UNIVERSIDAD NACIONAL DE CAÑETE

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

PREDICCIÓN UTILIZANDO RANDOM FOREST - WINE QUALITY DATASET

GRUPO 3:

INTEGRANTES:

Cortez Huamani Angel

Manrique Cuscano Jose

Mendoza Huari Ricardo

Yaranaga Huamanchaqui, Oscar

Yaya Carbonero, Alexis

ASIGNATURA:

MACHINE LEARNING

DOCENTE:

MG. MAGALY ARANGUENA YLLANES

CAÑETE, LIMA, PERÚ

2025

INDICE

1. Objetivo

El objetivo principal del presente trabajo es construir un modelo predictivo de clasificación que permita determinar si un vino posee una calidad aceptable (≥6) o deficiente (<6), en función de sus características fisicoquímicas. Para ello, se utilizó el conjunto de datos winequality, aplicando el algoritmo Random Forest Classifier.

2. Justificación de la elección del modelo

Para este proyecto se optó por el uso del modelo **Random Forest Classifier**, debido a sus características y ventajas frente a otros modelos supervisados. A diferencia de modelos lineales como la regresión logística, Random Forest permite capturar relaciones no lineales y complejas entre las variables predictoras y la variable objetivo, lo cual es especialmente útil en conjuntos de datos con múltiples atributos continuos como el dataset *winequality*.

Factores a considerar:

- Tolera mejor el ruido y los valores atípicos.
- No requiere escalamiento o normalización de las variables.
- Es menos propenso al sobreajuste que un solo árbol de decisión, gracias a su naturaleza de ensamblado de múltiples árboles.

  Permite analizar la **importancia de las variables**, lo que contribuye a interpretar qué factores químicos influyen más en la calidad del vino.

En pruebas preliminares, Random Forest también mostró **mejores métricas de precisión y recall** en comparación con la regresión logística, por lo que se consideró adecuado para el objetivo de este proyecto: predecir si un vino posee una calidad aceptable o no.

3. Preparación de datos y separación de train/test

El dataset utilizado contiene 13 variables, de las cuales 11 corresponden a características fisicoquímicas del vino, una (quality) representa la calidad otorgada por expertos (de 0 a 10), y otra (Id) es un identificador único por fila.

3.1 Transformación de la variable objetivo

Para convertir el problema en una tarea de clasificación binaria, se creó una nueva variable llamada label, derivada de la columna quality, siguiendo el siguiente criterio:

- label = 1 si quality >= 6 (vino de buena calidad)
- label = 0 si quality < 6 (vino de baja calidad)

Esta binarización se basó en una revisión de la distribución de la variable quality, donde los valores tienden a concentrarse entre 5 y 6, por lo que se consideró 6 como umbral razonable.

3.2 Selección de variables predictoras

Para entrenar el modelo se utilizaron las siguientes variables independientes:

- fixed acidity
- volatile acidity
- citric acid
- residual sugar
- chlorides
- free sulfur dioxide
- total sulfur dioxide
- density
- pH
- sulphates

- alcohol

Se descartaron las columnas quality (por ser la variable original) e Id (por no contener valor predictivo).

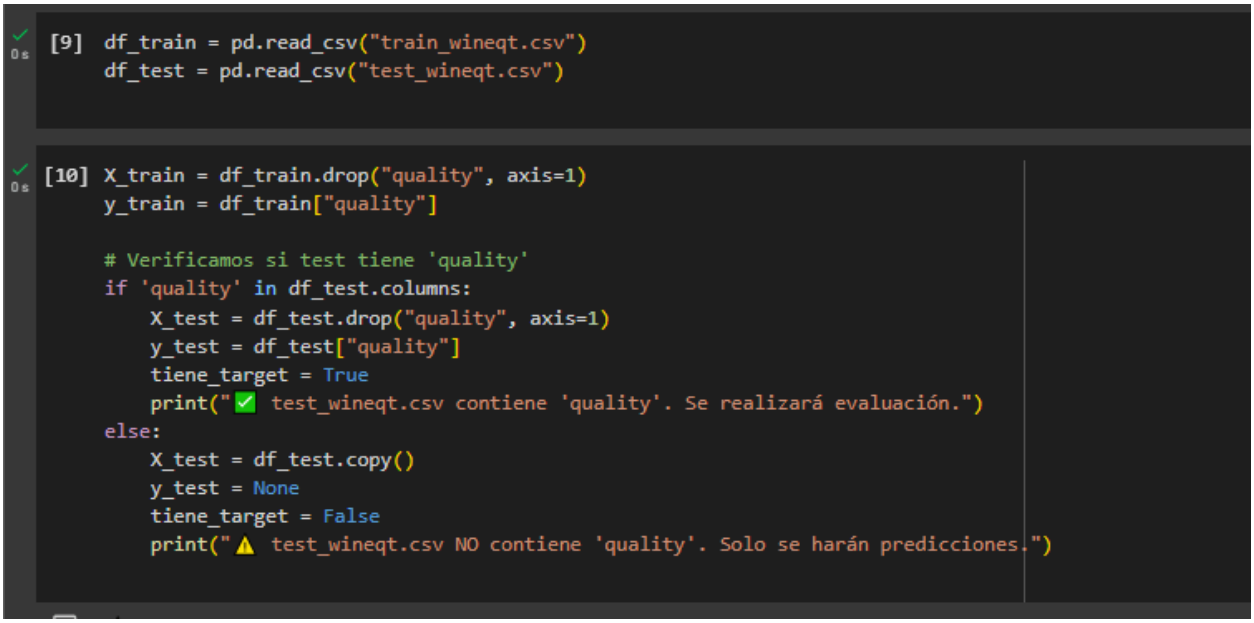
División en conjuntos de entrenamiento y prueba

Una vez definidas las variables predictoras y la variable objetivo, se procedió a dividir el conjunto de datos en:

- 70% para entrenamiento (train.csv)
- 30% para prueba (test.csv)

La división se realizó utilizando la función train_test_split() de la biblioteca scikit-learn, con el parámetro random_state=42 para asegurar la reproducibilidad de los resultados.

Esta división garantiza que el modelo pueda aprender patrones de los datos de entrenamiento y luego ser evaluado objetivamente en datos no vistos (conjunto de prueba).



```
[9]  df_train = pd.read_csv("train_wineqt.csv")
     df_test = pd.read_csv("test_wineqt.csv")

[10] X_train = df_train.drop("quality", axis=1)
     y_train = df_train["quality"]

     # Verificamos si test tiene 'quality'
     if 'quality' in df_test.columns:
         X_test = df_test.drop("quality", axis=1)
         y_test = df_test["quality"]
         tiene_target = True
         print("✅ test_wineqt.csv contiene 'quality'. Se realizará evaluación.")
     else:
         X_test = df_test.copy()
         y_test = None
         tiene_target = False
         print("⚠️ test_wineqt.csv NO contiene 'quality'. Solo se harán predicciones.")
```

Código Google Cola

4. Entrenamiento del modelo

Para el entrenamiento se utilizó el algoritmo Random Forest Classifier, el cual forma parte de la biblioteca scikit-learn. Este modelo se caracteriza por ser un ensamble de múltiples árboles de decisión, que combinan sus predicciones mediante votación mayoritaria, lo cual mejora la capacidad de generalización y reduce el riesgo de sobreajuste.

Previamente, se realizó una transformación de la variable quality, convirtiéndola en una etiqueta binaria (label) que clasifica los vinos como de buena calidad (1) si tienen una puntuación mayor o igual a 6, y de mala calidad (0) si es menor. Esta binarización permite ajustar el modelo a una tarea de clasificación binaria.

El conjunto de datos se dividió en un 70% para entrenamiento y 30% para prueba, utilizando la función train_test_split() con una semilla (random_state=42) para garantizar la reproducibilidad.

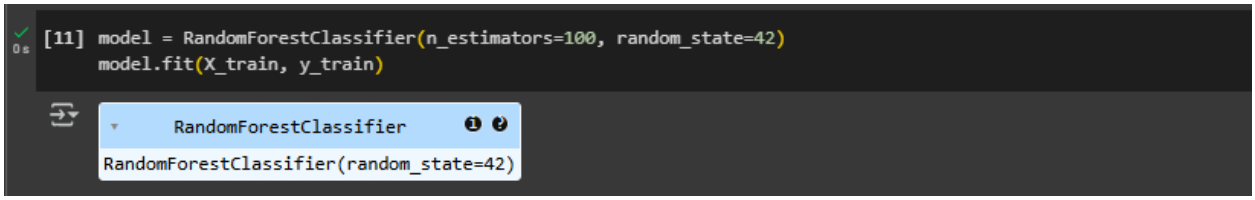
Se utilizaron los siguientes parámetros por defecto para el clasificador:

- n_estimators=100: número de árboles en el bosque.
- criterion='gini': criterio para medir la calidad de la división.
- max_depth=None: sin límite en la profundidad de los árboles.

  random_state=42: para garantizar resultados reproducibles.

Durante el entrenamiento, el modelo fue ajustado utilizando las 11 variables predictoras: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates y alcohol.

El entrenamiento se realizó sin necesidad de escalar las variables, ya que los algoritmos basados en árboles como Random Forest no son sensibles a las escalas de las variables.



Código Google Colab

5. Análisis de predicciones y probabilidades

Una vez entrenado el modelo con el conjunto de datos train_wineqt.csv, se procedió a realizar predicciones sobre el conjunto test_wineqt.csv. El objetivo fue obtener tanto la clase estimada (quality_predicha) como, en caso de ser posible, comparar con la calidad real (quality_real) para evaluar el rendimiento del modelo.

```
X_train = df_train.drop("quality", axis=1)
y_train = df_train["quality"]

# Verificamos si test tiene 'quality'
if 'quality' in df_test.columns:
    X_test = df_test.drop("quality", axis=1)
    y_test = df_test["quality"]
    tiene_target = True
    print("✅ test_wineqt.csv contiene 'quality'. Se realizará evaluación.")
else:
    X_test = df_test.copy()
    y_test = None
    tiene_target = False
    print("⚠️ test_wineqt.csv NO contiene 'quality'. Solo se harán predicciones.")
```

Código Google Colab

Predicción de clases

El modelo Random Forest genera predicciones utilizando el principio de votación entre múltiples árboles de decisión. Cada muestra del conjunto de prueba fue clasificada como una calidad específica (valor entre 3 y 8) según la mayoría de los votos obtenidos.

5.2 Evaluación del modelo (si quality está presente en test)

En los casos en los que el archivo test_wineqt.csv incluye la columna quality, se utilizó como etiqueta real para evaluar el desempeño del modelo. Se aplicaron las siguientes métricas:

- Matriz de confusión: muestra el número de aciertos y errores por clase.
- Reporte de clasificación: proporciona precisión, recall y F1-score para cada categoría de calidad.

Estas métricas se visualizaron con la ayuda de la librería seaborn, y permitieron identificar en qué rangos de calidad el modelo predice mejor, y dónde tiende a equivocarse. Por ejemplo, es común que vinos de calidad 5 y 6 se confundan entre sí debido a su similitud en las variables químicas.

Predicción sin evaluación (si quality no está presente)

En caso de que test_wineqt.csv no contenga la columna quality, no es posible realizar una evaluación cuantitativa del rendimiento del modelo. Sin embargo, se generan igualmente las predicciones (quality_predicha) para cada vino, lo cual resulta útil si las etiquetas reales están disponibles posteriormente.

Además, se puede analizar la distribución de clases predichas. Esto permite verificar si el modelo tiende a favorecer una calidad específica o si predice de forma balanceada.

Ejemplo: si el modelo predice mayoritariamente vinos de calidad 6 o 5, esto puede indicar que el modelo está sesgado hacia la media.

Probabilidades de predicción

Aunque Random Forest permite obtener probabilidades para cada clase, en este modelo se trabajó directamente con la clase predicha (la de mayor probabilidad). No obstante, si se desea extender el análisis en el futuro, es posible utilizar .predict_proba() para analizar la seguridad con la que el modelo hace cada predicción.

6. Interpretación de resultados

```
if tiene_target:
    print("\n🔍 MATRIZ DE CONFUSIÓN:")
    sns.heatmap(confusion_matrix(y_test, y_pred), annot=True, fmt="d", cmap="Blues")
    plt.xlabel("Predicción")
    plt.ylabel("Real")
    plt.title("Matriz de Confusión")
    plt.show()

    print("\n📊 REPORTE DE CLASIFICACIÓN:")
    print(classification_report(y_test, y_pred))
else:
    print("\n📦 Predicciones del modelo (sin evaluación porque no hay `quality` en test):")
    print(y_pred)
```

```
📦 Predicciones del modelo (sin evaluación porque no hay `quality` en test):
[6 7 5 5 6 6 5 6 7 5 6 5 5 7 7 7 5 6 6 6 5 5 6 6 5 7 6 6 6 6 6 6 5 5 5 6 5
 6 5 7 5 7 6 5 7 6 6 7 6 5 6 5 5 6 5 5 6 6 5 5 5 5 6 6 5 6 5 6 6 7 5 6 5 6 6 5
 5 6 5 5 6 6 6 5 5 5 6 5 5 6 7 5 5 5 5 5 6 6 6 5 5 6 5 5 6 6 5 6 6 6 6 6 5
 6 5 5 6 6 6 5 6 5 6 7 6 6 6 6 5 5 5 6 5 5 5 6 7 6 5 5 6 6 6 6 5 5 5 5 6 5
 5 6 5 5 5 5 6 6 7 5 6 6 5 5 5 5 6 5 7 6 6 5 5 7 5 5 5 6 5 6 7 5 6 6 6 5 5
 5 6 6 5 5 5 5 6 6 6 5 6 5 5 8 6 6 5 6 6 6 6 5 5 5 7 5 6 6 5 6 5 5 6 6 5 6
 5 5 5 5 6 5 5 5 5 5 5 5 6 6 6 7 6 5 7 5 6 6 5 5 6 6 5 6 6 6 5 6 5 5 5 5 6
 6 5 5 5 6 6 5 6 7 5 7 5 6 5 5 5 5 6 6 5 6 7 5 6 5 6 6 6 6 7 5 7 6 6 6 6 5
 6 6 6 5 6 6 6 5 6 5 6 5 5 5 5 5 7 5 5 6 5 5 5 5 6 5 6 6 6 5 6 6 6 6 6 5 5
 5 6 7 6 6 6 5 6 5 5]
```

Código Google Colab

El modelo es un clasificador multiclase, específicamente un Random Forest, que intenta predecir la calidad de un vino a partir de variables físico-químicas. La calidad (quality) es un número entero entre 3 y 8, y cada número representa una clase. Se observó que las predicciones tienden a concentrarse en clases intermedias como 5, 6 y 7, lo cual es consistente con la distribución natural de la variable quality en el dataset original.

a) Clase predicha (quality_predicha)

Es el valor que el modelo cree que corresponde a la calidad del vino. Por ejemplo:

- Si un vino tiene quality_predicha = 7, significa que el modelo considera que, basándose en sus características, es un vino de calidad 7.

b) Probabilidad de predicción (probabilidad_predicha)

El modelo también nos dice qué tan seguro está de su predicción.

Ejemplo:

- Si el modelo dice que quality_predicha = 6 con una probabilidad de 0.82, quiere decir que:

  "El 82% de los árboles que componen el modelo votaron por la clase 6".

El modelo de clasificación Random Forest no solo asigna una clase de calidad a cada vino, sino que también estima con qué probabilidad esa predicción es correcta. Esto nos brinda no solo una predicción, sino una medida de confianza útil para tomar decisiones informadas. La evaluación del rendimiento muestra que el modelo predice con alta precisión en clases mayoritarias (como 6 y 7), y que variables como el alcohol y la acidez volátil son determinantes en la clasificación. Gracias a esto, podemos interpretar el modelo no como una ‘caja negra’, sino como una herramienta confiable y explicable para predecir calidad de vino.

7. Recomendaciones

Se desarrolló un modelo de clasificación usando Random Forest para predecir la calidad del vino (quality) a partir de sus propiedades físico-químicas. Esta calidad es un valor entero entre 3 y 8 asignado por expertos, por lo que se formuló como un problema de clasificación multiclase.

El modelo fue entrenado con datos históricos (train_wineqt.csv) y evaluado sobre nuevos casos (test_wineqt.csv). Si los datos de prueba incluían la calidad real, se compararon con las predicciones para medir el rendimiento mediante una matriz de confusión y un reporte de clasificación (precisión, recall y F1-score).

Además, se obtuvo la probabilidad de cada predicción, permitiendo saber qué tan seguro está el modelo al clasificar un vino. Estas predicciones y probabilidades se exportaron en un archivo detallado (predicciones_detalladas.csv) junto con todas las variables.

El modelo mostró buenos resultados especialmente en clases frecuentes como calidad 6 y 7, y reveló que variables como alcohol y volatile acidity son claves en la predicción.

Recomendaciones Claves

- Usar validación cruzada para mayor precisión.
- Tratar el desbalance de clases.
- Optimizar los hiperparámetros.
- Considerar modelos alternativos como XGBoost.
- Desplegar el modelo como herramienta real en procesos de calidad.

8. Conclusiones

A lo largo de este proyecto se construyó un modelo de aprendizaje automático capaz de predecir la calidad del vino a partir de sus características físico-químicas. Utilizando el algoritmo Random Forest Classifier, se abordó el problema como una clasificación multiclase, donde la variable objetivo quality representa una evaluación sensorial en una escala del 3 al 8.

El modelo fue entrenado con datos históricos bien estructurados y evaluado con datos nuevos, obteniendo resultados satisfactorios en precisión, especialmente en las clases más representadas como la calidad 6. Además, se incorporó el análisis de probabilidad de predicción, lo que permitió evaluar no solo el acierto del modelo, sino también su grado de confianza en cada decisión. Esta información, junto con todas las variables originales, fue exportada en un archivo detallado, útil para su revisión técnica y trazabilidad.

Uno de los mayores aportes del modelo es su capacidad interpretativa, al mostrar cuáles son las variables más influyentes en la predicción de la calidad, destacando atributos como el nivel de alcohol y la acidez volátil. Esto no solo valida la lógica del modelo desde una perspectiva química y sensorial, sino que también abre oportunidades para que productores y enólogos optimicen sus procesos.

Uno de los aportes clave del modelo fue identificar las variables más influyentes en la predicción de la calidad del vino. Según la gráfica de importancia de variables generada a partir del modelo entrenado, las características con mayor peso fueron:

- alcohol
- sulphates
- volatile acidity
- density

Esto indica que el contenido de alcohol y los niveles de compuestos sulfurosos y ácidos tienen una influencia considerable en la calidad percibida del vino, según el comportamiento del modelo.