.100	0.473333	0.0255419	0.67	0.470213	0.33	0.470213	0.526667	4.73229
		1.000	0.473721	0.0219777	0.57	0.495076	0.43	0.495076	0.526279	4.73931
	0.100	0.001	0.465891	0.0247213	0.49	0.4999	0.51	0.4999	0.534109	4.63622
		0.010	0.656667	0.0417065	0.859899	0.0647846	0.463442	0.0989548	0.343333	4.67225
		0.100	0.473876	0.0261427	0.503106	0.493878	0.500812	0.4935	0.526124	4.77834
		1.000	0.46969	0.0247114	0.58	0.493559	0.42	0.493559	0.53031	4.85542
	1.000	0.001	0.690388	0.0524826	0.949483	0.0331235	0.43523	0.0966691	0.309612	4.34995
		0.010	0.825736	0.0300927	0.898328	0.0383883	0.75433	0.0531891	0.174264	3.87752
		0.100	0.709225	0.0439542	0.553373	0.0680222	0.869457	0.0547281	0.290775	4.74232
		1.000	0.479147	0.0278357	0.536346	0.492427	0.479654	0.488187	0.520853	4.75332
	10.000	0.001	0.831705	0.030029	0.915876	0.0356414	0.750012	0.0572708	0.168295	3.20391
		0.010	0.877674	0.0266869	0.870502	0.0389926	0.885427	0.0382899	0.122326	4.12175
		0.100	0.722171	0.0384284	0.544824	0.0576453	0.904342	0.0365981	0.277829	4.76934
		1.000	0.47876	0.0225239	0.519134	0.490706	0.494998	0.486942	0.52124	4.76934
	100.000	0.001	0.87938	0.026523	0.885712	0.0369079	0.873972	0.0411748	0.12062	3.37106
		0.010	0.872791	0.0257825	0.868957	0.0421782	0.877488	0.040342	0.127209	4.14377

		0.100	0.732636	0.0375548	0.560124	0.0605262	0.909243	0.0390813	0.267364	4.78035
		1.000	0.483643	0.0211415	0.697193	0.451825	0.320088	0.451077	0.516357	4.8464
	200.000	0.001	0.880465	0.0280826	0.887763	0.0444236	0.874388	0.0423836	0.119535	3.37707
		0.010	0.875581	0.0233429	0.863035	0.0457277	0.889486	0.0390554	0.124419	4.17179
		0.100	0.729147	0.0431027	0.552754	0.0665779	0.905257	0.0406396	0.270853	4.88444
		1.000	0.473798	0.0271491	0.673298	0.463073	0.341796	0.460004	0.526202	4.82839
	300.000	0.001	0.878915	0.0247766	0.886324	0.0359072	0.873138	0.0443412	0.121085	3.36406
		0.010	0.878682	0.0270403	0.86618	0.0414027	0.892048	0.0349393	0.121318	4.19081
		0.100	0.729767	0.0393753	0.556582	0.06118	0.907864	0.0425878	0.270233	4.80137
		1.000	0.480233	0.0256254	0.588351	0.482196	0.428308	0.479883	0.519767	4.89545

Apéndice 6. Resultados generados por el algoritmo red neuronal tipo perceptrón multicapa

N. de capas ocultas	Neuronas por capa	Accuracy	IntAccuracy	Sensibility	IntSensibility	Especificity	IntEspecificity	Error validación	Tiempo ejecución
1	20	0.893178	0.0255127	0.929231	0.0333343	0.858803	0.0544717	0.106822	325.637
	24	0.894651	0.0261528	0.9263	0.0316221	0.863591	0.0506733	0.105349	366.455
	28	0.89062	0.0236121	0.924056	0.0321194	0.857206	0.0415838	0.10938	400.448
	32	0.893256	0.0247056	0.927775	0.0328846	0.860083	0.0434086	0.106744	438.435
	36	0.893721	0.0212662	0.927769	0.0301056	0.86018	0.0400631	0.106279	470.956
2	20	0.90062	0.0249603	0.923908	0.0330248	0.878096	0.0424898	0.0993798	220.724
	24	0.903953	0.0273391	0.930186	0.0327481	0.878231	0.0524997	0.0960465	247.792
	28	0.89969	0.0222474	0.92538	0.0335615	0.875296	0.0452683	0.10031	263.301
	32	0.905581	0.0235225	0.927005	0.0316055	0.884653	0.0422383	0.0944186	279.846
	36	0.899302	0.0252741	0.913357	0.0377749	0.886256	0.0432169	0.100698	290.822