ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

EVALUACIÓN DEL DESEMPEÑO COMPUTACIONAL DE SISTEMAS DE RECOMENDACIÓN APLICADO A BASES DE DATOS FARMACOLÓGICAS

IMPLEMENTACIÓN DE UN SISTEMA DE RECOMENDACIÓN HÍBRIDO PARA PREDICCIÓN DE INTERACCIONES FARMACOLÓGICAS

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO/A EN CIENCIAS DE LA COMPUTACIÓN

JUAN PABLO RODRÍGUEZ ZUMÁRRAGA

juan.rodriguez03@epn.edu.ec

DIRECTOR: IVÁN MARCELO CARRERA IZURIETA

ivan.carrera@epn.edu.ec

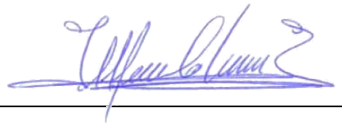
03 de marzo, 2023

CERTIFICACIONES

Yo, JUAN PABLO RODRÍGUEZ ZUMÁRRAGA declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

JUAN PABLO RODRÍGUEZ ZUMÁRRAGA

Certifico que el presente trabajo de integración curricular fue desarrollado por JUAN PABLO RODRÍGUEZ, bajo mi supervisión.



IVÁN MARCELO CARRERA IZURIETA

DIRECTOR

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el producto resultante del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

JUAN PABLO RODRÍGUEZ ZUMÁRRAGA

IVÁN MARCELO CARRERA IZURIETA

DEDICATORIA

El presente trabajo está dedicado a todas las personas que me han acompañado a lo largo de mis años de formación como persona y como profesional, especialmente a:

Mi madre, Virginia, que siempre ha sido mi guía y mi ejemplo a seguir, que todas sus enseñanzas y regaños me han llevado a ser la persona que soy hoy día.

Mi padre, Mauricio, que con su apoyo y fortaleza me han permitido aprovechar de la mejor manera posible mis talentos y aptitudes.

Mi hermana, Daniela, que con su motivación y comprensión me han permitido salir adelante en muchas etapas de mi vida.

Mi primo, Juan Carlos, al que considero como un hermano siendo mi compañero de aventuras y un soporte emocional de mi vida.

Mis abuelos paternos, Ruperto y Ana, que con su amor y cariño me han permitido mantenerme fuerte en las adversidades.

Mi abuelo materno, Rafael, que con su paciencia y consejos han podido encaminar mejor mis decisiones.

Mi abuela materna, Magdalena, que en paz descanse, que con el poco tiempo a mi lado supo brindarme un ejemplo de gran amor sincero e incondicional.

Mis tíos, primos y familiares más cercanos, que han formado parte de las diferentes etapas de mi vida y con los que poseo un cariño especial.

Mis amigos más cercanos, quienes han sido parte fundamental de mi día a día y que sin ellos no habría logrado superar muchas de las adversidades que me ha dado la vida.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ECUACIONES

RESUMEN

Uno de los problemas más graves que afecta a la sociedad humana, es el miedo a las enfermedades y las complicaciones de salud que se derivan de estas. Por ello, el ser humano y la sociedad ha desarrollado lo que conocemos como un *fármaco* para combatir dichos temores. Así, el proceso de desarrollo e investigación de las posibles estructuras químicas, que pudieran ser usados de forma segura en el ser humano, ha sido un trabajo de investigación muy complejo, siendo los avances tecnológicos la base de la modernización de métodos de experimentación y análisis para mejorar las propiedades o principios activos de los fármacos. Así como los modelos computacionales como los sistemas de recomendación, que se encargan de brindar la recomendación de un ítem a un usuario con base en datos históricos de interacciones que las relacionan.

La principal motivación del presente trabajo surge en plantear al sistema de recomendación como un posible enfoque en la solución al reposicionamiento de fármacos, la búsqueda de nuevas interacciones para compuestos químicos ya establecidos. Por este motivo, el presente proyecto centra sus esfuerzos en demostrar que el problema del reposicionamiento de fármacos puede ser solucionado a nivel computacional a través de un modelo de recomendación, que toma su flujo de datos a partir de los datos históricos de ensayos clínicos realizados para compuestos y células de diversos orígenes.

La experimentación efectuada nos dio una base de resultados con precisión aceptables, que nos indican la viabilidad de la aplicación de los sistemas de recomendación como una alternativa a ser tomada de punto inicial en futuras investigaciones enfocadas en el reposicionamiento de fármacos.

PALABRAS CLAVE: Sistemas de recomendación, reposicionamiento de fármacos, compuestos, células, ensayos.

ABSTRACT

One of the most serious problems affecting human society is the fear of disease and the health complications that result from it. For this reason, human beings and society have developed what we know as a *drug* to fight against these fears. Thus, the processes of development and research of possible chemical structures, which could be used safely in humans, has been a very complex research work, given that technological advances have been the basis for the modernization of experimental and analytical methods to improve the properties or active ingredients of drugs. One example of this is the development of computational models, such as recommendation systems, which are responsible for providing the recommendation of an item to a user based on historical data of interactions that relate them.

The main challenge that this work aims to address is in making use of recommender systems as a possible approach in the solution to drug repositioning, the search for new interactions for already established chemical compounds. For this reason, this work focuses its efforts on demonstrating that the problem of drug repositioning can be solved at the computational level through a recommendation model, which takes its data flow from historical data of clinical trials performed for compounds and cells of various origins.

Carried out experiments gave us a basis of results with acceptable precision, which indicates the feasibility of the application of recommender systems as an alternative to be taken as a starting point in future research focused on drug repositioning.

KEYWORDS: Recommendation systems, repositioning of drugs, compounds, cells, assays.

1 DESCRIPCIÓN DEL COMPONENTE DESARROLLADO

Los sistemas de recomendación están basados en el concepto de utilizar diferentes fuentes de información para inferir intereses de usuarios sobre ítems. El principio básico de la recomendación es que deben existir dependencias significativas entre las actividades centradas en los usuarios y las actividades centradas en los ítems. [1]

Debido a la existencia de varios resultados obtenidos según el tipo de aproximación empleada en el sistema de recomendación, pueden llegar a presentarse varios escenarios de consenso o discordancia entre los datos predichos. Por ello, se presenta como una posible solución a la diferencia de resultados el planteamiento de un sistema híbrido que, basándose en el resultado de varios sistemas, pueda llegar a dar un resultado mucho más acertado y preciso [2] al problema abordado dentro del trabajo. Tomando especial consideración la alta sensibilidad de los datos trabajados dentro del proyecto.

Uno de los desafíos más importantes para afrontar en los sistemas de recomendación híbridos es la presentación de un resultado consensuado por los diferentes resultados de los demás modelos trabajados dentro del proyecto. Para ello, en el presente trabajo se hará uso de una idea presentada por Aggarwal, que nos indica que muchos de estos modelos son abordados como un sistema de clasificación común en el campo del aprendizaje de máquina [3]. Así, a partir de las predicciones obtenidas de los modelos de filtrado colaborativo y basado en conocimiento; se someterán a evaluación varios sistemas de clasificación en búsqueda de un modelo capaz de ajustarse de la manera más adecuada al tipo de datos trabajados.

El desarrollo del presente trabajo plantea la implementación de un modelo que tomará como fuente de información una base de datos de interacciones farmacológicas, que serán trabajadas a lo largo de varias fases de preparación para que puedan ser ajustadas correctamente a las necesidades del objetivo del proyecto. Finalmente, se busca la evaluación del desempeño de las predicciones realizadas por el modelo implementado.

Es necesario indicar que pueden existir otras estrategias que se pueden dar para abordar el problema planteado. Sin embargo, dado el alcance del presente trabajo, se optará por la utilización de librerías computacionales que ya cuentan con la implementación completa de sistemas de clasificación. Las mismas que están bajo actualización constante para mejorar su eficiencia y precisión en los resultados obtenidos.

1.1 Objetivo general

Implementar un modelo computacional de un sistema de recomendación híbrido capaz de realizar predicciones sobre interacciones entre fármacos conocidos y un conjunto de líneas celulares.

1.2 Objetivos específicos

- Relacionar información de varias fuentes de datos sobre fármacos y líneas celulares que permitan obtener una visión más amplia de la información a ser manejada dentro del proyecto.
- Definir una arquitectura eficiente y ajustable a los datos previamente preparados, para el sistema de recomendación que será desarrollado dentro del proyecto.
- Comparar los resultados obtenidos de los diferentes modelos desarrollados a través de métricas estandarizadas previamente establecidas dentro del proyecto.

1.3 Alcance

El presente trabajo busca evaluar la capacidad que tiene un Sistema de Recomendación Híbrido para predecir las interacciones farmacológicas. Pero para ello debemos tener claro que la información almacenada en las bases de datos con las que se trabajará es limitada.

De igual manera, se debe considerar diferentes versiones, para una evaluación vía utilidad. Por ejemplo, para la base de datos ChEMBL, se toma la versión 30, publicada en marzo de 2022, para el entrenamiento del sistema de recomendación, y el conjunto de los nuevos registros publicados en la versión 31, de agosto de 2022, para evaluación. Asegurando de esta manera que la predictividad evaluada sea comparada con datos reales.

1.4 Marco teórico

Reposicionamiento de Fármacos

El desarrollo de nuevos compuestos farmacológicos requiere de un proceso largo, riguroso y costoso para las empresas farmacéuticas dedicadas al descubrimiento de nuevos medicamentos que pueden ser utilizados para tratar las diferentes dolencias que como seres humanos podemos llegar a tener. Gracias a los avances tecnológicos gigantescos que se han llevado a cabo durante la última década, la metodología de investigación y desarrollo de nuevos compuestos ha sido revolucionada, agregando muchas más facilidades que aceleran parte del proceso.

Una de las técnicas empleadas actualmente para este proceso, es el procedimiento conocido como *reposicionamiento de fármacos*, entendiéndose este término como descubrir nuevas funcionalidades que se pueden dar a compuestos farmacológicos. [4]

Otro concepto clave para la utilización del reposicionamiento, es lo que coloquialmente se conoce como una llave maestra. Concepto que se traslada hacia la farmacología, donde el objetivo final del desarrollo de fármacos consiste en la búsqueda de un compuesto que pueda afectar a diversas células sin la necesidad de algún reactivo adicional. Dejando de lado la búsqueda de una solución mágica, debemos tener claro las diferentes etapas de evaluación que se vive dentro de la preparación de un nuevo fármaco para su exposición al mercado. Siendo estas fases las de pruebas in-silico, pruebas in-vitro, pruebas in-vivo y ensayos clínicos. Particularmente, el proceso de reposicionamiento nos permite acelerar la investigación y selección de moléculas que serán tomadas en cuenta para la primera fase de pruebas in-silico a través de simulaciones computacionales de sus interacciones con las células objetivo.

Sistemas de Recomendación

El concepto de un sistema de recomendación se lo puede tratar como directrices de sugerencias que permiten identificar elementos en común para diferentes individuos, y buscar aquellos elementos distintos que puedan llegar a ser compartidos por una agrupación más grande de sujetos. Actualmente, los sistemas de recomendación son partes fundamentales de las interacciones que los usuarios realizan día a día en las diferentes plataformas web y móviles del mercado actual. Esto principalmente ocurre dada la gran cantidad de información que se maneja en cada una de estas plataformas generada a partir de la interactividad de cada individuo. [5]

Uno de los escenarios de aplicación para sistemas de recomendación, son sitios de compras en línea como Amazon o eBay. En estos sistemas, se realizan sugerencias a los usuarios sobre productos que podrían ser de su interés, en base en las características y gustos de personas que han interactuado en su plataforma. Estos sistemas tienen el objetivo de obtener el mayor provecho de las sugerencias acertadas al aumentar la estadía y consumo de estos usuarios dentro de su plataforma.

De esta necesidad de aumentar la interactividad en aplicaciones del mundo tecnológico, nacen los modelos computacionales capaces de utilizar diferentes fuentes de datos que le permitan inferir posibles intereses, aportando nuevos elementos que sean relevantes y novedosos para el usuario en cuestión.

Tipos de Sistemas de Recomendación

Filtrado Colaborativo

Para entender el funcionamiento de este tipo de sistema de recomendación, debemos plantear una base común que será compartida en todos los modelos. Dicha base parte del hecho de que tenemos registros finitos de las interacciones positivas y negativas realizadas entre los usuarios y aquellos ítems que se espera recomendar.

A partir de estos registros podemos llegar a plantear una matriz de interacción entre usuarios e ítems, donde aquellos valores desconocidos pueden ser inferidos basándose en la correlación de los registros.

Por ejemplo, si tenemos que un usuario A comparte casi las mismas interacciones positivas que un usuario B, y uno de ellos posee una interacción desconocida para el otro, podemos recomendarle dicho ítem dada la alta probabilidad de recibir una interacción positiva de dicha recomendación. [5]

Basados en Contenido

Los sistemas basados en contenido, como su nombre lo indica, tienen una base enfocada en los ítems que serán utilizados dentro de las recomendaciones. Teniendo en cuenta que no conocemos el universo de interacciones entre usuarios e ítems, sino únicamente los registros de un usuario en específico.

Por este motivo, podemos tomar ventaja del conocimiento que poseemos de las características que distinguen y singularizan a los ítems dentro del universo. A partir de dichas propiedades, podemos buscar ítems con un alto parecido entre aquellos que poseían interacciones positivas con el usuario para que sean objetivos de las nuevas recomendaciones dentro del sistema. [5]

Basados en Conocimiento

Los sistemas basados en conocimiento funcionan de una manera contraria a los basados en contenido, donde nuestro principal enfoque no será en los ítems del universo de recomendación, sino en los usuarios que serán los objetivos.

El concepto de conocimiento surge debido a que podemos tomar elementos en común entre usuarios, como el contexto cultural y predisposición en ciertas tendencias, para poder realizar las respectivas recomendaciones de aquellos ítems que encajan entre perfiles similares. [5]

Sistema de Recomendación Híbrido

Para entender el funcionamiento de los sistemas híbridos debemos tomar en cuenta que los sistemas expuestos con anterioridad poseen sus ventajas y desventajas, dependiendo del escenario en el cual sean utilizados. Por ello, siguiendo el principio informático de "Divide y vencerás" podemos tomar todos estos modelos sencillos y combinarlos para crear un modelo mucho más robusto que permitirá dar una aproximación mucho más exacta. Partiendo de este concepto, los modelos híbridos únicamente son composiciones y ensamblajes de otros sistemas de recomendación.

Existen diversas formas para ensamblar estos modelos de recomendaciones que nos permitirán aprovechar las características de los sistemas que los componen:

- Diseño de ensamblaje, en el cual tomamos las salidas de los otros sistemas de recomendación y los empleamos para producir una nueva salida. Por lo general, este tipo de sistemas se basan en clusterización, clasificación y análisis de datos atípicos.
- Diseño monolítico, en el cual se integran las diferentes entradas de datos de los sistemas en uno solo. Esto puede variar dependiendo de los sistemas base que se utilizarán para crear el modelo híbrido.
- Diseño mixto, el cual se basa en la combinación de las anteriores opciones expuestas con la finalidad de dar más robustez a un modelo de recomendación híbrido. [3]

2 METODOLOGÍA

2.1 Base de Datos ChEMBL

Para el desarrollo del presente trabajo se tomó como eje principal a la base de datos ChEMBL, la cual será el centro de datos de todo el proyecto que se desarrollará a lo largo del presente escrito. Por este motivo, se realizará una breve descripción relevante que contiene este repositorio, además de la utilidad que prestará a nuestra problemática [5]. Para empezar, debemos entender que el repositorio de datos contiene varias secciones con información química fundamental para diferentes escenarios.

La primera de ellas, y de la cual recabaremos una porción de datos que nos será de utilidad para la definición de fármacos, corresponde a la sección de Compuestos. Un sector importante de datos corresponde a la información de SMILES (especificación de introducción lineal molecular simplificada) de las moléculas químicas registradas. Esta cadena de texto nos permitirá describir de manera computacional cada una de las estructuras de las moléculas, las cuales serán tomadas de base en procesos futuros de preparación de datos que desembocará en el entrenamiento de los sistemas de recomendación. Cabe aclarar que esta característica, no necesariamente se la debe considerar como un identificador único dentro de la base de datos, dado que puede haber cierta duplicidad entre los compuestos registrados. Por este motivo, ChEMBL utiliza un identificar único en el esquema que nos permitirá posteriormente relacionar diferentes características entre tablas de las moléculas, esta llave primaria será conocida como *molregno* (Molecule Register Number). [6]

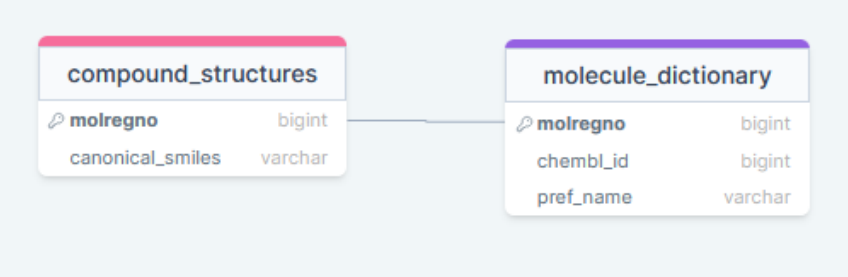


Figura 1:Abstracción de la sección de compuestos ChEMBL

La siguiente sección que será de gran importancia en el desarrollo del trabajo corresponde a la sección de *Targets* (Objetivos), que contiene la información más relevante acerca de las células en las cuales serán aplicadas los distintos fármacos descritos en la anterior sección de la base de datos. El sector en específico que será empleado para el trabajo corresponde al diccionario de células, donde se almacena información textual de utilidad para la definición de estos elementos. Para complementar los datos nos apoyaremos en

otra base de datos dedicada únicamente a estas células, conocida como *Cellosaurus*. [7] Dicho repositorio contiene información adicional que nos será de ayuda en la preparación de los datos, cabe aclarar que nuestros objetivos se centran en aquellas contenidas en el repositorio ChEMBL. Por ello, dentro de la definición de estos elementos tendremos almacenado un identificador que nos permitirá enlazar datos entre ambos repositorios de una manera mucho más sencilla durante el tratamiento de los datos computacionales. [6]

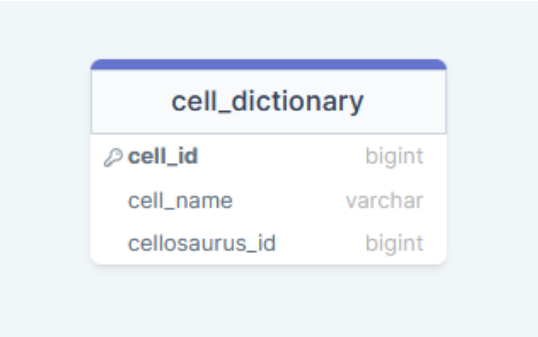


Figura 2: Abstracción de la sección de moléculas ChEMBL

Una vez que hemos definido los dos elementos principales que serán trabajados durante la creación de los modelos computacionales, debemos también recabar la información de la unión entre ellos. Por ello, nuestra última sección de interés en el repositorio corresponde a los *Datos Experimentales*. Esta zona posee dos grandes elementos de interés para el proyecto, aclarando que las dos están íntimamente relacionadas, la primera de ellas corresponde a los datos computacionales de ensayos de laboratorio que han sido registrados durante diversos trabajos. Adicionalmente, un ensayo en este contexto corresponde a las pruebas de laboratorio realizadas sobre una misma célula, y el comportamiento resultante de la interacción entre ambos elementos. Sin embargo, debido a que dentro de un mismo ensayo suelen existir varias pruebas efectuadas, el repositorio de datos optó por registrar esta información en un componente aparte, el cual en este contexto es llamado *Actividad*. Siendo estas actividades el segundo elemento de interés, donde están contenidas las circunstancias en las cuales fueron llevadas a cabo las pruebas donde tenemos datos del fármaco, cantidad empleada, unidad de medida, tipo de interacción, etc. [6]

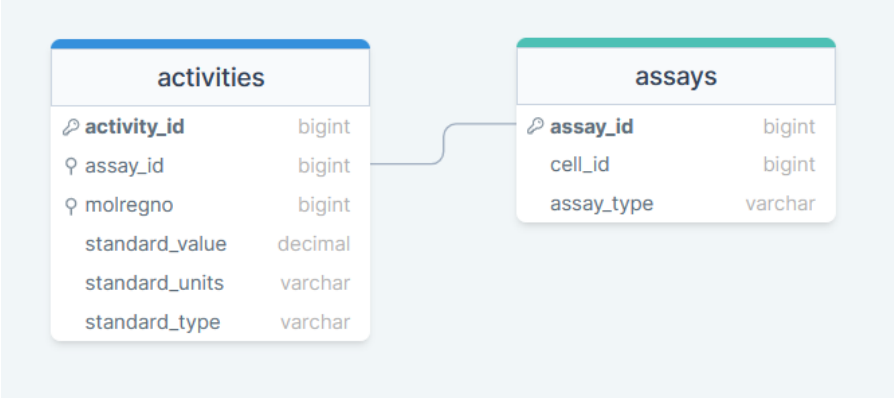


Figura 3: Abstracción de la sección de ensayos ChEMBL

Para finalizar, cada una de las secciones descritas anteriormente serán tratadas en diferentes etapas del proyecto según la necesidad del modelo a trabajar, de igual manera, únicamente nos centraremos en aquellas piezas de información de mayor relevancia para el modelo donde cierta información irrelevante en este contexto será omitida durante la preparación de los datos.

2.2 Preparación de Datos

Filtrado Colaborativo

Una vez hemos definido la fuente de la información de la cual tomaremos como base para el desarrollo del presente trabajo, debemos empezar con la preparación de los datos que serán utilizados dentro del primer sistema de recomendación.

De manera específica, nos vamos a centrar en los datos experimentales recolectados en el banco de datos que corresponden a los ensayos de laboratorio realizados a lo largo de diferentes trabajos científicos. Dicho en otras palabras, vamos a manejar los datos que relacionan el uso de fármacos en moléculas a través de la cantidad empleada y los resultados obtenidos de las interacciones entre ambos componentes.

Para efectuar la traducción de este procesamiento a un nivel de código, nos apoyaremos en un ambiente de ejecución del lenguaje Python que nos permite proceder a través de una división de la ejecución de porciones específicas de código, con la finalidad de tener un control más amplio en los resultados obtenidos de las instrucciones planteadas.

Con este planteamiento previo, procederemos a realizar una división de 3 fases del procesamiento de la información de los ensayos obtenida de la base de datos ChEMBL:

Extracción y estandarización de valores

Como antecedente a esta fase, debemos detallar una aclaración respecto a la fuente de datos. Dado que, este tipo de información se maneja de diversas formas en el procesamiento de datos y planteamiento de modelos de inteligencia artificial, todos los datos se encuentran contenidos en un archivo SQLite, herramienta de base de datos ligera y de fácil portabilidad.

Por este motivo, nuestra primera acción será la obtención de la información de los datos a través de una conexión a la base de datos, la misma que será almacenada en una estructura de datos conocida como *DataFrame*, que nos facilitará su tratamiento y procesamiento durante lo largo de los diferentes procesos efectuados.

```
df = pd.read_sql_query("SELECT ASY.ASSAY_ID, ACT.ACTIVITY_ID, ACT.MOLREGNO, \
ASY.CELL_ID, ACT.STANDARD_VALUE,ACT.STANDARD_UNITS, ACT.STANDARD_TYPE \
FROM ACTIVITIES ACT INNER JOIN ASSAYS ASY ON ACT.ASSAY_ID = ASY.ASSAY_ID \
WHERE ASY.CELL_ID IS NOT NULL \
AND standard_type IN ('IC50','GI50','EC50', 'CC50');", conn)
```

```
df
```

	assay_id	activity_id	molregno	cell_id	standard_value	standard_units	standard_type
0	42997	31881	295513	635	1000000.0	nM	CC50
1	42997	33039	107119	635	100000.0	nM	CC50
2	102567	33288	84235	522	50000.0	nM	CC50
3	102567	33291	84807	522	100000.0	nM	CC50
4	76931	33405	62917	308	50000.0	nM	CC50
...	...	...	...	...	...	...	...
3877799	2144911	23681327	2609875	726	25000.0	nM	IC50
3877800	2144911	23681328	2647402	726	20800.0	nM	IC50
3877801	2144911	23681329	2695826	726	25000.0	nM	IC50
3877802	2144911	23681330	2582982	726	25000.0	nM	IC50
3877803	2144911	23681331	2685920	726	7350.0	nM	IC50

3877804 rows × 7 columns

Figura 4: Extracción de ensayos de ChEMBL

Detallando más acerca de la consulta SQL realizada, debemos precisar que únicamente nos concentraremos en actividades específicas realizadas en los ensayos de los siguientes tipos:

- CC50 - Concentración de citotóxica del 50% en células estacionarias y en división de múltiples tipos de células y tejidos humanos relevantes para determinar el potencial de toxicidad específica del ciclo celular, de la especie o del tejido.
- IC50 - Concentración de fármaco requerida para una inhibición del 50%, adicionalmente permite medir la concentración de antígeno que bloque el 50% de la unión del anticuerpo-fago al antígeno inmovilizado.

- GI50 - Concentración que provoca una inhibición del crecimiento celular del 50%.
- EC50 - Concentración para poder producir un 50% de la respuesta máxima que permite comparar la potencia de los fármacos.

A continuación, debido a que tenemos una gran variedad de unidades de medición que fueron utilizadas durante las actividades registradas de cada ensayo de la base de datos, debemos realizar una estandarización de estas. Para ello, nuestra primera acción sobre estos datos es obtener todos los valores únicos de las unidades de nuestra estructura de datos.

```
df.standard_units.unique()
```

```
array(['nM', 'ug.mL-1', None, 'ucm', '%', "10'7uM", "10'10uM",
       "10'3ug/ml", 'mol', 'ppm', '10^3 uM', '10^4 uM', 'Ke nM-1', 'pnM',
       'ng/g', 'mg kg-1', 'pH', "10'8pM", "10'7pM", "10'6pM", "10'5pM",
       '10^2 uM', 'µM', 'PuM', '10^4M', "10'-8cm/s", 'ug', 'ppm g dm**-3',
       'p.p.m.', 'M ml-1', 'ug/g', 'ml', 'molar ratio', 'uL', '/uM',
       "10'-13 ug/ml", '/uM/s', 'mg.min/m3', 'umol/dm3', "10'-11uM",
       "10'-8mg/ml", "10'-4nM", "10'6uM", "10'5uM", '10^-5 uM'],
      dtype=object)
```

Figura 5: Extracción de unidades de ensayos

No obstante, es necesario clarificar que muchas de las unidades obtenidas no tienen una relación directa con la unidad estándar de medición µM (Micromoles), siendo este el motivo por el cual descartaremos algunas sin aplicar algún tipo de tratamiento a sus datos.

```
deleted_units = [None,'ucm', '%', 'ppm', 'Ke nM-1', 'pnM', 'ng/g', 'mg kg-1', 'pH', 'PuM',
"10'-8cm/s", 'ug', "ppm g dm**-3", 'p.p.m.', 'M ml-1', 'ug/g', 'ml', 'molar ratio', 'uL', '/uM/
s', 'mg.min/m3']
```

Figura 6: Unidades a ser eliminadas de los ensayos

```
df_filtered.standard_units.unique()
```

```
array(['nM', 'ug.mL-1', "10'7uM", "10'10uM", "10'3ug/ml", 'mol',
       '10^3 uM', '10^4 uM', "10'8pM", "10'7pM", "10'6pM", "10'5pM",
       '10^2 uM', 'µM', '10^4M', '/uM', "10'-13 ug/ml", 'umol/dm3',
       "10'-11uM", "10'-8mg/ml", "10'-4nM", "10'6uM", "10'5uM",
       '10^-5 uM'], dtype=object)
```

Figura 7: Unidades finales para los ensayos

Paralelamente, descartamos aquellas tuplas de información que no contienen valores numéricos en el registro de sus ensayos.

```
df_filtered = df_filtered[df_filtered.standard_value.notna()]
df_filtered
```

	assay_id	activity_id	molregno	cell_id	standard_value	standard_units	standard_type
0	42997	31881	295513	635	1000000.0	nM	CC50
1	42997	33039	107119	635	100000.0	nM	CC50
2	102567	33288	84235	522	50000.0	nM	CC50
3	102567	33291	84807	522	100000.0	nM	CC50
4	76931	33405	62917	308	50000.0	nM	CC50
...	...	...	...	...	...	...	...
3877799	2144911	23681327	2609875	726	25000.0	nM	IC50
3877800	2144911	23681328	2647402	726	20800.0	nM	IC50
3877801	2144911	23681329	2695826	726	25000.0	nM	IC50
3877802	2144911	23681330	2582982	726	25000.0	nM	IC50
3877803	2144911	23681331	2685920	726	7350.0	nM	IC50

Figura 8: Ensayos en bruto

Después empezaremos a tratar una por una las unidades restantes hasta que finalicemos con el proceso de estandarización. Para ello, utilizaremos la siguiente tabla de relación entre unidades:

Tabla 1: Factores de conversión para unidades de ensayos

Unidad estándar	Factor de conversión	Unidad a convertir	Valor unitario
µM	0.001	nM	1
µM	$\frac{1000}{301}$	$\frac{\mu g}{mL}$	1
µM	10000000	$10^{7}\mu M$	1
µM	10000000000	$10^{10}\mu M$	1
µM	$\frac{1000000}{301}$	$10^{3}\frac{\mu g}{mL}$	1
µM	0.000001	mol	1
µM	1000	$10^{3}\mu M$	1
µM	10000	$10^{4}\mu M$	1
µM	100	$10^{8}\rho M$	1
µM	10	$10^{7}\rho M$	1
µM	1	$10^{6}\rho M$	1
µM	0.1	$10^{5}\rho M$	1
µM	100	$10^{2}\mu M$	1
µM	10000000000	$10^{4}M$	1
µM	$\frac{1}{3010000000000}$	$10^{-13}\frac{\mu g}{mL}$	1
µM	1	$\frac{\mu mol}{dm^{3}}$	1
µM	0.00000000001	$10^{-11}\mu M$	1
µM	$\frac{100}{301}$	$10^{-8}\frac{mg}{mL}$	1
µM	0.1	$10^{-4}nM$	1

µM	1000000	$10^6 \mu M$	1
µM	100000	$10^5 \mu M$	1
µM	0.00001	$10^{-5} \mu M$	1

Análisis de las interacciones de los ensayos

Una vez hemos finalizado la fase de estandarización de los datos, procedemos a realizar la evaluación de los valores obtenidos para determinar si la interacción realizada es positiva o negativa con base en un umbral definido.

En este caso en concreto el umbral definido es $10 \mu M$, dicho de una manera más detallada, debemos verificar si la cantidad empleada del fármaco en la molécula corresponde a un valor viable sin requerir de una gran cantidad del compuesto que podría llegar a ser tóxica para la muestra utilizada.

```
def process_interaction(row):
    value = float(row['standard_value'])

    if (value) > threshold:
        interaction = -1
    else:
        interaction = 1

    return interaction
```

Figura 9: Algoritmo para análisis de interacción de un ensayo

```
standard_df
```

	assay_id	activity_id	molregno	cell_id	standard_value	standard_units	standard_type	interaction
0	42997	31881	295513	635	1000.00	uM	CC50	-1
1	42997	33039	107119	635	100.00	uM	CC50	-1
2	102567	33288	84235	522	50.00	uM	CC50	-1
3	102567	33291	84807	522	100.00	uM	CC50	-1
4	76931	33405	62917	308	50.00	uM	CC50	-1
...	...	...	...	...	...	...	...	...
3758482	2144911	23681327	2609875	726	25.00	uM	IC50	-1
3758483	2144911	23681328	2647402	726	20.80	uM	IC50	-1
3758484	2144911	23681329	2695826	726	25.00	uM	IC50	-1
3758485	2144911	23681330	2582982	726	25.00	uM	IC50	-1
3758486	2144911	23681331	2685920	726	7.35	uM	IC50	1

3758487 rows × 8 columns

Figura 10: Ensayos con su respectiva interacción

Sumarización de interacciones

Durante la última fase de la preparación de los datos, debemos puntualizar que existen interacciones repetidas entre fármacos y moléculas de diferentes ensayos que fueron registrados en la base de datos. Por este motivo, debemos hacer un compendio de

información para este tipo de casos y eliminar datos que pueden causar ruido en futuros procesamientos.

Primero, realizaremos una separación entre aquellos registros únicos y aquellos que se encuentran repetidos con diferentes valores en su medición y en su definición de interacción.

```
duplicate_assays = interactions_df[interactions_df.duplicated(['molregno', 'cell_id'], keep=False)]
no_duplicate_assays = interactions_df[~interactions_df.duplicated(['molregno', 'cell_id'], keep=False)]
```

Figura 11: División de ensayos duplicados y no duplicados

```
duplicate_assays.sort_values(['molregno', 'cell_id'], ascending=[True, True], inplace=True, ignore_index=True)
duplicate_assays
```

Figura 12: Ordenamiento de los ensayos duplicados

	assay_id	activity_id	molregno	cell_id	standard_value	standard_units	standard_type	interaction
0	840576	11017315	23	330	13.00000	uM	IC50	-1
1	840575	11017325	23	330	5.00000	uM	IC50	1
2	840574	11017335	23	330	9.00000	uM	IC50	1
3	840577	11017345	23	555	34.00000	uM	IC50	-1
4	840578	11017355	23	555	26.00000	uM	IC50	-1
...	...	...	...	...	...	...	...	...
498331	2144908	23680599	2712302	5555	1.32000	uM	IC50	1
498332	1998402	22082800	2712584	5673	0.52500	uM	IC50	1
498333	1998402	22082801	2712584	5673	0.52481	uM	IC50	1
498334	1998403	22082802	2712584	5673	100.00000	uM	IC50	-1
498335	1998403	22082803	2712584	5673	100.00000	uM	IC50	-1

498336 rows × 8 columns

Figura 13: Ensayos duplicados

Después comprobaremos si existe una discrepancia en los valores de interacciones obtenidos en la fase anterior, dicho en otras palabras, si es que todas las interacciones son positivas o negativas, caso contrario, desconocemos el estado de la interacción debido a que no hay un consenso en sus registros.

```python
def summarize_interactions(df):
    summarize_df = pd.DataFrame({
        'molregno': pd.Series(dtype='str'),
        'cell_id': pd.Series(dtype='str'),
        'interaction': pd.Series(dtype='int')
    })

    while True:
        if (df.empty): break

        molregno = df.iloc[0].molregno
        cell_id = df.iloc[0].cell_id

        aux_df = df[(df['molregno'] == molregno) & (df['cell_id'] == cell_id)]
        flag = len(aux_df.interaction.unique()) == 1

        if(flag):
            summarize_df.loc[len(summarize_df)] = [molregno, cell_id, df.iloc[0].interaction] #
            type: ignore
        else:
            summarize_df.loc[len(summarize_df)] = [molregno, cell_id, 0] # type: ignore

        df = df[~df.isin(aux_df)].dropna()

    return summarize_df
```

Figura 14: Algoritmo de sumarización de ensayos duplicados

	molregno	cell_id	interaction
0	1633228.0	697.0	0.0
1	1633240.0	449.0	1.0
2	1633242.0	502.0	-1.0
3	1633249.0	449.0	1.0
4	1633268.0	163.0	-1.0
...	...	...	...
3758360	2709669.0	726.0	1.0
3758361	2638089.0	726.0	1.0
3758362	2648910.0	726.0	1.0
3758363	2624206.0	726.0	1.0
3758364	2580311.0	726.0	1.0

3439989 rows × 3 columns

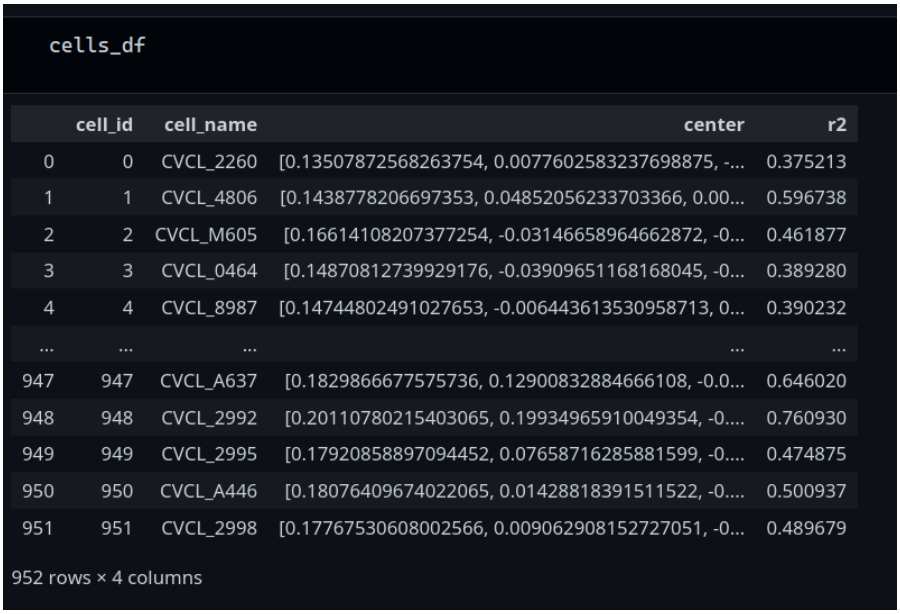
Figura 15: Ensayos sumarizados

Para finalizar, debemos realizar este proceso para las dos versiones de la base de datos ChEMBL que serán utilizadas en el desarrollo de los modelos de sistemas de recomendación.

Sistema de Recomendación Basado en Conocimiento

Paralelamente, al modelo de Filtrado Colaborativo que fue planteado en la anterior sección, se llevó a cabo el desarrollo de un modelo basado en conocimiento. La particularidad de este modelo es que nos centraremos en los ítems de la recomendación, en este caso en particular las células, más no en todo el conjunto de ensayos que describen las interacciones entre fármacos y células.

Por este motivo, nuestro primer paso a dar en el planteamiento del modelo corresponde a definir una forma de relacionar cada una de las células, de tal manera, que podamos establecer similitud entre los elementos que componen nuestro conjunto de datos. Con este propósito, haremos uso de un banco de datos que contiene una representación de la información de cada célula como una hiperesfera en un hiperespacio, añadiendo datos como la ubicación de su centro y radio en esta abstracción.



cells_df

	cell_id	cell_name	center	r2
0	0	CVCL_2260	[0.13507872568263754, 0.0077602583237698875, -...	0.375213
1	1	CVCL_4806	[0.1438778206697353, 0.04852056233703366, 0.00...	0.596738
2	2	CVCL_M605	[0.16614108207377254, -0.03146658964662872, -0...	0.461877
3	3	CVCL_0464	[0.14870812739929176, -0.03909651168168045, -0...	0.389280
4	4	CVCL_8987	[0.14744802491027653, -0.006443613530958713, 0...	0.390232
...	...	...	...	...
947	947	CVCL_A637	[0.1829866677575736, 0.12900832884666108, -0.0...	0.646020
948	948	CVCL_2992	[0.20110780215403065, 0.19934965910049354, -0....	0.760930
949	949	CVCL_2995	[0.17920858897094452, 0.07658716285881599, -0....	0.474875
950	950	CVCL_A446	[0.18076409674022065, 0.01428818391511522, -0....	0.500937
951	951	CVCL_2998	[0.17767530608002566, 0.009062908152727051, -0...	0.489679

952 rows × 4 columns

Figura 16: Dataset auxiliar con información de las células

De esta manera, se empleó el concepto geométrico de la intersección que consiste en la suma de los radios de las hiperesferas menos la distancia euclidiana dividida para la suma de los radios, como se muestra en la Ecuación 1.

$$interseccion = \frac{r1 + r2 - d}{r1 + r2}$$

Ecuación 1: Ecuación de Intersección

Es importante destacar que esta métrica nos permitirá calcular un valor numérico que será utilizado para obtener un listado de elementos más similares por cada una de las células en el conjunto de datos auxiliar. Asimismo, nos ofrece la posibilidad de utilizar estos valores en caso de que se necesite trabajar en futuros proyectos con las representaciones hiperespaciales de los objetos.

Continuando con el tratamiento de los datos requeridos para nuestro modelo, se aplicó la fórmula descrita anteriormente para cada uno de los ítems, obteniendo así un listado de células similares con base en la métrica propuesta.

cells_similarities_df

	cell_id	cell_name	proximate_elements
0	476	CVCL_FA09	{"889": 0.8698316673031014, "181": 0.869779466...
1	477	CVCL_D607	{"79": 0.8304440078494713, "479": 0.8298717591...
2	478	CVCL_1987	{"483": 0.9197208269967997, "569": 0.917173023...
3	479	CVCL_0218	{"569": 0.9447684634169393, "482": 0.941758805...
4	480	CVCL_1988	{"478": 0.8401073607908522, "490": 0.817789937...
...	...	...	...
947	471	CVCL_L294	{"81": 0.757882214552359, "894": 0.75760893798...
948	472	CVCL_6888	{"473": 0.781166660080495, "847": 0.7569554541...
949	473	CVCL_6889	{"472": 0.781166660080495, "847": 0.7755910769...
950	474	CVCL_D102	{"478": 0.8536340730044826, "569": 0.848229135...
951	475	CVCL_6892	{"92": 0.770451492661547, "90": 0.735991768200...

952 rows × 3 columns

Figura 17: Datos de las células con sus ítems más similares

Adicionalmente, nos apoyaremos en otro conjunto de datos que contiene la información de los fármacos y sus elementos con mayor similitud, que nos serán útiles en el algoritmo del sistema de recomendación al poseer más elementos que se relacionan con los ítems de los ensayos.

similarity_merge

	molregno	chembl_id	top3
0	97	CHEMBL2	[97, 312495, 2507043, 2511460]
1	115	CHEMBL3	[115, 835256]
2	146	CHEMBL4	[146, 210, 1795, 506559]
3	147	CHEMBL5	[147, 48536]
4	148	CHEMBL6246	[148]
...	...	...	...
38911	2537371	CHEMBL4802120	[2537371]
38912	2537372	CHEMBL4802121	[70140, 1076140, 2537372]
38913	2537374	CHEMBL4802123	[2537374]
38914	2537375	CHEMBL4802124	[4398, 302932, 1163762, 2537375]
38915	2537376	CHEMBL4802125	[1275851, 2505521, 2537376]

38916 rows × 3 columns

Figura 18: Datos de los fármacos con sus ítems más similares

Por último, se debe aclarar que los datos procesados en el anterior modelo correspondiente a los ensayos de la base de datos ChEMBL son requeridos de igual forma para el planteamiento del modelo basado en conocimiento.

Modelo Híbrido

Teniendo en cuenta el hecho de que el sistema basado en conocimiento no es capaz de realizar predicciones adecuadas dada la naturaleza de nuestra base de datos, se optó por un enfoque diferente en la preparación de los datos de entrada para el modelo híbrido del sistema de recomendación.

Con este planteamiento previo, procederemos a efectuar una división de 3 fases del preprocesamiento de la data requerida en el último sistema de recomendación propuesto.

Generación de nuevos ensayos

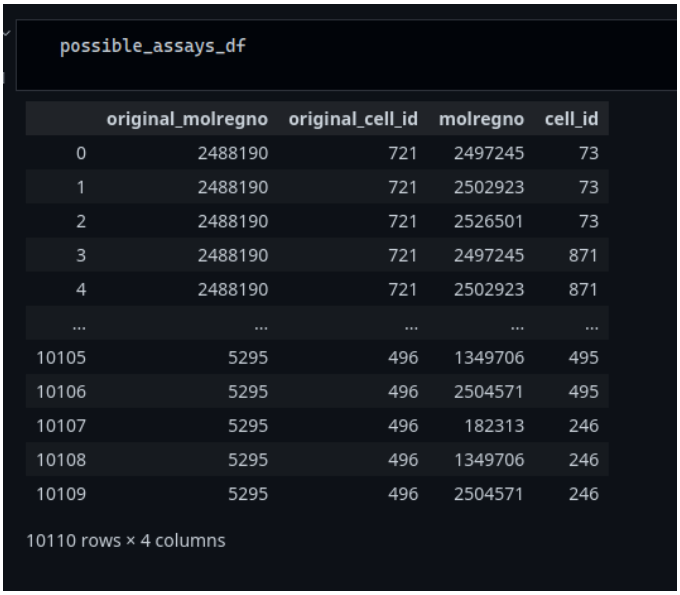
Para esta fase, hicimos uso de parte de la lógica del algoritmo basado en conocimiento donde generaremos un nuevo grupo de ensayos que serán formados a partir de las combinaciones de los fármacos y células con mayor similitud a los datos del ensayo de prueba original.

```python
def generate_possible_assays(assay):
    cell_top_elements_row = cell_similarities.loc[cell_similarities['cell_id'] == assay.cell_id]
    drugs_top_elements_row = drugs_similarities.loc[drugs_similarities['molregno'] == assay.
    molregno]

    if cell_top_elements_row.empty:
        return None

    if drugs_top_elements_row.empty:
        return None

    cells_top_elements = json.loads(
        cell_top_elements_row.proximate_elements.to_list()[0])

    if len(cells_top_elements) < 10:
        return None

    drugs_top_elements = drugs_top_elements_row.top3.to_list()[0]

    if len(drugs_top_elements) < 4:
        return None

    new_possible_assays = pd.DataFrame(
        columns=['original_molregno', 'original_cell_id', 'molregno', 'cell_id'])

    for cell in cells_top_elements.keys():
        for drug in drugs_top_elements:
            if drug == assay.molregno:
                continue

            possible_assay = pd.DataFrame(
                [{'original_molregno': assay.molregno, 'original_cell_id': assay.cell_id,
                'molregno': drug, 'cell_id': cell}])

            new_possible_assays = pd.concat(
                [new_possible_assays, possible_assay], axis=0, ignore_index=True)

            del possible_assay

    return new_possible_assays
```

Python

Figura 19: Algoritmo de generación de nuevos ensayos

Obteniendo como resultado un conjunto de nuevos ensayos que serán utilizados posteriormente como base los datos del modelo híbrido.



possible_assays_df

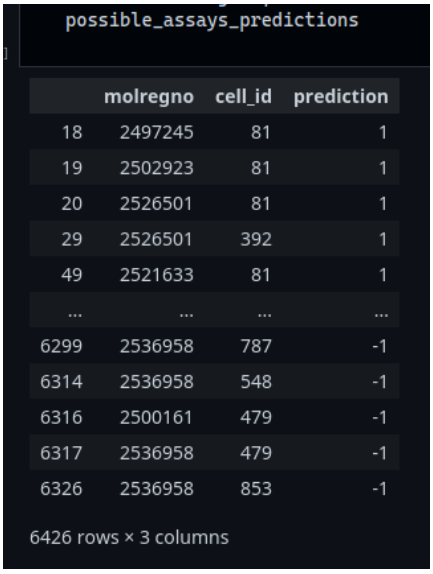
	original_molregno	original_cell_id	molregno	cell_id
0	2488190	721	2497245	73
1	2488190	721	2502923	73
2	2488190	721	2526501	73
3	2488190	721	2497245	871
4	2488190	721	2502923	871
...	...	...	...	...
10105	5295	496	1349706	495
10106	5295	496	2504571	495
10107	5295	496	182313	246
10108	5295	496	1349706	246
10109	5295	496	2504571	246

10110 rows × 4 columns

Figura 20: Dataset de nuevos ensayos

Prediciendo las interacciones del nuevo grupo de ensayos

Tomando como base este nuevo grupo de ensayos, nos apoyamos en el modelo de filtrado colaborativo obtenido en una fase anterior, con el cual realizaremos la predicción de las posibles interacciones que tendrán estos nuevos pares fármaco-célula.



possible_assays_predictions

	molregno	cell_id	prediction
18	2497245	81	1
19	2502923	81	1
20	2526501	81	1
29	2526501	392	1
49	2521633	81	1
...	...	...	...
6299	2536958	787	-1
6314	2536958	548	-1
6316	2500161	479	-1
6317	2536958	479	-1
6326	2536958	853	-1

6426 rows × 3 columns

Figura 21: Predicción de interacciones para los nuevos ensayos

Entre paréntesis, debemos esclarecer que se empleó el análisis de umbrales y valores discretos planteados en la sección correspondiente al filtrado colaborativo.

Uniendo los datos para el modelo híbrido

Con los datos de los nuevos ensayos, además de la posible interacción que poseen, podemos plantear la estructura que tendrán los datos para el modelo híbrido.

Como se mencionó anteriormente, estos nuevos ensayos están estrechamente relacionados con cada ensayo de prueba definido, dado que partieron de la combinación de los fármacos y células similares a los originales. Por ello, para proceder con la estructuración de este nuevo dataset, se buscaron las nuevas predicciones con base en el par fármaco-célula original.

```
linked_df = linked_data()
linked_df
```

	molregno	cell_id	interaction	p_0	p_1	p_2	p_3	p_4	p_5	p_6	...	p_21	p_22	p_23	p_24	p_25	p_26	p_27	p_28	p_29	p_30
0	1732984	741	1	1	1	1	1	1	1	1	...	1	1	-1	-1	1	1	1	1	-1	-1
1	1732985	741	1	1	1	1	1	1	1	1	...	1	1	1	1	1	-1	1	-1	1	-1
2	2211237	449	-1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
3	2406188	763	1	1	1	0	1	1	1	1	...	1	1	1	1	1	-1	1	1	-1	1
4	336913	434	0	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	-1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
332	712090	308	1	-1	1	1	1	-1	-1	-1	...	1	0	1	1	-1	-1	1	-1	-1	-1
333	14440	489	1	1	0	0	0	1	1	1	...	0	-1	-1	-1	1	1	1	1	1	1
334	8062	661	1	1	1	1	1	-1	1	-1	...	-1	-1	1	-1	-1	1	-1	-1	1	1
335	14440	661	1	1	1	1	1	1	1	1	...	1	0	0	0	0	0	0	1	1	1
336	14440	440	1	1	1	1	1	-1	-1	-1	...	1	0	0	0	1	1	1	1	1	1

337 rows × 34 columns

Figura 22: Conjunto de datos para el modelo híbrido

En síntesis, se tomaron estos nuevos datos y se juntaron con el valor de la predicción efectuada por el modelo de filtrado colaborativo para generar un dataset de 31 dimensiones que contienen todas las posibles interacciones para el conjunto de ensayos de prueba definido.

2.3 Filtrado Colaborativo

Para empezar con el desarrollo del sistema de recomendación, debemos usar los dos registros de datos obtenidos en la anterior sección del trabajo. Después de ello, efectuaremos una división de los datos que no constan en ambas estructuras de datos para que podamos utilizarlos como datos de prueba para el modelo obtenido.

```
difference_between_versions = new_version_df[~new_version_df.apply(tuple, 1).isin(old_version_df.
apply(tuple, 1))] # type: ignore
difference_between_versions
```

	molregno	cell_id	interaction
1415	1732984.0	741.0	1.0
1420	1732985.0	741.0	1.0
3378	1761966.0	5555.0	0.0
3631	1763584.0	627.0	1.0
3695	1779457.0	491.0	-1.0
...	...	...	...
3439984	2709669.0	726.0	1.0
3439985	2638089.0	726.0	1.0
3439986	2648910.0	726.0	1.0
3439987	2624206.0	726.0	1.0
3439988	2580311.0	726.0	1.0

24022 rows × 3 columns

Figura 23: Obtención del conjunto de ensayos de prueba

Nos apoyaremos en una librería externa del lenguaje Python que incorpora la lógica necesaria para entrenar un sistema de recomendación de filtrado colaborativo, conocida como *Surprise*. Gracias a este soporte externo, podemos concentrarnos en el entrenamiento del modelo a través de los parámetros establecidos, sin la necesidad de empezar de 0 en la lógica detrás del mismo. De igual manera, detallaremos a continuación los parámetros utilizados para el desarrollo del modelo computacional:

- rating_scale: Escala de valores establecidos para las interacciones del modelo. En este caso se empleó una escala de -1 a 1, siendo el valor que representa la interacción negativa y positiva respectivamente.
- n_factors: Número de elementos resultantes de la descomposición de los valores singulares, nos proporcionan información oculta de las variables originales, se emplean para inferir similitudes. En este caso, emplearemos un número de factores igual a 10.
- n_epochs: Número de épocas (iteraciones) que serán utilizadas en el entrenamiento del modelo. En este caso, se empleó 300 épocas para el entrenamiento del modelo.
- lr_all: Porcentaje de aprendizaje de los elementos a través de las épocas de entrenamiento del modelo. En este caso, se usó un valor de 0.002 durante el entrenamiento del modelo.
- reg_all: Término de regularización de los parámetros a través de las épocas de entrenamiento. En este escenario, se empleó un valor de 0.1.
- biased: Sesgo definido para el entrenamiento del modelo. Por defecto se emplea un valor de verdadero.

```
reader = Reader(rating_scale=(-1, 1))
```

Figura 24: Definición de la escala para las predicciones

```
param_grid = {
    'n_factors': [50],
    'n_epochs': [300],
    'lr_all': [0.002],
    'reg_all': [0.1],
    'biased' : [True]
} # Best values on training
```

Figura 25: Matriz de parámetros de entrenamiento para el modelo

El método de entrenamiento escogido para el modelo es el de Descomposición de Valores Singulares SVD, que corresponde a una técnica de factorización de matrices que nos permite descomponer una matriz A en una multiplicación de varias matrices más pequeñas.

Asimismo, con la ayuda de la librería *Surprise* realizamos el entrenamiento del modelo para nuestro conjunto de datos, posterior a ello, analizamos la precisión obtenida a través de la métrica del error medio cuadrático (RSME) resultando en un valor del 0,5093. Sin embargo, debido a la naturaleza propia de los datos donde la mayor parte de interacciones fármaco - célula son desconocidas, no se pudo obtener un acierto mucho mayor para los modelos entrenados.

```
trainset = data.build_full_trainset()
algo.fit(trainset)

<surprise.prediction_algorithms.matrix_factorization.SVD at 0x1a7870ac990>

predictions = algo.test(trainset.build_testset())

accuracy.rmse(predictions)

RMSE: 0.5093

0.5093020808562567
```

Figura 26: Entrenamiento del filtrado colaborativo

De cualquier manera, con esta versión entrenada del modelo de filtrado colaborativo procedimos a realizar las predicciones de los datos de entrenamiento, donde se puede evidenciar el funcionamiento de este primer sistema de recomendación.

```
test_set = data_diff.construct_testset(data_diff.raw_ratings)
```

```
predictions = collaborative_model.test(testset=test_set)
```

```
accuracy.rmse(predictions)
```

```
RMSE: 0.9874

0.98743034145976 35
```

Figura 27: Evaluación del modelo para los ensayos de prueba

```
predictions_df = pd.DataFrame(predictions)
predictions_df
```

	uid	iid	r_ui	est	details
0	1732984.0	741.0	1.0	0.196664	{'was_impossible': False}
1	1732985.0	741.0	1.0	0.933243	{'was_impossible': False}
2	1761966.0	5555.0	0.0	-0.373830	{'was_impossible': False}
3	1763584.0	627.0	1.0	1.000000	{'was_impossible': False}
4	1779457.0	491.0	-1.0	-0.015621	{'was_impossible': False}
...	...	...	...	...	...
24017	2709669.0	726.0	1.0	0.042628	{'was_impossible': False}
24018	2638089.0	726.0	1.0	0.042628	{'was_impossible': False}
24019	2648910.0	726.0	1.0	0.042628	{'was_impossible': False}
24020	2624206.0	726.0	1.0	0.042628	{'was_impossible': False}
24021	2580311.0	726.0	1.0	0.042628	{'was_impossible': False}

24022 rows × 5 columns

Figura 28: Obtención de las predicciones para los ensayos de prueba

Presentación de Resultados

Los resultados obtenidos de las predicciones efectuadas por el sistema de recomendación, dichos datos son representados como valores continuos en una escala de -1 a 1 sin corresponder correctamente a nuestra definición de interacción positiva, negativa o desconocida a través de una escala discreta de 1, -1 y 0 respectivamente. Por este motivo, debemos tratar esta información para que se adapte a la interpretación que estamos buscando en el contexto de nuestro proyecto.

Para realizar este proceso debemos definir un umbral el cual se encargará de establecer si el valor obtenido corresponde a una de las categorías de las interacciones, por esta razón, la primera acción a realizar es la búsqueda de los umbrales que nos permitirán discretizar los valores de manera óptima. De igual forma, utilizaremos un concepto clave en análisis de datos, que corresponde a la matriz de confusión como medio de evaluación de la precisión del umbral utilizado en las interacciones predichas.

```python
def calculate_confusion_matrix():
    confusion_matrices = pd.DataFrame(
        columns=['possitive_threshold', 'negative_threshold', 'true_values']
    )

    for possitive_threshold in np.linspace(0.01, 1, num=100, endpoint=False)
    :
        possible_positive = predictions_threshold(
            round(possitive_threshold, 2), True
        )

        for negative_threshold in np.linspace(-0.99, -0.01, num=99):
            possible_negative = predictions_threshold(
                round(negative_threshold, 2), False
            )

            possible_values = pd.concat([possible_positive,
            possible_negative])

            temp_conf_matrix = confusion_matrix(
                possible_values['r_ui'], possible_values['est_threshold']
            )

            true_values = sum(np.diagonal(temp_conf_matrix))

            aux_df = pd.DataFrame([{'possitive_threshold': round
            (possitive_threshold, 2),
                                    'negative_threshold': round
                                    (negative_threshold, 2),
                                    'true_values': true_values}])

            confusion_matrices = pd.concat([confusion_matrices, aux_df],
            axis=0, ignore_index=True)

            del possible_negative
            del temp_conf_matrix
            del true_values
            del aux_df

        del possible_positive

    return confusion_matrices
```

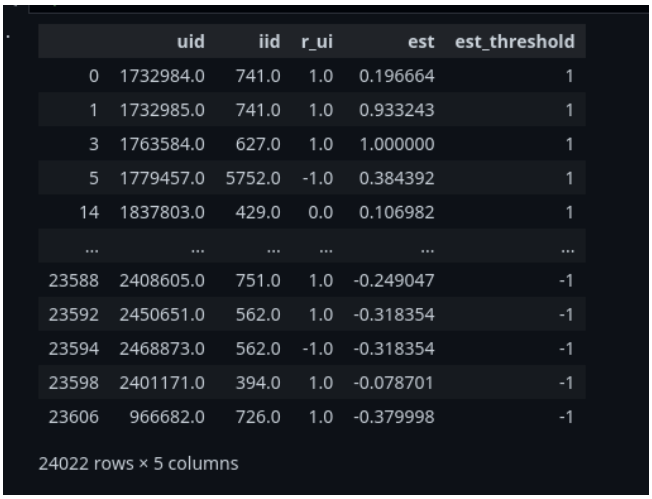
Figura 29: Algoritmo para la búsqueda de umbrales

En consecuencia, una vez finalizada la búsqueda de umbrales para la discretización de interacciones positivas y negativas, se determinó que los mejores valores corresponden a 0,01 y -0,01 respectivamente. Con estos valores procedemos a discretizar nuestro conjunto de interacciones predichas por el modelo de filtrado colaborativo.

Tabla 2: Mejores resultados para pruebas de umbrales

Umbral positivo	Umbral negativo	True Positive Rate (%)
0.01	-0.01	0.526
0.02	-0.01	0.523
0.03	-0.01	0.506
0.01	-0.02	0.491
0.02	-0.02	0.487
0.01	-0.03	0.486
0.01	-0.04	0.485
0.02	-0.03	0.483

0.02	-0.04	0.482
0.04	-0.01	0.479



	uid	iid	r_ui	est	est_threshold
0	1732984.0	741.0	1.0	0.196664	1
1	1732985.0	741.0	1.0	0.933243	1
3	1763584.0	627.0	1.0	1.000000	1
5	1779457.0	5752.0	-1.0	0.384392	1
14	1837803.0	429.0	0.0	0.106982	1
...	...	...	...	...	...
23588	2408605.0	751.0	1.0	-0.249047	-1
23592	2450651.0	562.0	1.0	-0.318354	-1
23594	2468873.0	562.0	-1.0	-0.318354	-1
23598	2401171.0	394.0	1.0	-0.078701	-1
23606	966682.0	726.0	1.0	-0.379998	-1

24022 rows × 5 columns

Figura 30: Ensayos de prueba procesados con los umbrales

2.4 Sistema de Recomendación Basado en Conocimiento

Para el tratamiento del sistema de recomendación, posterior a la obtención de los datos necesarios para el modelo, debemos detallar el algoritmo que será empleado para la obtención de las recomendaciones del conjunto de datos de los ensayos de prueba usados en el anterior modelo. Dichos datos pertenecen a la diferencia entre las versiones de ChEMBL empleadas en el presente proyecto.

En este contexto de trabajo, se procesará cada registro de los ensayos de prueba de manera individual de la siguiente forma:

1. De la célula que interactúa en el ensayo recuperaremos el listado de los ítems más similares obtenidos en la preparación de los datos.
2. En caso de no existir un registro de estos elementos, el sistema no podrá dar una recomendación al ensayo ingresado.
3. Caso contrario, por cada uno de los elementos de la lista se buscará en el banco de datos originales de los ensayos una interacción entre dicha célula y el fármaco original del ensayo.
4. Si existe un ensayo registrado de la interacción entre esa nueva célula y el fármaco original, se tomará en cuenta el tipo de interacción para dar la recomendación.

5. Adicionalmente, se recuperará en caso de existir el listado de los fármacos con más similitud al fármaco original.
6. Se realizará la búsqueda de interacciones entre la nueva célula y los nuevos fármacos, y en caso de existir se tomará en cuenta esta interacción en la recomendación.
7. Si no existen registros de ensayos entre las nuevas células y el conjunto de fármacos, el sistema no podrá dar una recomendación del ensayo procesado.

```python
def get_proximate_elements_interaction(assay):
    cell_top_elements_row = cell_similarities.loc[cell_similarities['cell_id'] == assay.cell_id]
    drugs_top_elements_row = drugs_similarities.loc[drugs_similarities['molregno'] == assay.molregno]

    if cell_top_elements_row.empty:
        return 5

    cells_top_elements = json.loads(
        cell_top_elements_row.proximate_elements.to_list()[0]
    )

    top_elements_interaction = []

    for cell in cells_top_elements.keys():
        cell_assay = old_version_assays_df[(old_version_assays_df['cell_id'] == cell) & (
            old_version_assays_df['molregno'] == assay.molregno)]

        if not cell_assay.empty:
            top_elements_interaction.append(cell_assay.interaction)

        if not drugs_top_elements_row.empty:
            drugs_top_elements = drugs_top_elements_row.top3.to_list()[0]

            for drug in drugs_top_elements:
                if drug == assay.molregno:
                    continue

                cell_assay = old_version_assays_df[(old_version_assays_df['cell_id'] == cell) & (
                    old_version_assays_df['molregno'] == drug)]

                if not cell_assay.empty:
                    top_elements_interaction.append(cell_assay.interaction)

    top_elements_interaction = np.array(top_elements_interaction)

    if np.size(top_elements_interaction) == 0:
        return 8

    interactions_count = np.unique(
        top_elements_interaction, return_counts=True)
    interactions_count = dict(
        zip(interactions_count[0], interactions_count[1])
    )

    return max(interactions_count, key=interactions_count.get)  # type: ignore
```

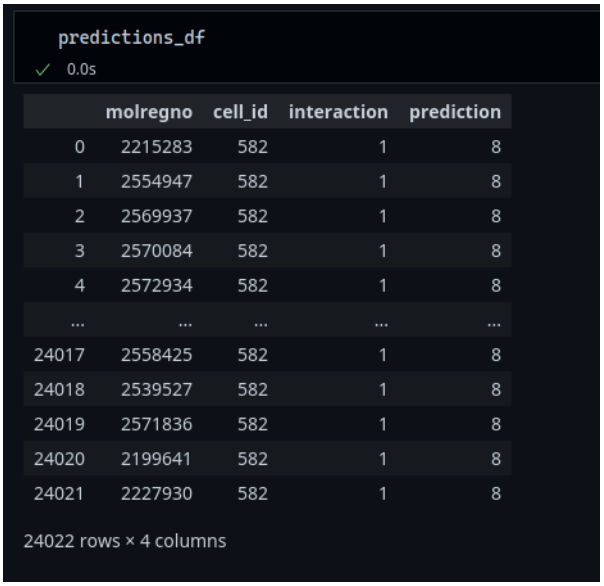
✓ 0.0s Pyth

Figura 31: Algoritmo para el sistema basado en conocimiento

Presentación de Resultados

Entre paréntesis, se definieron dos posibles valores fuera de lugar de la escala utilizada para el análisis de los resultados obtenidos por el algoritmo basado en conocimiento. En

específico, siendo el valor 5 para aquellos ensayos donde se desconocía los elementos más similares para la célula original, ocasionando un corte prematuro del algoritmo. Por otro lado, el valor 8 para aquellos ensayos donde pese a las combinaciones de nuevas células y fármacos efectuadas, el conjunto de datos de ensayos no poseía información de estas nuevas interacciones, resultando en una predicción desconocida de los ensayos de prueba.



predictions_df

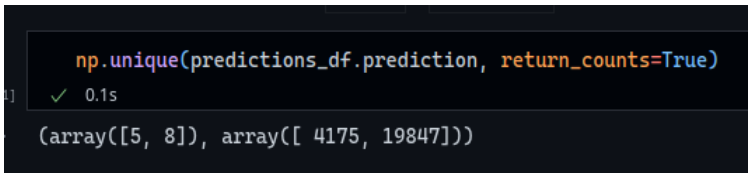
✓ 0.0s

	molregno	cell_id	interaction	prediction
0	2215283	582	1	8
1	2554947	582	1	8
2	2569937	582	1	8
3	2570084	582	1	8
4	2572934	582	1	8
...	...	...	...	...
24017	2558425	582	1	8
24018	2539527	582	1	8
24019	2571836	582	1	8
24020	2199641	582	1	8
24021	2227930	582	1	8

24022 rows × 4 columns

Figura 32: Predicciones del modelo basado en conocimiento

Finalmente, se pudo evidenciar que el valor correspondiente al desconocimiento de las nuevas interacciones en nuestro conjunto de datos original es el valor con mayor relevancia en la aplicación del algoritmo.



```
np.unique(predictions_df.prediction, return_counts=True)
```

✓ 0.1s

```
(array([5, 8]), array([ 4175, 19847]))
```

Figura 33: Valores únicos obtenidos por el modelo

2.5 Sistema de Recomendación Híbrido

Para el planteamiento de un modelo híbrido podemos tomar varios enfoques para realizar una predicción de los valores, siendo nuestro conjunto de datos compuesto de varias predicciones realizadas con la aplicación del filtrado colaborativo adicionalmente a la lógica de un modelo basado en conocimiento. Por este motivo, el planteamiento por el que se optó corresponde a tratar toda la tupla de datos como un problema de clasificación donde la entrada serán las predicciones y el valor objetivo será el valor real del ensayo.

aux

✓ 0.0s

	p_0	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	...	p_21	p_22	p_23	p_24	p_25	p_26	p_27	p_28	p_29	p_30
0	1	1	1	1	1	1	1	1	1	1	...	1	1	-1	-1	1	1	1	1	-1	-1
1	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	-1	1	-1	1	-1
2	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
3	1	1	0	1	1	1	1	1	1	1	...	1	1	1	1	1	-1	1	1	-1	1
4	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	-1	1	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
332	-1	1	1	1	-1	-1	-1	-1	-1	-1	...	1	0	1	1	-1	-1	1	-1	-1	-1
333	1	0	0	0	1	1	1	1	1	1	...	0	-1	-1	-1	1	1	1	1	1	1
334	1	1	1	1	-1	1	-1	0	1	-1	...	-1	-1	1	-1	-1	1	-1	-1	1	1
335	1	1	1	1	1	1	1	1	1	1	...	1	0	0	0	0	0	0	1	1	1
336	1	1	1	1	-1	-1	-1	1	1	1	...	1	0	0	0	1	1	1	1	1	1

337 rows × 31 columns

Figura 34: Datos de entrada para el SVM

```
y = hybrid_base_data['interaction']
y
```

✓ 0.0s

```
0      1
1      1
2     -1
3      1
4      0
      ..
332    1
333    1
334    1
335    1
336    1
Name: interaction, Length: 337, dtype: int64
```

Figura 35: Datos de salida para el SVM

En vista de ello, nos apoyaremos en el sistema de clasificación de vectores de soporte *Support Vector Machine* SVM, siendo un algoritmo de aprendizaje supervisado. Sintetizando el funcionamiento de este algoritmo, buscaremos un hiperplano que sea capaz de separar dos grupos de puntos en el espacio, donde tenemos vectores de soporte encargados de identificar la ubicación del hiperplano. [8]

Partiendo de este antecedente, definiremos la arquitectura del modelo y realizaremos el entrenamiento correspondiente para nuestro conjunto de datos además de visualizar el porcentaje de precisión del modelo.

Figura 36: Arquitectura del SVM

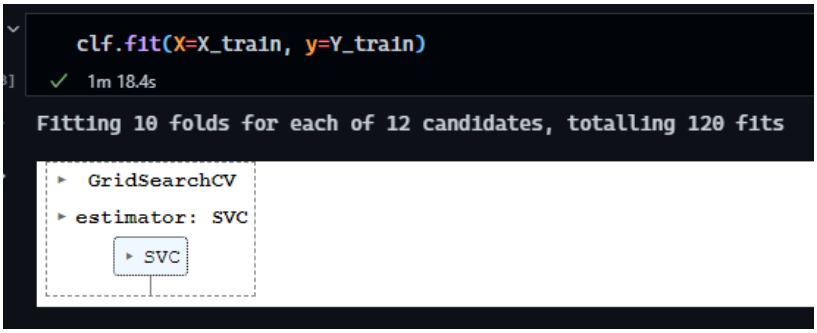


Figura 37: Invocación al entrenamiento del modelo

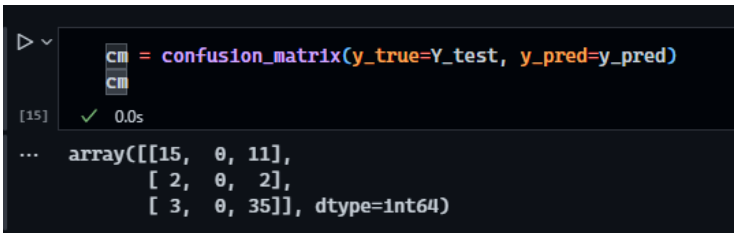


Figura 38: Matriz de confusión del SVM

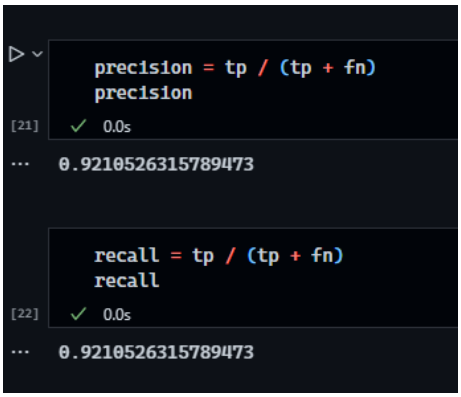


Figura 39: Métricas de evaluación del SVM

Como se puede observar en la Fig. 39, el resultado de precisión del modelo para los ensayos de prueba nos da un valor de 0.92. Finalmente, disponiendo de una matriz de confusión que nos muestra la distribución de los datos predichos para los pares fármaco-célula de prueba se visualiza en la Fig. 38.

3 PRUEBAS, RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1 Pruebas

Para el planteamiento de las pruebas efectuadas sobre el presente proyecto, cabe destacar que se realizó el mismo procedimiento para todos los modelos. Donde cada modelo fue puesto a prueba con el conjunto de ensayos que no constaban en las dos versiones de ChEMBL utilizadas, efectuando una predicción de su posible interacción y comparándola con la interacción original de los datos. Esta verificación nos permitirá establecer el posible comportamiento de los sistemas de recomendación para nuevos datos que sean creados en posteriores versiones de la fuente de datos.

De manera auxiliar a las pruebas se emplearon métricas de evaluación como el error medio cuadrático (RMSE) y matrices de confusión, siendo estas técnicas las más adecuadas para nuestro contexto.

3.2 Resultados

Filtrado Colaborativo

Para la evaluación del filtrado colaborativo tuvimos dos fases de análisis de resultados.

La fase donde se midió la precisión del modelo a través del RMSE, con los ensayos de prueba definidos. Aclarando que los resultados obtenidos de las predicciones estaban en una escala continua de -1 a 1.

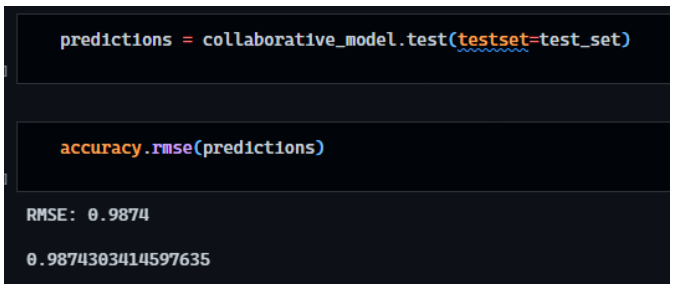


Figura 40: Porcentaje de acierto para el filtrado colaborativo

predictions_df

	uid	iid	r_ui	est
0	1732984.0	741.0	1.0	0.196664
1	1732985.0	741.0	1.0	0.933243
2	1761966.0	5555.0	0.0	-0.373830
3	1763584.0	627.0	1.0	1.000000
4	1779457.0	491.0	-1.0	-0.015621
...	...	...	...	...
24017	2709669.0	726.0	1.0	0.042628
24018	2638089.0	726.0	1.0	0.042628
24019	2648910.0	726.0	1.0	0.042628
24020	2624206.0	726.0	1.0	0.042628
24021	2580311.0	726.0	1.0	0.042628

24022 rows × 5 columns

Figura 41: Conjunto de predicciones discretas del filtrado colaborativo

Siendo la precisión de los datos de prueba de un RMSE de 0.9874, que nos indica que el modelo posee una buena caracterización de las interacciones para pares fármaco-célula que no ha procesado previamente.

La siguiente fase que correspondía al análisis de umbrales para la discretización de los valores continuos en la escala de -1, 0 o 1 dependiendo de si la interacción era negativa, desconocida o positiva respectivamente. Para ello, se hizo uso de matrices de confusión donde la suma de los valores de la diagonal correspondía a los valores que concordaban con la interacción original de los ensayos.

threshold_row

	possitive_threshold	negative_threshold	true_values
98	0.01	-0.01	12646

Figura 42: Definición de umbrales para el filtrado colaborativo

collaborative_predictions_df

	molregno	cell_id	prediction
0	1732984.0	741.0	1
1	1732985.0	741.0	1
2	1763584.0	627.0	1
3	1779457.0	5752.0	1
4	1837803.0	429.0	1
...	...	...	...
24017	2408605.0	751.0	-1
24018	2450651.0	562.0	-1
24019	2468873.0	562.0	-1
24020	2401171.0	394.0	-1
24021	966682.0	726.0	-1

24022 rows × 3 columns

Figura 43: Resultados del filtrado colaborativo

Sistema de Recomendación Basado en Conocimiento

Como se mencionó en la sección dedicada a este modelo, se utilizaron valores adicionales para identificar los distintos escenarios que se estaban dando dentro del algoritmo propuesto.

Siendo el valor 5 aquellas predicciones que no poseían la información sobre las células y sus elementos más similares. Y el valor 8 que hacía alusión a que los datos de los nuevos pares fármaco-células no existen dentro del conjunto de datos de entrenamiento.

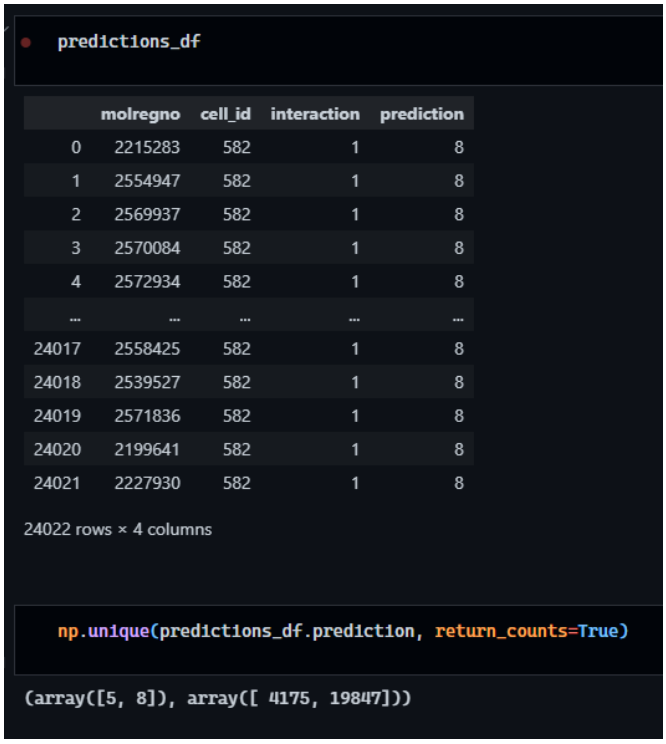


Figura 44: Resultados del modelo basado en conocimiento

Debido a que los datos con valor 8 fueron los predominantes durante las predicciones del conjunto de datos de prueba, podemos delimitar que la diferencia de datos entre las dos versiones de ChEMBL empleadas no es tan grande como esperábamos para este tipo de sistema de recomendación al tener muchas interacciones desconocidas dentro del conjunto de datos original.

Sistema de Recomendación Híbrido

Para este sistema se empleó la creación de un nuevo dataset que conectaba la lógica de nuevos pares fármaco-célula del sistema basado en conocimiento junto a las predicciones del modelo colaborativo para plantear posibles nuevas interacciones para los ensayos de prueba.

Para la evaluación del modelo en concreto se utilizaron dos métricas adicionales, la precisión que corresponde a la capacidad del modelo de distinguir los falsos positivos, y “recall”, la habilidad del modelo para encontrar todos los valores positivos, de los cuales se obtuvo los siguientes resultados.

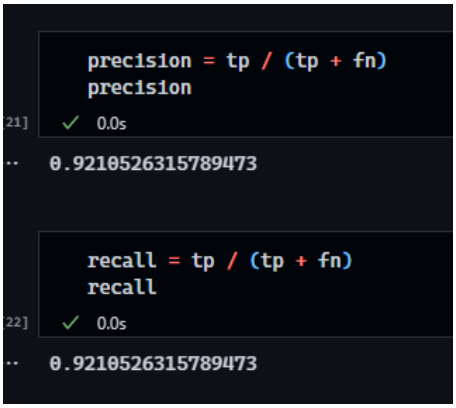


Figura 45: Resultados del modelo híbrido

El puntaje del modelo de predicción corresponde a un valor de 0.92 en su precisión y un recall de 0.92 las pruebas, donde podemos destacar que el modelo se favorece en que las predicciones realizadas poseen un valor acertado para los nuevos ensayos propuestos. Adicionalmente, el análisis de la matriz de confusión corrobora lo anteriormente descrito donde tenemos pocos valores que no están contenidos dentro de la diagonal de valores que corresponden a la predicción acertada de la interacción real de los ensayos.

Tabla 3:Precisión Sistemas de Recomendación

Sistema de Recomendación	**Precisión Pruebas**
Filtrado Colaborativo	0.9874
Sistema Híbrido	0.92

3.3 Conclusiones

- El planteamiento de un sistema de recomendación para interacciones farmacológicas es viable para el análisis de los pares fármaco-célula a través de modelos como el filtrado colaborativo que nos permiten predecir nuevos valores de interacción para datos desconocidos en la matriz de interacción de todos los pares posibles.
- El sistema de recomendación basado en conocimiento tiene como principal inconveniente que depende de que la mayoría de los datos que corresponden a la matriz de interacción se encuentren con un valor de interacción ya establecido, por este motivo, en el contexto de nuestra base de ensayos no es viable aplicar esta orientación dado el gran número de interacciones desconocidas.
- Un sistema de recomendación híbrido puede plantearse a través de diversos enfoques en la combinación de la lógica que poseen los demás sistemas, siendo viable la mezcla

de razonamientos para crear sistemas más confiables para la predicción de nuevos datos.

- Se logró una estandarización en el proceso de la creación de un modelo de sistema de recomendación para la base de datos ChEMBL, pudiendo aplicar toda la experimentación de datos a cualquier versión nueva o antigua de este banco de información.

3.4 Recomendaciones

- Se recomienda obtener una fuente de datos más amplia de células y fármacos, para la obtención de información de similitud entre los elementos, para enriquecer el funcionamiento del algoritmo basado en conocimiento, además de aumentar la viabilidad de este sistema para el contexto de bases farmacológicas.
- Se recomienda continuar con los estudios del comportamiento de los modelos para las versiones pasadas de la fuente de datos ChEMBL, de igual manera, realizar nuevas pruebas con las futuras actualizaciones de esta información.
- Se recomienda ampliar el estudio de las interacciones definidas en los ensayos de la base de datos ChEMBL, con la finalidad de emplear una data mucho más robusta en el llenado de la matriz de interacción requerida para todos los modelos planteados en el presente trabajo.

4 REFERENCIAS BIBLIOGRÁFICAS

[1] C. C. Aggarwal, "18.5 Recommender Systems," in *Data Mining The Textbook*, Switzerland, Springer International Publishing, 2015, pp. 604-608.

[2] R. Burke, «Hybrid Recommender Systems: Survey and Experiments,» *User Modeling and User-Adapted Interaction,* vol. 12, 2002.

[3] C. C. Aggarwal, «Ensemble-Based and Hybrid Recommender Systems,» de *Recommender Systems The Textbook*, Switzerland, Springer International Publisher, 2016, pp. 198-204.

[4] F. S. González, F. Prieto Martínez y J. L. Medino-Franco, « Descubrimiento y desarrollo de fármacos: Un enfoque computacional,» *Educación Química,* vol. 28, nº 1, pp. 51-58, 2017.

[5] C. C. Aggarwal, «An Introduction to Recommender Systems,» de *Recommender Systems The Textbook*, Switzerland, Springer International Publisher, 2016, pp. 1-20.

[6] EMBL's European Bioinformatics Institute, «ChEMBL,» EMBL's European Bioinformatics Institute, [En línea]. Available: https://www.ebi.ac.uk/chembl/#. [Último acceso: 23 Enero 2023].

[7] Expasy, «Cellosaurus,» Expasy, [En línea]. Available: https://www.cellosaurus.org/. [Último acceso: 23 Enero 2023].

[8] MathWorks, «Support Vector Machine (SVM),» MathWorks, [En línea]. Available: https://la.mathworks.com/discovery/support-vector-machine.html. [Último acceso: 25 Febrero 2023].

5 ANEXOS

ANEXO I: Enlace al Repositorio Digital

Enlace donde se encuentra almacenado todo el procesamiento y la información obtenida en cada una de las fases de la experimentación del proyecto.

https://gitlab.com/tic-drug-repositioning/recommender-system-hybrid