PREDECIR LA CALIDAD DE UN VINO

Presentación Técnica

AGENDA

01. Origen

02. Datos, Limpieza y Feature Engineering

03. Modelos

04. Modelo Seleccionado

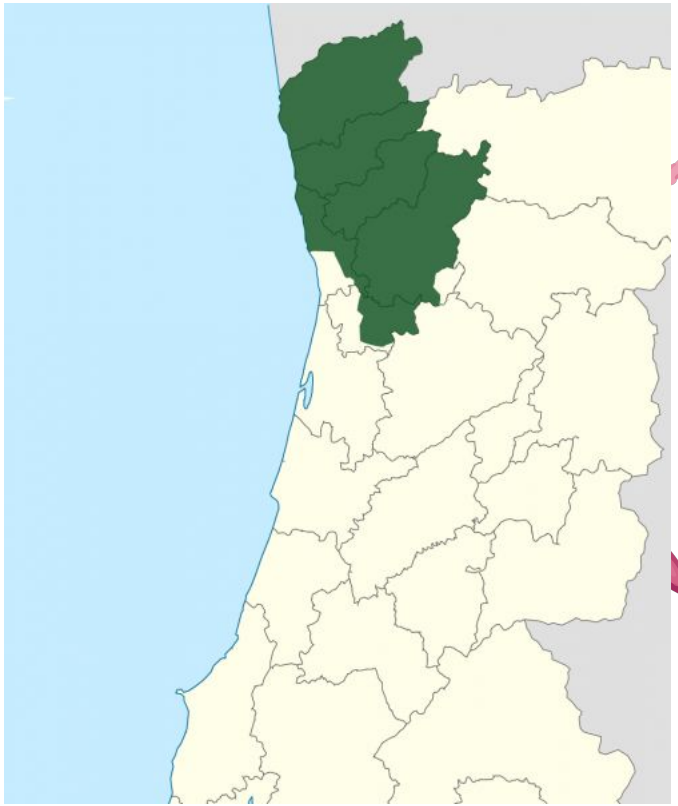
01. ORIGEN

Mi proyecto se centrará en el vino verde, un producto único de la región del Miño (noroeste de Portugal).

De graduación alcohólica media, es especialmente apreciado por su frescura (especialmente en verano).

Este vino representa el 15% de la producción total portuguesa, y alrededor del 10% se exporta, principalmente vino blanco.

01. ORIGEN



02. DATOS

En este DataSet se analizan las dos variantes más comunes, blanco y rosado, de la región denominación de origen del vino verde.
Las muestras se analizaron en la entidad de certificación oficial (CVRVV). La CVRVV es una organización interprofesional cuyo objetivo es mejorar la calidad y la comercialización del vino verde.
Los datos se registraron mediante un sistema informático (iLab), que gestiona automáticamente el proceso de análisis de muestras de vino, desde la solicitud del productor hasta el análisis de laboratorio y sensorial.

02. DATOS

```
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   wine_type             6497 non-null   object
 1   fixed acidity         6497 non-null   float64
 2   volatile acidity      6497 non-null   float64
 3   citric acid           6497 non-null   float64
 4   residual sugar        6497 non-null   float64
 5   chlorides             6497 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6497 non-null   float64
 10  sulphates             6497 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64
```

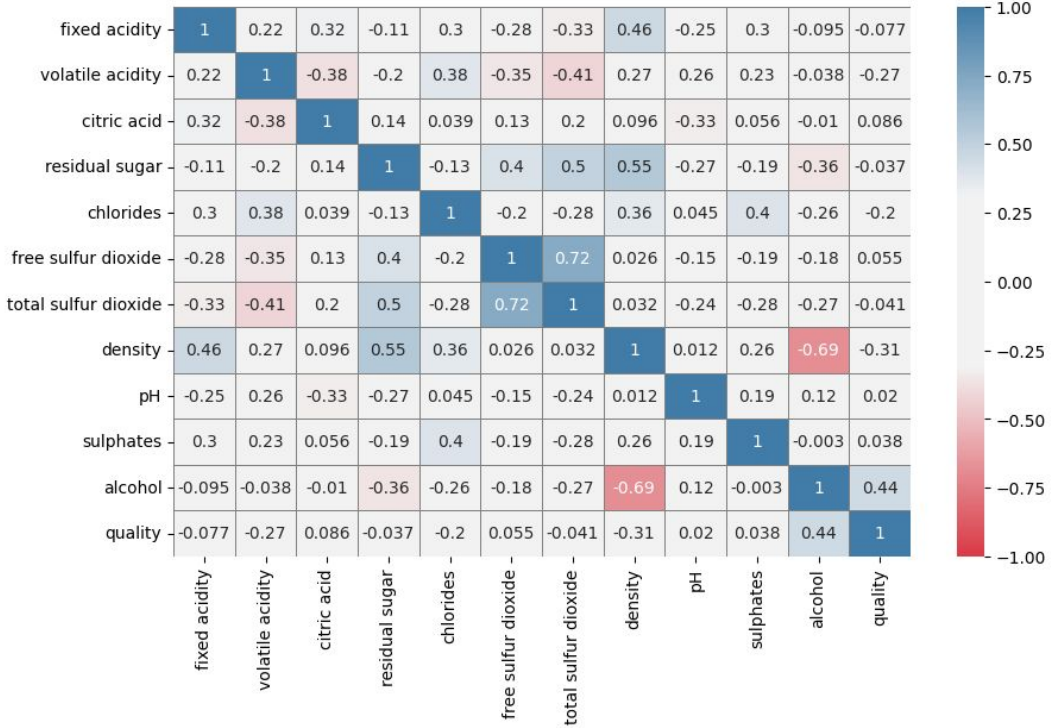
Input variables (basadas en análisis físico-químicos):

1 - Niveles muy altos = sabor agresivo. (4-10 g/L)
2 - En exceso olor a vinagre. (< 0.6 g/L)
3 - Da frescura pero puede ser artificial. (0-1 g/L)
4 - Dulzor. Seco (<4 g/L), semiseco (4-12 g/L), dulce (>12 g/L).
5 - En exceso = sabor salado o "agua de mar". (0.01-0.1 g/L.)
6 - En exceso = olor a cerilla. (10-50 mg/L).
7 - Límites legales (ej. 200 mg/L en vinos rosados)
8 - Vinos secos ≈ 0.990-1.000 g/mL.
9 - Estabilidad microbiológica y color. Vinos típicos: 3.0-3.8.
10 - Afecta la fermentación y conservación. Típico: 0.5-1 g/L.
11 - Cuerpo y percepción de dulzor. Típico: 9-15%.

Target variable (basado en datos sensoriales):

12 - quality (score entre 0 y 10).La media de al menos tres evaluaciones hechas por expertos en vino.

02. LIMPIEZA



NOS ENCONTRAMOS ANTE UN MODELO DE CLASIFICACIÓN EN EL QUE LA VARIABLE TARGET SE ENCUENTRA ENTRE 1 -10. REDUCIMOS LAS VARIABLES A TRES CALIDADES:

DESORDENAMOS LOS DATOS YA QUE ESTABAN PRIMERO LOS BLANCOS Y LUEGOS LOS TINTOS

02. LIMPIEZA

1.- NOS ENCONTRAMOS ANTE UN MODELO DE CLASIFICACIÓN EN EL QUE LA VARIABLE TARGET SE ENCUENTRA ENTRE 1 -10. REDUCIMOS LAS VARIABLES A TRES CALIDADES:

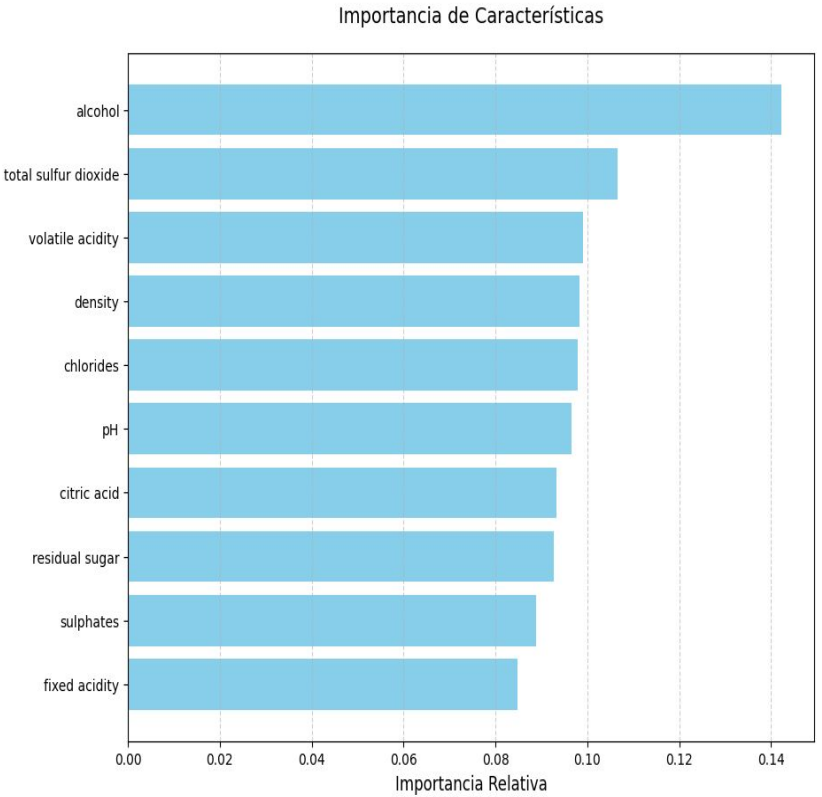
- BAJA (1 - 5)
- MEDIA (6 - 7)
- ALTA (8 - 10)

2.- DESORDENAMOS LOS DATOS YA QUE ESTABAN PRIMERO LOS BLANCOS Y LUEGOS LOS TINTOS

3.- BALANCEAMOS LOS DATOS YA QUE TENEMOS MUY POCOS VALORES PARA VINOS DE 'ALTA' CALIDAD

02. FEATURES

```
RangeIndex: 6497 entries, 0 to 6496
Data columns (total 14 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   wine_type             6497 non-null   object
 1   fixed acidity         6497 non-null   float64
 2   volatile acidity      6497 non-null   float64
 3   citric acid           6497 non-null   float64
 4   residual sugar        6497 non-null   float64
 5   chlorides             6497 non-null   float64
 6   free sulfur dioxide   6497 non-null   float64
 7   total sulfur dioxide  6497 non-null   float64
 8   density               6497 non-null   float64
 9   pH                    6497 non-null   float64
 10  sulphates             6497 non-null   float64
 11  alcohol               6497 non-null   float64
 12  quality               6497 non-null   int64
 13  quality_category      6497 non-null   object
```



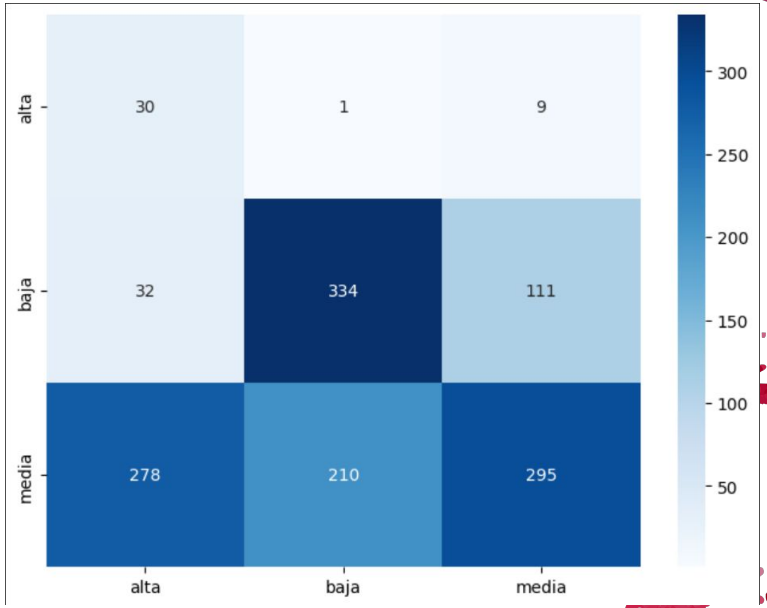
03. MODELOS

¡¡¡ COMENZAMOS CON LOS MODELOS !!!

03. MODELOS

EN EL PUESTO NUMERO 5: LOGISTIC REGRESSION

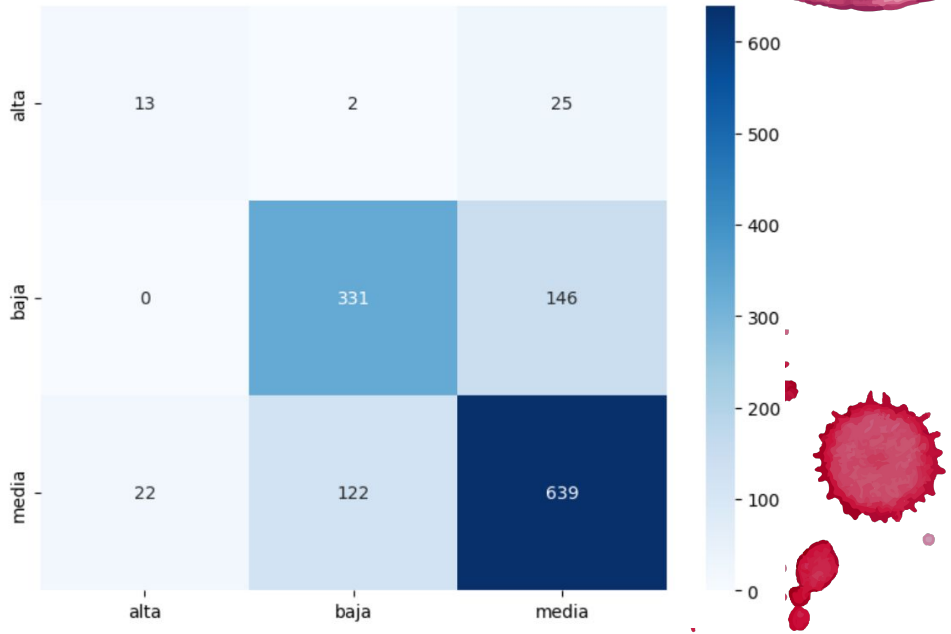
```
Accuracy: 0.5069230769230769
Precision (macro): 0.4706409014362902
Recall (macro): 0.608988570005703
F1-Score (macro): 0.4346675227414936

Matriz de Confusión:
[[ 30   1   9]
 [ 32 334 111]
 [278 210 295]]

Reporte de Clasificación:
              precision    recall  f1-score   support

        alta       0.09      0.75      0.16        40
        baja       0.61      0.70      0.65       477
       media       0.71      0.38      0.49       783

    accuracy                           0.51      1300
   macro avg       0.47      0.61      0.43      1300
weighted avg       0.66      0.51      0.54      1300

ROC-AUC

np.float64(0.7704415322927719)
```



03. MODELOS

EN EL PUESTO NUMERO 4: XGBClassifier

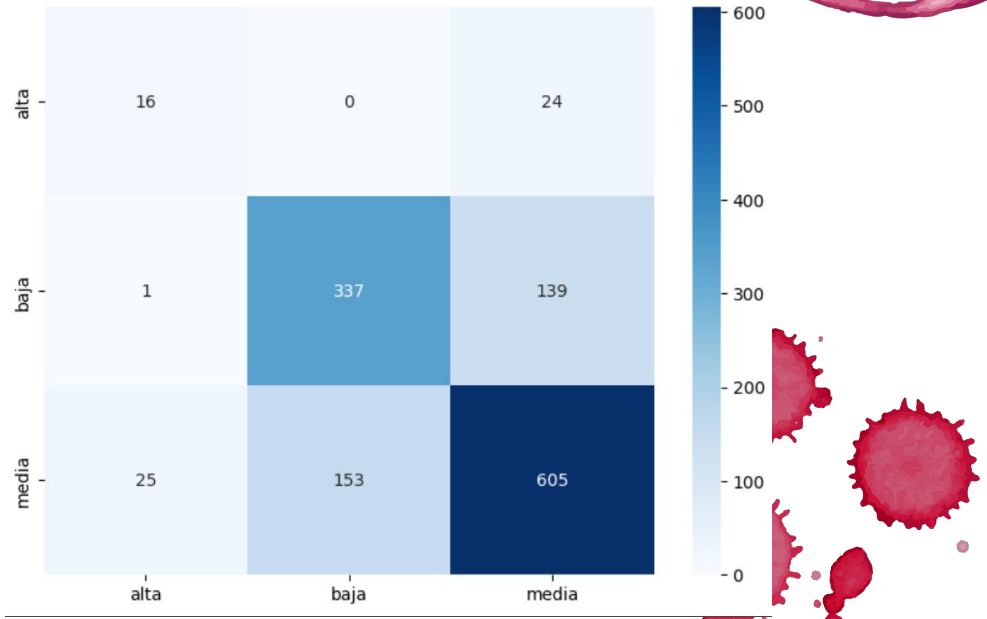
```
Accuracy: 0.7561538461538462
Precision (macro): 0.6292633292633293
Recall (macro): 0.6116707631509193
F1-Score (macro): 0.6197423276189552

Matriz de Confusión:
[[ 13   2  25]
 [  0 331 146]
 [ 22 122 639]]

Reporte de Clasificación:
              precision    recall  f1-score   support

        alta       0.37      0.33      0.35        40
        baja       0.73      0.69      0.71       477
       media       0.79      0.82      0.80       783

    accuracy                           0.76      1300
   macro avg       0.63      0.61      0.62      1300
weighted avg       0.75      0.76      0.75      1300

ROC-AUC

np.float64(0.8368389508113315)
```



03. MODELOS

EN EL PUESTO NUMERO 3: RandomForestClassifier

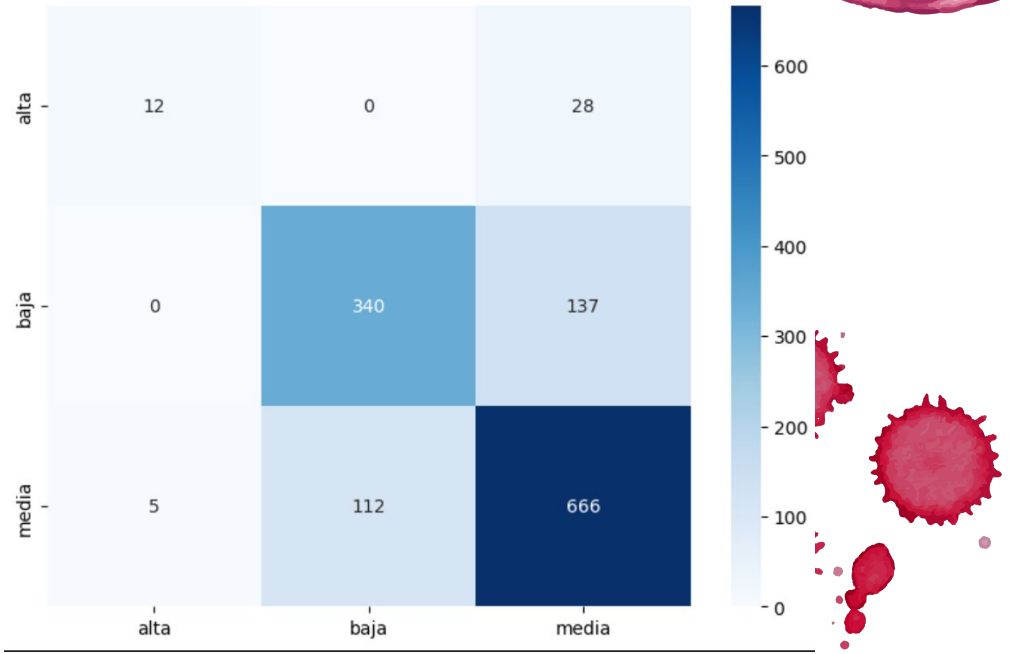
```
Accuracy: 0.7369230769230769
Precision (macro): 0.6188226332199546
Recall (macro): 0.626389390909018
F1-Score (macro): 0.6224622601789397

Matriz de Confusión:
[[ 16   0  24]
 [  1 337 139]
 [ 25 153 605]]

Reporte de Clasificación:
              precision    recall  f1-score   support

        alta       0.38      0.40      0.39        40
        baja       0.69      0.71      0.70       477
       media       0.79      0.77      0.78       783

    accuracy                           0.74      1300
   macro avg       0.62      0.63      0.62      1300
weighted avg       0.74      0.74      0.74      1300

ROC-AUC

np.float64(0.8440844719451794)
```



03. MODELOS

EN EL PUESTO NUMERO 2: RandomForestClassifier_GS

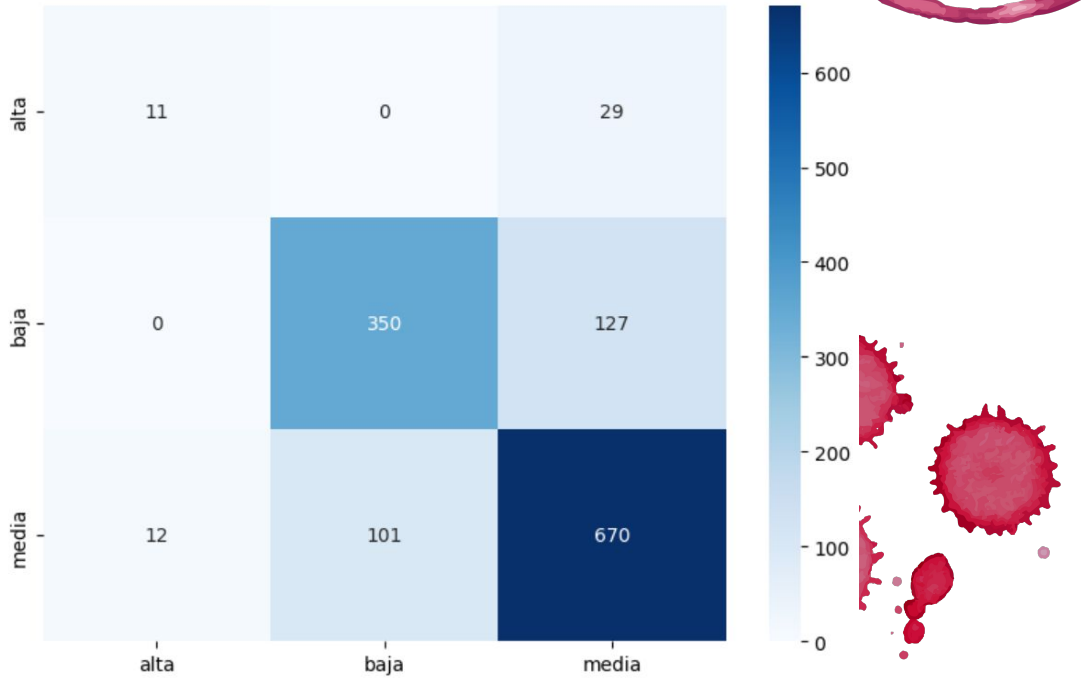
```
Accuracy: 0.7830769230769231
Precision (macro): 0.7531795952143358
Recall (macro): 0.6211209908672498
F1-Score (macro): 0.6594337673508183

Matriz de Confusión:
[[ 12   0  28]
 [  0 340 137]
 [  5 112 666]]

Reporte de Clasificación:
              precision    recall  f1-score   support

       alta       0.71      0.30      0.42        40
       baja       0.75      0.71      0.73       477
      media       0.80      0.85      0.83       783

    accuracy                           0.78      1300
   macro avg       0.75      0.62      0.66      1300
weighted avg       0.78      0.78      0.78      1300

ROC-AUC

np.float64(0.8516650747925855)
```



03. MODELOS

EN EL PUESTO NUMERO 1: AbaBoostClassifier

```
Accuracy: 0.7930769230769231
Precision (macro): 0.6884840330568892
Recall (macro): 0.6214786300071488
F1-Score (macro): 0.6454440357780405

Matriz de Confusión:
[[ 11   0  29]
 [  0 350 127]
 [ 12 101 670]]

Reporte de Clasificación:
              precision    recall  f1-score   support

        alta       0.48      0.28      0.35        40
        baja       0.78      0.73      0.75       477
       media       0.81      0.86      0.83       783

    accuracy                           0.79      1300
   macro avg       0.69      0.62      0.65      1300
weighted avg       0.79      0.79      0.79      1300

ROC _ AUC

np.float64(0.8917342290625161)
```



03. MODELOS

RESUMEN

	ROC AUC	ACCURACY
AdaBoostClassifier	0.873785	0.787692
RandomForestClassifier_GS	0.851665	0.783077
RandomForestClassifier	0.844084	0.736923
XGBClassifier	0.836839	0.756154
LogisticRegression	0.770442	0.506923

CONCLUSIONES

El **AdaBoostClassifier** destaca como el mejor modelo por las siguientes razones:

- **El ROC AUC más alto:** Este es un fuerte indicador de su capacidad para discriminar entre las clases de manera efectiva, especialmente si hay algún desequilibrio de clases
- **La Precisión más alta:** También logra la mejor precisión general, lo que significa que realiza la mayor cantidad de predicciones correctas.
- **Fuerte Rendimiento en Todas las Clases (según la Matriz de Confusión):** AdaBoost muestra un buen equilibrio de predicciones correctas en las tres clases ("alta", "baja", "media"), con menos clasificaciones erróneas críticas en comparación con otros modelos. El objetivo en una matriz de confusión es tener números altos a lo largo de la diagonal.

Por lo tanto, considerando tanto las métricas resumidas como los detalles de las matrices de confusión, el **AdaBoostClassifier** es el modelo más robusto y con mejor rendimiento entre los evaluados.

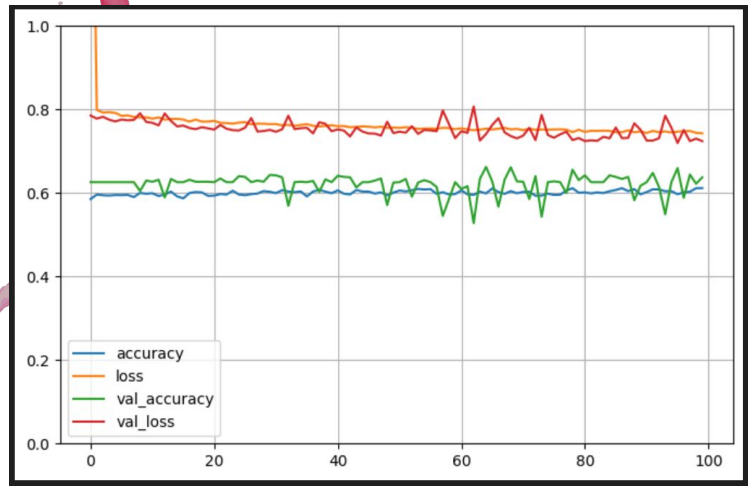
POR ÚLTIMO PERO FUERA DEL CONCURSO

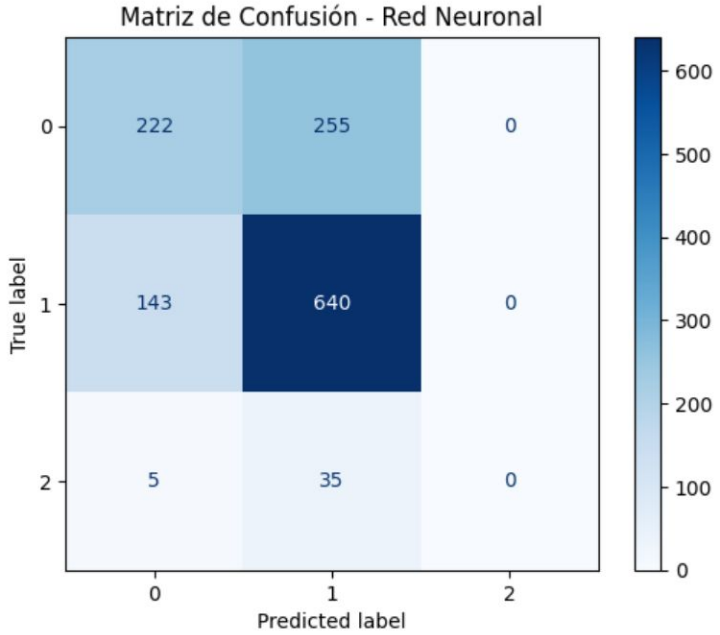
RED NEURONAL REENTRENADA

```
capas = [
    keras.layers.Flatten(input_dim=5),
    keras.layers.Dense(units = 300, activation='relu'),
    keras.layers.Dense(units = 64, activation='relu'),
    keras.layers.Dense(units = 3, activation='softmax')
]
```

POR ÚLTIMO PERO FUERA DEL CONCURSO

```
41/41 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.6624 - loss: 0.7100
```





Aunque las métricas no eran del todo malas, la matriz de confusión....

GRACIAS