UNIVERSIDAD DE ANTIOQUIA

PREDICCIÓN DE LOS MECANISMOS DE ACCIÓN (MoA) ASOCIADOS AL USO DE FÁRMACOS A TRAVÉS DE MODELOS DE MACHINE LEARNING

Autor(es)

Marcela Andrea Castrillón Buitrago

Yeison Fernando Villamil Franco

Universidad de Antioquia

Facultad de Ingeniería

Departamento de Ingeniería de Sistemas

Medellín, Colombia

2021

CC BY NC SA

Predicción de los Mecanismos de Acción (MoA) Asociados al Uso de Fármacos a través de Modelos de Machine Learning

Marcela Andrea Castrillón Buitrago
Yeison Fernando Villamil Franco

Tesis o trabajo de investigación presentada(o) como requisito parcial para optar al título de:
Especialista en Analítica y Ciencia de Datos

Asesores (a):
Julián David Arias Londoño

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería de Sistemas
Medellín, Colombia
2021.

CONTENIDO

LISTA DE FIGURAS 4

LISTA DE TABLAS 5

1. RESUMEN EJECUTIVO 6

2. DESCRIPCIÓN DEL PROBLEMA 7

2.1. Problema del Negocio 9

2.2. Aproximación desde la Analítica de Datos 10

2.3. Origen de los Datos 11

2.4. Métricas de desempeño 12

2.4.1. Etapa 1: One Vs Rest 13

2.4.2. Etapa 2 y 3: Clasificación multi-etiqueta 15

3. DATOS 16

3.1. Datasets 16

3.2. Descriptiva 17

4. PROCESO DE ANALÍTICA 21

4.1. Pipeline principal 21

4.1.1. Etapa 1: One Vs Rest 22

4.1.2. Etapa 2: Modelos de clasificación multi-etiqueta 22

4.1.3. Etapa 3: Modelos basados en Deep learning 23

4.2. Preprocesamiento 23

4.3. Modelos 23

4.3.1. Etapa 1: One Vs Rest 23

4.3.2. Etapa 2: Modelos de clasificación multi-etiqueta 25

4.3.3. Etapa 3: Modelos basados en Deep learning 27

5. METODOLOGÍA 29

5.1. Baseline 29
5.2. Validación 30
5.3. Iteraciones y Evolución 30
5.4. Herramientas 32
6. RESULTADOS 33
6.1. Métricas 33
6.1.1. Etapa 1: One Vs Rest 33
6.1.2. Etapa 2: Modelos de clasificación multi-etiqueta 36
6.1.3. Etapa 3: Modelos basados en Deep learning 36
6.2. Evaluación Cualitativa 37
6.3. Consideraciones de producción 37
7. CONCLUSIONES 38
8. BIBLIOGRAFIA 39

LISTA DE FIGURAS

Figura 1. Pruebas de laboratorio para la evaluación de los MoA. Tomado y modificado de [3]. .11
Figura 2. Respuestas de las pruebas de laboratorio en células. 11
Figura 3. Matriz de confusión para Etapa 1: *One Vs Rest*.....14
Figura 4. Distribuciones de las variables continuas. En verde se muestra la expresión génica y en rojo la viabilidad celular. 17
Figura 5. Variables que representan el tipo de perturbación, dosis y tiempo de dosis respectivamente para el conjunto de entrenamiento. 17
Figura 6. Variables de entrada del conjunto de entrenamiento. 19
Figura 7. Conjunto de proteínas con mayor número de activaciones. 20
Figura 8. Pipeline principal para el entrenamiento y predicción. 21
Figura 9. Arquitectura de red neuronal usando la librería de *tensorflow*. 26
Figura 10. Arquitectura de red neuronal usando la librería de *scikit-learn*. 26
Figura 11. Arquitectura de red neuronal para el primer proceso de la etapa 3. 27
Figura 12. Arquitectura de redes neuronales para la predicción de los MoA usando el codificador de la Figura 11. 28
Figura 13. Arquitectura de red neuronal usada para el *denoising autoencoder*. 28
Figura 14. Arquitectura de redes neuronales para la predicción de los MoA usando el codificador de la Figura 13. 29
Figura 15. Matriz global para la máquina de soporte. 35

LISTA DE TABLAS

Tabla 1. Resultados obtenidos durante la etapa 1. 33
Tabla 2. Resultados obtenidos durante la etapa 2. 36
Tabla 3. Resultados obtenidos durante la etapa 3. 36

1. RESUMEN EJECUTIVO

En el pasado los científicos obtenían los fármacos a partir de productos naturales o se basaban en remedios tradicionales, muchos de estos eran obtenidos de manera empírica mediante la observación de los efectos externos. En la actualidad y con el avance de la tecnología, el descubrimiento de fármacos ha cambiado de los enfoques empíricos del pasado a un modelo más específico basado en la comprensión del mecanismo biológicos subyacente de una enfermedad, evaluados principalmente en las células. El entendimiento de estos mecanismos es vital ya que permite conocer el efecto de los fármacos sobre las células con el fin de evitar la toxicidad en estas. Una de las formas para la evaluación de estos mecanismos es a través de ensayos de viabilidad celular que permiten, mediante pruebas de laboratorio, la recombinación de diferentes células con un único ADN (ácido desoxirribonucleico). Cada célula es tratada con diferentes tipos de fármacos y mediante espectrometría de masas es posible identificar las distribuciones de las respuestas de estas, donde se encuentran los mecanismos de acción. Esta evaluación crea unos picos de intensidad y permite identificar diferentes tipos de proteínas que son activadas con el uso de estos fármacos. Esta activación de proteínas es un mecanismo de respuesta celular conocido por sus siglas en inglés como *mechanisms of action* (MoA). Este MoA es básicamente la manera como un fármaco afecta una célula, haciendo que esta active una proteína (este tipo de activaciones pueden ser proteínas, molécula de RNA, etc). Una de las mayores dificultades que existe en la evaluación de fármacos, es que se necesitan muchas pruebas de laboratorio que permitan identificar las diferentes proteínas que están siendo activadas. Por esto, los modelos predictivos pueden ser una solución para usar bancos de células ya evaluados y poder relacionar sus respuestas con células a las cuales aún no se les han estudiado sus MoA.

Partiendo de las definiciones y planteamientos anteriormente mencionados, este estudio tiene como objetivo la experimentación de modelos predictivos de clasificación que permita a partir del uso de información génica y de viabilidad celular encontrar el conjunto de proteínas que están siendo activadas al uso de diferentes fármacos. Los datos disponibles constan de un conjunto de entrenamiento con información de 772 expresiones génicas y 100 variables de viabilidad celular, siendo las respuestas al uso de 23814 fármacos únicos. Es importante destacar que no se reciben información de las diferentes patologías estudiadas. Las variables de respuestas constan de un

archivo con 206 diferentes proteínas que fueron activadas al uso de estos fármacos. Es posible que existan varias activaciones por fármaco, ya que como se mencionó anteriormente, hay proteínas que tienen funcionalidades similares. Estas variables objetivo consisten en una asignación binaria de 0 para el caso de no activación y 1 el caso de activación.

Este estudio consiste en un problema de clasificación multi-etiqueta, sin embargo, de las 206 proteínas activadas, solo son tomadas aquellas que tienen más de 100 activaciones, esto quiere decir, aquellas que fueron activadas por más de 100 fármacos, dando como resultado un total de 41 proteínas. El estudio consta de tres etapas; la primera etapa consiste en el entrenamiento de modelos predictivos para cada una de las 41 proteínas seleccionadas en un esquema *One vs Rest*, los modelos usados en esta etapa son: regresión logística, *random forest*, máquinas de soportes (*support vector machine classifier* - SVC) y modelo de aumento de gradiente (XGBoost). La segunda etapa consiste en usar modelos multi-etiqueta, siendo estos: *random forest* y redes neuronales. Por último, la tercera etapa consiste en usar estructuras de redes neurales (*Deep learning*) con *autoencoders* para todo el problema multi-etiqueta. La métrica base para la evaluación de las tres etapas es la pérdida logarítmica (*cross-entropy*) que permite calcular la probabilidad de activación de las proteínas. Para este estudio, el mejor resultado obtenido fue para el modelo máquina de soporte (SVC), con un valor de pérdida de 0.0389.

En el siguiente repositorio se encuentran los notebooks que fueron usados para la ejecución del proyecto: https://github.com/yeivillamil/Proyecto_MecanismosdeAccion

2. DESCRIPCIÓN DEL PROBLEMA

Existe en la actualidad una gran necesidad de poder crear fármacos innovadores que combatan muchas de las enfermedades que no han sido tratadas con éxito. Sin embargo, el desarrollo de estos es difícil, costoso y en ocasiones ineficiente. En la última década, el descubrimiento de fármacos fenotípicos es una estrategia que ha sido introducida como primera clase, estos básicamente consisten en identificar el objetivo de moléculas farmacológicamente activas. Una de las dificultades de este tipo de estrategias es que se cree que las inestabilidades genéticas entre pacientes individuales es una de las principales causas de heterogeneidades de las respuestas de los fármacos [1].

Dos estrategias distinguibles para el estudio de fármacos mediante viabilidad celular son mecanismo-primero (*mechanisms-first*) y componente primero (*compound-first*), denominadas así por el descubrimiento tipo objetivo y tipo fenotípico respectivamente. La primera se centra en un objetivo de fármaco, siendo este un producto génico que proporciona el punto de partida de la invención de un agente terapéutico que modula una expresión, función o actividad. La segunda se basa solo en la respuesta individual del fenotipo sin relacionar la interacción génica. Esta primera estrategia basada en objetivo es comúnmente conocida como identificación MoA [2]. En medicina el término MoA es usado para definir como un fármaco o alguna otra sustancia produce un efecto en el cuerpo. Un ejemplo claro es identificar los MoA de un fármaco que son afectados en una célula, ya sea una enzima, función celular, etc. El conocimiento de los MoA ayuda a proporcionar información sobre la seguridad de los fármacos usados y cómo estos afectan el cuerpo [3]. El término objetivo se utiliza con mayor frecuencia en la literatura científica para describir el objetivo molecular específico (proteína, molécula de ARN, etc.) con la que interactúa un fármaco para iniciar una respuesta biológica. MoA se usa a menudo como sinónimo de objetivo, aunque algunos investigadores reservan este término para describir la acción del fármaco en un nivel superior de complejidad biológica, refiriéndose a un sistema o a procesos de señalización celular que se ven afectados por el fármaco a través de su interacción con una molécula específica. Los ejemplos más simples de objetivos moleculares incluyen inhibidores de enzimas, agonistas o antagonistas de los receptores de la superficie celular y bloqueadores de los transportadores de la membrana plasmática.

La actividad celular relacionada con el uso de fármacos está influenciada por múltiples factores, incluyendo afinidad y selectividad de componentes y la penetración de la participación de los fenotipos celulares [4]. Existen varios métodos usados para la investigación de los mecanismos de acción: biofísicos, computacionales y celulares. Los enfoques basados en células para investigar los mecanismos de acción incluyen un perfilamiento transcripcional después del tratamiento farmacológico de la célula, enfoque proteómico químico como *kinobeads* y ensayo de desplazamiento térmico celular para medir la interacción fármaco-proteína e imágenes multiplexadas o citometría de flujo para medir parámetros celulares tras el tratamiento farmacológico. A pesar de la utilidad de estos diferentes enfoques, obtener una imagen completa del mecanismo de acción, particularmente en células, sigue siendo un desafío.

La identificación del mecanismo y la validación de un objetivo son importantes y desafiantes. Hay muchos enfoques diferentes e informes recientes para describir estos enfoques. Lo que está claro es que ningún enfoque funciona para todos los medicamentos [5]. También está claro que algunos medicamentos están aprobados con mecanismos de acción desconocidos y la identificación del objetivo no es un criterio de aprobación. Además, los mecanismos moleculares de muchos medicamentos no se identifican hasta mucho después de su aprobación. Un programa de la efectividad de los fármacos avanza de manera más rápida cuando se conocen los mecanismos de acción.

Basados en la estrategia de objetivo para la identificación de los mecanismos de acción, son creados conjuntos de datos de células con un ADN único obtenidas en laboratorio. Mediante espectrometría de masas se realizan las mediciones de las diferentes proteínas, RNA u otra sustancia que se activa al uso de fármacos y esta información es almacenada en conjuntos de variables conocidos como viabilidad celular. Estas viabilidades celulares permiten conocer la distribución de moléculas de una sustancia, permitiendo identificar moléculas únicas o combinadas que son activadas por los fármacos empleados. Debido a que una de las dificultades existentes es poder predecir estas sustancias activas para células que poseen una funcionalidad similar, se crean conjuntos de datos que relacionan estas viabilidades celulares con comportamiento similares encontrados en estudios génicos, para poder usar estrategias computacionales que permitan encontrar los conjuntos de objetivos que están siendo alterados al uso de fármacos.

2.1. Problema del Negocio

El *ConnectivityMap* es un proyecto del Instituto Broad del MIT y Harvard, el Laboratorio de Ciencias de la Innovación de Harvard (LISH) y la biblioteca de fondos comunes de firmas celulares integradas basadas en red (LINCS). Ellos presentaron el desafío en *Kaggle*: Predicción de los mecanismos de acción (MoA), en el siguiente enlace: (https://www.kaggle.com/c/lish-moa) con el objetivo de avanzar en el desarrollo de fármacos mediante mejoras en los algoritmos de predicción del MoA.

En el LINCS existen bases de datos completas de perturbaciones de fármacos a gran escala. Estos datos permiten la realización de algoritmos predictivos que puedan predecir los diferentes efectos de los fármacos en las células mediante la activación objetivo, para este caso proteínas. Existe una gran dificultad a la hora de identificar cada uno de los objetivos activados, ya que se hace necesario

pruebas de laboratorio extensas que pueden ser costosas y pausadas. La creación de algoritmos, o en el caso de este estudio, la experimentación de modelos de clasificación permitiría predecir de manera eficiente la probabilidad de activación de diferentes objetivos para cada fármaco único y esta información poder relacionarla a diferentes tipos de células que tienen funcionalidades similares a las que están siendo usadas en el proceso de entrenamiento.

Este tipo de metodologías de implementación de modelos predictivos ayudaría principalmente al entendimiento rápido de los MoA, permitiendo así hacer más rápido los procesos de evaluación de la efectividad de los fármacos que están siendo usados para diferentes patologías. Adicionalmente reduciría la cantidad y los tiempos en pruebas de laboratorio.

2.2. Aproximación desde la Analítica de Datos

Debido a la dificultad que existe en poder predecir los mecanismos de acción de las células, cuando estas son alteradas por el uso de fármacos, se comienza un reto de experimentar modelos de *machine learning* y *deep learning* que permitan encontrar la probabilidad de activación de un objetivo (proteína, RNA, etc) dentro de la célula. Uno de los retos en este MoA a nivel computacional usando algoritmos matemáticos, es poder encontrar modelos que permitan predecir dicho objetivo a conjuntos de células que poseen distribuciones de espectrometría y funcionalidades similares. Esto permitiría relacionar estas células con aquellas que no han sido evaluadas y reduciría tiempos en el desarrollo de nuevos tipos de fármacos.

Así, el actual ejercicio consiste en un problema multi-etiqueta, donde se presentan no solo una respuesta, sino múltiples respuestas o etiquetas para un mismo conjunto de datos de entrada. Este tipo de datos puede ser abordado con una estrategia uno vs el resto (*OneVsRest*) que será la primera etapa de este trabajo, donde son entrenados cuatro diferentes tipos algoritmos de clasificación para cada una de las proteínas, permitiendo realizar predicciones y matrices de confusión individuales. En las etapas dos y tres será abordado el problema en conjunto como multi-etiqueta y usando de algoritmos de clasificación que soportan entrenar y predecir este tipo de problemas. En la etapa dos serán usados dos tipos de modelos y en la etapa tres serán usados modelos sofisticados de *Deep learning*.

2.3. Origen de los Datos

En la práctica, la forma de predecir los mecanismos de acción es colocando células en platos de ensayo donde cada una de estas contiene un ADN único y que permite identificar su alteración al momento de usar fármacos específicos, como muestra la Figura 1.

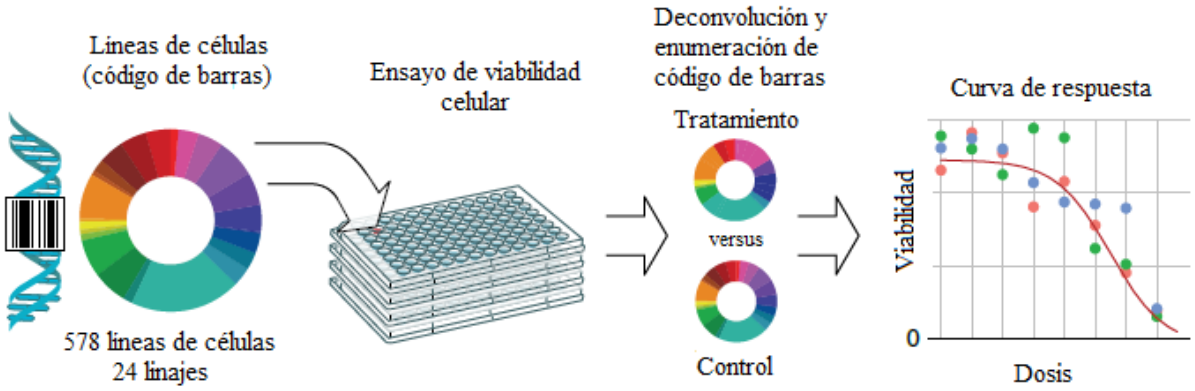

Figura 1. Pruebas de laboratorio para la evaluación de los MoA. Tomado y modificado de [3].

Este tipo de pruebas de la Figura 1 se realizan para poder medir la distribución del efecto que se está teniendo en cada célula al uso de fármacos. Es importante mencionar que siempre son probadas diferentes dosis, ya que esto permite identificar el nivel de efectividad y toxicidad de los fármacos dentro de la célula. La distribución obtenida viene acompañada de la proteína que fue activada en cada una de las pruebas y en cada experimento es posible que se activen diferentes tipos de proteínas. Las distribuciones miden la intensidad de respuesta como muestra la Figura 2.

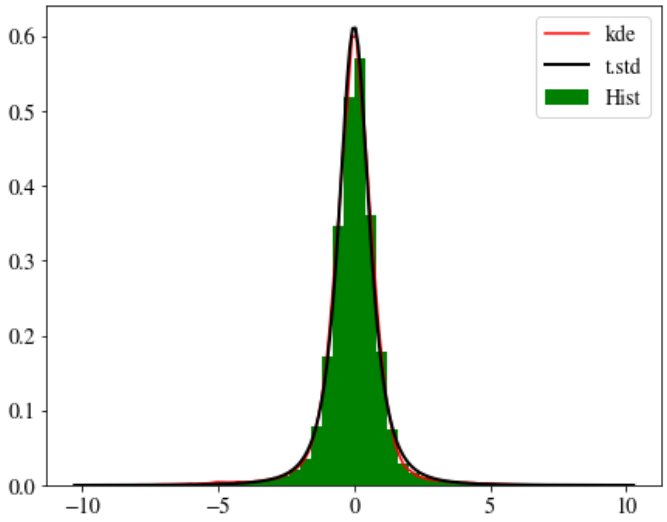

Figura 2. Respuestas de las pruebas de laboratorio en células.

Cada punto dentro la distribución representa la respuesta a una cierta perturbación, que, para este caso, el fármaco genera esta perturbación y la respuesta sería una proteína activada. Esta es una de las maneras de obtener las respuestas de los MoA de una célula.

Para cada tipo de células probadas en las pruebas de laboratorio, son asociadas expresiones génicas con funcionalidades similares a las células probadas. El conjunto de datos usado en este trabajo puede ser encontrado en la siguiente ruta https://www.kaggle.com/c/lish-moa. Los registros del conjunto de datos están representados por los fármacos probados para cada prueba de viabilidad celular y consta de 100 respuestas de células y 772 expresiones génicas. Los objetivos analizados como mecanismos de acción son 206 proteínas en total y están agrupados en tres clases principalmente. Estas son:

- Receptores: Objetivos que se encuentran en la superficie y que se pueden ser modulados positivamente por agonistas, las cuales se unen a receptores celulares y provocan una acción determinada. El caso contrario es antagonista.
- Enzimas: Se encuentran dentro de la célula. Son moduladas positivamente por activadores y negativamente por inhibidores y bloqueadores.
- Agentes: Son sencillamente moduladores.

Es importante mencionar que, para cada fármaco probado, no se mencionan los tipos de patologías y tanto las viabilidades celulares y expresiones génicas tienen un nombre genérico (c-1 a c-100 y g-1 a g-772 respectivamente) y cada fármaco posee una identificación (id).

2.4. Métricas de desempeño

El ejercicio de *Kaggle* es un problema de clasificación multi-etiqueta el cual consta de 872 variables de entrada siendo estas las viabilidades celulares y expresiones génicas, y como variables de salida las proteínas que están siendo activadas con el uso de fármacos. Cada una de estas proteínas dentro del conjunto de datos, tiene una respuesta positiva o negativa de activación, representadas con 1 o un 0 respectivamente. A partir de esto, la métrica de desempeño propuesta en el ejercicio de *Kaggle* fue la pérdida logarítmica (*cross-entropy*), la cual mide la probabilidad de que una muestra tenga una respuesta positiva para cada MoA. La Ecuación 1 muestra la métrica usada.

$$score = -\frac{1}{M}\sum_{m=1}^{M}\frac{1}{N}\sum_{i=1}^{N}\left[y_{i,m}\log\left(\widehat{y_{i,m}}\right) + \left(1 - y_{i,m}\right)\log\left(1 - \widehat{y_{i,m}}\right)\right] \quad (1)$$

Donde:

- N es el número de observaciones o fármacos (sig_id)
- M es el número de mecanismos de acción (proteínas)
- $\widehat{y_{i,m}}$ a probabilidad predicha de un MoA positivo
- $y_{i,m}$ es el Moa, 1 para respuesta positivo y 0 para respuesta negativa

2.4.1. Etapa 1: One Vs Rest

La primera etapa consistió en la experimentación de cuatro modelos de clasificación: regresión logística, *Random forest*, SVC y aumento de gradiente (*XGBoost*). Cada uno de los modelos fueron entrenados para cada una de las proteínas, abordando el problema de múltiples-etiquetas como de una sola etiqueta y permitiendo el cálculo de matrices de confusión, para evaluar la calidad de cada uno de los cuatros modelos. Las métricas propuestas en esta etapa son:

- *Accuracy*
- *Balanced accuracy*
- *Recall*
- *Precision*
- *F1_score*

Teniendo en cuenta que el problema de clasificación del ejercicio en este estudio consiste en un problema desbalanceado, se ha usado la métrica de exactitud balanceada (*balanced accuracy*) para verificar la tasa de positividad y negatividad de los modelos predictivos. La Figura 3 muestra una matriz de confusión, usada para la verificación de la calidad de los modelos de clasificación y adicionalmente las métricas medidas son presentadas en las ecuaciones 2, 3, 4, 5.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP} \quad (2)$$

$$balanced\_accuracy = \frac{TPR + TNR}{2} \begin{cases} TPR = \frac{TP}{TP + FN} = recall \\ TNR = \frac{TN}{TN + FP} \end{cases} \quad (3)$$

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$F1\_score = \frac{2 * precision * recall}{(precision * recall)} \quad (5)$$

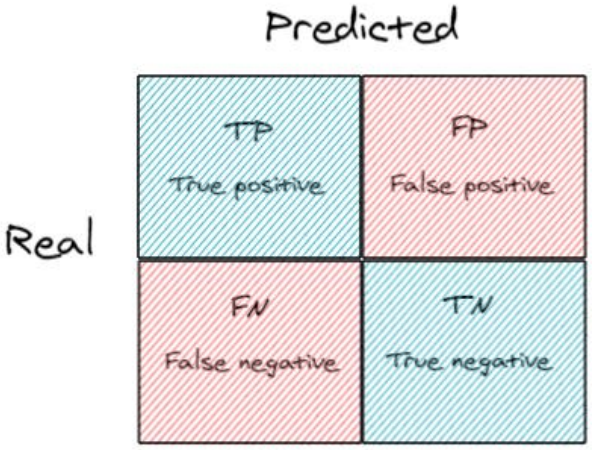

Figura 3. Matriz de confusión para Etapa 1: *One Vs Rest*

Como se ha mencionado, cuando existe un proceso de clasificación para una sola etiqueta es posible obtener y evaluar la calidad del modelo con las métricas anteriormente propuestas. Sin embargo, es importante mencionar que la métrica base propuesta y considerada para cada una de las etapas, es la pérdida logarítmica (Ecuación 1).

Estas cinco métricas de desempeño se calcularon usando las librerías de *scikit-learn* en la sección de métricas. Adicionalmente, es creada una función que permite el cálculo global de cada una de estas métricas de una manera sopesada. El cálculo sopesado fue realizado en función de cuán efectivo fue el modelo en predecir la clase minoritaria, representado las activaciones. Este cálculo fue realizado básicamente tomando en cuenta la relación entre el valor predicho como verdadero positivo (TP) y falso negativo (FN) como se muestra en la ecuación 6.

$$global_{metric} = \frac{1}{M}\sum_{i=1}^{M}\left(\frac{TP_{i,m}}{TP_{i,m} + FN_{i,m}}\right) * g_{metric_{i,m}} \quad (6)$$

Donde:

- TP: Es el verdadero positivo de cada una de las matrices de confusión (para cada proteína)
- FN: Es el verdadero negativo para cada una de las matrices de confusión (para cada proteína)
- g: Es la métrica (*accuracy, balanced-accuracy, recall_score, precision_score, f1_score*) para cada proteína.

Para el caso de la pérdida logarítmica, fue diseñada una función que permite crear una matriz con el valor de esta métrica para cada fila y columna. El valor final es el promedio de todas las filas calculadas para cada una de las proteínas. El procedimiento para el cálculo es el siguiente.

1. Se construye una matriz de ceros con la forma del conjunto de datos de prueba para cada proteína.
2. Son indicados los límites para cada una de las variables, intentando no tener valores indefinidos cuando se aplique el logaritmo.
3. Se calcula la pérdida logarítmica para cada fila y columna por medio de un *for loop* y usando la ecuación 1
4. Retornamos la matriz de pérdida logarítmico

Adicionalmente a las métricas de desempeño y matrices de confusión para cada proteína en esta etapa, es diseñada una función que permite el cálculo de una matriz de confusión global que identifica cuáles de los modelos están prediciendo de manera correcta las activaciones para cada una de las proteínas. Esta matriz será presentada en la sección de resultados. Debido a que la pérdida logarítmica debe ser promediada para todas las proteínas, aquí no fue usada la métrica global sopesada.

2.4.2. Etapa 2 y 3: Clasificación multi-etiqueta

El proceso multi-etiqueta se realizó en dos etapas, la primera consistió en usar modelos que soporten problemas de este tipo, para este caso fueron usados redes neuronales y *random forest*.

Para la segunda etapa se consideraron arquitecturas más sofisticadas como *autoencoder* y los *denoising autoencoders* basados en redes neuronales profundas. Para estas dos etapas, solo se usó la pérdida logarítmica como métrica de evaluación.

3. DATOS

Para el desarrollo de este proyecto fueron utilizados el conjunto de datos del reto Predicción de los mecanismos de acción (*Mechanisms of Action (MoA) Prediction*) de *Kaggle*, son datos de acceso público, y deben ser aceptados los términos y condiciones del reto para proceder a su descarga o su uso. Estos pueden ser encontrados en la siguiente ruta: https://www.kaggle.com/c/lish-moa/data. Los datos se encuentran en formato csv y el tamaño de estos es 205.95 MB.

Estos contienen un archivo de entrenamiento, con sus variables de entrada siendo las expresiones génicas y viabilidad celular, y el conjunto de salida siendo las proteínas por predecir. El conjunto de datos cuenta con un archivo de prueba que no contiene proteínas evaluadas. El reto consiste en encontrar de un modelo de *machine learning* o *deep learning* que permita predecir la probabilidad de activación de las proteínas con una métrica de pérdida logarítmica lo más baja posible.

3.1. Datasets

El conjunto de entrenamiento tiene una dimensión de 23814 filas y 876 columnas; 772 expresiones génicas (g-), 100 características de viabilidad celular (c-). Se tiene una columna *cp_type* que indica muestras tratadas con un compuesto (*cp_vehicle*) o con una perturbación de control (*ctrl_vehicle*); las perturbaciones de control no tienen MoA. Posee otras dos columnas de *cp_time* y *cp_dose* que indican la duración del tratamiento (24, 48, 72 horas) y la dosis (alta o baja) respectivamente. La primera columna de *sig_id* contiene las diferentes muestras o fármacos como un identificador. La viabilidad celular y expresiones génicas son variables continuas. Las variables que representan las perturbaciones y dosis son variables categóricas, y finalmente, el tiempo de dosis es una variable numérica. Para el caso de las variables categóricas, el conjunto de datos debe ser filtrado por la perturbación que representan el tratamiento con un compuesto (*cp_vehicle*).

El conjunto de datos que representa las variables de salida consta de 207 columnas, las cuales 206 son las proteínas activadas y la primera corresponde a el id de cada fármaco. Los registros contienen

las proteínas como variables numéricas con representación binaria, asignando 1 para activación y 0 para no activación, en cada uno de los MoA.

3.2. Descriptiva

La Figura 4 muestra el comportamiento de dos de las variables continuas que contiene el conjunto de entrenamiento. En rojo es mostrada la distribución de la respuesta de viabilidad celular y en verdad la respuesta de la expresión génica.

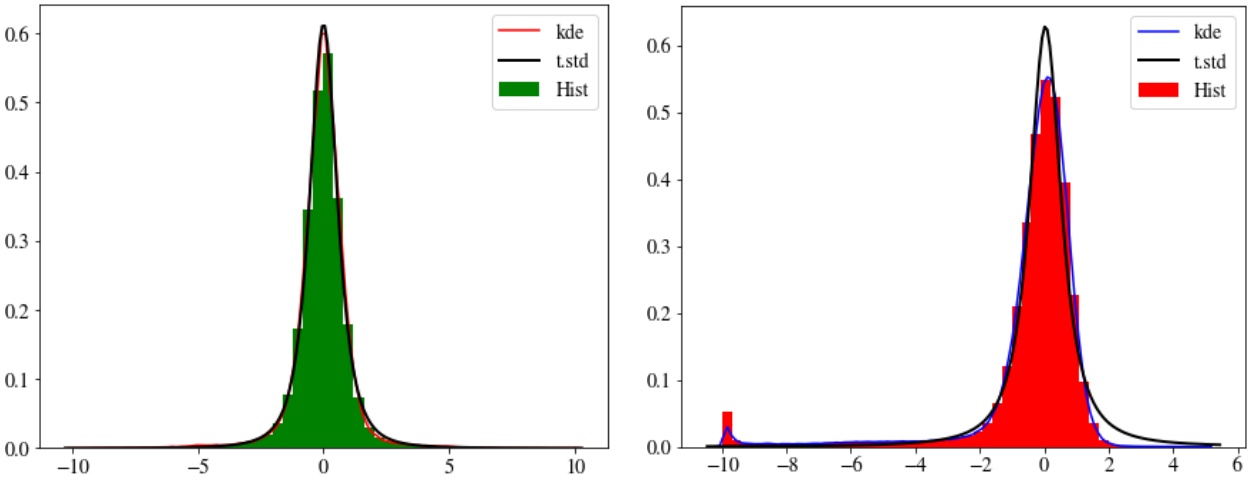

Figura 4. Distribuciones de las variables continuas. En verde se muestra la expresión génica y en rojo la viabilidad celular.

La Figura 5 muestra el conteo de las variables *cp_type*, *cp_time* y *cp_dose*.

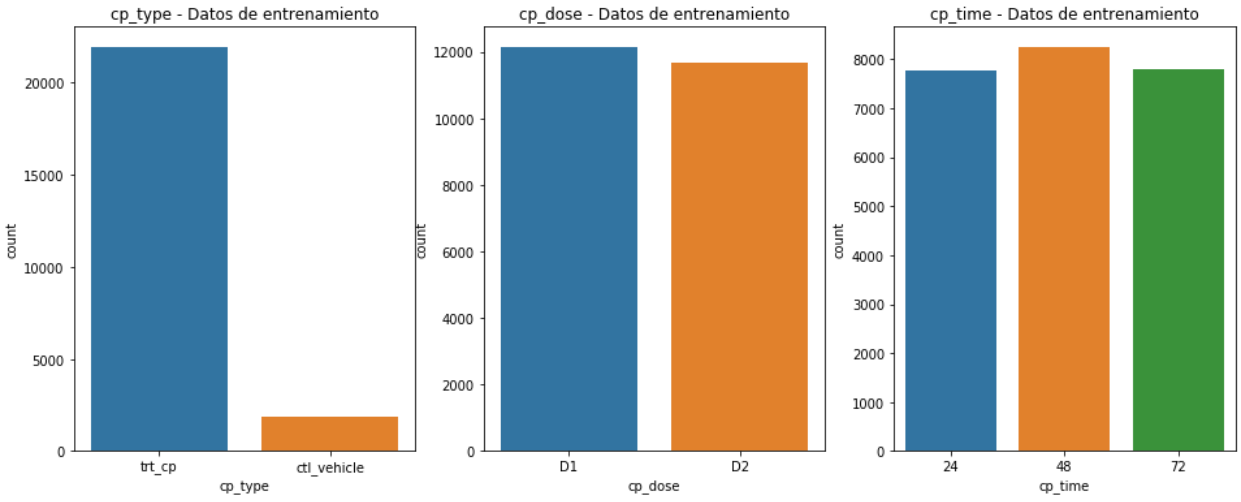

Figura 5. Variables que representan el tipo de perturbación, dosis y tiempo de dosis respectivamente para el conjunto de entrenamiento.

Como se observa en la Figura 5, el tipo de perturbación por compuesto *trt_cp* contiene la mayoría de las observaciones del conjunto de datos. Para el caso de las diferentes dosis y el tiempo de aplicación, tienen un comportamiento similar. Es importante mencionar que las distribuciones de las expresiones génicas y viabilidades celulares mostradas en la Figura 4 son similares entre sí.

A continuación, en la Figura 6, se utiliza t-SNE como técnica de reducción de dimensionalidad esto con el objetivo de representar este conjunto de datos de alta dimensionalidad, en un espacio de baja dimensionalidad. El conjunto de entrenamiento tiene una alta dimensionalidad con 774 características, mediante esta técnica se representan los datos en 2 dimensiones. En la Figura 6, se observan 8 sub gráficas, las primeras 4 corresponden a las expresiones génicas y las 4 siguientes corresponden a las muestras de viabilidad celular.

En las primeras 4 sub-gráficas correspondientes a las muestras de expresiones génicas. Se encuentra la primera sub gráfica Leiden, donde se observa una clusterización de los datos de las expresiones génicas, en 17 grupos. En la segunda sub gráfica se observa los datos *cp_type* que indican las muestras tratadas con un compuesto (*cp_vehicle*) que se encuentran coloreadas en naranja, o con una perturbación de control (*ctrl_vehicle*) que se encuentran en color gris, a partir de esto se encuentra mayor proporción de expresiones génicas tratadas con un compuesto (*cp_vehicle*). En las sub-gráficas *cp_time* se observa la duración del tratamiento (24, 48, 72 horas) la cual contiene en una escala de colores dependiendo de la duración. En la sub gráfica *cp_dose* se indica si la dosis es alta (D1) o baja (D2) respectivamente.

En las siguientes 4 sub-gráficas corresponden a las muestras de viabilidad celular. Se encuentra la primera sub gráfica Leiden, donde se observa una clusterización de los datos de las expresiones génicas, en 8 grupos. En la segunda sub gráfica se observa los datos *cp_type* que indican las muestras tratadas con un compuesto (*cp_vehicle*), en color naranja, o con una perturbación de control (*ctrl_vehicle*) en color gris. Para este caso también se tiene una mayor proporción de muestras de viabilidad celular tratadas con un compuesto (*cp_vehicle*). En la sub-gráfica *cp_time* se observa la duración del tratamiento (24, 48, 72 horas), la escala de colores depende del tratamiento. En la sub gráfica *cp_dose* se indica si la dosis es alta o baja D1 y D2 respectivamente.

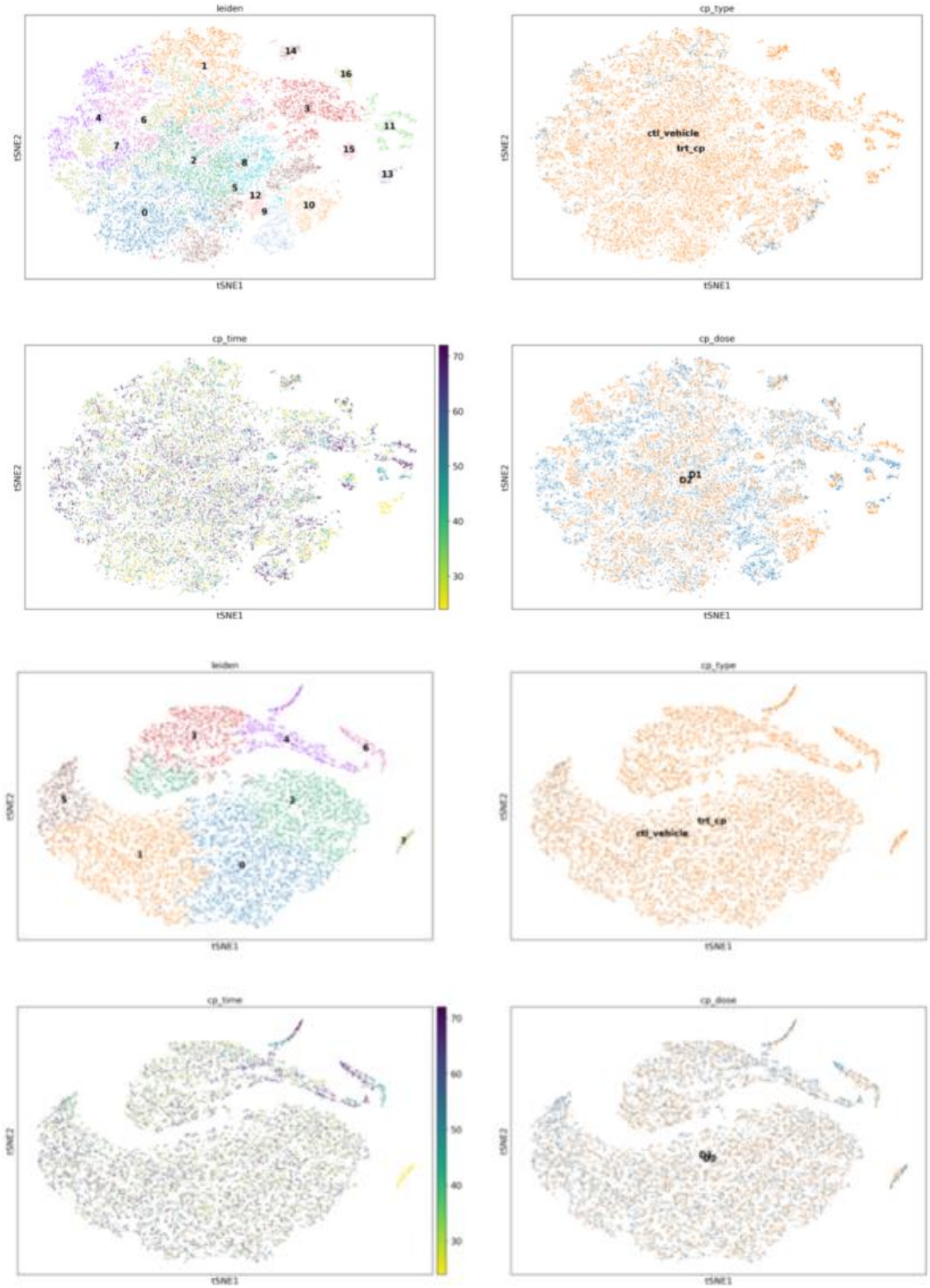

Figura 6. Variables de entrada del conjunto de entrenamiento.

En la Figura 7, se puede observar algunas de las variables de salida, del conjunto de datos que contiene las proteínas activadas. A partir de este conjunto se obtiene uno nuevo que contiene las 41 proteínas con activación mayor a 100, las cuales son usadas para el entrenamiento y predicción de los diferentes modelos.

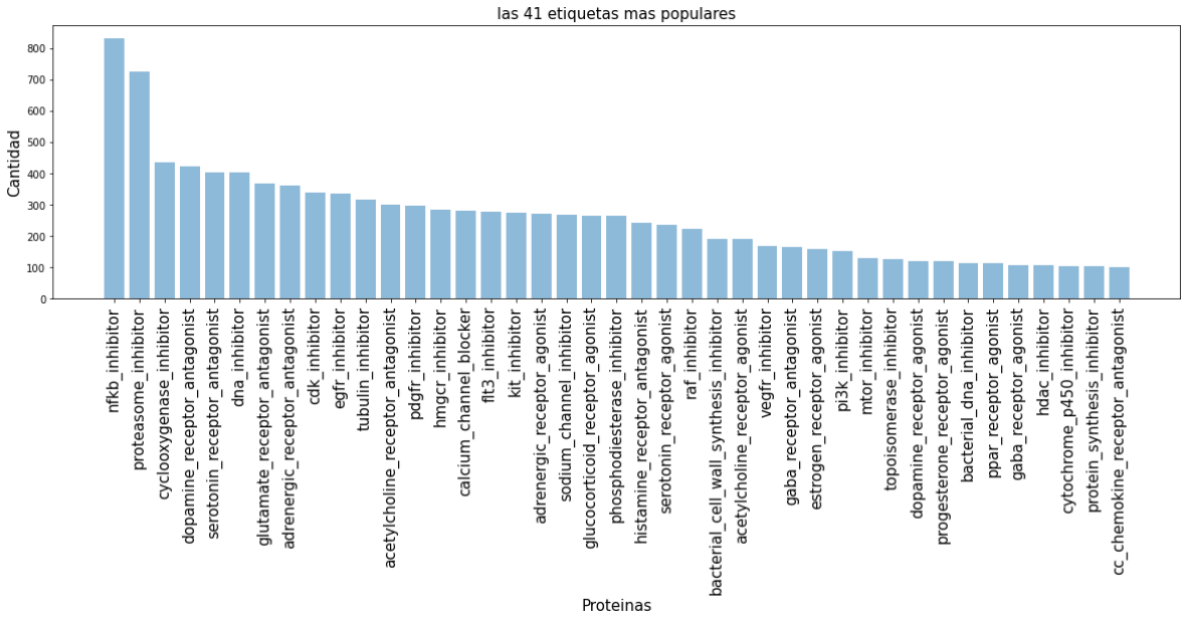

Figura 7. Conjunto de proteínas con mayor número de activaciones.

4. PROCESO DE ANALÍTICA

4.1. Pipeline principal

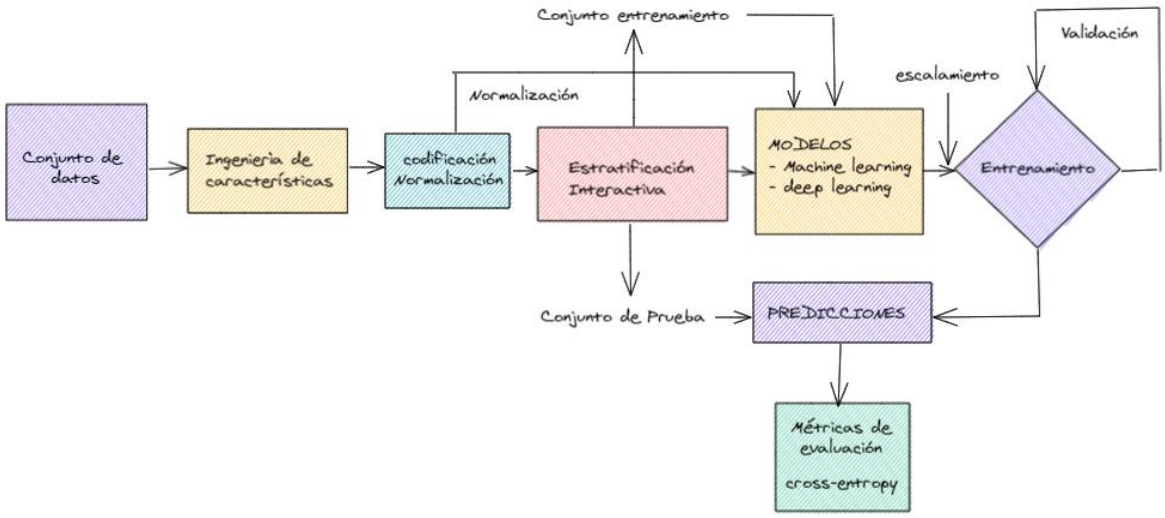

Figura 8. Pipeline principal para el entrenamiento y predicción.

El pipeline principal es mostrado en la Figura 8. Durante las diferentes etapas descritas en el pipeline es importante mencionar que, durante el proceso de entrenamiento y prueba, fue usado el conjunto de datos asignado en *kaggle* para entrenamiento, ya que existe un archivo de prueba adicional que no contiene las variables de salida o proteínas activadas. La primera parte del flujo consistió en la realización de un proceso de análisis exploratorio e ingeniería de características donde se verificó si existían datos nulos, se realizaron gráficos para observar las distribuciones de cada una de las características y se verifica el comportamiento de los estadísticos de cada una de estas variables. Para la codificación de las variables categóricas, fue mapeada cada una de estas y cambiado cada valor de formato *string* a formato numérico (mostrado en la sección preprocesamiento).

En el flujo de la Figura 8, se muestra un recuadro de partición de los datos con un método iterativo. El método usado es tomado de la librería *scikit-multilearn* [6]. Debido a que las variables de salida no contienen la misma cantidad de valores para cada una de las clases, siendo este un problema desbalanceado, el módulo de partición iterativa nos permite tener un balance en la salida, al asignar el mismo porcentaje de la clase minoritaria tanto en el conjunto de entrenamiento como el conjunto de prueba. Esto garantiza que durante el proceso tengamos una cantidad significativa de valores de

la clase minoritaria en los conjuntos de datos particionados. Para el escalamiento de las variables fue utilizado el método de máximos y mínimos, ya que con este se obtuvieron los mejores resultados durante evaluaciones preliminares. Como fue mencionado al inicio, el proceso de entrenamiento fue abordado en tres diferentes etapas, las cuales serán descritas a continuación.

4.1.1. Etapa 1: One Vs Rest

Durante la etapa 1 de este trabajo, se realizó el entrenamiento de los modelos de *machine learning* para cada una de las 41 proteínas con mayor activación en un esquema *One vs Rest*. Este tipo de procesos permite la creación de matrices de confusión y analizar la calidad del modelo en cada proteína.

Durante la etapa 1, el pipeline usado permitió realizar el escalamiento de las variables y el entrenamiento de los modelos. Para cada uno de los modelos entrenados, fueron variados ciertos hiper-parámetros de cada modelo mediante el uso del módulo *GridSearchCV*. Este módulo permite no solo la combinación de los hiper-parámetros, también permite realizar una validación cruzada con el objetivo de evitar *overfitting* durante el proceso de entrenamiento. Posteriormente al entrenamiento, fueron calculadas las métricas mencionadas en la sección 2 con su respectiva matriz de confusión.

Durante esta etapa fue asignado un paso adicional usando una librería que permite la creación de muestras sintéticas cuando existen problemas desbalanceados. Este proceso es conocido como SMOTE (*Synthetic minority oversampling technique*) [7], el cual permitió la creación de muestras adicionales de la clase minoritaria. La cantidad de muestras fueron solo el 20% del total minoritario de cada proteína, esto con el objetivo de no generar *overfitting* durante el proceso de entrenamiento.

4.1.2. Etapa 2: Modelos de clasificación multi-etiqueta

En la segunda etapa de la experimentación el problema de clasificación fue abordado usando modelos que soportan problemas multi-etiqueta. El pipeline fue similar al descrito en la Figura 8. Sin embargo, para este caso la combinación de parámetros y la validación fue realizada de manera manual por medio de *for loops*, particionando el conjunto de entrenamiento en diferentes *folds* y posteriormente, entrenando y validando para cada partición. *Random Forest* y redes neuronales fueron los modelos para esta etapa.

4.1.3. Etapa 3: Modelos basados en Deep learning

Para la última etapa de la experimentación y siguiendo un pipeline similar al mostrado en la Figura 8, fue abordado nuevamente el problema de clasificación multi-etiqueta, pero usando arquitecturas sofisticadas de redes neuronales mediante la aplicación de *autoencoders* y *denoising autoencoders*.

4.2.Preprocesamiento

Como se mencionó anteriormente, el conjunto de datos que se usa para este estudio no posee datos nulos, ni valores que no corresponden a las variables. Así que la primera parte del proceso de procesamiento para el análisis exploratorio solo consistió en revisar para cada variable su información estadística tal como; media, desviación estándar, mediana, mínimo, máximo.

Durante el proceso de entrenamiento, se debe realizar un pequeño preprocesamiento que nos permita tener nuestras variables en el mismo formato. Las variables como expresiones génicas y viabilidad celular son de formato numérico. Sin embargo, el tipo de dosis y el tiempo de dosis deben ser convertidos a formato numérico, ya que son formato *string*. Para esto, se mapean estas dos variables y se le asignan el siguiente formato:

- Tiempo de dosis: {24:1, 48:2, 72:3}
- Tipo de dosis: {D1:1, D2:2}

Con esa configuración son transformadas las variables *string* a numéricas. Uno de los principales preprocesamientos de datos es el escalado de las variables numéricas, paso que fue insertado dentro del pipeline para el entrenamiento. Para este caso fueron probados el escalado estándar (*standard scaler*) el cual es basado en la media y desviación estándar de cada variable, escalado de valores máximos y mínimos, y finalmente el escalado robusto en función del primer y tercer cuartil de los datos. Los mejores resultados se obtuvieron con el escalado de máximos y mínimos, así que decide continuar con este.

4.3. Modelos

Como fue mencionado anteriormente, el proceso de experimentación fue abordado en tres etapas. A continuación, se definirán los modelos usados en cada una de las etapas y la razón de su uso.

4.3.1. Etapa 1: One Vs Rest

Durante esta etapa fueron usados los siguientes modelos:

Regresión logística: La regresión logística es un método de clasificación lineal que permite estimar la probabilidad de una variable binaria a partir de ciertas características. Esta está fundamentada en la función logística (*logistic function*) la cual es una curva en forma de S, también llamada curva sigmoide [8].

Debido a que el problema de clasificación de los mecanismos de acción es binario, donde muestra la activación o no de las proteínas con un 1 y 0 respectivamente, la función sigmoide de la regresión logística nos permite abordar este tipo de predicciones binarias.

Random Forest (Bosques aleatorios): El *random forest* es un método de aprendizaje en conjunto o más conocido por sus siglas en inglés como ensemble learning que está formado por un conjunto de árboles de decisión individuales y permite predecir las clases en un problema de clasificación. Cada árbol es entrenado con una muestra generada mediante el proceso *bootstrapping* [9].

Este tipo de estructura de árboles de decisión individuales permite que las variables de entrada sean particionadas de una manera efectiva para encontrar la mejor solución al problema de clasificación, en este caso, una activación de cada proteína.

Máquinas de soporte (support vector classifier): Las máquinas de soporte están fundamentadas en el máximo margen de clasificación y se relaciona con un hiperplano que separa las clases a predecir. Este hiperplano es construido en un espacio N-dimensional (N siendo el número de características de entrada) que clasifica y separa de manera distinta cada una de las clases. Estos son los límites de decisión del modelo.

Este tipo de separaciones por medio de hiperplanos permite a las máquinas de soporte ser un método robusto para separar de manera acertada las clases del problema binario de activación de proteínas al uso de fármacos. Es por lo que este método nos permite abordar el problema de clasificación [10].

XGBoost (aumento de gradiente extremo): El método de aumento de gradiente o *boosting* cae en la clasificación de los métodos de aprendizaje en conjunto o ensemble learning como fue mencionado anteriormente para el modelo *random forest*. Este tipo de técnica trata de tomar clasificadores débiles en conjunto para crear un clasificador robusto mediante una técnica *bagging* [11].

El método *XGBoost* permite aplicar una regularización sobre los árboles de decisión débiles usados en un *random forest*. Esto permite que el este modelo de *boosting* sea más robusto y pueda generar mejores clasificadores para el problema binario en estudio. Adicionalmente, este método *XGBoost* permite el uso de herramientas como GPU para paralelizar y aumentar la velocidad de las corridas.

4.3.2. Etapa 2: Modelos de clasificación multi-etiqueta

Durante la etapa 2 de este estudio, se realizó el entrenamiento de los modelos de machine learning abordando el problema como clasificación multi-etiqueta. Es decir que las clases no eran mutuamente excluyentes, y para un fármaco se podían activar varias proteínas. En esta etapa se usaron modelos que soportan este tipo de problemas, tales como:

Random forest: Este método fue utilizado en la etapa 1 en el proceso *One vs Rest*, sin embargo, también permite abordar un problema multi-etiqueta.

Redes neuronales: Una red neuronal es un modelo simplificado que emula el modo en cómo el cerebro humano procesa la información. Estas unidades de procesamiento se organizan por capas; una capa de entrada, capas ocultas y una capa de salida. Cada una de estas capas contiene una cantidad de nodos que hacen referencia a la cantidad de información o variables que se quiere procesar [12].

Para esta sección de la etapa 2, se crearon estructuras de redes neuronales usando librerías de *scikit learn* y *tensorflow* (tf). Usando la librería de tf fue diseñada la arquitectura mostrada en la Figura 9. Se utiliza la función de activación ‘relu’ en las capas ocultas y en la capa de salida se usa una función de activación sigmoide que convierte cada puntuación del nodo final entre 0 y 1 independientemente de las otras puntuaciones. La razón por la que la puntuación final es independiente es porque cada proteína puede activar varios fármacos. También, se especifica el número de etiquetas de destino que corresponde a las 41 proteínas que tienen más de 100 activaciones.

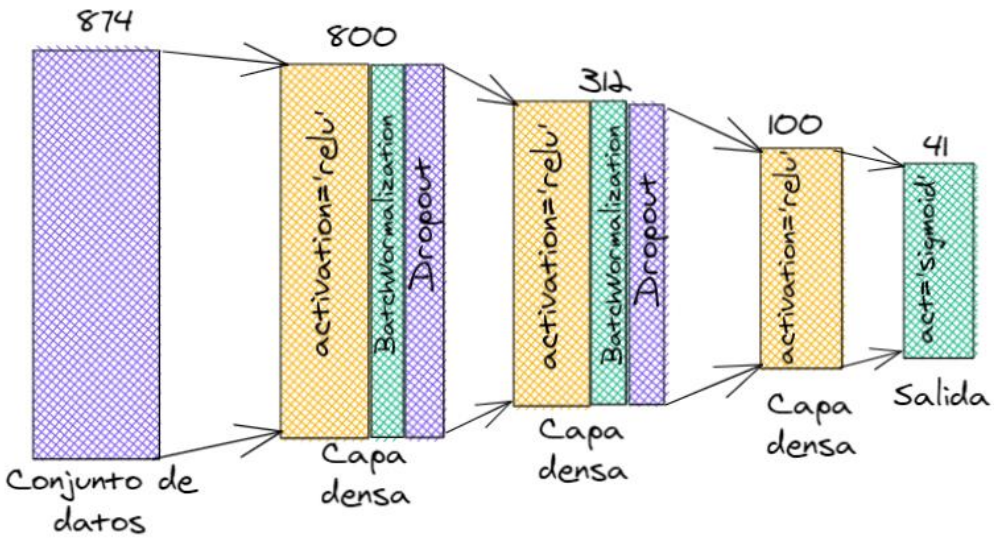

Figura 9. Arquitectura de red neuronal usando la librería de *tensorflow*.

Para la red neuronal de la librería de *scikit-learn*, fue diseñada una arquitectura sencilla con una capa de entrada de 874, siendo la cantidad de variables de viabilidad celular y expresiones génicas, una capa oculta con activación tangente hiperbólica y 900 nodos, y una capa de salida con las 41 proteínas a predecir. Esta arquitectura es mostrada en la Figura 10.

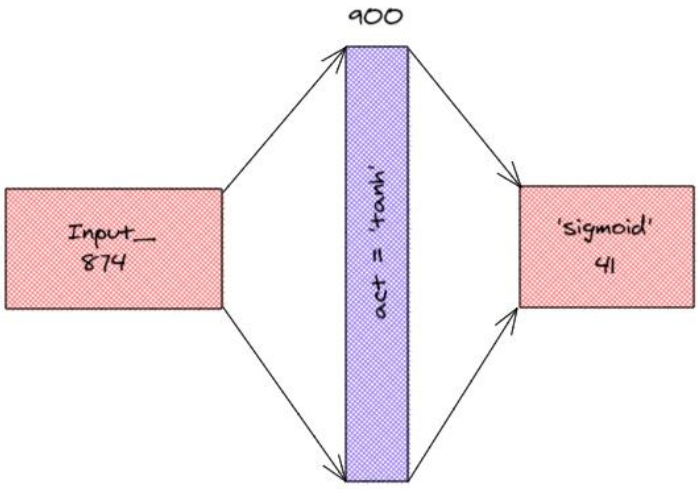

Figura 10. Arquitectura de red neuronal usando la librería de *scikit-learn*.

4.3.3. Etapa 3: Modelos basados en Deep learning

En la última etapa de experimentación son utilizados modelos basados de *deep learning* más sofisticado. Son formadas arquitecturas de *autoencoders* [13] y *denoising autoencoders* [14], generando ruido para entrenar las variables de entrada y un *deep autoencoder* donde es diseñada una arquitectura para la predicción de los MoA, esto con el objetivo de abordar el problema multi-etiqueta. Esta sección consiste en dos procesos, el primero se diseña una arquitectura de *autoencoder* para entrenar el conjunto de datos original y el codificador de este es usado en una nueva arquitectura de redes neuronales. El segundo proceso consiste en el entrenamiento de *autoenconder* agregando ruido al conjunto de datos original y el codificador de este es usado en una nueva arquitectura de red neuronal. El esquema de *autoencoder* del primer proceso es mostrado en la Figura 11:

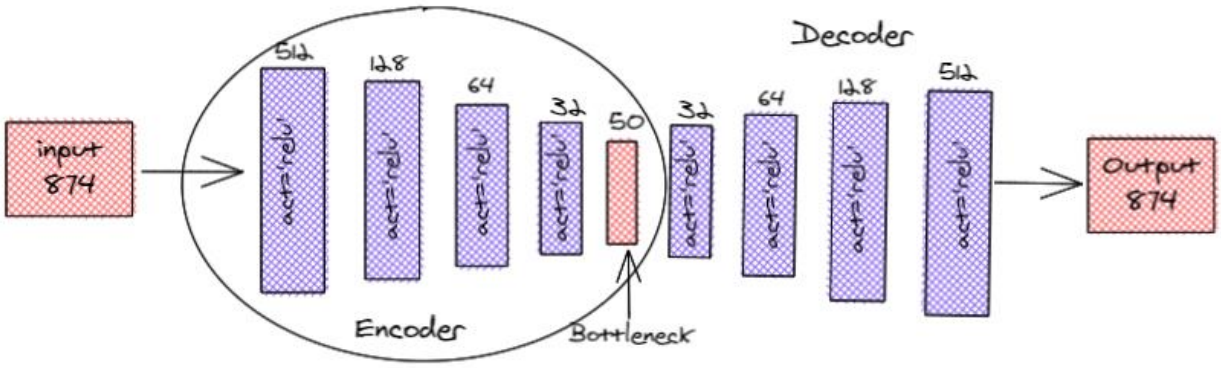

Figura 11. Arquitectura de red neuronal para el primer proceso de la etapa 3.

Subsiguiente al entrenamiento usando la arquitectura anterior con *autoencoder*, es usado el codificador en una nueva arquitectura la cual realizará la predicción de las 41 proteínas seleccionadas en la parte inicial. La arquitectura de red neuronal diseñada y usando el codificador del primer proceso es mostrado en la Figura 12.

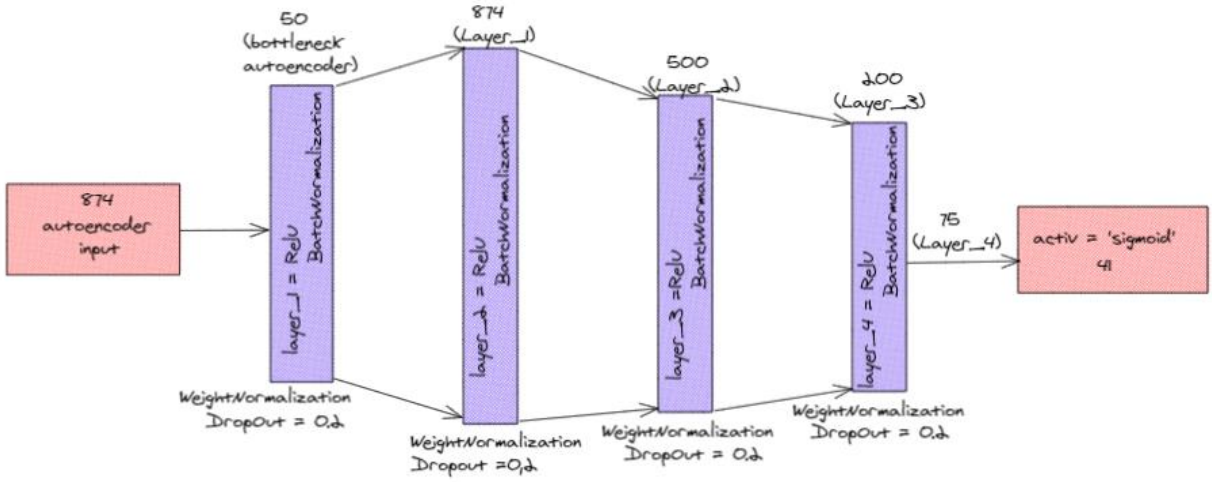

Figura 12. Arquitectura de redes neuronales para la predicción de los MoA usando el codificador de la Figura 11.

Para el proceso de 2 de la etapa tres, se usa un *denoising autoencoder* donde se genera ruido blanco mediante una distribución aleatoria normal con 0.2 de media y desviación estándar de 0.05. Esto tiene como objetivo evitar entrenar la siguiente arquitectura con algún dato atípico. El ruido tiene la forma del conjunto de datos de entrenamiento (*X_train*) con 17558 filas y 874 columnas. La Figura 13 muestra el esquema de la arquitectura *de denoising autoencoder*.

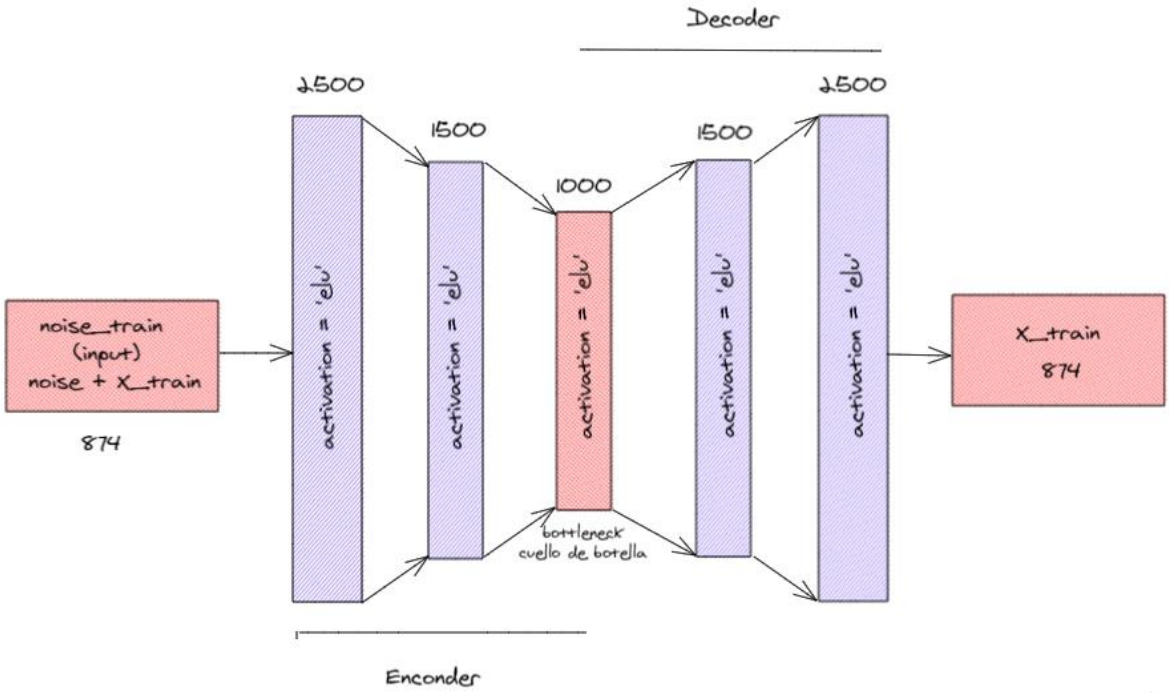

Figura 13. Arquitectura de red neuronal usada para el *denoising autoencoder*.

Posterior al entrenamiento usando la arquitectura de *denoising autocoder*, el input y la capa que representa el cuello de botella o codificador son insertados en una nueva arquitectura, la cual realizará la predicción de las 41 proteínas seleccionadas en la parte inicial. Esta arquitectura puede ser observada en la Figura 14.

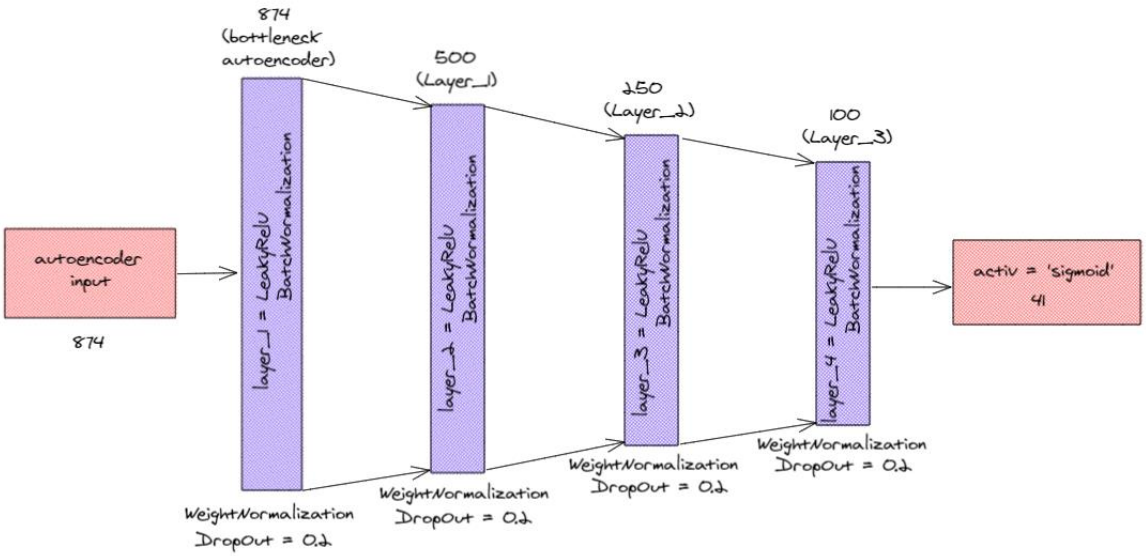

Figura 14. Arquitectura de redes neuronales para la predicción de los MoA usando el codificador de la Figura 13.

5. METODOLOGÍA

5.1. Baseline

Durante las pruebas iniciales fueron tomadas dos proteínas, una con activaciones mayores a 100 y otra con activaciones menores a 50. Los modelos de clasificación inicialmente probados durante el estudio fueron: MultinomialNB (modelo de Bayes multinomial para clasificación), regresión logística, *random forest* y la máquina de soporte (SVC). Este esta primera iteración no fue considerada la pérdida logarítmica como métrica de evaluación, sin embargo, fueron calculadas las siguientes métricas: *accuracy*, *balanceda_accuracy* y *f1_score*. Las principales dificultades que fueron observadas durante la etapa inicial fueron; el entendimiento técnico del problema, el desbalance del conjunto de datos, y la alta dimensionalidad del problema.

A partir de las dificultades observadas y los resultados obtenidos en esta parte inicial, se decide usar una partición iterativa de los datos con una librería especialmente enfocada a tratar con datos desbalanceados [6], considerando que para las proteínas que se tenían una cantidad de activaciones

menores a 50, no se obtuvieron los resultados esperados debido a la partición de los datos. En la revisión de los resultados de los modelos, se verifica que la SVC tuvo un mejor rendimiento comparado con los otros modelos. Este modelo fue la base para las diferentes etapas ejecutadas durante la experimentación. Por sus buenos resultados, el modelo de regresión logística y *random forest* fueron también considerados en estas etapas.

5.2. Validación

Para cada etapa del actual estudio, se realizó la partición del conjunto de datos mediante un módulo de la librería *scikit-multilearn*. Este módulo (*iterative_train_test_split*) permite tratar particiones en un conjunto de datos desbalanceados y de múltiples etiquetas. A partir de esta librería son realizadas las particiones asignando un 70% para entrenamiento y 30% para prueba.

Durante el proceso de entrenamiento, se usaron librerías de *scikit-learn* que permitían el particionamiento del conjunto de datos para la validación. Los módulos usados para este particionamiento fueron *StratifiedShuffleSplit* y *StratifiedKFold*, y la cantidad de *folds* o particiones para la mayoría de las corridas realizadas en cada una de las etapas del estudio fueron 5.

Para la primera etapa en el proceso *One Vs Rest*, donde fueron entrenados los modelos para cada una de las proteínas, se usó el módulo *GridSearchCV* que permitía realizar la combinación de hiper-parámetros, entrenamiento y validación. Obteniendo de estos pasos anteriores, la mejor métrica y combinación de parámetros para cada uno de los modelos. Durante la etapa dos, donde fue abordado el problema como multi-etiqueta, se realizó el respectivo entrenamiento, validación y combinación de parámetros manera manual a partir de *for loops* para cada modelo. Para la última etapa donde fueron usados modelos sofisticados de *Deep learning* se realiza en dos procesos. El primero son entrenados los modelos a partir de *for loops* como la etapa 2 para la parte de combinación de parámetros y validación y para el segundo proceso fue usado un *wraper* de *Keras* que permite realizar el entrenamiento de las arquitecturas de redes neuronales usando el módulo *GridSearchCV*.

5.3. Iteraciones y Evolución

A partir de los resultados obtenidos con las primeras iteraciones donde fueron usadas dos proteínas, con mayor y menor cantidad de activaciones, se procede a revisar cada uno de los modelos de *machine learning* para clasificación con el objetivo de definir cuáles de estos nos permitían tratar

el problema de clasificación multi-etiqueta y adicionalmente, módulos de partición que puedan tratar el conjunto de datos desbalanceados. Esto se logra observando que para las particiones del conjunto de datos se hacía necesario contar con módulos que permitieran hacer una partición adecuada e iterativa, ya que para la proteína que tiene activaciones menores a 50, la cantidad de datos de la clase minoritaria asignada al conjunto de prueba después de la partición no era significativa y los resultados no eran los esperados. Adicionalmente, se hace necesario la evaluación y ejecución del estudio con modelos sofisticados con arquitectura especiales de *Deep learning*.

Para las siguientes iteraciones se decide usar un módulo de partición iterativa de la librería *scikit-multilearn*, esto con el fin de darle el peso adecuado a las clases desbalanceadas y poder asignar el mismo porcentaje de estas clases en los conjuntos de entrenamiento y prueba para cada proteína. Se realizan iteraciones con esta librería y los diferentes modelos mencionados en las secciones anteriores para cada una de las tres etapas. Durante la etapa 1 se tuvo como objetivo entrenar los modelos para cada una de las proteínas siendo usados en orden jerárquica descendente a partir de los resultados obtenidos de las primeras iteraciones: regresión logística, *random forest*, SVC *y XGBoost*. Este último fue usado debido a buenos resultados obtenidos en propuestas revisadas en la literatura. En esta etapa y considerando que, para muchas de las 41 proteínas seleccionadas, aún existía una diferencia significativa en las dos clases se decide a usar técnicas para la creación de datos sintéticos para el entrenamiento de los modelos, la técnica usada es conocida como SMOTE. Este módulo fue importado de la librería *imblearn* y el porcentaje creado fue del 20% total de la clase minoritaria para cada proteína. El objetivo del uso de esta librería era observar si durante la etapa del entrenamiento, estos datos adicionales permitían una mejora en los resultados.

Durante la segunda etapa de esta experimentación, donde el problema de clasificación fue abordado como multi-etiqueta. Así, se deciden experimentar tres modelos, el primero fue el *modelo random forest* y el segundo fue un modelo de redes neuronales por medio de dos librerías diferentes, *scikit-learn* y *tensorflow*. De la librería de *tensorflow* se empieza la experimentación de redes neuronales sencillas y estos resultados son comparados con la arquitectura de redes neuronales que ofrece la librería de *scikit-learn*.

Finalmente, en la tercera etapa se utilizaron modelos sofisticados de *Deep learning*, como arquitecturas de redes neurales con *autenconder*s. La idea de los *autoencoders* era que las 874

variables de entrada quedaran condensadas en la capa más interna en el *bottleneck* o cuello de botella para reconstruir la misma información, y después utilizar otra arquitectura para buscar una menor pérdida logarítmica. Para este caso fueron usados dos tipos de *autoencoders*, uno donde solo fueron entrenados los datos originales y otro donde fue generado ruido blanco sobre los datos de entrenamiento, también llamado *denoising autoencoders*. Con esta última etapa finalizó la serie de experimentos del actual estudio.

5.4. Herramientas

Debido al alta dimensionalidad del problema se requería alto costo computacional, este costo se lograba identificar en las iteraciones realizadas en máquinas locales, donde podían tardar para modelos más sofisticados como máquinas de soporte, *XGBoost* y redes neuronales, entre 3 a 5 días. Es así como se decidió la implementación de la librería de *machine learning* que aprovecha recursos como GPU (https://docs.rapids.ai/api/cuml/stable/), la cual que permite la paralelización del método de máquinas de soporte y era posible pasar de iteraciones de 4 días a solo media hora. El acelerador de GPU usado fue proporcionado por la plataforma *Kaggle* que asigna por semana un total de 35 horas. Este acelerador de *kaggle* también fue usado para las diferentes iteraciones de la etapa 2 y 3, reduciendo el tiempo computacional. A continuación, serán listadas librerías y módulos usados:

- **sklearn.model_selection**: KFold, StratifiedKFold, StratifiedShuffleSplit, ShuffleSplit, train_test_split, GridSearchCV
- **sklearn.linear_model**: LogisticRegression
- **sklearn.preprocessing**: LabelEncoder, MinMaxScaler, RobustScaler
- **sklearn.metrics**: accuracy_score, f1_score, balanced_accuracy_score, plot_confusion_matrix, confusion_matrix, log_loss, recall_score, precision_score, ConfusionMatrixDisplay
- **sklearn.tree**: DecisionTreeClassifier
- **sklearn.ensemble**: RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
- **skmultilearn.model_selection**: iterative_train_test_split, IterativeStratification
- **sklearn.svm**: SVC
- **xgboost**: XGBClassifier
- **sklearn.pipeline**: Pipeline
- **sklearn.utils.class_weight**: compute_class_weight

- **imblearn.over_sampling**: SMOTE
- **tensorflow.keras.layers:** layer, Input, Dense
- **tensorflow.keras.callbacks:** ReduceLROnPlateau, ModelCheckpoint, EarlyStopping
- **tensorflow_addons**
- **cuml.svm:** SVC

6. RESULTADOS

6.1. Métricas

A continuación, serán presentados los resultados para cada una de las etapas.

6.1.1. Etapa 1: One Vs Rest

La Tabla 1 muestra los resultados obtenidos durante la etapa 1.

Tabla 1. Resultados obtenidos durante la etapa 1.

Modelos	**Medidas**					
	Sin SMOTE					
	log loss	*Accuracy*	*bal_accuracy*	*recall*	*precision*	*f1_score*
Regresión logística	0.1613	0.424	0.359	0.293	0.195	0.223
Random Forest	0.241	0.413	0.348	0.282	0.168	0.183
Máquina de soporte vectorial	0.0384	0.454	0.382	0.302	0.183	0.212
XGBoost	0.0416	0.449	0.443	0.438	0.158	0.193
	Con SMOTE					
Regresión logística	0.159	0.450	0.383	0.315	0.186	0.217
Random Forest	0.368	0.405	0.342	0.278	0.175	0.187
Máquina de soporte vectorial	0.039	0.438	0.379	0.319	0.194	0.219
XGBoost	0.0423	0.449	0.451	0.440	0.159	0.192

Como se evidencia en la Tabla 1, el error sopesado calculado para las métricas *accuracy*, *balanced-accuracy*, *recall_score*, *precision_score*, *f1_score*, tuvieron una leve mejoría al aplicar la técnica de SMOTE, ya que se tuvo un 20% más de muestras durante el proceso de entrenamiento. Sin embargo, no en todos los casos con el SMOTE, mejoraron significativamente la pérdida logarítmica, como se evidencia en los modelos *random forest*, SVC y XGBoost. Los mejores resultados de pérdida logarítmica fueron obtenidos cuando fue usada la SVC.

Dos de las métricas que permiten evaluar la calidad del modelo para esta primera etapa son: *recall* y *precision*. La primera nos permite conocer la tasa de verdaderos positivos y la segunda nos permite identificar que tanto el modelo se está equivocando a la hora de predecir la clase monitoria (verdaderos positivos), como lo muestran las ecuaciones de la sección 2. Como se evidencia en la Tabla 1, el modelo XGBoost presenta un valor de *recall* mayor, sin embargo, la métrica *precision* es baja. Mostrando que puede tener una mejor clasificación de la clase minoritaria, pero se equivoca más con la clase mayoritaria, con más falsos positivos. A partir de esto, se observa que la SVC continúa siendo un modelo bueno, teniendo un mejor resultado de estas dos métricas. Para evaluar cuál de los modelos predice de manera efectiva las dos clases, fue construida una matriz global para cada uno de estos. La Figura 15 muestra la matriz global para el mejor resultado de la primera etapa, siendo la SVC.

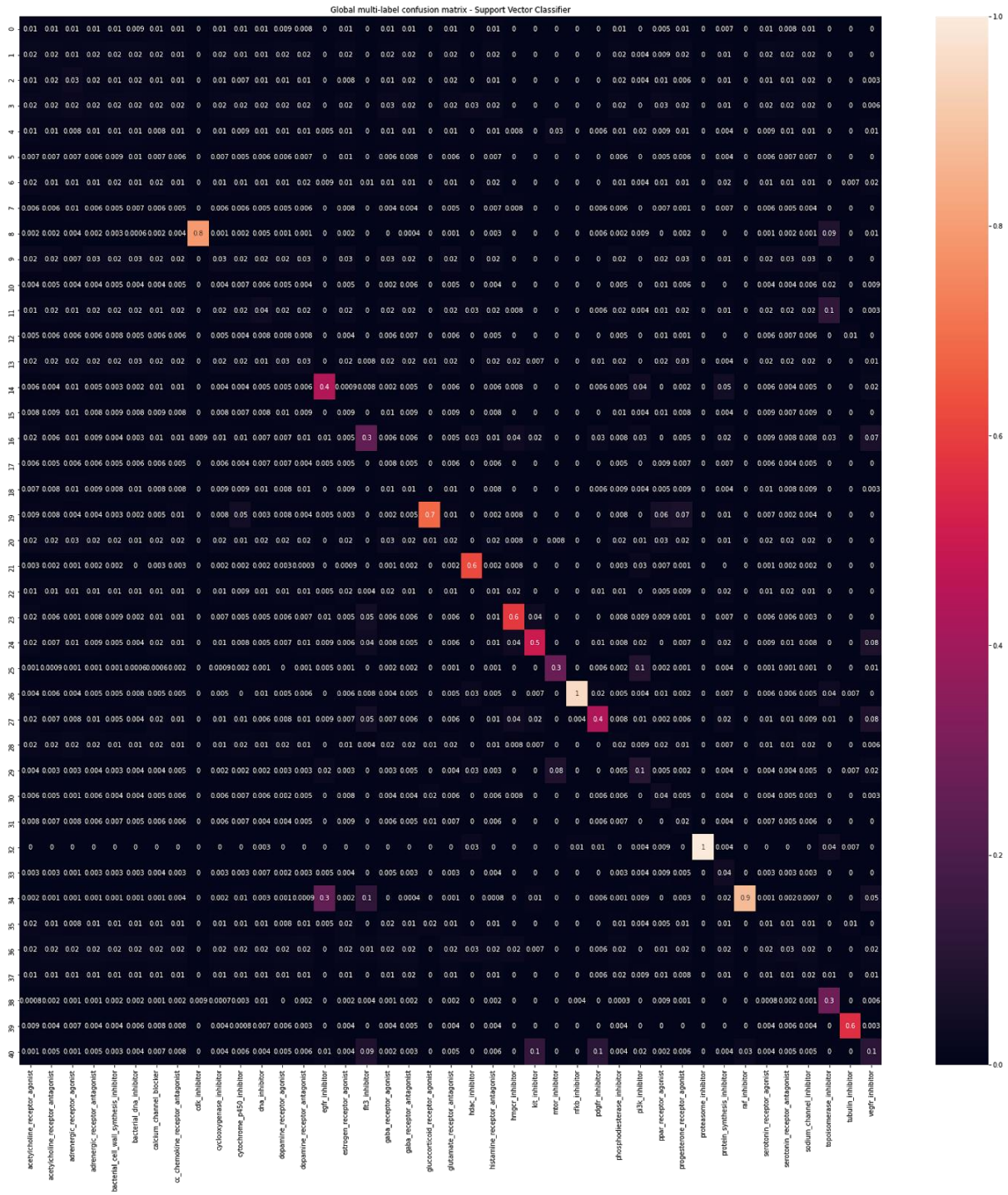

Figura 15. Matriz global para la máquina de soporte.

Aquí se pueden identificar las proteínas donde la SVC consiguió predecir de manera eficiente tanto la clase mayoritaria como minoritaria. Los recuadros de la diagonal principal muestran los valores que el modelo consiguió predecir de manera correcta en colores claros y los recuadros por fuera de

la diagonal que presentan colores oscuros son aquellos donde el desbalance de las clases continúa siendo fuerte y es posible que los modelos no sean capaces de predecir de manera correcta.

6.1.2. Etapa 2: Modelos de clasificación multi-etiqueta

La Tabla 2 muestra los resultados obtenidos usando los modelos que soportan el problema multi-etiqueta.

Tabla 2. Resultados obtenidos durante la etapa 2.

Modelos multi-etiqueta	
	log loss
Red Neuronal (*tensorflow*)	0.042
Random Forest	0.047
Red neuronal (*scikit-learn*)	0.058

Como fue mencionado en la sección 4, son usadas dos diferentes arquitecturas de redes neuronales para esta etapa. Se puede observar en la Tabla 2 que cuando el problema es abordado como multi-etiqueta, el modelo *random forest* es capaz de obtener una pérdida logarítmica mucho menor que la obtenida en la etapa 1. Sin embargo, para esta etapa el mejor resultado fue el obtenido con la red neuronal descrito en la Figura 9.

6.1.3. Etapa 3: Modelos basados en Deep learning

Como fue descrito en la sección 4, para este caso fueron probados dos modelos de redes neuronales usando *autoencoders*. El primero *autoencoder* fue entrenado solo con los datos de entrenamiento y el segundo fue entrenado generando ruido sobre los datos de entrenamiento. La Tabla 3 presenta los resultados obtenidos durante esta etapa.

Tabla 3. Resultados obtenidos durante la etapa 3.

Modelos - Deep Learning	
Redes neuronales	log loss
Denoising autoenc	0.041
autoencoder	0.047

Se observa que usando la arquitectura de *autoencoder* agregando ruido sobre los datos de entrenamiento, se obtiene un mejor resultado. Siendo esta una buena estrategia de redes neuronales.

En la ejecución de las arquitecturas de redes neuronales tanto en la etapa 2 como la 3, se evidencia que se obtienen resultados similares con respecto a la métrica de evaluación. Sin embargo, el mejor resultado obtenido para las 41 proteínas seleccionadas fue usando la SVC de la etapa 1, con un valor de pérdida de 0.038.

6.2. Evaluación Cualitativa

Una de las principales apuestas para este tipo de proyectos es poder encontrar un modelo predictivo que permita el cálculo de la probabilidad de que una o un conjunto de proteínas sean inhibidas o activadas con el uso de diferentes tipos de fármacos. Es por esto por lo que la métrica de evaluación, siendo la pérdida logarítmica, es un buen indicativo de medir la efectividad de este tipo de ejercicios. De las principales dificultades de este tipo problema de clasificación es el desbalance de las clases que están siendo predichas. Esto puede ocasionar que durante el proceso de entrenamiento se tenga un buen resultado de la pérdida, debido a que se tienen más valores de la clase minoritaria en este conjunto, pero sea un mal predictor con el archivo de prueba. Esto genera que exista *underfitting* durante el proceso. Así que es importante el uso de técnicas que permitan tratar datos desbalanceados para evitar este tipo de casos de *overfitting* o *underfitting* y se pueda tener un mejor resultado de métrica y mejores modelos predictivos.

Un modelo con un valor de pérdida bajo es un indicativo de que el modelo predictivo es bueno a la hora de predecir la probabilidad de activación. Esta es una de las maneras de evaluación para verificar la eficiencia del desempeño de un fármaco sobre ciertas patologías cuando son usados modelos predictivos. Se espera tener un modelo que permita predecir de manera individual o conjunta esta probabilidad, ya que como se tienen tantas pruebas de laboratorio, se hace difícil la predicción poco eficiente de cada una de las proteínas. A partir de esto, se evidencia que la SVC se alinea con los resultados esperados para la pérdida logarítmica, ya que este fue el valor más pequeño obtenido. Sin embargo, es importante mencionar que en las etapas 2 y 3, también son obtenidos resultados satisfactorios ya que la métrica del negocio estuvo en el rango entre 0.041 y 0.047. Estos resultados nos permiten concluir que tanto la SVC como las redes neuronales son modelos predictivos eficientes para tratar este tipo de problemas farmacológicos.

6.3. Consideraciones de producción

Dentro de las consideraciones técnicas que pueden ser tenidas en cuenta a la hora de considerar la puesta en producción de este tipo de problemas farmacológicos, es inicialmente relacionarlo a un

tipo de proyecto llamado mapas de conectividad (https://clue.io/). Este tipo de herramientas muestra las relaciones que existen entre diferentes tipos de células ya probadas de manera experimental. Estos mapas de conectividad permiten identificar las relaciones funcionales entre células, logrando identificar aquellas que ya poseen una evaluación de los mecanismos de acción con aquellas que aún no fueron evaluadas con respecto a este tipo de respuestas. Esto permite no solo disminuir errores mayores en los modelos predictivos, también poder relacionar muchos tipos de células con funcionalidades similares.

A partir de lo anterior, es posible primero realizar diferentes procesos de entrenamiento con el objetivo de calcular la eficacia de modelos a la hora de predecir estos mecanismos de acción. Esto usando diferentes conjuntos de datos y evaluando las diferentes proteínas activadas, ya que se espera que, si las células cuentan con funcionalidades similares, estas proteínas sean las mismas o significativamente similares a otros conjuntos de datos sin evaluación. Esto ayudaría inicialmente a tener modelos predictivos eficientes, a partir del seguimiento que se puede tener en la medición de la métrica de desempeño y también a poder relacionar los datos usados para en el entrenamiento para nuevos conjuntos en dos pasos, en el primer es necesario conocer los mapas de conectividad para ver las diferentes células y sus funcionalidades, esto permitiría saber de manera clara cuales activan proteínas similares y con esto pasar a un segundo paso, donde este nuevo conjunto de datos será usado para predecir las diferentes activaciones. El aporte de este tipo despliegues ayudaría a las empresas farmacéuticas a reducir los tiempos de evaluación, pruebas de laboratorio y por ende un ahorro presupuestal significativo.

7. CONCLUSIONES

- Este proyecto muestra de manera general un contexto de las estrategias usadas para el estudio de los efectos de los fármacos a través de los mecanismos de acción (MoA), y de cómo estas son abordadas desde diferentes perspectivas. Así, que el entendimiento del conjunto de datos fue fundamental para abordar el problema e identificar las principales variables que influyen en la predicción de estos mecanismos de acción. Este campo sigue estando en continuo crecimiento y exploración, el cual, por medio del uso de tecnología, algoritmos computacionales, etc.; permiten grandes descubrimientos para el beneficio de la humanidad.
- La ejecución del proyecto permitió la construcción de arquitecturas de modelos de *machine learning* y *Deep learning* sofisticados para cada una de las etapas, con el objetivo de obtener

mejoras en las métricas de decisión y así aportar modelos predictivos necesarios para el estudio eficiente de los mecanismos de acción de las células cuando son probados diferentes productos farmacológicos. Una de las principales ventajas de estos modelos predictivos, es que permite reducir tiempo de pruebas de laboratorio y la aceleración en el proceso de desarrollo de nuevos fármacos.

- La aplicación de metodologías de validación durante el proceso de entrenamiento para cada una de las etapas fue un insumo fundamental para la búsqueda de la mejor combinación de hiper-parámetros. Influyendo significativamente en la mejora de la pérdida logarítmica.
- Uno de los principales objetivos durante el estudio fue reducir la pérdida logarítmica durante cada una de las etapas ejecutadas. Este objetivo se cumple, considerando que las pérdidas obtenidas en algunos modelos de la primera etapa son altos, sin embargo, la máquina de soporte continúa siendo el mejor resultado obtenido.
- Para trabajos futuros, se espera la implementación de cada una de las etapas anteriormente ejecutadas, pero para el total de proteínas que se tienen en el conjunto de datos. Este desafío sería una actividad que permita seguir probando los modelos con datos desbalanceados, ya que las proteínas faltantes contienen activaciones menores a 100, y permitiría tener una evaluación y comparación mayor con los resultados que fueron obteniendo para este ejercicio de *kaggle*.

8. BIBLIOGRAFIA

[1] F. Li *et al.*, "DrugMoaMiner: A computational tool for mechanism of action discovery and personalized drug sensitivity prediction," *3rd IEEE EMBS Int. Conf. Biomed. Heal. Informatics, BHI 2016*, pp. 368–371, 2016, doi: 10.1109/BHI.2016.7455911.

[2] D. C. Swinney and J. A. Lee, "Recent advances in phenotypic drug discovery," *F1000Research*, vol. 9, pp. 1–9, 2020, doi: 10.12688/f1000research.25813.1.

[3] S. M. Corsello *et al.*, "Discovering the anticancer potential of non-oncology drugs by systematic viability profiling," *Nat. Cancer*, vol. 1, no. 2, pp. 235–248, 2020, doi: 10.1038/s43018-019-0018-6.

[4] E. Gonçalves *et al.*, "Drug mechanism-of-action discovery through the integration of pharmacological and CRISPR screens," *Mol. Syst. Biol.*, vol. 16, no. 7, pp. 1–14, 2020, doi:

10.15252/msb.20199405.

[5] R. L. Davis, “Mechanism of Action and Target Identification: A Matter of Timing in Drug Discovery,” *iScience*, vol. 23, no. 9, p. 101487, 2020, doi: 10.1016/j.isci.2020.101487.

[6] K. Sechidis, G. Tsoumakas, and I. Vlahavas, “On the stratification of multi-label data,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6913 LNAI, no. PART 3, pp. 145–158, 2011, doi: 10.1007/978-3-642-23808-6_10.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. Philip Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” *J. Artif. Intell. Res.*, vol. 16, no. 2, pp. 321–357, 2002, doi: 10.1002/eap.2043.

[8] P. Yarnold and L. G. Grimm, “Logistic Regression,” in *Reading and understanding multivariate statistics*, 1995, pp. 217–244.

[9] T. K. Ho, “Random Decision Forests,” *Proc. Int. Conf. Doc. Anal. Recognition, ICDAR*, vol. 1, pp. 278–282, 1995, doi: 10.1109/ICDAR.1995.598994.

[10] I. Steinwart and A. Christmann, *Support Vector Machines*. Springer Publishing Company, Incorporated, 2008.

[11] T. Chent and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 785–794, 2016.

[12] C. M. Bishop, *Neural Networks for Pattern Recognition*. 1995.

[13] P. Baldi, “Autoencoders, Unsupervised Learning, and Deep Architectures,” *ICML Unsupervised Transf. Learn.*, pp. 37–50, 2012, doi: 10.1561/2200000006.

[14] P. Vincent and H. Larochelle, “Extracting and Composing Robust Features with Denoising,” in *Proceedings of the 25 th International Conference on Machine Learning*, 2008, pp. 1096–1103, doi: https://doi.org/10.1145/1390156.1390294.