Caso de estudio

PID_00284576

Jordi Gironés Roig

Tiempo mínimo de dedicación recomendado: 2 horas

Universitat Oberta de Catalunya

Jordi Gironés Roig

Licenciado en Matemáticas por la Universidad Autónoma de Barcelona y diplomado en Empresariales por la Universitat Oberta de Catalunya. Ha desarrollado la mayor parte de su carrera profesional en torno de la solución SAP, en sus vertientes operativas con S4-HANA y estratégica con SAP-BI. Actualmente trabaja en la industria químico-farmacéutica como responsable de aplicaciones corporativas para Esteve Pharmaceuticals y colabora con la UOC en asignaturas relacionadas con la analítica de datos.

El encargo y la creación de este recurso de aprendizaje UOC han sido coordinados por el profesor: Julià Minguillón Alfonso

Primera edición: septiembre 2021

Av. Tibidabo, 39-43, 08035 Barcelona
Autoría: Jordi Gironés Roig
Producción: FUOC

Índice

Introducción 5

1. Presentación del caso 7

2. Paso 1: establecer un objetivo analítico 8

3. Paso 2: verificar los datos 10

4. Paso 3: preprocesado de datos 13
4.1. Errores de escritura 13
4.2. Errores en valores 13
4.3. Gestión de valores nulos 15

5. Paso 4: análisis exploratorio 19

6. Paso 5: clasificación 22
6.1. Separación del juego de datos 22
6.2. Generación del modelo 23
6.3. Gestión del *overfitting* 23
6.3.1. *Cross-validation* 24
6.4. *Parameter tunning* 24

7. Paso 6: reproducibilidad 27

Resumen 28

Introducción

Utilizaremos el formato de caso de estudio para presentar un flujo analítico contextualizado, puesto que entendemos que la analítica de datos debe enfocarse de un modo holístico donde el dato y lo que este describe forman un núcleo inseparable.

Adicionalmente, lo haremos siguiendo un decálogo de buenas prácticas para que el estudiante pueda tener una buena base desde la que crecer como analista profesional para llegar a ser más efectivo y sobre todo más colaborativo.

1. Presentación del caso

Aprovechando nuestro conocimiento en los ámbitos de la minería de datos y los análisis clínicos, hemos fundado una *start-up*, New Diagnosis LLC, dedicada a **facilitar herramientas de diagnóstico** a hospitales.

Conocedores del potencial que supone poder trabajar con fuentes de datos externas generadas por terceros, nuestra propuesta es firmar contratos de colaboración con diferentes hospitales para que nos cedan datos sobre analíticas de pacientes que usaremos para entrenar nuestros modelos predictivos con el objetivo de facilitar y **acelerar el proceso de diagnóstico** en futuros casos.

Figura 1. New Diagnosis LLC

Nuestro objetivo es dar un paso más allá y ser capaces de identificar variantes o grados de intensidad de la propia enfermedad, posibilitando así la implementación de tratamientos menos generalistas y más orientados a cada tipología concreta. Incrementar la precisión en el diagnóstico y el tratamiento redundará también en la reducción de efectos secundarios, que en muchas ocasiones pueden acabar suponiendo un grave problema para el paciente.

2. Paso 1: establecer un objetivo analítico

El primer paso que deberíamos dar como analistas de datos es el de establecer nuestro objetivo analítico. Dicho de otro modo, redactar la pregunta o plantear el problema que trataremos de resolver. Recordemos la siguiente máxima:

> Una buena respuesta requiere una buena pregunta.

Del mismo modo, también deberemos fijar qué medidas vamos a usar para medir el grado de cumplimiento del objetivo planteado. La siguiente pregunta nos puede ayudar en esta primera tarea.

> ¿Has especificado la tipología de pregunta analítica a la que te enfrentas (por ejemplo, clasificación, segmentación, asociación, regresión...) antes de empezar a trabajar con los datos?

Vamos a clasificar variantes de un tipo de enfermedad a partir de analíticas tomadas a pacientes que previamente sabíamos que sufrían dicha variante. De modo que se trata de un problema de clasificación.

> ¿Has establecido cómo vas a medir el grado de cumplimiento de tu objetivo analítico?

Dado que nos enfrentamos a un problema de clasificación, hemos pensado en usar como medida el concepto de precisión: **proporción de variantes clasificadas correctamente**.

Para cuantificar el rendimiento de nuestro modelo de clasificación, el responsable de datos de New Diagnosis LLC ha establecido que quiere llegar al 90 % de precisión.

> ¿Has entendido bien el contexto del objetivo planteado y sus posibles aplicaciones?

Estamos construyendo una parte de la funcionalidad de nuestro motor de clasificación, responsable de las habilidades de clasificación de variantes de enfermedades etiquetadas y a partir de análisis clínicos asociados. En el futuro, este motor se integrará con cada vez más hospitales para incrementar sus capacidades de aprendizaje.

¿Has establecido un procedimiento para recopilar datos para la demo?

El responsable de datos de New Diagnosis LLC nos ha explicado que los científicos de campo aprovecharán los primeros contratos de intercambio de información firmados con dos hospitales que van a ceder los datos debidamente anonimizados y relacionados con tres variantes de una enfermedad identificada a partir de marcadores presentes en los análisis clínicos habituales.

Inicialmente disponemos de datos de ciento ochenta pacientes sobre los que se han realizado estudios tanto de diagnóstico como de seguimiento de la enfermedad con el objetivo de completar mejor el proceso de captura de información.

Finalmente, los datos recogidos se recopilan en un juego de datos que se almacena en un repositorio GitHub de una empresa privada.

¿Te has planteado si realmente tu objetivo analítico puede ser respondido a partir de los datos disponibles?

El juego de datos del que disponemos para esta demo contiene solo datos de tres variantes de una enfermedad. En consecuencia, el modelo de clasificación generado solo servirá para clasificar una de las tres variantes sobre las que se ha capturado información en el juego de datos. Si quisiéramos construir un clasificador más general deberemos recopilar más datos.

3. Paso 2: verificar los datos

El siguiente paso es echar un vistazo al juego de datos, porque este siempre puede contener errores y es importante que los gestionemos antes de iniciar nuestro estudio analítico.

En general, vamos a tratar de responder las siguientes preguntas:

- ¿El juego de datos contiene errores?
- ¿Hay cosas extrañas entre los datos?
- ¿Voy a necesitar corregir o incluso eliminar parte de los datos?

Empecemos por leer el juego de datos: observamos que nuestro juego de datos contiene 180 filas o diagnósticos de pacientes y 18 columnas de las cuales 17 corresponden a marcadores sobre los análisis clínicos que se les ha practicado.

Visualicemos en la figura 2, por ejemplo, las diez primeras columnas, donde apreciamos que la columna *class* contiene el diagnóstico y el resto de columnas son los distintos marcadores clínicos.

Figura 2. Visualización inicial

	class	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
0	variantA	6.4	4.9	2.9	1.8	15.93	3.51	4.33	17.7	129.2
1	variantA	6.2	4.4	2.9	1.8	14.90	3.58	4.04	13.3	102.2
2	variantA	6.0	4.6	2.8	1.8	14.86	4.16	4.57	20.7	103.2
3	variantA	5.9	4.5	3.0	1.8	16.07	3.75	4.40	18.9	115.2
4	variantA	6.3	5.0	2.9	1.8	14.94	4.39	4.77	23.1	120.2

Empezamos bien, parece que el juego de datos tiene buen aspecto.

La primera fila del juego de datos contiene los nombres de las características y estos son suficientemente descriptivos, de modo que es fácil reconocer qué representa cada columna.

Cada fila del juego de datos representa la analítica de un paciente: 17 medidas y 1 clase, que nos indicará a qué variante de enfermedad corresponde.

Una de las primeras cosas que deberíamos gestionar son los valores nulos.

Por suerte, nuestros clínicos han tenido la precaución de identificar con NA aquellas medidas que no han podido tomar.

Como siguiente paso, siempre es recomendable visualizar la distribución de nuestros datos, prestando una atención especial a los valores extremos o *outliers*.

Empecemos por listar en la figura 3 los estadísticos básicos de nuestro juego de datos.

Figura 3. Estadísticos básicos

	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
count	150.000000	150.000000	150.000000	145.000000	178.000000	178.000000	178.000000	178.000000	178.000000
mean	6.901727	4.454667	5.258667	2.836552	14.700618	4.136348	4.266517	21.594944	101.941573
std	1.503198	0.433123	1.764420	0.755058	0.811827	1.117146	0.274344	3.339564	14.282484
min	0.068000	3.400000	2.500000	1.700000	12.730000	2.540000	3.260000	12.700000	72.200000
25%	6.400000	4.200000	3.100000	2.000000	14.062500	3.402500	4.110000	19.300000	90.200000
50%	7.000000	4.400000	5.850000	2.900000	14.750000	3.665000	4.260000	21.600000	100.200000
75%	7.700000	4.700000	6.600000	3.400000	15.377500	4.882500	4.457500	23.600000	109.200000
max	9.200000	5.800000	8.400000	4.100000	16.530000	7.600000	5.130000	32.100000	164.200000

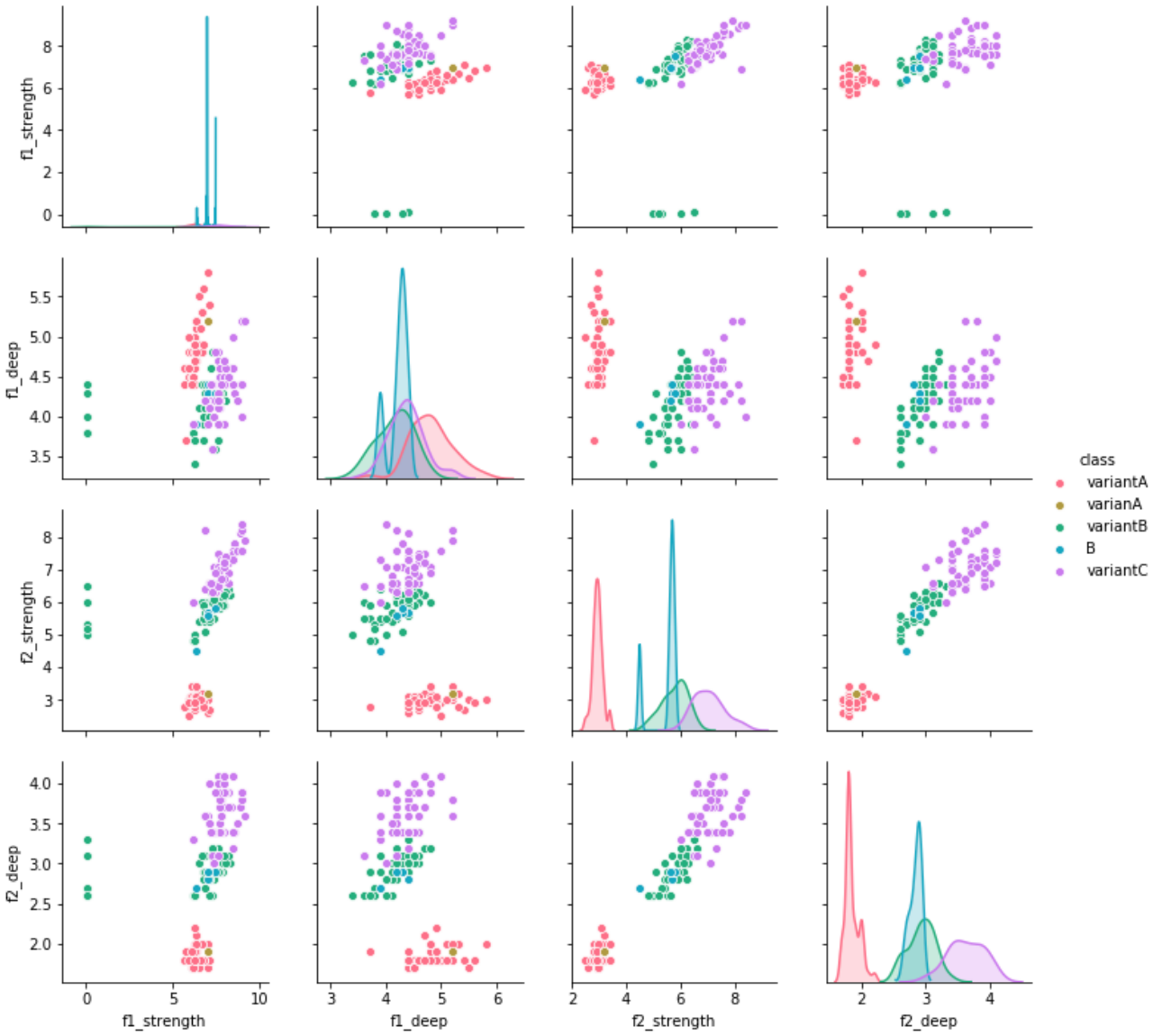
De esta tabla se desprende, por ejemplo, que en todas las características tenemos valores ausentes, ya que en ninguna de ellas alcanzamos el valor 180 en la variable *count*.

El problema de este tipo de estadísticas es que solo nos serán útiles si ya de antemano sabemos que los valores obtenidos deberían estar dentro de cierto rango. Sin embargo, aun así merece la pena visualizarlas, ya que pueden ayudar a identificar irregularidades de una forma rápida y que de otro modo no apreciaríamos.

En la figura 4 vamos a crear una matriz de gráficos de puntos *scatterplot matrix* solo a partir de las cuatro primeras características. Las matrices de gráficos de puntos muestran un histograma de cada característica en la diagonal, y los gráficos de puntos para cada par de características del juego de datos. Nos proporciona de un modo esquemático una visualización muy general de nuestros datos.

Adicionalmente podemos asignar un color distinto a cada clase para facilitar la posibilidad de identificar patrones.

Figura 4. Matriz de gráficos



A partir de la matriz de gráficos analicemos más problemas en el juego de datos:

- Hay cinco clases, cuando solo debería haber tres: probablemente debido a algunos errores en los nombres de las clases.

- Hay algunos valores extremos que aparentemente deben ser errores: una fila de *f1_deep* para *variantA* está claramente fuera del rango habitual y varias filas de *f1_strength* para *variantB* tienen valores cercanos a cero. Deberemos buscar una razón.

- Hemos eliminado las filas con valores nulos para poder generar los gráficos.

En cada uno de estos tres casos deberemos tomar una decisión, lo que nos lleva al siguiente paso.

4. Paso 3: preprocesado de datos

Hasta ahora hemos identificado varios problemas con respecto a los datos, de modo que deberemos resolverlos antes de seguir con el análisis.

4.1. Errores de escritura

Tenemos cinco clases, mientras que solo deberían haber tres. Tenemos errores de escritura en el juego de datos.

Después de interactuar con el equipo de científicos, hemos visto que uno de ellos olvidó añadir la palabra *variant* en algunas de las filas correspondientes a *variantB*.

La otra clase sobrante, *varianA*, simplemente era un error tipográfico que olvidaron corregir.

4.2. Errores en valores

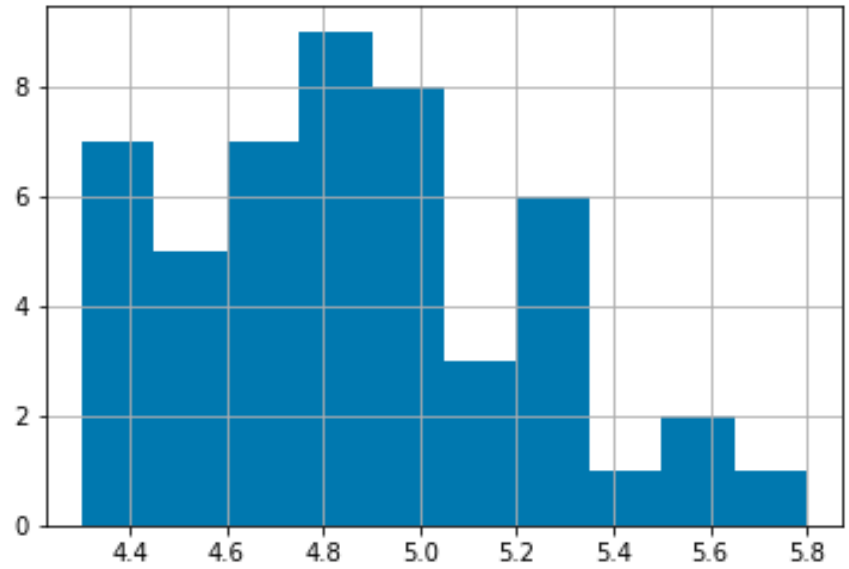
Hay algunos valores extremos que aparentemente son errores: una fila de *f1_deep* perteneciente a *variantA* está claramente fuera del rango normal, y varias filas de *f1_strength* pertenecientes a *variantB* tienen valores anormalmente bajos por algún motivo.

La gestión de valores extremos no es en absoluto una materia obvia en el mundo de la analítica de datos. Normalmente no es fácil discernir si se trata de un error numérico o si realmente responde a una anomalía en la que deberíamos profundizar. Por este motivo, deberemos ser muy cuidadosos cuando trabajemos con este tipo de valores. Si tomamos la decisión de excluirlos, deberemos documentar concretamente qué datos hemos excluido y por supuesto argumentarlo con razones sólidas.

En el caso concreto de las filas anómalas de *variantA*, nuestros compañeros científicos nos han explicado que es imposible para la enfermedad estudiada del tipo *variantA* tener una medida *f1_deep* por debajo de 3.9. Claramente este valor debe ser revisado.

¡Muy bien! En la figura 5 observamos como todos los diagnósticos *variantA* están por encima de 3.9.

Figura 5. Diagnósticos de *variantA*



Nuestro siguiente objetivo será gestionar los valores cercanos a cero en la medida de *f1_strength* para el caso *variantB*. Empecemos por visualizar estas filas en la figura 6.

Figura 6. Valores cercanos a cero

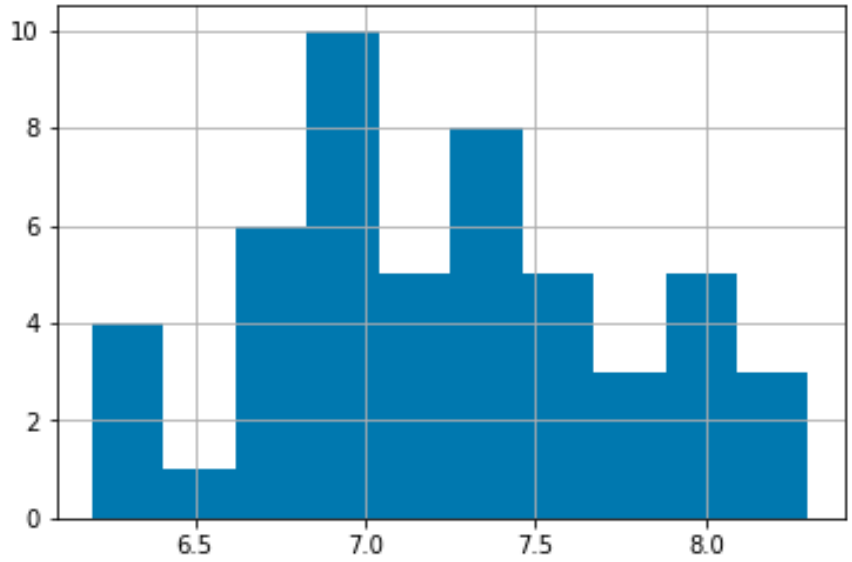
	class	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
86	variantB	0.080	4.4	6.5	3.3	13.86	3.41	4.21	24.9	92.2
87	variantB	0.073	4.3	6.0	3.1	13.35	3.47	4.52	28.1	90.2
88	variantB	0.070	4.0	5.0	2.6	13.34	3.86	4.36	23.7	86.2
89	variantB	0.068	3.8	5.3	2.7	13.78	3.13	4.20	25.7	72.2
90	variantB	0.068	3.8	5.2	2.6	13.78	3.63	4.22	20.6	83.2

¿Que puede haber sucedido? Todos estos valores de *f1_strength*, cercanos a cero, podría ser que se hubieran tomado en milímetros en lugar de centenas de milímetros.

Efectivamente, el equipo de científicos nos confirma nuestras sospechas, de modo que por nuestra parte procedemos a realizar la conversión de unidad pertinente.

¡Perfecto! En la figura 7 apreciamos que ya no tenemos valores *outliers* que podían haber tenido un impacto muy negativo en nuestro estudio.

Figura 7. Valores corregidos para *variantB*



4.3. Gestión de valores nulos

Hemos tenido que prescindir de las filas con valores nulos para poder generar los gráficos.

Empecemos por visualizar en la figura 8 algunas de las filas con valores nulos.

Figura 8. Valores nulos

	class	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
7	variantA	6.3	4.8	3.0	NaN	15.76	3.95	4.51	19.7	123.2
8	variantA	5.7	4.3	2.9	NaN	16.53	3.44	4.07	16.1	99.2
9	variantA	6.2	4.5	3.0	NaN	15.56	3.15	4.17	18.1	100.2
10	variantA	6.7	5.1	3.0	NaN	15.80	3.96	4.20	20.1	107.2
11	variantA	6.1	4.8	3.1	NaN	15.82	3.28	4.22	18.9	97.2
109	variantB	NaN	NaN	NaN	NaN	13.31	3.15	4.60	22.1	96.2
110	variantB	NaN	NaN	NaN	NaN	13.16	5.54	3.72	21.6	109.2

Observamos como para los casos de *variantB* las primeras cuatro características tienen todas valores nulos, concretamente 21 filas, de modo que optamos por eliminar estas entradas, ya que nuestro estudio se centra precisamente en estas cuatro columnas y se trata de demasiadas filas. Con tantas filas, rellenar con medias generaría un sesgo hacia la media que no creemos bueno para generalizar.

En la figura 9 podemos ver el efecto de este primer paso de eliminación de filas.

Figura 9. Tras eliminar filas

	class	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
7	variantA	6.3	4.8	3.0	NaN	15.76	3.95	4.51	19.7	123.2
8	variantA	5.7	4.3	2.9	NaN	16.53	3.44	4.07	16.1	99.2
9	variantA	6.2	4.5	3.0	NaN	15.56	3.15	4.17	18.1	100.2
10	variantA	6.7	5.1	3.0	NaN	15.80	3.96	4.20	20.1	107.2
11	variantA	6.1	4.8	3.1	NaN	15.82	3.28	4.22	18.9	97.2

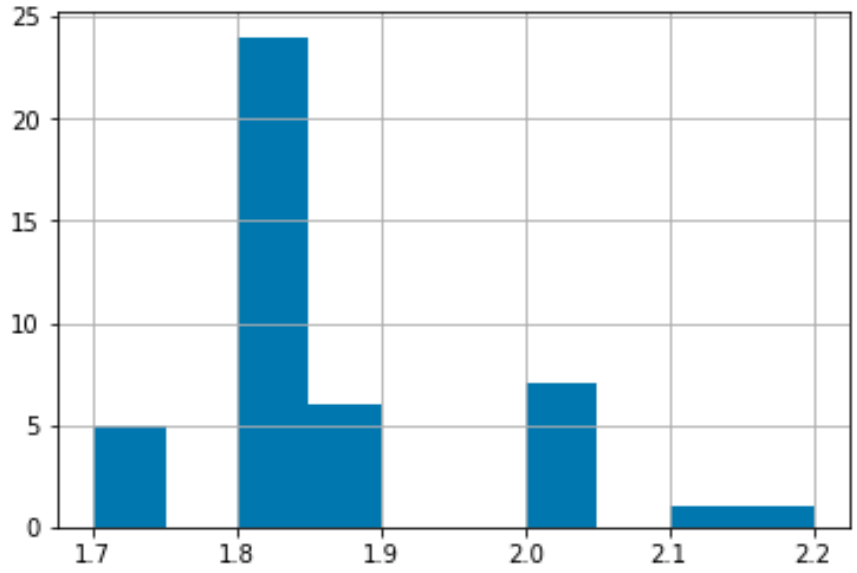
Nos queda todavía un resto de valores nulos en la característica *f2_deep* de los casos *variantA*. Eliminando todas estas filas estamos pagando un precio alto, puesto que todas ellas pertenecen a la clase *variantA* y solo tenemos el problema en una característica.

El problema que nos encontramos es que nuestro juego de datos queda sesgado, ya que todos los problemas de valores nulos se concentran bajo la misma tipología de enfermedad. En consecuencia, tendremos menos información sobre esta clase y este hecho nos podría llevar a construir un modelo incorrecto.

Bajo estas circunstancias, una buena opción sería rellenar los valores nulos con la media de esta característica dentro de su clase: si sabemos que los valores de esta medida deberían estar dentro de un cierto rango, entonces tiene sentido rellenarlos con valores como la media.

Para hacerlo nos apoyaremos en la figura 10, en la que visualizamos los rangos de valores para la característica *f2_deep*.

Figura 10. Rango de valores en *variantA*



La mayoría de valores *f2_deep* para *variantA* se mantienen en el rango 1.8-1.9, de modo que optaremos por informar de los valores nulos con el valor de la media.

En la figura 11 vemos el efecto de esta sustitución de valores.

Figura 11. Rellenado con medias

	class	f1_strength	f1_deep	f2_strength	f2_deep	concavity	smoothness	compactness	texture	area
7	variantA	6.3	4.8	3.0	1.85	15.76	3.95	4.51	19.7	123.2
8	variantA	5.7	4.3	2.9	1.85	16.53	3.44	4.07	16.1	99.2
9	variantA	6.2	4.5	3.0	1.85	15.56	3.15	4.17	18.1	100.2
10	variantA	6.7	5.1	3.0	1.85	15.80	3.96	4.20	20.1	107.2
11	variantA	6.1	4.8	3.1	1.85	15.82	3.28	4.22	18.9	97.2

¡Perfecto! Ya no tenemos valores nulos en nuestro juego de datos.

Nota

El juego de datos contiene más valores nulos en otras características, pero para simplificar simplemente nos estamos centrando en las cuatro primeras.

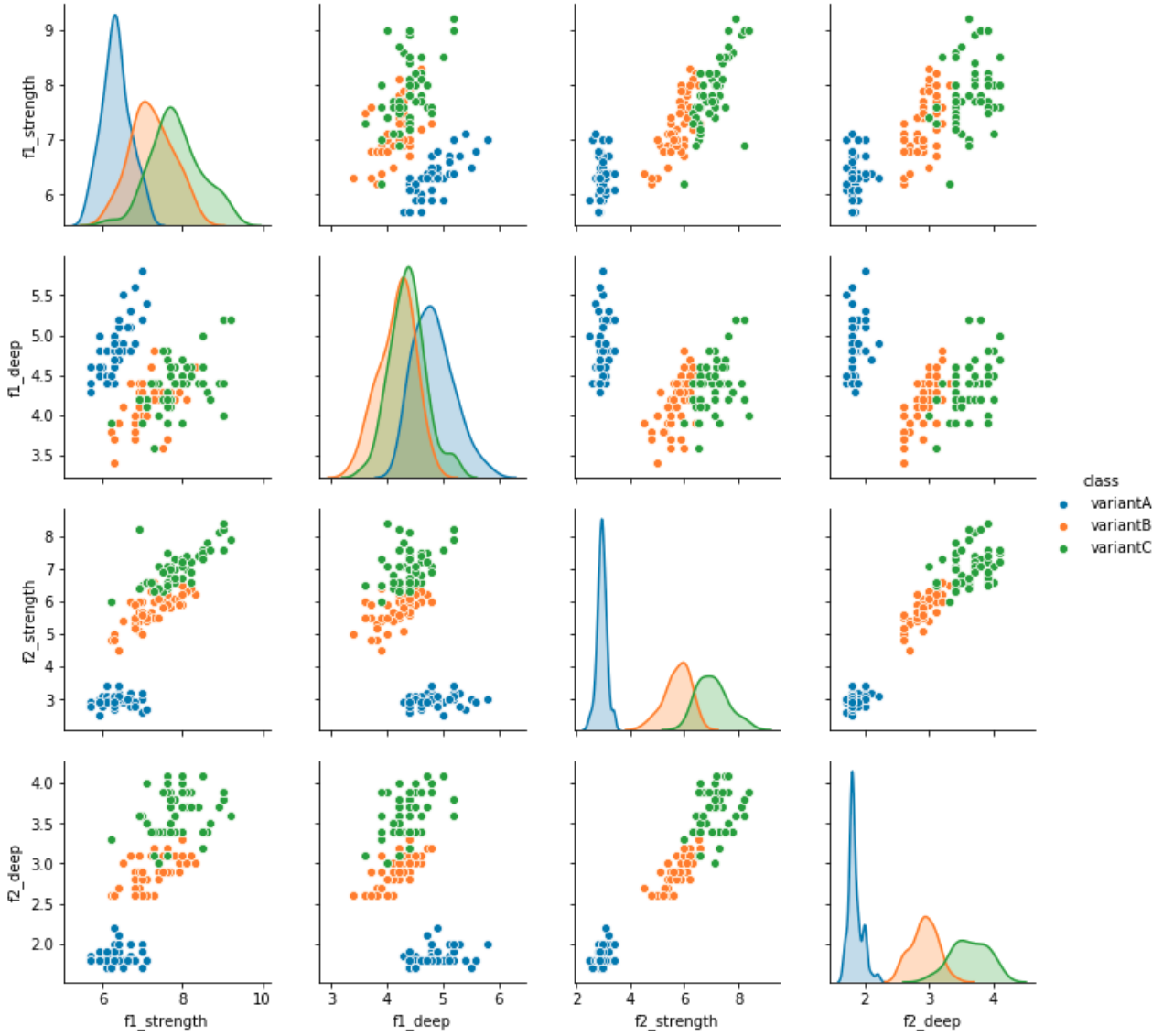
Tras el preprocesado de datos, merece la pena generar una matriz de gráficos para echar un vistazo general a los datos y certificar que las correcciones que hemos aplicado hasta el momento funcionan. Veámoslo en la figura 12.

Las pequeñas modificaciones que hemos hecho sobre el juego de datos original nos han servido para mostrar algunos de los aspectos que hay que tener en cuenta y valorar las opciones que tenemos para gestionarlos.

Las buenas prácticas que se deben tener en cuenta deberían ser:

- Asegurarse de que los datos se han registrado correctamente.
- Asegurarse de que los datos están dentro de rangos de valores aceptables, y usar el conocimiento de expertos en la materia para valorar si los valores son normales o no.
- Gestionar los valores nulos: ya sea informando de valores nuevos o eliminando las filas afectadas.
- Nunca acondicionar los datos manualmente porque puede inducir a nuevos errores y no es una práctica reproducible.
- Usar código como una forma de dejar constancia de las acciones de preprocesado de datos que se han llevado a cabo.
- Generar gráficos en este punto del estudio analítico para verificar de un modo visual que el juego de datos tiene un buen aspecto.

Figura 12. Resumen de situación



5. Paso 4: análisis exploratorio

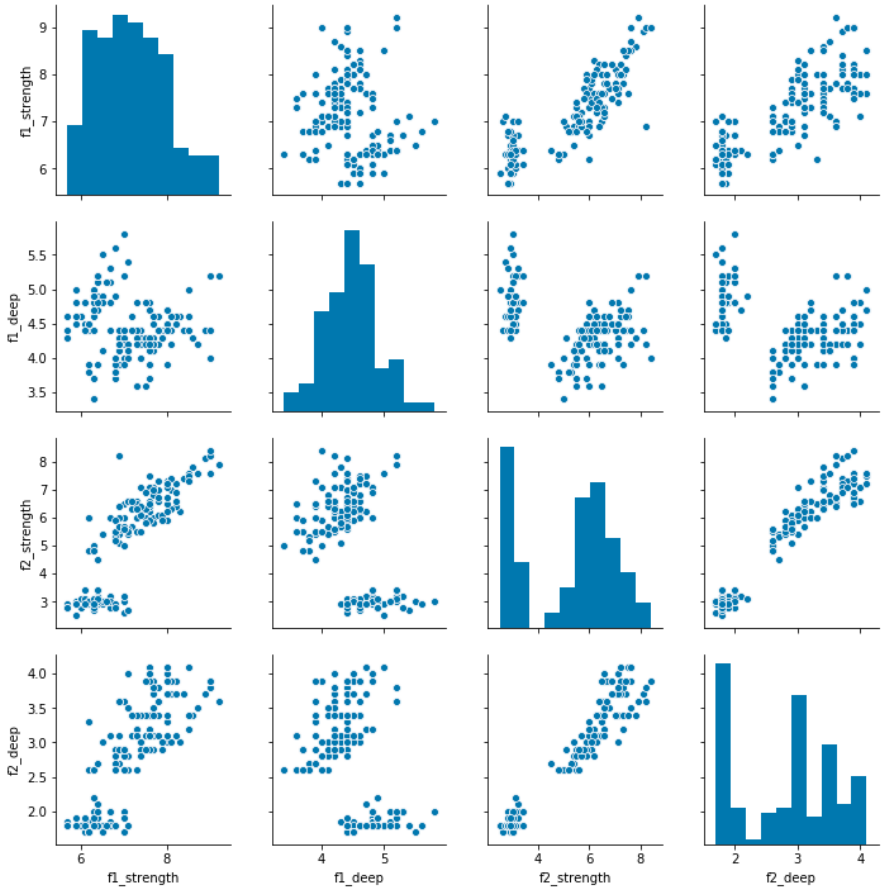
En la fase de análisis exploratorio es en la que empezamos a profundizar más en los datos. En este estadio ya no tenemos valores extremos ni valores nulos ni errores en los datos. De modo que podemos empezar a plantear preguntas del siguiente estilo:

- ¿Qué tipo de distribuciones tengo en mi juego de datos?
- ¿Existen correlaciones entre características?
- ¿Hay factores concretos que puedan explicar estas correlaciones?

Esta es la fase en la que explotamos más a fondo las capacidades de visualización que tengamos al alcance. Generaremos muchos gráficos que no necesariamente van a ser bonitos, pero no debemos preocuparnos de eso en esta fase, ya que serán de uso interno.

Volvamos a usar la matriz de gráficos que ya conocemos de pasos anteriores. Veamos la figura 13.

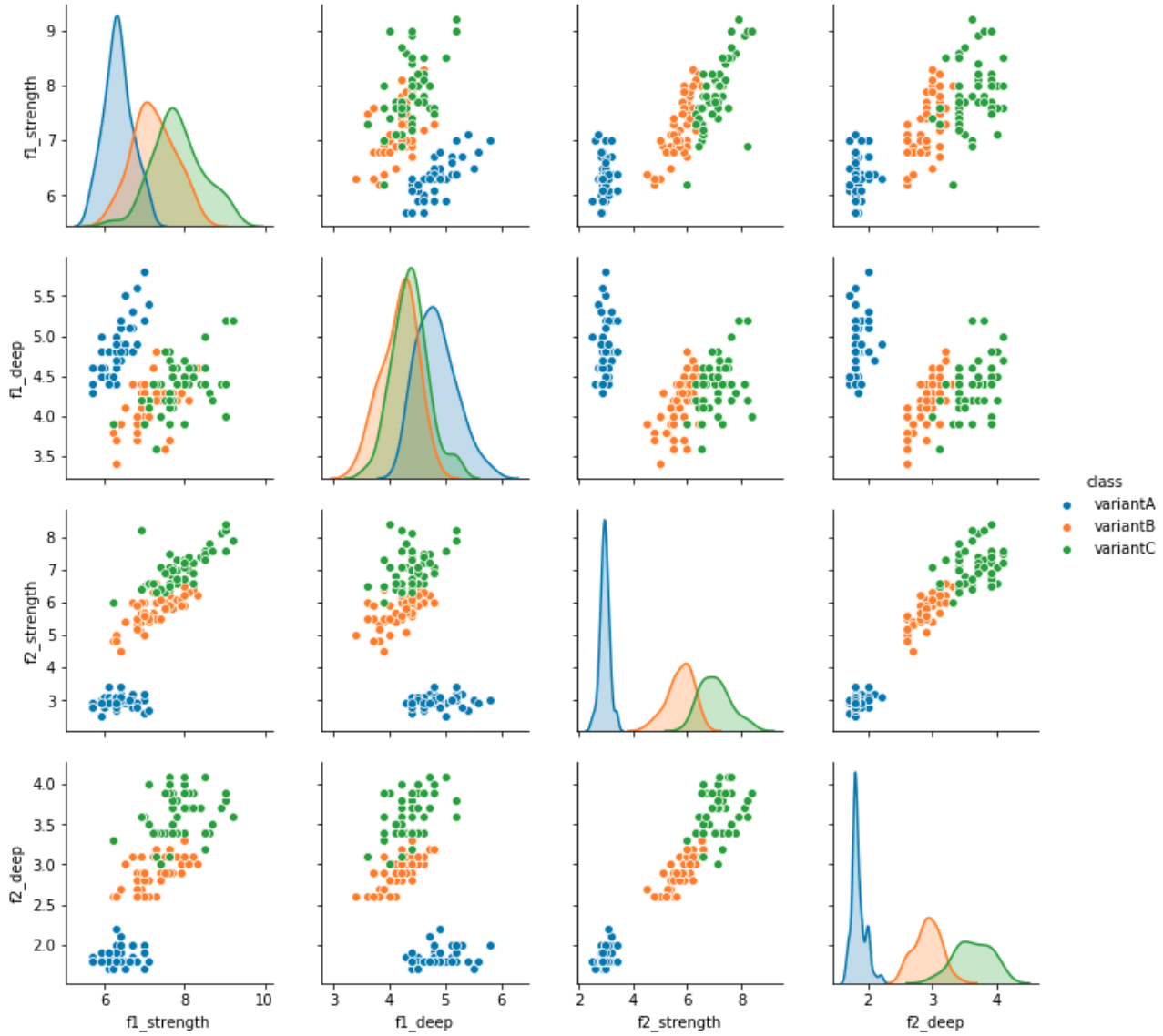
Figura 13. Matriz de gráficos



Observamos que las características f1 (*f1_strength*, *f1_deep*) están normalmente distribuidas, hecho que nos beneficia si vamos a usar modelos que presuponen una distribución normal de datos.

Sin embargo, observamos algo extraño en las características f2. Quizá haya particularidades entre las diferentes variantes de la enfermedad. Asignemos colores distintos a cada clase para entender mejor lo que puede estar ocurriendo. Analicemos la figura 14.

Figura 14. Matriz de gráficos con colores



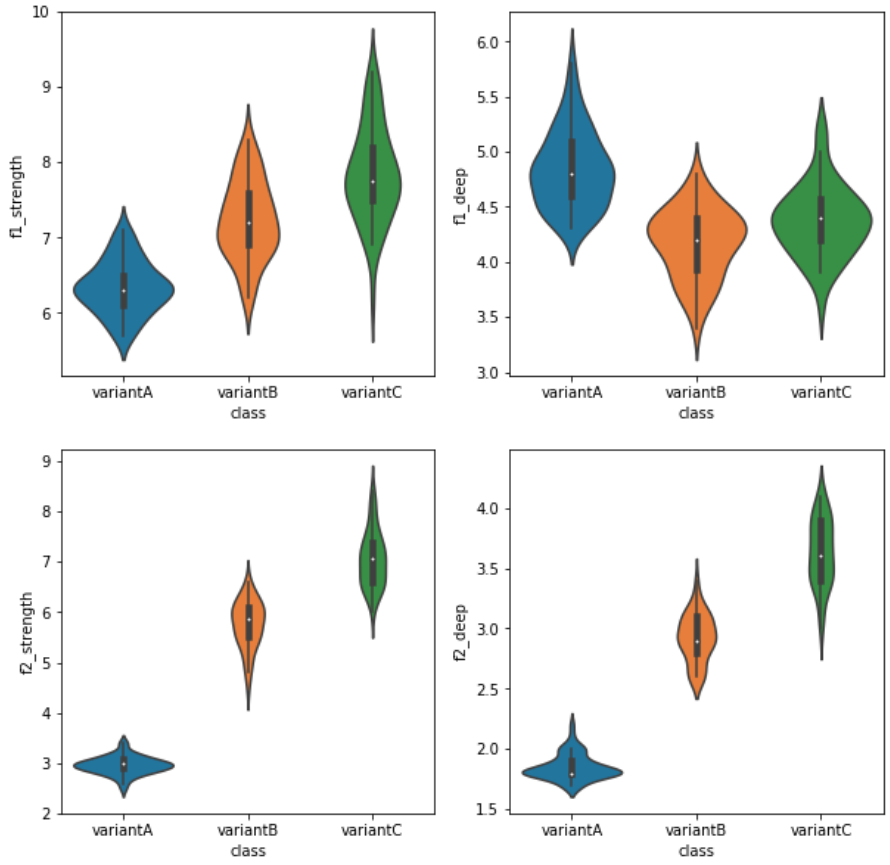
Ahora vemos claro que la distribución extraña que veíamos antes se debe a la mezcla de las distintas clases. Buenas noticias, puesto que apreciamos que efectivamente dentro de cada variante se entrevé una distribución cercana a la normal. Adicionalmente, este comportamiento distinto por variante facilitará el trabajo de clasificación, ya que las distribuciones de *variantA* con respecto al resto de variantes son claramente distintas.

Por otro lado, también anticipamos posibles dificultades en el proceso de clasificación entre *variantB* y *variantC*, ya que las distribuciones de sus características se solapan.

También apreciamos correlaciones entre las medidas f2, así como entre las medidas f1. Los científicos nos aseguran que este hecho era de esperar y que está plenamente dentro de la normalidad.

Usaremos gráficos de violín en la figura 15, *violin plots*, para comparar las distribuciones de las características entre distintas variantes. Los gráficos de violín contienen la misma información que los *box plots*, pero con el añadido de aportar información sobre las zonas de densidad de datos.

Figura 15. Gráficos de violín



Estamos ya en disposición de iniciar la fase de modelado, la más atractiva para la mayoría del público.

6. Paso 5: clasificación

Merece la pena observar que en este punto llevamos mucho trabajo hecho con los datos y todavía no hemos generado ni un solo modelo.

Las tareas anteriores pueden parecer aburridas y tediosas, pero sin las tareas de preprocesado, verificación y exploración, generaríamos ahora modelos de clasificación sin apenas capacidad de generalización.

Apuntad la siguiente máxima: ***bad data leads to bad models***. Verificad siempre vuestros datos antes de nada.

Lo primero que vamos a necesitar para alimentar nuestro generador de modelos de clasificación es proporcionarle dos juegos de datos, el de entrenamiento y el de pruebas.

6.1. Separación del juego de datos

Un *training set* es un subconjunto aleatorio del juego de datos que usaremos para entrenar nuestros modelos.

Un *testing set* es un subconjunto aleatorio del juego de datos (mutuamente excluyente del *training set*) que usaremos para validar la capacidad predictiva de los modelos generados.

Especialmente en juegos de datos dispersos como el nuestro, es fácil que los modelos caigan en el sobreentrenamiento o *overfit*. Es decir, el modelo se adapta en exceso a los datos que conoce y en consecuencia es incapaz de generalizar sobre datos que no ha visto nunca. Por este motivo es importante generar el modelo con un juego de datos y validarlo con otro totalmente distinto.

Es importante notar que una vez hayamos separado en dos nuestro juego de datos (*training set* + *testing set*), deberíamos tratar el *testing set* como si jamás hubiera existido, es decir, no debe participar para nada en el proceso de entrenamiento del modelo; de lo contrario, nos estaremos haciendo trampas a nosotros mismos.

En esta fase habremos separado el juego de datos.

6.2. Generación del modelo

Una vez tenemos nuestro juego de datos separado en *training set* y *testing set* y atendiendo a las directrices de nuestro responsable de datos, vamos a proceder a generar un modelo de clasificación.

Una de las mejores opciones que tenemos es la de empezar por los **árboles de decisión**; estos tienen una propiedad interesante: son escalar-invariantes, es decir, la escala en la que están representadas las características del juego de datos no afecta en absoluto a su rendimiento, contrariamente a otros modelos de minería de datos, que sí requieren que los datos estén previamente normalizados. En otras palabras, no importa si nuestras características toman valores entre 0 y 1 o los toman entre 0 y 1.000; los árboles de decisión funcionarán igualmente en ambos casos.

Hay algunos parámetros con los que podemos jugar para tratar de mejorar el rendimiento de los árboles de decisión. El proceso de ajuste de parámetros en la función de generación del modelo se conoce como proceso de *tunning*.

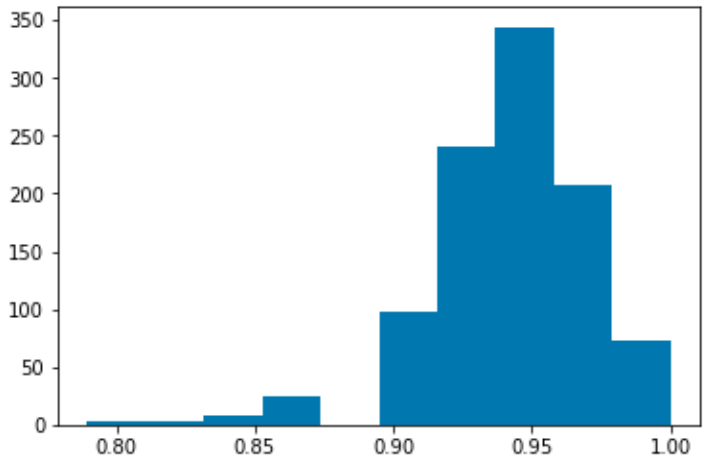
6.3. Gestión del *overfitting*

Dependiendo de cómo se han distribuido los datos entre el *training set* y el *testing set*, nuestro modelo puede llegar a tener niveles de predicción del 80 % al 100 %.

Para ver este efecto, podemos repetir mil veces la tarea de generación aleatoria de juegos de datos *training set* y *testing set* y la generación de árbol de decisión.

En la figura 16 podemos ver los distintos niveles de precisión que alcanza el algoritmo de clasificación en función de la separación del juego de datos que hacemos.

Figura 16. Niveles de precisión



Obviamente nos encontramos ante un problema. No es bueno que en función de la distribución de datos que hagamos tengamos tanta diferencia en la capacidad predictiva de nuestro modelo de clasificación. Este fenómeno recibe el nombre de sobreentrenamiento o *overfitting*. El modelo aprende a clasificar el *training set* tan bien que después es incapaz de generalizar en nuevos datos lo aprendido.

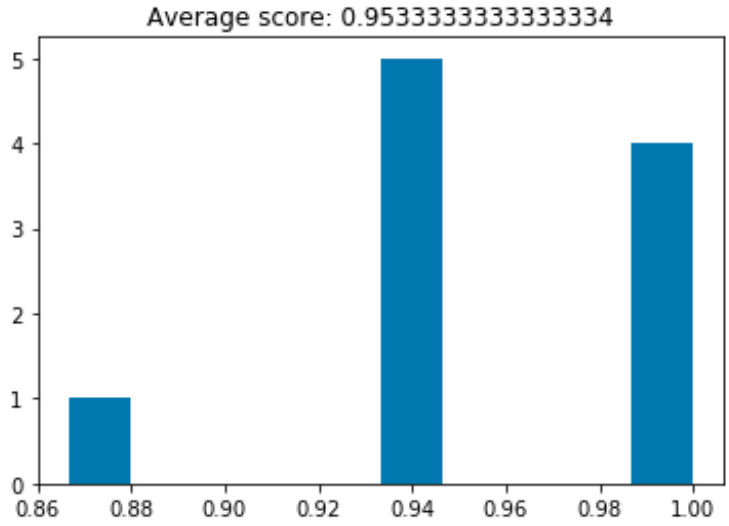
6.3.1. *Cross-validation*

El problema del sobreentrenamiento es la principal razón por la que muchos analistas optan por la funcionalidad del *k-fold cross-validation* en sus modelos: separar el juego de datos original en *k* subconjuntos, para usar uno de ellos como *testing set*, y el resto de subconjuntos como *training set*. Este proceso se repite *k* veces, de modo que cada subconjunto se usa como *testing set* exactamente una vez.

k = 10 es la fórmula más utilizada, de modo que vamos a probarla.

En la figura 17 apreciamos como ahora tenemos unas puntuaciones de predicción más consistentes de nuestro clasificador.

Figura 17. Precisión tras cross-validation



6.4. *Parameter tunning*

Cada modelo de minería de datos tiene asociados una serie de parámetros sobre los que podemos hacer ajustes con la intención de mejorar el modelo. Por ejemplo, podría pasar que si reducimos significativamente un parámetro de nuestro árbol de decisión, la precisión del modelo decaiga significativamente.

En consecuencia, necesitamos disponer de un proceso que nos permita escoger la combinación óptima de parámetros para alcanzar los niveles más altos posibles de rendimiento en nuestros modelos.

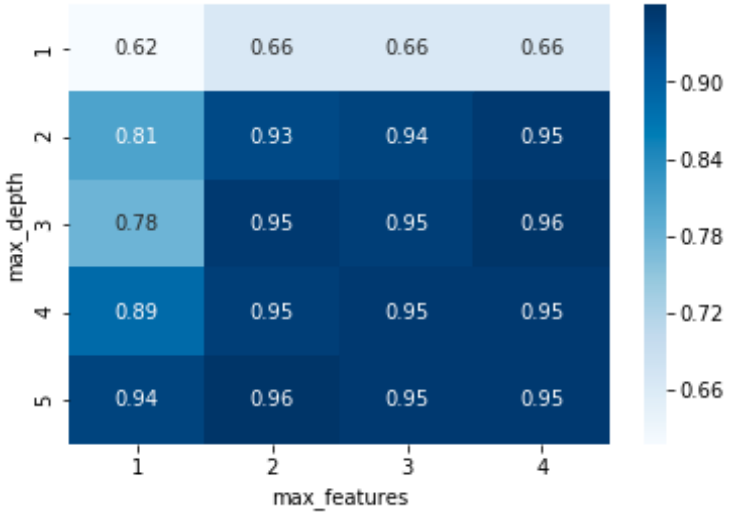
Uno de los métodos más habituales para *parameter tuning* de modelos es *grid search*.

La idea tras este modelo es simple: probar sobre un rango de parámetros hasta encontrar los valores que ofrecen un mejor rendimiento del modelo.

Debemos centrar nuestra intuición como analistas en escoger el mejor rango posible, para que sea el método *grid search* el que se encargue de determinar los valores óptimos dentro del rango que nosotros le hemos fijado. Se trata de un proceso iterativo de adecuación de rangos y búsqueda de valores óptimos.

En la figura 18 visualizamos los niveles de precisión alcanzados tras iterar procesos de *tunning* sobre los parámetros: (*max_deep* y *max_features*).

Figura 18. Niveles de precisión tras *tunning*



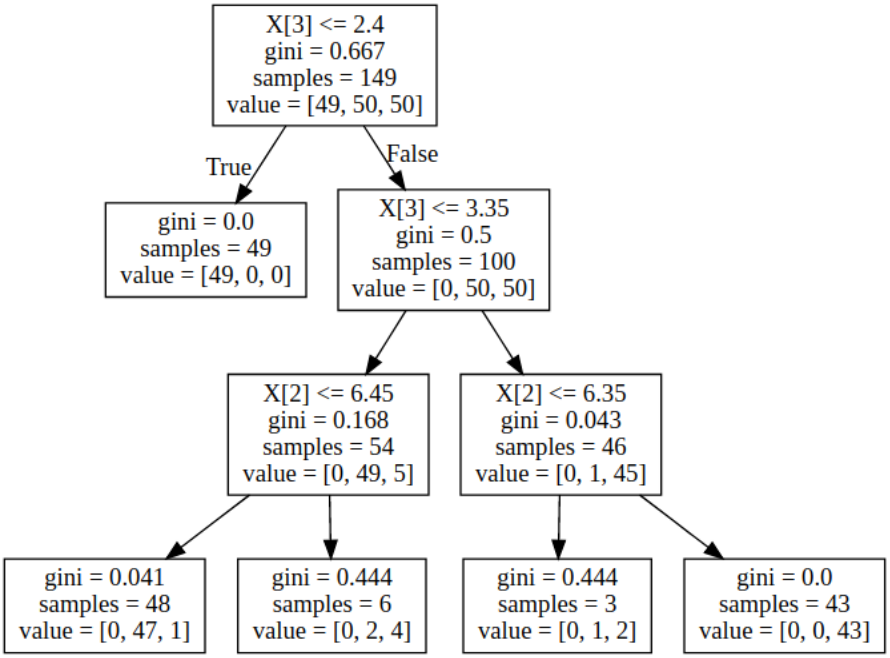
Con este gráfico disponemos de una visión global del espacio de parámetros: con esto sabemos que necesitamos un valor *max_depth* de como mínimo 2 para disponer de un árbol de decisión con precisión aceptable.

Sin embargo, el parámetro *max_features* parece no ser demasiado significativo siempre y cuando mantengamos dos características, hecho que tiene sentido porque nuestro juego de datos tiene solo cuatro.[1]

[1] Recordemos que una de nuestras clases era fácilmente separable usando tan solo una característica.

Podemos visualizar nuestro árbol de decisión en la figura 19, en la que apreciamos qué lógica sigue para tomar sus decisiones.

Figura 19. Árbol de decisión



Acabamos de generar un árbol de decisión a partir de las cuatro primeras características de nuestro juego de datos:

- *X*[0] = *f1_strength*
- *X*[1] = *f1_deep*
- *X*[2] = *f2_strength*
- *X*[3] = *f2_deep*

Observamos como en el primer nodo se plantea la primera pregunta: filas con *f2_deep* por debajo de 2.337.

La respuesta positiva toma value = [49,0,0] indicando que tenemos 49 casos de *variantA*.

La respuesta negativa toma *value* = [0,50,50] indicando que tenemos cincuenta casos de *variantB* y cincuenta casos más de *variantC*.

7. Paso 6: reproducibilidad

Asegurarse de que el trabajo que hemos hecho es reproducible es el último y probablemente más importante paso en cualquier análisis. Como norma general, no deberíamos dar demasiada importancia a un descubrimiento que no podamos reproducir. Como tal, si nuestro análisis no es reproducible es como si no lo hubiéramos hecho.

Un análisis como este se ha hecho deliberadamente extenso para hacer que nuestro estudio sea reproducible. Hemos documentado paso a paso todo lo que íbamos haciendo, disponemos de un registro escrito de texto de lo que hicimos y de por qué lo hicimos. Todo esto otorga credibilidad al propio estudio, ya que lo convierte en verificable y por supuesto en mejorable en tanto en cuanto puede suponer a su vez un punto de partida para nuevas iniciativas.

Resumen

Se ha seleccionado este formato por la sencillez del juego de datos, por el buen encaje en una idea de negocio y sobre todo por el desarrollo meticuloso y riguroso del proceso analítico en el que cada paso está inspirado en un decálogo de buenas prácticas que podrán servir al estudiante para crecer en este ámbito de conocimiento.

Además, el juego de datos ofrece posibilidades más allá de las exploradas en este *notebook*, puesto que tan solo se han usado cuatro de las diecisiete características descriptivas disponibles.