UNIVERSIDAD
SAN SEBASTIAN

FACULTAD DE INGENIERÍA
INGENIERÍA CIVIL INFORMÁTICA
SEDE BELLAVISTA

Algoritmos de machine learning para el análisis y predicción de permeabilidad molecular

Tesis para optar al título de Ingeniero Civil informática

Director: Alfredo Jesús Pereira Toloza

Codirector: Álvaro Muñoz Castro

Estudiante: Sebastián Alejandro Beas Aguilar

Santiago, Chile

2025

HOJA DE CALIFICACIÓN

En _____, el_____ de. _____ de _____ los abajo firmantes dejan constancia que el (la) estudiante. _____ de la carrera o programa de _____ ha aprobado la tesis para optar al título o grado académico de _____ _ con una nota de _____

_____
Profesor Evaluador

_____
Profesor Evaluador

_____
Profesor Evaluador

Índice

Índice de Tablas

Índice de Figuras

Resumen

El propósito de esta tesis fue implementar modelos clasificadores de machine learning que puedan predecir eficazmente la capacidad molecular de atravesar la Barrera Hematoencefálica. Con el fin de acelerar el proceso de síntesis de nuevos fármacos para tratar enfermedades del Sistema Nervioso Central.

Esto se hizo a través del entrenamiento de 4 modelos clasificadores diferentes: KNN, SVM, Random Forest y Gaussian Naive Bayes. Para esto se usó la base de datos pública B3DB la cual contiene moléculas previamente etiquetadas, luego se realizó un proceso de selección de características aplicando Información Mutua y RFEcv de manera continua. Finalmente, los parámetros de los modelos fueron optimizados a través de la técnica grid search. Se crearon versiones de los modelos sin selección de características con fines comparativos.

Luego de este proceso se obtuvo como principal resultado un modelo Random Forest que logró un AUC de 0.96 y una Especificidad de 0.93. Además, se obtuvieron las importancias de características para el modelo anteriormente mencionado, en donde algunas de las principales variables fueron: TPSA, qed y NOCount.

A partir de estos resultados se puede concluir que los modelos sí pueden predecir eficazmente la permeabilidad de las moléculas. Además, algunos de los modelos generados superan levemente modelos generados por otros autores usando los mismos datos.

Palabras clave: Machine learning, Permeabilidad, Random Forest, RFEcv.

Abstract

The purpose of this thesis was to implement machine learning classification models that can effectively predict the molecular ability to cross the blood-brain barrier. The aim was to accelerate the process of synthesizing new drugs to treat diseases of the central nervous system.

This was done by training four different classification models: KNN, SVM, Random Forest, and Gaussian Naive Bayes. For this, the public B3DB database was used, which contains previously labeled molecules. Then, a feature selection process was performed by continuously applying Mutual Information and RFEcv. Finally, the model parameters were optimized using the grid search technique. Versions of the models without feature selection were created for comparison purposes.

After this process, the main result was a Random Forest model that achieved an AUC of 0.96 and a specificity of 0.93. In addition, the feature importance for the model was obtained, where some of the main variables were: TPSA, qed, and NOCount.

Based on these results, it can be concluded that the models can effectively predict the permeability of molecules. In addition, some of the models generated slightly outperform models generated by other authors using the same data.

Keywords: Machine learning, Permeability, Random Forest, RFEcv

Introducción

Hoy en día las enfermedades del sistema nervioso central son una de las patologías que afecta a más personas en el mundo, en donde para la mayoría aún no hay un tratamiento completamente efectivo.

Las enfermedades neurodegenerativas, como el Alzheimer o el Parkinson, representan uno de los desafíos médicos más complejos de nuestro tiempo. Estas patologías no solo no han podido ser tratadas al 100% sino que su naturaleza progresiva y devastadora ha causado un temor generalizado en la sociedad por mucho tiempo.

De hecho, solo un 8% de las prescripciones para tratar enfermedades del Sistema Nervioso Central (SNC) son efectivos. Lo que demuestra la dificultad existente en poder crear fármacos que consigan tratar estas patologías.

Esto se debe principalmente a la Barrera Hematoencefálica. Esta es una estructura totalmente selectiva que cumple la función de proteger el sistema nervioso central, esto lo hace regulando el paso de sustancias desde la sangre al tejido cerebral, previniendo el paso de toxinas o patógenos dañinos, y de desregulaciones hormonales. De esta manera la Barrera Hematoencefálica se transforma en un problema para el desarrollo de fármacos enfocados en el SNC ya que no permite estos consigan llegar al cerebro y lograr su efecto.

En los últimos años se han realizado muchos intentos para poder darle fin a este permanente problema y cada vez la sociedad se ha acercado más a encontrar una cura para estas enfermedades, pero como se mencionó anteriormente aún es un desafío por superar.

Este problema no es solo un desafío médico, sino también económico y social. El costo de cuidado de pacientes con Alzheimer/Parkinson es inmenso, y el fracaso en encontrar fármacos efectivos que penetren la Barrera Hematoencefálica representa millones de pérdidas en investigación y desarrollo.

Actualmente existen muchos métodos que intentan predecir la permeabilidad de las moléculas (habilidad para traspasar la Barrera Hematoencefálica), ya sea métodos probados en animales o *in vitro*, muchos de los cuales requieren muchos recursos, tiempo o personas, por lo que son muy costosos. Por ello el desarrollo de fármacos efectivos para tratar enfermedades del sistema nervioso central se ve retrasado.

Es aquí en donde entran los métodos computacionales, en específico el machine learning. Esta herramienta es capaz de predecir la efectividad de los fármacos en cuestión de segundos, acelerando así el desarrollo de medicamentos que logren atravesar la barrera. Esto permite a los investigadores orientar sus esfuerzos en compuestos mucho más prometedores.

En esta tesis se abordará el desafío de la permeabilidad de las moléculas usando técnicas de Machine learning. Con el objetivo de clasificar las moléculas respecto a su capacidad de atravesar la Barrera Hematoencefálica, a la que en adelante nos referiremos como BBB (por sus singles en inglés *Blood Brain Barrier*). En consecuencia, la cualidad de una molécula para atravesar dicha barrera será definida como BBB+, mientras que se utilizará BBB- para las que no lo consigan.

Objetivos

Objetivo General:

Desarrollar y evaluar modelos de machine learning para predecir permeabilidad de moléculas a partir de su caracterización química, con el fin de apoyar procesos de diseño y síntesis de fármacos efectivos contra enfermedades del Sistema Nervioso Central.

Objetivos Específicos.

1. Recolectar, curar y organizar datos moleculares a partir de bases de datos públicas.
2. Generar descriptores moleculares estructurales y fisicoquímicos utilizando herramientas computacionales.
3. Desarrollar modelos predictivos utilizando algoritmos KNN, SVM, Random Forest y Naive Bayes ajustando sus hiperparámetros para optimizar desempeño.
4. Evaluar el desempeño de los modelos mediante métricas estadísticas y validación cruzada.

Revisión bibliográfica

Marco teórico

Contexto biológico

Para iniciar se dará un breve contexto biológico respecto al tema de la investigación.

Sistema nervioso central:

El sistema nervioso central o SNC está compuesto por el cerebro y la médula espinal. Este se encarga de recibir información sensorial, procesar dicha información y generar una respuesta motora (Squire, 2013). Es así como luego de recibir una señal externa el cerebro envía una señal eléctrica a través de la médula espinal hacia los músculos y/o glándulas para generar una respuesta.

Debido a lo anterior el SNC se vuelve un sistema vital para la homeostasis en el cuerpo humano.

Barrera Hematoencefálica BHE o BBB del inglés Blood Brain Barrier:

La Barrera Hematoencefálica es una barrera selectivamente permeable que regula el paso de moléculas desde el torrente sanguíneo hacia el sistema nervioso central. Su principal función es la protección del sistema nervioso central, evitando que traspasen toxinas nocivas y regulando la homeostasis en el cerebro (Abbott et al., 2010).

Algunas de las principales propiedades que definen el paso de una molécula a través de la Barrera Hematoencefálica hacia el SNC son:

- Lipofilicidad (LogP): Es la capacidad de un compuesto para disolverse en grasa o aceite.
- Área de superficie polar (TPSA del inglés Topological Polar Surface Area): Suma de la superficie de todos los átomos polares en una molécula.
- Peso Molecular (MolWt de Mol Weight en inglés): suma de todas las masas atómicas de los átomos de una molécula.

Estas propiedades químicas son críticas para el diseño de fármacos eficaces en el SNC, ya que se ha demostrado que la lipofilicidad y la superficie polar son clave a la hora de definir la permeabilidad cerebral (Pajouhesh & Lenz,

2005). De hecho, es debido a esta permeabilidad selectiva que la barrera actúa como un “cuello de botella”, provocando que muchos fármacos no resulten eficaces a la hora de tratar enfermedades del SNC (Pardridge, 2005)

Es debido a esta estructura por la que algunos fármacos no resultan eficaces a la hora de tratar enfermedades que afectan el sistema nervioso central.

Enfermedades del SNC:

En la actualidad existe un gran número de personas que padecen o han padecido, enfermedades que afectan el sistema nervioso central, las cuales en su mayoría aún no poseen ningún tratamiento efectivo (Gribkoff & Kaczmarek, 2017). Algunas de las cuales son:

- Alzheimer, corresponde al 60%-70% de los casos de demencia en el mundo(Organización Mundial de la Salud, 2025).
- Accidente Cerebrovascular (ACV), en 2021 fue una de las principales causas de muerte globalmente y lo sigue siendo(Organización Mundial de la Salud, 2024).
- Parkinson. Según la OMS (Organización Mundial de la Salud, 2023) en 2019 más de 8,5 millones de personas padecían esta enfermedad.

Metodologías basadas en Machine Learning.

Machine Learning:

(Mitchell, 1997) define el aprendizaje automático como como un proceso en el que un programa mejora su desempeño (D) en una tarea (T) a través de la experiencia (E). En el contexto actual podemos seguir viendo su definición ya que los datos pueden ser la experiencia, la tarea el reconocimiento de patrones y el desempeño son las métricas de desempeño como la precisión y la exactitud.

En términos más actuales, el machine learning es una rama de la Inteligencia Artificial la cual se centra en analizar datos y usar algoritmos sin recibir instrucciones explicitas, con el fin de aprender de los datos consiguiendo predecir valores y reconocer patrones no reconocibles a simple vista. Estos valores a predecir pueden ser continuos o discretos dependiendo del algoritmo.

Existen 2 principales tipos de algoritmos de machine learning:

Aprendizaje supervisado.

Este grupo de algoritmos se centra en analizar datos previamente etiquetados para entender la relación entre los datos y su etiqueta con el fin de predecir la etiqueta de datos no vistos.

Aprendizaje no supervisado.

Grupo de algoritmos centrado en encontrar patrones y relaciones dentro de los datos entregados.

Con el fin de optimizar el análisis, los datos son estructurados matricialmente en donde cada fila es una observación/registro y las columnas corresponden a sus atributos. Técnicamente, a las columnas se les denomina características o atributos, y la variable dependiente o a predecir recibe el nombre de etiqueta o clase objetivo.

La principal herramienta usada para medir el desempeño de los modelos clasificación es la matriz de confusión, de donde nacen múltiples métricas importantes.

Matriz de confusión:

Matriz que muestra la cantidad de aciertos y desaciertos de un modelo predictivo de clasificación, en donde un eje demuestra los valores reales y el otro los valores predichos. Separando en 4 clases para un problema de clasificación binaria:

- Verdaderos positivos (VP): representa la cantidad de predicciones acertadas para la clase positiva.
- Falsos positivos (FP): representa la cantidad de todos los casos negativos identificados como positivos.
- Falsos negativos (FN): representa la cantidad de todos los casos positivos que fueron identificados como negativos.
- Verdaderos negativos (VN): representa la cantidad de predicciones acertadas para la clase negativa.

Figura 1. Matriz de confusión.

De esta matriz nacen diferentes métricas importantes.

Métricas de desempeño.

Exactitud: Tasa de predicciones acertadas del modelo. Se mide de la siguiente manera.

$$Exactitud = \frac{VP + VN}{VP + VN + FP + FN} \quad (1)$$

En casos de desbalanceo de clases (las clases objetivo tienen cantidades de registros muy diferentes) esta medida es muy mala. Por ejemplo: en un set de datos de 100 registros en donde la clase A representa 90 de ellos, aun cuando el modelo solo clasifique como A, la precisión será de 90%.

Precisión: Tasa de acierto que tiene el modelo al predecir la clase positiva. Se mide de la siguiente manera:

$$Precision = \frac{VP}{VP + FP} \quad (2)$$

Esta métrica es importante cuando el coste de los Falsos Positivos es alto y queremos optimizar los modelos para evitar estos casos.

***Sensibilidad* o Recall:** Mide la capacidad del modelo de detectar los realmente positivos.

$$Sensibilidad = \frac{VP}{VP + FN} \quad (3)$$

Métrica importante cuando se quiere que no pase desapercibido ningún caso positivo.

Especificidad: Mide la capacidad del modelo de detectar los registros realmente falsos (también se le conoce como Sensibilidad/Recall para la clase negativa).

$$Especificidad = \frac{VN}{VN + FP} \quad (4)$$

Como contraparte de la métrica anterior, esta es importante si no se desea que los casos negativos pasen desapercibidos.

F1-score: Esta métrica combina ambas Precisión y sensibilidad. Es la media armónica entre estas dos medidas.

$$F1 - score = 2\frac{Precision * Sensibilidad}{Precision + Sensibilidad} \quad (5)$$

Al combinar ambas medidas, esto lo hace más robusto que la Exactitud ante set de datos desbalanceados.

Área bajo la curva ROC: Mide la capacidad del modelo para distinguir (discriminar) entre clases. Se calcula graficando la Sensibilidad frente a la Tasa de Falsos Positivos a través de distintos umbrales de decisión.

Algoritmos

K-Nearests Neighbors (KNN)(Cover & Hart, 1967):

Algoritmo de aprendizaje supervisado que asume que los puntos cercanos son parecidos. De esta manera asigna a un nuevo punto la misma clase de los puntos más cercanos mediante una votación en el caso de clasificación y un promedio en el caso de regresión.

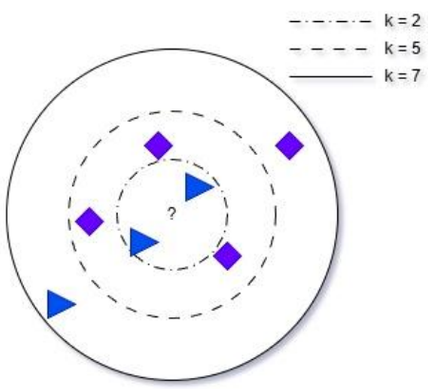


Figura 2. Ejemplo clasificación KNN.

Para este algoritmo el hiperparámetro más importante es k (el número de vecinos) ya que este valor gobierna la clasificación de nuevos registros. Debido a esto encontrar el valor de k correcto es imprescindible: un valor de k alto puede significar robustez ante el ruido en el set de datos, pero generalizar en exceso causando subajuste, por otro lado, un valor bajo causa alta sensibilidad ante el ruido, causando sobreajuste.

Para encontrar los puntos más cercanos comúnmente se utiliza la distancia Euclidiana.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2} \quad (6)$$

En caso de empate en la votación el algoritmo asigna la clase del nuevo punto según cual sea la clase moda dentro de todo el set de datos, en caso de que las clases tengan el mismo numero de instancias se asigna el primer valor de moda que encuentra.

Para resolver ese problema de empates además de otros se creó una versión mejorada de KNN.

Weighted KNN(Dudani, 1976):

Versión de KNN que asigna pesos a los vecinos más cercanos, de manera que un vecino que se encuentre mas cerca del punto a predecir tenga mas fuerza que un punto que se encuentre mas lejos. El valor del peso es el inverso de la distancia.

$$w_i = \frac{1}{d(x_q, x_i)} \quad (7)$$

Random Forest(Breiman, 2001)**:**

Algoritmo de aprendizaje supervisado que entrena un conjunto de árboles de decisión diferentes para realizar una predicción (o Bagging). En donde cada árbol predice el mismo registro y se hace una votación, en el caso de ser clasificación, para definir el valor a predecir o se calcula un promedio en el caso de regresión.

La manera en que se generan estos árboles de decisión es lo mas importante de los algoritmos. Cada árbol es independiente y cada uno de ellos esta creado en base a vectores generados igualitariamente distribuidos. Estos vectores definen que cantidad de filas muestrean del set de datos original y que columnas o características a elegir, logrando que cada árbol sea independiente uno del otro, pero creados con las mismas reglas.

Máquina de vectores de soporte (SVM de su nombre en inglés Support Vector Machine)(Cortes et al., 1995):

Es un algoritmo de aprendizaje supervisado que busca el plano que mejor separa las clases objetivo. Esto lo logra buscando el hiperplano que maximiza la distancia entre las 2 clases.

Esto se puede observar en el diagrama de (Yang et al., 2019).

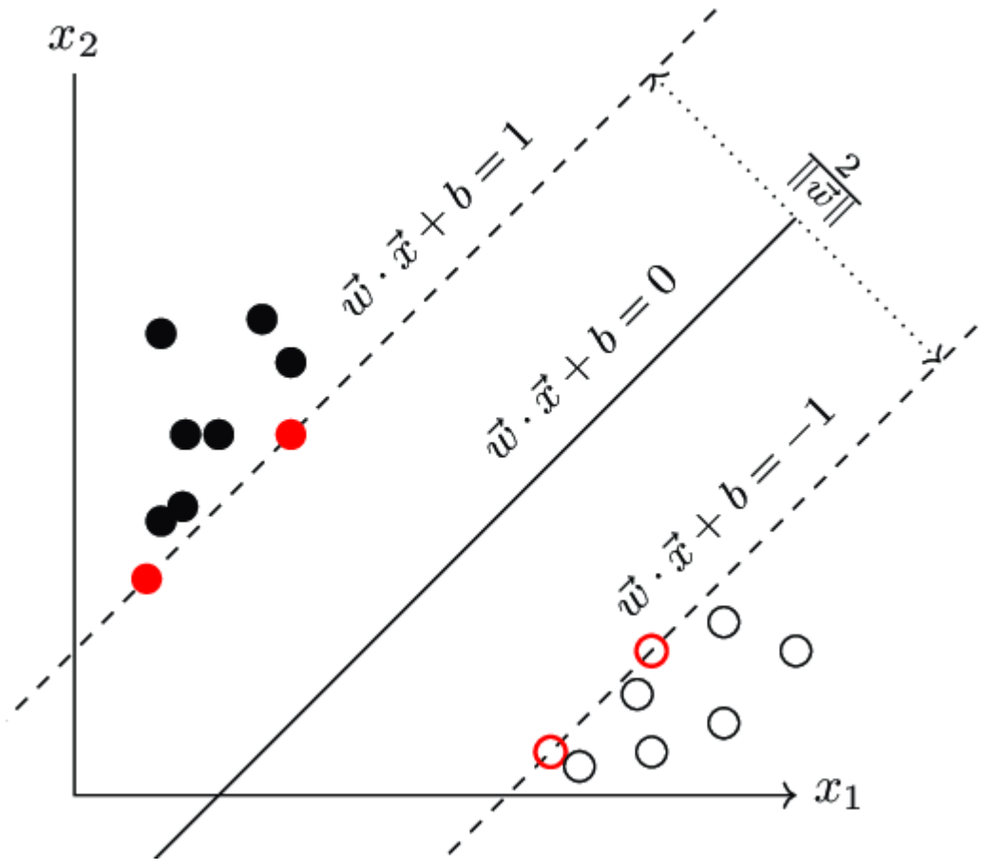


Figura 3. Diagrama explicativo de SVM.

El hiperplano que separa ambas clases se define como:

$$f(x) = (w \cdot x + b) \tag{8}$$

La fórmula principal que usa SVM es la del margen suave, que permite que los puntos puedan equivocarse, o no estar bien separados, pero le agrega un costo a eso. Esto debido a que en la práctica no todos los puntos se pueden separar perfectamente.

$$\min_{w,b,\xi}\left(\frac{1}{2}\left|\left|w\right|\right|^2 + C\sum_{i=1}^{n}\xi_i\right) \quad (9)$$

En donde:

- w es el vector de pesos, perpendicular al plano separador.
- $\left|\left|w\right|\right|^2$es la norma Euclidiana al cuadrado del vector de pesos
- C es el hiperparámetro de regulación o costo, un valor alto penaliza fuertemente los errores de clasificación, lo que lleva a un margen estrecho, un valor pequeño permite mas errores a cambio de un margen más ancho.
- ξ_i es la variable de holgura. Mide cuanto le falta a un punto i para estar en el lado correcto del margen.

De esta manera:

- $\frac{1}{2}\left|\left|w\right|\right|^2$ busca maximizar la separación de ambas clases. O en otras apalabras maximizar el margen.
- $C\sum_{i=1}^{n}\xi_i$ es la sumatoria de errores de clasificación, busca minimizar el error de clasificación.

De este modo el algoritmo busca el punto óptimo en donde la suma de ambos, la separación de clases y la sumatoria de errores, sea mínima.

***Gaussian Naive Bayes*:**

Algoritmo de aprendizaje supervisado con bases estadísticas. Este asume que todas las columnas del set de datos son completamente independientes entre si, de ahí viene el término “Naive”. En específico esta versión del algoritmo asume que las columnas siguen una distribución (o Gaussiana). Esto lo hace un algoritmo muy rápido y barato computacionalmente hablando.

Como lo dice su nombre basa en el teorema de Bayes, el cual explica la probabilidad de que ocurra un evento en base a otro.

$$P(A/B) = \frac{P\,(B/A) * P(A)}{P(B)} \quad (10)$$

Este algoritmo define que clase predecir dependiendo de cual de las clases tiene más probabilidades de ocurrir. Esto lo logra utilizando la Función de Densidad de Probabilidad Gaussiana.

$$P(x_i|y) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (11)$$

Luego de obtener las probabilidades de un registro nuevo, estas son multiplicadas y el resultado de ese producto es la probabilidad que se usa para la decisión de clase final.

Al momento de entrenar este algoritmo predictivo, el modelo lo único que hace es guardar la varianza y la media para cada columna, separándolo por clase.

Selección de características

Existen 3 tipos de algoritmos de selección de características:

- Filtro: Algoritmos que principalmente usan métodos estadísticos en los datos para realizar la selección, entregando una puntuación a cada característica y luego seleccionando k mejores características. Solo revisan la interacción entre una variable y la clase objetivo, no la relación entre variables y la clase objetivo.
  Son muy rápidos y baratos computacionalmente
- Envoltura: Entrenan un modelo predictivo repetidas veces obteniendo una métrica que optimizar cada vez. Este tipo de selección de características trata el problema como una búsqueda, ya que iteran múltiples veces hasta encontrar la combinación de características que maximice una métrica de desempeño previamente definida.
  Debido a que entrenan un modelo predictivo en cada iteración estos algoritmos se vuelven muy costosos computacionalmente.
- Integrados o embebidos: Se refiere a mecanismos internos que tienen ciertos modelos predictivos al momento de entrenarse, los cuales los ayudan a decidir cuáles son las características más importantes.

Para lograr un mejor rendimiento de los modelos se usaron 2 algoritmos de selección de características de diferentes tipos.

Información mutua (Mutual Information):

Algoritmo de selección de características del tipo filtro, basado en la teoría de la información y la Entropía. Este busca la relación entre una columna y la clase objetivo calculando cuanta incertidumbre de la clase objetivo se reduce al conocer la columna analizada. Logrando atrapar relaciones de diferentes tipos, no solo lineales como lo hace la correlación de Pearson.

La entropía, la incertidumbre base, es definida como:

$$H(Y) = -\sum_y p(y) \log_2\big(p(y)\big) \tag{12}$$

Pero lo información mutua se calcula exactamente con la Divergencia de Kullback-Leibler entre la distribución conjunta de los datos y el producto de sus distribuciones marginales.

$$I(X;Y) = \sum_{x,y} p(x,y) \log_2 \left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{13}$$

En esta ecuación:

$p(x,y)$ representa la co-ocurrencia entre las 2 variables o distribución real.

$p(x)p(y)$ representa la distribución teórica si las 2 variables fueran independientes.

Por lo tanto, la información mutua cuantifica cuanta información se gana de la variable objetivo al rechazar la hipótesis de que las variables son estadísticamente independientes. De este modo, un valor alto indica una alta dependencia, ya sea lineal o no lineal, entre la característica analizada y la variable objetivo.

Recursive Feature Elimination with cross-validation (RFEcv)

RFEcv, o Eliminación recursiva de características con validación cruzada en español, es un algoritmo de selección de características del tipo de envoltura. Este es una variación del RFE normal, que consiste en entrenar un modelo y prueba su

desempeño, luego elimina un cierto número de características y entrena otro modelo. Así hasta encontrar la combinación de características que maximiza cierta métrica de desempeño. La diferencia principal con RFE normal es que en ese se define una cantidad de características a las que llegar.

Para seleccionar que características eliminar depende de que algoritmo se usó para el entrenamiento.

Este proceso de entrenar un nuevo modelo en cada iteración lo hace muy costoso computacionalmente por lo que comúnmente se usa luego de métodos de selección de atributos de tipo filtro.

Balanceo de clases

Desbalance de clases:

Este término hace referencia a una diferencia significativa en la cantidad de registros para cada una de las clases objetivo.

El problema del desbalance de clases es que puede causar sesgo dentro de los modelos predictivos, de manera que este puede aprender mucho de una de las clases, pero ignorar la minoritaria.

En base a este problema se desarrollaron diferentes técnicas para contrarrestarlo.

Balanceo de clases:

El balanceo de clases comprende un grupo de técnicas orientadas a mitigar la diferencia en la cantidad de registros de las clases. Esto lo pueden lograr ya sea, eliminando registros de la clase mayoritaria (*under-sampling*) o agregando registros a la clase minoritaria (*over-sampling)*.En esta tesis se usará SMOTE, una de las estrategias de sobremuestreo mas usadas en la literatura.

SMOTE: Synthetic Minority Over-sampling Technique (Chawla et al., 2002)**:**

Esta técnica de sobremuestreo crea registros sintéticos entre una muestra de la clase minoritaria y un vecino cercano de esa muestra. De manera que cada registro nuevo es calculado como:

$$Z = P + \lambda(Q - P) \tag{14}$$

En donde:

- Z es el nuevo registro sintético.
- P es un registro de la clase minoritaria.
- Q es un vecino cercano de P.
- λ es un numero al azar entre el 0 y el 1, incluyéndolos, siguiendo una distribución normal.

Estado del arte

Enfoques clásicos y modelos de referencia:

Historicamente la predicción de la permeabilidad de la Barrera Hematoencefálica (BBB) fue abordada mediante modelos lineales y bayesianos. Autores como (Martins et al., 2012) establecieron líneas bases usando modelos como Maquinas de Vectores de Soporte y Random forest bajo un enfoque bayesiano, logrando precisiones cercanas al 95%. Además en un trabajo mas reciente (V. Kumar et al., 2024) usó métodos más antiguos como lo es el Análisis Discriminante Lineal (LDA), demostrando que métodos más simples aún pueden ser competitivos. Sin embargo, estos trabajos a menudo carecían de generalización debido a que usaban set de datos pequeños y no estandarizados.

Cajas negras, Deep Learning y Ensambles:

Recientemente la literatura a comenzado a preferir modelos de alta complejidad (“Cajas Negras”) en un intento de capturar relaciones no lineales en las moléculas. (Shaker et al., 2021) creó un modelo usando el algoritmo Light Gradient Boosting Machine (LGBM) junto con descriptores generados por el software Dragon, sacrificando interpretabilidad por potencia, al que llamaron LightBBB. Por otro lado, (Tang et al., 2022) propusieron Deep-B un modelo que combina procesamiento de lenguaje natural y visión por computadora en una visión de Deep Learning compleja. Por su lado, (R. Kumar et al., 2022) presentaron DeepPred-BBB, el cual usa redes neuronales convolucionales(CNN). Sin embargo, aunque estos modelos lograron buenas métricas de desempeño, autores recientes critican su falta de interpretabilidad, debido a su naturaleza “Caja Negra”, lo que impide entender las estructuras químicas que permiten la permeabilidad en una molécula.

Estrategias para el balance de datos:

Un desafío importante es el desbalance de clases que actualmente existe, ya que los data sets poseen mayor cantidad de moléculas permeables (BBB+) que no permeables (BBB-).(Shi et al., 2021) abordaron explícitamente este problema probando múltiples estrategias de sobremuestreo, incluyendo SMOTE, ADASYN y Upsampling. Su estudio concluyó que la combinación de Upsampling con XGBoost superaba a otras técnicas, alcanzando una exactitud del 96% en un set de validación externo. Sin embargo, Shi et al. entrenaron sus modelos con un data set más pequeño y antiguo (aprox. 2354 moléculas), lo cual es significativamente menor al estándar actual del data set B3DB (7807 moléculas), la cual se utiliza en esta tesis. Además, el uso agresivo de técnicas sintéticas de sobremuestreo como SMOTE puede causar sobreajuste en el modelo.

Interpretabilidad y nuevos descriptores:

Como respuesta a los modelos de Deep Learning o “Caja Negra”, (Jia & Sosso, 2024) buscaron maximizar la interpretabilidad de sus modelos, usando el dataset B3DB. En lugar de usar miles de descriptores complejos, usaron los “cliques”, fragmentos moleculares funcionales, junto con modelos Naive Bayes y Random Forest . Su trabajo demostró que es posible conseguir igualar el rendimiento de modelos más complejos como DeepPred-BBB o LightBBB sin sacrificar interptretabilidad. Aunque con este trabajo lograron identificar algunos grupos funcionales clave, su modelo Naive Bayes es un clasificador más débil en términos de potencia predictiva pura en comparación con los ensambles más avanzados.

La brecha existente:

A pesar de los avances previamente mencionados, existe una brecha notable en la literatura. Por un lado, a pesar de su buen desempeño, los modelos de Shi et al fueron entrenados en una cantidad baja de registros(aprox. 2354 moléculas) en comparación a las de B3DB (aprox. 7800 moléculas) lo que hace que no tengan una buena capacidad de generalización, además obtuvieron una diferencia de desempeño considerable en el set de testeo externo lo que puede ser causa de un sobreajuste debido a la generación sintética de registros que utilizaron antes de su selección de características. Por otro lado, los modelos de (Jia & Sosso, 2024) son un poco débiles en términos de desempeño e ignoraron completamente el factor 3D y la conectividad global de las moléculas.

Es por eso por lo que esta tesis propone el uso de los descriptores de RDKit junto con fingerprints de Morgan. Además, se propone el uso de diferentes técnicas de selección de características con el posterior aplicacíon de SMOTE para balancear clases.

Metodología.

Recolección de datos.

Para este estudio se utilizó el conjunto de datos público llamado ‘B3DB’. Este recurso ha sido utilizado anteriormente en trabajos similares, como el de Jia & Sosso, quienes desarrollaron diferentes modelos para predecir la permeabilidad con el objetivo de conseguir resultados interpretables (Jia & Sosso, 2024).

Este set de datos cuenta con 7807 moléculas diferentes, de las cuales 2851 no atraviesan la barrera (BBB-) y 4956 si lo logran (BBB+). Se puede ver una clara diferencia entre la cantidad de registros entre los 2 tipos de moléculas, esto es conocido como desbalance de clases. Esto causa sesgo a la hora de la predicción de los modelos.

Caracterización de los datos y preprocesamiento.

La caracterización de moléculas del set de datos se refiere al proceso de generar nuevas columnas(también conoció como características) para agregar información relevante en la que los modelos predictivos se puedan apoyar para realizar sus funciones.

Para la generación de columnas para el modelo se eligieron 2 grupos de descriptores moleculares.

1) Set completo de descriptores de RDKit: Esta es una librería de Python enfocada en el análisis químico. La cual ofrece un total de 217 descriptores numéricos en la versión 2025.3.5.

   Se escogió este set de descriptores ya que abarca un rango amplio de características moleculares 2d. Contiene propiedades fisicoquímicas, conteo de átomos y enlaces, índices topológicos, entre otros.
2) Fingerprint de Morgan: Se genero un fingerprint de Morgan de 1024 bits y 2 de radio (Rogers & Hahn, 2010). Esto entrega 1024 descriptores binarios, en

donde cada uno de estos representa la presencia o ausencia de subestructuras moleculares específicas. Este conjunto igualmente fue calculado usando la librería RDKit.

Para el preprocesamiento se usó diferentes técnicas.

1) Chequeo de columnas que contengan solo 0's. Esto se hizo debido a que por la naturaleza de los descriptores es muy probable que algunas columnas no contengan registros diferentes de 0 al analizar propiedades muy específicas. Se elimino 5 columnas pertenecientes a los descriptores de RDKit.

2) Chequeo de valores nulos. Solo se encontraron 5 registros/filas que contenían valores nulos. Los cuales fueron eliminados.

3) Limpieza varianza 0. Se elimino todas las columnas que tuvieran varianza igual a 0 ya que estas no aportan ninguna información valiosa a los modelos predictivos.

4) Umbral de coeficiente de variación. Luego de eliminar todas las columnas con varianza 0 también se eliminó las que poseían un Coeficiente de variación menor o igual a 15%. Esto debido a que un coeficiente de variación bajo significa que los datos presentan poca dispersión relativa lo que significa que estas características no sean muy discriminativas a la hora de diferenciar la clase objetivo (BBB+/BBB-). A continuación, se presenta la formula del coeficiente de variación:

$$Coeficiente\ de\ variación\ = \frac{\sigma}{\mu} = \frac{Desviacion\ estandar}{Media} \qquad (15)$$

5)Análisis de correlación. Se utilizó un criterio de Correlación de Pearson, eliminando las columnas que tuvieran un coeficiente mayor a 0.98 con otra. Eligiendo cuál de las 2 columnas a eliminar según cual tenía un menor coeficiente de correlación con la variable objetivo. Se realizó este filtro ya que un alto coeficiente de correlación significa que 2 características pueden ser explicadas entre sí, por lo que 1 de las 2 realmente no aporta nueva información al modelo.

Luego de la generación de descriptores y su posterior preprocesamiento se terminó con un Dataset de 7800 filas y 1209 columnas.

Selección de características.

Para comenzar el proceso de selección de características se realizó una división de entrenamiento y testeo con 80% de los datos para entrenamiento y el 20% sobrante para testeo.

También, previo a la selección de atributos se crea una copia de respaldo del conjunto de datos para una futura comparación del resultado de los modelos entre un conjunto sin ningún cambio y el que si paso por el proceso de selección de atributos.

Con el fin de mejorar el rendimiento de los modelos se aplicaron 2 algoritmos de selección de características. Esto de manera continua entregando al segundo algoritmo el set de datos con las características seleccionas por el primero.

1) Mutual Information o información mutua en español. Algoritmo que mide la dependencia entre 2 variables calculando cuanta incertidumbre de la variable objetivo se pierde si se conoce cierta característica. Esto nos entrega un ranking de valores en donde se mantuvo el 75% de las columnas con puntuación más alta.

2)RFEcv. Recursive feature elimination with cross validation o en español, eliminación recursiva de características con validación cruzada (Guyon et al., 2002). Algoritmo que prueba un modelo con K características y con cada iteración elimina un cierto porcentaje de estas. A diferencia del RFE normal en el que se define un numero de columnas a mantener, este lo hace automáticamente seleccionando la mejor cantidad de columnas, lo que hace que sea mucho más costoso computacionalmente.

En este caso el algoritmo puntuó los modelos según su f1-score ya que es considerada una métrica de evaluación más robusta por si sola. Se usó validación cruzada de 5 pliegues.

Finalmente, luego del proceso de selección de características se obtuvo un nuevo set de datos con 897 características.

Entrenamiento y optimización de hiperparámetros.

Debido a que el set de datos esta desbalanceado en una razón de 2851 registros para la clase negativa (BBB-) y 4956 registros para la clase positiva (BBB+) se decidió balancear los datos usando la técnica SMOTE, la cual crea registros sintéticos de la clase más pequeña para igualar la cantidad de registros por clase.

Este proceso de balanceo de clases se aplicará luego de la selección de características debido a que de lo contrario puede generar sobreajuste en los modelos.

Para cumplir con la tarea de predecir la permeabilidad de las moléculas se entrenará diferentes modelos de machine learning supervisado, esto significa que se analizará un conjunto de datos previamente etiquetados para aprender de ellos y lograr predecir la permeabilidad de moléculas desconocidas por el modelo.

Se eligieron 4 modelos diferentes para entrenar, cada uno de estos pertenecientes a diferentes paradigmas de aprendizaje para obtener una mejor comparación.

Los algoritmos fueron:

Weighted KNN(Dudani, 1976): Una versión de KNN(Cover & Hart, 1967) que considera la distancia de los puntos para asignarles un peso, de manera que los puntos más cercanos tienen más fuerza que los lejanos a la hora de asignar una clase.

Random Forest(Breiman, 2001): Algoritmo que entrena varios árboles de decisión diferentes, para luego hacer que cada uno de estos prediga un punto y luego hacer una votación entre todos estos modelos para hacer la predicción “oficial”, lo que lo hace robusto y resistente a sobre-ajuste.

Máquina de Vectores de Soporte (SVM de su nombre en inglés Super Vector Machine)(Cortes et al., 1995): Algoritmo robusto ante alta dimensionalidad que busca el hiperplano que mejor separe la clase objetivo. Puede usar diferentes kernel para aumentar la dimensión de los datos de manera de que en esa dimensión la clase objetivo si sea separable.

Gaussian Naive Bayes: Predice la clase objetivo de manera rápida y estadísticamente, asumiendo que todas las columnas o características siguen una distribución normal. Elegido para otorgar comparación ante modelos más robustos.

Cada una de las versiones de estos algoritmos fueron las de la librería scikit-learn de Python.

Todos estos modelos fueron entrenados en los datos de entrenamiento, que es el 80% de los datos originales.

Se empleó la técnica de búsqueda por rejilla (en ingles grid search) con el objetivo de hallar la combinación óptima de hiperparámetros dentro del espacio de

búsqueda definido. Para la evaluación del desempeño de los modelos generados por grid-search se usó una validación cruzada de 5 pliegues.

Aquí cada uno de los parámetros que se optimizo:

Tabla 1. Hiperparámetros optimizados por algoritmo.

Modelo	Hiperparámetros
Weighted KNN	- *'n_neighbors'*: rango de 1 a 20.
Random Forest	- *'n_estimators'*: 100 a 1000 con saltos de 50. - '*criterion*': gini, entropy o log_loss.
SVM	- *'Kernel'*: rbf o poly - *'C'*: 0.1, 1, 10, 100. - *'Gamma'* (solo para el kernel 'rbf'): 1, 0.1, 0.01, 0.001. - '*Degree*' (solo para el kernel 'poly'): 2, 3, 4, 5.
Gaussian Naive Bayes	No se optimizo parámetros.

Evaluación e interpretación de resultados.

El rendimiento final de los modelos de clasificación se evaluó sobre el 20% de los datos (el set de testeo/prueba) utilizando las siguientes métricas:

.

- Exactitud: Tasa de predicciones acertadas entre todas las predicciones.
- Precisión: Tasa de acierto cuando el modelo predice Verdadero o la clase positiva.
- Sensibilidad o Recall: Mide la capacidad del modelo de detectar los positivos entre positivos.
- Especificidad: Mide la capacidad del modelo de detectar los falsos entre los falsos

- F1-score: Medida que combina Precisión y sensibilidad. Es la media armónica entre estas dos medidas. Es buena para datos desbalanceados.
- Área bajo la curva ROC: Mide la capacidad del modelo de distinguir entre clases. Esto a través de la Sensibilidad (o tasa de verdaderos positivos) y la tasa de falsos positivos.

Con el fin de tener una mayor comprensión química de los resultados se obtendrá la importancia de características(Feature Importance) para ambos modelos de Random Forest. Esta importancia se deriva de la métrica de impureza utilizada durante el entrenamiento (internamente se utiliza como criterio de división). La impureza de cada característica representa la reducción de impureza que aporta. Estas medidas de importancia son entregadas de manera que la suma total de ellas sea igual a 1.

Resultados

Creación del set de datos caracterizado

Como resultado de la caracterización, preprocesamiento y selección de características se obtuvo un conjunto de datos con 897 columnas y 7800 filas.

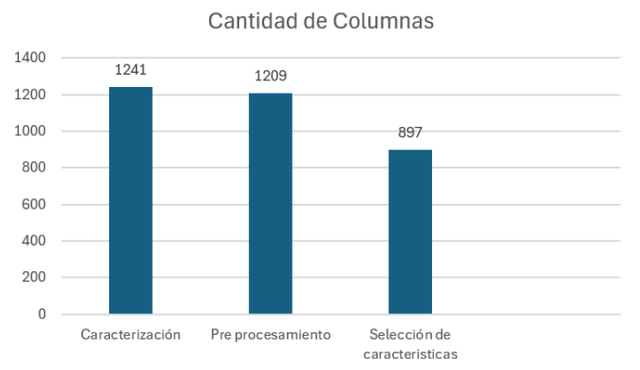


Figura 4. Columnas por etapa.

Como parte del análisis exploratorio de datos se generó 3 boxplots de 3 principales descriptores generados por RDKit los cuales fueron: la lipofilicidad (MolLogP), el peso molecular (MolWT) y la superficie polar total (TPSA). En la

Figura 5 se puede ver que las características MolLogP y MolWt no presentan una diferencia entre la distribución de las clases (BBB-/BBB+) muy significativa. Por otro lado, la característica TPSA si demuestra una pequeña separación en la distribución de las clases como se puede apreciar en la Tabla 2.

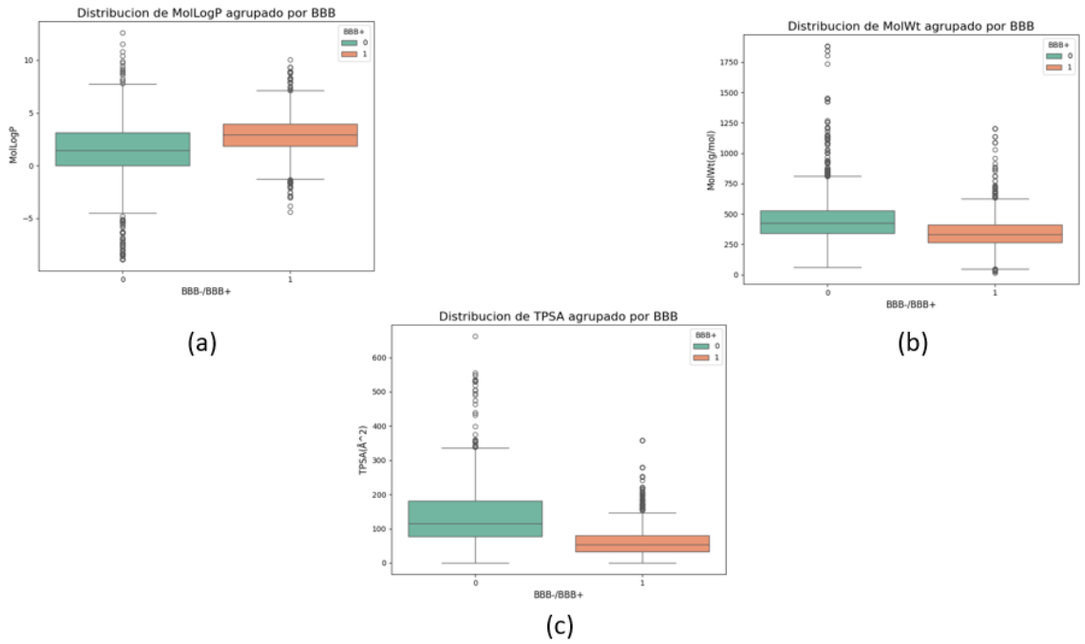


Figura 5. Boxplots de 3 propiedades importantes.

Muestra MolLogP(a), MolWt(b) y TPSA(c) agrupadas por clase objetivo(BBB-/BBB+).

Tabla 2. Promedios de características MolWt, MolLogP y TPSA según clase.

Promedio por clase.		
	BBB+	**BBB-**
MolWt	2,877	1,454
MolLogP	340,182	464,717
TPSA	60,100	133,924

.

Ademas debido a la gran cantidad de columnas que se tienen se hace un análisis de correlación para verificar redundancia de información en el dataset.

En la Figura 6se puede que ver que las correlaciones existentes dentro de los descriptores de RDKit siguen una distribucion normal, pero aun asi hay varias columnas dentro de este grupo que estan altamente correlacionadas. Como se ve en la Figura 7.

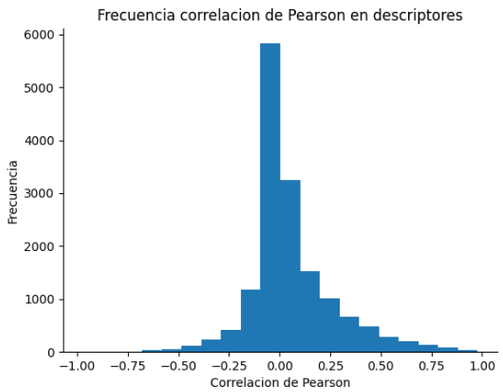
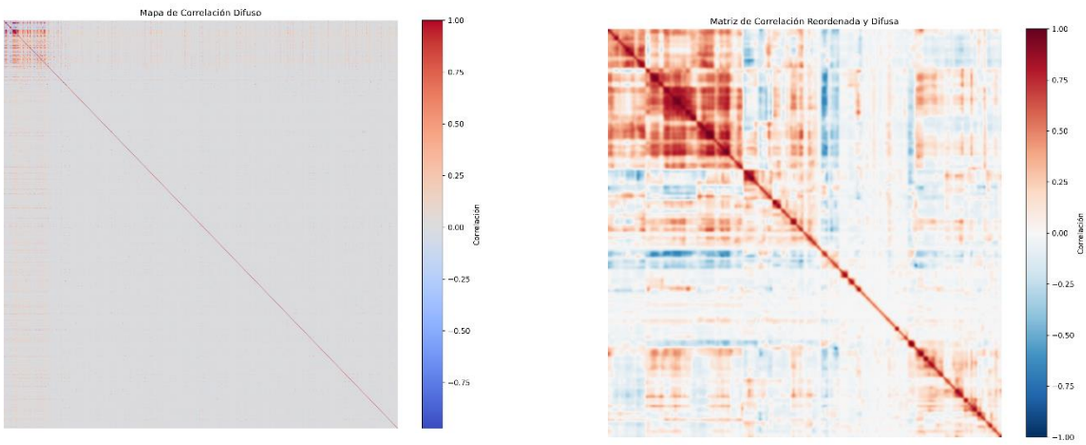


Figura 6. Distribución de correlación de Pearson de descriptores de RDKit.



(a) (b)

Figura 7. Mapas de calor de la correlación de Pearson de columnas.

Gráfico (a) corresponder a todas las columnas del conjunto de datos, el (b) solo a los descriptores de RDKIT.

Entrenamiento y evaluación.

Se entrenaron 4 modelos diferentes: KNN, SVM, Random Forest y Naive Bayes. Generando 2 grupos, un grupo al que solo se le aplicó SMOTE y otro que paso por un proceso de Selección de características antes que SMOTE. Finalizando con 8 modelos predictivos diferentes. Además, a cada uno de estos modelos se les optimizo los hiperparámetros usando la estrategia grid-search.

Se calcularon múltiples métricas de desempeño para cada uno de los modelos de predicción, los resultados fueron ordenados en tablas para su visualización. La Tabla 3 contiene los resultados de los algoritmos que solo se les aplicó el algoritmo SMOTE, la Tabla 4 contiene los resultados de los algoritmos a los que se les aplicó SMOTE con una posterior Selección de Características.

Se puede ver que como regla general todos los algoritmos tienen una mayor Sensibilidad que Especificidad. También es visible poca variación en la Exactitud de los modelos predictivos rondando los valores de 0.86 en promedio.

Se ve como 2 algoritmos lograron resultados muy cercanos en ambas tablas, los cuales son Random Forest y SVM, estos 2 algoritmos son los mejores en base a las métricas. Pero finalmente el mejor es el algoritmo Random Forest sin selección de características debido a su alto valor en AUC logrando un 0.96.

Por otro lado, el algoritmo que peor desempeño tuvo fue Gaussian Naive Bayes, que obtuvo los peores resultados en ambos grupos. Esto debido a su baja Especificidad de 0.72 y su baja Exactitud obteniendo 0.84 puntos en su versión con Selección de características.

Tabla 3. Tabla con métricas de desempeño de modelos solo con SMOTE.

Modelos solo con SMOTE						
Modelo	**Precisión**	**Sensibilidad**	**Especificidad**	**F1-score**	**AUC**	**Exactitud**
KNN	**0,90**	0,87	0,84	0,85	0,88	0,86
RF	0,89	**0,93**	0,80	0,87	**0,96**	**0,88**
SVM	**0,90**	0,92	0,83	0,87	x	**0,88**
GNB	0,84	0,87	0,72	0,80	0,84	0,82

Tabla 4. Tabla con métricas de desempeño de modelos con SMOTE y selección de características.

Modelos con SMOTE + Selección de características						
Modelo	**Precisión**	**Sensibilidad**	**Especificidad**	**F1-score**	**AUC**	**Exactitud**
KNN	**0,90**	0,89	0,83	0,86	0,89	0,87
RF	0,89	**0,93**	0,80	0,87	**0,94**	**0,88**
SVM	0,89	**0,93**	0,80	0,87	X	**0,88**
GNB	0,85	0,90	0,72	0,82	0,86	0,84

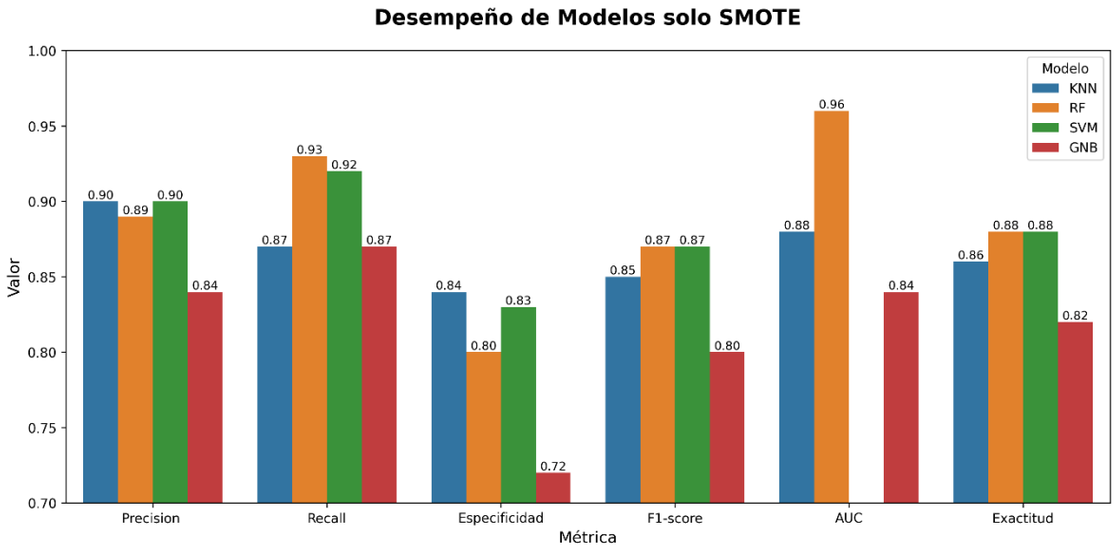


Figura 8. Gráfico de barras de métricas de desempeño modelos SMOTE.

Separado por métrica para cada uno de los modelos a los que solo se les aplicó SMOTE

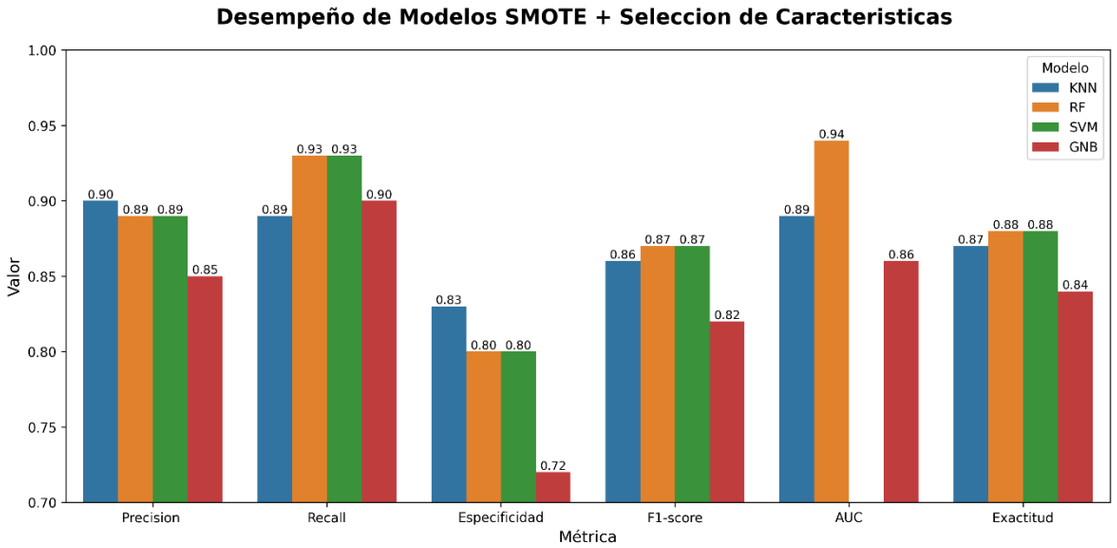


Figura 9. Gráfico de barras de métricas de desempeño modelos sel. Características.

Separado por métrica para cada uno de los modelos a los que se les aplicó SMOTE y Selección de características.

En la Figura 10 se ve la matriz de confusión del modelo Random Forest sin selección de características. Es desde aquí de donde se calculan todas sus métricas de desempeño. Se puede ver una tasa mayor de casos BBB+ acertados que para la clase BBB-.

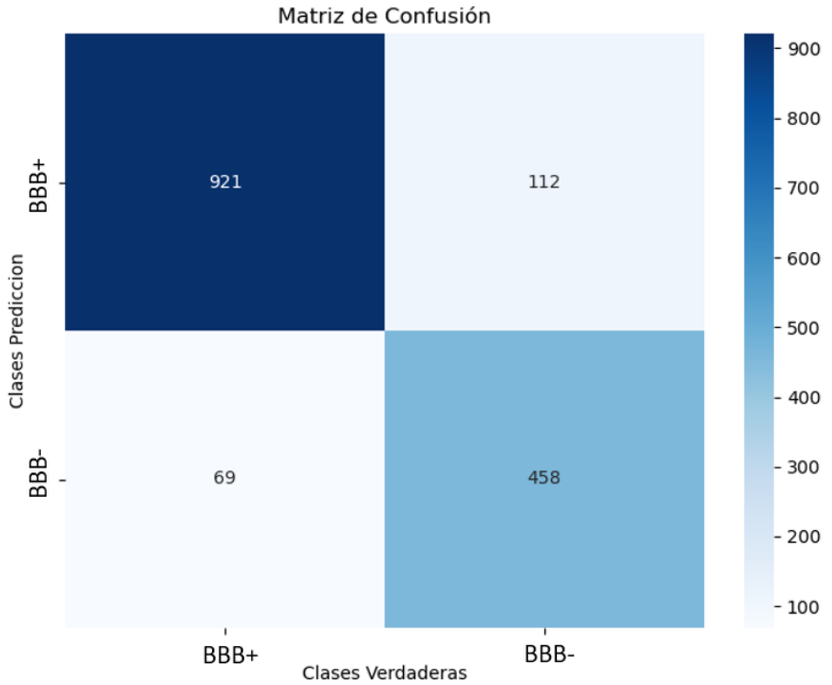


Figura 10. Matriz de confusión de modelo Random Forest sin Sel. características.

Con el fin de lograr entender cuáles son las características mas valiosas a la hora de predecir si una molécula es capaz de atravesar o no la Barrera Hematoencefálica se obtuvo la importancia de características de los modelos Random Forest que fueron entrenados.

Tanto en el modelo con Selección de características (Figura 11) como en el del modelo sin ella (Figura 12) se puede ver el mismo tipo de distribución, en donde solo algunos descriptores tienen una alta importancia y luego esta disminuye.

Podemos ver que algunas de las características más importantes se repiten en ambos modelos Random Forest. Algunas de las más importantes: TPSA, qed, NOcount, NHOHcount. No se puede apreciar ninguna de las características generadas por el fingerprint de Morgan.

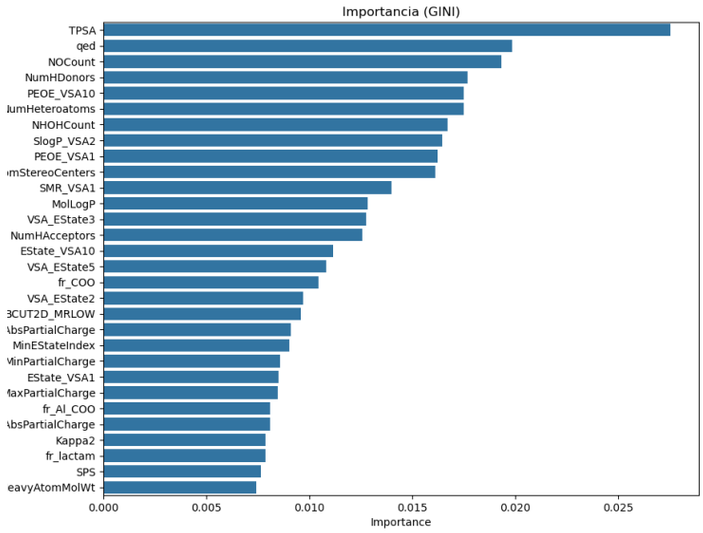


Figura 11. Importancia de características para modelo Random Forest con selección de características.

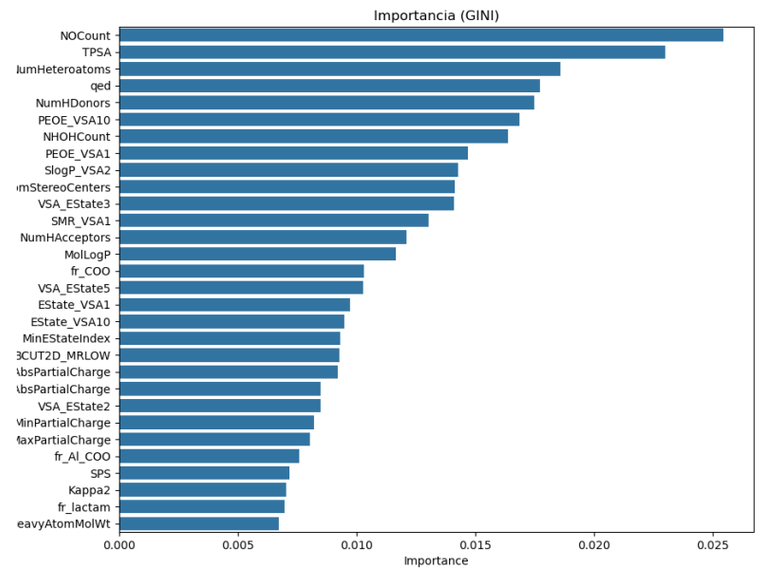


Figura 12. Importancia de características para modelo Random Forest sin selección de características.

En base a los a la Figura 11 se generó un PairPlot, Figura 13, de las 5 características más importantes, con el fin de comprender la razón de la importancia de estas.

En ninguna de estas características se puede ver una diferenciación clara entre las 2 clases (BBB-/BBB+).

Cabe recalcar que este gráfico se construyó usando el set de datos sin SMOTE, para capturar la naturaleza real de los registros.

Figura 13. PairPlot columnas importantes.

Se ven gráficos de dispersión y de distribución univariada de las 5 propiedades/características que tienen más importancia en uno de los modelos Random Forest.

Como se puede ver en la Figura 14, también se obtuvo las curvas ROC para obtener el AUC(Area Under the Curve). Se puede apreciar que los algoritmos Random Forest obtuvieron la mejor puntuación con un valor de 0.96 y el peor resultado fue el de los algoritmos Gaussian Naive Bayes con un valor de 0.84, por otro lado, KNN obtuvo valores dentro de la media con 0.90 y 0.88.

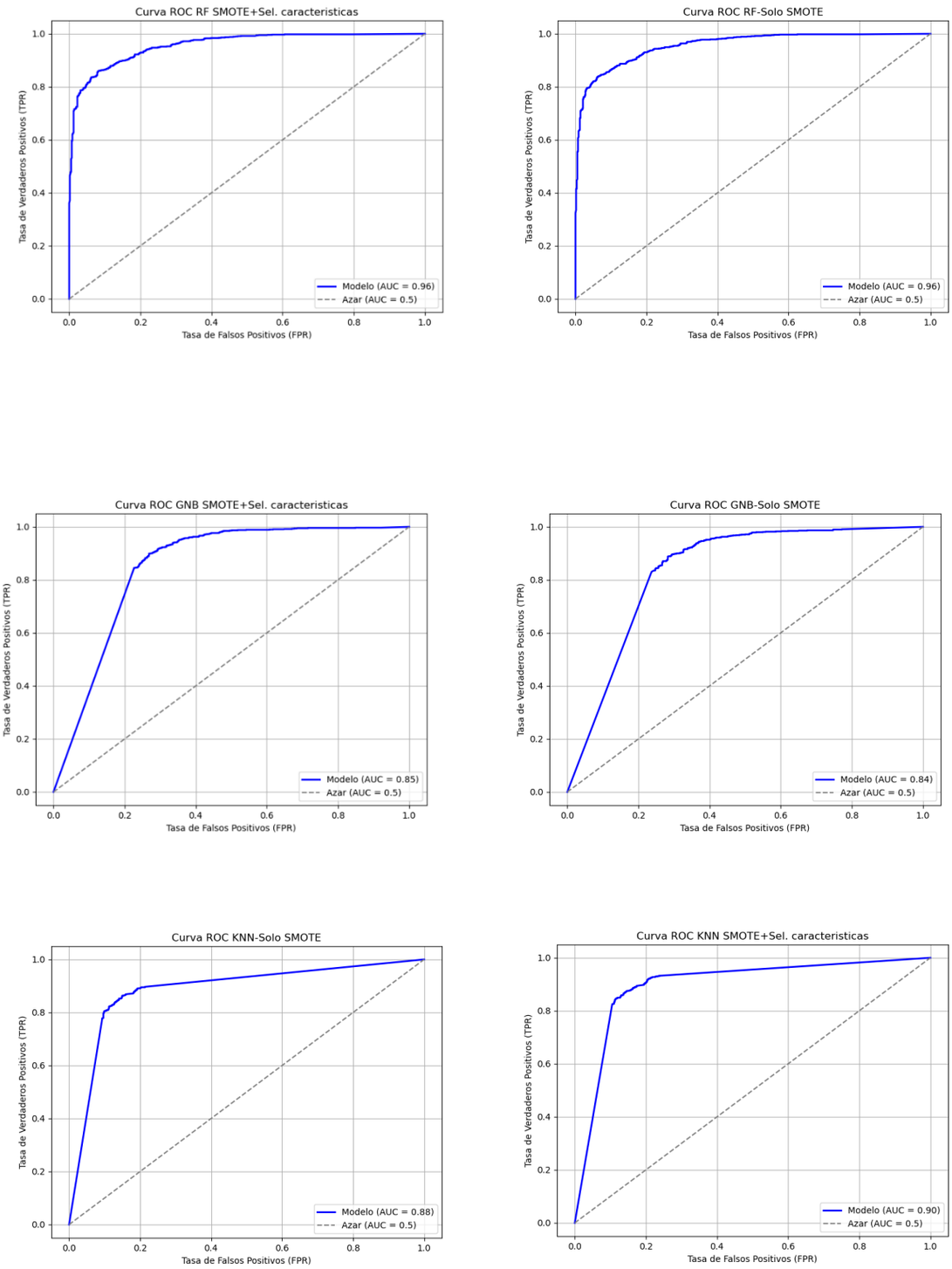


Figura 14. Curvas ROC.

Característica operativa del receptor o ROC acrónimo del inglés Reciever Operating Characteristic. Junto con el área bajo la curva (AUC de Area Under the Curve) y una curva demostrativa de un AUC igual a 0.5.

Discusión

En el presente estudio se implementaron 4 algoritmos de clasificación diferentes (KNN, SVM, Random Forest y Naive Bayes). En donde, para efectos comparativos se generaron 2 variantes de cada uno: con selección de características y otra sin ella. Este proceso de selección se implementó del modo que los resultados del primer algoritmo fueron entregados al segundo, con el fin de encontrar los mejores descriptores que facilitaran la predicción. Luego de esto, se le aplico la técnica de balanceo SMOTE al set de datos de entrenamiento para cada uno de los algoritmos.

Como se puede ver en la Tabla 3 y la Tabla 4 los resultados obtenidos por los modelos predictivos presentaron limitaciones en términos de Exactitud, ya que ninguno de estos supero el umbral de 0.9 a diferencia del modelo de (Shi et al., 2021) que obtuvo un 0.96 de Exactitud con su modelo de XGBoost . Aunque algunos de los modelos obtuvieron muy buenos resultados en otras métricas como lo es el modelo de Random Forest sin selección de características, que obtuvo 0.96 de AUC. En otras métricas algunos de los modelos obtuvieron un 0.93 en Sensibilidad y/ o 0.83 en lo cual se podría considerar bueno.

En comparación a otros trabajos que usaron el mismo set de datos, estos resultados son un poco más altos que el de (Jia & Sosso, 2024) que tiene un 0.90 y 0.84 de Sensibilidad y Especificidad respectivamente. Pero en términos generales el desempeño de los modelos no fue tan alto como lo esperado.

Se puede ver que en norma general todos los modelos desarrollados tienen una mejor Sensibilidad que Especificidad. Lo que significa que son muy capaces de predecir correctamente la clase positiva o en este caso, las moléculas que consiguen atravesar la Barrera Hematoencefálica, de hecho. Esto se puede ver en la matriz de confusión del modelo Random Forest sin selección de características en donde hay una cantidad mucho menor de casos positivos(BBB+) clasificados erróneamente en comparación a los negativos (BBB-).

En el análisis del PairPlot (Figura 13) se revela que, a pesar de ser las características con mayor importancia, ninguna de ellas demuestra una separación evidente entre las clases. Esto sugiere que el rendimiento no depende de características aisladas, sino que de la interacción combinada de dichas características.

Hablando del proceso de Selección de características, se puede ver que no causó una mejora en los modelos de Random Forest y SVM, esto debido a que ambos modelos son conocidos por ser robustos ante grandes cantidades de características. Por otro lado, se puede ver una mejora menor en el desempeño de los algoritmos KNN y Gaussian Naive Bayes, aun así, esta mejora solo es de alrededor de 0.2 puntos porcentuales lo que no se podría considerar un aumento significativo.

Con respecto a las características mas importantes, lo primero a resaltar es que ninguno de los descriptores del fingerprint de Morgan se encuentran en el top de importancia, lo que puede significar 2 cosas: este fingerprint no aporta al modelo o ninguno de las columnas generadas por el mismo aporta por si sola.

En términos de las características mas decisivas a la hora de definir la permeabilidad, como era de esperar la superficie polar total (TPSA) y la similaridad a fármacos orales existentes (qed) son bastante importantes. Pero también se puede ver otros descriptores que también obtuvieron alta importancia: como el calculo de polaridad especifica en PEOE_VSA1 (polaridad baja) y PEOE_VSA10 (polaridad alta) lo que puede significar que es importante saber exactamente que tipo de polaridad tiene una molécula y no solo su carga total. Por otro lado, una característica a destacar es AtomStereoCenters que cuenta el número de átomos que generan estereoisometría o, en otras palabras, que este descriptor se encuentre dentro del top en importancia significa que al modelo le importa la forma 3D de la molécula. En base a estos descubrimientos se puede llegar a la conclusión que el conocer estas características sobre una molécula puede orientar la búsqueda de fármacos permeables.

Conclusiones

En conclusión, se consiguió desarrollar un modelo capaz de predecir características moleculares, específicamente la permeabilidad de los fármacos o habilidad de traspasar la Barrera Hematoencefálica.

Los resultados evidencian que el modelo de Random Forest sin selección de características fue superior que los demás modelos en múltiples métricas de desempeño, alcanzando un 0.96 de AUC y un 0.93 de Sensibilidad. Lo que indica

que tiene una muy alta capacidad de discernir entre las 2 clases (BBB+ y BBB-). Es por eso que se determinó a este modelo como el más idóneo y por esa misma razón este puede ser usado como punto de partida para futuras mejoras.

Como trabajo futuro, se sugiere explorar que el desempeño de los modelos se puede mejorar utilizando otros fingerprints como lo puede ser el MACCS (Durant et al., 2002) ya sea reemplazando o agregándolos a los que se usaron en esta tesis, es interesante agregarlo ya que los mejores algoritmos fueron los más robustos ante gran dimensionalidad. También se debe probar otros métodos de selección de características diferentes a Información Mutua y RFEcv ya que estos en conjunto no lograron una mejora significativa en los resultados. Por otro lado, se pueden implementar otros algoritmos predictivos para expandir el área de búsqueda y encontrar la mejor combinación de algoritmos predictivos y selección de características. Además, todavía queda la posibilidad de expandir el set de datos incluyendo una mayor cantidad de moléculas etiquetadas, lo que puede traer consigo un mayor desempeño y una mejor generalización.

Referencias.

Abbott, N. J., Patabendige, A. A. K., Dolman, D. E. M., Yusof, S. R., & Begley, D. J. (2010). Structure and function of the blood–brain barrier. *Neurobiology of Disease*, *37*(1), 13–25. https://doi.org/10.1016/J.NBD.2009.07.030

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324/METRICS

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/JAIR.953

Cortes, C., Vapnik, V., & Saitta, L. (1995). Support-vector networks. *Machine Learning 1995 20:3*, *20*(3), 273–297. https://doi.org/10.1007/BF00994018

Cover, T. M., & Hart, P. E. (1967). Nearest Neighbor Pattern Classification. *IEEE Transactions on Information Theory*, *13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Dudani, S. A. (1976). The Distance-Weighted k-Nearest-Neighbor Rule. *IEEE*

Transactions on Systems, Man and Cybernetics, *SMC-6*(4), 325–327. https://doi.org/10.1109/TSMC.1976.5408784

Gribkoff, V. K., & Kaczmarek, L. K. (2017). The need for new approaches in CNS drug discovery: Why drugs have failed, and what can be done to improve outcomes. *Neuropharmacology*, *120*, 11–19. https://doi.org/10.1016/J.NEUROPHARM.2016.03.021

Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning*, *46*(1–3), 389–422. https://doi.org/10.1023/A:1012487302797/METRICS

Jia, H., & Sosso, G. C. (2024). Transparent Machine Learning Model to Understand Drug Permeability through the Blood-Brain Barrier. *Journal of Chemical Information and Modeling*, *64*(23), 8718–8728. https://doi.org/10.1021/acs.jcim.4c01217

Kumar, R., Sharma, A., Alexiou, A., Bilgrami, A. L., Kamal, M. A., & Ashraf, G. M. (2022). DeePred-BBB: A Blood Brain Barrier Permeability Prediction Model With Improved Accuracy. *Frontiers in Neuroscience*, *16*, 858126. https://doi.org/10.3389/FNINS.2022.858126/BIBTEX

Kumar, V., Banerjee, A., & Roy, K. (2024). Breaking the Barriers: Machine-Learning-Based c-RASAR Approach for Accurate Blood–Brain Barrier Permeability Prediction. *Journal of Chemical Information and Modeling*, *64*(10), 4298–4309. https://doi.org/10.1021/ACS.JCIM.4C00433

Martins, I. F., Teixeira, A. L., Pinheiro, L., & Falcao, A. O. (2012). A Bayesian Approach to in Silico Blood-Brain Barrier Penetration Modeling. *Journal of Chemical Information and Modeling*, *52*(6), 1686–1697. https://doi.org/10.1021/CI300124C

Mitchell, T. M. (1997). Machine learning. Tom M. Mitchell. Published by McGraw-Hill, Maidenhead, U.K., International Student Edition, 1997. ISBN: 0-07-115467-1, 414 pages. Price: U.K. £22.99, soft cover. In *Software Testing, Verification and Reliability* (Vol. 9, Issue 3). McGraw-Hill. https://doi.org/10.1002/(sici)1099-1689(199909)9:3<191::aid-stvr184>3.0.co;2-e

Organización Mundial de la Salud. (2023, August 9). *Enfermedad de Parkinson*. https://www.who.int/es/news-room/fact-sheets/detail/parkinson-disease

Organización Mundial de la Salud. (2024, August 7). *Las diez causas principales de defunción*. https://www.who.int/es/news-room/fact-sheets/detail/the-top-10-causes-of-death

Organización Mundial de la Salud. (2025, March 31). *Demencia*. https://www.who.int/es/news-room/fact-sheets/detail/dementia

Pajouhesh, H., & Lenz, G. R. (2005). Medicinal chemical properties of successful central nervous system drugs. *NeuroRX 2005 2:4*, *2*(4), 541–553. https://doi.org/10.1602/NEURORX.2.4.541

Pardridge, W. M. (2005). The blood-brain barrier: Bottleneck in brain drug development. *NeuroRX 2005 2:1*, *2*(1), 3–14. https://doi.org/10.1602/NEURORX.2.1.3

Rogers, D., & Hahn, M. (2010). Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, *50*(5), 742–754. https://doi.org/10.1021/ci100050t

Shaker, B., Yu, M. S., Song, J. S., Ahn, S., Ryu, J. Y., Oh, K. S., & Na, D. (2021). LightBBB: computational prediction model of blood–brain-barrier penetration based on LightGBM. *Bioinformatics*, *37*(8), 1135–1139. https://doi.org/10.1093/BIOINFORMATICS/BTAA918

Shi, Z., Chu, Y., Zhang, Y., Wang, Y., & Wei, D. Q. (2021). Prediction of blood-brain barrier permeability of compounds by fusing resampling strategies and extreme gradient boosting. *IEEE Access*, *9*, 9557–9566. https://doi.org/10.1109/ACCESS.2020.3047852

Squire, L. R. . (2013). *Fundamental neuroscience*. 1127.

Tang, Q., Nie, F., Zhao, Q., & Chen, W. (2022). A merged molecular representation deep learning method for blood–brain barrier permeability prediction. *Briefings in Bioinformatics*, *23*(5), 1–10. https://doi.org/10.1093/BIB/BBAC357

Yang, J., Awan, A. J., & Vall-Llosera, G. (2019). *Support Vector Machines on Noisy Intermediate Scale Quantum Computers*. https://arxiv.org/pdf/1909.11988