Universidad de Valladolid

FACULTAD DE CIENCIAS

TRABAJO DE FIN DE GRADO

Grado en Físicas

Espectrometría, viticultura y Machine Learning: creación de una base de datos espectrales de distintos vinos para entrenar a un sistema de Inteligencia Artificial.

Autor: Alfonso Martínez, Marcos

Tutoras: Belmonte Sainz-Ezquerra, María Teresa
Fernández del Reguero, Patricia

Año: 2024

Índice general

Capítulo 1

Resumen.

En este trabajo vamos a realizar un estudio sobre la posibilidad de generar modelos de Machine Learning que sean capaces de encontrar patrones en unas muestras de vino utilizando como datos los espectros de absorbancia obtenidos. Mediante el uso de colorimetría, estadística y programación, vamos a generar una serie de datos obtenidos a partir de los espectros con los que podremos alimentar los modelos. Dependiendo de los datos y muestras que tengamos, tendremos que decidir el tipo de problema con el que estamos tratando, si es de clasificación o de agrupación. Finalmente compararemos los modelos para ver similitudes y diferencias en el comportamiento, así como los patrones que se hayan identificado para poder decidir si sería posible, con una base de datos suficientemente grande, elaborar un modelo competente en el futuro. A lo largo del trabajo veremos y entenderemos cómo se entrenan y cómo funcionan los diferentes algoritmos.

Capítulo 2

Abstract.

In this project we will be performing an study on the possibility of generating Machine Learning models capable of finding patterns in the absorbtion spectral data of some wine samples. Making use of colorimetry, statistics and programming, we will be generating a series of data from the spectral data which we will use to train the models. Depending on the data and samples we got, we will define the kind of situation we are in, if it's a classification or a clustering problem. Lastly, we will be comparing the models' results, the similarities and differences, and the patterns they have identified to understand the viablity of this line of work in the future. Throughout this project we will be explaining the steps needed to train the different algorithms and how they work.

Capítulo 3

Fundamento teórico.

3.1. Vinos, tipos de vino y producción.

El sector de producción de vino es uno de los principales en la industria española. Se conoce al vino como la bebida alcohólica obtenida de la fermentación del mosto de la uva. Los principales compuestos del vino son: agua, alcoholes, ácidos, azúcares, compuestos fenólicos, compuestos nitrogenados y compuestos volátiles. Estos compuestos varían dependiendo del tipo de vino, edad, condiciones, y la cosecha de la uva utilizada. Podemos ver los compuestos en la Tabla 3.1, que entra más en detalle sobre los diferentes compuestos, así como sus proporciones (los rangos aportan valores aproximados de los componentes esenciales, ya que hay alrededor de 1000 identificados hasta ahora) [1].

Tabla 3.1: Componentes de los vinos. Fuente: [1]

Componente	**Proporciones por litro**	**Comentarios**
Gases disueltos		
dióxido de carbono	0-50 cc	
dióxido de azufre		
Total	80-200 g	Mayor en vinos dulces
Libre	10-50 mg	Mayor en vinos inestables
Sustancias volátiles		
agua	700-900 g	
etanol (alcohol)	8.5-15 % por vol	Mayor en vinos fortalecidos, menor en vinos de baja graduación
alcoholes superiores	0.15-0.5 g	
acetaldehido	0.005-0.5 g	Mayor en vinos de jerez y similares
esteres	0.1-0.3 g	
ácido acético	0.35-0.6 g	
Sustancias fijas		
azúcar residual	0.8-180 g	Depende del tipo de vino, mayor en vinos dulces
glicerol	5-12 g	
fenólicos	0,2-0.5 g; 1.5-4.0 g	Menor rango para vinos blancos, mayor para tintos
Ácidos orgánicos		
ácido tartárico	3-10 g	Depende del tipo de uva
ácido málico	0-4 g	Depende del clima y la conversión maloláctica
ácido láctico	0-1 g	
ácido succínico	0.2-1.5 g	
ácido cítrico	0-1 g	En vinos a los que se les ha añadido
Sales minerales		
sulfatos	0.1-0.4 g	Expresado como sales potásicas
cloruros	0.25-0.85 g	
fosfatos	0.08-0.5 g	
Minerales		
potasio	0.7-1.5 g	
calcio	0.06-0.9 g	
hierro	0.002-0.006 g	

Los vinos se pueden clasificar de dos maneras, por color o por edad [2].

Por color:

- **Vinos tintos**: Se obtienen de las uvas tintas. Principalmente producidos por la maceración y fermentación. El proceso de maceración consiste en un período de contacto del mosto con la piel, lo que permite la coloración de este y del vino con la subsecuente fermentación.
- **Vinos blancos**: Obtenidos de uvas blancas. El principal método consiste en el prensado de la uva y fermentación sin piel.
- **Vinos rosados**: Se pueden obtener por varios métodos, entre ellos prensado de la uva o maceración, pero reduciendo el contacto con la piel. Mezclar vinos está prohibido en muchos países.

Por edad:

- **Vino joven**: Son vinos que no han pasado por barrica o, de haberlo hecho, ha sido entre 3 y 6 meses.
- **Vino crianza**: Han pasado envejeciendo al menos 24 meses. Primero, se les deja en barrica de roble 12 meses, y después se embotella. Las botellas se almacenan horizontalmente en lugares oscuros.
- **Vino reserva**: Los vinos reserva envejecen mínimo 36 meses. Para ser considerados reserva, tienen que haber permanecido al menos 1 año en barrica y 6 meses en botella, en las mismas condiciones que el caso anterior. Para vinos blancos y rosados el proceso completo se reduce a 2 años, con 6 meses en barrica.
- **Vino gran reserva**: Tienen el período de envejecimiento más largo de todos, aproximadamente 5 años, 2 años en barrica de roble y 2 años en botella (como mínimo). Para blancos y rosados, el total es de 4 años con al menos 6 meses en barrica.

Europa se posiciona como la principal productora de vino del mundo. Dentro de Europa podemos encontrar a los 5 principales productores de vino: Francia, Italia, España, Alemania y Portugal. Podemos apreciar que ha habido una disminución en la producción de vino bastante pronunciada en España y portugal, siendo Españala más afectada en este sector, ya que en 5 años ha bajado su producción un 19 %. Esta disminución se puede achacar a las condiciones climáticas adversas, que no han permitido unas buenas cosechas [3]. Los datos se pueden observar a continuación en la Tabla 3.2.

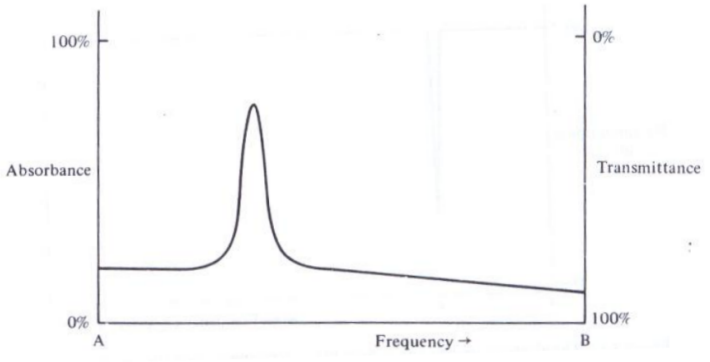
Tabla 3.2: Principales productores de Europa en 5 años. Medidas en millones de hl. Fuente: [3].

País	2018	2019	2020	2021	Prov. 2022	Prev. 2023	23/22 Var.	23/22 Var.(%)	Media 5 años	Var. 5 años (%)
Francia	49.2	42.2	46.7	37.6	45.8	45.8	-0.0	0 %	44.3	3 %
Italia	54.8	47.5	49.1	50.2	49.8	43.9	-5.9	-12 %	50.3	-13 %
España	44.9	33.7	40.9	35.5	35.7	30.7	-5.0	-14 %	38.1	-19 %
Alemania	10.3	8.2	8.4	8.4	8.9	9.0	0.1	1 %	8.9	2 %
Portugal	6.1	6.5	6.4	7.4	6.8	7.4	0.6	8 %	6.6	12 %

3.2. Espectroscopía UV - Vis.

Antes de entrar en espectroscopía, vamos a definir lo que es un espectro de absorción. Si hacemos pasar un haz de luz blanca a través de una muestra de una sustancia, a determinadas longitudes de onda esta interaccionará con las moléculas de la muestra. Las moléculas "absorberán" la luz y utilizarán la energía para excitarse. Si representamos en una gráfica las longitudes de onda (o frecuencias) a las que ha ocurrido este fenómeno, es lo que se conoce como espectro de absorción. En la gráfica, representaremos la cantidad de luz absorbida (absorbancia), pero también podemos representar la luz que ha atravesado nuestra muestra, conocida como transmitancia. Las longitudes de onda en las que se produzca este fenómeno presentarán picos. En la Figura 3.1 podremos apreciar un ejemplo de un espectro de absorbancia y transmitancia de una muestra con un pico característico [4].

Figura 3.1: Representación de espectros de absorbancia y transmitancia. Fuente: [4].



La espectroscopía se puede definir como el estudio de la interacción entre las ondas electromagnéticas y la materia. En nuestro caso, analizaremos cómo nuestras muestras absorben la luz en el espectro UV - Visible a través de los espectros de absorción obtenidos. Los espectros de las sustancias dan información sobre su composición, concentración y estudio de su estructura. La espectroscopía Ultravioleta – Visible es una de las más útiles y es ampliamente usada para análisis cuantitativos. La principal ventaja de este tipo de espectroscopía es que es ampliamente aplicable, requiere una preparación mínima de las muestras (o ninguna) y es fácil de usar [5] [6]. Esta técnica se puede aplicar a materia orgánica e inorgánica por igual, trabajando normalmente entre las longitudes de onda de 100 y 1000 nm.

Para que podamos aplicar esta técnica, la sustancia a analizar debe tener propiedades de absorción o emisión en ese rango. Esto dará forma al espectro cuando lo analicemos, presentando picos característicos. A partir de esta técnica vamos a poder sacar varios parámetros característicos del espectro de absorbancia para cada muestra como son el índice de componentes polifenólicos IPT280, los parámetros de Glories y los parámetros del espacio CIELab, los cuales explicaremos a continuación.

3.3. Parámetros a calcular.

Índice IPT280: Índice de polifenoles totales, IPT por sus siglas. Entre los componentes polifenólicos se encuentran los flavanoles. Estos presentan un pico de absorbancia a 280 nm, en el rango UV. También presentan una absorción típica en 520 nm, característica de las sustancias con color rojo. Este índice se obtiene midiendo la absorbancia del vino a 280 nm, donde presentan un máximo los compuestos mencionados [7] [8].

$$IPT280 = A_{280} \cdot 100 \tag{3.1}$$

Parámetros de Glories: El color es probablemente la principal característica que los consumidores perciben del vino. Es conocido que este aporta información de la calidad, ya que está relacionado con el tipo de vino, su edad y conservación.

El método de Glories (1984) calcula los parámetros de intensidad de color, tonalidad, y los diferentes porcentajes de color (amarillo, rojo, azul y la proporción de color rojo producida por cationes de flavilo). Estos parámetros se pueden obtener a partir de los datos del espectro de absorbancia a las longitudes de onda de 420, 520 y 620 nm. Con estos podremos sacar los siguientes parámetros [9].

$$Densidad\ de\ color \quad CD = A_{420} + A_{520} \tag{3.2}$$

$$Intensidad\ de\ color \quad CI = A_{420} + A_{520} + A_{620} \tag{3.3}$$

$$Tinte \quad T = \frac{A_{420}}{A_{520}} \tag{3.4}$$

$$Proporcion\ de\ color\ rojo\ producido\ por\ cationes\ de\ flavilo \quad dA\,\% = \frac{A_{520} - \frac{A_{420}-A_{620}}{2}}{A_{520}} \cdot 100 \tag{3.5}$$

$$Proporcion\ de\ color\ amarillo \quad Y\,\% = \frac{A_{420}}{CI} \cdot 100 \tag{3.6}$$

$$Proporcion\ de\ color\ rojo \quad R\,\% = \frac{A_{520}}{CI} \cdot 100 \tag{3.7}$$

$$Proporcion\ de\ color\ azul \quad B\,\% = \frac{A_{620}}{CI} \cdot 100 \tag{3.8}$$

Espacios de color CIEXYZ 1931 y CIELAB.

El espacio de color CIEXYZ fue establecido en 1931 por la comisión internacional de L'Eclairage (CIE). Fue el primer método utilizado para calcular el color en los alimentos. El método consiste en determinar los valores triestímulo X, Y y Z que se obtienen a partir de la medida de la transmitancia de las longitudes de onda comprendidas en el espectro visible, [380, 780] nm. El sistema de valores triestímulo se basa en tres estímulos de colores primarios, correspondientes al rojo, verde y azul. Estos estímulos de colores primarios son distribuciones espectrales definiendo lo que se llama el observador estándar [10]. Las fórmulas son las siguientes:

$$X = \frac{K}{N} \int_\lambda S(\lambda) I(\lambda) \overline{x}(\lambda) \tag{3.9}$$

$$Y = \frac{K}{N} \int_\lambda S(\lambda) I(\lambda) \overline{y}(\lambda) \tag{3.10}$$

$$Z = \frac{K}{N} \int_\lambda S(\lambda) I(\lambda) \overline{z}(\lambda) \tag{3.11}$$

Las integrales las calcularemos en el rango del espectro visible, con λ (longitud de onda) comprendida en el rango de [380, 780] nm.

Con N siendo:

$$N = \int_\lambda I(\lambda) \overline{y}(\lambda) \tag{3.12}$$

Y $K = 100$ para normalizar los valores.

$S(\lambda)$ se trata de la transmitancia de la muestra que estemos analizando, que podremos obtener a partir de la absorbancia que ya tenemos. Para poder calcular este dato, podemos utilizar la siguiente relación para calcular el % de transmitancia a partir del dato de la absorbancia:

$$\%T = 10^{2-A} \tag{3.13}$$

$I(\lambda)$ en este caso es la distribución de energía espectral del iluminante utilizado.

El color de los objetos depende la luz con la que lo iluminamos. El término iluminante se usa para significar la distribución espectral teórica de la luz que ilumina un objeto. En nuestro caso, usaremos el iluminante D65, recomendado por el CIE en 1964. Este intenta representar la luz de día con una temperatura de color de aproximadamente 6500 K [11]. La distribución de energía espectral del iluminante D65 se encuentra tabulada, así que obtendremos su valor de esa forma.

Finalmente, las funciones $\overline{x}$, $\overline{y}$, $\overline{z}$ son las funciones de correspondencia de colores. Son la cantidad de cada uno de los estímulos primarios X, Y y Z necesarias para que el ojo humano perciba la mezcla del mismo modo que lo haría como si fuese luz monocromática de una determinada longitud de onda. Estas funciones las podremos obtener con un cálculo aproximado de la siguiente manera [12]:

$$\overline{x}(\lambda) = \sum_{i=0}^{2} \alpha_{xi}\ exp(-\frac{1}{2}[(\lambda - \beta_{xi}) S(\lambda - \beta_{xi}, \gamma_{xi}, \delta_{xi})]^2) \tag{3.14}$$

$$\overline{y}(\lambda) = \sum_{i=0}^{1} \alpha_{yi}\ exp(-\frac{1}{2}[(\lambda - \beta_{yi}) S(\lambda - \beta_{yi}, \gamma_{yi}, \delta_{yi})]^2) \tag{3.15}$$

$$\overline{z}(\lambda) = \sum_{i=0}^{1} \alpha_{zi}\ exp(-\frac{1}{2}[(\lambda - \beta_{zi})S(\lambda - \beta_{zi}, \gamma_{zi}, \delta_{zi})]^2) \tag{3.16}$$

Con S siendo una función selectora que se puede representar con dos funciones de Heaviside, H:

$$S(x, y, z) = y(1 - H(x)) + zH(x) \tag{3.17}$$

Las constantes se encuentran tabuladas, las cuales podremos sacar de la Tabla 3.3:

Tabla 3.3: Constantes de los sumatorios tabuladas. Fuente: [12].

	$\overline{x}_0$	$\overline{x}_1$	$\overline{x}_2$	$\overline{y}_0$	$\overline{y}_1$	$\overline{z}_0$	$\overline{z}_1$
α	0.362	1.056	-0.065	0.821	0.286	1.217	0.681
β	442.0	599.8	501.1	568.8	530.9	437.0	459.0
γ	0.0624	0.0264	0.0490	0.0213	0.0613	0.0845	0.0385
δ	0.0374	0.0323	0.0382	0.0247	0.0322	0.0278	0.0725

Una vez calculemos estos datos, podremos obtener las coordenadas del espacio CIELab. El sistema CIELab proporciona un espacio de color tridimensional en el que las posiciones numéricas de los colores muestran mejor cómo se perciben. El eje x positivo se extiende hasta el valor de +60, correspondiendo al color rojo. Este eje se llama +a*. Perpendicularmente, y en sentido positivo, tenemos el color amarillo en el eje y, llamado +b*. El eje -a* va hacia el color verde y el -b*, hacia el azul. EL valor del eje z positivo corresponde al valor de la luminancia, siendo el eje L*. [10]

Estos valores los podremos calcular con transformaciones simples tal que:

$$L^* = 116(\frac{Y}{Y_n})^{1/3} - 16 \tag{3.18}$$

$$a^* = 500[(\frac{X}{X_n}) - (\frac{Y}{Y_n})^{1/3} - (\frac{Z}{Z_n})^{1/3}] \tag{3.19}$$

$$b^* = 200[(\frac{Y}{Y_n})^{1/3} - (\frac{Z}{Z_n})^{1/3}] \tag{3.20}$$

Con X_n, Y_n y Z_n los valores triestímulo del punto blanco de referencia correspondiente al iluminante que utilicemos, en nuestro caso [13]:

$$\begin{aligned} X_n &= 95{,}0489 \\ Y_n &= 100 \\ Z_n &= 108{,}8840 \end{aligned} \tag{3.21}$$

Finalmente podremos sacar 2 parámetros adicionales que están relacionados con estos parámetros recientemente calculados.

$$CIE\ 1976\ a, b\ (CIELAB)\ croma:\ C^*_{ab} = (a^{*2} + b^{*2})^{1/2} \tag{3.22}$$

$$CIE\ 1976\ a, b\ (CIELAB)\ tono:\ h_{ab} = arctan(b^*/a^*) \tag{3.23}$$

Correspondiendo C^*_{ab} con el croma y h_{ab} con el tono.

3.4. Machine learning.

Machine learning (aprendizaje automático), es la ciencia de programar ordenadores de manera que aprendan de los datos que les alimentamos, dándoles la habilidad de aprender sin estar explícitamente programados para ello. El ejemplo más claro de esto es el filtro de spam que todos tenemos en nuestro correo o smartphone.

Para explicar mejor cómo funciona esto, podemos utilizar el ejemplo del filtro de spam. Si quisiéramos crear uno a mano, primero tendríamos que analizar los mensajes de spam que nos llegan y detectar

palabras o frases que los identifiquen, como “tienes un paquete esperando en aduanas”, “tu cuenta ha sido bloqueada” o “contactamos contigo para ofrecerte trabajo en remoto”. También podemos analizar el remitente del mensaje, asunto, etc. Estas reglas habría que implementarlas a mano, e ir introduciendo nuevas hasta que finalmente tengamos un filtro suficientemente bueno.

Todos estos pasos anteriores serían realizados automáticamente por nuestro algoritmo de Machine Learning, encontrando patrones simplemente alimentando el modelo con mensajes diferenciados entre spam y no spam. La principal ventaja de todo esto es el mantenimiento, ya que, si surgen nuevos prototipos de spam o se modifican los antiguos, en nuestro modelo creado a mano habría que modificar las reglas una a una y crear nuevas, hasta el infinito. En cambio, con el modelo de Machine Learning, podremos reentrenarlo para que detecte estos nuevos patrones.

En general, el aprendizaje automático es muy útil en casos como problemas para los que las soluciones existentes son demasiado tediosas o requieren de demasiadas reglas, problemas demasiado complejos para los que las soluciones tradicionales no sirven, entornos fluctuantes que requieren de una adaptación continua, o sacar conclusiones de problemas complejos y grandes cantidades de datos.

Aunque hay muchas maneras de diferenciar modelos, la principal división entre modelos de Machine Learning que se puede hacer es si han sido entrenados o no con supervisión humana (supervisados, no supervisados, semisupervisados y entrenamiento por refuerzo) [14] [15].

Aprendizaje supervisado.

En este tipo de aprendizaje, en los datos que alimentamos al modelo se encuentran las soluciones, llamadas etiquetas.

Los problemas más típicos son de clasificación o regresión. En los problemas de clasificación, los datos caen en clases (spam o no spam, por ejemplo), y nuestro modelo lo que hace es aprender a clasificar los nuevos datos de los que no tenemos la respuesta. Los problemas de regresión lo que hacen es predecir un resultado numérico en un continuo, como por ejemplo el precio de un determinado producto o lo que se estima que se va a gastar un cliente en el futuro.

Los algoritmos más importantes de aprendizaje supervisado son: K-Nearest Neighbors (K primeros vecinos o KNN), Regresión lineal, Regresión logística, Máquinas de vector soporte (Support Vector Machines o SVMs), Árboles de decisión y Random Forests, Redes neuronales.

Aprendizaje no supervisado.

La diferencia principal con el anterior grupo es que en este caso los datos no tienen etiquetas, el modelo intenta encontrar patrones sin saber la respuesta. Entre estos modelos podemos encotrar aplicaciones de clusterización, detección de anomalías o reducción de la dimensionalidad.

Los principales modelos en este caso son: K–Means, DBSCAN, Hierarchical Cluster Analysis (HCA o clústerning jerárquico aglomerativo.), One-Class SVM, Isolation Forest, Kernel PCA, Locally Linear Embedding (LLE), t-Distributed Stochastic Neighbor Embedding (t-SNE), Apriori, Eclat.

Aprendizaje semi supervisado.

En algunos casos, vamos a tener datos que no estén etiquetados y otros sí, ya que etiquetar es un proceso que requiere de bastante tiempo y recursos. Algunos algoritmos pueden encargarse de esto. Un ejemplo para entender esto sería los algoritmos que se han ido implementando en las galerías de fotos de los móviles. Si subimos una serie de fotos en los que salen varios elementos en común, el algoritmo es capaz de identificar estos elementos (por ejemplo una persona, un edificio, etc.) en las diferentes fotografías, ahora estaría en nuestra mano asociarlas una etiqueta para que en el futuro ya el algoritmo sea capaz de hacerlo solo.

La mayoría de estos algoritmos son combinaciones de los anteriores.

Aprendizaje por refuerzo.

El aprendizaje por refuerzo se trata de un sistema bastante diferente. En este caso, el modelo se llama agente. Este observa el entorno y puede seleccionar y realizar acciones. Estas acciones son recompensadas o penalizadas. El sistema aprende por sí mismo cuál es la mejor estrategia, llamada política, para conseguir el mayor número de recompensas. La política es lo que define las acciones que el agente tomará en cada situación en la que se encuentre. Los ejemplos más notables de este caso son los robots que aprenden a andar, o los que juegan al ajedrez.

En nuestro caso, anticipándonos a los resultados que vamos a mostrar, vamos a plantear nuestro caso como un problema de clasificación no supervisado, ya que no tenemos una determinada etiqueta sobre la que queramos realizado una predicción, por lo que haremos será utilizar algoritmos de agrupación que creen clústeres de vinos para después analizar las características comunes a esas agrupaciones y sacar conclusiones. A continuación desarrollaremos más sobre los principales algoritmos.

3.4.1. Algoritmos de aprendizaje supervisado.

Aunque hemos dicho que usaremos algoritmos de aprendizaje no supervisado, nos ayudaremos en un algoritmo de aprendizaje supervisado para entrenar uno de los modelos no supervisados.

K-nearest neighbors, KNN.

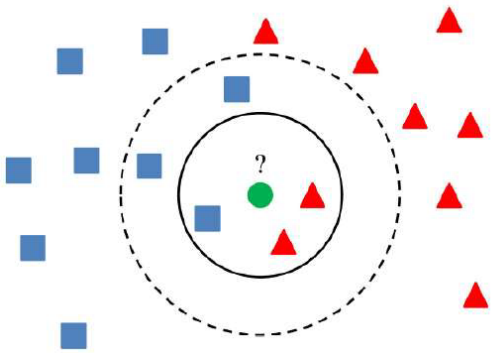
Este es un modelo peculiar, ya que no saca ninguna función de los datos de entrenamiento, sino que se aprende el set de datos del que se alimenta.

Se puede resumir en el siguiente comportamiento:

1. Se elige un número k de vecinos y una métrica para la distancia (normalmente suele ser euclídea).
2. Encuentra el número k de vecinos más cercanos entre los datos.
3. Asigna una etiqueta al dato que queramos clasificar de acuerdo a una votación, la de la etiqueta que tenga el mayor número de vecinos (los que sean más cercanos de acuerdo a los datos).

Lo idóneo es que el número k sea un número impar, ya que así nunca vamos a tener un empate en la votación. No existe una forma de saber el mejor valor para k a priori, depende de cada conjunto de datos, generalmente se prueba con varios y nos quedamos con el mejor. Si tomamos la figura 3.2, si tomásemos 3 primeros vecinos, la nueva muestra se clasificaría como un triángulo rojo, pero si tomásemos 5, lo haría como un cuadrado azul.

Figura 3.2: Representación visual del comportamiento del KNN. Fuente: [16]



La principal ventaja de este modelo es que se adapta según vamos obteniendo nuevos datos. Por otra parte, la principal desventaja es que la complejidad de clasificar nuevos ejemplos crece con el número de ejemplos con el que entrenemos el modelo. Esto se puede evitar si nuestros datos tienen pocas dimensiones (features) o una estructura eficiente. Por eso, el almacenamiento puede ser un problema si trabajamos con grandes cantidades de datos.

3.4.2. Algoritmos de aprendizaje no supervisado.

Para la parte de algoritmos no supervisados nos vamos a centrar en la parte de clustering o agrupamiento. El objetivo del clustering es agrupar elementos similares en clusters para después poder analizar las caracerísticas comunes a cada grupo. Como en la clasificación, cada elemento es asignado a un grupo, salvo que en este caso es una tarea no supervisada. Se trata de una buena herramienta para crear modelos para análisis de datos, análisis de imágenes, etc. En este trabajo nos vamos a centrar en 3 de ellos: K-Means, DBSCAN y clústering jerárquico aglomerativo. Hemos seleccionado K-Means por ser de los más simples de implementar y ejecutar, a la par que eficiente. A diferencia de K-Means que se basa en distancias, DBSCAN es un algoritmo basado en densidad de puntos, por lo que lo usaremos y comprobaremos como se comportan algoritmos con estas características. Finalmente, entrenaremos modelos de clústering jerárquico aglomerativo por ser de los más típicos a la hora de agrupar elementos basados en su similitud.

K-Means.

El K-Means es un algoritmo simple y eficiente. Fue propuesto en Stuart Lloyd at Bell Labs en 1957, así que a veces a este algoritmo se le conoce como Lloyd-Forgy. La principal ventaja que tiene es que converge rápidamente. Una de las cosas que tenemos que tener en cuenta a la hora de entrenar este algoritmo es que le tenemos que indicar el número de centroides que va a tener nuestro modelo. Los centroides serán el centro de cada clúster, y cada dato será asignado a uno de los k clústeres.

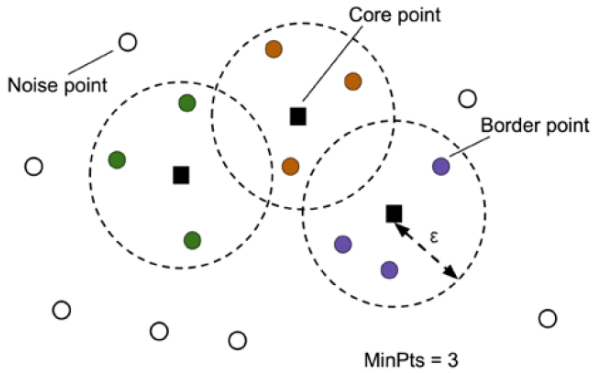
El algoritmo funciona de la siguiente manera, primero coloca los centroides aleatoriamente, después añade etiquetas a los datos, actualiza los centroides, añade etiquetas, y así hasta que los centroides se dejan de mover. El algoritmo tiene garantizado converger en un número finito de iteraciones (generalmente son pocas, por lo que es bastante eficiente).

Aunque el K-Means es muy bueno, no es perfecto. Hay que realizar varias iteraciones y tenemos que determinar nosotros el número de clústeres que vamos a tener. Además, no se comporta muy bien cuando los clústeres tienen tamaños variables, diferentes densidades o formas extrañas.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

Este algoritmo define los clústeres como regiones continuas de alta densidad. Por cada registro, el algoritmo cuenta cuantos registros hay en una pequeña distancia ε. Esta región se conoce como la ε neighborhood. Si un registro tiene al menos un número umbral que definiremos (incluido el mismo registro) de registros en esta región, entonces decimos que es un registro núcleo o core. Los puntos que se encuentran en la región pero no son núcleo, son conocidos como puntos frontera o border. Todos los registros que se encuentren en la región de un núcleo pertenecen al mismo clúster, por lo que podemos decir que una serie de núcleos vecinos forman un mismo clúster. Cualquier registro que no sea núcleo y no tenga uno en su región, es considerado una anomalía o ruido. Este comportamiento se puede ver representado en la Figura 3.3.

Figura 3.3: Representación visual del comportamiento del DBSCAN. Fuente: [15].



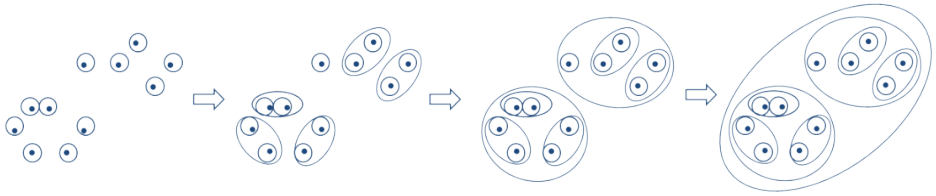
Este algoritmo funciona muy bien si los clústeres son densos y están bien definidos, separados por regiones de baja densidad.

En resumen, este algoritmo funciona muy bien para clústeres de cualquier forma y tamaño, siempre que la densidad varíe significativamente entre estos.

Clustering jerárquico.

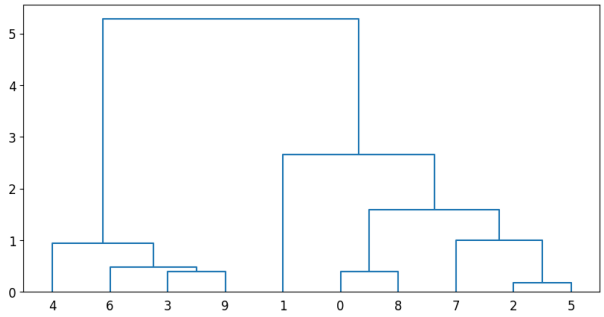
El algoritmo de clústering jerárquico de aglomerativo probablemente sea el más fácil de explicar. Consiste en ir agrupando los registros por proximidad, como construyendo un árbol de abajo hacia arriba. Podemos imaginar un conjunto de gotas de agua que, cuando se acercan lo suficiente unas a otras, se juntan formando una gota mayor, hasta crear una única gota, como se puede ver en la figura 3.4.

Figura 3.4: Representación del comportamiento del clustering jerárquico [17].



Si en vez de representar en el plano los datos lo hiciésemos en forma de árbol, representando los clústeres como ramas que van ascendiendo, tendríamos lo que se conoce como dendograma. El dendograma permite visualizar el comportamiento del modelo muy fácilmente. También nos permite ver dónde deberíamos poner el límite de clústeres, permitiendo elegir este valor de manera visual. Otra forma de limitar el número de clústeres es estableciendo una distancia máxima entre puntos. Un ejemplo de dendrograma lo podemos ver en la Figura 3.5, representando una agrupación de 10 puntos.

Figura 3.5: Ejemplo de dendrograma.



3.4.3. Análisis de Componentes Principales, PCA.

El PCA es un método estadístico que permite simplificar la complejidad de espacios muestrales con muchas dimensiones a la vez que conserva su información.

Supóngase que existe una muestra con n individuos cada uno con p variables (X_1 , X_2, ..., X_p), es decir, el espacio muestral tiene p dimensiones. PCA permite encontrar un número de factores subyacentes (z<p) que explican aproximadamente lo mismo que las p variables originales. Donde antes se necesitaban p valores para caracterizar a cada individuo, ahora bastan z valores. Cada una de estas z nuevas variables recibe el nombre de componente principal.

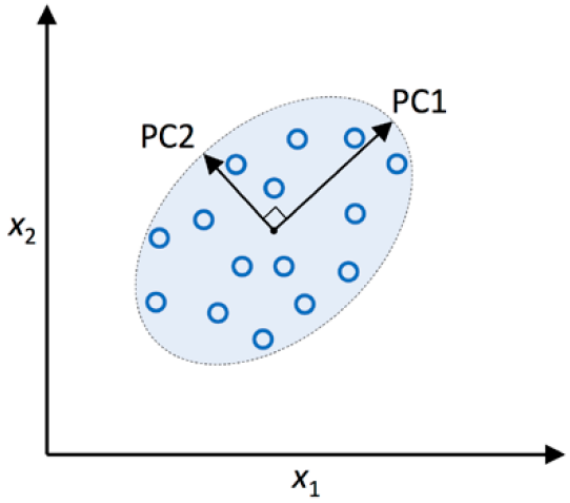
El método de PCA permite por lo tanto "condensar" la información aportada por múltiples variables en solo unas pocas componentes. Esto lo convierte en un método muy útil de aplicar previa utilización de otras técnicas estadísticas tales como regresión, clustering, etc.

Para calcular el PCA debemos obtener la matriz de covarianza de nuestros datos. (Ejemplo para 2 variables).

$$V(b) = V\begin{pmatrix} b_0 \\ b_1 \end{pmatrix} = \begin{bmatrix} var(b_o) & cov(b_0, b_1) \\ cov(b_0, b_1) & V(b_1) \end{bmatrix} \tag{3.24}$$

Realizamos transformaciones lineales de nuestra matriz de covarianza: transformaciones de los puntos en el plano. Las transformaciones nos devuelven dos vectores (PC1 y PC2, en la Figura 3.6). Estos son los vectores propios. Las transformaciones nos devolverán los valores propios. Los valores más altos son los que representan la mayor varianza de nuestros datos. En el método PCA, cada una de las componentes se corresponde con un autovector, y el orden de componente se establece por orden de autovalor.

Figura 3.6: PCA aplicado a 2 variables x_1 y x_2. Fuente: [15].



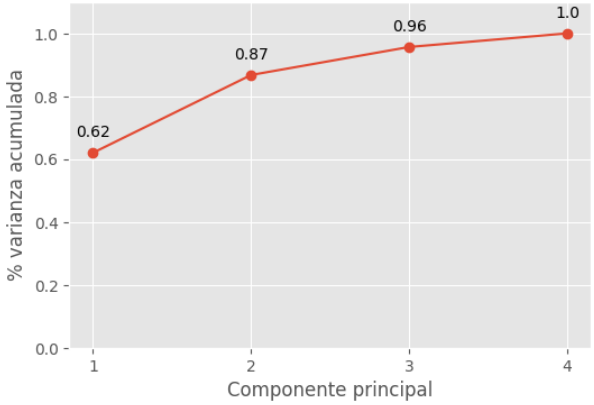
Los dos nuevos vectores se conocen como componentes principales. A la hora de calcular las componentes principales, cada una se obtiene por combinación lineal de las variables originales. Se pueden entender como nuevas variables obtenidas al combinar de una determinada forma las variables originales. La primera componente principal de un grupo de variables (X_1 , X_2, ..., X_p) es la combinación lineal de dichas variables que tiene mayor varianza.

El proceso para calcular las componentes principales es el siguiente:

1. Centralización de las variables: se resta a cada valor la media de la variable a la que pertenece. Con esto se consigue que todas las variables tengan media cero.
2. Se obtiene cada componente a través de la optimización de combinaciones lineales para obtener aquellas con la máxima varianza.
3. Una vez calculada la primera, se obtiene la segunda y se repite iterativamente hasta calcular todas las posibles componentes, o hasta que se decida detener el proceso (por ejemplo indicando un número máximo de componentes a calcular).

Algunos puntos a tener en cuenta antes de aplicar PCA son los siguientes. PCA identifica direcciones cuya varianza es mayor. Por ello deberemos tener los datos en la misma escala. Al trabajar con varianzas, PCA es altamente sensible a outliers. Es muy recomendable estudiar si los hay. ¿Cuánta información presente en el set de datos original se pierde al proyectar las observaciones en un espacio de menor dimensión? Una forma de ver esto es elaborando una gráfica con la varianza acumulada de las componentes principales, Figura 3.7, donde elegiremos con cuántas nos quedamos.

Figura 3.7: Varianza acumulada de 4 componentes principales.



Se puede entender la Figura 3.7 de la siguiente forma. Una vez calculadas las componentes principales, para saber el número de componentes que nos conviene utilizar nos fijamos en la varianza. Las primeras componentes se calculan de forma que si proyectamos nuestros datos sobre ellas, perdemos la mínima información. Por ejemplo, si proyectamos únicamente sobre la componente 1, conseguiríamos explicar el 62 % de la varianza original, si proyectamos sobre la 1 y la 2, un 87 %. lo óptimo es maximizar la varianza acumulada minimizando el número de componentes a utilizar, aunque conviene buscar un número alto como umbral de varianza para no perder demasiada información, por ejemplo un 90 %.

Capítulo 4

Muestras, tratamiento y herramientas de trabajo.

En este apartado vamos a ver las muestras con las que se ha trabajado, el tratamiento de estas antes de obtener sus espectros y los materiales utilizados para obtener dichos espectros. También vamos a ver las distintas librerías dentro del lenguaje Python utilizadas para analizar y limpiar los datos, así como para entrenar los modelos de Machine Learning.

Figura 4.1: Fotografía del espacio de trabajo.

En la Figura 4.1 se puede ver la mesa del laboratorio donde se tomaron los espectro de absorción de las muestras de vino así como estas almacenadas en tubos y los materiales utilizados. Junto al ordenador podemos ver el espectrómetro utilizado, el cual describiremos más adelante y, en pantalla, un espectro de absorción de uno de los vinos tintos.

4.1. Muestras y tratamiento.

Como veremos a continuación, tenemos 43 muestras de vinos (tintos, rosados y blancos) proporcionadas por el equipo de investigación de enología de la Universidad de Valladolid, en concreto del área de microbiología de la Escuela Técnica Superior de Ingenierías Agrarias de Palencia. Cada muestra es diferente a las demás, ya sea por el tipo de vino, el año, el tipo de uva, tratamiento de riego, etc., para poder tener diferentes características. Esto lo podemos ver detalladamente en la Tabla 4.1

Tabla 4.1: Muestras analizadas. Fuente: Laboratorio de enología de la Universidad de Valladolid.

Muestra	Vino	Variedad	Año	Otros datos
1	Rosado G1-testigo	-	2018	Botella vino 3 abierto 14/09/23
2	Blanco R1D2	Verdejo	2016	Botella vino 5 abierto 14/09/23
3	Rosado G1-testigo	-	2018	Abierto 14/09/23 y conservado en falcon 4ºC
4	Tinto Pesquera T7A III-IV	Tempranillo	2022	Botella vino 7 abierto 147/09/23
5	Blanco R1D2	Verdejo	2016	Abierto 14/09/23 y conservado en falcon 4ºC
6	Tinto Pinna Fidelis. Comercial.	Tempranillo	2006	Botella vino 8 abierto 14/09/23
7	Tinto Pesquera T7A III-IV	Tempranillo	2022	Abierto 14/09/23 y conservado en falcon 4ºC
8	Tinto Pinna Fidelis. Comercial.	Tempranillo	2006	Abierto 14/09/23 y conservado en falcon 4ºC
9	Tinto G2-testigo	-	2018	Abierto 14/09/23 y conservado en falcon 4ºC
10	Blanco R1D1D2	Verdejo	2018	Abierto 14/09/23 y conservado en falcon 4ºC
11	Blanco R1D1D2	Verdejo	2017	Abierto 14/09/23 y conservado en falcon 4ºC
12	Tinto G2-testigo	-	2018	Botella vino 9 abierto 14/09/23
13	Blanco R1D1D2	Verdejo	2018	Botella vino 10 abierta 14/09/23
14	Blanco R1D1D2	Verdejo	2017	Botella vino 11 abierta 14/09/23
15	Rosado maceración G1	-	2008	Recién abierta (27/09/23)
16	Tinto Rosum Toro. Comercial	Tinta de Toro	2016	Abierta en 2018
17	Tinto Rioja Siglo Crianza. Comercial	-	1992	Abierta en 1996
18	Blanco. Cigales. Las Luceras. Comercial	Verdejo	2012	Abierta en ¿?
19	Tinto G1 Regadío	Tempranillo	2014	Recién abierta (27/09/23)
20	Blanco R2D3D4	Verdejo	2015	Recién abierta (27/09/23)
21	Blanco R2D1D2	Verdejo	2015	Recién abierta (27/09/23)
22	Blanco R2T3T4	Verdejo	2017	Recién abierta (27/09/23)
23	Blanco R1T1T2	Verdejo	2017	Recién abierta (27/09/23)
23 Nuevo	Blanco	Verdejo	???	???
24	Blanco R2D1D2	Verdejo	2018	Recién abierta (27/09/23)
25	Blanco R1D3D4	Verdejo	2018	Recién abierta (27/09/23)
26	Tinto Pesquera espontánea T07 I-II	Tempranillo	2022	Recién abierta (27/09/23)
27	Tinto Pesquera espontánea T07 III-IV	Tempranillo	2022	Recién abierta (27/09/23)
28	Tinto Pesquera espontánea T7P I-II	Tempranillo	2022	Recién abierta (27/09/23)
29	Tinto Pesquera espontánea T7P III-IV	Tempranillo	2022	Recién abierta (27/09/23)
30	Tinto Pesquera T0P I-II	Tempranillo	2022	Recién abierta (27/09/23)
31	Tinto Pesquera T03 I-II	Tempranillo	2022	Recién abierta (27/09/23)
32	Tinto Pesquera T07 I-II	Tempranillo	2022	Recién abierta (27/09/23)
33	Tinto Pesquera T7P I-II	Tempranillo	2022	Recién abierta (27/09/23)
34	Tinto Pesquera T15 I-II	Tempranillo	2022	Recién abierta (27/09/23)
35	Alteración de vino 30 con K2S2O5	Tempranillo	2022	30 mg/100 mL K2S2O5
36	Alteración de vino 30 con tartárico	Tempranillo	2022	0,6 mg/100 mL tartárico
37	Alteración de vino 30 con cítrico	Tempranillo	2022	0,6 mg/100 mL cítrico
38	Tinto Pesquera T03 I-II	Tempranillo	2023	Depósito. Recién acabada la fermentación
39	Tinto Pesquera T03 III-IV	Tempranillo	2023	Depósito. Recién acabada la fermentación
40	Tinto Pesquera T07 I-II	Tempranillo	2023	Depósito. Recién acabada la fermentación
41	Tinto Pesquera T07 III-IV	Tempranillo	2023	Depósito. Recién acabada la fermentación
42	Tinto Pesquera T15 I-II	Tempranillo	2023	Depósito. Recién acabada la fermentación
43	Tinto Pesquera T15 III-IV	Tempranillo	2023	Depósito. Recién acabada la fermentación

Entre la información que encontramos en la tabla tenemos:

- **Muestra:** Indica el número de la muestra de vino.
- **Vino:** El tipo de vino analizado, indicando si se trata de tinto, blanco o rosado.
- **Variedad:** Indica la variedad del vino.
- **Año:** Año del vino.
- **Otros datos:** Información relevante sobre los vinos, como por ejemplo cuando se abrió la botella, cómo se conservó o si ha sido alterado.

Además, tenemos también la siguiente información sobre las muestras:

- Tinto Pesquera I-II y III-IV son duplicados, se han elaborado igual con la misma uva pero en distintos depósitos.
- Tinto Pesquera T03, T07... es el mismo viñedo pero con distintos tratamientos de riego.
- Las muestas 1-14 se toma muestra recién abierta la botella y se guarda en falcon a 4°C y dos semanas mas tarde se toma otra muestra de la botella abierta.
- Muestras 15-19, varios comerciales y, ya que las muestras fueron suministradas por el equipo de investigación de enología, algunos de ellos son vinos de prácticas..
- Muestras 20-25, misma uva Verdejo, distintos tratamientos de riego.
- Muestras 26-29, se elabora con fermentación espontánea, en el resto de los vinos se inoculan levaduras.

- Muestras 30-34, mismo viñedo, distintos riegos.
- Muestras 35-37, los han alterado en el falcon.
- Muestras 38-43, vendimia de 2023, recién terminada la fermentación.

4.2. Espectrómetro UV – Vis.

El instrumento utilizado para analizar las muestras ha sido el Espectrofotómetro Modelo 6850 UV/Visible, 230V, JENWAY, Figura 4.2. Este espectrofotómetro tiene banda variable, doble rayo y una interfaz local integrada. Los modos incluidos son específicos para fotometría, concentraciones, varias longitudes de onda, escaneo de espectros, cuantificación, cinética y análisis de ADN y proteínas. Se puede controlar directamente desde un ordenador.

Figura 4.2: Espectrómetro utilizado.

A continuación, podremos ver en la Tabla 4.2 los datos más técnicos del mismo.

Tabla 4.2: Especificaciones técnicas del espectrómetro. Fuente: [18].

Rango	190 a 1100 nm
Resolución	0.1 nm
Precisión	± 0.3 nm (a 0.5 y 1 nm ancho de banda) ± 0.5 nm (a 2, 4 y 5 nm ancho de banda)
Repetibilidad	± 0.2 nm
Ancho de banda espectral	Variable 0.5 / 1 / 2 / 4 y 5 nm
Transmitancia	0 a 200 %
Absorbancia	-0.3 a 3.0A
Precisión	±0.3 %T (0 – 100 %T), ±0.002 A (0 – 0.5 A)
Reproducibilidad	±0.001 Abs (0 a 0.5 Abs) ±0.002 Abs (0.5 a 1.0 Abs)
Resolución	0.1 %T, 0.001 A
Luz parásita	<0.05 % a 360nm y 220nm
Ruido	0.0005 A
Estabilidad	±0.001 A a 500 nm después de 15 min
Altura del rayo	15 mm
Fuente d eluz	Lámparas de Tungsteno y Deuterio

A la hora de hacer las medidas, se han realizado disoluciones de las muestras de vino con las siguientes concentraciones:

- **Blanco:** 1:10 (200 μl vino + 1800 μl agua ultrapura Milli-Q)
- **Clarete:** 1:20 (100 μl vino + 1900 μl agua ultrapura Milli-Q)
- **Tintos:** 1:20 (100 μl vino + 1900 μl agua ultrapura Milli-Q)

Para analizar las muestras dentro del espectrofotómetro, se introducen en una cubeta de cuarzo Figura 4.3. Se utiliza cuarzo ya que, a diferencia del plástico, esta cubeta permite realizar mediciones en el espectro UV. Esta se lavaba después de cada medición, ya que no es de usar y tirar, como las de plástico, por ser bastante más cara que estas.

Figura 4.3: Cubeta utilizada para analizar las muestras.

Una vez tenemos las muestras preparadas, las introduciremos en el espectrómetro, que funciona de la siguiente manera: en primer lugar, nuestra fuente de luz corresponde a lámparas de deuterio y tungsteno, que como hemos visto en el fundamento teórico, serán nuestro iluminante D65, la fuente de luz blanca. Un haz de luz de esta fuente pasará a través de un monocromador, que hará que solo pasen las longitudes de onda seleccionadas, permitiendo realizar el barrido. Este haz a la longitud de onda determinada pasará a través de nuestra cubeta con la muestra. Como hemos mencionado antes, el cuarzo nos permite realizar mediciones en el espectro UV - Visible. El haz que ha atravesado la muestra finalmente llega a un detector que, generalmente, es una placa fotosensible. Al incidir la luz sobre esta, se generará una corriente que amplificaremos y representaremos en nuestra gráfica, dando lugar al espectro de absorción de la muestra introducida una vez se haya completado el barrido por las longitudes de onda.

4.3. Análisis de los datos y entrenamiento de modelos.

Los datos de las muestras, así comos los espectros obtenidos, fueron analizados utilizando el lenguaje Python. La principal ventaja de python es que ofrece una gran cantidad de librerías enfocadas al análisis de datos, el cual va a ser un gran protagonista en este trabajo, así como potentes librerías de Machine Learning. Para nuestro trabajo hemos usado las siguientes librerías:

- **Funciones matemáticas:** NumPy y Math.
- **Análisis de datos:** Pandas.
- **Visualización de datos:** Matplotlib, Seaborn y Plotly.
- **Colorimetría:** Colour.
- **Machine Learning:** Scikit-learn.

Capítulo 5

Resultados.

Ahora que tenemos las muestras y sus espectros de absorción, toca analizar los espectros de absorbancia obtenidos. El objetivo de este análisis será el de entender los picos que aparecen (o no), asociando estos a las características de las muestras. Finalmente, a partir de estos espectros, seleccionaremos las variables que consideremos adecuadas para poder entrenar los modelos.

Este último paso es de gran importancia, ya que seleccionar unas buenas variables mejorará en gran medida el resultado de nuestros modelos. Si los alimentamos con datos que no aporten demasiada información relevante, lo único que conseguiremos es disminuir su rendimiento, aumentar el coste de entrenamiento y que nuestros resultados finales no sean los mejores que podemos obtener.

5.1. Espectros UV - Vis.

Los datos de los espectros se encuentran en formato CSV, con un archivo asociado a cada tipo de vino, por lo que lo primero que se hizo fue cargar los datos y etiquetar cada espectro con el número de su muestra. El formato de origen de los espectros de los vinos tintos venía estructurado de acuerdo a la Tabla 5.1

Tabla 5.1: Formato de los datos.

	0	1	2	3	4	5	6	7	8	9	...
0	250.0	2.2645	1.6707	1.6671	1.620373	1.994519	2.113855	1.987342	1.985722	2.354524	...
1	250.5	2.1884	1.6341	1.6280	1.604110	1.937360	2.055637	1.932661	1.920947	2.287396	...
2	251.0	2.1191	1.5996	1.5925	1.590057	1.886296	1.997585	1.881168	1.865247	2.220316	...
3	251.5	2.0649	1.5675	1.5587	1.576151	1.841649	1.948357	1.835738	1.818526	2.156875	...
4	252.0	2.0119	1.5378	1.5298	1.562423	1.796473	1.900175	1.789820	1.771967	2.098593	...
...	...	...	...	...	...	...	...	...	...	...	...
1697	1098.5	0.0092	0.0078	0.0083	0.008241	0.007559	0.009756	0.007684	0.010508	0.008366	...
1698	1099.0	0.0109	0.0088	0.0094	0.009149	0.009071	0.010840	0.009012	0.011891	0.009735	...
1699	1099.5	0.0081	0.0058	0.0066	0.006506	0.006184	0.008041	0.005793	0.009183	0.006733	...
1700	1100.0	0.0104	0.0076	0.0094	0.008347	0.008613	0.010083	0.007955	0.011234	0.008695	...

Nos encontramos con un Dataframe que tiene 24 columnas, 23 de las muestras de vino (columnas 1-23), y la columna 0, que representa la longitud de onda λ en nm. Las muestras para los espectros de absorción se han tomado desde 250 nm a 1100 nm en intervalos de 0,5 nm. Tomando la columna 0 como nuestro eje X y renombrando cada columna a la corresponidente muestra, podremos representar nuestros datos en gráficas como se puede apreciar en la Figura 5.1 para los vinos tintos, la Figura 5.2 para vinos blancos y Figura 5.3 para vinos rosados.

Repitiendo el mismo procedimiento para los vinos blancos y rosados, tenemos las siguientes gráficas.

Figura 5.1: Espectros de los vinos tintos.

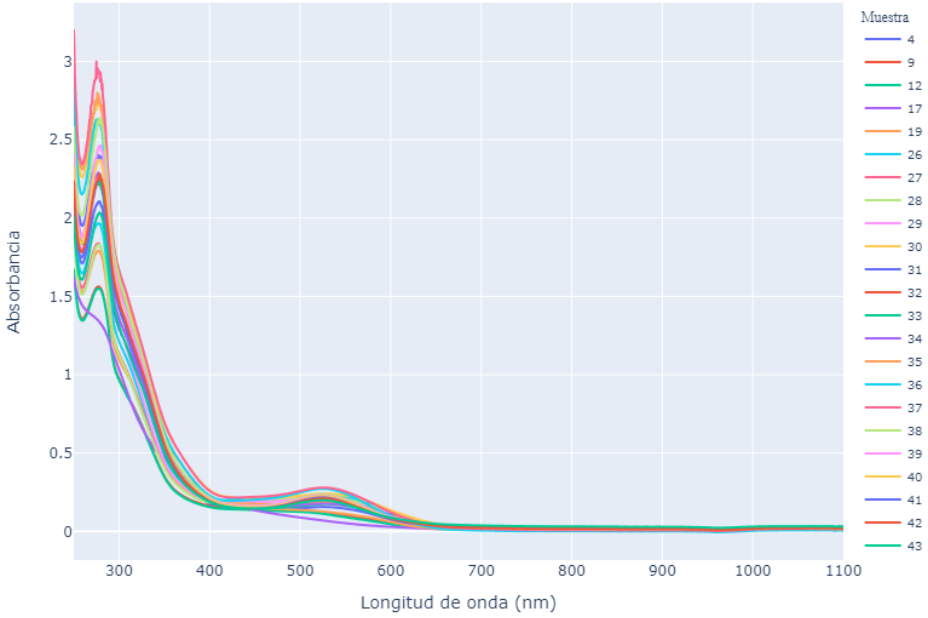


Figura 5.2: Espectros de los vinos blancos.

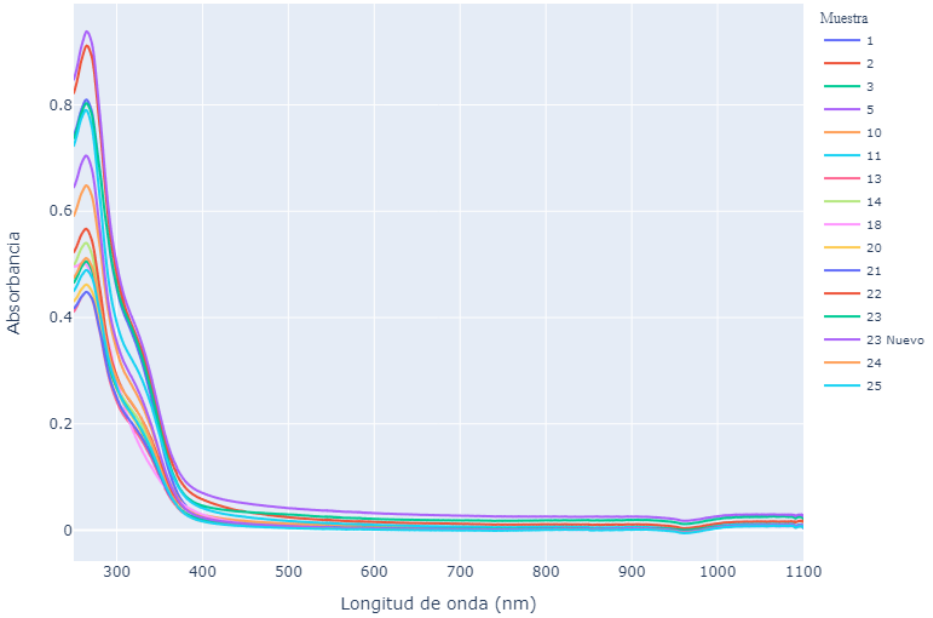
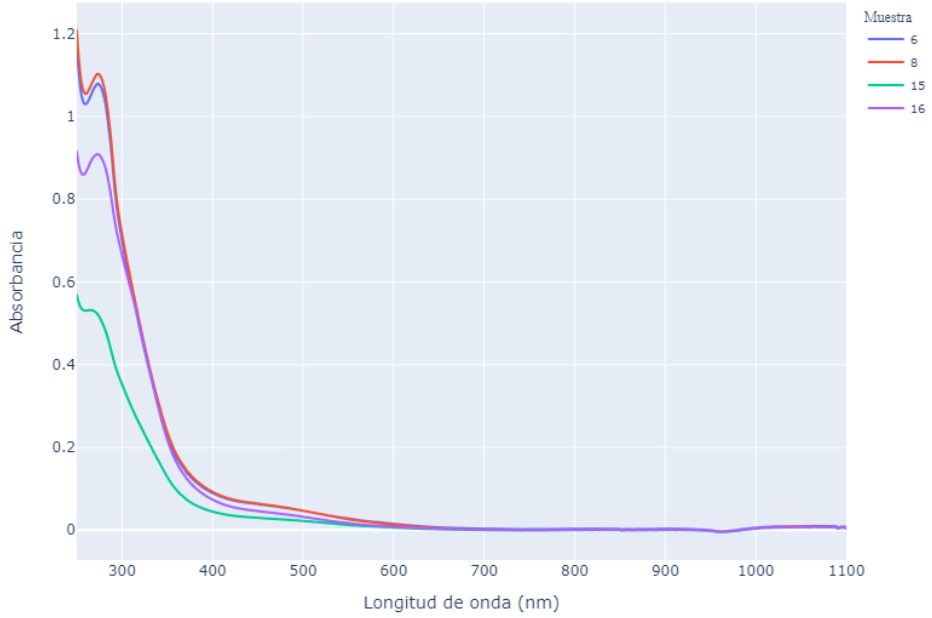


Figura 5.3: Espectros de los vinos rosados.



Para los vinos tintos, se puede apreciar en la Figura 5.1 que casi toda la información se encuentra hasta más o menos los 650 nm. El mayor pico está presente sobre los 280 nm, lo cuál tiene sentido, ya que unos compuestos presentes en el vino, los flavanoles, presentan un pico de absorción en esa longitud de onda. Estos compuestos también presentan un pico alrededor de los 520 nm, lo cual también podemos apreciar en la gráfica, aunque en bastante menor medida. La absorción en la parte azul del espectro se traducirá en que veremos los vinos de un color más rojo, lo cual se puede comprobar en los espectros de los vinos tintos. También se puede apreciar que hay una muestra que, aunque presenta una absorbancia elevada, no presenta un pico como tal. Se corresponde a la Muestra 17. Esta muestra es peculiar, ya que se trata de un vino del año 1992. Estos compuestos disminuyen su concentración a medida que el vino envejece, lo que se traduce en un pico menos pronunciado para esta muestra.

En las gráficas de los vinos blancos, Figura 5.2 y rosados, Figura 5.3, el pico principal ha disminuido bastante, más para los blancos que para los rosados. En el caso de los blancos, este pico se ha desplazado un poco hacia los 270 nm. Para ambos casos, el pico que aparecía sobre los 520 nm ha desaparecido completamente.

Algo que se puede ver también, es que los vinos más viejos muestran un menor número de flavanoles, lo que se puede deducir de un pico de menor intensidad a 280 nm (véase la muestra de vino tinto más vieja, la 17 que hemos mencionado anteriormente).

5.2. Procesado de los espectros obtenidos.

En este apartado nos vamos a encargar de transformar los espectros de los vinos en parámetros con los que alimentar nuestros modelos. Como hemos mencionado anteriormente, seleccionar unos buenos parámetros marcará la diferencia entre unos buenos resultados o unos menos buenos. El proceso que se seguirá para obtener dichos parámetros será el descrito en el Capítulo 3, aplicando las fórmulas necesarias para la transformación. De cara a la obtención de los parámetros CIELab, se ha hecho uso de la librería Colour de Python, que implementa estos cálculos en sus funciones sin necesidad de definir unas nuevas por nuestra parte, y siguiendo la misma lógica descrita en dicho apartado.

Lo primero que vamos a hacer es transformar nuestros espectros de absorbancia en espectros de transmitancia, ya que para obtener los parámetros CIELab tenemos que trabajar con transmitancia. La transformación es bastante simple, la podemos realizar de la siguiente manera:

$$\%T = 10^{2-A} \tag{5.1}$$

Siendo $\%T$ el porcentaje de transmitancia a una determinada longitud de onda λ y A la absorbancia a esa misma λ. Lo que nos quedaría de acuerdo a la Tabla 5.2.

Tabla 5.2: Transmitancias.

	Longitud de onda	4	9	12	17	19	26	27	28	29	...
0	250.0	0.543876	2.134519	2.152286	2.396774	1.012700	0.769387	1.029575	1.033423	0.442055	...
1	250.5	0.648037	2.322202	2.355049	2.488227	1.155154	0.879758	1.167721	1.199646	0.515946	...
2	251.0	0.760151	2.514201	2.555642	2.570058	1.299284	1.005576	1.314716	1.363807	0.602121	...
3	251.5	0.861192	2.707073	2.762485	2.653683	1.439962	1.126271	1.459695	1.518707	0.696827	...
4	252.0	0.972971	2.898678	2.952569	2.738905	1.597817	1.258418	1.622482	1.690569	0.796906	...
...	...	...	...	...	...	...	...	...	...	...	...
1696	1098.0	97.364357	97.926448	97.813770	97.739250	97.937046	97.425580	97.925546	97.200612	97.603637	...
1697	1098.5	97.903902	98.220016	98.107001	98.120330	98.274536	97.778642	98.246254	97.609480	98.092093	...
1698	1099.0	97.521416	97.994116	97.858826	97.915399	97.932987	97.534890	97.946292	97.299140	97.783370	...
1699	1099.5	98.152191	98.673379	98.491783	98.513103	98.586171	98.165526	98.674969	97.907734	98.461625	...
1700	1100.0	97.633757	98.265258	97.858826	98.096384	98.036320	97.705047	98.184967	97.446445	98.017811	...

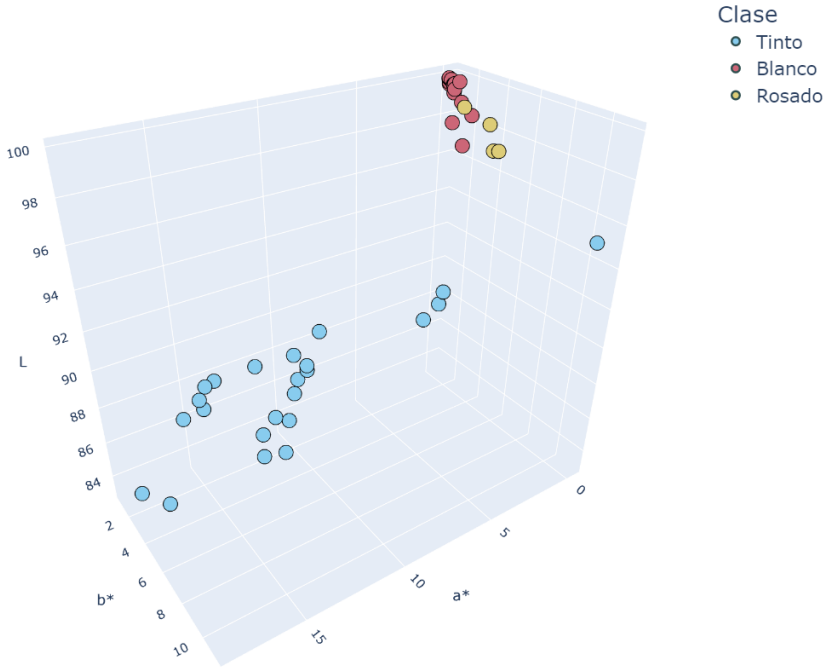
Una vez tenemos las transmitancias, podemos calcular los parámetros triestímulo XYZ asociados a cada muestra para posteriormente sacar los parámetros CIELab. Los parámetros L^*, a^* y b^* obtenidos se muestran en la Tabla 5.3.

Tabla 5.3: Parámetros CIELab para las primeras 5 muestras.

Muestra	L	a	b
4	89.692604	8.696581	2.785636
9	92.430320	5.506093	7.520574
12	92.913566	5.236831	7.455891
17	95.451000	0.392608	11.245866
19	91.565801	5.952165	7.084635

Y, representando en el espacio, obtendremos los siguientes puntos, etiquetados por su tipo y mostrados en la Figura 5.4.

Figura 5.4: Representación 3D de los vinos por clase en el espacio CIELab.



A simple vista se pueden ver agrupaciones bastante fácilmente, pero también tenemos algún caso en el que las muestras no se encuentran tan juntas. Podemos hacer 2 grandes distinciones a simple vista, vinos tintos y blancos junto a rosados. Si analizamos algunos de esos puntos en el espacio, tenemos información bastante interesante. Las muestras que más llaman la atención pertenecen a los tintos. Estas muestras corresponden a las 17, que se encuentra en solitario en la parte derecha; las muestras 9, 12 y 19, que se

encuentran a la izquierda de la 17; y finalmente la 36 y 37, que se encuentran a la izquierda en la parte inferior junto al eje b*.

Una vez analizadas las muestras en el espacio CIELab, procedemos a sacar los parámetros de Glories junto con el índice IPT280. También calcularemos los mencionados croma C^*_{ab} y tono h_{ab}. Unificando todos los parámetros en una misma tabla, obtendremos lo que buscábamos, Tabla 5.4

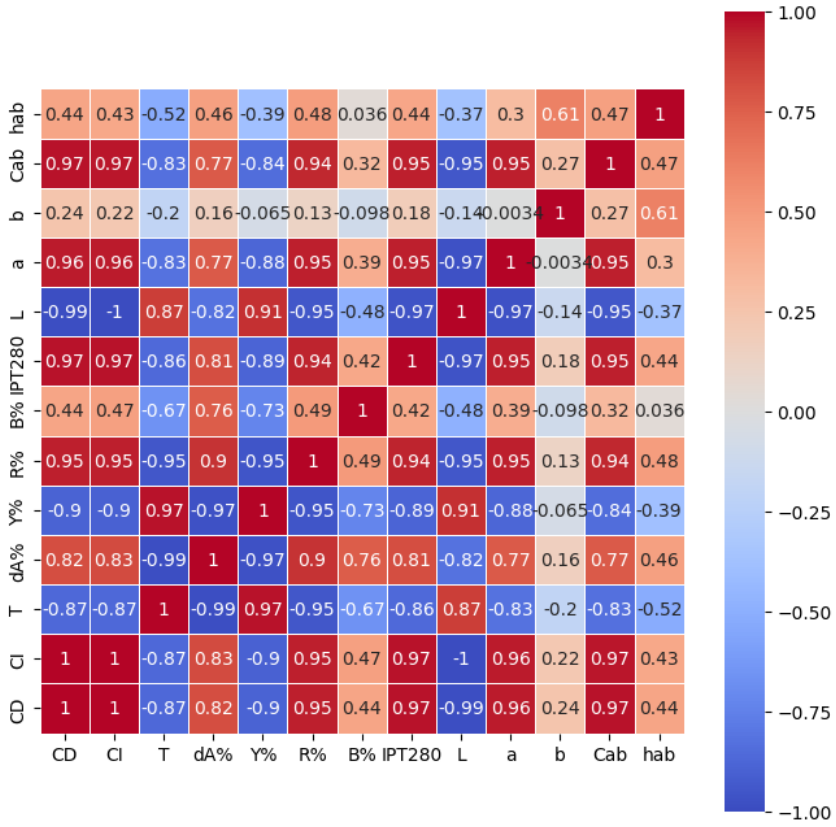
Tabla 5.4: Tabla con todas las variables.

Muestra	CD	CI	T	dA %	Y %	R %	B %	IPT280	L	a	b	Cab	hab
4	0.302100	0.355200	0.945267	69.832582	41.328829	43.721847	14.949324	208.3900	89.692604	8.696581	2.785636	9.131829	0.309988
9	0.272200	0.306200	1.204049	53.562753	48.563031	40.333116	11.103854	154.8500	92.430320	5.506093	7.520574	9.320734	0.938826
12	0.259000	0.289200	1.223176	51.802575	49.273859	40.283541	10.442600	153.8800	92.913566	5.236831	7.455891	9.111241	0.958478
17	0.248197	0.271148	2.379727	-3.360023	64.451886	27.083733	8.464381	132.2866	95.451000	0.392608	11.245866	11.252717	1.535899
19	0.291853	0.329676	1.243729	52.352489	49.071816	39.455405	11.472779	176.8803	91.565801	5.952165	7.084635	9.253125	0.872048

Para analizar estas variables, utilizaremos lo que se conoce como matriz de correlaciones, Tabla 5.5. La matriz de correlaciones se trata de una tabla que muestra las relaciones lineales entre las variables que tenemos, comparando todas a pares. Esto se traduce en que la tabla nos mostrará la pendiente de la recta de cada par de variables, siendo 1 si las variables están muy relacionadas linearmente, y -1 si están inversamente relacionadas.

Cabe destacar que correlación no implica causalidad, es posible que tengamos una correlación alta entre dos variables pero que físicamente una no esté relacionada con la otra. Esto puede deberse a una tercera variable (como por ejemplo en nuestro caso, la edad). En nuestro caso, usaremos esta matriz de correlaciones para identificar qué variables nos aportan una información similar y dejar únicamente la que mejor se adapte a nuestro estudio previo.

Figura 5.5: Matriz de correlaciones.



Como hemos comentado anteriormente, vamos a revisar las correlaciones obtenidas para ver si podemos deshacernos de alguna variable y de esta manera simplificar los datos quitando información poco relevante. Tenemos que tener cuidado a la hora de eliminar basándonos únicamente en sus relaciones lineales ya que, por ejemplo, los parámetros CIELab L^*, a^* y b^* obtienen coeficientes altos entre ellos, cuando en realidad son 3 coordenadas del espacio de color independientes una de otra. La limpieza de variables que vamos a hacer entonces será de variables con correlaciones similares respecto al resto de

variables, es decir, variables que tengan unos coeficientes similares para todas las demás. Una vez decidido esto, podemos ver 2 claros ejemplos en la matriz:

- **CD y CI:** La densidad de color y la intensidad de color se comportan de forma muy similar. Esto tiene sentido, ya que en la propia definición la única diferencia es que a la intensidad de color se le agrega un término adicional, así que nos quedaremos con este último dato.
- **T - %Y - L:** En este caso tenemos el tinte, la proporción de color amarillo, la coordenada L del espacio de color. Si nos basamos únicamente en los resultados de la matriz, nos vamos a quedar con L, ya que es la que tiene una correlación más baja con el resto de variables en general.

Con lo que finalmente nos quedaríamos con 9 de las 13 columnas iniciales.

5.3. Estandarizado y Análisis de componentes principales (PCA).

Una vez hemos elegido nuestras variables, vamos a aplicar estandarizado y PCA para que nuestros modelos entrenen con datos de mejor calidad.

5.3.1. Estandarizado.

A veces, tenemos el caso de que algunas variables tienen unos valores muy altos respecto a otras variables del mismo set de datos que manejamos. Esto provoca que estas columnas con números grandes "pesen" más que el resto. Aquí entra en juego la estandarización de los datos.

Estandarizar es una transformación reversible que consiste en restar a cada dato la media de su columna y, posteriormente, dividir el mismo valor por la desviación típica de la misma columna. Lo que conseguimos con esto es que todos los datos tengan media cero y desviación unitaria, es decir, que estén centrados en el 0 y alrededores (varianza unitaria).

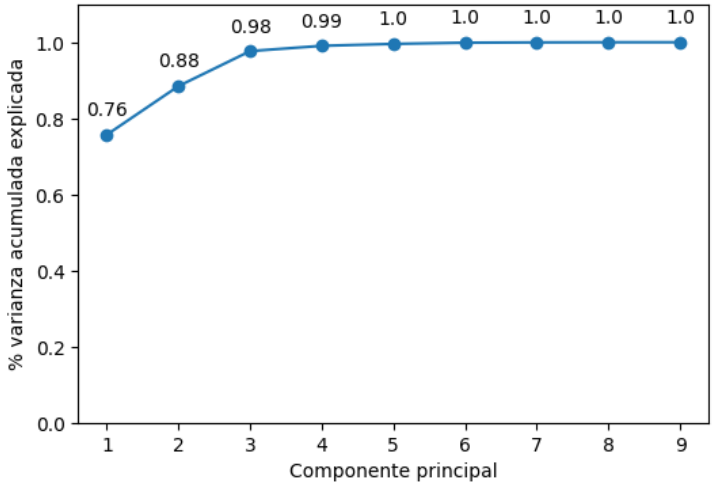
La principal ventaja de todo esto es que ahora podremos "comparar" entre las distintas columnas, teniendo todos los datos un peso similar para los algoritmos. De cara a los distintos modelos, realmente sólo nos haría falta estandarizar si vamos a usar algún algoritmo basado en distancias, por lo que no es un proceso necesario. Pero si decidimos hacerlo, no va a estropear nada, ya que no afectará al modelo. El estandarizado lo realizaremos aplicando la transformación previamente descrita a nuestros datos utilizando Python, ya que hay funciones diseñadas para esta función sin necesidad de nosotros definir una nueva. La función utilizada pertenece a la librería Scikit-learn y es la siguiente:

```
SandardScaler()
```

5.3.2. Análisis de componentes principales (PCA).

De cara a reducir dimensiones, como vimos anteriormente, una de las técnicas más utilizadas es el análisis de componentes principales. PCA identifica direcciones cuya varianza es mayor, por lo que es importante que nuestros datos estén estandarizados, como hicimos en el apartado anterior. Si tenemos el cuenta el porcentaje de información perdida respecto a los datos originales, vamos a analizar la varianza acumulada después de calcular las componentes principales, representada en la Figura 5.6. Calcularemos un número de componentes igual al de nuestras variables, y desde ahí tomaremos una decisión.

Figura 5.6: Varianza acumulada.



Como podemos ver, con 3 variables podemos explicar un 98 % de la varianza. Siendo este valor mayor a 90 %, podemos considerar las 3 primeras variables para nuestro modelo. Tomando 2 variables tendríamos un 88 %, lo cual es un valor bastante aceptable pero no tan bueno como el otro, por lo que nos decantamos por 3. Una vez decidido que vamos a usar 3 componentes principales, proyectamos nuestros parámetros sobre este espacio para obtener los vectores de la Tabla 5.5

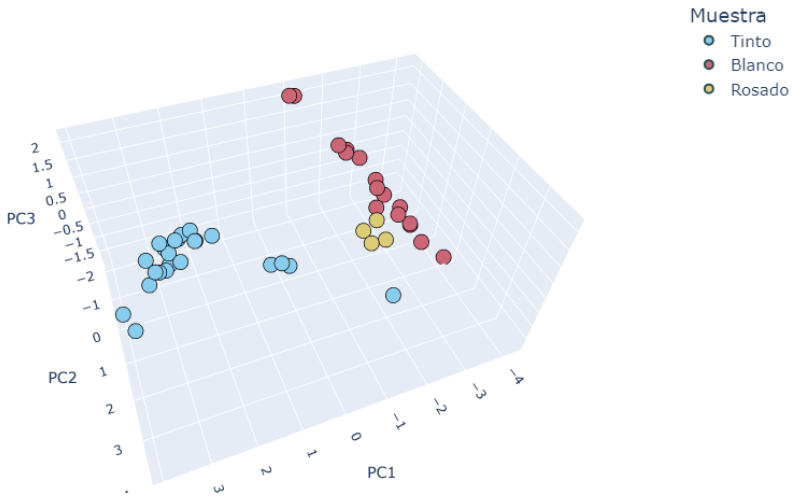
Tabla 5.5: 3 componentes principales.

	CI	dA %	R %	B %	IPT280	L	a	b	Cab
PC1	0.377471	0.344128	0.375667	0.210082	0.372190	-0.376947	0.367067	0.066459	0.366146
PC2	0.088293	-0.182287	-0.014502	-0.557648	0.080898	-0.017215	-0.028717	0.773760	0.203528
PC3	-0.057876	0.358238	-0.056781	0.623085	-0.116246	0.109045	-0.301220	0.581706	-0.149459

De cara a explicar cómo obtenemos las componentes principales, es una simple combinación lineal de nuestras antiguas variables. Por poner un ejemplo de la Tabla 5.5, usemos la primera componente:
$PC1 = 0{,}377471CI + 0{,}344128dA\,\% + 0{,}375667R\,\% + 0{,}210082B\,\% + 0{,}372190IPT280 - 0{,}376947L + 0{,}367067a^* - 0{,}066459b^* + 0{,}366146C^*_{ab}$

Finalmente sólo nos queda proyectar nuestros datos sobre estas componentes para tener los datos deseados. Al tener 3 componentes, podremos representar en el espacio nuestros datos, como se puede ver en la figura 5.7.

Figura 5.7: Representación en el espacio de componentes principales.



A diferencia de la Figura 5.4, en este caso los vinos blancos se encuentran más alejados entre sí, y los rosados se han separado un poco también. Esto es debido a que las componentes principales buscan que exista la mayor varianza posible al proyectar nuestros datos sobre los nuevos vectores. En concreto, la variable que presenta mayor varianza, como hemos visto en la Figura 5.6, es la primera componente principal PC1 con un 76 %.

5.4. Modelos y resultados.

Ahora que tenemos ya nuestros datos procesados y preparados, podemos proceder a entrenar nuestros modelos. Como no tenemos una etiqueta clara para realizar nuestras predicciones, vamos a utilizar modelos de aprendizaje no supervisado de clusterización. Al aplicar estos modelos, los algoritmos lo que harán será agrupar nuestros datos en diferentes clústeres. Una vez tengamos nuestros datos agrupados, podremos analizar el comportamiento para ver las características que ha tenido en cuenta a partir de los datos que les hemos proporcionado. Los algoritmos elegidos para nuestro trabajo serán el K-Means, DBSCAN y clustering jerárquico aglomerativo.

Una cosa que tenemos que tener en cuenta es el número de muestras que tenemos. De cara a este trabajo, nos va a servir para comprobar si para este grupo obtenemos resultados que tengan sentido y así poder seguir con el estudio.

5.4.1. K-Means.

El primer algoritmo que vamos a utilizar es el K-Means, que ya hemos explicado en el fundamento teórico. Para comenzar a trabajar con este algoritmo, tenemos que tener en cuenta que hay que definir de antemano el número de clústeres que queremos que aparezcan en el resultado final. Hacer esto a ciegas puede resultar en una gran pérdida de tiempo, ya que tendríamos que entrenar un modelo para cada número de clústeres final y analizar los resultados a posteriori. Afortunadamente, existen dos formas de obtener el número de clústeres óptimo (o por lo menos, una buena aproximación): observando las inercias y estudiando el Silhouette score.

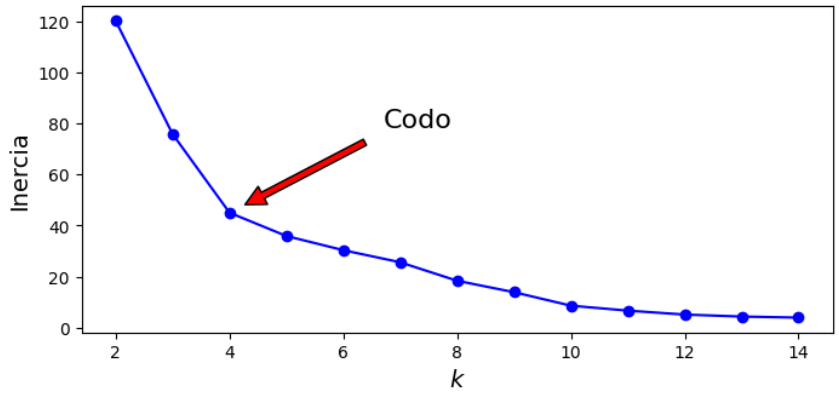
La inercia consiste en la suma de los cuadrados de las distancias de cada punto x a su centroide asignado C:

$$I = \sum_{k=1}^{K} \sum_{i=1}^{n_k} ||x_i - C_k||^2 \tag{5.2}$$

Si calculamos varias inercias y las representamos, obtendremos una línea curva descendiente. La inercia tiende a disminuir a medida que aumentamos el número de clústeres. Entonces, para encontrar el número

óptimo, lo que deberemos buscar es el punto en el que la pendiente cambie bastante, lo que se conoce como codo. En la Figura 5.8 podemos apreciar esto mismo, siendo k el número de clústeres para el que calculamos la inercia..

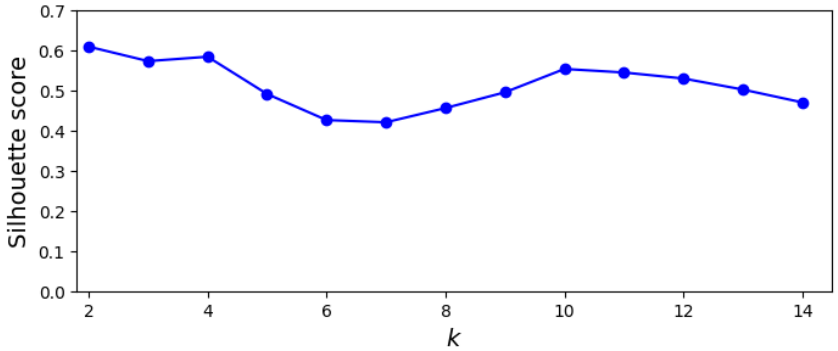
Figura 5.8: Codo en la gráfica de inercias.



Se puede apreciar que nuestro codo estaría principalmente en el punto correspondiente a 4 clústeres, así que vamos a analizar los resultados para este número de clústeres. Si nos fijamos bien, en k=10 también tenemos un codo menos pronunciado. Tener 10 clústeres con el número de muestras que tenemos no es algo ideal pero igualmente vamos a analizarlo y ver qué conclusiones podemos sacar. Además de las inercias, podemos ver los Silhouette scores y así observar otro método para aproximar el número de clústeres óptimo.

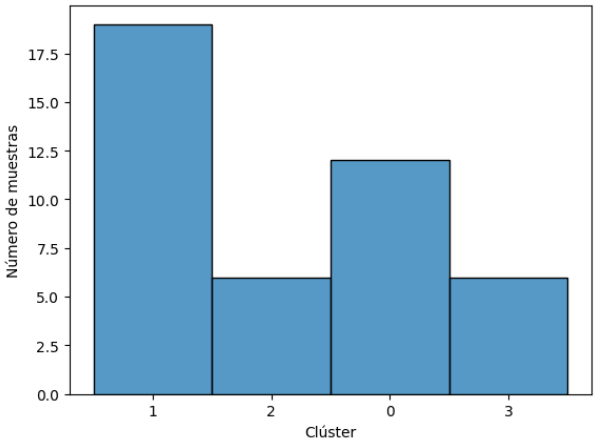
El silhouette score calcula la media de $\frac{b-a}{max(a,b)}$, donde a es la distancia media de un punto a los puntos asociados a su mismo clúster, y b es la distancia media a los puntos del clúster más próximo. Este valor va de -1 a +1, indicando los valores positivos que el punto está bien asignado dentro del clúster, y los valores negativos que el punto puede haber sido mal asignado. Nos interesa buscar los valores más altos para este caso, los cuales están representados en la Figura 5.9.

Figura 5.9: Gráfica de los Silhouette scores.



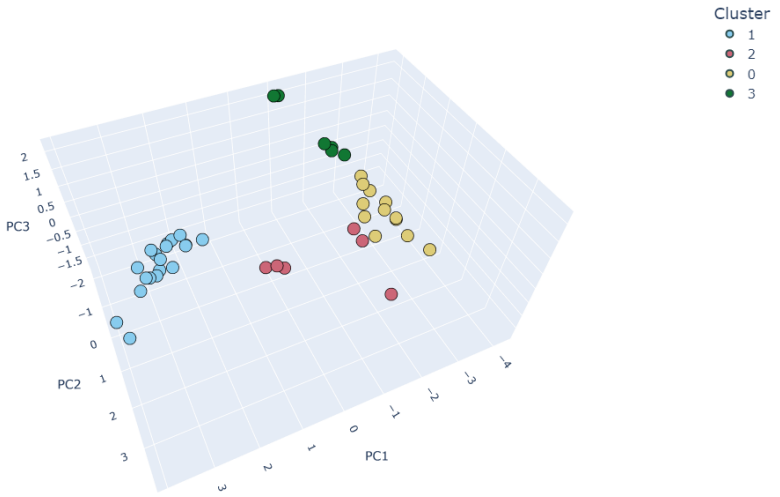
Viendo la gráfica, podemos apreciar que el valor para k=4 es bastante alto, pero el de k=10 también, por lo que tiene sentido seguir con ambos. Procedemos a entrenar nuestro modelo comenzando con k=4. Lo primero que mostraremos es el número de muestras asociadas a cada clúster de los 4 que le hemos indicado usar, representado en la figura.

Figura 5.10: Distribución de muestras en los 4 clústeres.



Podemos observar en la Figura 5.10 que para el clúster 1 tenemos 19 muestras, para el clúster 2 hay 6 muestras, en el clúster 0 tenemos 12 muestras y finalmente, en el clúster 3 hay 6 muestras. Ahora podemos hacer una representación en el espacio de componentes principales y ver cómo se han agrupado las muestras, analizando similitudes y diferencias entre clústeres.

Figura 5.11: Representación en el espacio de los 4 clústeres.



Como podemos ver en la Figura 5.11, el clúster 1 está muy bien definido pero los demás parece que tienen muestras que, a simple vista, parecería que pertenecen a otras agrupaciones. Vamos a analizar las muestras de cada clúster para ver qué información podemos sacar.

- **Clúster 0:** Muestras 1, 13 - 16, 18, 20 - 25. Se tratan todos de vinos blancos salvo las muestras 15 y 16, que son rosados. La muestra 15 se trata de un vino rosado macerado de 2008 y la 16, un vino rosado comercial de 2016 y abierta en 2018.
- **Clúster 1:** Muestras 4, 26 - 43. Todas estas muestras se tratan de vinos tintos de los años 2022 y 2023, sin hacer ninguna distinción por uva, tipo de riego, tipo de fermentación, etc.
- **Clúster 2:** Muestras 6, 8, 9, 12, 17 y 19. Las muestras 6 y 8 son ambas la misma muestra de vino rosado pero tomadas con 2 semanas de diferencia. Estas se han agrupado con las muestras 9, 12, 17 y 19, que son las muestras de vino tinto de mayor edad. En el caso de las muestras 9 y 12, se trata del mismo caso que con los rosados, misma muestra tomada con 2 semanas de diferencia. La muestra 17 es la más antigua de todas, siendo del año 1992 y la 19, un vino tinto de 2014.

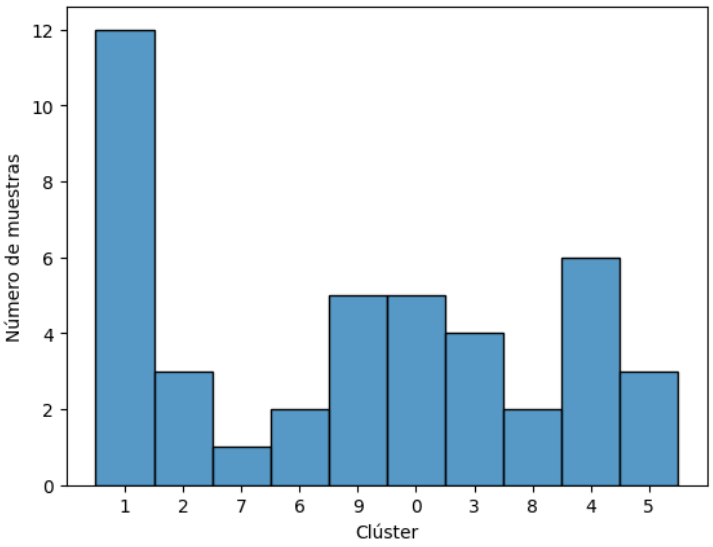
- **Clúster 3:** Muestras 2, 3, 5, 10, 11 y 23. En este clúster son todos vinos blancos, muestras comprendidas entre los años 2016 y 2018. Las muestras 2 y 5 son del mismo vino pero tomadas con 2 semanas de diferencia.

Evaluando las muestras de los clústeres, podemos sacar algunas conclusiones. Lo primero que vemos es una separación entre vinos blancos y vinos tintos. Los rosados, en cambio, se encuentran agrupados junto a los otros dos pero en diferentes clústeres. En el clúster 1 tenemos los vinos tintos de años similares (2022 y 2023) comparten características similares, por eso aparecen agrupados juntos. A medida que envejecen, el pico característico del espectro disminuye y se asemeja más al de los vinos rosados o blancos. En el clúster 2 encontramos la muestra 17, y es la que más se diferencia de todas las demás, mientras que las muestras 9, 12 y 19, incluidad en el mismo clúster, no siendo tan viejas pero tampoco tan jóvenes como el resto de tintos, se encuentran más cerca del clúster 1. Las dos muestras de vino rosado que pertenecen al clúster 2 aparecen muy cerca del clúster 0, en el que estaban los otros 2 vinos rosados junto a los vinos blancos. El algoritmo los ha agrupado aquí seguramente por culpa nuestra de no tratar la muestra 17 como un outlier, haciendo que el centroide de ese clúster quede más cerca de estos dos puntos.

Si analizamos ahora los clústeres de vinos blancos, 0 y 3, encontraremos alguna característica interesante. Fijándonos en el clúster 0, vemos que están la mayoría de muestras de vinos blancos elaborados con el mismo tipo de uva verdejo (salvo la 23), junto a la muestra 1, 13 y 14. Estas tres se tomaron nada más abrir la botella. La muestra 15 es un vino rosado macerado, lo que puede indicar que este método afecta al color del vino asemejándolo al de uno blanco, y la muestra 16, un vino rosado también pero de 2016 y abierto en 2018. Con respecto al clúster 3, encontramos aquí las muestras 2, 23 y todas las muestras de vino blanco que se tomaron a las 2 semanas de abrir la botella, que serían las muestras 3, 5, 10, y 11. Esta es la principal diferencia que encontramos con el otro clúster que contiene vinos blancos, el clúster 0, lo que parece indicar que para los vinos blancos afecta al color de forma apreciable el que hayan sido abiertas y conservadas.

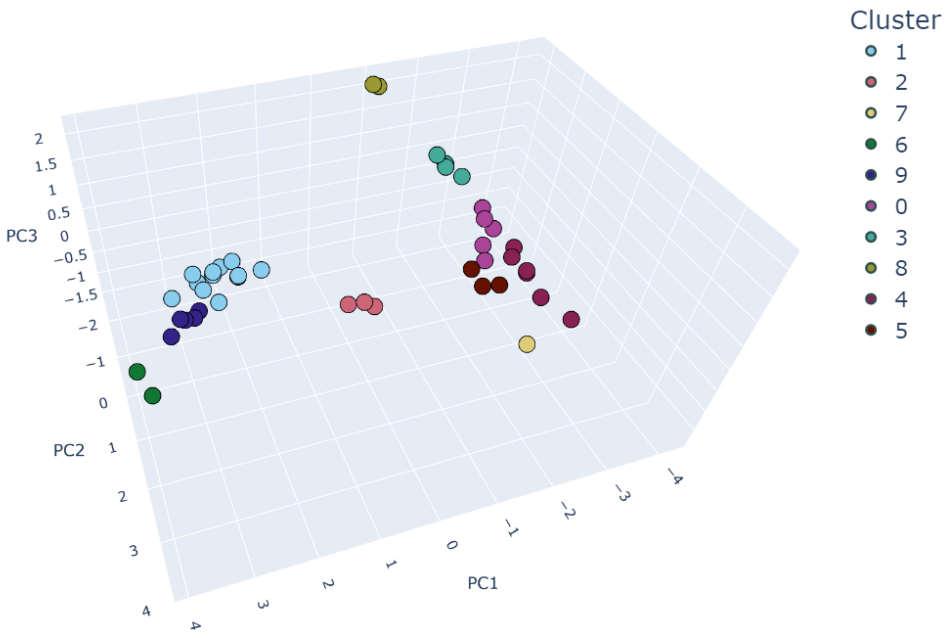
Vamos a ver el resultado para 10 clústeres.

Figura 5.12: Distribución de muestras en los 10 clústeres.



Apreciamos en la Figura 5.12 que tenemos un número de muestras importante asociadas a un único clúster, el clúster 1 con 12 registros. El clúster que menos muestras tiene, con 1, es el clúster 7. Después tenemos los clústeres 6 y 8, con 2 muestras cada uno. Los clústeres 2 y 5 con 3 muestras, el clúster 3 con 4 muestras, los clústeres 9 y 0 con 5 muestras, y finalmente el clúster 6 con 6 muestras. Podemos representar esto en el espacio y así también analizar qué muestras han sido agrupadas.

Figura 5.13: Representación en el espacio de los 10 clústeres.



De esta la Figura 5.13 podremos sacar la siguiente información:

- **Clúster 0:** Muestras 1, 13, 14, 15 y 23 Nuevo. En el caso de la muestra 15, se trata de un rosado, el resto son blancos. Los vinos 13 y 14 se elaboraron similarmente pero son de distintos años.
- **Clúster 1:** Muestras 4, 26-35 y 43. Las muestras 26-29 se elaboraron con fermentación espontánea. Las muestras 30-34 son del mismo viñedo pero con diferentes riegos. La muestra 4 tiene unas características bastante similares a las de estos vinos, todos ellos de 2022. La muestra 35 fue alterada. La única muestra más diferente es la 43, siendo de 2023 y recién acabada la fermentación.
- **Clúster 2:** Muestras 9, 12 y 19. Las muestras 9 y 12 son del mismo vino tinto de 2018, solo que tomadas con 2 semanas de diferencia, mientras que la 19 se trata de un vino de 2014. Ambos están clasificados como Tintos G 2 y 1 respectivamente.
- **Clúster 3:** Muestras 2, 10, 11 y 23. Las muestras 10 y 11 son bastante similares, siendo ambas Blanco R1D1D2 pero de los años 2018 y 2017 respectivamente. La muestra 2 se parece en características, Blanco R1D2 pero de 2016. La muestra 23 difiere un poco en características, Blanco R1T1T2, pero es del año 2017.
- **Clúster 4:** Muestras 20, 21, 22, 24 y 25. Todos estos vinos se elaboraron con la misma uva verdejo, pero con distintos tratamientos de riego.
- **Clúster 5:** Muestras 8, 6 y 16. Estos son todos vinos rosados.
- **Clúster 6:** Muestras 36 y 37. Alteraciones de vinos tintos.
- **Clúster 7:** Muestra 17. Se trata de un vino tinto de 1992. En la gráfica de los espectros ya vimos que no presentaba un pico como el resto, por lo que es entendible que sea la única muestra peculiar.
- **Clúster 8:** Muestras 3 y 5. Se tratan de vinos blancos, muestras del mismo vino tomadas con 2 semanas de separación.
- **Clúster 9:** Muestras 38-42. Todas estas muestras se tratan de tintos recién fermentados. Todos del mismo año.

Como vemos, el algoritmo ha agrupado los vinos de una manera que tiene bastante sentido, ha sido capaz de encontrar bastantes de las clasificaciones que teníamos originalmente en el archivo sin tener nada de esta información. Aumentando el número de clústeres, conseguimos que las agrupaciones sean de menor número de muestras, pero con el número de estas que manejamos no tiene demasiado sentido tener un número tan grande. Aún así lo hemos hecho para analizar el comportamiento de los algoritmos y ver si es posible que encuentren patrones o características comunes sin haberlas indicado. Por ejemplo,

en el clúster 1 que es el más abundante, se encuentran agrupados vinos tintos de 2022 procedentes del mismo viñedo pero usando un tratamiento de riego diferente, por lo que parece que esto no haya afectado demasiado a la hora analizar las muestras. También se encuentran en este clúster vinos elaborados con fermentación espontánea del mismo año, lo que nos indica que este método no hace que el vino sea demasiado diferente a los que no se ha aplicado el mismo. La muestra 43, se elaboró con el mismo tipo de uva que la muestra 34, solo que un año más tarde. Finalmente, en este clúster hay una muestra de vino tinto de 2022 alterada con una sal de potasio, por lo que parece que esta alteración no afecta demasiado a las características que hemos tenido en cuenta en este análisis.

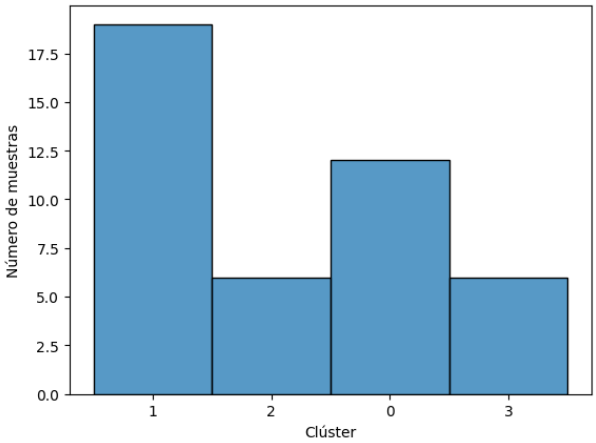
Si continuamos con el clúster 0, aquí encontramos varias muestras de vino blanco y una de rosado, la 15. Como hemos visto en las figuras de los espectros, con el paso del tiempo el pico característico a 280 nm disminuye, por lo que tiene sentido que este vino, siendo de 2008, pueda estar en este clúster. También tendríamos que analizar todas sus características, ya que por lo que vemos se trata de un vino macerado. El resto de vinos blancos, fueron abiertos para tomar una muestra y posteriormente almacenados para tomar otra muestra 2 semanas después, las cuales se encuentran en otro clúster.

En el clúster 2 tenemos vinos tintos más viejos que la mayoría, las muestras 9 y 12 de 2018 y tomadas con 2 semanas de diferencia, y la muestra 19 de 2014. De nuevo confirmando que **la edad afecta al color**. En el clúster 3 tenemos las muestras 10 y 11, que corresponden con las muestras tomadas 2 semanas más tarde de las 13 y 14 que encontrábamos en el clúster 0. Las muestras 20 a 22, 24 y 25 se encuentran en el mismo clúster, el 4. Todos estos vinos blancos se elaboraron con el mismo tipo de uva verdejo. Para el clúster 5 tenemos los otros 3 vinos rosados. Las muestras 6 y 8 son el mismo vino comercial pero tomados con 2 semanas de diferencia. La muestra 16 también se trata de una muestra de vino comercial, del año 2016 y abierto en 2018, lo que podría explicar que se le agrupe con un vino de 2006. El clúster 6 comprende las muestras alteradas de vino tinto, 36 y 37. Estas fueron alteradas con ácido tartárico y ácido cítrico respectivamente. A diferencia de la muestra 35 encontrada en el clúster 1, aquí si que ha afectado al color. La muestra 17 la encontramos en el clúster 7, la que podríamos considerar un outlier dada su edad. No la hemos eliminado por la cantidad de muestras que manejamos y así también analizamos cómo afecta al entrenamiento.

En el clúster 8 están presentes las muestras 3 y 5 de vino blanco. Estas muestras son tomadas dos semanas después de las muestras 1 y 2 respectivamente. Dado que estos vinos son similares al resto, sin ninguna característica destacable, es extraño que estas muestras aparezcan tan distantes al resto, por lo que tendríamos que conocer con más detalle todos los datos de su elaboración para poder sacar más información. De momento, lo que sabemos es que el estar 2 semanas conservados después de abiertos ha afectado a estos vinos de forma diferente al resto. Por último, en el clúster 9 encontramos todos los vinos tintos de 2023 de la vendimia de ese año y recién acabada la fermentación, a diferencia del clúster 1, en el que todos eran del año 2022 salvo 1.

Ahora vamos a ver como se hubiera comportado si únicamente hubiésemos estandarizado los datos, sin aplicar PCA. Como el procedimiento para buscar un número óptimo de clústeres es igual que en el caso anterior, podemos omitir el proceso en este caso, ya que se encuentra realizado en el Notebook. Tendríamos también dos picos, para k=4 y k=10. Para comparar, usaremos k=4, ya que sólo queremos ver el comportamiento de nuestras muestras reduciendo las dimensiones. En la Figura 5.14 podemos ver la distribución.

Figura 5.14: Distribución de las muestras sin aplicar PCA.



Si analizamos las muestras en concreto:

- **Clúster 0:** Muestras 1, 13 - 16, 18, 20 - 25.
- **Clúster 1:** Muestras 4, 26 - 43.
- **Clúster 2:** Muestras 6, 8, 9, 12, 17 y 19.
- **Clúster 3:** Muestras 2, 3, 5, 10, 11 y 23.

Se trata de la misma agrupación que teníamos en el caso anterior aplicando PCA para 4 clústeres. En este caso, al no poder representar en el espacio los datos, vamos a realizar aquí todo el análisis. En este caso no podremos representar las muestras en el espacio, ya que tenemos más de 3 parámetros. Como conlusión de este ejercicio, podemos sacar que el aplical PCA nos ha beneficiado, ya que no hemos perdido nada de información, a su vez reduciendo el número de parámetros. Para nuestro modelo, esto puede no significar una gran diferencia en rendimiento, pero si se tratase de un set de datos de miles o millones de registros, hubiese sido extremadamente beneficioso en lo que a rendimiento y coste se refiere.

Para acabar con el K-Means, concluiremos diciendo que este algoritmo ha hecho un buen trabajo a la hora de clasificar nuestras muestras, por lo que es un candidato a seguir en un futuro si se amplía la cantidad de muestras a utilizar para poder tener unos modelos mejores.

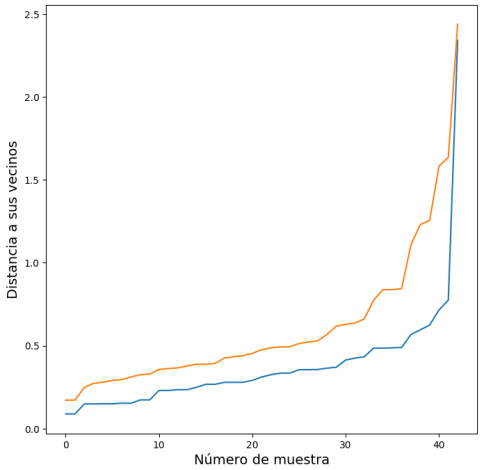
5.4.2. DBSCAN.

Ahora vamos a realizar los mismos modelos utilizando el algoritmo de DBSCAN. Para poder usar este modelo, tenemos que indicarle dos parámetros: ε, que nos indica la distancia máxima que puede haber entre dos registros, y min_samples, que indica el número de muestras que habrá en una región ε. ε nos indica si un punto se encuentra dentro de una región o es considerado un outlier.

Como para el K-Means, tendremos que buscar los valores óptimos para ε y para min_samples. Para esto vamos a usar el algoritmo KNN (K Nearest Neighbors), con el que podremos sacar un silhouette score. Esto lo haremos ya que representando las distancias de los puntos a sus vecinos, podremos sacar un "codo" para ver qué ε serían demasiado pequeñas y cuáles demasiado grandes. Estos dos valores que estamos buscando se conocen como hiperparámetros, y se utilizan para controlar el modelo que queremos entrenar externamente. En nuestro caso, encontraremos los mejores buscando la combinación de ambos que maximice el Silhouette score.

Para generar la Figura 5.15, hemos tomado las distancias a los 2 primeros vecinos, lo cual queda de la siguiente manera.

Figura 5.15: Distancia de cada punto a sus 2 primeros vecinos.



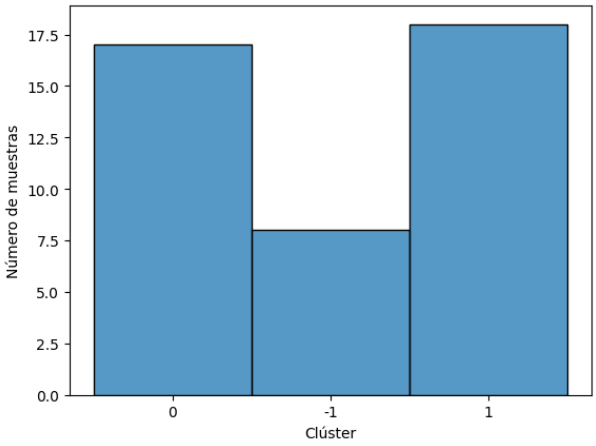
Una vez analizada esta gráfica, vemos que tenemos un codo a una distancia aproximada de entre 0.4 y 1, así que vamos a buscar unos hiperparámetros cerca de esos valores. Finalmente obtenemos los valores de la Tabla 5.6 para los mejores hiperparámetros, que se elegirán usando el Silhouette score como métrica.

Tabla 5.6: Tabla con los mejores hiperparámetros.

Muestras	Eps	Score
4	1.1	0.510932
3	0.9	0.505002
3	1.0	0.505002
3	1.1	0.505002
5	1.1	0.486264

Analizando la Tabla 5.6 podemos ver que los mejores hiperparámetros los obtenemos para min_samples $= 4$ y $\varepsilon = 1.1$. Vamos a entrenar al modelo y vemos como se comporta, realizando un análisis de la distribución de muestras asociadas a cada clúster como se puede ver en la Figura 5.16.

Figura 5.16: Distribución de las muestras por clúster para DBSCAN.



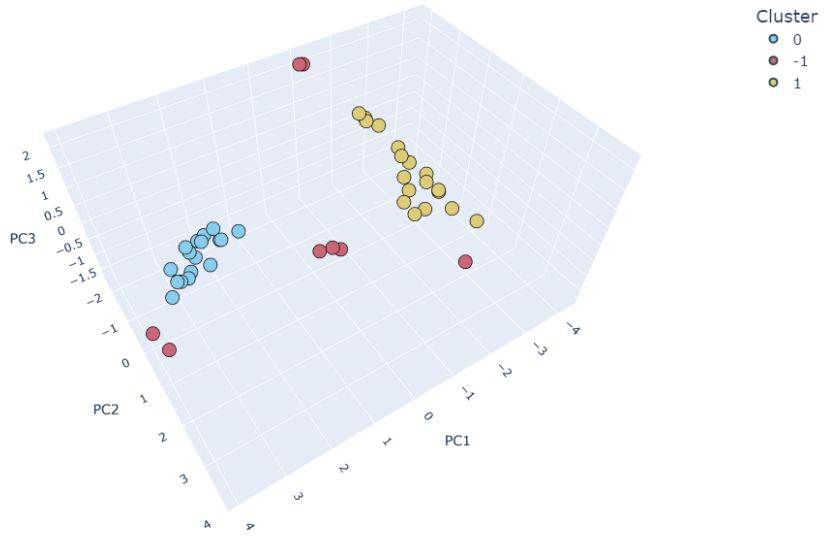
Como habíamos mencionado en el fundamento teórico, a este algoritmo no hay que indicarle el número de clústeres a generar de primeras, sino que eso lo hace solo. Lo primero que notamos a simple vista vemos que tenemos 3 clústeres (2 en realidad, ya que el clúster -1 corresponde a las muestras que no han sido clasificadas en ninguna agrupación, ruido), que se distribuyen de la siguiente manera:

- **Clúster 0:** 17 muestras.
- **Clúster 1:** 18 muestras.

- **Clúster -1:** 8 muestras.

Si representamos esto en el espacio, obtenemos la Figura 5.17 siguiente gráfica.

Figura 5.17: Gráfica en 3D de los clústeres generados por DBSCAN.



Podemos ver que se han creado 2 clústeres principales, diferenciando los tintos en uno, y los blancos + rosados en otro. El cluster -1 corresponde a "outliers", datos que no han sido categorizados por el algoritmo en uno de los dos clústeres principales. Si analizamos los outliers tendremos las siguientes muestras. Muestras 36 y 37 de vinos tintos alterados con ácido tartárico y cítrico respectivamente. Muestras 9, 12, 17 y 19 de vinos tintos, que corresponden con las de mayor edad. Muestras 3 y 5 de vino blanco, las cuales ya hemos visto que fueron tomadas con 2 semanas de diferencia de sus respectivas muestras. Este tiempo que pasaron en conserva parece que afectó a estas muestras de manera significativa, ya que para el resto de vinos blancos en las mismas condiciones no ha habido una diferencia tan significativa. Viendo este resultado, no vamos a analizar el resto de muestras en detalle, ya que nos aporta menos información que la que tenemos en un principio. Incluso si no tuviéramos datos de los vinos, podríamos diferenciarlos por color, lo que nos generaría 3 agrupaciones, además de una para outliers.

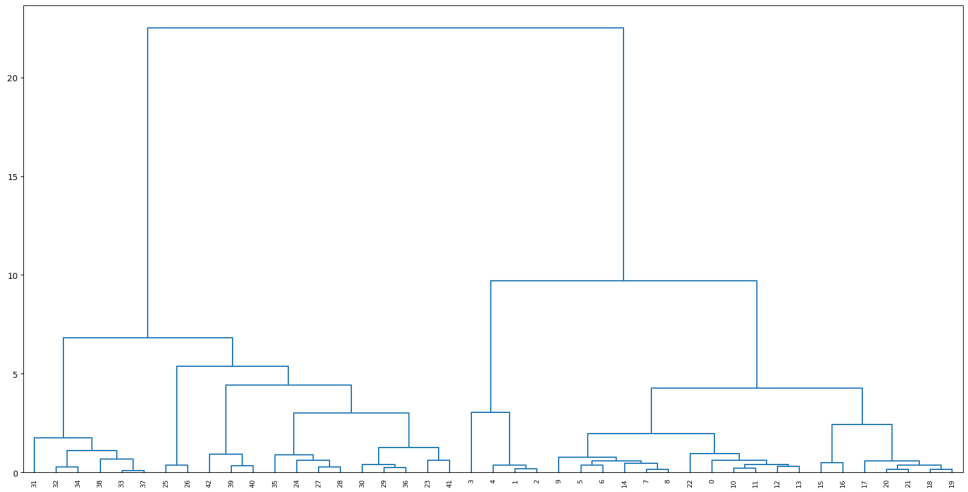
Si realizamos el mismo ejercicio para los datos sin aplicar PCA, obtenemos un resultado bastante similar, por lo que no vamos a presentarlo aquí. Todos los cálculos y las gráficas se pueden encontrar en el Jupyter Notebook junto al resto del trabajo.

El principal motivo por el que este algoritmo no ha trabajado muy bien, es porque busca clústeres basados en densidad de puntos. En nuestro caso esta densidad no se alcanza del todo, generando bastantes outliers, y los que se encuentran juntos en teoría, realmente no lo están tanto. De ahí que se genere una división entre vinos tintos por un lado y blancos y rosados por otro.

5.4.3. Clústering jerárquico aglomerativo.

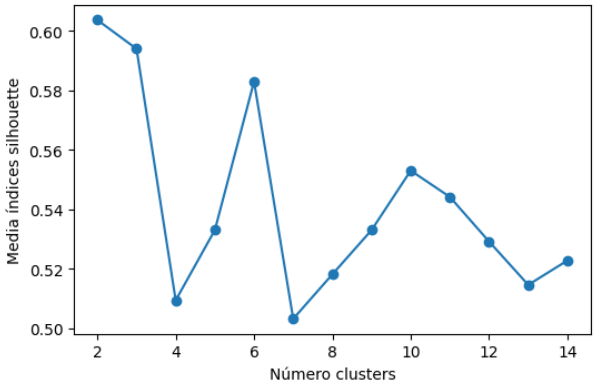
Por último, vamos a aplicar un modelo de clustering jerárquico aglomerativo para ver el comportamiento con los datos que tenemos. Entre los parámetros a tener en cuenta se encuentran: n_clusters, que determina el número de clústers que se van a generar, y distance_threshold que será la distancia (altura del dendrograma) a partir de la cual se dejan de unir los clústeres. Para empezar, usaremos n_clusters=None y distance_threshold=0 para dejar que crezca completamente, así lo visualizaremos entero. Si entrenamos el modelo con los datos proyectados sobre las componentes principales y visualizamos, obtenemos el dendrograma al completo representado en la Figura 5.18

Figura 5.18: Dendograma generado por el clústering jerárquico aglomerativo.



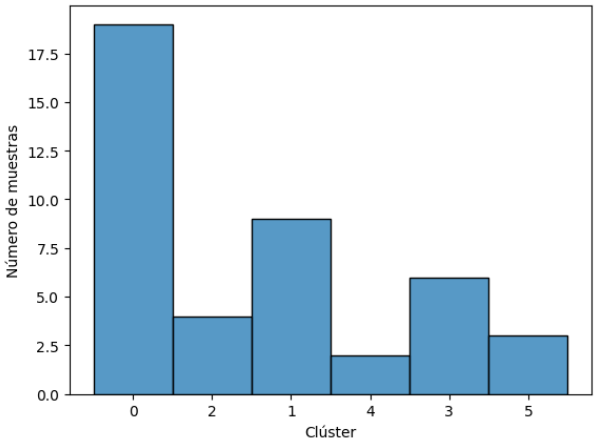
Podemos apreciar como al ir creciendo de abajo a arriba se van generando clústeres agrupando pares de muestras o clústeres. Una forma de elegir el número de clústeres que queremos es visualmente a partir del dendrograma, elegir un umbral y pasarle ese valor al modelo. Por ejemplo, si quisiéramos 4 clústeres, el umbral estraría aproximadamente en una altura de 6 unidades. Si en vez de eso queremos utilizar un método basado en métricas, podemos utilizar de nuevo el Silhouette Score para que nos indique el número óptimo. Para encontrar este número, representamos los datos para obtener la Figura 5.19 y ahí seleccionar.

Figura 5.19: Silhouette scores para el clústering jerárquico.



En este caso tenemos 2 picos claramente marcados, para 6 clústeres y, en menor medida, para 10. Podemos apreciar que de nuevo tenemos 10 clústeres como valor óptimo, al igual que en el K-Means para el mismo caso del PCA. Vamos a analizar primero el caso de los 6 clústeres comenzando por la distribución de muestras, Figura 5.20.

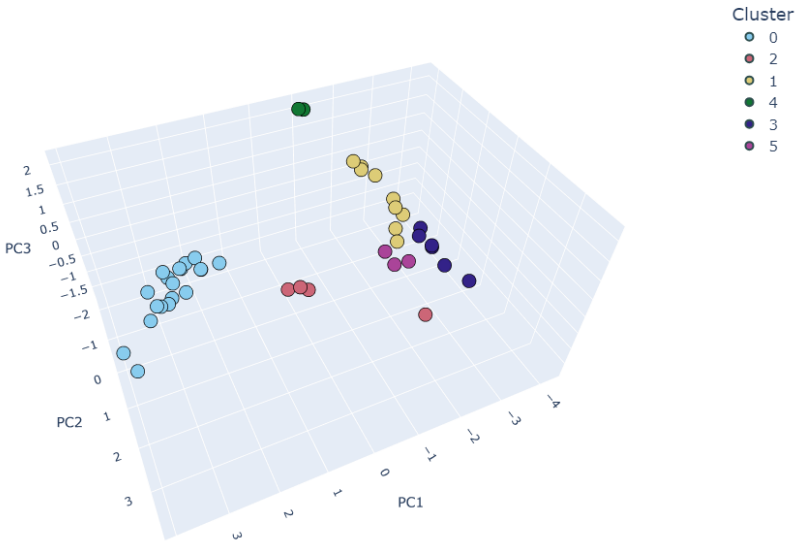
Figura 5.20: Distribución de las muestras en los clústeres.



Tenemos la siguiente distribución, que se puede analizar en la Figura 5.21.

- **Clúster 0:** 19 muestras.
- **Clúster 1:** 9 muestras.
- **Clúster 2:** 4 muestras.
- **Clúster 3:** 6 muestras.
- **Clúster 4:** 2 muestras.
- **Clúster 5:** M3 muestras.

Figura 5.21: Representación 3D de los clústeres generados por el algoritmo.



- **Clúster 0:** Muestras 4, 26-43. Se tratan de todas las muestras de vino tinto salvo 4, las de los vinos con mayor edad. Una vez más, tenemos que los vinos de edad similar se suelen agrupar. Esta vez también se han incluido las muestras alteradas, que en otros casos estaban agrupadas en un clúster propio.
- **Clúster 1:** Muestras 1, 2, 10, 11, 13, 14, 15, 23 y 23 Nuevo. La muestra 15 se trata de un vino rosado, en concreto, el vino rosado macerado, lo cual puede indicar que este método afecta al color

del vino. Las muestras 10-11 y 13-14 se tratan de muestras del mismo vino pero tomadas con dos semanas de diferencia, lo que explicaría su agrupamiento. El resto de muestras, 1, 2, 23 y 23 Nuevo se tratan de vinos blancos diferentes. Adicionalmente, estos dos últimos fueron elaborados con el mismo tipo de uva verdejo.

- **Clúster 2:** Muestras 9, 12, 17 y 19. Se trata de los vinos tintos con mayor edad, los que no estaban agrupados junto al resto de tintos. Lo que indica que la edad afecta bastante al color de este tipo de vinos.
- **Clúster 3:** Muestras 20, 21, 22, 24 y 25. Estos son todos vinos blancos elaborados con el mismo tipo de uva verdejo, como en el caso de las muestras 23 y 23 Nuevo. Son de diferentes años, pero parece que afecta más el tipo de uva que la edad en este caso.
- **Clúster 4:** Muestras 3 y 5. Estas muestras son del mismo tipo de vino blanco que las muestras 1 y 2, pero tomadas con 2 semanas de diferencia. Parece que a estos vinos les ha afectado en gran medida este intervalo de tiempo, ya que para el resto de muestras suelen agruparse juntas si tienen estas características.
- **Clúster 5:** Muestras 6, 8 y 16. Son los vinos rosados no macerados. Esta agrupación puede indicarnos que los vinos macerados van a ver alterado su color, en caso de este vino, apróximandolo más a los vinos blancos.

En la comparativa de estos clústeres vamos a encontrar clústeres de vinos tintos, rosados y blancos. Las agrupaciones de vinos tintos corresponden a los clústeres 0 y 2, los cuales han sido diferenciados por edad. El clúster 2 agrupa las muestras de vino tinto con mayor edad: 9, 12, 17 y 19, mientras que el 0 incluye a todas las demás. De nuevo, vemos que la edad afecta significativamente al color del vino tinto. En el clúster 5 encontramos 3 de los 4 vinos rosados, las muestras 6, 8 y 16, mientras que la muestra 15 se encuentra agrupada junto a los vinos blancos. Esta muestra es la única muestra macerada, lo que parece indicar que esta característica hace que su color se asemeje más al de los vinos blancos. Por último, para los vinos blancos vamos a tener 3 clústeres, los número 1, 3 y 4. El clúster con mayor diferencia al resto es el 4, que incluye a las muestras 3 y 5. Como hemos visto anteriormente, a estas muestras les afectó de manera notable el haber sido tomadas 2 semanas después de abrir la botella y conservarla. Los clústeres 1 y 3 se encuentran más próximos. A diferencia del clúster 1, todos los vinos del clúster 3 se elaboraron con el mismo tipo de uva verdejo. En el clúster 1 tenemos los vinos las muestras 1 y 3, que corresponden al mismo vino de las muestras del clúster 4 pero tomadas nada más abrir la botella. También encontramos los pares de vinos 10-13 y 11-14, que tienen la misma característica que los vinos 1-3 y 2-5, pero en este caso perteneciendo todos al mismo clúster. Esto nos indica que las muestras 1-3 y 2-5 tienen que tener alguna característica especial que desconocemos a partir de la información que tenemos inicialmente.

Si comparamos estos resultados a los obtenidos para K-Means con 4 clústeres, vemos que las dos nuevas agrupaciones que hemos obtenido se tratan de las de vinos rosados y las dos muestras de vinos blancos de comportamiento diferente al resto. Por lo demás, los vinos han sido agrupados de forma similar, los vinos tintos de 2022 y 2023 por un lado y los de mayor edad por otro, y presenta la separación de vinos blancos que hemos visto en todos los casos. También mencionar que hemos ganado información sobre los vinos rosados, cosa que perdíamos en los 4 clústeres del K-Means.

Hasta ahora, las agrupaciones de este algoritmo parece que muestran los resultados con mayor sentido. 4 clústeres puede que se nos quedara corto, incluso no clasificaba muy bien algunos datos, mientras que 10 clústeres era un número demasiado elevado teniendo en cuenta la cantidad de muestras que se manejan. Es verdad que respecto a 10, se pierde algo de información, como la separación de vinos tintos alterados o por edad más cercana, pero para una primera aproximación muestra que se puede obtener bastante información de los espectros de absorción de los vinos. Para comparar resultados, también se realizó el entrenamiento para 10 clústeres, obteniendo unos resultados prácticamente iguales a los del K-Means, por lo que no los mostraremos en esta memoria debido a la cantidad de espacio limitada, pero se pueden encontrar en el Jupyter Notebook complementario a esta memoria si se quieren consultar.

Como primeras conclusiones, podemos decir principalmente que uno de los factores que más afecta al color es la edad, sobre todo si se trata de vinos tintos. Dependiendo de la alteración, también tenemos que el color varía, como hemos podido comprobar en el clúster que suele aparecer para muestras 36 y 37. Para los vinos blancos, podemos decir que tomar muestras con días de diferencia desde la apertura de la botella hace que el color varíe más significativamente que en los vinos tintos o rosados, haciendo que aparezcan agrupados en diferentes clústeres, y que el tipo de uva con el que se elaboran también tiene

un peso mayor que en los tintos, viendo como en la mayoría de casos está presente esta diferencia en los clústeres creados. Por último, en el caso de vinos rosados, hemos visto como el macerado se encontraba más próximo a los vinos blancos, siendo agrupado con estos en todos los casos, por lo que podríamos tener una primera idea de que este método de elaboración afecta al color. Con una sola muestra, la número 17 en este caso, es difícil afirmarlo, pero se puede tener en cuenta a la hora de ampliar el número de muestras.

Para acabar con este apartado, hemos realizado el mismo análisis para los datos sin aplicar el PCA. El número de clústeres que obtuvimos fue de 5 y 10. En el caso de 5 clústeres, la única diferencia es que las muestras de vinos rosados que teníamos en la figura 5.21 correspondiente al clúster 5 se veían incluidas junto a los vinos blancos, por lo que perdíamos esa información. Para 10 clústeres, el resultado era exactamente el mismo, por lo que se podrá encontrar en el Notebook por el mismo motivo que mencionábamos previamente, la falta de espacio, ya que estaríamos repitiendo el análisis de la distribución.

Capítulo 6

Conclusiones.

Para finalizar este trabajo, vamos a analizar las conclusiones que hemos ido obteniendo a lo largo del mismo, así como comentar la posible línea de trabajo en el futuro.

En primer lugar, destacar que a pesar del número de muestras reducido, se han obtenido unos resultados interesantes para los modelos de aprendizaje no supervisado. En concreto, los modelos de K-Means y Clústering Jerárquico Aglomerativo han sido los que han dado resultados más remarcables. Si analizamos las agrupaciones que han realizado estos modelos, podemos sacar resultados interesantes. Empezando por los vinos tintos, **la principal característica por la que los algoritmos separaban las agrupaciones de estos vinos era la edad**. En la mayoría de casos veíamos como casi siempre las muestras más longevas de tintos se encontraban en su propio clúster. Para el caso de K-Means con 10 clústeres y clústering jerárquico con 6 y 10 clústeres también existía una diferencia entre las dos muestras de vino tinto alteradas con ácido tartárico y cítrico, lo que nos indica que esto afecta al color, mientras que la alteración con una sal de potasio no era separada. Si entramos en los casos de 10 clústeres, obtuvimos el mismo resultado en las agrupaciones para ambos sorprendentemente. En el caso de los tintos en concreto, de nuevo teníamos nuevos clústeres en los que las diferencias eran la edad, llegando a diferenciar vinos tintos de los años 2022 y 2023.

Dejando de lado las diferencias y fijándonos en las similitudes, **los vinos tintos del mismo año (para vinos jóvenes) y de edad similar (para vinos tintos de mayor edad) eran agrupados juntos en todos los casos.** Cuando aumentábamos el número de clústeres, también podíamos apreciar que de distinguía entre vinos de 2022 y 2023. También encontrábamos que **los vinos que compartían tipo de uva se agrupaban**, siendo esta una característica común en tintos y blancos, como comentaremos a continuación. Para finalizar con los tintos, los que fueron elaborados con fermentación espontánea también fueron agrupados juntos y en el mismo clúster que los de su año, lo que parece indicar que este método de elaboración no afecta demasiado al color.

Continuando con los vinos blancos, aquí encontramos dos principales resultados. El primero, que **el tomar las muestras con 2 semanas de diferencia habiendo sido conservadas afecta al color** en mayor o menor medida. Este caso se puede apreciar de forma notable en el caso de las muestras 3 y 5, apareciendo repetidamente en un único clúster separadas del resto o como outliers. El resto de muestras con estas características, aparecían agrupadas con las muestras tomadas recién abiertas las botellas o con otros vinos blancos, pero nunca por separado de estos. Como caso general, podemos decir que los vinos blancos que han sido abiertos y conservados lo verán reflejado en su color. Por otro lado y, como veníamos anticipando con los vinos tintos, **los vinos blancos elaborados con el mismo tipo de uva son agrupados juntos**, lo que nos puede ayudar en el futuro si tenemos una base de datos grande con esta información.

Acabando con las muestras de vino, llegamos a los rosados. Lamentablemente, no disponíamos de una cantidad de muestras equiparable a los otros dos tipos de vinos, pero hemos podido ver algún resultado interesante. Por ejemplo, en **el caso del vino rosado macerado, este era agrupado junto a los vinos blancos en todos los casos**, incluso cuando el resto de vinos rosados eran agrupados en un único clúster, lo que nos puede indicar que este método de elaboración afecta al color. De nuevo, con el limitado número de muestras, puede que no sea suficiente para realizar una afirmación, pero nos puede indicar una posible característica a tener en cuenta en el futuro si disponemos de muestras de otros vinos macerados.

Por otro lado, el algoritmo de DBSCAN ha sido el que peores resultados nos ha dado comparando con el resto. Esto se debe a que no son las condiciones ideales para aplicar este algoritmo, ya que su comportamiento busca agrupaciones basadas en densidad de registros. El mejor caso en el que este algoritmo es muy eficiente, es cuando los puntos tienen formas irregulares pero con gran densidad. Los otros algoritmos se basan en distancias, y, por como estaban distribuidas nuestras muestras, han hecho que tengan un mejor desempeño. Lo que sí podemos destacar a su favor es que ha identificado los outliers más característicos, como serían las muestras de vino tinto alteradas, las muestras de vino tinto de mayor edad o las dos muestras de vino blanco que al tomarlas dos semanas después habían variado su color notablemente.

Otra cosa que cabe destacar es que dos algoritmos distintos hayan realizado una clasificación igual para el mismo número de clústeres, K-Means y clústering jerárquico aglomerativo con k=10. Aunque esto haya sido interesante de analizar, por el número de muestras que manejamos, tal cantidad de clústeres no es algo óptimo, por eso hemos analizado los casos de 4 clústeres con K-Means y 6 con clústering jerárquico. Si profundizamos en el análisis con K-Means, **sin aplicar PCA, hemos obtenido un resultado igual, sin perder información**. Esto probablemente se ha debido a que el número de muestras que manejamos no es muy grande, pero aún así hemos visto cómo podemos hacer que el entrenamiento y los modelos sean más eficientes sin gran pérdida de información de nuestros datos. Como anticipábamos, a los algoritmos que se basan en distancias les ayuda este procesamiento previo. También cabe mencionar que nos ha ayudado a mejorar las tareas de visualización, permitiendo proyectar nuestros datos sobre un espacio de 3 componentes principales.

Mencionando los casos de K-Means con 4 clústeres y clústering jerárquico con 6 clústeres podemos comentar los resultados, de los cuales **hemos obtenido resultados bastantes similares** como hemos visto, a pesar de tener 2 clústeres de diferencia. En el caso de los tintos hemos obtenido la misma clasificación, separados los más jóvenes de los más viejos. Uno de los clústeres nuevos era uno para vinos rosados, aunque una de estas muestras, la 15 (vino macerado) seguía estando agrupado con los blancos. El otro clúster que aparecía era el de las dos muestras de vino blanco (3 y 5) que fueron tomadas a las 2 semanas de abrir las botellas, como hemos visto en el análisis hecho.

Para concluir las observaciones del trabajo hecho hasta ahora, cabe destacar la gran cantidad de información que se puede obtener del color del vino, también sumando a ello que la técnica utilizada de espectroscopía UV-Vis ha demostrado ser versátil, precisa, rápida y bastante eficiente en cuanto a costes se refiere, por lo que es una buena opción de cara a futuro.

De cara a trabajo futuro, lo principal sería conseguir una base de muestras suficientemente grande y amplia en variedad para ser capaces de entrenar modelos más precisos que aporten gran cantidad de información. También sería interesante incluir variables como el tipo de uva utilizada, el tratamiento de la cosecha, el riego que se ha utilizado, los tiempos exactos que ha pasado en barril y/o en botella, etc., ya que cuanta más información tengamos, más precisos serán nuestros modelos. Una cosa con la que tendremos que tener cuidado es de no generar overfitting, pero con un procesamiento de datos adecuado, se puede generar una potente herramienta que sea capaz de ayudarnos a tomar mejores decisiones en las diferentes etapas de creación y embotellado del vino, algo en lo que hay mucho interés dada la región en la que nos encontramos. Como hemos visto, los algoritmos han sido capaces de agrupar por características como el tipo de uva, tipo de riego utilizado o el método de elaboración, aunque de nuevo, se necesitaría una base de datos considerable. También sería conveniente utilizar más modelos para poder comparar y conseguir entrenar uno con un rendimiento mayor.

Otro punto que habría que tener en cuenta es el tema del tratamiento de outliers. En nuestro caso, teníamos la muestra 17, que se trataba de un vino tinto de 1992, alejándose considerablemente de la edad del resto. En este trabajo hemos decidido no eliminarla por dos motivos. El primero, que se trata de un primer acercamiento para estudiar la aplicación del aprendizaje automático a las medidas de los espectros obtenidos por espectroscopía UV-Vis y comprobar su viabilidad. El segundo, que el número de muestras era limitado, por lo que de cara a entrenar los modelos, hemos decidido conservarla.

El machine learning es una ciencia que ha ganado mucha popularidad en los últimos años. Esta disciplina ha crecido gracias a internet, a que es fácil de obtener recursos para aprenderla y a la gran demanda que hay de profesionales en el área. Como último comentario, este estudio ha mostrado que es posible entrenar un modelo que sea capaz de identificar patrones en muestras de vino únicamente a través de sus espectros de absorbancia, mostrando el potencial del aprendizaje automático y los posibles usos que se le pueden dar en la industria como en este caso, tratándose de un trabajo multidisciplinar.

Bibliografía

[1] Jancis Robinson and Julia Harding. *The Oxford companion to wine*. American Chemical Society, 2015.

[2] Qué es un vino joven, crianza, reserva y gran reserva. `https://www.bodegasriojanas.com/que-es-un-vino-joven-crianza-reserva-y-gran-reserva/`. Accessed: 2024-09-16.

[3] OIV Master and OIV Patronage. Compendium of international methods of wine and must analysis. *International Organisation of Vine and Wine*, 2024.

[4] C. N. Banwell and Elaine M. McCash. *Fundamentals of Molecular Spectroscopy*. McGraw-Hill, 1983.

[5] Rocío Ríos-Reina, Silvina Mariela Azcarate, J Camiña, and Raquel M Callejón. Assessment of uv–visible spectroscopy as a useful tool for determining grape-must caramel in high-quality wine and balsamic vinegars. *Food chemistry*, 323:126792, 2020.

[6] María Luz García Carretero et al. Análisis del contenido polifenólico en vinos tintos mediante lengua electrónica, ftir y uv-vis. 2017. TFG de la Universidad de Valladolid.

[7] Jose Luis Aleixandre-Tudo and Wessel Du Toit. The role of uv-visible spectroscopy for phenolic compounds quantification in winemaking. *Frontiers and new trends in the science of fermented food and beverages*, pages 200–204, 2018.

[8] Juan García Cazorla, María Xirau Vayreda, and Robert Azorín Romero. *Técnicas usuales de análisis en enología*. Panreac Química, 2005.

[9] María J Martelo-Vidal and Manuel Vázquez. Classification of red wines from controlled designation of origin by ultraviolet-visible and near-infrared spectral analysis. *Ciência e técnica vitivinícola*, 29(1):35–43, 2014.

[10] William Ross McCluney. *Introduction to radiometry and photometry*. Artech House, 2014.

[11] Janos Schanda. Cie colorimetry. *Colorimetry: Understanding the CIE system*, 3:25–78, 2007.

[12] Chris Wyman, Peter-Pike Sloan, and Peter Shirley. Simple analytic approximations to the cie xyz color matching functions. *J. Comput. Graph. Tech*, 2(2):11, 2013.

[13] Whitepoint. `https://es.mathworks.com/help/images/ref/whitepoint.html`. Accessed: 2024-09-16.

[14] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. .º'Reilly Media, Inc.", 2022.

[15] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt publishing ltd, 2019.

[16] What is k-nearest neighbours? `https://spotintelligence.com/2023/08/22/k-nearest-neighbours/`. Accessed: 2024-09-16.

[17] T. Fuertes. Hierarchical clustering, using it to invest. `https://quantdare.com/hierarchical-clustering/`, 2016. Accessed: 2024-09-16.

[18] Bibby Scientific. *6850 Spectrophotometer Operating Manual*. Bibby Scientific, 2012.