

## Módulo 2: Uso de framework o biblioteca de aprendizaje máquina para la implementación de una solución.

Daniel Saldaña Rodríguez  
A00829752

Para esta implementación de un método de aprendizaje de datos se seleccionó una base de datos proveniente de Kaggle, esta base de datos cuenta con 13 variables que son usadas para predecir si el paciente tiene un riesgo elevado de sufrir un ataque al corazón. Hay solo dos posibles resultados, 0 (No hay un riesgo elevado), y 1 (Hay un riesgo elevado), por lo que este es un problema de clasificación binaria. Entre estas variables podemos encontrar la edad, sexo, niveles de colesterol, tipo de dolor de pecho y nivel de azúcar entre otras. La base de datos puede ser encontrada en esta liga:

<https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis-prediction-dataset>

Primero se utilizaron una variedad de modelos para encontrar el que tuviera el mejor rendimiento, a continuación se puede ver cada modelo utilizado y el accuracy score que obtuvo:

- K-Nearest Neighbor: 0.590164
- Support Vector Classifier: 0.655738
- Regresión Logística: 0.803279
- Decision Tree Classifier: 0.737705
- Gaussian Naive Bayes: 0.819672
- Random Forest Classifier: 0.803279
- Gradient Boosting Classifier: 0.819672
- MLPClassifier: 0.836066

De estos se eligieron Gaussian Naive Bayes, Random Forest y Gradient Boosting para analizar más a fondo, MLP tuvo un resultado más alto, sin embargo, se concluyó que el resultado variaba mucho entre cada iteración.

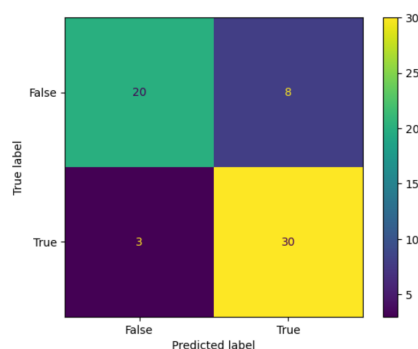
### Gaussian Naive Bayes:

Accuracy: 0.81967

Precisión: 0.78947

Recall: 0.90909

F1: 0.84507



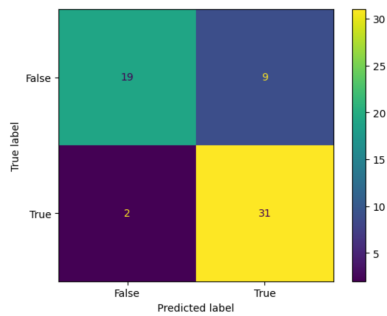
### Random Forest Classifier:

Accuracy: 0.81967

Precisión: 0.775

Recall: 0.93939

F1: 0.84932



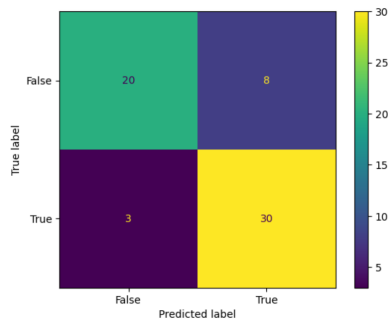
### Gradient Boosting Classifier:

Accuracy: 0.81967

Precisión: 0.78947

Recall: 0.90909

F1: 0.84507



Se puede ver que estos tres modelos tienen estadísticas muy similares, se seleccionó Gradient Boosting para hacer un ajuste de hiper parámetros y se obtuvieron los siguientes resultados:

```
GradientBoostingClassifier(learning_rate=0.01, max_depth=4)
```

Accuracy: 0.85246

Precisión: 0.78571

Recall: 1.0

F1: 0.88

