INFORME FINAL - INTRODUCCIÓN CIENCIA DE DATOS

Detección de cáncer de mama

Universidad Católica Argentina

Facultad de Química e Ingeniería

Licenciatura en Ciencia de Datos

Estudiantes: Chocobares Juan Cruz,

Formenti Agustin,

Mendes Lorenzo,

Morenico Andres.

Docente: Pavon Claudio Gonzalo.

Fecha de entrega: 25 de junio de 2025.

Detección de Cáncer de Mama con Machine Learning

1. Introducción

En este proyecto aplicamos técnicas de Machine Learning supervisado para clasificar cáncer de mama como benignos o malignos usando el dataset “Breast Cancer Wisconsin (Diagnostic)”. Para asegurar un buen flujo de trabajo y colaboración, versionamos todo en GitHub, organizando el código en carpetas (notebooks/, data/, reports/) y documentando cada paso con commits claros y un README en cada carpeta.

A. Glosario de Variables

Variables	Descripción
radius_mean	Radio promedio del núcleo celular (media del radio medido en píxeles).
texture_mean	Desviación estándar de los niveles de gris; mide variaciones en la textura.
perimeter_mean	Longitud promedio del contorno del núcleo.
area_mean	Área promedio del núcleo.
smoothness_mean	Suavidad promedio: variación local en el contorno.
compactness_mean	Compacticidad promedio: (perímetro^2/área) - 1.
concavity_mean	Concavidad promedio: gravedad de las concavidades en el contorno.
concave points_mean	Cantidad promedio de puntos cóncavos en el contorno.
symmetry_mean	Simetría promedio de la forma del núcleo.
fractal_dimension_mean	Dimensión fractal promedio: complejidad del contorno ($1 \leq$ valor ≤ 2).
radius_se	Error estándar del radio.
texture_se	Error estándar de la textura.
perimeter_se	Error estándar del perímetro.
area_se	Error estándar del área.
smoothness_se	Error estándar de la suavidad.
compactness_se	Error estándar de la compacticidad.

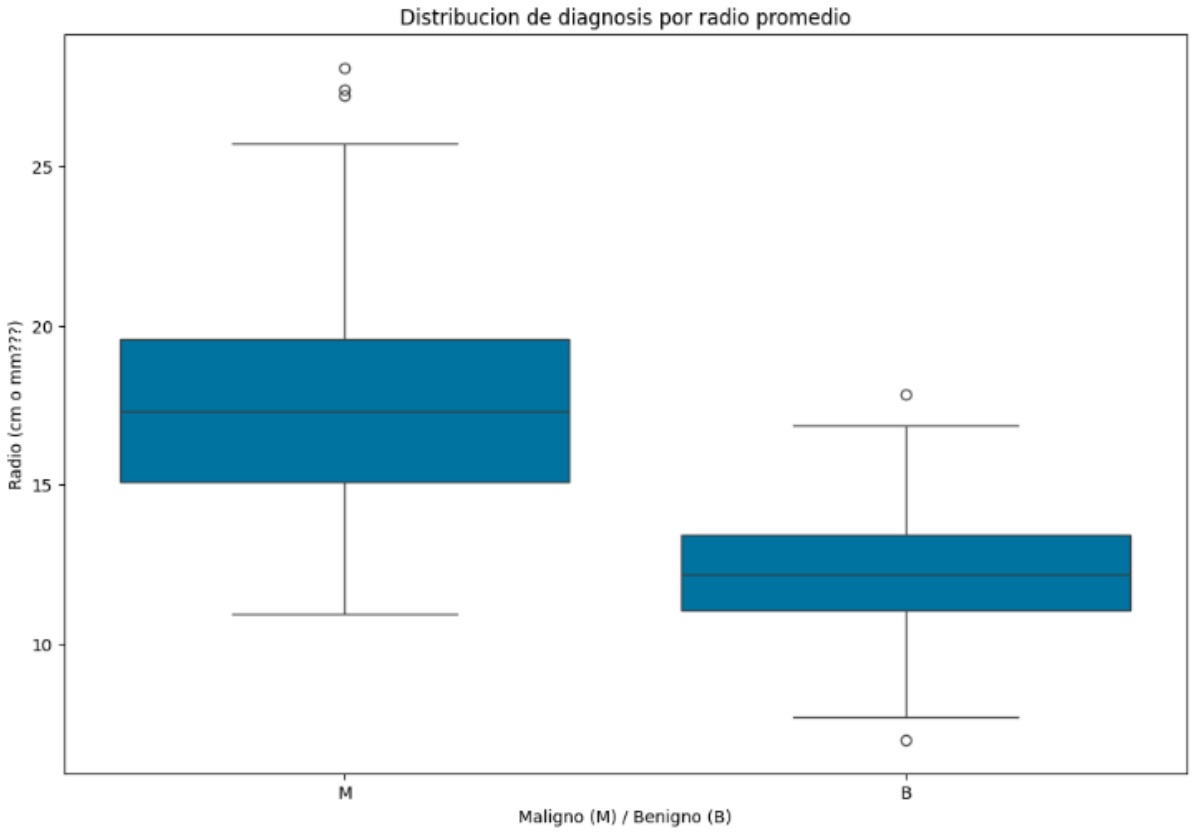
concavity_se	Error estándar de la concavidad.
concave points_se	Error estándar de los puntos cóncavos.
symmetry_se	Error estándar de la simetría.
fractal_dimension_se	Error estándar de la dimensión fractal.
radius_worst	Peor (máximo) radio observado.
texture_worst	Peor (máxima) textura observada.
perimeter_worst	Peor (máximo) perímetro observado.
area_worst	Peor (máxima) área observada.
smoothness_worst	Peor (máxima) suavidad observada.
compactness_worst	Peor (máxima) compacticidad observada.
concavity_worst	Peor (máxima) concavidad observada.
concave points_worst	Peor (máximo) número de puntos cóncavos.
symmetry_worst	Peor (máxima) simetría observada.
fractal_dimension_worst	Peor (máxima) dimensión fractal observada.

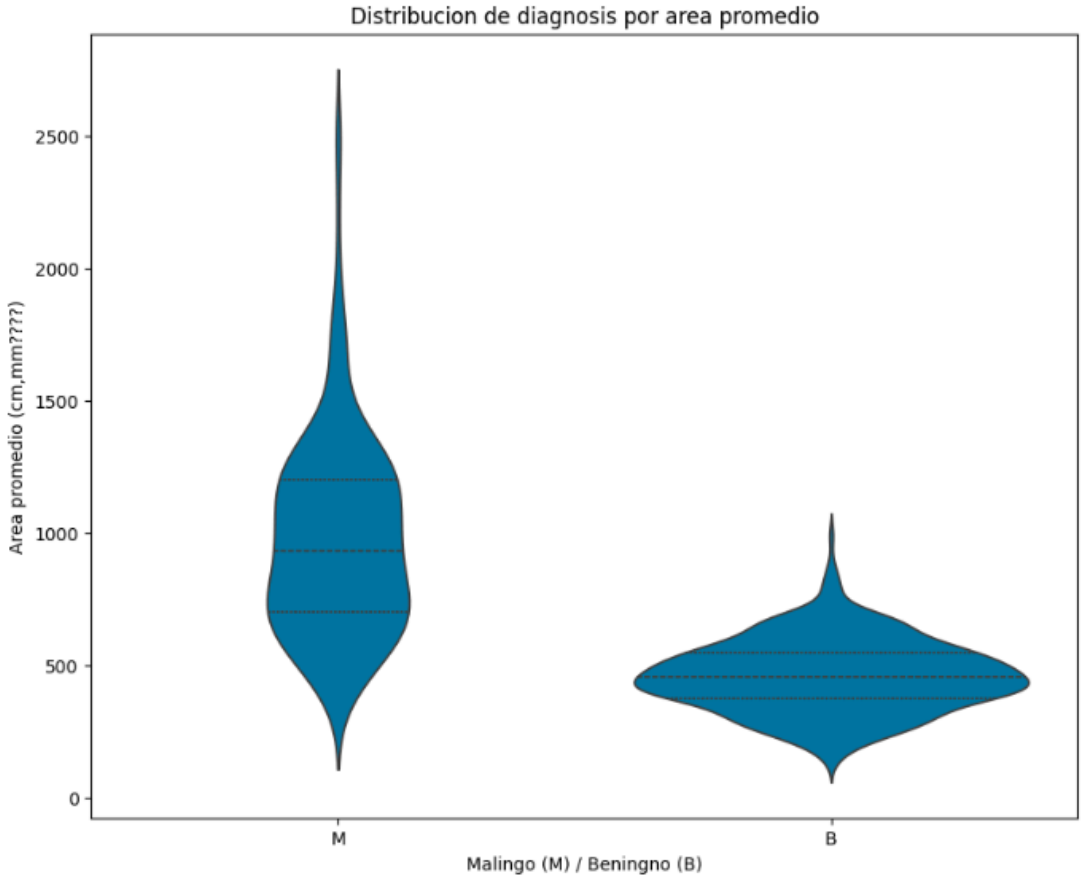
B. Explicación de Métricas de Evaluación

Métrica	Definición
Accuracy	(TP + TN) / (TP + TN + FP + FN) - Porcentaje de predicciones correctas en total.
Precision	TP / (TP + FP) - De las predicciones 'maligno', proporción realmente correctas.
Recall	TP / (TP + FN) - De los verdaderos malignos, proporción detectada correctamente.
F1-Score	2*(precision*recall)/(precision+recall) - Media armónica de precision y recall.
ROC AUC	Área bajo la curva ROC; mide la capacidad global de separación de clases.

2. Exploración y Visualización Inicial

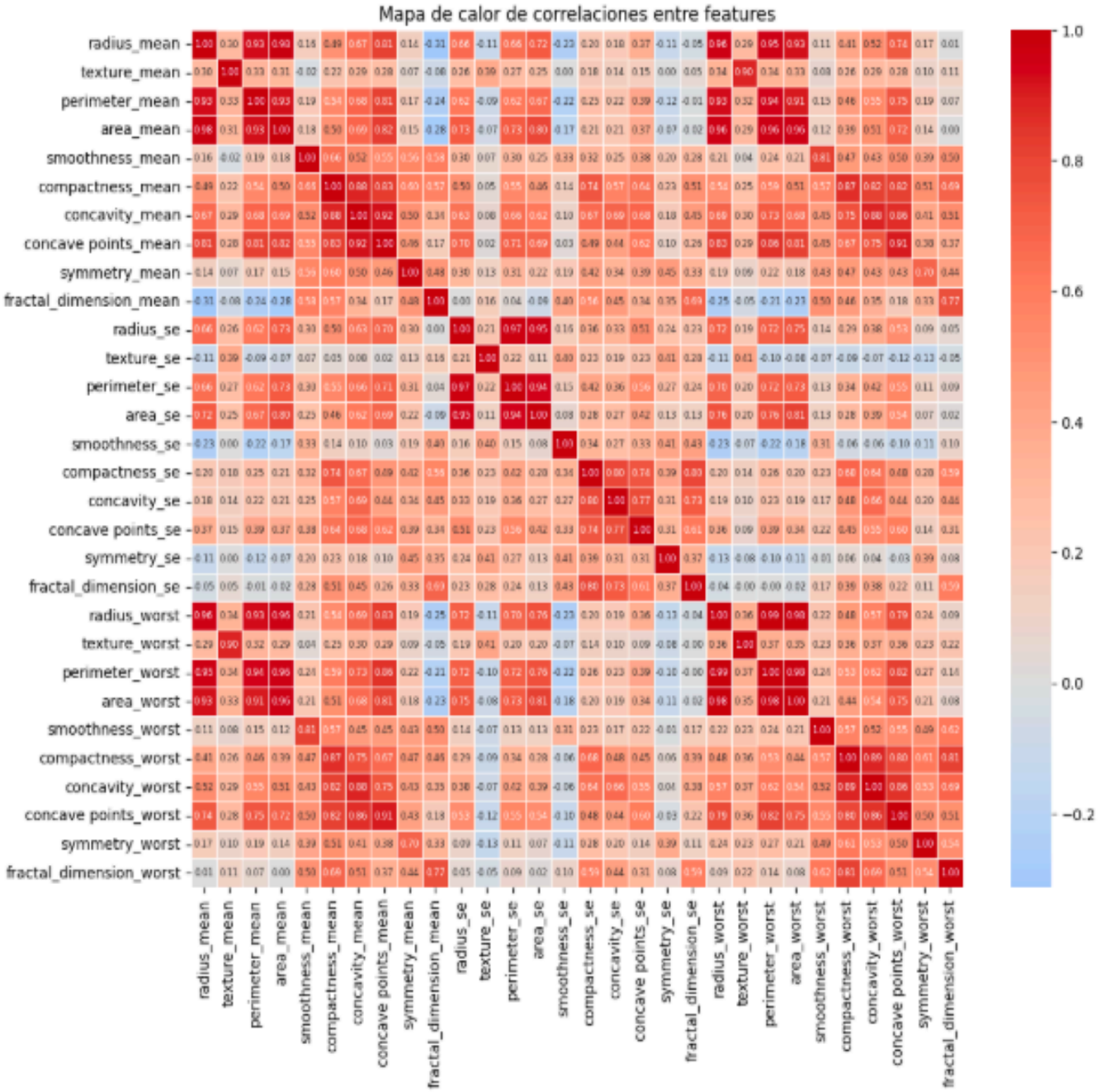
• Se realizaron boxplots y violin plots para entender distribuciones y outliers.





• Se realizaron estudios individuales de cada una de las columnas que contenían outliers ([“radius_mean”], [“texture_mean”], [“perimeter_mean”])

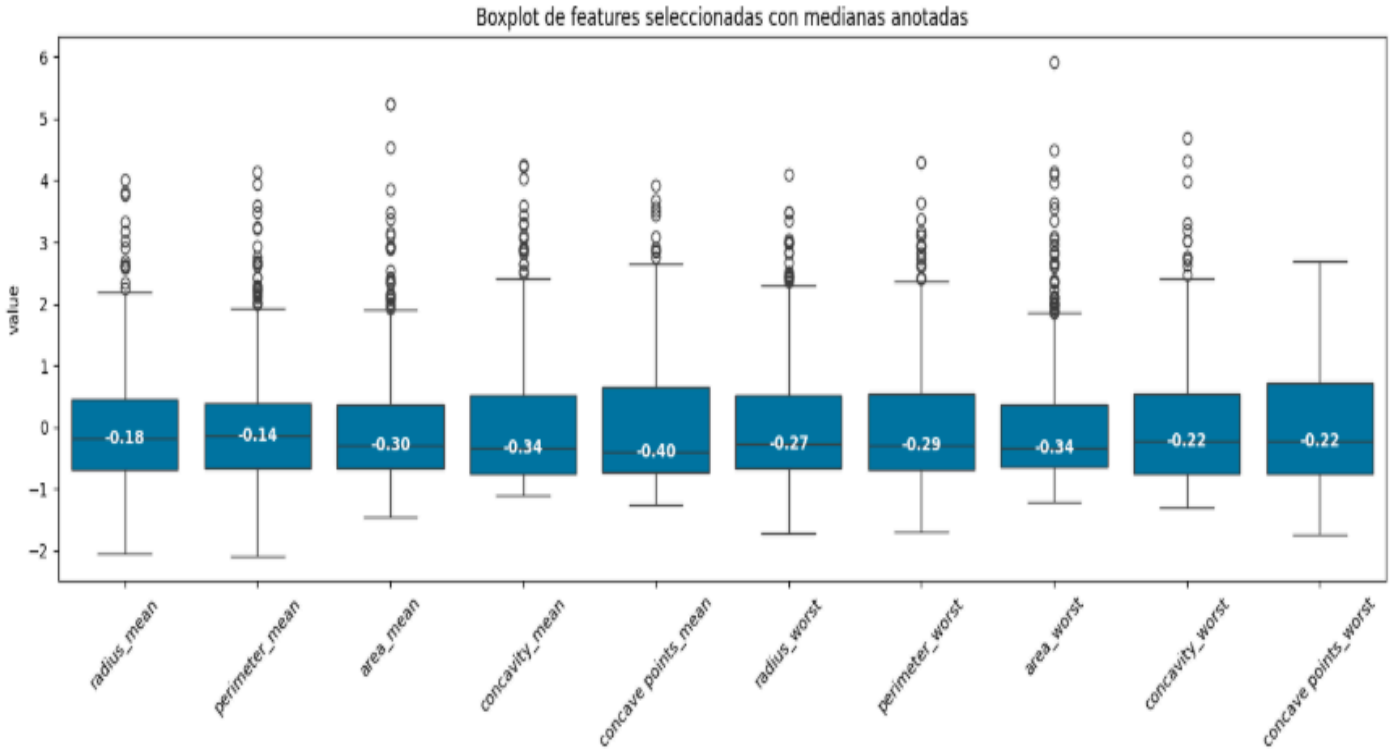
- Se incluyó un heatmap de correlaciones para detectar relaciones lineales.



3. Limpieza y Preprocesamiento

• Eliminación de columnas irrelevantes ([“id”], [“Unnamed: 32”]).
• Codificación de diagnosis de B/M a 0/1.
• La limpieza se realizó con la ayuda de la herramienta SelectKBest, donde seleccionamos las 10 mejores features en base al test ANOVA F que mide la varianza entre clases de cada variable. Esto se justifica por la falta de conocimiento médico de quien realiza el modelo, evitando cualquier tipo de sesgo.

• Se realizó un estudio sobre las 10 features seleccionadas, construyendo boxplots y chequeando la cantidad de outliers de cada una.



	variable	outliers	pct_outliers
7	area_worst	35	6.151142
2	area_mean	25	4.393673
1	perimeter_mean	24	4.217926
3	concavity_mean	18	3.163445
5	radius_worst	17	2.987698
6	perimeter_worst	15	2.636204
0	radius_mean	14	2.460457
8	concavity_worst	12	2.108963
4	concave points_mean	10	1.757469
9	concave points_worst	0	0.000000

• Se utilizó RobustScaler para escalar features y mitigar outliers, ya que el estudio de las variables deja en claro una gran cantidad de valores atípicos en cada una, y este tipo de escalado, es muy amigable con estos tipos de valores.

4. Modelos Entrenados

Se entrenaron tres modelos:
1. Regresión Logística (baseline sin ajuste de hiper parámetros).
2. Random Forest (GridSearchCV para n_estimators, max_depth, min_samples_split).
3. Decision Tree (GridSearchCV para max_depth, min_samples_split, criterion).

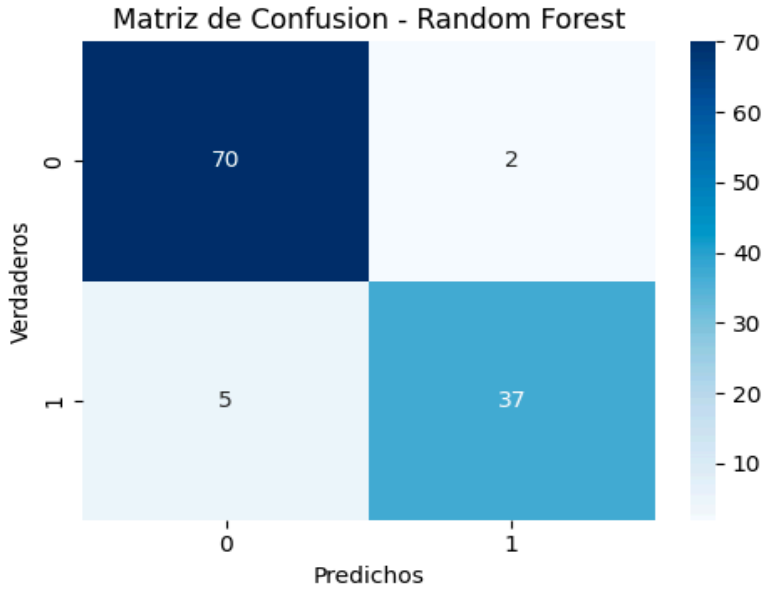
5. Resultados

Se evaluaron métricas: Accuracy, Precision, Recall, F1-Score y ROC AUC.

Tabla comparativa y curvas ROC indicaron que Random Forest obtuvo el mayor ROC AUC, mientras que Regresión Logística ofreció excelente equilibrio y simplicidad.

• Random Forest: Mejor ROC AUC (99%)

```
=== Random Forest (GridSearchCV) ===
Mejores parámetros: {'max_depth': None, 'min_samples_split': 10, 'n_estimators': 50}
Accuracy:  0.9386
Precision: 0.9487
Recall:    0.8810
F1 Score:  0.9136
ROC AUC:   0.9911
```



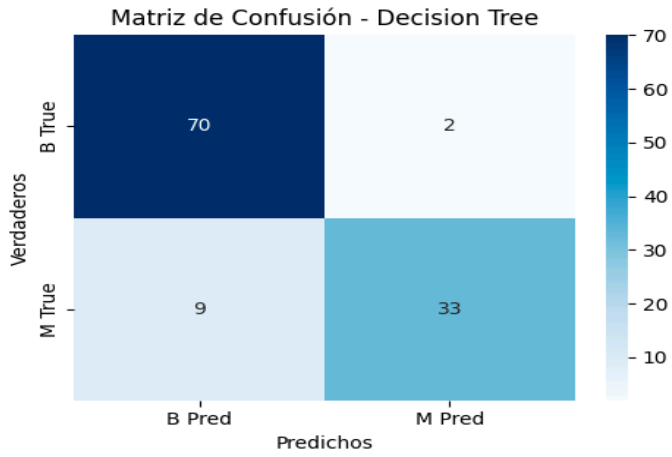
- Más métricas por clase:

```
Accuracy benigno (clase 0):  0.9722
Accuracy maligno (clase 1):  0.8810

Classification Report:
```

	precision	recall	f1-score	support
Benigno	0.93	0.97	0.95	72
Maligno	0.95	0.88	0.91	42

• Decision Tree: Quedó un poco rezagado (AUC ~0.93)

```
=== Decision Tree (GridSearchCV) ===
Mejores parámetros: {'criterion': 'entropy', 'max_depth': 5, 'min_samples_leaf': 4, 'min_samples_split': 10}
Accuracy:  0.9035
Precision: 0.9429
Recall:    0.7857
F1 Score:  0.8571
ROC AUC:   0.9643
```

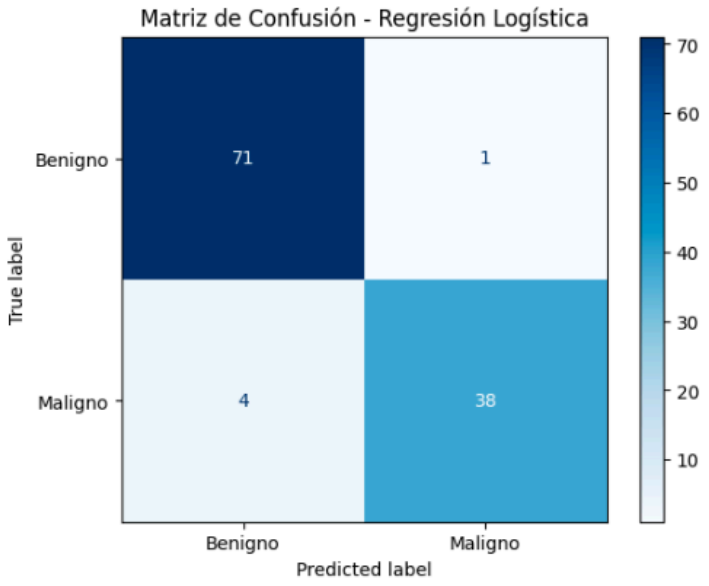


- Más métricas por clase:

```
Accuracy benigno (clase 0) Decision Tree: 0.9722
Accuracy maligno (clase 1) Decision Tree: 0.7857

Classification Report - Decision Tree:

              precision    recall  f1-score   support

     Benigno       0.89      0.97      0.93        72
     Maligno       0.94      0.79      0.86        42
```

• Regresión Logística: Alcanzó AUC ~0.95 y es fácil de explicar (coeficientes).

```
=== Regresión Logistica ===
Accuracy:  0.9561
Precision: 0.9744
Recall:    0.9048
F1 Score:  0.9383
ROC AUC:   0.9974
```

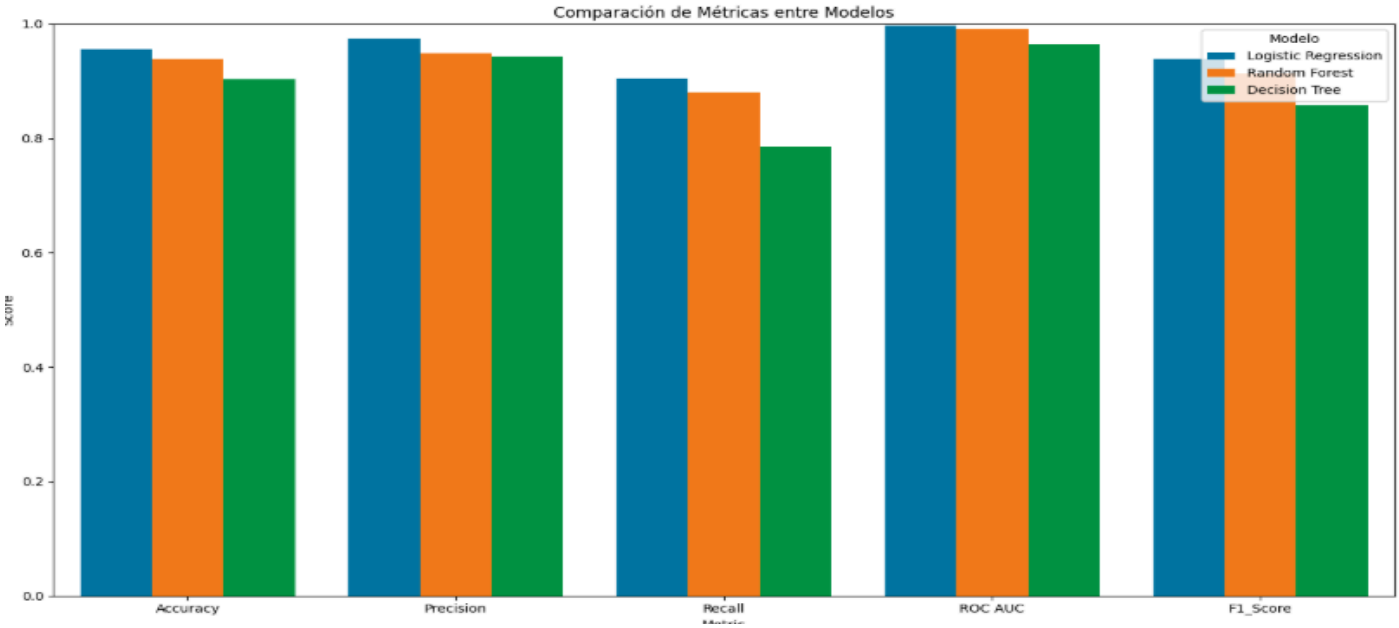


- Más métricas por clase:

```
Modelo: Regresión Logística
              precision    recall  f1-score   support

     Benigno       0.95      0.99      0.97        72
     Maligno       0.97      0.90      0.94        42
```

6. Comparación de modelos

Realizamos una comparación entre modelos para ver su rendimiento en cada métrica

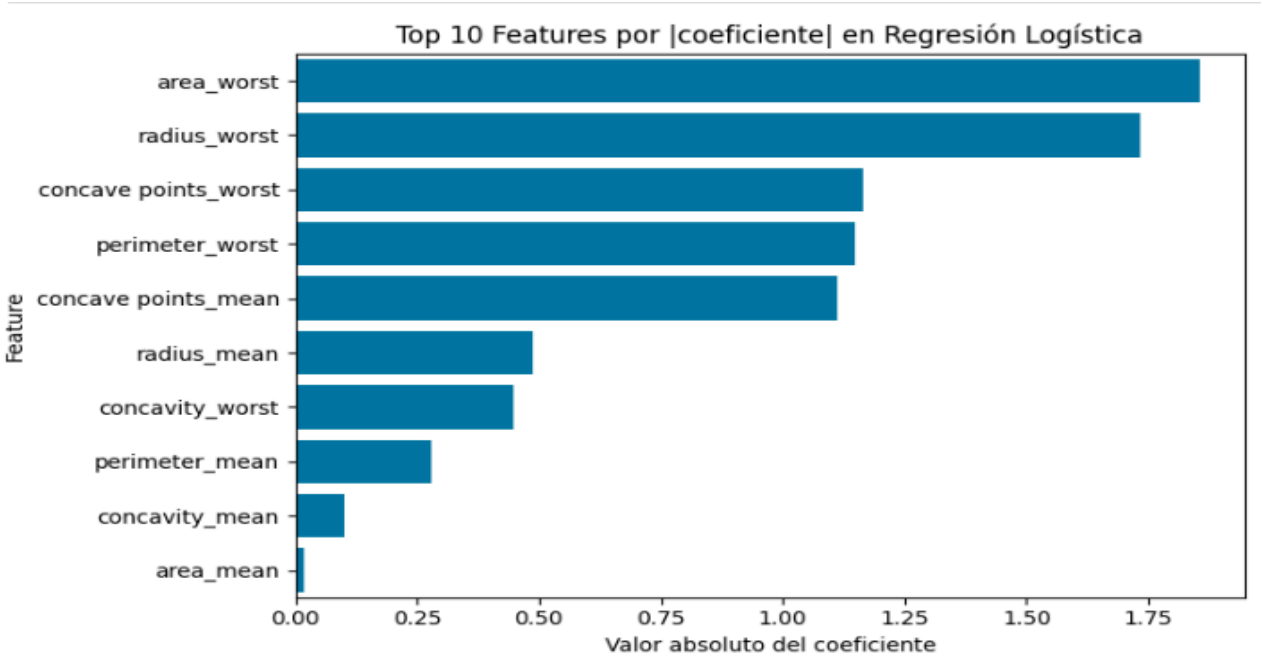



```
=== Comparación de Modelos ===
                 Model  Accuracy  Precision    Recall   ROC AUC  F1_Score
0  Logistic Regression  0.956140   0.974359  0.904762  0.997354  0.938272
1        Random Forest  0.938596   0.948718  0.880952  0.991071  0.913580
2        Decision Tree  0.903509   0.942857  0.785714  0.964286  0.857143
```

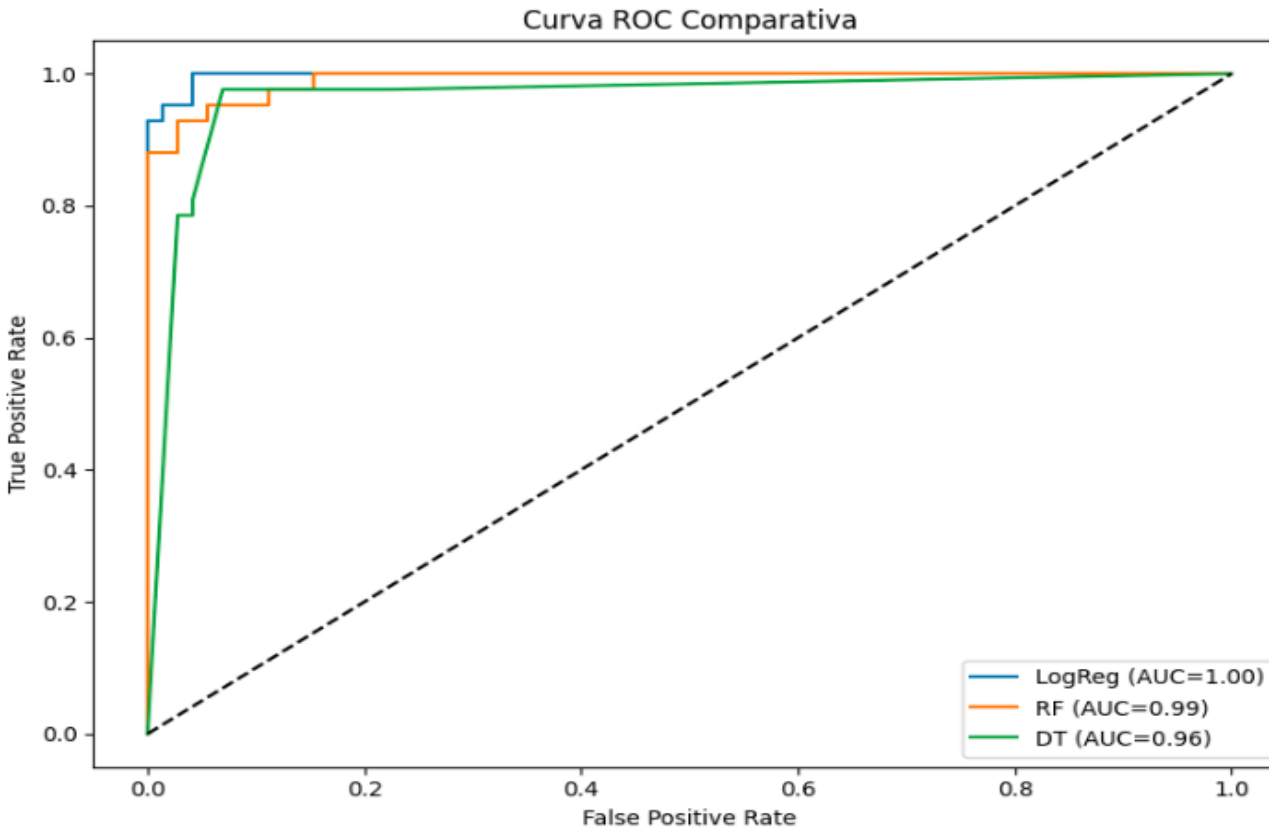
7. Conclusión

Recomendamos implementar el modelo de Regresión Logística en producción por su balance entre performance e interpretabilidad.

Además, se realizó un estudio de las 10 mejores features de este modelo, para entender cuáles eran las que más impacto tienen sobre él.



Para finalizar, se agregó un gráfico sobre las curvas ROC comparativas que denotan que modelo es mejor al separar clases.



8. Versionamiento y Repositorio

- GitHub: Se utilizaron distintas branchs para garantizar un buen trabajo y /README en cada carpeta.
- Colab: El modelo fue creado en colab para poder tener una buena dinámica.
- Commits descriptivos en cada etapa.

9. Notas:

- Todo el trabajo fue realizado en lenguaje Python.
- Los modelos se han entrenado utilizando la librería scikit-learn.
- Los gráficos y tablas se realizaron utilizando las librerías Matplotlib y Seaborn

10. Referencias:

- Kaggle: Breast Cancer Dataset.