UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO DE CIENCIAS E INGENIERIAS

Aplicación de modelos de Feature Selection y Machine Learning para identificar inhibidores potentes de la tirosinasa

Proyecto de Investigación

Pedro Santiago Salazar Casares

Ingeniería en Sistemas

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero en Sistemas

Quito, 22 de mayo de 2019

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

COLEGIO CIENCIAS E INGENIERIAS

HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN

Aplicación de modelos de Feature Selection y Machine Learning para identificar inhibidores potentes de la tirosinasa

Pedro Santiago Salazar Casares

Calificación: _____

Nombre del profesor, Título académico: Noel Pérez, PhD.

Firma del profesor _____

Nombre del profesor, Título académico: Yovani Marrero-Ponce, PhD.

Firma del profesor _____

Quito, 22 de mayo de 2019

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante: _____

Nombres y apellidos: Pedro Santiago Salazar Casares

Código: 00116701

Cédula de Identidad: 1722904206

Lugar y fecha: Quito, 22 de mayo de 2019

RESUMEN

Los inhibidores de la tirosinasa son fármacos utilizados para el tratamiento de la hiperpigmentación de la piel, pero la baja efectividad y seguridad de los inhibidores actuales exigen el continuo descubrimiento de nuevos compuestos de este tipo. Sin embargo, los métodos existentes *in vitro* e *in silico* (computacionales) para este fin presentan altos costos y una eficiencia limitada. Por lo tanto, es necesario obtener nuevas herramientas computacionales eficientes que permitan el descubrimiento de inhibidores potentes de la tirosinasa.

En este trabajo se implementaron nuevos modelos computacionales de clasificación, que combinan técnicas de selección de características (*Feature Selection*) y aprendizaje automático (*Machine Learning*), para la identificación de inhibidores potentes de la tirosinasa en bases de datos de moléculas químicas. Los mejores modelos de clasificación obtuvieron una exactitud total (*accuracy*) de 97.20% y 89.76% en el conjunto de validación, lo cual representa un avance significativo en la identificación de estos compuestos químicos.

Palabras clave: minería de datos, multiclasificación, feature selection, machine learning, inhibidores, tirosinasa.

ABSTRACT

Tyrosinase inhibitors are drugs used for the treatment of skin hyperpigmentation, but the low effectiveness and safety of current inhibitors require the discovery of new compounds of this kind. However, current *in vitro* and *in silico* (computational) methods for this purpose present high costs and limited efficiency. Therefore, it is necessary to get new and efficient computational tools that allow the discovery of strong Tyrosinase inhibitors.
This work presents computational classification models, which combine Feature Selection and Machine Learning techniques, for the identification of strong Tyrosinase inhibitors on chemical molecules databases. The best classification models obtained an accuracy of 97.20% and 89.76% in the validation set, which represents a significant advance in the identification of these chemical compounds.

Key words: data mining, multiclassification, feature selection, machine learning, inhibitors, tyrosinase.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS Y FIGURAS

ACRÓNIMOS

ACV: Accidente Cerebrovascular
ADL: Análisis discriminante lineal.
C4.5: Árbol de decisión C4.5.
CARDD*:* Diseño Racional de Fármacos Asistido por Ordenador.
DT: Tabla de decisión.
KNN: K vecinos más próximos.
MLP: Perceptrón Multicapa.
NB: *Naïve-Bayes.*
RB: Red bayesiana.
RF: *Random Forest*.
RLB: Regresión Logística Binaria.
ROC: Característica Operativa del Receptor
SVM: *Support Vector Machine.*
TOMOCOMD*:* Diseño Topológico Computacional de Moléculas.

CAPÍTULO 1: INTRODUCCIÓN

1.1 Antecedentes

La melanina es un pigmento que, entre otras funciones, protege a la piel de los efectos perjudiciales de la radiación ultravioleta (Brenner y Hearing, 2008). Sin embargo, la hiperpigmentación, ocasionada por la sobreproducción y acumulación de melanina causa problemas estéticos y enfermedades graves (Norlund y otros, 1998). Por ejemplo, el melanoma, el cual es un tipo de cáncer de piel común y agresivo, causa la muerte de una persona en el mundo cada hora (American Cancer Society, 2018). Como el melanoma y otras enfermedades exhiben un alto grado de pigmentación, se propone que la reducción de la melanogénesis (proceso de formación de la melanina) mejorarán sus diagnósticos y tratamientos (Riley, 1997).

La enzima tirosinasa regula el proceso de melanogénesis, por lo que es considerada la diana farmacológica de elección en el tratamiento de la hiperpigmentación (Chen y Kubo, 2002). Aunque los inhibidores de la tirosinasa se utilizan en el tratamiento de estos desórdenes de la piel, la baja efectividad y seguridad de estos químicos limitan su uso. Por lo tanto, la búsqueda de nuevos agentes inhibidores de esta enzima más seguros y eficaces sigue siendo de gran interés para las industrias farmacéuticas y cosméticas mundiales (Seo y otros, 2003).

El diseño y descubrimiento de fármacos asistido por computadores ha emergido como una alternativa para el mundo real de síntesis y bioensayos (Hann y Green, 1999). El cribado virtual (VS, por sus siglas en inglés *Virtual Screening*) es un proceso computacional que consiste en filtrar moléculas, a partir de bases de datos químicos, para seleccionar aquellas que tengan un alto potencial de convertirse en fármacos. El objetivo de aplicar este filtrado es seleccionar los mejores candidatos a fármacos, a partir de un enorme conjunto de compuestos posibles, para su posterior optimización. Las ventajas de su uso son el ahorro de tiempo y

dinero, así como la capacidad de analizar una cantidad de candidatos que puede rondar el orden de los billones (Scior et al, 2007).

1.1.1 Feature Selection.

Existen características redundantes e irrelevantes que incrementan la complejidad y disminuyen la exactitud de la clasificación. Por lo tanto, para mitigar esos efectos negativos y generar modelos prometedores, es importante la remoción de las características innecesarias.

La selección de características (*Feature Selection* en inglés) facilita la visualización y el entendimiento de los datos, disminuye los requisitos de medición y almacenamiento, reduce los tiempos y recursos computacionales del entrenamiento de modelos y limita la maldición de dimensionalidad (*curse of dimensionality* en inglés) para mejorar el rendimiento de las predicciones. Debido a las complicaciones que se presentan para buscar el subconjunto óptimo de características de entre todo el espacio de subconjuntos posibles, se suele elegir un subconjunto satisfactorio para el problema tratado.

Las técnicas de *Feature Selection* pueden ser categorizadas en dos paradigmas, en función de cómo realizan la búsqueda de las posibles hipótesis en el espacio de características con la construcción del modelo de clasificación: de filtrado o filtros (univariados y multivariados) y de envoltura.

Las técnicas de filtrado evalúan la relevancia de las características al observar solo las propiedades intrínsecas de los datos. En la mayoría de los casos, se calcula una puntuación de relevancia de la función y se eliminan las de puntuación baja. Posteriormente, este subconjunto de características se presenta como entrada al algoritmo o técnica de clasificación (Guyon y Elisseeff, 2003). A partir de este algoritmo, es posible variar las estrategias de búsqueda y las medidas de evaluación para diseñar diferentes modelos de filtros individuales. Dado que los métodos de filtrado son independientes de cualquier clasificador, no hereda ningún sesgo y será computacionalmente eficiente. Sin embargo, el

mejor subconjunto obtenido debe someterse a un estudio de correlación para resolver el problema de redundancia entre características.

Los métodos de envoltura, también llamados *wrappers*, integran la búsqueda de hipótesis del modelo y la búsqueda de subconjuntos de características en la misma configuración. Siempre y cuando se defina el procedimiento de búsqueda (para encontrar un posible subconjunto de características en todo el espacio de características), se generan y evalúan otros subconjuntos de características. La evaluación se realiza mediante una técnica de *Machine Learning* específica, lo que hace que este método sea extremadamente dependiente del clasificador empleado. Por lo tanto, el procedimiento de búsqueda se envuelve alrededor del modelo de clasificación. Sin embargo, como el espacio de los subconjuntos de características crece exponencialmente con el número de características, los métodos de búsqueda heurística (aleatorios y deterministas) tienen más probabilidades de usarse para guiar la búsqueda de un subconjunto óptimo (Guyon y Elisseeff, 2003).

Los algoritmos de envoltura y filtrado son muy similares. La función de evaluación es la principal diferencia. Los métodos de filtrado utilizan una medida independiente para la evaluación de cada subconjunto generado. Mientras tanto, los métodos de envoltura evalúan que tan buenos son ellos aplicando una técnica de *Machine Learning* sobre el subconjunto de características. Como ventaja, devuelve un subconjunto final de características que proporcionará mejores rendimientos de clasificación. Sin embargo, esta mejora tiene el inconveniente de que son más costosos computacionalmente que los métodos de filtrado.

Por otro lado, los métodos de *Feature Selection* también pueden ser categorizados dependiendo de la estrategia de búsqueda usado. Las siguientes estrategias o métodos de búsqueda son los más comúnmente usados (Ladha y Deepa, 2011):

- Selección hacia adelante
- Eliminación hacia atrás
- Selección hacia adelante *Stepwise*
- Eliminación hacia atrás *Stepwise*

- Mutación aleatoria

1.1.2 Machine Learning.

Arthur Samuel, pionero en el desarrollo de juegos informáticos e inteligencia artificial, en 1959 definió al aprendizaje automático (*Machine Learning* en inglés) como el "campo de estudio que da a las computadoras la capacidad de aprender sin ser explícitamente programado". Este campo interdisciplinario tiene relaciones estrechas con la inteligencia artificial, reconocimiento de patrones, minería de datos, estadística, teoría de la probabilidad, entre otros. En principio, los sistemas informáticos basados en *Machine Learning* funcionan de manera más consistente que los seres humanos.

Las técnicas de *Machine Learning* son usadas para el aprendizaje de máquinas sin intervención humana a partir de un conjunto de datos provistos. Estás técnicas buscan entrenar modelos que sean capaces de brindar predicciones con alta precisión y exactitud. Un buen predictor generalmente proporcionará predicciones con bajo sesgo y varianza en cualquier momento.

Existen diferentes tipos de aprendizaje para el entrenamiento de los modelos de *Machine Learning*. En el aprendizaje supervisado, los modelos aprenden a partir de las características de los datos y sus respectivas salidas las cuales son conocidas. Existen dos tipos de aprendizaje supervisado: clasificación y regresión. En la clasificación, las salidas pertenecen a un conjunto finito de categorías. Por otro lado, en los problemas de regresión la salida es un valor numérico continuo. En el aprendizaje no supervisado, los modelos no conocen el valor de salida, por lo que se entrenan a partir de la correlación de las características. El ejemplo más conocido de aprendizaje no supervisado es el problema de agrupación (más conocido por su denominación en inglés: *clustering*).

1.1.3 Multiclasificador.

Se conoce que un único modelo, o clasificador, no es suficiente para optimizar la predicción en ciertos problemas. Los multiclasificadores suelen mejorar el resultado de clasificación que ofrece el mejor modelo individual. Este rendimiento se obtiene tras combinar las salidas de los modelos obtenidos a partir de diferentes datos de entrenamiento, subconjuntos de características y técnicas de *Machine Learning*.

Los multiclasificadores pueden ser tan simples como una ponderación estadística o una combinación inteligente de las salidas (Rodríguez-Abed, 2007). Se han desarrollado algoritmos que sirven para cubrir problemas generales, como *Bagging* (Kuncheva y otros, 2002) y *Boosting* (Shipp y Kuncheva, 2002), aunque existen otros para tratar problemas específicos. Ambos son métodos de re-muestreo con reemplazamiento, en el primero de manera aleatoria y en el segundo reforzando los casos con mayor error de clasificación, que combinan las salidas usando la técnica de *voto pesado*.

1.2 Motivación y objetivo

Las limitantes de los compuestos químicos inhibidores de la tirosinasa y de los métodos *in silico* existentes hacen necesario el uso de herramientas computacionales eficientes, para identificar inhibidores potentes.

Como objetivo se propone implementar nuevos modelos computacionales eficientes, que combinen técnicas de *Feature Selection* y *Machine Learning*, para identificar inhibidores potentes de la tirosinasa en bases de datos de moléculas químicas

Como hipótesis se plantea: si se implementan nuevos modelos computacionales eficientes, que combinen técnicas de *Feature Selection* y *Machine Learning*, entonces se mejora las limitaciones existentes para identificar inhibidores potentes de la tirosinasa en bases de datos de moléculas químicas.

1.3 Tareas de investigación

Las tareas de investigación de este trabajo fueron las siguientes:

- Estado del arte de *Feature Selection, Machine Learning y* multiclasificadores.
- Estado del arte de *Feature Selection, Machine Learning y* multiclasificadores en estudios químicos.
- Selección de modelos a través del multiclasificador propuesto.
- Experimentación y validación de los modelos.

1.4 Organización del trabajo

La organización de este trabajo es la siguiente:

- El capítulo 1 presenta la introducción, la cual provee los antecedentes de la búsqueda de inhibidores de la tirosinasa. También contiene una descripción de *Feature Selection*, *Machine Learni*ng y multiclasificadores. Finalmente, se enuncian las limitantes que motivaron el desarrollo de este trabajo, sus objetivos, su hipótesis y las tareas de investigación.
- El capítulo 2 describe el estado del arte de los temas principales vinculados a este trabajo: (1) *Feature Selection*, (2) *Machine Learning*, y (3) Multiclasificadores.
- El capítulo 3 detalla el método propuesto y la metodología usada en este trabajo.
- El capítulo 4 presenta las conclusiones y recomendaciones de este trabajo.

CAPÍTULO 2: ESTADO DEL ARTE

2.1 Feature Selection

Junto con el avance exponencial de la tecnología en las últimas décadas, se ha experimentado un crecimiento similar en la capacidad de recolección de datos. Sin embargo, la posesión de una enorme cantidad de información, con decenas de miles de propiedades, resulta intratable si no se la aplica un correcto procesamiento de los datos. Una de las tareas más conocidas es la remoción de características irrelevantes y redundantes (*Feature Selection*). Existen múltiples objetivos de esta tarea, siendo los más importantes (Yu y otros, 2005):

- Evitar el *overfitting* y mejorar el rendimiento de los modelos
- Proveer modelos más rápidos y efectivos atendiendo al costo computacional

La selección del subconjunto óptimo de características para un problema de aprendizaje supervisado requiere una búsqueda exhaustiva de todos los posibles subconjuntos de la cardinalidad escogida. En aplicaciones prácticas de *Machine Learning*, usualmente se busca un subconjunto satisfactorio de variables en lugar del óptimo.

Roy y otros (2015) usaron una red neuronal profunda (un algoritmo de *Machine Learning* perteneciente a *Deep Learning*) para seleccionar las 30 mejores características, dentro de un espacio inicial de 426, que permitiría alcanzar un mejor reconocimiento de movimientos en videos. El método propuesto consistió en analizar la contribución individual de cada característica al potencial de activación de la primera capa intermedia de la red neuronal. Este es un nuevo enfoque que difiere del proceso convencional de selección de características usando *wrappers*. Sin embargo, en dicho estudio no se realizó una comparación con respecto a otros clasificadores. Ellos decidieron comparar con la técnica de

reducción de espacio como el análisis de componentes principales (PCA por sus siglas en inglés).

Qin y compañía (2015) utilizaron un nuevo método de selección de características basado en *Deep Learning* para obtener una mejor clasificación de escenas de teledetección. Usando redes de creencia profunda, su enfoque se basa en la selección de características en base al error de reconstrucción de las entradas obtenido a partir del entrenamiento no supervisado. Este enfoque se destaca por la no generación exhaustiva de subconjuntos. El modelo inicia con todo el conjunto de características y entrega un subconjunto teóricamente ideal. La desventaja de este enfoque es que se hace un filtrado en base al valor del error de reconstrucción de cada entrada (característica), sin tomar en cuenta explícitamente la redundancia entre ellas. Sin embargo, la redundancia podría ser solucionada por las relaciones que se generan entre todas las características durante el entrenamiento de la red neuronal. El enfoque presentado por Qin y su grupo de investigación también es rescatado como una alternativa a los métodos de selección de características convencionales (filtros y *wrappers*).

Dalia y otros (2019) presentaron un método de *Feature Selection* para superar las limitaciones existentes en el reconocimiento de escritura a mano. La gran variabilidad de escritura de cada persona hace que la selección de conjuntos de características apropiadas sea compleja. En este estudio, se introdujo un enfoque compuesto por (1) diferentes medidas univariadas para producir un ranking de características y (2) un método de búsqueda exhaustiva para elegir el subconjunto de características que maximice los resultados de clasificación. En los experimentos, se consideró uno de los conjuntos de características más efectivo y ampliamente utilizado para este problema. Los resultados experimentales, obtenidos mediante el uso de bases de datos de palabras reales de caracteres escritos a mano, confirmaron la efectividad de la propuesta.

Salah y otros (2003) usaron algoritmos genéticos como métodos de *Feature Selection* para mejorar la clasificación de vinos en Chile. Ellos señalaron que la clasificación de vinos, tanto por su variedad como su lugar de producción, se realiza procesando diferentes características físicas (color, densidad), químicas (fenoles, aminoácidos) y organolépticas (aromas, sabores). Para una mejor clasificación de cepas de vinos y tras contar con un cuantioso número inicial de propiedades, se deben trabajar con las que realmente aporten información vital al problema, produzcan menor cantidad de errores y no sean redundantes entre sí. Se seleccionaron 29 características, partiendo de un conjunto de 6751, las cuales permitieron una clasificación correcta del 99.1%.

Ying (2004) hizo un estudio comparativo de métodos de *Feature Selection* para el descubrimiento de medicamentos para una agresiva reducción de dimensionalidad. En esta investigación logró remover hasta el 99% de características. Utilizó dos algoritmos de aprendizaje muy conocidos: SVM y *Naïve-Bayes*. Comprobó que el primero no se benefició tanto de la reducción de características como el segundo. Por otro lado, los métodos más efectivos para *Feature Selection* resultaron ser aquellos que implementaron los criterios de selección *Information Gain* y Chi Cuadrado, Aquellos métodos que utilizaron el criterio información mutua (*Mutual information*) brindaron los peores resultados.

Zhi-Zhong y otros (2018) hicieron el lanzamiento de un programa de computación denominado ECoFFeS. Este programa simple y eficiente permite a los desarrolladores de medicamentos aprovechar la computación evolutiva para *Feature Selection*. La computación evolutiva es definida como una rama de la inteligencia artificial "dedicada al estudio de una clase de algoritmos basados en los principios Darwinianos de la selección natural" (Melián, Moreno-Pérez y Moreno-Vega, 2009). El uso de algoritmos de este campo permite la resolución de problemas de optimización combinatoria. ECoFFeS integra algoritmos evolutivos, combinadores de evaluación, técnicas de clasificación y regresión. También

cuenta con una técnica de ejecución en paralelo para reducir el tiempo de análisis total (Zhi-Zhong et al, 2018).

2.2 Machine Learning

Machine Learning ofrece un gran potencial para una clasificación efectiva y eficiente de una gran variedad de problemas de la vida cotidiana.

Navarro, Fernández, Borraz y Alonso (2017) presentaron un sistema autónomo basado en sensores para detectar peatones en una aplicación de un vehículo autónomo. Eso se logró mediante un procesamiento de los datos y aplicación de algoritmos de *Machine Learning*. El trabajo detalla el análisis exhaustivo del rendimiento de tres algoritmos distintos: *Naïve-Bayes*, k vecinos más cercanos (KNN) y máquinas de soporte vectorial (SVM). Los algoritmos fueron entrenados con 1931 casos. El rendimiento final del método midió un escenario de tráfico real, que contenía 16 peatones y 469 muestras de no peatones, mostrando un *accuracy* de 96.2%, sensibilidad de 81.2%, y especificidad de 96.8%.

Addo, Guegan y Hassani (2018) construyeron modelos de clasificación binarios, basados *Machine Learning* y *Deep Learning*, en datos reales para predecir la probabilidad de incumplimiento del préstamo. Las 10 características más importantes de estos modelos fueron conservadas (a partir de un conjunto de 181) y usadas en el proceso de modelado para probar la estabilidad de los clasificadores binarios, comparando su desempeño en datos externos. Los autores concluyen que los modelos basados en árboles son más estables que los modelos basados en redes neuronales artificiales multicapa, obteniendo en sus mejores modelos un *accuracy* de 99.3% y 97.9% en el conjunto de validación, respectivamente. Este escenario permite cuestionar el uso intensivo de *Deep Learning* en las empresas bancarias.

El análisis de discriminante lineal (ADL) ha sido una de las técnicas más utilizadas en el descubrimiento y diseño de fármacos (González-Diaz y otros, 2005). Otras técnicas, como el método de *k*-Vecinos más Cercanos (KNN, por sus siglas en inglés) (Abidin y Perizzo,

2006), redes bayesianas y árboles de decisión, también se han expandido en esta área de trabajo.

Louvina (2010) buscó identificar nuevos compuestos que detengan la actividad de la leishmaniasis (una enfermedad parasitaria). Mediante el uso de análisis discriminante lineal (ADL) en un conjunto de 1357 instancias (366 activos), consiguió un *accuracy* de 76.40%. Dicho resultado, sin embargo, fue superior a otros obtenidos por investigaciones similares. El bajo desempeño pudo estar condicionado al uso de un conjunto de entrenamiento con clases bastante desbalanceadas.

Poorinmohammad y otros (2014) usaron máquinas de soporte vectorial (SVM en inglés) para crear modelos de predicción de péptidos anti-VIH (virus de inmunodeficiencia humana). La clasificación con modelos computacionales alcanzó un *accuracy* de 96.76%. Un posible sesgo se evitó tras balancear el número de compuestos activos e inactivos. Los resultados del estudio fueron muy prometedores para el hallazgo, con un pequeño margen de error, de péptidos candidatos que combatan al VIH.

Malik y otros (2016) hicieron una detección no invasiva del nivel de glucosa en la sangre en ayunas tras analizar los parámetros electroquímicos en la saliva. En un estudio cuya mayor debilidad fue la escasa cantidad de casos (175), al usar las técnicas de regresión logística, red neuronal artificial y la máquina de soporte vectorial se alcanzó un *accuracy* aproximado de 85% (con la última técnica).

2.3 Multiclasificador

Dietterich (2000) en uno de sus estudios justificó cómo un multiclasificador puede ser mejor que un clasificador individual de tres modos: estadística, figurativa y computacional. Sin embargo, a pesar del uso intensivo de *Feature Selection* y *Machine Learning* en distintos problemas, los multiclasificadores no son utilizados con mucha frecuencia. En ocasiones, las soluciones solo se limitan a reducir al mejor subconjunto de características, obtenido por el

uso de un solo método de *Feature Selection*, en combinación con una única técnica de *Machine Learning*, para la construcción de un solo modelo. Se debería procurar por la variedad e integración de modelos obtenidos a partir de distintos datos de entrenamiento, subconjuntos de características y técnicas de *Machine Learning*.

Takemura, Shimizu y Hamamoto (2010) propusieron un multiclasificador basado en el algoritmo *AdaBoost* y *Feature Selection* para la clasificación de tumores de mama en imágenes ultrasónicas. Se calcularon un total de 208 características para la discriminación. Se utilizó un multiclasificador entrenado por un algoritmo de aprendizaje del tipo *AdaBoost* multiclase (*AdaBoost.M2*), combinado con un proceso de selección de características secuencial. Una prueba de validación cruzada de 10 iteraciones validó el rendimiento y los resultados se compararon con los de un clasificador basado en la distancia de Mahalanobis y una técnica SVM multiclase. Un total de 200 carcinomas, 50 fibroadenomas y 50 quistes fueron utilizados en los experimentos, demostrándose que la combinación de un clasificador entrenado por *AdaBoost.M2* es útil para la discriminación de los tumores.

Xie y otros (2017) desarrollaron un método novedoso para clasificar los tumores melanocíticos como benignos o malignos mediante el análisis de imágenes de dermatoscopia digital. Los experimentos se llevaron a cabo en dos bases de datos diversas que incluyen imágenes de razas caucásicas y de coloración amarilla. El algoritmo sigue tres pasos importantes: primero, las lesiones se extraen utilizando una red neuronal autogenerada; segundo, se extraen las características descriptivas del color, la textura y el borde; y tercero, las lesiones se clasifican con un multiclasificador que combina redes neuronales de propagación hacia atrás (*backpropagation* en inglés) con redes neuronales difusas para lograr un mejor rendimiento. Los resultados muestran que la precisión de la clasificación (94.17% y 91.11% en cada base de datos) se mejoró sustancialmente por el uso de las nuevas funciones de borde y el multiclasificador propuesto.

Subudhi y otros (2019) presentaron un método automatizado basado en un sistema de decisión para detectar el accidente cerebrovascular (ACV) isquémico en imágenes de resonancia magnética. Fueron utilizadas dos técnicas de *Machine Learning* para construir los modelos de clasificación para este sistema: SVM y el multiclasificador *Random Forest*. Un total de 192 exploraciones de resonancia magnética se consideraron para la evaluación. El sistema propuesto detectó de manera eficiente las lesiones por ACV con una precisión del 93,4% utilizando *Random Forest*, el cual superó ligeramente el desempeño del clasificador SVM.

Huong y otros (2015) propusieron el uso de sistemas multiclasificadores para la predicción de agentes despigmentantes. Para tal fin, se obtuvo una inmensa variedad de modelos individuales a través del uso de diferentes técnicas de *Machine Learning* en dos conjuntos de datos. Esto permitió el desarrollo de modelos que clasifican la capacidad de compuestos para despigmentar la piel y otro grupo de modelos que determinan la potencia de despigmentación. Producto de esta investigación se obtuvieron en las series de predicción un *accuracy* de 95.52% para los primeros modelos y 88.89% para los segundos. Los modelos permitieron encontrar 31 nuevos compuestos inhibidores e identificar 52 inhibidores como potentes, todo corroborado con ensayos en laboratorio.

CAPÍTULO 3: MÉTODO PROPUESTO

3.1 Descripción del multiclasificador

En este trabajo se utilizó un multiclasificador para la obtención de modelos a partir de dos conjuntos de datos diferentes. Los modelos obtenidos a partir del primer conjunto permitirán clasificar la capacidad de moléculas químicas para inhibir la tirosinasa. Los modelos obtenidos a partir del segundo conjunto determinarán la potencia con la que las moléculas pueden inhibir a la enzima.

El multiclasificador utilizó técnicas de *Feature Selection* para obtener diferentes subconjuntos de características. Se elaboraron métodos de filtrado, de envoltura y evaluadores de correlación entre subconjuntos. Los métodos de filtrado emplearon los criterios de selección *Information Gain* (Ambielli, 2017), *Relief* (Robnik-Sikonja y Kononenko, 1997), *One R* y Chi cuadrado (Quevedo, 2011). También se usó *CfsSubsetEval* (Hall, 1998), un evaluador de subconjuntos basado en correlación, con los métodos de búsqueda *Best First*, *Greedy Stepwise* y algoritmo genético (Da Silva, 2011). El evaluador de correlación resuelve la redundancia de características que los métodos de filtrado no solucionan. Para los métodos de envoltura se usaron los tres métodos de búsqueda mencionados y como evaluador *WrapperSubsetEval*, un esquema de evaluación de atributos que usa técnicas de *Machine Learning* para obtener los subconjuntos. El propósito de usar diferentes métodos de *Feature Selection* es obtener un gran número de subconjuntos que representen diferentemente al conjunto original de características.

A partir de cada subconjunto, el multiclasificador entrenó modelos utilizando diferentes técnicas de *Machine Learning*. Las técnicas se emplearon en dos ámbitos diferentes: (1) por los *wrappers* en selección de características (como algoritmos de aprendizaje) y (2) para construir los modelos: redes bayesianas (RB) (Friedman y otros, 1997), *Naïve Bayes* (NB) (Ying, 2004), el Análisis Discriminante Lineal (ADL) (Raschka, 2014), regresión logística

binaria (RLB) (Hosmer y Lemeshow, 1989), la red perceptrón multicapa (MLP) (Ruck y otros, 1990), las Máquinas con Soporte Vectorial (SVM por sus siglas en inglés) (Christianini y Shawe-Taylor, 2000), K vecinos más próximos (KNN por sus siglas en inglés), tabla de decisión (Kohavi, 1995), *One R* (Holte, 1993), el árbol C4.5 (Quinlan, 1993) y *Random Forest* (RF) (Breiman, 2001).

También se usaron diferentes técnicas *ensemble* para la construcción de modelos: *Bagging*, *Boosting* (usando el método *AdaBoostM1*) (Freund y Schapire, 1996), *Stacking* (Wolpert, 1992) y voto (mayoritario o promedio de probabilidades). Para los dos primeros se usaron las técnicas de *Machine Learning* antes descritas. Para los otros dos se usaron diversas combinaciones de las mismas técnicas. Los modelos obtenidos con las técnicas *ensemble* también fueron considerados modelos individuales, pues igualmente fueron usados para alimentar a los modelos de multiclasificación. El propósito de usar diferentes técnicas de *Machine Learning* es contar con una alta diversidad de fronteras de decisión para ser aprovechadas en la combinación de los modelos.

Después, se obtuvieron modelos de multiclasificación a través del cumplimiento de dos requisitos: obtener un conjunto de modelos individuales diversos y elegir el método apropiado de combinación.

Una alta diversidad en la construcción de modelos individuales puede lograrse:

- Cambiando el conjunto de entrenamiento, como lo hacen *bagging* y *boosting*.
- Manipulando el conjunto de atributos.
- Obteniendo modelos entrenados con diferentes técnicas de clasificación.

La combinación de las salidas de los modelos individuales puede realizarse de dos formas: selección o fusión. En el primero tan solo se elige al "mejor" clasificador. En el segundo utiliza una función para combinar las salidas de los diferentes clasificadores (Kuncheva, 2014). El enfoque de fusión fue usado para este multiclasificador. Se empleó una

función que permitió la construcción de nuevos conjuntos de datos. Estos fueron usados para obtener modelos de multiclasificación.

Un modelo calcula la probabilidad de pertenencia de cada instancia a cada una de las clases. En base a esos valores (salidas), el modelo asigna la instancia a la clase con el mayor valor de probabilidad. Como este problema es de clasificación binaria, los modelos generan dos salidas por cada instancia evaluada. Para que los modelos devuelvan un valor por instancia en lugar de dos, se utilizó la función del cálculo de diferencia de probabilidades (ΔP%) para unificarlos:

$$\Delta P\% = 100\ (P_{activo} - P_{inactivo}) \qquad \textbf{(1)}$$

$$\Delta P\% = 100\ (P_{potente} - P_{débil}) \qquad \textbf{(2)}$$

Para modelos de activación, en la ecuación (1) P_{activo} representa la probabilidad de que un compuesto sea activo, $P_{inactivo}$ de que sea inactivo. Para modelos de potencia, en la ecuación (2) $P_{potente}$ representa la probabilidad de que un compuesto sea potente, $P_{débil}$ de que sea moderado o débil. De esta manera, los modelos (activación/potencia) clasifican los compuestos que tienen un ΔP% > 0 como activos/potentes y aquellos con ΔP% < 0 como inactivos/débiles, en un rango de -100 a 100.

Se formaron conjuntos de datos de probabilidades al juntar estas salidas. Las instancias de estos nuevos conjuntos de datos serían las moléculas de los conjuntos originales, mientras que las características serían los modelos. A partir de esta concepción, se procedió a construir modelos de multiclasificación usando el mismo procedimiento de obtención de modelos individuales.

La figura 1 presenta el diagrama del multiclasificador propuesto y sus cuatro tipos de diversidad (datos, variables o características, modelos clasificadores y las combinaciones de ellos).

Los criterios para seleccionar los mejores modelos fueron:

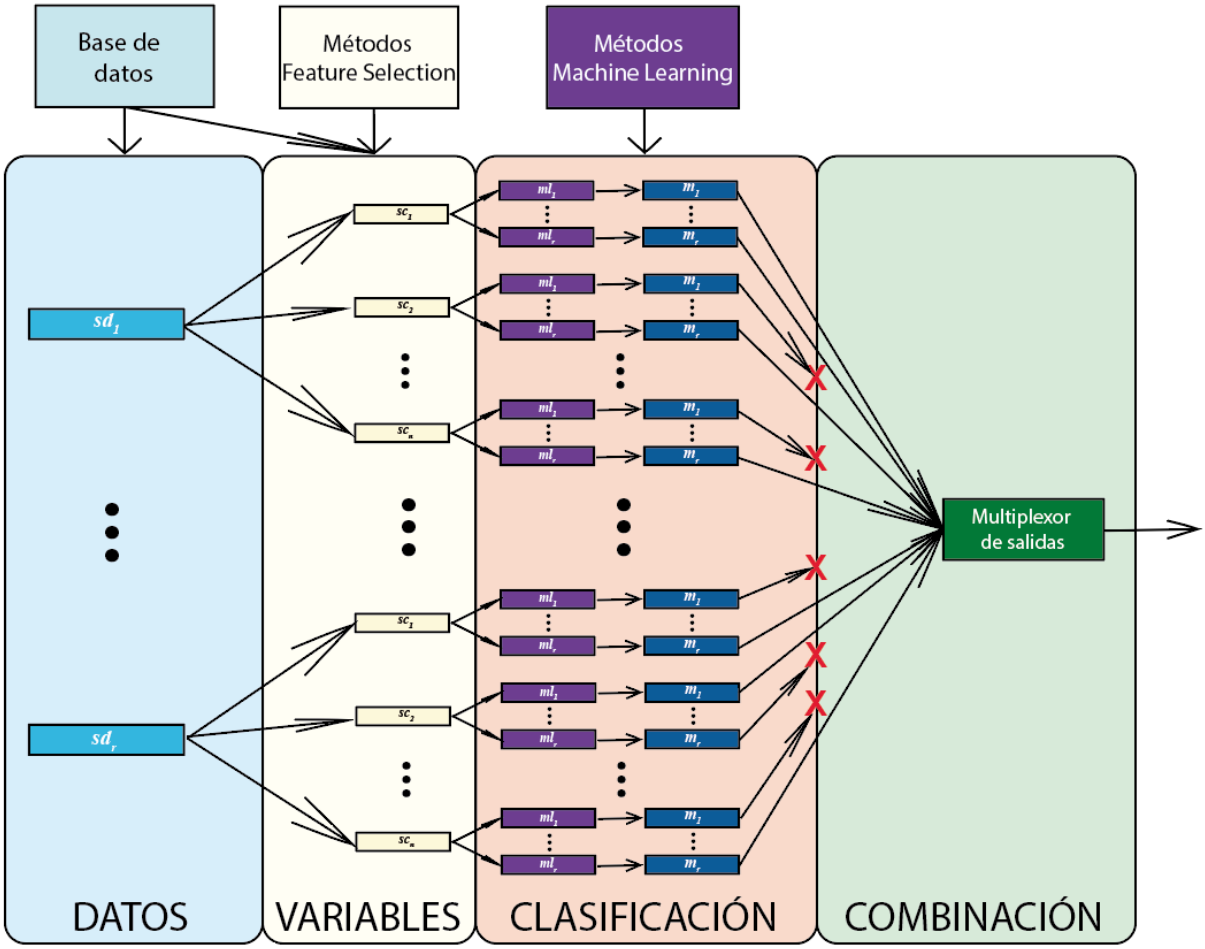

Figura 1: Diagrama del multiclasificador propuesto.

- Valor de *accuracy* en el conjunto de validación, comparándolo con:
    - El *accuracy* de los modelos entrenados a partir del conjunto original de variables.
    - El *accuracy* de los modelos (1) obtenidos de un mismo subconjunto o (2) entrenados por la misma técnica de *Machine Learning*.
- Tamaño de los subconjuntos (por menor número de variables). Se basa en el principio de parsimonia ("*Occam's Razor*"), el cual indica que cuando las explicaciones múltiples para un fenómeno ofrecen resultados iguales o muy similares se elige la más simple (Ariew, 1976).
- Nivel de correlación entre los subconjuntos. Idealmente se buscan modelos obtenidos a partir de subconjuntos sin características compartidas.

- Según las técnicas de *Machine Learning* usadas para sus entrenamientos. Basados una vez más en el principio de parsimonia, se eligieron los modelos obtenidos a partir de las técnicas de mayor eficiencia algorítmica.

3.2 Metodología experimental

La metodología de este trabajo se compone de:

- Conjuntos de datos experimentales
- Configuración del método propuesto
- Experimentación y validación de los modelos

3.2.1 Conjuntos de datos experimentales.

El presente trabajo se ha realizado usando dos conjuntos de datos químicos de clasificación binaria. Las características de este problema, que describen cuantitativamente propiedades de las moléculas, se calcularon con el módulo ***CARDD*** (acrónimo de Diseño Racional de Fármacos Asistido por Ordenador), implementado en el programa ***TOMOCOMD*** (acrónimo de Diseño Topológico Computacional de Moléculas) (Marrero-Ponce y Romero, 2002). Para este trabajo, no se realizó un procesamiento de datos debido a que los dos conjuntos fueron entregados como se describe a continuación:

- El primer conjunto se compone inicialmente de 1429 casos (701 activos y 728 inactivos) y 118 características. La salida de este conjunto determina si un compuesto es capaz de inhibir a la tirosinasa. Los modelos entrenados con este conjunto de datos serán denominados modelos de activación.
- El segundo conjunto se compone inicialmente de 515 casos (339 potentes y 176 débiles) y 66 características. La salida de este conjunto determina si un compuesto activo inhibe a la tirosinasa potentemente o de forma moderada o débil. Los modelos entrenados con este conjunto de datos serán denominados modelos de potencia.

Ambos conjuntos son particionados en conjuntos de entrenamiento y validación. Los conjuntos de entrenamiento contaron con mayor número de muestras y sirvieron para fijar los parámetros de los modelos. Los conjuntos de validación se usaron para evaluar las predicciones de los modelos entrenados, por lo que permite ajustar sus hiperparámetros.

La tabla 1 y 2 muestran la distribución de las instancias (para entrenamiento y validación) de los dos conjuntos de datos proporcionados para esta investigación:

Conjunto 1	Activos	Inactivos	Total
Entrenamiento	526	546	***1072***
Validación	175	182	***357***
Total	***701***	***728***	**1429**

Tabla 1: Repartición del primer conjunto de datos en entrenamiento y validación.

Conjunto 2	Potentes	Débiles	Total
Entrenamiento	255	133	***388***
Validación	84	43	***127***
Total	***339***	***176***	**515**

Tabla 2: Repartición del segundo conjunto de datos en entrenamiento y validación.

3.2.2 Configuración del método propuesto

En los métodos de filtrado se establecieron los *rankings* usando los siguientes criterios de evaluación:

- *Information Gain*
- *Relief*, que utilizó los 10 vecinos más cercanos. No los asignó un peso preferencial en función de la distancia. Todas las instancias fueron utilizadas para la estimación de características.
- O*ne R*, que usó validación cruzada de 10 iteraciones.
- Prueba de Chi cuadrado.

Los métodos de búsqueda, usados tanto por el evaluador de correlación de características como por los métodos de envoltura, fueron *Best First*, *Greedy Stepwise* y algoritmo genético.

- *Best First* realizó la búsqueda hacia adelante (inicia con un conjunto vacío de variables y va insertando una a la vez hasta encontrar un subconjunto satisfactorio). Se esperaron 5 nodos consecutivos sin mejora para terminar la búsqueda.
- *Greedy Stepwise* realizó la búsqueda exhaustiva hacia adelante.

- En el algoritmo genético usó una probabilidad de recombinación del 60%, una probabilidad de mutación del 3.3%, un tamaño de la población de 20 y corrió un máximo de 20 generaciones.

Para los métodos de envoltura y construcción de modelos, se usaron los siguientes algoritmos y sus correspondientes configuraciones de los hiperparámetros más importantes:

- Las redes bayesianas utilizaron un estimador simple con un valor *alpha* de 0.5, el algoritmo de búsqueda K2 con *score* Bayes y máximo un nodo padre. Inician como una red *Naïve-Bayes*.
- *Naïve-Bayes.*
- ADL utiliza un valor de *ridge* de $1 * 10^{-6}$.
- La regresión logística utilizó un valor de *ridge* de $1 * 10^{-8}$.
- La red MLP contó con 500 épocas de entrenamiento, *learning rate* de 0.3 sin decaimiento, *momentum* de 0.2 y tres capas (una de entrada, una intermedia y una de salida). El número de neuronas de la capa de entrada era el número de características del subconjunto en uso. El número de neuronas de la capa intermedia se calculó mediante la función:

$$n = (a + c)\ /2 \qquad (3)$$

  donde *n* es el número de neuronas de la capa intermedia, *a* es el número de características y *c* el número de clases. La capa de salida tuvo dos neuronas (una para cada clase). La función de activación de todas las neuronas fue *Sigmoid* (Han y Morag, 1995).
- SVM utilizó la función *kernel* polinomial grado 1 como núcleo, regresión logística como calibrador (misma configuración que la anteriormente descrita), un valor de épsilon de $1 * 10^{-12}$, el parámetro de coste C de 1.0 y una tolerancia de 0.001.

- Para el KNN se usó solo al vecino más próximo (k=1) y como algoritmo de búsqueda la distancia euclidiana (es la distancia en línea recta de dos puntos en un espacio euclidiano).
- La tabla de decisión utilizó *Best First* como algoritmo de búsqueda, la validación cruzada *leave-one-out* y la raíz del error cuadrático medio (RMSE por sus siglas en inglés) como medida de evaluación.
- O*ne R* usó validación cruzada de 10 iteraciones.
- El árbol C4.5 utilizó el factor de confianza para la poda de 0.25. Un valor más pequeño aumenta la cantidad de podas. Se trabajó con un mínimo de 2 instancias por hoja.
- *Random Forest* utilizó 100 árboles y 100 iteraciones. Para el número de características aleatorias elegidas se utilizó la siguiente función:

$$n = \log_2(p) + 1 \qquad (\mathbf{4})$$

donde *n* es el número de características elegidas y *p* es el número de predictores.

Para todos los algoritmos se trabajó con un tamaño de lote (*batch size*) de 100 bloques.

Debido a las ventajas que las redes neuronales presentan frente a otras técnicas (Donges, 2018), se realizó un trabajo más exhaustivo utilizando la red MLP. Está técnica fue usada como algoritmo de aprendizaje dentro de muchos modelos *wrapper*, garantizando su superioridad con respecto a otras técnicas en los métodos de *Feature Selection*. La diversidad en este grupo se consiguió alterando dos parámetros de los MLP: el número de épocas de entrenamiento y el *learning rate* (taza de aprendizaje). El número de épocas varió entre 500 y 2500 con intervalos de 500 unidades. El *learning rate* varió entre 0.05 y 0.3 con intervalos de 0.05 unidades.

Las técnicas *ensemble* usadas para la construcción de modelos fueron *Bagging*, *Boosting* (en su modalidad *AdaBoostM1*), *Stacking*, voto mayoritario y voto promedio de probabilidades.

- *Bagging* usó como clasificador todas las técnicas antes descritas, una a la vez. Utilizó 10 hilos diferentes y realizó 10 iteraciones.
- *AdaBoostM1* usó como clasificador todas las técnicas antes descritas, una a la vez. Utilizó 10 hilos diferentes y el valor 100 como umbral de peso para la poda.
- Para *Stacking*, se realizaron combinaciones de tres técnicas de *Machine Learning* diferentes entre sí para alimentar a otra técnica distinta. Se usó la validación cruzada de 10 iteraciones para la validación interna de los modelos generados.
- Para la técnica de voto se utilizaron dos reglas de combinación: voto mayoritario o por promedio de probabilidades (promedio de las salidas de los modelos).

Para el desarrollo de este trabajo se utilizaron las implementaciones de las técnicas en Weka, el cual es un paquete de software escrito en Java para trabajos de minería de datos y *Machine Learning* (Witten y otros, 2016).

3.2.3 Experimentación y validación de los modelos.

Los pasos de experimentación del método propuesto para la obtención de modelos, que evalúen la capacidad y potencia de inhibición de la tirosinasa, fueron los siguientes:

- Primero, se aplicaron métodos de *Feature Selection* sobre los conjuntos de datos, adquiriendo diversos subconjuntos ‘óptimos’ de características.
- Segundo, se crearon modelos partiendo de los subconjuntos de características y usando varias técnicas de *Machine Learning*. Cada técnica se utilizó sobre cada subconjunto, obteniendo así una gran diversidad de modelos.
- Tercero, los modelos fueron validados para comprobar su confiabilidad y eficiencia con datos externos.

- Cuarto, se discriminaron los modelos en varias etapas usando los criterios de selección propuestos (Sección 3.1).
- Quinto, se crearon conjuntos de datos de probabilidades a partir de las salidas de los mejores modelos individuales.
- Sexto, se crearon modelos de multiclasificación a partir de esos nuevos conjuntos de datos.
- Séptimo, se discriminaron los nuevos modelos usando nuevamente los criterios de selección propuestos.

La diversidad entre los modelos individuales es un factor determinante para el resultado final de los modelos de multiclasificación. Kuncheva (2014) planteó cuatro niveles por donde se puede introducir diversidad. En este trabajo, la diversidad se obtuvo de la siguiente manera en cada nivel:

- A nivel de base de datos, se crearon diferentes conjuntos de entrenamiento utilizando los métodos *ensemble Bagging* y *Boosting*.
- A nivel de atributos, se obtuvieron los modelos a partir de subconjuntos diferentes.
- A nivel de clasificador, se emplearon diversas técnicas (RB, NB, ADL, RLB, MLP, SVM, IBK, DT, *One R*, C4.5, RF) que crearon diferentes fronteras de decisión del conjunto de entrenamiento (modelos individuales).
- A nivel de combinación, se diversificaron los multiclasificadores empleando *Bagging*, *Boosting*, *Stacking* y voto ponderado y no ponderado. También se crearon modelos de multiclasificación que se entrenaron en base a las salidas de los mejores modelos individuales.

Los modelos debieron someterse a la validación interna y externa para comprobar su calidad de predicción. La validación interna permite medir el rendimiento interno de los

modelos. En este caso, se efectuó la validación cruzada con el 5, 10, 15, 20, 25 y 30% dejando para la predicción (*Leave-Group-Out* en inglés).

La validación externa determina la capacidad de predicción en datos que no ha visto. Puede ocurrir que el modelo no generalice bien a pesar de haberse entrenado adecuadamente (*overfitting*), o que sea muy sencillo para el problema tratado (*underfitting*). Se utilizó la medida de la cantidad de casos correctamente clasificados, o exactitud total (*accuracy*, ***Ac***) (Baldi y otros, 2000):

		Predicción	
		Positivos	Negativos
Real	Positivos	Verdaderos positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos positivos (FP)	Verdaderos negativos (VN)

Tabla 3: Matriz de confusión binaria.

$$Ac = 100\frac{VP+VN}{VP+FP+VN+FN} \quad (5)$$

3.3 Resultados

3.3.1 Resultados del primer conjunto de datos.

A partir del primer conjunto de datos:

- Se obtuvieron 70 subconjuntos de características heterogéneos.
- Se consiguieron 1181 modelos tras la aplicación de las técnicas de *Machine Learning* en todos los subconjuntos obtenidos.
- Se seleccionaron los 13 mejores modelos, construidos a partir de 9 subconjuntos y 9 técnicas diferentes.
- Estos mejores modelos individuales fueron combinados para obtener 66 modelos de multiclasificación.
- Se seleccionaron los dos mejores modelos de este tipo, construidos a partir de 5 modelos y 2 técnicas distintas.

- El *accuracy* de los mejores modelos fue mayor al de sus similares obtenidos con el conjunto original de variables (118).

La tabla 4 resume los 5 mejores modelos individuales y los 2 mejores de multiclasificación para el primer conjunto de datos (modelos de activación).

La primera columna sirve para enumerar los modelos. La segunda columna detalla las técnicas de *Machine Learning* usadas para el entrenamiento de los modelos. La tercera columna muestra (1) el tamaño de los subconjuntos de características usados para la obtención de los modelos individuales, y (2) los modelos individuales usados para el entrenamiento de los modelos de multiclasificación. La cuarta columna exhibe el *accuracy* de los modelos en el conjunto de validación.

Mejores modelos individuales			
Modelo	Técnica de Machine Learning	Subconjunto (# características)	Accuracy (%)
1	Random Forest	23 (1)	96.36
2	AdaBoostM1 (C4.5)	23 (1)	96.36
3	Voto mayoritario (MLP, RF, C4.5)	18	96.36
4	Stacking (C4.5 – NB,SVM,RF)	14	96.08
5	KNN	23 (2)	94.12
Mejores modelos de multiclasificación			
Modelo	Técnica de Machine Learning	Subconjunto (modelos)	Accuracy (%)
A	ADL	2, 4, 5	97.20
B	Red bayesiana	1, 3, 4, 5	97.20

Tabla 4: Mejores modelos de activación.

La tabla 4 demuestra la eficiencia de los modelos individuales. Ellos logran clasificar correctamente más del 94% de casos externos usando una pequeña cantidad de características iniciales. Por ejemplo, el modelo 4 obtuvo un *accuracy* de 96.08% en el conjunto de validación empleando solo 14 de las 118 características originales.

La combinación de estos modelos eficientes permitió obtener modelos de multiclasificación con un mayor desempeño que los individuales. El mejor modelo (*accuracy* de 97,20% en el conjunto de validación) fue obtenido a partir de las salidas de tres modelos individuales y entrenado con la técnica ADL. Estos modelos se obtuvieron con la aplicación

de tres técnicas de *Machine Learning* (KNN, *Boosting* y *Stacking*) sobre tres subconjuntos diferentes (de 23, 23 diferente al anterior, y 14 variables).

A partir de la tabla 4, se observa que el *accuracy* de los modelos individuales es muy similar, siendo el mismo en tres de ellos (modelos 1, 2 y 3). Esta situación parece indicar que: (1) se ha explorado el conjunto de datos con tal diversidad que no se podría obtener un mayor rendimiento significativo, o (2) es probable que se presente un error en el conjunto de datos. Debido a que los resultados y materiales utilizados son muy similares a los empleados por Huong y otros (2015), la primera opción parece resultar más plausible. El mismo comportamiento se observa en los modelos de multiclasificación.

3.3.2 Resultados del segundo conjunto de datos.

A partir del segundo conjunto de datos:

- Se obtuvieron 66 subconjuntos de características heterogéneos.
- Se consiguieron 987 modelos tras la aplicación de las técnicas de *Machine Learning* en todos los subconjuntos obtenidos.
- Se seleccionaron los 9 mejores modelos, construidos a partir de 8 subconjuntos y 6 técnicas diferentes.
- Estos mejores modelos individuales fueron combinados para obtener 123 modelos de multiclasificación.
- Se seleccionaron los dos mejores modelos de este tipo, construidos a partir de 4 modelos y 2 técnicas distintas.
- El *accuracy* de los mejores modelos fue mayor al de sus similares obtenidos con el conjunto original de variables (66).

La tabla 5 resume los 5 mejores modelos individuales y los 2 mejores de multiclasificación para el segundo conjunto de datos (modelos de potencia). La tabla 5 es análoga a la tabla 4, pero en esta ocasión para los modelos de potencia.

Mejores modelos individuales			
Modelo	Técnica de Machine Learning	Subconjunto (# características)	Accuracy (%)
1	Bagging (RF)	19	88.98
2	Bagging (RF)	7	88.19
3	Stacking (ADL – NB,SVM,C.45)	11	88.19
4	Voto por promedio (RB,MLP,C4.5)	16	88.19
5	MLP	15	87.40
Mejores modelos de multiclasificación			
Modelo	Técnica de Machine Learning	Subconjunto (modelos)	Accuracy (%)
A	Red bayesiana	2, 3, 5	89.76
B	ADL	2, 4	88.98

Tabla 5: Mejores modelos de potencia.

La tabla 5 demuestra una buena eficiencia de los modelos individuales. Ellos logran clasificar correctamente más del 87% de casos externos usando una pequeña cantidad de características iniciales. Por ejemplo, el modelo 2 obtuvo un *accuracy* de 88.19% en el conjunto de validación empleando solo 7 de las 66 características originales.

La combinación de estos modelos eficientes permitió obtener modelos de multiclasificación con mayor desempeño que los individuales. El mejor modelo (*accuracy* de 89.76% en el conjunto de validación) fue obtenido a partir de las salidas de tres modelos individuales y entrenado con la red bayesiana. Los tres modelos se obtuvieron con la aplicación de tres técnicas de *Machine Learning* (*Bagging*, *Stacking* y la red MLP) sobre tres subconjuntos diferentes (de 7, 11, y 15 variables).

A partir de la tabla 5, se observa que el *accuracy* de los modelos individuales es bastante similar, siendo el mismo en tres de ellos (modelos 2, 3 y 4). Al igual que en los modelos de activación, podría deberse a: (1) una amplia exploración del conjunto de datos, o (2) la presencia de un error en el conjunto de datos. Como los resultados y materiales utilizados son similares a los empleados por Huong y otros (2015), la primera opción parece resultar más plausible. Los modelos de multiclasificación también presentaron un comportamiento similar al de los individuales.

3.3.3 Comparación de resultados con otros trabajos.

Los mejores resultados obtenidos en este trabajo fueron comparados con los estudios descritos en el capítulo 2. La tabla 6 presenta una comparación visual de los valores de *accuracy* que regularmente se obtienen en esta clase de investigaciones.

La primera columna enuncia a los autores de los estudios. La segunda columna señala el objetivo de las investigaciones. La tercera columna indica el tamaño del conjunto de entrenamiento y el porcentaje de casos positivos que poseían. La cuarta columna muestra el *accuracy* en el conjunto de validación. La quinta y sexta columna son similares a la tercera y cuarta, pero para los modelos de potencia.

	Objetivo	**Tamaño del conjunto de entrenamiento clasificación (% activos)**	**Mayor *accuracy* (%) (validación) en clasificación**	**Tamaño del conjunto de entrenamiento potencia (% potentes)**	**Mayor *accuracy* (%) (validación) en potencia**
Louvina, 2010	Anti-leishmanisidas	1357(26.97)	76.4	N/A	N/A
Poorinmohammad et al, 2014	Péptidos anti-VIH	1051 (51.57) (total de casos)	96.76	N/A	N/A
Malik et al, 2016	Niveles de glucosa en la sangre en ayunas	175 (total de casos)	85.00	N/A	N/A
Navarro et al, 2017	Detección de peatones	1931 (N/A)	96.20	N/A	N/A
Xie et al, 2017	Identificación de melanoma	240 (33.33) (total de casos)	94.17	N/A	N/A
Addo et al, 2018	Probabilidad de incumplimiento del préstamo	117019 (1.48)	99.30	N/A	N/A
Subudhi et al, 2019	ACV isquémico	N/A	94.30	N/A	N/A
Huong et al, 2015	Inhibidores de la tirosinasa	1072 (49.07)	95.52	398 (64.57)	88.89
Este trabajo	**Inhibidores de la tirosinasa**	**1072 (49.07)**	**97.20**	**388 (65.72)**	**89.76**

Tabla 6: Comparación de resultados con otros trabajos.

La ausencia de datos en las últimas dos columnas de la mayoría de los estudios justamente resalta la labor de aquellas investigaciones (la presente y la de Huong y compañía) que buscaron una discriminación más aguda del objetivo (en ambas la identificación de inhibidores potentes de la tirosinasa). Cabe destacar que, a excepción del estudio de Huong y otros, las comparaciones no son directas debido a que no compartieron el mismo objetivo, ni se usaron los mismos datos y técnicas para la modelación.

De la tabla 6 se puede observar que esta investigación obtuvo el segundo mayor *accuracy* en conjunto de validación, y el primero si solo consideramos los estudios en el área de medicina. El mejor resultado (Addo, Guegan y Hassani, 2018) cuenta con una cantidad de casos totalmente mayor a los otros estudios (117019), aunque se advierte que es un conjunto de datos completamente desbalanceado (los casos activos representan el 1.48% del total de ellos), por lo que existe la posibilidad de que presente un alto sesgo.

Por otro lado, este trabajo y el de Huong y otros comparten la misma meta: obtener modelos de clasificación de activación y potencia de inhibidores de la tirosinasa. También se usó la misma base de datos de activación y una prácticamente igual de potencia en esta ocasión. Ambos estudios también resuelven usar sistemas multiclasificadores para obtener modelos más eficientes. Por lo tanto, se puede destacar que este trabajo obtuvo un *accuracy* ligeramente mayor, tanto para activación (1,68% mayor) como para potencia (0.87% mayor). Esta diferencia es apreciable tomando en consideración que el porcentaje de error es bajo en ambos estudios.

Sin embargo, se reconoce que a los modelos obtenidos en esta investigación les resta ser evaluados con más mediciones y pruebas estadísticas para tener plena confiabilidad de sus capacidades de predicción. Así mismo, una vez corroborada su eficiencia podrán ser usados para la comparación con otros estudios publicados en revistas científicas.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

Al finalizar la presente investigación se llegaron a las siguientes conclusiones:

1. Se logró implementar nuevos modelos computacionales eficientes, que combinen técnicas de *Feature Selection* y *Machine Learning*, para identificar inhibidores potentes de la tirosinasa en bases de datos de moléculas químicas.
2. El uso de un multiclasificador permitió obtener modelos que superaron en *accuracy* a los mejores modelos individuales.
3. Los mejores modelos obtenidos presentan un mayor *accuracy*, en el conjunto de validación, en comparación a trabajos similares.
4. Esta comparación no es exacta debido a que los compuestos buscados fueron diferentes, no se utilizaron los mismos datos de entrenamiento y validación y se aplicó una mayor diversidad de técnicas en esta investigación.

4.2 Recomendaciones

Para futuros trabajos se recomienda:

1. Validar los mejores modelos usando otras medidas utilizadas en estudios similares. En investigaciones para el descubrimiento de medicamentos se suelen utilizar *accuracy*, sensibilidad, especificidad, el coeficiente de correlación de Mathews y el área bajo la curva ROC (siglas en inglés de característica operativa del receptor).
2. Explorar otras herramientas computacionales que permitan generar nuevos modelos identificadores de otros tipos de fármacos. En este trabajo solo se utilizó Weka, que es un programa bastante sólido aunque ahora deprecado. En la actualidad, estas investigaciones son realizadas con otras herramientas como MATLAB (MathWorks, 2019) y bibliotecas, con muy buenas implementaciones de *Machine Learning*, para el

lenguaje de programación Python. La aplicación de diferentes herramientas permitirá una mayor diversidad y mejor comparación de resultados.

3. Planificar los recursos computacionales para la búsqueda exhaustiva de posibles hipótesis. Aunque se consiguió la generación de más de 2000 modelos, se podría haber explorado más a través del uso de estaciones de trabajo acorde a las exigencias de esta investigación.
4. Una vez corroborada su eficiencia, la utilización jerárquica de los dos grupos de modelos obtenidos permitirá una descripción más completa de la actividad inhibitoria de la tirosinasa. Un modelo de activación actuará primero para descartar aquellos compuestos que no tienen capacidad alguna de inhibir la enzima. Un modelo de potencia medirá la fuerza de inhibición de los compuestos que pasen el primer filtro.

REFERENCIAS BIBLIOGRÁFICAS

Abidin, T. & Perrizo, W. (2006). SMART-TV: A Fast and Scalable Nearest Neighbor Based Classifier for Data Mining. *Proceedings of the ACM SAC-06 symposium on Applied computing, Dijon, France* (pp. 536-540), New York: ACM Press. doi: 10.1145/1141277.1141403

Addo, P., Guegan, D. & Hassani, B. (2018). Credit risk analysis using machine and deep learning models. *Risks, 6*(2), 38-56.

Ambielli, B. (2017). Information entropy and information gain. *Bambielli's blog*. Obtenido el 27 de noviembre de 2018 de https://bambielli.com/til/2017-10-22-information-gain/

American Cancer Society (2018). *Cancer facts & figures 2018*. Atlanta, pp. 4,23-24. Obtenido el 01 de abril de 2019 de https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2018/cancer-facts-and-figures-2018.pdf

Ariew, R. (1976). Maquinilla de afeitar de Ockham: Un análisis histórico y filosófico del principio de parsimonia de Ockham, Champaign-Urbana: Universidad de Illinois.

Baldi, P., Brunak, S., Chauvin, Y., Andersen, C.A. & Nielsen, H. (2000). Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics, 16*(5), 412-424.

Breiman, L. (2001). Random Forests. *Machine Learning. 45*(1), 5-32.

Brenner, M. & Hearing, V. J. (2008). The protective role of melanin against UV damage in human skin. *Photochemistry and photobiology, 84*(3), 539–549.

Chen, Q. & Kubo, I. (2002). Kinetics of Mushroom Tyrosinase Inhibition by Quercetin. *Journal of Agriculture and Food Chemistry, 50*(14), 4108-4112.

Christianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines*, New York: Cambridge University Press.

Da Silva, S. (25/04/2011). *Seleção de características por meio de algoritmos genéticos para aprimoramento de rankings e de modelos de classificação* (tesis de doctorado). Universidade de São Paulo, São Paulo, Brasil. Obtenido el 10 de noviembre de 2019 de http://www.teses.usp.br/teses/disponiveis/55/55134/tde-19072011-151501/pt-br.php

Dalia, N., De Stefano, C., Fontanella, F. & Scotto di Freca, A. (2019). A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognition Letters, 121*, 77-86.

Das, S. (2001). Filters, wrappers and a boosting-based hybrid for feature selection. *ICML*, Citeseer, 74-81.

Dietterich T.G. (2000) Ensemble Methods in Machine Learning. Proceedings Multiple Classifier Systems 2000. Lecture Notes in Computer Science, vol 1857. Springer-Verlag Berlin: Berlin. doi: 10.1007/3-540-45014-9_1

Donges, N. (17/04/2018). Pros and Cons of Neural Networks. *Towards Data Science*. Obtenido el de 30 de octubre de 2018 de https://towardsdatascience.com/hype-disadvantages-of-neural-networks-6af04904ba5b

Fialho, A. et al. (2010). Predicting outcomes of septic shock patients using feature selection based on soft computing techniques. *Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications. 81*, E. Hüllermeier, E. & otros (Eds.): Springer Berlin Heidelberg, 65-74.

Freund, Y. & Schapire, R.E. (1996). Experiments with a new boosting algorithm. *Thirteenth International Conference on Machine Learning, San Francisco*, 148-156.

Friedman, N., Geiger, D. & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine Learning, 29,* 131-163. doi: 10.1023/A:1007465528199

González-Diaz, H., Uriarte, E. & Ramos de Armas, R. (2005). Predicting stability of Arc repressor mutants with protein stochastic moments. *Bioorganic & Medical Chemistry*. *13*(2), 323-331.

Goodfellow, I. et al. (2016). *Deep Learning*. Cambridge, Estados Unidos: The MIT Press. Obtenido el 30 de octubre de 2019 de http://www.deeplearningbook.org/

Guyon, I., Elisseeff, A. (03/2003). An Introduction to Variable and Feature Selection. *Jorunal of Machine Learning Research*. Vol. 3 (1), p. 1157-1182. Obtenido el 26 de octubre de 2018 de http://www.jmlr.org/papers/v3/guyon03a.html

Hall, M.A. (1998). *Correlation-based feature selection for machine learning* (tesis de doctorado). The University of Waikato, Hamilton, Nueva Zelanda.

Han, J. & Morag, C. (1995). The influence of the sigmoid function parameters on the speed of backpropagation learning. In Mira, José; Sandoval, Francisco (eds.). *From Natural to Artificial Neural Computation*. Lecture Notes in Computer Science. 930. p. 195–201.

Hann, M. & Green, R. (1999). Chemoinformatics--a new name for an old problem? *Current Opinion in Chemcal Biology*. *3*(4), 379-383. doi: 10.1016/S1367-5931(99)80057-X

Holte, R.C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning. 11*(1), 63-91.

Huong, L. et al. (2015). Multi-Criteria decision making: the best choice for the modeling of chemicals against hyper-pigmenatation? *Current Bionformatics. 10*(5), 520-532.

Kohavi, R. (1995). The Power of Decision Tables. *Proceedings of the 8th European Conference on Machine Learning*, 174-189. doi: 10.1007/3-540-59286-5_57

Kuncheva, L.I. (2014). *Combining Pattern Classifiers: methods and algorithms* 2º ed., Hoboken, New Jersey: Wiley Interscience.

Kuncheva, L.I., Skurichina, M. & Duin, R.P.W. (2002). An experimental study on diversity for bagging and boosting with linear classifiers. *Information Fusion, 3*(2), 245-258. Obtenido el 13 de abril de 2019 de http://pages.bangor.ac.uk/~mas00a/papers/lkmsrdif02.pdf

Kuncheva, L.I. & Whitaker, C.J. (2001). Ten measures of diversity in classifier ensembles: limits for two classifiers. *IEEE Workshop on Intelligent Sensor Processing, 10*, 1-6.

Ladha, L. & Deepa, T. (2011). Feature selection methods and algorithms. *International Journal on Computer Science and Engineering. 3*(5), p. 1787-1797.

Louvina, J. (2010). *Identificación de nuevos compuestos líderes con actividad antileishmaniásica a través de estudios in silico* (tesis de licenciatura). Universidad Central "Martha Abreu" de Las Villas, Santa Clara, Cuba.

Malik, S. et al. (2016). Non-invasive detection of fasting blood glucose level via electrochemical measurement of saliva. *Springerplus. 5*(1), 701.

MathWorks. (2019). *MATLAB* (Versión R2019a) [software].

Marrero-Ponce, Y. & Romero, V. (2002). *TOMOCOMD (TOpological MOlecular COMputer Design) para Windows* (Versión 1.0) [software]. Universidad Central de Las Villas. La versión 1.0 es una versión experimental preliminar, una futura versión profesional se puede solicitar a Y. Marrero: yovanimp@uclv.edu.cu o ymarrero77@yahoo.es

Melián, B., Moreno-Pérez, J. & Moreno-Vega, J. (2009). Introducción a la Computación Evolutiva. *Números, 71*, 21-27.

Navarro, P., Fernández, C., Borraz, R. & Alonso, D. (2017). A machine learning approach to pedestrian detection for autonomous vehicles using high-definition 3D range data. *Sensors*, *17*(1), 18.

Nishioka, K. (1978). Particulate tyrosinase of human malignant melanoma. *European Journal of Biochemistry, 85*(1), 137-146.

Poorinmohammad, N. et al. (2015). Computational prediction of anti-HIV-1 peptides and in vitro evaluation of anti-HIV-1 activity of HIV-1 P24-derived peptides. *Journal of peptide science. 21*(1), 10-16.

Qin, Z. et al. (2015). Deep learning based featured selection for remote sensing scene classification. *IEEE Geoscience and remote sensing letters. 12* (11), 2321-2325.

Quevedo, F. (2011). La prueba de ji-cuadrado. *Medwave. 11*(12), 1-5. doi: 10.5867/medwave.2011.12

Quinlan, R. (1993). *C4.5: Programs for Machine Learning*, San Mateo, CA: Morgan Kaufmann Publishers.

Raschka, S. (03/08/2014). Linear Discriminant Analysis. *Sebastian Raschka*. Obtenido el 20 de febrero de 2019 de https://sebastianraschka.com/Articles/2014_python_lda.html

Riley, P. A. (1997) Melanin. The international journal of biochemistry & cell biology. 29(11), 1235-1239.

Robnik-Sikonja, M. & Kononenko, I. (1997). An adaptation of Relief for attribute estimation in regression. *Proceedings of the Fourteenth International Conference on Machine Learning*, 296-304.

Rodríguez-Abed, A. (2007). *Nuevo sistema multiclasificador jerárquico. Posibilidades de aplicación* (Tesis de maestría). Universidad Central "Martha Abreu" de Las Villas, Santa Clara, Cuba.

Roy, D. et al. (01/10/2015). Feature selection using deep neural networks. *IEEE Xplore*. Obtenido el 29 de octubre de 2018 de https://ieeexplore.ieee.org/document/7280626

Ruck, D., Rogers, S. & Kabrisky, M. (1990). Feature selection using a multilayer perceptron. *Journal of Neural Network Computing. 2*(2), 40-48.

Salah, S. et al. (01/2003). Selección de características usando algoritmos genéticos para clasificación de vinos chilenos. *ResearchGate*. Obtenido el 26 de octubre de 2018 de https://www.researchgate.net/publication/255611559_Seleccion_de_Caracteristicas_usando_Algoritmos_Geneticos_para_Clasificacion_de_Vinos_Chilenos

Scior T., Bernard P., Medina-Franco J.L. & Maggiora G.M. (2007). Large Compound Databases for Structure-Activity Relationships Studies in Drug Discovery. *Mini-Reviews in Medicinal Chemistry. 7*(8):851-60. doi: 10.1016/S1367-5931(99)80057-X

Seo, S.Y., Sharma, V.K. & Sharma, N. (2003). Mushroom tyrosinase: recent prospects. *Journal of Agriculture and Food Chemistry. 51*(10), 2837-2853.

Shipp, C.A. & Kuncheva, L.I. (2002). An investigation into how AdaBoost affects classifier diversity. *Proceedings IPMU 2002, Annecy, France,* 203-208. Obtenido el 13 de abril de 2019 de http://pages.bangor.ac.uk/~mas00a/papers/cslkIPMU02.pdf

Subudhi, A., Dash, M. & Sabut, S. (2019). *Biocybernetics and biomedical engineering. 39*(2).

Takemura, A., Shimizu, A. & Hamamoto, K. (2010). Discrimination of breast tumors in ultrasonic images using an ensemble classifier based on the AdaBoost algorithm with feature selection. *IEEE Transactions on medical imaging. 29*(3), 598-609.

Varshavsky R. et al. (2006). Novel unsupervised feature filtering of biological data. *Bioinformatics. 22*(14), p. e507-e513.

Venkatesh, S. & Lipper, R.A. (2000). Role of the development scientist in compound lead selection and optimization. *J Pharm Sci. 89*(2), 145-154.

Witten, I.H., Frank, E., Hall, M.A. & Pal, J.P. (2016). *The WEKA Workbench* (Versión 3.8.3) [software]. Apéndice en línea para "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, 4º Ed.

Wolpert, D.H. (1992). Stacked generalization. *Neural Networks. 5*(2), 241-259.

Xie, F., Fan, H., Li, Y., Jiang, Z., Meng, R., & Bovik, A.C. (2017). Melanoma Classification on Dermoscopy Images Using a Neural Network Ensemble Model. *IEEE Transactions on Medical Imaging. 36*, 849-858.

Ying, L. (13/04/2004). A comparative study on feature selection methods for drug discovery. *Journal of Chemical Information and Computer Sciences. 44* (5), p. 1823-1828.

Yu, J. et al. (2005). Ovarian cancer identification based on dimensionality reduction for high-throughput mass spectrometry data. *Bioinformatics. 21*(10), p. 2200-2209

Zhi-Zhong, L. & al. (30/03/2018). ECoFFeS: A software using evolutionary computation for feature selection in drug discovery. *IEEE Access.* Obtenido el 03 de noviembre de 2018 de https://ieeexplore.ieee.org/document/8328818