

Asignación de Clasificación

Ryoji Takahashi

December 1, 2024

1 Descripción de las Asignaciones y Objetivos

Las asignaciones de clasificación son:

1. Elegir un conjunto de datos de clasificación abierto distinto al Iris (fuente de referencia).
2. En un notebook de Jupyter, realizar un análisis exploratorio de datos (EDA), incluyendo limpieza de datos, transformaciones, agregaciones y visualizaciones según corresponda.
3. Seleccionar, entrenar y probar el(los) modelo(s) considerado(s) apropiado(s).
4. Justificar el modelo elegido basado en métricas de rendimiento.
5. Sacar conclusiones del ejercicio realizado.
6. Preparar un entregable con todos los archivos necesarios para reproducir el análisis y poner el modelo entrenado en producción (integración DevOps).

1.1 Conjunto de Datos

1. El conjunto de datos de **calidad del vino** fue seleccionado para este estudio. Las calidades del vino fueron clasificadas de **3** a **8** (**8** como una calidad buena). Podría clasificarse como una clasificación multi-clase, sin embargo, realicé una clasificación binaria categorizando como por encima de 7 es un buen vino (etiquetado como 1), y menos de 7 no es un buen vino (etiquetado como 0). Como resultado, el número de vinos no buenos es **1382** y de buenos vinos es **217**. Este es un conjunto de datos desbalanceado.

1.2 Análisis Exploratorio de Datos (EDA)

Antes de discutir los métodos de clasificación, quisiera abordar el análisis exploratorio de datos (EDA). Los detalles del EDA se muestran en el notebook. No se presentaron valores faltantes (NAs) y los tipos de datos eran apropiados. Sin embargo, si se presentaran valores faltantes, deberían manejarse cuidadosamente ya sea eliminándolos o imputándolos.

Las correlaciones de las características se muestran en la Figura 1 al final de la sección del EDA. La "densidad" tiene una fuerte correlación positiva con

el "azúcar residual", mientras que tiene una fuerte correlación negativa con el "alcohol". "Alcohol" tiene una correlación positiva con la "calidad", mientras que el "dióxido de azufre libre" y el "ácido cítrico" tienen casi ninguna correlación con la "calidad". Estos hallazgos son muy importantes para la ingeniería de características posterior.

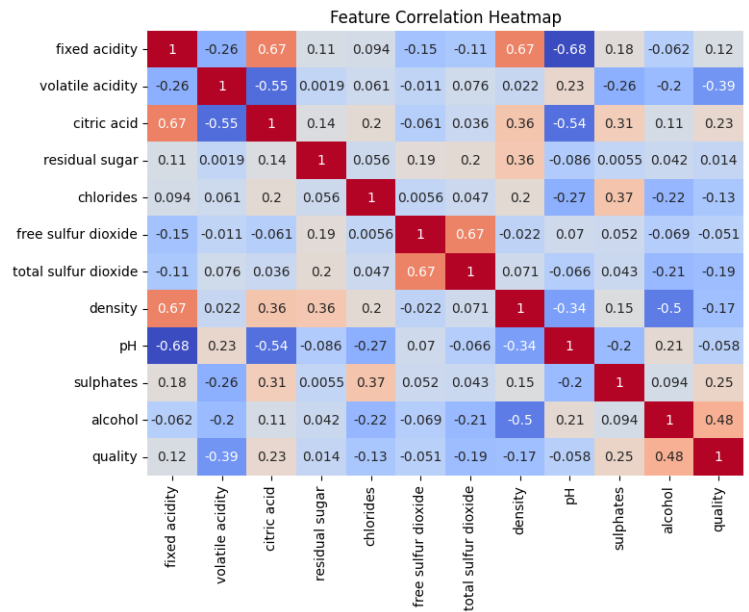


Figure 1: Correlaciones

1.3 Métodos de Clasificación

3. Construcción de modelos de aprendizaje automático. Como mencioné, este es un conjunto de datos desbalanceado. Para manejar tales conjuntos de datos, hay varias opciones, como la Técnica de Sobremuestreo de Minorías Sintéticas (SMOTE), el ajuste de umbral y los métodos de ensamble. Como uno de los métodos de ensamble, el *Random Forest (RF)*, bagging con árboles de decisión independientes. Adicionalmente, *XGBoost*, bagging con $L1$ y $L2$ regularización es un método popular. Por lo tanto, implementé estos dos métodos para la clasificación binaria, después de escalar los datos utilizando *StandardScaler* (transformando a $\mu = 0$, $\sigma = 1$). Como es estándar, el conjunto de datos fue dividido en 80 % de entrenamiento y 20 % de prueba. El *Random Forest (RF)* tuvo una precisión y un recall de **0.67** y **0.60**, respectivamente. Para mejorar esta

precisión, realicé ajuste de hiperparámetros utilizando una búsqueda en cuadrícula. Según mi experiencia, logré mejorar los resultados, sin embargo, no fue posible mejorarlos significativamente. En el código, dejé las líneas de la búsqueda en cuadrícula, y sigo investigando. Luego, apliqué *XGBoost* con ajuste de hiperparámetros usando *optuna*. Los resultados mejoraron en comparación con *RF*, con puntajes de precisión y recall de 0.67 y 0.73, respectivamente. Por supuesto, para desplegar en producción, también sería importante repetir el entrenamiento y (validación) prueba con ajustes adicionales de hiperparámetros.

La Figura 2 muestra la matriz de confusión (*confusion matrix*, CM) de los resultados de *XGBoost*. La matriz de confusión es una de las varias métricas de evaluación utilizadas para medir el rendimiento de un modelo de clasificación. Las métricas de rendimiento del modelo, como precisión y recall, se calcularon a partir de la matriz.

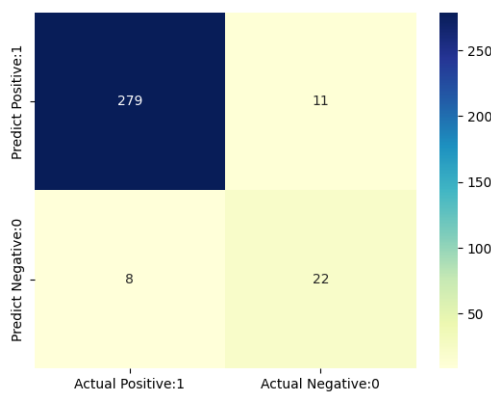


Figure 2: Matriz de Confusión (*Confusion Matrix*)

También es común mostrar la curva Característica Operativa del Receptor (*Receiver Operating Characteristic*, ROC). En la Figura 3. Es una representación gráfica del rendimiento de un clasificador binario en diferentes umbrales de clasificación. No utilicé técnicas de ajuste de umbral, que grafican la Tasa de Verdaderos Positivos (*True Positive Rate*, TPR) frente a la Tasa de Falsos Positivos (*False Positive Rate*, FPR).

La Figura 4 muestra las importancias de las características. Como en el código, tanto *RF* como *XGBoost* coincidieron en las importancias de las características. Este gráfico es importante para la ingeniería de características que se discutirá en la última sección.

Finalmente, para la entrega y reproducción, guardé los modelos entrenados en archivos *pkl*, como se indica en el código, para que se puedan cargar los modelos entrenados y realizar pruebas.

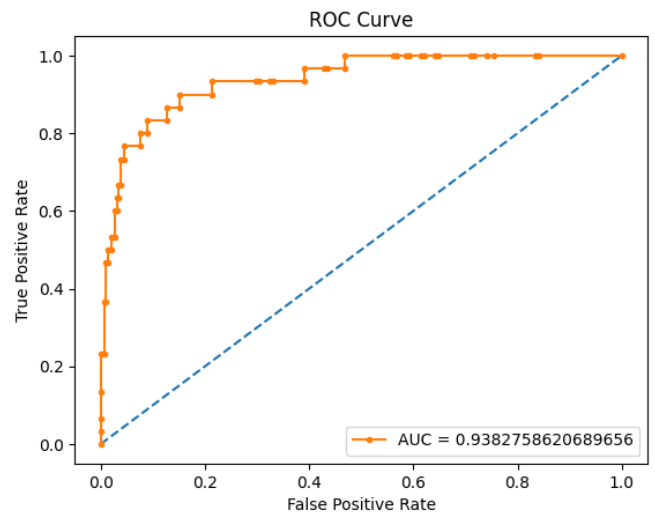


Figure 3: Curva ROC (*ROC Curve*)

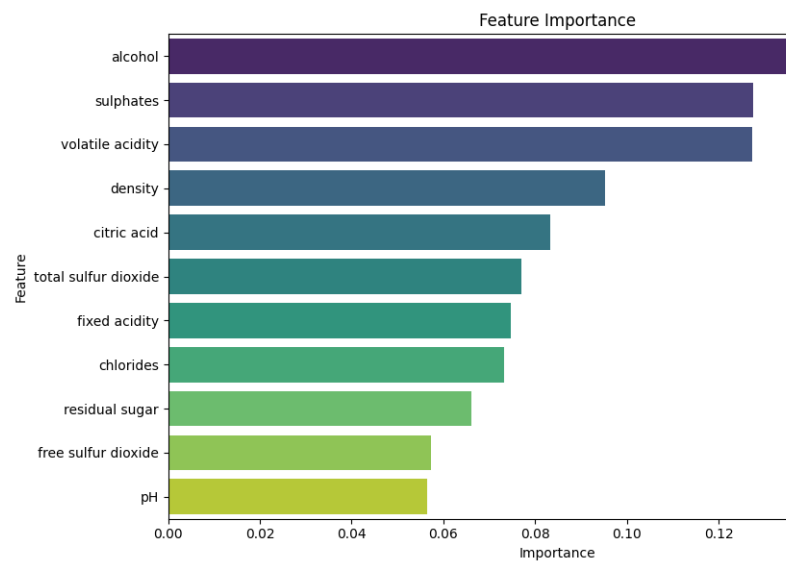


Figure 4: Importancia de las Características (*Feature Importance*)

2 Conclusión y Discusión Adicional

En los conjuntos de datos del mundo real (RWD), a menudo me he encontrado con conjuntos de datos desbalanceados. He aplicado SMOTE en algunos estudios. Sin embargo, he encontrado que es menos efectivo para mejorar la precisión en conjuntos de datos desbalanceados "generales". Usualmente, como en el ejemplo anterior, *RF* y *XGBoost* con *StandardScaler* ofrecen mejor desempeño. También me gustaría señalar que el "compromiso sesgo-varianza" es un tema importante. Ya que una mejor precisión será un compromiso entre el sesgo y la varianza (complejidad). La ingeniería de características, como seleccionar características importantes y eliminar las no importantes, también podría mejorar la modelización de estos conjuntos de datos desbalanceados.