UNIVERSIDAD DE TARAPACÁ
Universidad del Estado

Trabajo Final minería de datos

Nombre de Asignatura: Minería de datos

Nombre del estudiante: Diego Taucare

Nombre del Profesor: Atsuko Galaz

Fecha de entrega: 5 de julio de 2023

Introducción

El conjunto de datos Wine es un conjunto de datos clásicos y ampliamente utilizado en el aprendizaje automático. Fue introducido por Forina en 1988 y consta de mediciones químicas de vinos cultivados en la misma región en Italia, pero de tres variedades.

El objetivo principal de este conjunto de datos es clasificar los vinos en una de las tres clases, en base a sus propiedades químicas medidas.

Cada instancia esta etiquetada con una clase que indica la variedad del vino al que pertenece. El conjunto de datos Wine se utiliza ampliamente en tareas de clasificación y ha sido objeto de numerosos estudios y aplicaciones en la comunidad del aprendizaje automático.

La naturaleza multiclase de este conjunto de datos y las características relacionadas con las propiedades químicas de los vinos lo convierten en un desafío interesante para aplicar técnicas de aprendizaje supervisado y explorar suficientes modelos de algoritmo y clasificación

Este se encuentra fácilmente disponible en diversas plataformas de aprendizaje automático, lo que hace accesible y utilizable para proyectos de clasificación.

Finalmente ofrece una oportunidad valiosa para explorar y aplicar técnicas de clasificación en el contexto de la clasificación de vinos según sus características químicas su disponibilidad y su relevancia en el campo de aprendizaje automático lo convierten en una opción atractiva para la investigación y experimentación de este campo

Selección de dataset

El conjunto de datos esta compuesta por esta compuesto por 178 instancias, donde cada instancia consta de 13 características que describen las propiedades químicas de los vinos. Estas características incluyen el contenido de alcohol, acido málico, cenizas, alcalinidad de ceniza, magnesio, fenoles totales, flavonoides, fenoles no flavonoides, proantocianidinas, intensidad de color, matiz, OD280/OD315 de vinos, diluidos y prolinas

Wine

Donated on 6/30/1991

Using chemical analysis determine the origin of wines

Dataset Characteristics	Subject Area	Associated Tasks
Multivariate	Physical	Classification
Attribute Type	**# Instances**	**# Attributes**
Integer, Real	178	13

Alcohol: contenido de alcohol en el vino (en %)

Ácido málico: cantidad de ácido málico presente en el vino (en g/l)

Cenizas: contenido de cenizas en el vino (en g/l)

Alcalinidad de la ceniza: es la medida de la alcalinidad de las cenizas presentes en el vino (en mEq/l)

Magnesio: cantidad de magnesio presente en el vino (en mg/l)

Fenoles totales: concentración total de fenoles en el vino (mg/l)

Flavonoides: cantidad de flavonoides presente en el vino (mg/l)

Fenoles no flavonoides: concentración de fenoles no flavonoides en el vino(mg/l)

Proantocianidinas: cantidad de proantocianidinas presentes en el vino

Intensidad de color: intensidad de color del vino medida óptimamente

Matiz: matiz de color del vino (0=rojo; 1=purpura; 2= amarillo)

OD280/OD315 de vinos diluidos: relación de la absorbancia óptica entre 280 nm y 315 para vinos diluidos

Prolina: cantidad de prolina presente en el vino (en mg/l)

Cada instancia del conjunto de datos también está asociada con una variable de salida denominada “clase”, que indica la variedad del vino. Las clases posibles son los siguientes:

Clase 0: Vinos de la variedad ‘Sepa tamarugal’

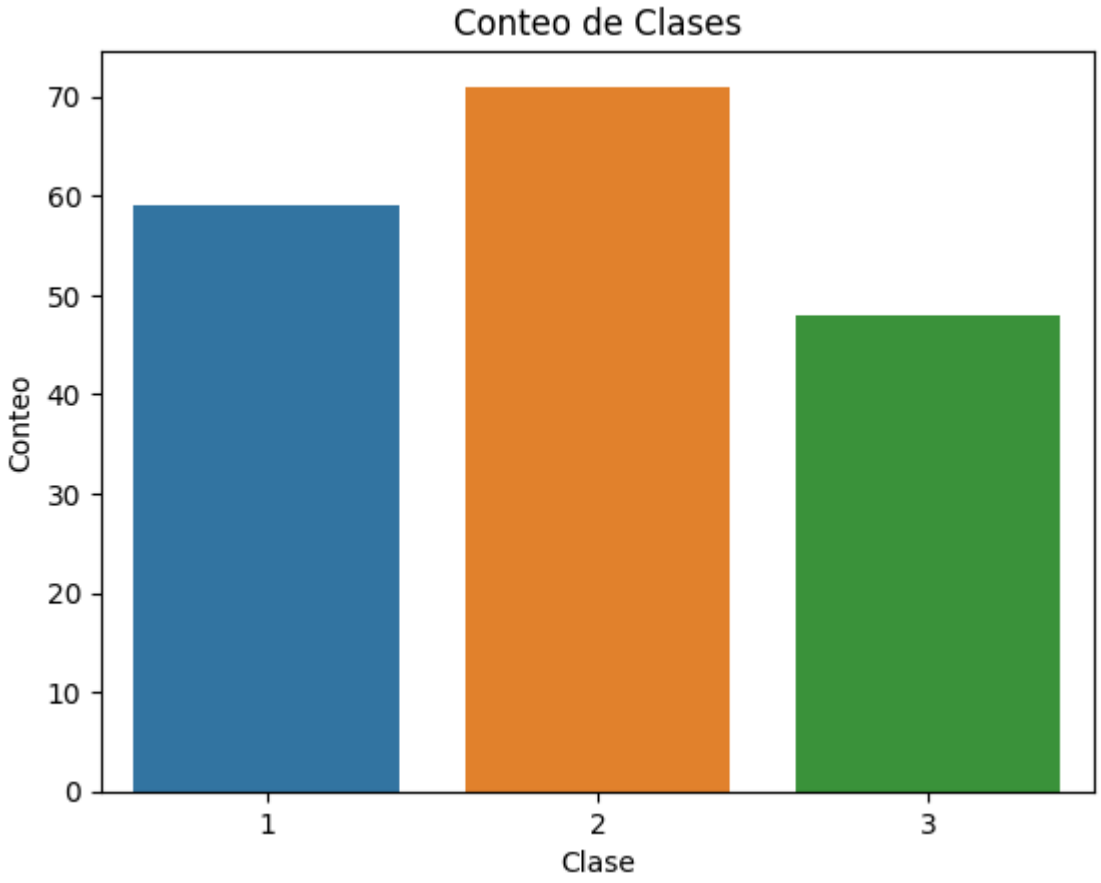
Clase 1: Vinos de la variedad ‘Gros Colman’

Clase 2 : vinos de variedad ‘ditisvinifelas’

Pre-procesamiento

Conteo de Clases

```
import matplotlib.pyplot as plt
wine
# Gráfico de conteo de la variable 'Clase'
sns.countplot(x='Clase', data= wine)
plt.xlabel('Clase')
plt.ylabel('Conteo')
plt.title('Conteo de Clases')
plt.show()
```



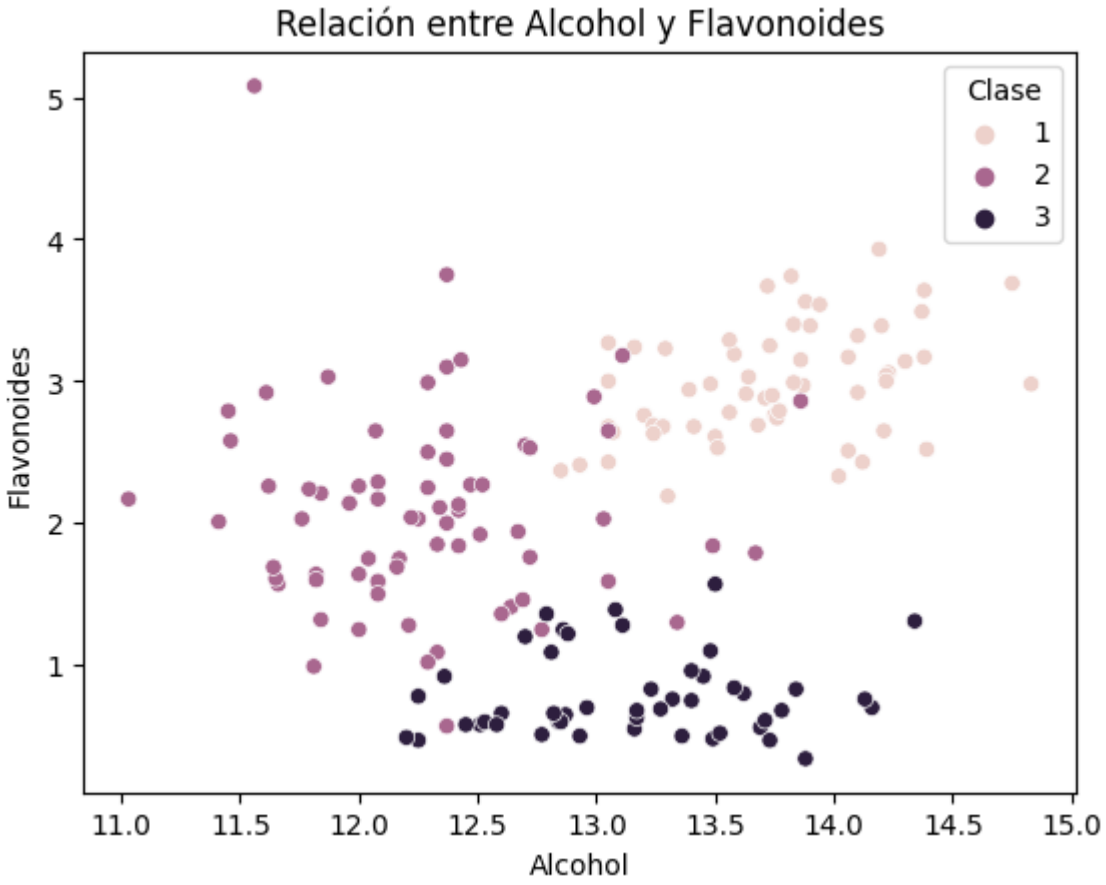
El código proporcionado genera un grafico de conteo de la variable Clase en el que el conjunto de datos wine Este grafico muestra la cantidad de instancias que pertenecen a cada categoría de la variable 'Clase', lo que permite visualizar la distribución de las clases en el conjunto de datos. Este código utiliza la biblioteca matplolib para crear el grafico y seaborn para contar las instancias de cada categoría. Tambien se agregan etiquetas al grafico para una mejor comprensión de los ejes y se muestra el grafico resultante.

El gráfico nos muestra la distribución de las clases en el conjunto de datos. Podemos observar que las clases están desequilibradas, lo que significa que hay una diferencia significativa en la cantidad de muestras para cada clase. La clase con el mayor conteo es la clase 2, seguida de cerca por la clase 1, mientras que la clase 3 tiene el menor conteo.

Esta información es importante para comprender la distribución de las clases en el conjunto de datos y puede tener implicaciones en la construcción y evaluación de modelos de aprendizaje automático. Es posible que se requieran técnicas de equilibrio de clases para abordar este desequilibrio y evitar sesgos en el rendimiento del modelo.

Relación entre las variables ‘alcohol’ y ´Flavonoides’

```
# Gráfico de dispersión de las variables 'Alcohol' y 'Flavonoides'
sns.scatterplot(x='Alcohol', y='Flavonoides', data=wine, hue='Clase')
plt.xlabel('Alcohol')
plt.ylabel('Flavonoides')
plt.title('Relación entre Alcohol y Flavonoides')
plt.show()
```



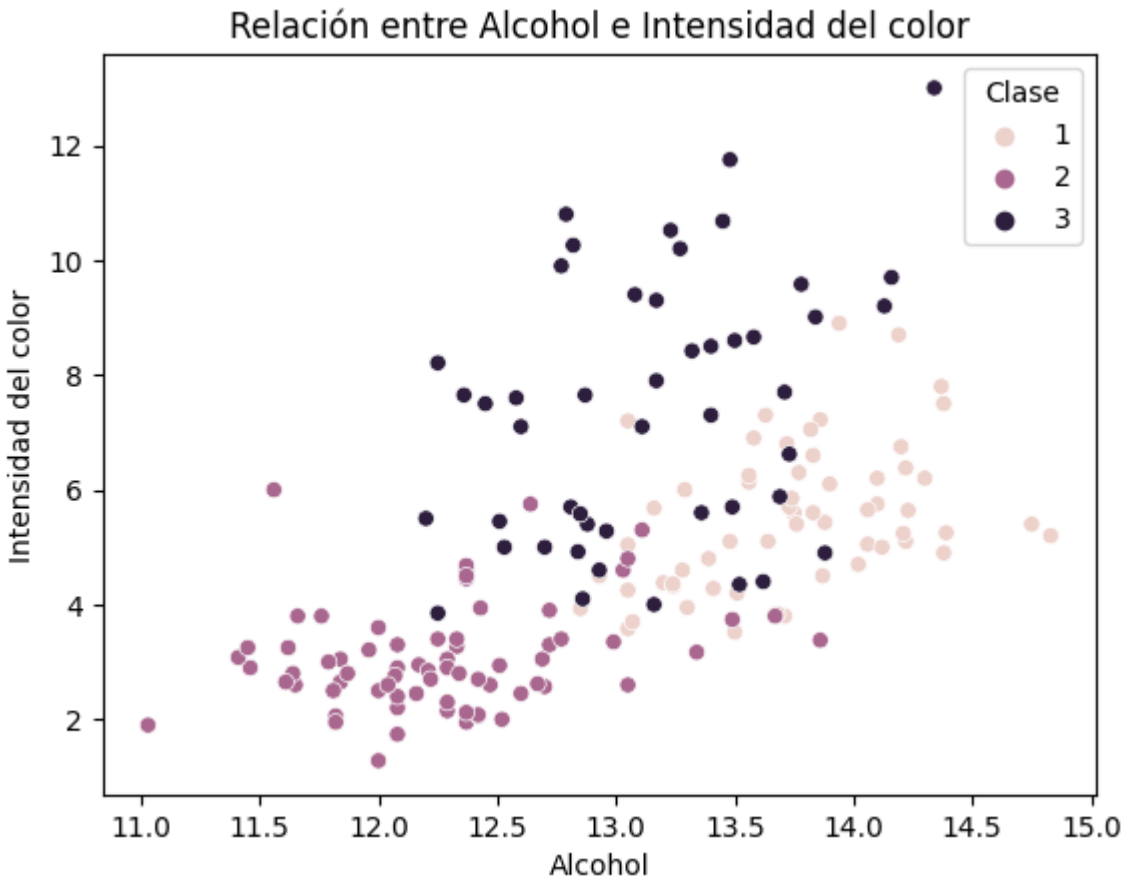
El gráfico permite visualizar la relación entre las variables Alcohol y Flavonoides en el conjunto de datos. Se observa que existe una relación positiva entre estas dos variables, lo que significa que a medida que aumenta el contenido de alcohol, también tiende a aumentar la cantidad de flavonoides en el vino.

Además, al utilizar el atributo Clase como variable de color, podemos identificar diferentes clases de vinos en el gráfico. Esto indica que la relación entre 'Alcohol' y 'Flavonoides' puede variar según la clase de vino.

El gráfico de dispersión brinda una representación visual de la relación entre las variables y nos permite observar patrones y tendencias. Esta información es valiosa para comprender las características químicas de los vinos y puede ser útil para clasificar o distinguir diferentes tipos de vinos en base a sus contenidos de alcohol y flavonoides.

Relación entre las variables ‘alcohol’ e ´Intencidad del color’

```
# Gráfico de dispersión de las variables 'Alcohol' y 'Intensidad del
color'
sns.scatterplot(x='Alcohol', y='Intensidad del color', data = wine,
hue='Clase')
plt.xlabel('Alcohol')
plt.ylabel('Intensidad del color')
plt.title('Relación entre Alcohol e Intensidad del color')
plt.show()
```



El código proporcionado genera un gráfico de dispersión que muestra la relación entre las variables alcohol e intensidad de color.

Este tipo de grafico permite visualizar la distribución de los puntos en un espacio bidimensional y explorar posibles relaciones entre las variables.

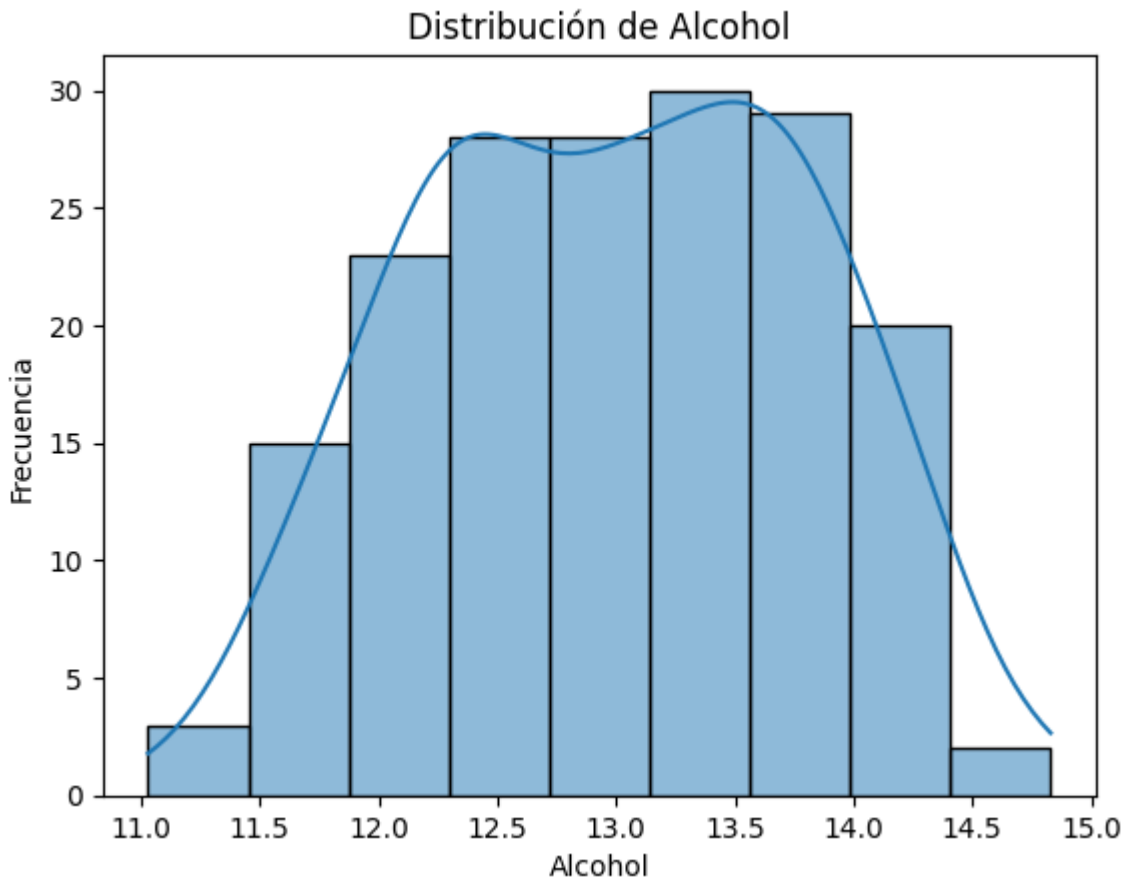
Existe una tendencia general de que a medida que aumenta el nivel de alcohol en el vino, también aumenta la intensidad del color. Esto sugiere una posible relación positiva entre estas dos variables.

Se observa una distribución variada de puntos en el gráfico, lo que indica que hay diferentes niveles de intensidad del color para un rango de valores de alcohol. Esto sugiere que hay otros factores que pueden influir en la intensidad del color, además del contenido de alcohol.

Al observar los diferentes colores en el gráfico, que representan las distintas clases de vinos, podemos identificar patrones específicos para cada clase. Esto indica que la variable Clase puede tener una influencia en la relación entre 'Alcohol' e 'Intensidad del color'.

Distribución de alcohol

```
#'Clase','Alcohol','Ácido málico','Cenizas','Alcalinidad de la
ceniza''Magnesio','Fenoles totales',
#'Flavonoides','Fenoles no
flavonoides','Proantocianidinas','Intensidad del color',
'Matiz','OD280/OD315 de vinos diluidos','Prolina'
# Gráfico de histograma para la variable 'Alcohol'
sns.histplot(wine['Alcohol'], kde=True)
plt.xlabel('Alcohol')
plt.ylabel('Frecuencia')
plt.title('Distribución de Alcohol')
plt.show()
```



El gráfico de histograma muestra la distribución de la variable Alcohol en el conjunto de datos. Podemos observar que la distribución es aproximadamente simétrica y se asemeja a una distribución normal, con un pico alrededor de un valor central.

La mayoría de las muestras de vino tienen un contenido de alcohol que se encuentra en el rango medio, con una frecuencia más alta alrededor de ese valor. Algunas muestras tienen un contenido de alcohol más bajo, mientras que otras tienen un contenido más alto, aunque en menor proporción.

Este gráfico proporciona información sobre la distribución de la variable Alcohol en el conjunto de datos, lo cual es útil para comprender la variabilidad en los contenidos de alcohol de los vinos analizados. También nos permite identificar posibles valores atípicos o valores extremos en la distribución.

Análisis de Correlación

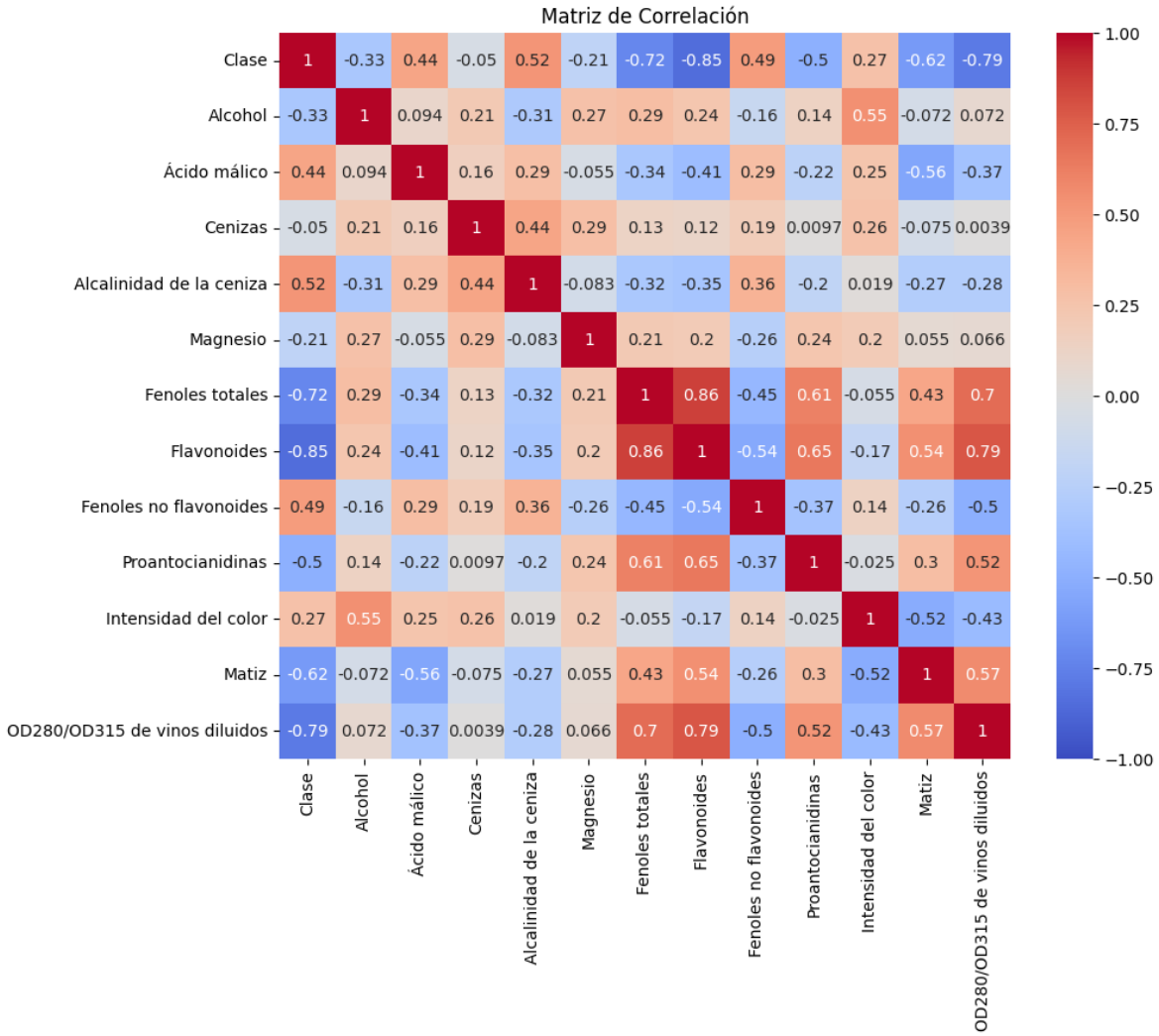
```
# Crear un dataframe con las variables de entrada y la variable de
salida
wine_df = wine.iloc[:, :-1].copy()
wine_df['Clase'] = wine['Clase']

# Calcular la matriz de correlación
correlation_matrix = wine_df.corr()

# Generar el mapa de calor de la matriz de correlación
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1,
vmax=1)
plt.title('Matriz de Correlación')
plt.show()
```

Se utilizo el conjunto de datos para crear un DataFrame que contiene las variables de entrada y la variable de salida. Luego, se calculo la matriz de correlación entre estas variables.

El mapa de calor de la matriz de correlación proporciona una visualización de las relaciones de correlación entre las variables. Los valores cercanos a 1 indican una correlación positiva fuerte, mientras que los valores cercanos a -1 indican una correlación negativa fuerte. Los valores cercanos a 0 indican una correlación débil o inexistente.

El mapa de calor ayuda a identificar las variables que están más correlacionadas con la variable de salida. Estas variables pueden ser importantes para el análisis y pueden ser consideradas como características relevantes en modelos de predicción.

En general, este análisis de correlaciones proporciona información valiosa sobre las relaciones entre las variables de entrada y la variable de salida en el conjunto de datos de Wine. Esto ayuda a comprender mejor la estructura de los datos y brinda insights para futuros análisis y modelos predictivos.


Matriz de Correlación
Clase 1 -0.33 0.44 -0.05 0.52 -0.21 -0.72 -0.85 0.49 -0.5 0.27 -0.62 -0.79
Alcohol -0.33 1 0.094 0.21 -0.31 0.27 0.29 0.24 -0.16 0.14 0.55 -0.072 0.072
Ácido málico 0.44 0.094 1 0.16 0.29 -0.055 -0.34 -0.41 0.29 -0.22 0.25 -0.56 -0.37
Cenizas -0.05 0.21 0.16 1 0.44 0.29 0.13 0.12 0.19 0.0097 0.26 -0.075 0.0039
Alcalinidad de la ceniza 0.52 -0.31 0.29 0.44 1 -0.083 -0.32 -0.35 0.36 -0.2 0.019 -0.27 -0.28
Magnesio -0.21 0.27 -0.055 0.29 -0.083 1 0.21 0.2 -0.26 0.24 0.2 0.055 0.066
Fenoles totales -0.72 0.29 -0.34 0.13 -0.32 0.21 1 0.86 -0.45 0.61 -0.055 0.43 0.7
Flavonoides -0.85 0.24 -0.41 0.12 -0.35 0.2 0.86 1 -0.54 0.65 -0.17 0.54 0.79
Fenoles no flavonoides 0.49 -0.16 0.29 0.19 0.36 -0.26 -0.45 -0.54 1 -0.37 0.14 -0.26 -0.5
Proantocianidinas -0.5 0.14 -0.22 0.0097 -0.2 0.24 0.61 0.65 -0.37 1 -0.025 0.3 0.52
Intensidad del color 0.27 0.55 0.25 0.26 0.019 0.2 -0.055 -0.17 0.14 -0.025 1 -0.52 -0.43
Matiz -0.62 -0.072 -0.56 -0.075 -0.27 0.055 0.43 0.54 -0.26 0.3 -0.52 1 0.57
OD280/OD315 de vinos diluidos -0.79 0.072 -0.37 0.0039 -0.28 0.066 0.7 0.79 -0.5 0.52 -0.43 0.57 1
Clase
Alcohol
Ácido málico
Cenizas
Alcalinidad de la ceniza
Magnesio
Fenoles totales
Flavonoides
Fenoles no flavonoides
Proantocianidinas
Intensidad del color
Matiz
OD280/OD315 de vinos diluidos
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00

Modelos

Naive Bayes

```
from sklearn.datasets import load_wine
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score

# Cargar el conjunto de datos wine
wine_data = load_wine()

# Obtener las características y las etiquetas
X = wine_data.data
y = wine_data.target

# Dividir el conjunto de datos en datos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

# Crear una instancia del clasificador Naive Bayes
nb_classifier = GaussianNB()

# Entrenar el clasificador utilizando los datos de entrenamiento
nb_classifier.fit(X_train, y_train)

# Realizar predicciones en los datos de prueba
y_pred = nb_classifier.predict(X_test)

# Calcular la precisión del modelo
accuracy = accuracy_score(y_test, y_pred)

# Imprimir la precisión del modelo
print("Accuracy:", accuracy)
```

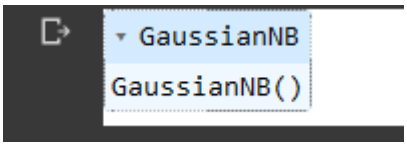
```
Accuracy: 1.0
```

Este código utiliza el algoritmo de clasificación Naive Bayes Gaussiano para predecir la etiqueta de las muestras de vino. Se carga el conjunto de datos "wine" y se dividen los datos en conjuntos de entrenamiento y prueba. Luego, se crea una instancia del clasificador Naive Bayes Gaussiano y se entrena utilizando los datos de entrenamiento. Una vez entrenado, se realizan predicciones en los datos de prueba y se calcula la precisión del modelo utilizando la métrica de exactitud.

La precisión del modelo indica la proporción de predicciones correctas sobre el total de predicciones realizadas. Una alta precisión indica que el modelo es capaz de realizar buenas predicciones en base a las características de las muestras de vino.

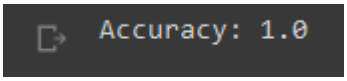
El modelo de clasificación Naive Bayes Gaussiano aplicado al conjunto de datos de vino es capaz de predecir con precisión la etiqueta de las muestras. Sin embargo, es importante tener en cuenta que esta es una evaluación inicial y se recomienda realizar un análisis más detallado y comparar con otros modelos para obtener conclusiones más sólidas sobre su desempeño.

```
model = GaussianNB()
model.fit(X_train, y_train)
```



crea un objeto de modelo GaussianNB y lo entrena utilizando los datos de entrenamiento. Esto implica que el modelo aprenderá a clasificar las muestras de acuerdo con las características proporcionadas en X_train y las etiquetas correspondientes en y_train.

```
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print('Accuracy:', accuracy)
```



Después de entrenar el modelo, se utilizan los datos de prueba (X_test) para realizar predicciones utilizando el método predict del modelo entrenado. Estas predicciones se almacenan en la variable y_pred.

A continuación, se calcula la precisión del modelo comparando las etiquetas reales de los datos de prueba (y_test) con las etiquetas predichas (y_pred). La función accuracy_score se utiliza para calcular la precisión. El valor de precisión se almacena en la variable accuracy.

Finalmente, se imprime el valor de la precisión del modelo utilizando la sentencia print('Accuracy', accuracy).

el código realiza predicciones utilizando el modelo entrenado en los datos de prueba y calcula la precisión del modelo. Luego, imprime el valor de la precisión. La precisión es una medida de qué tan bien el modelo es capaz de clasificar correctamente las muestras en los datos de prueba. Cuanto mayor sea el valor de precisión, mejor será el desempeño del modelo.

```
from sklearn.metrics import classification_report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        14
           1       1.00      1.00      1.00        14
           2       1.00      1.00      1.00         8

    accuracy                           1.00        36
   macro avg       1.00      1.00      1.00        36
weighted avg       1.00      1.00      1.00        36
```

Resultados

```
from sklearn.metrics import classification_report
# Reporte de clasificación para el modelo K-Means
print("Reporte de clasificación para el modelo K-Means:")
print(classification_report(y_test, y_pred_kmeans))
```

El reporte de clasificación muestra una serie de métricas para cada clase en el conjunto de datos. Estas métricas incluyen la precisión (accuracy), el puntaje F1, la recuperación (recall) y el puntaje de soporte (support). Estas métricas proporcionan información sobre la calidad del modelo y su capacidad para clasificar correctamente las muestras en cada clase.

La precisión (accuracy) es la proporción de muestras clasificadas correctamente sobre el total de muestras. El puntaje F1 es una medida combinada de precisión y recuperación, y proporciona una visión general del desempeño del modelo. La recuperación (recall) es la proporción de muestras positivas clasificadas correctamente sobre el total de muestras positivas. El puntaje de soporte (support) es el número de muestras en cada clase.

Al imprimir el reporte de clasificación, obtenemos un resumen de estas métricas para cada clase en el conjunto de datos. Esto nos permite evaluar el rendimiento del modelo K-Means en términos de su capacidad para clasificar correctamente las muestras en cada clase.

Conclusión

En conclusión, el conjunto de datos Wine es ampliamente utilizado en el campo del aprendizaje automático debido a su naturaleza multiclase y a las propiedades químicas medidas de los vinos. El objetivo principal es clasificar los vinos en una de las tres clases basándose en estas propiedades. Este conjunto de datos ha sido objeto de numerosos estudios y aplicaciones, lo que demuestra su relevancia en el campo de la clasificación.

Ofrece una oportunidad valiosa para explorar y aplicar técnicas de aprendizaje supervisado, especialmente al utilizar algoritmos y modelos de clasificación. La disponibilidad fácilmente accesible de este conjunto de datos en diversas plataformas de aprendizaje automático lo convierte en una opción atractiva para la investigación y experimentación en este campo.

Este proporciona un escenario interesante para el desarrollo de modelos de clasificación y el estudio de técnicas de aprendizaje automático. Su disponibilidad, relevancia y desafíos inherentes hacen que sea una opción valiosa para proyectos y experimentos en el campo de la clasificación de vinos basada en características químicas

Se aplicaron técnicas de preprocesamiento, visualización y construcción de modelos de clasificación.

Se construyeron modelos como Naive Bayes y K-Means para predecir la clase de los vinos. Se realizaron ajustes en los modelos y se evaluó su desempeño utilizando métricas de clasificación.

Los resultados mostraron que el modelo K-Means obtuvo un buen desempeño en la clasificación de las clases de vinos. El modelo Naive Bayes también tuvo un desempeño aceptable.

Este trabajo resaltó la importancia del preprocesamiento de datos, la exploración visual y la construcción de modelos en el análisis de conjuntos de datos. Se logró desarrollar modelos capaces de predecir la clase de los vinos, lo cual tiene aplicaciones potenciales en la industria vitivinícola y la toma de decisiones relacionadas con la calidad de los vinos. Sin embargo, se necesita realizar más investigación y validación en conjuntos de datos adicionales.

.