Aprenentatge Automàtic (APA)

Enero 2023

Práctica de regresión o clasificación

Oriol Gavin
Jairo El Yazidi

1.- Trabajo a realizar y objetivos

Para poder realizar esta práctica lo primero y más fundamental fue la elección del problema que se pretende resolver, es decir, el conjunto de datos con el que se trabaja, el tipo de predicción qué se quiere hacer (regresión o clasificación) y sobre qué variable o clases. Después de una búsqueda de un conjunto de datos que cumpliese con todas las características exigidas a través de las diversas webs que se nos proporcionaron para ello dimos con uno llamado *Student Performance Data Set*. Se trata de un conjunto con 649 instancias (más de 500), 33 variables (más de 10) numéricas y categóricas, con lo cual se requiere algún tipo de preprocesamiento, y que además está concebido tanto para regresión como clasificación. La elección de este *dataset* además nos pareció de lo más apropiada porque sus datos son de fácil comprensión a la hora de explicar los resultados y porque se trataba de las calificaciones de un conjunto de estudiantes portugueses, siendo el mundo estudiantil un lugar en el que nosotros todavía estamos inmersos.

Para nuestro caso la decisión conjunta fue que el trabajo fuese únicamente sobre el subconjunto "student-mat", que corresponde a las notas de la asignatura de matemáticas, la cual nos es mucho más familiar e interesante que el otro conjunto con las notas de la asignatura de lengua portuguesa. Además decidimos que consistiera en un ejercicio de regresión sobre la variable objetivo denominada como "G3". Esta variable corresponde a la nota sobre 20 obtenida en el tercer trimestre del curso académico y el conjunto también cuenta con sus homólogos G1 y G2. El objetivo principal entonces consiste en predecir cuál es la nota del último trimestre en función de variables predictoras como:

- La escuela en la que el alumno estudia, de entre las dos que se contemplan.
- El sexo y la edad.
- El tipo de zona en la que vive, urbana o rural.
- El tamaño de su familia, pequeña o grande.
- Si los padres están juntos o separados.
- El nivel de estudios de la madre y del padre.
- El tipo de empleo de la madre y del padre.
- La razón por la que eligieron tal escuela.
- La persona de la familia que ejerce de tutor legal.
- El tiempo invertido en ir de casa a la escuela y viceversa.
- Las horas invertidas en estudiar.
- El número de suspensos previos.
- Si reciben ayuda académica de alguien fuera de la escuela y de la familia.
- Si reciben clases particulares pagadas en la asignatura en cuestión.
- Si realizan algún tipo de actividad extraescolar.
- Si fueron a la guardería de pequeños.
- Si quieren realizar estudios universitarios en el futuro.
- Si tienen acceso a internet en casa.

- Si tienen pareja.
- La calidad de la relación con la familia.
- El tiempo que dedican al ocio personal y a salir con sus amigos.
- El nivel de consumo de alcohol entre semana y los fines de semana.
- El nivel de salud.
- El número de ausencias en clase.

A través de este informe desarrollaremos todas las etapas por las que hemos pasado para llevar a cabo la mejor regresión posible, que no fue una tarea tan sencilla como pensamos inicialmente, y analizaremos los resultados obtenidos, que tampoco fueron tan buenos como imaginamos pero de los que se pueden sacar importantes conclusiones.

2.- Estudios previos sobre los datos

Una vez seleccionados los datos pasamos a ver algunos análisis hechos sobre estos. Recordamos que los datos corresponden a las notas de varios alumnos de secundaria en matemáticas y lengua portuguesa de dos colegios distintos en Portugal. Los datos fueron recogidos a base de cuestionarios a los alumnos e informes pedidos a las familias. El estudio fue impulsado principalmente por Paulo Cortez, profesor en el *Departamento de Sistemas de Información, Universidad de Minho, Portugal*.

El principal estudio de los datos viene hecho también por él mismo y Alice Silva. En dicho estudio se realiza un análisis tanto de Regresión como de Clasificación. Para ello, según cuentan, han usado la librería de código abierto RMiner. El autor es el propio Paulo y proporciona herramientas para usar muchos de los métodos vistos en el curso de APA: Random Forest, Redes Neuronales, Support Vector Machines, Decision Tree y Naïve Bayes de entre otras más. En concreto, las 5 mencionadas son las usadas en el estudio de Paulo y Alice. Los mejores resultados no vienen siempre dados por el mismo método, por lo que usar varios siempre viene bien.

En resumen, el estudio revela que tanto la predicción como la regresión de la nota ‘G3’ se complica mucho sin las notas de los trimestres anteriores ‘G1’ y ‘G2’. Para poner un ejemplo, con G1 y G2 llegan a conseguir clasificaciones con un 92% de acierto mientras que, sin dichas variables, a duras penas llegan al 70% (para la asignatura de matemáticas).

También revelan la importancia de las variables para el método de Random Forest. Siempre que G1 y G2 están presentes, dichas variables se llevan casi todo el protagonismo, mientras que cuando no están, factores como la cantidad de ausencias del estudiante, o los suspensos previos son los más relevantes.

Nosotros sacamos diferentes conclusiones de este trabajo que nos ayudaron a decidir los diferentes métodos para el remuestreo y para los resultados. Primero decidimos que uno de nuestros métodos de remuestreo sería, tal y como hicieron ellos, el error cuadrático medio para, a parte de comparar resultados, ver si nos acercamos a lo que ellos hicieron (uno de ellos, a parte, queríamos probar con otros más).
Otra conclusión fue que queríamos repetir cuantos menos métodos de regresión mejor, así que decidimos usar solamente el SVM en común (nosotros usamos el SVM RBF, en su proyecto no se especifica cuál se usó), sin embargo, también trataremos de ver la importancia de cada variable así como el árbol de decisión representado visualmente.

Así que, una vez sacadas diferentes decisiones basadas en el proyecto, pasamos a realizar nuestra propia exploración y experimentación.

3.- Exploración de los datos

Lo primero que se tuvo en cuenta relativo a la exploración de los datos, una vez conocidas y entendidas todas las variables del conjunto, fue el preprocesamiento que se llevaría a cabo. Antes de realizar ningún tipo de transformación en los datos que pudiera alterar su aspecto original vimos necesario analizar la presencia de valores perdidos o faltantes y también los anómalos o *outliers*. El primero de estos dos no supuso un problema en absoluto puesto que, como se puede ver en al apartado correspondiente del *notebook*, todas las variables del conjunto tienen valores para todos los ejemplos, lo cual también es mencionado en la documentación del *dataset* elegido.
En lo que respecta a *outliers*, no vimos necesario realizar ningún tipo de transformación de los datos que los pudiera corregir. Las variables categóricas no tienen valores anómalos porque están ceñidas dentro de una pequeña variedad de respuestas posibles y las numéricas, como se puede apreciar en el *boxplot* del cuaderno, o bien tienen muy pocos valores lejos de la media o bien, es el caso de la variable *absences*, eliminar los valores anómalos restaría significado a los datos, ya que si un alumno ha faltado repetidas veces a clase no significa que los datos sean incorrectos sino que ese alumno potencialmente sacará peores notas.

Habiendo visto en el primer apartado cuáles son todos los atributos que se toman en consideración para la regresión no es difícil ver la cantidad de variables categóricas, nominales y binarias es importante. La siguiente fase del preprocesamiento, como bien se puede observar en el *notebook* de la práctica, es precisamente la codificación de estos atributos de manera que puedan ser tratados por los modelos. Nuestra estrategia fue finalmente usar la codificación *one-hot* porque aporta más significado teniendo solo 0 y 1. Si usamos categorías 1, 2, 3, 4, etcétera, el hecho, por ejemplo, de que 2 sea el doble de 1 no tiene verdaderamente

ese significado sino simplemente otra categoría sin relación, así que quisimos evitar esa interpretación por parte de los modelos.

Para una posible selección o extracción de características el paso previo era visualizar correctamente la importancia que todas ellas toman a la hora de realizar una predicción. La manera más sencilla y clara de ver esta información es mediante el Análisis de Componentes Principales o PCA, así que el primer paso era estandarizar los datos para que todos estuvieran comprendidos dentro del rango 0-1. Este es un requisito previo para poder calcular el PCA adecuadamente pero además este nuevo conjunto con los datos normalizados también nos será útil para el funcionamiento de algunos de los modelos que entrenaremos más adelante, ya que requieren de la misma transformación.
El ajuste del método PCA dio lugar a dos nuevas gráficas que se pueden observar en el *notebook*, el *scree plot* y el de variancia explicada. Estas dos figuras nos hacen ver que hay una gran diferencia entre atributos en lo que respecta a la importancia. Ninguna de las variables se lleva todo el protagonismo pero algunas de ellas tienen aportaciones de prácticamente cero. En la documentación del *dataset* también hay un dato interesante acerca de esto, y es que las variables G1 y G2 tienen tanta correlación con la variable objetivo que su importancia es mucho mayor a nivel proporcional. Este apunte sumado a los resultados observados del PCA nos llevaron a eliminar tanto G1 como G2 del conjunto de variables predictoras, en este caso no por ser irrelevantes o redundantes sino por no querer simplificar demasiado la faena.
El otro dato que también aportó el PCA, por supuesto, es la visualización de los datos en una dimensionalidad reducida. Como se puede observar en las representaciones 2D y 3D del apartado de visualización del *notebook*, no se puede apreciar ninguna frontera ni distinción clara entre los distintos valores de G3. Tanto los valores más cercanos a 20 como los próximos a 0 se encuentran esparcidos a lo largo y ancho del gráfico sin estar especialmente agrupados. Este hecho complica las cosas notablemente a la hora de la regresión porque no hay una manera sencilla de separar las buenas y malas notas, pero veremos más adelante qué resultados obtenemos con todos los modelos a ajustar.

4.- Protocolo de remuestreo

Tal y como concluimos al inicio con los estudios previos de los datos, uno de nuestros protocolos de remuestreo es el MSE. Además, basado en lo que hemos ido estudiando durante el curso y lo que más hemos usado, decidimos también usar la validación cruzada.

No obstante, el valor del MSE no es interpretable genéricamente, es decir, un valor de 0 es un modelo perfecto y valor más grandes suelen ser peores, no obstante, ese valor depende del problema, por lo que un valor de 5, puede ser bueno aquí, pero

malo en otro tipo de datos, así que decidimos usar el MSE normalizado. Este valor sí es inmediatamente interpretable y nos sirve para ir sacando conclusiones y resultados con cada ejecución. Al final, dado que el MSE normalizado es 1-R2, decidimos quedarnos simplemente con el R2. Recordamos pues que, un R2 de 1 significa una regresión perfecta, valores de entre 0.85 para arriba son de las mejores predicciones posibles, y menos de 0.7 empiezan a ser bastante mediocres.

La justificación sobre el uso de la validación cruzada es porqué ha sido el método más usado a lo largo del curso, y uno de los que consideramos más representativos de la veracidad del modelo. Recordamos que en esta métrica se devuelve la media de diferentes ejecuciones individuales, así que en el fondo, lo que está haciendo es devolver un valor mucho más general y representativo.

5.- Resultados con métodos lineales

En este apartado mostraremos y analizaremos los resultados obtenidos por los modelos lineales que hemos entrenado en el *notebook* de nuestra práctica. Concretamente las métricas que nos interesan son las que hemos indicado en el apartado anterior, por considerarlas un buen protocolo de remuestreo. De todos los modelos lineales que se habrían podido elegir, los tres seleccionados han sido regresión lineal, regresión LASSO y k-vecinos más cercanos (KNN), y en sus respectivos bloques explicaremos qué nos ha llevado a escogerlos.

Regresión lineal

Se trata del primer método de regresión realizado en esta práctica y también es el más sencillo. De hecho, la razón que nos ha conducido a seleccionarlo como uno de nuestros tres métodos realizados es precisamente su simplicidad. Los modelos que se ajustan más adelante van aumentando su grado de complejidad, especialmente en la sección de métodos no lineales, pero partir de una regresión lineal nos permite ver si realmente existe algún tipo de mejora respecto al modelo más básico que se puede entrenar. Si los subsiguientes métodos no muestran un cambio significativo a mejor, querrá decir que no vale la pena el aumento de complejidad del modelo para unos resultados tan parecidos a este método base.

Entremos en la materia que nos atañe, que son los resultados del método. Nada más comenzar a entrenar el modelo, al ser el primero, hubo que tomar una decisión importante. Como bien se ha comentado en apartados anteriores, las variables predictoras G1 y G2 tienen una muy alta correlación con la variable objetivo e incluirlas en el conjunto predictor puede simplificar demasiado la regresión. Así pues, hicimos la prueba al entrenar este modelo de diferentes maneras. Tanto para el conjunto de entrenamiento como de test, estandarizado y no, creamos tres

versiones que contenían, respectivamente, G1 y G2, solo G1 y, finalmente, ni G1 ni G2.
El que contiene ambas variables alcanza unos resultados sorprendentemente buenos, con una puntuación de validación cruzada de 79,89 % y un R2 de 78,86 %, casi un 80 % en ambas métricas. Que un modelo tan simple como este obtenga tan buenos resultados solo puede indicar dos cosas: que el ejercicio de regresión está siendo demasiado fácil y que el resto de modelos a entrenar tienen muy poco margen de mejora. De esta manera, decidimos que claramente no sería nuestra opción elegida.
La siguiente versión pensamos que podía ser un término medio. Vimos que, en los pesos que el modelo anterior asignaba a cada variable, tanto G1 como G2 recibían el mismo coeficiente, pero aunque numéricamente no hubiera una predilección por ninguno de los dos, racionalmente cabía esperar que la nota del segundo trimestre tuviese más importancia en lo que respecta al tercero, así que esta fue la que eliminamos provisionalmente. Los resultados solo con la variable G1 nos ofrecían una puntuación de validación cruzada de 55,59 % y un R2 de 62,95 %. Estos valores son sensiblemente peores que en el anterior caso, pero al tratarse del modelo más básico posible pensamos que con estar por encima del 50 % era suficiente. Además, por supuesto, esta versión da mucho más margen de mejora a los modelos que vienen a continuación.
La última versión de regresión lineal fue entrenada con el mismo conjunto pero sin G1 ni G2. Sabíamos que la tarea sería mucho más complicada que las anteriores, pero no esperábamos los resultados obtenidos. La puntuación de validación cruzada desciende hasta un valor negativo de -5,08 % mientras que el R2 ofrece un mísero 8,67 %.

Creemos que los resultados obtenidos por la versión que solo contiene G1 no son suficientemente buenos y que un valor algo mejor no habría estado de más, pero en este caso, con las alternativas presentadas, está claro que, para no simplificar tanto las cosas ni obtener resultados ridículamente malos, en el medio está la virtud.

Regresión LASSO

Este segundo modelo lineal fue elegido para establecerlo como la mejora directa de la regresión anterior. Este modelo realiza una selección de variables y eso lo hace más interesante a la hora de observar los pesos que asigna a cada variable. La elección de esta función de `sklearn` concretamente nos supone además una pequeña ayuda porque ajusta sus hiperparámetros automáticamente. Para el entrenamiento y test de este regresor y todos los siguientes ya no hicimos más pruebas y usamos el conjunto de datos solo con G1, como comentamos anteriormente.

El primer aspecto importante que se puede comentar sobre el modelo en cuestión es el que aparece en nuestro *notebook* justo después de entrenarlo, los coeficientes

de cada variable. Como ya hemos dicho, es uno de los aspectos más atractivos que nos ha conducido a entrenar este modelo y, efectivamente, se puede ver que tan solo 13 variables, de las 57 que actualmente tiene este conjunto, obtienen un peso diferente de cero. La razón de que haya tantas variables ahora es que se han añadido muchas más que no estaban en el conjunto original al hacer *one-hot encoding*. En el apartado de Análisis de interpretabilidad explicaremos más en profundidad los valores concretos que han sido asignados a las variables por este modelo y los demás.

En cuanto a la calidad de los resultados predichos podemos decir que ha habido una mejora respecto al modelo base. La puntuación de validación cruzada ha subido más de 6 puntos desde el 55,59 % anterior hasta el 61,98 % actual, de manera que supera la barrera psicológica del 60 % y supone una mayor fiabilidad, aunque todavía no demasiado alta. Por lo que respecta al R2, podemos ver en el gráfico que su valor, aunque menos, también mejora en comparación al previo, en este caso ascendiendo desde un 62,95 % hasta un 65,21 %. En conclusión, se trata del mejor modelo hasta ahora y, aunque no obtiene resultados todo lo satisfactorios que se querría, puede ser indicativo de una tendencia al alza para los siguientes modelos.

K-Nearest Neighbors

Este tercer regresor es el último de los modelos lineales que entrenamos en esta práctica y nos pareció una buena opción por su particular manera de funcionar. Se trata de uno de los modelos más interesantes a nuestro parecer porque aprovecha los valores de los ejemplos más cercanos para predecir uno en concreto, de manera que influye mucho el tipo de cálculo de la distancia entre instancias o la cantidad de ejemplos a tener en cuenta. Lo primero que hacemos en este caso es ajustar los hiperparámetros necesarios, ya que en este caso no funcionan solos como los de LASSO. Concretamente hemos realizado una búsqueda en cuadrícula para:

- n_neighbors: El número k de vecinos a tener en cuenta para predecir cada ejemplo.
- weights: El tipo de peso que cada ejemplo aporta a la regresión.
    - uniform: Hace que cada ejemplo cercano tenga el mismo peso a la hora de predecir independientemente de la distancia a la que se encuentre.
    - distance: Para predecir, cada ejemplo cercano tiene un peso inverso a la distancia con la instancia objetivo.
- leaf_size: El tamaño de hoja para el KDTree.
- metric: El tipo de métrica con la que calcular la distancia entre ejemplos, que puede ser de tipo L1, L2 o coseno.

Entrando en materia sobre los resultados de calidad del modelo podemos ver de entrada que hay al menos 5 combinaciones de parámetros que ofrecen la misma puntuación de validación cruzada. Así pues, estamos nuevamente frente a una

mejora de los resultados puesto que se asciende ligeramente desde el 61,98 % anterior hasta el 66,29 % que aquí se observa. No obstante, la métrica que más gana en este nuevo modelo entrenado es la otra, R2. Se puede apreciar una subida de hasta 8 puntos en este valor puesto que se alcanza una puntuación del 73,20 %. De este modo, no es muy complicado darse cuenta de que este es el mejor modelo entrenado hasta ahora y, por tanto, el mejor de los tres lineales que se ajustan en este informe.

Para mayor claridad, la tabla que se muestra al final del apartado de modelos lineales resume perfectamente los resultados de los tres modelos anteriores y muestra la tendencia al alza que estos han seguido. La incluimos aquí también para sintetizar la evolución, en este caso sobre 100 y no sobre 1.

	R. Lineal	R. LASSO	K-NN
Puntuación R2	62,95 %	65,21 %	73,20 %
Validación cruzada	55,59 %	61,98 %	66,29 %

6.- Resultados con métodos no lineales

Multi-Layer Perceptron

El primer método no lineal que hemos elegido es Multi-Layer Perceptron, o más comúnmente MLP. Recordemos que un MLP es una red neuronal multicapa que permite resolver problemas no lineales. Tiene la forma siguiente:

CAPA INPUT -> CAPAS OCULTAS -> CAPA OUTPUT

Al ser el primer modelo no lineal, esperamos que nos dé una primera idea sobre si realmente valdrá la pena usar métodos polinómicos complejos para predecir los valores de nuestro conjunto.

Para simular el modelo en nuestros experimentos hemos usado la librería SKLearn, en concreto la parte de neural_network para obtener la función MLPRegressor. Además, a diferencia de con los modelos anteriores, hemos usado los datos escalados.

Lo primero que hemos hecho obviamente es la exploración de parámetros. MLP tiene una cantidad abismal de posibles parámetros a la hora de entrenar el modelo, así que para decidir qué parámetros explorar nos basamos en lo visto tanto en clases de Laboratorio durante el curso, como en los ejercicios entregables que hemos ido haciendo. Por tanto, los parámetros elegidos fueron:

- hidden_layer_sizes: Prueba diferentes números de capas ocultas en la red neuronal.
- activation: Cambia la función de activación de las capas ocultas.
    - relu: Elimina valores negativos (max(0,x))
    - identity: Retorna el mismo valor (f(x) = x)
    - logistic: Más conocida como sigmoide (f(x) = 1 / (1 + exp(-x)))
- alpha: Este parámetro regula la penalización L2.
- learning_rate_init: Valor inicial de la tasa de aprendizaje y valor de los pesos de los pasos.
- n_iter_no_change: Máximo número de épocas permitidas.
- learning_rate: Cambios en la tasa de aprendizaje:
    - constant: No cambia en toda la ejecución.
    - invscaling: El valor decrece de manera gradual durante la ejecución.
    - adaptive: El valor cambia cuando hay pérdida de precisión.

Con los datos explorados usando la llamada BayesSearchCV obtenemos los parámetros de las 5 mejores ejecuciones (ordenadas por el valor de validación cruzada). La mejor validación cruzada obtenida fue de 0.5083 y los parámetros que nos dieron dicho valor fueron:

- hidden_layer_sizes: 10
- Función de activación: Identity
- alpha: 0.01
- learning_rate_init: 0.1
- learing_rate: Constant
- n_iter_no_change: 40

Finalmente, los resultados para los protocolos de remuestreo elegidos fueron 0.5083 para la mejor validación cruzada y 0.58 para el error cuadrático. Para ser un modelo polinómico, nos sorprendieron estos resultados tan pobres ya que realmente teníamos esperanza en los modelos no polinómicos, no obstante, nos quedaban aún dos más por probar.

Gradient Boosting

Nuestro último método de regresión aplicado fue Gradient Boosting. Personalmente, teníamos esperanzas en él ya que es uno de los que mejores rendimientos nos había dado, en los laboratorios, en conjuntos de datos que parecían complicados, mejores resultados.

Una vez más hemos usado la versión de sklearn. Y hemos explorado los siguientes parámetros usando una BayesSearchCV:

- n_estimators: Número de escenarios que tratará el algoritmo.
- loss: Función de pérdida a optimizar:

    - squared_error
    - absolute_error
    - Huber: Combina el squared_error y el absolute_error.
    - quantile
- criterion: Función que evalúa la calidad de una partición:
    - friedman_mse
    - squared_error
- max_depth: Nodos máximos en el árbol.
- min_samples_leaf: Número mínimo de ejemplos por hoja del árbol.
- learning_rate: Tasa de aprendizaje.

Hay muchos más parámetros que estos admitidos por la función GradienBoostingRegressor, pero estos son los que habíamos explorado, al menos una vez, durante el curso.

Ejecutada la búsqueda Bayesiana, obtenemos que la mejor validación cruzada es de 0.629 con los parámetros:

- n_estimators: 25
- loss: huber
- criterion: squared_error
- max_depth: 5
- min_samples_leaf: 2
- learning_rate: 0.1

Y, al hacer la gráfica comparativa valores reales - valores predichos y calcular el R2, nos dió un valor de 0.635. Si bien sigue sin ser suficientemente bueno, vistos los resultados hasta el momento tampoco desentona negativamente. Los resultados en general no son de los mejores, así que no justifican el uso de gradient boosting para la predicción.

Random Forest

Ya que nos decidimos por Gradient Boosting, elegimos también Random Forest, ambos modelos están fuertemente relacionados entre sí por el uso de árboles de decisión y creíamos que sería una comparación interesante.

Aquí la exploración era parámetros muy parecidos a gradient boosting, pero prescindiendo de alguno: n_estimators, criterion, max_depth i min_samples_leaf. Como ya están explicados no lo repetiremos.
No obstante, los mejores parámetros fueron algo distintos. Como criterio de calidad ganó *absolute_error,* el número de escenarios fueron 100 y lo otro no es tan relevante. Se pueden ver en el Colab.

Los resultados fueron una Validación Cruzada del 69,9% y un R2 de 75,9%.

Han sido los mejores resultados. Al final, Random Forest es uno de los métodos más potentes, por lo que es cierto que le pertocan los mejores resultados, no obstante, dado que gradient boosting no era para nada bueno, tampoco estábamos tan seguros de esto.

	MLP	Gradient Boosting	Random Forest
Puntuación R2	57,96 %	63,51 %	75,98 %
Validación cruzada	50,83 %	62,93 %	69.91 %

7.- Modelo final elegido

Una vez completadas las dos secciones de modelos de regresión tenemos un ganador entre los lineales y otro para los no lineales. Como se ha comentado anteriormente, el modelo lineal que obtiene mejores resultados tanto en R2 como en la puntuación de validación cruzada es K-vecinos más cercanos (KNN), mientras que el mejor rendimiento de los no lineales ha sido con diferencia el Random Forest. Con los valores comentados en los apartados anteriores ya podemos hacernos una idea de cuál es nuestro modelo elegido, pero hagamos antes una comparativa más para que sea más claro y evidente.

	K-NN	Random Forest
Puntuación R2	73,20 %	75,99 %
Validación cruzada	66,29 %	69,92 %

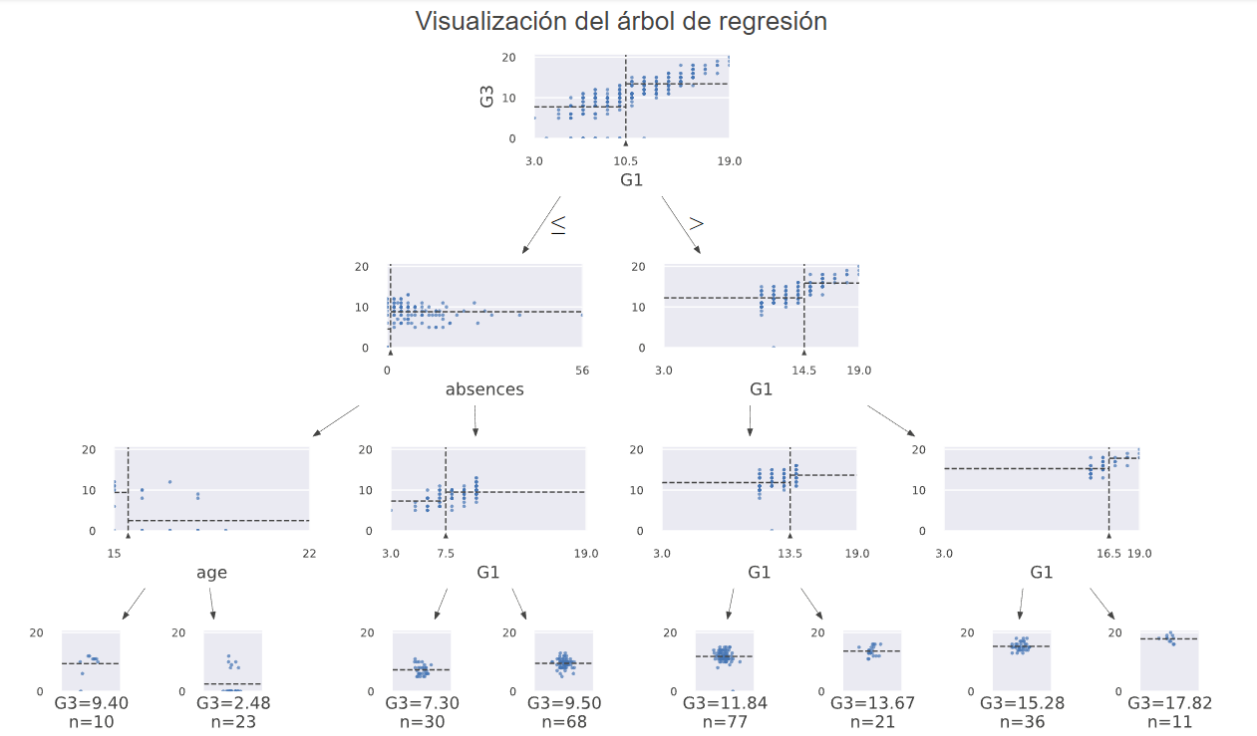
K-Nearest Neighbors obtiene mejores resultados incluso que el resto de modelos no lineales, lo cual lo convierte en un gran candidato a modelo elegido. Aun así, al entrenar el modelo de Random Forest con sus mejores hiperparámetros podemos ver que el rendimiento es incluso mayor que con K-NN.
En definitiva, nuestro modelo ganador y, por tanto, el elegido para la regresión de nuestro conjunto de datos es Random Forest. La justificación de esto se basa meramente en la calidad de las predicciones hechas, de manera que tanto en validación cruzada como en R2 obtenemos unos porcentajes de casi el 70 % o más. Estos valores suponen una mejora considerable respecto a, por ejemplo, el modelo de regresión lineal que tomamos como base y, además, superar la barrera psicológica del 70 % transmite bastante más fiabilidad en cuanto a la validez de las notas predichas.
En concreto estamos hablando de que el error cuadrático medio del modelo elegido es inferior al 25 %, así que se podría decir de una manera generalizada que la nota es acertada 3 de cada 4 veces o que por lo general el valor real no se aleja más de 1 cuarto de la predicción. La razón de que la puntuación de validación cruzada sea

ligeramente más baja que el R2 es que para calcularla se realiza una media de varios valores, los errores obtenidos al considerar como subconjunto de test una porción diferente del conjunto total. En nuestro caso el número elegido de porciones distintas a tener en cuenta ha sido 10, de manera que entre esas 10 ejecuciones algunas han obtenido resultados inferiores al 70 % pero otras, como nuestra partición, han obtenido valores mayores que podrían llegar al 75 % o más.
Desde luego que sabemos que los resultados de las métricas del modelo elegido como ganador no son todo lo buenas que se querría para considerar que un regresor es completamente fiable, pero creemos que unos porcentajes de entre el 70 y 80 % son bastante adecuados y que por supuesto hay que considerar el cierto grado de complejidad que hemos añadido al eliminar la variable G2 del conjunto, que tenía una gran correlación.

8.- Análisis de interpretabilidad

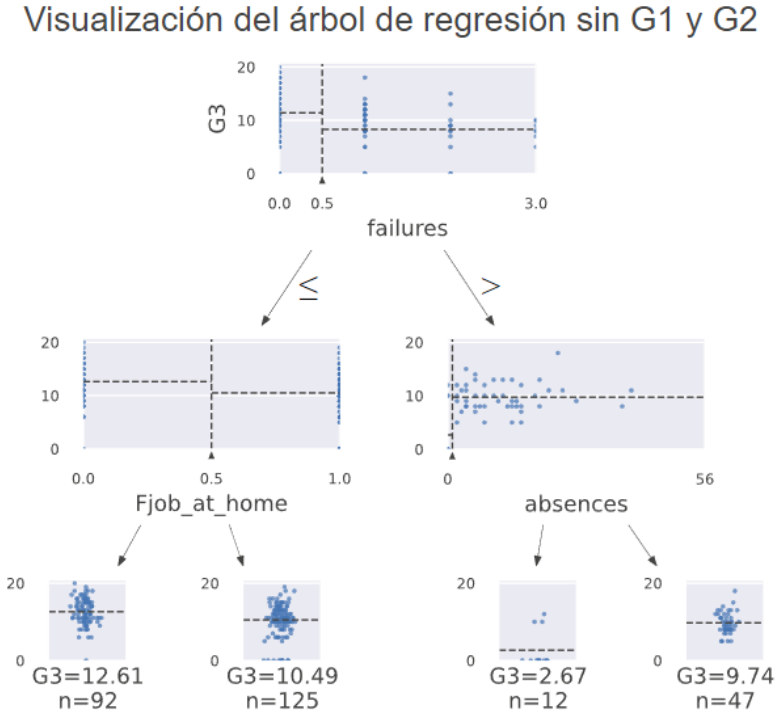
Para interpretar los resultados y analizar los atributos más importantes, quisimos ejecutar un árbol de decisión, sobre todo para la visualización que nos otorga la librería dtreeviz. Para profundizar un poco más, pese a que dijimos que no se tendría en cuenta, hicimos una ejecución con los datos sin G1 y G2. De esta manera podríamos obtener dos tipos de conclusiones: Una sobre la regresión que conseguimos con los métodos y podríamos considerar más teórica, y otra sobre la importancia de diversos factores de una persona sobre su educación, que se podría considerar más social. Pero repito, la segunda conclusión no es la más importante para este trabajo.

Las ejecuciones para los árboles fueron como sigue.
Con G1, el resultado fue el siguiente:



G1 se lleva casi todo el peso, y las particiones que hace no son demasiadas, ya que no tiene en cuenta demasiadas variables.

Sin G1 y G2 obtenemos el siguiente árbol:



Aquí el árbol es aún más pequeño, eso fue también sorprendente ya que pensábamos que, al haber más variables con peso, realizaría más particiones, sin embargo parece ser que esas variables siguen siendo tan insignificantes que las ignora.
Por tanto, podemos concluir que G1 tiene casi todo el peso y, en su ausencia, los suspensos y las ausencias toman más protagonismo.

Para los modelos lineales, nos basamos en los pesos que otorgan los métodos a sus variables para ver qué atributos eran los más valorados.

Regresión Lineal
Este modelo da pesos a todas las variables. Esto puede ser mejor o peor. Viendo nuestros resultados diríamos que en este caso es peor, y es justificable. Al final, realmente hay muchas variables que no aportan ninguna información relevante, pero como la regresión lineal les intenta dar un peso, acaba por ser menos precisa que otras. Sin embargo, en Knn es distinto, porque tener todas las variables el mismo peso (que no es que sea el mismo, sinó que directamente no tienen).

Regresión LASSO
Se puede ver que tan solo 13 variables, de las 57 que actualmente tiene este conjunto, obtienen un peso diferente de cero. Si tenemos en cuenta el orden que siguen las variables dentro del conjunto podemos también analizar qué atributos se están llevando ese peso mayor. No resulta ninguna sorpresa ver que G1 es la variable que más importancia tiene a la hora de predecir el resultado de la variable objetivo y se lleva un coeficiente de 0,985. La segunda y tercera variables en orden descendente de peso no se encuentran demasiado lejos la una de la otra pero sí reciben mucha menos importancia que G1. Se trata curiosamente de “age” y “romantic_no”, es decir, la edad de los estudiantes y el indicador de que no tienen pareja. Reciben un peso respectivamente de -0,360 y de 0,355, o sea que tienen casi la misma magnitud pero con correlación inversa o directa según el signo.

K-NN
Como habíamos adelantado, este método da el mismo peso a todos sus atributos, es decir, para calcular las distancias no distingue según sea una variable teóricamente con más peso o no. Esto puede ser beneficioso, y de hecho es el mejor método lineal, ya que la nota es simplemente un factor más. En el contexto de los estudiantes, creemos que agrupar según atributos iguales o parecidos creemos que es algo bastante positivo. Veamos un ejemplo, imaginemos dos hermanos muy parecidos que por lo general se comportan de manera parecida (amigos en común, extraescolares, situación con la familia, ...), pero a uno de ellos un examen le va muy mal por cierta razón y la nota de un trimestre baja. Un regresor que sea la nota lo que más tiene en cuenta, le costará predecir una futura nota final, sin embargo, con KNN nos da la impresión que, viendo que en verdad ese estudiante es muy parecido a otro con mejor nota, intentará asignarle una mejor nota.

Creemos que será un buen método para consultar ejemplos.

Para los no lineales, usamos la función permutation_importance que nos permitía extraer los atributos importantes.

MLP
MLP asigna una importancia exageradamente superior a G1 respecto al resto de variables. Y no tiene ninguna otra relevancia importante en ninguna otra variable. Asigna algo de importancia en casi cada variable, lo cual habíamos visto que era incluso contraproducente, y viendo que sus resultados son bastante malos también, parece que podemos seguir manteniendo nuestra hipótesis.

Gradient Boosting
Gradient Boosting también da mucho peso a G1, lo cual no nos extraña, pero a diferencia de MLP deja muchas variables sin peso asignado o con pesos ridículos. Las únicas variables a destacar son entonces G1 y las ausencias. Es el primer modelo que asigna una importancia medianamente relevante a alguna variable distinta a G1, así que eso puede ser un paso importante para mejorar las predicciones.

Aquí decidimos hacer un experimento extra sólamente para poder sacar alguna conclusión extra y acompañar a los resultados del Árbol de Regresión. Probamos Gradient Boosting con el conjunto de datos sin G1 y G2, es decir, el regresor no cuenta con ninguna nota del alumno, para calcular su nota final. Tal y como vimos con el primer experimento, la validación cruzada y R2 eran absurdamente bajos, y reafirmó que una predicción fiable quedaba totalmente descartada. No obstante lo interesante fueron las variables a las que se le daba más peso para intentar predecir la nota. Estas fueron: las asignaturas suspendidas previamente y las ausencias. Este resultado nos sorprendió gratamente ya que eran las mismas variables que, en el estudio comentado al principio, tenían más peso en ausencia de G1 y G2. Lo que nos indicaba que estábamos haciendo las cosas bien. Además también coinciden perfectamente con las del árbol de decisión, tanto con G1 como sin él.

Random Forest
Para random forest esperábamos unos pesos bastante parecidos a Gradient Boosting ya que al final ambos modelos se ayudan de los árboles de decisión, sin embargo, habiendo visto que sus resultados eran mejores, teníamos esperanza en ver algún cambio o alguna asignación algo distinta.

Al final los resultados fueron los mismos, mucho peso a G1, algo a las ausencias del alumno y muchas variables sin ningún peso o valores ínfimos.
Referente a la segunda parte de la interpretación, hemos elegido los siguientes ejemplos. Mostraremos los atributos que consideramos más importantes para la predicción de cada estudiante según lo que hemos observado en el análisis de los

atributos y, si es posible, mostraremos algún atributo que pueda tener sentido para cada caso en particular.

Estudiante 389

Failures	Absences	G1	G2	G3
1	0	6	5	0

Este caso puede ser curioso para los regresores ya que ni sus ausencias son muy elevadas, sus suspensos tampoco, no tiene notas muy elevadas en ningún trimestre, pero no son 0, y sin embargo su nota final es de 0, la cual es muy baja y probablemente impredecible. Además no hay un parámetro lógico que el regresor no tenga muy en cuenta y sin embargo pueda ser socialmente relevante como la situación familiar o los problemas de salud.

Estudiante 4

Failures	Absences	G1	G2	G3
3	10	7	8	10

Este caso también llama bastante la atención, pese a no aprobar ni el primer ni el segundo trimestre, aprueba el tercero y, si se tuviese que guiar uno por sus ausencias o suspensos, nadie diría que sería capaz de aprobar el tercero. No obstante, hay una variable que se llama “paid”que hace relación a si tiene o no clases extra, y en su caso es que sí, por tanto, pese a que el regresor no lo tiene demasiado en cuenta, podría llegar a entenderse su aprobado final.

Estudiante 137

Failures	Absences	G1	G2	G3
0	0	11	0	0

Finalmente, este estudiante parece que no empieza tan mal el curso, pero sin embargo lo acaba de forma catastrófica. Contando que además nuestros modelos no cuentan con la nota de G2, realmente parece imposible su predicción acertada, más aún viendo sus ausencias y suspensos. En este caso tampoco hay ninguna variable social que pueda justificar sus notas.

Para ver las predicciones elegiremos los regresores KNN y Random forest ya que eran los mejores en sus respectivas categorías (lineales o no).

Las predicciones fueron, no sorprendentemente, erróneas, siendo estas para KNN:
Estudiante 389 -> 2.857
Estudiante 4 -> 8.428
Estudiante 137 ->10.428

Y de hecho, pudimos calcular las distancias a sus respectivos vecinos más cercanos y observamos ciertas cosas también interesantes. Tanto el 389 como el 4 tienen un ejemplo a distancia 0 y poco, por lo que són muy parecidos entre sí y su nota estará muy relacionada. Nos pareció extraño, pero tenía sentido ya que era nuestro propio que estaba dentro del conjunto de entrenamiento, así que nos fijamos en los otros ejemplos que se encontraban entre 4 y 5 de distancia. Por lo general, los ejemplos tienen notas parecidas, sin embargo, por ejemplo con el estudiante 389, había uno de los más cercanos con nota final de 12, otro con 10, así que a pesar de que una gran mayoría tenían 0, ciertos valores más grandes decanta la predicción final por un poco más de 0.

Esta explicación se puede extrapolar al resto de ejemplos, ya que KNN hace lo mismo.

Para el Random Forest:
Estudiante 389 -> 1.505
Estudiante 4 -> 9.255
Estudiante 137 ->11.155

Está claro que el estudiante 137 es un caso imposible de predecir sin contar con la nota G2 (como mínimo), no obstante, en los otros se acercan bastante más de lo que creíamos que lo harían.

Para profundizar también un poco más en algún ejemplo, sacamos el decision_path del Random forest con mejores parámetros y visualizamos alguno de los árboles formados por el Random Forest para ver qué decisiones podría haber seguido nuestro estudiante. Una de las cosas que más nos sorprendió al visualizar los árboles fue ver que realmente Random Forest usa todos los atributos. Los de mayor importancia aparecían en todos los árboles (o en casi todos), pero todos los atributos tenían al menos un nodo de decisión. Siempre empieza teniendo en cuenta G1, y en los pasos inferiores es cuando prueba otros atributos.

Por ejemplo, vimos que del primer árbol usaba el primer nodo y algunos más del medio, no obstante este primero era fácil de visualizar. Vimos que la primera clasificación que hacía, era respecto a G1, en este caso si dicha nota era mayor o no a 13,5. En este caso, todos nuestros estudiantes se irían en el caso de no, es decir, de la izquierda.

Si siguiéramos visualizando árboles y que nodos se visualizan de cada árbol, podríamos reconstruir el path por completo y ver qué atributos se valoraban de nuestros estudiantes en concreto. Al haber 100 árboles distintos y más de 10000 nodos, decidimos no indagar tantísimo ya que era una pérdida de tiempo. Lo que sí nos quedamos es con la manera de hacerlo. En el Colab hay las instrucciones necesarias para visualizar lo comentado y, si se usan de la forma adecuada se podría llegar a ver bien el *path* de decisión.

Como conclusión de este apartado nos quedamos con que los casos complicados son difíciles de predecir ya que de G1 a G3 puede cambiar mucho la nota. KNN lo podría llegar a predecir ya que al si bastantes personas tuviesen casos parecidos, los clasificaría igual, pero Random Forest basa su primera decisión en G1, por lo que sí después no es capaz de reconducir las decisiones hasta un G3 adecuado, no llegará a la predicción correcta ya que G1 lo habrá alejado mucho al principio. También nos quedamos con que pese a no ser correctas, en bastantes ocasiones la predicción se acercaba más de lo que esperábamos.

Y finalmente, con la larga exploración que hace Random Forest y sus árboles que son sorprendentemente distintos al árbol de decisión. No para mal, sinó todo lo contrario ya que al realizar más distinciones, usa más atributos y parece que consigue mejores resultados. Usar más atributos aquí, es algo distinto a que tengan más importancia. Atributos con más importancia són los de permutation_importance, y son bastantes pocos.

9.- Autoevaluación y conclusiones

Para finalizar el informe de esta práctica de regresión es necesario realizar una pequeña autoevaluación de los resultados obtenidos y nuestro desarrollo del *notebook* y la documentación. Creemos que en general nuestro ritmo de trabajo ha sido el adecuado porque, sabiendo la carga de trabajo que podía implicar, nos pusimos manos a la obra con antelación. Dedicamos un primer esfuerzo inicial a elegir las métricas y los modelos que hemos incluido y a entrenar estos debidamente y en una segunda fase de la práctica nos concentramos en analizar todos los resultados obtenidos y a sacar conclusiones de ellos.

En general opinamos que los resultados finales son bastante reveladores en cuanto al funcionamiento de los modelos seleccionados y a su adecuación a nuestro problema en cuestión. La variedad de modelos entrenados nos ha permitido ver, entre otras cosas, qué variables era mejor eliminar o conservar en el conjunto de datos, ya que tomamos los resultados de la regresión lineal inicial como ejemplo para incluir G1 o G2.
También hemos sido capaces de entender mejor cómo funcionan todos los modelos entrenados, con sus particularidades que les hacen tener mejores o peores

puntuaciones, ya que en este caso hay modelos que asignan pesos a todos los atributos del conjunto (regresión lineal), que los asignan tan solo a algunas variables (regresión LASSO), que dan peso a los propios ejemplos y no a las variables (K-NN) o que utilizan estructuras arbóreas para decidir (Random Forest), por mencionar algunos.
Finalmente también hemos podido comprobar, como ya venimos viendo durante los problemas entregables de la asignatura, que para un mismo problema se pueden obtener predicciones altamente cercanas a los valores reales o ridículamente malas simplemente en función del modelo elegido, desde puntuaciones de poco más del 50 % de acierto hasta más del 75 % que obtiene el Random Forest elegido.

Por supuesto, el estudio de la práctica podría haber sido mucho más amplio si se hubieran tenido en cuenta muchos más modelos para la comparación, siendo algunos ejemplos la regresión Ridge o las SVM con kernels lineal, cuadrático o RBF. Creemos que la puntuación de validación cruzada y la de R2 son métricas suficientemente variadas e importantes, pero también es cierto que se podría haber extendido el trabajo si se hubieran empleado otro tipo de métricas que quizás en algunos contextos son más explicativas del rendimiento de un modelo y que no hemos podido ver.

La conclusión final, pues, es que el análisis comparativo ha sido bastante exhaustivo, que los modelos seleccionados han aportado la variedad necesaria para sacar conclusiones provechosas, que las métricas han sido perfectamente útiles para ver sin ninguna ambigüedad qué modelos eran mejores y, sobre todo, que ha sido realmente productivo y didáctico poder realizar un estudio sobre un conjunto de datos elegido por nosotros de una manera tan rigurosa y completa, desde la selección de variables y el preprocesamiento hasta la elección final de métodos ganadores.

10.- Referencias bibliográficas

Conjunto de datos
Adjuntamos el link oficial de donde sacamos el conjunto de datos, y todos los detalles necesarios sobre este.
https://archive.ics.uci.edu/ml/datasets/student+performance

Web de la asignatura
De la web de la asignatura sacamos de referencia todos los laboratorios, los archivos .ipynb extras para cada modelo y las transparencias de teoría de donde sacar información.
https://sites.google.com/upc.edu/aprenentatge-automatic?pli=1

Web oficial SKLearn
Todos los modelos han sido implementados con las versiones de SKLearn, así que todas sus descripciones de las implementaciones han sido visitadas regularmente para resolver todo tipo de dudas.
https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

https://scikit-optimize.github.io/stable/auto_examples/index.html

https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LassoCV.html

https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html

https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPRegressor.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

https://scikit-learn.org/stable/modules/permutation_importance.html

https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html

Web oficial de statsmodel
Para obtener los pesos correctamente de la regresión lineal, implementamos la versión de statsmodel ya que nos ofrecía la función summary muy útil.
https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html

https://www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLSResults.html

StackOverflow
Las dudas más concretas que teníamos las buscábamos en stackOverflow, que es la página de la que mejores respuestas obtenemos.
https://stackoverflow.com/questions/51734180/converting-statsmodels-summary-object-to-pandas-dataframe

https://stackoverflow.com/questions/48870380/randomforestclassifier-object-has-no-attribute-tree

https://stackoverflow.com/questions/67051417/how-to-predict-a-single-sample-with-keras

Otros
Muchas veces StackOverflow no era suficiente y teníamos que profundizar más en algunos blogs de internet.
https://towardsdatascience.com/beautiful-decision-tree-visualizations-with-dtreeviz-af1a66c1c180

https://garg-mohit851.medium.com/random-forest-visualization-3f76cdf6456f

https://towardsdatascience.com/what-is-one-hot-encoding-and-how-to-use-pandas-get-dummies-function-922eb9bd4970

Estudios previos del conjunto de datos

Paulo Cortez and Alice Silva, *USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE*, Dep. Information Systems/Algoritmi R&D Centre University of Minho, 4800-058 Guimarães, PORTUGAL

Se puede acceder al estudio en concreto aquí
http://www3.dsi.uminho.pt/pcortez/student.pdf

Esta página nos sirvió para comprobar los modelos mencionados en el estudio referenciado.
https://rdrr.io/cran/rminer/