Universidad de Costa Rica

Facultad de Ciencias

Escuela de Química

“Predicción del Coeficiente de Distribución (log D_{pH}) *n*-octanol/agua con Modelos de Machine Learning”

Trabajo Final de Graduación presentado como requisito para optar por el grado de Licenciatura en Química.

Kenneth Geovanny López Pérez

Ciudad Universitaria Rodrigo Facio

San Pedro, Montes de Oca

2022

El presente Trabajo Final de Graduación ha sido aceptado por la Escuela de Química de la Facultad de Ciencias de la Universidad de Costa Rica, como requisito parcial para optar por el grado de Licenciatura en Química.

Juan José Araya Barrantes, Ph.D.
Director Escuela de Química

William Zamora Ramírez, Ph.D.
Director de Tesis

Mauricio Gutiérrez Arguedas, Ph.D.
Asesor de Tesis

Jorge Castro Castro, M.Sc.
Asesor de Tesis

Víctor Hugo Soto Tellini, Ph.D.
Miembro del Tribunal

Kenneth Geovanny López Pérez, B.Sc.
Postulante

Índice

Índice ... 3
Agradecimientos ... 5
Lista de Abreviaturas ... 6
Resumen ... 8
Justificación ... 9
Objetivos ... 10
1. Introducción ... 12
1.1. Machine Learning: Aprendizaje Automático ... 12
1.1.1. Aspectos generales ... 12
1.2. Modelos de predicción ... 13
1.2.1. Regresión lineal múltiple ... 13
1.2.2. Bosques aleatorios ... 14
1.2.3. Máquinas de soporte vectorial ... 16
1.2.4. Mínimos cuadrados parciales ... 20
1.2.5. Redes neuronales ... 21
1.2.6. XGBoosting ... 23
1.2.7. k-vecinos más próximos ... 25
1.3. Evaluación y validación de modelos ... 25
1.4. Coeficiente de partición ... 28
1.4.1. Aspectos generales ... 28
1.4.2. Modelos de predicción de ML ... 30
1.5. Constante de acidez ... 36
1.5.1. Aspectos generales ... 36
1.5.2. Modelos de predicción de ML ... 37
1.6. Coeficiente de distribución ... 40
1.6.1. Aspectos generales ... 40
1.6.2. Modelos de predicción de ML ... 41
2. Metodología ... 47
2.1. Coeficiente de partición neutro (log P_N) ... 47
2.1.1. Base de datos ... 47
2.1.2. Descriptores ... 50

2.1.3. Modelos 54
2.1.4. Evaluación y validación 57
2.2. Coeficiente de partición iónico (log P_I) 57
2.2.1. Base de datos 57
2.2.2. Descriptores 58
2.2.3. Modelos 59
2.2.4. Evaluación y validación 60
2.3. Constante de acidez (pKa) 61
2.3.1. Base de datos 61
2.3.2. Descriptores 63
2.3.3. Modelos 68
2.3.4. Evaluación y validación 68
2.4. Coeficiente de distribución (log D_{pH}) 69
3. Resultados y discusión 71
3.1. Coeficiente de partición neutro (log P_N) 71
3.2. Coeficiente de partición iónico (log P_I) 85
3.3. Constante de acidez (pK_a) 94
3.4. Coeficiente de distribución (log D_{pH}) 105
4. Conclusiones y recomendaciones 114
Bibliografía 117
Apéndices 130

Agradecimientos

A toda mi familia, por la formación que me dieron y que me ha permitido cumplir mis metas. Gracias por el apoyo en todo momento, por sus oraciones y por el amor que me han dado.

Lista de Abreviaturas

2D	Bidimensional
3D	Tridimensional
ADMET	Absorción- Distribución- Metabolismo- Excreción-Toxicidad
AI	Artificial Intelligence
ANN	Artificial Neural Network
DB	Drug Bank
DFT	Density Functional Theory
DL	Deep Learning
DNN	Deep Neural Network
GGA	Generalized Gradient Approximation
GPU	Graphics Processing Unit
HOMO	Highest Occupied Molecular Orbital
IUPAC	International Union of Pure and Applied Chemistry
kNN	k-Nearest Neighbors
log D	Logaritmo en base diez del coeficiente de distribución
log D_{pH}	Logaritmo en base diez del coeficiente de distribución a un pH dado
log P_I	Logaritmo en base diez del coeficiente de partición iónico
log P_N	Logaritmo en base diez del coeficiente de partición neutro
LUMO	Lowest Occupied Molecular Orbital
MAE	Mean Absolute Error
ML	Machine Learning

MLP	Multilayer Perceptron
MLR	Multiple Linear Regression
MSE	Mean Squared Error
MW	Molecular Weight
NN	Neural Network
PCA	Principal Component Analysis
pH	Negativo del logaritmo en base diez de la concentración de hidronio
pK_a	Negativo del logaritmo en base diez de la constante de equilibrio ácida
PLS	Partial Least Squares
PSA	Polar Surface Area
QM	Quantum Mechanics
QSAR	Quantitative Structure Activity Relationship
QSPR	Quantitative Structure Property Relationship
RF	Random Forest
RMSE	Root Mean Squared Error
SVM	Support Vector Machines
TPSA	Topological Polar Surface Area
XGB	Xtreme Gradient Boosting

Resumen

El coeficiente de distribución (log D_{pH}) en *n*-octanol/agua es un descriptor de la lipofilicidad de las moléculas, característica relevante en la química medicinal, toxicología y en otras áreas de la química. Este coeficiente, toma en cuenta la distribución de la molécula neutra e ionizada en cada fase; por lo que dependerá del coeficiente de partición del compuesto neutro (log P_N), del coeficiente del compuesto ionizado (log P_I) y de la constante de equilibrio ácida (pK_a). Estos tres valores se pueden utilizar para calcular el coeficiente de distribución a cualquier pH de interés.

Los modelos de *Machine Learning (ML)* se basan en el aprendizaje automático a partir de observaciones para luego realizar clasificación o predicción de otras observaciones. Son una de las herramientas *in silico* con más auge en los últimos años en la predicción de propiedades físicas y químicas, como lo es el log *D*. Existen gran cantidad de algoritmos de *ML* para la predicción del log $D_{7.4}$, por ser el pH fisiológico.

En este trabajo se utilizaron algoritmos de *ML* para predecir individualmente el coeficiente de partición del compuesto neutro (log P_N), del coeficiente del compuesto ionizado (log P_I) y de la constante de equilibrio ácida (pK_a). Las predicciones individuales de las propiedades se sometieron validaciones cruzadas y externas para elegir el mejor modelo de predicción para cada una. En el caso de la predicción de log P_I el algoritmo con mejor desempeño fue de *Random Forest* (RF); para log P_N y pK_a fue XGBoosting. Luego se integraron los tres y se obtuvieron predicciones del coeficiente de distribución (log D_{pH}) a diferentes valores de pH para un set de prueba.

Para el set de prueba se obtuvo un RMSE de 0.76 y de 0.96 para un set de validación externa unidades de log *D*. La evidencia obtenida sugiere que el desempeño del modelo propuesto es comparable y mejor en algunos casos que softwares de licencia consolidados en la predicción del log *D*.

Justificación

Un problema crítico en el desarrollo de drogas es la caracterización de las propiedades físicas que influyen directamente en la farmacocinética (PK).[1] El porcentaje de aprobación por droga que entra a fase I es menor al 10%, la principal razón de fracaso es la eficacia de la droga, que incluye razones PK's.[2] Se estima que el costo de llevar una droga al mercado ha aumentado en un 150% y un 75% de los costos corresponde a los fracasos en su desarrollo. Los modelos *in silico* son una herramienta poderosa para la reducción en gasto y tiempo en la predicción de propiedades físicas.[3] La relevancia de la lipofilicidad en ADMET (Absorción- Distribución-Metabolismo- Excreción-Toxicidad) de las drogas es conocida desde el siglo pasado.[4]

Para describir la lipofilicidad se utiliza la partición del compuesto entre n-octanol y agua. El log P_N corresponde al logaritmo de la partición de la molécula neutra. El log D_{pH}, llamado coeficiente de distribución, toma en cuenta la partición tanto de la molécula neutra como ionizada, por lo tanto, va a variar con el pH de la fase acuosa para las moléculas que tengan grupos ionizables. [5]

Existe una plétora de métodos computacionales para la predicción de log P_N y log D_{pH}; todos con ventajas y desventajas.[4] Una de las herramientas poderosas para la predicción de propiedades físicas y químicas es la Inteligencia Artificial (AI) a través de algoritmos de *Machine Learning* (ML).[1,6] Una de las desventajas principales de los métodos actuales es que predicen únicamente el coeficiente de distribución al pH fisiológico (7.4). Sin embargo, a lo largo del tracto gastrointestinal, donde se da el proceso de absorción de la mayoría de fármacos, el pH puede variar en el rango 1.7-8.0.[7] La principal motivación de este trabajo es poder predecir efectivamente el coeficiente de partición a cualquier pH.

Objetivos

Los objetivos del presente trabajo son los siguientes:

General:

Crear un modelo con algoritmos de Machine Learning para la predicción del coeficiente de distribución de moléculas neutras, monopróticas y/o monobásicas a cualquier condición de pH.

Específicos:

1. Generar descriptores estructurales, topológicos y electrostáticos para bases de datos que incluye moléculas neutras y monocargadas.
2. Elaborar modelos basados en algoritmos de Machine Learning para la predicción del coeficiente de partición de moléculas neutras (log P_N) e ionizadas (log P_I), y pK_a.
3. Integrar los modelos de predicción de log P_N, log P_I y pK_a en uno de predicción de log D_{pH}.

Capítulo 1:

Introducción

1. Introducción

1.1. Machine Learning: Aprendizaje Automático

1.1.1. Aspectos generales

Machine Learning (ML) es un campo de estudio que utiliza algoritmos computacionales para transformar datos empíricos en modelos con utilidad.[8] El aprendizaje automático (traducción de ML) es una rama de la Inteligencia Artificial (AI, por sus siglas in inglés) introducida en 1959 por Arthur Samuel. [9] Este amplio término incluye algoritmos estadísticos que tienen la capacidad de predicción o decisión basados en la inferencia de los datos disponibles sin instrucciones específicas en el código. [10] Estos algoritmos tienen la propiedad de corregir con nuevos entrenamientos errores que el programa pueda cometer y así mejorar su rendimiento. [9]

La utilización de ML para resolver problemas relacionados con la química no es algo nuevo. Entre los primeros abordajes en la química se encuentran el reconocimiento de patrones moleculares. [10] Una de las investigaciones pioneras consistió en la predicción de la estructura secundaria de proteínas a partir de información previa de secuencias de aminoácidos de proteínas cuya estructura secundaria era conocida. [11] El uso de algoritmos de ML ha continuado su legado y llegado a tener resultados excelentes en la predicción de estructuras tridimensionales de proteínas recientemente. [12,13] El reciente auge del ML inició con el cambio de siglo y ha aumentado con las mejoras en el almacenamiento de datos, incremento del poder de procesamiento (aparición de GPU's) y la continua optimización de algoritmos de ML (ejemplo *Deep Learning*). [14]

Los algoritmos de ML se pueden clasificar en dos clases principales: supervisados y no supervisados. [10] En el aprendizaje supervisado se tiene un set de datos de entrada, *input*, que cada observación tiene un vector, *X*, que está asociado a una respuesta conocida, *Y*, *output*. El proceso de aprendizaje buscar crear una función, $Y = f(X)$, que pueda ser utilizada para obtener nuevos valores a partir de nuevos vectores X' que se desconoce sus valores Y'. El aprendizaje no supervisado se utiliza cuando solo se cuenta con los datos del input y con ninguna variable de output. [6] En otras palabras los datos no cuentan con etiquetas, debido a esto es difícil evaluar el desempeño de estos modelos ya que no se cuenta con outputs con los que comparar. Los algoritmos más comunes de este tipo son los de agrupamiento y de reducción de dimensiones. [15]

Con los algoritmos de ML se pueden realizar diversos tipos de tareas que se pueden principalmente clasificar de la siguiente manera:

- Clasificación: busca asignar una categoría a cada observación.
- Regresión: busca predecir un valor o valores para cada observación.
- Ranqueo: busca ordenar las observaciones bajo un criterio.
- Agrupamiento: busca agrupar las observaciones en subgrupos más homogéneos.
- Reducción dimensional: busca transformar la representación inicial de las observaciones en una con menos dimensiones. [15]

1.2. Modelos de predicción

1.2.1. Regresión lineal múltiple

El modelo de regresión lineal es de los más simples y su comprensión es importante ya que brinda información importante para generar modelos más complejos. Este modelo busca una relación lineal entre una variable dependiente y un grupo de variables independientes. Si el modelo tiene solo una variable independiente, corresponde a un modelo de regresión lineal simple; si tiene dos o más, es un modelo de regresión lineal múltiple (MLR, por sus siglas en inglés). [9] Un modelo MLR genera predicciones $\hat{y}$, de una variable de interés y; como una función lineal de una matriz X de n, d dimensiones, $X \in \mathbb{R}^{n \times d}$. Donde n corresponde al número de observaciones y d a los descriptores. Se genera un vector w que corresponde a los pesos que se le asignan a cada descriptor y un intercepto b. El vector w se calcula de manera que se reduzca al mínimo la suma de las diferencias de cuadrados entre los valores de y y los predichos $\hat{y}$. En la ecuación 1 se muestra la relación matemática de la regresión lineal múltiple. [10]

$$y(X) = Xw + b = \begin{bmatrix} x_{1,1} & \cdots \\ \cdots & x_{n,d} \end{bmatrix} (w_1 \quad \cdots \quad w_d) + b \qquad [1]$$

La regresión lineal múltiple también se puede representar como se muestra en la ecuación 2.

$$y(x) = b + \sum_{i=1}^{d} x_i w_i \quad [2]$$

A pesar de su aparente simplicidad, los modelos de MLR pueden proveer predicciones confiables y una fácil interpretación de como los inputs afectan el output del modelo. El método de minimización más famoso que se utiliza es el de los mínimos cuadrados [16] . El coste computacional de esta operación escala clásicamente a d^3. [10] La expresión matemática se muestra en la ecuación 3.

$$LS = \sum_{i=1}^{d} (y - \hat{y})^2 \quad [3]$$

1.2.2. Bosques aleatorios

La unidad básica de los bosques aleatorios son los árboles de decisión binaria, estos consisten en modelos que parten el conjunto de datos dando resultado a subgrupos. [9] Al tener varios de árboles de decisión, se generan "ramas" que corresponden a decisiones secuenciales, el final de cada rama termina en un set de nodos llamados "hojas". En el caso de que el algoritmo se usado como regresión, las hojas representan valores numéricos y en clasificación corresponden a las etiquetas de cada una de las categorías. Este tipo de modelo es de aprendizaje supervisado, utiliza datos para genera las secuencias de árboles de decisión y posteriormente aplicarlos para generar predicciones. [10] En la Figura 1 se muestra gráficamente una secuencia de árboles de decisión para la regresión.

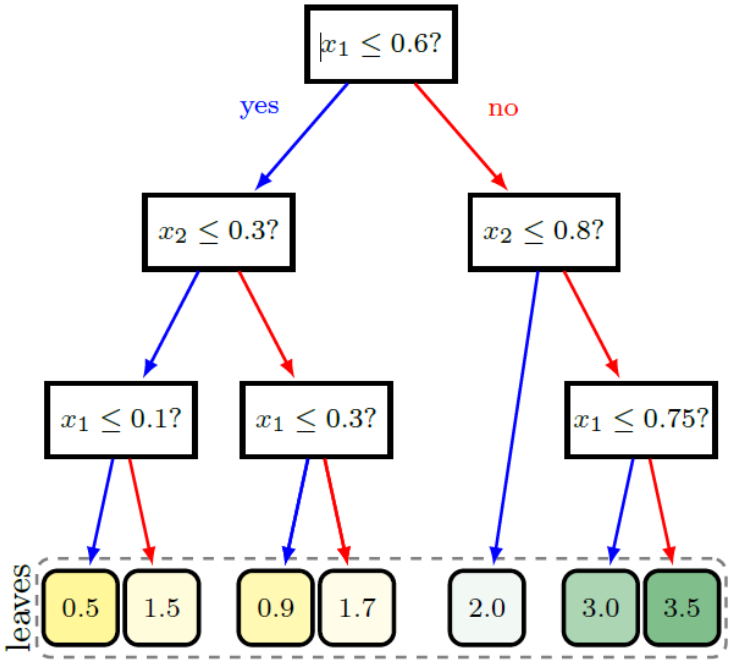


Figura 1. Representación gráfica de un modelo de regresión basado en árboles de decisión binaria. [10]

La partición se hace de manera que la desviación estándar de los subgrupos sea minimizada. El proceso se repite hasta que se alcanzan las "hojas" del modelo. En las hojas del modelo pueden existir observaciones individuales o varias en la misma hoja; cuántas observaciones se encuentren en la misma hoja corresponde al tamaño de la hoja. Un modelo puede tener las hojas suficientes para tener una observación por hojas, pero esto resultaría en un modelo complicado y probablemente en un sobreajuste. [10]

El uso de solamente un árbol usualmente resulta en un desempeño pobre y mala exactitud. Para solucionar esto se pueden entrenar múltiples árboles y agregar los resultados. El método utilizado se llama "*bagging*", en este se muestrea cierta parte del set de entrenamiento permitiendo observaciones duplicadas y omisiones. Cada uno de los *subsets* muestreados tiene el mismo tamaño y con cada uno se entrena un árbol simple, al ser un algoritmo rápido de entrenar es un perfecto candidato para utilizar el método de *bagging*. Posteriormente se promedian los resultados de cada uno de los árboles individuales y se obtiene un solo resultado. En la Figura 2 se representa gráficamente el proceso de bagging aplicado a árboles de decisión. [17]

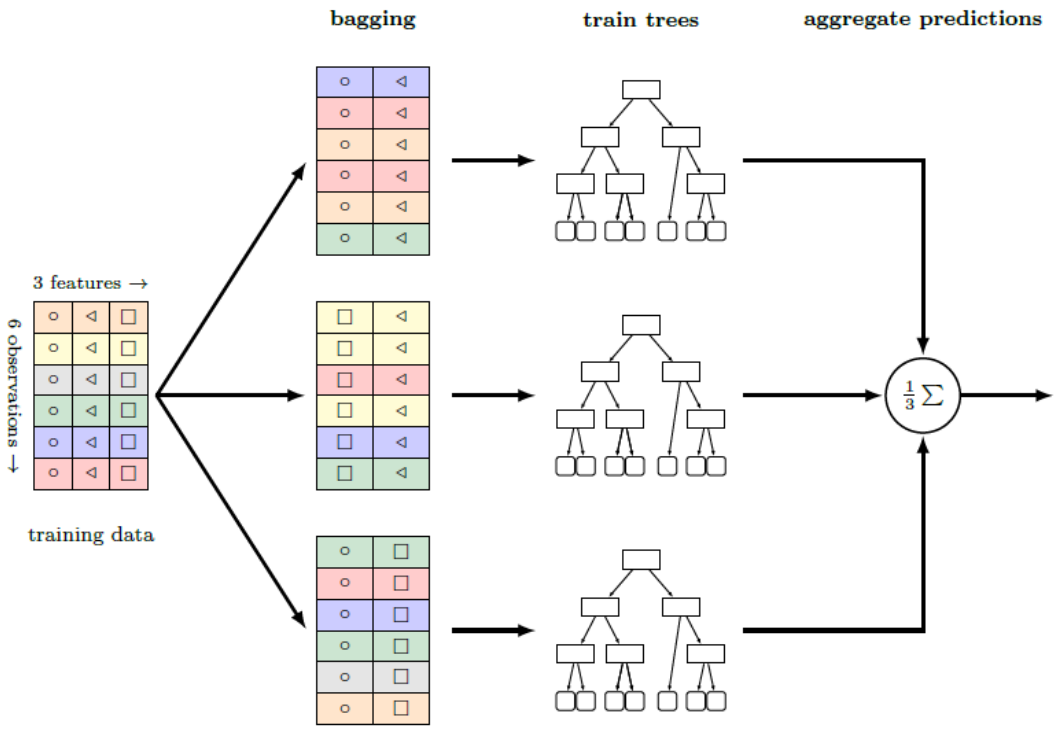


Figura 2. Representación gráfica del proceso de bagging. Se muestran los muestreos del set de entramiento y la generación de modelos a partir de cada uno de los subsets para al final agregar los resultados de los modelos en un solo promediando. [10]

Cuando se aplica la metodología de bagging existe la posibilidad que los árboles sean muy similares entre sí, especialmente si hay algún descriptor que es muy relevante para la separación de los datos ya que será elegido para iniciar la mayoría de árboles. Esto resultaría en outputs similares en cada árbol, altamente correlacionados y sin reducción de la variabilidad en los resultados finales. Para evitar esto los bosques aleatorios o *random forests* (su traducción del inglés) toman de manera aleatoria solamente una parte de las variables independientes. De esta manera se obtiene una visión parcial del problema y se reduce la correlación entre las predicciones de cada árbol. Si este procedimiento se repite dejando cada variable por fuera, se puede obtener un ordenamiento de la importancia de las variables. [9]

1.2.3. Máquinas de soporte vectorial

Las máquinas de soporte vectorial (SVM, por sus siglas en inglés) son una técnica de aprendizaje supervisado que se basa en la separación de dos grupos basado en la información suministrada. Las SVM buscan una función que separe los grupos y que maximice la distancia

entre la función y los puntos más cercanos de cada uno de los grupos. Si no existe una función lineal que pueda separar los grupos se recurre a la transformación o redimensionalización de los datos. [18] Para entender el funcionamiento de las SVM se debe iniciar con el problema más simple, la clasificación lineal. Dos grupos de datos pueden ser clasificados por infinita cantidad de hiperplanos con función lineal. Sin embargo, se busca el hiperplano que pueda maximizar el margen entre la función y los grupos. En la Figura 3, se muestra como un hiperplano cualquiera puede separar los grupos y luego el hiperplano de mayor margen. [15]

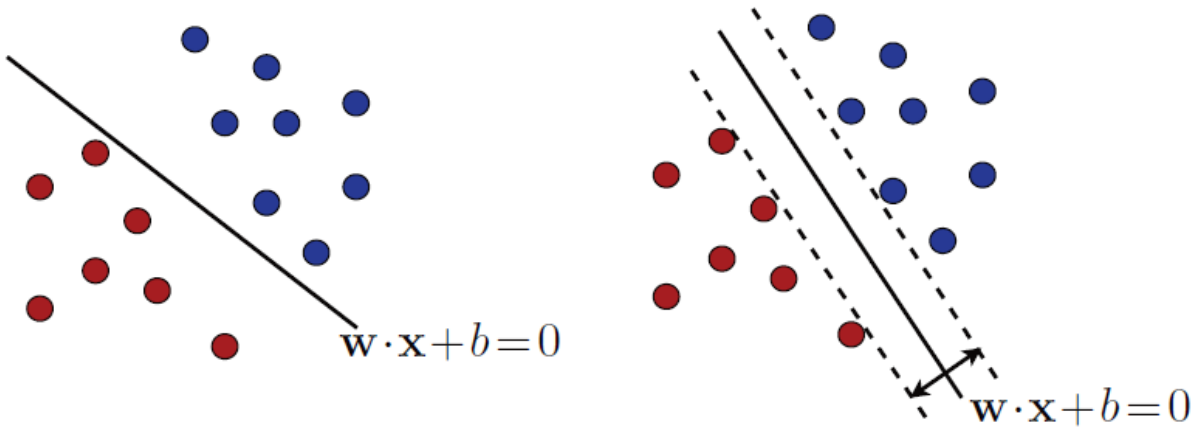


Figura 3. Clasificación de dos grupos por un hiperplano cualquiera (izquierda) y clasificación de dos grupos con un hiperplano que maximice el margen entre los grupos y la función. [15]

En el caso de hiperplanos lineales, w corresponde a un vector y b a un escalar. Como existen infinidad de planos que los puedan separar, el mejor plano es el de mayor margen, a este se le llama el hiperplano óptimo. Los puntos más cercanos equidistan del hiperplano óptimo y las rectas paralelas a este que contienen estos puntos se llaman vectores de soporte. Sus ecuaciones se muestran en la ecuación 4 [9]

$$w \cdot x + b = k \qquad \text{ó} \qquad w \cdot x + b = -k \qquad [4]$$

El valor del margen corresponde a $\frac{1}{\|w\|}$. Por lo que el problema de encontrar el hiperplano se reduce a encontrar el vector w y el escalar b que minimicen la expresión mostrada en la ecuación 5 ya que es equivalente al valor del margen. [9]

$$\frac{1}{2}\|w\|^2 \qquad [5]$$

Existen casos, que son mayoría, donde los grupos no se pueden separar perfectamente. Para esto se introduce las variables de holgura que flexibilizan la búsqueda del hiperplano óptimo. Representadas con la letra ξ, la variable de holgura mide la distancia entre los vectores del soporte y el punto que viola el margen máximo. Se dice ahora que el modelo tiene márgenes blandos. En la Figura 4 se ilustra este concepto. [15]

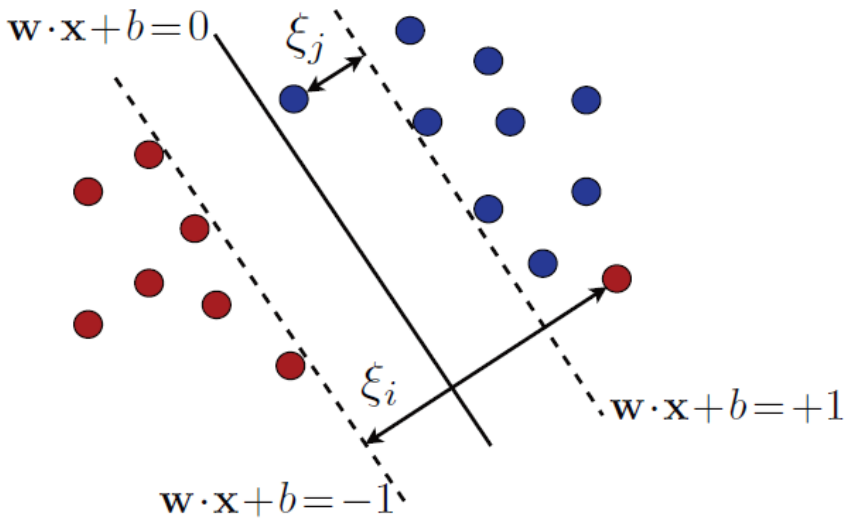


Figura 4. Representación de la clasificación con máquinas de soporte vectorial con márgenes blandos de dos grupos cada uno con una variable de holgura. [15]

Ahora la maximización del margen va a tener un costo asociado a las variables de holgura. Este costo se muestra en la ecuación 6 que corresponde a la minimización que se busca ahora, donde C es un valor mayor o igual a cero. Si se escoge un valor grande para la constante C, se consigue que la suma de las variables de holgura sea pequeña y se obtienen menos elementos no separables. Si se elige un valor bajo de C, la cantidad de valores que violen el margen será mayor. [9]

$$\frac{1}{2}\|w\|^2 + C\cdot\sum_{i=0}^{n}\xi_i \qquad [6]$$

En el caso de que los conjuntos no pueden ser separados linealmente se realizan transformaciones del espacio original a un espacio de mayor dimensión en donde sí sea posible la separación. A esta transformación se le denomina “truco de kernel”. [9] A una función se le llama kernel K cuando para un espacio X, en el cual dos puntos $x, x' \in X$; donde K(x, x’) es igual al

producto punto de los vectores resultantes al utilizar la función de transformación Φ en cada punto. La expresión general de una función kernel se muestra en la ecuación 7 y la Figura 5 ilustra lo que se quiere realizar con este método. [15]

$$K(x, x') = \Phi(x) \cdot \Phi(x') \qquad [7]$$

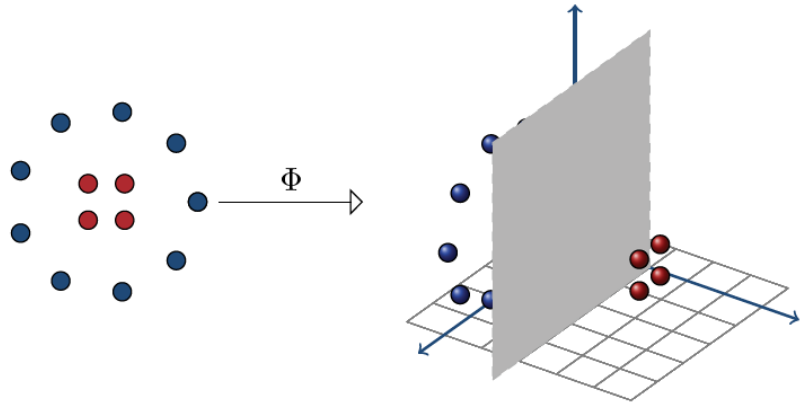


Figura 5. Representación de uso del método kernel para el cambio de dimensión de un espacio donde los grupos no son separables linealmente. [15]

Existen infinita cantidad de funciones kernel posibles, las más utilizadas son la lineal, polinomial, sigmoidea y Gaussiana. Los usos de cada una de ellas dependen de la cantidad y tipo de datos que se tengan. A manera general, se puede decir que el kernel lineal se utiliza para categorizar textos, el polinomial para clasificación de imágenes, el sigmoideo se aplica en redes neuronales y el Gaussiano cuando se cuenta con gran cantidad de información. En el Cuadro I se muestran las expresiones generales de las funciones kernel mencionadas. [9]

Cuadro I. Expresiones matemáticas generales de los principales kernels.

Kernel	$K(x, x')$
Lineal	$x \cdot x'$
Polinomial	$(x \cdot x' + 1)^d$
Sigmoideo	$\tanh(\kappa x \cdot x' - \delta)$
Gaussiano	$exp\left(\frac{-\|x - x'\|^2}{2\sigma^2}\right)$

En el caso del modo de regresión de las SVM no se busca un hiperplano para la clasificación en clases, si no un hiperplano regresor que mejor se ajuste al conjunto de datos de entrenamiento. Se considera una distancia ε, de modo que se espera que los puntos estén dentro de la banda o tubo que diste ε del hiperplano. A la hora de definir el hiperplano solo se toman en cuenta los que disten más de ε y en este caso serán los vectores de soporte. En la Figura 6 se representa como ahora se quiere englobar los puntos dentro de ε y que ahora los vectores ξ son los que se salen del margen. En términos de la función de coste, como lo es la ecuación 6, significa que un número pequeño de C dejaría que muchos puntos se salgan de la banda y un valor alto de C significaría que pocos puntos se encuentran afuera de este ($\xi \rightarrow 0$). [19]

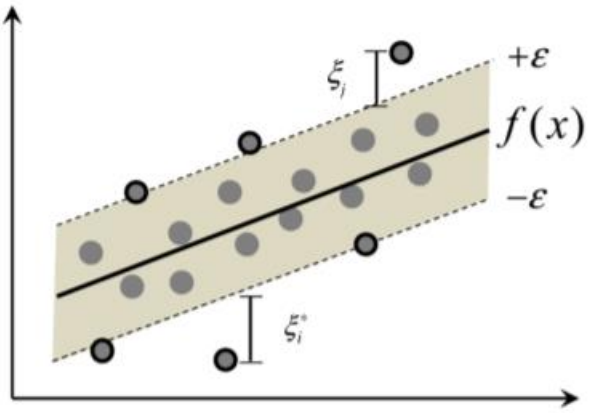


Figura 6. Representación gráfica de un SVR lineal con una distancia margen ε. [19]

1.2.4. Mínimos cuadrados parciales

El método de mínimos cuadrados parciales (PLS, por sus siglas en inglés, *Partial Least Squares*) es útil para generar modelos predictivos cuando existen muchos factores y son colineales. Este tipo de modelo se centra en la predicción de respuestas, no en la relación existente entre las variables y la respuesta. El PLS se desarrolló en la década de los 60's como una técnica econométrica, pero ha migrado a ser ampliamente utilizada en ingeniería química y quimioinformática. [20]

Esta técnica combina aspectos de la regresión múltiple y análisis de componentes principales (PCA, por sus siglas en inglés: *Principal Component Analysis*). Si se tiene una matriz *Y* que contiene una cantidad *I* de observaciones y *K* variables dependientes; y una matriz *X* que contiene *I* observaciones y *J* variables independientes. Con un PCA se explica y se encuentran los

componentes principales que explican y eliminan problemas de colinealidad en X pero esto no garantiza que estos componentes sean relevantes para Y. PLS busca los componentes, llamados vectores latentes, que simultáneamente descomponen X y Y con la restricción que de que estos componentes expliquen lo más posible la covarianza entre estas matrices. [21]

1.2.5. Redes neuronales

Las redes neuronales artificiales (*ANN, Artificial Neural Networks*) se pueden considerar como modelos matemáticos que tratan de imitar el procesamiento de información del cerebro. Los inicios de este tipo de programación parecida a cerebros se dieron en la década de los 40. Los elementos básicos de procesamiento de las redes neuronales son las neuronas. Cada neurona está caracterizada por un nivel de actividad (representando el estado de polarización de una neurona), valores de entrada (conexiones de dendritas de otras neuronas), un valor de sesgo (el estado de reposo de la neurona), y valores de salida (el axón y sus terminales). Estos aspectos son representados matemáticamente, donde cada conexión tiene un peso y de eso dependerá la activación o no de la neurona. Estos pesos pueden ser positivos (excitación de la neurona) o negativos (inhibitorios). [22] En la Figura 7 se pueden apreciar los elementos generales de una neurona.

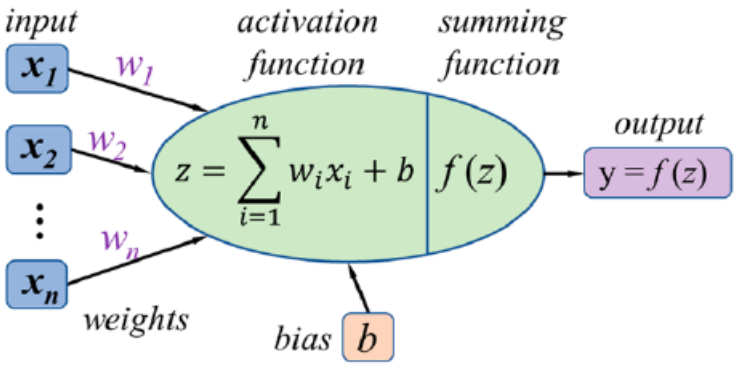


Figura 7. Esquema de los elementos generales de una neurona artificial. [6]

Las redes neuronales típicas tienen una arquitectura donde las neuronas artificiales están acomodadas en capas: una capa de input, capas ocultas y la capa de output. Las capas ocultas pueden ser solo una o las que usuario considere necesarias para el problema específico. Para entrenar las *ANNs* se pueden utilizar variedad de técnicas de aprendizaje como por ejemplo algoritmos de descenso de gradiente. La arquitectura más básica de una red neuronal es input →

ocultas → output, a este tipo de redes se les llama *feed-forward*. Uno de los ejemplos más comunes de este tipo de redes es el *Perceptron*. [6]

El MLP (*Multilayer Perceptron*) consisten en ANNs donde todas las capas, excepto la capa input, se conforman de neuronas y tienen funciones de activación no lineales. Las funciones de activación más comunes tienen forma sigmoidea e intentan asemejar la activación de una neurona biológica. Cada nodo está conectado a todos los nodos de la capa siguiente. El aprendizaje de las MLP es supervisado y se da por retropropagación. [23] La retropropagación se refiere a la distribución del error en la estructura de la red, asignando parte del error a cada nodo. Existen arquitecturas más complejas como las *Recurring Neural Networks*, *Boltzmann Machines*, *Convolutional Neural Networks* y *Message Passing Networks*. Cada una tiene una variación en la disposición, conexiones y funciones de activación; su uso dependerá del problema con el que se enfrenta. [6]

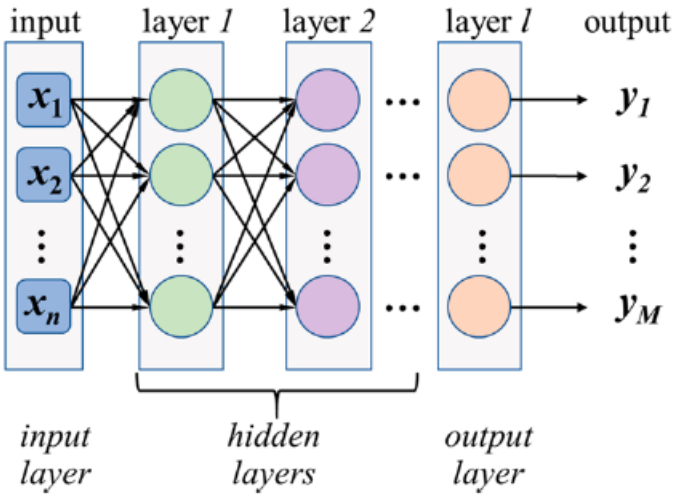
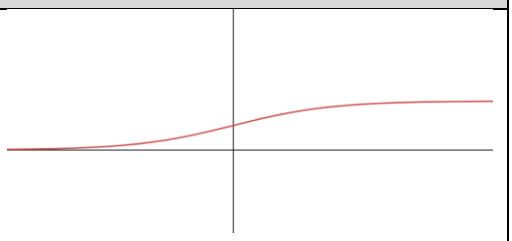
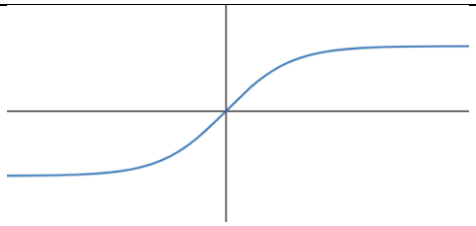
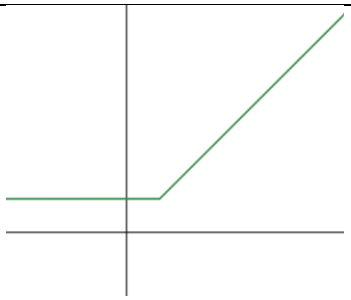


Figura 8. Esquema de una arquitectura básica de una red neuronal *feed-forward*. [6]

Las funciones de activación transforman una señal input en una señal output y esta funcionará como input de otra neurona. Estas son necesarias para definir si una señal pasa a la siguiente neurona en la red, no de existir funciones de activación las redes neuronales actuarían como un modelo de regresión lineal. Entre las funciones de activación más comunes se encuentran la logística sigmoidea, tangente hiperbólica y unidad lineal rectificada. Estas funciones tienen en común que limitan la amplitud del output y tienen un punto umbral de activación. [24] En el Cuadro II se muestran la ecuaciones y gráficas generales de estas funciones.

Otro parámetro importante de las ANNs son los optimizadores utilizados en el entrenamiento. Los optimizadores son algoritmos o métodos que determinan atributos de una red neuronal artificial como los pesos o el ritmo de aprendizaje, con el objetivo de reducir una función la pérdida. [25] Entre las funciones de optimización más comunes se encuentran algoritmos de convergencia-lineal estocástica *LBFGS* [26], descenso de gradiente estocástico *SGD* [27] y optimización basada en gradientes de primer orden *ADAM* [28].

Cuadro II. Resumen de la ecuaciones y gráficas generalizadas de las funciones de activación más comunes en las redes neuronales artificiales.

Función de activación	$f(x)$	Gráfica general
Logística sigmoidea	$\frac{1}{1+e^{-x}}$	
Tangente hiperbólica	$\tanh(x)$	
Unidad lineal rectificada	$\max(k,x)$	

1.2.6. XGBoosting

Los métodos de ensamble son aquellos que combinan varias técnicas de predicción para obtener mejores resultados. Uno de estos métodos es el *boosting*, el cual tiene como idea utilizar un algoritmo de aprendizaje débil para construir uno de aprendizaje fuerte. [15] Uno de los métodos de este tipo que ha destacado es el boosting de árboles, específicamente el XGBoost. [29] En este

método, a diferencia de los árboles de decisión, en cada árbol ocurre una regresión que asigna cierto puntaje a cada hoja. De igual manera los datos son clasificados por los árboles, solo que la predicción final corresponderá a la suma de los puntajes de cada hoja. En la Figura 9 se muestra un ejemplo animado de cómo funciona el ensamble de árboles. [30]

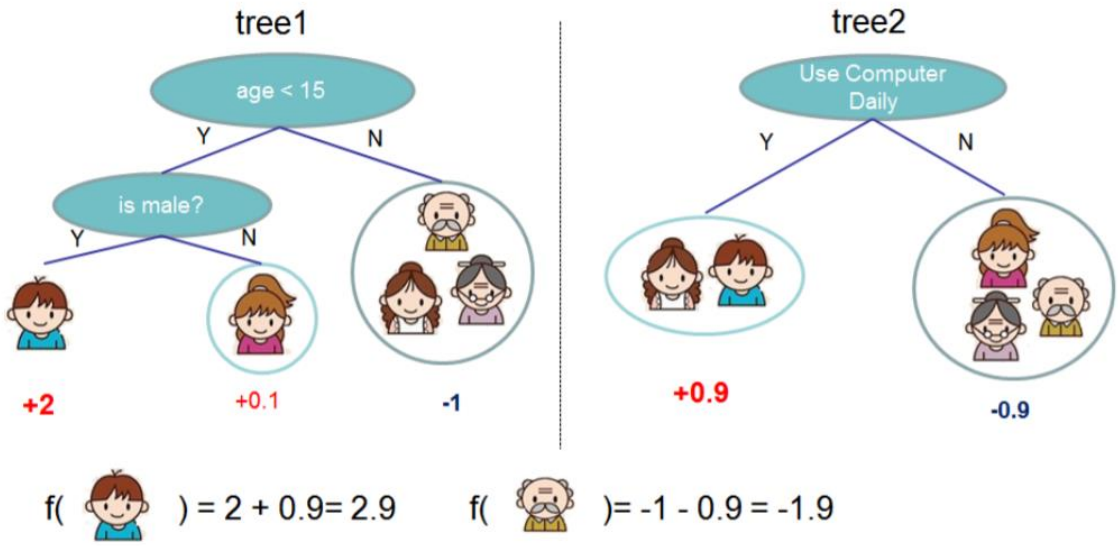


Figura 9. Representación gráfica de un ejemplo de ensamble de árboles para la regresión. [30]

El aprendizaje de cada árbol se realiza uno a la vez y se va adicionando un árbol a la vez para el entrenamiento, no es posible entrenar todos los árboles a la vez. La mayoría de los paquetes que utilizan este algoritmo incluyen un término de regularización que define la complejidad de cada árbol. El XGBoost genera un gradiente estadístico para cada una de las observaciones que se encuentra en cada hoja, se suma los de cada hoja y con esto se asigna un puntaje para evaluar que tan bueno es el árbol. En la Figura 10, se muestra cómo se realiza esto para asignar un puntaje al árbol y en la ecuación 8 cómo se integran estas funciones de puntajes. El primer término de esta ecuación se refiere al de la hoja izquierda, el segundo al de la hoja derecha, el tercero a la hoja original y el último término a la regularización. Si esta función de gane es negativa, es decir el primer término es menor que γ, se deja de adicionar ramas al árbol. [31]

$$\frac{1}{2}\left[\frac{G_L^2}{H_L+\lambda}+\frac{G_R^2}{H_R+\lambda}-\frac{(G_L+G_R)^2}{H_L+H_R+\lambda}\right]-\gamma \qquad [8]$$

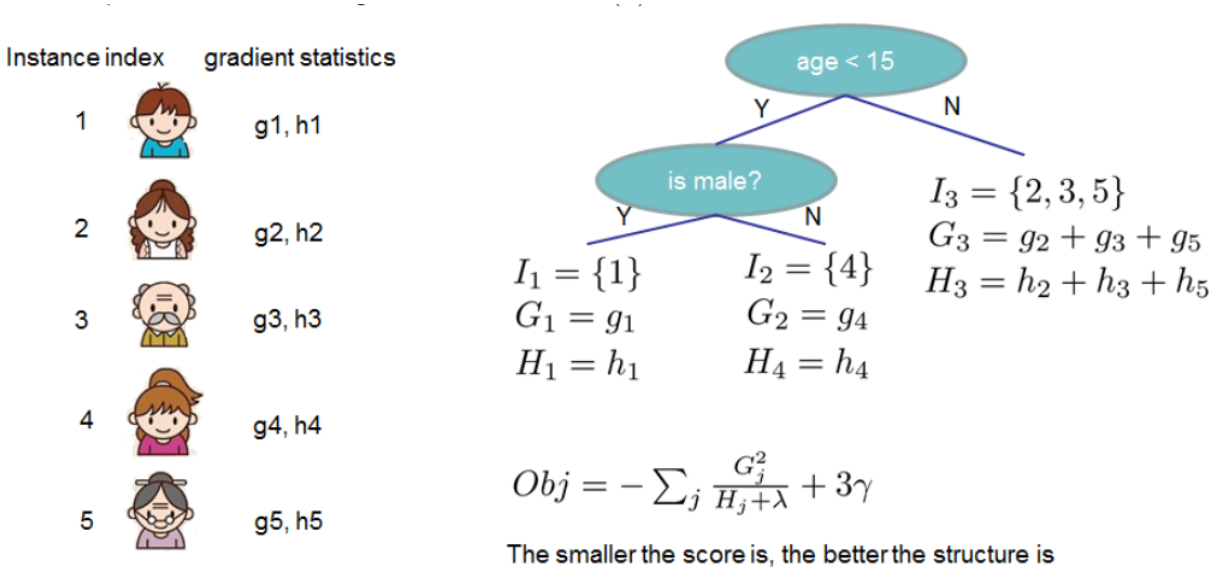


Figura 10. Representación gráfica de la función de puntaje que realiza el XGBoost para optimizar los modelos.

1.2.7. k-vecinos más próximos

El algoritmo de los *k*-vecinos más próximos es de los más simples de ML. Se basa en asumir que "cosas que se parecen deben ser parecidas". Este algoritmo mayoritariamente utilizado para clasificación, tiene como objetivo calcular las distancias entre observaciones según una función entre dos elementos en un espacio dado según las dimensiones de los vectores utilizados como input. Existen diversas funciones para calcular distancias, la más simple la Euclidiana. El algoritmo reordena las observaciones según las distancias para posteriormente obtener un output. En el caso de clasificación se tiene como output una etiqueta que defina un espacio con *k* observaciones; para regresión es un ponderado de la variable output de los *k* vecinos del punto. [32]

1.3. Evaluación y validación de modelos

En los casos donde el problema es de regresión, como el de la presente investigación, para evaluar el desempeño de los modelos se utiliza principalmente el error cuadrático medio (*MSE, Mean Squared Error*) y el error absoluto medio (*MAE, Mean Absolut Error*). Las ecuaciones de estos errores se muestran en las ecuaciones 8 y 9, respectivamente. En ambas ecuaciones, n corresponde al número de observaciones del set donde se están generando las predicciones, y_i a el valor considerado como verdadero de la variable que se quiere predecir y $\hat{y}_i$ al valor predicho por

el modelo. Un tercer error es usualmente calculado, que corresponde a la raíz cuadrada del error cuadrático medio (*RMSE, Root Mean Squared Error*), la ecuación correspondiente se muestra en la ecuación 10. [33]

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad [8]$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \quad [9]$$

$$RMSE = \sqrt{MSE} \quad [8]$$

Una manera rápida y fácil de comparar si el modelo tiene buen desempeño es al comparar la desviación estándar del set y el RMSE. Se quiere que el RMSE esté por debajo de la mitad de la desviación estándar para considerar el modelo con un desempeño aceptable. [33]

Una métrica utilizada en la evaluación de modelos de regresión es el coeficiente de determinación, conocido también como R^2. Este es informativo y veraz, ya que da cabida a mejor interpretabilidad de su resultado al tener un rango predicho; a diferencia de las métricas de errores que pueden ir desde 0 a infinito. El R^2 puede interpretarse como la proporción de la varianza en la variable dependiente que es predecible a partir de las variables independientes. Matemáticamente el coeficiente de determinación se expresa como se muestra en la ecuación 10, donde $\bar{y}$ corresponde al promedio de los valores y_i. [34]

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad [10]$$

Un método utilizado para la validación de modelos es la validación cruzada de k-iteraciones (*k-fold cross validation*). En esta el número de observaciones se reparten uniformemente en k subsets, $D_1, D_2, \ldots, D_k$, de tamaños casi iguales o iguales. Cabe destacar que las observaciones se encuentran exclusivamente en uno de los subgrupos y no se repiten. En cada iteración de la validación se usa como test set el subset D_i y el resto se utilizan para entrenar el modelo colectivamente. Con esta validación se asegura que toda observación fue parte del test set una vez. Se reporta un promedio de los errores calculados. [35] En la Figura 11 se muestra la validación cruzada de k-iteraciones de manera esquemática, en este caso $k = 5$, lo que quiere decir que la

totalidad de los datos será divida en 5 subgrupos. Cada fila representa una iteración, el suubset naranja corresponde al test set y los azules al training set.

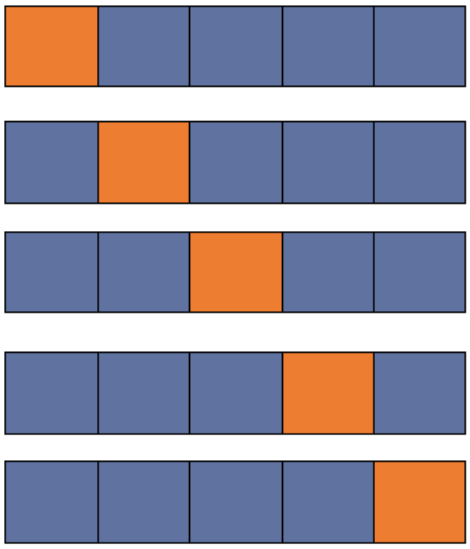


Figura 11. Representación esquemática de la validación cruzada de k-iteraciones. Se representa en naranja el test set y el azul los subsets que integran el training set en la iteración.

El uso de un set de datos independiente para evaluar el desempeño del modelo se conoce como validación externa. Este tipo de validación es importante para evidenciar que el modelo puede ser generalizado independientemente de la fuente de las observaciones y puede dar información importante sobre el dominio de aplicabilidad del modelo. [36]

Para completar la evaluación de un modelo es necesario definir el dominio de aplicabilidad. El dominio de aplicabilidad de modelos QSAR (*Quantitative Structure Activity Relationship*) o QSPR (*Quantitative Structure Property Relationship*) es el espacio físico-químico y estructural en cual se encuentro el set de entrenamiento de un modelo. Por lo tanto, ese mismo espacio es el cuál es aplicable hacer predicciones de otros compuestos. Generalmente el dominio de aplicabilidad se define con los mismos descriptores que utiliza el modelo. Compuestos fuera del dominio de aplicabilidad serían puntos de extrapolación y no interpolación. [37]

Existen diversos métodos para la evaluación del dominio de aplicabilidad. Los más simples consistentes en métodos de distancia. Por ejemplo, el método de rangos en el espacio de descriptores tiene como hipótesis que el dominio de aplicabilidad es el rango entre el valor menos

y el valor mayor de cada descriptor y de la variable a predecir. Otro método comúnmente utilizado es el de *leverage* (aprovechamiento) donde mediante una matriz de los valores de aprovechamiento, H, que proyecta ortogonalmente los vectores de los descriptores en un espacio del tamaño de las columnas, como se muestra en la ecuación 11. El espacio del dominio de aplicabilidad se define con tres desviaciones estándar y el límite se define con la ecuación 12, donde p corresponde a la cantidad de descriptores y n a la de observaciones. Los resultados de estos cálculos se resumen en un gráfico de Williams. [37]

$$H = X(X^T X)^{-1} X^T \quad [11]$$

$$h^* = 3\frac{(p+1)}{n} \quad [12]$$

1.4. Coeficiente de partición

1.4.1. Aspectos generales

La lipofilicidad de una molécula se refiere a su habilidad de disolverse en grasas, aceites, lípidos y sustancias no polares. [38] La definición de la IUPAC es: "*Lipofilicidad representa la afinidad de una molécula o una fracción de ella por un ambiente lipofílico*". [39] En modelos biológicos, la lipofilicidad, simula el evento que ocurre cuando un compuesto traspasa la membrana celular, que su por naturaleza es hidrofóbica. [38] La lipofilicidad también tiene un papel relevante en la unión de las drogas con las proteínas, ya que una alta lipofilicidad genera una fuerza motriz para que la molécula escape de la fase acuosa y se una a la proteína diana. [40] Por lo tanto, es ampliamente conocido que la lipofilicidad de la molécula juega un papel importante en la determinación de la idoneidad de moléculas candidatas a drogas. [41] Desde el siglo anterior se conoce que la lipofilicidad tiene alta relevancia en los parámetros ADMET (Absorción, Distribución, Metabolismo, Excreción y Toxicidad) y desde entonces existe exhaustiva literatura sobre cuál es la lipofilicidad idónea de las drogas. [42]

Ha sido ampliamente reportado que una alta lipofilicidad aumenta la posibilidad que la molécula sea promiscua, es decir no sea la unión con la proteína diana no sea específica. Mientras que una baja lipofilicidad generalmente exhibe malas propiedades ADMET. [38] La lipofilicidad tiene una correlación negativa con la solubilidad, la cual es necesaria para la biodisponibilidad; esto deja un pequeño rango en el cual la lipofilicidad es adecuada. [40] Para manejar este problema

se han mediciones experimentales de la razón de concentraciones de dos disolventes inmiscibles para cuantificar la lipofilicidad. [43]

Un coeficiente de partición se define como la razón de la concentración de un compuesto entre dos medios cuando llega al equilibrio. Este coeficiente se utiliza en escala logarítmica por conveniencia. [44] A pesar de la existencia de otras alternativas para la descripción de la lipofilicidad como cromatografía de liposomas inmovilizados, partición liposomas/agua o membranas artificiales inmovilizadas; el coeficiente de partición (*P*) en un sistema bifásico de un disolvente orgánico y agua sigue siendo el más utilizado para describir la lipofilicidad de un compuesto. La fase orgánica representa el ambiente lipofílico y el agua el hidrofílico. [45]

Existen diversos disolventes orgánicos que han sido utilizados para determinar el coeficiente de partición, por ejemplo: di-*n*-butiléter, cloroformo, alcanos, tolueno y *n*-octanol. [46] Para distinguir la partición de moléculas neutras y moléculas ionizadas, se tiene el coeficiente de partición neutro (log P_N) y el coeficiente de partición iónico (log P_I). El log P_N no depende si la molécula corresponde a un ácido o una base, las siguientes ecuaciones representan el equilibrio de un ácido y la ecuación del coeficiente de partición respectivo. [7]

$$HA_{(agua)} \leftrightharpoons HA_{(n-octanol)} \qquad [13]$$

$$\log P_N = \log\left(\frac{[HA]_{n-octanol}}{[HA]_{agua}}\right) \qquad [14]$$

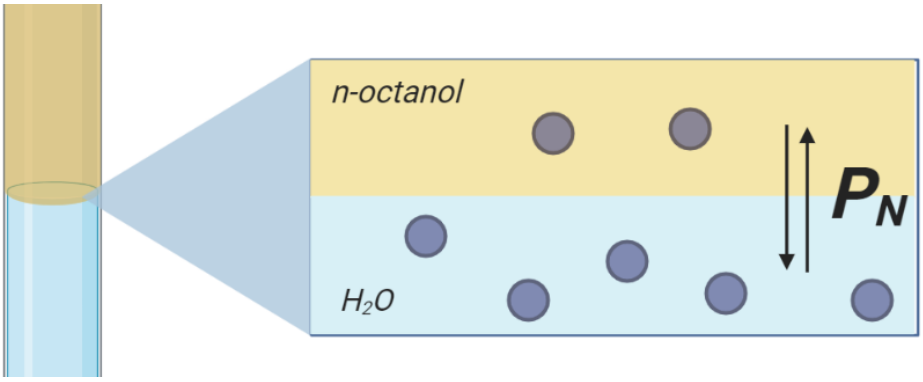


Figura 12. Representación gráfica de la partición *n*-octanol/agua.

El coeficiente de partición iónico involucra las especies ionizadas, en el caso de los ácidos los aniones (base conjugada) y en el caso de las bases los cationes (ácido conjugado). [46] A diferencia del log P_N, el log P_I no ha sido ampliamente estudiado, principalmente en el caso de los aniones. [47] En las ecuaciones 15 y 16 se muestran las ecuaciones generales para el coeficiente de partición de un ácido y una base conjugados, respectivamente.

$$\log P_I = \log\left(\frac{[A^-]_{n-octanol}}{[A^-]_{agua}}\right) \quad [15]$$

$$\log P_I = \log\left(\frac{[BH^+]_{n-octanol}}{[BH^+]_{agua}}\right) \quad [16]$$

1.4.2. Modelos de predicción de ML

De los métodos más simples de para la regresión del tipo en aprendizaje supervisado se encuentran los modelos de regresión lineal múltiple (MLR). [6] Existen abordajes diferentes de los descriptores utilizados para este tipo de modelos. Algunos se basan en aspectos estructurales, como *Molecular Fingerprints* basados en parametrización binaria en la presencia de algún fragmento de la molécula. [48] Los *Fingerprints* se generan a partir de diferentes SMARTS, este es un lenguaje para describir los patrones de una molécula; se puede considerar que es una extensión de las reglas de SMILES. [49] Un total de 8047 bits binarios se generaron para cada molécula, pero se eliminan los que están correlacionados o que tienen poca ocurrencia para evitar errores en los modelos. Luego de esto para el log P_N quedan en total 1681. Con bits estos se obtuvieron un RMSE de 0.607, pero si se reducían aún más a 600 bits binarios se mejoraba un poco el RMSE a 0.569 para el test set. [48]

En otras investigaciones se ha optado por utilizar descriptores relacionados con el área de la molécula. Por ejemplo: el área polar superficial, área polar superficial excluyendo ciertos átomos, área hidrofóbica, área más hidrofóbica, área hidrofóbica promedio, entre otras. Los autores consideran que estos descriptores son apropiados ya que describen bien la manera en que la molécula interaccionará con el medio. Estas han obtenido buenos resultados en la predicción de log P_N incluso con un training set de 147 moléculas y un modelo tan simple como el MLR se obtuvieron RMSE de 0.4836. Este modelo fue entrenado con pocos compuestos simples que fueran

anillos aromáticos, por lo que el dominio de aplicabilidad va a estar restringido a compuestos similares. [50]

Otras estrategias novedosas de MLR corresponden a la generación de hologramas de las moléculas basados en la estructura. Estos incluyen la carga, conteo del tipo de átomos, clasificadores del tipo de átomo y número de átomos enlazados; pero no información sobre la conectividad entre estos. Se realiza un pretratamiento para eliminar las moléculas con tipos de átomos que eran escasos, menos de tres apariciones. También el modelo se aplica para moléculas que no caen dentro de la aplicabilidad del modelo haciendo restas y sumas de la molécula más parecida que sí esté en el dominio. Para el entrenamiento del modelo en este caso no se utilizaron valores experimentales de log P_N, si no que se hizo un promedio entre cuatro log P_N computados: AlogP, XlogP2, SlogP y XlogP3. Este abordaje fue reportado bajo el nombre de JPlogP y se hizo bajo la lógica de que promediar valores de otros modelos puede hacer una compensación de los errores singulares en cada modelo. El modelo se compara con varios modelos conocidos, incluidos con los que se entrenó. Este presentó el menor RMSE entre los modelos con otras estrategias comparables. [51]

Las NN han sido utilizadas desde hace casi dos décadas para la predicción de la lipofilicidad. Uno de los log P_N computados más conocidos, AlogP, en su versión 2.1 fue desarrollado bajo algoritmos de este tipo. Como descriptores utilizaron conteo de átomos de hidrógeno, tipos de hidrógenos según el átomo que esté enlazado y 73 descriptores electrotopológicos. [52] El tipo de red neuronal del modelo fue asociativa, una mezcla entre *k-nearest neighbors* y una red neuronal. [53]

Mas recientemente para un reto a ciegas de predicción de propiedades físicas, SAMPL6, se utilizó una red neuronal profunda. Esta red se entrenó con 12000 moléculas usando como input una huella dactilar de cada molécula que incluía para cada átomo: número de átomos pesados enlazados, valencia de los átomos el número de hidrógeno enlazados, número atómico, carga, masa, hidrógenos enlazados y si el átomo está en un anillo. Esto se generó a partir de un SMILES y resulto un número entero de 32 bits para cada átomo. Por lo tanto, para cada molécula se tiene un arreglo de arreglos según los átomos están enlazados. El modelo de NN se entrenó con 12000 moléculas y se probó con 2000. [54]

Se reportaron dos redes neuronales con diferentes arquitecturas, la primera tiene un total de cinco *hidden layers*: tres capas de 512 unidades y dos capas de 256 unidades. La segunda arquitectura solo tenía tres capas: de 512, 256 y 128 capas respectivamente. Ambas arquitecturas tienen solo una capa como *output*. El modelo de cinco capas obtuvo mejores resultados con un RMSE de 0.62, mientras que el de tres capas fue de 0.85 para las moléculas del SAMPL6. De igual manera para el test set el mejor rendimiento lo dio el de 5 capas. Debido a la gran cantidad de datos, los autores reportan que el ambiente químico descrito es bastante amplio. [54]

Otro novedoso modelo desarrollado para el SAMPL6 mezcla mecánica cuántica con *Machine Learning*. Entre los descriptores utilizados se encuentran momentos dipolares, superdeslocalizabilidad electrofílica, dureza de los puentes de hidrógeno, energías de ionización, afinidad electrónica, nucleofilicidad, polarizabilidad, área de van der Waals, volumen de van der Waals y la superdeslocalizabilidad del HOMO y LUMO. Cada uno de estos descriptores calculados en agua y octanol. En este caso se utilizaron los algoritmos de MLR y mínimos cuadrados parciales (PLS). [55]

Se utilizaron 97 moléculas como training set. El modelo con los mejores resultados fue de PLS con un total de 74 descriptores. Sin embargo, estos modelos no fueron tan buenos; el menor RMSE fue de 0.87. Los modelos se compararon con cálculos de estructura electrónica de la molécula con DFT y GGA hechos por los mismos autores. Ninguno de estos modelos tubo mejores resultados que el mejor de ML. [55]

En algunas investigaciones los modelos con algoritmos SVM han dado resultados prometedores al nivel de las NN. Un estudio comparativo de SVM, NN y MLR con 3516 moléculas en el set de entrenamiento y 428 en el de evaluación; se obtuvieron errores menores para el caso del SVM. Los descriptores de este modelo fueron características de superficie y volumen de la molécula: relaciones de área hidrofílica/volumen de la molécula, volumen de la molécula accesible al agua, etc. [56]

En una investigación se evaluaron en conjunto de modelos para predecir varias propiedades físicas, entre ellas el log P_N, se compararon los algoritmos SVM, MLR, PLS y Random Forest (RF). El algoritmo que presento mejores resultados fue el SVM, con RMSE de 0.451. Las moléculas del modelo incluían productos químicos industriales, medicamentos, fertilizantes, fragancias, aditivos de comida, petroquímicos y pesticidas. Las predicciones de esta gama de

moléculas fueron hechas con códigos binarios de la presencia o ausencia del fragmento en la estructura de la molécula, se hicieron modelos con códigos de 1681 y 600 dígitos. El segundo mejor algoritmo fue el RF. [48]

Recientemente se reportó un modelo que utiliza una arquitectura de redes neuronales llamada SAMPN (*self attention based message-passing neural network*). Como *input* del modelo se utilizan grafos de la molécula generados a partir del SMILES, los nodos representan los átomos y las aristas los enlaces. Cada nodo contiene: tipo de átomo, número de átomos pesados enlazados, carga, quiralidad, hibridación y aromaticidad. Las aristas contienen información de: tipo de enlace, si pertenece a un anillo y estereoquímica. En este tipo de algoritmo se pasa la información de los átomos entre ellos de manera que reciben la información de los átomos vecinos. Cuanto se transmite la información a través de todos los nodos, se crea un vector con toda la información y este es el que entra en la NN. Como mejor resultado se obtuvo un RMSE de 0.571, con el SAMPN. Si no se utiliza el self-attention (SA) antes de la NN, los valores de RMSE son mayores. [57]

El SA permite que se conozca la relación entre los fragmentos/átomos y no se un modelo aditivo simplemente. El flujo de la información entre los nodos y aristas se puede ver resumido en la Figura 13. Como se visualiza en la Figura 14 las contribuciones de cada átomo a la lipofilicidad no están aisladas si no que están distribuidas entre los átomos. Con el mismo abordaje también se hizo un algoritmo de RF; sin embargo, los resultados fueron malos en comparación con el SAMPN. [57]

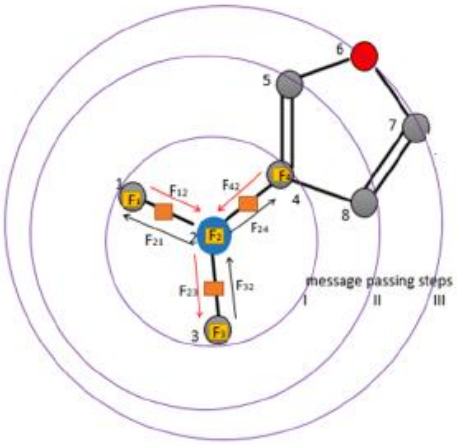


Figura 13. Esquema del flujo de información del modelo SAMPN para la generación del vector con la información de los vecinos y del átomo. Se muestra el ejemplo para el átomo 2. [57]

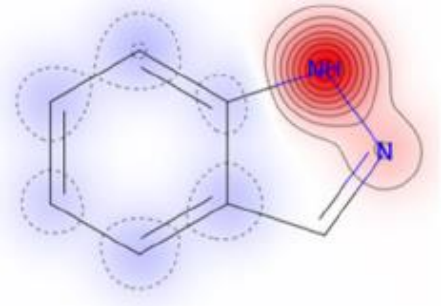


Figura 14. Esquema de contribución a la lipofilicidad de los fragmentos de la molécula. En azul se colorea las partes de la molécula que aumentan la lipofilicidad y en rojo los que la reducen. [57]

En otro reto a ciegas, aún más reciente, varios modelos empíricos destacaron en la predicción del coeficiente de partición de un set de *N-acilsulfonamidas*. En este reto, SAMPL7, entre las sumisiones ranqueadas destacó un método de regresión lineal múltiple que se basó en el conteo de grupos funcionales y en ciertas propiedades importantes que se pueden correlacionar con el coeficiente de partición (Donadores/aceptores de puentes de hidrógeno, área polar superficial, número de anillos aromáticos, etc.). Este método destacó con un RMSE de 0.58 unidades y un MAE de 0.41; siendo el mejor según los lineamientos del concurso. Una de las claves de este modelo, con nombre *TFE-MLR*, fue el tener un set de entrenamiento con un espacio químico muy similar al de las moléculas del reto. A pesar de tener un dominio de aplicabilidad muy restringido, este método da como evidencia que puede que algoritmos muy complejos no sean necesarios para la predicción del log P_N, sino la clave está en tener un set de entrenamiento altamente similar a los objetivos. [58]

El segundo método con mejor desempeño consistió de *Chemprop* [59] el cuál utiliza el algoritmo de Message Passing Neural Networks (MPNN) creado por un grupo de investigación del MIT. Este algoritmo ha sido utilizado en la predicción de diversas propiedades, factibilidad de funcionar como antibiótico e inhibición de SARS-Cov. El set de entrenamiento utilizado para *Chemprop* fue una versión procesada del set de log *P* de OPERA [60] según el coeficiente de similitud de Tanimoto (> 0.25) con las moléculas del SAMPL7. El RMSE de este modelo fue de 0.66 y el MAE de 0.48. [61]

Otro método destacado en el SAMPL7 también fue empírico, *ClassicalGSG*. Este método utiliza como inputs atributos atómicos que se asignan según campos de fuerza clásicos de mecánica molecular. Estos inputs entrenan redes neuronales, con un set de datos de 41 000 valores de log *P*.

Las redes neuronales utilizadas en *ClassicalGSG* fueron de MLP. El RMSE fue de 0.77 y el MAE de 0.62 para la submisión ranqueada. [62]

Fuera de retos a ciegas, recientemente se reportó un método que utiliza representaciones moleculares de 300 dimensiones para cada subestructura presente en la molécula y luego los agrega en un solo vector. Esta representación de secuencias o combinaciones de estos vectores se reportó bajo el nombre de *Mol2Vec*. Estos vectores se utilizaron para entrenar diversas redes neuronales de Deep Learning como MLP, Conv1D, LSTM y ensamblados de estos métodos. El mejor algoritmo de predicción de log P fue un ensamblado de MLP y Conv1D, con un RMSE de 0.589 y MAE de 0.441. [63]

Otro abordaje novedoso y reciente en la predicción de log *P* es el uso de aumentado de datos y redes neuronales profundas (*DNN, Deep Neural Network*). El aumentado de los datos se realizó a partir del SMILES original, canónico y con hidrógenos explícitos; los grafos para entrenar las DNN fueron generados con todos estos SMILES para todos los posibles tautómeros de las moléculas. Para determinar el log *P* del compuesto se hace una ponderación según la fracción de cada tautómero, $\log P_{\text{exp}} = \log \sum P_{\text{i}} * f_{\text{i}}$. El modelo se creó con 12 500 datos de log *P* experimentales, se dividió este set proporción 80/20 para el training set y test set respectivamente. Se creó un modelo sin tomar en cuenta los tautómeros y otro en como se describió anteriormente se les toma en cuenta. La arquitectura de las DNN fue de dos capas ocultas, con 64 y 128 neuronas. [64]

Según lo indagado no hay un modelo de ML que se centre en la predicción exclusiva de del coeficiente de partición iónico, los modelos se centran en la predicción del coeficiente de partición neutro. Lo que existe al momento es la predicción de coeficientes de partición iónicos dentro de modelos que en realidad fueron entrenados con el propósito de predecir el log P_N. Por ejemplo, modelos reportados consideran que datos de compuestos con valores de log P_N constituyen valores de log P_I, pero de igual manera se utilizaron para construir el modelo mezclados con datos de log P_N. [64] Otros modelos de predicción de log P_N tienen compuestos iónicos en sus sets de entrenamiento, como aminas cuaternarias, cuyos valores en realidad constituirían mediciones de log P_I. [48]

En una revisión sobre determinación de partición *n*-octanol-agua de surfactantes, se encontró que modelos QSPR tienen varias limitaciones en su predicción. Entre las limitaciones

encontradas para los modelos computacionales fueron: la falta de aminas cuaternarias, ausencia de contraiones, uso de solo SMILES neutros y correcciones de substracción de los contraiones. Los resultados arrojan que los modelos predicen eficientemente la partición de surfactantes neutros y la gran mayoría tiene errores altos para la predicción de aniónicos, catiónicos y anfóteros; principalmente para los últimos. [65]

1.5. Constante de acidez

1.5.1. Aspectos generales

Según la definición de Arrhenius los ácidos corresponden a sustancias que se disocian en agua para formar iones hidronio y las bases a sustancias que se disocian en agua para formar iones hidróxido. La definición de Brønsted-Lowry dice que los ácidos son especies que pueden donar un protón y una base la cual pueda aceptar un protón. La fuerza que tendrá un ácido estará dad por su grado de ionización, el cual depende de la constante de disociación ácida. [66]

En la ecuación 17 se muestra la ecuación general para la disociación de un ácido en agua. HA representa el ácido en su forma neutral y A^- en su forma desprotonada, lo que correspondería a su base conjugada. Cuanto mayor sea la constante de disociación ácida, más se disociará. La expresión general de la constante de acidez se muestra en la ecuación 18. [66]

$$HA + H_2O \overset{K_a}{\rightleftharpoons} H_3O^+ + A^- \qquad [17]$$

$$K_a = \frac{[H_3O^+][A^-]}{[HA]} \qquad [18]$$

Como los valores de la constante de acidez y de la concentración de iones hidronio abarcan rangos tan amplios se suelen expresar en pK_a y pH, respectivamente. Estos valores corresponden al menos logaritmo en base 10 de la K_a y [H_3O^+], en las ecuaciones 19 y 20 se muestran estas definiciones matemáticas. [66]

$$\mathrm{p}K_a = -\log K_a \qquad [19]$$

$$\mathrm{pH} = -\log\,[H_3O^+] \qquad [20]$$

La fuerza de una base se puede también expresar en términos de la constante de acidez de su ácido conjugado. Como se muestra en la ecuación 21, la base protonada (ácido conjugado), BH^+, se disocia en la base, B. A pesar de que también se podría expresar en términos de la constante de disociación básica, se suele utilizar la pK_a para comparar tanto ácidos y bases.

$$BH^+ + H_2O \overset{K_a}{\rightleftharpoons} B + H_3O^+ \quad [21]$$

$$K_a = \frac{[B][H_3O^+]}{[BH^+]} \quad [22]$$

La influencia de la pK_a en propiedades biofarmacéuticas de las drogas y compuestos es ampliamente conocida en la industria farmacéutica. En los sistemas fisiológicos es de importancia ya que el estado de ionización va a afectar la tasa de difusión a través de membranas celulares, la solubilidad, permeabilidad, farmacocinética y ADME. El principal rango de pK_a para drogas es de 2-12 unidades. [67]

1.5.2. Modelos de predicción de ML

En 2003 se propuso un modelo de PLS que utilizó *fingerprints* basado en conectividad, tipos de átomos y grupos funcionales que puedan afectar la ionización según su posición. El máximo número de elementos del cada *fingerprint* es de 165 para hasta cinco niveles de conectividad desde el sitio de ionización. El set de entrenamiento para este modelo contenía 625 ácidos y 412 bases. Los autores muestran los resultados separando los ácidos y las bases; en el caso de los ácidos al comparar los valores experimentales con los predichos se obtuvo un R^2 de 0.98 y de 0.99 para las bases. Los errores del modelo no son reportados. [68]

En 2006 se reportó un modelo para la predicción de la pK_a específicamente de ácidos carboxílicos alifáticos y alcoholes. El algoritmo utilizado fue la regresión lineal múltiple. Los tipos de descriptores utilizados fueron representaciones de moléculas con esferas topológicas, PETRA (Descriptores relacionados con la reactividad como distribución de carga, polarizabilidad, resonancia y otras propiedades físico-químicas), inducción de cargas de átomos conectados, estéricos y E-State (descriptor relacionado con el número cuántico principal, electrones de valencia, electrones sigma y la distancia topológica entre los átomos). En esta investigación se utilizó una red neuronal para seleccionar el set de entrenamiento y el set de prueba. Se contó con

1122 ácidos carboxílicos alifático como data set y con 288 para los alcoholes. En el caso de los ácidos se obtuvo un R^2 de 0.810 en una validación cruzada de k = 5, para los alcoholes 0.805. El error estándar de predicción fue mayor en el caso de los alcoholes. [69]

Un modelo que utiliza propiedades semiempíricas de química cuántica y descriptores basados en información molecular fue propuesto para predecir la pK_a de compuestos parecidos a drogas. Los autores utilizaron PLS como algoritmo y realizaron las validaciones cruzadas con k = 7. Como set de datos de partió de aproximadamente diez mil compuestos, de los cuales un 20% se utilizó como test set. Los modelos entrenados por aparte para tipos específicos de ácidos (alcoholes, ácidos carboxílicos, fenoles, etc.) o bases (amidinas, aminas, anilinas, etc.) dan mejores resultados que el modelo combinado, el cual presenta un RMSE de 0.81 unidades. Entre los principales restos mencionados por los autores es la inclusión de más tipos de grupos funcionales ionizables en el set de entrenamiento. Una de las ventajas de este modelo es que incluye moléculas multipróticas. El modelo fue creado para Novartris y se creó una aplicación para el cálculo de la pK_a. [70]

Un modelo enfocado en aminas alifáticas fue desarrollado probando cinco diferentes algoritmos de ML (RF, XGBoost, PLS, SVR y LASSO) para un set de 14 499 datos de pK_a experimentales. Cada molécula fue representada con descriptores provenientes de un *Rooted Fingerprint*, este corresponde a un tipo de huella dactilar que recoleta información estructural de los átomos vecinos al átomo que se selecciona como raíz. El algoritmo de ML con mejor desempeño fue el LASSO, seguido del XGBoost. El mejor modelo obtuvo buenos resultados para un set externo de 726 moléculas, para un RMSE de 0.45, MAE de 0.33 y R^2 de 0.84. [71]

Otro modelo enfocado en un solo tipo de moléculas fue una red neuronal propuesta para la predicción de la pK_a de ácidos benzoicos. En esta propuesta utilizaron 519 valores experimentales, de los cuales 136 corresponde a mediciones en agua, el resto son pK_as en disolventes orgánicos: DMA, DMF, acetona, isopropanol, metanol y DMSO. Para entrenar al modelo se utilizaron descriptores tanto para los ácidos benzoicos, 221, como para los disolventes, 19. El RMSE de las predicciones fue de 0.21, el cual es un valor bastante bueno para pK_a. En la misma investigación se intentó incluir fenoles al modelo y el RMSE subió a 0.59, siendo los fenoles la principal fuente de error. [72]

Con abordajes diferentes al resto de modelos mencionados hasta el momento, se realizó un modelo híbrido que integra un modelo de poblaciones de entropías con una red neuronal de función

de base radial. El desempeño de este modelo es excelente, el valor del RMSE para el set de prueba es de 0.04. Sin embargo, es de notar que el set de entrenamiento consiste solamente de sustancias neutras y básicas, y cuenta con tan solo 74 moléculas y el de prueba con 20. [73]

La gran mayoría de trabajos en predicción de pK_a se realizan en agua como disolvente. Sin embargo, existe una investigación donde se utilizan algoritmos de ML para predecir la pK_a en 39 disolventes diferentes, entre ellos DMSO, EtOH y MeCN. Se introdujo SPOC, una herramienta para obtener descriptores combinados de estructuras y propiedades físico-químicas. Se crearon dos modelos, uno holístico con los 39 disolventes y uno por cada disolvente por separado para los seis disolventes con los que se tenía mayor cantidad de datos. Los mejores resultados se obtuvieron al utilizar los algoritmos de redes neuronales y XGBoost entre todos los utilizados, con un RMSE de 1.34 y 1.43, respectivamente. En todos los casos el modelo holístico obtuvo mejores resultados, los menores errores se obtuvieron para los valores en EtOH/H_2O, MeOH y DMF; todos con MAE menor a la unidad. Con este modelo se obtuvo un RMSE de 1.07 unidades de pK_a para el reto a ciegas SAMPL6. [74]

Una investigación en 2019 aplicó modelos similares a QSAR aplicados en el pasado para la predicción de pK_a. Con un set de 7912 compuestos, que al curar se mantuvo un 79% de este, se utilizó para entrenar tres tipos de algoritmos de ML: SVM combinado con kNN, XGBoost y DNN. Como descriptores se utilizaron conteos de fragmentos, fingerprints binarios y descriptores moleculares continuos. Los tres tipos de modelos tuvieron desempeños similares con RMSE alrededor de 1.5 unidades. [75]

En un artículo de 2020, del cual se tomaron los valores experimentales de pK_a para entrenar el modelo del presente trabajo, se construyeron modelos utilizando 200 descriptores del módulo de Python RDKit y una *Morgan fingerprint* de 4096 bits de radio 3 para las moléculas en sus estados neutros. Los algoritmos de ML probados fueron RF, SVR, MLP y XGBoost. Al realizar una validación cruzada con $k = 5$, el modelo con mejor desempeño fue el RF al utilizar todos los descriptores y el *fingerprint* como inputs. [76]

Recientemente se reportaron varios modelos novedosos de *Active Learning*, el cual corresponde a un tipo de modelos de ML que consulta las etiquetas de nuevos datos (diferentes al del entrenamiento inicial) para mejorar el desempeño del modelo. En esta investigación se realizaron varios experimentos donde fueron entrenados con *Morgan fingerprints* de 2048 bits. El desempeño de los modelos propuestos no fue bueno, incluso los autores llaman como un fracaso

en la predicción de pK_a. Como recomendación los autores mencionan que este tipo de algoritmo usualmente utilizan sets de datos mucho más grandes. [77]

1.6. Coeficiente de distribución

1.6.1. Aspectos generales

El coeficiente de distribución (log *D*) se define como la razón de la concentración de un compuesto en un la fase lipofílica y la concentración en fase acuosa de todas las especies a un pH dado. [78] A diferencia del log *P*, el log *D* considera la partición de tanto las especies ionizadas y no ionizadas entre ambas fases. [79] El log *D* solo se utiliza en contextos donde se trabaje con moléculas ionizables y dependerá del pH de la fase acuosa. [7] En un contexto donde el 95% de las drogas aprobadas corresponden a moléculas ionizables, siendo 75% bases y 20% ácidos. [80]

Los modelos iniciales teoría de partición establecían que solo las especies neutras se particionan entre las fases. Por lo tanto, para el coeficiente de distribución no se toma en cuenta la concentración de la especie iónica en la fase lipofílica. En la Figura 15 se muestra el esquema de este modelo para un ácido. [81] En estas circunstancias el coeficiente de distribución depende del coeficiente de partición neutro y de la pKa de la molécula como se muestra en la ecuación 24, donde $\delta = pH - pK_a$ para ácidos y $\delta = pK_a - pH$ para bases.

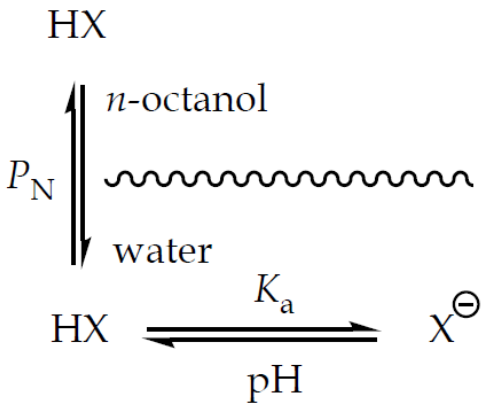


Figura 15. Esquema de mecanismo de partición *n*-octanol/agua para un ácido (HX) sin tomar en cuenta la partición de especies iónicas. [81]

$$\log D_{pH} = \log\left(\frac{[HA]_{n-octanol}}{[HA]_{agua} + [A^-]_{agua}}\right) \quad [23]$$

$$\log D_{pH} = \log P_N - \log\left(1 + 10^{\delta}\right) \quad [24]$$

Investigaciones posteriores donde se evidenció la partición de especies iónicas, un modelo más elaborado para el coeficiente de distribución fue propuesto. Donde la corrección inmediata fue incluir la concentración de la especie iónica en la fase lipofílica [81]. En la Figura 16 se muestra el esquema que soporta este modelo y en las ecuaciones 25 y 26 la expresión del coeficiente de distribución y la ecuación que lo relaciona con el coeficiente de partición iónica, neutra y la pK_a. Los valores de δ son iguales que en el modelo tradicional.

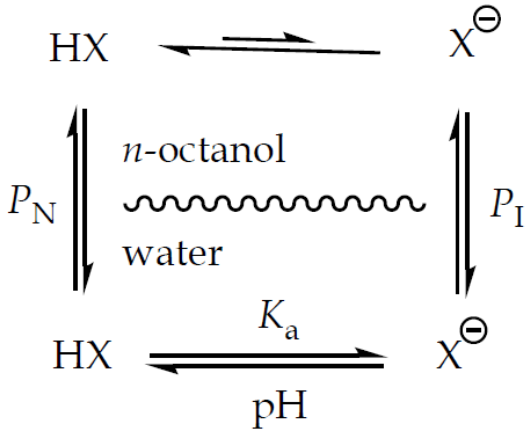


Figura 16. Esquema de mecanismo de partición *n*-octanol/agua para un ácido (HX) al tomar en cuenta la partición de especies iónicas. [81]

$$\log D_{pH} = \log\left(\frac{[HA]_{n-octanol} + [A^-]_{n-octanol}}{[HA]_{agua} + [A^-]_{agua}}\right) \quad [25]$$

$$\log D_{pH} = \log\left(P_N + P_I \cdot 10^{\delta}\right) - \log\left(1 + 10^{\delta}\right) \quad [26]$$

Como es evidenciado en las ecuaciones anteriores el coeficiente de distribución dependerá del pH de la fase acuosa. Las condiciones variables de pH en el cuerpo van a generar que los compuestos se encuentren en una mezcla de las especies neutras e ionizadas. A pesar de que los mecanismos bioquímicos mantienen el pH de la sangre a 7.4, a lo largo del tracto gastrointestinal el pH puede variar desde 1 hasta 8 unidades.[80] En el campo del diseño de fármacos, el log *D* se reporta al pH fisiológico (pH = 7.4) y en su gran mayoría las bases de datos son de solamente log $D_{7.4}$. [82]

1.6.2. Modelos de predicción de ML

La predicción de los coeficientes de distribución se ha centrado en el log $D_{7.4}$. Muchas alternativas han sido propuestas y han mejorado con el paso de los años. En 1997 se propuso un modelo a partir de varias regresiones lineales que predecían la partición y protonación de las moléculas. Las ecuaciones involucran la partición de los microestados de protonación según el pH, de manera que con sumatorias y productorias de los parámetros obtenidos de las regresiones (P, k y K) se pudiera obtener el coeficiente de distribución. Esta es de las pocas investigaciones que ha intentado obtener valores de log *D* a varios valores de pH. A pesar de que las ecuaciones derivadas tienen cierto significado físico, ya que se fundamentan en los equilibrios de partición de los microestados de protonación; los resultados obtenidos no son de la mejor calidad. El mejor de los modelos presentados tiene más de 0.5 de error para el 50% de set de prueba. Este modelo fue presentado bajo el nombre *PrologD*. [83]

En 2005 fue reportado un modelo para log $D_{7.4}$ con redes neuronales Bayesianas regularizadas (BRNN), usando bases de datos de AstraZeneca para entrenarlo. [84] Este modelo se basa en penalizar los pesos grandes que puedan darse en algunas neuronas, para reducir el sobreajuste y que el modelo sea más generalizado.[85] Luego de la estandarización y eliminación de moléculas repetidas el training set utilizado fue de 5000 moléculas, separadas cada una en un clúster. Los clústeres fueron creados con el total de las moléculas y se seleccionó uno de cada clúster para el training set para tener diversidad química en el modelo. El resto de moléculas son parte del test set, para un total de 3189. Para este modelo de BRNN se generaron 122 descriptores de propiedades 2D, 3D y de carga. Tras la eliminación de propiedades correlacionadas y de poca varianza se entrenó la BRNN con 56 descriptores no especificados por los autores. En cuanto a la arquitectura de la BRNN se utilizaron 43 nodos y se entrenó por 450 ciclos. Al probar una librería de compuestos se obtuvo un RMSE de 0.45. Los autores separaron luego las moléculas en bases, ácidos, zwitteriones y neutros; los errores más bajos fueron para las bases y neutros, mientras que los más altos para los ácidos.[84]

Un método para la predicción de log $D_{7.4}$ con SVM fue propuesto, en este caso se usó un total de 1130 moléculas, de las cuales 80% se usó como training set. Un total de 121 descriptores 2D fueron generados para entrenar el modelo, los mejores descriptores se seleccionaron mediane un algoritmo genético de regresión lineal múltiple. Se observó que luego de la utilización de 30 descriptores la mejoría del modelo no era significativa por lo que este fue el número final de

descriptores utilizados. Al hacer un análisis eliminando uno de los 30 descriptores a la vez, se notó que los que más influencia tenían en el modelo eran los descriptores relacionados con puentes de hidrogeno, polaridad y área superficial de la molécula. Como resultados de este modelo SVM se obtuvo un RMSE de 0.56, a modo de comparación se creó un modelo de PLS. El PLS obtuvo peores resultados que el SVM, para un RMSE de 0.69. Además, se hicieron comparaciones con modelos comerciales como Marvin y el SVM obtuvo mejores resultados. Al determinar el dominio de aplicabilidad del modelo se encontraron varios valores atípicos, entre ellos el péptido Ac−Tyr −Gly−Gly−Gln−NH_2 y el antibiótico rifampin. De esto se puede intuir que el modelo no es bueno para péptidos y macrociclos. [86]

Como se mencionó en uno de los modelos de log *P* las moléculas se pueden representar como grafos. Un modelo con un abordaje similar utiliza los grafos convolucionados para generar redes neuronales para predecir log $D_{7.4}$. El desempeño del método *Graph Convolutional Deep Neural Network* (GC-DNN) con tres capas ocultas fue mejor que para algoritmos que utilizaban como inputs vectores con la información del grafo que usa el GC y otros descriptores, RF y FC-DNN. Los autores hacen énfasis en lo importante que es la información de conectividad de la molécula y atribuyen a eso el mejor desempeño del método GC-DNN. [87]

Un estudio con el objetivo de crear un punto de referencia para comparar métodos de ML en el diseño de drogas (mecánica cuántica, propiedades físico-químicas, biofísica y fisiológicas), se comparan los métodos de: MPNN, DAG, GC, KRR, XGBoost, Singletask NN, RF y Weave. En el caso del log $D_{7.4}$ el mejor modelo fue el de GC, lo que muestra que la representación de las moléculas como grafos tiene un buen potencial con RMSE debajo de 0.7. El segundo modelo con mejor desempeño es el Weave, este modelo también se basa en información gráfica, la diferencia radica en que transmite mejor la información entre átomos lejanos, pero aumenta la complejidad de la convolución. [88]

Otro algoritmo utilizado es el *Cross Conformal Prediction* (CCP), consiste en una regresión que predice un intervalo según la similitud del dato con el training set. En este caso el training set se divide en un set de entrenamiento y uno de calibración; se puede dividir en varios sets más para entrenar y calibrar. El intervalo que predice también va a depender del intervalo de confianza que el usuario seleccione. Este algoritmo fue usado con 1.5 millones de compuestos, para predecir log $D_{7.4}$. Sin embargo, este gran set corresponde a valores de log *D* calculados y no experimentales.

Como descriptores se utilizaron vectores de 1068830 enteros que contienen información sobre los átomos, su ambiente químico, átomos vecinos y otros atributos 2D de la molécula. A pesar de los buenos resultados que se muestran en el artículo, RMSE 0.41, estos se evalúan contra un log *D* calculado. El error de este modelo contendrá intrínsecamente el error del modelo con que se obtuvieron los valores para el entrenamiento. [89]

Recientemente se publicó una investigación donde se comparan varios algoritmos de ML para la predicción de log $D_{7.4}$. Un training set de 3501 moléculas fue utilizado. Como descriptores se utilizaron: 19 propiedades físicas, 14 de teoría de Huckel, 18 de área superficial, 42 de conteo de átomos/enlaces, 16 de conectividad/forma, 33 de proximidad y distancia, 13 de farmacóforos y 50 de cargas parciales; para un total de 205. De estos se eliminan los altamente correlacionados ($R^2 \geq 0{,}95$) o los de varianza cercana a cero. Ocho algoritmos fueron probados: RF, RVM, SVM, Cubist, GP, GB, XGBoost y DL. [90]

El RVM es un modelo de maximización de expectativa baja, reduce el número de SV requeridos para modelar el límite de decisión en el hiperparámetro espacial creado con los vectores. Cubist se basa en la construcción de árboles de regresión que hacen que la predicción se haga basado en una regresión lineal y no en valores discretos como algoritmos que usan arboles de decisión. GP se basa en inferencia probabilística Bayesiana. GB es *gradient boosting*, algoritmo predecesor del XGBoost. [90]

De todos los modelos desarrollados el que obtuvo los mejores parámetros de desempeño la investigación fue el XGBoost, es acorde a lo esperado ya que este algoritmo ha sido el ganador de varias competencias recientemente. Los autores hicieron un modelo consenso, promediando las predicciones de los tres mejores modelos XGBoost, GB y SVM. Los descriptores más relevantes del modelo fueron log *P* y log *S* computados. Al establecer el dominio de aplicabilidad del modelo se notó que los *outliers* eran principalmente compuestos péptido-miméticos.[90]

La mayoría de estudios de ML en descriptores lipofílicos se centra en moléculas pequeñas. Mediante algoritmos LASSO y SVM se generaron modelos para la predicción de log $D_{7.4}$ de péptidos y péptido miméticos. El algoritmo LASSO (*Least absolute shrinkage and selection operator*) consiste en una regresión regularizada multivariable donde se puede controlar la dimensionalidad del modelo. Se generaron 120 descriptores de atributos 1D-2D de los péptidos, los descriptores fueron seleccionados por minimización del RMSE con el algoritmo LASSO o por

PCA. Con el algoritmo LASSO se seleccionaron 11 descriptores y con PCA 20. La mayoría de estos relacionados con carga y área polar superficial. El set para el modelo consistió de 243 péptidos disponibles al público y 800 péptidos/péptido miméticos de AstraZeneca. El mejor de los modelos resulto ser el SVM con los descriptores seleccionados por LASSO. Sin embargo, los autores hicieron un ponderado según los RMSE obtenidos para calcular un modelo consenso entre los SVM con los descriptores de LASSO y PCA. Este modelo obtuvo mejor desempeño para varios test sets. Obtuvo como mejor resultado un RMSE de 0.38 para un test set de 64 péptidos.[91]

Recientemente se reportó un modelo donde se utilizaron data sets públicos (*ChEMBL* [92,93]), de licencia (*BioByte* [94]) y privados (*Genentech*) para la predicción de log *D*. Se utilizaron en primera instancia valores experimentales de log *D* a pH de 7.4, para entrenar algoritmos de cubista, RF y SVM. Como descriptores utilizados fueron: ClogP, ClogD, pKa, carga formal, fracción ionizada, fingerprints ECFP4[95], índices electrotopológicos, conteos de átomos, donadores/aceptores de puente de hidrógeno y TPSA. De esta manera se obtuvieron los mejores resultados con SVM, con un RMSE de 0.66 con el set de licencia filtrado a obtener un 75%. Posteriormente utilizan valores predichos de pK_a por softwares comerciales para utilizar la ecuación 6 para despejar y obtener valores de log *P*. Entre las principales limitaciones expuestas se encuentra la baja calidad de las predicciones de pK_a.[96] Este abordaje sería inverso a lo que se propone en este trabajo, ya que el objetivo es predecir valores de log *P* y pK_a, para obtener el valor de log *D* a cualquier pH.

La compañía farmacéutica Pfizer desarrolló un modelo de ML para la predicción del coeficiente de distribución de datos obtenidos por la técnica de shake-flask miniatura (SFlogD, n = 200 mil) y el método cromatográfico (ElogD, n = 80 mil). A partir de estos datos construyeron modelos de regresión lineales, que con los valores arrojados entrenaron el modelo de ML. El modelo fue basado en XGBoost, con un set de entrenamiento del 80% y de testeo del 20%. Para seleccionar de cuál metodología obtener el dato se emplearon una serie de reglas basadas en si la molécula es un zwiterión y el valor de log *D*. Al probar los modelos obtuvieron que para el SFlogD un 71.0% tienen errores menores a 0.5 unidades de log D y un 75.9% para el ElogD. No se reportan los valores de RMSE, MSE o MAE. [97]

Capítulo 2:

Metodología

2. Metodología

2.1. Coeficiente de partición neutro (log P_N)

2.1.1. Base de datos

Para construir los modelos de predicción del coeficiente de partición neutro (log P_N) se utilizaron dos bases de datos para obtener valores experimentales y el código SMILES de las moléculas correspondientes; la primera se obtuvo de In *Silico Prediction of Physicochemical Propertiers of Environmental Chemicals Using Molecular Fingerprints and Machine Learning* [48] y la segunda de *DrugBank*[98]. Por facilidad, a la primera base de datos se le denominó ENV y la segunda DB. De estas bases de datos se extrajeron las observaciones que tenían un SMILES correcto y valor de (log P_N) experimental. La verificación de la validez del código SMILES se realizó con el módulo de Python RDkit [99], se dejó por fuera las moléculas cuya conversión de SMILES a representación molecular (Mol) condujera a un error o advertencia.

La base de datos ENV contenía un total de 13 819 compuestos y DB 1 102. Se fusionaron las dos bases de datos mencionadas en una denominada como ALL para fines del código, en total se tenían 14 921 compuestos. Se eliminaron compuestos duplicados convirtiendo todas las moléculas de código SMILES a formato Mol y luego viceversa. De esta manera dejó solo una de las observaciones en casos donde hayan SMILES duplicados. Luego de eliminar duplicados, se obtuvo un total de 14 477 compuestos.

Para asegurar que la base de datos correspondiera solo de moléculas neutras, se procedió a eliminar sales con la herramienta *SaltRemover* de RDkit [99]. Se eliminaron un total de 93 sales. En el caso de compuestos que estuvieran cargados se utilizó la fórmula *GetFormalCharge* para obtener la carga de las moléculas, se eliminaron un total de 16 moléculas cuya carga era diferente de cero. En este punto se contaban con un total de 14 368 moléculas neutras.

Una de los pasos para la preparación de datos es el recorte de la base de datos en búsqueda de anomalías u *outliers*. En caso de que se dejen estos datos los algoritmos pueden arrastrar el sesgo que estos introducen. [100] Para hacer el filtrado se tomó tanto en cuenta la regla de Lipinski para log P_N (log $P_N < 5$) y de la masa molecular (MW, *Molecular Weight* < 500) [101], como la caracterización estadística del dataset. En la Figura 17 se puede observar la dispersión de las observaciones según log P_N y MW, es claro que existen puntos que se alejan de la mayoría de la población. Estos outliers

se encuentran a altas MW y valores extremos del log P_N. En las Figuras 18 y 19 se puede observar la distribución de frecuencias para estas dos propiedades.

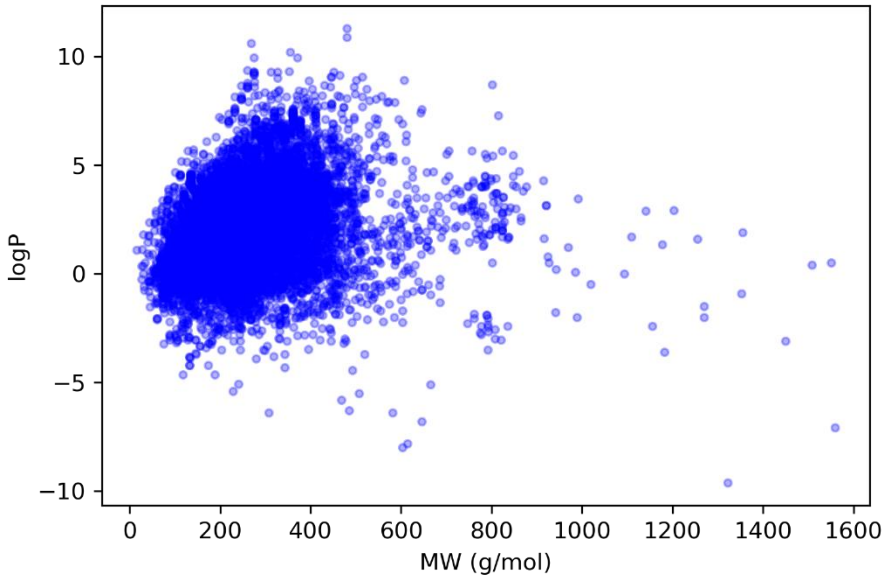


Figura 17. Dispersión MW vs log P_N de la base de datos previo al filtrado.

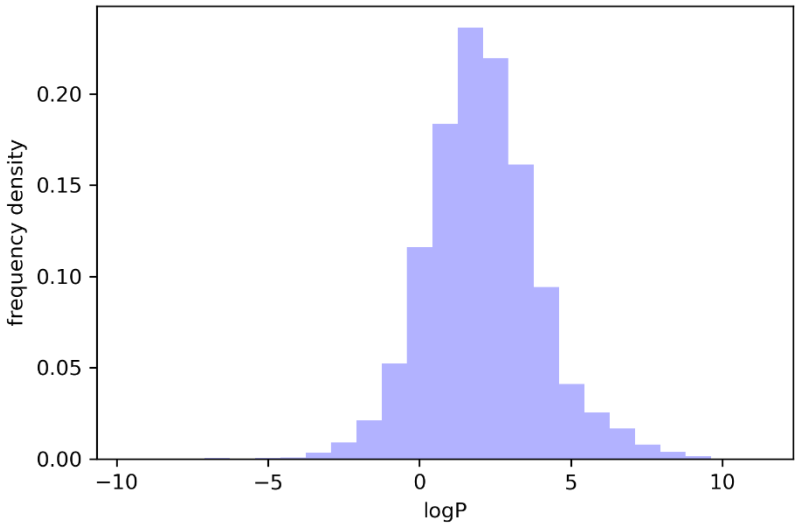


Figura 18. Distribución de densidad de frecuencias de log P_N de la base de datos previo al filtrado.

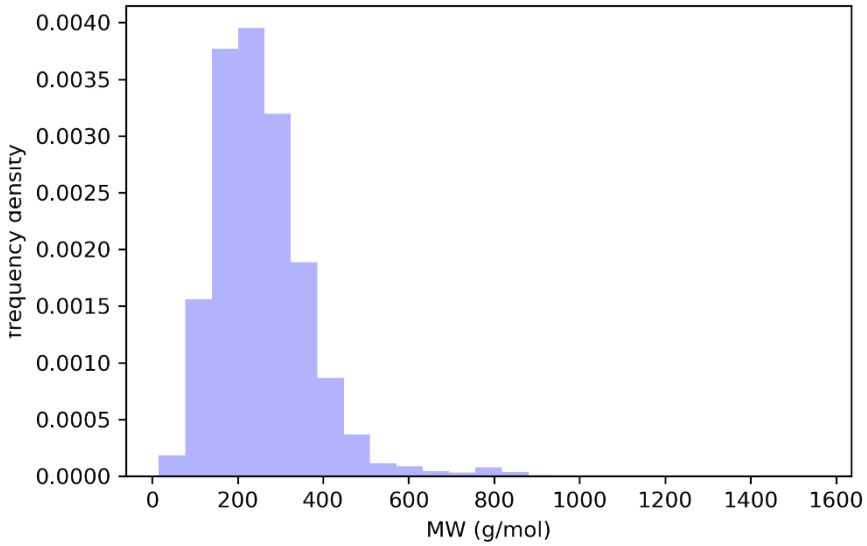


Figura 19. Distribución de densidad de frecuencias de MW de la base de datos previo al filtrado.

En el Cuadro III se muestra un resumen estadístico de las distribuciones para log P_N y MW, se incluyen los rangos si se hiciera un filtrado con dos y tres desviaciones estándar. Se tomó la decisión de hacer el filtrado con tres desviaciones estándar ya que el límite inferior para log P_N con dos desviaciones estándar dejaría por fuera información importante de compuestos menos lipofílicos y en el caso de la masa molecular el límite superior se encuentra inferior a 500 g/mol, lo que deja un límite muy abrupto para moléculas que rondan los 500 g/mol.

Cuadro III. Resumen estadístico de la base de datos (N = 14 368) para el log P_N y MW.

Propiedad	Promedio	Mediana	σ	$\bar{x}$ - 2 σ	$\bar{x}$ + 2 σ	$\bar{x}$ - 3 σ	$\bar{x}$ + 3 σ
log P_N	2.08	2.0	1.87	-1.66	5.83	-3.54	7.70
MW (g/mol)	258	240	118	22	494	-97	613

Al filtrar por log P_N se dejaron por fuera un total de 115 observaciones y por MW 212. Luego del filtrado, el número total de observaciones es de 14 041. En la Figura 20 se muestra la dispersión para estas dos propiedades luego del filtrado de la base de datos.

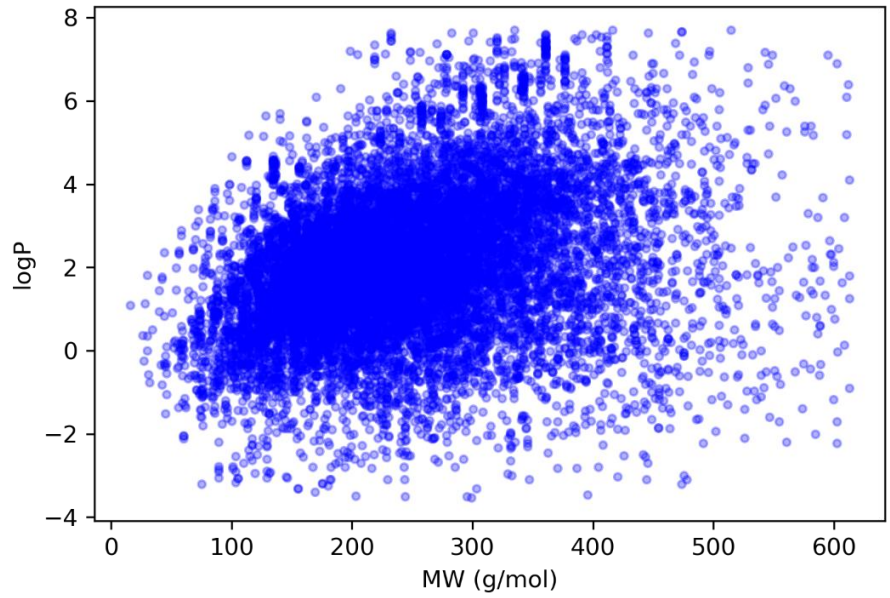


Figura 20. Dispersión MW vs log P_N de la base de datos luego del filtrado.

2.1.2. Descriptores

Para el cálculo de descriptores que puedan funcionar como variables independientes para la predicción del log P_N se optó por utilizar el módulo RDkit [99], se utilizó conteo de grupos funcionales (todos los posible proporcionados por el módulo), masa molecular, refractividad molar, número de aceptores de puente de hidrógeno, número de donadores de puente de hidrógeno, número de enlaces rotables, número de anillos aromáticos, número de anillos alifáticos y área polar superficial; para un total de 96 descriptores.

Entre descriptores se buscó cuáles tienen eran igual a cero para todas las observaciones, se identificó tres descriptores y se procedió a eliminarlos: fr_diazo, fr_isocyan y fr_prisulfonamd. La multicolinealidad se da cuando una o más variables independientes tienen una alta correlación lineal. Cuando esto ocurre se debe dejar solo una de las variables, de lo contrario se estaría introduciendo un sesgo al modelo. Como regla empírica se dice que hay un alta multicolinealidad cuando el coeficiente de determinación es mayor a 0.95. [102] Para los 93 descriptores restantes, se calculó el coeficiente de correlación entre cada columna y se identificaron los que tuvieran un $R^2 > 0.95$. En el Cuadro IV se muestran los descriptores altamente correlaciones y su R^2

respectivo. Se eliminaron los descriptores mostrados en la segunda columna, para dejar un total de 89.

Cuadro IV. Descriptores con alta multicolinealidad identificados en el data set.

Descriptor 1	**Descriptor 2**	$\boldsymbol{R^2}$
fr_Ar_NH	fr_Nhpyrrole	1.00
fr_COO	fr_COO2	1.00
fr_phenol	fr_phenol_noOrthoHbond	0.97
fr_phos_acid	fr_phos_ester	0.98

En el Cuadro V se muestra una breve descripción de todos los 89 descriptores remanentes que fueron utilizados para el entrenamiento de los modelos.

Cuadro V. Descriptores calculados para la predicción del log P_N.

Descriptor	Descripción
fr_Al_COO	Número de ácidos carboxílicos alifáticos
fr_Al_OH	Número de grupos hidroxilos alifáticos
fr_Al_OH_noTert	Número de grupos hidroxilos alifáticos excluyendo tert-OH
fr_ArN	Número de grupos funcionales con N enlazados a un anillo aromático
fr_Ar_COO	Número de ácidos carboxílicos aromáticos
fr_Ar_N	Número de nitrógenos aromáticos
fr_Ar_NH	Número de aminas aromáticas
fr_Ar_OH	Número de hidroxilos aromáticos
fr_COO	Número de ácidos carboxílicos
fr_C_O	Número de carbonilos
fr_C_O_noCOO	Número de carbonilos excluyendo ácidos carboxílicos/carboxilatos
fr_C_S	Número de tiocarbonilos
fr_HOCCN	Número de C(OH)CCN-Ctert-alquil or C(OH)CCNcíclico
fr_Imine	Número de iminas
fr_NH0	Número de aminas terciarias

fr_NH1	Número de aminas secundarias
fr_NH2	Número de aminas terciarias
fr_N_O	Número de hidroxilaminas
fr_Ndealkylation1	Número de grupos XCCNR
fr_Ndealkylation2	Número de aminas tert-alicíclicas
fr_SH	Número de tioles
fr_aldehyde	Número de aldehídos
fr_alkyl_carbamate	Número de alquil carbamatos
fr_allylic_oxid	Número de sitios de oxidación alílica
fr_amide	Número de amidas
fr_amidine	Número de grupos amidinos
fr_aniline	Número de anilinas
fr_aryl_methy	Número de sitios aril metil para hidroxilación
fr_azide	Número de azidas
fr_azo	Número de grupos azo
fr_barbitur	Número de grupos barbiturato
fr_benzene	Número de anillos de benceno
fr_benzodiazepine	Número de benzodiazepinas
fr_bicyclic	Número de biciclos
fr_dihydropyridine	Número de dihidropiridinas
fr_epoxide	Número de anillos epóxido
fr_ester	Número de ésteres
fr_ether	Número de éteres
fr_furan	Número de anillos furano
fr_guanido	Número de grupos guanidino
fr_hdrzine	Número de grupos hidrazina
fr_hdrzone	Número de grupos hidrazona
fr_imidazole	Número de anillos imidazol
fr_imide	Número de grupos imida
fr_isothiocyan	Número de isotiocianatos

fr_ketone	Número de cetonas
fr_ketone_Topliss	Número de cetonas excluyendo diarilo, α,β-insaturadas, dienonas y con heteroátomos en el C α
fr_lactam	Número de betalactamas
fr_lactone	Número de ésteres cíclicos (lactonas)
fr_methoxy	Número de grupos metoxi
fr_morpholine	Número de anillos de morfolina
fr_nitrile	Número de grupos nitrilo
fr_nitro	Número de grupos nitro
fr_nitro_arom	Número de grupos nitro en anillos aromáticos
fr_nitro_arom_nonortho	Número de grupos nitro en anillos aromático excluyendo posición orto
fr_nitroso	Número de grupos nitroso
fr_oxazole	Número de anillos de oxazol
fr_oxime	Número de grupos oxima
fr_para_hydroxylation	Número de sitios de para-hidroxilación
fr_phenol	Número de fenoles
fr_phos_acid	Número de grupos de ácido fosfórico
fr_piperdine	Número de grupos piperidina
fr_piperzine	Número de grupos piperazino
fr_priamide	Número de amidas primarias
fr_pyridine	Número de anillos de piridina
fr_quatN	Número de aminas cuaternarias
fr_sulfide	Número de tioéteres
fr_sulfonamd	Número de sulfonamidas
fr_sulfone	Número de sulfonas
fr_term_acetylene	Número de acetilenos terminales
fr_tetrazole	Número de anillos de tetrazol
fr_thiazole	Número de tiazoles
fr_thiocyan	Número de tiocianatos

fr_thiophen	Número de tiofenos
fr_unbrch_alkane	Número de alquenos no ramificados (al menos 4 C)
fr_urea	Número de grupos urea
MW	Masa molar
MR	Refractividad molar computada
HBA	Número de aceptores de puentes de hidrógeno
HBD	Número de donadores de puentes de hidrógeno
RotBonds	Número de enlaces rotables
AromRings	Número de anillos aromáticos
AliphRings	Número de anillos alifáticos
PSA	Área polar superficial topológica
C	Número de carbonos
Cl	Número de cloros
F	Número de flúores
Br	Número de bromos
I	Número de yodos

2.1.3. Modelos

Es una buena práctica iniciar con un modelo de MLR para establecer una base de referencia del desempeño previo a tratar con modelos no lineales. [10] Para la programación del modelo de MLR se utilizó de la librería *scikit-learn* el módulo *linear_model* [103]. En el Cuadro VI se presenta a modo de resumen las diferentes condiciones de modelos MLR que se efectuaron. Para todas se calculó el RMSE en función del porcentaje del dataset utilizado como training set. Con los resultados de los modelos de MLR se tomaron decisiones para construir el resto de modelos con los otros algoritmos.

Cuadro VI. Condiciones utilizadas para generar los modelos de MLR.

Modelo	Descripción de las condiciones utilizadas
MLR-1	Totalidad de descriptores
MLR-2	25% de los descriptores que más correlacionan con el log P_N
MLR-3	50% de los descriptores que más correlacionan con el log P_N
MLR-4	75% de los descriptores que más correlacionan con el log P_N
MLR-5	Solamente descriptores de conteo grupos funcionales
MLR-6	Totalidad de descriptores, pero regularizados
MLR-7	25% de los descriptores que más correlacionan con el log P_N, pero regularizados
MLR-8	50% de los descriptores que más correlacionan con el log P_N, pero regularizados
MLR-9	75% de los descriptores que más correlacionan con el log P_N, pero regularizados
MLR-10	Solamente descriptores de conteo grupos funcionales, pero regularizados

Para este análisis inicial se consideró solamente el RMSE para evaluar los modelos. Basándose en los resultados del MLR se eligió las condiciones que mejor desempeño tuvieron para probar con el resto de modelos por probar. El segundo tipo de algoritmo probado fue el PLS con el módulo PLSRegression. [103] Se probó con la totalidad de los descriptores, con dos y tres componentes para porcentajes del dataset como training set desde 50 hasta 99%, para efectos de codificación se les llamó PLS-2 y PLS-3 respectivamente.

En el caso del algoritmo de boosting, se utilizó el paquete de Python *xgboost*[104] y se le asignó el código XGB. Se calculó el RMSE para modelos entrenados con training sets correspondientes al 80-99% del dataset. De la misma manera se realizó para el modelo de Random Forest, RF, creado con el módulo *RandomForestRegressor* de *sckit-learn*. [103]

Varias arquitecturas de ANN de perceptrón fueron entrenadas con el algoritmo de MLP del módulo *MLPRegressor* de *scikit-learn*. El modo de activación de las neuronas fue *'relu'* y la función de resolución de funciones *'adam'*.[103] Para el diseño de las arquitecturas se tomaron en cuenta varias recomendaciones de la literatura: el número neuronas en capas ocultas debe estar entre el tamaño de la capa output y el del input, la mayoría de problemas se pueden solucionar al usar solo dos capas de neuronas y que la cantidad de neuronas ocultas debe ser menor que el doble de las neuronas del input. [105]

Se probaron varios abordajes mencionados en la literatura para el cálculo del tamaño de las capas ocultas, las ecuaciones 27, 28 y 29 muestran las fórmulas utilizadas, donde N_h corresponde al número de neuronas de la capa oculta, N_i a las de la capa input y N_o a las de la capa output. [106] Sin embargo, la selección final de la arquitectura se debe realizar con prueba y error. [105] A partir de los resultados preliminares se trabajó posteriormente con esa arquitectura como base para luego mejorarla. En el Cuadro VII se muestra un resumen de las arquitecturas probadas.

$$N_h = \frac{\sqrt{1 + 8N_i} - 1}{2} \quad [27]$$

$$N_h = \sqrt{N_i N_o} \quad [28]$$

$$N_h = \frac{4N_i^2 + 3}{N_i^2 - 8} \quad [29]$$

Cuadro VII. Resumen de las arquitecturas de redes neuronales artificiales MLP utilizadas. Los tamaños de las capas ocultas se expresan según la cantidad de neuronas en cada una.

Modelo	Tamaño de la/las capas ocultas
NN-1	120
NN-2	170
NN-3	400
NN-4	1024, 512
NN-5	120, 10
NN-6	128, 64
NN-7	170, 18
NN-8	256, 128
NN-9	400, 200
NN-10	400, 4
NN-12	500, 100
NN-13	512, 256
NN-14	1024, 512, 256
NN-15	128, 128, 128

NN-16	256, 128, 64
NN-17	256, 256, 256
NN-18	400, 200, 100
NN-19	512, 256, 128
NN-20	512, 512, 512

2.1.4. Evaluación y validación

Al tener todos los resultados de los algoritmos se procedió a elegir los dos con mejor desempeño para realizar validaciones cruzadas y validaciones con test set externos. Además, si el tipo de algoritmo lo permite, se realizan pruebas optimización en la arquitectura o en parámetros del algoritmo. En el caso de las redes neuronales se probaron las diferentes funciones de activación ofrecidas por el módulo utilizado.

En el caso de las validaciones cruzadas de k-iteraciones se realizaron con $k = 10$, ya que según evidencia empírica de la literatura sugiere que se obtiene una baja varianza y un bajo sesgo, esto también ocurre con $k = 5$. [107] Las validaciones externas se realizaron con dos bases de datos SAMPL6 [108] ($n = 11$) y SAMPL7 [109] ($n = 22$). En ambos tipos de validación se utilizó RMSE, MSE, MAE y R^2 como métricas de evaluación.

Se realizó también el cálculo la matriz de leverage para establecer el dominio de aplicabilidad del modelo. En la gráfica de Williamson se incluyó el training set, test set y los dos sets de validación externa.

2.2. Coeficiente de partición iónico (log P_I)

2.2.1. Base de datos

La base de datos para la predicción del coeficiente de partición iónico se construyó a partir de 104 valores obtenidos (AVD) del libro *Absorption and Drug Development* [7]. De estas observaciones se eliminó la morfina y el enalaprilato que tenían carga 2+ y se salen de los objetivos planteados. La base AVD se separó en dos, MON para los compuestos simple monocargados y ZW para los compuestos zwiteriónicos monocargados, con 76 y 26 observaciones respectivamente. Además, se obtuvieron valores provenientes de las bases de datos ENV [48] y DB [98] donde previamente se habían filtrado 93 compuestos que correspondían a sales (SALTS). De

estas sales se recuperaron 17 observaciones, luego de filtrar compuestos que correspondieran aminas cuaternarias y aquellos que no tenían valores de log P_N.

Para cada base se programó un script para que cada observación tuviera el SMILES neutro y cargado (sin contraión) de cada molécula con la librería *RDkit* [99] de Python y *OpenBabel* [110]. En el Cuadro VIII se resumen los tamaños de las subbases de datos empleadas y como se fusionaron para obtener dos bases finales con las qué probar si incluir o no los zwiteriones.

Cuadro VIII. Resumen del tamaño de las bases de datos empleadas para la predicción del log P_I. El número de observaciones de las subbases se muestra entre paréntesis.

Base de datos	Número de observaciones	Subbases de datos
ALL_logPI	93	MON (76)
		SALTS (17)
ALL_logPI_Zw	119	MON (76)
		SALTS (17)
		ZW (26)

2.2.2. Descriptores

Se calcularon los mismos descriptores que para la base de datos de log P_N, con la diferencia de que se agregó la refractividad molar y el área polar superficial de las moléculas, pero en su forma cargada. El total de descriptores calculados fue de 99. Se realizó un filtro para eliminar los descriptores cuyas columnas estaban en cero; para la base ALL_logPI se eliminaron 33 y 27 para ALL_logPI_Zw. El total de descriptores remantes para el entrenamiento fue de 66 y 72, respectivamente.

Para predecir el log P_I no se optó por predecir el valor de este directamente, se optó por predecir la diferencia entre el log P_N y el log P_I. A esta diferencia se le llamó Delta (log P_N - log P_I) o por facilidad seguirá siendo mencionado como Delta. En la Figura 22 y Figura 23 se muestran los histogramas de la distribución de log P_N, log P_I y el Delta; para las bases de datos con y sin zwiteriones, respectivamente.

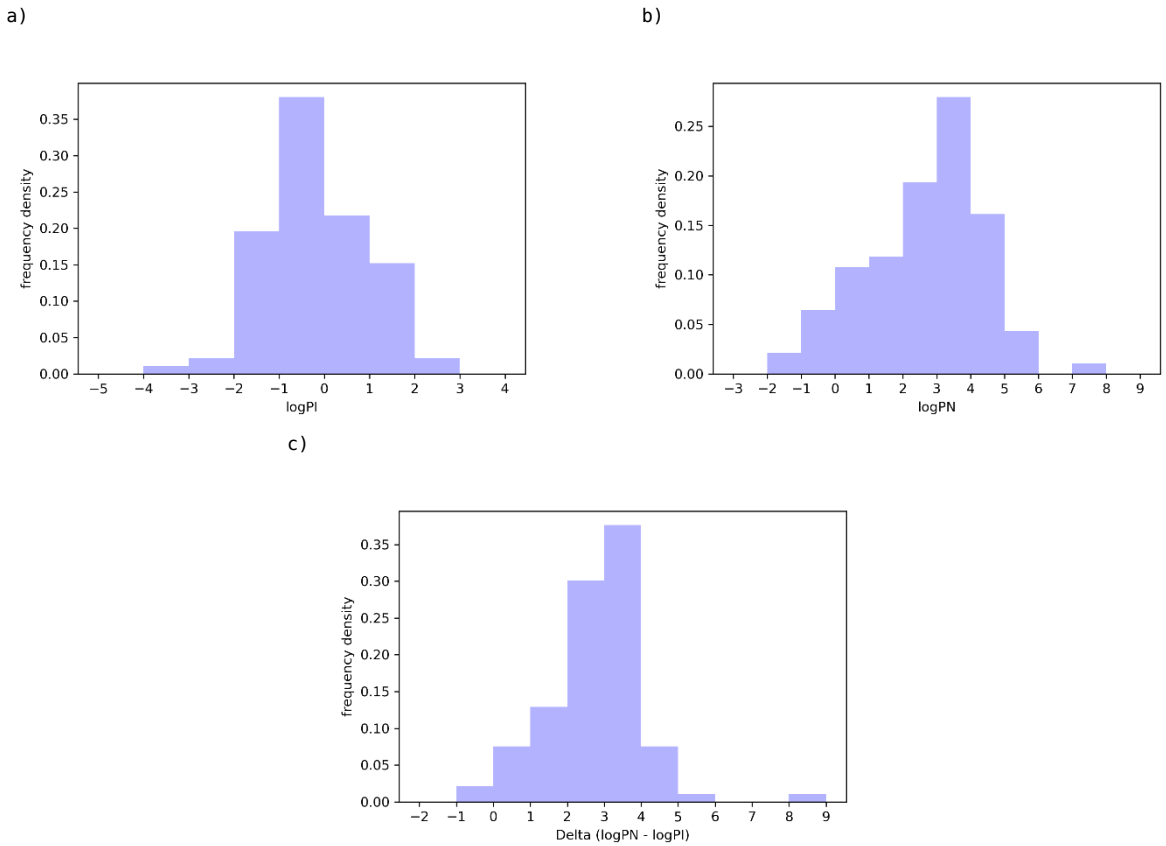


Figura 22. Distribución de densidad de frecuencias de los valores de a) log P_I, b) log P_N y c) Delta para la base de datos ALL_logPI.

2.2.3. Modelos

De igual manera que para la predicción del log P_N se probaron inicialmente varios tipos de algoritmos y de acuerdo al RMSE para el test set se deciden elegir dos algoritmos con mejor desempeño para realizar evaluaciones más detalladas. Debido a que la variable a predecir es la diferencia entre el log P_N y el log P_I, en los MLR exploratorios se calculó la diferencia entre la refractividad molar del compuesto neutro y cargado; de igual manera con el área polar superficial. Dada la poca cantidad de observaciones disponibles, se testeó también si el desempeño de los modelos mejoraba al reducir el número de descriptores cuyas observaciones fueran mayores a cero en al menos 70% de los datos. En el Cuadro VI se presenta un resumen de varias condiciones con las que se probaron tanto los modelos de MLR, RF, PLS, NN, SVM y XGBoost para predecir el Delta.

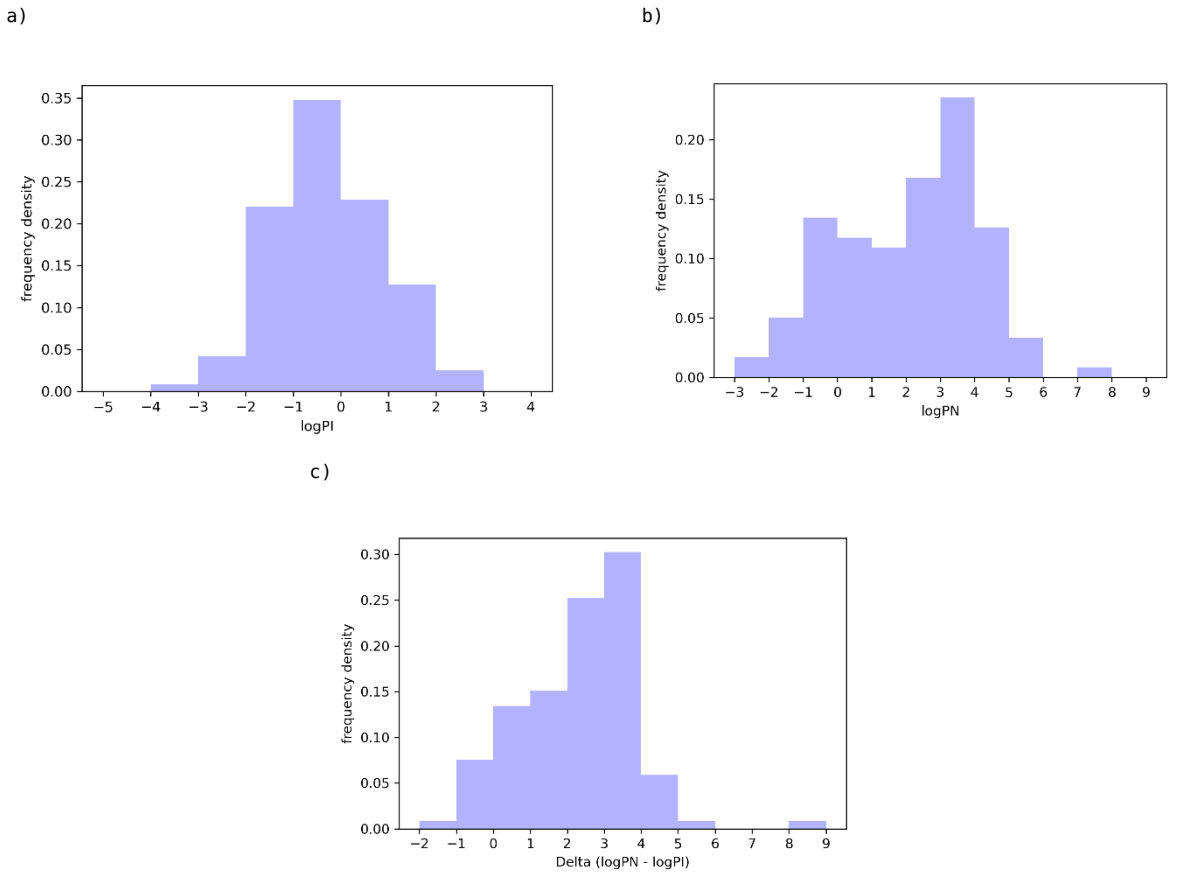


Figura 23. Distribución de densidad de frecuencias de los valores de a) log P_I, b) log P_N y c) Delta para la base de datos ALL_logPI_Zw.

2.2.4. Evaluación y validación

Los modelos con mejor desempeño con el test set fueron seleccionados para hacer validación cruzada y externa. De acuerdo a las métricas (RMSE, R^2, MSE y MAE) de estas validaciones y las del test set, se elige el modelo con mejor desempeño. El set utilizado para la validación externa[111], se filtró para utilizar solamente moléculas monopróticas y monobásicas. Además, se realizó un gráfico de Williams para definir el dominio de aplicabilidad de los modelos.

Cuadro IX. Modelos de predicción de Delta (log P_N - log P_I)) de regresión lineal múltiple.

Código de condición	Dataset	Delta_MR y Delta_PSA	Porcentaje mínimo de observaciones diferentes de cero	Número de descriptores
I-1	ALL_logPI	No	0	66
I-2	ALL_logPI	Sí	0	64
I-3	ALL_logPI	No	70	28
I-4	ALL_logPI	Sí	70	26
I-5	ALL_logPI_Zw	No	0	72
I-6	ALL_logPI_Zw	Sí	0	70
I-7	ALL_logPI_Zw	No	70	30
I-8	ALL_logPI_Zw	Sí	70	26

2.3. Constante de acidez (pKa)

2.3.1. Base de datos

La base de datos utilizada se recuperó de *Machine Learning meets* pK_a [76]. La base de datos fue pre-curada por los autores; de manera que no contuviera sales, moléculas con boro, selenio o silicio; moléculas que violaran más de una de las reglas de Lipinski, ni duplicados. Los autores mencionan que el rango de pK_a la base es entre 2 y 12 unidades. Además, que los SMILES de las moléculas corresponden a los estados de protonación a 7.4 unidades de pH. El total de la base fue de 5994 observaciones, 2398 ácidos y 3596 bases. Sin embargo, se encontraron ciertas inconsistencias en moléculas que debían estar cargadas o neutras al pH indicado. En el Cuadro X se muestra un resumen de lo observado respecto a las cargas de las moléculas de la base de datos.

Las moléculas con cargas múltiples se descartaron por salirse de los objetivos de esta investigación. Para el resto de moléculas se utilizó el software *dimorphite_dl 2.4* [112] para generar los posibles estados de protonación de las moléculas. La mayoría de moléculas, 2742, presentaban

solo dos estados de protonación (neutro y cargado), lo cual se alinea con el objetivo de predicción de pK_a para monopróticas y monobásicas.

Cuadro X. Resumen de las cargas de las moléculas de la base de datos previo a ser curada.

	Ácidos	Bases
Totalidad	**2398**	**3596**
Cargados correctos	1140	1663
Cargados incorrectos	34	446
Neutros correctos	827	1277
Neutros incorrectos	317	170
Cargas múltiples	80	40

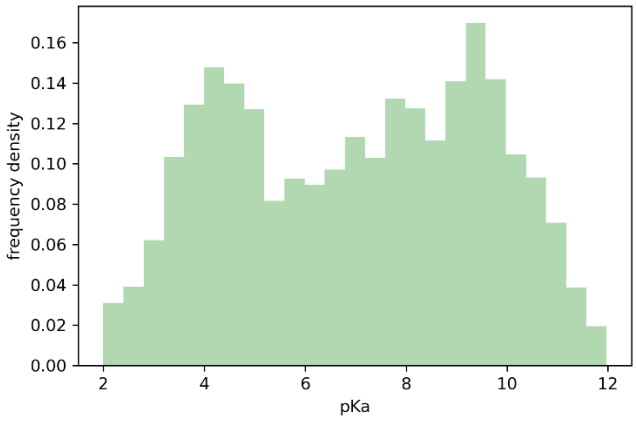
Para poder alimentar a los modelos con una mayor cantidad de datos, mediante reglas generales de acidez y basicidad se construyeron scripts con la librería *rdkit* [99] para seleccionar los dos estados de ionización involucrados en el equilibrio de la pKa correspondiente en los casos donde *dimorphite_d*l 2.4 arrojaba más de dos estados de ionización. De esta manera, se logró recuperar un total de 4363 observaciones con los dos SMILES de los estados de protonación involucrados en el equilibrio. En el Cuadro XI se muestra el resumen estadístico de la base de datos resultante.

Cuadro XI. Descripción estadística de la base de datos de pKa resultante luego del curado.

Parámetro estadístico	Base de datos completa	Ácidos	Bases
Número de observaciones	4363	1844	2517
Promedio	7.00	6.35	7.48
Desviación Estándar	2.50	2.65	2.28
Mínimo	2.00	2.01	2.00
Máximo	11.97	11.97	11.94

Como se puede observar en la siguiente figura, la distribución de valores de pKa se aleja de ser una distribución normal debido a la diferencia entre valores de ácidos y bases. Al observar los histogramas por separado por esta categoría se puede observar que tampoco siguen la distribución normal.

a)



b) c)

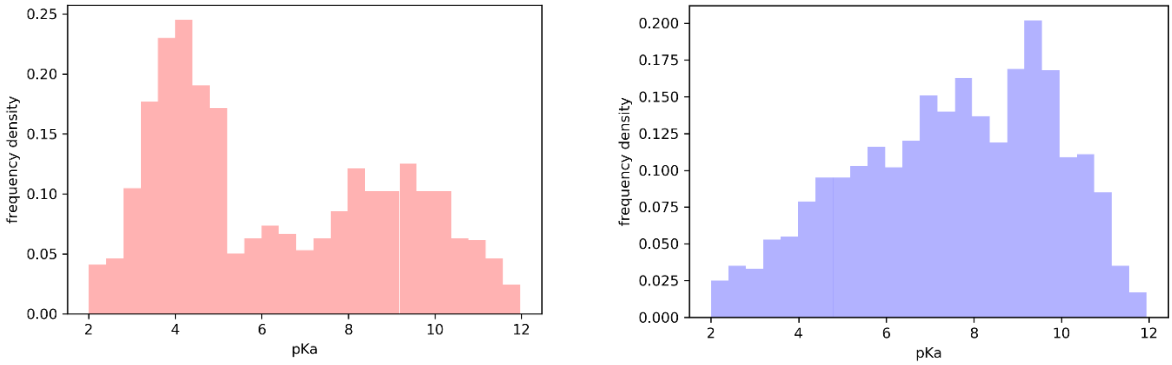


Figura 24. Distribución de valores de pKa de la base de datos curada para a) todas las observaciones, b) ácidos y c) bases.

2.3.2. Descriptores

Para cada una de las observaciones se calcularon todos los descriptores posibles del módulo *rdkit.Chem.Descriptors* tanto para la molécula neutra como para la molécula cargada, luego se calculó la diferencia entre cada descriptor. Se eliminaron los casos donde no existía diferencias entre los descriptores. Los valores de las diferencias se utilizarán para entrenar los modelos.

Además, se agregó un descriptor binario para indicar si la molécula corresponde a un ácido o una base. Esta base de datos se definió como "*pKa_descriptors*" y tiene una totalidad de 109 descriptores. Se generó un set adicional de descriptores con *Morgan Circular Fingerprints* [113] de 1024 bits y radio de 3. Con esto cada observación de pKa quedó con 1033 descriptores. Esta base de datos de denominó "*pKa_descriptors_fp*".

En el siguiente cuadro se muestran los descriptores utilizados. Los fingerprints por su estructura no tienen identificación de lo que representa cada bit.

Cuadro XII. Descriptores utilizados a parte del conteo de fragmentos utilizados en la predicción de la constante de acidez.

Descriptor	Descripción
MaxEStateIndex	Valor máximo de índice de estado electropológico. [114]
MinEStateIndex	Valor mínimo de índice de estado electropológico. [114]
MaxAbsEStateIndex	Valor absoluto máximo de índice de estado electropológico. [114]
MinAbsEStateIndex	Valor absoluto mínimo de índice de estado electropológico. [114]
qed	Estimación cuantitativa de la semejanza a drogas. Estimación implementada por *RDkit* [99] basada en el primer método propuesta para estimar el QED basado en características de las moléculas como masa molar, log P, donadores y aceptores de puentes de hidrógeno, enlaces rotables, aromaticidad y área polar superficial. [115]
MolWt	Masa molecular promedio.
HeavyAtomMolWt	Masa molecular al tomar en cuenta solo átomos pesados (ignorando hidrógenos).
ExactMolWt	Masa molecular exacta.
MaxPartialCharge	Máxima carga parcial de la molécula.
MinPartialCharge	Mínima carga parcial de la molécula.
MaxAbsPartialCharge	Máxima carga parcial absoluta de la molécula.
MinAbsPartialCharge	Mínima carga parcial absoluta de la molécula.
FpDensityMorgan1	
FpDensityMorgan2	

FpDensityMorgan3	Índice de densidad de la huella dactilar circular de conectividad de Morgan para la molécula. El número del descriptor corresponde al radio con el cuál se hacen las iteraciones para construir la huella dactilar. [95]
BCUT2D_MWHI BCUT2D_MWLOW BCUT2D_CHGHI BCUT2D_CHGLO BCUT2D_LOGPHI BCUT2D_LOGPLOW BCUT2D_MRHI BCUT2D_MRLOW	Implementación de *RDkit* basada en descriptores que contienen información topológica y atómica para obtener a partir de información 2D propiedades como refractividad molar, log P, masa molar y carga Gasteiger. [116]
BalabanJ	Índice topológico basado en distancia altamente discriminante entre isómeros basado en distancias. [117]
BertzCT	Índice general de complejidad molecular basado en teoría de grafos. [118]
Chi0 Chi1 Chi0n Chi1n Chi2n Chi3n Chi4n Chi0v Chi1v Chi2v Chi3v Chi4v	Índice de conectividad molecular basado en teoría de grafos tomando las moléculas en esqueleto. El número corresponde al orden de las subgrafos utilizadas para calcular los índices: 0 corresponde usando los átomos (vértices), 1 a un enlace, 2 a dos enlaces, 3 a tres enlaces y 4 a cuatro enlaces. Los índices con subíndice "v" corresponde a índices que toman en cuenta los electrones de valencia como elementos del grafo, el resto calcula solo con los dominios enlazantes. [119]
HallKierAlpha	Corresponde a la inclusión de la identidad de los átomos (elemento e hibridación) en el cálculo de los atributos Kappa que se basan solamente en la figura de las moléculas. [119]
Kappa1 Kappa2 Kappa3	Atributo de figura basado en el número de trayectorias mínimas y máximas, de orden *n*, en la molécula. Un orden, *n*, igual 1 corresponde a un enlace, 2 a trayectorias de dos enlaces y 3 a de tres enlaces. [119]

LabuteASA	Área superficial aproximada basada en las áreas de van der Waals de los átomos individuales y corrección de las áreas por los enlaces según sean los elementos involucrados en este. [120]
PEOE_VSA1	Ecualización parcial de la electronegatividad de los orbitales [121] pesada según las contribuciones de cada átomo a el área superficial de van der Waals aproximada. El subíndice del descriptor representa los límites de los intervalos (- ∞, -0.3, -0.25, -0.20, -0.15, -0.10, -0.05, 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, + ∞) [120,122]
PEOE_VSA10	
PEOE_VSA11	
PEOE_VSA12	
PEOE_VSA13	
PEOE_VSA14	
PEOE_VSA2	
PEOE_VSA3	
PEOE_VSA4	
PEOE_VSA5	
PEOE_VSA6	
PEOE_VSA7	
PEOE_VSA8	
PEOE_VSA9	
SMR_VSA1	Refractividad molar computada pesada según las contribuciones de cada átomo a el área superficial de van der Waals aproximada. El subíndice del descriptor representa los límites de los intervalos (- ∞, -0.3, -0.25, -0.20, -0.15, -0.10, -0.05, 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, + ∞) [120,122]
SMR_VSA10	
SMR_VSA2	
SMR_VSA3	
SMR_VSA4	
SMR_VSA5	
SMR_VSA6	
SMR_VSA7	
SMR_VSA9	
SlogP_VSA1	Log P computado pesado según las contribuciones de cada átomo a el área superficial de van der Waals aproximada. El subíndice del descriptor representa los límites de los intervalos (- ∞, -0.3, -0.25, -0.20, -0.15, -0.10, -0.05, 0, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, + ∞) [120,122]
SlogP_VSA10	
SlogP_VSA12	
SlogP_VSA2	
SlogP_VSA3	
SlogP_VSA4	
SlogP_VSA5	
SlogP_VSA6	
SlogP_VSA8	
TPSA	Área polar superficial topológica calculada como suma de contribuciones de fragmentos. [123]
EState_VSA1	Descriptores calculados con índices de estados electrotopológicos [124] y contribuciones del área superficial desarrollado por RDkit. [99]
EState_VSA10	
EState_VSA2	
EState_VSA3	
EState_VSA4	

EState_VSA5	
EState_VSA6	
EState_VSA7	
EState_VSA8	
EState_VSA9	
VSA_EState1	Descriptores calculados con índices de estados electrotopológicos [124] y contribuciones del área superficial desarrollado por RDkit. [99]
VSA_EState10	
VSA_EState2	
VSA_EState3	
VSA_EState4	
VSA_EState5	
VSA_EState6	
VSA_EState7	
VSA_EState8	
VSA_EState9	
FractionCSP3	Fracción de carbonos con hibridación sp^3. [99]
NHOHCount	Número de grupos NH y OH. [99]
NumAliphaticCarbocycles	Número de carbociclos alifáticos. [99]
NumAliphaticHeterocycles	Número de heterociclos alifáticos. [99]
NumAliphaticRings	Número de anillos alifáticos. [99]
NumAromaticCarbocycles	Número de carbociclos aromáticos. [99]
NumAromaticHeterocycles	Número de heterociclos aromáticos. [99]
NumAromaticRings	Número de anillos aromáticos. [99]
NumHAcceptors	Número de grupos aceptores de puentes de hidrógeno. [99]
NumHDonors	Número de grupos donadores de puentes de hidrógeno. [99]
NumRotatableBonds	Número de enlaces rotables. [99]
MolLogP	Log *P* calculado con contribuciones atómicas según elementos y características de este como aromaticidad, alifaticidad, grupo funcional al que pertenece, si es ionizable, etc. [125]
MolMR	Refractividad molar calculada con contribuciones atómicas según elementos y características de este como aromaticidad, alifaticidad, grupo funcional al que pertenece, si es ionizable, etc. [125]

Tipo	Variable categórica binaria, según la molécula sea ácido o base. Se determina a partir de las cargas de la molécula ionizada. Programación propia utilizando *RDkit*. [99]

2.3.3. Modelos

En el Cuadro XIII se muestran los algoritmos y/o arquitecturas de los algoritmos probados para la predicción de la pKa. Para generar los modelos fueron utilizados los mismos módulos de Python ya mencionados utilizados en la predicción de las otras dos propiedades.

Cuadro XIII. Algoritmos utilizados para la construcción de modelos de predicción de pK_a.

Modelo	Descripción
MLR	Regresión lineal múltiple
PLS-2	Mínimos cuadrados parciales con 2 componentes
PLS-3	Mínimos cuadrados parciales con 3 componentes
RF	Bosques aleatorios
SVM	Máquinas de soporte vectorial
kNN	k vecinos más próximos
XGB	Xtreme Gradient Boosting
NN 512 512 512	Red neuronal de 3 capas con 512 neuronas cada una
NN 512 256 128	Red neuronal de 3 capas con 512, 256 y 128 neuronas
NN 256 256 128	Red neuronal de 3 capas con 256, 256 y 128 neuronas

2.3.4. Evaluación y validación

Se construyeron modelos para ambos con solo los descriptores y con los descriptores + el *fingerprint*. Los algoritmos utilizados se muestran en el Cuadro XIII. Se eligieron los dos algoritmos con mejor desempeño, basados en el RMSE, para luego realizar las validaciones cruzadas y externas y finalmente elegir el modelo con mejor desempeño. La validación cruzada se realizó con $k = 10$ y los sets externos fueron los del SAMPL6 [108] y SAMPL7 [109]. En ambas validaciones se calcularon el RMSE, R^2, MSE y MAE para decidir el mejor modelo. Posteriormente se realizó un análisis por separado para ácidos y bases del modelo escogido. Se graficó un Williams plot para determinar el dominio de aplicabilidad.

2.4. Coeficiente de distribución (log D_{pH})

Para la predicción se inició con la predicción del log P_N y Delta de la molécula, a partir de estos se calculó el log P_I realizando la resta log P_I = log P_N – Delta. Con la predicción de la pK_a ya se tienen los tres valores necesarios para predecir el log *D* al pH que se requiera usando la ecuación 24. Por comparación, también se calculó los valores con la ecuación 26 que solo necesita el log P_N y pK_a. En este caso, no existe un set de entrenamiento para log *D*, recopilaron varios valores a diferentes pH de diferentes fuentes [7] [109] [126] para crear una base de datos de prueba ($n = 289$).

Para las predicciones se calcularon las métricas que se utilizaron en la evaluación de los modelos anteriores, RMSE, R^2, MSE y MAE. Además, se graficaron los perfiles para las moléculas que tenían valores de log *D* a varios pH y se realizaron comparaciones con los valores predichos por la herramienta comercial de Chemaxon *JChem for Office*. [127]

Capítulo 3:

Resultados y Discusión

3. *Resultados y discusión*

3.1. Coeficiente de partición neutro (log P_N)

Con el modelo de regresión lineal múltiple se hicieron los primeros análisis para determinar los porcentajes de partición training/test óptimos, así como el tipo de descriptores a utilizar. Al probar las condiciones mostradas en el Cuadro IV, se obtuvieron mejores resultados al utilizar la totalidad de los descriptores y no filtrando según su correlación con el log P_N, es decir con el modelo MLR-1. Esto se puede evidenciar en las siguientes figuras donde el modelo MLR-1 tiene mejor desempeño que los modelos MLR-2, MLR-3, MLR-4 y MLR-5.

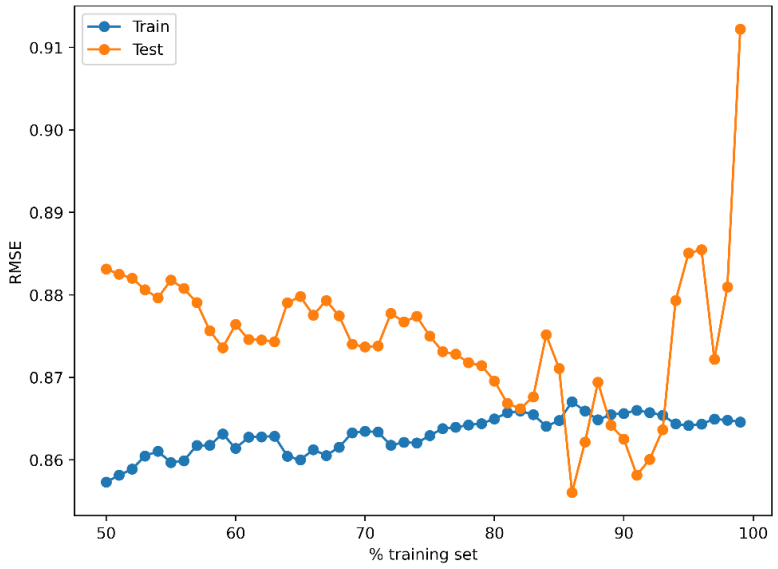


Figura 25. Variación del RMSE con el porcentaje del dataset utilizado para entrenar el modelo MLR-1 en la predicción del coeficiente de partición neutro.

El menor RMSE con el modelo MLR-1 se tiene al entrenarlo con el 86% y es de 0.86. Como se puede observar en la Figura 33, ninguno de los otros modelos con menos cantidad de descriptores obtiene mejores resultados. En el caso de los modelos MLR-6 al MLR-10, que consisten en las mismas condiciones de los MLR-1 al MLR-5 pero con los descriptores regularizados, se obtuvieron los mismos resultados que sin regularizar. Por lo que se opta para el resto de algoritmos siguientes no regularizar.

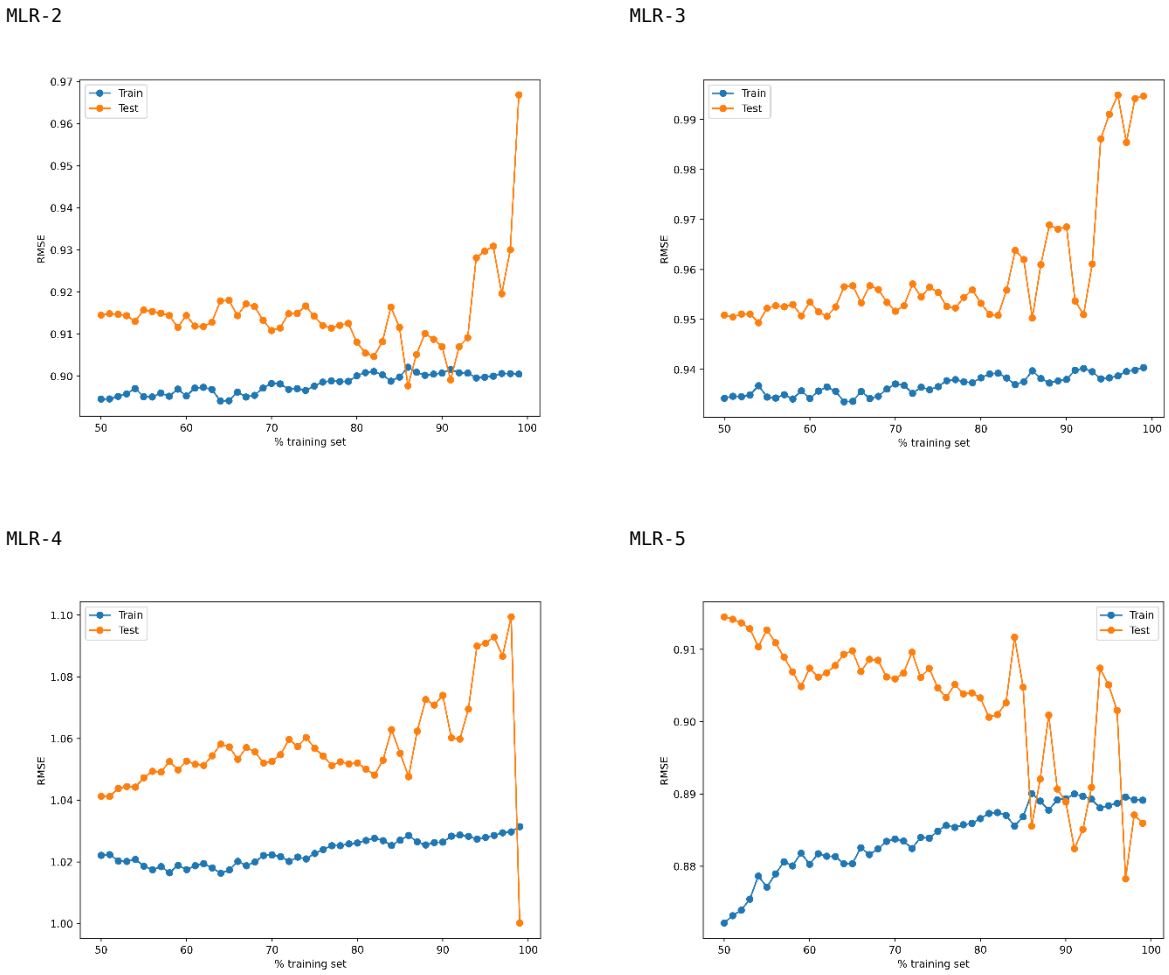


Figura 26. Variación del RMSE con el porcentaje del dataset utilizado para entrenar los modelos MLR-2, MLR-3, MLR-4 y MLR-5 en la predicción del coeficiente de partición neutro.

Los resultados del PLS se muestran en las Figuras 27 y 28, para los modelos con 2 y 3 componentes respectivamente. El menor RMSE para PLS-2 fue de 0.98 al usar el 91% y para PLS-3 fue de 0.94 con 86%. Es evidente que el modelo con tres componentes tiene mejor desempeño que el de dos, pero no mejora el de MLR-1.

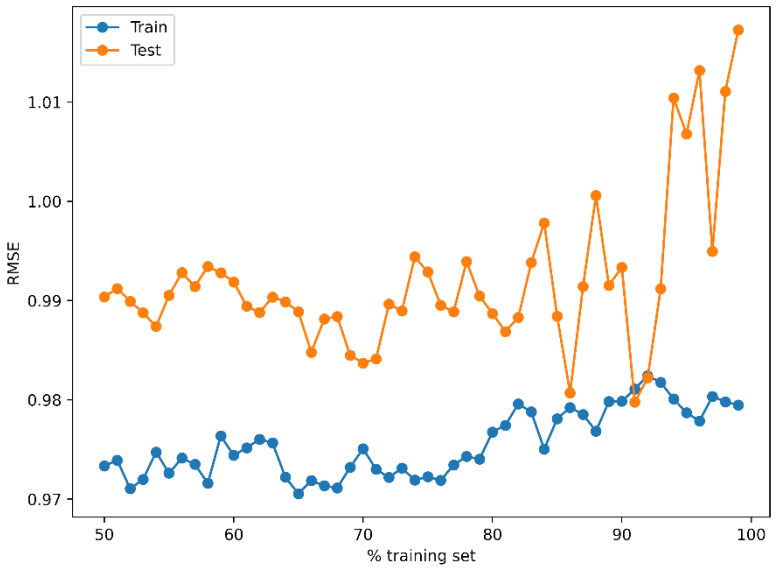


Figura 27. Variación del RMSE con el porcentaje del dataset utilizado para entrenar el modelo PLS-2 en la predicción del coeficiente de partición neutro.

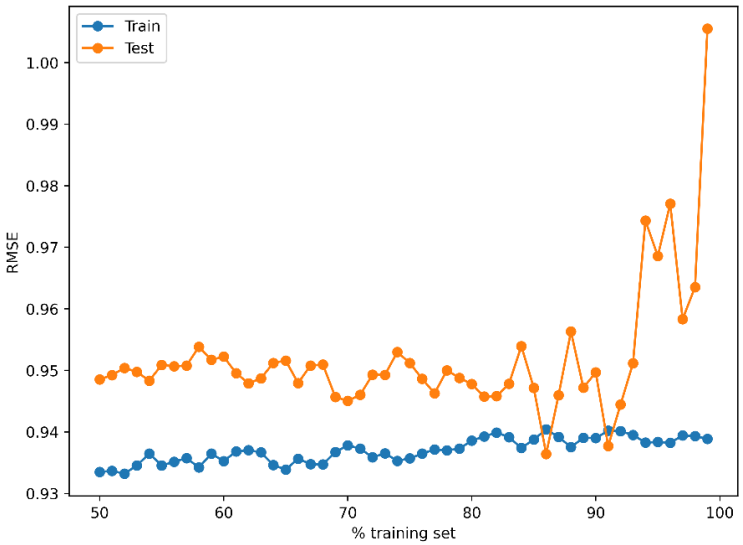


Figura 28. Variación del RMSE con el porcentaje del dataset utilizado para entrenar el modelo PLS-3 en la predicción del coeficiente de partición neutro.

La Figura 29 muestra el resultado para el modelo XGB, se puede observar que la variación del RMSE no es tan abrupta como en algunos de los modelos anteriores, lo que lo hace un buen candidato. El menor RMSE se obtuvo con el 98%, sin embargo, este porcentaje es muy alto ya que deja solo un 2% como test set, lo cual conduciría a una evaluación no representativa del modelo.[128]

Al acomodar por los valores y tomando en cuenta que se deje un porcentaje considerable para el test set, el mejor RMSE es de 0.62 al 87%.

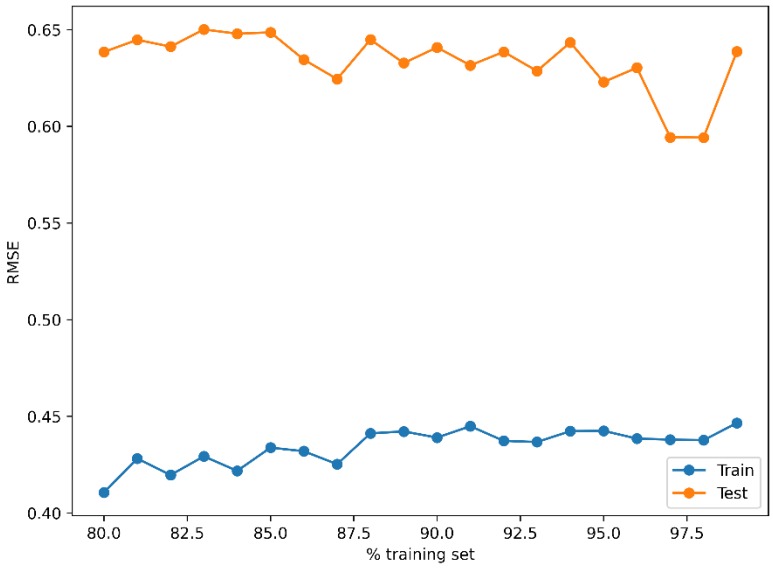


Figura 29. Variación del RMSE con el porcentaje del dataset utilizado para entrenar el modelo XGB en la predicción del coeficiente de partición neutro.

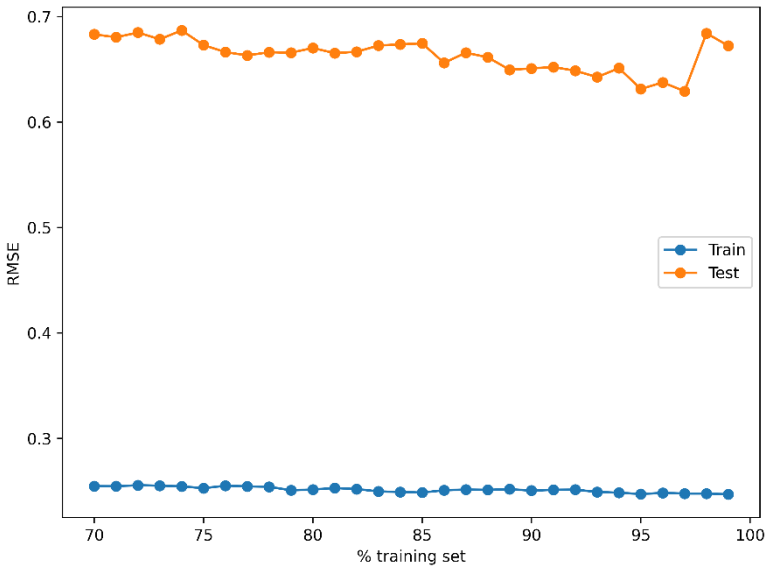


Figura 30. Variación del RMSE con el porcentaje del dataset utilizado para entrenar el modelo RF en la predicción del coeficiente de partición neutro.

En el caso del modelo RF, se obtuvo un menor RMSE en comparación con el MLR-1 pero no menor que el del XGB. Como se puede observar en la Figura 30, el RMSE del modelo no varía abruptamente con el porcentaje del dataset utilizado para el entrenamiento y al tener un bajo RMSE también hace este modelo un buen candidato.

Al evaluar los tres modelos de SVM se obtuvo el menor RMSE con el modelo SVM-1 que corresponde al kernel lineal, el valor fue de 0.90. Los modelos SVM-2 y SVM-3 tuvieron RMSE mayores a la unidad, por lo que no fueron considerado como opciones viables para predecir el log P_N. En el caso de los modelos SVM cabe resaltar que solo se probó a un porcentaje fijo ya que el tiempo de entrenamiento de los modelos es bastante alto.

En el caso de los modelos de NN, se muestra en el Cuadro XIV un resumen de los RMSE más bajos para cada uno de los modelos y el porcentaje correspondiente al cual se obtuvieron. Como se puede apreciar, los mejores modelos corresponden a NN-14, NN-17, NN-19 y NN-20. En la Figura 31 se muestran las variaciones del RMSE con el porcentaje utilizado como training set para estos cuatro modelos. Basándose en el RMSE y que la variación alrededor del punto donde es mínimo, se determina que el modelo que parece el mejor candidato para predecir el log P_N es el NN-20.

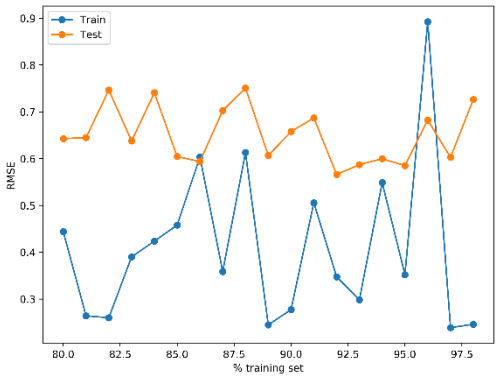
El Cuadro XV muestra a manera de resumen los valores de RMSE más bajos por cada algoritmo utilizado y el porcentaje del dataset utilizado como training respectivo. De este cuadro se puede inferir que los mejores resultados se obtienen al tomar 87 % del dataset como training set y el algoritmo de redes neuronales NN-20. Con base en los resultados del Cuadro XVI se determinó que los dos algoritmos electos para realizar las validaciones y evaluaciones de hiperparámetros (en casos aplicables) corresponden al XGB y NN-20.

Previo a las validaciones se entrenaron los modelos con los sets de entrenamiento conformados con un 87% del dataset y con la misma semilla aleatoria. En el caso del modelo NN-20 se probaron las tres funciones de activación ofrecidas por el módulo y se mantuvo la función de entrenamiento *'adam'*. La métrica de evaluación utilizada fue el RMSE. Con estos resultados el modelo de redes neuronales que fue escogido para continuar con las validaciones fue el de activación con tangente hiperbólica.

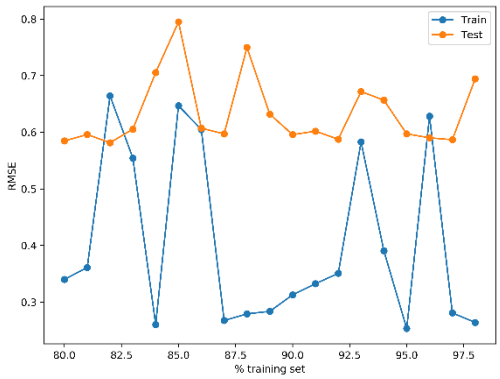
Cuadro XIV. Resumen de RMSE más bajos y porcentajes óptimos utilizados como training set para cada uno de los modelos de NN utilizados en la predicción de log P_N.

Modelo	Arquitectura	Porcentaje training set (%)	RMSE
NN-1	120	91	0.69
NN-2	170	81	0.67
NN-3	400	90	0.69
NN-4	1024, 512	84	0.73
NN-5	120, 10	83	0.64
NN-6	128, 64	90	0.68
NN-7	170, 18	84	0.65
NN-8	256, 128	81	0.75
NN-9	400, 200	81	0.70
NN-10	400, 4	90	0.63
NN-12	500, 100	83	0.62
NN-13	512, 256	93	0.69
NN-14	1024, 512, 256	92	0.57
NN-15	128, 128, 128	88	0.59
NN-16	256, 128, 64	88	0.61
NN-17	256, 256, 256	82	0.58
NN-18	400, 200, 100	89	0.59
NN-19	512, 256, 128	89	0.58
NN-20	512, 512, 512	87	0.58

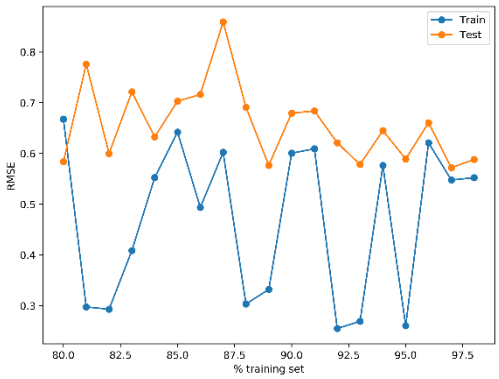
NN-14



NN-17



NN-19



NN-20

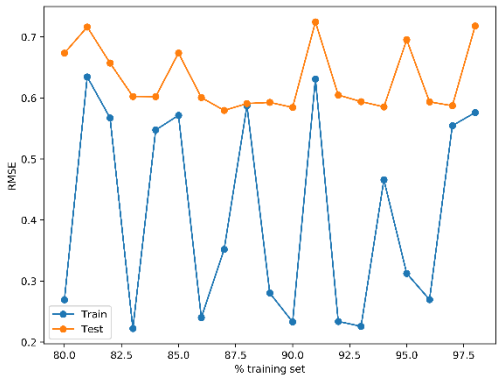


Figura 31. Variación del RMSE con el porcentaje del set de datos utilizado como set de entrenamiento de modelos de redes neuronales (NN-14, NN-17, NN-19 y NN-20) de perceptrón múltiple con diferentes arquitecturas en la predicción del coeficiente de partición neutro.

Cuadro XV. Resumen de RMSE y porcentajes utilizados como training set por algoritmo en la predicción del coeficiente de partición neutro.

Modelo	Porcentaje training set (%)	RMSE
MLR-1	86	0.86
PLS-3	86	0.94
XGB	87	0.62
RF	89	0.65
SVM-1	87*	0.90
NN-20	87	0.58

Cuadro XVI. RMSE de los mejores modelos de predicción de log P_N para el mismo test set. Para los modelos de redes neuronales (NN) se prueban diferentes funciones de activación.

Modelo	RMSE
XGB	0.62
NN-20 logistic	0.59
NN-20 tanh	0.56
NN-20 relu	0.64

Para comparar los dos mejores modelos, XGB y NN-20 tanh, se calcularon los parámetros de desempeño (R^2, RMSE, MSE y MAE) para tanto el set de entrenamiento como para el set de prueba de log P_N. Como se muestra en el Cuadro XVII el modelo de redes neuronales tiene un coeficiente de determinación más cercano a uno y los tres tipos de errores calculados son menores. En el Cuadro XVII también se muestran las métricas de desempeño para las predicciones del software de ChemAxon, estas son peores que ambos modelos propuestos. En las Figuras 32 y 33 se muestran las regresiones de los datos predichos por el modelo XGB y NN-20 tanh respectivamente, con los valores experimentales.

Cuadro XVII. Parámetros estadísticos de la evaluación del desempeño de los dos mejores modelos de predicción de log P_N y comparación con el modelo de licencia ChemAxon.

Set / Modelo	Training set				Test set			
	R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB	0.94	0.42	0.18	0.32	0.88	0.62	0.39	0.44
NN-20 tanh	0.98	0.33	0.11	0.25	0.91	0.56	0.32	0.40
ChemAxon	0.84	0.73	0.54	0.51	0.82	0.76	0.58	0.53

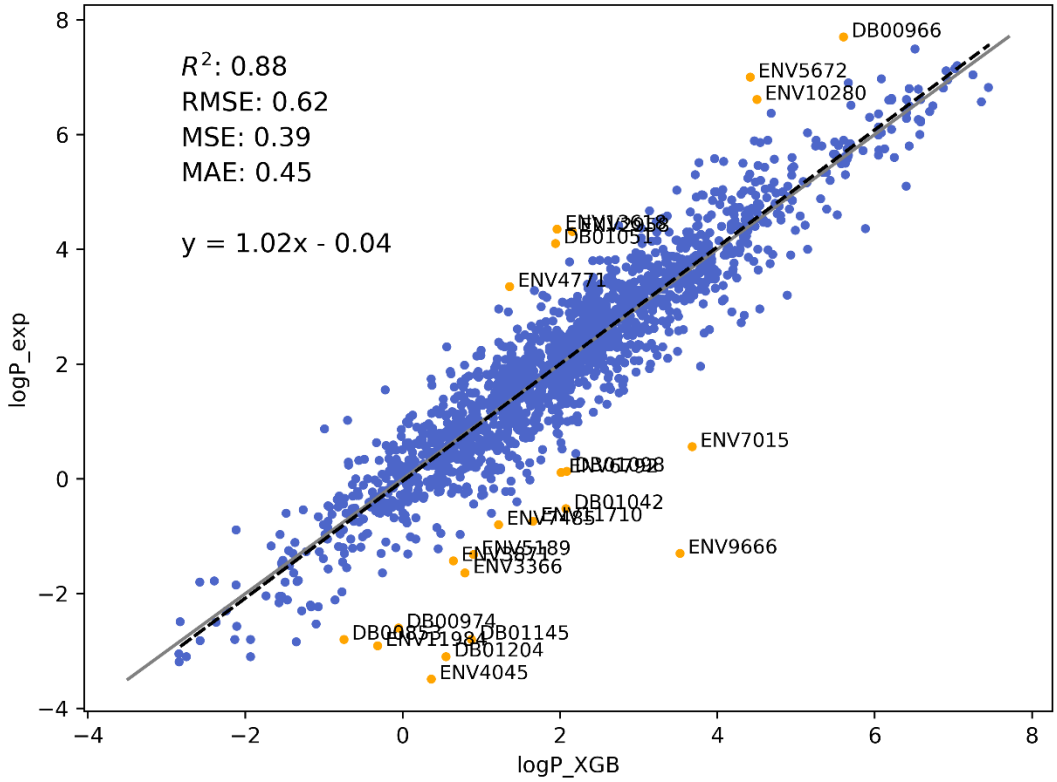


Figura 32. Regresión de los valores de log P_N predichos con el modelo XGB con los valores experimentales del test set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra punteada corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

Al analizar en ambas regresiones se nota que ambas tienen la misma cantidad de puntos fuera del rango ± 3·RMSE. En el caso del XGB estos puntos se encuentran más distribuidos a valores extremos del coeficiente de partición neutro, mientras que en el NN-20 tanh se encuentran más distribuidos en todo el rango. El comportamiento del XGB es más esperado ya que en los valores extremos hay menos cantidad de datos, como se evidencia en la Figura 23. Esto representa una ventaja del XGB ya que no tantos compuestos tienen valores extremos del coeficiente de partición.

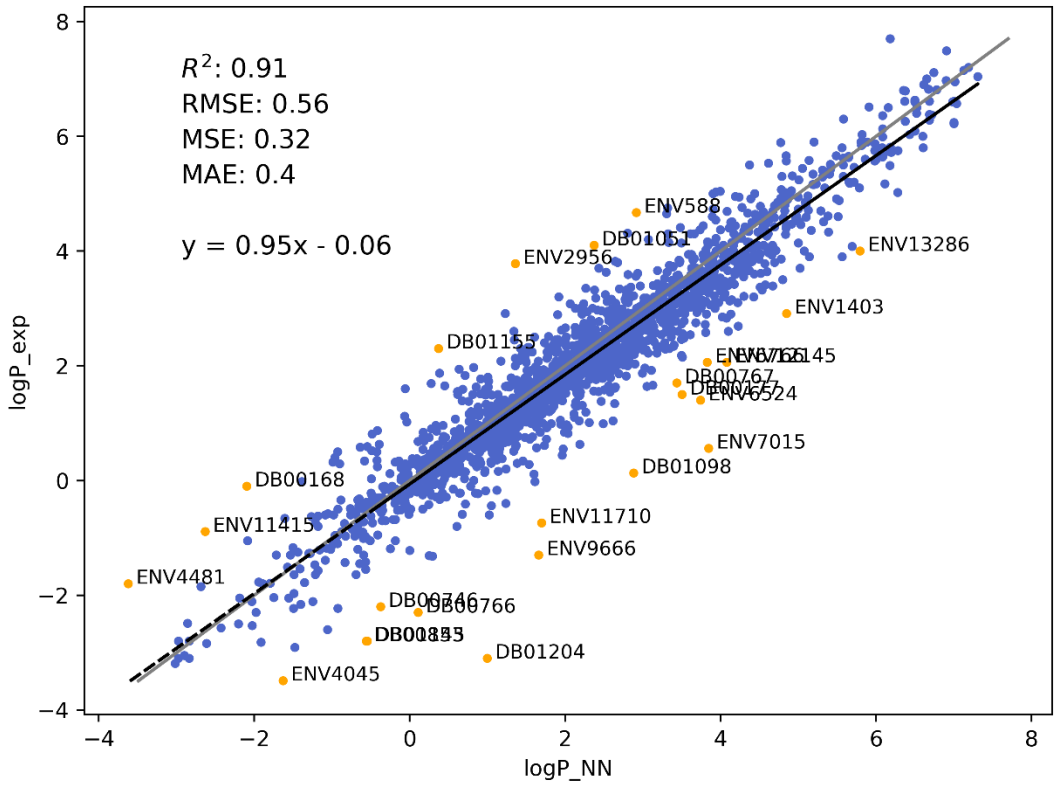


Figura 33. Regresión de los valores de log P_N predichos con el modelo NN-20 tanh con los valores experimentales del test set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra punteada corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

En las Figuras 34 y 35, se muestran las regresiones de los valores predichos y valores experimentales de log P_N para el set de entrenamiento. Como se puede observar la regresión es más cercana a la función identidad, como es de esperar ya que con estas mismas moléculas se entrenaron los modelos. Se puede apreciar que las predicciones con NN-20 tanh se encuentran más cercanas la función identidad, lo que es esperable ya que los errores de este modelo son menores que los del modelo XGB, para el training set.

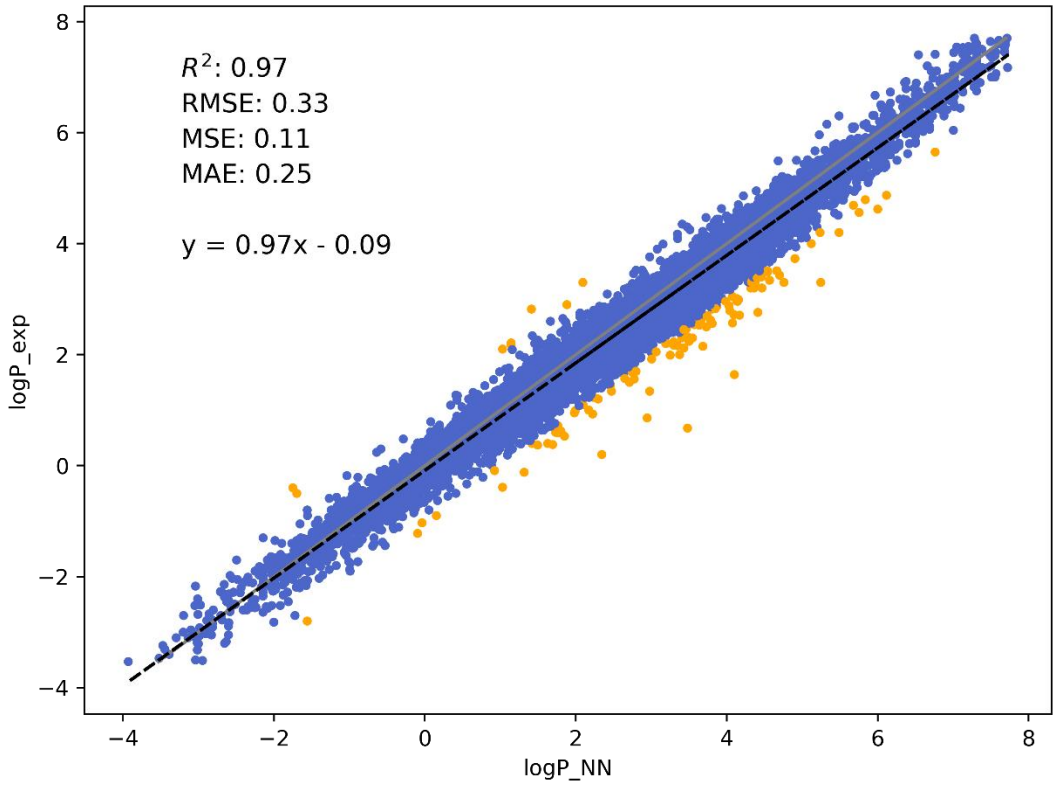


Figura 34. Regresión de los valores de log P_N predichos con el modelo NN-20 tanh con los valores experimentales del training set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra punteada corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

Una posible causa de los errores de los modelos es el no considerar los tautómeros. Usualmente los tautómeros tienen diferente hidrofobicidad entre sí, el ignorar la presencia de tautomerismo lleva a que los modelos computacionales tengan errores en la predicción de propiedades. Investigaciones han comparado como modelos computaciones predicen diferentes coeficientes de partición para los tautómeros de una misma molécula. [129]

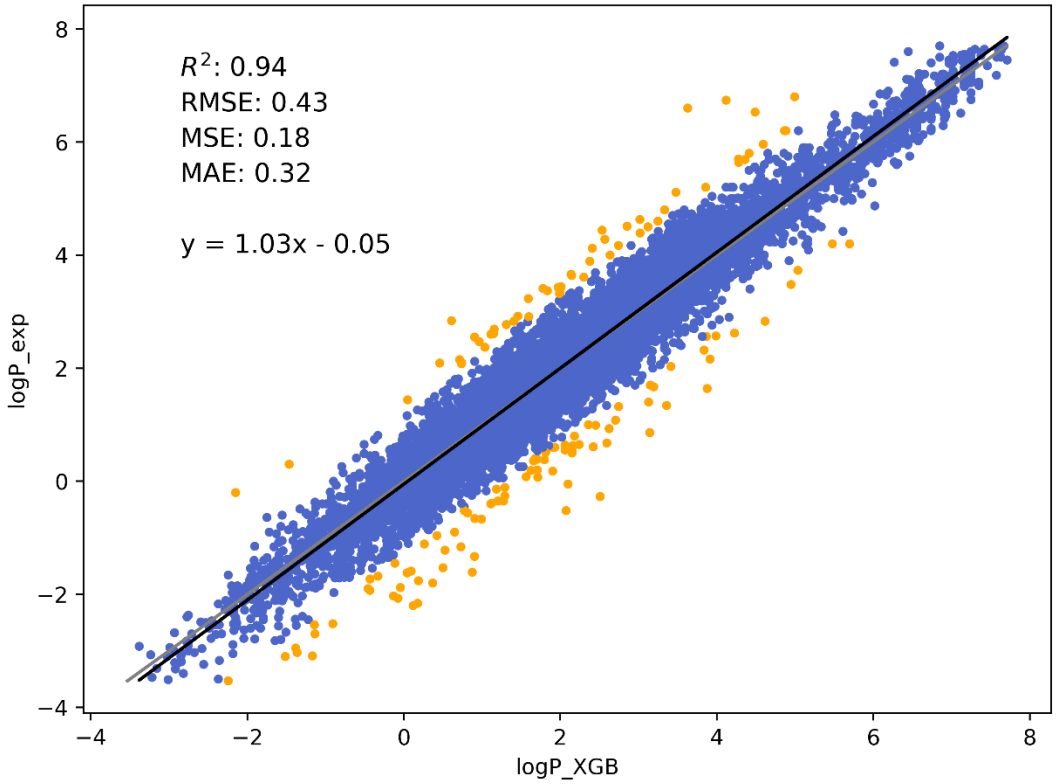


Figura 35. Regresión de los valores de log P_N predichos con el modelo NN-20 tanh con los valores experimentales del training set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra punteada corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

Cabe destacar que entre los dos modelos existen moléculas en común que la predicción se sale de los rangos ya especificados. En la Figura 36 se muestran las estructuras de estos compuestos. Es importante analizar estos compuestos a la luz del dominio de aplicabilidad de los modelos que se presentará posteriormente.

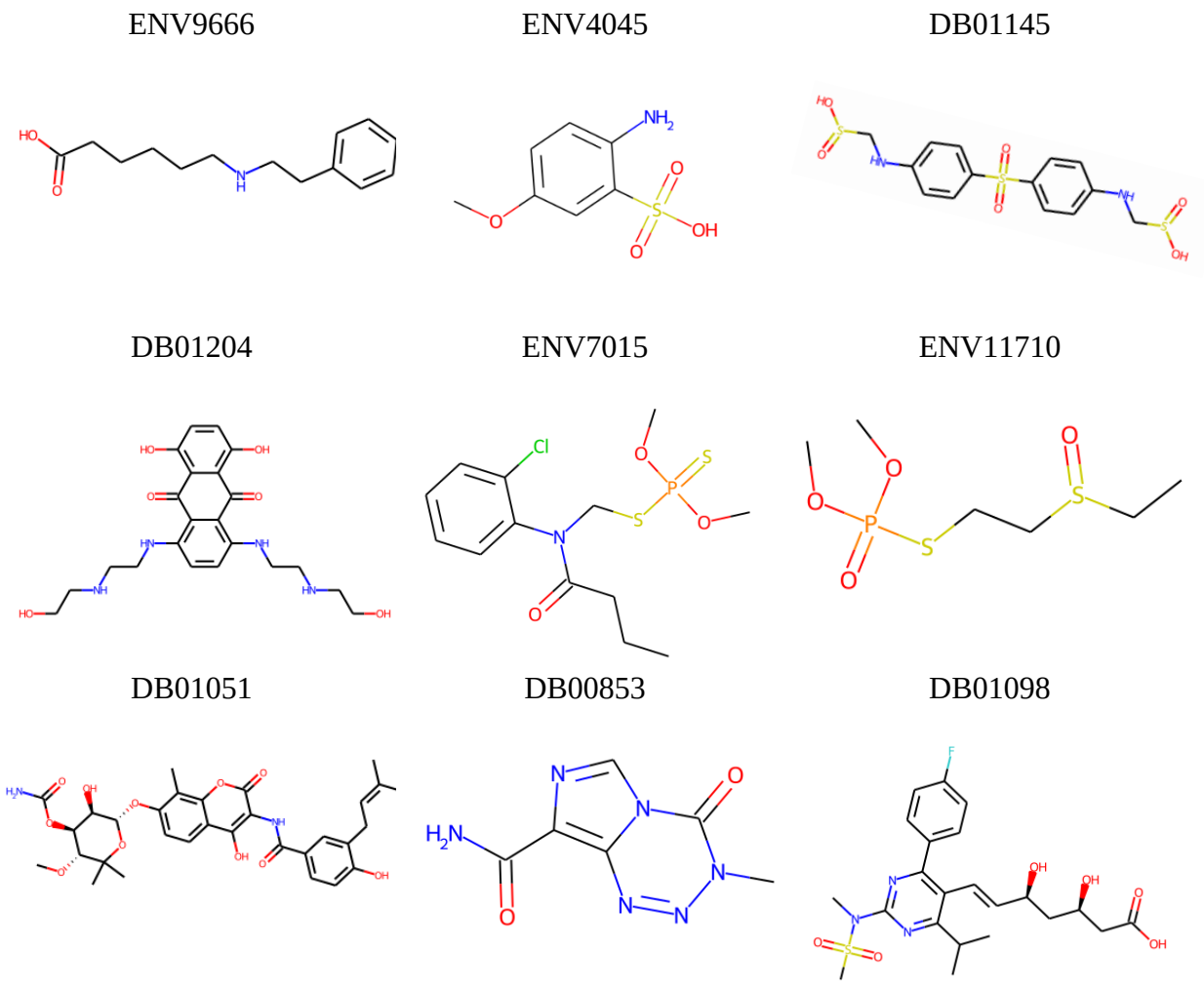


Figura 36. Estructura de las moléculas cuya predicción de log P_N se desvía más de 3·RMSE en ambos modelos XGB y NN-20 tanh.

En las validaciones cruzadas de *k*-iteraciones se utilizó $k = 10$, de manera que los resultados de las métricas, mostradas en el Cuadro XVIII, corresponden al promedio de los diferentes 10 test sets en cada iteración. De nuevo, el modelo con mejores métricas corresponde al de redes neuronales, la comparación es bastante semejante a la de la evaluación general simple de los modelos.

Cuadro XVIII. Parámetros estadísticos de la evaluación del desempeño de los dos mejores modelos de predicción de log P_N.

Set / Modelo	Tiempo entrenam. / min	Training set				Test set			
		R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB	0.98	0.94	0.44	0.19	0.32	0.88	0.63	0.39	0.45
NN-20	222.4	0.96	0.32	0.11	0.24	0.90	0.56	0.21	0.40

Para complementar la validación cruzada, se realizó una validación con dos sets externos, SAMPL6 y SAMPL7. Los resultados de las métricas de desempeño se muestran en el Cuadro X, como se puede observar el desempeño del modelo XGB es mejor que el del NN-20, contrario a las evaluaciones realizadas previamente. En el caso del set externo SAMPL6, el RMSE del modelo XGB fue 0.35 unidades de log P_N menor y para el SAMPL7 0.30. Esto representa diferencias de 41% y 26% respectivamente, lo cual se considera significativo. Esta diferencia es mucho mayor que la de las evaluaciones en los test sets donde el modelo NN-20 tiene un RMSE mejor en 0.07 unidades, esto solamente representa una diferencia del 11%, lo cual no se considera significativo. [130] Por esta razón el modelo que se seleccionó para los posteriores cálculos del coeficiente de distribución es el XGB.

Cuadro XIX. Parámetros estadísticos de la evaluación del desempeño de los dos mejores modelos de predicción de log P_N en la validación con dos sets externos SAMPL6 y SAMPL7.

Set / Modelo	SAMPL6				SAMPL7			
	R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB	0.58	0.50	0.25	0.42	0.18	0.84	0.70	0.66
NN-20	0.50	0.85	0.72	0.74	0.11	1.14	1.29	0.93

En la Figura 37 se muestra el gráfico de William para evaluar el dominio de aplicabilidad. Los puntos fuera de los límites corresponden a puntos de moléculas que está fuera del dominio del modelo, es decir predicciones que no se pueden realizar con seguridad a como está construido el modelo. Para el modelo electo, XGB, el 96.00% del training set y el 96.00% del test set se encuentra dentro del dominio de aplicabilidad del modelo. Este porcentaje es bastante favorable considerando el tamaño de la base datos ($n = 14041$), lo que quiere decir que se cubre un espacio químico amplio.

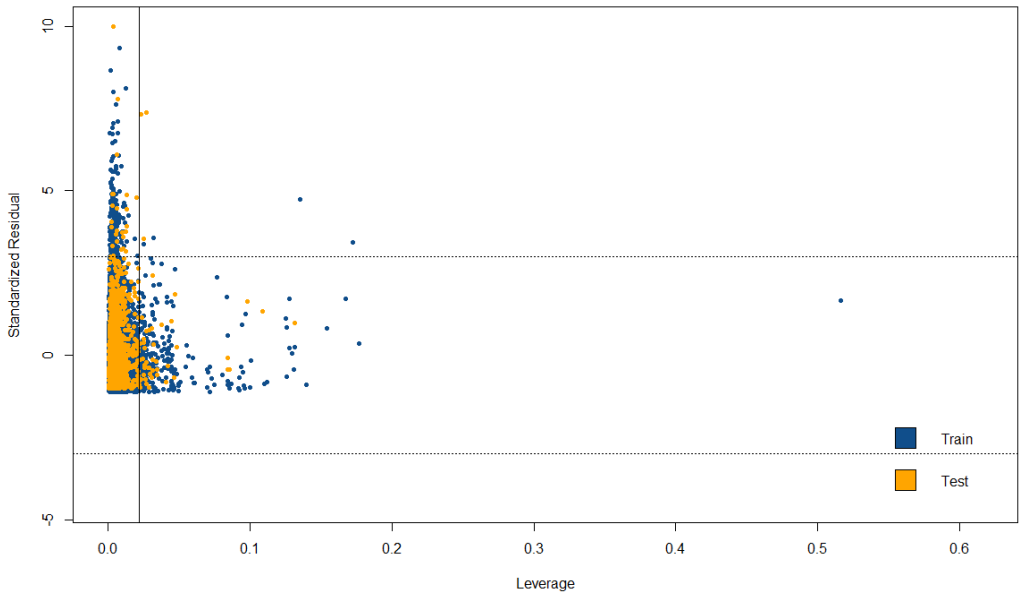


Figura 37. Gráfico de William para la evaluación del dominio de aplicabilidad del modelo XGB para la predicción del coeficiente de partición neutro. En azul se muestran los puntos del training set y en naranja el test set. Los límites de residuos estándar son tres desviaciones estándar y el valor límite de apalancamiento *(h*)* es de 0.022.

Al examinar las moléculas que se encuentran fuera del dominio de aplicabilidad con las moléculas que presentan mayores errores, mostradas en la Figura 36, se nota que la totalidad de estas se encontraban fuera del dominio de aplicabilidad del modelo. Incluso al evaluar todas aquellas moléculas del test set que tuvieron un error mayor a 3·RMSE con el modelo XGB, también se encontró que estaban fuera del dominio. Esto nos indica que la predicción del log P_N para moléculas que se encuentran fuera del dominio de aplicabilidad implica errores importantes. Conclusiones y recomendaciones

3.2. Coeficiente de partición iónico (log P_I)

Dada la poca cantidad de datos para log P_I, se considerarán modelos exitosos solo aquellos con un set de entrenamiento menor al 85% del dataset inicial, para asegurar que el test set no sea muy pequeño. Cabe recalcar de nuevo, que los errores y predicciones analizadas en esta sección no son directamente de log P_I, si no de la diferencia entre el log P_N y log P_I.

Se probaron seis diferentes tipos de descriptores, detallados anteriormente, con seis algoritmos de ML. En el Cuadro XX se muestran los RMSE para estas 36 posibles combinaciones. Los

mejores resultados se obtuvieron con el RF en la condición I-6 y con XGB con la condición I-5. En las Figuras 38 y 39 se analiza la variación del RMSE de los valores predichos, con el porcentaje del dataset utilizado para el entrenamiento. Se puede notar que el modelo XGB-I-5 tiene un comportamiento más estable luego del 88%, para el RF-I-5 es un menos estable. Es deseable que sea constante descartar de que no sea simplemente una buena combinación en el test set. Se nota que ambas condiciones I-5 e I-6 son las que tienen mayor cantidad de descriptores, incluyen los de las moléculas ionizadas, se puede inferir que son de importancia en la predicción.

En el Cuadro XXI se muestras las métricas para evaluar el desempeño en la predicción de la diferencia entre el coeficiente de distribución neutro y el iónico. Un punto alto de ambos modelos es que presentan un mejor desempeño que *JChem* de *ChemAxon.* [127] Se puede notar que en caso del modelo XGB-I-5 para el set de entrenamiento tiene excelentes resultados, pero para el set de prueba resultados peores que el modelo RF-I-6. Esto último es importante de notar, ya que la utilidad pensada del modelo es que pueda predecirlo para compuestos novedosos que sean prospectos, donde un buen desempeño solo en el training set no es suficiente.

Cuadro XX. Resultados del RMSE por condiciones y algoritmo en la predicción del Delta. Entre paréntesis se muestra el porcentaje del set utilizado como training set con que se obtuvo ese RMSE. Se destacan en negrita los modelos con mejor desempeño.

	MLR	PLS	RF	SVM	NN	XGB
I-1	2.32 (88)	0.98 (80)	0.81 (78)	0.99 (85)	1.02 (80)	0.66 (83)
I-2	1.19 (89)	1.00 (80)	0.82 (83)	0.89 (84)	0.71 (84)	0.72 (83)
I-3	1.42 (78)	1.01 (80)	1.02 (84)	0.76 (80)	0.98 (80)	0.88 (83)
I-4	1.36 (78)	1.05 (80)	0.95 (80)	0.77 (80)	1.13 (83)	0.95 (83)
I-5	3.06 (81)	0.82 (89)	0.56 (90)	0.70 (90)	1.13 (86)	**0.53 (88)**
I-6	2.30 (81)	0.79 (89)	**0.51 (90)**	0.67 (90)	0.95 (89)	0.69 (88)
I-7	1.36 (83)	0.89 (89)	0.55 (90)	0.68 (90)	0.81 (90)	0.72 (90)
I-8	1.34 (80)	0.87 (89)	0.58 (86)	0.82 (90)	0.77 (90)	0.58 (89)

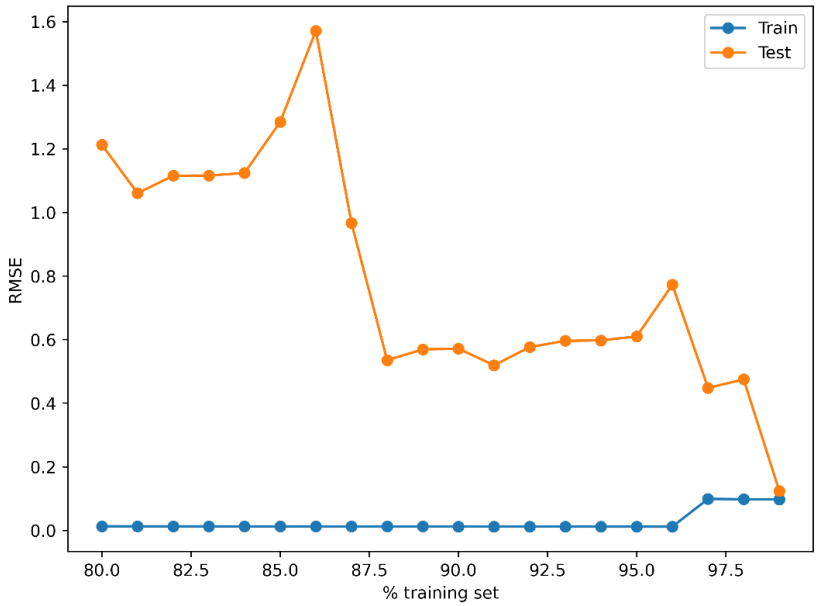


Figura 38. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo XGB-I-5 para la predicción del Delta (log P_N y log P_I).

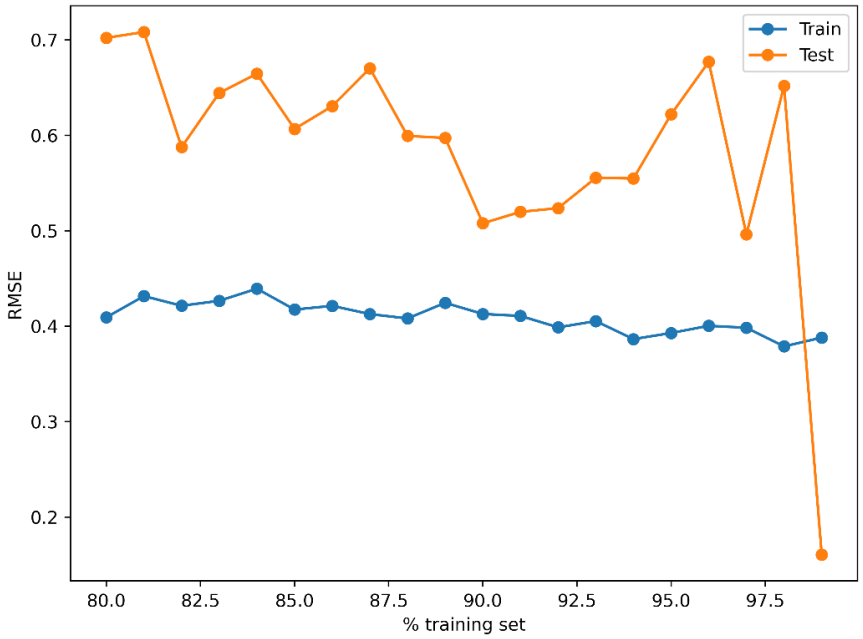


Figura 39. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo RF-I-6 para la predicción del Delta (log P_N y log P_I).

Cuadro XXI. Parámetros estadísticos de la evaluación del desempeño de los dos mejores modelos de predicción del Delta y comparación con el software de licencia ChemAxon.

Set / Modelo	Training set				Test set			
	R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB-I-5	0.99	0.04	0.00	0.01	0.83	0.54	0.29	0.43
RF-I-6	0.95	0.41	0.17	0.29	0.88	0.51	0.26	0.40
ChemAxon	0.25	1.59	2.53	1.11	0.82	0.75	0.57	0.63

De una manera más clara en la Figura 41 se puede notar donde casi todos los puntos de la regresión de valores predichos por el modelo XGB-I-5 contra experimentales caen sobre la función identidad, pero en la Figura 40 que corresponde al test set está más dispersos. Las Figuras 42 y 43 contienen las regresiones para el modelo RF-I-6, en la del training set se nota un punto que tiene un error significativo, el compuesto MON69 que corresponde al tamoxifeno. Posteriormente se notará que este compuesto se encuentra fuera del dominio de aplicabilidad del modelo, una de las posibles razones es que el Delta de este compuesto es el más alto de todo el dataset y no hay observaciones que describan apropiadamente valores tan altos. La estructura del tamoxifeno ionizado se muestra en la Figura 47.

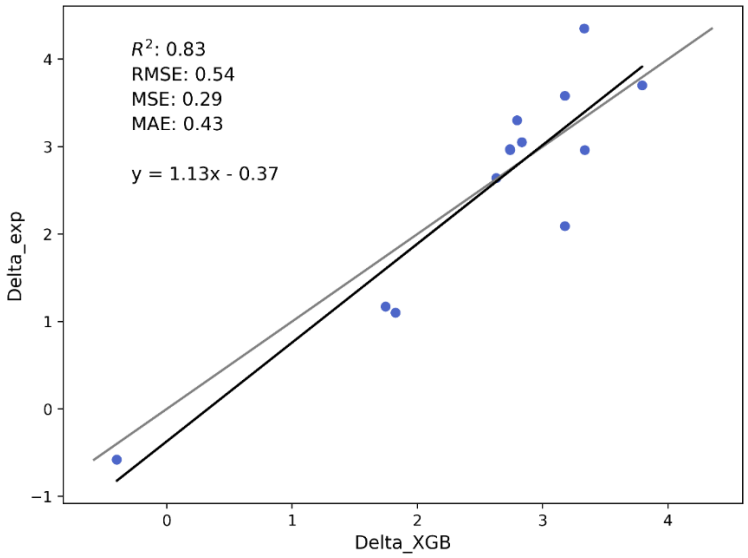


Figura 40. Regresión de los valores de Delta predichos con el modelo XGB-I-5 con los valores experimentales del test set. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

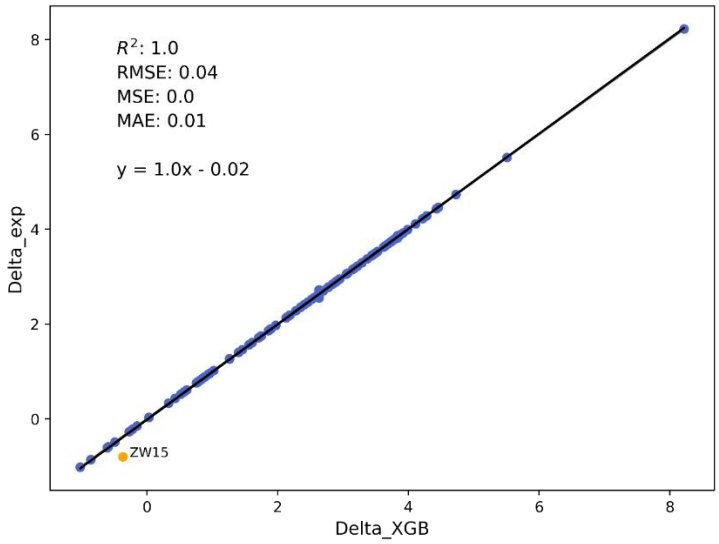


Figura 41. Regresión de los valores de Delta predichos con el modelo XGB-I-5 con los valores experimentales del training set. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

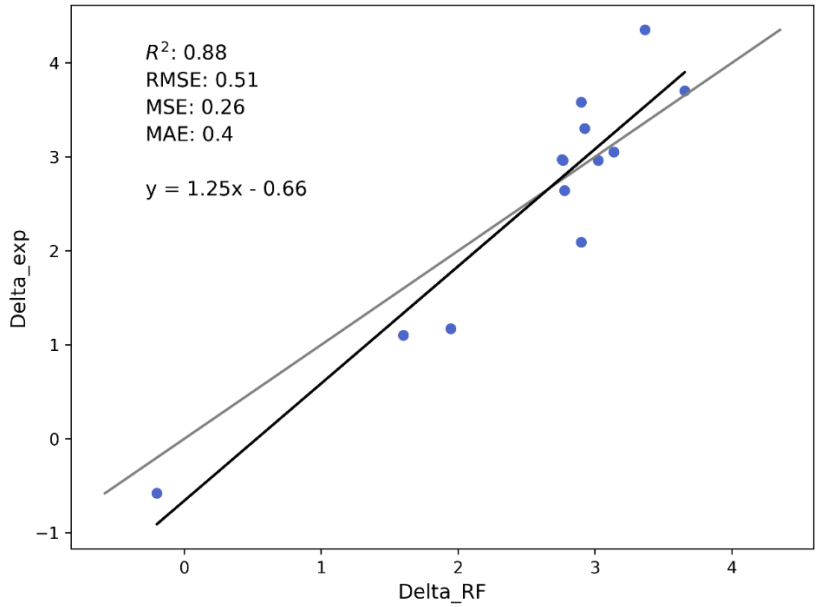


Figura 42. Regresión de los valores de Delta predichos con el modelo RF-I-6 con los valores experimentales del test set. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

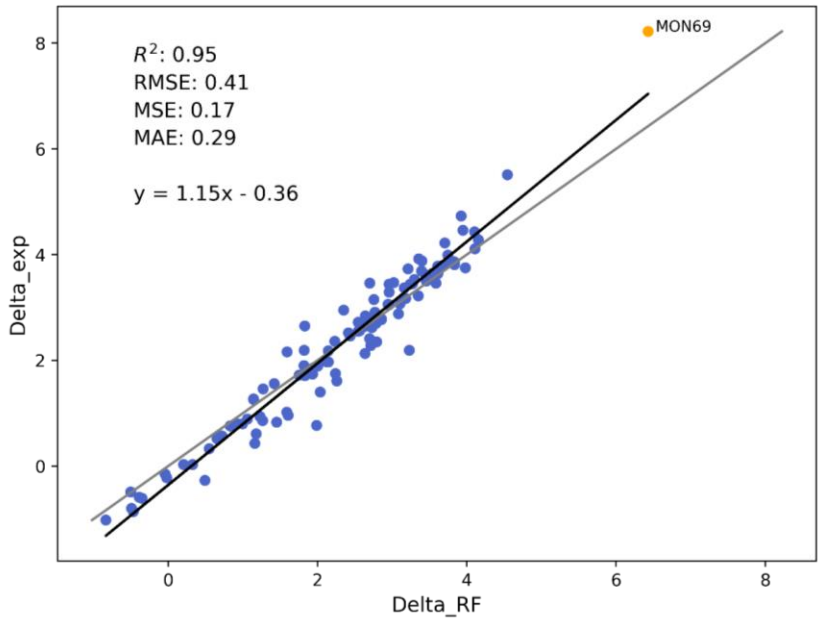


Figura 43. Regresión de los valores de Delta predichos con el modelo RF-I-6 con los valores experimentales del training set. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

La evidencia mostrada al momento indica que el modelo a elegir es el RF-I-l ya que tiene mejor desempeño para el test set. Para corroborar esto se realizó una validación cruzada de *k*-iteraciones con $k = 10$, los resultados las métricas de esta validación se muestran en el Cuadro XXII. En la validación se notó el mismo patrón que en los resultados obtenidos previamente, el modelo XGB-I-5 tienen mejor desempeño en el training set y el RF-I-6 mejor en el test set.

Cuadro XXII. Parámetros estadísticos de desempeño de la validación cruzada de *k*-iteraciones, con $k = 10$, de los dos mejores modelos de predicción del Delta.

Set / Modelo	Tiempo entrenam. / min	Training set				Test set			
		R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB-I-5	0.02	1.0	0.09	0.01	0.02	0.43	1.07	1.34	0.77
RF-I-6	0.15	0.93	0.40	0.16	0.28	0.55	0.95	1.03	0.71

A parte de la validación cruzada también se realizó una validación con un set externo, las métricas de desempeño se muestran en el Cuadro XXII. En este caso del RMSE, MSE y MAE fue menor con el algoritmo XGBoost, pero este tiene un menor coeficiente de determinación. A pesar de que no hay consenso en la comunidad de ML sobre si tienen más importancia los errores o el R^2, hay autores que mencionan que el último da más información sobre la exactitud del modelo. [131] Además, al no ser tan grandes las diferencias entre los errores, se optó por decidir seleccionar el modelo RF-I-6.

Cuadro XXIII. Parámetros estadísticos de desempeño de la validación externa de los dos mejores modelos de predicción del Delta.

Set / Modelo	External set			
	R^2	RMSE	MSE	MAE
XGB-I-5	0.65	0.56	0.32	0.43
RF-I-6	0.78	0.57	0.33	0.50

En las Figuras 44 y 45 se muestran las regresiones de los valores predichos por ambos modelos y los valores experimentales. A pesar de que se aparenta que los valores predichos están lejos de los experimentales, si se nota en la escala la mayoría de puntos tienen errores menores a la unidad. En la Figura 45 se puede notar que la gran mayoría de errores son negativos. Si se comparan las ecuaciones de las rectas de ajuste, la pendiente de la del modelo con el algoritmo RF tiene un valor más cercano a uno y un intercepto más cercano a cero que el modelo con XGB. Lo que refuerza la evidencia para seleccionar el modelo RF-I-6.

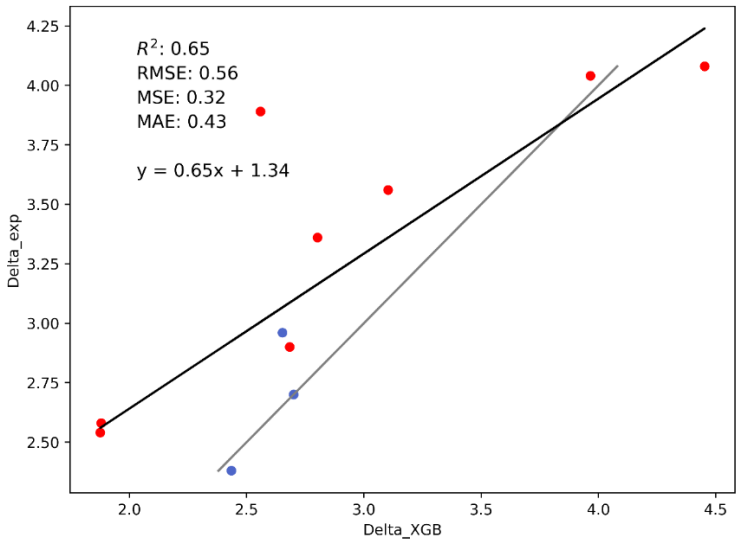


Figura 44. Regresión de los valores de Delta predichos con el modelo XGB-I-5 con los valores experimentales del set externo.[111] En rojos se muestran los ácidos y en azul las bases. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

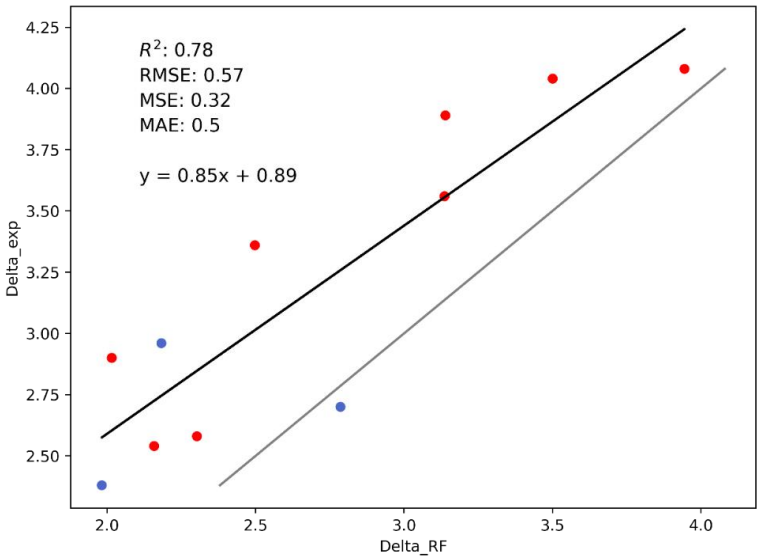


Figura 45. Regresión de los valores de Delta predichos con el modelo RF-I-6 con los valores experimentales del set externo.[111] En rojos se muestran los ácidos y en azul las bases. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

Al realizar el análisis del dominio de aplicabilidad de obtuvo en gráfico de Williams mostrado en la Figura 46. Como se pueden notar existen dos puntos que está fuera de los límites del dominio de aplicabilidad, ya que sus residuales estandarizados son mayores al límite superior de 3. Con esto significa que un 98.11% del training set y un 100% del test set se encuentra dentro del dominio de aplicabilidad. El valor de h* en este caso es de 2.07, pero ningún valor se sale de este límite.

En las Figuras 47 se muestran las estructuras de los dos compuestos que se salen del dominio de aplicabilidad: el tamoxifeno y el clorumbacil. Como se mencionó previamente el tamoxifeno es el compuesto del set con el mayor valor de Delta y no hay otros valores en rangos similares, lo cual dificulta que el modelo esté bien entrenado para predecir Deltas altos. En el caso del clorumbacil no hay una razón evidente que justifique el error tan alto, lo cual es una desventaja de los modelos de ML.

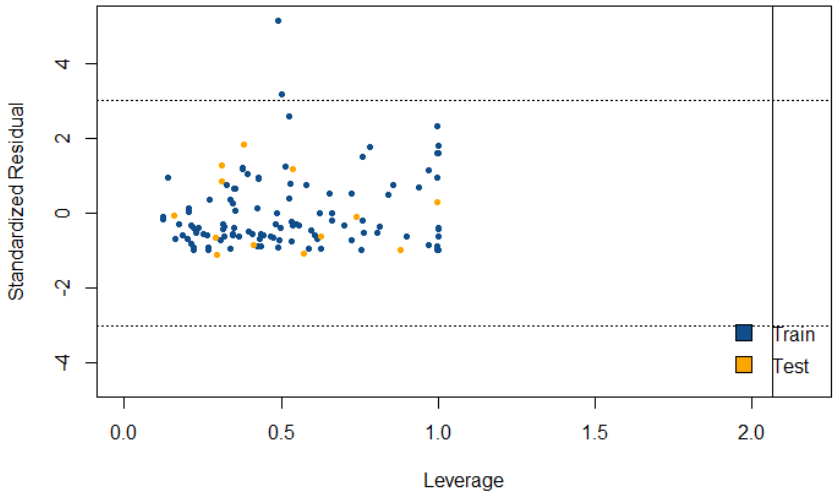


Figura 46. Gráfico de Williams para la evaluación del dominio de aplicabilidad del modelo RF-I-6 para la predicción del Delta. En azul se muestran los puntos del training set y en naranja el test set. Los límites de residuos estándar son tres desviaciones estándar y el valor límite de apalancamiento *(h*)* es de 2.7.

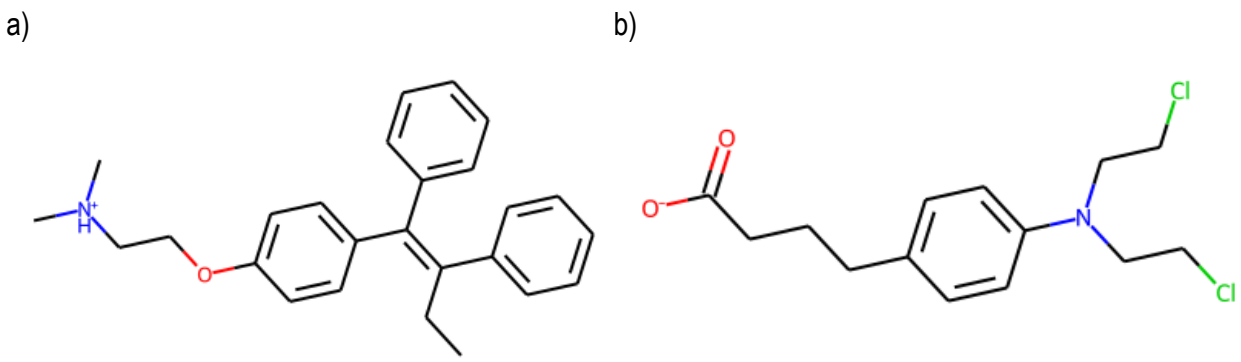


Figura 47. Estructura de los compuestos que se salen del dominio de aplicabilidad del modelo RF-I-6: a) tamoxifeno y b) clorumbacil ionizados.

3.3. Constante de acidez (pK_a)

En el Cuadro XXIV se muestran los RMSE para el test set para las 20 combinaciones posibles entre algoritmos de ML y las dos condiciones de descriptores usadas. En ambas condiciones se utilizan los mismos descriptores que corresponden a las diferencias entre los valores entre la molécula neutra y la molécula ionizad, aparte de esto la segunda condición incluye una huella dactilar molecular de la molécula neutra. Los modelos con mejor desempeño preliminar corresponden al de RF y XGB. Para el RF se obtuvo el mismo RMSE en ambas condiciones de descriptores, para el XGB se obtiene mejores resultados al incluir el *fingerprint*. Se seleccionaron estos dos modelos, en ambas condiciones para evaluaciones posteriores.

En las Figuras 48-51 se muestran las variaciones del RMSE con el porcentaje utilizado como set de entrenamiento. En todas se nota que el RMSE para el training set es mucho más bajo que el del test set, como es de esperar. En todos los casos no se ven variaciones abruptas en el RMSE, se observa una tendencia constante, lo cual es deseable en los modelos. Por lo tanto, se continuará con los cuatro modelos para realizar validaciones y evaluaciones posteriores. Algo relevante de notar es que para tres de los cuatro modelos el porcentaje del data set utilizado para el entrenamiento es del 88%. Por lo tanto, para igualar las condiciones y que sean comparables los cuatro modelos se utilizó este porcentaje para entrenarlos.

Cuadro XXIV. Resultados del RMSE por condiciones y algoritmo en la predicción de la pK_a. Entre paréntesis se muestra el porcentaje del dataset utilizado como training set con que se obtuvo ese RMSE. Se destacan en negrita los modelos con mejor desempeño.

Modelo	Dataset	
	pKa_descriptors	pKa_descriptors_fp
MLR	1.26 (84)	1.24 (88)
PLS-2	1.61 (90)	1.48 (87)
PLS-3	1.51 (87)	1.35 (87)
RF	**0.85 (88)**	**0.85 (88)**
SVM	1.29 (84)	1.21 (88)
kNN	1.25 (83)	1.26 (83)
XGB	**0.84 (86)**	**0.78 (88)**
NN 512 512 512	1.08 (88)	0.96 (88)
NN 512 256 128	1.12 (83)	0.93 (88)
NN 256 256 128	1.12 (84)	0.99 (88)

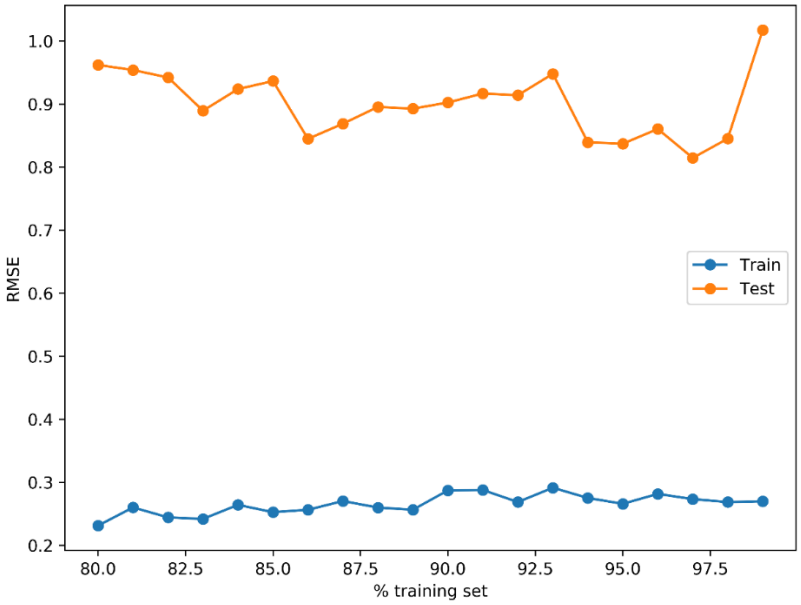


Figura 48. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo XGB- pKa_descriptors para la predicción de la pK_a.

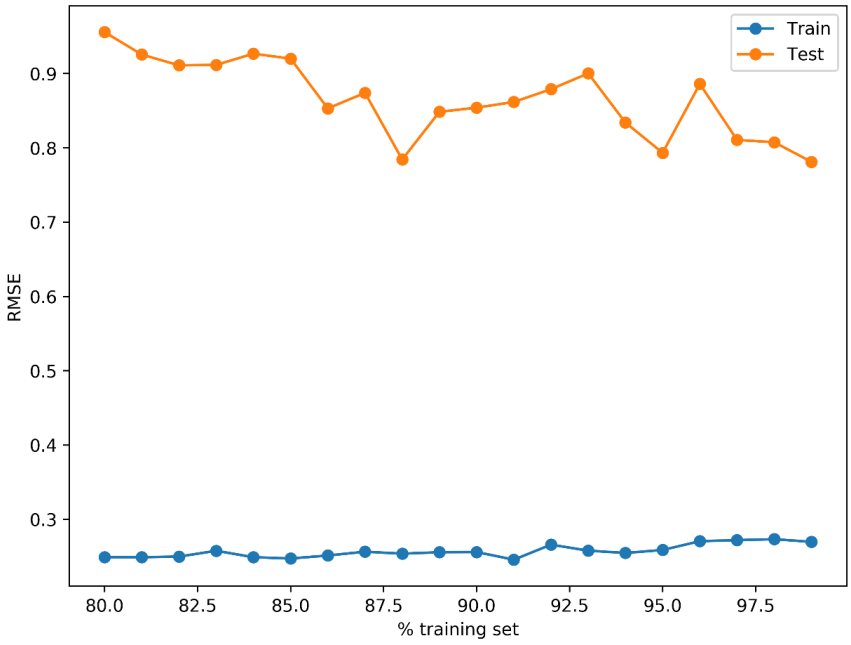


Figura 49. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo XGB- pKa_descriptors_fp para la predicción de la pK_a.

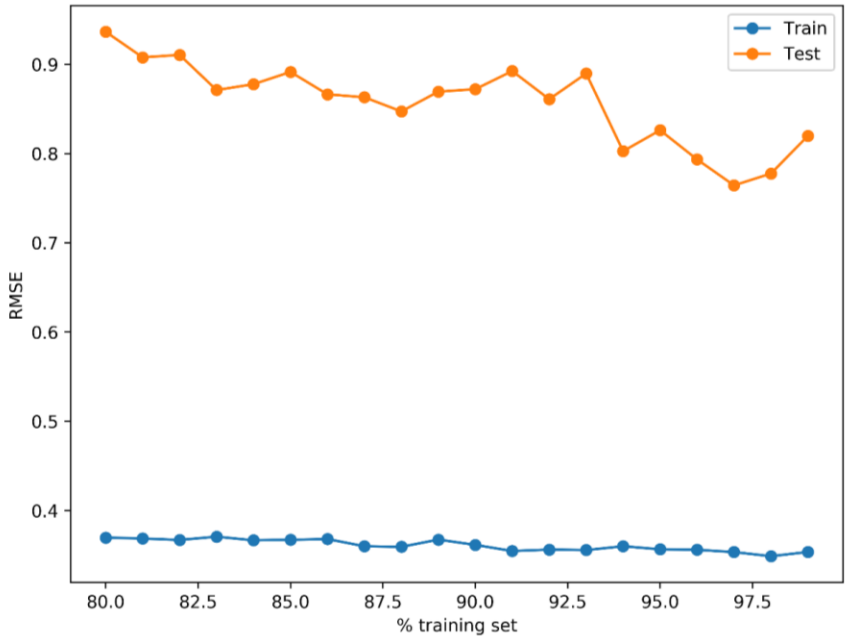


Figura 50. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo RF- pKa_descriptors para la predicción de la pK_a.

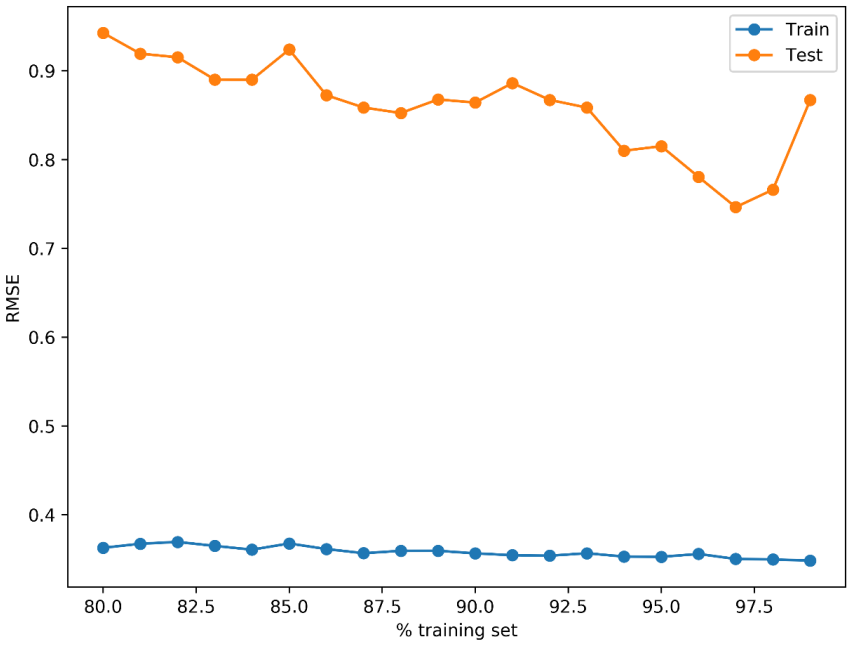


Figura 51. Variación del RMSE con el porcentaje del dataset utilizado como training set con el modelo RF- pKa_descriptors_fp para la predicción de la pK_a.

Al evaluar los cuatro modelos y compararlos con el software de ChemAxon se obtienen las métricas mostradas en el Cuadro XXV. En el caso del training set el mejor desempeño lo obtiene el modelo XGB-descr-fp, con un coeficiente de determinación casi perfecto de 0.99 y un RMSE bastante bajo de 0.25 unidades. Cabe destacar que para el training set los errores de los valores de pK_a predichos por JChem de ChemAxon son mucho mayores que los del resto de modelos propuestos.

Para el test set, los modelos propuestos también tienen mejor desempeño que ChemAxon. De nuevo el modelo XGB-descr-fp es el que tiene las mejores métricas. En revisiones hechas para modelos empíricos de pKa ha quedado evidenciado que los modelos enfocados en un solo tipo de compuestos tienen mejor desempeño, pero su dominio de aplicabilidad es muy restringido. Los modelos generales, usualmente rondan un RMSE superior a la unidad y en los mejores casos un poco menos de la unidad, con rangos usuales de 0.7-1.5. Generalmente los modelos de mecánica cuántica tienen mejores desempeños que los de ML. [132] Esto confirma que el desempeño de los modelos propuestos es bastante satisfactorio.

Cuadro XXV. Parámetros estadísticos de la evaluación del desempeño de los dos mejores modelos de predicción de la pK_a y comparación con el software de licencia ChemAxon.

Set / Modelo	Training set				Test set			
	R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB-descr	0.98	0.26	0.07	0.19	0.86	0.90	0.80	0.59
XGB-descr-fp	0.99	0.25	0.06	0.19	0.90	0.78	0.62	0.53
RF-descr	0.98	0.36	0.13	0.23	0.88	0.84	0.71	0.55
RF-descr-fp	0.98	0.36	0.13	0.23	0.88	0.86	0.74	0.55
ChemAxon	0.81	1.15	1.33	0.77	0.81	1.12	1.26	0.78

En las validaciones cruzadas de *k*-iteraciones se utilizó $k = 10$, de manera que los resultados de las métricas, mostradas en el Cuadro XXVI, corresponden al promedio de los diferentes 10 test sets en cada iteración. Se puede notar que a diferencia de los resultados del Cuadro XXV, los modelos de RF y XGB tienen desempeños casi que similares. El XGB-descr-fp tiene RMSE y MSE menores en 0.01 que el XGB-descr. Además, se nota que el tiempo de entrenamiento es aproximadamente cuatro veces menor para el XGB-descr. Esto pone en una nueva perspectiva ya que no es claro que el XGB-descr sea superior como aparentaba serlo.

Cuadro XXVI. Parámetros estadísticos de desempeño de la validación cruzada de *k*-iteraciones, con $k = 10$, de los dos mejores modelos de predicción de la pK_a.

Set / Modelo	Tiempo entrenam. / min	Training set				Test set			
		R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB-descr	0.51	0.99	0.27	0.07	0.19	0.86	0.94	0.88	0.61
XGB-descr-fp	3.44	0.99	0.26	0.07	0.19	0.86	0.93	0.87	0.61
RF-descr	8.30	0.98	0.36	0.13	0.23	0.86	0.94	0.89	0.60
RF-descr-fp	29.8	0.98	0.36	0.13	0.22	0.86	0.94	0.88	0.59

Para tener un panorama más claro se realizó una validación con dos sets externos. En ambos casos, el modelo que tienen mejor desempeño es el XGB-descr entre los modelos propuestos en este trabajo. En el caso del SAMPL6, con el modelo XGB-descr se obtuvo un RMSE de 1.21

unidades el cual es menor que el obtenido con el software de referencia. Al comparar las métricas con las del reto a ciegas de donde se tomó el set externo se nota que este modelo estaría en el top 10. [133]

Cuadro XXVII. Parámetros estadísticos de desempeño de la validación externa de los dos mejores modelos de predicción de la pK_a.

Set / Modelo	SAMPL6				SAMPL7			
	R^2	RMSE	MSE	MAE	R^2	RMSE	MSE	MAE
XGB-descr	0.83	1.21	1.46	0.94	0.71	1.70	2.90	1.49
XGB-descr-fp	0.66	1.51	2.29	1.16	0.29	2.37	5.60	1.96
RF-descr	0.84	1.33	1.78	1.07	0.76	2.15	4.61	1.89
RF-descr-fp	0.80	1.42	2.00	1.17	0.68	2.04	4.14	1.82
ChemAxon	0.86	1.25	1.55	0.99	0.90	0.78	0.61	0.62

En el caso del SAMPL7, ChemAxon tiene un mejor desempeño que los cuatro modelos propuestos en este trabajo. Al comparar solamente entre los modelos obtenidos, el XGB-descr también es el que obtuvo mejor desempeño, tanto en coeficiente de determinación como en los tres tipos de errores. Los errores del resto de modelos son bastante altos, esto pone al XGB-descr en primer lugar para elección del modelo a seleccionar. Si se comparan con el resto de modelos submitidos en el reto a ciegas SAMPL7, el modelo XGB-descr hubiese obtenido el segundo lugar, siendo superado solamente por un modelo de mecánica cuántica, en la revisión del reto también utilizaron como referencia ChemAxon y este obtuvo el mejor desempeño. [134]

Para abundar más en el modelo a seleccionar, XGB-descr, se calcularon las métricas de desempeño separado por ácidos y bases para el SAMPL6. Es claro que para las bases las predicciones son más exactas, el RMSE de las bases es incluso menor a la unidad. El MSE es dos veces mayor para ácidos que para las bases. Esto indica que una posible debilidad del modelo sea la predicción de pK_a para ácidos. En la Figura 52 se puede observar de manera gráfica como los ácidos, mostrados en rojo, tiene errores mayores que las bases. La Figura 53 muestra la regresión para el SAMPL7, este set consiste solamente de ácidos, lo que puede explicar el por qué no fue superior a ChemAxon en esta validación externa.

Cuadro XXVIII. Parámetros estadísticos de desempeño de la validación externa de los dos mejores modelos de predicción de la pK_a con las moléculas separadas en ácidos y bases.

Modelo / Set	XGB-descr			
	R^2	RMSE	MSE	MAE
SAMPL6 ácidos	0.69	1.50	2.25	1.27
SAMPL6 bases	0.63	0.98	0.96	0.74

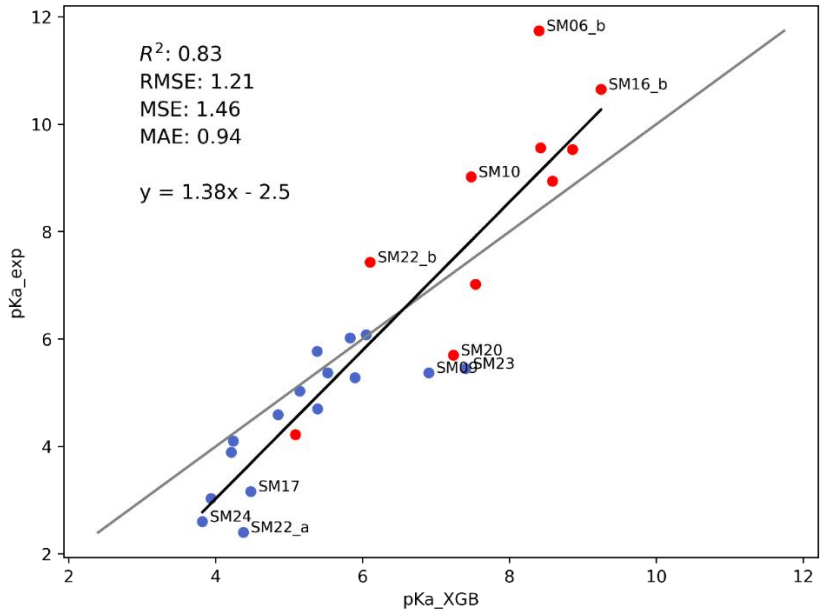


Figura 52. Regresión de los valores de pK_a predichos con el modelo XGB con los valores experimentales del SAMPL6. En rojos se muestran los ácidos y en azul las bases. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

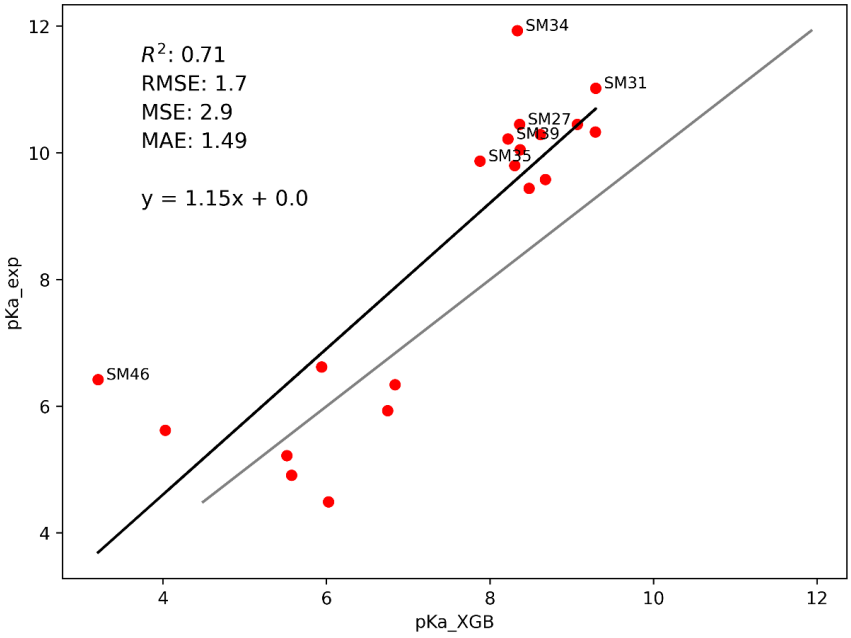


Figura 53. Regresión de los valores de pK_a predichos con el modelo XGB con los valores experimentales del SAMPL7. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

En las Figuras 54 y 55, se muestran las regresiones al comparar los valores predichos por XGB-descr y los experimentales, para el test set y training set, respectivamente. En ambos casos se puede notar que la pendiente de la línea de tendencia es bastante cercana a uno, atributo que es deseable en el modelo. En naranja se muestran las moléculas con errores más grandes, para el caso del test set nueve moléculas se consideran outliers, cinco ácidos y cuatro bases.

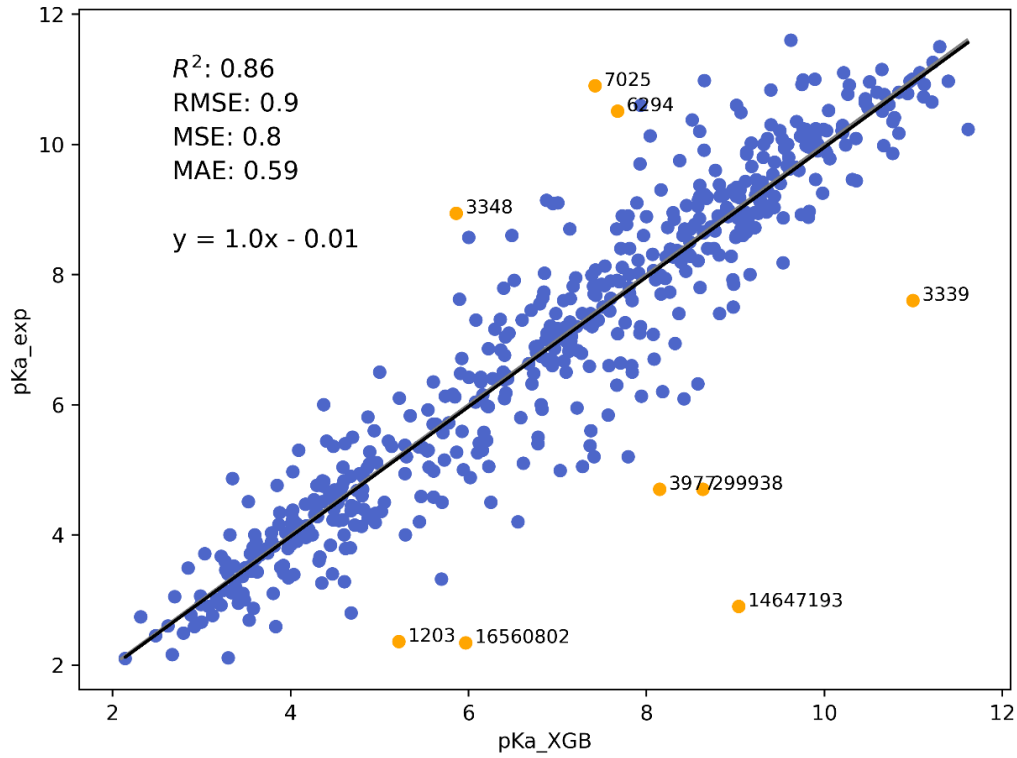


Figura 54. Regresión de los valores de pK_a predichos con el modelo XGB-descr con los valores experimentales del test set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

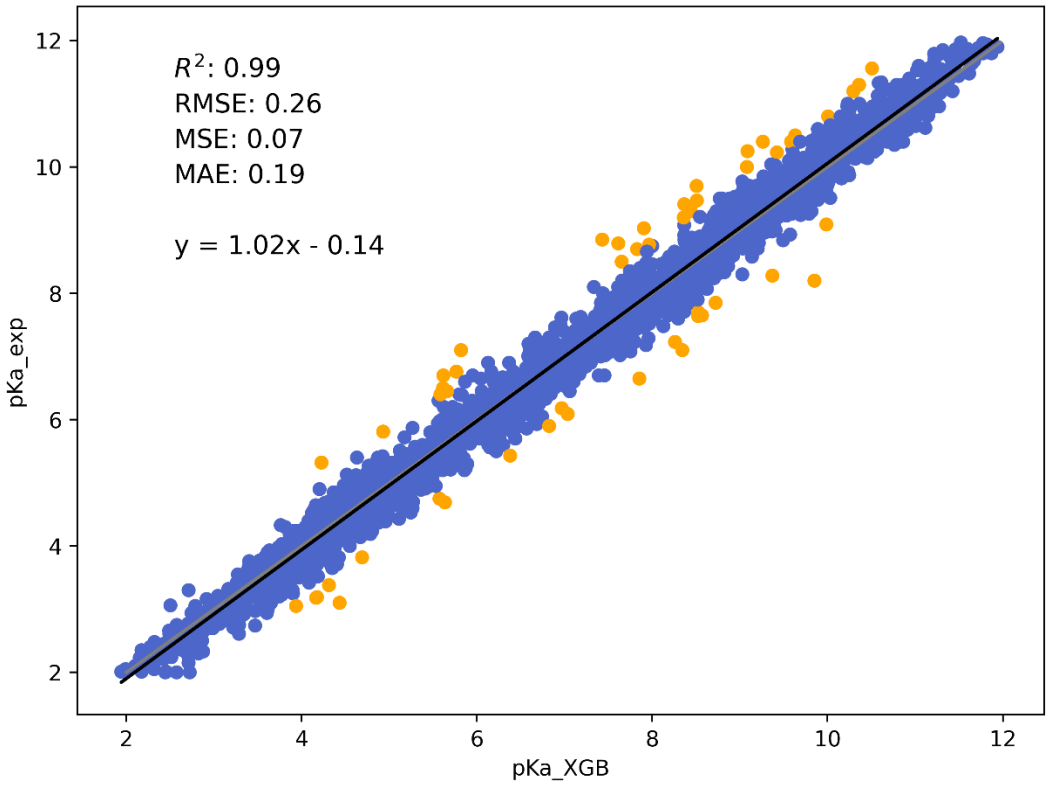


Figura 55. Regresión de los valores de pK_a predichos con el modelo XGB-descr con los valores experimentales del training set. En naranja se muestran los valores cuyo error absoluto es mayor a tres veces el RMSE del modelo. La línea negra corresponde a la línea de tendencia generada por los puntos, la línea gris a la función identidad.

Al analizar las moléculas con mayores errores, estas se encuentran fuera del dominio de aplicabilidad. En la Figura 56 al observar las estructuras se puede notar que la molécula 16560802 tiene un grupo que no es monoprótico, probablemente de ahí viene el error en la predicción, ya que el modelo no está diseñado para moléculas con más de un sitio ionizable. Otra posible fuente de error es que el modelo propuesto no toma en cuenta la tautomerización, fenómeno que ocurre en la molécula 1203. Además, también se nota la presencia de heterociclos análogos en las moléculas 14647193, 7025 y 3339; lo cual puede ser indicativo de que se necesitan más datos de este tipo de moléculas o que los descriptores utilizados no describen correctamente la variación de pK_a. Al realizar el gráfico de Williams y determinar las moléculas que están fuera del dominio de aplicabilidad, las nueve moléculas efectivamente están fuera de este.

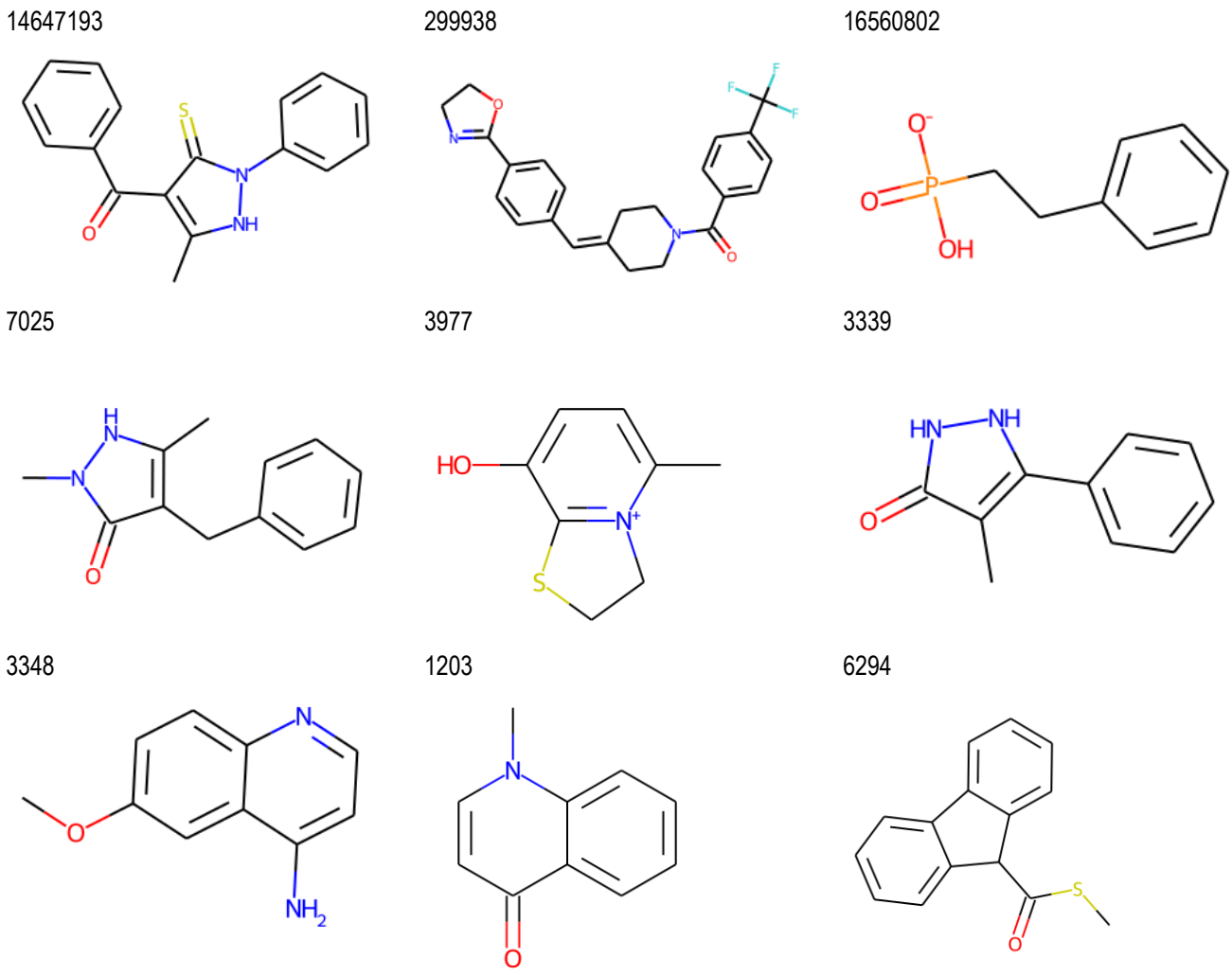


Figura 56. Estructuras de los nueve compuestos con mayor error en la predicción de pK_a con el modelo XGB-descr.

El gráfico de Williams mostrado en la Figura 57, muestra como existen moléculas fuera del dominio de aplicabilidad tanto en el training como en el test set. El valor límite de apalancamiento (h^*) del modelo es de 0.086. Un 95.86% del training set se encuentra dentro del dominio de aplicabilidad y un 94.84% del test set. Estos son porcentajes son menores si se comparan con los de los modelos de log P_N y log P_I. Esto quiere decir que la descripción de cómo se comporta la pK_a es más difícil de predecir, en retos a ciegas esto se ha notado previamente.[133,134]

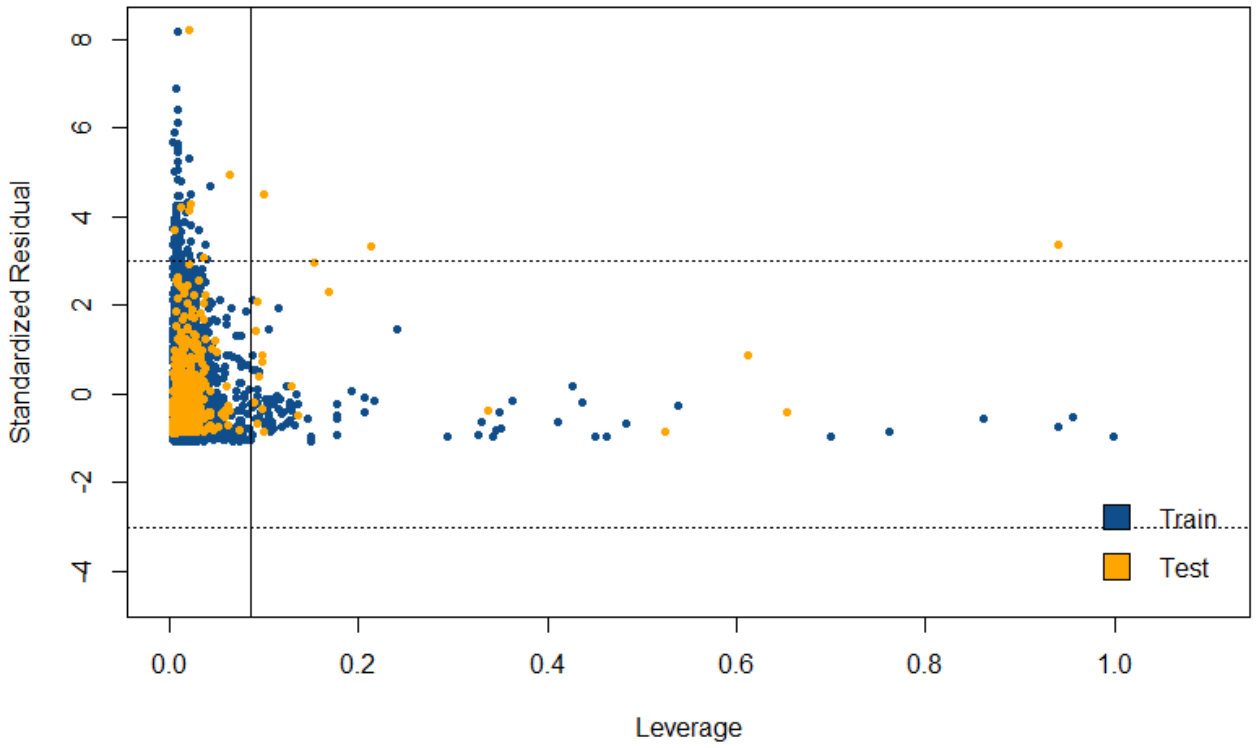


Figura 57. Gráfico de Williams para la evaluación del dominio de aplicabilidad del modelo XGB para la predicción la pK_a. En azul se muestran los puntos del training set y en naranja el test set. Los límites de residuos estándar son tres desviaciones estándar y el valor límite de apalancamiento *(h*)* es de 0.086.

En otras investigaciones[76] se han utilizado abordajes similares donde se calculan con *RDkit*[99] los mismos descriptores utilizados en este modelo, pero la clave del por qué se obtuvieron mejores desempeños radica en que no fueron solamente calculados para la molécula en su estado neutro o ionizado, si no en ambos. De manera que se pudieran restar los valores y relacionar este cambio con los valores de la constante de acidez.

3.4. Coeficiente de distribución (log D_{pH})

Al tener los tres modelos para predecir log P_N, log P_I y pK_a se integraron según las dos ecuaciones físico-químicas que describen el fenómeno de distribución entre una fase orgánica, *n*-octanol en este caso, y la fase acuosa. La ecuación 24 no incluye la partición de los compuestos ionizados, mientras que la ecuación 26 sí, de esta manera se propusieron dos modelos para la predicción del log D_{pH}. Ambos modelos toman como input el código SMILES de la molécula

neutra, el código SMILES de la molécula cargada y el pH al que se quiere conocer el coeficiente de distribución.

En el Cuadro XIX se muestras los parámetros estadísticos para evaluar el desempeño de los modelos y la comparación con el software comercial ChemAxon[127]. Se puede observar que en general el modelo que toma en cuenta la partición iónica tiene menores errores que el modelo que no; y un mejor coeficiente de determinación. El modelo de ChemAxon presenta un mejor desempeño, sin embargo, para ser un primer acercamiento al problema de predicción a diferentes pH el modelo log *D*-1 al menos es comparable con este.

Cuadro XIX. Parámetros estadísticos de la evaluación del desempeño de los modelos de predicción de log D_{pH} y comparación con el software comercial ChemAxon[127] para todas las observaciones del set de prueba.

Parámetro / Modelo	R^2	RMSE	MSE	MAE
log *D*-1 (con P_i)	0.75	0.92	0.84	0.66
log *D*-2 (sin P_i)	0.66	1.18	1.38	0.77
ChemAxon	0.85	0.66	0.44	0.50

En la Figura 58 se muestra la regresión lineal para comparar los valores predichos por el modelo y los valores experimentales. Se observa en naranja una cantidad considerable de puntos que tienen un error mayor al RMSE (0.92), la mayoría de los errores observados provienen de un alto error en la predicción de la pK_a.

El compuesto con el que se obtuvieron mayores errores corresponde a la benzoxazinona, en la Figura 59 se muestra el perfil de las predicciones y valores experimentales en un rango de pH 8-13. Se aprecia que los errores a pH extremo son mayores a las 3 unidades. Al buscar la razón de este error tan grande se encontró que la predicción de la pK_a para este compuesto fue de 7.44 y el valor experimental es de 13.2. [135] Esta gran diferencia en la pK_a significará que según las ecuaciones 24 y 26 la gran mayoría del compuesto se encuentre ionizado, lo cual significará que la predicción será un log *D* mucho más hidrofílico. Una de las razones por las que este compuesto tiene tan mala predicción de pK_a es por la ausencia de grupos funcionales oxazinona en el set de entrenamiento.

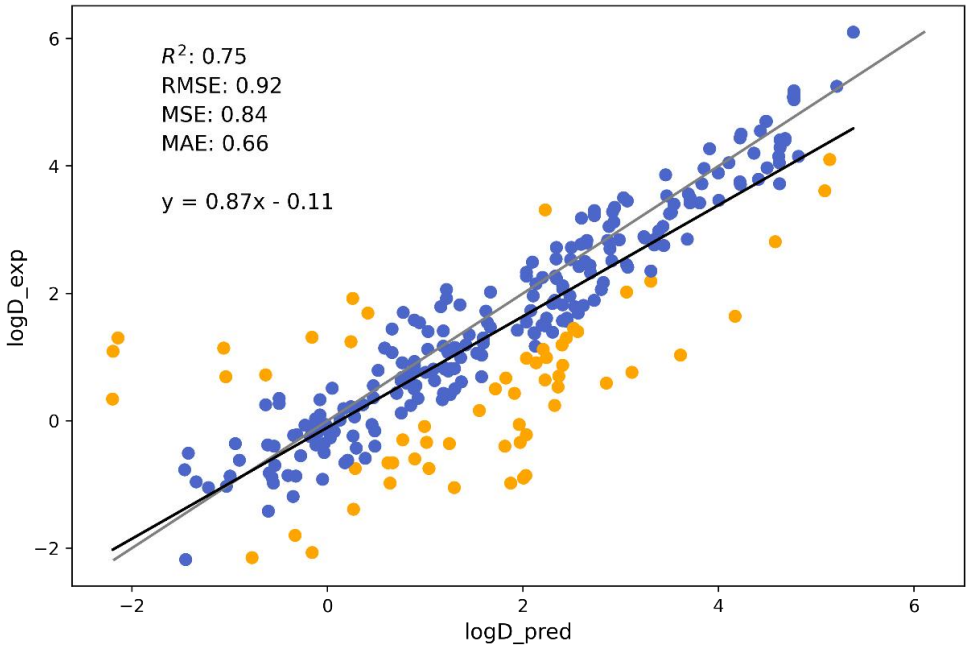


Figura 58. Comparación de los valores de log *D* predichos para el set de prueba con los valores experimentales. Se muestran en naranja las predicciones con un error mayor a valores del RMSE.

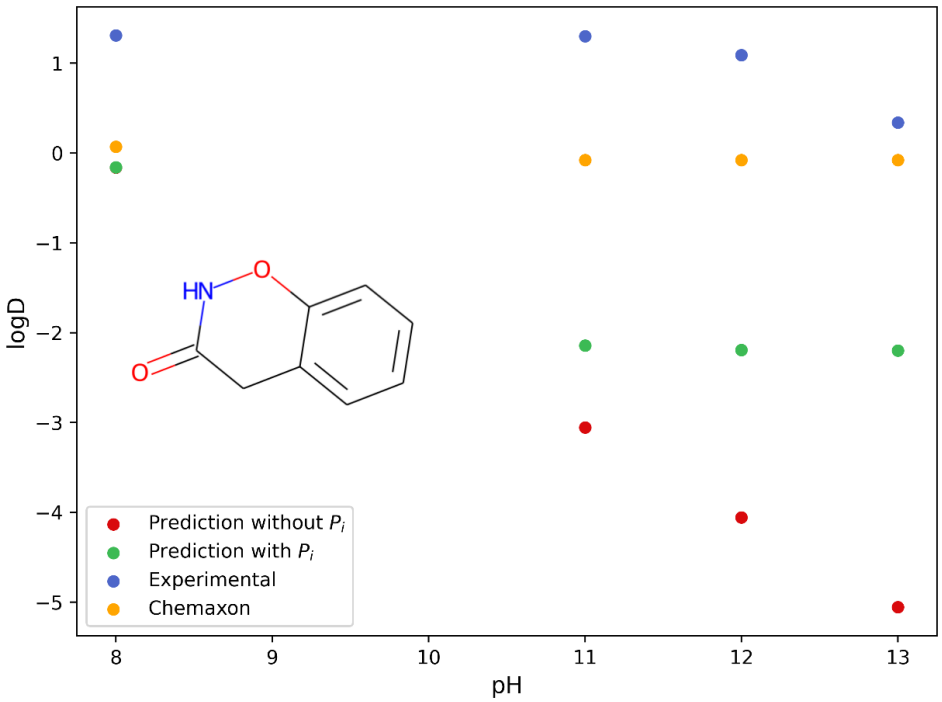


Figura 59. Perfil de variación de las predicciones de log *D* y valores experimentales con el pH del compuesto benzoxazinona.

La principal razón por la que el modelo log *D*-1 tiene mejor desempeño que el modelo log *D*-2 es que a pH extremos la predicción de log *D*-2 es pobre debido a que la mayoría de la molécula va a estar ionizada y este no toma en cuenta su partición. En el Cuadro XXX se comparan los parámetros estadísticos para los dos modelos tomando en cuenta las predicciones solo a pH mayor a 9 o menor a 5 unidades. Los errores para el modelo log *D*-2 aumentan en comparación con el set completo, mientras que los del log *D*-1mejoran.

Cuadro XXX. Parámetros estadísticos de la evaluación del desempeño de los modelos de predicción de log D_{pH} y comparación con el software comercial ChemAxon[127] para todas las observaciones a pH menor de 5 o mayor de 9.

Parámetro / Modelo	R^2	RMSE	MSE	MAE
log *D*-1 (con P_i)	0.82	0.76	0.58	0.50
log *D*-2 (sin P_i)	0.72	1.45	2.10	0.84
ChemAxon	0.90	0.53	0.28	0.40

En los siguientes ejemplos de perfiles para las moléculas de difunisal, naproxeno y DDA se puede observar cómo las predicciones con log *D*-1 siguen la forma sigmoidea de la variación del log *D* con el pH experimentalmente. Mientras que las predicciones con log *D*-2 se comportan de manera lineal a pH extremo.

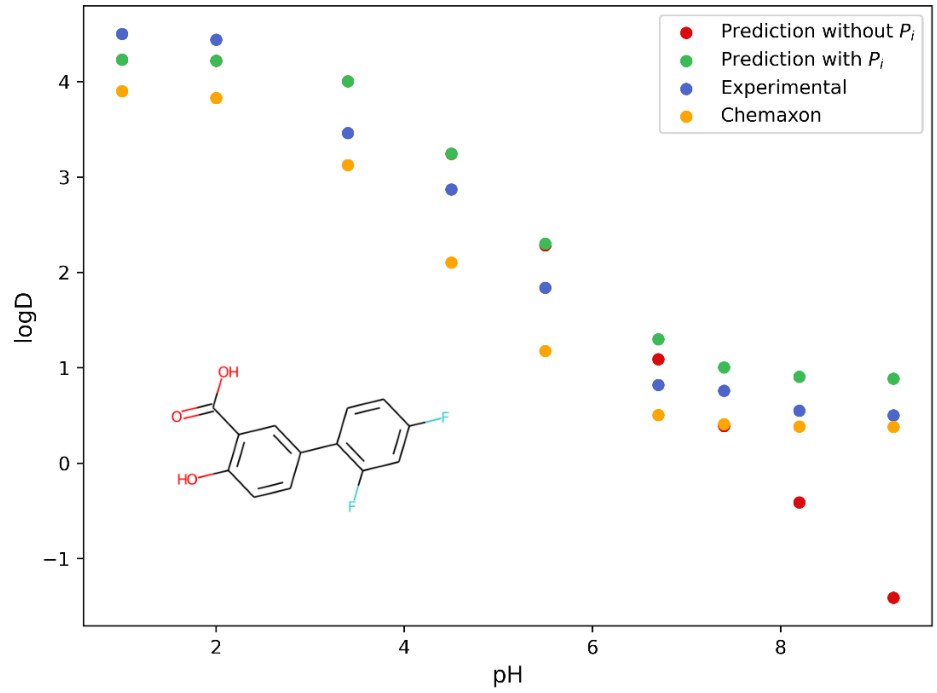


Figura 60. Perfil de variación de las predicciones de log *D* y valores experimentales con el pH del compuesto difunisal.

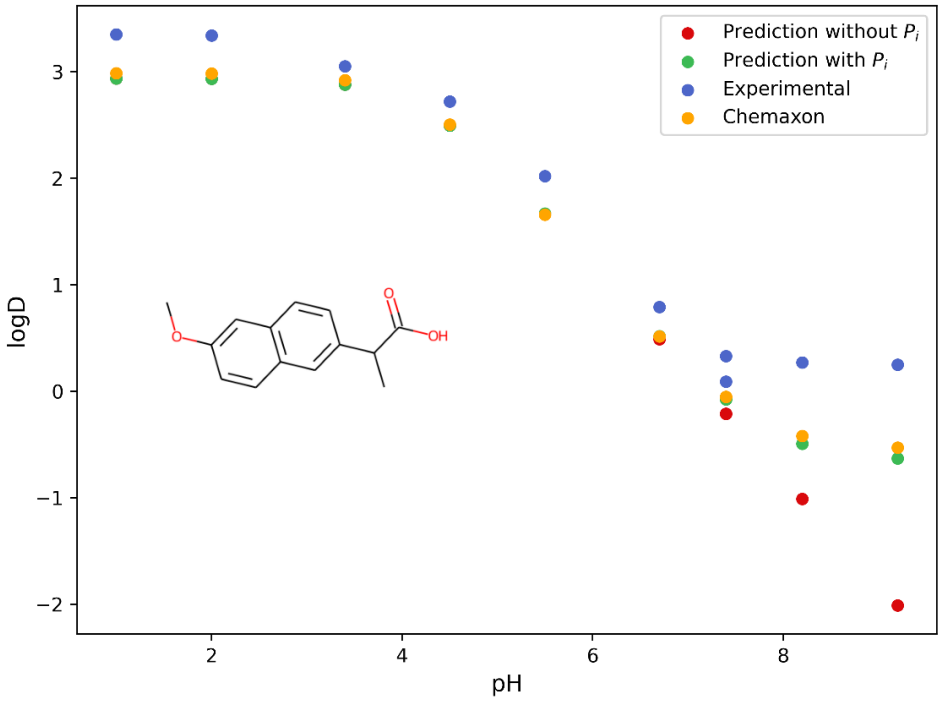


Figura 61. Perfil de variación de las predicciones de log *D* y valores experimentales con el pH del compuesto naproxeno.

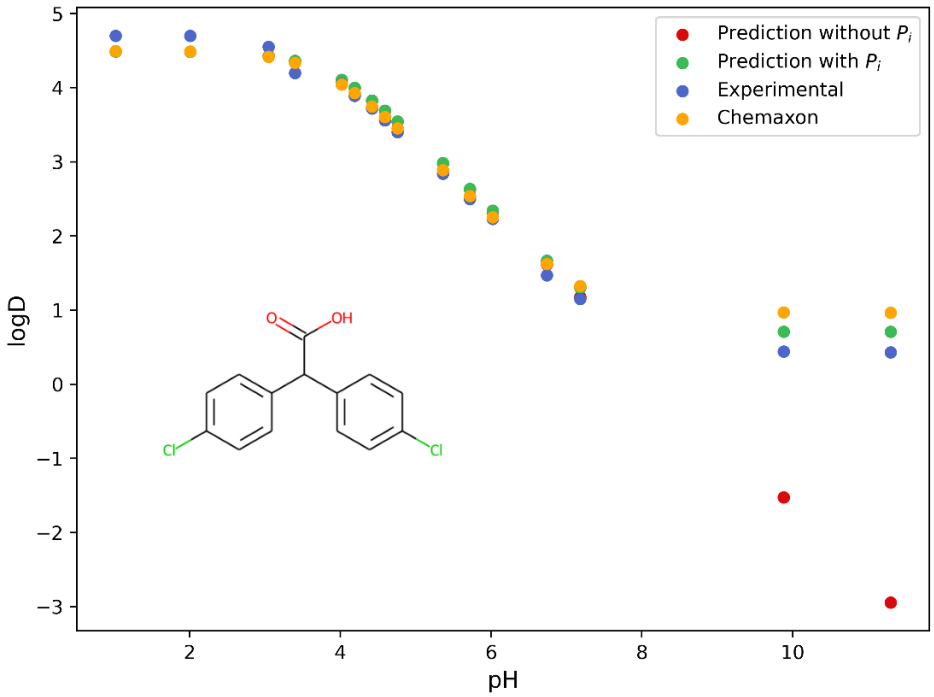


Figura 62. Perfil de variación de las predicciones de log *D* y valores experimentales con el pH del compuesto DDA.

Es importante destacar que la mayoría de los compuestos en este set de prueba estaban presentes en algunos de los tres sets de entrenamiento de las propiedades individuales. Por esta razón, se compararon los modelos con moléculas que no estuvieran presentes del todo. Para esto se utilizó el set externo del SAMPL7 [109], cuyos resultados se muestran en el Cuadro XXXI. Se puede observar que en este caso el modelo log *D*-1 tiene errores menores que ChemAxon[127], lo que demuestra que este modelo propuesto tiene un buen desempeño en la predicción del coeficiente de distribución.

Cuadro XXXI. Parámetros estadísticos de la evaluación del desempeño de los modelos de predicción de log D_{pH} y comparación con el software comercial ChemAxon[127] para el set externo SAMPL7.

Parámetro / Modelo	R^2	RMSE	MSE	MAE
log *D*-1 (con P_i)	0.55	0.96	0.93	0.77
log *D*-2 (sin P_i)	0.43	1.24	1.55	0.90
ChemAxon	0.19	1.07	1.14	0.89

Al comparar con otro tipo de métodos submitidos al reto a ciegas SAMPL7, se observa que el modelo propuesto presenta un mucho mejor desempeño que el primer lugar del reto. Cabe destacar que el método log *D*-1 es el único que tiene un RMSE menor a una unidad y en comparación con los otros métodos que tienen un buen desempeño es el único empírico, el resto corresponde a modelos físicos basados en mecánica cuántica. En el Cuadro XXXII se resumen los parámetros de desempeño de los tres mejores métodos del SAMPL7 para la predicción del coeficiente de distribución. [134]

Cuadro XXXII. Parámetros estadísticos de la evaluación del desempeño de los modelos con mejor desempeño en la predicción de log D_{pH} en el reto a ciegas SAMPL7. [134]

Parámetro / Modelo	Tipo de Modelo	R^2	RMSE	MSE	MAE
TFE IEFPCM MST + IEFPCM/MST	Mecánica Cuántica	0.55	1.27	1.61	0.98
EC_RISM	Mecánica Cuántica	0.53	1.69	2.86	1.43
TFE-NHLBI-TZVP-QM + TZVP-QM	Mecánica Cuántica	0.25	1.72	2.96	1.47

Para continuar con la evaluación del modelo, una de las moléculas, sulindaco, que se encontraba en el set de prueba para log *D*, pero no está incluida en ninguno de los tres sets de entrenamiento. El perfil de coeficiente de partición del sulindaco en un rango de pH de 3.5-8.2 se muestra en la Figura 63. Se puede apreciar que la tendencia de los log *D* predichos con los modelos propuestos, conforme aumenta el pH, es similar a la tendencia de los valores experimentales. Sin embargo, es evidente la existencia de un error positivo, esto se puede interpretar como que la predicción de pK_a es buena, pero la de log P_N no es tan buena. El valor de pK_a predicho es de 4.57 unidades y el experimental de 4.7; el log PN predicho es de 3.71 mientras que el valor experimental es de 3.42. Esto confirma que el principal error viene de la predicción del coeficiente de partición neutro. El coeficiente de partición iónico no tiene mucha influencia ya que como se observa los puntos del modelo que incluye el log P_i y el que no están prácticamente superpuestos.

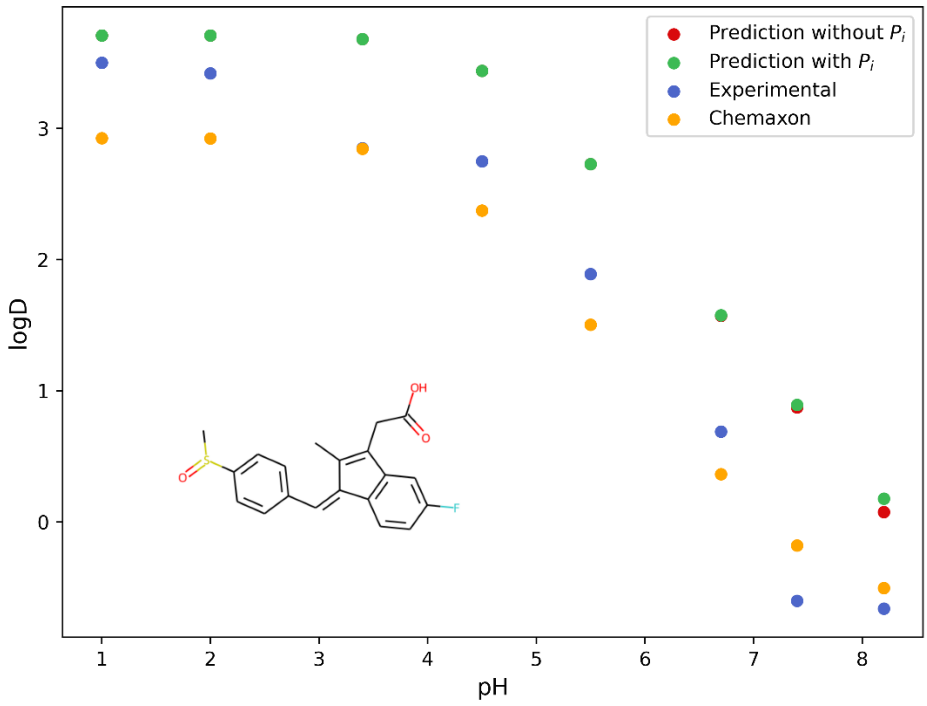


Figura 63. Perfil de variación de las predicciones de log *D* y valores experimentales con el pH del compuesto sulindaco.

Capítulo 4:

Conclusiones y recomendaciones

4. Conclusiones y recomendaciones

Se presentaron tres modelos de ML para la predicción de log P_N, log P_I y pK_a; que resultan en desempeños similares e incluso mejores en algunos casos al compararlos con software bajo licencia utilizados comúnmente. Todos los modelos utilizan herramientas de software libre y programables en Python.

En el caso de log P_N, se utilizaron descriptores de conteo de grupos funcionales y estructurales de las moléculas neutras. El algoritmo seleccionado fue el de XGBoosting, con este se obtuvo un RMSE 0.62 para el test set, a pesar de no ser el más bajo. La decisión radica en el mejor desempeño en set externos.

Para la predicción del log P_I el algoritmo seleccionado fue el de RF. El entrenamiento se hizo con descriptores de conteo de grupos funcionales de la molécula neutra y con diferencias en descriptores relacionados con la polaridad de la molécula en su estado iónico. Al realizar el análisis con varios arreglos de descriptores se llegó a la conclusión que estos últimos descriptores mencionados mejoran el desempeño en la predicción la partición iónica. Es claro que a pesar del buen desempeño en la predicción de los Delta (log P_N - log P_I), es necesario la ampliación del dominio de aplicabilidad y espacio químico del modelo.

La predicción de la pK_a fue el reto de predicción más arduo en comparación con las otras dos propiedades. A pesar de esto, se obtuvieron resultados favorables, el modelo seleccionado presentó un RMSE de 0.90 para el test set, mejores métricas en el set externo SAMPL6 y las mejores para el SAMPL7 de los modelos propuestos en el trabajo. El algoritmo utilizado fue el XGBoosting. Los descriptores del modelo son bastante simples, pero a pesar de que en otras investigaciones se han hecho abordajes similares, la clave está en incluir descriptores para la molécula tanto en su estado neutro como ionizado. Para este modelo es necesario la mejora principalmente para ácidos.

Los tres modelos son simples y presentan tiempos de entrenamiento/predicción bajos en comparación con modelos que tienen desempeños similares pero que requieren mucho más tiempo y coste computacional, como los de mecánica cuántica. Al integrar los tres modelos se obtuvo un modelo efectivo del coeficiente de distribución a diferentes pH. La evidencia mostrada sugiere que es necesario la inclusión del fenómeno de partición iónico en la predicción del log *D*, especialmente para pH extremos. En varios de los perfiles calculados, se notó que el modelo que

más influía en los fallos era el de la pK_a. Se recomienda a futuro buscar el aumento de la base datos o descriptores que describan mejor la constante de acidez.

Para el set de prueba la mayoría de los compuestos estaban presentes en al menos uno de los sets de entrenamiento de alguna de las tres propiedades. Es recomendable la búsqueda de sets externos que no estén presentes para evaluar de una mejor manera el desempeño del modelo integrado. Una de las principales limitaciones es la falta de datos en la literatura pH diferente del pH fisiológico. Con el set de prueba no se logró obtener mejores métricas que el software de licencia ChemAxon, pero para el set externo sí. Esto es un buen indicativo de que el modelo sí tiene un buen desempeño con moléculas fuera de los sets de entrenamiento, pero se requiere más evidencia con otros sets que sean más diversos y con diferentes valores de pH para evaluar correctamente el modelo.

Bibliografía

Bibliografía

(1) Shen, J.; Nicolaou, C. A. Molecular Property Prediction: Recent Trends in the Era of Artificial Intelligence. *Drug Discovery Today: Technologies* **2019**, *32–33*, 29–36. https://doi.org/10.1016/J.DDTEC.2020.05.001.

(2) Dowden, H.; Munro, J. Trends in Clinical Success Rates and Therapeutic Focus. *Nat Rev Drug Discov* **2019**, *18* (7), 495–496. https://doi.org/10.1038/D41573-019-00074-Z.

(3) Leelananda, S. P.; Lindert, S. Computational Methods in Drug Discovery. *Beilstein journal of organic chemistry* **2016**, *12*, 2694–2718. https://doi.org/10.3762/BJOC.12.267.

(4) Waring, M. J. Lipophilicity in Drug Discovery. *Expert Opinion on Drug Discovery* **2010**, *5* (3), 235–248. https://doi.org/10.1517/17460441003605098.

(5) Young, R. J. Physical Properties in Drug Design. *Topics in Medicinal Chemistry* **2014**, *9*, 1–68. https://doi.org/10.1007/7355_2013_35.

(6) Yang, X.; Wang, Y.; Byrne, R.; Schneider, G.; Yang, S. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. *Chemical Reviews* **2019**, *119* (18), 10520–10594. https://doi.org/10.1021/ACS.CHEMREV.8B00728.

(7) Avdeef, A. *Absorption and Drug Development*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2012. https://doi.org/10.1002/9781118286067.

(8) Edgar, T. W.; Manz, D. O. Machine Learning. *Research Methods for Cyber Security* **2017**, 153–173. https://doi.org/10.1016/B978-0-12-805349-2.00006-6.

(9) Veliz Capuñay, C. *Aprendizaje Automático : Introducción al Aprendizaje Profundo*; 2020.

(10) Janet, J. P.; Kulik, H. J.; Yamilee Morency, C. majors, F. U.; Mary Kate Caucci, C. major, F. U. *Machine Learning in Chemistry*; American Chemical Society: Washington, DC, USA, 2020. https://doi.org/10.1021/acs.infocus.7e4001.

(11) Qian, N.; Sejnowski, T. J. Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. *Journal of Molecular Biology* **1988**, *202* (4), 865–884. https://doi.org/10.1016/0022-2836(88)90564-5.

(12) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature 2021 596:7873* **2021**, *596* (7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2.

(13) Callaway, E. “It Will Change Everything”: DeepMind’s AI Makes Gigantic Leap in Solving Protein Structures. *Nature* **2020**, *588* (7837), 203–204. https://doi.org/10.1038/D41586-020-03348-4.

(14) Gawehn, E.; Hiss, J. A.; Brown, J. B.; Schneider, G. Advancing Drug Discovery via GPU-Based Deep Learning. *Expert Opin Drug Discov* **2018**, *13* (7), 579–582. https://doi.org/10.1080/17460441.2018.1465407.

(15) Mohri, M.; Rostamizadeh, A.; Talwalkar, A. Foundations of Machine Learning Second Edition. *Вестник Казнму* **2018**, *№3* (1), с.30.

(16) Louppe, G. Understanding Random Forests: From Theory to Practice. **2014**.

(17) Bbeiman, L. Bagging Predictors. **1996**, *24*, 123–140.

(18) Boyle, B. H. *Support Vector Machines : Data Analysis, Machine Learning, and Applications*; Nova Science Publishers, 2011.

(19) Martín Guareño, J. J. Support Vector Regression: Propiedades y Aplicaciones, Universidad de Sevilla, Sevilla, 2016.

(20) Tobias, R. D. An Introduction to Partial Least Squares Regression. *Proceedings of the twentieth annual SAS users group international conference* **1995**, *20*.

(21) Abdi, H. Partial Least Squares (PLS) Regression. *Encyclopedia for research methods for the social sciences* **2003**, *6* (4), 792–795.

(22) Fullér, R. Artificial Neural Networks. *Introduction to Neuro-Fuzzy Systems* **2000**, 133–170. https://doi.org/10.1007/978-3-7908-1852-9_2.

(23) Cain, G. *Artificial Neural Networks : New Research*; Nova Science Publishers, Inc., 2016.

(24) Sharma, S.; Sharma, S.; Athaiya, A. ACTIVATION FUNCTIONS IN NEURAL NETWORKS. *International Journal of Engineering Applied Sciences and Technology* **2020**, *4*, 310–316.

(25) Sanket, D. *Various Optimization Algorithms For Training Neural Network*. Towards Data Science. https://towardsdatascience.com/optimizers-for-training-neural-network-59450d71caf6 (accessed 2022-02-08).

(26) Moritz, P.; Nishihara, R.; Jordan, M. I. A Linearly-Convergent Stochastic L-BFGS Algorithm. PMLR May 2, 2016, pp 249–258.

(27) Ketkar, N. Stochastic Gradient Descent. *Deep Learning with Python* **2017**, 113–132. https://doi.org/10.1007/978-1-4842-2766-4_8.

(28) Kingma, D. P.; Lei Ba, J. Adam: A Method for Stochastic Optimization. *ICLR* **2015**.

(29) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. https://doi.org/10.1145/2939672.

(30) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. **2016**. https://doi.org/10.1145/2939672.2939785.

(31) *Introduction to Boosted Trees — xgboost 1.5.2 documentation*. https://xgboost.readthedocs.io/en/stable/tutorials/model.html (accessed 2022-01-17).

(32) Shalev-Shwartz, Shai.; Ben-David, Shai. *Understanding Machine Learning : From Theory to Algorithms*; 2014.

(33) Bowles, M. Machine Learning in Python: Essential Techniques for Predictive Analysis: Chapter 7 Building Ensemble Models with Python. *Machine Learning in Python: Essential Techniques for Predictive Analysis* **2015**, 360.

(34) Chicco, D.; Warrens, M. J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Comput. Sci.* **2021**. https://doi.org/10.7717/peerj-cs.623.

(35) Han, J.; Kamber, M.; Pei, J. Classification: Basic Concepts. *Data Mining* **2012**, 327–391. https://doi.org/10.1016/B978-0-12-381479-1.00008-3.

(36) Ho, S. Y.; Phua, K.; Wong, L.; bin Goh, W. W. Extensions of the External Validation for Checking Learned Model Interpretability and Generalizability. *Patterns* **2020**, *1* (8), 100129. https://doi.org/10.1016/J.PATTER.2020.100129.

(37) Kar, S.; Roy, K.; Leszczynski, J. Chapter 6 Applicability Domain: A Step Toward Confident Predictions and Decidability for QSAR Modeling. *Methods in Molecular Biology* **2018**, *1800*. https://doi.org/10.1007/978-1-4939-7899-1_6.

(38) Arnott, J. A.; Planey, S. L. The Influence of Lipophilicity in Drug Discovery and Design. *http://dx.doi.org/10.1517/17460441.2012.714363* **2012**, *7* (10), 863–875. https://doi.org/10.1517/17460441.2012.714363.

(39) Mc Naught, a. D.; Wilkinson, a. Compendium of Chemical Terminology-Gold Book. *Iupac* **2012**, 1670. https://doi.org/10.1351/goldbook.

(40) Hongmao, S. Quantitative Structure–Property Relationships Models for Lipophilicity and Aqueous Solubility. In *A Practical Guide to Rational Drug Design*; Elsevier, 2016; pp 193–223. https://doi.org/10.1016/B978-0-08-100098-4.00006-5.

(41) Waring, M. J. Defining Optimum Lipophilicity and Molecular Weight Ranges for Drug Candidates-Molecular Weight Dependent Lower LogD Limits Based on Permeability. *Bioorg Med Chem Lett* **2009**, *19* (10), 2844–2851. https://doi.org/10.1016/J.BMCL.2009.03.109.

(42) Waring, M. J. Lipophilicity in Drug Discovery. *Expert Opinion on Drug Discovery* **2010**, *5* (3), 235–248. https://doi.org/10.1517/17460441003605098.

(43) Ginex, T.; Vazquez, J.; Gilbert, E.; Herrero, E.; Luque, F. J. Lipophilicity in Drug Design: An Overview of Lipophilicity Descriptors in 3D-QSAR Studies. *Future Medicinal Chemistry* **2019**, *11* (10), 1177–1193. https://doi.org/10.4155/FMC-2018-0435.

(44) Johanson, G. Modeling of Disposition. *Comprehensive Toxicology: Second Edition* **2010**, *1–14*, 153–177. https://doi.org/10.1016/B978-0-08-046884-6.00108-1.

(45) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Advanced Drug Delivery Reviews* **1997**, *23* (1–3), 3–25. https://doi.org/10.1016/S0169-409X(96)00423-1.

(46) Caron, G.; Ermondi, G.; Scherrer, R. A. Lipophilicity, Polarity, and Hydrophobicity. *Comprehensive Medicinal Chemistry II* **2007**, *5*, 425–452. https://doi.org/10.1016/B0-08-045044-X/00135-8.

(47) Bouchard, G.; Carrupt, P.-A.; Testa, B.; Gobry, V.; Girault, H. H. Lipophilicity and Solvation of Anionic Drugs. *Chemistry - A European Journal* **2002**, *8* (15), 3478. https://doi.org/10.1002/1521-3765(20020802)8:15<3478::AID-CHEM3478>3.0.CO;2-U.

(48) Zang, Q.; Mansouri, K.; Williams, A. J.; Judson, R. S.; Allen, D. G.; Casey, W. M.; Kleinstreuer, N. C. In Silico Prediction of Physicochemical Properties of Environmental Chemicals Using Molecular Fingerprints and Machine Learning. *Journal of Chemical Information and Modeling* **2017**, *57* (1), 36–49. https://doi.org/10.1021/ACS.JCIM.6B00625/SUPPL_FILE/CI6B00625_SI_002.XLSX.

(49) *Daylight Theory: SMARTS - A Language for Describing Molecular Patterns*. https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html (accessed 2022-06-01).

(50) Bahmani, A.; Saaidpour, S.; Rostami, A. A Simple, Robust and Efficient Computational Method for n-Octanol/Water Partition Coefficients of Substituted Aromatic Drugs. *Scientific Reports 2017 7:1* **2017**, *7* (1), 1–14. https://doi.org/10.1038/s41598-017-05964-z.

(51) Plante, J.; Werner, S. JPlogP: An Improved LogP Predictor Trained Using Predicted Data. *Journal of Cheminformatics* **2018**, *10* (1), 1–10. https://doi.org/10.1186/S13321-018-0316-5/TABLES/4.

(52) Tetko, I. v.; Tanchuk, V. Y. Application of Associative Neural Networks for Prediction of Lipophilicity in ALOGPS 2.1 Program. *J Chem Inf Comput Sci* **2002**, *42* (5), 1136–1145. https://doi.org/10.1021/CI025515J.

(53) Tetko, I. v. Associative Neural Network. *Neural Processing Letters 2002 16:2* **2002**, *16* (2), 187–199. https://doi.org/10.1023/A:1019903710291.

(54) Prasad, S.; Brooks, B. R. A Deep Learning Approach for the Blind LogP Prediction in SAMPL6 Challenge. *Journal of Computer-Aided Molecular Design* **2020**, *34* (5), 535–542. https://doi.org/10.1007/S10822-020-00292-3/FIGURES/6.

(55) Patel, P.; Kuntz, D. M.; Jones, M. R.; Brooks, B. R.; Wilson, A. K. SAMPL6 LogP Challenge: Machine Learning and Quantum Mechanical Approaches. *Journal of Computer-Aided Molecular Design* **2020**, *34* (5), 495–510. https://doi.org/10.1007/S10822-020-00287-0/TABLES/9.

(56) Chen, H. F. In Silico Log P Prediction for a Large Data Set with Support Vector Machines, Radial Basis Neural Networks and Multiple Linear Regression. *Chemical Biology & Drug Design* **2009**, *74* (2), 142–147. https://doi.org/10.1111/J.1747-0285.2009.00840.X.

(57) Tang, B.; Kramer, S. T.; Fang, M.; Qiu, Y.; Wu, Z.; Xu, D. A Self-Attention Based Message Passing Neural Network for Predicting Molecular Lipophilicity and Aqueous Solubility. *Journal of Cheminformatics* **2020**, *12* (1), 1–9. https://doi.org/10.1186/S13321-020-0414-Z/FIGURES/4.

(58) Lopez, K.; Pinheiro, S.; Zamora, W. J. Multiple Linear Regression Models for Predicting the N-octanol/Water Partition Coefficients in the SAMPL7 Blind Challenge. *Journal of Computer-Aided Molecular Design* **2021**, *35* (8), 923–931. https://doi.org/10.1007/S10822-021-00409-2/FIGURES/7.

(59) *Chemprop*. http://chemprop.csail.mit.edu/ (accessed 2022-06-03).

(60) Mansouri, K.; Grulke, C. M.; Judson, R. S.; Williams, A. J. OPERA Models for Predicting Physicochemical Properties and Environmental Fate Endpoints. *Journal of Cheminformatics* **2018**, *10* (1). https://doi.org/10.1186/S13321-018-0263-1.

(61) Lenselink, E. B.; Stouten, P. F. W. Multitask Machine Learning Models for Predicting Lipophilicity (LogP) in the SAMPL7 Challenge. *Journal of Computer-Aided Molecular Design* **2021**, *35* (8), 901–909. https://doi.org/10.1007/S10822-021-00405-6/TABLES/3.

(62) Donyapour, N.; Dickson, A. Predicting Partition Coefficients for the SAMPL7 Physical Property Challenge Using the ClassicalGSG Method. *Journal of Computer-Aided Molecular Design* **2021**, *35* (7), 819–830. https://doi.org/10.1007/S10822-021-00400-X/TABLES/6.

(63) Datta, R.; Das, D.; Das, S. Efficient Lipophilicity Prediction of Molecules Employing Deep-Learning Models. *Chemometrics and Intelligent Laboratory Systems* **2021**, *213*, 104309. https://doi.org/10.1016/J.CHEMOLAB.2021.104309.

(64) Ulrich, N.; Goss, K. U.; Ebert, A. Exploring the Octanol–Water Partition Coefficient Dataset Using Deep Learning Techniques and Data Augmentation. *Communications Chemistry 2021 4:1* **2021**, *4* (1), 1–10. https://doi.org/10.1038/s42004-021-00528-9.

(65) Hodges, G.; Eadsforth, C.; Bossuyt, B.; Bouvy, A.; Enrici, M. H.; Geurts, M.; Kotthoff, M.; Michie, E.; Miller, D.; Müller, J.; Oetter, G.; Roberts, J.; Schowanek, D.; Sun, P.; Venzmer, J. A Comparison of Log K Ow (n-Octanol–Water Partition Coefficient) Values for Non-Ionic, Anionic, Cationic and Amphoteric Surfactants Determined Using Predictions and Experimental Methods. *Environmental Sciences Europe* **2019**, *31* (1), 1–18. https://doi.org/10.1186/S12302-018-0176-7/TABLES/6.

(66) Wade, G. *Química Orgánica*; Pearson, 2017; Vol. 1.

(67) Manallack, D. T. The PKa Distribution of Drugs: Application to Drug Discovery. *Perspectives in Medicinal Chemistry* **2007**, *1*, 25. https://doi.org/10.1177/1177391x0700100003.

(68) Xing, L.; Glen, R. C.; Clark, R. D. Predicting PKa by Molecular Tree Structured Fingerprints and PLS. *Journal of Chemical Information and Computer Sciences* **2003**, *43* (3), 870–879. https://doi.org/10.1021/CI020386S/ASSET/IMAGES/LARGE/CI020386SF00006.JPEG.

(69) Jinhua, Z.; Kleinöder, T.; Gasteiger, J. Prediction of PKa Values for Aliphatic Carboxylic Acids and Alcohols with Empirical Atomic Charge Descriptors. *Journal of Chemical Information and Modeling* **2006**, *46* (6), 2256–2266. https://doi.org/10.1021/CI060129D/ASSET/IMAGES/LARGE/CI060129DF00009.JPEG.

(70) Jelfs, S.; Ertl, P.; Selzer, P. Estimation of PKa for Druglike Compounds Using Semiempirical and Information-Based Descriptors. *Journal of Chemical Information and Modeling* **2007**, *47* (2), 450–459. https://doi.org/10.1021/CI600285N/SUPPL_FILE/CI600285N-FILE002.XLS.

(71) Lu, Y.; Anand, S.; Shirley, W.; Gedeck, P.; Kelley, B. P.; Skolnik, S.; Rodde, S.; Nguyen, M.; Lindvall, M.; Jia, W. Prediction of p Ka Using Machine Learning Methods with Rooted Topological Torsion Fingerprints: Application to Aliphatic Amines. *Journal of Chemical Information and Modeling* **2019**, *59* (11), 4706–4719. https://doi.org/10.1021/ACS.JCIM.9B00498/ASSET/IMAGES/LARGE/CI9B00498_0002.JPEG.

(72) Jover, J.; Bosque, R.; Sales, J. QSPR Prediction of PKa for Benzoic Acids in Different Solvents. *QSAR and Combinatorial Science* **2008**, *27* (5), 563–581. https://doi.org/10.1002/QSAR.200710095.

(73) Li, M.; Zhang, H.; Chen, B.; Wu, Y.; Guan, L. Prediction of PKa Values for Neutral and Basic Drugs Based on Hybrid Artificial Intelligence Methods. *Scientific Reports 2018 8:1* **2018**, *8* (1), 1–13. https://doi.org/10.1038/s41598-018-22332-7.

(74) Yang, Q.; Li, Y.; Yang, J.; Liu, Y.; Zhang, L.; Luo, S.; Cheng, J. Holistic Prediction of the p K a in Diverse Solvents Based on a Machine-Learning Approach . *Angewandte Chemie* **2020**, *132* (43), 19444–19453. https://doi.org/10.1002/ANGE.202008528.

(75) Mansouri, K.; Cariello, N. F.; Korotcov, A.; Tkachenko, V.; Grulke, C. M.; Sprankle, C. S.; Allen, D.; Casey, W. M.; Kleinstreuer, N. C.; Williams, A. J. Open-Source QSAR Models for PKa Prediction Using Multiple Machine Learning Approaches. *Journal of Cheminformatics* **2019**, *11* (1), 1–20. https://doi.org/10.1186/S13321-019-0384-1/TABLES/13.

(76) Baltruschat, M.; Czodrowski, P. Machine Learning Meets PKa. *F1000Res* **2020**, *9*. https://doi.org/10.12688/F1000RESEARCH.22090.2.

(77) Francoeur, P. G.; Peñaherrera, D.; Koes, D. R. Active Learning for Small Molecule PKa Regression; a Long Way To Go. **2022**. https://doi.org/10.26434/CHEMRXIV-2022-8W1Q0.

(78) Scherrer, R. A.; Howard, S. M. Use of Distribution Coefficients in Quantitative Structure-Activity Relationships. *Journal of Medicinal Chemistry* **1977**, *20* (1), 53–58. https://doi.org/10.1021/JM00211A010.

(79) Low, Y. W. (Ivan); Blasco, F.; Vachaspati, P. Optimised Method to Estimate Octanol Water Distribution Coefficient (LogD) in a High Throughput Format. *European Journal of Pharmaceutical Sciences* **2016**, *92*, 110–116. https://doi.org/10.1016/J.EJPS.2016.06.024.

(80) Bhal, S. K.; Kassam, K.; Peirson, I. G.; Pearl, G. M. The Rule of Five Revisited: Applying Log D in Place of Log P in Drug-Likeness Filters. *Molecular Pharmaceutics* **2007**, *4* (4), 556–560. https://doi.org/10.1021/MP0700209/SUPPL_FILE/MP0700209SI20070511_093159.PDF.

(81) Zamora W. J. Toward Refined Theoretical Models for the Description of Lipophilicity in Biomolecules, Universitat de Barcelona, Barcelona, 2019.

(82) Aliagas, I.; Gobbi, A.; Lee, M.-L.; Sellers, B. D. Comparison of LogP and LogD Correction Models Trained with Public and Proprietary Data Sets. *Journal of Computer-Aided Molecular Design 2022 36:3* **2022**, *36* (3), 253–262. https://doi.org/10.1007/S10822-022-00450-9.

(83) Csizmadia, F.; Tsantili-Kakoulidou, A.; Panderi, I.; Darvas, F. Prediction of Distribution Coefficient from Structure. 1. Estimation Method. *Journal of Pharmaceutical Sciences* **1997**, *86* (7), 865–871. https://doi.org/10.1021/JS960177K.

(84) Bruneau, P.; McElroy, N. R. LogD7.4 Modeling Using Bayesian Regularized Neural Networks. Assessment and Correction of the Errors of Prediction. *Journal of Chemical Information and Modeling* **2005**, *46* (3), 1379–1387. https://doi.org/10.1021/CI0504014.

(85) Burden, F.; Winkler, D. Bayesian Regularization of Neural Networks. *Methods in Molecular Biology* **2008**, *458*, 23–42. https://doi.org/10.1007/978-1-60327-101-1_3.

(86) Wang, J. B.; Cao, D. S.; Zhu, M. F.; Yun, Y. H.; Xiao, N.; Liang, Y. Z. In Silico Evaluation of LogD7.4 and Comparison with Other Prediction Methods. *Journal of Chemometrics* **2015**, *29* (7), 389–398. https://doi.org/10.1002/CEM.2718.

(87) Hop, P.; Allgood, B.; Yu, J. Geometric Deep Learning Autonomously Learns Chemical Features That Outperform Those Engineered by Domain Experts. *Molecular Pharmaceutics* **2018**, *15* (10), 4371–4377. https://doi.org/10.1021/ACS.MOLPHARMACEUT.7B01144/ASSET/IMAGES/MEDIUM/MP-2017-01144R_M008.GIF.

(88) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science* **2018**, *9* (2), 513–530. https://doi.org/10.1039/C7SC02664A.

(89) Lapins, M.; Arvidsson, S.; Lampa, S.; Berg, A.; Schaal, W.; Alvarsson, J.; Spjuth, O. A Confidence Predictor for LogD Using Conformal Regression and a Support-Vector Machine. *Journal of Cheminformatics* **2018**, *10* (1), 1–10. https://doi.org/10.1186/S13321-018-0271-1/FIGURES/5.

(90) Fu, L.; Liu, L.; Yang, Z. J.; Li, P.; Ding, J. J.; Yun, Y. H.; Lu, A. P.; Hou, T. J.; Cao, D. S. Systematic Modeling of Log D7.4 Based on Ensemble Machine Learning, Group Contribution, and Matched Molecular Pair Analysis. *Journal of Chemical Information and Modeling* **2020**, *60* (1), 63–76. https://doi.org/10.1021/ACS.JCIM.9B00718/SUPPL_FILE/CI9B00718_SI_002.XLSX.

(91) Fuchs, J. A.; Grisoni, F.; Kossenjans, M.; Hiss, J. A.; Schneider, G. Lipophilicity Prediction of Peptides and Peptide Derivatives by Consensus Machine Learning. *Medchemcomm* **2018**, *9* (9), 1538–1546. https://doi.org/10.1039/C8MD00370J.

(92) Lapins, M.; Arvidsson, S.; Lampa, S.; Berg, A.; Schaal, W.; Alvarsson, J.; Spjuth, O. A Confidence Predictor for LogD Using Conformal Regression and a Support-Vector Machine. *Journal of Cheminformatics* **2018**, *10* (1), 1–10. https://doi.org/10.1186/S13321-018-0271-1/FIGURES/5.

(93) Fu, L.; Liu, L.; Yang, Z. J.; Li, P.; Ding, J. J.; Yun, Y. H.; Lu, A. P.; Hou, T. J.; Cao, D. S. Systematic Modeling of Log D7.4 Based on Ensemble Machine Learning, Group Contribution, and Matched Molecular Pair Analysis. *Journal of Chemical Information and Modeling* **2020**, *60* (1), 63–76. https://doi.org/10.1021/ACS.JCIM.9B00718/SUPPL_FILE/CI9B00718_SI_002.XLSX.

(94) *BioByte*. http://www.biobyte.com/ (accessed 2022-06-04).

(95) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50* (5), 742–754. https://doi.org/10.1021/CI100050T/ASSET/IMAGES/LARGE/CI-2010-00050T_0017.JPEG.

(96) Aliagas, I.; Gobbi, A.; Lee, M.-L.; Sellers, B. D. Comparison of LogP and LogD Correction Models Trained with Public and Proprietary Data Sets. *Journal of Computer-Aided Molecular Design 2022 36:3* **2022**, *36* (3), 253–262. https://doi.org/10.1007/S10822-022-00450-9.

(97) chang, george; Woody, N.; Keefer, C. Providing the 'Best' Lipophilicity Assessment in a Drug Discovery Environment. **2021**. https://doi.org/10.26434/CHEMRXIV.14292485.V1.

(98) Wishart, D. S.; Knox, C.; Guo, A. C.; Shrivastava, S.; Hassanali, M.; Stothard, P.; Chang, Z.; Woolsey, J. DrugBank: A Comprehensive Resource for in Silico Drug Discovery and Exploration. *Nucleic Acids Res* **2006**, *34* (Database issue). https://doi.org/10.1093/NAR/GKJ067.

(99) Landrum, G. RDKit. 2018.

(100) Panesar, A. Preparing Data. *Machine Learning and AI for Healthcare* **2021**, 167–187. https://doi.org/10.1007/978-1-4842-6537-6_6.

(101) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and Computational Approaches to Estimate Solubility and Permeability in Drug Discovery and Development Settings. *Adv Drug Deliv Rev* **2001**, *46* (1–3), 3–26. https://doi.org/10.1016/S0169-409X(00)00129-0.

(102) Swamynathan, M. Mastering Machine Learning with Python in Six Steps. *Mastering Machine Learning with Python in Six Steps* **2017**. https://doi.org/10.1007/978-1-4842-2866-1.

(103) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel V. and Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer P. and Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(104) Chen, T.; Guestrin, C. XGBoost. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp 785–794. https://doi.org/10.1145/2939672.2939785.

(105) Heaton, Jeff. Introduction to Neural Networks with Java. **2008**, 438.

(106) Yotov, K.; Hadzhikolev, E.; Hadzhikoleva, S. Determining the Number of Neurons in Artificial Neural Networks for Approximation, Trained with Algorithms Using the Jacobi Matrix. *TEM Journal* **2020**, 1320–1329. https://doi.org/10.18421/TEM94-02.

(107) James, G.; Witten, D.; Hastie, T.; Tibshirani, R. Resampling Methods. **2013**, 175–201. https://doi.org/10.1007/978-1-4614-7138-7_5.

(108) Isik, M.; Rizzi, A.; Mobley, D. L.; Shirts, M.; Bergazin, D. T. MobleyLab/SAMPL6: SAMPL6 Part II - Release the Evaluation Results of Log *P* Predictions. **2019**. https://doi.org/10.5281/ZENODO.2651393.

(109) Bergazin, T. D.; Mobley, D. L.; Amezcua, M.; Grosjean, H.; Isik, M.; Slochower, D.; Chodera, J.; Tielker, N.; Ray, D.; Sasmal, S.; Murakumo, K. Samplchallenges/SAMPL7: Version 1.1: Update LogP Analysis; Release PHIP2 Analysis. **2021**. https://doi.org/10.5281/ZENODO.5637494.

(110) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *Journal of Cheminformatics* **2011**, *3* (1), 33. https://doi.org/10.1186/1758-2946-3-33.

(111) Scherrer, R. A.; Donovan, S. F. Automated Potentiometric Titrations in KCl/ Water-Saturated Octanol: Method for Quantifying Factors Influencing Ion-Pair Partitioning. *Analytical Chemistry* **2009**, *81* (7), 2768–2778. https://doi.org/10.1021/AC802729K/SUPPL_FILE/AC802729K_SI_001.PDF.

(112) Ropp, P. J.; Kaminsky, J. C.; Yablonski, S.; Durrant, J. D. Dimorphite-DL: An Open-Source Program for Enumerating the Ionization States of Drug-like Small Molecules. *Journal of Cheminformatics* **2019**, *11* (1), 1–8. https://doi.org/10.1186/S13321-019-0336-9/TABLES/3.

(113) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50* (5), 742–754. https://doi.org/10.1021/CI100050T/ASSET/IMAGES/MEDIUM/CI-2010-00050T_0018.GIF.

(114) Hall, L. H.; Mohney, B.; Kier, L. B. The Electrotopological State: An Atom Index for QSAR. *Quantitative Structure-Activity Relationships* **1991**, *10* (1), 43–51. https://doi.org/10.1002/QSAR.19910100108.

(115) Bickerton, G. R.; Paolini, G. v.; Besnard, J.; Muresan, S.; Hopkins, A. L. Quantifying the Chemical Beauty of Drugs. *Nature Chemistry* **2012**, *4* (2), 90–98. https://doi.org/10.1038/NCHEM.1243.

(116) Pearlman, R. S.; Smith, K. M. Software for Chemical Diversity in the Context of Accelerated Drug Discovery. *Drugs of the Future* **1998**, *23* (8), 885–895. https://doi.org/10.1358/DOF.1998.023.08.858430.

(117) Balaban, A. T. Highly Discriminating Distance-Based Topological Index. *Chemical Physics Letters* **1982**, *89* (5), 399–404. https://doi.org/10.1016/0009-2614(82)80009-2.

(118) Bertz, S. H. The First General Index of Molecular Complexity. *J Am Chem Soc* **1981**, *103* (12), 3599–3601. https://doi.org/10.1021/JA00402A071/ASSET/JA00402A071.FP.PNG_V03.

(119) Hall, L. H.; Kier, L. B. The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling. In *Reviews in Computational Chemistry*; Lipkowitz, K. B., Boyd, D. B., Eds.; John Wiley & Sons, Ltd, 1991; Vol. 2, pp 367–422. https://doi.org/10.1002/9780470125793.CH9.

(120) Labute, P. A Widely Applicable Set of Descriptors. *Journal of Molecular Graphics and Modelling* **2000**, *18* (4–5), 464–477. https://doi.org/10.1016/S1093-3263(00)00068-1.

(121) Gasteiger, J.; Marsili, M. Iterative Partial Equalization of Orbital Electronegativity—a Rapid Access to Atomic Charges. *Tetrahedron* **1980**, *36* (22), 3219–3228. https://doi.org/10.1016/0040-4020(80)80168-2.

(122) Chemical Computing Group. *QuaSAR-Descriptor*. QuaSAR-Descriptor. http://www.cadaster.eu/sites/cadaster.eu/files/challenge/descr.htm (accessed 2022-05-04).

(123) Ertl, P.; Rohde, B.; Selzer, P. Fast Calculation of Molecular Polar Surface Area as a Sum of Fragment-Based Contributions and Its Application to the Prediction of Drug Transport Properties. *J Med Chem* **2000**, *43* (20), 3714–3717. https://doi.org/10.1021/JM000942E.

(124) Kier, L.; Hall, L. *Molecular Structure Descriptoon: The Electrotopological State*; Academic Press, 1999.

(125) Wildman, S. A.; Crippen, G. M. Prediction of Physicochemical Parameters by Atomic Contributions. *Journal of Chemical Information and Computer Sciences* **1999**, *39* (5), 868–873. https://doi.org/10.1021/CI990307L/ASSET/IMAGES/LARGE/CI990307LF00002.JPEG.

(126) Tsantili-Kakoulidou, A.; Panderi, I.; Csizmadia, F.; Darvas, F. Prediction of Distribution Coefficient from Structure. 2. Validation of Prolog D, an Expert System. *Journal of Pharmaceutical Sciences* **1997**, *86* (10), 1173–1179. https://doi.org/10.1021/JS9601804.

(127) ChemAxon. *JChem for Office | ChemAxon*. https://chemaxon.com/products/jchem-for-office (accessed 2022-04-29).

(128) Cawley, G. C.; Talbot, N. L. C. On Over-Fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research* **2010**, *11*, 2079–2107.

(129) Martin, Y. C. Let's Not Forget Tautomers. *Journal of Computer-Aided Molecular Design* **2009**, *23* (10), 693–704. https://doi.org/10.1007/S10822-009-9303-2/TABLES/2.

(130) *How to compare regression models*. https://people.duke.edu/~rnau/compare.htm (accessed 2022-06-01).

(131) Chicco, D.; Warrens, M. J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Computer Science* **2021**, *7*, 1–24. https://doi.org/10.7717/PEERJ-CS.623/SUPP-1.

(132) Lee, A. C.; Crippen, G. M. Predicting PKa. *Journal of Chemical Information and Modeling* **2009**, *49* (9), 2013–2033. https://doi.org/10.1021/CI900209W/ASSET/IMAGES/LARGE/CI-2009-00209W_0001.JPEG.

(133) Işık, M.; Rustenburg, A. S.; Rizzi, A.; Gunner, M. R.; Mobley, D. L.; Chodera, J. D. Overview of the SAMPL6 PK a Challenge: Evaluating Small Molecule Microscopic and Macroscopic PK a Predictions. *Journal of Computer-Aided Molecular Design* **2021**, *35* (2), 131–166. https://doi.org/10.1007/S10822-020-00362-6/FIGURES/12.

(134) Bergazin, T. D.; Tielker, N.; Zhang, Y.; Mao, J.; Gunner, M. R.; Francisco, K.; Ballatore, C.; Kast, S. M.; Mobley, D. L. Evaluation of Log P, PK a, and Log D Predictions from the SAMPL7 Blind Challenge. *Journal of Computer-Aided Molecular Design* **2021**, *35* (7), 771–802. https://doi.org/10.1007/S10822-021-00397-3/TABLES/4.

(135) Rabel, S. R.; Sun, S.; Maurin, M. B.; Patel, M. Electronic and Resonance Effects on the Lonization of Structural Analogues of Efavirenz. *AAPS Journal* **2001**, *3* (4). https://doi.org/10.1208/PS030428.

Apéndices

Apéndices

El código, sets de datos y modelos principales utilizados en la elaboración de este trabajo de investigación se incluyen en el siguiente repositorio. Incluido se encuentra un jupyter notebook configurado para la predicción del coeficiente de distribución al introducir el SMILES de la molécula neutra, el SMILES de la molécula cargada y el pH deseado.

https://github.com/KennethLopPer/CBIO3_logD