“Predicción de respuesta a fármacos quimioterapéuticos a partir de datos genómicos”

Alfonso Esteban Lasso
Máster en Bioinformática y Bioestadística
Área 4
Nombre Consultor/a: Héctor Tejero Franco
Nombre Profesor/a responsable de la asignatura: Jeroni Luna Cornadó
06/06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	Predicción de respuesta a fármacos quimioterapéuticos a partir de datos genómicos.
Nombre del autor:	*Alfonso Esteban Lasso*
Nombre del consultor/a:	*Héctor Tejero Franco*
Nombre del PRA:	*Jeroni Luna Cordanó*
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	*Máster en Bioinformática y Bioestadística*
Área del Trabajo Final:	*4*
Idioma del trabajo:	*Español*
Palabras clave	*Predicción* *Cáncer* *Algoritmo*

Resumen del Trabajo:

El cáncer es una de las principales causas de mortalidad en el mundo. A pesar de los grandes avances realizados en las últimas décadas, en muchos casos los tumores no responden al tratamiento estándar o bien desarrollan resistencia durante el mismo. Para facilitar futuros tratamientos personalizados se están desarrollando una serie de tecnologias genómicas de alto rendimiento, entre ellas algoritmos de Machine Learning de predicción de respuesta a fármacos.

Con esto en mente he aplicado una serie de algoritmo de Machine Learning con distintas combinaciones de personalización de ajuste con la finalidad de desarrollar modelos capaces de predecir, con la mayor precisión posible, la respuesta a fármacos registrados en el GDSC y DepMap Broad Institute.

Estos algoritmos han sido entrenados y ajustados para 4 fármacos al azar para ver la precisión de predicción sobre sus correspondientes datos de expresión en base a los valores AUC obtenidos por estas dos Instituciones con resultados favorables y obteniéndo las 2 mejores combinaciones probadas:

El fármaco que mejor se predice es el Erlotinib con un 87,64% de precisión acertando predicciones en una partición de datos 60/20/20, mediante Random Search en Random Forest cuando los valores Auc están discretizados.

Cuando los predictores han sido entrenados como valores continuos el mejor valor de R^2 obtenido ha sido 1 correspondiente a la predicción de respuesta para los fármacos Erlotinib, Rapamycin y Sunitinib mediante el modelo ajustado de regularización de Ridge con hiperparámetros seleccionados mediante RandomSearch y con una particion 80/20 en 10 folds validación cruzada.

Abstract:

Cancer is one of the leading causes of death in the world. Despite the great advances made in recent decades, in many cases the tumors do not respond to standard treatment or develop resistance during treatment. In order to facilitate future personalized treatments, a series of high-performance genomic technologies are being developed, including Machine Learning algorithms for predicting drug response.

With this in mind, I have applied a series of Machine Learning algorithms with different combinations of adjustment customization in order to develop models capable of predicting, as accurately as possible, the response to drugs registered at the GDSC and DepMap Broad Institute.

These algorithms have been trained and adjusted for 4 random drugs to see the accuracy of prediction on their corresponding expression data based on the AUC values obtained by these two institutions with favorable results and obtaining the 2 best combinations tested:

The drug that is best predicted is Erlotinib with 87.64% accuracy, accurate predictions in a 60/20/20 data partition, by means of Random Search in Random Forest when the Auc values are discretised.

When the predictors have been trained as continuous values, the best R2 value obtained was 1 corresponding to the response prediction for Erlotinib, Rapamycin and Sunitinib drugs using the adjusted Ridge regularization model with hyperparameters selected by RandomSearch and with an 80/20 partition in 10 en validación cruzada folds.

Índice

Contenido

Lista de figuras

1. Introducción

1.1 Contexto y justificación del Trabajo:

El cáncer es una de las principales causas de mortalidad en el mundo. A pesar de los grandes avances realizados en las últimas décadas, en muchos casos los tumores no responden al tratamiento estándar o bien desarrollan resistencia durante el mismo. El desarrollo de las tecnologías genómicas de alto rendimiento permite una caracterización genómica profunda de los tumores. La búsqueda de marcadores de respuesta a fármacos antitumorales es de gran interés y entraría dentro de la idea general de "medicina personalizada": darle a cada paciente el tratamiento adecuado según sus características socio-biológicas particulares. El desarrollo de estas tecnologías de alto rendimiento ha permitido también la aplicación de algoritmos de inteligencia artificial o machine learning para tratar de producir, a partir de datos experimentales generados a priori, algoritmos que puedan aprender de dichos datos y ser usados en otros contextos para la predicción de fenotipos biológicos de interés como la respuesta a fármacos.

1.2 Objetivos del Trabajo:

Los objetivos principales han sido divididos en dos:

1- Desarrollar un script que genere una matriz de expresión para cualquier fármaco incluido en las bases de datos GDSC y GenomeCancerProyect.org con los valores de sensibilidad a ese fármaco en las líneas celulares en los que se aplica y los valores de expresión de los genes asociados a esas líneas celulares tumorales:

1.1 Combinar los distintos datasets y metadatos para crear en última instancia la matriz de expresión deseada.

1.2 Inputar y preprocesar los distintos datos de interés para su posterior análisis.

2- Desarrollar un modelo óptimo de machine learning para predecir de forma automática la respuesta a fármacos quimioterapéuticos a raíz de las matrices de expresión generadas en el objetivo anterior.

2.1 Una vez desarrollado el script para generar matrices de expresión usar los distintos tipos de validación cruzada para tener los datos más óptimos para el modelo de análisis.

2.2 Probar distintos algoritmos de machine learning para conseguir los mejores análisis en la predicción de sensibilidad a fármacos.

1.3 Enfoque y método seguido:

A partir de los datasets obtenidos en la enciclopedia de líneas celulares cancerígenas (https://portals.broadinstitute.org/ccle) y del cancer center del hospital general de Massachusetts (https://www.cancerrxgene.org/), preprocesar los datos hasta obtener la matriz de expresión para un fármaco concreto con sus valores AUC y los valores de expresión de los genes de las líneas celulares tumorales donde actúa el fármaco mediante inputar y fusionar dataframes con origen en los anteriores datasets para después entrenar los datos obtenidos en la matriz de expresión con distintas formas de validación cruzada y analizarlos mediante distintos algoritmos, tales como Random Forest, regresión de Ridge o LASSO, hasta concluir con el modelo y los parámetros de este que mejor se ajusten. Todo esto desarrollado en R por la familiaridad del estudiante en este lenguaje de programación aunque si sobra tiempo se extrapolará a lenguajes como Python para usar el máximo de herramientas disponibles en la elaboración de este software predictor y sacar el mayor número de conclusiones posibles en el tiempo establecido.

De esta forma se consolidarán los conocimientos aprendidos durante el máster de machine learning y programación en R, principalmente, además de comprender en profundidad la forma de usar los distintos algoritmos y sus aplicaciones en biomedicina usando el mayor número de herramientas disponibles al alcance y por tanto encontrando la forma más apropiada de lograr el objetivo principal del proyecto: Desarrollo de predictores automáticos de respuesta a fármacos quimioterapeúticos.

1.4 Planificación del Trabajo:

Para la realización del proyecto he necesitado un ordenador con sistema operativo Linux, el software de programación instalado R y Rstudio. A continuación se muestra el plan de Trabajo seguido con las tareas programadas para las fechas en las que se ha desarrollado este y el seguimiento de las mismas. Al final de este se encuentra un sumatorio con las horas totales dedicadas a las tareas que comprenden el proyecto.

Fecha	Proyecto / Actividad	Horas	Descripción actividad	Seguimiento
20/02/2019	Definición de los contenidos del trabajo	5	Reunión para debatir sobre el tema en concreto del TFM	Cumplido
21/02/2019	Definición de los contenidos del trabajo	5	Búsqueda y primer contacto con los pappers que usaremos en el trabajo.	Cumplido
22/02/2019	Definición de los contenidos del trabajo	5	Continuación de lectura y comprensión de pappers asociados al estudio que se va a llevar a cabo	Cumplido
23/02/2019	Definición de los contenidos del trabajo	0	Fin de semana	
24/02/2019	Definición de los contenidos del trabajo	0	Fin de semana	
25/02/2019	Definición de los contenidos del trabajo	5	Continuación de lectura y comprensión de pappers asociados al estudio que se va a llevar a cabo	Cumplido
26/02/2019	Definición de los contenidos del trabajo	5	Continuación de lectura y comprensión de pappers asociados al estudio que se va a llevar a cabo	Cumplido
27/02/2019	Definición de los contenidos del trabajo	5	Inicio de búsqueda y comprensión de datasets	Cumplido
28/02/2019	Definición de los contenidos del trabajo	5	Búsqueda y comprensión de datasets	Cumplido
01/03/2019	Definición de los contenidos del trabajo	5	Búsqueda y comprensión de datasets y realización de PEC 0	Cumplido
02/03/2019	Definición de los contenidos del trabajo	0	Fin de semana	
03/03/2019	Definición de los contenidos del trabajo	0	Fin de semana	
04/03/2019	Definición de los contenidos del trabajo	5	Realización de PEC 0 y entrega	Cumplido
05/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
06/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
07/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
08/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
09/03/2019	Plan de trabajo	0	Fin de semana	
10/03/2019	Plan de trabajo	0	Fin de semana	
11/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
12/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
13/03/2019	Plan de trabajo	5	Búsqueda y comprensión de datasets	Cumplido
14/03/2019	Plan de trabajo	5	Realización de PEC1.	Cumplido
15/03/2019	Plan de trabajo	5	Realización de PEC1.	Cumplido
16/03/2019	Plan de trabajo	0	Fin de semana	
17/03/2019	Plan de trabajo	0	Fin de semana	
18/03/2019	Plan de trabajo	5	Realización de PEC1 y entrega.	Cumplido
19/03/2019	Desarrollo del trabajo - Fase 1	5	Inicio de la realización del primer objetivo general: desarrollo de la matriz de expresión	Cumplido
20/03/2019	Desarrollo del trabajo - Fase 1	5	Creación de dataframes a partir de los datasets iniciales con los datos de interés	Cumplido
21/03/2019	Desarrollo del trabajo - Fase 1	5	Preprocesamiento e inputación de datos	Cumplido
22/03/2019	Desarrollo del trabajo - Fase 1	5	Preprocesamiento e inputación de datos	Cumplido
23/03/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
24/03/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
25/03/2019	Desarrollo del trabajo - Fase 1	5	Preprocesamiento e inputación de datos	Cumplido
26/03/2019	Desarrollo del trabajo - Fase 1	5	Preprocesamiento e inputación de datos	Cumplido
27/03/2019	Desarrollo del trabajo - Fase 1	5	Preprocesamiento e inputación de datos	Cumplido
28/03/2019	Desarrollo del trabajo - Fase 1	5	Creación de matriz del fármaco a elegir con sus respectivos valores AUC y en las líneas celulares asociadas a estos	Cumplido
29/03/2019	Desarrollo del trabajo - Fase 1	5	Creación de matriz del fármaco a elegir con sus respectivos valores AUC y en las líneas celulares asociadas a estos	Cumplido
30/03/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
31/03/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
01/04/2019	Desarrollo del trabajo - Fase 1	5	Creación de la matriz de genes de expresión asociados a cada línea celular tumoral	Cumplido
02/04/2019	Desarrollo del trabajo - Fase 1	5	Creación de la matriz de genes de expresión asociados a cada línea celular tumoral	Cumplido
03/04/2019	Desarrollo del trabajo - Fase 1	5	Creación de la matriz de genes de expresión asociados a cada línea celular tumoral	Cumplido
04/04/2019	Desarrollo del trabajo - Fase 1	5	Creación de la matriz de genes de expresión asociados a cada línea celular tumoral	Cumplido
05/04/2019	Desarrollo del trabajo - Fase 1	5	Fusión de matrices para crear la matriz de expresión final	Cumplido
06/04/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
07/04/2019	Desarrollo del trabajo - Fase 1	0	Fin de semana	
08/04/2019	Desarrollo del trabajo - Fase 1	5	Comprobación y revisión del script con distintos fármacos	Cumplido
09/04/2019	Desarrollo del trabajo - Fase 1	5	Comprobación y revisión del script con distintos fármacos	Cumplido
10/04/2019	Desarrollo del trabajo - Fase 1	5	Comprobación y revisión del script con distintos fármacos	Cumplido
11/04/2019	Desarrollo del trabajo - Fase 1	5	Comprobación y revisión del script con distintos fármacos	Cumplido
12/04/2019	Desarrollo del trabajo - Fase 1	5	Comprobación y revisión del script con distintos fármacos	Cumplido
13/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
14/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
15/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
16/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
17/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
18/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
19/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
20/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
21/04/2019	Desarrollo del trabajo - Fase 1	0	Semana Santa	
22/04/2019	Desarrollo del trabajo - Fase 1	5	Realización del informe de desarrollo del trabajo Fase 1	Cumplido
23/04/2019	Desarrollo del trabajo - Fase 1	5	Realización del informe de desarrollo del trabajo Fase 1	Cumplido
24/04/2019	Desarrollo del trabajo - Fase 1	5	Realización y entrega informe desarrollo del trabajo Fase 1.	Cumplido

25/04/2019	Desarrollo del trabajo - Fase 2	5	Inicio de la realización del segundo objetivo general: distintos tipos de validación cruzada y métodos analíticos para conseguir un buen predictor de sensibilidad a fármacos con machine learning	Cumplido
26/04/2019	Desarrollo del trabajo - Fase 2	5	Lectura y compresión sobre los distintos tipos de validación cruzada	Cumplido
27/04/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
28/04/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
29/04/2019	Desarrollo del trabajo - Fase 2	5	Lectura y compresión sobre los distintos tipos de validación cruzada	Cumplido
30/04/2019	Desarrollo del trabajo - Fase 2	5	Puesta a prueba de los distintos tipos de validacion cruzada en nuestra matriz de expresión final	Cumplido
01/05/2019	Desarrollo del trabajo - Fase 2	5	Puesta a prueba de los distintos tipos de validacion cruzada en nuestra matriz de expresión final	Cumplido
02/05/2019	Desarrollo del trabajo - Fase 2	5	Puesta a prueba de los distintos tipos de validacion cruzada en nuestra matriz de expresión final	Cumplido
03/05/2019	Desarrollo del trabajo - Fase 2	5	Lectura y compresión de los distintos algoritmos a usar para el análisis de los datos entrenados por validación cruzada y comprensión de los distintos parámetos a ajustar	Cumplido
04/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
05/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
06/05/2019	Desarrollo del trabajo - Fase 2	5	Lectura y compresión de los distintos algoritmos a usar para el análisis de los datos entrenados por validación cruzada y comprensión de los distintos parámetos a ajustar	Cumplido
07/05/2019	Desarrollo del trabajo - Fase 2	5	Lectura y compresión de los distintos algoritmos a usar para el análisis de los datos entrenados por validación cruzada y comprensión de los distintos parámetos a ajustar	Cumplido
08/05/2019	Desarrollo del trabajo - Fase 2	5	Aplicación de los distintos algoritmos	Cumplido
09/05/2019	Desarrollo del trabajo - Fase 2	5	Aplicación de los distintos algoritmos	Cumplido
10/05/2019	Desarrollo del trabajo - Fase 2	5	Compresión de los resultados obtenidos tras los análisis	Cumplido
11/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
12/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
13/05/2019	Desarrollo del trabajo - Fase 2	5	Compresión de los resultados obtenidos tras los análisis	Cumplido
14/05/2019	Desarrollo del trabajo - Fase 2	5	Compresión de los resultados obtenidos tras los análisis	Cumplido
15/05/2019	Desarrollo del trabajo - Fase 2	5	Compresión de los resultados obtenidos tras los análisis	Cumplido
16/05/2019	Desarrollo del trabajo - Fase 2	5	Realización del informe de desarrollo del trabajo Fase 2	Cumplido
17/05/2019	Desarrollo del trabajo - Fase 2	5	Realización del informe de desarrollo del trabajo Fase 2	Cumplido
18/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
19/05/2019	Desarrollo del trabajo - Fase 2	0	Fin de semana	
20/05/2019	Desarrollo del trabajo - Fase 2	5	Realización y entrega informe desarrollo del trabajo Fase 2.	Cumplido
21/05/2019	Realización de la memoria	5	Deducir conclusiones y discusiones a raíz de los distintos resultados obtenidos	Cumplido
22/05/2019	Realización de la memoria	5	Aplicación del modelo de machine learning SVM	Cumplido
23/05/2019	Realización de la memoria	5	Deducir conclusiones y discusiones a raíz de los distintos resultados obtenidos	Cumplido
24/05/2019	Realización de la memoria	5	Inicio realización memoria y organización del contenido de esta.	Cumplido
25/05/2019	Realización de la memoria	0	Fin de semana	
26/05/2019	Realización de la memoria	0	Fin de semana	
27/05/2019	Realización de la memoria	5	Realización de memoria hasta Introducción (incluida)	Cumplido
28/05/2019	Realización de la memoria	5	Realización del resto de capítulos hasta conclusión (sin incluir)	Cumplido
29/05/2019	Realización de la memoria	5	Realización del resto de capítulos hasta conclusión (sin incluir)	Cumplido
30/05/2019	Realización de la memoria	5	Realización de la conclusión hasta anexos	Cumplido
31/05/2019	Realización de la memoria	5	Realización de la conclusión hasta anexos	Cumplido
01/06/2019	Realización de la memoria	0	Fin de semana	
02/06/2019	Realización de la memoria	0	Fin de semana	
03/06/2019	Realización de la memoria	5	Revisión y correción memoria	Cumplido
04/06/2019	Realización de la memoria	5	Revisión y correción memoria	Cumplido
05/06/2019	Elaboración de la presentación	5	Realización esquemática de cómo distribuir la información de la memoria en la presentación	Por comenzar
06/06/2019	Elaboración de la presentación	5	Realización esquemática de cómo distribuir la información de la memoria en la presentación	Por comenzar
07/06/2019	Elaboración de la presentación	5	Aplicación del esquema a la presentación	Por comenzar
08/06/2019	Elaboración de la presentación	0	Fin de semana	
09/06/2019	Elaboración de la presentación	0	Fin de semana	
10/06/2019	Elaboración de la presentación	5	Aplicación del esquema a la presentación	Por comenzar
11/06/2019	Elaboración de la presentación	5	Revisión y perfeccionamiento de la presentación	Por comenzar
12/06/2019	Elaboración de la presentación	5	Revisión y perfeccionamiento de la presentación	Por comenzar
	Horas totales	380		

1.5 Breve sumario de productos obtenidos:

Nueve archivos .R con los scripts de programación empleados y comentados además de nombrados por la función que cumplen y que generan los respectivos modelos ajustados mencionados (para ejecutar el resto de archivos .R primero hay que ejecutar el archivo "creacionmatriz.R" para generar la matriz que emplearán el resto de archivos para ejecutarse):

- **Creacionmatriz.R:** Script que al ejecutarse genera la matriz de expresión para el fármaco escogido y se usa posteriormente en el entrenamiento de los distintos modelos de predicción.

- **H2ocontinuosRF.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de Random Forest con búsqueda de hiperparámetros en H2o y los predictores con valores continuos; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente ambos modelos (Las particiones 80/20 y 60/20/20 han de ejecutarse por separado para que el software H2o seleccione las nuevas particiones).

- **H2odiscretizadosRF.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de Random Forest con búsqueda de hiperparámetros en H2o y los predictores con valores discretizados; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente y mediante matrices de confusión ambos modelos (Las particiones 80/20 y 60/20/20 han de ejecutarse por separado para que el software H2o seleccione las nuevas particiones).

- **randomsearchcontinuosRF.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de Random Forest con búsqueda de hiperparámetros en random search y los predictores con valores continuos; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente ambos modelos.

- **randomsearchdiscretizadosRF.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de Random Forest con búsqueda de hiperparámetros en random search y los predictores con valores discretizados; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente y mediante matrices de confusión ambos modelos.

- **Ridgelassoelasticcontinuosrandomsearch.R:** Script que al ejecutarse genera seis arhivos .rds con los modelos ajustados de Ridge, LASSO y Elastic Net con búsqueda de hiperparámetros en random search y los predictores con valores continuos; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente estos modelos.

- **Ridgelassoelasticdiscretizadosrandomsearch.R:** Script que al ejecutarse genera seis arhivos .rds con los modelos ajustados de Ridge, LASSO y Elastic Net con búsqueda de hiperparámetros en random search y los predictores con valores discretizados; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente y mediante matrices de confusión estos modelos.

- **SVMLinealcontinuos.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de SVM Lineal con búsqueda de hiperparámetros en random search y los predictores con valores continuos; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente estos modelos.

- **SVMLinealdiscretizados.R:** Script que al ejecutarse genera dos arhivos .rds con los modelos ajustados de SVM Lineal con búsqueda de hiperparámetros en random search y los predictores con valores discretizados; para las particiones de datos 80/20 y 60/20/20 además de poder visualizarse graficamente y mediante matrices de confusión estos modelos.

1.6 Breve descripción de los otros capítulos de la memoria:

Materiales: Lista de materiales empleados en el desarrollo del proyecto con descripción detallada de cada uno de ellos: databases, bibliotecas usades en el desarrollo del script, software empleado y ordenadores.

Métodos: Todos los métodos empleados en el desarrollo del proyecto explicados con mayor profundidad, desde la explicación de la creación de la matriz de expresión, distintas formas empleadas en la partición de los datos, las distintas búsquedas de hiperparámetros empleadas, los distintos modelos empleados y dependiendo de la clase de predictores que se usen y las distintes formas de representación de los resultados obtenidos.

Resultados: Hiperparámetros óptimos obtenidos para cada modelo empleado, matrices de confusión y representaciones gráficas de los distintos modelos ajustados en la predicción de respuesta a fármacos quimioterapéuticos para los 4 fármacos escogidos al azar.

Discusión: Deducciones obtenidas al observar los resultados.

Conclusión: Distintas conclusiones obtenidas a raíz de la discusión de los resultados obtenidos.

2. Resto de capítulos:

Materiales:

Los materiales utilizados para la realización de este proyecto han sido los siguientes:

Datasets:

Para la obtención de las bases de datos, que contendrán los datos genómicos de expresión en líneas celulares tumorales sobre las que actúan los fármacos de estudio, he accedido a las páginas de internet asociadas al GDSC (The Genomics of Drug Sensitivity in Cancer Project) y al Broad Institute DepMap:

1. El archivo obtenido del GDSC es el asociado al archivo "v17.3_fitted_dose_response.xls" que contiene la información relativa a la concentración inhibitoria media máxima natural (IC50) y valores de área bajo la curva dosis-respuesta (AUCs) para todas las combinaciones de línea celular/medicamento examinadas por la colaboración entre el Proyecto del Genoma del Cáncer en el Wellcome Sanger Institute (UK) y el Centro de Terapias Moleculares, Massachusetts General Hospital Cancer Center (USA).
   Estos datos han sido seleccionados por ser el mayor recurso público de información sobre la sensibilidad a los medicamentos en las células cancerosas y los marcadores moleculares de la respuesta a los medicamentos. Los datos están disponibles gratuitamente y sin restricciones. El GDSC contiene actualmente datos de sensibilidad a los medicamentos para casi 75.000 experimentos, que describen la respuesta a 138 medicamentos anticancerosos a través de casi 700 líneas de células cancerosas.

2. Los archivos obtenidos del DepMap son los asociados a los archivos "DepMap-2019q1-celllines.csv" y "CCLE_depMap_19Q1_TPM.csv" que corresponden a los metadatos asociados a los valores de expresión expresados en transcripciones por millón (TPM), es decir, por cada 1.000.000 de moléculas de ARN en la muestra de ARN-seq, x provenía de este gen/transcripción, en las líneas celulares tumorales de estudio. Estos valores han sido obtenidos, a partir de archivos RNAseq utilizando los pipelines GTEx, por el Broad Institute DepMap. Estos archivos han sido seleccionados por contener 56202 genes y 1201 líneas celulares, datos de caracterización genómica del proyecto CCLE (El proyecto Cancer Cell Line Encyclopedia) el cual comenzó en 2008 como una colaboración entre el Instituto Broad y los Institutos Novartis de Investigación Biomédica y su Instituto de Genómica de la Fundación de Investigación Novartis.

Con estos 2 archivos de valores, tanto los valores AUC dosis/respuesta como los valores de expresión, y el archivo de metadatos he conseguido aplicar una serie de modelos de Machine Learning para observar cúal se ajustaría mejor en la predicción de respuesta a fármacos usando como valores predictivos los valores AUC y ver si estos valores serían buenos predictores para la predicción de respuesta a fármacos con nueva información.

Programación:

El desarrollo de este proyecto ha sido realizado mediante lenguaje de programación R y en la interfaz gráfica de Rstudio, instalado en el sistema operativo de Linux (Ubuntu 18.04).

R es un lenguaje de programación y un entorno de software libre para computación estadística y gráficos apoyado por la R Foundation for Statistical Computing. R y sus bibliotecas implementan una amplia variedad de técnicas estadísticas y gráficas, incluyendo modelos lineales y no lineales, pruebas estadísticas clásicas, análisis de series de tiempo, clasificación, clustering y otros. Este tipo de lenguaje de programación ha sido elegido por mi familiaridad con éste y sus bibliotecas y paquetes lo que me ha permitido algo de fluidez a la hora de aplicar el código de programación para la resolución de los objetivos propuestos en este proyecto.

El sistema operativo Linux es un sistema operativo libre tipo Unix; multiplataforma (son implementados e interoperan en múltiples plataformes informáticas), multiusuario y multitarea (permite que varios procesos o aplicaciones se ejecuten aparentemente al mismo tiempo). Ha sido elegido por su velocidad de procesamiento de código en comparación con otros sistemas operativos.

Rstudio es un entorno de desarrollo integrado (IDE) para el lenguaje de programación R, dedicado a la computación estadística y gráficos. Incluye una consola, editor de sintaxis que apoya la ejecución de código, así como herramientas para el trazado, la depuración y la gestión del espacio de trabajo facilitándome el desarrollo y revisión del código y haciéndome más accessibles visualmente los resultados.

Bibliotecas usadas en RStudio con sus respectivas versiones y funciones dentro de este proyecto:

library(ROCR) (1.0.7): visualizar curvas ROC en datos discretizados.
library(e1071)(1.7.1): entrenamiento de modelos con SVM.
library(glmnet)(2.0.16): entrenamiento de modelos con Ridge, Lasso y Elastic Net.
library(caret) (6.0): búsqueda de hiperparámetros con random search y formación de matrices de confusión en predicciones discretizadas.
library(miscTools)(0.6.22): para calcular la R^2 en modelos continuos.
library(randomForest) (4.6): entrena modelos de tipo random Forest.
library(H2o) (3.22.1): búsqueda de hiperparámetros con el software H2o.
library(readxl) (1.3.0): lectura de archivos .xls.
library(data.table) (1.12.0): para manipular los datasets iniciales.
library(resample): para ordenar dataframes por sus varianzas más altas.
library(plyr): para seleccionar y copiar listas de nombres.
Library(caTools): para realizar las particiones de datos.
library(ggplot2): para la representacion gráfica de los datos continuos.

Para el desarrollo de este proyecto, además, han sido usados dos ordenadores para agilizar los procesos de lectura de código que en algunos casos podia alargarse durante más de 20 minutos por cada búsqueda de hiperparámetros realizada.
Los ordenadores usados han sido:

- Un ordenador de sobremesa prestado por el Cnio (Centro Nacional de Investigaciones Oncológicas) con estas características principales:
  PC Sobremesa - Acer Aspire XC-885, AXC-885, CI5-8400, 8GB, 1TB+128GB, GT720, W10, Negro. Procesador: Intel® Core™ i5-8400: Número Procesador: i5-8400: Modelo Procesador: Core i5: Tamaño de caché: 9 MB: Tamaño memoria RAM: 8 GB: Configuración RAM: 8; Ranuras RAM: 4

- Un ordenador portàtil de mi propiedad con estas características principales:
  Portátil - Legion Y530-15ICH 15, Intel® Core™ I5-8300H, 8 GB, 1TB, GTX 1050-4GB; Resolución: 1920 x 1080; Calidad de imagen: Full-HD; Tipo de pantalla: 15.6" Full HD; Tipo de RAM: DDR4; Tamaño memoria RAM: 8 GB; Procesador: Intel® Core™ i5-8300H (4 x 2.3 GHz); Modelo Procesador: Core i5.

Métodos:

Creación de la matriz de expresión:

El archivo obtenido en el GDSC (“v17.3_fitted_dose_response.xls”) ha sido leído mediante la función read_excel(), del paquete “readxl”. Viendo que consta de 224202 observaciones, correspondientes al número de combinaciones de línea celular/medicamento examinados, y 13 variables, correspondientes a los resultados obtenidos del ajuste de los datos por el GDSC, he seleccionado únicamente las columnas: variables asociadas al nombre del fármaco (Nombre principal del medicamento), línea celular (Nombre primario de la línea celular) y valor AUC (Área bajo la curva para el modelo ajustado presentado como fracción de la superficie total entre la más alta y la más baja concentración de cribado) , ya que son las 3 variables que van a representar la respuesta a fármacos. Esta selección ha sido sintetizada, mediante la función data.frame() incluida en el software base de R, en un dataframe (tabla o estructura bidimensional en la que cada columna contiene valores de una variable y cada fila contiene un conjunto de valores de cada columna) con el mismo número de observaciones y solo las columnas de interés.
El siguiente paso ha sido la creación de una variable que contenga el nombre del fármaco al que queremos predecir su respuesta. En este estudio han sido escogidos al azar 4 fármacos (Erlotinib, Rapamycin, Sunitinib y Paclitaxel) para las pruebas de precisión del predictor. Una vez creada la variable he creado un bucle que comprueba si este fármaco se encuentra en el dataframe y si es así selecciona todas las observaciones correspodientes a este fármaco con las líneas celulares en las que ha sido probado y los valores AUC obtenidos obteniendo así un dataframe personalizado para cada fármaco de ínterés. Formado por 2 variables (nombre de la línea celular y valor AUC) como columnas y las observaciones como filas y guardando éste, mediante la función saveRDS() implementada en el software base de R, como un archivo .rds (“farmaco.rds”) para cargarlo directamente al entrenar los modelos de interés y no tener que volver a ejecutar de nuevo todo el código empleado hasta ahora.

El siguiente paso ha sido leer, mediante la función read.csv() implementada en el software base de R, el archivo “DepMap-2019q1-celllines.csv” que corresponde a los metadatos asociados a los valores de expresión de los genes expresados en las líneas celulares tumorales obtenidos del Broad Institute DepMap.
Este archivo consiste en 1677 observaciones (filas) y en 9 variables (columnas), de las cuales seleccionaremos la variable asociada a la ID designada por DepMap y la variable asociada a la ID de Cosmic para cada observación obteniendo, mediante la función data.frame(), un dataframe de 2 variables y 1677 observaciones.
Este dataframe lo he usado para, mediante la funcion merge() implementada en el software base de R, fusionarlo con otro dataframe, creado a su vez con las variables: nombre de la línea celular y la variable asociada a la ID de Cosmic (ambas variables pertenecientes al archivo antes leído“v17.3_fitted_dose_response.xls”), y con 224202 observaciones.
De esta fusión de dataframes he obtenido una asociación común entre la nomenclatura del DepMap y Cosmic para los nombres de las líneas celulares tumorales ya que dependiendo de la base de datos usada tendrán una u otra ID.
Esta asociación consiste en un dataframe formado por 2 variables, nombre de la línea celular y la ID del DepMap, y 969 observaciones.

El próximo paso ha sido la lectura del archivo "CCLE_depMap_19Q1_TPM.csv", mediante la función fread(), del paquete “tdyr”, que consta de 1165 observaciones y 57821 variables (57820 genes de expresión y la ID associada al DepMap).

Una vez leído el archivo, he fusionado éste, mediante la función merge(), con el dataframe creado en el paso anterior por las columnas en común de la ID asociada al DepMap y obteniéndo así un dataframe de 656 observaciones, que consisten en los nombres de las líneas celulares tumorales asociadas a esas IDs del DepMap, y 57820 variables, correspondientes a los nombres de los genes de expresión.
Lo siguiente ha sido fusionar el dataframe que contiene las líneas celulares y los genes de expresión, con el dataframe de respuesta a fármacos, el cual fue guardado como se indica anteriormente ("farmaco.rds"), mediante la función merge() y obteniendo un dataframe, en el que las filas serán las líneas celulares tumorales que poseen en común tanto los genes de expresión del dataframe que los contenía como los valores AUC asociados al fármaco escogido contenidos en el archivo "farmaco.rds".
Las observaciones de este dataframe varían según el fármaco escogido, ya que cada fármaco tiene unas líneas celulares concretas en las que ha sido probado y unos valores de expresión asociados a estas líneas celulares tumorales, aunque este siempre tendra 57821 variables (57820 genes de expresión y una columna con los valores AUC).

Por último, con la función sapply(), implementada en el software base de R, he eliminado los posibles Nas y guardado este dataframe, mediante la función saveRDS(), con el nombre "finalmatrix.rds" para seguir usándola en los distintos modelos implementados en este proyecto sin necesidad de tener que ejecutar todo el código ejecutado hasta ahora.
Estos Nas son generados tras la última fusión de dataframes debido a que al conservarse como variables los nombres de los genes de expresión se generan huecos vacios en aquellas observaciones en las que el fármaco no ha sido probado en la linea celular tumoral a la que pertenece el gen expresado en cuestión.

Tipo de datos:

Dependiendo de la naturaleza de los datos se ha seguido un desarrollo de análisis u otro. Los modelos según el tipo de dato son:

1. Modelos Continuos:

Si las predicciones se realizan en base a mediciones de resumen de respuesta continuas, es decir, mediciones con números reales. En mi caso los valores Auc del dataframe "finalmatrix.rds".
Para la preparación de estos datos he leido el archivo "finalmatrix.rds" con la función readRDS() y he filtrado el dataframe para seleccionar los 200 genes de expresión con las varianzas más altas (medida escogida por mi tutor en el Cnio al azar y no muy grande para seleccionar las columnas con suficiente varianza para ser informativos en la predicción) para agilizar la ejecución de código mediante las funciones: sort(), del paquete "DescTool", para clasificar por orden ascendiente o descendiente las varianzas obtenidas mediante la función colVars(), del paquete "resample"; y names() y subset(), implementados en el software base de R, con los que he seleccionado los nombres de los 200 genes de expresión con las varianzas más altas y subdividido el dataframe inicial seleccionando solo las columnas con estos nombres.
El resultado final de este preprocesamiento será un dataframe con 201 variables (los 200 genes con las varianzas más altas y 1 variable con los valores AUC que he usado posteriormente como predictores) a diferencia de las 57821 variables con las que partíamos inicialmente.

2. Modelos Categóricos:

Si las predicciones se realizan en base a mediciones de resumen de respuesta categóricas, es decir, mediciones dicotómicas. En mi caso con los valores Auc del dataframe "finalmatrix.rds", previa discretización de sus valores continuos, siguiendo los parámetros indicados (sensible =< 3º quartil, resistente < 3º quartil de los datos). Es decir, tras realizar el mismo proceso de preprocesamiento de datos que para los modelos continuos, he calculado mediante la function quantile(), implementado en el software base de R, el tercer cuartil de los valores AUC para

despúes clasificar estos valores en sensibles o resistentes dependiendo de si se encuentran por encima o por debajo del tercer cuartil respectivamente y transformando este vector generado en un vector de clase factor mediante la function factor(), implementado en el software base de R, obteniendo así mediciones de respuesta discretizadas.

Partición de datos:

Los algoritmos empleados en Machine learning funcionan haciendo predicciones o decisiones basadas en datos mediante la construcción de un modelo matemático a partir de datos de entrada.

La partición de datos consiste en la división en varios conjuntos de datos o Sets, que siguen la misma distribución de probabilidad, los datos obtenidos de estos datos de entrada (en mi caso el conjunto de datos utilizado es el generado anteriormente con el nombre de "finalmatrix.rds").

En particular, tres conjuntos de datos o sets se utilizan comúnmente en las diferentes etapas de la creación del modelo:

1. **Training Set o conjunto de datos de entrenamiento:** un conjunto de ejemplos utilizados para el aprendizaje del modelo.

2. **Validation Set o conjunto de datos de validación:** un conjunto de ejemplos utilizados para ajustar los hiperparámetros de un modelo.

3. **Test Set o conjunto de datos de prueba:** es un conjunto de datos que es independiente del conjunto de datos de entrenamiento, pero que sigue la misma distribución de probabilidad que el conjunto de datos de formación y se utiliza únicamente para evaluar el rendimiento de un clasificador completamente especificado.

Si un modelo que se ajusta al conjunto de datos de entrenamiento también se ajusta bien al conjunto de datos de prueba, se ha producido un sobreajuste mínimo.

Un mejor ajuste del conjunto de datos de formación en comparación con el conjunto de datos de prueba suele indicar un sobreajuste.

Entre todos los tipos de particiones he escogido 2 por sus características diferenciales:

Tras los preprocesamientos, tanto para modelos continuos como categóricos, he realizado la partición de datos de dos formas distintas:

1. 80/20 (con 10 iteraciones en validación cruzada):

Consiste en la división del conjunto de datos en un 80% de estos para formar el training set y el 20% restante para formar el test set.
En este caso no hay datos de validación pues he usado la en validación cruzada con 10 iteraciones para el ajuste de hiperparámetros.

Este tipo de validación cruzada es utilizada para evaluar los resultados de un análisis estadístico y garantizar que son independientes de la partición entre datos de entrenamiento y prueba.
Consiste en repetir y calcular la media aritmética obtenida de las medidas de evaluación sobre diferentes particiones, en este caso 10. Se utiliza en entornos donde

el objetivo principal es la predicción y se quiere estimar la precisión de un modelo que se llevará a cabo a la práctica como es uno de los objetos principales de este proyecto.

Los datos de muestra se dividen en 10 subconjuntos. Uno de los subconjuntos se utiliza como datos de prueba y el resto como datos de entrenamiento. El proceso de validación cruzada es repetido durante 10 iteraciones, con cada uno de los posibles subconjuntos de datos de prueba. Finalmente se realiza la media aritmética de los resultados de cada iteración para obtener un único resultado. Este método es muy preciso puesto que evaluamos a partir de K combinaciones de datos de entrenamiento y de prueba, pero aun así tiene una desventaja y es que es lento desde el punto de vista computacional.

En este tipo de partición he dividido los datos de train y test en la proporción indicada previamente mendiante las funciones sample.split(), obtenida del paquete "caTools", y subset() consiguiendo los train y test sets con las proporciones deseadas e indicando como variable predictora o clase la columna de los valores AUC.

2. 60/20/20:

Consiste en la división del conjunto de datos en un 80% de estos para formar el training set, de los cuales el 20% serán escogidos posteriormente para formar el set de validación, y el 20% restante para formar el test set.
Para conseguir este tipo de partición he usado la funcion sample(), implementado en el software base de R, para dividir los datos en un 80% (training set), los cuales he usado para volver a dividir mediante el mismo método en un 60% y 20% (este último utilizado como validation set), y 20% (test set).
Para comprobar que las particiones se han realizado correctamente he usado la funcion reduce(), obtenido del paquete "purr".

- Búsqueda de hiperparámetros:

Los hiperparámetros de un modelo son una característica de este que externa al modelo y cuyos valores no pueden ser estimados a partir de los datos. El valor del hiperparámetro debe ajustarse antes de que comience el proceso de aprendizaje.

En contraste, un parámetro es una característica interna del modelo y su valor si puede ser estimado a partir de los datos.

Sabiendo esto, para el mejor ajuste en la aplicación de los modelos tanto discretos como continuos he seleccionado 3 formas de búsqueda de hiperparámetros para comparar con cual se obtendrían mejores valores de "Accuracy" o "Rsme" dependiendo de si este modelo es discreto o continuo respectivamente:

H2O: "es un producto creado por la compañía H2O.ai con el objetivo de combinar los principales algoritmos de machine learning y aprendizaje estadístico con el Big Data. Gracias a su forma de comprimir y almacenar los datos, H2O es capaz de trabajar con millones de registros en un único ordenador, empleando todos sus núcleos, o en un cluster de muchos ordenadores. Internamente, H2O está escrito en Java y sigue el paradigma *Key/Value* para almacenar los datos y *Map/Reduce* para sus algoritmos. Gracias a sus API, es posible acceder a todas sus funciones desde R**,** Python o Scala (lenguajes de programación)**,** así como por una interfaz web llamada Flow. Aunque la principal ventaja de H2O frente a otras herramientas es su escalabilidad, sus algoritmos son igualmente útiles cuando se trabaja con un volumen de datos reducido" [71].

"El manejo de H2O puede hacerse íntegramente desde R: iniciar el cluster, carga de datos, entrenamiento de modelos, predicción de nuevas observaciones, etc. Es importante tener en cuenta que, aunque los comandos se ejecuten desde R, los datos

se encuentran en el cluster de H2O, no en memoria. Solo cuando los datos se cargan en memoria, se les pueden aplicar funciones propias de R" [71].

1. **Random search:** La búsqueda aleatoria (RS) es una familia de métodos de optimización numérica que no requieren que se optimice el gradiente del problema por lo que se puede usar tanto en datos categorizados como continuos.

2. **Grid search:** Consiste en una búsqueda exhaustiva a través de un subconjunto especificado manualmente del espacio de hiperparámetros de un algoritmo de aprendizaje. Un algoritmo de Grid Search debe estar guiado por alguna métrica de rendimiento, normalmente medida por la validación cruzada en el conjunto de entrenamiento o por la evaluación en un conjunto de validación retenido, ambas técnicas probadas.

Modelos:

1. Random forest:

También conocidos en castellano como "'Bosques Aleatorios"', es una combinación de árboles predictores tal que cada árbol depende de los valores de un vector aleatorio probado independientemente y con la misma distribución para cada uno de estos. Es una modificación sustancial de bagging.

La idea esencial del bagging es promediar muchos modelos ruidosos pero aproximadamente imparciales, y por tanto reducir la variación. Los árboles son los candidatos ideales para el bagging, dado que ellos pueden registrar estructuras de interacción compleja en los datos que tienen relativamente baja parcialidad.

Cada árbol es construido usando el siguiente algoritmo:

"El algoritmo de capacitación para bosques aleatorios aplica la técnica general de la agregación de bootstrap. Dado un conjunto de entrenamiento X = x1, ..., xn con respuestas Y = y1,..., yn, "folds" repetidamente (B veces) selecciona una muestra aleatoria con sustitución del conjunto de entrenamiento y se ajusta a los árboles de estas muestras" [81]:

"Para b = 1, ..., B:

1. Ejemplo, con reemplazo, n ejemplos de entrenamiento de X, Y; llamémosle Xb, Yb.
2. Entrenar un árbol de clasificación o regresión f sub b en Xb, Yb" [81].

"Después del entrenamiento, se pueden hacer predicciones para muestras no vistas x' promediando las predicciones de todos los árboles de regresión individuales en x"" [81]:

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} f_b(x')$$

Ilustración 1 : Fórmula Random Forest

o por mayoría de votos en el caso de los árboles de clasificación" [81].

"Este procedimiento de bootstrapping conduce a un mejor rendimiento del modelo porque disminuye la varianza del modelo, sin aumentar el sesgo. Esto significa que mientras que las predicciones de un solo árbol son altamente sensibles al ruido en su conjunto de entrenamiento, el promedio de muchos árboles no lo es, siempre y cuando los árboles no estén correlacionados. El simple hecho de entrenar muchos árboles en un solo conjunto de entrenamiento daría árboles fuertemente correlacionados (o incluso el mismo árbol muchas veces, si el algoritmo de entrenamiento es determinista); el muestreo de arranque es una forma de descorrelacionar los árboles mostrándoles diferentes conjuntos de entrenamiento" [81].

"Además, puede hacerse una estimación de la incertidumbre de la predicción como la desviación estándar de las predicciones de todos los árboles de regresión individuales en x[81]:

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}(f_b(x') - \hat{f})^2}{B-1}}.$$ "[81].

Ilustración 2 : Estimación de la incertidumbre

"El número de muestras/árboles, B, es un parámetro libre. Típicamente, se utilizan de varios cientos a varios miles de árboles, dependiendo del tamaño y la naturaleza del conjunto de entrenamiento. Un número óptimo de árboles B se puede encontrar usando la validación cruzada, u observando el error fuera de iteraciones en: el error de predicción medio en cada muestra de entrenamiento x_i, usando sólo los árboles que no tenían x_i en su muestra de bootstrap. El error de entrenamiento y prueba tiende a nivelarse después de que un cierto número de árboles han sido colocados" [81].

"El procedimiento anterior describe el algoritmo original de entrenamiento de árboles. Los bosques aleatorios difieren de una sola manera de este esquema general: utilizan un algoritmo de aprendizaje en árbol modificado que selecciona, para cada candidato dividido en el proceso de aprendizaje, un subconjunto aleatorio de las características. La razón de hacer esto es la correlación de los árboles en una muestra de arranque ordinaria: si una o unas pocas características son predictores muy fuertes para la variable de respuesta (salida objetivo), estas características serán seleccionadas en muchos de los árboles B, causando que se correlacionen. Un análisis de cómo el entrenamiento y la proyección subespacial aleatoria contribuyen a las ganancias de precisión bajo diferentes condiciones es dado por Ho" [81].

Sabiendo esto y para aplicar en R este modelo se ha utilizado la función randomforest(), del paquete "randomForest", para entrenar los datos de entrenamiento divididos anteriormente en las distintas particiones de datos con los hiperparámetros obtenidos previamente y mediante:

- **H2o:**
  En este método de busqueda de hiperparámetros he usado integramente el paquete "H2o" para todas sus funciones siendo estas:
  - **H2o.int():** inicia el software.
  - **as.H2o():** para transformar los datos de entrenamiento en un dataframe.
  - **Setdiff():** para seleccionar el vector clase necesario para el entrenamiento.
  - **List():**
    a) para crear una lista indicando los intervalos de busqueda de cada hiperparámetro:

    1. Ntrees: Número de árboles a crecer. Esto no debe ser ajustado a un número demasiado pequeño, para asegurar que cada fila de

entrada sea pronosticada por lo menos unas cuantas veces. (Intervalo: 200:500 escogiendo valores de 150 en 150).

2. Mtries: Número de variables muestreadas aleatoriamente como candidatas en cada división. (Intervalo: 15:35, de 10 en 10 valores).

3. Max_depth: profundidad máxima de rango del bosque generado (Intervalo: 20:40, de 5 en 5 valores).

4. Mins_rows: minimo numero de files usadas por modelo (Intervalo: 1:5, de 2 en 2 valores).

5. Nibns: el número de ubicaciones que deben incluirse en el histograma y luego dividirse de la mejor manera posible. (Intervalo: 10:30, de 5 en 5 valores).

6. Sample_rate: especifica que cada árbol del conjunto debe muestrear (sin reemplazar) del conjunto completo de datos de formación utilizando una tasa de muestreo específica por clase en lugar de un factor de muestreo global. (Valores: 0.55, 0.632, 0.75).

b) para crear una lista indicando los criterios de busqueda:

1. strategy: tipo de modelo dependiendo si es discreto o continuo.

2. Stopping_metric: si es discreto: “Accuracy”; o continuo “Rmse”.

3. Stopping_tolerance: Esta opción especifica el valor de tolerancia por el cual un modelo debe mejorar antes de que cese el entrenamiento. En mi caso con un valor de 0.005.

4. stopping_rounds: para detener el modelo de entrenamiento cuando la opción seleccionada para stop_metric no mejora para este número especificado de rondas de entrenamiento, basado en una media móvil simple. En mi caso con un valor de 10.

5. max_runtime_sec: esta opción especifica el tiempo de ejecución máximo en segundos que desea asignar para completar el modelo. Si se excede este tiempo de ejecución máximo antes de que se complete la construcción del modelo, entonces el modelo fallará. En mi caso con un valor de 30*60.

- **H2o.grid():** para entrenar el modelo indicando los parámetros:

1. Algorithm: randomForest en mi caso.

2. Grid_id: indicando el nombre del otuput resultante.

3. Training_frame: para indicar los datos de entrenamiento que se van a usar.

4. Hypers_params: para indicar el intervalo de hiperpparámetros a probar.

5. Search_criteria: para indicar los criterios de busqueda seleccionados.

- **H2o.Grid():** para guardar los mejores hiperparámetros obtenidos en la busqueda indicando en los parámetros:

    1. Grid_id: el nombre de la id que hemos reservado para nuestro modelo.

    2. Sort_by: ordenar los resultados obtenidos de la busqueda por "Accuracy" para modelos discretos y "Rmse" para continuos.

    3. Decreasing: para indicar si queremos que la lista de resultados este ordenada de mayor a menor y viceversa.

Para ambas formas de partición de datos realizadas en este proyecto el procedimiento ha sido el mismo, con la única diferencia que en la partición 60/20/20 los datos para la búsqueda de hiperparámetros han sido los de validación, sin validación cruzada, en vez de los de entrenamiento.

- **Random Search:**

En este método de búsqueda de hiperparámetros he seleccionado el parámetro tunelength= 20, un parámetro control (trcontrol) y el parámetro de métrica "Accuracy" en modelos discretizados y "Rmse" en modelos continuos.

El parámetro control sirve para indicar, mediante la funcion traincontrol() del paquete "caret", que deseo que este control sea un 10 iteraciones en validación cruzada repetido 5 veces, para la función train(), obtenida del paquete "caret".

En el parámetro "tunelength" indico que quiero obtener los 20 mejores parámetros principales o más relevantes obtenidos en las distintas iteraciones y repeticiones del modelo, dependiendo de si es continuo (mtry, RMSE, Rsquared y Mae) o discretizado (mtry, Accuracy y Kappa).

Tras este tipo de búsqueda de hiperparámetros para el ajuste del modelo de Random Forest, he realizado el entrenamiento del modelo con la función random forest(), del paquete "RandonForest", especificando en esta el parámetro mtry (Número de variables muestreadas aleatoriamente como candidatas en cada división) más optimo obtenido en la búsqueda.

Para ambas formas de partición de datos realizadas en este proyecto el procedimiento ha sido el mismo, con la única diferencia que en la particion 60/20/20 los datos para la búsqueda de hiperparámetros han sido los de validación, sin validación cruzada, en vez de los de entrenamiento.

2. Regularización de Ridge:

Es muy similar a la regresion de los mínimos cuadrados (el procedimiento de ajuste de mínimos cuadrados estima β0, β1,...,βp utilizando los valores que minimizan),

$$\text{RSS} = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 .$$

Ilustración 3 : Fórmula mínimos cuadrados

exceptuándo que los coeficientes de Ridge se estiman minimizando una cantidad ligeramente diferente. En particular, las estimaciones del coeficiente de regresión de Ridge β̂R son los valores que minimizan:

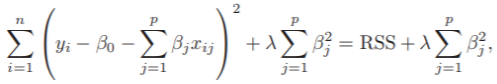


Ilustración 4 : Fórmula Ridge

donde $\lambda \geq 0$ es un parámetro de tuning, que se determinará por separado. La ecuación compensa dos criterios diferentes. La regresión de Ridge busca estimaciones de coeficientes que se ajusten bien a los datos, haciendo el RSS pequeño. Sin embargo, el segundo término, $\lambda \sum_j \beta_j^2$,

Ilustración 5 : Penalización de contracción

llamado penalización por contracción, es pequeño cuando β1,...,βp están cerca de cero, por lo que tiene el efecto de reducir las estimaciones de βj a cero. El parámetro de ajuste λ sirve para controlar el impacto relativo de estos dos términos en las estimaciones del coeficiente de regresión. Cuando λ = 0, el término de la penalización no tiene efecto, y la regresión de Ridge producirá las estimaciones de los mínimos cuadrados. La selección de un buen valor para λ, por tanto, es crítica.
.

Para la busqueda de hiperparámetros en el ajuste de los modelos de regresion de Ridge he usado únicamente la función cv.glmnet(), del paquete "glmnet", debido al tiempo invertido durante el desarrollo del primer algoritmo que he probado, random Forest, en los modelos con particiones 80/20 con 10 iteraciones en validación cruzada.
Los parámetros especificados en esta función para el ajuste del modelo han sido:

1. **Family:** indica el tipo de regresión. En nuestro caso binomial para discretizado y por defecto en continuos.

2. **Alpha:** 0 (valor que corresponde a la regresión Ridge).

3. **Type.measure:** tipo de mediación a especificar dependiendo de si es discretizado (auc) o continuo (mse).

4. **Niteraciones ens:** número de pliegues. Al ser 10 iteraciones en validación cruzada será 10.

Para los modelos con particiones 60/20/20 he usado grid search para la busqueda de hiperparámetros mediante el set de validación, gracias a la funcion expand.grid() del paquete "caret", para más tarde aplicar estos hiperparámetros al entrenamiento del modelo ajustado con la funcion glmnet(), del paquete "glmnet", con los datos del set de entrenamiento.

Los hiperparámetros indicados en la grid search han sido:

- **Lambda:** los valores indicados en el intervalo han sido de 10 R^2 -3 a 3 siendo este intervalo de longitud 100.
- **Alpha:** 0 para la regresión de Ridge.

Los parámetros indicados en la función glmnet() son:

- **Family:** binomial en el caso de los modelos con predictores discretizados y por defecto en los continuos.
- **Alpha:** 0 para la regresión de Ridge.
- **Lambda:** obtenida de la búsqueda mediante grid search.

3. Lasso:

Lasso, o Least Absolute Shrinkage and Selection Operator, es muy similar conceptualmente a la regresión de Ridge. También añade una penalización por coeficientes distintos de cero, pero a diferencia de la regresión de Ridge que penaliza la suma de coeficientes cuadrados (penalización L2), LASSO penaliza la suma de sus valores absolutos (penalización L1). Como resultado, para valores altos de λ, muchos coeficientes se ponen exactamente a cero con LASSO, lo que nunca se da en el caso de la regresión de Ridge.

La única diferencia entre las funciones de LASSO y de Ridge está en los términos de la penalización. En LASSO, la pérdida se define como:

$$L_{lasso}(\hat{\beta}) = \sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2 + \lambda \sum_{j-1}^{m}|\hat{\beta}_j|.$$

Ilustración 6 : Fórmula Lasso

Para la búsqueda de hiperparamétros y entrenamiento del modelo se ha seguido el mismo procedimiento que para la regresión de Ridge aunque cambiando el valor alpha por el que corresponde a las regresiones de LASSO, alpha =1.

4. ElasticNet:

Elastic Net surgió por primera vez como resultado de la crítica al método LASSO, cuya selección de variables puede ser demasiado dependiente de los datos y, por lo tanto, inestable. La solución es combinar las penalizaciones de regresión de Ridge y LASSO para obtener lo mejor de ambos. El objetivo de Elastic Net es minimizar la siguiente función de pérdida:

$$L_{enet}(\hat{\beta}) = \frac{\sum_{i-1}^{n}(y_i - x_i'\hat{\beta})^2}{2n} + \lambda(\frac{1-\alpha}{2}\sum_{j-1}^{m}\hat{\beta}_j^2 + \alpha\sum_{j-1}^{m}|\hat{\beta}_j|),$$

Ilustración 7 : Fórmula Elastic Net

donde α es el parámetro de mezcla entre Ridge (α = 0) y LASSO (α = 1).

Ahora, hay dos parámetros para tunear: λ y α.

5. Support Vector Machine (SVM) Linear:

También llamadas redes vectoriales de apoyo, son modelos de aprendizaje supervisados con algoritmos de aprendizaje asociados que analizan los datos utilizados para la clasificación y el análisis de regresión. Dado un conjunto de ejemplos de entrenamiento, cada uno marcado como perteneciente a una u otra de dos categorías, un algoritmo de entrenamiento SVM construye un modelo que asigna nuevos ejemplos a una u otra categoría, convirtiéndolo en un clasificador lineal binario no probabilístico. Un modelo SVM es una representación de los ejemplos como puntos en el espacio, mapeados de manera que los ejemplos de las categorías separadas se dividen por una brecha clara y lo más amplia posible. Los nuevos ejemplos se mapean en ese mismo espacio y se predice que pertenecen a una categoría basada en el lado del hueco en el que caen.

Para la busqueda de hiperparamétros he usado la función tune(), del paquete "e1071", indicando el modelo de entrenamiento, en este caso "svm", los datos de entrenamiento, el kernel (el kernel usado en el entrenamiento y la predicción, en este caso linear, y el

rango con una lista de los hiperparámetros a evaluar, en este caso cost (violación del coste de las restricciones), mediante la funcion de creación de listas list().

Una vez deducidos los mejores hiperparámetros se ha entrenado el modelo ajustado mediante la funcion svm(), del paquete "e1071", indicando los datos de entrenamiento, el kernel ("linear"), la family o família (binomial en caso discretizado), el valor de cost obtenido en la búsqueda de hiperparámetros y el número de validaciones cruzadas que va a realizar (en el caso de la 10 iteraciones en validación cruzada).

Creación de gráficas:

1. Modelos categóricos:

Para representar los datos obtenidos en el ajuste de los modelos categóricos he escogido:

a. Matriz de confusión: es una herramienta que permite la visualización del proceso obtenido de un algoritmo de clasificación que se emplea en aprendizaje supervisado. Cada columna de la matriz representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. Uno de los beneficios de las matrices de confusión es que facilitan ver si el sistema está confundiendo dos clases o no. Como medida orientativa de precisión de predicción ha sido escogido el parámetro orientativo "Accuracy" que se muestra justo debajo de la matriz propiamente dicha [4-9].

Para su obtención hemos realizado una predicción de los test set con los modelos ajustados mediante la función predict(), perteneciente al paquete "stats", para a continuación representar la matriz de confusión mediante la función confusiónMatrix(), obtenida del paquete "caret".

b. Curva Roc: es una representación gráfica de la sensibilidad frente a la especificidad para un sistema clasificador binario según se varía el umbral de discriminación. Otra interpretación de este gráfico es la representación de la razón o ratio de verdaderos positivos (VPR = Razón de Verdaderos Positivos) frente a la razón o ratio de falsos positivos (FPR = Razón de Falsos Positivos) también según se varía el umbral de discriminación (valor a partir del cual decidimos que un caso es positivo).

   El mejor método posible de predicción se situaría en un punto en la esquina superior izquierda, o coordenada (0,1) del espacio ROC, representando un 100% de sensibilidad (ningún falso negativo) y un 100% también de especificidad (ningún falso positivo). A este punto (0,1) también se le llama una clasificación perfecta. Por el contrario, una clasificación totalmente aleatoria (o adivinación aleatoria) daría un punto a lo largo de la línea diagonal, que se llama también línea de no-discriminación, desde el extremo inferior izquierdo hasta la esquina superior derecha (independientemente de los tipos de base positivas y negativas).

   Para su obtención hemos realizado una predicción de los test set con los modelos ajustados anteriormente y especificando que la predicción sea de tipo probabilidad mediante la función predict() y el parámetro type="prob". A continuación mediante la función prediction(), perteneciente al paquete "ROCR", he seleccionado la segunda columna obtenida de la ultima predicción que corresponde a la clase sensible, para

con los valores categorizados del Testset, se represente a través de las funciones performance(), del paquete “ROCR”, y plot(), incluido en el software base de R, la curva Roc resultado.

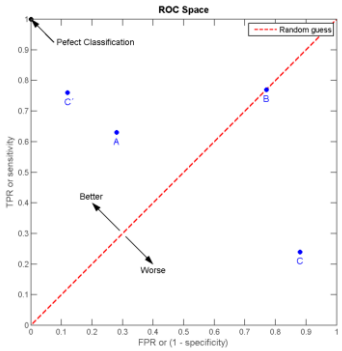


Ilustración 8 : Curva ROC

2. Modelos continuos:

Para representar los datos obtenidos en el ajuste de los modelos continuos he escogido la representación gráfica de R^2 y la distribución de puntos obtenidos sobre el hiperplano: encuentra una "superficie" que intenta separar los ejemplos negativos y positivos con el margen más grande posible a ambos lados del hiperplano. En este caso, bi-dimensional, la "superficie" es una línea.

Hay muchas formas de hacer este tipo de representaciones pero me parece más fàcil de interpretar de este modo.
Lo que distingue a este tipo de representaciones es que el hiper-plano resultante se consigue logrando que el margen que separa los datos sea el mayor posible.

Mientras más lejos esté el hiper-plano de los puntos a los que clasifica, mejor.
Asociando estas ideas con la R^2 se obtiene una medida de precisión o bondad del modelo siendo este mejor ajustado o con mejor precisión cuanto mayor sea su valor. Estos se encuentran entre 0 y 1[4-9].

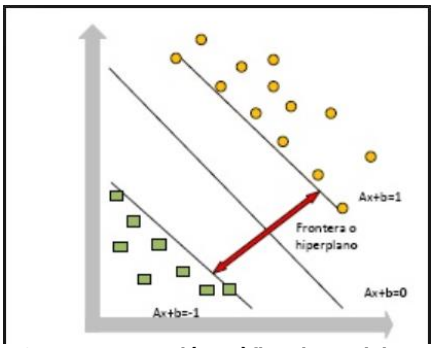


Ilustración 9 : Representación gràfica de modelos continuos.

Para su obtención he realizado una predicción de los test set con los modelos ajustados mediante la función predict() para a continuación mediante la función rsq(), definida por comodidad por mí, calcular los valores de R^2 asociados al modelo ajustado con los datos del test set y representar gráficamente estos valores en una gráfica mediante la función ggplot(), del paquete “ggplot2” representando así la precisión del modelo o su validez.

Resultados:

Búsqueda de hiperparámetros:

Los resultados obtenidos en las distintas búsquedas de hiperparámetros son los siguientes:

Random forest H2o:

En la búsqueda de hiperparámetros para el algoritmo random Forest, mediante el software de H2o, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en validación cruzada:

```
Number of models: 639
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
  max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1        40      1.0     15    25    200        0.75 rf_grid2_model_635  0.5090909090909090
2        20      1.0     25    25    200       0.632 rf_grid2_model_622  0.5333333333333333
3        35      3.0     35    15    500        0.75 rf_grid2_model_600  0.5333333333333333
4        30      5.0     35    20    200        0.75 rf_grid2_model_137  0.5393939393939393
5        30      5.0     35    15    500       0.632 rf_grid2_model_487  0.5393939393939393

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
634        35      5.0     15    15    350       0.632 rf_grid2_model_147                0.8
635        20      5.0     15    20    200       0.632 rf_grid2_model_127  0.8060606060606060
636        30      3.0     15    15    350        0.55 rf_grid2_model_167  0.8060606060606060
637        25      1.0     15    25    350        0.75 rf_grid2_model_573 0.8121212121212121
638        35      5.0     25    30    200        0.55  rf_grid2_model_43 0.8181818181818181
639        35      5.0     15    10    500       0.632 rf_grid2_model_602 0.8242424242424242
```

Ilustración 10 : Erlotinib

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 575
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1          20      1.0     15    20    350        0.75 rf_grid2_model_124 0.2514970059880239
2          30      1.0     25    30    200        0.75 rf_grid2_model_365 0.2634730538922155
3          20      1.0     35    10    200        0.75 rf_grid2_model_370 0.2634730538922155
4          25      1.0     35    20    200        0.75  rf_grid2_model_16 0.2754491017964071
5          30      1.0     35    30    350        0.75  rf_grid2_model_23  0.329341317365269

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
570        35      1.0     35    20    500        0.55 rf_grid2_model_511 0.7784431137724551
571        25      1.0     35    20    500       0.632 rf_grid2_model_477 0.7844311377245509
572        35      3.0     35    10    200        0.55 rf_grid2_model_424 0.7844311377245509
573        20      5.0     15    10    350        0.75 rf_grid2_model_133 0.7844311377245509
574        20      3.0     35    10    350       0.632 rf_grid2_model_296 0.7964071856287425
575        20      1.0     35    15    200       0.632 rf_grid2_model_381 0.8083832335329342
```

Ilustración 11 : Rapamycin

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 131
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1          25      1.0     35    30    200        0.75 rf_grid2_model_122  0.4184782608695652
2          30      1.0     35    15    500       0.632  rf_grid2_model_37  0.4565217391304348
3          25      1.0     25    20    200        0.55  rf_grid2_model_27  0.4565217391304348
4          40      3.0     15    10    350       0.632  rf_grid2_model_62  0.4891304347826086
5          30      3.0     35    20    200        0.75  rf_grid2_model_50  0.4945652173913043

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
126        40      5.0     25    30    500        0.75  rf_grid2_model_35 0.7771739130434783
127        40      3.0     25    10    200       0.632 rf_grid2_model_125  0.782608695652174
128        40      3.0     25    30    200        0.75  rf_grid2_model_69  0.7880434782608696
129        20      3.0     25    10    350        0.55 rf_grid2_model_136  0.7934782608695652
130        35      5.0     25    15    350       0.632  rf_grid2_model_85  0.7934782608695652
131        20      1.0     25    30    500        0.55  rf_grid2_model_25  0.7989130434782609
```

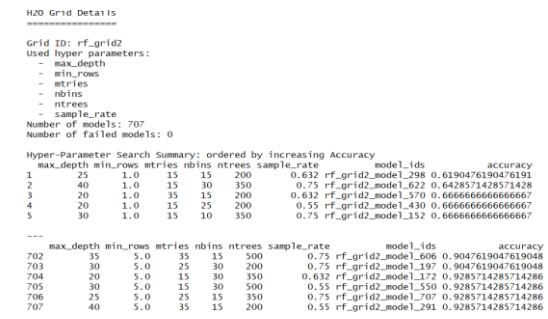
Ilustración 12 : Sunitinib

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 507
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1          20      1.0     35    15    200        0.75 rf_grid2_model_368  0.4347826086956522
2          40      1.0     35    15    200        0.75 rf_grid2_model_211  0.4402173913043478
3          25      1.0     15    25    350       0.632 rf_grid2_model_473 0.4456521739130434
4          35      1.0     15    30    200       0.632 rf_grid2_model_131 0.4510869565217391
5          35      1.0     35    25    200        0.75 rf_grid2_model_257  0.4565217391304348

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
502        30      1.0     25    25    500       0.632 rf_grid2_model_106 0.8097826086956521
503        35      5.0     35    30    500        0.55 rf_grid2_model_446 0.8152173913043479
504        35      5.0     35    15    200       0.632 rf_grid2_model_442 0.8152173913043479
505        20      5.0     25    25    350        0.55 rf_grid2_model_495 0.8206521739130435
506        40      3.0     35    10    500        0.55 rf_grid2_model_339 0.8206521739130435
507        35      5.0     35    10    200        0.55 rf_grid2_model_395 0.8260869565217391
```

Ilustración 13 : Paclitaxel

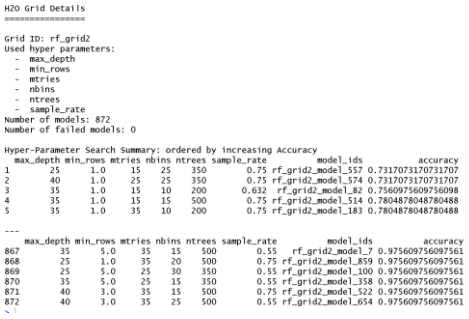
Se puede observar que, para la búsqueda de hiperpárametros con H2o en Random Forest con los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 11 a 14), el output nos devuelve, ordenados de forma ascendente por el valor “Acuracy”, que es el valor indicativo de la precisión de los modelos entrenados con predictores discretizados, la mejor combinación de hiperparámetros obtenidos, siendo en este caso el modelo número 639 para el Erlotinib, el 573 para el Rampamycin, el 507 para el Sunitinib y el 131 para el paclitaxel.

2. Clasificación 60/20/20:



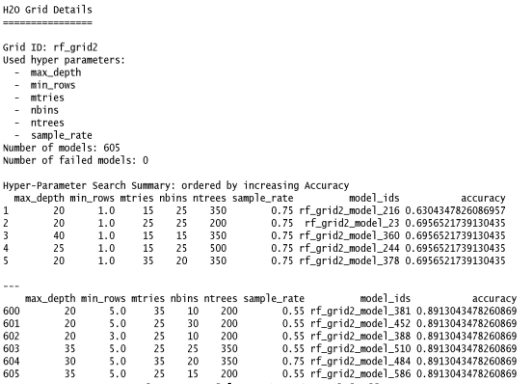
```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 707
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1          25      1.0     15    15    200       0.632 rf_grid2_model_298 0.6190476190476191
2          40      1.0     15    30    350       0.75 rf_grid2_model_622 0.6428571428571428
3          20      1.0     35    15    200       0.632 rf_grid2_model_570 0.6666666666666667
4          20      1.0     15    25    200       0.55 rf_grid2_model_430 0.6666666666666667
5          30      1.0     15    10    350       0.75 rf_grid2_model_152 0.6666666666666667

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
702        35      5.0     35    15    500       0.75 rf_grid2_model_606 0.9047619047619048
703        30      5.0     25    30    200       0.75 rf_grid2_model_197 0.9047619047619048
704        20      5.0     15    30    350       0.632 rf_grid2_model_172 0.9285714285714286
705        30      5.0     15    30    500       0.55 rf_grid2_model_550 0.9285714285714286
706        25      5.0     25    15    350       0.75 rf_grid2_model_707 0.9285714285714286
707        40      5.0     35    15    200       0.55 rf_grid2_model_291 0.9285714285714286
```

Ilustración 14 : Erlotinib



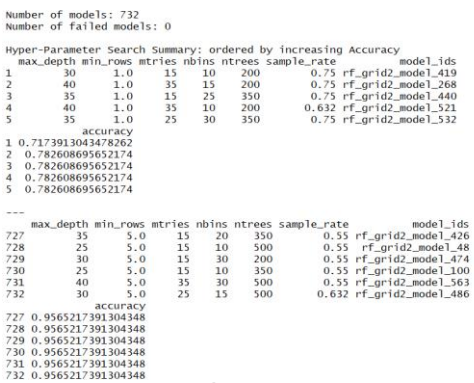
```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 872
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1          25      1.0     15    25    350       0.75 rf_grid2_model_557 0.7317073170731707
2          40      1.0     25    25    350       0.75 rf_grid2_model_574 0.7317073170731707
3          35      1.0     15    10    200      0.632  rf_grid2_model_82 0.7560975609756098
4          35      1.0     15    15    500       0.75 rf_grid2_model_514 0.7804878048780488
5          35      1.0     35    10    200       0.75 rf_grid2_model_183 0.7804878048780488

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
867        35      5.0     35    15    500       0.55  rf_grid2_model_7 0.9756097560975610
868        25      1.0     35    20    500       0.75 rf_grid2_model_859 0.9756097560975610
869        25      5.0     25    30    350       0.55 rf_grid2_model_100 0.9756097560975610
870        35      5.0     25    15    350       0.55 rf_grid2_model_358 0.9756097560975610
871        40      3.0     35    15    500       0.75 rf_grid2_model_522 0.9756097560975610
872        40      3.0     35    25    500       0.55 rf_grid2_model_654 0.9756097560975610
```

Ilustración 15 : Rapamycin



```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 605
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
  max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
1        20      1.0     15    25    350       0.75 rf_grid2_model_216 0.6304347826086957
2        20      1.0     25    25    200       0.75  rf_grid2_model_23 0.6956521739130435
3        40      1.0     15    15    350       0.75 rf_grid2_model_360 0.6956521739130435
4        25      1.0     15    25    500       0.75 rf_grid2_model_244 0.6956521739130435
5        20      1.0     35    20    350       0.75 rf_grid2_model_378 0.6956521739130435

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids           accuracy
600        20      5.0     35    10    200       0.55 rf_grid2_model_381 0.8913043478260869
601        20      5.0     25    30    200       0.55 rf_grid2_model_452 0.8913043478260869
602        20      3.0     25    10    200       0.55 rf_grid2_model_388 0.8913043478260869
603        35      5.0     25    25    350       0.55 rf_grid2_model_510 0.8913043478260869
604        30      5.0     35    20    350       0.75 rf_grid2_model_484 0.8913043478260869
605        35      5.0     25    15    200       0.55 rf_grid2_model_586 0.8913043478260869
```

Ilustración 16 : Sunitinib

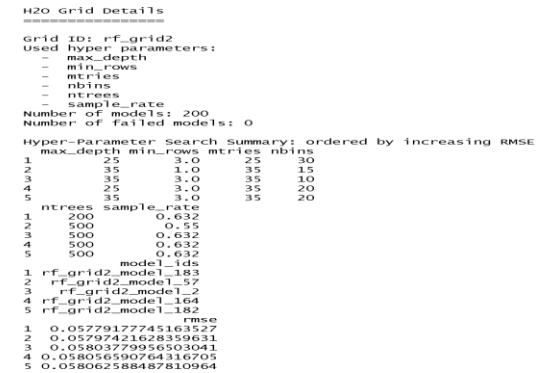


```
Number of models: 732
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing Accuracy
  max_depth min_rows mtries nbins ntrees sample_rate          model_ids
1        30      1.0     15    10    200       0.75 rf_grid2_model_419
2        40      1.0     35    15    200       0.75 rf_grid2_model_268
3        35      1.0     15    25    350       0.75 rf_grid2_model_440
4        40      1.0     35    10    200      0.632 rf_grid2_model_521
5        35      1.0     25    30    350       0.75 rf_grid2_model_532
           accuracy
1 0.7173913043478262
2  0.782608695652174
3  0.782608695652174
4  0.782608695652174
5  0.782608695652174

---
    max_depth min_rows mtries nbins ntrees sample_rate          model_ids
727        35      5.0     15    20    350       0.55 rf_grid2_model_426
728        25      5.0     15    10    500       0.55  rf_grid2_model_48
729        30      5.0     15    30    200       0.55 rf_grid2_model_474
730        25      5.0     15    10    350       0.55 rf_grid2_model_100
731        40      5.0     35    30    500       0.55 rf_grid2_model_563
732        30      5.0     25    15    500      0.632 rf_grid2_model_486
           accuracy
727 0.9565217391304348
728 0.9565217391304348
729 0.9565217391304348
730 0.9565217391304348
731 0.9565217391304348
732 0.9565217391304348
```

Ilustración 17 : Paclitaxel

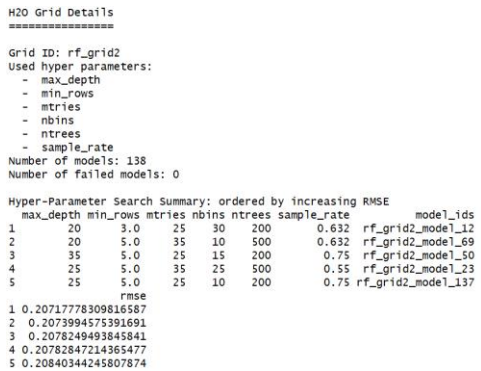
Se puede observar que, para la búsqueda de hiperpárametros con H2o en Random Forest con los valores Auc discretizados y una partición 60/20/20 (Figuras 15 a 18), el output nos devuelve, ordenados de forma ascendente por el valor “Acuracy”, que es el valor indicativo de la precisión de los modelos entrenados con predictores discretizados, la mejor combinación de hiperparámetros obtenidos, siendo en este caso el modelo número 872 para el Erlotinib, el 707 para el Rampamycin, el 605 para el Sunitinib y el 732 para el paclitaxel.

3. Regresión 10 iteraciones en validación cruzada:



```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 200
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries nbins
1        25       3.0      25    30
2        35       1.0      35    15
3        35       3.0      35    10
4        25       3.0      35    20
5        35       3.0      35    20
  ntrees sample_rate
1    200       0.632
2    500        0.55
3    500       0.632
4    500       0.632
5    500       0.632
          model_ids
1 rf_grid2_model_183
2  rf_grid2_model_57
3   rf_grid2_model_2
4 rf_grid2_model_164
5 rf_grid2_model_182
                rmse
1  0.05779177745163527
2  0.05797421628359631
3  0.05803779956503041
4 0.05805659076431670 5
5 0.05806258848781096 4
```

Ilustración 18 : Erlotinib



```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 138
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries nbins ntrees sample_rate          model_ids
1        20       3.0      25    30    200       0.632  rf_grid2_model_12
2        20       5.0      35    10    500       0.632  rf_grid2_model_69
3        35       5.0      25    15    200        0.75  rf_grid2_model_50
4        25       5.0      35    25    500        0.55  rf_grid2_model_23
5        25       5.0      25    10    200        0.75 rf_grid2_model_137
                rmse
1 0.20717778309816587
2  0.2073994575391691
3  0.2078249493845841
4 0.20782847214365477
5 0.20840344245807874
```

Ilustración 19 : Rapamycin

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 622
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries
1        35      1.0     35
2        20      1.0     35
3        20      1.0     35
4        25      1.0     35
5        40      1.0     15
  nbins ntrees sample_rate
1    10    200        0.55
2    20    200       0.632
3    10    200        0.55
4    30    200        0.55
5    20    350       0.632
           model_ids
1 rf_grid2_model_541
2  rf_grid2_model_61
3 rf_grid2_model_266
4 rf_grid2_model_307
5 rf_grid2_model_101
                rmse
1 0.13409818201891596
2  0.1350149306231596
3 0.13597092797798782
4 0.13612394376562018
5  0.1362076284734795
```

Ilustración 20 : Sunitinib

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 67
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries nbins ntrees sample_rate         model_ids               rmse
1        35      3.0     35    10    500        0.75  rf_grid2_model_7 0.18590182231050623
2        25      1.0     35    25    350        0.55 rf_grid2_model_62   0.186507515169332
3        25      3.0     35    15    500       0.632 rf_grid2_model_55 0.18652676229430912
4        35      5.0     25    25    350        0.75 rf_grid2_model_28  0.1867682884499087
5        20      1.0     15    10    350        0.75 rf_grid2_model_25 0.18681693657289708
```

Ilustración 21 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros con H2o en Random Forest con los valores Auc continuos y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 19 a 22), el output nos devuelve, ordenados de forma descendente por el valor “rmse”, que es el valor indicativo de la precisión de los modelos entrenados con predictores continuos, la mejor combinación de hiperparámetros obtenidos, siendo en este caso el modelo número 1 para todos los fármacos, ya que cuanto menor es el valor “rmse” mejor es la predicción del modelo.

4. Regresión 60/20/20:

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 1013
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries nbins
1        30      1.0     35    20
2        30      1.0     35    25
3        35      1.0     35    30
4        40      1.0     35    10
5        25      1.0     35    10
  ntrees sample_rate
1    200        0.632
2    200        0.632
3    350        0.632
4    350         0.75
5    500         0.75
           model_ids
1 rf_grid2_model_431
2 rf_grid2_model_906
3  rf_grid2_model_63
4 rf_grid2_model_402
5 rf_grid2_model_233
                rmse
1  0.0371273054489573
2 0.037232844326378724
3  0.0372840643567045 8
4  0.0373293949236562
5  0.0373354603532703 2
```

Ilustración 22 : Erlotinib

```
H2O Grid Details
================

Grid ID: rf_grid2
Used hyper parameters:
  -  max_depth
  -  min_rows
  -  mtries
  -  nbins
  -  ntrees
  -  sample_rate
Number of models: 638
Number of failed models: 0

Hyper-Parameter Search Summary: ordered by increasing RMSE
  max_depth min_rows mtries nbins ntrees sample_rate         model_ids               rmse
1        40      3.0     35    30    200        0.75 rf_grid2_model_185 0.11604646077532413
2        35      3.0     35    15    350        0.75 rf_grid2_model_625 0.11627436133515903
3        40      1.0     35    10    200       0.632  rf_grid2_model_14  0.1163269670351596
4        25      3.0     35    15    350        0.75  rf_grid2_model_86 0.11678948037068647
5        25      3.0     35    30    500        0.55 rf_grid2_model_575 0.11696541513799079
```

Ilustración 23 : Rapamycin

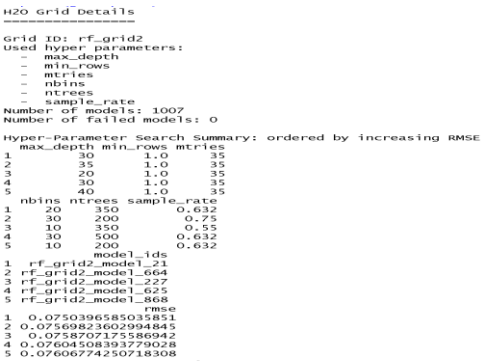


Ilustración 24 : Sunitinib

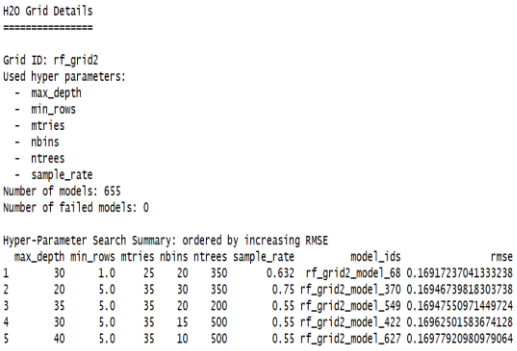


Ilustración 25 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros con H2o en Random Forest con los valores Auc continuos y una partición 60/20/20 (Figuras 23 a 26), el output nos devuelve, ordenados de forma descendente por el valor “rmse”, que es el valor indicativo de la precisión de los modelos entrenados con predictores continuos, la mejor combinación de hiperparámetros obtenidos, siendo en este caso el modelo número 1 para todos los fármacos, ya que cuanto menor es el valor “rmse” mejor es la predicción del modelo.

RandomForest Random Search:

En la búsqueda de hiperparámetros para el algoritmo random Forest, mediante Random Search, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en validación cruzada:

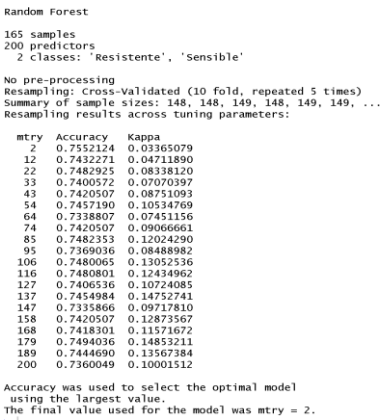


Ilustración 26 : Erlotinib

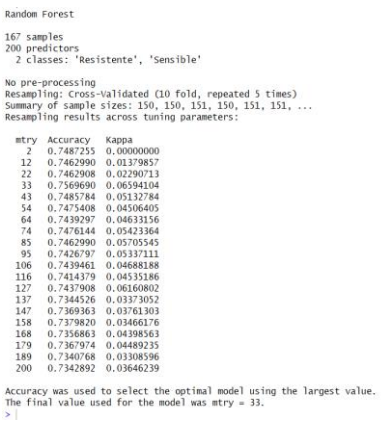


Ilustración 27 : Rapamycin

```
Random Forest

184 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 165, 166, 165, 166, 165, 166, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.7547816  0.05968109
   12   0.7472893  0.07671152
   22   0.7514345  0.09199125
   33   0.7484589  0.08783149
   43   0.7526763  0.09709124
   54   0.7504403  0.09589209
   64   0.7471070  0.08507840
   74   0.7515583  0.10146824
   85   0.7472308  0.09531523
   95   0.7450740  0.08719827
  106   0.7438321  0.09818193
  116   0.7486481  0.11180107
  127   0.7427279  0.09039005
  137   0.7395046  0.08827387
  147   0.7428449  0.09027776
  158   0.7395631  0.09021855
  168   0.7406811  0.09191327
  179   0.7352425  0.07459015
  189   0.7396354  0.08699446
  200   0.7382112  0.09846298

Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was mtry = 2.
```

Ilustración 28 : Sunitinib

```
Random Forest

184 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Cross-Validated (10 fold, repeated 5 times)
Summary of sample sizes: 165, 166, 165, 166, 165, 166, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.7504472  0.00000000
   12   0.7504472  0.00000000
   22   0.7526109  0.01567542
   33   0.7526109  0.01958577
   43   0.7527933  0.03278367
   54   0.7526694  0.03317038
   64   0.7493292  0.01712037
   74   0.7483419  0.02851277
   85   0.7526041  0.04193769
   95   0.7482766  0.03293662
  106   0.7505642  0.04943944
  116   0.7482181  0.04471137
  127   0.7516684  0.06736455
  137   0.7513106  0.05718868
  147   0.7524217  0.07184479
  158   0.7449295  0.04658025
  168   0.7482766  0.05170356
  179   0.7481527  0.05339562
  189   0.7449364  0.04596823
  200   0.7471070  0.05072258

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 43.
```

Ilustración 29 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros con random search en Random Forest con los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 27 a 30), el output nos devuelve una lista con los valores “mtry” probados y sus correspondientes valores de “Accuracy” y “Kappa”; y el mejor valor de “mtry”. Para el Erlotinib ha sido mtry=2, para el Rapamycin mtry=33, para el Sunitinib mtry=2 y para el Paclitaxel mtry=43.

2. **Clasificación 60/20/20**:

```
Random Forest

 41 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 41, 41, 41, 41, 41, 41,
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.7781545  -0.01409279
   12   0.7649640  -0.03054936
   22   0.7556829  -0.04484760
   33   0.7587001  -0.04808565
   43   0.7499327  -0.05550002
   54   0.7445718  -0.05206490
   64   0.7442147  -0.05478697
   74   0.7390480  -0.06076734
   85   0.7359711  -0.06116398
   95   0.7361616  -0.04746642
  106   0.7277806  -0.07152835
  116   0.7306378  -0.06731783
  127   0.7280346  -0.07299104
  137   0.7329235  -0.05359390
  147   0.7282251  -0.07268295
  158   0.7276774  -0.07516359
  168   0.7256816  -0.07523342
  179   0.7282251  -0.07268295
  189   0.7227012  -0.07918483
  200   0.7275901  -0.05933343

Accuracy was used to select the optimal model
 using the largest value.
The final value used for the model was mtry = 2.
```

Ilustración 30 : Erlotinib

```
Random Forest

 42 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 42, 42, 42, 42, 42, 42, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.8571362  0.00000000
   12   0.8571362  0.00000000
   22   0.8617114  0.05365854
   33   0.8624353  0.06433331
   43   0.8571721  0.03385772
   54   0.8599549  0.10146065
   64   0.8597883  0.07706449
   74   0.8593780  0.09365112
   85   0.8593780  0.09365112
   95   0.8614833  0.11193684
  106   0.8564833  0.10733078
  116   0.8589833  0.11096714
  127   0.8516303  0.08876469
  137   0.8589833  0.11096714
  147   0.8468944  0.11214041
  158   0.8563225  0.14485543
  168   0.8462055  0.13011763
  179   0.8562055  0.15604001
  189   0.8415748  0.14519875
  200   0.8440748  0.17738097

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 33.
```

Ilustración 31 : Rapamycin

```
Random Forest

 46 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 46, 46, 46, 46, 46, 46, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.6681525  0.14730896
   12   0.6505230  0.13641471
   22   0.6375452  0.09736870
   33   0.6475368  0.13389283
   43   0.6339257  0.09308613
   54   0.6412590  0.10779914
   64   0.6486538  0.12594882
   74   0.6351178  0.10788000
   85   0.6342845  0.11939519
   95   0.6377786  0.11022984
  106   0.6343204  0.11350854
  116   0.6302960  0.11824885
  127   0.6289824  0.11407220
  137   0.6285485  0.10471933
  147   0.6300457  0.12114389
  158   0.6313655  0.11385279
  168   0.6219014  0.09978864
  179   0.6190378  0.10420377
  189   0.6174432  0.09089337
  200   0.6264766  0.11250527

Accuracy was used to select the optimal model using the
 largest value.
The final value used for the model was mtry = 2.
```

Ilustración 32 : Sunitinib

```
Random Forest

 46 samples
200 predictors
  2 classes: 'Resistente', 'Sensible'

No pre-processing
Resampling: Bootstrapped (25 reps)
Summary of sample sizes: 46, 46, 46, 46, 46, 46, ...
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.7832871  -0.007040619
   12   0.7637120  -0.022434133
   22   0.7590570  -0.018755034
   33   0.7641067  -0.002623423
   43   0.7590871  -0.009772243
   54   0.7472400  -0.036354056
   64   0.7474067  -0.033236763
   74   0.7425675  -0.030942941
   85   0.7389289  -0.033620235
   95   0.7386511  -0.036481041
  106   0.7439485  -0.029981273
  116   0.7302575  -0.050999887
  127   0.7250262  -0.041655406
  137   0.7274373  -0.040546414
  147   0.7391602  -0.021180409
  158   0.7347818  -0.011273824
  168   0.7183458  -0.054282558
  179   0.7147104  -0.046250121
  189   0.7303236  -0.014217459
  200   0.7213607  -0.041665203

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 2.
```

Ilustración 33 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros con random search en Random Forest con los valores Auc discretizados y una partición 60/20/20 (Figuras 31 a 34), el output nos devuelve una lista con los valores "mtry" probados y sus correspondientes valores de "Accuracy" y "Kappa"; y el mejor valor de "mtry". Para el Erlotinib ha sido mtry=2, para el Rapamycin mtry=33, para el Sunitinib mtry=2 y para el Paclitaxel mtry=2.

3. Regresión 10 iteraciones en validación cruzada:

```
RMSE was used to select
 the optimal model using
 the smallest value.
The final value used for
 the model was mtry = 2.
```

Ilustración 34 : Erlotinib

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 74.
```

Ilustración 35 : Rapamycin

```
RMSE was used to select
 the optimal model using
 the smallest value.
The final value used for
 the model was mtry = 2.
```

Ilustración 36 : Sunitinib

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 22.
```

Ilustración 37 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros con random search en Random Forest con los valores Auc continuos y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 35 a 38), el output nos devuelve una lista con los valores "mtry" probados y sus correspondientes valores de "RMSE", "R^2" y "MAE"; y el mejor valor de "mtry". Para el Erlotinib ha sido mtry=2, para el Rapamycin mtry=74, para el Sunitinib mtry=2 y para el Paclitaxel mtry=22.

4. Regresión 60/20/20:

```
RMSE was used to select
 the optimal model using
 the smallest value.
The final value used for
 the model was mtry = 2.
```

Ilustración 38 : Erlotinib

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 2.
```

Ilustración 39 : Rapamycin

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 2.
```

Ilustración 40 : Sunitinib

```
RMSE was used to select the optimal model using the smallest value.
The final value used for the model was mtry = 2.
```

Ilustración 41 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros con random search en Random Forest con los valores Auc continuos y una partición 60/20/20 (Figuras 39 a 42), el output nos devuelve una lista con los valores "mtry" probados y sus correspondientes valores de "RMSE", "R^2" y "MAE"; y el mejor valor de "mtry". Para el Erlotinib ha sido mtry=2, para el Rapamycin mtry=2, para el Sunitinib mtry=2 y para el Paclitaxel mtry=2.

Regresión Ridge:

En la búsqueda de hiperparámetros para el algoritmo regularización de Ridge, mediante Random Search, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en validación cruzada:

```
$lambda.min
[1] 2.025937

$lambda.1se
[1] 133.2931
```

Ilustración 42 : Erlotinib

```
$lambda.min
[1] 10.74029

$lambda.1se
[1] 87.11782
```

Ilustración 43 : Rapamycin

```
$lambda.min
[1] 2.539249

$lambda.1se
[1] 126.379
```

Ilustración 44 : Sunitinib

```
$lambda.min
[1] 7.514733

$lambda.1se
[1] 88.43427
```

Ilustración 45 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión logística de Ridge con los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 43 a 46), el output nos devuelve los mejores valores obtenidos para los parámetros lambda.min y lambda.1se.

2. Clasificación 60/20/20:

```
Tuning parameter 'alpha' was held constant at a value of 0
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 0
 and lambda = 1000.
```

Ilustración 46 : Erlotinib

```
Tuning parameter 'alpha' was held constant at a value of 0
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 0
 and lambda = 1000.
```

Ilustración 47 : Rapamycin

```
Tuning parameter 'alpha' was held constant at a value of 0
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 0
 and lambda = 1000.
```

Ilustración 48 : Sunitinib

```
Tuning parameter 'alpha' was held constant at a value of 0
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 0
 and lambda = 1000.
```

Ilustración 49 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión logística de Ridge con Random Search, los valores Auc discretizados y una partición 60/20/20 (Figuras 47 a 50); el output nos devuelve el mejor valor obtenido para el parámetro lambda (ya que al ser Ridge, alpha siempre es igual a 0). Siendo estos lambda=1000 en los 4 farmacos probados.

3. Regresión 10 iteraciones en validación cruzada:

```
$lambda.min
[1] 0.2223282

$lambda.1se
[1] 20.25772

attr(,"class")
[1] "cv.glmnet"
```

Ilustración 50 : Erlotinib

```
$lambda.min
[1] 0.9610083

$lambda.1se
[1] 96.10083
```

Ilustración 51 : Rapamycin

```
$lambda.min
[1] 1.176573

$lambda.1se
[1] 70.53372
```

Ilustración 52: Sunitinib

```
$lambda.min
[1] 0.6516813

$lambda.1se
[1] 42.87628
```

Ilustración 53: Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión logística de Ridge con los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 51 a 546), el output nos devuelve los mejores valores obtenidos para los parámetros lambda.min y lambda.1se.

4. Regresión 60/20/20:

```
Tuning parameter 'alpha' was held constant at a value of 0
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 0 and lambda = 1.873817.
```

Ilustración 54 : Erlotinib

```
Tuning parameter 'alpha' was held constant at a value of 0
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 0 and lambda = 17.47528.
```

Ilustración 55 : Rapamycin

```
Tuning parameter 'alpha' was held constant at a value of 0
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 0 and lambda = 4.328761.
```

Ilustración 56 : Sunitinib

```
Tuning parameter 'alpha' was held constant at a value of 0
RMSE was used to select the optimal model using the smallest value.
The final values used for the model were alpha = 0 and lambda = 1000.
```

Ilustración 57 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión de Ridge con Random Search, los valores Auc continuos y una partición 60/20/20 (Figuras 55 a 58); el output nos devuelve el mejor valor obtenido para el parámetro lambda (ya que al ser Ridge, alpha siempre es igual a 0) y el

valor de “RMSE” más pequeño asociado. En el Erlotinib se ha obtenido lambda=1,873817, para el Rapamycin lambda=17.47528, para el Sunitinib lambda=4,328761 y para el Paclitaxel lambda =1000.

Regresión Lasso:

En la búsqueda de hiperparámetros para el algoritmo regularización de LASSO, mediante Random Search, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en validación cruzada:

```
$lambda.min
[1] 0.07627523

$lambda.1se
[1] 0.1214517
```

Ilustración 58 : Erlotinib

```
$lambda.min
[1] 0.04138802

$lambda.1se
[1] 0.06590144
```

Ilustración 59 : Rapamycin

```
$lambda.min
[1] 0.1206349

$lambda.1se
[1] 0.1387008
```

Ilustración 60 : Sunitinib

```
$lambda.min
[1] 0.04201344

$lambda.1se
[1] 0.08057802
```

Ilustración 61 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros en LASSO los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 59 a 62), el output nos devuelve los mejores valores obtenidos para los parámetros lambda.min y lambda.1se.

2. Clasificación 60/20/20:

```
Tuning parameter 'alpha' was held constant at a value of 1
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 1
 and lambda = 0.04977024.
```

Ilustración 62 : Erlotinib

```
Tuning parameter 'alpha' was held constant at a value of 1
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 1
 and lambda = 0.05722368.
```

Ilustración 63 : Rapamycin

```
Tuning parameter 'alpha' was held constant at a value of 1
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 1
 and lambda = 1000.
```

Ilustración 64 : Sunitinib

```
Tuning parameter 'alpha' was held constant at a value of 1
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha = 1
 and lambda = 1000.
```

Ilustración 65 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros en LASSO con Random Search, los valores Auc discretizados y una partición 60/20/20 (Figuras 63 a 66); el output nos devuelve el mejor valor obtenido para el parámetro lambda (ya que al ser lasso, alpha siempre es igual a 1). Siendo estos lambda=0,04977024 para el Erlotinib, lambda=0,05722358 para el Rapamycin y lambda =1000 tanto para el Sunitinib como para el Paclitaxel.

3. Regresión 10 iteraciones en validación cruzada:

```
$lambda.min
[1] 0.01279277

$lambda.1se
[1] 0.0492973
```

Ilustración 66 : Erlotinib

```
$lambda.min
[1] 0.02736986

$lambda.1se
[1] 0.08756349
```

Ilustración 67 : Rapamycin

```
$lambda.min
[1] 0.01384606

$lambda.1se
[1] 0.06732786
```

Ilustración 68 : Sunitinib

```
$lambda.min
[1] 0.01279277

$lambda.1se
[1] 0.0492973
```

Ilustración 69 : Paclitaxel

Se puede observar que, para la búsqueda de hiperpárametros en LASSO con Random Search, los valores Auc continuos y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 67 a 70); el output nos devuelve el mejor valor obtenido para el parámetro lambda (ya que al ser lasso, alpha siempre es igual a 1). Siendo estos lambda=0,0492973 para el Erlotinib, lambda=0,08756349 para el Rapamycin, lambda =0.06732786 para el Sunitinib y lambda=0.0492973 para el Paclitaxel.

4. Regresión 60/20/20:

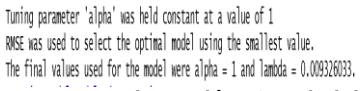


Ilustración 70 : Erlotinib

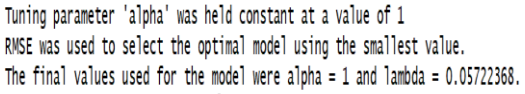


Ilustración 71 : Rapamycin

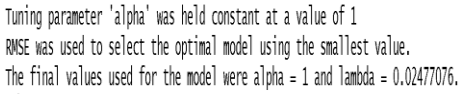


Ilustración 72 : Sunitinib

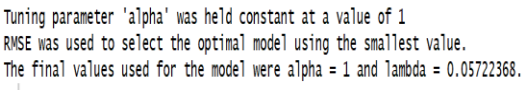


Ilustración 73 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en LASSO con Random Search, los valores Auc continuos y una partición 60/20/20 (Figuras 71 a 74); el output nos devuelve el mejor valor obtenido para el parámetro lambda (ya que al ser lasso, alpha siempre es igual a 1). Siendo estos lambda=0.009326033 para el Erlotinib, lambda=0,05722368 para el Rapamycin, lambda =0,02477076 para el Sunitinib y lambda=0.05722368 para el Paclitaxel.

Regresión Elastic Net:

En la búsqueda de hiperparámetros para el algoritmo regularización de Elastic Net, mediante Random Search, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en Validación cruzada:

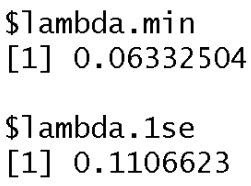


Ilustración 74 : Erlotinib

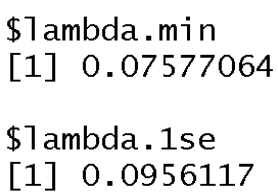


Ilustración 75 : Rapamycin

```
$lambda.min
[1] 0.08314895

$lambda.1se
[1] 0.126379
```

Ilustración 76 : Sunitinib

```
$lambda.min
[1] 0.01657097

$lambda.1se
[1] 0.04830525
```

Ilustración 77 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión logística Elastic Net los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 75 a 78), el output nos devuelve los mejores valores obtenidos para los parámetros lambda.min y lambda.1se.

2. Clasificación 60/20/20:

```
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha
 = 0.1473684 and lambda = 0.2952464.
```

Ilustración 78 : Erlotinib

```
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha
 = 0.5736842 and lambda = 0.07747661.
```

Ilustración 79 : Rapamycin

```
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha
 = 0.1947368 and lambda = 0.2139757.
```

Ilustración 80 : Sunitinib

```
Accuracy was used to select the optimal model using
 the largest value.
The final values used for the model were alpha
 = 0.2421053 and lambda = 0.06136475.
```

Ilustración 81 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión logística Elastic Net con Random Search, los valores Auc discretizados y una partición 60/20 (Figuras 79 a 82); el output nos devuelve el mejor valor obtenido para el parámetro lambda y alpha. Siendo estos alpha=0.1473684 y lambda=0.2952464 para el Erlotinib; alpha=0.5736842 y lambda=0.07747661 para el Rapamycin; alpha=0.1947368 y lambda=0.2139757 para el Sunitinib; y alpha=0.242105 y lambda=0.06136475 para el Paclitaxel.

3. Regresión 10 iteraciones en validación cruzada:

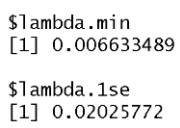


Ilustración 82 : Erlotinib

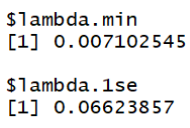


Ilustración 83 : Rapamycin

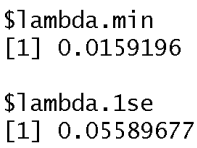


Ilustración 84 : Sunitinib

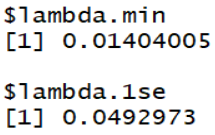


Ilustración 85 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión Elastic Net con Random Search, los valores Auc continuos y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 83 a 86); el output nos devuelve el mejor valor obtenido para el parámetro lambda y alpha. Siendo estos lambda=0.02025772 para el Erlotinib; lambda=0.06623857 para el Rapamycin; lambda=0,05589677 para el Sunitinib; y lambda=0.0492973 para el Paclitaxel.

4. Regresión 60/20/20:

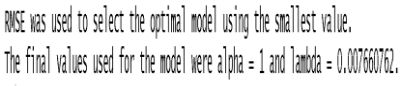


Ilustración 86 : Erlotinib

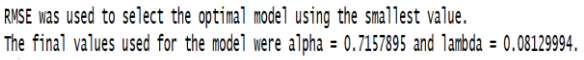


Ilustración 87 : Rapamycin

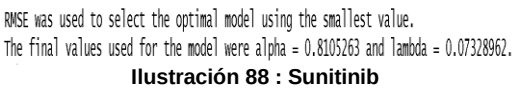


Ilustración 88 : Sunitinib

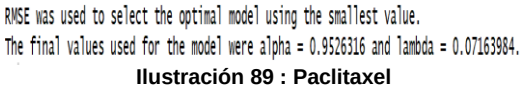


Ilustración 89 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en una regresión Elastic Net con Random Search, los valores Auc continuos y una partición 60/20/20 (Figuras 87 a 90); el output nos devuelve el mejor valor obtenido para el parámetro lambda y alpha. Siendo estos alpha=1 y lambda=0.007660762 para el Erlotinib; alpha=0,7157895 y lambda=0.08129994 para el Rapamycin; alpha=0.805263 y lambda=0,07328962 para el Sunitinib; y alpha=0.9526316 y lambda=0.07163984 para el Paclitaxel.

SVM:

En la búsqueda de hiperparámetros para el algoritmo regularización de SVM, mediante Random Search, y ordenados por el tipo de predictor (discretizados o continuos) y las distintas particiones de datos realizadas, se han obtenido los siguientes valores óptimos:

1. Clasificación 10 iteraciones en validación cruzada:

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.3095588

- Detailed performance results:
    cost gamma     error dispersion
1  1e-01   0.5 0.3095588 0.08402967
2  1e+00   0.5 0.3099265 0.08114281
3  1e+01   0.5 0.3099265 0.08114281
4  1e+02   0.5 0.3099265 0.08114281
5  1e+03   0.5 0.3099265 0.08114281
6  1e-01   1.0 0.3095588 0.08402967
7  1e+00   1.0 0.3099265 0.08114281
8  1e+01   1.0 0.3099265 0.08114281
9  1e+02   1.0 0.3099265 0.08114281
10 1e+03   1.0 0.3099265 0.08114281
```

Ilustración 90 : Erlotinib

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.3522059

- Detailed performance results:
    cost gamma     error dispersion
1  1e-01   0.5 0.3522059  0.1754875
2  1e+00   0.5 0.3882353  0.1544332
3  1e+01   0.5 0.3882353  0.1544332
4  1e+02   0.5 0.3882353  0.1544332
5  1e+03   0.5 0.3882353  0.1544332
6  1e-01   1.0 0.3522059  0.1754875
7  1e+00   1.0 0.3882353  0.1544332
8  1e+01   1.0 0.3882353  0.1544332
9  1e+02   1.0 0.3882353  0.1544332
10 1e+03   1.0 0.3882353  0.1544332
```

Ilustración 91 : Rapamycin

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.294152

- Detailed performance results:
    cost gamma     error dispersion
1  1e-01   0.5 0.2941520 0.08680704
2  1e+00   0.5 0.3216374 0.08570350
3  1e+01   0.5 0.3216374 0.08570350
4  1e+02   0.5 0.3216374 0.08570350
5  1e+03   0.5 0.3216374 0.08570350
6  1e-01   1.0 0.2941520 0.08680704
7  1e+00   1.0 0.3216374 0.08570350
8  1e+01   1.0 0.3216374 0.08570350
9  1e+02   1.0 0.3216374 0.08570350
10 1e+03   1.0 0.3216374 0.08570350
```

Ilustración 92 : Sunitinib

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.304386

- Detailed performance results:
    cost gamma     error dispersion
1  1e-01   0.5 0.3043860 0.08680977
2  1e+00   0.5 0.3426901 0.08324979
3  1e+01   0.5 0.3426901 0.08324979
4  1e+02   0.5 0.3426901 0.08324979
5  1e+03   0.5 0.3426901 0.08324979
6  1e-01   1.0 0.3043860 0.08680977
7  1e+00   1.0 0.3426901 0.08324979
8  1e+01   1.0 0.3426901 0.08324979
9  1e+02   1.0 0.3426901 0.08324979
10 1e+03   1.0 0.3426901 0.08324979
```

Ilustración 93 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en SVMLineal con Random Search, los valores Auc discretizados y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 91 a 94); el output nos devuelve el mejor valor obtenido para los parámetros coste, error y gamma. Siendo estos cost=0.1 y gamma=0.5 para el Erlotinib; cost=0.1 y gamma=0.5 para el Rapamycin; cost=0.1 y gamma=0.5 para el Sunitinib; y cost=0.1 y gamma=0.5 para el Paclitaxel.

2. Clasificación 60/20/20:

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.365

- Detailed performance results:
    cost gamma error dispersion
1  1e-01   0.5 0.365  0.2667187
2  1e+00   0.5 0.365  0.2667187
3  1e+01   0.5 0.365  0.2667187
4  1e+02   0.5 0.365  0.2667187
5  1e+03   0.5 0.365  0.2667187
6  1e-01   1.0 0.365  0.2667187
7  1e+00   1.0 0.365  0.2667187
8  1e+01   1.0 0.365  0.2667187
9  1e+02   1.0 0.365  0.2667187
10 1e+03   1.0 0.365  0.2667187
```

Ilustración 94 : Erlotinib

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.38

- Detailed performance results:
    cost gamma error dispersion
1  1e-01   0.5  0.38  0.1917753
2  1e+00   0.5  0.38  0.1917753
3  1e+01   0.5  0.38  0.1917753
4  1e+02   0.5  0.38  0.1917753
5  1e+03   0.5  0.38  0.1917753
6  1e-01   1.0  0.38  0.1917753
7  1e+00   1.0  0.38  0.1917753
8  1e+01   1.0  0.38  0.1917753
9  1e+02   1.0  0.38  0.1917753
10 1e+03   1.0  0.38  0.1917753
```

Ilustración 95 : Rapamycin

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.24

- Detailed performance results:
    cost gamma error dispersion
1  1e-01   0.5  0.24   0.132916
2  1e+00   0.5  0.24   0.132916
3  1e+01   0.5  0.24   0.132916
4  1e+02   0.5  0.24   0.132916
5  1e+03   0.5  0.24   0.132916
6  1e-01   1.0  0.24   0.132916
7  1e+00   1.0  0.24   0.132916
8  1e+01   1.0  0.24   0.132916
9  1e+02   1.0  0.24   0.132916
10 1e+03   1.0  0.24   0.132916
```

Ilustración 96 : Sunitinib

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost gamma
  0.1   0.5

- best performance: 0.32

- Detailed performance results:
    cost gamma error dispersion
1  1e-01   0.5  0.32  0.1974842
2  1e+00   0.5  0.32  0.1974842
3  1e+01   0.5  0.32  0.1974842
4  1e+02   0.5  0.32  0.1974842
5  1e+03   0.5  0.32  0.1974842
6  1e-01   1.0  0.32  0.1974842
7  1e+00   1.0  0.32  0.1974842
8  1e+01   1.0  0.32  0.1974842
9  1e+02   1.0  0.32  0.1974842
10 1e+03   1.0  0.32  0.1974842
```

Ilustración 97 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en SVMLineal con Random Search, los valores Auc discretizados y una partición 60/20/20 (Figuras 95 a 98); el output nos devuelve el mejor valor obtenido para los parámetros coste, error y gamma. Siendo estos cost=0.1 y gamma=0.5 para el Erlotinib; cost=0.1 y gamma=0.5 para el Rapamycin; cost=0.1 y gamma=0.5 para el Sunitinib; y cost=0.1 y gamma=0.5 para el Paclitaxel.

3. Regresión 10 iteraciones en validación cruzada:

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.005003296

- Detailed performance results:
   cost       error dispersion
1 1e-01 0.005003296 0.00501501
2 1e+00 0.005003296 0.00501501
3 1e+01 0.005003296 0.00501501
4 1e+02 0.005003296 0.00501501
5 1e+03 0.005003296 0.00501501
```

Ilustración 98 : Erlotinib

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.05013818

- Detailed performance results:
   cost      error dispersion
1 1e-01 0.05013818 0.01730346
2 1e+00 0.05013818 0.01730346
3 1e+01 0.05013818 0.01730346
4 1e+02 0.05013818 0.01730346
5 1e+03 0.05013818 0.01730346
```

Ilustración 99 : Rapamycin

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.0218232

- Detailed performance results:
   cost     error  dispersion
1 1e-01 0.0218232 0.009095201
2 1e+00 0.0218232 0.009095201
3 1e+01 0.0218232 0.009095201
4 1e+02 0.0218232 0.009095201
5 1e+03 0.0218232 0.009095201
```

Ilustración 100 : Sunitinib

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.04054482

- Detailed performance results:
   cost      error dispersion
1 1e-01 0.04054482 0.01241908
2 1e+00 0.04054482 0.01241908
3 1e+01 0.04054482 0.01241908
4 1e+02 0.04054482 0.01241908
5 1e+03 0.04054482 0.01241908
```

Ilustración 101 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en SVMLineal con Random Search, los valores Auc continuos y una partición 80/20 con 10 iteraciones en validación cruzada (Figuras 99 a 102); el output nos devuelve el mejor valor obtenido para el parámetro coste. Siendo estos cost=0.1 para el Erlotinib, cost=0.1 para el Rapamycin, cost=0.1 para el Sunitinib y cost=0.1 para el Paclitaxel.

4. Regresión 60/20/20:

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.001311882

- Detailed performance results:
   cost       error  dispersion
1 1e-01 0.001311882 0.001166777
2 1e+00 0.001311882 0.001166777
3 1e+01 0.001311882 0.001166777
4 1e+02 0.001311882 0.001166777
5 1e+03 0.001311882 0.001166777
```

Ilustración 102 : Erlotinib

```
Parameter tuning of ‘svm’:

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.05284324

- Detailed performance results:
   cost      error dispersion
1 1e-01 0.05284324 0.03517232
2 1e+00 0.05284324 0.03517232
3 1e+01 0.05284324 0.03517232
4 1e+02 0.05284324 0.03517232
5 1e+03 0.05284324 0.03517232
```

Ilustración 103 : Rapamycin

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.02824238

- Detailed performance results:
   cost      error dispersion
1 1e-01 0.02824238 0.04923379
2 1e+00 0.02824238 0.04923379
3 1e+01 0.02824238 0.04923379
4 1e+02 0.02824238 0.04923379
5 1e+03 0.02824238 0.04923379
```

Ilustración 104 : Sunitinib

```
Parameter tuning of 'svm':

- sampling method: 10-fold cross validation

- best parameters:
 cost
  0.1

- best performance: 0.04398324

- Detailed performance results:
   cost      error dispersion
1 1e-01 0.04398324  0.0139346
2 1e+00 0.04398324  0.0139346
3 1e+01 0.04398324  0.0139346
4 1e+02 0.04398324  0.0139346
5 1e+03 0.04398324  0.0139346
```

Ilustración 105 : Paclitaxel

Se puede observar que, para la búsqueda de hiperparámetros en SVMLineal con Random Search, los valores Auc continuos y una partición 60/20/20 (Figuras 103 a 106); el output nos devuelve el mejor valor obtenido para el parámetro coste. Siendo estos cost=0.1 para el Erlotinib, cost=0.1 para el Rapamycin, cost=0.1 para el Sunitinib y cost=0.1 para el Paclitaxel.

Matrices de confusión y Gráficas:

Los resultados de las predicciones realizadas con los modelos ajustados y entrenados sobre los datos Asociados a los Test Sets, han sido representados mediante matrices de confusion y curvas Roc en modelos discretizados representando la precision de estos por el valor "Accuracy" y curvas de representacion de R^2 para representar la precision del modelo en modelos continuos.

Random forest H2o:

1. Clasificación 10 iteraciones en validación cruzada:

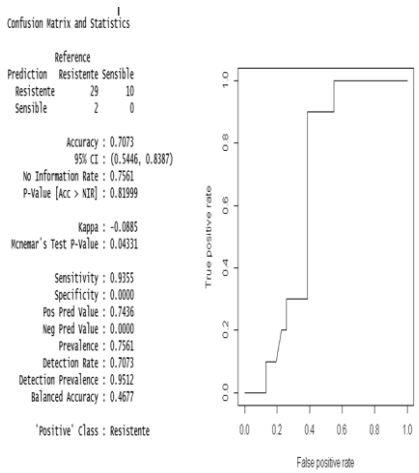


Ilustración 106 : Erlotinib

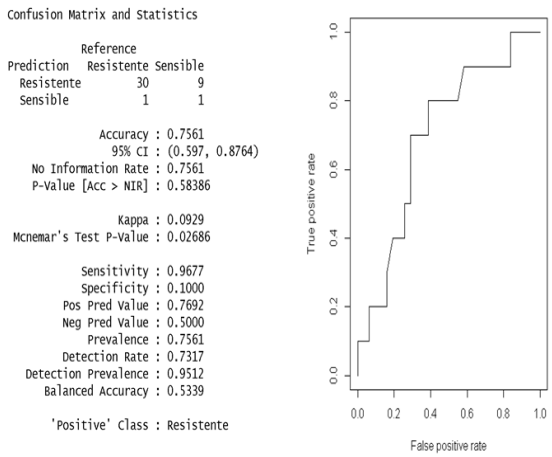


Ilustración 107 : Rapamycin

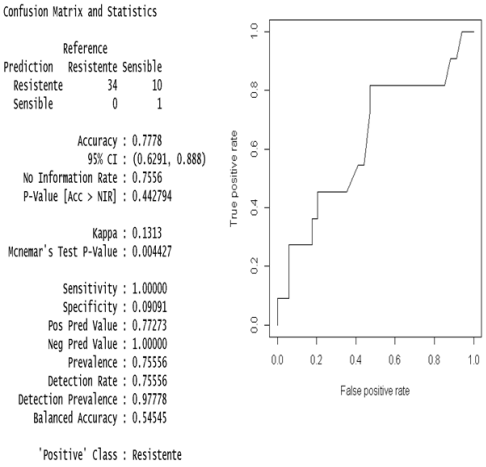


Ilustración 108 : Rapamycin

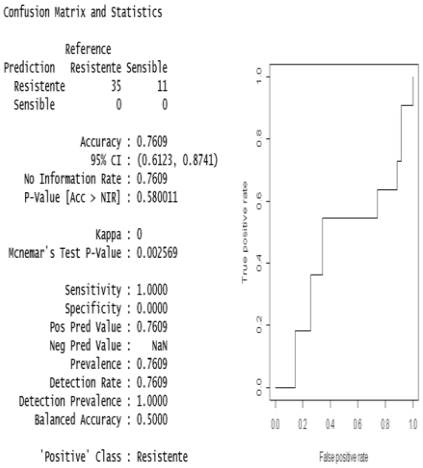


Ilustración 109 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 107 a 110) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7561 para el Erlotinib, un 0,7073 para el Rapamycin, un 0,7778 para el Sunitinib y un 0,7609 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representacion gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

2. Clasificación 60/20/20:

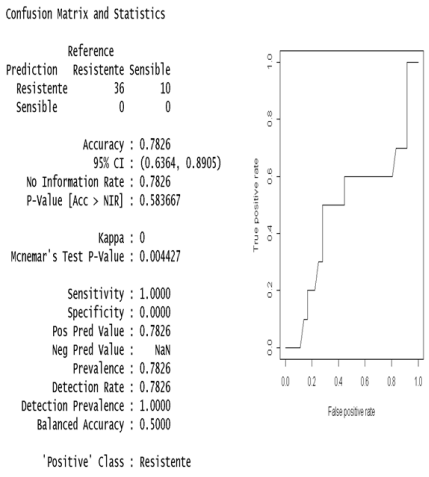


Ilustración 110 : Erlotinib

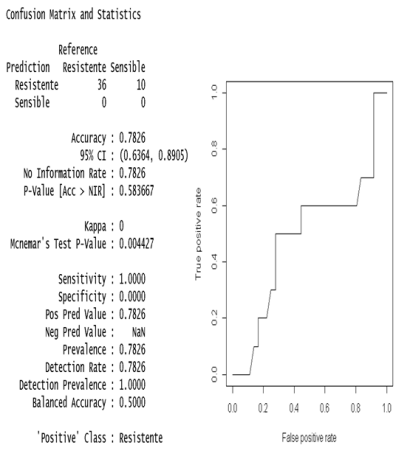


Ilustración 111: Rapamycin

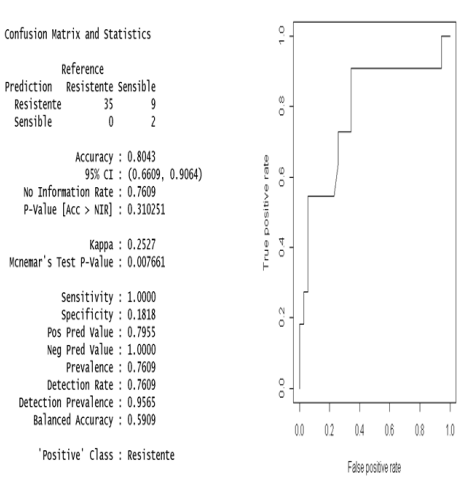


Ilustración 112 : Sunitinib

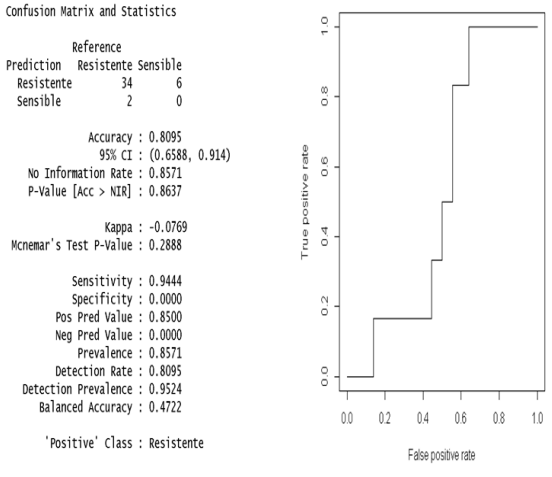


Ilustración 113 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 60/20/20 (Figuras 111 a 114) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7826 para el Erlotinib, un 0,7073 para el Rapamycin, un 0,8043 para el Sunitinib y un 0,8095 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representacion gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 iteraciones en validación cruzada:

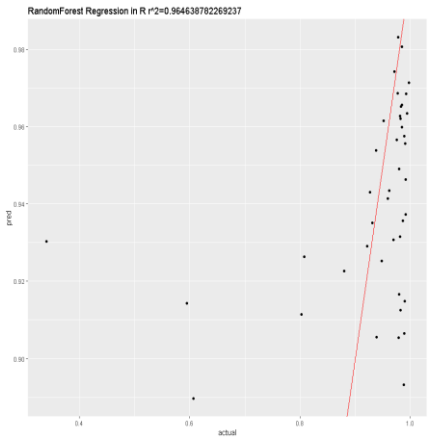


Ilustración 114 : Erlotinib

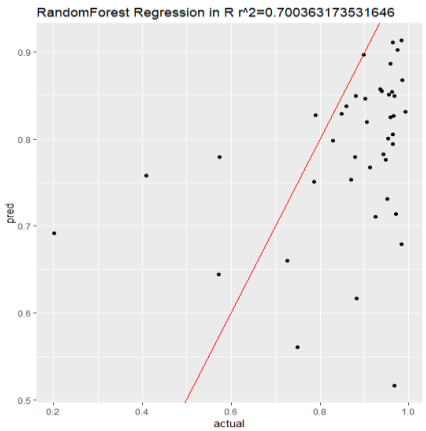


Ilustración 115 : Rapamycin

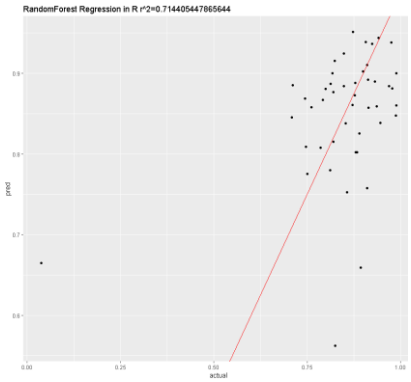


Ilustración 116 : Sunitinib

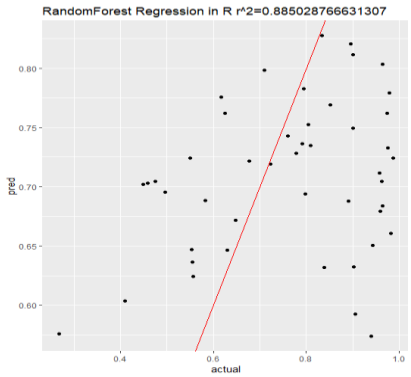


Ilustración 117 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de Random forest con hiperparámetos seleccionados mediante H2o y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 115 a 118) se ha obtenido un modelo con una bondad ("R^{2}") del 0,96463882269237 para el Erlotinib, un 0.7003632 para el Rapamycin, un 0.7144054 para el Sunitinib y un 0.8850288 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predición de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

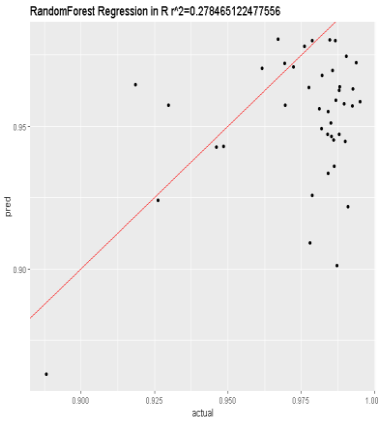


Ilustración 118 : Erlotinib

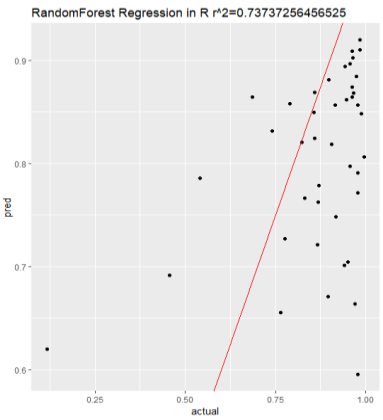


Ilustración 119 : Rapamycin

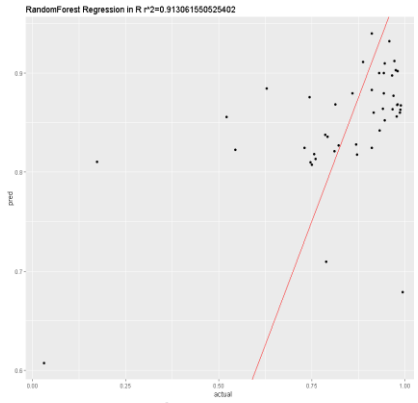


Ilustración 120 : Sunitinib

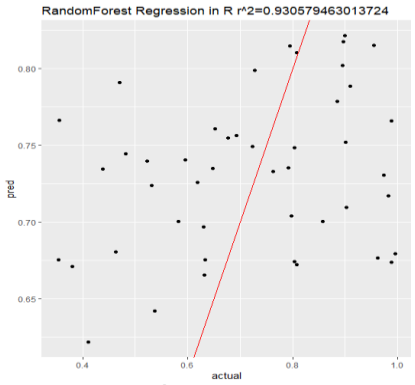


Ilustración 121 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 60/20/20 (Figuras 119 a 122) se ha obtenido un modelo con una bondad ("R^2") del 0.2784651 para el Erlotinib, un 0.7373726 para el Rapamycin, un 0.9130616 para el Sunitinib y un 0.9305795 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predición de este cuanto mayor es el valor de R^2.

Random Forest with Random Search:

1. Clasificación 10 iteraciones en validación cruzada:

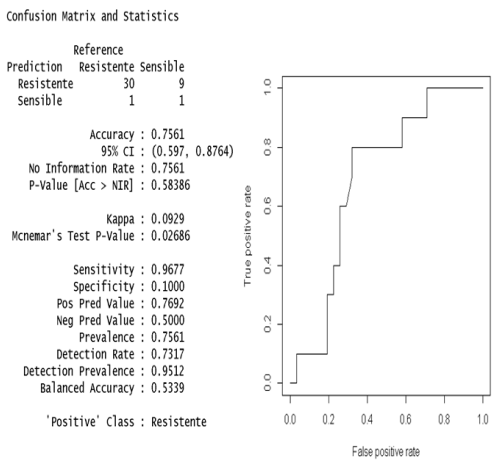


Ilustración 122 : Erlotinib

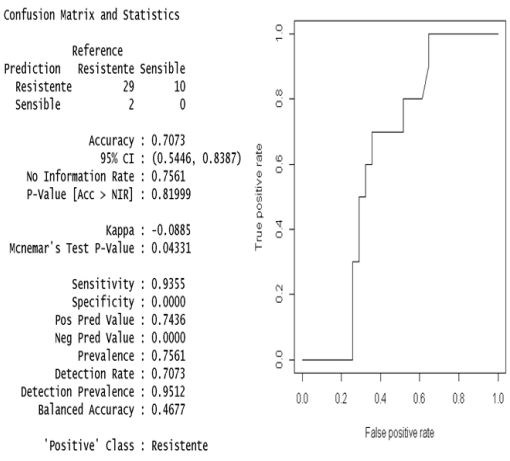


Ilustración 123 : Rapamycin

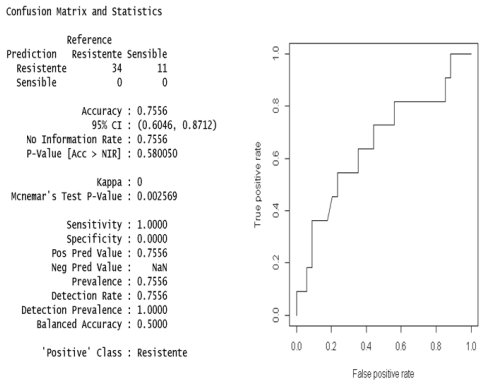


Ilustración 124 : Sunitinib

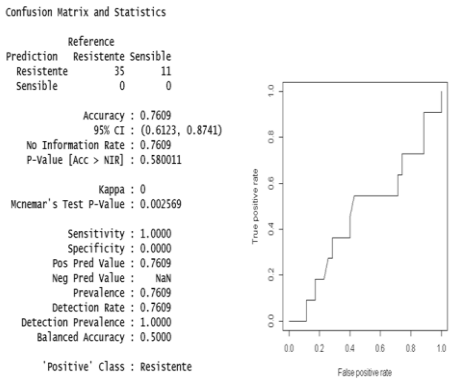


Ilustración 125 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 123 a 126) se ha obtenido un modelo con una precision ("Accuracy") del 0,7561 para el Erlotinib, un 0,7073 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,7609 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

2. Clasificación 60/20/20:

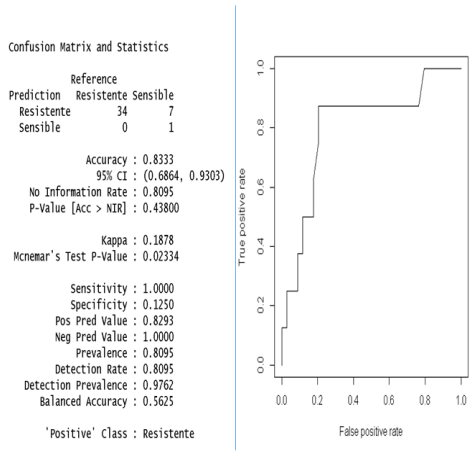


Ilustración 126 : Erlotinib

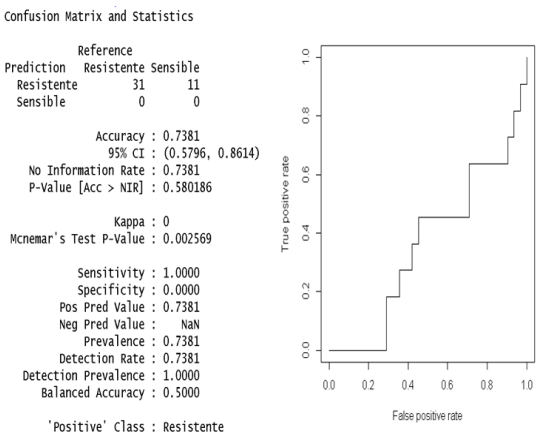


Ilustración 127 : Rapamycin

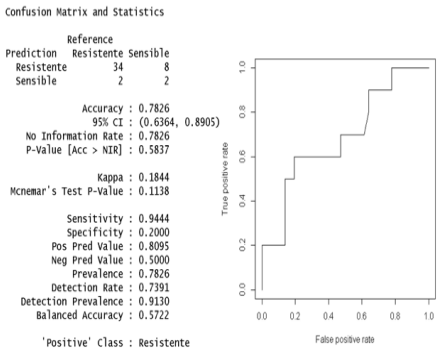


Ilustración 128 : Sunitinib

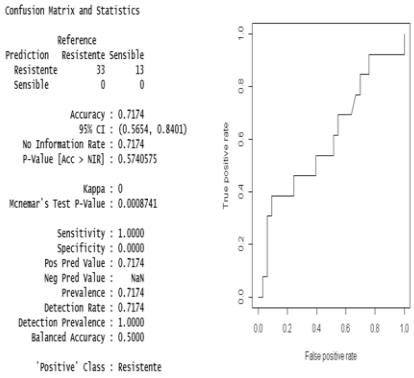


Ilustración 129 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 127 a 130) se ha obtenido un modelo con una precision (“Accuracy”) del 0,8333 para el Erlotinib, un 0,7381 para el Rapamycin, un 0,7826 para el Sunitinib y un 0,7174 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje “x” el ratio de falsos positivos y el eje “y” el ratio de verdaderos positivos) de la precision de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 en validación cruzada:

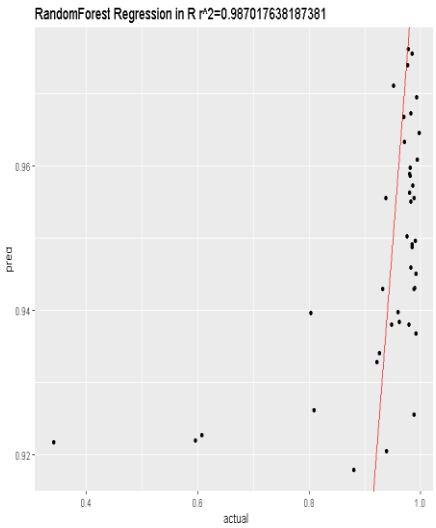


Ilustración 130 : Erlotinib

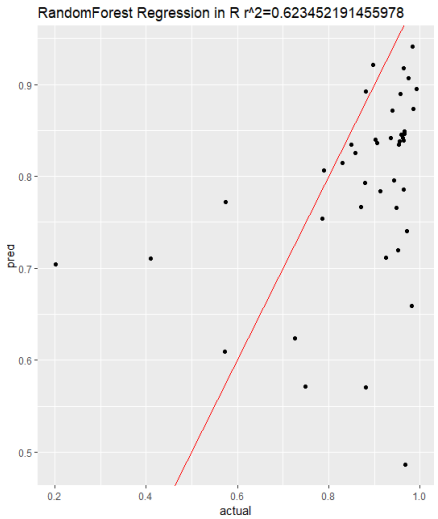


Ilustración 131 : Rapamycin

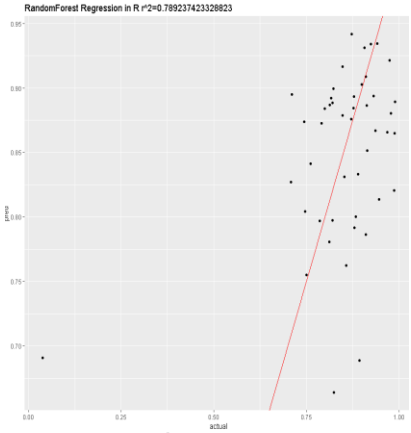


Ilustración 132 : Sunitinib

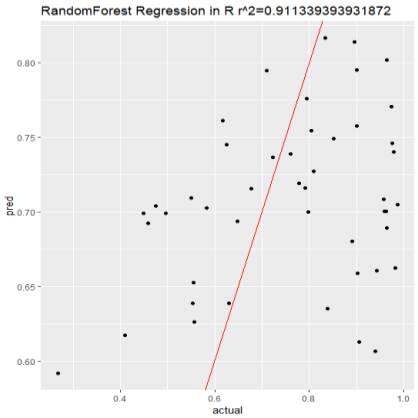


Ilustración 133 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 131 a 134) se ha obtenido un modelo con una precisión ("R^{2}") del 0.9870176 para el Erlotinib, un 0.6234522 para el Rapamycin, un 0.7892374 para el Sunitinib y un 0.9113394 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo major la predición de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

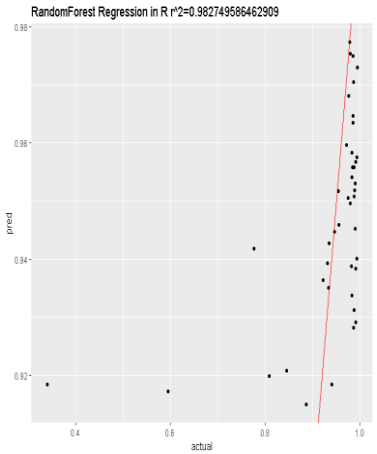


Ilustración 134 : Erlotinib

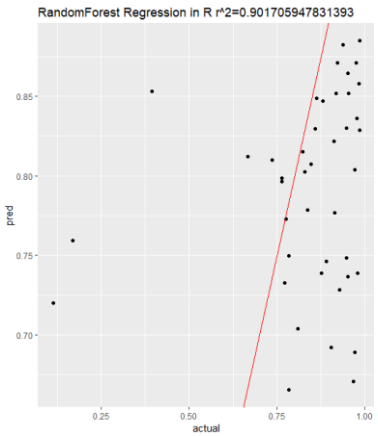


Ilustración 135 : Rapamycin

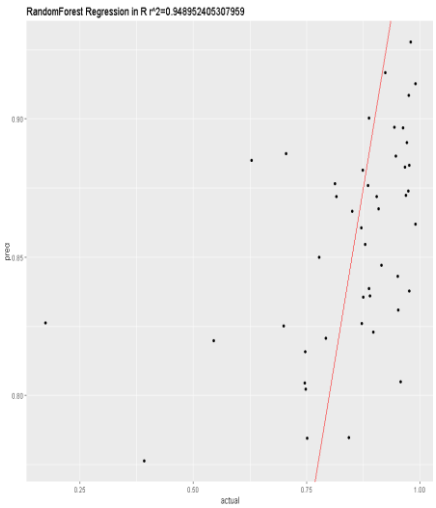


Ilustración 136 : Sunitinib

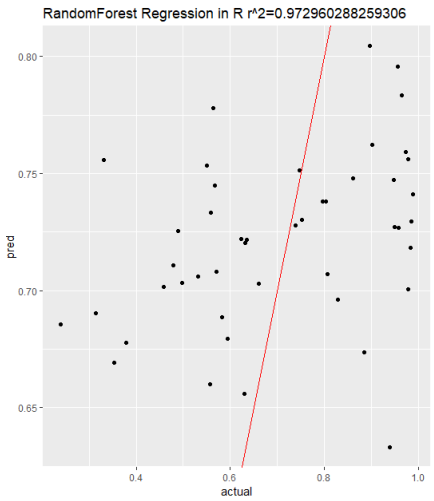


Ilustración 137 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 135 a 138) se ha obtenido un modelo con una precision ("R^2") del 0.9827496 para el Erlotinib, un 0.9017059 para el Rapamycin, un 0.9489524 para el Sunitinib y un 0.9729603 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predición de este cuanto mayor es el valor de R^2.

Regresión Ridge:

1. Clasificación 10 iteraciones en validación cruzada:

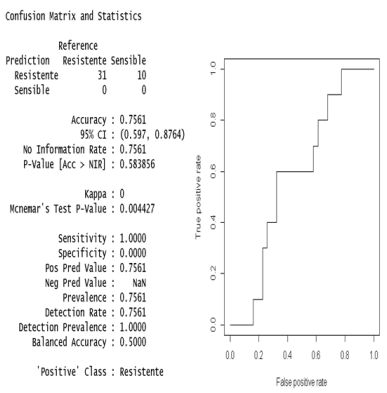


Ilustración 138 : Erlotinib

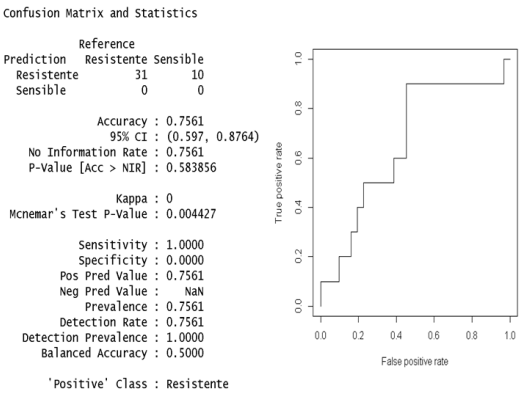


Ilustración 139 : Rapamycin

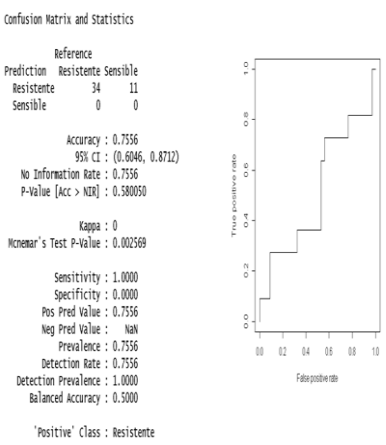


Ilustración 140 : Sunitinib

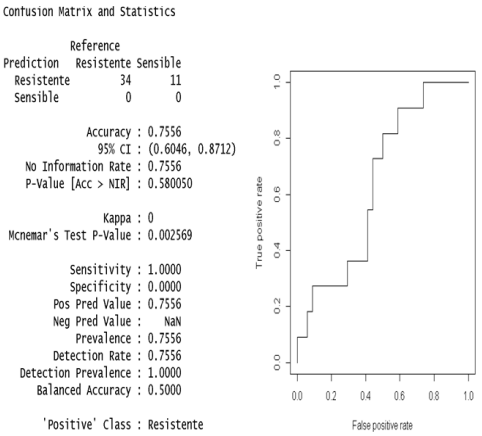


Ilustración 141 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 139 a 142) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7561 para el Erlotinib, un 0,7561 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,7556 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precision de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc(cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

.

2. Clasificación 60/20/20:

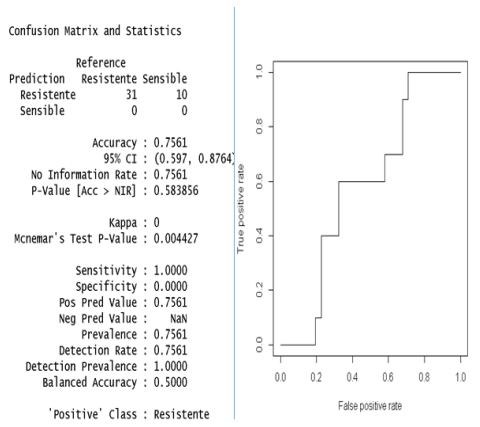


Ilustración 142 : Erlotinib

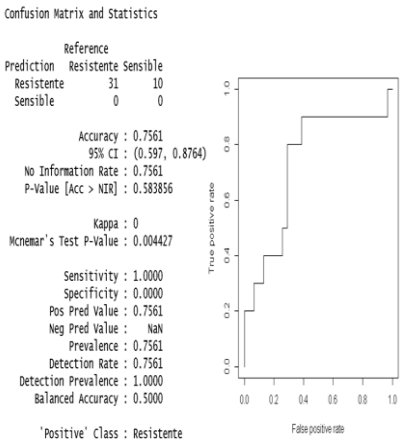


Ilustración 143 : Rapamycin

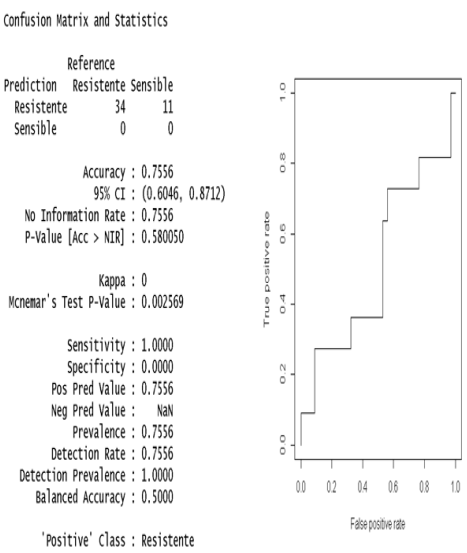


Ilustración 144 : Sunitinib

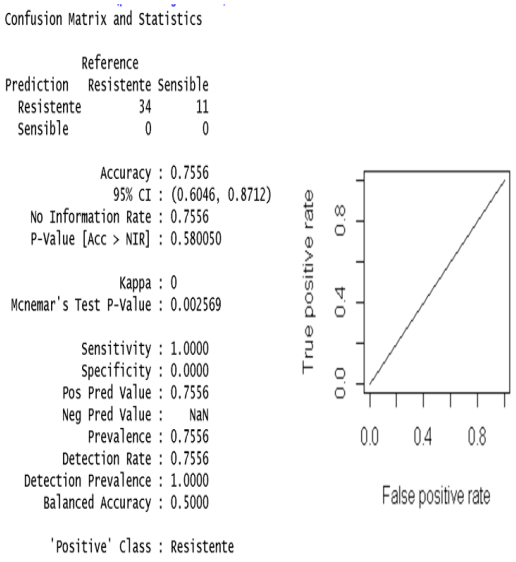


Ilustración 145 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 143 a 146) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7561 para el Erlotinib, un 0,7561 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,7556 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de Falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 iteraciones en validación cruzada:

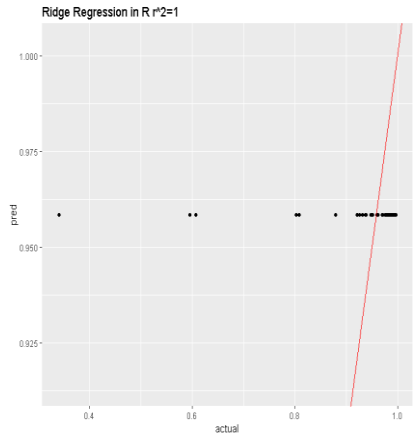


Ilustración 146 : Erlotinib

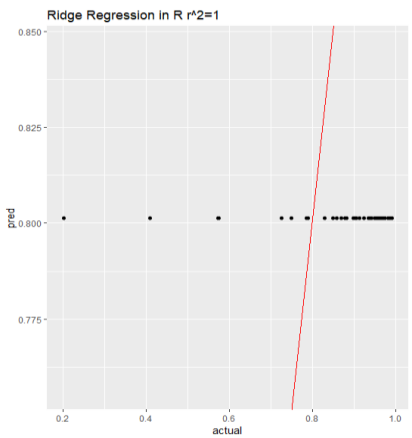


Ilustración 147 : Rapamycin

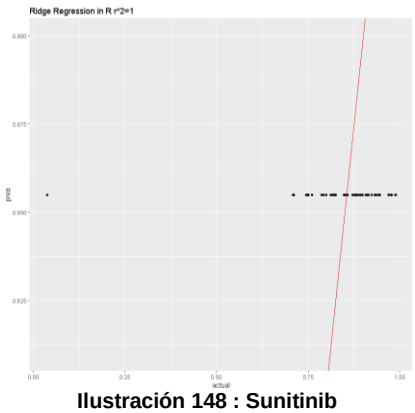


Ilustración 148 : Sunitinib

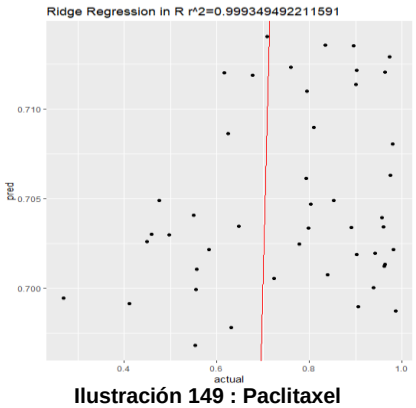


Ilustración 149 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 147 a 150) se ha obtenido un modelo con una precisión ("R^2") del 1 para el Erlotinib, un 1 para el Rapamycin, un 1 para el Sunitinib y un 0.9993495 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicción de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

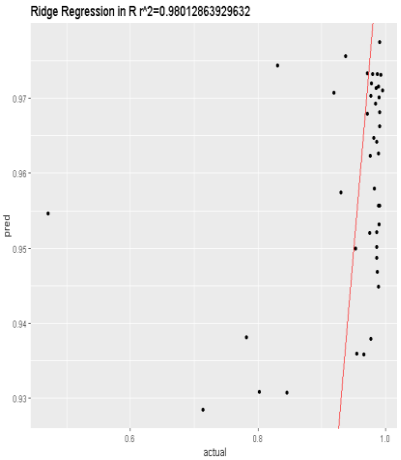


Ilustración 150 : Erlotinib

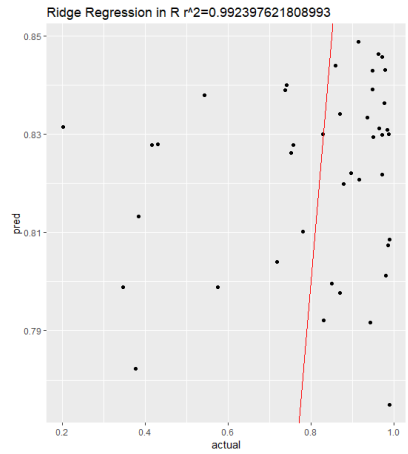


Ilustración 151 : Rapamycin

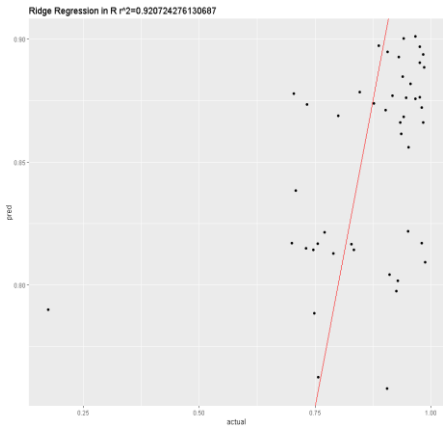


Ilustración 152 : Sunitinib

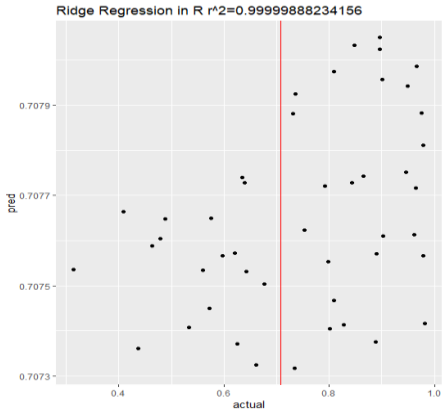


Ilustración 153 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 151 a 154) se ha obtenido un modelo con una precisión ("R^2") del 0.9801286 para el Erlotinib, un 0.9923976 para el Rapamycin, un 0.9207243 para el Sunitinib y un 0.9999989 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicción de este cuanto mayor es el valor de R^2.

Regresión LASSO:

1. Clasificación 10 iteraciones en validación cruzada:

```
Confusion Matrix and Statistics

          Reference
Prediction   Resistente Sensible
  Resistente         27        8
  Sensible            4        2

               Accuracy : 0.7073
                 95% CI : (0.5446, 0.8387)
    No Information Rate : 0.7561
    P-Value [Acc > NIR] : 0.8200

                  Kappa : 0.0821
 Mcnemar's Test P-Value : 0.3865

            Sensitivity : 0.8710
            Specificity : 0.2000
         Pos Pred Value : 0.7714
         Neg Pred Value : 0.3333
             Prevalence : 0.7561
         Detection Rate : 0.6585
   Detection Prevalence : 0.8537
      Balanced Accuracy : 0.5355

       'Positive' Class : Resistente
```

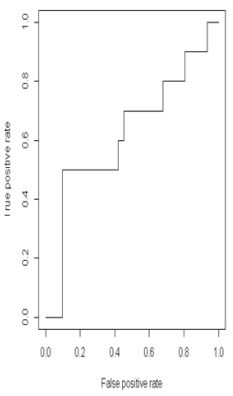


Ilustración 154 : Erlotinib

```
Confusion Matrix and Statistics

          Reference
Prediction   Resistente Sensible
  Resistente         34       10
  Sensible            0        1

               Accuracy : 0.7778
                 95% CI : (0.6291, 0.888)
    No Information Rate : 0.7556
    P-Value [Acc > NIR] : 0.442794

                  Kappa : 0.1313
 Mcnemar's Test P-Value : 0.004427

            Sensitivity : 1.00000
            Specificity : 0.09091
         Pos Pred Value : 0.77273
         Neg Pred Value : 1.00000
             Prevalence : 0.75556
         Detection Rate : 0.75556
   Detection Prevalence : 0.97778
      Balanced Accuracy : 0.54545

       'Positive' Class : Resistente
```

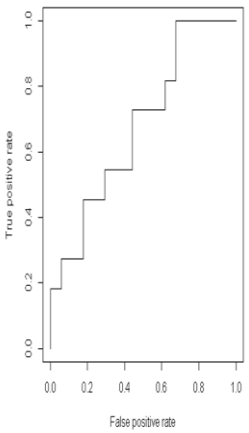


Ilustración 155 : Rapamycin

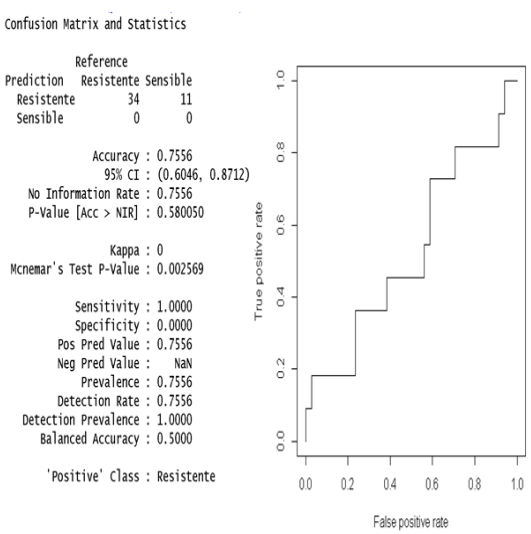


Ilustración 156 : Sunitinib

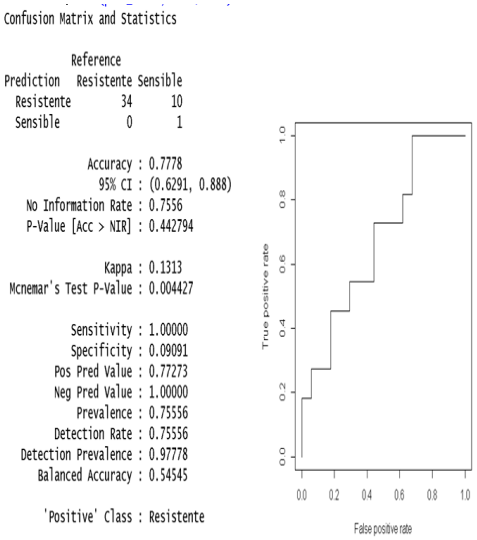


Ilustración 157 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 155 a 158) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7073 para el Erlotinib, un 0,7778 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,7778 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

2. Clasificación 60/20/20:

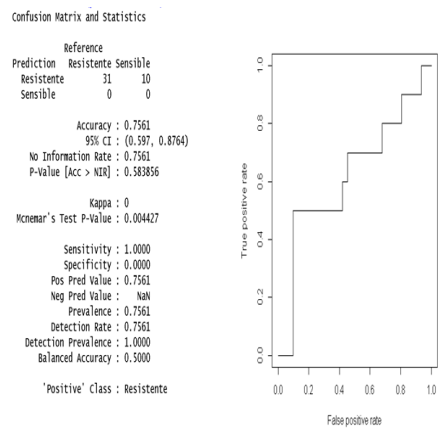


Ilustración 158 : Erlotinib

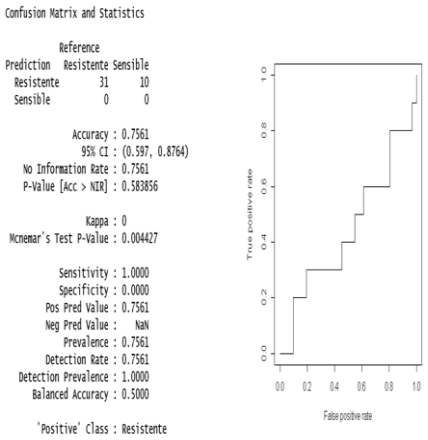


Ilustración 159 : Rapamycin

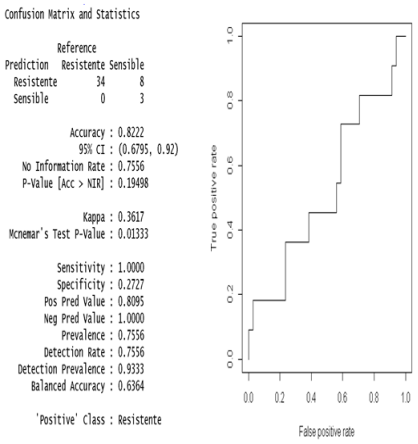


Ilustración 160 : Sunitinib

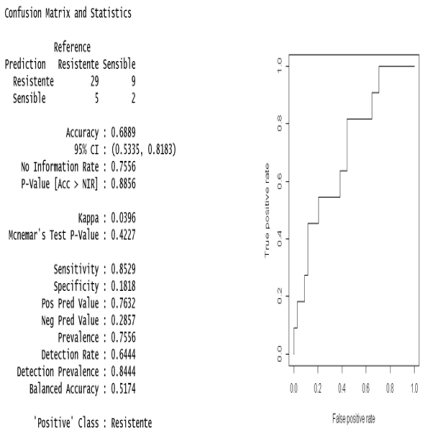


Ilustración 161 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 159 a 162) se ha obtenido un modelo con una precisión ("Accuracy") del 0,8764 para el Erlotinib, un 0,7561 para el Rapamycin, un 0,8222 para el Sunitinib y un 0,6889 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 iteraciones en validación cruzada:

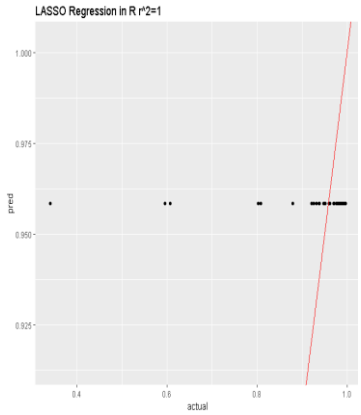


Ilustración 162 : Erlotinib

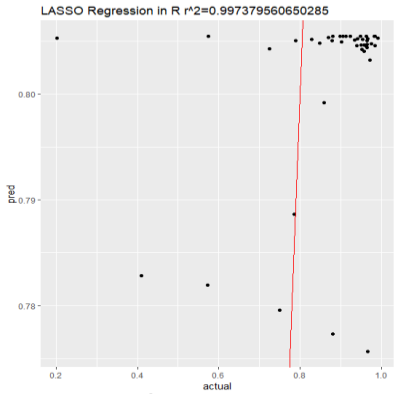


Ilustración 163 : Rapamycin

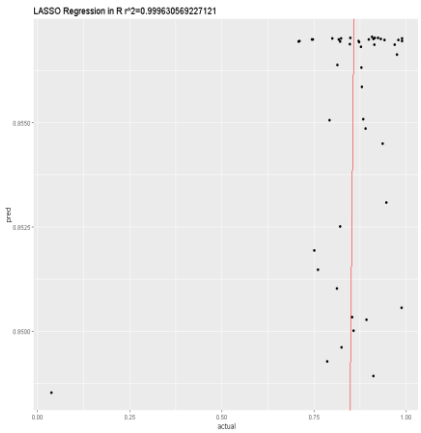


Ilustración 164 : Sunitinib

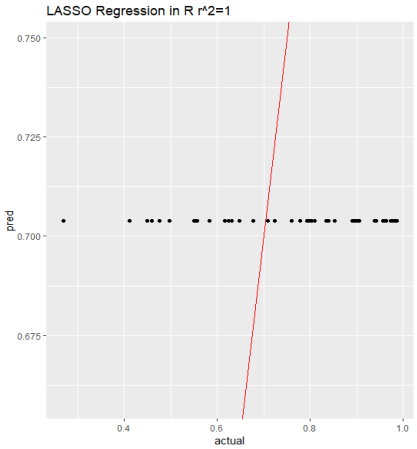


Ilustración 165 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 163 a 166) se ha obtenido un modelo con una precisión ("R^2") de 1 para el Erlotinib, un 0.9973796 para el Rapamycin, un 0.9996306 para el Sunitinib y un 1 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predición de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

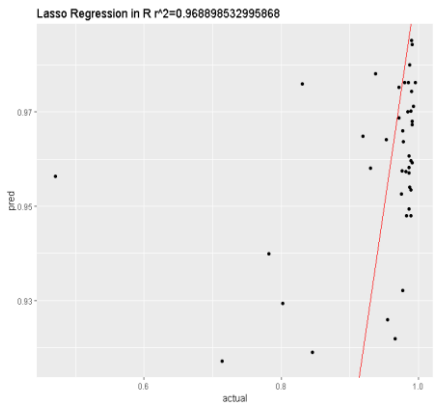


Ilustración 166 : Erlotinib

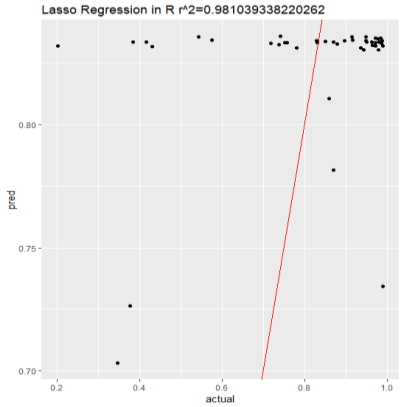


Ilustración 167 : Rapamycin

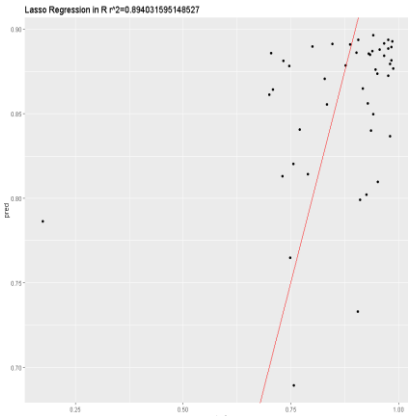


Ilustración 168 : Sunitinib

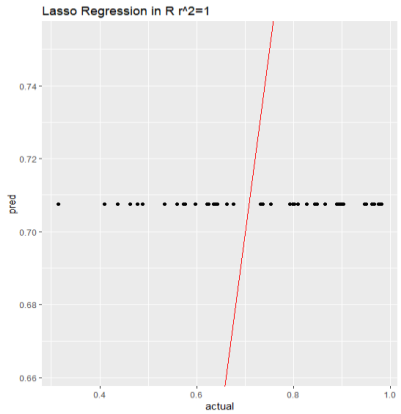
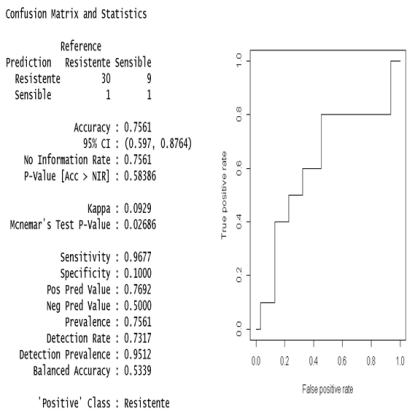


Ilustración 169 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 167 a 170) se ha obtenido un modelo con una precisión ("R^2") del 0.9688985 para el Erlotinib, un 0.9810393 para el Rapamycin, un 0.8940316 para el Sunitinib y un 1 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicion de este cuanto mayor es el valor de R^2.

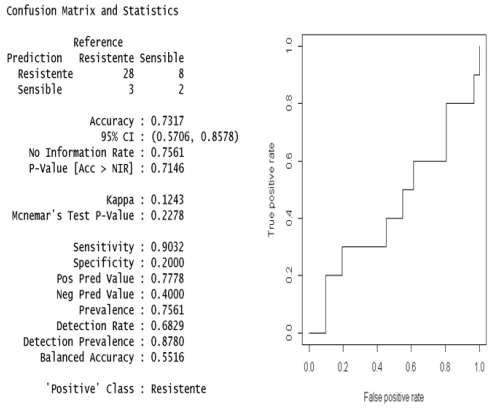
Regresión Elastic Net:

1. Clasificacion 10 iteraciones en validación cruzada:



```
Confusion Matrix and Statistics

          Reference
Prediction  Resistente Sensible
  Resistente         30        9
  Sensible            1        1

               Accuracy : 0.7561
                 95% CI : (0.597, 0.8764)
    No Information Rate : 0.7561
    P-Value [Acc > NIR] : 0.58386

                  Kappa : 0.0929
 Mcnemar's Test P-Value : 0.02686

            Sensitivity : 0.9677
            Specificity : 0.1000
         Pos Pred Value : 0.7692
         Neg Pred Value : 0.5000
             Prevalence : 0.7561
         Detection Rate : 0.7317
   Detection Prevalence : 0.9512
      Balanced Accuracy : 0.5339

       'Positive' Class : Resistente
```


Ilustración 170 : Erlotinib



```
Confusion Matrix and Statistics

          Reference
Prediction  Resistente Sensible
  Resistente         28        8
  Sensible            3        2

               Accuracy : 0.7317
                 95% CI : (0.5706, 0.8578)
    No Information Rate : 0.7561
    P-Value [Acc > NIR] : 0.7146

                  Kappa : 0.1243
 Mcnemar's Test P-Value : 0.2278

            Sensitivity : 0.9032
            Specificity : 0.2000
         Pos Pred Value : 0.7778
         Neg Pred Value : 0.4000
             Prevalence : 0.7561
         Detection Rate : 0.6829
   Detection Prevalence : 0.8780
      Balanced Accuracy : 0.5516

       'Positive' Class : Resistente
```


Ilustración 171 : Rapamycin

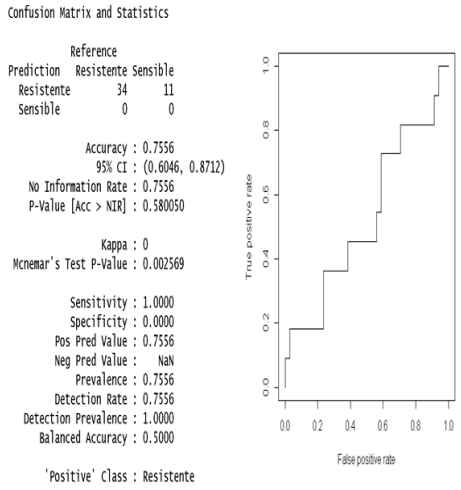


Ilustración 172 : Sunitinib

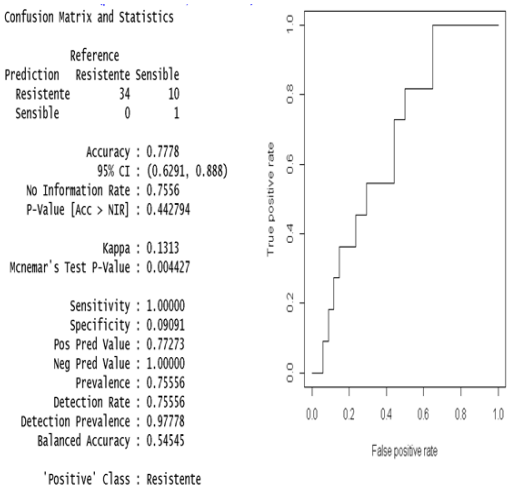


Ilustración 173 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística ElasticNet con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 171 a 174) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7561 para el Erlotinib, un 0,7317 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,7778 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representacion gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

.

2. Clasificación 60/20/20:

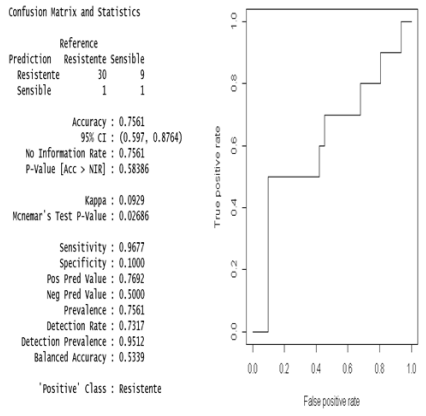


Ilustración 174 : Erlotinib

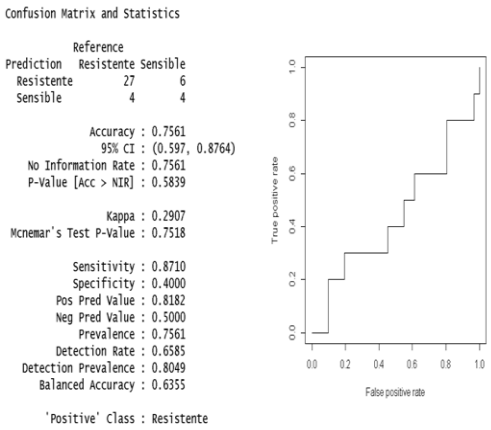


Ilustración 175 : Rapamycin

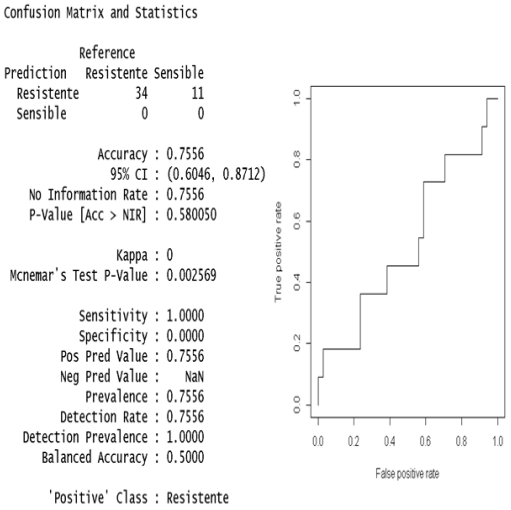


Ilustración 176 : Sunitinib

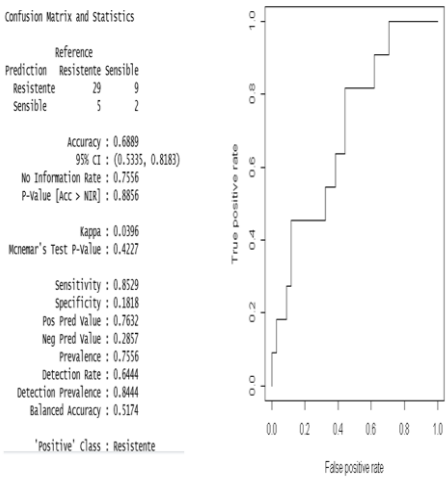


Ilustración 177 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística ElasticNet con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 175 a 178) se ha obtenido un modelo con una precisión ("Accuracy") del 0,7561 para el Erlotinib, un 0,7561 para el Rapamycin, un 0,7556 para el Sunitinib y un 0,6889 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 iteraciones en validación cruzada:

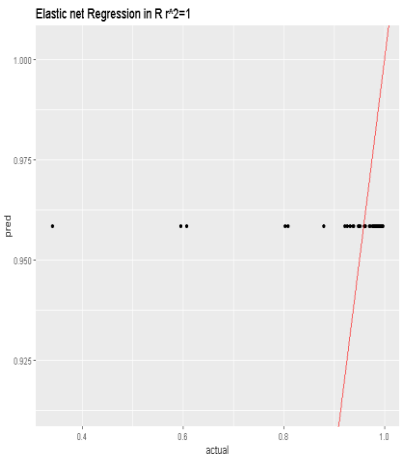


Ilustración 178 : Erlotinib

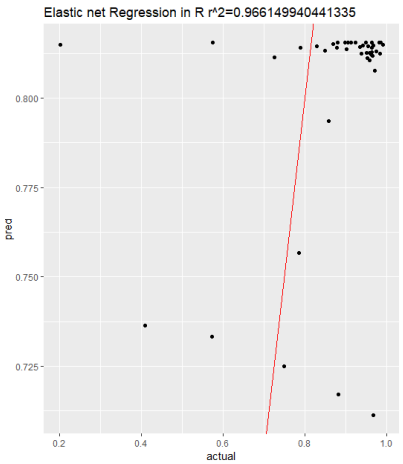


Ilustración 179 : Rapamycin

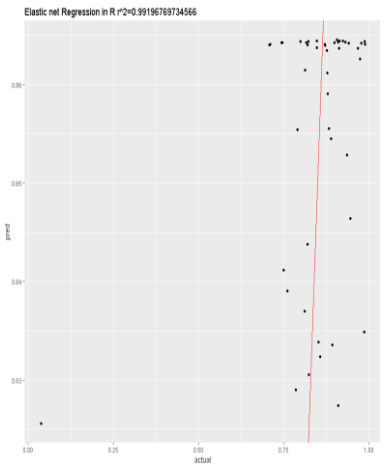


Ilustración 180 : Sunitinib

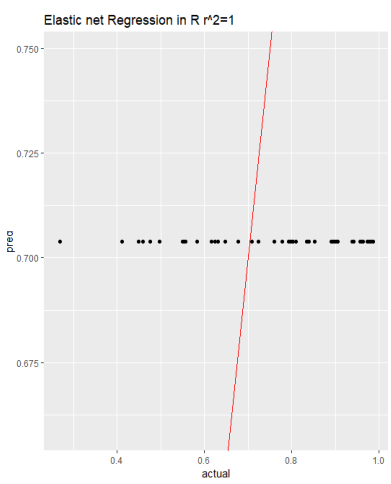


Ilustración 181 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización Elastic Net con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 179 a 182) se ha obtenido un modelo con una precisión ("R^2") de 1 para el Erlotinib, un 0.9661499 para el Rapamycin, un 0.9919677 para el Sunitinib y un 1 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicción de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

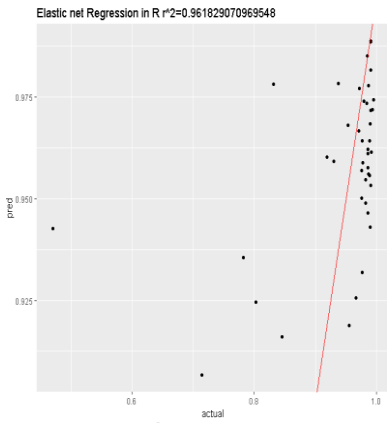


Ilustración 182 : Erlotinib

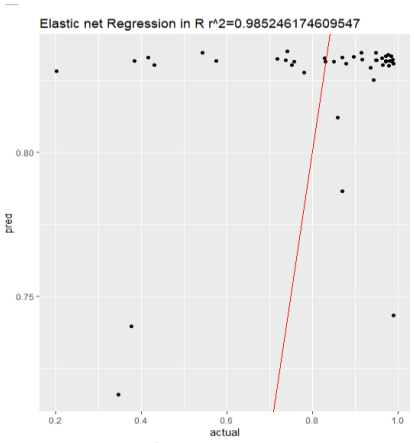


Ilustración 183 : Rapamycin

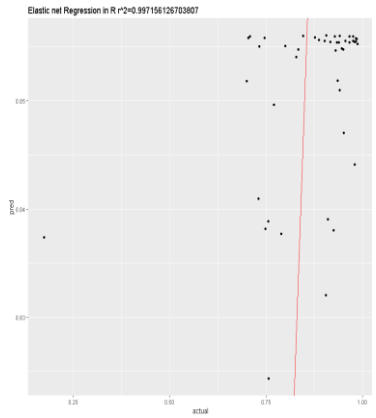


Ilustración 184 : Sunitinib

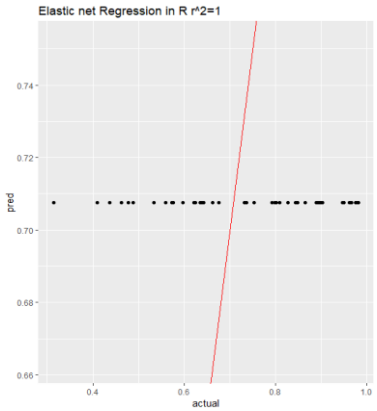


Ilustración 185 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de regularización Elastic Net con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 183 a 186) se ha obtenido un modelo con una precisión ("R^2") del 0.9618291 para el Erlotinib, un 0.9852462 para el Rapamycin, un 0.9971561 para el Sunitinib y un 1 para el Paclitaxel. Representados posteriormente en la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicción de este cuanto mayor es el valor de R^2.

SVM:

1. Clasificación 10 iteraciones en validación cruzada:

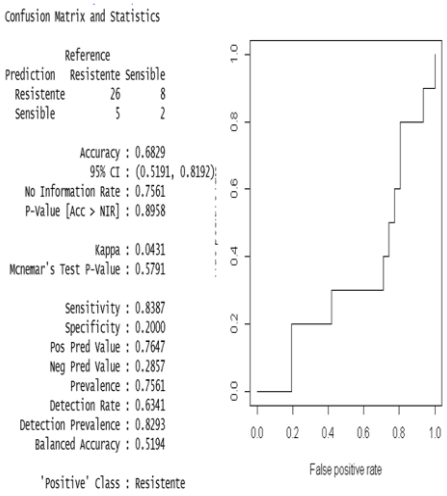


Ilustración 186 : Erlotinib

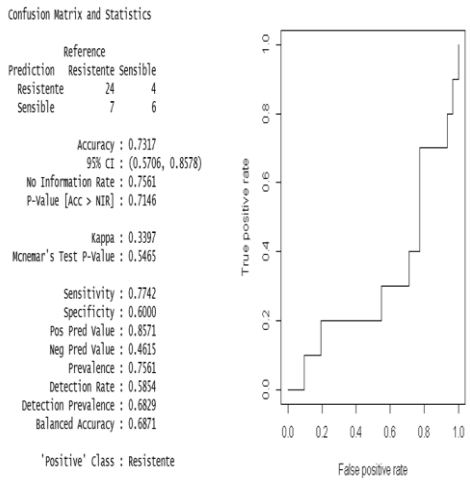


Ilustración 187 : Rapamycin

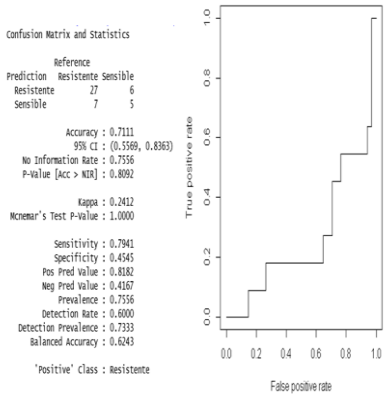


Ilustración 188 : Sunitinib

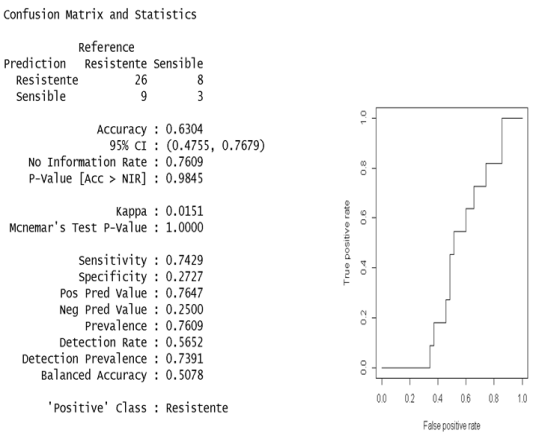


Ilustración 189 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 187 a 190) se ha obtenido un modelo con una precisión ("Accuracy") del 0,6829 para el Erlotinib, un 0,7317 para el Rapamycin, un 0,7111 para el Sunitinib y un 0,6304 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representación gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea major será el modelo) o el punto más cercano al punto (0,1).

2. Clasificación 60/20/20:

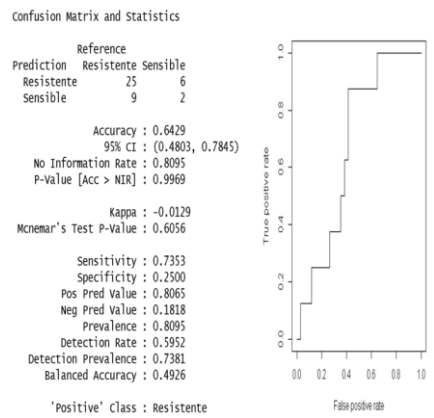


Ilustración 190 : Erlotinib

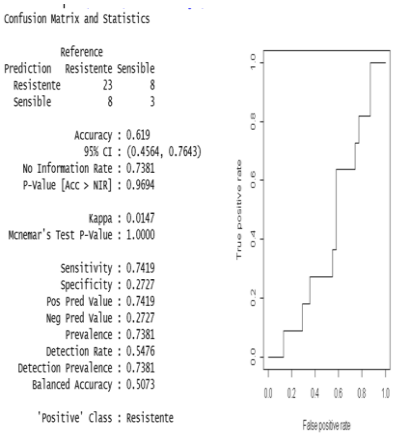


Ilustración 191 : Rapamycin

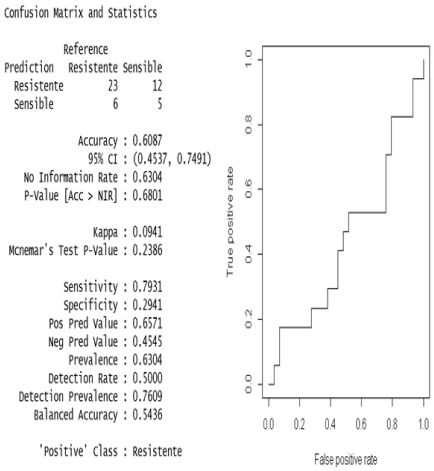


Ilustración 192 : Sunitinib

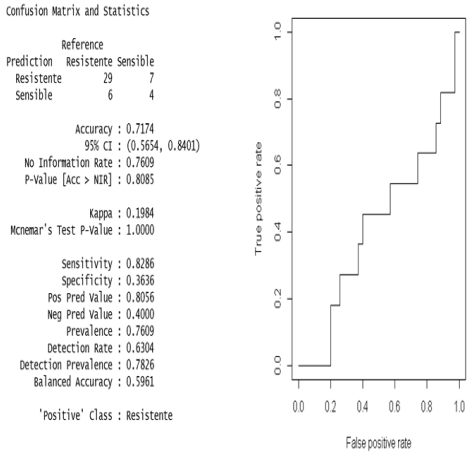


Ilustración 193 : Paclitaxel

Para los datos discretizados entrenados mediante el modelo ajustado de SVM lineal con hiperpárametos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 191 a 194) se ha obtenido un modelo con una precisión ("Accuracy") del 0,6429 para el Erlotinib, un 0,619 para el Rapamycin, un 0,6087 para el Sunitinib y un 0,7174 para el Paclitaxel. Representados posteriormente en las curvas Roc vemos la representacion gráfica, siendo el eje "x" el ratio de falsos positivos y el eje "y" el ratio de verdaderos positivos) de la precisión de cada modelo mencionado pudiendo apreciar la efectividad de estos para predecir por el Área bajo la curva Roc (cuanto mayor sea mejor será el modelo) o el punto más cercano al punto (0,1).

3. Regresión 10 iteraciones en validación cruzada:

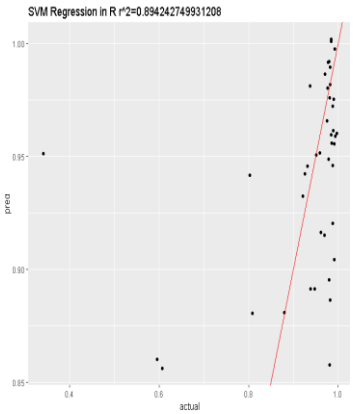


Ilustración 194 : Erlotinib

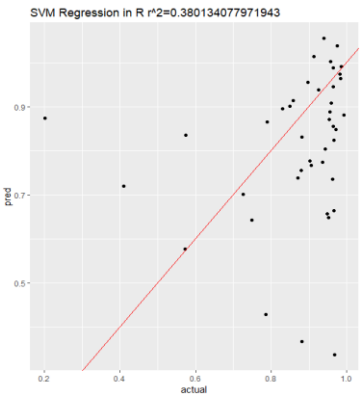


Ilustración 195 : Rapamycin

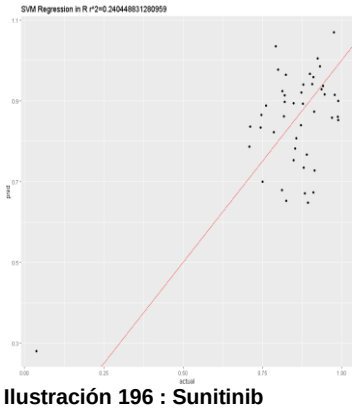


Ilustración 196 : Sunitinib

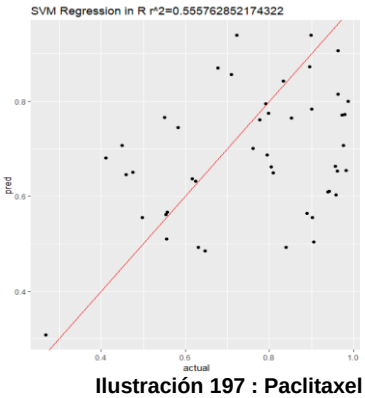


Ilustración 197 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada (Figuras 195 a 198) se ha obtenido un modelo con una precisión ("R^2") del 0,89424274 para el Erlotinib, un 0.3801341 para el Rapamycin, un 0.2404488 para el Sunitinib y un 0.5557629 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la precisión de cada modelo ajustado para cada fármaco siendo mejor la predicción de este cuanto mayor es el valor de R^2.

4. Regresión 60/20/20:

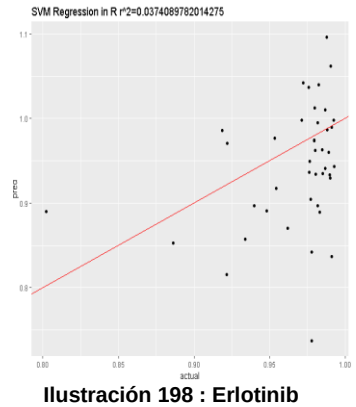


Ilustración 198 : Erlotinib

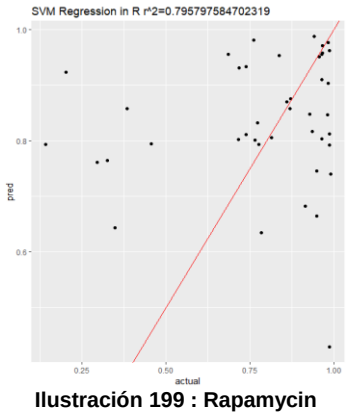


Ilustración 199 : Rapamycin

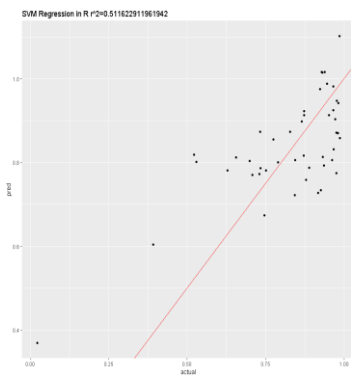


Ilustración 200 : Sunitinib

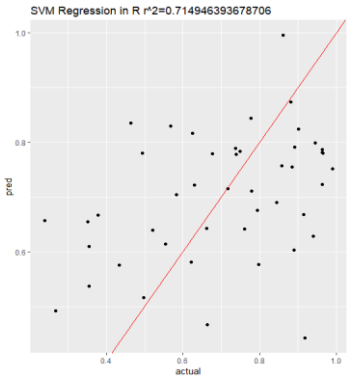


Ilustración 201 : Paclitaxel

Para los datos continuos entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 (Figuras 199 a 202) se ha obtenido un modelo con una precisión ("R^2") del 0.03740898 para el Erlotinib, un 0,150022645209041 para el Rapamycin, un 0.5116229 para el Sunitinib y un 0.7149464 para el Paclitaxel. Representados posteriormente la recta correspondiente a cada valor de R^2 en un gráfico de puntos se puede apreciar la preccisión de cada modelo ajustado para cada fármaco siendo major la predicción de este cuanto mayor es el valor de R^2.

Discusión:

Observando los resultados obtenidos he podido apreciar que en los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Sunitinib con una precisión del 77,78% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos discretizados entrenados mediante el modelo ajustado de Random forest con hiperparámetos seleccionados mediante H2o y con una partición 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Paclitaxel con una precisión del 80,95% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Erlotinib con un valor de

0,96463882269237 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante H2o y con una partición 60/20/20 se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Paclitaxel con un valor de 0.9305795 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Paclitaxel con una precisión del 76,09% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos discretizados entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Erlotinib con una precisión del 83,3% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el menor valor de R^2 ha sido al predecir la respuesta para el fármaco Erlotinib con un valor de 0.9870176 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de random forest con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que el menor valor de R^2 ha sido al predecir la respuesta para el fármaco Erlotinib con un valor de 0.9827496 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para los fármacos Erlotinib y Rapamycin con una precisión del 75,61 % en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para los fármacos Erlotinib y Rapamycin con una precisión del 75,61 % en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos continuos entrenados mediante el modelo ajustado de regularización Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el mayor valor de R^2 ha sido al predecir la respuesta para los fármacos Erlotinib, Rapamycin y Sunitinib con un valor de 1 viendo también mediante la posición que ocupa su

recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de regularización Ridge con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Paclitaxel con un valor de 0.9999989 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para los fármacos Paclitaxel y Rapamycin con una precisión del 77,78% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Erlotinib con una precisión del 87,64% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta.

Para los datos continuos entrenados mediante el modelo ajustado de regularización LASSO con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el mayor valor de R^2 ha sido al predecir la respuesta para los fármacos Erlotinib y Paclitaxel con un valor de 1 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de regularización LASSO con hiperparámetos seleccionados mediante RandomSearch y con una particion 60/20/20 se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Paclitaxel con un valor de 1 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística ElasticNet con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Paclitaxel con una precisión del 77,78% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta.

Para los datos discretizados entrenados mediante el modelo ajustado de regularización logística ElasticNet con hiperparámetos seleccionados mediante RandomSearch y con una particion 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para los fármacos Erlotinib y Rapamycin con una precisión del 75,61% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos continuos entrenados mediante el modelo ajustado de regularización Elastic Net con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el mayor valor de R^2 ha sido al predecir la respuesta para los fármacos Erlotinib y Paclitaxel con un valor de 1 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de regularización Elastic Net con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Paclitaxel con un valor de 1 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos discretizados entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Rapamycin con una precisión del 73,17% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos discretizados entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que la mayor precisión ha sido al predecir la respuesta para el fármaco Paclitaxel con una precisión del 71,74% en sus predicciones de que sean ciertas viendo también mediante la posición que ocupa su curva ROC que efectivamente posee el mayor área bajo esta. Dicho esto se puede observar que la precisión obtenida al entrenar este tipo de modelo para estos 4 fármacos no dista mucho entre ellas.

Para los datos continuos entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 80/20 en 10 iteraciones en validación cruzada se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Erlotinib con un valor de 0,89424274 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

Para los datos continuos entrenados mediante el modelo ajustado de SVM lineal con hiperparámetos seleccionados mediante RandomSearch y con una partición 60/20/20 se observa que el mayor valor de R^2 ha sido al predecir la respuesta para el fármaco Paclitaxel con un valor de 0.7149464 viendo también mediante la posición que ocupa su recta con respecto a las dos distribuciones de puntos correspondientes a cada clase (sensible/resistente) efectivamente posee el mayor de los errores cuadráticos y por tanto mayor bondad y ajuste.

3. Conclusiones:

Observando los resultados se puede concluir con que el fármaco que mejor se predice es el Erlotinib con un 87,64% de precisión acertando predicciones con una partición de datos 60/20/20, usando un 20% de esta partición como datos de validación en la búsqueda de hiperparámetros mediante Random Search en Random Forest cuando los valores Auc, usados como predictores a lo largo de todo el proyecto, están discretizados.

Cuando los predictores han sido entrenados como valores continuos el mejor valor de R^2 obtenido ha sido 1 correspondiente a la predicción de respuesta para los fármacos Erlotinib, Rapamycin y Sunitinib mediante el modelo ajustado de regularización de Ridge con hiperparámetos seleccionados mediante RandomSearch y con una particion 80/20 en 10 iteraciones en validación cruzada.

Con estas dos conclusiones se puede deducir que de los 4 fármacos escogidos al azar para desarrollar un algoritmo de machine learning de predicción de respuesta, el que mejor es capaz de predecirse es el Erlotinib para datos categorizados, Erlotinib, Rapamycin y Paclitaxel con datos continuos; la mejor partición de datos sería mediante la división de estos en un 20% de validación para la búsqueda de hiperparámetros, un 80% para el entrenamiento del modelo ajustado y un 20% para predecir o testear el modelo ajustado siendo categóricos y 80/20 con 10 iteraciones en validación cruzada cuando son continuos; y los mejores algoritmos en la predicción de respuesta a fámarcos quimioterapeúticos, hasta ahora, son el Random Forest con valores discretizados y la regresión Ridge para valores continuos.

Los objetivos planteados inicialmente, tanto los parciales como los principales, han sido cumplidos satisfactoriamente en el período de tiempo planteado en el plan de trabajo. Este plan de trabajo se planteó con suficiente tiempo para que al iniciar la asignatura del proyecto de fin de grado ya estuviera planteado y organizado con la empresa pública Centro Nacional de Investigaciones Oncológicas por lo que ha habido muy pocos imprevistos, ya que se disponía de suficiente tiempo en la mayoría de tareas y los posibles problemas que pudieran surgir eran paliados con tiempo suficiente hasta el próximo inicio de tarea.

El único problema que ha surgido a la hora de desarrollar el pleno cumplimiento de los objetivos ha sido el cambio de trabajo de mi tutor a mediados de Mayo por motivos políticos. Esto ha desembocado en que las últimas dos semanas de Mayo no ha sido posible la supervisión de mi trabajo de forma tan continuada como hasta entonces haciendo que el desarrollo de los algoritmos ajenos a Random Forest hayan sido ralentizados y por consiguiente teniendo que alargar varias horas la dedicación a estas tareas durante los fines de semana, los cuales han estado libres el resto del tiempo de desarrollo del proyecto.

En un futuro y con un proyecto más ambicioso se puede estudiar una mayor combinación de fármacos, algortimos de machine learning y formas de división de datos para el desarrollo de su correspondiente modelo implementando, además, un mayor número de hiperparámetros y parámetros a ajustar para obtener una conclusiones más precisas y certeras sobre el mejor algoritmo de predicción de respuesta a fármacos quimioterapeúticos en líneas celulares tumorales mediante el uso de valores AUC, o incluso otros posibles bioindicadores que puedan servir como posibles predictores de respuesta.

4. Glosario:

- **Línea celular de cultivo:** Células de un tipo único (humano, animal o vegetal) que se han adaptado para crecer continuamente en el laboratorio y que se usan en investigación[23].

- **Fármaco:** Medicamento legal que se usa para prevenir, tratar o aliviar los síntomas de una enfermedad o una afección anormal[24].

- **IC50:** Medida de la potencia de una sustancia para inhibir una función biológica o bioquímica específica. Medida cuantitativa indica la cantidad de un medicamento u otra sustancia (inhibidor) que se necesita para inhibir un proceso biológico determinado (o un componente de un proceso, es decir, una enzima, célula, receptor celular o microorganismo) a la mitad. Los valores se expresan típicamente como concentración molar[25].

- **Valor Auc:** Valor que representa el área bajo la curva de dosis-respuesta ajustada.

- **Metadato:** Literalmente sobre dato, son datos que describen otros datos[26].

- **RNA-Seq:** También llamada WTSS, utiliza la secuenciación de próxima generación (NGS) para revelar la presencia y cantidad de ARN en una muestra biológica en un momento dado[27].
-
- **GTEx:** Es un repositorio contiene pipelines de análisis para: Alineación, cuantificación y control de calidad del ARN-secuencia; mapeo y anotación eQTL; cuantificación de expresiones específicas de alelos; y generación de la anotación colapsada utilizada para la cuantificación de la expresión a nivel genético[28].

- **Expresión de un gen:** La expresión génica es el proceso por el cual la información de un gen se utiliza en la síntesis de un producto genético funcional. Estos productos suelen ser proteínas, pero en los genes que no codifican las proteínas, como los genes de transferencia de ARN (tRNA) o pequeños genes de ARN nuclear (snRNA), el producto es un ARN funcional.

- **Linux:** Es una familia de sistemas operativos de código abierto tipo Unix basados en el núcleo de Linux, un núcleo de sistema operativo lanzado por primera vez el 17 de septiembre de 1991 por Linus Torvalds. Linux suele estar empaquetado en una distribución Linux (o distro para abreviar) [29].

- **Clustering:** El análisis de conglomerados o clustering es la tarea de agrupar un conjunto de objetos de tal manera que los objetos del mismo grupo (llamados conglomerados) sean más similares (en cierto sentido) entre sí que con los de otros grupos (conglomerados). Es una tarea principal de la minería de datos exploratoria y una técnica común para el análisis de datos estadísticos, utilizada en muchos campos, incluyendo el aprendizaje automático, el reconocimiento de patrones, el análisis de imágenes, la recuperación de información, la bioinformática, la compresión de datos y los gráficos por computadora[30].

- **Erlotinib:** Es un inhibidor de la tirosina quinasa del receptor del factor de crecimiento epidérmico (EGFR) que se utiliza en el tratamiento del cáncer de pulmón de células no pequeñas, el cáncer de páncreas y varios otros tipos de cáncer. Normalmente se comercializa bajo el nombre comercial de Tarceva[31].

- **Rapamycin:** Compuesto macrólido obtenido de Streptomyces hygroscopicus que actúa bloqueando selectivamente la activación transcripcional de las citocinas, inhibiendo así la producción de citocinas[32].

- **Sunitinib:** Es un inhibidor de la tirosina quinasa (RTK) de moléculas pequeñas y múltiples[33].

- **Paclitaxel:** Es un agente quimioterapéutico comercializado bajo la marca Taxol entre otros. Usado como tratamiento para varios tipos de cáncer, el paclitaxel es un inhibidor mitótico que fue aislado por primera vez en 1971 de la corteza del tejo del Pacífico que contiene hongos endofitos que sintetizan el paclitaxel[34].

- **Dataframe:** Es una tabla o una estructura bidimensional en la que cada columna contiene valores de una variable y cada fila contiene un conjunto de valores de cada columna[35].

- **DepMap ID:** ID asociada a cada experimento registrado por el consorcio Broad Institute DepMap.

- **Cosmic ID:** ID asociada a cada experimento registrado en la base de datos de Cosmic.

- **Na:** Valor que falta es aquel cuyo valor es desconocido. Los valores que faltan se representan en R con el símbolo NA[36].

- **Varianza:** Es una medida de dispersión definida como la esperanza del cuadrado de la desviación de dicha variable respecto a su media[37].

- **Variable:** Una variable o escalar es una ubicación de almacenamiento (identificada por una dirección de memoria) emparejada con un nombre simbólico asociado (un identificador), que contiene cierta cantidad de información conocida o desconocida a la que se refiere como un valor[38].

- **Medicion dicotómica:** Que se puede responder con si o no.

- **discretización:** Proceso de transferir funciones continuas, modelos, variables y ecuaciones a contrapartes discretas[39].

- **Cuartiles:** Son los tres valores de la variable que dividen a un conjunto de datos ordenados en cuatro partes iguales[40].

- **Algoritmo:** En matemáticas e informática, un algoritmo es una especificación inequívoca de cómo resolver una clase de problemas. Los algoritmos pueden realizar cálculos, procesamiento de datos, razonamiento automatizado y otras tareas[41].

- **Machine Learning:** El aprendizaje automático es una aplicación de la inteligencia artificial (IA) que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia sin estar programados explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden acceder a los datos y utilizarlos para aprender por sí mismos[42].

- **Sobreajuste:** En aprendizaje automático, el sobreajuste (también es frecuente emplear el término en inglés overfitting) es el efecto de sobreentrenar un algoritmo de aprendizaje con unos ciertos datos para los que se conoce el resultado deseado[43].

- **Media aritmética:** También llamada promedio o media, de un conjunto infinito de números es el valor característico de una serie de datos cuantitativos, objeto de estudio que parte del principio de la esperanza matemática o valor esperado, se obtiene a partir de la suma de todos sus valores dividida entre el número de sumandos[44].

- **Bagging:** La agregación de bootstrap, también llamada bagging, es un meta-algoritmo del conjunto de aprendizaje de la máquina diseñado para mejorar la estabilidad y la precisión de los algoritmos de aprendizaje de la máquina utilizados en la clasificación estadística y la regresión[45].

- **RMSE:** El error cuadrático medio es la desviación estándar de los residuos (errores de predicción). Los residuos son una medida de cuán lejos están de los puntos de datos de la línea de regresión; RMSE es una medida de cuán esparcidos están estos residuos. En otras palabras, le dice cuán concentrados están los datos alrededor de la línea de mejor ajuste. El error cuadrático medio de la raíz se utiliza comúnmente en climatología, pronóstico y análisis de regresión para verificar los resultados experimentales[46].

- **MAE:** En estadística, el error absoluto medio es una medida de la diferencia entre dos variables continuas. Supongamos que X e Y son variables de observaciones emparejadas que expresan el mismo fenómeno[47].

- **Kappa:** El coeficiente kappa de Cohen (κ) es una estadística que mide el acuerdo entre calificadores para elementos cualitativos (categóricos). Generalmente se piensa que es una medida más robusta que el simple cálculo de un acuerdo porcentual, ya que κ tiene en cuenta la posibilidad de que el acuerdo ocurra por casualidad[48].

- **El análisis de regresión múltiple** es una técnica de análisis multivariable en el que se establece una relación funcional entre una variable dependiente o a explicar y una serie de variables independientes o explicativas, en la que se estiman los coeficientes de regresión que determinan el efecto que las variaciones de las variables independientes tienen sobre el comportamiento de la variable dependiente. El modelo más utilizado es el modelo lineal, pues es el que requiere estimar un menor número de parámetros[49].

- **Sensibilidad (equivalente a la tasa de positivos verdaderos):** Proporción de casos positivos que están bien detectadas por la prueba. La definición matemática es: Sensibilidad = VP / (VP + FN) [50].

- **Especificidad (también llamada Tasa de verdaderos negativos):** Proporción de casos negativos que son bien detectadas por la prueba. La definición matemática es: Especificidad = VN / (VN + FP) [50].

- **Tasa de falsos positivos (FPR):** Proporción de casos negativos que la prueba detecta como positivos[50].

- **Tasa de verdaderos positivos (VPR):** Proporción de casos negativos que la prueba detecta como positivos[50].

5. Bibliografía:

Datasets:

1. Broad Institute DepMap:https://depmap.org/portal/ (20/2/2019)
2. GDSC: https://www.cancerrxgene.org/ (25/2/2019)
3. https://www.cancer.gov/espanol/publicaciones/diccionario/def/linea-celular-de-cultivo (27/2/2019)

Artículos:

4. Paul Geeleher, Nancy J Cox and R Stephanie Huang, Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines, Genome Biology 2014, 15:R47.

5. IN SOCK JANG1, ELIAS CHAIBUB NETO, JUSTIN GUINNEY, STEPHEN H. FRIEND, ADAM A. MARGOLIN1, SYSTEMATIC ASSESSMENT OF ANALYTICAL METHODS FOR DRUG SENSITIVITY PREDICTION FROM CANCER CELL LINE DATA, Pacific Symposium on Biocomputing 2014.

6. Menden MP, Iorio F, Garnett M, McDermott U, Benes CH, et al. (2013) Machine Learning Prediction of Cancer Cell Sensitivity to Drugs Based on Genomic and Chemical Properties. PLoS ONE 8(4): e61318. doi:10.1371/journal.pone.0061318

7. Costello JC1, Heiser LM2, Georgii E3, Gönen M4, Menden MP5, Wang NJ6, Bansal M7, Ammad-ud-din M4, Hintsanen P8, Khan SA4, Mpindi JP8, Kallioniemi O8, Honkela A9, Aittokallio T8, Wennerberg K8; NCI DREAM Community, Collins JJ10, Gallahan D11, Singer D11, Saez-Rodriguez J5, Kaski S12, Gray JW6, Stolovitzky G13, A community effort to assess and improve drug sensitivity prediction algorithms, Nat Biotechnol. 2014 Dec;32(12):1202-12. doi: 10.1038/nbt.2877. Epub 2014 Jun 1.

8. Jordi Barretina, Giordano Caponigro[…]Levi A. Garraway, The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity,Nature volume 483, pages 603–607 (29 March 2012).

9. Mathew J. Garnett, Elena J. Edelman[…]Cyril H. Benes, Systematic identification of genomic markers of drug sensitivity in cancer cells, Nature volume 483, pages 570–575 (29 March 2012).

Libros:

10. Julian J.Faraway, Linear Models with R, 2º edición, Editoral CRCPress Taylor y Francis Group, 711 Third Avenue, NY 10017.

11. Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction to Statistical Learning with applications in R, Springer, New York 2013.

URLS:

12. https://www.cancer.gov/espanol/publicaciones/diccionario/def/farmaco
13. https://github.com/broadinstitute/gtex-pipeline

14. https://www.drugbank.ca/drugs/DB00530
15. https://www.drugbank.ca/drugs/DB00877
16. https://www.drugbank.ca/drugs/DB01268
17. https://www.drugbank.ca/drugs/DB01229
18. https://www.tutorialspoint.com/r/r_data_frames.htm
19. https://faculty.nps.edu/sebuttre/home/R/missings.html
20. https://www.superprof.es/apuntes/escolar/matematicas/estadistica/descriptiva/cuartiles.html
21. https://www.expertsystem.com/machine-learning-definition/
22. https://www.statisticshowto.datasciencecentral.com/rmse/
23. https://www.cancer.gov/espanol/publicaciones/diccionario/def/linea-celular-de-cultivo (28/5/2019)
24. https://www.cancer.gov/espanol/publicaciones/diccionario/def/farmaco (28/5/2019)
25. https://en.wikipedia.org/wiki/IC50 (28/5/2019)
26. https://es.wikipedia.org/wiki/Metadatos (28/5/2019)
27. https://en.wikipedia.org/wiki/RNA-Seq (28/5/2019)
28. https://github.com/broadinstitute/gtex-pipeline (28/5/2019)
29. https://en.wikipedia.org/wiki/Linux (28/5/2019)
30. https://en.wikipedia.org/wiki/Cluster_analysis (28/5/2019)
31. https://www.drugbank.ca/drugs/DB00530 (28/5/2019)
32. https://www.drugbank.ca/drugs/DB00877 (1/6/2019)
33. https://www.drugbank.ca/drugs/DB01268 (1/6/2019)
34. https://www.drugbank.ca/drugs/DB01229 (1/6/2019)
35. https://www.tutorialspoint.com/r/r_data_frames.htm (1/6/2019)
36. https://faculty.nps.edu/sebuttre/home/R/missings.html (1/6/2019)
37. https://es.wikipedia.org/wiki/Varianza (1/6/2019)
38. https://en.wikipedia.org/wiki/Variable_(computer_science) (1/6/2019)
39. https://es.wikipedia.org/wiki/Discretizaci%C3%B3n (1/6/2019)
40. https://www.superprof.es/apuntes/escolar/matematicas/estadistica/descriptiva/cuartiles.html (1/6/2019)
41. https://en.wikipedia.org/wiki/Algorithm (1/6/2019)
42. https://www.expertsystem.com/machine-learning-definition/(1/6/2019)
43. https://es.wikipedia.org/wiki/Sobreajuste (1/6/2019)
44. https://es.wikipedia.org/wiki/Media_aritm%C3%A9tica (1/6/2019)
45. https://en.wikipedia.org/wiki/Bootstrap_aggregating (1/6/2019)
46. https://www.statisticshowto.datasciencecentral.com/rmse/ (1/6/2019)
47. https://en.wikipedia.org/wiki/Mean_absolute_error (1/6/2019)
48. https://en.wikipedia.org/wiki/Cohen%27s_kappa (1/6/2019)
49. http://www.eumed.net/tesisdoctorales/2008/mrl/Analisis%20de%20regresion%20multiple.htm (1/6/2019)
50. https://www.xlstat.com/es/soluciones/funciones/analisis-detallado-de-sensibilidad-y-especificidad (1/6/2019)
51. https://www.xlstat.com/es/soluciones/funciones/analisis-detallado-de-sensibilidad-y-especificidad (1/6/2019)
52. https://www.xlstat.com/es/soluciones/funciones/analisis-detallado-de-sensibilidad-y-especificidad (1/6/2019)
53. https://www.xlstat.com/es/soluciones/funciones/analisis-detallado-de-sensibilidad-y-especificidad (1/6/2019)

54. https://en.wikipedia.org/wiki/IC50
55. https://es.wikipedia.org/wiki/Metadatos
56. https://en.wikipedia.org/wiki/RNA-Seq
57. https://en.wikipedia.org/wiki/Linux
58. https://en.wikipedia.org/wiki/Cluster_analysis
59. https://es.wikipedia.org/wiki/Varianza
60. https://en.wikipedia.org/wiki/Variable_(computer_science)
61. https://es.wikipedia.org/wiki/Discretizaci%C3%B3n
62. https://en.wikipedia.org/wiki/Algorithm
63. https://es.wikipedia.org/wiki/Sobreajuste
64. https://es.wikipedia.org/wiki/Media_aritm%C3%A9tica

65. https://en.wikipedia.org/wiki/Bootstrap_aggregating
66. https://en.wikipedia.org/wiki/Mean_absolute_error
67. https://en.wikipedia.org/wiki/Cohen%27s_kappa

Librerías:

68. https://cran.r-project.org/web/packages/ROCR/ROCR.pdf (26/5/2019)
69. https://cran.r-project.org/web/packages/e1071/index.html (26/5/2019)
70. https://cran.r-project.org/web/packages/glmnet/index.html (26/5/2019)
71. https://cran.r-project.org/web/packages/caret/index.html (26/5/2019)
72. https://cran.r-project.org/web/packages/miscTools/index.html (26/5/2019)
73. https://cran.r-project.org/web/packages/randomForest/randomForest.pdf (26/5/2019)
74. https://cran.r-project.org/web/packages/h2o/h2o.pdf (26/5/2019)
75. https://cran.r-project.org/web/packages/readxl/readxl.pdf (26/5/2019)
76. https://cran.r-project.org/web/packages/data.table/index.html (26/5/2019)
77. https://cran.r-project.org/web/packages/resample/index.html (26/5/2019)
78. https://www.rdocumentation.org/packages/plyr/versions/1.8.4 (26/5/2019)
79. https://cran.r-project.org/web/packages/caTools/index.html (26/5/2019)
80. https://www.rdocumentation.org/packages/ggplot2/versions/3.1.1(26/5/2019)

Otras:

81. https://en.wikipedia.org/wiki/Random_forest (30/5/2019)

6. Anexos:

Tutorial de instalación de R en ubuntu con librerías:

Requisitos previos:
Para seguir con este tutorial, necesitará un servidor Ubuntu 18.04 con:
-al menos 1 GB de RAM
-un usuario no root con privilegios -sudo

Paso 1 - Instalación de R:
Debido a que R es un proyecto de rápida evolución, la última versión estable no siempre está disponible desde los repositorios de Ubuntu, así que empezaremos por añadir el repositorio externo mantenido por CRAN.

Primero agreguemos la clave GPG relevante:

sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys E298A3A825C0D65DFD57CBB651716619E084DAB9

Una vez que tengamos la clave de confianza, podremos añadir el repositorio. Tenga en cuenta que si no está usando 18.04, puede encontrar el repositorio relevante de la lista R Project Ubuntu, llamado para cada versión.

sudo add-apt-repository 'deb https://cloud.r-project.org/bin/linux/ubuntu bionic-cran35/'

Ahora, necesitaremos ejecutar la actualización después de esto para incluir los manifiestos de paquetes del nuevo repositorio.

sudo apt update

En este punto, estamos listos para instalar R con el siguiente comando.

sudo apt install r-base

Si se le pide que confirme la instalación, pulse y para continuar.

En el momento de escribir este artículo, la última versión estable de R de CRAN es la 3.5.1, que se muestra cuando se inicia R.

Como estamos planeando instalar un paquete de ejemplo para cada usuario del sistema, iniciaremos R como root para que las librerías estén disponibles para todos los usuarios automáticamente. Alternativamente, si ejecuta el comando R sin el comando sudo, se puede configurar una biblioteca personal para su usuario.

sudo -i R

Paso 2 - Instalación de los paquetes R desde CRAN:

Parte de la fuerza de R es su abundancia disponible de paquetes adicionales. Para propósitos de demostración, instalaremos “caret”, una librería que produce gráficos ASCII que incluyen scatterplot, line plot, density plot, acf y bar charts:

install.packages('caret')

Para más información:
(https://www.r-project.org/)