Exploración de la Calidad de Vino

Aplicación de técnicas de procesamiento de datos

Proyecto Final

<u>Espacio curricular:</u> Procesamiento de datos

<u>Año:</u> 2023

<u>Integrantes (GRUPO 11):</u>

- ➔ **Andrés Chaile**
- ➔ **Gustavo Godoy**
- ➔ **Romina Ulerich**
- ➔ **Diego Ormeño**
- ➔ **Adrian Sequeira**
- ➔ **Martín Oller**
- ➔ **Claudia Yon**

Explorando la calidad del vino

Una aproximación a través de técnicas de procesamiento de datos

Introducción

En el presente trabajo el objetivo es establecer, por medio de técnicas de procesamiento de datos, las variables más influyentes en la calidad sensorial del vino.Vamos a emplear dos conjuntos de datos centrados en la evaluación de la calidad de vinos, específicamente en las variantes de vino blanco y vino tinto del reconocido "Vinho Verde" portugués.

Estos datasets provienen del repositorio de Machine Learning de la Universidad de California Irvine (UCI). En este caso, solo contamos con información sobre las características físico-químicas del vino, que se utilizan como variables de entrada, así como los datos relacionados con sus propiedades sensoriales, que se consideran como la variable de salida en el estudio.

Descripción de las variables de entrada (basados en tests físico químicos):

1. **fixed acidity / acidez fija.** La mayoría de los ácidos involucrados con el vino son fijos o no volátiles (no se evaporan fácilmente).
2. **volatile acidity / acidez volátil.** La cantidad de ácido acético en el vino, que en niveles demasiado altos puede provocar un sabor desagradable a vinagre.
3. **citric acid / ácido cítrico.** Encontrado en pequeñas cantidades el ácido cítrico puede agregar 'frescura' y sabor a los vinos.
4. **residual sugar / azúcar residual.** La cantidad de azúcar que queda después de que se detiene la fermentación. Es raro encontrar vinos con menos de 1 gramo/litro y vinos con más de 45 gramos/litro se consideran dulces.
5. **chlorides / cloruros.** La cantidad de sal en el vino.
6. **free sulfur dioxide / dióxido de azufre libre.** La forma libre de SO2 existe en equilibrio entre el SO2 molecular (como gas disuelto) y el ion bisulfito; previene el crecimiento microbiano y la oxidación del vino.
7. **total sulfur dioxide / dióxido de azufre total.** Cantidad de formas libres y ligadas de S02; en bajas concentraciones, el SO2 es mayormente indetectable en el vino, pero en concentraciones de SO2 libres superiores a 50 ppm, el SO2 se vuelve evidente en la nariz y el sabor del vino.

8. **density / densidad.** La densidad del agua es cercana a la del agua dependiendo del porcentaje de alcohol y azúcar contenido.
9. **pH / pH.** Describe qué tan ácido o básico es un vino en una escala de 0 (muy ácido) a 14 (muy básico); la mayoría de los vinos están entre 3 y 4 en la escala de pH
10. **sulphates / sulfatos** Un aditivo del vino que puede contribuir a los niveles de dióxido de azufre (S02), que actúa como antimicrobiano y antioxidante
11. **alcohol / alcohol.** El porcentaje de contenido de alcohol del vino

Descripción de la variable de salida:

1. **quality / calidad.** Variable de salida o target (basada en datos sensoriales, puntuación entre 0 y 10). Indica qué tan bueno es el vino en este estándar de calidad.

~~Luego de~~ recolectar los datos, se procederá a realizar las tareas de limpieza, integración y transformación necesarias para poder realizar el análisis exploratorio,procesamiento, y análisis de datos.

Metodología y/o Resultados

Los métodos y técnicas utilizadas en el procesamiento y análisis de datos los explicamos en las siguientes etapas:

1. Recolección y preparación de datos

Como se mencionó antes los datasets provienen del repositorio de Machine Learning de la Universidad de California Irvine (UCI).

En primer lugar, con la ayuda de la librería Pandas, se importaron dos archivos CSV, 'winequality-red.csv' y 'winequality-white.csv', ubicados en la carpeta "data". Luego, los datos se almacenan en las variables vino_tinto y vino_blanco. Como segundo paso se explora las características de los datos (cantidad de filas y columnas, tipos de datos, cantidad de registros). Además se explora con las medidas estadísticas descriptivas de los datos (como el tamaño de la muestra, media, desvío estándar, mínimo, máximo y cuartiles)

I. Para la eliminación de los duplicados se utiliza primero el método `duplicated()` para saber si existen, y `sum()` para sumar y conocer sus cantidades si es el caso. Se usa el método `drop_duplicates` para eliminar los duplicados del conjunto de datos. Y por último, se verifica si el proceso resultó exitoso En cuanto a las

operaciones de transformación y consolidación de datos. Aquí está el desglose de lo que se realizó en cada parte:

II. Se agregó una nueva columna llamada "categoria":
    A. Se asignó el valor "rojo" a todos los registros en el DataFrame `vino_tinto_sd`.
    B. Se asignó el valor "blanco" a todos los registros en el DataFrame `vino_blanco_sd`.

III. Se concatenaron los conjuntos de datos de vino tinto y vino blanco en el DataFrame `df_vinos`:
    A. Se utilizó la función `pd.concat()` para concatenar los DataFrames `vino_tinto_sd` y `vino_blanco_sd`.
    B. El parámetro `ignore_index=True` se utilizó para restablecer los índices del DataFrame resultante.

IV. Se definió tres funciones de validación de la integración:
    A. `validar_filas()` comparó el número de filas esperadas con el número de filas en un DataFrame dado y mostró un mensaje de validación exitosa o una lista de las filas esperadas y encontradas en caso de discrepancia.
    B. `validar_columnas()` verificó si todas las columnas esperadas estaban presentes en un DataFrame y mostró un mensaje de validación exitosa o una lista de las columnas faltantes en caso de discrepancia.
    C. `validar_integracion()` utilizó las dos funciones anteriores para validar la integración de los conjuntos de datos de vino tinto y vino blanco en el DataFrame `df_vinos`.

V. Se llamó a la función `validar_integracion()` para validar la integración de los datos.

VI. Se creó un diccionario llamado `nuevos_nombres_col` que contenía los nombres de columna originales y sus correspondientes nuevos nombres.

VII. Se renombraron las columnas del DataFrame `df_vinos` utilizando el diccionario `nuevos_nombres_col`.

VIII. Se imprimió información resumida sobre el DataFrame `df_vinos` utilizando el método `info()` de pandas. Se mostró el rango de índices, el número de filas no nulas y el tipo de datos de cada columna. También se proporcionó el uso de memoria aproximado del DataFrame.

En cuanto al almacenamiento del DataSet se realizó en un archivo CSV, y además se verificó si se guardaron correctamente en la siguiente secuencia:

I. Se definió una variable `ruta_archivo_final` que contiene la ruta y nombre de archivo para el archivo CSV donde se almacenará el conjunto de datos final.
II. Se utilizó un bloque `try-except` para manejar posibles errores durante el proceso de guardado y lectura del archivo CSV.
III. Se guardó el DataFrame `df_vinos` en el archivo CSV utilizando el método `to_csv()` de pandas. El parámetro `index=False` se utilizó para evitar que se guarde el índice del DataFrame en el archivo.
IV. Se leyó el archivo CSV recién guardado en un nuevo DataFrame llamado `df_guardado` utilizando la función `pd.read_csv()`.
V. Se realizó una comparación entre el DataFrame original `df_vinos` y el DataFrame recién leído `df_guardado` utilizando el método `equals()`. Si los dos DataFrames son iguales, se imprimió el mensaje "El dataset ha sido preparado y está listo para su análisis".
VI. Si los DataFrames no son iguales, se imprimió el mensaje "Error: Los datos no coinciden después del guardado".
VII. Si se produjo algún error durante el guardado o la lectura del archivo CSV, se capturó la excepción y se imprimió un mensaje de error indicando la descripción de la excepción.

De esta manera el DataSet ya está listo para su análisis.

2. Análisis exploratorio de datos:

En primera instancia se importaron las bibliotecas `seaborn`, `numpy` y `matplotlib.pyplot`, lo que permite utilizar las funcionalidades proporcionadas por estas bibliotecas en el código que sigue a continuación.

Para conocer los tipos de datos de cada columna en el DataFrame `df_vinos`. Se realizó los siguiente:

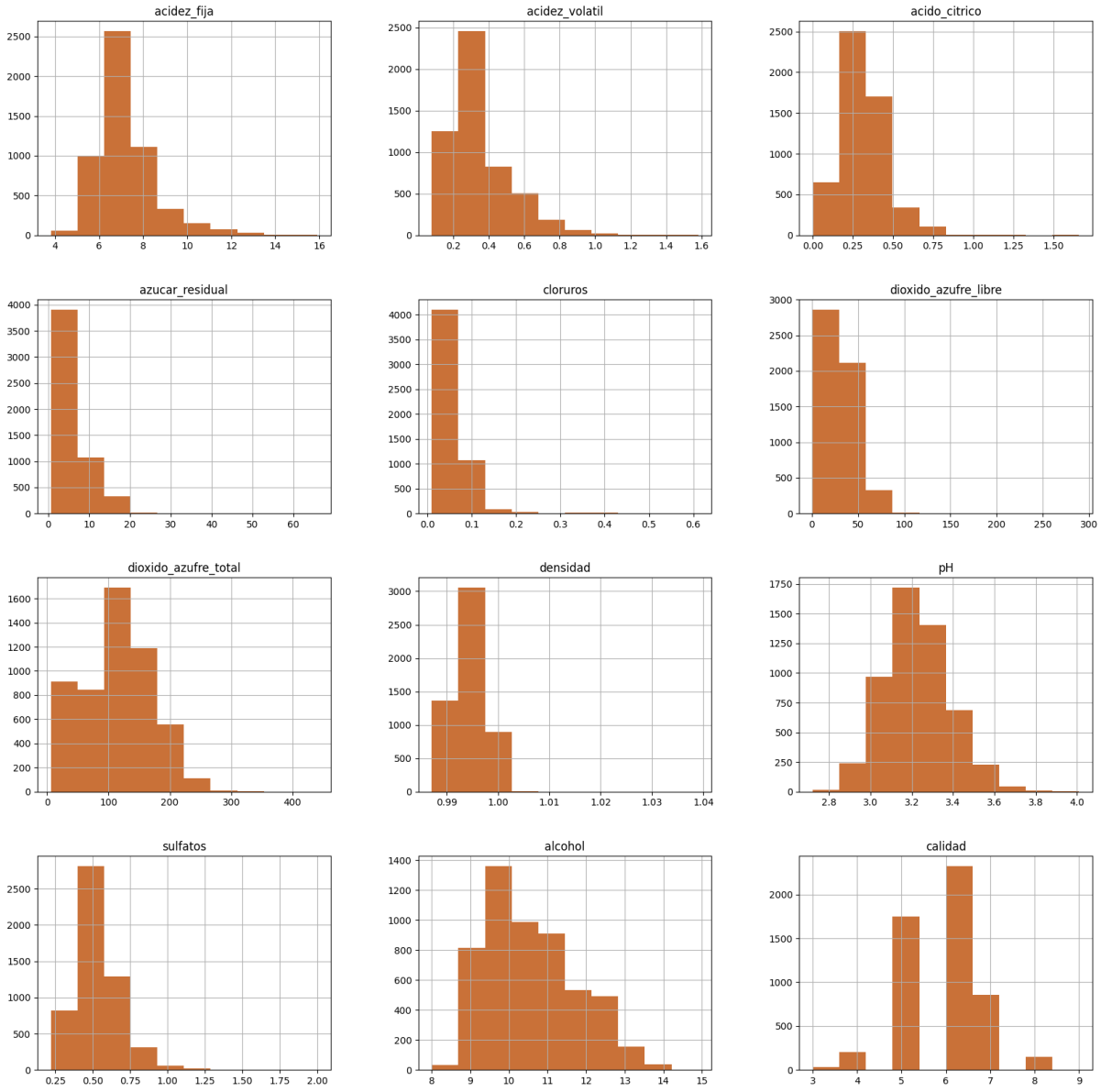
I. Se utilizó el atributo `dtypes` del DataFrame `df_vinos` para obtener los tipos de datos de cada columna.
II. Se mostraron los tipos de datos de las columnas en forma de tabla, donde cada columna se muestra junto con su tipo de dato correspondiente.

```
acidez_fija              float64
acidez_volatil           float64
acido_citrico            float64
azucar_residual          float64
cloruros                 float64
dioxido_azufre_libre     float64
dioxido_azufre_total     float64
densidad                 float64
pH                       float64
sulfatos                 float64
alcohol                  float64
calidad                    int64
categoria                 object
dtype: object
```

Para conocer las medidas estadísticas descriptivas:

I. Se utilizó el método `describe()` en el DataFrame `df_vinos` para obtener un resumen estadístico que incluye medidas como la media, la desviación estándar y los cuartiles para cada columna del DataFrame. Los resultados se mostraron en forma de tabla.

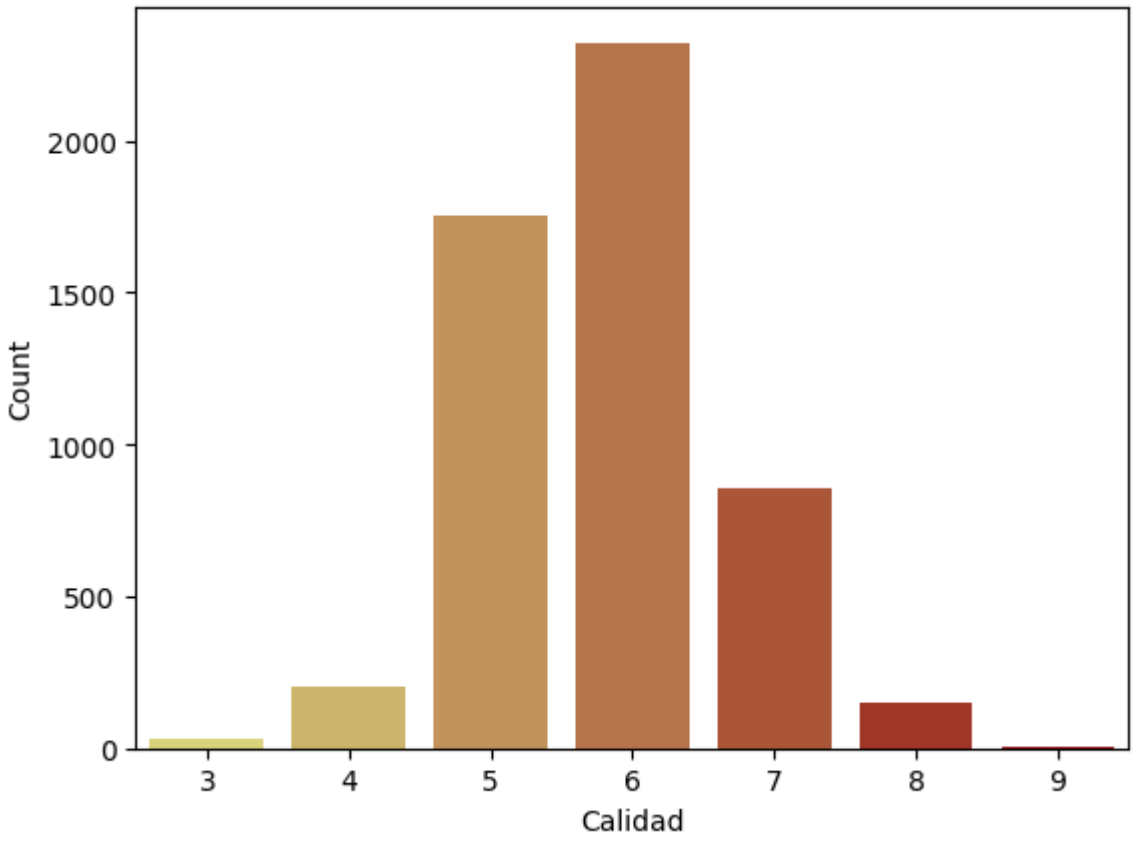
II. Se generaron histogramas para cada una de las variables del DataFrame utilizando el método `hist()` del DataFrame `df_vinos`. El parámetro `figsize=(20,20)` se utilizó para ajustar el tamaño de la figura del histograma. Todos los histogramas se mostraron juntos utilizando `plt.show()`.

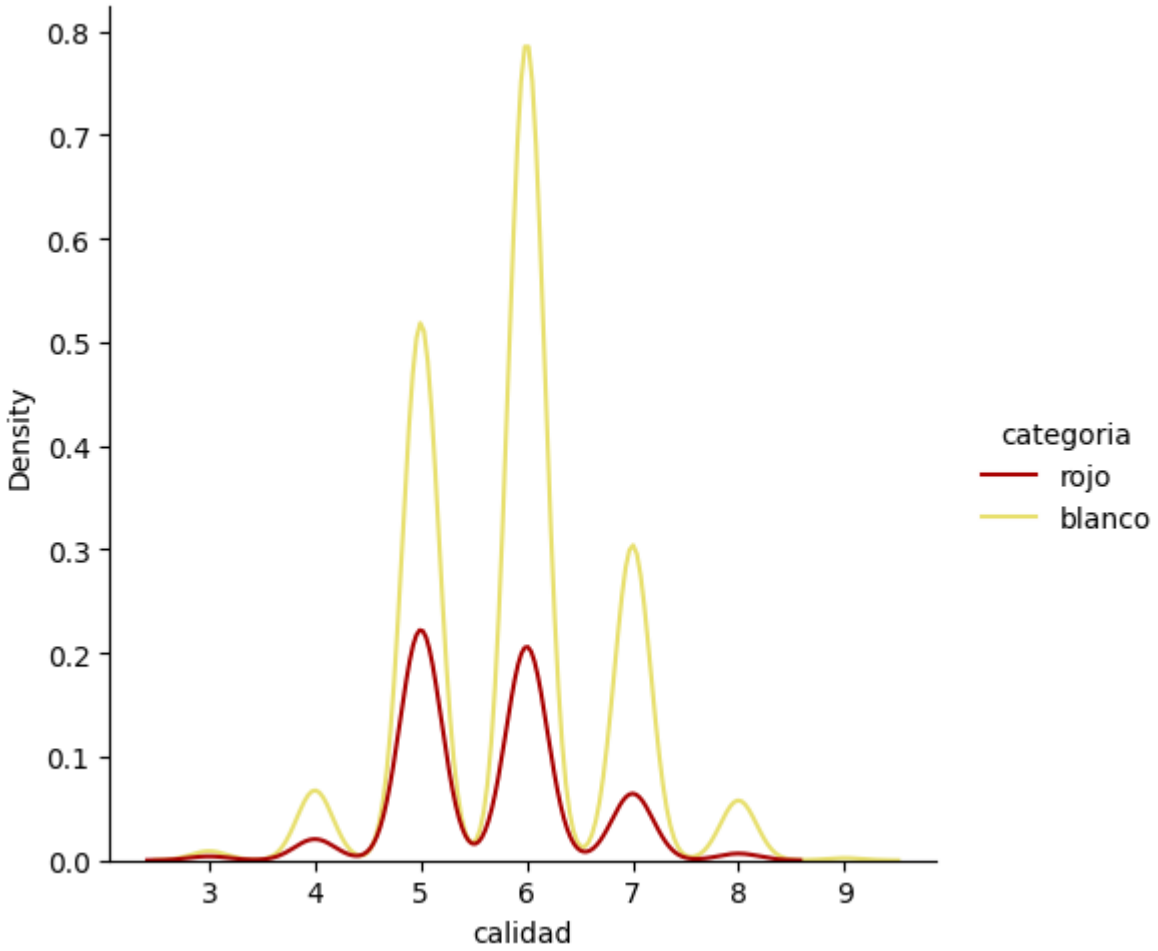


III. Se realizó un análisis de diagramas de caja para cada variable en relación a las categorías de vinos. Se utilizó un bucle `for` para iterar sobre las columnas del DataFrame, excepto las dos últimas. Se generaron diagramas de caja utilizando el método `boxplot()` de seaborn (`sns.boxplot()`) con la variable de categoría `x` y la variable de interés `y`. Se utilizó una paleta de colores personalizada para representar las categorías de vinos. Se asignaron títulos a cada diagrama de caja utilizando información extraída del nombre de la columna. Cada diagrama de caja se mostró utilizando `plt.show()`.

Buscando conocer la distribución de la variable “calidad” se hacen varias operaciones. Se examina la variable "calidad" en términos de conteo de registros y se presentan visualmente las distribuciones mediante gráficos de barras y KDE.

Los pasos seguidos son de la siguiente manera: En primer lugar, se extrae la columna "calidad" del DataFrame "df_vinos" y se almacena en una variable llamada "calidad". Luego, se utiliza la función "`value_counts()`" para contar la cantidad de registros para cada nivel de calidad, generando una serie llamada "`cantidad_por_calidad`". A continuación, se crea un gráfico de barras utilizando la función "`countplot()`", donde se muestra la distribución de registros según los niveles de calidad. Además, se divide el análisis según la columna "categoria" y se crea un gráfico de densidad de kernel (KDE) utilizando la función "`displot()`". Este último gráfico muestra la distribución de la variable "calidad" para cada categoría, permitiendo visualizar las diferencias entre ellas.





Se observa que la calidad entre 5 y 7 son las que tienen mayor frecuencia en ambos tipos de vinos.

3. Procesamiento y análisis de datos:

3.1. Segmentación de la variable calidad

Se llevó a cabo la segmentación de la variable "calidad" en el dataset de vinos. Se definieron cinco categorías para clasificar los vinos en función de su calidad: "Bajo" para los vinos con calidad inferior a 5, "Medio Bajo" para aquellos con calidad igual a 5, "Medio" para los vinos con calidad igual a 6, "Medio Alto" para los de calidad igual a 7, y "Alto" para los vinos con calidad superior a 7.

A continuación, se aplicó la categorización a través de la función "`apply`" en la columna "calidad" del dataframe "df_vinos". Se utilizó una expresión `lambda` para evaluar el valor de "x" (la calidad) y asignar la categoría correspondiente según las condiciones establecidas.

El resultado de esta categorización se guardó en una nueva columna llamada "categoria_calidad".

Posteriormente, se imprimieron las primeras filas del dataframe "df_vinos" utilizando el método "`head()`", lo que permitió observar cómo se agregó la columna "categoria_calidad" con las categorías asignadas a cada vino en función de su calidad. También e utiliza la función `describe` con el parámetro `include=[object]` para obtener estadísticas descriptivas de las columnas categóricas "categoria" y "categoria_calidad" del dataframe. Esto permite conocer la cantidad de registros, el número de categorías únicas, la categoría más frecuente y su frecuencia.

	categoria	categoria_calidad
count	5320	5320
unique	2	5
top	blanco	Medio
freq	3961	2323

3.2. Relación entre variables: Correlación.

3.2.1 Tratamientos de outliers

Se realizó un tratamiento de outliers utilizando el método del rango intercuartílico (IQR, por sus siglas en inglés). En primer lugar, se identifican las columnas de interés en el DataFrame `df_vinos`, excluyendo las columnas 'categoria' y 'categoria_calidad'. Estas columnas se almacenan en la variable `columns_of_interest`.

A continuación, se calculan los límites del rango intercuartílico (Q1 y Q3) para cada columna mediante el uso de la función `quantile()` aplicada al DataFrame `df_vinos` y pasando el argumento correspondiente. El resultado se almacena en las variables `Q1` y `Q3`, respectivamente. Luego, se calcula el IQR restando Q1 a Q3 y se guarda en la variable `IQR`.

Para detectar los outliers, se establecen límites inferiores y superiores multiplicando 1.5 por el IQR y restando o sumando estos valores a Q1 y Q3, respectivamente. Los límites inferior y superior se almacenan en las variables `lower_bound` y `upper_bound`.

A continuación, se utiliza la operación lógica `|` (OR) para verificar si algún valor en cada columna es menor que el límite inferior o mayor que el límite superior. Esto se realiza mediante la comparación de las columnas del DataFrame `df_vinos` con los límites inferior y superior. El resultado es una serie booleana llamada `outliers`.

Para imprimir los registros con outliers, se utilizó `df_vinos[outliers]`, lo que devuelve un nuevo DataFrame que contiene solo las filas que cumplen con esta condición. Estas filas representan los registros que contienen outliers en al menos una de las columnas seleccionadas.

A continuación, se filtran los registros sin outliers utilizando el operador `~` (complemento lógico) aplicado a la serie booleana `outliers`. El resultado se asigna al DataFrame `df_vinos_filtered`, que contiene únicamente los registros que no presentan outliers en ninguna de las columnas seleccionadas.

Finalmente, se imprime el DataFrame `df_vinos_filtered`, mostrando los registros filtrados sin outliers.

3.3. Comparación de las variables dependientes con y sin filtrar outliers

La comparación se realizó entre las variables dependientes del DataFrame original `df_vinos` y el DataFrame filtrado `df_vinos_filtered`, centrándose en la variable categórica "categoria_calidad".

Además, se muestra que la categoría de calidad más frecuente en ambos DataFrames es "Medio". Esto se confirma mediante el bucle `for` que itera sobre las categorías únicas en la variable "categoria_calidad" del DataFrame filtrado y las imprime.

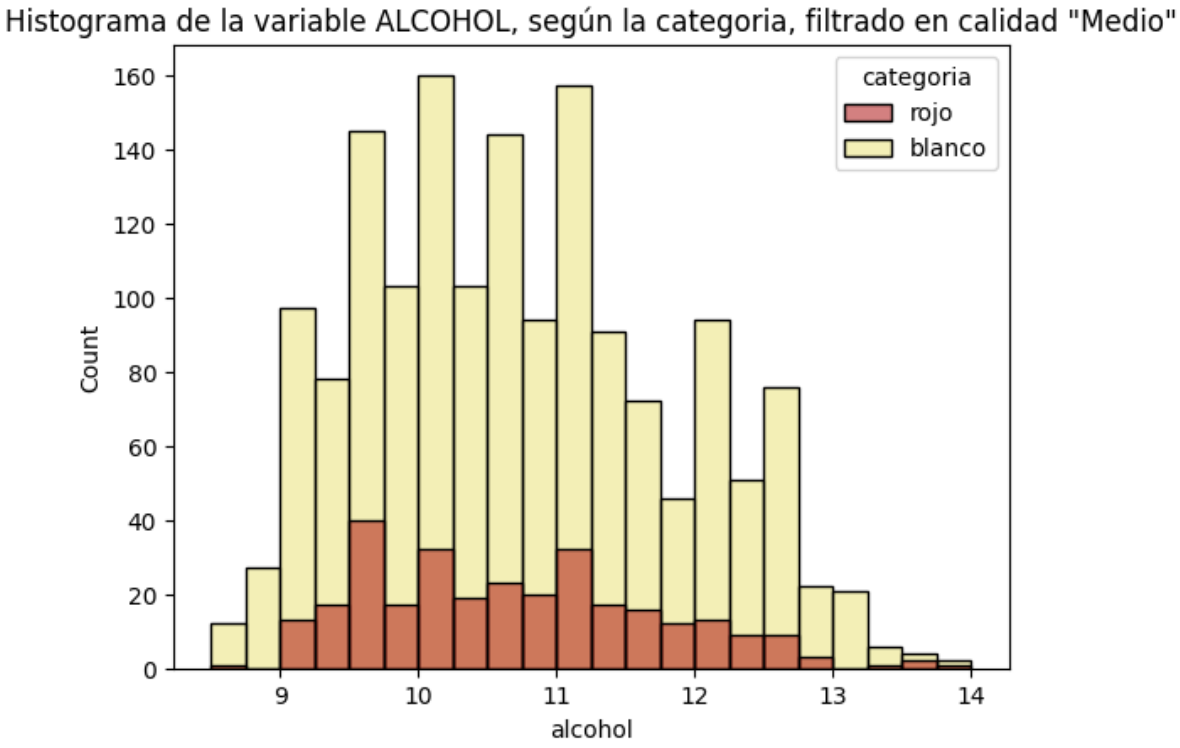
```
Medio Bajo
Medio Alto
Bajo
Medio
```

Con el objetivo de realizar una comparativa entre las categorías de vino (rojo y blanco), se selecciona la categoría de calidad "Medio" como referencia.

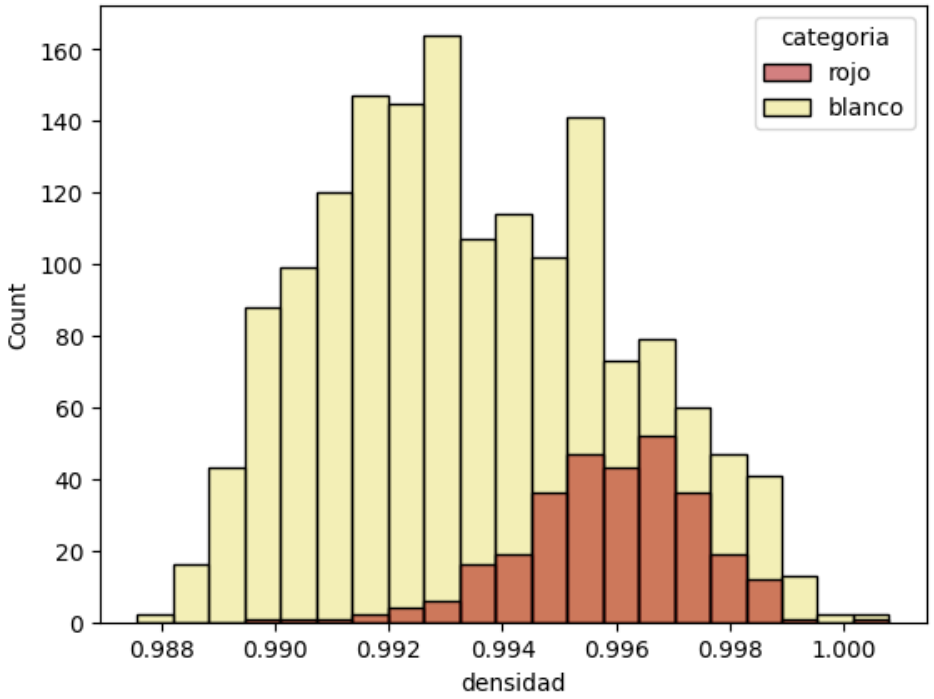
3.4. Comparaciones de Categoria de Calidad "Medio"

Se generaron visualizaciones para analizar la categoría de calidad "Medio" en un conjunto de datos de vinos.

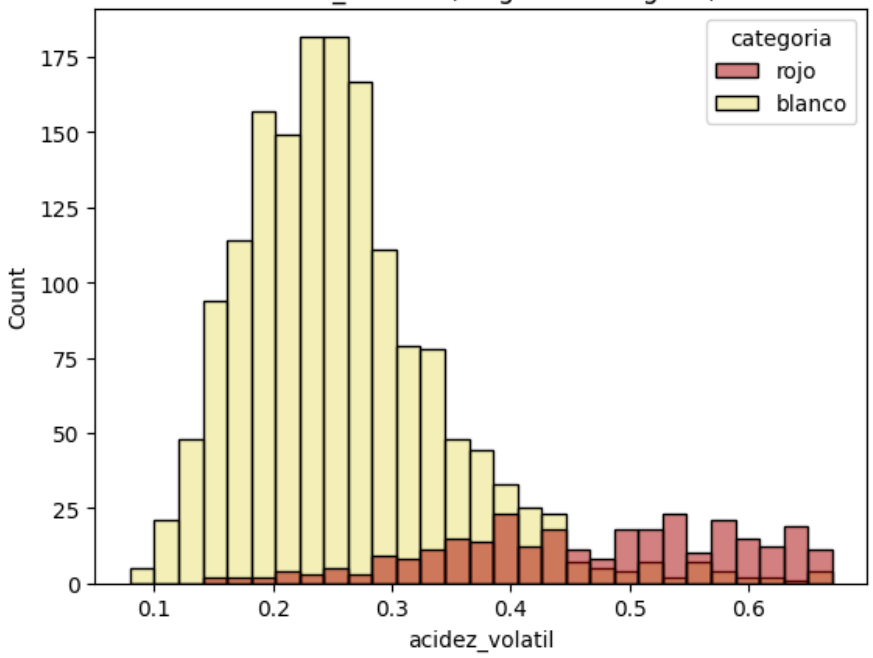
- Se crean histogramas para las variables "alcohol", "densidad", "cloruros" y "acidez_volatil" en la categoría "Medio". Cada histograma muestra la distribución de los valores de la variable correspondiente, diferenciando las categorías con colores distintos.



Histograma de la variable DENSIDAD, según la categoria, filtrado en calidad "Medio"



Histograma de la variable ACIDEZ_VOLATIL, según la categoria, filtrado en calidad "Medio"



- Además de los histogramas, se genera un gráfico de dispersión (scatterplot) que muestra la relación entre las variables "densidad" y "azucar_residual" en la categoría "Medio". También se utiliza el color para diferenciar las categorías en el gráfico.

3.5. Comparaciones de las medias entre categoria de vinos

Se realizaron comparaciones de las medias entre las categorías de vinos "blanco" y "tinto".

Primero, se calcula el promedio de cada campo del DataFrame `df_vinos_filtered` para la categoría de vinos tintos, utilizando la función `mean()` y filtrando las filas que tienen el valor de "categoria" igual a "rojo". El resultado se guarda en la variable `def_tinto`.

Luego, se realiza el mismo cálculo del promedio para la categoría de vinos blancos, filtrando las filas que tienen el valor de "categoria" igual a "blanco". El resultado se guarda en la variable `def_blanco`.

Después, se crea un nuevo DataFrame llamado 'resultado' utilizando la función `pd.concat()`, que combina los promedios de cada categoría en columnas distintas.

A continuación, se renombran los títulos de las columnas en el DataFrame 'resultado' utilizando el método `rename()` y un diccionario de reemplazo.

Por último, se imprimen en pantalla el DataFrame 'resultado', que muestra los promedios de cada campo para las categorías de vinos "blanco" y "tinto".

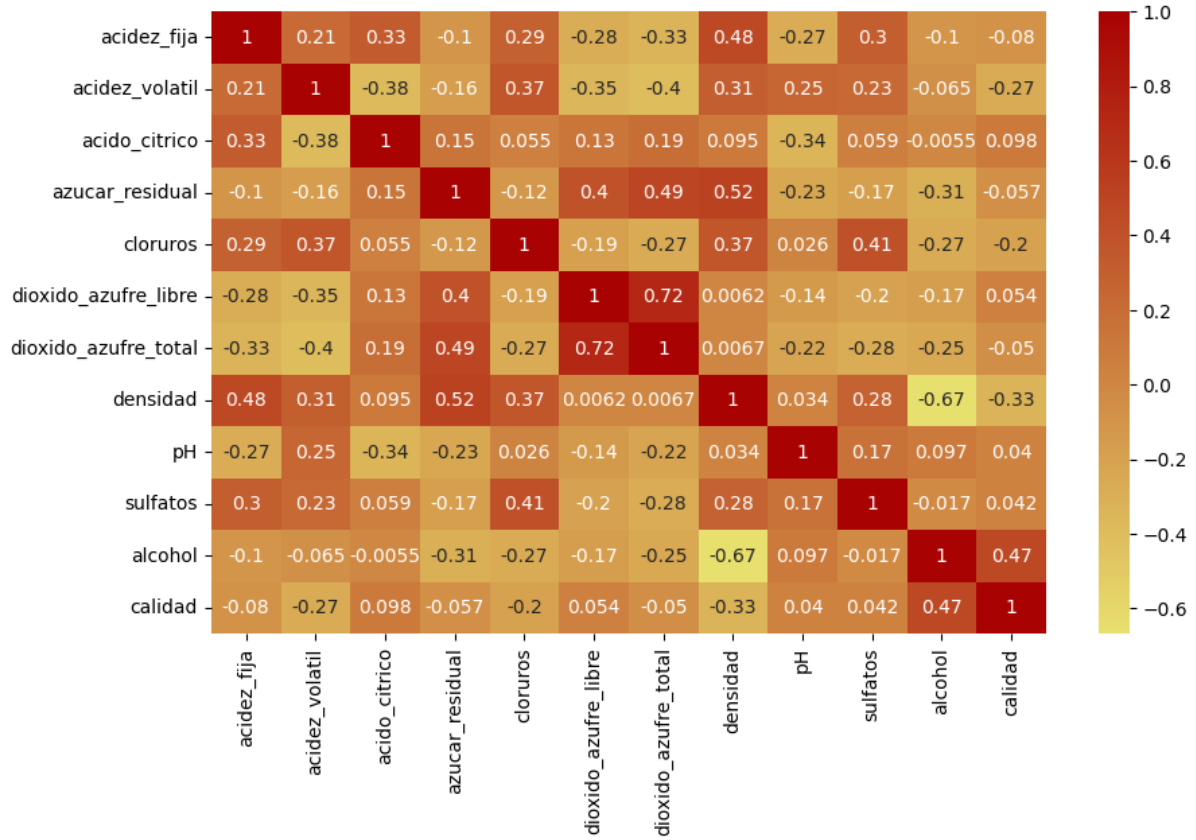
	Blanco	Tinto
acidez_fija	6.825739	7.621951
acidez_volatil	0.275471	0.483680
acido_citrico	0.321832	0.232568
azucar_residual	5.460638	2.397489
cloruros	0.042853	0.076085
dioxido_azufre_libre	33.746602	16.855811
dioxido_azufre_total	134.960106	49.309182
densidad	0.993552	0.996159
pH	3.197367	3.342712
sulfatos	0.486587	0.616557
alcohol	10.630175	10.402463
calidad	5.825946	5.625538

3.6. Correlaciones

Se realizó un análisis de correlaciones en un conjunto de variables seleccionadas del DataFrame `df_vinos`.

Primero, se calcula la matriz de correlaciones utilizando el método `corr()` sobre las columnas específicas del DataFrame. Estas columnas se seleccionan mediante el método `filter()` y se incluyen en la lista proporcionada.

A continuación, se muestra en pantalla la matriz de correlaciones utilizando `vinos_corr`. La matriz muestra los valores de correlación entre las variables seleccionadas. Luego, se crea un gráfico de calor utilizando la función `heatmap()` de la biblioteca Seaborn. Además, se incluyen los valores de correlación en cada celda del gráfico utilizando el parámetro `annot=True`. Finalmente, se muestra el gráfico de calor en una figura con tamaño de 10x6 utilizando `plt.figure()` y `plt.show()`.



4. Visualización de datos:

4.1. Distribuciones de variables con outliers

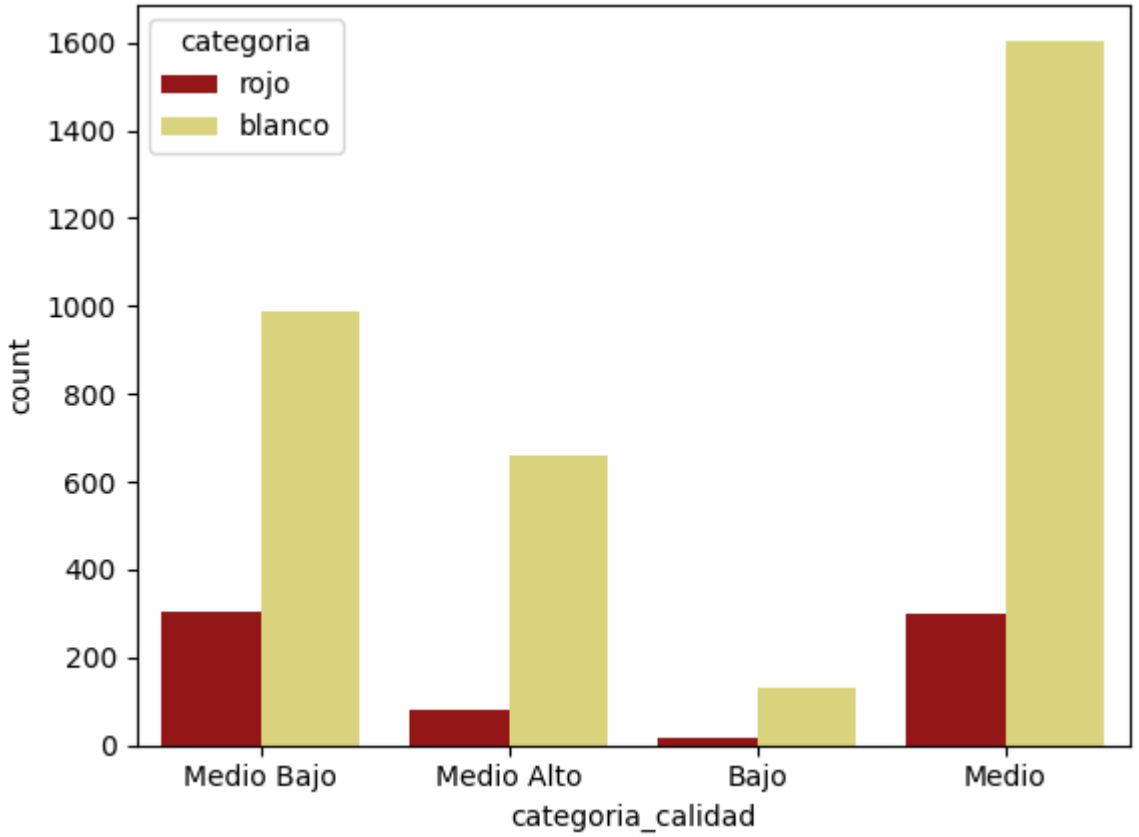
Se utiliza la función `pairplot()` de la biblioteca Seaborn (`sns`) para crear una matriz de gráficos de dispersión que muestra las distribuciones de las variables del DataFrame `df_vinos`. Cada gráfico de dispersión representa la relación entre dos variables y se colorea según la categoría especificada por la columna `'categoria'`. Finalmente, se muestra el pairplot utilizando `plt.show()` para visualizarlo en una figura separada.

4.2. Distribuciones de variables sin outliers

Se utiliza la función `pairplot()` de la biblioteca Seaborn (`sns`) para crear una matriz de gráficos de dispersión que muestra las distribuciones de las variables filtradas en el DataFrame `df_vinos_filtered`. Cada gráfico de dispersión representa la relación entre dos variables y se colorea según la categoría especificada por la columna `'categoria'`. Finalmente, se muestra el pairplot utilizando `plt.show()` para visualizarlo en una figura separada.

4.3. Cantidades de vinos segun su categoria y calidad

Se usó la función `countplot()` de la biblioteca Seaborn (`sns`) para crear un gráfico de barras que muestra las cantidades de vinos según su categoría y calidad en el DataFrame `df_vinos_filtered`. La variable `'categoria_calidad'` se muestra en el eje x, y los colores de las barras se asignan según la columna `'categoria'`.



4.4. Correlación de la calidad con resto de las variables

Se calculó la correlación de la calidad del vino con respecto a otras variables seleccionadas en el DataFrame `df_vinos_filtered`. Se utiliza la función `corr()` para calcular la matriz de correlación entre las variables seleccionadas, y luego se selecciona la columna `'calidad'` y se ordena en orden descendente.

Finalmente, se imprime en pantalla la serie de correlación de la calidad del vino con respecto a las otras variables, mostrando así la fuerza y la dirección de la correlación entre ellas.

```
calidad                1.000000
alcohol                0.453872
acido_citrico          0.102150
dioxido_azufre_libre   0.074745
sulfatos               0.059069
pH                     0.052871
azucar_residual       -0.056042
dioxido_azufre_total  -0.068406
acidez_fija           -0.094186
acidez_volatil        -0.230552
cloruros              -0.261744
densidad              -0.333244
Name: calidad, dtype: float64
```

Conclusiones

Uno de los hallazgos más importantes de nuestro análisis fue que la calidad de los vinos, con mayor frecuencia, se encuentra en el rango de 5 a 7. Esto indica que la mayoría de los vinos evaluados en nuestro conjunto de datos se consideran de calidad promedio a buena. Con la diferencia que la calidad de vinos rojos estaba más concentrada entre el rango 5 y 6; los vinos blancos mucho más concentrados en la calidad 6. Posiblemente esto último puede deberse a que en el DataSet hay más registros de vinos blancos que de vinos rojos, por lo cual también 6 es la calidad de vino en general más frecuente.

Basado en el análisis realizado, se encontró que el vino rojo, en comparación con el vino blanco, presenta características distintivas en términos de su composición química. Específicamente, se observó que el vino rojo tiene niveles más altos de densidad y acidez volátil. Es probable que la acidez volátil haya influido sensitivamente al valorar la calidad del vino rojo, que como se dijo al principio en niveles altos puede provocar un sabor desagradable.

Por otro lado, en el análisis de las muestras registradas, se observó que el vino blanco exhibe mayores niveles de ácido cítrico, cloruros, azúcar residual, dióxido de azufre libre, dióxido de azufre total y sulfatos en comparación con el vino tinto. Estos componentes juegan un papel importante en la determinación de la calidad del vino, ya que su presencia en cantidades específicas puede influir en su sabor y características organolépticas.

Sin embargo, al analizar la correlación entre estos componentes y la calidad del vino de forma individual, se encontró que no existe una relación lineal fuerte entre cada uno de ellos

y la calidad del producto. Esto sugiere que otros factores o combinaciones de componentes podrían estar influyendo en la calidad del vino.

Se recomienda realizar un análisis de correlación combinando dos o más componentes, como ácido cítrico y azúcar residual, cloruros y dióxido de azufre libre, o dióxido de azufre total y sulfatos por ejemplo, en relación con la calidad del vino. Esto permitiría una evaluación más precisa de las interacciones entre estos componentes y su impacto en la calidad global del vino. Si bien un análisis de correlación no implica causalidad, la identificación de patrones y comprensión de las relaciones entre las variables de estudio, es un buen punto de partida para futuras investigaciones que permitan estudiar estrategias orientadas a aumentar la calidad de ambas variedades de vinos.

Referencias

I. https://deepnote.com/@platzi-escuela-datos/Proyecto-Analisis-Exploratorio-de-Datos-32b92291-829a-46c8-a4bc-eebbfab60359
II. https://rstudio-pubs-static.s3.amazonaws.com/697219_e89c752713fc451b89a4e22021ff4756.html
III. https://www.youtube.com/watch?v=shcDnxhP12c
IV. https://youtube.com/playlist?list=PL3yz5wZ3_mGv3ewPYWCimw8CJpgys337e