

PONTIFICIA UNIVERSIDAD JAVERIANA

PROYECTO FINAL DE INTRODUCCIÓN A LA INTELIGENCIA ARTIFICIAL

INTEGRANTES:

SARA JULIANA PEÑA GÓMEZ

DANIELA FLOREZ

PROFESOR

JULIO OMAR PATIÑO

BOGOTÁ. 2024

1. INTRODUCCIÓN:

El objetivo de este informe es evaluar la eficacia de diferentes modelos de redes neuronales para clasificar la calidad del vino rojo y blanco. Utilizando un conjunto de datos que incluye características químicas del vino, se entrenaron y compararon tres modelos: un perceptrón, una red neuronal con una capa oculta y una red neuronal con dos capas ocultas. Cada modelo fue evaluado en términos de precisión (accuracy), precisión positiva (precision), sensibilidad (recall), y la puntuación F1 (F1 Score) para determinar su capacidad de predecir correctamente la calidad del vino como buena o mala.

2. COMPRENSIÓN DEL DATASET

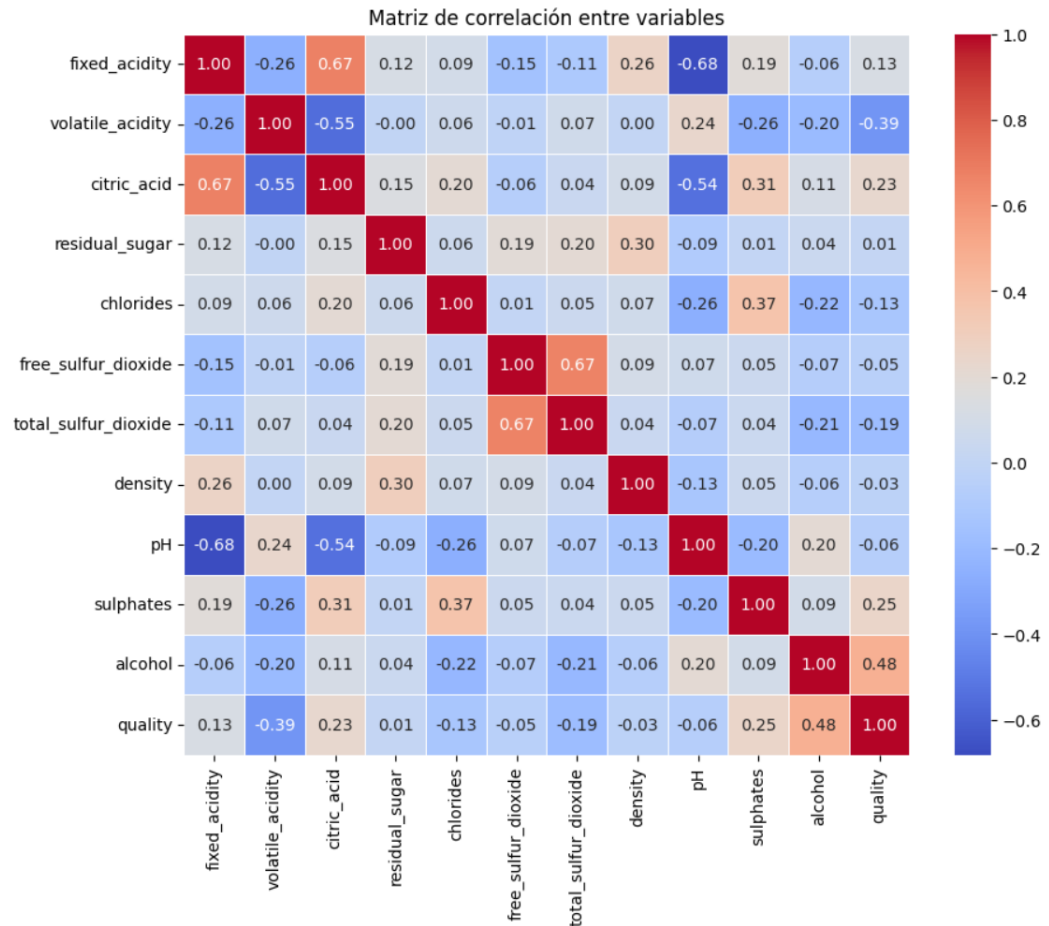
a. ¿Qué información presenta el dataset?

Este dataset presenta dos bases de datos de dos tipos de vinos: rojo y blanco. Cada uno de ellos presenta 12 variables. Estas hablan de los diferentes tipos de ácidos (fija, volátil, cítrica), también del sobrante de azúcar, cloruros, la cantidad de dióxido de sulfuro libre y el total de dióxido de sulfuro. Además, cuenta con la densidad, el pH, sulfatos, alcohol y la calidad del vino.

Todos los datos son de tipo continuo, exceptuando los datos calidad que es un entero y el color que es categórico.

b. Análisis Gráfico

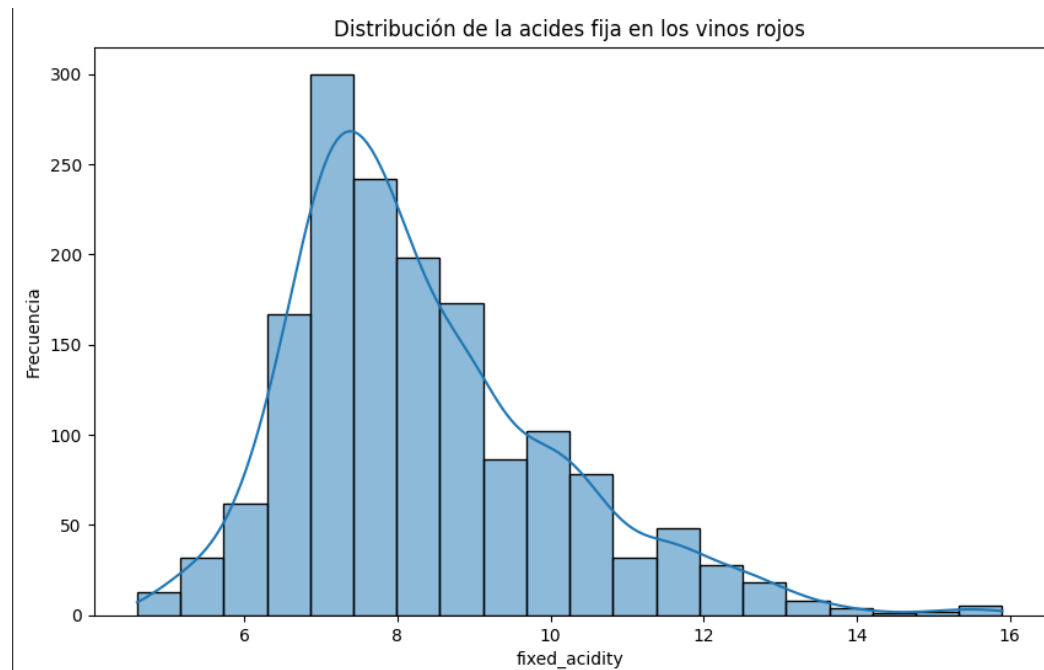
Para realizar el análisis transformamos los datos a tipo flotante para poder realizar los gráficos de forma correcta. Antes de decidir cuáles variables son aptas para el procesamiento y análisis, se realizó una matriz de correlación de ambas bases de datos. Esta es la matriz de los vinos rojos:



Gráfica 1. Matriz Correlación Vino Rojo

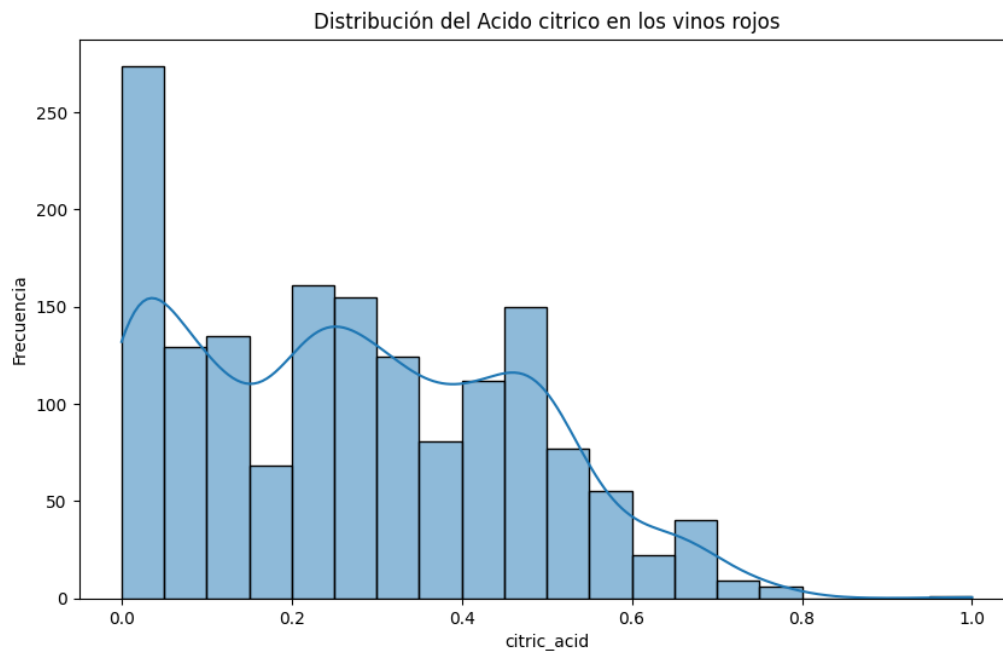
Esta matriz muestra cual es la correlación entre cada variable. Si es 1 o lo más cercano a este quiere decir que es positiva y es posible realizar una correlación entre ambas variables. Si es el caso contrario, es decir, lo más cercano a -1 o como en este caso -0.6, es negativa. Por último, si su valor esta de color gris quiere decir que es neutra por lo que su correlación no será significativa. Como se puede observar, las variables con más correlación son: “fixed_acidity” con “citric_acid”, “free_sulfur_dioxide” con “total_sulfur_dioxide”, “alcohol” con “density” y “sulphates” con “chlorides”. Sin embargo, variables como: “citric_acid” y “sulphates” tienen una buena correlación. Las variables que se van a tomar para analizar y procesar son: “fixed_acidity”, “citric_acid”, “alcohol” y “quality”.

Estas son las distribuciones de las variables escogidas:



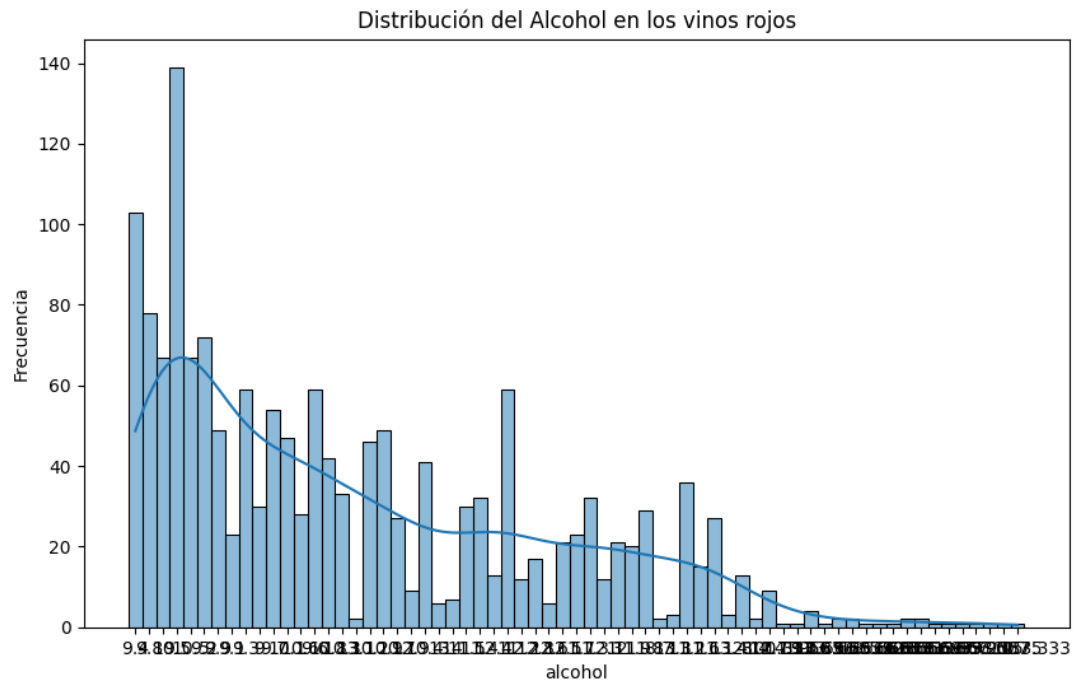
Gráfica 2. Distribución de la Acidez Fija.

La forma de la gráfica sugiere que la distribución de la acidez fija en los vinos rojos sigue una distribución aproximadamente normal (campana de Gauss), aunque con una ligera asimetría hacia la derecha. La mayor concentración de vinos tiene una acidez fija entre 7 y 8. Además, hay menos vinos con acidez fija muy alta (más de 10) o muy baja (menos de 6).



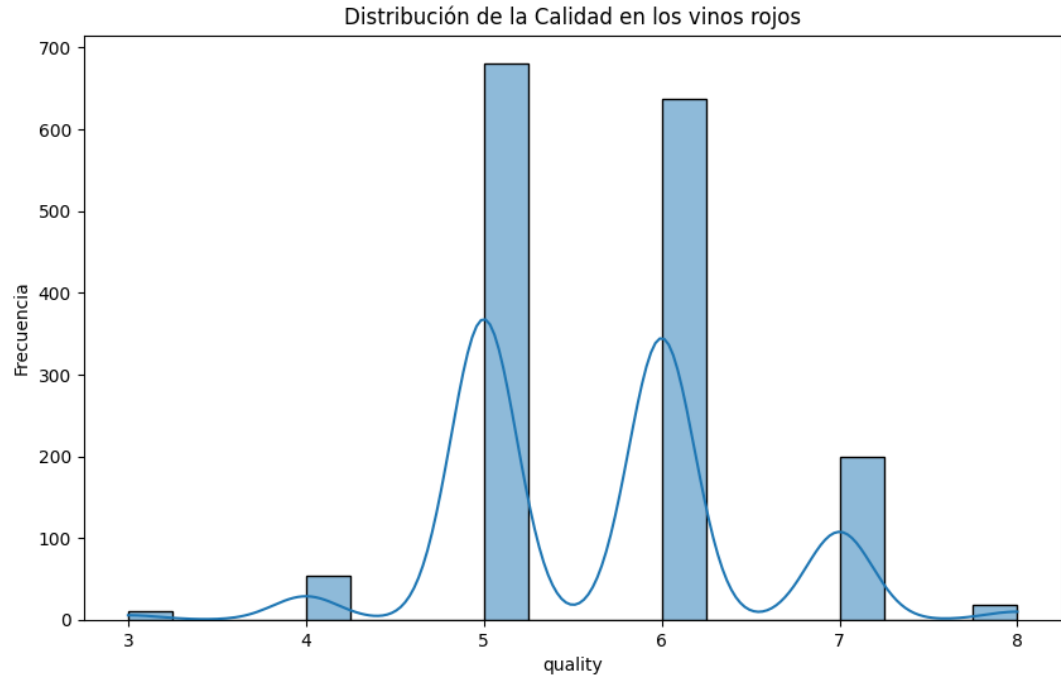
Gráfica 3. Distribución Ácido Cítrico

Esta gráfica muestra una distribución más asimétrica, con un claro sesgo hacia la derecha. La mayor parte de los vinos tienen un contenido de ácido cítrico muy bajo, alrededor de 0 a 0.1. A medida que aumenta el contenido de ácido cítrico, la frecuencia de los vinos disminuye.



Gráfica 4. Distribución del Alcohol en Vinos Rojos

La forma de esta grafica muestra un claro sesgo hacia la derecha, sin embargo, al tener tantos datos y con valores altos, no es posible determinar de forma clara cuál es la tendencia en los vinos rojos.

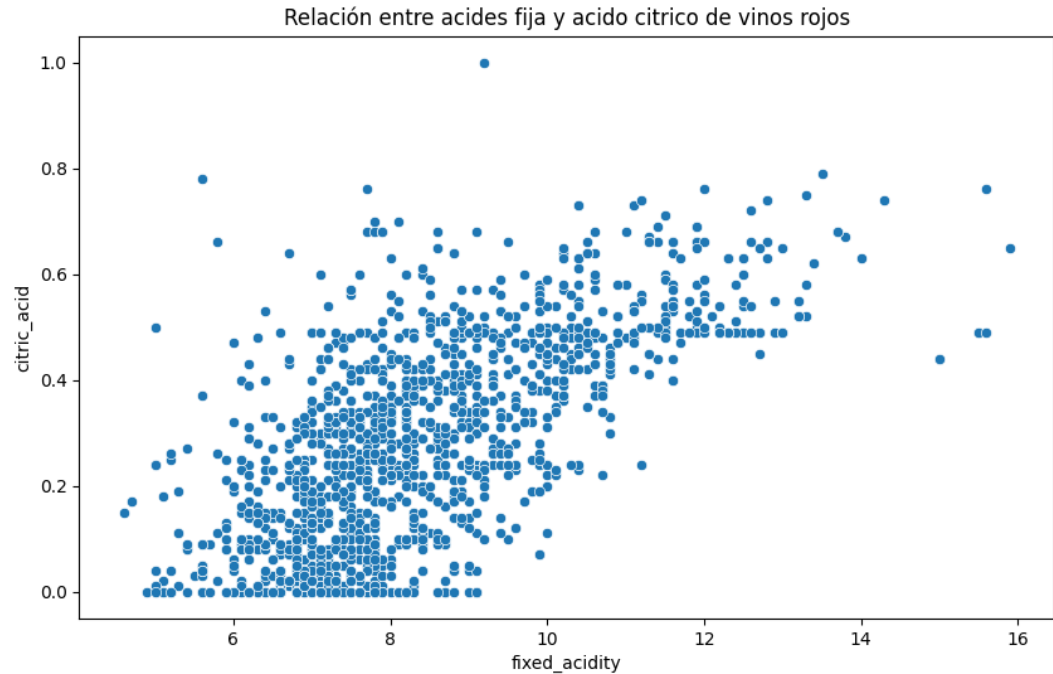


Gráfica 5. Histograma Distribución de la Calidad en Vinos Rojos

De esta gráfica, podemos interpretar que:

- No existen vinos tintos con una puntuación de calidad de 1 o 6.
- La puntuación de calidad más común para los vinos tintos es 7, con una frecuencia de 700.
- El segundo puntaje de calidad más común para los vinos tintos es 5, con una frecuencia de 500.
- El puntaje de calidad menos común para los vinos tintos es 0, con una frecuencia de 3.

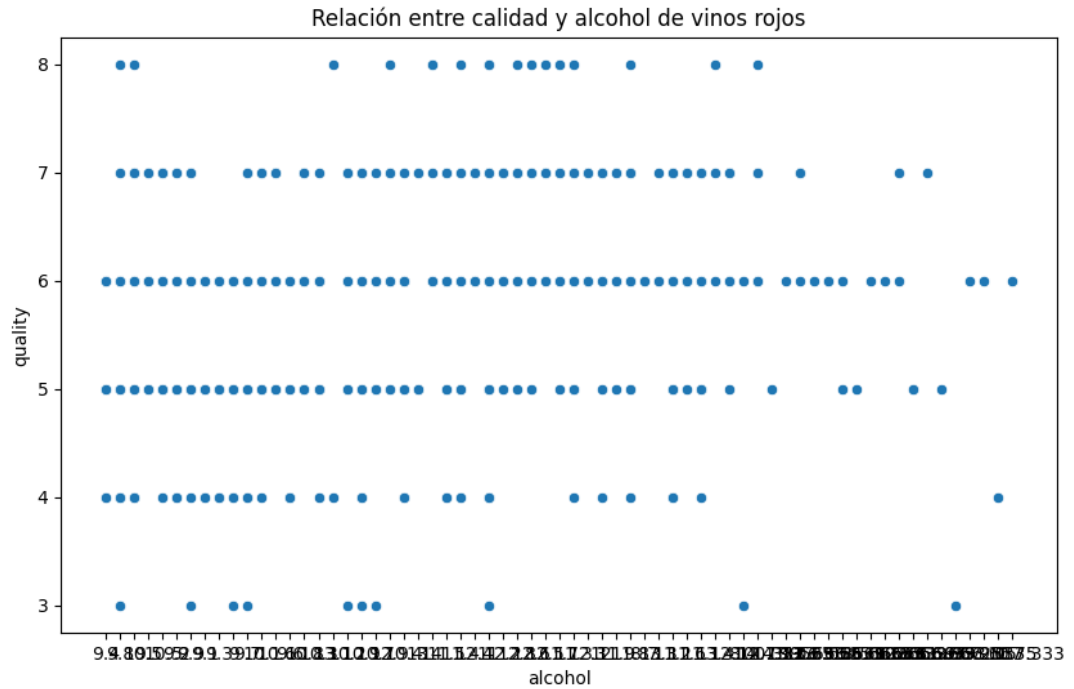
Las siguientes gráficas, muestran la correlación entre las variables escogidas. Este método es para asegurar y comprobar que los datos de la matriz son correctos y se puede hacer un análisis correcto entre las variables seleccionadas.



Gráfica 6. Correlación entre Acides Fija y Acides Cítrico.

La relación entre la acidez fija y la acidez total es importante en la elaboración del vino porque puede afectar el sabor, el equilibrio y la estabilidad del vino. Los niveles de acidez fija más altos pueden contribuir a la acidez y la frescura de un vino, mientras que los niveles de acidez total más altos pueden hacer que un vino se sienta más refrescante y vivaz en el paladar.

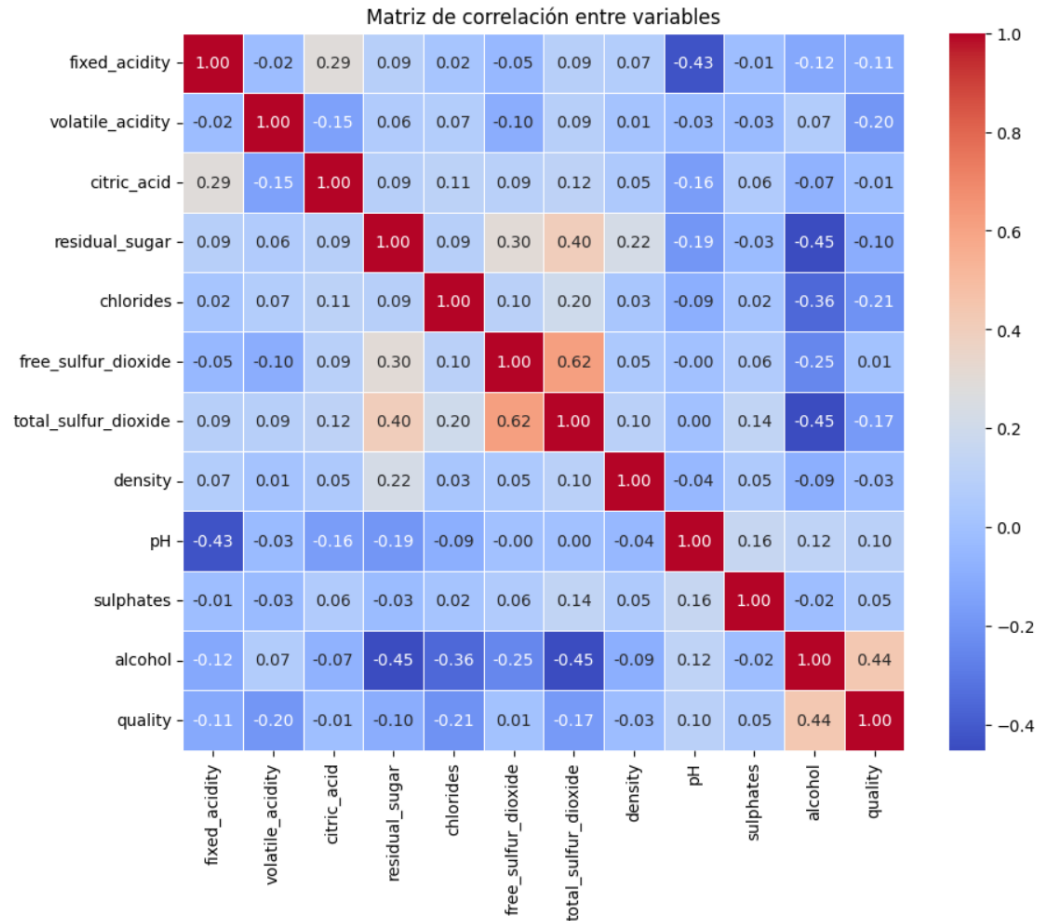
Esta gráfica muestra una correlación positiva baja., podemos observar que a medida que disminuye la acidez fija, la acidez total también tiende a disminuir. Esto sugiere que los vinos con niveles fijos de acidez más bajos pueden tener un sabor ácido menos pronunciado y resultar menos refrescantes en el paladar.



Gráfica 7. Correlación entre Alcohol y Calidad.

En esta gráfica, muestra la relación entre la calidad del vino y el contenido de alcohol. Sin embargo, al no ser los datos de forma clara, muestra la idea de que los vinos de mayor calidad tienden a tener un cierto rango de contenido alcohólico, o que los vinos con un contenido alcohólico específico tienen más probabilidades de ser de alta calidad.

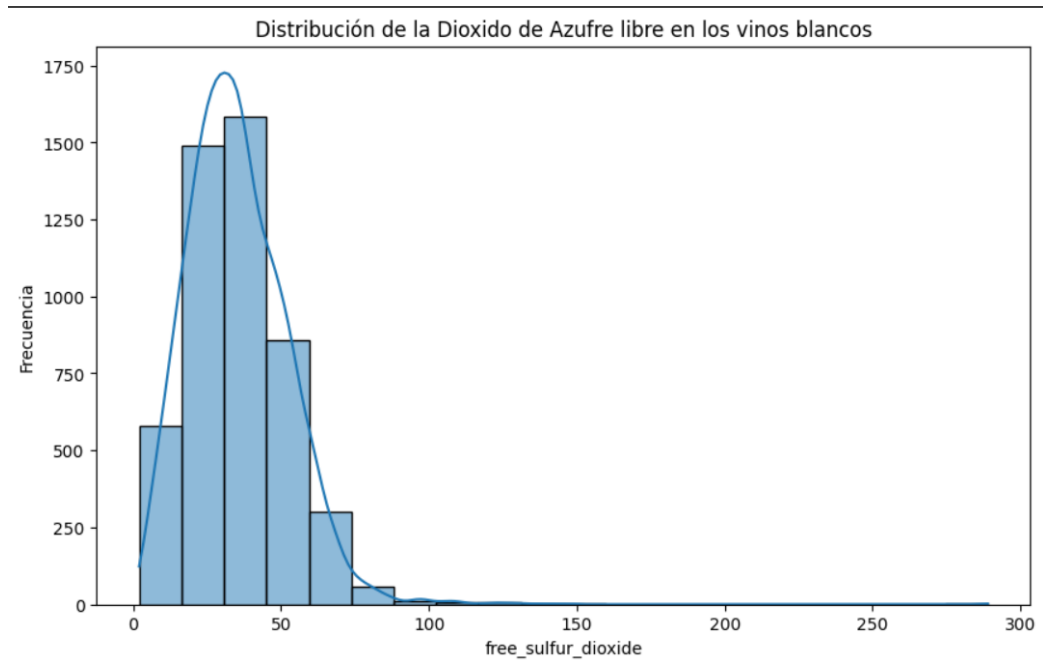
Vinos Blancos



Gráfica 8. Matriz de Correlación.

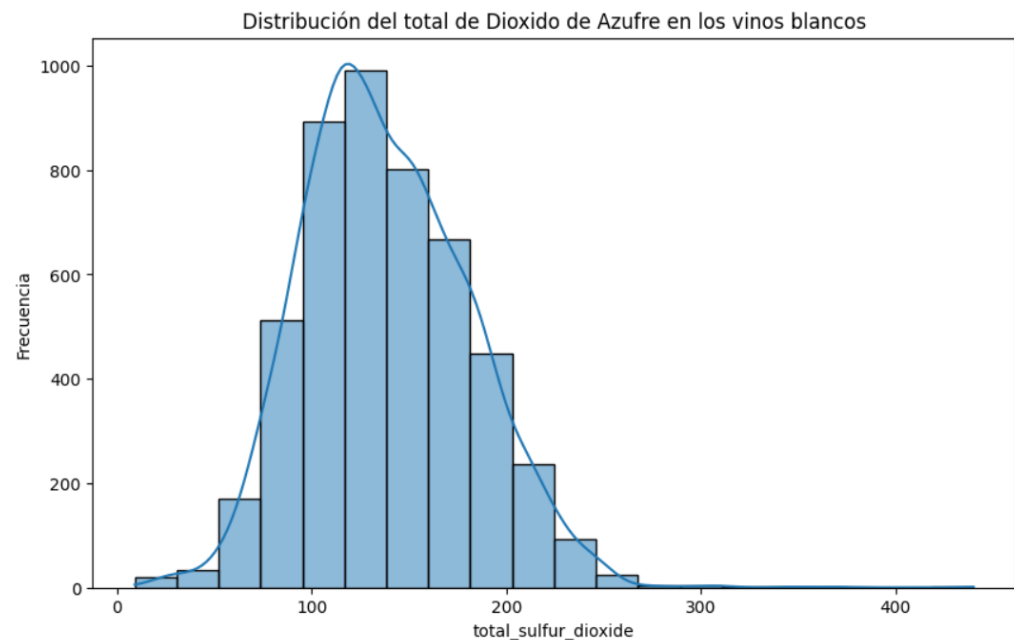
En este caso, no se pueden escoger las mismas variables, ya que su correlación es diferente. En este caso, las variables a escoger son: “total_sulfur_dioxide”, “free_sulfur_dioxide”, “alcohol” y “quality”

Estas son las distribuciones de las variables escogidas:



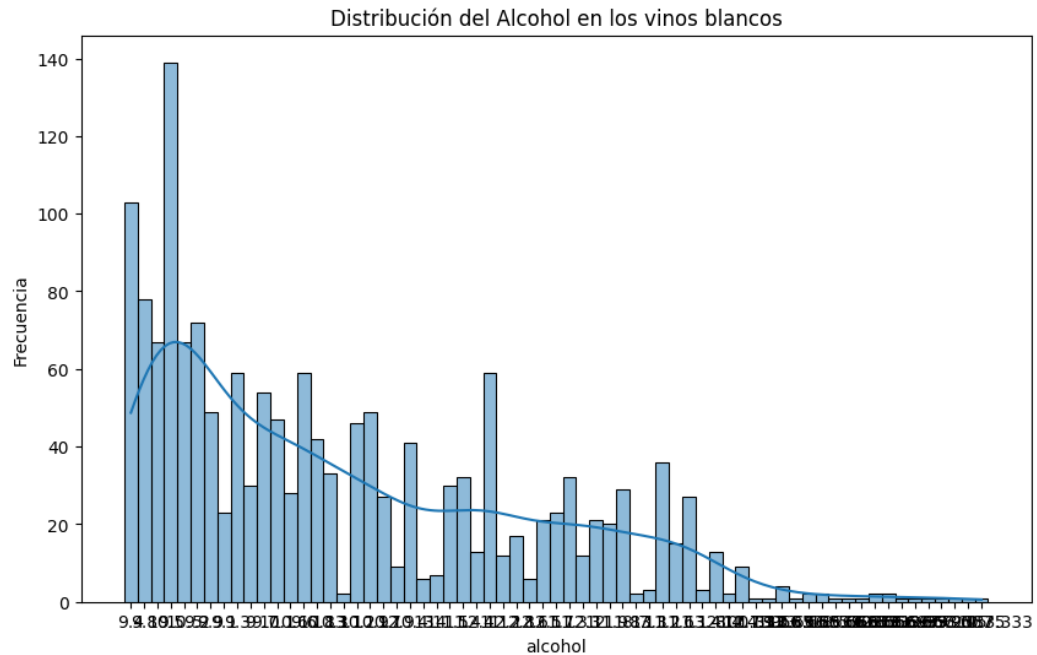
Grafica 9. Histograma Dióxido de Azufre Libre

El histograma muestra una distribución ligeramente sesgada a la derecha, con un pico de alrededor de 30-40 mg/L. Esto indica que la mayoría de los vinos blancos tienden a tener niveles de dióxido de azufre libre dentro de este rango.



Grafica 10. Histograma Total Dióxido de Azufre

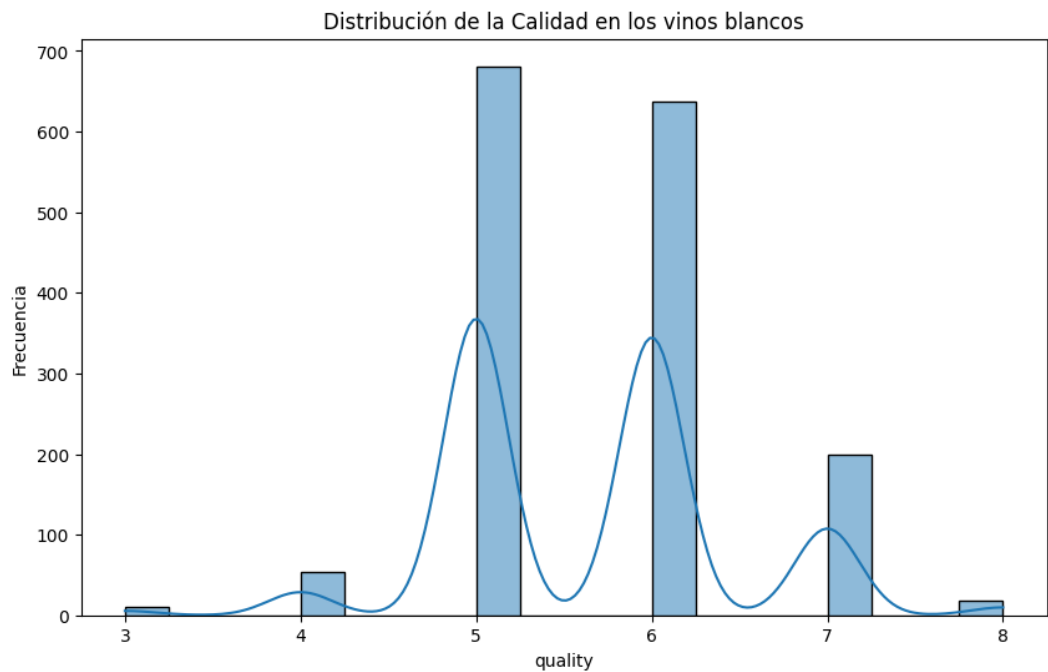
Este histograma también muestra una distribución sesgada a la derecha, con un pico ligeramente más alto, alrededor de 100-120 mg/L para el dióxido de azufre total.



Grafica 11. Histograma Alcohol

La mayoría de los vinos blancos de este conjunto de datos tienen un contenido de alcohol entre el 9% y el 13%.

Hay un pico de alrededor del 9,8% de alcohol, lo que indica que se trata de un valor común. La distribución está ligeramente sesgada hacia la derecha, con menos vinos con niveles de alcohol más altos.



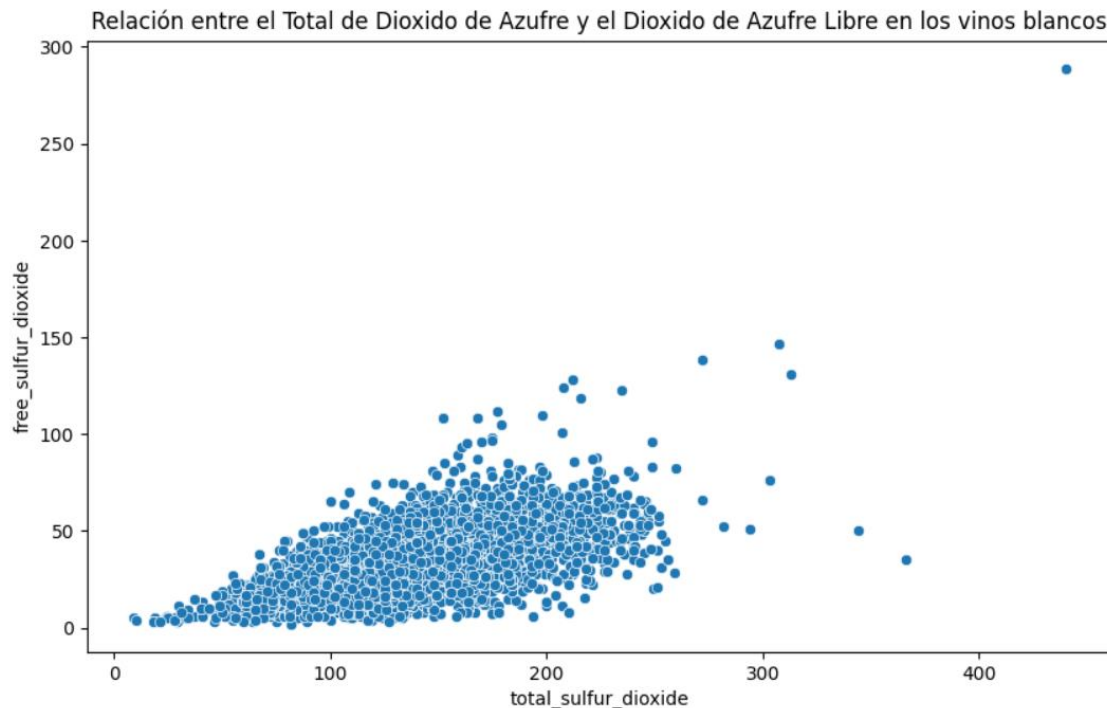
Gráfica 12. Histograma Calidad

La mayoría de los vinos se agrupan en torno a calificaciones de calidad de 5 y 6.

La distribución sugiere que las puntuaciones de calidad 5 y 6 son las más habituales para estos vinos blancos.

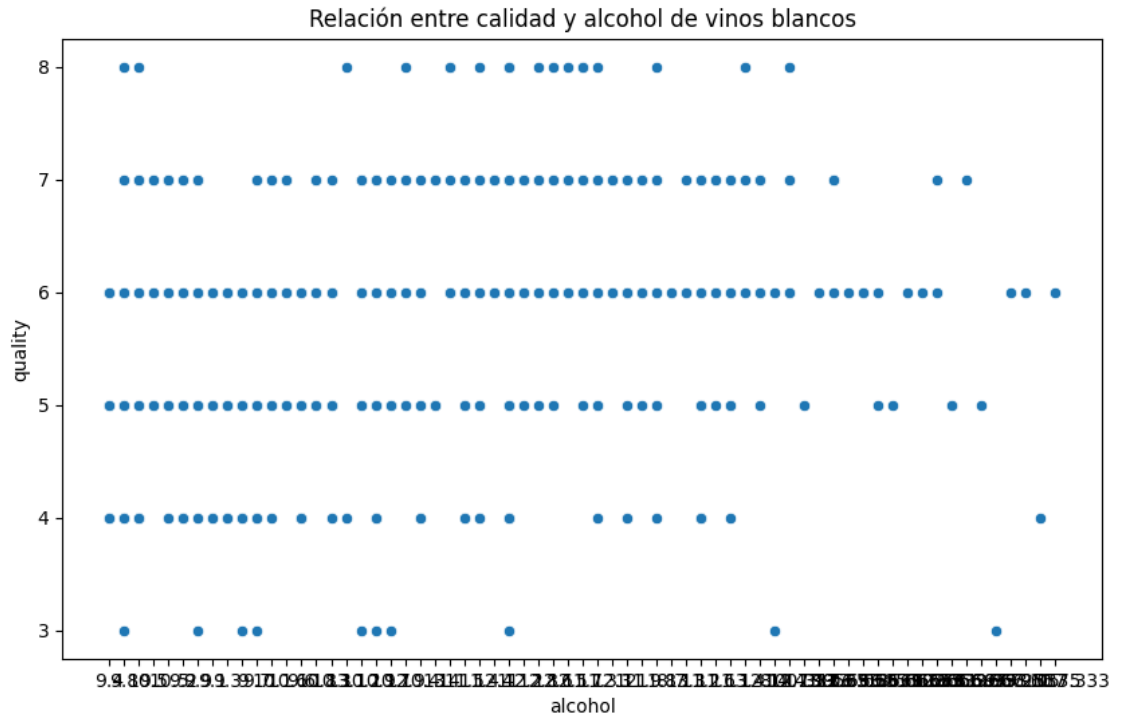
Muy pocos vinos recibieron puntuaciones en los extremos de la escala de calidad (3, 4, 7, 8).

Las siguientes gráficas muestran la correlación de las variables escogidas para los vinos blancos como en los vinos rojos:



Gráfica 13. Correlación entre Dióxido de Azufre Libre y Dióxido de Azufre Total.

Se observa una correlación positiva entre ambas variables. Esto significa que, en general, a medida que aumenta el dióxido de azufre total, también aumenta el dióxido de azufre libre. Los puntos no se ajustan perfectamente a una línea recta, lo que indica que la relación entre las variables no es perfectamente lineal. Hay una dispersión considerable de los datos, especialmente a medida que aumenta el dióxido de azufre total. Además, hay algunos valores atípicos en la gráfica, particularmente en la parte superior derecha, donde hay algunos vinos con niveles muy altos de dióxido de azufre total y libre. El dióxido de azufre se utiliza comúnmente en la elaboración del vino como conservante y antioxidante.



Gráfica 14. Correlación entre Acides Fija y Acides Cítrico.

Esta gráfica, tiene la misma tendencia y relación a la de los vinos rojos.

3. PROCESAMIENTO DE LA INFORMACIÓN

- Vinos Rojos:

Se usarán dos métodos para realizar un mejor procesamiento de información. Los métodos son: Dumificación de Variables Categóricas y Búsqueda de Datos Atípicos. La dumificación de variables categóricas es el proceso de convertir estas variables en una forma que pueda ser proporcionada a algoritmos de aprendizaje automático. Como es el caso de la variable calidad. Al tener tantos valores se decidió que un vino es de alta calidad si el valor de este es mayor a 5, el cual va a tomar el valor de 1, mientras si es menor a 5, tomara el valor de 0. Esto con el fin de que el entrenamiento del modelo sea más sencillo y los resultados más fáciles de entender. En el caso de la búsqueda de datos atípicos son datos u observaciones que se desvían significativamente del resto de los datos en un conjunto de datos. La búsqueda de datos atípicos implica identificar y manejar estos valores para evitar que distorsionen el análisis o los modelos de aprendizaje automático. Cómo se observa en la gráfica de la distribución del alcohol y de la relación entre la calidad y el alcohol.

Por otro lado, tenemos las otras variables: Acides Fija y Acides Cítrica. Para estas se tratará la Búsqueda de Datos Atípicos. Para poder obtener un mejor análisis sobre como estas variables afectan al vino rojo.

- Vinos Blancos

Se realizará una normalización de los datos para las variables “alcohol” y “quality”. Por otro lado, se realizará una Búsqueda de datos atípicos para las variables de sulfato.

4. CONSTRUCCIÓN DEL DATASET

- a. ¿Qué proporción de conjunto de entrenamiento y de pruebas?

La proporción seleccionada es 80-20. Se considera que, de esta forma, el 80% es para el entrenamiento de la red neuronal y el 20% es para pruebas. Se selecciono esta proporción de acuerdo con el tamaño de los datos ya que cada base de datos presenta mas o menos 1600 valores.

- b. ¿Cómo cambia el modelo si existen cambios en las proporciones de los dos conjuntos de trabajo?

Al cambiar la proporción del modelo puede afectar la capacidad de aprendizaje del modelo y la capacidad de generalización. Sin embargo, con esta proporción escogida se puede asegurar que el tamaño de datos que hacen parte del entrenamiento permite una mejor evaluación, mejorando su capacidad para capturar patrones y relaciones en los datos. Y el contar con el 20% de los datos para pruebas, permite una estimación fiable del rendimiento del modelo, por lo cual permite una evaluación robusta y minimiza el riesgo de sobreajuste en el tiempo de ejecución.

5. ELABORACIÓN DEL MODELO VINOS ROJOS

1. Preparación de los Datos:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Cargamos los datos
vino_rojo = pd.read_csv('winequality-red.csv', delimiter=';')
vino_blanco = pd.read_csv('winequality-white.csv', delimiter=';')

# Convertir columnas numéricas y manejar errores
datos_vinoRojo_numericos = vino_rojo.apply(pd.to_numeric, errors='coerce')

# Eliminar filas con NaN
datos_vinos_numericos = datos_vinoRojo_numericos.dropna()

# Convertir columnas numéricas y manejar errores
datos_vinoBlanco_numericos = vino_blanco.apply(pd.to_numeric, errors='coerce')

# Eliminar filas con NaN
datos_vinoB_numericos = datos_vinoBlanco_numericos.dropna()
```

Imagen 1. Carga y limpieza de archivos.

Se cargan dos conjuntos de datos desde archivos CSV. Luego, se intenta convertir todas las columnas a tipo numérico, reemplazando los valores no convertibles con “NaN”. Y se eliminan las columnas que tengan esta característica.

2. Búsqueda de valores Atípicos en Alcohol y Especificamos cuales variables van a ser entrenadas (Calidad y Alcohol)

```

import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Cargar los datos
data = pd.read_csv('winequality-red.csv', delimiter=';')

# Asegúrate de que la columna 'alcohol' sea numérica
data['alcohol'] = pd.to_numeric(data['alcohol'], errors='coerce')

# Eliminar filas con valores NaN en la columna 'alcohol'
data = data.dropna(subset=['alcohol'])

# Calcular IQR y eliminar outliers
Q1 = data['alcohol'].quantile(0.25)
Q3 = data['alcohol'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
data_filtered = data[(data['alcohol'] >= lower_bound) & (data['alcohol'] <= upper_bound)]

```

Imagen 2. Carga y limpieza de archivos y de variables.

El código proporciona un proceso completo de preprocesamiento de datos y preparación para el entrenamiento de un modelo de aprendizaje automático utilizando un conjunto de datos de calidad del vino tinto. Primero, se cargan los datos y se asegura que la columna 'alcohol' sea numérica, eliminando cualquier fila con valores nulos en esta columna. Luego, se eliminan los valores atípicos en la columna 'alcohol' usando el rango intercuartílico (IQR). Después, la columna 'alcohol' se utiliza como característica y la columna 'quality' se convierte en una tarea binaria (vino bueno o malo), basada en si la calidad es mayor a 5. Finalmente, los datos se dividen en conjuntos de entrenamiento y prueba utilizando `train_test_split` de scikit-learn, asegurando que un 20% de los datos se reserven para pruebas, y se imprimen las formas de los conjuntos resultantes para verificación.

```

# La columna característica es 'alcohol' y la columna objetivo es 'quality'
X = data_filtered[['alcohol']].astype(float)
y = data_filtered['quality'].astype(int)

# Convertimos la calidad a una tarea binaria (bueno o malo)
y = (y > 5).astype(int)

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

# Verificar las formas de los conjuntos
print(f'X_train shape: {X_train.shape}, y_train shape: {y_train.shape}')
print(f'X_test shape: {X_test.shape}, y_test shape: {y_test.shape}')

```

Imagen 3. Definición de variables.

Establece las entradas (X) y salidas (y) del modelo de aprendizaje automático. En concreto, está seleccionando la columna 'alcohol' del DataFrame `data_filtered` como la variable característica (X), que se utilizará para predecir la calidad del vino. Esta columna se convierte

a tipo float para asegurar que todos los valores sean números decimales, adecuados para cálculos numéricos. Al mismo tiempo, la columna 'quality' se selecciona como la variable objetivo (y), que es la categoría o valor que queremos predecir. Esta columna se convierte a tipo int para asegurar que los valores sean enteros, apropiados para una variable de clasificación.

a. Definición de Modelos de Redes Neuronales

```
X_train shape: (1264, 1), y_train shape: (1264,)
X_test shape: (317, 1), y_test shape: (317,)
```

Muestra la proporción de los datos: 80-20

Imagen 4. Proporción de los datos.

```
# Definir los modelos
def create_perceptron():
    model = Sequential([
        Dense(units=1, activation='sigmoid', input_shape=(1,), use_bias=True)
    ])
    return model

def create_single_hidden():
    model = Sequential([
        Dense(units=1, activation='sigmoid', input_shape=(1,), use_bias=True),
        Dense(units=1, activation='sigmoid', use_bias=True)
    ])
    return model

def create_double_hidden():
    model = Sequential([
        Dense(units=2, activation='sigmoid', input_shape=(1,), use_bias=True),
        Dense(units=2, activation='sigmoid', use_bias=True),
        Dense(units=1, activation='sigmoid', use_bias=True)
    ])
    return model

# Lista de modelos
models = [
    ('Perceptrón', create_perceptron),
    ('Una capa oculta', create_single_hidden),
    ('Dos capas ocultas', create_double_hidden)
]

# Diccionario para almacenar resultados
results = {}

# Entrenar y evaluar cada modelo
for name, create_model in models:
    model = create_model()
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=10, batch_size=1, verbose=1)
    y_pred = model.predict(X_test)
    y_pred = (y_pred > 0.5).astype(int)
```

Imagen 5. Definición de los modelos.

El código define y evalúa tres tipos de modelos de redes neuronales para una tarea de clasificación binaria: un perceptrón simple, un modelo con una capa oculta y un modelo con dos capas ocultas. Cada modelo se define usando Sequential de tensorflow.keras, con capas Dense y activación 'sigmoid'. Los modelos se compilan con el optimizador 'adam' y la función de pérdida 'binary_crossentropy', y luego se entrenan con los datos de entrenamiento durante 10 épocas. Después del entrenamiento, se generan predicciones para los datos de prueba y se

convierten en valores binarios (0 o 1). Se calculan diversas métricas de evaluación (exactitud, precisión, recall, F1-score y matriz de confusión) para cada modelo y se almacenan en un diccionario llamado results para compararlos.

```
# Entrenar y evaluar cada modelo
for name, create_model in models:
    model = create_model()
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=10, batch_size=1, verbose=1)
    y_pred = model.predict(X_test)
    y_pred = (y_pred > 0.5).astype(int)

    # Calcular métricas
    acc = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Guardar resultados
    results[name] = {
        'accuracy': acc,
        'precision': precision,
        'recall': recall,
        'f1_score': f1,
        'conf_matrix': conf_matrix
    }
```

Imagen 6. Entrenamiento y evaluación de cada modelo.

Es importante recalcar cual es la característica principal de cada modelo y su función dentro del proyecto:

- El perceptrón es el modelo más simple, con una única capa de salida. Este tipo de modelo es capaz de resolver problemas linealmente separables, lo que significa que puede encontrar una línea (o un plano en dimensiones más altas) que separa dos clases. Sin embargo, su capacidad para capturar relaciones complejas entre las características y las etiquetas es limitada debido a su simplicidad.
- El Modelo con Capa Oculta, introduce una capa oculta con una neurona adicional entre la entrada y la salida. La capa oculta permite al modelo aprender representaciones más complejas y no lineales de los datos, lo que puede mejorar la capacidad del modelo para capturar patrones que un perceptrón simple no podría.
- El Modelo con Dos Capas Ocultas, cada una con dos neuronas. La adición de más capas y neuronas permite al modelo aprender funciones de mapeo mucho más complejas y capturar relaciones no lineales profundas entre las características y las etiquetas. Este tipo de modelo puede capturar estructuras más sofisticadas en los datos, lo que puede conducir a un mejor rendimiento en tareas complicadas.

3. Búsqueda de Valores Atípicos en las Variables de Ácido Fijo y Ácido Cítrico.

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Cargar los datos
data = pd.read_csv('winequality-red.csv', delimiter=';')

# Función para identificar outliers usando el método IQR
def identify_outliers(df, feature):
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[feature] < lower_bound) | (df[feature] > upper_bound)]
    return outliers

# Identificar outliers en 'fixed_acidity'
outliers_fixed_acidity = identify_outliers(data, 'fixed_acidity')
print(f'Outliers in fixed_acidity:\n{outliers_fixed_acidity}')

# Identificar outliers en 'citric_acid'
outliers_citric_acid = identify_outliers(data, 'citric_acid')
print(f'Outliers in citric_acid:\n{outliers_citric_acid}')

# Visualización de outliers
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.boxplot(y=data['fixed_acidity'])
plt.title('Fixed Acidity Outliers')

plt.subplot(1, 2, 2)
sns.boxplot(y=data['citric_acid'])
plt.title('Citric Acid Outliers')

plt.show()

```

Imagen 7. Carga y limpieza de variables.

Se repite el mismo proceso que en las variables anteriores. Esto se realiza con la finalidad de que las redes puedan tener un mejor campo de entrenamiento y no se consuman tantos recursos mientras esto ocurre.

```

# Eliminar outliers en 'fixed_acidity' y 'citric_acid'
def remove_outliers(df, feature):
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[feature] >= lower_bound) & (df[feature] <= upper_bound)]
    return df

data_cleaned = remove_outliers(data, 'fixed_acidity')
data_cleaned = remove_outliers(data_cleaned, 'citric_acid')

# Seleccionar las características y la columna objetivo
X = data[['fixed_acidity', 'citric_acid']]
y = data['quality']

# Convertir la calidad a una tarea binaria (bueno o malo)
y = (y > 5).astype(int)

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

# Verificar las formas de los conjuntos
print(f'X_train shape: {X_train.shape}, y_train shape: {y_train.shape}')
print(f'X_test shape: {X_test.shape}, y_test shape: {y_test.shape}')

```

Imagen 8. Carga y limpieza de archivos.

El proceso es el mismo, sin embargo, se gráfica y se muestra como es el comportamiento de las variables al hacer un tratamiento de datos específico.

a. Definición y Entrenamiento de las Redes Neuronales

```

def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=10, batch_size=1, verbose=0)
    y_pred = (model.predict(X_test) > 0.5).astype(int)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    return accuracy, precision, recall, f1, cm

# Perceptrón
model_perceptron = Sequential([
    Dense(units=1, activation='sigmoid', input_shape=(2,), use_bias=True)
])
accuracy_perceptron, precision_perceptron, recall_perceptron, f1_perceptron, cm_perceptron = train_and_evaluate_model(model_perceptron, X_train, y_train, X_test, y_test)
print(f'Perceptrón:\nAccuracy: {accuracy_perceptron}\nPrecision: {precision_perceptron}\nRecall: {recall_perceptron}\nF1 Score: {f1_perceptron}\nConfusion Matrix:\n{cm_perceptron}\n')

# Red neuronal con una capa oculta
model_one_hidden = Sequential([
    Dense(units=2, activation='relu', input_shape=(2,), use_bias=True),
    Dense(units=1, activation='sigmoid', use_bias=True)
])
accuracy_one_hidden, precision_one_hidden, recall_one_hidden, f1_one_hidden, cm_one_hidden = train_and_evaluate_model(model_one_hidden, X_train, y_train, X_test, y_test)
print(f'Una capa oculta:\nAccuracy: {accuracy_one_hidden}\nPrecision: {precision_one_hidden}\nRecall: {recall_one_hidden}\nF1 Score: {f1_one_hidden}\nConfusion Matrix:\n{cm_one_hidden}\n')

# Red neuronal con dos capas ocultas
model_two_hidden = Sequential([
    Dense(units=2, activation='relu', input_shape=(2,), use_bias=True),
    Dense(units=2, activation='relu', use_bias=True),
    Dense(units=1, activation='sigmoid', use_bias=True)
])
accuracy_two_hidden, precision_two_hidden, recall_two_hidden, f1_two_hidden, cm_two_hidden = train_and_evaluate_model(model_two_hidden, X_train, y_train, X_test, y_test)
print(f'Dos capas ocultas:\nAccuracy: {accuracy_two_hidden}\nPrecision: {precision_two_hidden}\nRecall: {recall_two_hidden}\nF1 Score: {f1_two_hidden}\nConfusion Matrix:\n{cm_two_hidden}\n')

```

Imagen 9. Definición del modelo y entrenamiento de las redes.

La función `train_and_evaluate_model` centraliza el proceso de compilación, entrenamiento y evaluación de los modelos. Configura el modelo con el optimizador 'adam' y la función de pérdida 'binary_crossentropy', y establece 'accuracy' como métrica de evaluación. Entrena el modelo usando los datos de entrenamiento (`X_train` y `y_train`) durante 10 épocas con un tamaño de lote de 1, sin mostrar la salida de progreso (`verbose=0`). Realiza predicciones sobre los datos de prueba (`X_test`) y las convierte en valores binarios (0 o 1) usando un umbral de 0.5. Calcula varias métricas de rendimiento (exactitud, precisión, recall, F1-score

y matriz de confusión) usando las predicciones y las etiquetas verdaderas (y_{test}). Por último, Devuelve las métricas calculadas.

Luego, tres modelos diferentes se crean y evalúan: un perceptrón simple, una red neuronal con una capa oculta y otra con dos capas ocultas. Cada uno de estos modelos se entrena con los mismos datos y se evalúa con las mismas métricas, lo que permite una comparación clara de su rendimiento en términos de exactitud, precisión, recall, F1-score y matriz de confusión.

VINOS BLANCOS:

1. Búsqueda de Valores Atípicos en las variables: Dióxido de Azufre Libre y Dióxido de Azufre Total.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Cargar los datos
data = pd.read_csv('winequality-white.csv', delimiter=';')

# Función para identificar outliers usando el método IQR
def identify_outliers(df, feature):
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    outliers = df[(df[feature] < lower_bound) | (df[feature] > upper_bound)]
    return outliers

# Identificar outliers en 'azufre libre'
outliers_free_sulfur_dioxide = identify_outliers(data, 'free_sulfur_dioxide')
print(f'Outliers in free_sulfur_dioxide:\n{outliers_free_sulfur_dioxide}')

# Identificar outliers en 'azufre total'
outliers_total_sulfur_dioxide = identify_outliers(data, 'total_sulfur_dioxide')
print(f'Outliers in total_sulfur_dioxide:\n{outliers_total_sulfur_dioxide}')

# Visualización de outliers
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
sns.boxplot(y=data['free_sulfur_dioxide'])
plt.title('Free Sulfur Dioxide Outliers')

plt.subplot(1, 2, 2)
sns.boxplot(y=data['total_sulfur_dioxide'])
plt.title('Total Sulfur Dioxide Outliers')

plt.show()
```

Imagen 10. Carga y limpieza de variables.

Se realiza el mismo proceso que en los vinos rojos. Se especifica cuales variables van a ser parte del entrenamiento de los modelos.

```
# Eliminar outliers en 'fixed_acidity' y 'citric_acid'
def remove_outliers(df, feature):
    Q1 = df[feature].quantile(0.25)
    Q3 = df[feature].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[feature] >= lower_bound) & (df[feature] <= upper_bound)]
    return df

data_cleaned = remove_outliers(data, 'free_sulfur_dioxide')
data_cleaned = remove_outliers(data_cleaned, 'total_sulfur_dioxide')

# Seleccionar las características y la columna objetivo
X = data[['free_sulfur_dioxide', 'total_sulfur_dioxide']]
y = data['quality']

# Convertir la calidad a una tarea binaria (bueno o malo)
y = (y > 5).astype(int)

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

# Verificar las formas de los conjuntos
print(f'X_train shape: {X_train.shape}, y_train shape: {y_train.shape}')
print(f'X_test shape: {X_test.shape}, y_test shape: {y_test.shape}')
```

Imagen 11. Búsqueda de datos atípicos en las variables.

a. Definición y Entrenamiento de las Redes Neuronales

```
X_train shape: (3891, 1), y_train shape: (3891,)
X_test shape: (973, 1), y_test shape: (973,)
```

Imagen 12. Proporción del modelo de los vinos blancos.

Se muestra la proporción 80-20 usada en este modelo y se ve que la cantidad de datos es bastante mayor que la de los vinos rojos.

```
def train_and_evaluate_model(model, X_train, y_train, X_test, y_test):
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=10, batch_size=1, verbose=0)
    y_pred = (model.predict(X_test) > 0.5).astype(int)
    accuracy = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    cm = confusion_matrix(y_test, y_pred)
    return accuracy, precision, recall, f1, cm

# Perceptrón
model_perceptron = Sequential([
    Dense(units=1, activation='sigmoid', input_shape=(2,)), use_bias=True
])
accuracy_perceptron, precision_perceptron, recall_perceptron, f1_perceptron, cm_perceptron = train_and_evaluate_model(model_perceptron, X_train, y_train, X_test, y_test)
print(f'Perceptrón:\nAccuracy: {accuracy_perceptron}\nPrecision: {precision_perceptron}\nRecall: {recall_perceptron}\nF1 Score: {f1_perceptron}\nConfusion Matrix:\n{cm_perceptron}\n')

# Red neuronal con una capa oculta
model_one_hidden = Sequential([
    Dense(units=2, activation='relu', input_shape=(2,)), use_bias=True,
    Dense(units=1, activation='sigmoid', use_bias=True)
])
accuracy_one_hidden, precision_one_hidden, recall_one_hidden, f1_one_hidden, cm_one_hidden = train_and_evaluate_model(model_one_hidden, X_train, y_train, X_test, y_test)
print(f'Una capa oculta:\nAccuracy: {accuracy_one_hidden}\nPrecision: {precision_one_hidden}\nRecall: {recall_one_hidden}\nF1 Score: {f1_one_hidden}\nConfusion Matrix:\n{cm_one_hidden}\n')

# Red neuronal con dos capas ocultas
model_two_hidden = Sequential([
    Dense(units=2, activation='relu', input_shape=(2,)), use_bias=True,
    Dense(units=2, activation='relu', use_bias=True),
    Dense(units=1, activation='sigmoid', use_bias=True)
])
accuracy_two_hidden, precision_two_hidden, recall_two_hidden, f1_two_hidden, cm_two_hidden = train_and_evaluate_model(model_two_hidden, X_train, y_train, X_test, y_test)
print(f'Dos capas ocultas:\nAccuracy: {accuracy_two_hidden}\nPrecision: {precision_two_hidden}\nRecall: {recall_two_hidden}\nF1 Score: {f1_two_hidden}\nConfusion Matrix:\n{cm_two_hidden}\n')
```

Imagen 13. Definición y entrenamiento de los modelos.

Se implemento la misma estrategia usada en las variables de acides con el vino rojo. Aunque su implementación se vea diferente frente a las variables de alcohol y calidad tienen la misma finalidad.

2. Búsqueda de Valores Atípicos en las Variables de Alcohol y Calidad

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, accuracy_score, precision_score, recall_score, f1_score
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Cargar datos
data = pd.read_csv('winequality-white.csv', delimiter=';')

# Asegurarse de que los datos sean numéricos
data['alcohol'] = pd.to_numeric(data['alcohol'], errors='coerce')
data['quality'] = pd.to_numeric(data['quality'], errors='coerce')

# Eliminar filas con valores NaN en la columna 'alcohol'
data = data.dropna(subset=['alcohol'])

# Calcular IQR y eliminar outliers
Q1 = data['alcohol'].quantile(0.25)
Q3 = data['alcohol'].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR
data_filtered = data[(data['alcohol'] >= lower_bound) & (data['alcohol'] <= upper_bound)]

# La columna característica es 'alcohol' y la columna objetivo es 'quality'
X = data[['alcohol']]
y = data['quality']

# Convertimos la calidad a una tarea binaria (bueno o malo)
y = (y > 5).astype(int)

# Dividir el conjunto de datos en entrenamiento y prueba (80-20)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Verificar las formas de los conjuntos
print(f'X_train shape: {X_train.shape}, y_train shape: {y_train.shape}')
print(f'X_test shape: {X_test.shape}, y_test shape: {y_test.shape}')
```

Imagen 14. Carga y limpieza de variables.

Se hizo el mismo procedimiento que en los Vinos Rojos. Se prepararon las variables para que los modelos de redes neuronales puedan entrenar.

a. Definición y Entrenamiento de las Redes Neuronales


```

# Definir los modelos
def create_perceptron():
    model = Sequential([
        Dense(units=1, activation='sigmoid', input_shape=(1,), use_bias=True)
    ])
    return model

def create_single_hidden():
    model = Sequential([
        Dense(units=1, activation='sigmoid', input_shape=(1,), use_bias=True),
        Dense(units=1, activation='sigmoid', use_bias=True)
    ])
    return model

def create_double_hidden():
    model = Sequential([
        Dense(units=2, activation='sigmoid', input_shape=(1,), use_bias=True),
        Dense(units=2, activation='sigmoid', use_bias=True),
        Dense(units=1, activation='sigmoid', use_bias=True)
    ])
    return model

# Lista de modelos
models = [
    ('Perceptrón', create_perceptron),
    ('Una capa oculta', create_single_hidden),
    ('Dos capas ocultas', create_double_hidden)
]

# Diccionario para almacenar resultados
results = {}

# Entrenar y evaluar cada modelo
for name, create_model in models:
    model = create_model()
    model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
    model.fit(X_train, y_train, epochs=10, batch_size=1, verbose=1)
    y_pred = model.predict(X_test)
    y_pred = (y_pred > 0.5).astype(int)

    # Calcular métricas
    acc = accuracy_score(y_test, y_pred)
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    conf_matrix = confusion_matrix(y_test, y_pred)

    # Guardar resultados
    results[name] = {
        'accuracy': acc,
        'precision': precision,
        'recall': recall,
        'f1_score': f1,
        'conf_matrix': conf_matrix
    }

```

Imagen 15. Definición y entrenamiento de los modelos.

Se implemento la misma estrategia que en lo vinos rojos para realizar una mejor comparación sobre como en cada tipo de vino, los modelos definen la calidad del vino.

6. ANALISIS DE RESULTADOS

a. VINOS ROJOS

i. ALCOHOL Y CALIDAD

Perceptrón: Accuracy: 0.5426 Precision: 0.5356 Recall: 0.9518 F1 Score: 0.6855 Confusion Matrix: [[14 137] [8 158]]	Análisis comparativo: Accuracy: Perceptrón: 0.5426 Una capa oculta: 0.5268 Dos capas ocultas: 0.5237
Una capa oculta: Accuracy: 0.5268 Precision: 0.5253 Recall: 1.0000 F1 Score: 0.6888 Confusion Matrix: [[1 150] [0 166]]	Precision: Perceptrón: 0.5356 Una capa oculta: 0.5253 Dos capas ocultas: 0.5237
Dos capas ocultas: Accuracy: 0.5237 Precision: 0.5237 Recall: 1.0000 F1 Score: 0.6874 Confusion Matrix: [[0 151] [0 166]]	Recall: Perceptrón: 0.9518 Una capa oculta: 1.0000 Dos capas ocultas: 1.0000
	F1_score: Perceptrón: 0.6855 Una capa oculta: 0.6888 Dos capas ocultas: 0.6874

Imagen 16. Resultados de las variables Alcohol y Calidad de los Vinos Rojos.

- **Accuracy:** Es la proporción de muestras correctamente clasificadas sobre el total de muestras. En este caso, el perceptrón tiene una precisión del 54.26%, lo que significa que aproximadamente el 54.26% de las muestras fueron clasificadas correctamente. Para el modelo con una capa oculta, la precisión es del 52.68%, y para el modelo con dos capas ocultas, es del 52.37%. Esto indica la habilidad general del modelo para hacer predicciones correctas.
- **Precision (Precisión):** Es la proporción de verdaderos positivos sobre todos los positivos predichos por el modelo. En otras palabras, mide la calidad de las predicciones positivas. Por ejemplo, el perceptrón tiene una precisión del 53.56%, lo que significa que cuando predice que un vino es de alta calidad, el 53.56% de las veces está en lo correcto. Para el modelo con una capa oculta y el modelo con dos capas ocultas, la precisión es similar.
- **Recall (Recuperación o Sensibilidad):** Es la proporción de verdaderos positivos sobre todos los positivos reales en los datos. Por ejemplo, el recall del perceptrón es del 95.18%, lo que indica que, de todos los vinos de alta calidad en el conjunto de datos de prueba, el 95.18% fueron correctamente identificados por el modelo. Para los modelos con una y dos capas ocultas, el recall es del 100%, lo que significa que identificaron todos los casos positivos en el conjunto de prueba.
- **F1 Score:** Es la media armónica de precisión y recall. Proporciona una medida única que combina ambas métricas. Un F1 Score más alto indica un mejor equilibrio entre precisión y recall. En este

caso, el perceptrón tiene un F1 Score del 68.55%, mientras que los modelos con una y dos capas ocultas tienen F1 Scores ligeramente superiores (68.88% y 68.74% respectivamente), lo que indica un rendimiento ligeramente mejor en la clasificación.

- Matriz de confusión:

- Perceptrón:

- **14:** Verdaderos negativos (TN) - El modelo predijo que 14 vinos eran de baja calidad y eran realmente de baja calidad.
 - **137:** Falsos positivos (FP) - El modelo predijo que 137 vinos eran de alta calidad, pero eran de baja calidad.
 - **8:** Falsos negativos (FN) - El modelo predijo que 8 vinos eran de baja calidad, pero eran de alta calidad.
 - **158:** Verdaderos positivos (TP) - El modelo predijo que 158 vinos eran de alta calidad y eran realmente de alta calidad.

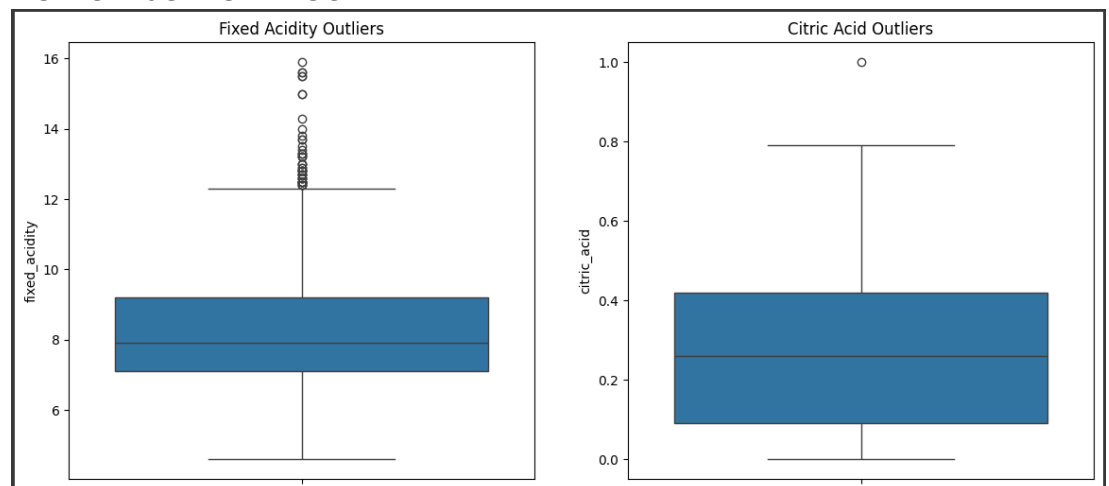
- Una capa oculta

- **1:** Verdaderos negativos (TN) - El modelo predijo que 1 vino era de baja calidad y era realmente de baja calidad.
 - **150:** Falsos positivos (FP) - El modelo predijo que 150 vinos eran de alta calidad, pero eran de baja calidad.
 - **0:** Falsos negativos (FN) - El modelo predijo que 0 vinos eran de baja calidad, pero eran de alta calidad.
 - **166:** Verdaderos positivos (TP) - El modelo predijo que 166 vinos eran de alta calidad y eran realmente de alta calidad.

- Dos capas ocultas

- **0:** Verdaderos negativos (TN) - El modelo predijo que 0 vinos eran de baja calidad y eran realmente de baja calidad.
 - **151:** Falsos positivos (FP) - El modelo predijo que 151 vinos eran de alta calidad, pero eran de baja calidad.
 - **0:** Falsos negativos (FN) - El modelo predijo que 0 vinos eran de baja calidad, pero eran de alta calidad.
 - **166:** Verdaderos positivos (TP) - El modelo predijo que 166 vinos eran de alta calidad y eran realmente de alta calidad.

ii. ACIDO FIJO Y CITRICO



Gráfica 15. Diagrama de Caja y Bigotes de las variables acidez fija y cítrica.

El gráfico tiene dos variables: 'acidez fija' y 'ácido cítrico'. Ambos se miden en el eje x y el eje y respectivamente. Los puntos representan diferentes vinos y su posición indica sus niveles de acidez.

Acidez fija: Esta es una medida del total de ácidos titulables contenidos en un vino, excluyendo los ácidos málico, tartárico, láctico y cítrico. Se expresa en gramos por litro (g/L). Los valores en el contexto oscilan entre 8 y 16, y un valor atípico tiene una acidez fija de 10. Ácido cítrico: este es un ácido orgánico débil que se encuentra en pequeñas cantidades en muchas frutas, incluidas las uvas. Es uno de los cinco ácidos tartáricos primarios del vino. Los valores en el contexto oscilan entre 0 y 0,4, con dos valores atípicos marcados como '08-' y '04' (que podrían significar 0,08 y 0,04 respectivamente, ya que el '-' podría usarse para indicar decimales).

En el gráfico se puede ver que a medida que aumenta la acidez fija, el contenido de ácido cítrico generalmente disminuye. Esto tiene sentido, ya que la acidez fija mide una combinación de ácidos, y un aumento en uno podría conducir a una disminución en la proporción de otros, como el ácido cítrico.

```
Perceptrón:
Accuracy: 0.51875
Precision: 0.551440329218107
Recall: 0.7486033519553073
F1 Score: 0.6350710900473934
Confusion Matrix:
[[ 32 109]
 [ 45 134]]

10/10 [=====] - 0s 2ms/step
Una capa oculta:
Accuracy: 0.559375
Precision: 0.559375
Recall: 1.0
F1 Score: 0.7174348697394789
Confusion Matrix:
[[ 0 141]
 [ 0 179]]

10/10 [=====] - 0s 3ms/step
Dos capas ocultas:
Accuracy: 0.525
Precision: 0.5470383275261324
Recall: 0.8770949720670391
F1 Score: 0.6738197424892703
Confusion Matrix:
[[ 11 130]
 [ 22 157]]
```

Imagen 17. Resultados de las variables acidez fija y cítrica.

- **Accuracy (Precisión):**
 - **Perceptrón:** 51.88%
 - **Una capa oculta:** 55.94%

- **Dos capas ocultas:** 52.50%

La precisión indica el porcentaje de predicciones correctas realizadas por el modelo. En este caso, todos los modelos tienen una precisión baja, lo que significa que poco más de la mitad de las predicciones son correctas.

- **Precision (Precisión):**

- **Perceptrón:** 55.14%
- **Una capa oculta:** 55.94%
- **Dos capas ocultas:** 54.70%

La precisión mide la proporción de vinos predichos como de alta calidad que realmente son de alta calidad. Una precisión más alta significa menos falsos positivos, es decir, menos vinos de baja calidad clasificados incorrectamente como de alta calidad. Los valores de precisión están alrededor del 55%, lo que indica que casi la mitad de los vinos predichos como de alta calidad son, de hecho, de baja calidad.

- **Recall (Sensibilidad):**

- **Perceptrón:** 74.86%
- **Una capa oculta:** 100%
- **Dos capas ocultas:** 87.71%

El recall mide la capacidad del modelo para identificar correctamente los vinos de alta calidad. Un recall del 100% en el modelo de una capa oculta significa que todos los vinos de alta calidad fueron correctamente identificados. El perceptrón y el modelo de dos capas ocultas también tienen un recall alto, lo que indica que la mayoría de los vinos de alta calidad se identifican correctamente.

- **F1 Score:**

- **Perceptrón:** 63.51%
- **Una capa oculta:** 71.74%
- **Dos capas ocultas:** 67.38%

El F1 Score es una media armónica de precisión y recall. Un F1 Score más alto indica un mejor equilibrio entre precisión y recall. El modelo de una capa oculta tiene el F1 Score más alto, lo que sugiere que es el mejor en balancear precisión y recall.

- **Matriz de Confusión:**

- **Perceptrón:**
 - **32:** Vinos de baja calidad correctamente identificados.
 - **109:** Vinos de baja calidad incorrectamente clasificados como de alta calidad.
 - **45:** Vinos de alta calidad incorrectamente clasificados como de baja calidad.
 - **134:** Vinos de alta calidad correctamente identificados.
- **Una capa oculta**
 - **0:** No hay vinos de baja calidad correctamente identificados.
 - **141:** Vinos de baja calidad incorrectamente clasificados como de alta calidad.
 - **0:** No hay vinos de alta calidad incorrectamente clasificados como de baja
 - **179:** Vinos de alta calidad correctamente identificados como alta calidad (TP).
- **Dos capas ocultas**
 - **11:** Vinos de baja calidad correctamente identificados como baja calidad (TN).
 - **130:** Vinos de baja calidad incorrectamente clasificados como de alta calidad (FP).
 - **22:** Vinos de alta calidad incorrectamente clasificados como de baja calidad (FN).
 - **157:** Vinos de alta calidad correctamente identificados como alta calidad (TP).

b. **VINOS BLANCOS**

i. **ALCOHOL Y CALIDAD**

```

Perceptrón:
Accuracy: 0.6772867420349434
Precision: 0.6772867420349434
Recall: 1.0
F1 Score: 0.8075980392156863
Confusion Matrix:
[[ 0 314]
 [ 0 659]]

31/31 [=====] - 0s 1ms/step
Una capa oculta:
Accuracy: 0.6772867420349434
Precision: 0.6772867420349434
Recall: 1.0
F1 Score: 0.8075980392156863
Confusion Matrix:
[[ 0 314]
 [ 0 659]]

31/31 [=====] - 0s 1ms/step
Dos capas ocultas:
Accuracy: 0.6772867420349434
Precision: 0.6772867420349434
Recall: 1.0
F1 Score: 0.8075980392156863
Confusion Matrix:
[[ 0 314]
 [ 0 659]]

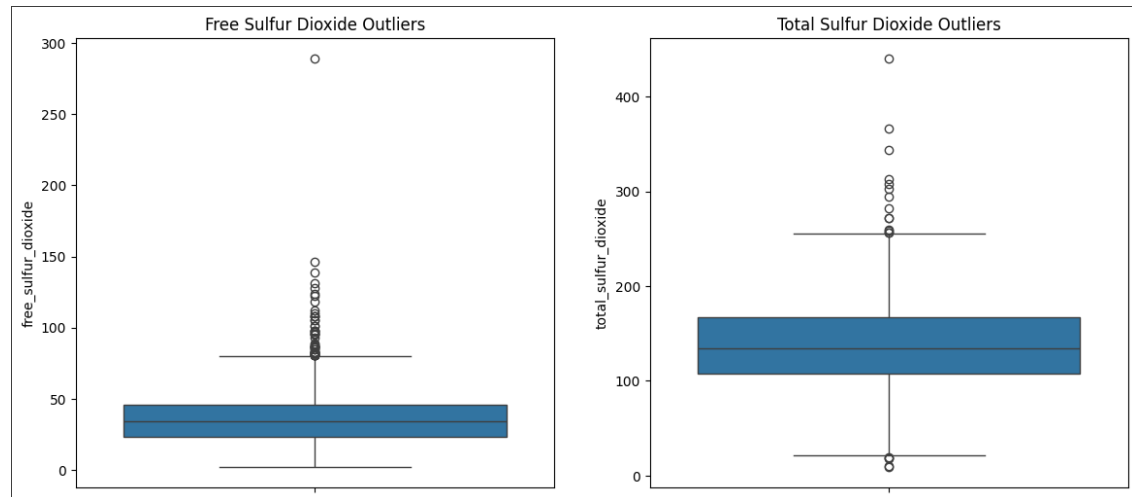
Análisis comparativo:
Accuracy:
Perceptrón: 0.6772867420349434
Una capa oculta: 0.6772867420349434
Dos capas ocultas: 0.6772867420349434
Precision:
Perceptrón: 0.6772867420349434
Una capa oculta: 0.6772867420349434
Dos capas ocultas: 0.6772867420349434
Recall:
Perceptrón: 1.0
Una capa oculta: 1.0
Dos capas ocultas: 1.0
F1 Score:
Perceptrón: 0.8075980392156863
Una capa oculta: 0.8075980392156863
Dos capas ocultas: 0.8075980392156863

```

Imagen 18. Resultados de las variables alcohol y calidad.

- **Accuracy:**
0.6773: Esto significa que aproximadamente el 67.73% de las predicciones realizadas por cada modelo son correctas. Dado que todas las métricas son iguales, esto sugiere que todos los modelos tienen un comportamiento idéntico en términos de clasificaciones correctas.
- **Precision (Precisión):**
0.6773: Esto indica que el 67.73% de las predicciones positivas son correctas. En otras palabras, cuando el modelo predice que un vino es de alta calidad, tiene razón el 67.73% de las veces.
- **Recall (Sensibilidad):**
1.0000: Un recall de 1.0 indica que el modelo está identificando todos los vinos de alta calidad correctamente, sin perder ninguno (no hay falsos negativos).
- **F1 Score:**
0.8076: El F1 Score es la media armónica de la precisión y el recall, y un valor de 0.8076 indica un buen equilibrio entre ambas métricas. Dado que el recall es perfecto, el F1 Score es relativamente alto.
- **Matriz de confusión**
Para todos los modelos:
 - **Verdaderos negativos (TN): 0** - No hay predicciones correctas de vinos de baja calidad.
 - **Falsos positivos (FP): 314** - El modelo clasificó incorrectamente 314 vinos de baja calidad como de alta calidad.
 - **Falsos negativos (FN): 0** - No hay predicciones incorrectas de vinos de alta calidad.
 - **Verdaderos positivos (TP): 659** - El modelo clasificó correctamente 659 vinos de alta calidad.

ii. DIOXIDO DE AZUFRE TOTAL Y LIBRE



Gráfica 16. Diagrama de Caja y Bigotes de las variables Dióxido de azufre total y libre.

El dióxido de azufre libre (SO₂) es un compuesto químico que se utiliza en la elaboración de vinos blancos para protegerlos de la oxidación y mantener su frescura. El nivel de SO₂ libre en un vino blanco puede variar, pero se considera que los niveles óptimos están entre 300 y 50 mg/L. En el contexto proporcionado, se mencionan los niveles de SO₂ libre en un vino blanco, que van desde 300 mg/L hasta 50 mg/L. También se mencionan los niveles de SO₂ total, que incluyen tanto el SO₂ libre como el SO₂ combinado. Los niveles de SO₂ total van desde 400 mg/L hasta 100 mg/L. En cuanto a los outliers (valores atípicos) mencionados en el contexto, se refieren a valores que están fuera del rango normal de SO₂ libre y total. Estos valores pueden ser causados por errores en la medición o por problemas en el proceso de elaboración del vino.

```

Perceptrón:
Accuracy: 0.5969387755102041
Precision: 0.7490566037735849
Recall: 0.6024279210925645
F1 Score: 0.6677880571909167
Confusion Matrix:
[[188 133]
 [262 397]]

31/31 [=====] - 0s 1ms/step
Una capa oculta:
Accuracy: 0.6724489795918367
Precision: 0.6724489795918367
Recall: 1.0
F1 Score: 0.8041488712629652
Confusion Matrix:
[[ 0 321]
 [ 0 659]]

31/31 [=====] - 0s 1ms/step
Dos capas ocultas:
Accuracy: 0.6724489795918367
Precision: 0.6724489795918367
Recall: 1.0
F1 Score: 0.8041488712629652
Confusion Matrix:
[[ 0 321]
 [ 0 659]]

```

Imagen 19. Resultado de las variables dióxido de azufre libre y total.

- **Accuracy:**

- **Perceptrón:** 59.69%
- **Una capa oculta:** 67.24%
- **Dos capas ocultas:** 67.24%

La precisión indica el porcentaje de predicciones correctas. En este caso, los modelos con una y dos capas ocultas tienen una precisión similar y superior a la del perceptrón, lo que indica que alrededor de dos tercios de las predicciones son correctas.

- **Precision (Precisión):**

- **Perceptrón:** 74.91%
- **Una capa oculta:** 67.24%
- **Dos capas ocultas:** 67.24%

La precisión mide la proporción de vinos predichos como de alta calidad que realmente son de alta calidad. El perceptrón tiene una precisión mayor (74.91%) en comparación con los otros dos modelos (67.24%), lo que significa que cuando el perceptrón predice un vino como de alta calidad, es más probable que realmente lo sea.

- **Recall (Sensibilidad):**

- **Perceptrón:** 60.24%
- **Una capa oculta:** 100%
- **Dos capas ocultas:** 100%

El recall mide la capacidad del modelo para identificar correctamente los vinos de alta calidad. Los modelos con una y dos capas ocultas identifican correctamente todos los vinos de alta calidad (recall del 100%), mientras que el perceptrón identifica alrededor del 60% de los vinos de alta calidad.

- **F1 Score:**

- **Perceptrón:** 66.78%
- **Una capa oculta:** 80.41%
- **Dos capas ocultas:** 80.41%

El F1 Score es una media armónica de precisión y recall. Un F1 Score más alto indica un mejor equilibrio entre precisión y recall. Los modelos con una y dos capas ocultas tienen un F1 Score más alto (80.41%), lo que sugiere que son mejores en balancear precisión y recall en comparación con el perceptrón.

- **Matriz de Confusión:**

- **Perceptrón:**
 - **188:** Vinos de baja calidad correctamente identificados.
 - **133:** Vinos de baja calidad incorrectamente clasificados como de alta calidad.
 - **262:** Vinos de alta calidad incorrectamente clasificados como de baja calidad.
 - **397:** Vinos de alta calidad correctamente identificados.
- **Una Capa Oculta**
 - **0:** No hay vinos de baja calidad correctamente identificados.
 - **321:** Vinos de baja calidad incorrectamente clasificados como de alta calidad.
 - **0:** No hay vinos de alta calidad incorrectamente clasificados como de baja calidad.
 - **659:** Vinos de alta calidad correctamente identificados.
- **Dos Capas Ocultas**
 - **0:** No hay vinos de baja calidad correctamente identificados.
 - **321:** Vinos de baja calidad incorrectamente clasificados como de alta calidad.
 - **0:** No hay vinos de alta calidad incorrectamente clasificados como de baja calidad.
 - **659:** Vinos de alta calidad correctamente identificados.

7. CONCLUSIONES

Estas son las conclusiones frente a lo que significaron los resultados de los modelos frente a las diferentes variables:

- Alcohol y Calidad en Vinos Rojos y Vinos Blancos:
 - Los resultados indican que, para el vino rojo, el modelo con perceptrón simple, una capa oculta y dos capas ocultas tienen un rendimiento similar, con precisiones alrededor del 52-54% y F1 Scores en el rango de 68-69%, aunque todos muestran una alta tasa de falsos positivos. En cambio, para el vino blanco, todos los modelos presentan una mejor predicción con una precisión del 67.73%, un recall perfecto del 100% y un F1 Score de 0.8076, aunque también con muchos falsos positivos. La predicción del vino blanco es claramente mejor, dado su mayor equilibrio entre precisión y recall y su mayor exactitud comparada con los modelos de vino rojo.
- Ácidos Fijos y Cítricos
 - Los resultados proporcionan una visión sobre cómo los modelos de clasificación de vinos, basados en los niveles de ácido fijo y ácido cítrico, evalúan la calidad del vino. La precisión y precisión muestran que alrededor del 55% de los vinos clasificados como de alta calidad por los modelos pueden ser incorrectos, lo que sugiere una tendencia a sobreestimar la calidad. Sin embargo, el alto recall, especialmente en el modelo con una capa oculta, indica una capacidad para identificar la mayoría de los vinos de alta calidad. El F1 Score más alto en este modelo resalta su capacidad para lograr un equilibrio entre precisión y recall
- Dióxido de Azufre Libre y Total:
 - Los resultados proporcionan una visión de cómo los modelos de clasificación de vinos, utilizando los niveles de dióxido de azufre libre y total, evalúan la calidad del vino. La precisión indica que los modelos con una y dos capas ocultas tienen una precisión similar y superior al perceptrón, alrededor del 67%, lo que sugiere que aproximadamente dos tercios de las predicciones son correctas. Sin embargo, la precisión del perceptrón es ligeramente mayor (74.91%), lo que implica que es más preciso en la identificación de vinos de alta calidad en comparación con los otros modelos. En términos de recall, los modelos con una y dos capas ocultas muestran un rendimiento perfecto del 100%, identificando correctamente todos los vinos de alta calidad, mientras que el perceptrón tiene un recall del 60.24%. En cuanto al F1 Score, los modelos con una y dos capas ocultas muestran un valor más alto (80.41%), indicando un mejor equilibrio entre precisión y recall en comparación con el perceptrón.