



Universidad de Castilla-La Mancha
Escuela Superior de Ingeniería Informática

Trabajo Fin de Grado
Grado en Ingeniería Informática
Tecnología Específica de
Computación

Aplicación de ciencia de datos a medicina personalizada: clasificación de mutaciones genéticas en tumores.

Yunior Machado Hernández

Febrero, 2021



Trabajo Fin de Grado
Grado en Ingeniería Informática
Tecnología Específica de
Computación

Aplicación de ciencia de datos a medicina
personalizada: clasificación de mutaciones
genéticas en tumores.

Autor: Yunior Machado Hernández

Directores: José Antonio Gámez Martín
Juan Carlos Alfaro Jiménez

Febrero, 2021

*Dedicado a mi familia y a todos
aquellos que hicieron esto posible.*

Declaración de Autoría

Yo, **Yunior Machado Hernández**, con **DNI 49904609-Z**, declaro que soy el único autor del trabajo fin de grado titulado **“Aplicación de ciencia de datos a medicina personalizada: clasificación de mutaciones genéticas en tumores”** y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 11 de Febrero de 2021

Fdo: Yunior Machado Hernández

Resumen

En este TFG abordamos un problema de clasificación de medicina personalizada donde estudiamos algunas de las técnicas de minería de datos más importantes. Dicho problema consiste en clasificar 9 tipos de mutaciones genéticas a partir del gen donde se encuentra dicha mutación, el aminoácido que ha sido transformado y la evidencia clínica donde se explican las conclusiones alcanzadas a partir de las pruebas clínicas. Esta última parte es la que más información aporta, y en la que se centrará principalmente nuestro estudio.

Veremos como con el procesamiento de texto (evidencia clínica) logramos clasificar con un mínimo de error los distintos tipos de tumores que nos presentan en el problema, demostrando así la importancia del mismo para este tipo de casos tanto para la clasificación como para la trazabilidad y soporte de nuestro modelo.

Agradecimientos

Agradezco a mis tutores, José Antonio Gámez Martín y Juan Carlos Alfaro Jiménez por ayudarme en estos tiempos difíciles, toda su atención y dedicación a la guía de este proyecto.

También agradezco a mi familia todo el apoyo y comprensión que me han dado durante todos estos años de carrera.

Otra parte que merece reconocimiento es toda la comunidad universitaria (amigos, compañeros, algunos profesores...), ya que propiciaron un ambiente adecuado para el desarrollo tanto profesional como personal.

Índice general

Capítulo 1	Introducción	1
1.1	Marco general de la medicina personalizada.	2
1.1.1	Tipos de Datos	3
1.1.2	Manejo de los Datos	5
1.1.3	Análisis e Interpretación de los Datos	6
1.1.4	Conclusiones	7
1.2	Motivación	9
1.3	Objetivos	9
1.4	Estructura del proyecto	10
1.5	Justificación de la adquisición de Competencias	10
Capítulo 2	Estado del Arte	13
2.1	Preprocesamiento	13
2.1.1	Codificación	14
2.1.2	Selección de variables	15
2.1.3	Tratamiento del desbalanceo	17
2.1.4	Preprocesamiento del Texto	20
2.2	Clasificadores	22
2.2.1	KNN	22
2.2.2	Naive Bayes	23
2.2.3	Árbol de Decisión	25
2.2.4	Random Forest	26
2.2.5	Support Vector Machines	26
2.3	Validación y Selección de Modelos	27
2.3.1	Métricas de Evaluación	28

2.3.2	Validación Cruzada	30
2.3.3	Búsqueda o Ajuste de Hiperparámetros	31
Capítulo 3	Exploración y Datos Disponibles	33
3.1	Datos Disponibles	33
3.2	Análisis Exploratorio de los Datos	34
3.2.1	Distribución de las Clases y Agrupamientos.	35
Capítulo 4	Experimentos y Resultados	41
4.1	Introducción	41
4.2	Metodología	41
4.2.1	Clasificación sin texto.	42
4.2.2	Clasificación solo con texto.	43
4.2.3	Clasificación con todas las variables.	45
4.3	Resultados	46
4.3.1	Clasificación sin texto	46
4.3.2	Clasificación con solo el texto	47
4.3.3	Clasificación con todas las variables	49
4.4	Conclusiones	50
Capítulo 5	Conclusiones y Trabajo Futuro	53
5.1	Conclusiones	53
5.2	Trabajo futuro	54
Bibliografía		55
Anexo I.	Título del anexo	57

Índice de figuras

Figura 1. Banco de Dato de Medicina Personalizada [1]	3
Figura 2. Grandes corporaciones de investigación biotecnológica [1]	6
Figura 3. Algoritmos de aprendizaje automático habitualmente considerados en Medicina Personalizada [1].....	8
Figura 4. <i>Knowledge Discovery</i>	14
Figura 5. Tipos de preprocesado [2]	14
Figura 6. Técnicas de reducción [2]	15
Figura 7. Binarizado	15
Figura 8. Conjunto de datos después de aplicar SMOTE	18
Figura 9. Proceso SMOTE [6].....	19
Figura 10. Tomek Link	19
Figura 11. <i>Bag of Words</i>	21
Figura 12. Ejemplo de ejecución de un árbol de regresión.....	26
Figura 13. Proceso de creación de un árbol de decisión dentro del <i>Random Forest</i>	27
Figura 14. SVM	27
Figura 15. SVM <i>One-to-Rest</i>	28
Figura 16. Área ROC	30
Figura 17. Validación Cruzada de $K=5$ iteraciones	31
Figura 18. Palabras Frecuentes	35
Figura 19. Tf-idf 1-gramas	36
Figura 20. Distribución de las Clases	37
Figura 21. Frecuencia Relativa de Genes agrupados por Clase.....	37
Figura 22. Longitud del Texto de cada Clase	38
Figura 23. Palabras Frecuentes de cada Clase	39
Figura 24. Esquema de validación y testeo de nuestros modelos.	42
Figura 25. Resultados Clasificación sin Texto.....	46
Figura 26. Resultados NB	47
Figura 27. Resultados Clasificación con Texto	48

Figura 28. Resultados SVC.....	49
Figura 29. Resultados Clasificación con toda la información.	50
Figura 30. Resultados SVC.....	50

Índice de tablas

Tabla 1. Configuración texto 1	47
Tabla 2. Configuración texto 2	49

Capítulo 1

Introducción

En las últimas décadas se ha hablado mucho sobre la medicina personalizada y, concretamente, como va a cambiar el tratamiento del cáncer el análisis genético.

Habitualmente, el diagnóstico de tumores cancerosos se había enfocado de manera sistémica:

1. Cada especialista busca los tumores que corresponden a su especialidad.
2. Los tumores son detectados mediante pruebas diagnósticas específicas como, por ejemplo, la mamografía o la elevación en los niveles PSA (antígeno prostático específico), o también de manera causal, mediante pruebas genéricas u otro tipo de exploración.
3. Interpretación de los resultados de las pruebas. Si no se llega a un diagnóstico claro, realizar más pruebas.

Usando aprendizaje automático y análisis de datos, no solo es posible agilizar el proceso y aliviar la carga de los expertos, sino también reducir el coste de la prueba diagnóstica, aumentar la periodicidad, y llevarlo a cabo de manera masiva. De esta manera, muchos casos de cáncer podrían ser diagnosticados a tiempo, logrando reducir la agresividad del tratamiento o la mortalidad debido a esta enfermedad. Además, una detección tardía podría implicar un gran sufrimiento y, desde un punto de vista económico, un coste sensiblemente elevado.

Una necesidad del proceso diagnóstico susceptible de ser eliminada mediante el aprendizaje automático es el contraste en las resonancias magnéticas. El gadolinio es el componente clave en los materiales de contraste usados más a menudo en los exámenes por resonancia magnética (RM), esto genera una mayor complejidad en la prueba y pueden dar lugar a depósitos en el organismo, susceptibles de provocar algunos efectos secundarios. Esto puede evitarse entrenarse con un algoritmo con imágenes obtenidas mediante contraste, con bajos niveles del mismo y sin él. Además, posibilita que las imágenes obtenidas puedan ser examinadas algorítmicamente mediante visión computarizada con una fiabilidad comparable, o incluso mayor en tareas concretas que la de un radiólogo humano, y que únicamente las imágenes que generen algún tipo de dudas puedan pasar a un examen detallado mediante radiólogos cualificados.

Con la superposición de aprendizaje automático y medicina, se puede llegar a conseguir pruebas diagnósticas menos intrusivas, más seguras, con un enfoque amplio y que podemos plantearnos llevar a cabo con una periodicidad mayor, posibilitando una detección más temprana de los problemas.

En este Trabajo de Fin de Grado (en adelante TFG) probaremos y analizaremos algunas de las técnicas más populares hoy en día de Minería de Datos y aprendizaje automático aplicadas a la medicina personalizada, incluyendo: exploración del conjunto de datos, técnicas de preprocesado, algoritmos de clasificación y validación de modelos. Para ello, contamos con un conjunto de datos extraído de una competición de *kaggle*, centrada en el diagnóstico de 9 tipos de tumores.

A continuación, describiremos el marco actual de la medicina personalizada, sus principales retos y objetivos.

1.1 Marco general de la medicina personalizada.

En las últimas décadas ha habido una gran transformación sin precedentes en la investigación biomédica, conduciéndola a un nuevo paradigma basado en datos debido a la mejora en la disponibilidad y recolección masiva de datos llevado a cabo por grandes

organizaciones como *Global Alliance for Genomics and Health* (GA4GH), infraestructuras de información como ELIXIR y *Big Data to Knowledge* (BD2K), etc. (Figura 1). El alto rendimiento en cuanto a la secuenciación del genoma y una considerable reducción en los costes de procesamiento ha conducido a algunas grandes organizaciones sanitarias a realizar la extracción de información de registros clínicos y datos de imágenes [1].

Large international consortia focusing on Personalized Medicine		
Initiative	Research focus	Link
Human Genome Diversity Project (HGDP)	General	www.hagsc.org/hgdp/
Global Network of Personal Genome Projects (PGP)	General	www.personalgenomes.org/
The Encyclopedia of DNA Elements (ENCODE)	General	www.encodeproject.org/
The NIH Roadmap Epigenomics Mapping Consortium (Roadmap)	General	www.roadmapepigenomics.org/
International Human Epigenome Consortium (IHEC)	General	http://ihc-epigenomes.org/
Cohorts for Heart and Aging Research in Genomic Epidemiology (CHARGE)	Cardiovascular and age-related diseases	www.chargeconsortium.com/
International Cancer Genome Consortium (ICGC)	Cancer	https://icgc.org/
The Cancer Genome Atlas (TCGA)	Cancer	https://cancergenome.nih.gov/
International Rare Disease Consortium (IRDIRC)	Rare diseases	www.irdirc.org/

Figura 1. Banco de Dato de Medicina Personalizada [1]

Un escenario típico en el análisis de datos biomédicos son las variables heterogéneas, multiespectrales, además de observaciones incompletas, imprecisas y no estructuradas (texto clínico). Por ello, un sistema de clasificación se evalúa en el campo de la salud teniendo en cuenta siete características: [1]

1. Reconocimiento de Patrones.
2. Análisis de datos no estructurados.
3. Soporte de Decisiones.
4. Predicción.
5. Trazabilidad.
6. Seguridad.
7. Eficiencia.

1.1.1 Tipos de Datos

La información que genera cada paciente aumenta cada año tanto en términos de volumen como en complejidad. Por ejemplo, la neuroimagen está actualmente produciendo más de diez petabytes cada año (nueve veces más que las últimas 3 décadas). Al mismo tiempo, se espera que los datos genómicos superen ampliamente en la próxima década otras áreas del big data como la astronomía [1]. Las imágenes son el

tipo de dato biomédico que mayor volumen posee ya que no solo se trata de imágenes de gigapíxeles de resolución, mostrando tejidos y organismos subcelulares, sino que también se incluyen metadatos y mediciones cuantitativas. En este aspecto, el desarrollo de plataformas integradoras para la escalabilidad y análisis de datos de imágenes junto con datos genéticos y anotaciones funcionales son de suma importancia [1].

La información sanitaria en formato digital incluye tanto datos estructurados (p. ej., códigos ICD (*International Statistical Classification of Diseases and Related Health Problems*)) como no estructurados (p. ej. descripciones de síntomas). Este recurso, cuya función inicial era la comunicación entre médicos, presenta una valiosa herramienta para la investigación y el desarrollo de modelos. Por otra parte, las tecnologías de cuantificación paralela masiva de datos genómicos, como la secuenciación del genoma completo (WGS) y la del exoma completo (WEX), también están desempeñando un papel clave en la aceleración del descubrimiento biomédico. Junto con la secuenciación del genoma, plataformas experimentales como la secuenciación de transcriptomas (RNA-seq y perfil ribosómico), perfilado de proteoma (espectrometría de masas) y el perfilado del interactoma (captura del cromosoma, ChIP-seq), contribuyen a que la información biomédica sea más accesible, rápida y económica [1].

Avances recientes en genómica, incluidos los de secuenciación genómica unicelular y del transcriptoma, identificación de ADN tumoral en circulación (ADNct) mediante biopsia líquida, y secuenciación de genomas bacterianos en muestras humanas (metagenómica), ya están teniendo un gran impacto en medicina, y están destinados a ser integrados en los estándares de la práctica médica [1].

Además de las imágenes, datos multiómicos y EHRs (*Electronic Health Record*), los datos de salud generados por el paciente (PGHD) a partir de dispositivos portátiles e implantables, se están convirtiendo en un tipo de big data cada vez más relevante en Medicina Personalizada. También los sensores para mediciones biométricas en tiempo real están promoviendo el desarrollo en nuevas áreas como el “stream computing”, que se ocupa del procesamiento y análisis de flujos de datos en tiempo real [1].

1.1.2 Manejo de los Datos

Son muchas las áreas donde se aplica el *Big Data*, entre ellas están el desarrollo de biomarcadores, investigación básica de cáncer, enfermedades raras, neurodegeneración, diabetes, patologías cardiovasculares, etc. El desarrollo de estas áreas en el marco de la Medicina Personalizada es relativamente alto en agencias sociales y gubernamentales, que requieren grandes esfuerzos de colaboración, experiencia colectiva y gestión distribuida (Figura 2). Muchos de estas grandes plataformas internacionales de investigación están comprometidas con el desarrollo de soluciones de medicina personalizada, tales como modelos de cerebro personalizados para pacientes con epilepsia intratable, investigación desarrollada dentro del Proyecto Cerebro Humano (www.humanbrainproject.eu), una iniciativa emblemática de la Comisión Europea. El modelo de investigación por consorcio, iniciado por movimientos comunitarios y cada vez más adoptado por agencias gubernamentales y el sector privado, es la base de todos los proyectos biomédicos a gran escala, como el caso del icónico Proyecto del Genoma Humano. La Comisión Europea ha invertido más de 2.600 millones de euros en la investigación de proyectos de medicina personalizada a través de fundaciones como FP7 y Horizonte 2020, y lanzó el Consorcio Internacional de Medicina Personalizada (PerMed), entre cuyos éxitos destaca el proyecto BLUEPRINT para el estudio de los mecanismos epigenéticos de la hematopoyesis [1].

Los datos biomédicos se adquieren de forma altamente distribuida, con formatos heterogéneos, y suelen ser de contenido sensible (datos privados). En este último aspecto, debido a la ley de protección de datos de la UE (2016/679 (GDPR)), la anonimidad y uso ético de los datos tendrá un impacto significativo en el diseño de actividades de investigación biomédica. En consecuencia, se utilizarán técnicas criptográficas basadas en *blockchain* para la anonimidad de los pacientes. Además, la computación en la nube se está convirtiendo en una forma generalizada de crear y ofrecer soluciones de software y almacenamiento (IBM Cloud, Google Cloud, Amazon Web Services, ...etc.), al exigir requisitos para limitar las amenazas a la seguridad. Para ser eficaces, los datos biomédicos deben ser seguros, pero también deben ser localizables, accesibles, interoperables y reutilizables (FAIR). De hecho, mantener la continuidad de las principales fuentes de datos, como en el caso de la Plataforma de Datos ELIXIR

(www.elixireurope.org/platforms/data), es crucial para evitar el aislamiento de los datos en los almacenes de datos [1].

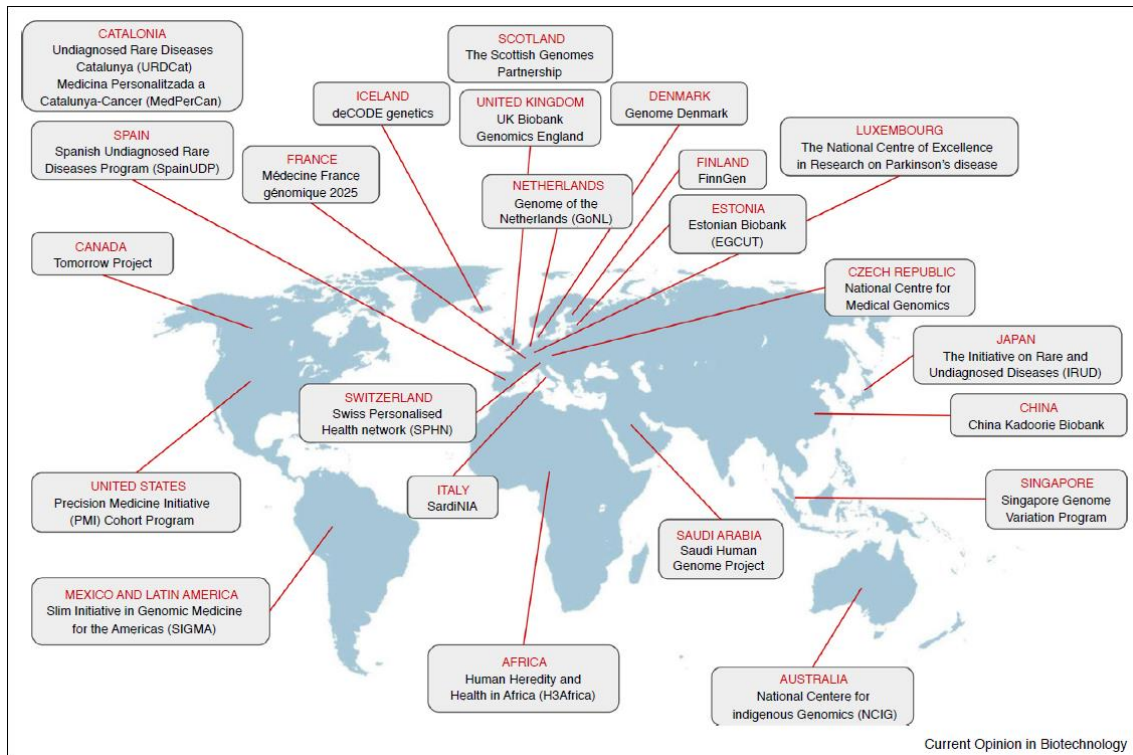


Figura 2. Grandes corporaciones de investigación biotecnológica [1]

Otro apartado fundamental es la computación de alto rendimiento (HPC). Los superordenadores y la programación paralela son esenciales para abordar problemas complejos con tareas de datos masivos. Las iniciativas europeas incluyen el gran ecosistema de datos Nube Abierta de Ciencia Europea (EOSC, <https://eosc-hub.eu/>), y EuroHPC para el desarrollo de supercomputadoras. Estas iniciativas tienen como objetivo proporcionar a la industria y a las autoridades públicas soluciones de clase HPC, además de un almacenamiento de datos de primera clase, gestión y transporte [1].

1.1.3 Análisis e Interpretación de los Datos

Aunque todo lo descrito anteriormente representa un gran trabajo y esfuerzo, de poco valdría sin la posibilidad de extraer información de estos grandes volúmenes de datos. Uno de los desafíos más difíciles para los sistemas informáticos consiste en el análisis de datos en *streaming* y las simulaciones basadas en datos (por ejemplo, el diseño de pacientes virtuales). Otro reto a tener en cuenta, es que los datos biomédicos comprenden un entramado de información complementaria proveniente de múltiples

fuentes heterogéneas, también denominados datos de vista-múltiple, que presentan distintas formas de representar instancias de datos [1].

Los métodos de aprendizaje automático se pueden aplicar eficazmente a ofrecer soluciones de integración para datos de múltiples vistas para explicar un evento o predecir un resultado. Por ejemplo, los modelos lineales generalizados (GLM), que son una generalización flexible de la regresión lineal ordinaria que permite variables de respuesta que tienen modelos de distribución de errores distintos de una distribución normal. Junto con los GLM, los modelos comunes de aprendizaje automático para datos multivisión son los modelos bayesianos como el clasificador Naive Bayes, los *ensembles* como el *Random Forest*, las redes neuronales y el *Deep learning* (Figura 3). La rápida popularidad del *Deep learning* se debe a las novedosas mejoras en el hardware, paquetes de software fáciles de usar, y la disponibilidad de grandes conjuntos de datos que se ajustan a vastos espacios de parámetros. El *Deep learning* ha sido ampliamente aplicado en la integración y modelización de datos biomédicos. En particular, ha funcionado eficazmente en la clasificación de imágenes y vídeos médicos, a menudo en combinación con el procesamiento de EHRs, además de incluirse en los sistemas de apoyo a las interacciones médico-computadora [1].

Por último, pero no menos importante, la computación cognitiva, apoyada en la neurociencia y en diversas técnicas de procesamiento del lenguaje natural (PLN) y lingüística computacional, persigue un proceso dinámico de observación, interpretación, evaluación y decisión. Sin embargo, tiene algunas limitaciones relacionadas con la comprensión correcta del significado contextual y la incertidumbre de la información. El ejemplo más popular de sistema cognitivo es el de IBM Watson, que se ha aplicado recientemente para el diagnóstico de varios tipos de cáncer y enfermedades neurológicas [1].

1.1.4 Conclusiones

El nuevo paradigma basado en datos está transformando significativamente el campo de la salud y de la investigación biomédica, pudiendo cambiar la manera de procesar la información clínica y molecular, las cuales abarcan, en cuanto a la escala, tasa, formas y

contenido de los datos generados, las cuatro dimensiones: volumen, velocidad, variedad y veracidad respectivamente. Actualmente disponemos de una gran cantidad de datos multiómicos, imágenes, dispositivos médicos y EHR provenientes de estudios de población a gran escala, que revelan sutiles, pero grandes diferencias en la genética humana y que permiten la aplicación de la medicina personalizada, involucrándose paralelamente la innovación y sostenibilidad de la gestión de la investigación y las infraestructuras. Entre los grandes desafíos del análisis de datos biológicos se encuentran el desarrollo de aplicaciones efectivas en áreas donde encontrar conexiones y conocimientos puede ser difícil debido a la abundancia y complejidad de los sistemas biológicos. Los métodos avanzados del *deep learning* y las plataformas para la computación cognitiva, son indudablemente las herramientas del futuro del análisis de macrodatos biomédicos. Deduciendo así que fomentar el progreso en estas áreas será crítico para la innovación en el cuidado de la salud y la medicina personalizada [1].

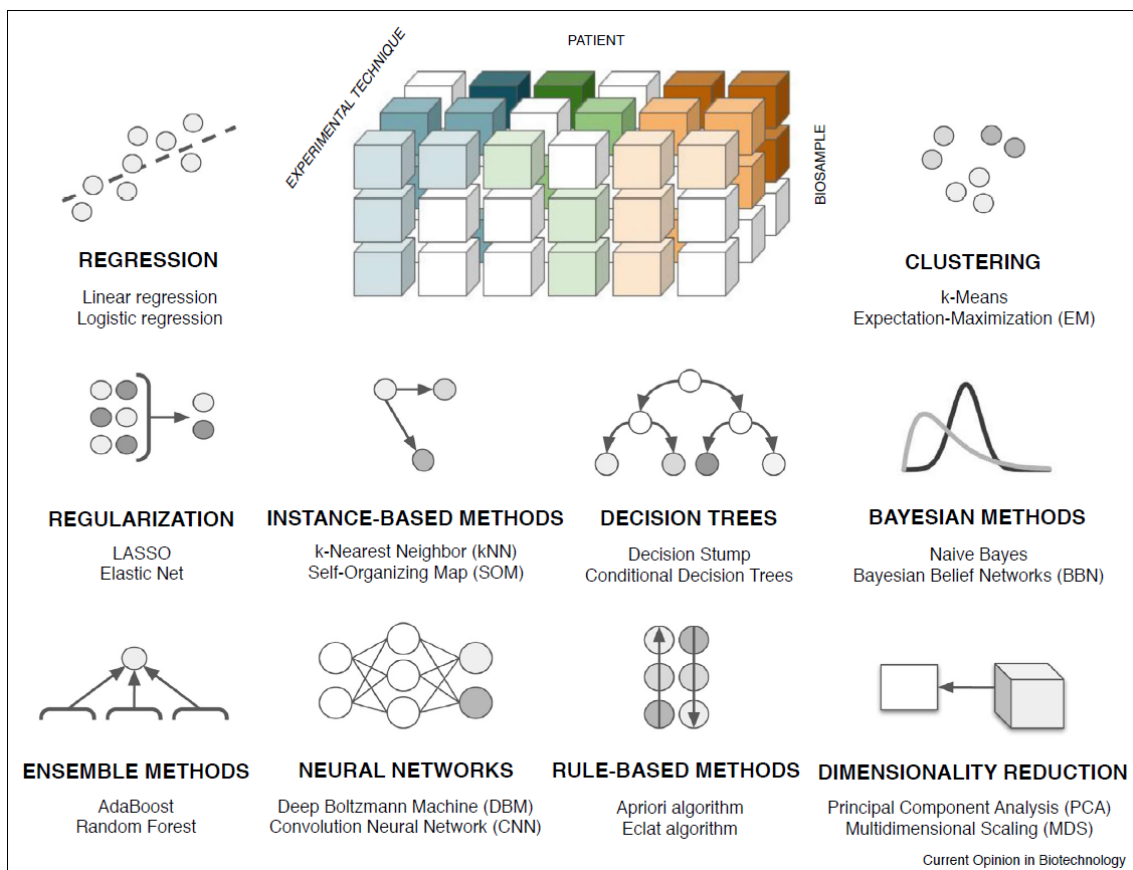


Figura 3. Algoritmos de aprendizaje automático habitualmente considerados en Medicina Personalizada [1]

1.2 Motivación

La enfermedad del cáncer siempre ha sido uno de los grandes retos de la medicina moderna, y, aunque ha habido numerosos avances en cuanto al diagnóstico y el tratamiento, siguen inabordables masivamente hablando en muchas ocasiones. Por tanto, mejorar la eficiencia disminuyendo la carga de los profesionales de la medicina para que puedan centrarse en aquellos casos que mayor dificultad plantean, sería de mucha ayuda.

El problema que abordamos pone de manifiesto esta necesidad, reclamando una solución para clasificar varios tipos de tumores disponiendo de anotaciones de expertos donde se han distinguido miles de mutaciones a partir de ellas. El cáncer, es una de las enfermedades más pronunciadas y temidas en las últimas décadas, sobre la que se ha llevado a cabo numerosos estudios acerca de su origen y tratamiento. Gracias a esto, ha habido muchos descubrimientos y buenos resultados en el área, aunque la mayoría pueden llegar a ser inabordables debido a temas económicos o de esfuerzo por parte de las organizaciones sanitarias. Por tanto, la automatización, y más concretamente, el aprendizaje automático llega a ser necesario no solo para una investigación más exhaustiva, sino también para que los tratamientos puedan ser llevados a cabos en zonas más empobrecidas o con menor presencia sanitaria.

1.3 Objetivos

Una vez secuenciado, un tumor puede tener miles de mutaciones genéticas, el reto está en distinguir las mutaciones que contribuyen a que el tumor crezca (*drivers*) de las neutrales (*passengers*). A menudo esa interpretación de mutaciones genéticas se hace manualmente, lo que consume mucho tiempo si un especialista tiene que revisar y clasificar cada mutación basándose en la evidencia del texto-base clínico.

Nuestro objetivo principal será obtener buenos modelos, que ayuden a realizar esta clasificación automáticamente, usando técnicas de aprendizaje automático a partir de estas anotaciones.

Esta no es una tarea trivial ya que la interpretación de la evidencia clínica es un gran desafío incluso para los especialistas humanos. Por lo tanto, modelar la evidencia clínica (texto) será fundamental para obtener un buen modelo.

De manera transversal, indagaremos en la naturaleza del problema (análisis exploratorio) y deduciremos cuales son las mejores técnicas y algoritmos de aprendizaje automático con los que obtendremos los mejores modelos. Además, se realizará un estudio de la influencia del texto y las demás variables por separado.

1.4 Estructura del proyecto

Para empezar, en la introducción pasaremos a explicar brevemente el problema que estamos abordando, donde se pondrá a prueba nuestros distintos algoritmos. Además de explicar el panorama actual de la medicina personalizada y sus posibilidades, seguido de los objetivos y problemas de la medicina moderna que pretendemos solventar. A continuación, le seguiría el estado del arte donde explicaremos las distintas técnicas de preprocesado, clasificación, y validación de modelos de la minería de texto. A partir de aquí, en materiales y métodos le sigue la descripción de la información que se nos proporciona, métricas que utilizaremos, los distintos clasificadores y la metodología a seguir. Y para acabar, presentaremos el estudio del experimento realizado y las conclusiones.

1.5 Justificación de la adquisición de Competencias

A continuación, pasamos a describir las competencias mencionadas en la propuesta del Trabajo de Fin de Grado, seguido de las actividades realizadas para profundizar en ellas y/o aplicarlas:

- Capacidad para conocer los fundamentos teóricos de los lenguajes de programación y las técnicas de procesamiento léxico, sintáctico y semántico asociadas, y saber aplicarlas para la creación, diseño y procesamiento de lenguajes.

Usada en el procesamiento de datos no estructurados (texto)

- Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Aplicada en la selección y uso de las distintas comparaciones de las técnicas y parámetros empleados en el preprocesado de los datos y el entrenamiento de los algoritmos.

- Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.

Aplicada en la selección de métodos, construcción y validación de los modelos de clasificación.

- Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes entornos inteligentes.

Aplicada en el procesamiento y transformación de los datos y la representación de la información no estructurada (texto).

- Capacidad para conocer y desarrollar técnicas de aprendizaje computacional y diseñar e implementar aplicaciones y sistemas que las utilicen, incluyendo las dedicadas a extracción automática de información y conocimiento a partir de grandes volúmenes de dato.

Aplicada en la selección de algoritmos de aprendizaje automático a aplicar así como en el diseño, ejecución y análisis de los experimentos con las distintas técnicas.

Capítulo 2

Estado del Arte

En este capítulo describiremos los métodos utilizados para resolver el problema, centrándonos en las técnicas de minería de texto, pues son núcleo de nuestro proyecto. Primero comenzaremos por las técnicas de preprocesamiento aplicadas al conjunto de datos para posteriormente abordar los algoritmos de clasificación utilizados, seguido de los métodos de validación y evaluación de modelos utilizando distintas métricas.

2.1 Preprocesamiento

Se denomina preprocesamiento al conjunto de técnicas utilizadas para representar un conjunto de datos de la manera más adecuada para el entrenamiento posterior de los modelos predictivos. Dentro del proceso de la extracción de conocimiento a partir de los datos (KDD), el preprocesamiento es una de las fases más importantes pues, debido a la imperfección de los datos reales, estos suelen contener inconsistencias y redundancias a ser resueltas (**Figura 4**). El preprocesamiento incluye disciplinas como preparación o reducción y transformación de datos (**Figura 5**). En la primera, se incluye la transformación, integración, limpieza y normalización de datos. Por el contrario, la segunda tiene como objetivo mejorar la eficiencia y eficacia de los algoritmos reduciendo la complejidad de los datos mediante la selección y/o construcción de variables, instancias o discretización (**Figura 6**).

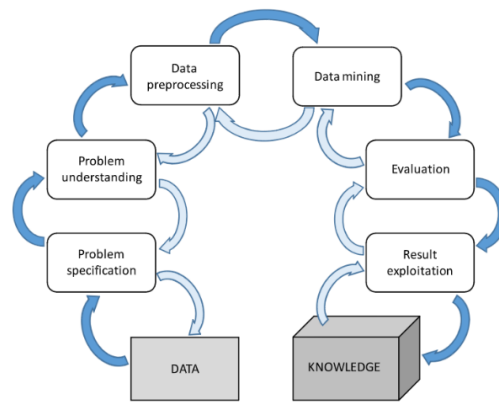


Figura 4. *Knowledge Discovery in Databases (KDD)*[2]

2.1.1 Codificación

Cada conjunto de datos tiene variables o columnas, que representan una parte de los datos susceptibles de ser medidos. Las variables categóricas son un tipo común de variables no numéricas que contienen valores discretos (normalmente de tipo *string*), que pueden tener asociados un orden (por ejemplo, variable altura: baja, media, alta).

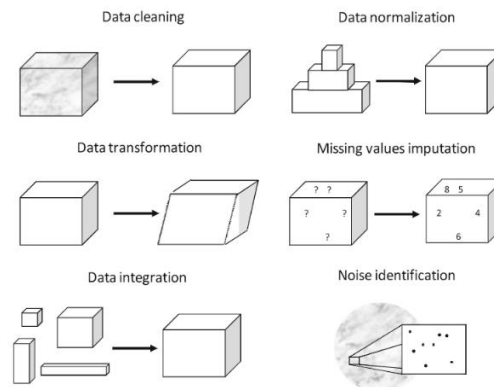


Figura 5. Tipos de preprocesado [2]

Algunos algoritmos de aprendizaje no son capaces de manejar este tipo de variables directamente, por lo que es necesario transformar el conjunto de datos a una representación basada en valores numéricos discretos. Las dos técnicas más habituales son [2]:

- **Numerizado:** Transforma cada valor posible de la variable categórica en un número entero.
- **Binarizado o *dummyficado*:** Transforma cada valor posible de la variable categórica en una variable binaria (Figura 7).

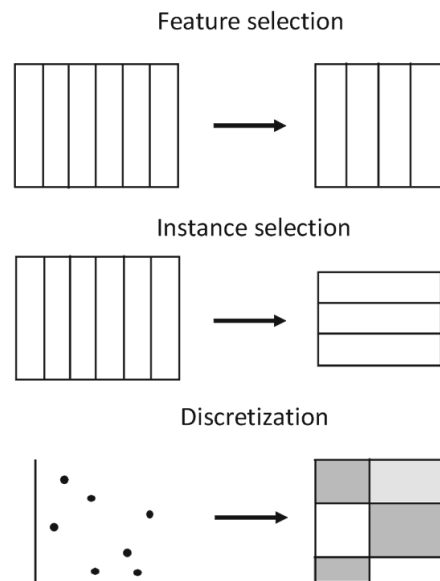


Figura 6. Técnicas de reducción [2]

2.1.2 Selección de variables

La selección de variables consiste en eliminar aquellas variables que aporten información pobre o redundante, para evitar correlaciones accidentales durante el aprendizaje de los modelos, aumentando así la capacidad de generalización. El objetivo sería obtener un subconjunto de variables que maximicen la información útil para un clasificador, de esta manera, reduciendo tanto el sobreajuste como el tiempo de ejecución y la memoria. A su vez, la visualización y trazabilidad de los modelos con un menor número de variables son más fáciles de interpretar [2].

Color			
Green			
Red			
Blue			

Color Green	Color Red	Color Blue
1	0	0
0	1	0
0	0	1

Figura 7. Binarizado

2.1.2.1 Eliminación recursiva de atributos.

La eliminación recursiva de atributos consiste en seleccionar subconjuntos de variables cada vez más pequeños (recursivamente), y al mismo tiempo que se maximiza la información contenida en el subconjunto de variables seleccionado. Primero, se entrena un clasificador con todo el conjunto de variables. Tras ello, se obtiene la importancia de cada atributo (por ejemplo, los coeficientes de un modelo lineal) y se eliminan aquellos con menor información. Este proceso se repite de manera recursiva hasta se alcanza un

número de específico de atributos (previamente definido), o bien converge en la métrica seleccionada [3].

Esto último tiene la ventaja de no tener que especificar el número de variables a elegir, sino que simplemente la eliminación se detiene cuando se halla una cota en la función de puntuación (*score*), lo cual garantiza que nos quedemos con las variables que maximicen dicha puntuación.

2.1.2.2 Selección de los K mejores. Prueba X^2 de Pearson.

Como su nombre indica, dicha técnica consiste en la selección de k variables que tengan la mejor puntuación basándonos en un estadístico (p.e información mutua). Las demás variables son eliminadas.

Prueba X^2 de Pearson

La prueba X^2 de Pearson se considera una prueba no paramétrica que mide la discrepancia entre una distribución observada y otra teórica (bondad de ajuste). Esta discrepancia se mide de la siguiente manera:

$$X^2 = \sum_i \frac{(\text{observada}_i - \text{teórica}_i)^2}{\text{teórica}_i} \quad (1)$$

Cuanto mayor sea el valor de X^2 , menos verosímil es que la hipótesis nula (que asume la igualdad entre ambas distribuciones) sea correcta. De la misma forma, cuanto más se aproxima a cero el valor de X^2 , más ajustadas están ambas distribuciones. También se utiliza para probar la independencia de dos variables entre sí, mediante la presentación de los datos en tablas de contingencia.

Esta puntuación se puede utilizar para seleccionar las n variables con los valores más altos para el estadístico chi-cuadrado (X^2), que debe contener solo variables no negativas, como valores booleanos o frecuencias (p.e: frecuencia de términos en la clasificación de documentos), en relación con las clases. Al medir la dependencia entre variables estocásticas, esta función "elimina" las variables que tienen más probabilidades de ser independientes de la clase y, por lo tanto, irrelevantes para la clasificación [4].

2.1.3 Tratamiento del desbalanceo

La mayoría de los conjuntos de datos representan las clases de manera desigual, es decir, la probabilidad con que aparecen los casos para cada valor de la clase son diferentes. Esto no supone un problema si esa desigualdad es pequeña. Sin embargo, cuando una o más clases son poco comunes, situación habitual en medicina (p.e enfermo vs sano), los modelos no suelen identificarlas de forma correcta. Por ello, se han propuesto muchas soluciones para afrontar este problema [2].

Estas soluciones suelen basarse en [6]:

- Muestreo de Datos: El conjunto de entrenamiento se modifica para lograr que haya una distribución de clases más equilibrada [7].
- Modificación del algoritmo: Los algoritmos de aprendizaje se adaptan para tratar este tipo de problemas explícitamente [6]. Normalmente usando clasificación basada en coste.

La mayoría de estudios en problemas desbalanceados han demostrado que la pérdida significativa de rendimiento se debe principalmente a la distribución sesgada de las clases, dada por la razón de desequilibrio (IR), esto es, la razón del número de instancias en la clase mayoritaria respecto al número de ejemplos en las clases minoritarias. Sin embargo, existen otros factores que contribuyen a tal degradación [6] :

- La falta de densidad de información en los datos de entrenamiento.
- Solapamiento entre clases.
- La importancia de las instancias fronterizas para realizar una buena discriminación entre las clases y su relación con ejemplos ruidosos.
- Las posibles diferencias en la distribución de los datos de entrenamiento y de prueba.

2.1.3.1 Preprocesamiento de conjunto de datos desbalanceados

Las técnicas de re-muestreo (*resampling*) pueden ser de varios tipos [6]:

- Submuestreo: Se crea un subconjunto del conjunto de datos original eliminando instancias (normalmente la clase mayoritaria).
- Sobremuestreo: Se crea un conjunto de datos mayor que el original replicando algunas instancias, esto es, creando nuevos casos a partir de los existentes.

- Híbridas: Combinación de ambas técnicas.

Dentro de estas familias de métodos, los más simples son los no heurísticos, como el submuestreo y sobremuestreo aleatorio. En el submuestreo aleatorio, el mayor inconveniente es que se pueden descartar datos potencialmente útiles para el proceso de aprendizaje, mientras que, en el caso del sobremuestreo, se puede aumentar la probabilidad de un sobreajuste, ya que se realizan copias exactas de las existentes [6].

Para hacer frente a estos problemas, se han propuestos métodos más sofisticados. Entre ellos se encuentra *Synthetic Minority Over-sampling Technique* (SMOTE). La idea principal, es crear nuevos ejemplos de clases minoritarias interpolando las instancias de la clase minoritaria que se encuentren juntas para sobremuestrear el conjunto de entrenamiento (Figura 8). De cada ejemplo x_i de la clase minoritaria, se hallan los k vecinos más cercanos pertenecientes a la misma clase, de estos k vecinos, se eligen al azar n instancias ($x_{i1}, x_{i2}, \dots, x_{in}$) (dicho valor de n depende de cuanto queramos sobremuestrear la clase), a través de las cuales se generarán casos sintéticos r_i por interpolación aleatoria, a lo largo de segmentos que las unen con x_i (Figura 9). Sin embargo, en las técnicas de sobremuestreo (especialmente para el algoritmo SMOTE) el problema de la sobregeneralización se debe en gran medida a la forma en que se crean las muestras sintéticas. Precisamente, SMOTE genera la misma cantidad de muestras de datos sintéticos para cada ejemplo minoritario original y lo hace sin tener en cuenta los ejemplos vecinos, lo que aumenta la ocurrencia de superposición entre clases [6].

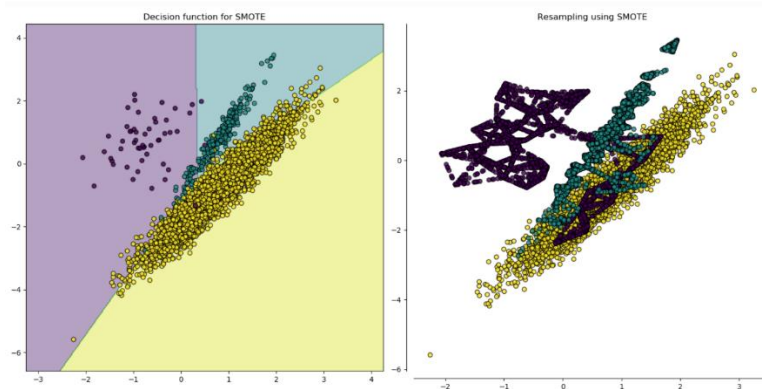


Figura 8. Conjunto de datos después de aplicar SMOTE

Con respecto al submuestreo, la mayoría de los enfoques existentes se basan en técnicas de limpieza de datos. Algunos de los trabajos más importantes en esta área incluyen la técnica de edición en los vecinos más cercanos propuesta por Wilson (ENN), la selección unilateral (OSS), el método de condensación aplicado usando la regla del vecino más

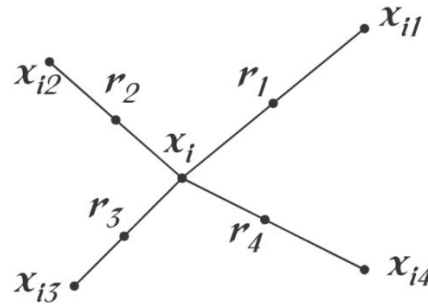


Figura 9. Proceso SMOTE [6]

cercano. Por ejemplo, *Tomek Links* (Figura 10), donde se muestran dos instancias de distinta clase, donde cada una es el vecino de la otra, y se pasa a eliminar aquella instancia que tenga la clase mayoritaria, o ambas según la estrategia que se desee tomar, y la regla de limpieza del vecindario, basada en ENN. En definitiva, la combinación del aumento de instancias con técnicas de limpieza de datos podría disminuir el solapamiento que se introduce por los métodos de muestreo, es decir, las integraciones de SMOTE con ENN o SMOTE con *Tomek Links*.

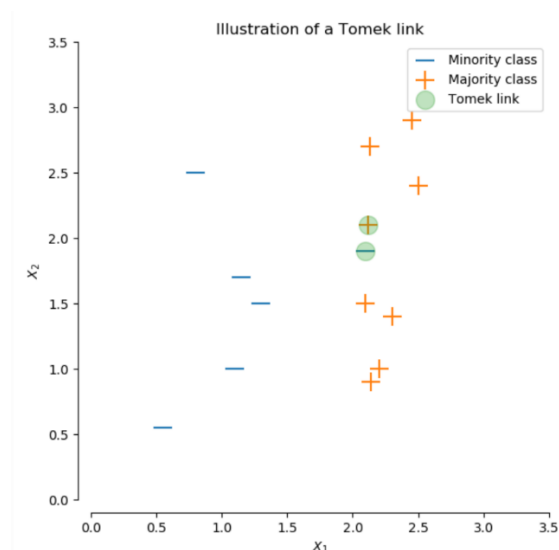


Figura 10. Tomek Link

2.1.4 Preprocesamiento del Texto

En este apartado, presentamos métodos para generar un conjunto de datos estructurado a partir de uno no estructurado (texto). Dichos métodos consisten en la eliminación de ruido (palabras que no aportan información, como los pronombres, artículos, etc.) y la construcción de atributos.

2.1.4.1 Tokenización

La *tokenización* es un método de preprocesamiento que divide un fragmento de texto en palabras, frases, símbolos u otros elementos significativos llamados *tokens*. El objetivo principal de este paso es la extracción de las palabras en una oración [8]. Por ejemplo, del fragmento de texto “Ayer comí lentejas” se transforma en “Ayer”, “comí”, “lentejas”.

2.1.4.2 Stop Words

Los textos y documentos incluyen una serie de palabras que no contienen información relevante para los algoritmos de clasificación (e.g., "a", "arriba", "a través", "después", "otra vez", etc.). La técnica más habitual para tratar con estas palabras es eliminarlas del conjunto de texto durante el procesado del mismo [8].

2.1.4.3 Lematizado

La lematización es un proceso habitual en el Procesamiento del Lenguaje Natural (PLN), que reemplaza el sufijo de una palabra por otra diferente o elimina el sufijo de una palabra por completo para obtener la forma básica de la palabra (lema) (p.e: géneros -> género)[8].

2.1.4.4 N-Gramas

Un n-grama es una subsecuencia contigua de n *tokens* que aparecen en un texto, pudiendo n tomar cualquier valor entre uno y la longitud del texto $- 1$. Los casos más utilizados son $n=1$, $n=2$ y $n=3$ los cuales llamamos unigramas (p.e lista de *tokens*), bigramas y trigramas respectivamente [8].


2.1.4.5 Bag of Words

El modelo de *bag of words* (BoW) es una representación reducida y simplificada de un documento. La técnica BoW se utiliza en varios dominios, como visión por computadora,

procesamiento del lenguaje natural (PLN), detección de spam mediante clasificadores bayesianos, filtros, etc. En un BoW, un cuerpo de texto, un documento o una oración se considera como una bolsa de palabras (*bag of words*) en una matriz, donde la relación semántica entre las palabras se ignora, obteniendo así un vocabulario. Si bien se ignoran la gramática y el orden de aparición, la multiplicidad (frecuencia de aparición) se cuenta y se puede usar más tarde para determinar los puntos de enfoque de los documentos (Figura 11) [8].

D1: He is a lazy boy. She is also lazy.

D2: Neeraj is a lazy person.



	He	She	lazy	boy	Neeraj	person
D1	1	1	2	1	0	0
D2	0	0	1	0	1	1

Figura 11. *Bag of Words*

2.1.4.6 TF-IDF

La frecuencia inversa de documentos (IDF) es un método utilizado junto con la frecuencia de los términos (TF) para destacar aquellas palabras de mayor frecuencia, pero que no sean implícitamente comunes en el *corpus* (texto sin estructura) y destacar aquellas, asignando para ello un mayor peso a las palabras con términos de alta o baja frecuencia en el documento. Esta combinación de TF e IDF se conoce como *Term Frequency-Inverse document frequency* (TF-IDF). El peso de un término en un documento por TF-IDF viene dado por [8]:

$$W(d, t) = TF(d, t) * \log\left(\frac{N}{df(t)}\right) \quad (1)$$

siendo N el número de documentos y $df(t)$ es el número de documentos que contienen el término t en el corpus. Aunque TF-IDF intenta superar el problema de los términos comunes en el documento, todavía sufre de algunas otras limitaciones descriptivas. Por ejemplo, TF-IDF no puede tener en cuenta la similitud entre las palabras del documento, ya que cada palabra es independiente y está indexada [8].

2.2 Clasificadores

A continuación, pasamos a describir todos los algoritmos que posteriormente se utilizarán en el estudio del problema.

2.2.1 KNN

El método de los k vecinos más cercanos (en inglés, *k-nearest neighbors*, abreviado *knn*) es un método de clasificación supervisada no paramétrico que sirve para estimar la función de densidad $F(x/C_j)$ de las variables predictoras x por cada clase C_j , o bien la probabilidad a posteriori de que un elemento x pertenezca a la clase C_j a partir de la información proporcionada por el conjunto de entrenamiento [9].

Es un tipo de aprendizaje vago (*lazy learning*), donde la función se aproxima solo localmente y todo el cómputo es diferido a la clasificación. La normalización de datos puede mejorar considerablemente la exactitud del algoritmo [9].

2.2.1.1 Algoritmo

Los ejemplos de entrenamiento son vectores en un espacio característico multidimensional, cada ejemplo está descrito en términos de p atributos considerando q clases para la clasificación. Los valores de los atributos del i -ésimo ejemplo (donde $1 \leq i \leq n$) se representan por el vector p -dimensional [9]:

$$\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{pi}) \in X \quad (2)$$

El espacio es particionado en regiones por localizaciones y etiquetas de los ejemplos de entrenamiento. Un punto en el espacio es asignado a la clase C si esta es la clase más frecuente entre los k ejemplos de entrenamiento más cercanos. Generalmente se usa la distancia euclídea [9]:

$$d(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{r=1}^p (x_{ri} - x_{rj})^2} \quad (3)$$

La fase de entrenamiento del algoritmo consiste en almacenar los vectores característicos y las etiquetas de las clases de los ejemplos de entrenamiento. En la fase de clasificación, la evaluación del ejemplo (del que no se conoce su clase) es representada por un vector en el espacio característico. Se calcula la distancia entre los vectores almacenados y el nuevo vector, y se seleccionan los k ejemplos más cercanos. El nuevo ejemplo es clasificado con la clase que más se repite en los vectores seleccionados [9].

2.2.2 Naive Bayes

Los métodos ingenuos de Bayes son un conjunto de algoritmos de aprendizaje supervisados basados en la aplicación del teorema de Bayes con la suposición "ingenua" de independencia condicional entre cada par de atributos dado el valor de la variable de clase. El teorema de Bayes establece la siguiente relación, dada la variable de clase y y un vector de variables dependientes (x_1, x_2, \dots, x_n) [10]:

$$P(y|x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots, x_n|y)}{P(x_1, \dots, x_n)} \quad (4)$$

Utilizando la suposición de independencia condicional [10]:

$$P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y) \quad (5)$$

para todo i . Entonces el teorema de Bayes puede simplificarse a:

$$P(y|x_1, \dots, x_n) = \frac{P(y) \prod_{i=1}^n P(x_i|y)}{P(x_1, \dots, x_n)} \quad (6)$$

Dado que $P(x_1, \dots, x_n)$ es constante dado la entrada o instancia a clasificar podemos usar la siguiente regla de clasificación:

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y) \Rightarrow \hat{y} = \arg \max_y P(y) \prod_{i=1}^n P(x_i|y) \quad (7)$$

y se usará la máxima verosimilitud para estimar $P(y)$ y $P(x_i|y)$; la primera es entonces la frecuencia relativa de la clase y en el conjunto de entrenamiento y la segunda es la frecuencia relativa de la instancia x_i conocida la clase [10].

A pesar de sus simples supuestos, los clasificadores de Bayes han funcionado bastante bien en muchas situaciones del mundo real, como la clasificación de documentos y el filtrado de spam. Requieren una pequeña cantidad de datos de entrenamiento para estimar los parámetros necesarios [10].

2.2.2.1 Naive Bayes Multinomial

Este clasificador implementa el algoritmo Naive Bayes para datos distribuidos multinomialmente, y es una de las dos variantes clásicas de Naive Bayes que se utilizan en la clasificación de texto (donde los datos son representados vectorialmente). La distribución está parametrizada por vectores $\theta_y = (\theta_{y1}, \dots, \theta_{yn})$ para cada clase y , donde n es el número de variables (en la clasificación de texto, equivale al tamaño del vocabulario especificado) y θ_{yi} es la probabilidad $P(x_i|y)$ de la variable i que aparece en la muestra de la clase y [10].

El parámetro θ_y es estimado por la versión suavizada de máxima verosimilitud:

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n} \quad (8)$$

Donde $N_{yi} = \sum_{x \in T} x_i$ es el numero de veces que la variable i aparece en la muestra de la clase y [10].

El suavizado a priori $\alpha \geq 0$ da cuenta de las características que no están presentes en las muestras de aprendizaje y evita probabilidades cero en cálculos posteriores. El ajuste $\alpha = 1$ se denomina suavizado de Laplace, mientras que $\alpha < 1$ se denomina suavizado de *Lidstone* [10].

2.2.2.2 Naive Bayes Complementario

Naive Bayes Complementario (de las siglas en inglés, CNB) es una adaptación del algoritmo estándar multinomial de naive Bayes (MNB) que es especialmente adecuado para conjuntos de datos no balanceados, donde existe al menos una clase minoritariamente representada. Específicamente, CNB usa estadísticas del complemento de cada clase para calcular los pesos del modelo. Los inventores de CNB muestran empíricamente que las estimaciones de parámetros para CNB son más estables que las de MNB. Además, CNB supera regularmente a MNB (a menudo por un margen considerable) en tareas de clasificación de texto. El procedimiento para calcular los pesos es el siguiente [11]:

$$\hat{\theta}_{ci} = \frac{\alpha_i + \sum_{j:y_j \neq c} d_{ij}}{\alpha + \sum_{j:y_j \neq c} \sum_k d_{kj}}; \mathbf{w}_{ci} = \log \hat{\theta}_{ci}; \mathbf{w}_{ci} = \frac{w_{ci}}{\sum_j |w_{cj}|} \quad (9)$$

donde las sumas son sobre todos los documentos j que no están en la clase c , d_{ij} es el valor de *tf-idf* del término i en el documento j , α_i es un hiperparámetro de suavizado como el que se encuentra en MNB. La regla de clasificación es [11]:

$$\hat{c} = \arg \min_c \sum_i t_i w_{ci} \quad (10)$$

2.2.3 Árbol de Decisión

Los árboles de decisión (de las siglas en inglés, DT) son un método de aprendizaje supervisado no paramétrico que se utiliza para clasificación y regresión. El objetivo es crear un modelo que prediga el valor de una variable objetivo aprendiendo reglas de decisión simples inferidas de los atributos de los datos [12].

Un árbol puede ser "aprendido" mediante el fraccionamiento del conjunto inicial en subconjuntos basados en una prueba de valor del atributo. Este proceso se repite en cada subconjunto derivado de una manera recursiva (particionamiento recursivo). La recursividad termina cuando el subconjunto en un nodo tiene todo el mismo valor de la variable objetivo, o cuando la partición ya no agrega valor a las predicciones. Este proceso de inducción *top-down* de los árboles de decisión (ITDAD) es un ejemplo de un algoritmo voraz, y es, con mucho, la estrategia más común para aprender árboles de decisión a partir de datos [12].

Por ejemplo, en la **Figura 12**, los árboles de decisión aprenden de los datos para aproximar una curva sinusoidal con un conjunto de reglas de decisión si-entonces-si no. Cuanto más profundo es el árbol, más complejas son las reglas de decisión y más ajustado es el modelo [12].

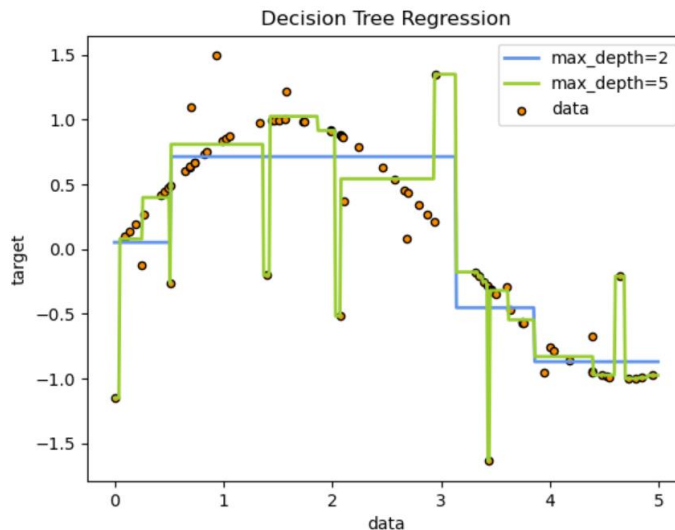


Figura 12. Ejemplo de ejecución de un árbol de regresión.

2.2.4 Random Forest

Random Forest es un ensemble (conjunto) de árboles de decisión que se construyen a partir de una muestra extraída con reemplazo de las instancias del conjunto de entrenamiento. Al expandir cada nodo durante la construcción de uno de los árboles, la mejor división se elige de un subconjunto aleatorio de tamaño n . Logrando reducir así la correlación entre los distintos arboles del ensemble.

Con esto logramos reducir la varianza implícita en los árboles de decisión a costa de un ligero aumento del sesgo. Para la clasificación, se promedia la predicción probabilística de todos los árboles o se hace una votación donde cada árbol vota por la clase más probable según su entrenamiento y se elige la clase más votada (Figura 13) [13].

2.2.5 Support Vector Machines

Support Vector Machines (SVM) es un algoritmo de aprendizaje supervisado que puede usarse tanto para clasificación (*support vector classification* (SVC)) como para la regresión (*support vector regression* SVR). El objetivo de un SVM es encontrar un hiperplano que maximice la separación de las instancias en dos clases. Aquellas instancias cuya distancia al hiperplano (margen) sea mínima, se les llamará *support vectors points* (Figura 14).

Cuanto mayor sea la distancia del hiperplano, mayor es la probabilidad de clasificar correctamente una instancia. La computación del producto escalar de las instancias

(separación) depende de una función *kernel* (p.e: lineal, polinomial, gaussiano, radial y sigmoidal) [14].

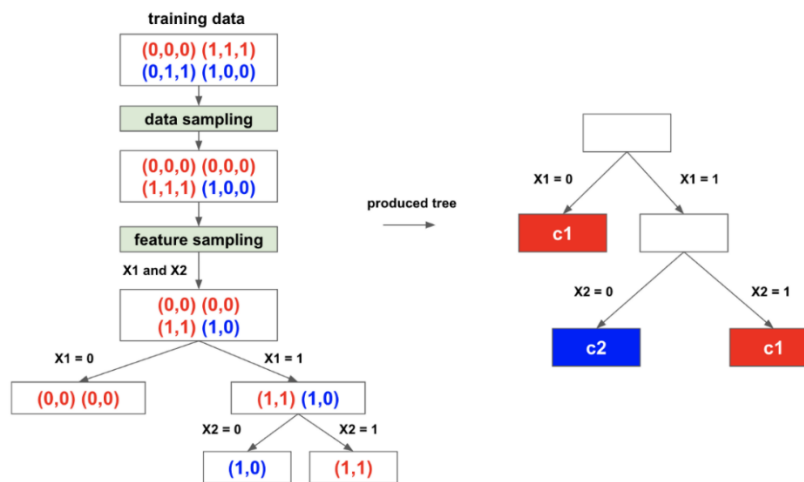


Figura 13. Proceso de creación de un árbol de decisión dentro del *Random Forest*

2.2.5.1 SVM Multiclase

SVM no soporta la clasificación multiclase originalmente, ya que separa los datos en dos partes. Por tanto, convertimos un problema de clasificación de múltiples clases, en otros de múltiples clasificaciones binarias. Uno de los métodos es conocido como *One-to-Rest*, donde tendremos un hiperplano que separa cada clase de las demás (Figura 15) [14].

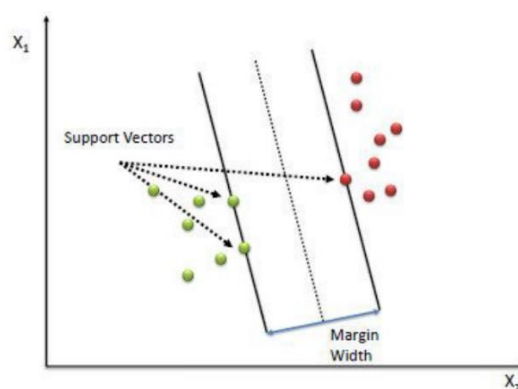


Figura 14. SVM

2.3 Validación y Selección de Modelos

Para poder comprobar como de malo/bueno es nuestro modelo, necesitamos un conjunto de datos que no haya sido “visto” durante el entrenamiento (conjunto test), pues si no, el clasificador simplemente repetiría las clasificaciones aprendidas (sobreajuste) y no sería capaz de predecir de manera correcta un conjunto de datos

distinto al de entrenamiento (error debido a la variabilidad de los datos). Para que este proceso tenga validez, el conjunto test solo se puede usar una vez para probar nuestro modelo. Esto plantea un problema, pues antes de usar el test es difícil saber si nuestro modelo clasificará de manera correcta datos que no haya “visto” nunca (generalizar).

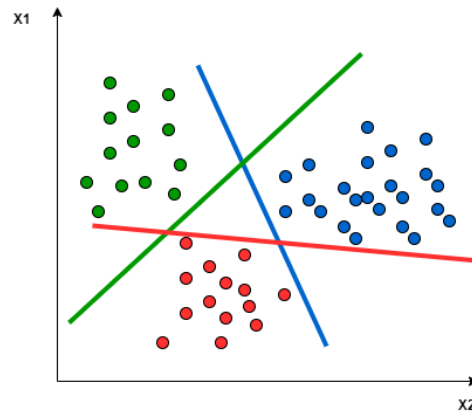


Figura 15. SVM *One-to-Rest*

Otro aspecto a tener en cuenta es la métrica que utilizaremos para evaluarlo, pues, según el objetivo a conseguir (tener mayor acierto, mayor precisión, menor entropía...etc.) se deben utilizar unas u otras para dirigir la selección de los mejores modelos. A continuación, veremos que métricas utilizaremos, así como las distintas técnicas de búsqueda de hiperparámetros y su validación.

2.3.1 Métricas de Evaluación

Para evaluar nuestro modelo hemos utilizado como métrica principal, tanto para evaluar como para estimar los mejores parámetros de nuestro clasificador, la función de pérdida conocida como *log-loss*, ya que es la métrica propuesta en la competición de *Kaggle* cuyo conjunto de datos estamos utilizando. Las demás las usamos como información complementaria.

2.3.1.1 Función de Pérdida *Log-loss*

La función de pérdida *log-loss* mide el rendimiento de un modelo de clasificación donde la entrada de predicción es un valor de probabilidad entre 0 y 1 para cada valor de la clase. El objetivo de nuestros modelos de aprendizaje automático es minimizar este valor. Un modelo perfecto tendría una pérdida de 0. La función de pérdida logarítmica aumenta a medida que la probabilidad prevista difiere de la etiqueta real. Por lo tanto, predecir

una probabilidad de 0.012 cuando la etiqueta de observación real es 1 sería mala y daría como resultado una pérdida de registro alta. Tenemos la fórmula:

$$-\sum_{c=1}^M y_{o,c} \log p_{o,c} \quad (11)$$

Donde $y_{o,c}$ es el valor real de la clase c en el ejemplo o , M es el número de clases y $p_{o,c}$ es la probabilidad predicha de la clase c en la observación o . El signo negativo en el sumatorio es debido a que los logaritmos de números menores que 1 son negativos, lo cual es más difícil de interpretar intuitivamente. Es decir, es más fácil de interpretar que debemos disminuir el error positivo, a maximizar el error negativo.

2.3.1.2 Tasa de Acierto (*accuracy*)

La fórmula de la tasa de acierto es muy sencilla:

$$\text{tasa de acierto} = \frac{\text{número de predicciones correctas}}{\text{número total de predicciones}} \quad (12)$$

Esto plantea ciertas limitaciones, ya que, en un problema altamente desbalanceado, tendríamos una tasa de acierto alta si escogemos siempre la clase mayoritaria, la cual puede no ser de nuestro objetivo clasificar. Tanto esta métrica como las siguientes son funciones de pérdida 0-1 ya que se basan en fallar/acertar, no en la probabilidad de predecir la clase real como en *log-loss*.

2.3.1.3 F1-score

Es la media armónica entre la precisión y el *recall*:

$$F1 = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (13)$$

Donde la precisión es el número de instancias predichas correctamente como la clase c de un total de predicciones de la clase c y el *recall* el número de instancias correctamente predichas como la clase c de un total de instancias de clase c .

2.3.1.4 Área bajo ROC (*Receiver Operating Characteristic Curve*)

La curva de la función ROC viene representada por el *recall* (TPR) y la tasa de falsos positivos (FPR o falsas alarmas) (Figura 16). La línea discontinua nos muestra la puntuación de un mal clasificador o una clasificación aleatoria.

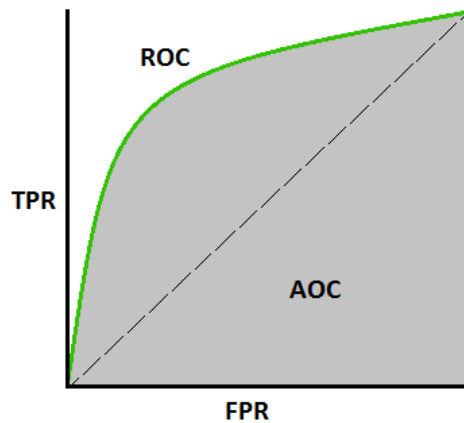


Figura 16. Área ROC

2.3.2 Validación Cruzada

Una posible solución para el problema propuesto en la introducción sobre la imposibilidad de observar, o la utilización de una única vez el conjunto test, es la extracción de una parte del conjunto de entrenamiento para obtener predicciones del modelo entrenado (conjunto de validación). De esta manera, podemos analizar que problemas tiene nuestro modelo e ir solventándolos. Esto plantea el inconveniente de como elegir las instancias del conjunto de validación de manera que:

- Sean una representación fiable del conjunto de datos.
- No sea excesivamente grande, pues esto limita las instancias con las que entrenar.
- La selección sea lo más aleatoria posible, teniendo en cuenta los anteriores puntos.

2.3.2.1 Validación Cruzada Estratificada de K iteraciones

En la validación cruzada estratificada de k iteraciones, el conjunto de entrenamiento se divide en k subconjuntos seleccionados aleatoriamente sin reemplazo, de manera que la proporción de clases se mantenga. Uno de los subconjuntos se utiliza como validación y el resto como entrenamiento, este proceso se repite durante k iteraciones donde se escoge en cada una un conjunto de validación distinto hasta haber probado con todos los

subconjuntos. Finalmente, se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado.

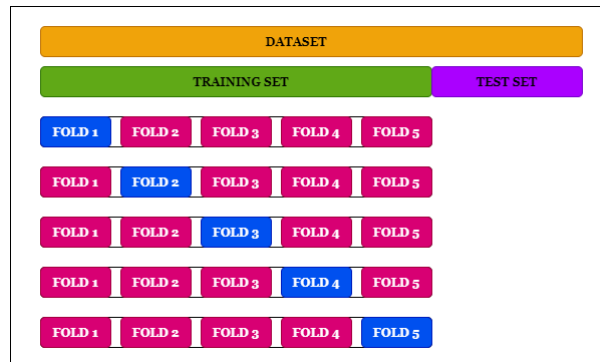


Figura 17. Validación Cruzada de $K=5$ iteraciones

2.3.3 Búsqueda o Ajuste de Hiperparámetros

Cada clasificador tiene parámetros que pueden influir drásticamente en su rendimiento y en el tipo de error que se comete (p.e el número de vecinos en KNN). Por eso, elegir los valores de los parámetros no es trivial. Esta búsqueda de parámetros puede llevarse a cabo por fuerza bruta (búsqueda exhaustiva), a menudo inviable, pues algunos parámetros toman valores en todo \mathbb{R} . O bien, de manera aleatoria, eligiendo un número de candidatos c de entre todas las combinaciones posibles.

Capítulo 3

Exploración y Datos Disponibles

3.1 Datos Disponibles

Nos son proporcionados dos tipos distintos de datos muy diferenciados, por un lado, tenemos la descripción de las mutaciones genéticas, la cual está constituida por variables discretas; y por otro lado tenemos la evidencia clínica, que es un texto donde se describe el por qué se llega a esa clasificación.

Descripción de los archivos:

- *training_variants*: archivo csv (*comma separated file*) con la descripción de la mutación genética, cuyos campos son:
 - *ID*: identificación de la fila utilizada para vincular la mutación con la evidencia clínica.
 - *Gene*: el gen donde se encuentra la mutación.
 - *Variation*: el aminoácido que cambia para esta mutación.
 - *Class*: tipo de tumor encontrado.
- *training_text*: un archivo delimitado por '||' que contiene la evidencia clínica (texto) utilizada para clasificar las mutaciones genéticas.

Tiene los campos:

- *ID*: identificación de la fila utilizada para vincular la evidencia clínica a la mutación genética.
- *Text*: la evidencia clínica utilizada para clasificar la mutación genética. Se trata de texto no estructurado, por tanto, es necesario preprocesar este atributo para que pueda ser utilizado por los algoritmos de aprendizaje.

3.2 Análisis Exploratorio de los Datos

Hay un total de 3321 casos disponibles, por tanto, la tarea de exploración no es algo trivial, ya que necesitamos hacer uso de herramientas de visualización y estadísticos para poder comprender mejor la base de datos que se nos proporciona.

Variables Discretas

Haciendo uso de un análisis superficial de las variables discretas tenemos que:

- La variable **Gene** tiene 264 valores posibles.
- La variable **Variation** tiene 2996 valores posibles. Obsérvese que son casi tantos como casos, por lo que a priori esta variable no sería de mucho interés.
- Hay 9 clases.
- No hay valores nulos.

Texto

Hemos aplicado los procesos visto en el capítulo 2 sección 1.4, donde primeramente ejecutamos el tokenizado, obteniendo 281586 tokens, le sigue el lematizado, luego le sigue BoW, donde como salida tendremos un conjunto de palabras con sus respectivas frecuencias, realizaremos una nube de palabras, donde a mayor frecuencia de dicha palabra, mayor es su tamaño en la “nube” (Figura 18).

Como se puede deducir de la nube de palabras, las palabras más frecuentes son “mutat” (*mutation* lematizada), “cell” (lo cual tiene sentido ya que estamos hablando de tumores, que son mutaciones de células). También sobresalen “background” (antecedentes del paciente), “oncogen”, “monomer”, “gene”.

Frecuencia Relativa de cada Gen en cada Clase

A continuación, vemos un diagrama de barras donde en cada barra se muestra la frecuencia relativa de los 20 genes más frecuentes, usando un color para cada gen: (Figura 21).

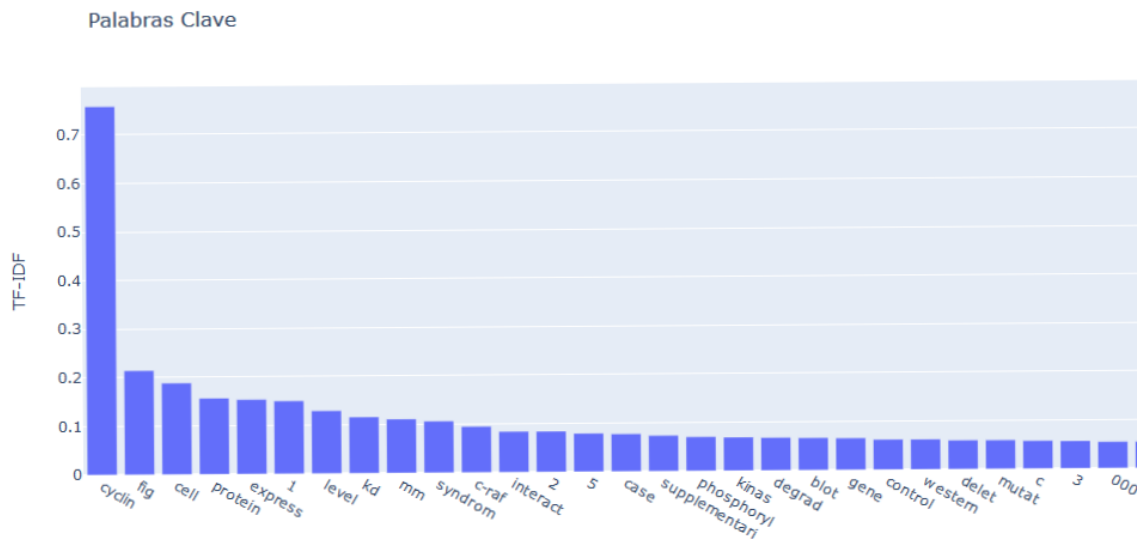


Figura 19. Tf-idf 1-gramas

Vemos que:

- La clase 5 tiene el gen **BRCA1** muy frecuente con diferencia (casi el 40% de los casos). Le sigue el **BRCA2** con una frecuencia de casi el 10%
- En la clase 6 el **BRCA2** aparece el 30% de los casos y el **BRCA1** el 20% aproximadamente. Lo que significa que entre los dos ocupan la mitad de los casos.
- En la clase 9 tenemos el gen **SF3B1** con una frecuencia del 40% de los casos.
- En las clases 8 y 9, hay poca variabilidad de genes (puede que sea porque están infrarrepresentadas)

Distribucion de cada Clase

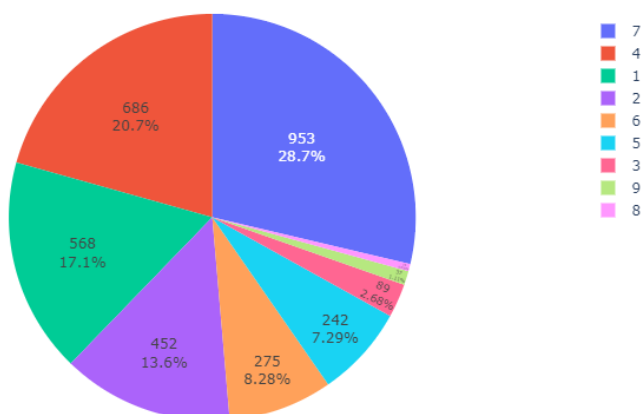


Figura 20. Distribución de las Clases

Longitud del Texto de cada Clase

A continuación, vemos un diagrama de violines, el cual nos muestra la distribución de la longitud de texto de cada clase usando una curva de densidad. El ancho de cada curva se corresponde con la frecuencia aproximada de los valores de longitud del texto (Figura 22). Nótese que los valores extremos se “extienden” para que la gráfica no quede acotada solo para los valores dados, es por eso que se pueden ver valores por debajo de 0.



Figura 21. Frecuencia Relativa de Genes agrupados por Clase

De este diagrama podemos deducir que la mayoría de los textos están acotados superiormente por 20.000 caracteres, lo cual hace bastante difícil la interpretación del

contexto a base de lectura “manual”. Para ello, haremos uso de una nube de palabras agrupadas por clase, de esta manera podemos ver las palabras más frecuentes usadas en cada clase y poder deducir a partir de ahí que tipo de tumor estamos clasificando (Figura 23).

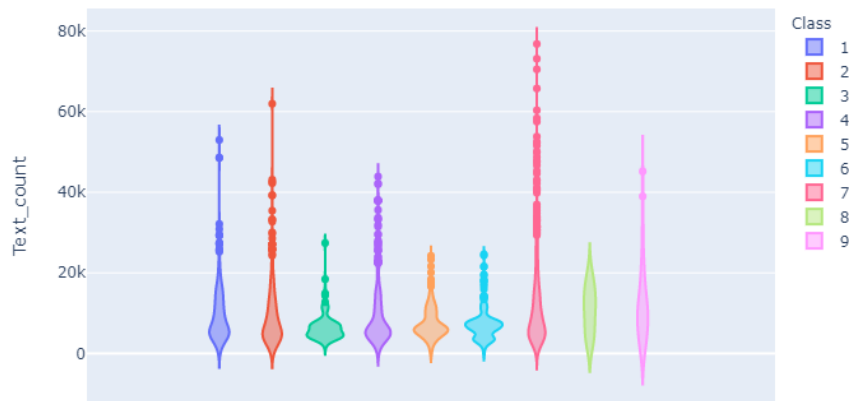


Figura 22. Longitud del Texto de cada Clase

De aquí podemos deducir que:

- La Clase 1 es un tipo de tumor relacionado con el **sexo** (masculino o femenino), **células estromales** (presentes en la medula espinal).
- La Clase 2 está relacionada con el **pulmón, quinasas y ciclinas**. Probablemente se deba a quinasas dependientes de ciclina (**cdk**).
- La Clase 3 está relacionada con la proteína **tumprss2** y **glioma** (tipo de tumor presente en el cerebro y la medula espinal).
- La clase 4 está relacionada con **monómeros** (molécula de poca masa), **oncogenes** (gen anormal) y **leucemia mielomonocítica** (células de ascendencia común reproduciéndose descontroladamente).
- La clase 5 está relacionada también con **monómeros** y **oncogenes**.
- La clase 6 está relacionada con **pulmón** y **cáncer**.
- La clase 7 está relacionada con la **vía pi3k/akt/mtor** (vía de señalización intracelular) y **familia** (probablemente porque sea hereditario).
- La clase 8 está relacionada con el **adn**.

- La clase 9 está relacionada con **arn**, sistema nervioso somático, gen **sf3b1**, inhibidor **ezh2** y grupo de proteínas **prc2**.

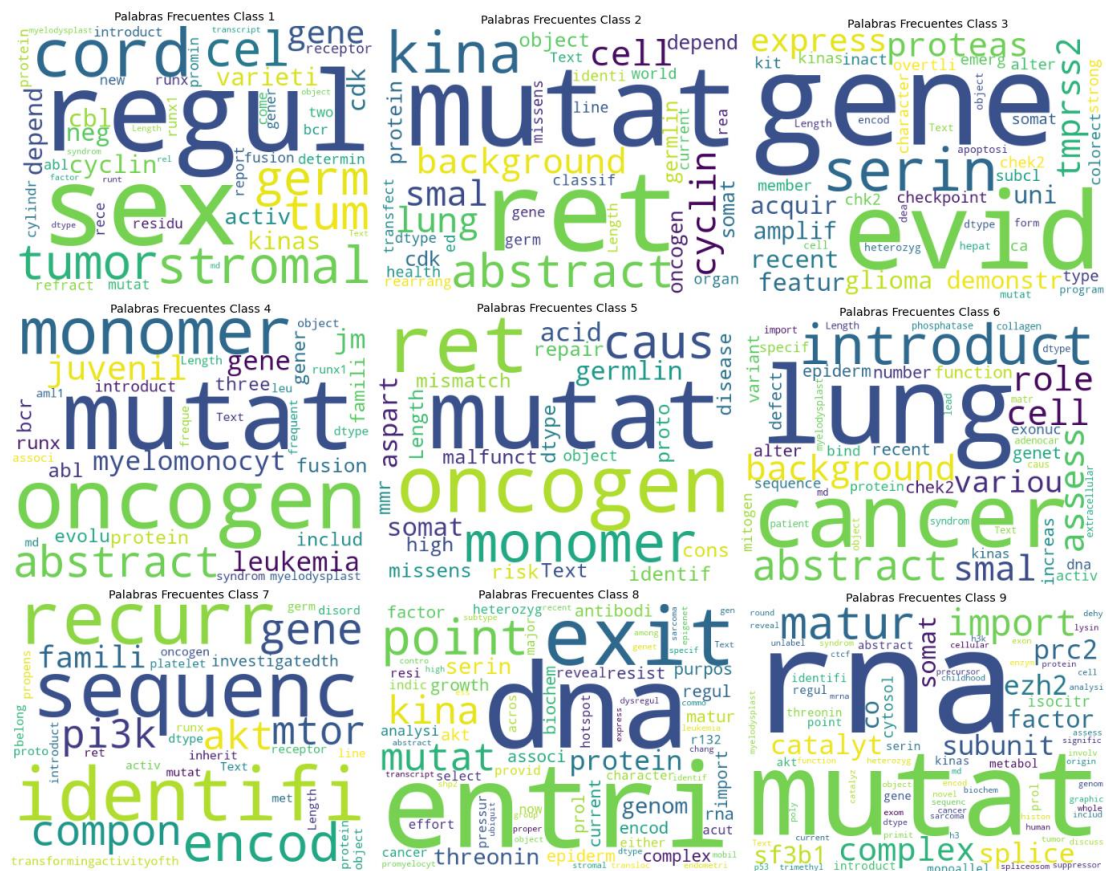


Figura 23. Palabras Frecuentes de cada Clase

A continuación, se llevará a cabo un análisis predictivo que confirmará o no estas tendencias detectadas en el análisis exploratorio.

Capítulo 4

Experimentos y Resultados

4.1 Introducción

Una vez realizada la exploración, pasamos al diseño de los experimentos, preprocesado, ejecución de los clasificadores y conclusiones de los resultados obtenidos.

4.2 Metodología

Para una mejor organización y comparación del impacto del texto, hemos separado en tres partes los experimentos. Por una parte, tenemos la clasificación sin texto, por otra la clasificación solo con texto, y por último la clasificación con todas las variables. En cada una de las partes, probaremos los distintos algoritmos descritos en el capítulo 2 y evaluaremos sus resultados con cada una de las técnicas de preprocesado propuestas.

Para la validación/testeo de nuestros modelos, seguimos el esquema de la **Figura 24**. Donde lo primero es separar los datos disponibles en *training/test* (2). Los datos de entrenamiento (training) los usaremos por un lado para el ajuste de hiperparámetros (1), usando la validación cruzada como validación (3) y como métrica para “dirigir” el ajuste, *log-loss*.

Por otro, para el entrenamiento de nuestro clasificador (5) con la configuración obtenida del proceso anterior (4) y así obtener nuestro modelo.

Finalmente, usaremos el test al final del procedimiento como prueba definitiva de nuestro modelo (6), siendo este último resultado, el que usaremos en nuestras gráficas de comparación de modelos.

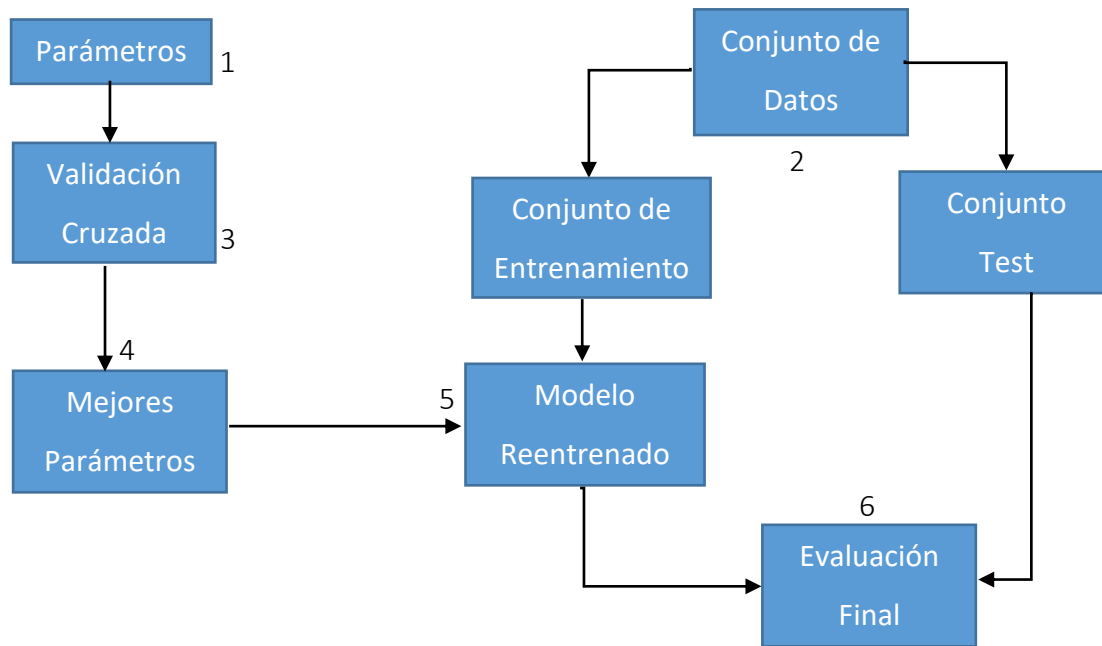


Figura 24. Esquema de validación y testeo de nuestros modelos.

4.2.1 Clasificación sin texto.

En esta parte solo tenemos las variables *Variants* y *Gene*. Como vimos en el apartado de exploración, la variable *Variants* no aporta casi información, pues casi hay un valor para cada instancia, además de presentar un grave peligro de sobreajuste. Por tanto, solo tenemos disponible para clasificar la variable *Gene*. Esto supone una desventaja, ya que disponemos de poca información para clasificar, pero también da lugar a poder probar numerosas técnicas que suponen un alto coste computacional (p.e la búsqueda exhaustiva) y la posibilidad de comparación de la combinación de algunas de ellas.

Preprocesado

Lo primero que debemos hacer es “preparar” la variable *Gene* para que ésta pueda ser utilizada por nuestros clasificadores, pues, al ser una variable categórica, necesitamos binarizarla antes. Una vez binarizada, la variable *Gene* se convierte en 264 variables binarias, con lo cual, probaremos a utilizar la eliminación recursiva de variables, usando un árbol de decisión como selector (ya que es un clasificador sencillo pero eficiente para obtener la importancia de las variables) y la 5-validación cruzada estratificada como validación.

Además, probaremos a utilizar el aumento de instancias con SMOTE para mejorar la clasificación de las clases más infrarepresentadas (sobre todo la 8 y la 9) y reducir la influencia de aquellas que estén sobrerrepresentadas (La 7 y la 4). Además del aumento de instancias, probaremos también un “balanceo implícito” con la modificación de los pesos de la siguiente manera:

$$\frac{\text{número de ejemplos}}{\text{número de clases} * \text{número de ocurrencias de la clase y}} \quad (15)$$

Clasificadores y Parámetros a ajustar:

- KNN: probamos el número de vecinos en el rango $[1, \sqrt{\text{numero de instancias}} \approx 60)$
- Naive Bayes (NB): probamos con y sin suavizado de Laplace.
- Árbol de Decisión: criterios de división según entropía (ganancia de información), o la impureza de Gini, que es una medida de cuán a menudo un elemento elegido aleatoriamente del conjunto sería clasificado incorrectamente si fue clasificado de manera aleatoria de acuerdo con la distribución de las clases.

4.2.2 Clasificación solo con texto.

En esta parte, el reto consiste en el tratamiento del texto (datos no estructurados). Como veremos, el problema de esta parte será la gran cantidad de datos (mayor probabilidad de ruido) de la que dispondremos y el elevado coste computacional.

Preprocesado

En este apartado utilizamos los procesos descritos en el capítulo 2, sección 1.4 (preprocesado del texto).

Primero debemos convertir el texto en una estructura de datos con la que se pueda clasificar, para ello usamos BoW. Luego, eliminamos aquellas palabras del lenguaje que no nos aporten información (*stop words*) y algunas que encontramos en la exploración (p.e “fig”, “conclusion”, “table”, etc.). Seguidamente, cambiamos la frecuencia de aparición de cada palabra almacenada en el BoW por el *tf-idf*, la cual es una puntuación más afinada.

También, probaremos la técnica de selección de variables de los $k=100$ mejores utilizando el estadístico chi-cuadrado como selector. Finalmente, probaremos también el aumento de instancias como en el apartado anterior, salvo en el caso del NB Complementario, ya que este compensa de manera implícita las clases infrarepresentadas.

Clasificadores y Parámetros a ajustar

- KNN: evaluamos el número de vecinos de forma similar a la clasificación sin texto.
- NB Multinomial
- NB Complementario
- *Random Forest* (RF): evaluamos los criterios de división según entropía (ganancia de información), o la impureza de Gini. Tenemos cien árboles para clasificar, tomando $\sqrt{\text{numero de variables}}$ para encontrar la mejor ramificación.
- SVC: evaluamos la función con la que dividiremos las instancias para clasificarlas (polinómica, rbf, sigmoidal y lineal).

Ajuste de Hiperparámetros General

Hay que tener en cuenta que, en este apartado, el coste computacional es elevado, con lo cual, no podemos hacer una búsqueda exhaustiva. Como alternativa, usaremos la búsqueda aleatoria ya que el número de candidatos se mantiene constante, independiente del número de parámetros que vayamos a evaluar, con lo cual el tiempo

de ejecución no varía. Además, para reducir aún más el número de entrenamientos para el ajuste, haremos una 3-validación cruzada, en lugar de la 5-cv usada en el apartado anterior.

Algunos parámetros que evaluaremos son propios de la clasificación con texto e independientes de los clasificadores, estos son:

- Número de palabras para clasificar con mejor *tf-idf*. Donde probaremos desde 1000 hasta 9000 yendo de mil en mil y finalmente con todas las palabras.
- N-gramas. Donde probaremos todas las posibilidades: palabras individuales (1,1), parejas de palabras (2,2), palabras individuales y parejas (1,2).
- Usar *tf-idf* o trabajar directamente con la frecuencia.
- Normalización del *tf-idf*. Si normalizamos con la norma de Manhattan (norma-l1), donde $||\vec{u}|| = (|u_1| + |u_2| + \dots + |u_n|)$, o con la norma Euclídea (norma-l2), donde $||\vec{u}|| = \sqrt{u_1^2 + u_2^2 + \dots + u_n^2}$.

4.2.3 Clasificación con todas las variables.

Por último, probaremos a clasificar con toda la información disponible. Aquí debemos tratar con atributos de distinto tipo, y por tanto, con distinto preprocesado. Por un lado, tenemos *Gene*, una variable categórica que debemos binarizar, y por otro el texto, que es un dato no estructurado. Para lidiar con ello, utilizaremos el preprocesado visto en la clasificación sin texto para la variable *Gene* y el de la clasificación con solo el texto para la evidencia clínica, exceptuando las técnicas de selección y aumento.

Clasificadores y parámetros a ajustar

Los parámetros a probar son iguales a los del apartado anterior. Los clasificadores son:

- NB Multinomial
- *Random Forest* (RF)
- SVC

4.3 Resultados

A continuación, pasamos a ver los resultados de los clasificadores y distintas técnicas de preprocesado, así como los resultados del mejor clasificador con las métricas complementarias (tasa de acierto, f1-score y ROC).

4.3.1 Clasificación sin texto

A continuación, veremos la configuración optima de los parámetros de cada clasificador:

- KNN: número de vecinos = 59.
- Naïve Bayes: con suavizado de Laplace.
- Árbol de Decisión: criterio de división = 'gini'

Gráfica y Análisis de Resultados

Como se puede observar en la **Figura 25**, a pesar de las numerosas técnicas y parámetros que se han probado, se obtienen resultados no muy buenos, ya que la variable Gene aporta poca información para clasificar.

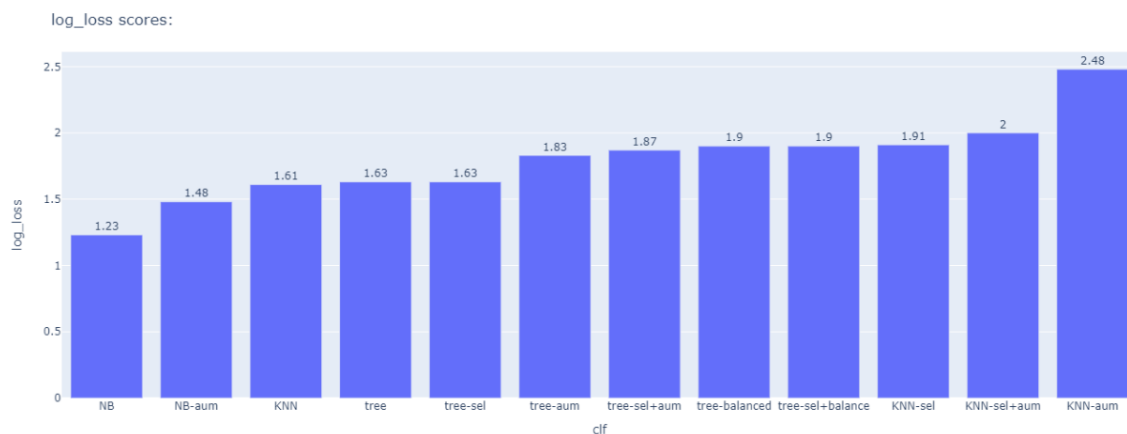


Figura 25. Resultados Clasificación sin Texto

Cabe destacar que nuestro mejor clasificador en este apartado ha sido el NB (**Figura 26**). Se puede observar que, a pesar de los malos resultados en acierto o en f1-macro, el ROC es bastante bueno, lo que significa que, aunque no hayamos acertado con la clase verdadera, si que le hemos dado una probabilidad bastante cercana a la clase elegida.

Esto tiene mucho sentido pues estamos ajustando la métrica *log_loss*, la cual está relacionada con reducir el error de probabilidad.

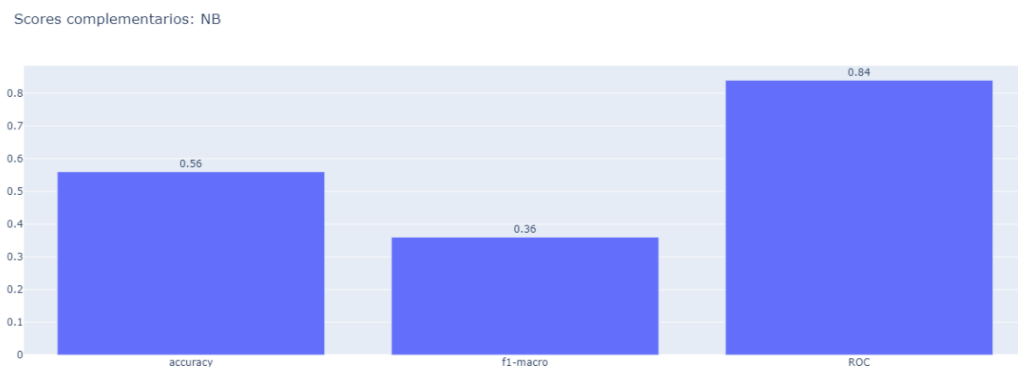


Figura 26. Resultados NB

4.3.2 Clasificación con solo el texto

A continuación, veremos la configuración subóptima de los parámetros de cada clasificador:

- KNN: número de vecinos = 59.
- Random Forest: criterio de división = 'gini'
- SVC: función para marcar la frontera = 'rbf'

Pasamos a ver la configuración subóptima respecto al texto:

Clasificador	Número de términos	n-gramas	Tf-idf	Norma tf-idf
KNN	5000	(1, 2)	False	l2
NB Multinomial	1000	(2, 2)	True	l2
NB Complementario	5000	(1, 1)	True	l2
Random Forest	1000	(1, 2)	True	l2
SVC	7000	(1, 1)	True	l2

Tabla 1. Configuración texto 1

Como norma general, se observa que el tf-idf es mejor función de puntuación que solamente la frecuencia, además que la normalización euclídea es la mejor en todos los casos. Un dato a tener en cuenta es que el tf-idf no es un buen discriminador para el SVC,

ya que necesita 7000 términos (muchos de ellos tendrán un tf-idf bajo) para lograr una buena clasificación.

Gráfica y Análisis de Resultados

Como se puede observar en la **Figura 27**, se obtienen resultados bastante mejores que en la clasificación sin texto, aunque las técnicas de aumento de instancias, balanceo de clases (NB Complementario) y selección de variables (salvo en el caso del NB Multinomial) siguen sin dar resultados. También se advierte que, a excepción del SVC, todos los demás clasificadores “pierden” contra el mejor clasificador del apartado anterior (NB), esto es porque dichos clasificadores, a pesar de ser más complejos, no lograron discriminar bien entre el ruido y la información útil.

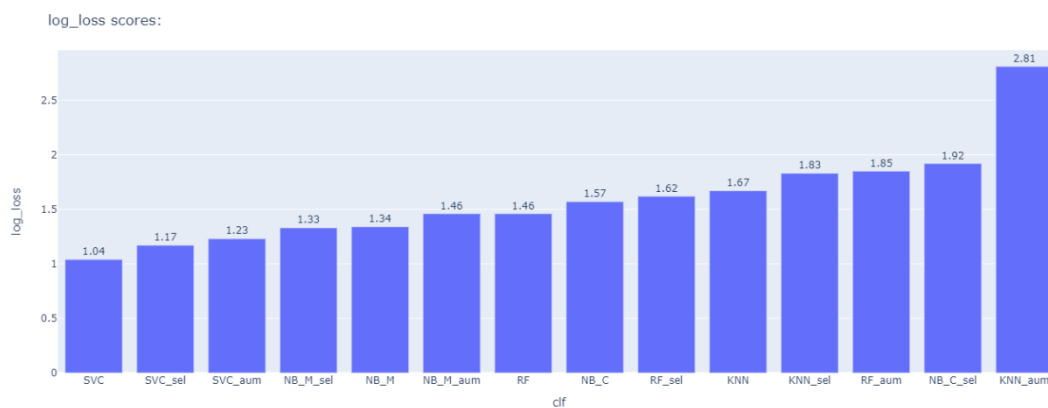


Figura 27. Resultados Clasificación con Texto

Pasamos a mostrar los resultados de las métricas complementarias del SVC (**Figura 28**). Notamos una mejoría general en todas las puntuaciones respecto al NB de la clasificación sin texto, sobre todo en la f1-macro.

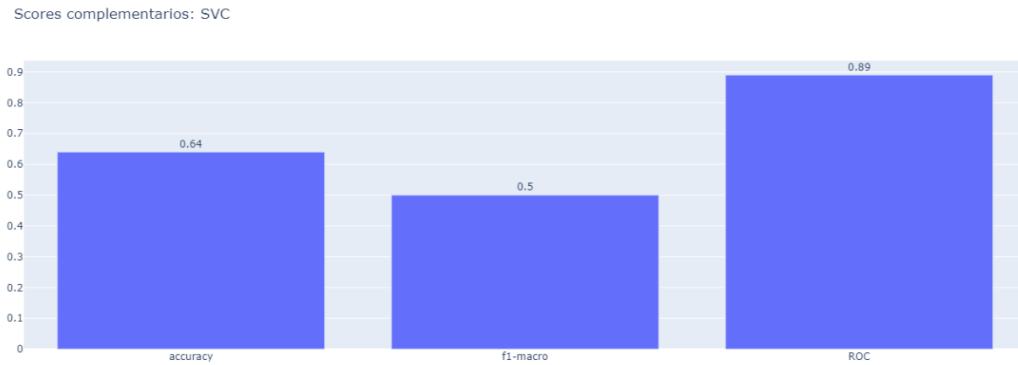


Figura 28. Resultados SVC

4.3.3 Clasificación con todas las variables

A continuación, veremos la configuración subóptima de los parámetros de cada clasificador:

- SVC: kernel = 'rbf'
- Random Forest: criterio de división = 'entropía'

Pasamos a ver la configuración subóptima respecto al texto:

Clasificador	Número de términos	n-gramas	Tf-idf	Norma tf-idf
NB Multinomial	3000	(2, 2)	True	l1
Random Forest	5000	(1, 1)	True	l1
SVC	7000	(1, 1)	True	l2

Tabla 2. Configuración texto 2

Vemos que el tf-idf sigue siendo mejor que la frecuencia y que las palabras individualmente aportan más información que en parejas.

Gráfica y Análisis de Resultados

La Figura 29 nos muestra una leve mejoraría para el clasificador SVC y una grande para el NB Multinomial, ya que logra “ponerse a la altura” del NB en su versión categórica aplicado con las variables no textuales de la clasificación sin texto. Por otra parte, el RF empeora su puntuación.

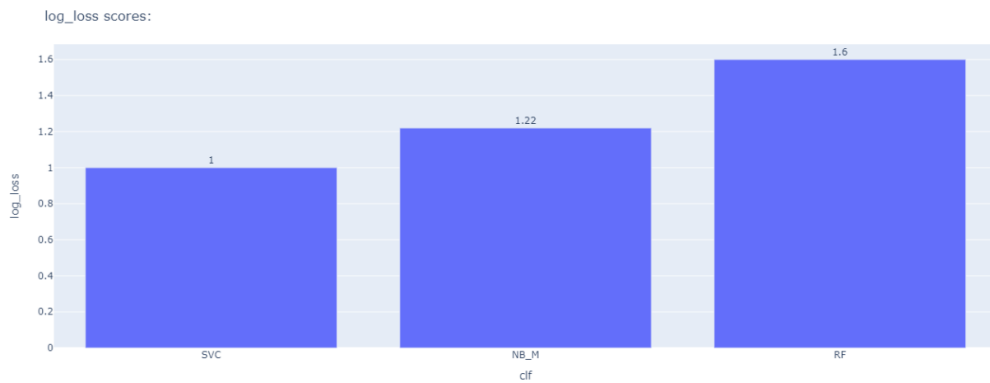


Figura 29. Resultados Clasificación con toda la información.

Vemos que las métricas complementarias del SVC de la Figura 30 son idénticas a las del apartado anterior. Demostrando que la gran cantidad de información aportada por el texto opaca a la generada por la variable *Gene*.

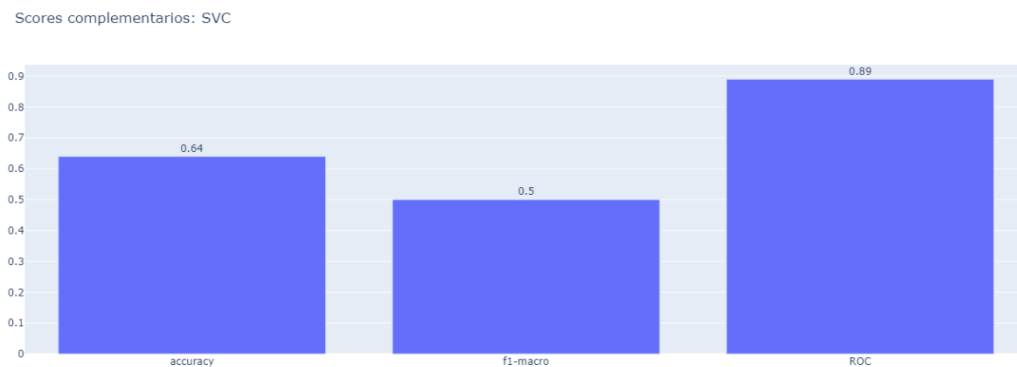


Figura 30. Resultados SVC

4.4 Conclusiones

Como demuestran los resultados, el procesamiento del lenguaje natural es una parte fundamental en este problema, ya que es posible extraer bastante información a partir de la evidencia clínica, y, por tanto, mejorar la clasificación de los tumores. Uno de los mayores retos ha sido la eliminación del ruido, motivo por el cual algunos clasificadores aun siendo más sencillos han mejorado su puntuación en la clasificación sin texto. De esta manera, hemos podido comprobar la importancia del preprocesado respecto a la obtención del *tf-idf* frente a tener en cuenta solo la frecuencia de los términos, además de la ampliación del conjunto de palabras a eliminar por ser poco relevantes (aparte de las comunes como los artículos, preposiciones, etc.). Cabe destacar, que las técnicas de

selección de variables, aumento de instancias o balanceo, no llegaron a mejorar en gran medida la clasificación como cabía esperar.

De todos los clasificadores, el KNN ha sido el que peor resultados ha tenido, seguido por el árbol de decisión y el RF. El NB ha demostrado funcionar de manera satisfactoria tanto en la clasificación sin texto como con todas las variables, aunque al parecer con solo texto no clasifica tan bien. El mejor clasificador con diferencia es el SVC, llegando a aprovechar bastante bien la información disponible. Como norma general, en todos los casos se consiguió una buena puntuación AUC, lo cual en el ámbito de la medicina es bastante bueno ya que, la clase real tiene la suficiente probabilidad como para que sea tomada en cuenta.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1 Conclusiones

La medicina personalizada es una de las aplicaciones más significativas del análisis de datos a los profesionales de la medicina y a los investigadores, ya que reduce en gran medida la carga de trabajo rutinario que estos tienen que hacer para llevar a cabo una investigación y llegar a realizar un diagnóstico. En esta tarea, el PLN es de vital importancia, pues los médicos toman muchas anotaciones y mucha de la información del paciente es difícil de representar sin ellas.

Se ha podido observar, como tener mayor información o clasificadores más complejos no garantiza una mejor clasificación si no se discrimina bien entre el ruido y la información útil. Otro aspecto fundamental fue el un ajuste de hiperparámetros, ya que muchos clasificadores llegaron a ser bastante sensibles al mismo, además de servir como preprocesado, ya que en algunos podíamos “balancear” las clases o “relajar” las fronteras de decisión para generalizar mejor. Otro aspecto a tener en cuenta fue la importancia de la exploración para llegar a entender mejor el dominio del problema e identificar información útil para la clasificación.

5.2 Trabajo futuro

Como trabajo posterior, contando con una mayor capacidad computacional, podría hacerse una búsqueda mas completa o uniforme (*gridsearch*) en lugar de aleatoria para el ajuste de hiperparámetros para la clasificación con todas las variables, además de “construir” un metclasificador (conjunto de clasificadores) con los mejores clasificadores.

Bibliografía

- [1] D. Cirillo and A. Valencia, “Big data analytics for personalized medicine,” *Curr. Opin. Biotechnol.*, vol. 58, pp. 161–167, 2019, doi: 10.1016/j.copbio.2019.03.004.
- [2] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, “Big data preprocessing: methods and prospects,” *Big Data Anal.*, vol. 1, no. 1, p. 9, Dec. 2016, doi: 10.1186/s41044-016-0014-0.
- [3] Z. Yin, Y. Wang, L. Liu, W. Zhang, and J. Zhang, “Cross-subject EEG feature selection for emotion recognition using transfer recursive feature elimination,” *Front. Neurobot.*, vol. 11, no. APR, Apr. 2017, doi: 10.3389/fnbot.2017.00019.
- [4] S. T. Nihan, “Karl Pearsons chi-square tests,” *Educ. Res. Rev.*, vol. 15, no. 9, pp. 575–580, 2020, doi: 10.5897/err2019.3817.
- [5] T. Hastie, J. H. (Jerome H. . Friedman, and T. Hastie, *The elements of statistical learning : data mining, inference, and prediction : with 200 full-color illustrations*. Springer, 2001.
- [6] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, “An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics,” *Inf. Sci. (Ny)*, vol. 250, pp. 113–141, Nov. 2013, doi: 10.1016/j.ins.2013.07.007.
- [7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002, doi: 10.1613/jair.953.
- [8] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text classification algorithms: A survey,” *Inf.*, vol. 10, no. 4, pp. 1–68, 2019, doi:

- 10.3390/info10040150.
- [9] V. García, R. A. Mollineda, and J. S. Sánchez, "On the k-NN performance in a challenging scenario of imbalance and overlapping," *Pattern Anal. Appl.*, vol. 11, no. 3–4, pp. 269–280, Sep. 2008, doi: 10.1007/s10044-007-0087-5.
 - [10] H. Zhang, "The optimality of Naive Bayes," *Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004*, vol. 2, pp. 562–567, 2004.
 - [11] J. D. M. Rennie, L. Shih, J. Teevan, and D. Karger, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers," *Proceedings, Twent. Int. Conf. Mach. Learn.*, vol. 2, no. 1973, pp. 616–623, 2003.
 - [12] W. Y. Loh, "Classification and regression trees," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011, doi: 10.1002/widm.8.
 - [13] Y. L. Pavlov, "Random forests," *Random For.*, pp. 1–122, 2019, doi: 10.1201/9780429469275-8.
 - [14] Y. Tang, Y. Q. Zhang, and N. V. Chawla, "SVMs modeling for highly imbalanced classification," *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 39, no. 1, pp. 281–288, 2009, doi: 10.1109/TSMCB.2008.2002909.

Anexo I. Título del anexo
