Resultados y Evaluación de Modelos: Clasificación de email de spam: pre-procesamiento y baselines

Adrian Pineda Sanchez
A00834710

Regresión Logística:

```
DEBUG:: Los labels completos de regresión logística son:
[0 0 0 1 1 1 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0
 0 1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 1 0
 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 0 0 1 0 0 1 1 0
 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1 0 1 0 1
 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 0
 1 1 1 1 1 1 0 1 1 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 0 0
 0 1 0 1 0 0 1 0 1 1 1 0 0 1 1 0 1 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 0 1 0 1
 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 1 1
 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1
 0 0 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1
 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 1 0 1 1 1 0
 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1
 0 0 0 1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1
 1 1 1 0 0 0 1 0 1 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 1 0 1 1 0 0 1 0 0 1 1 1 1 0
 1 0 0 1 1 1 1 0 0 1 0 1 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 0 1 0 1 1 1 0 1 1 1 0 0
 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 1 0
 0 0 0 1 0 1 1 1]
```

+ Code + Markdown

```
from sklearn.metrics import accuracy_score

acc_score = accuracy_score(test_y, predicted_labels)

print("DEBUG::El accuracy score de regresión logistica es::")
print(acc_score)
```

```
DEBUG::El accuracy score de regresión logística es::
0.9866666666666667
```

- Precisión (Accuracy): 0.9866
- Observaciones: La regresión logística proporcionó un excelente rendimiento, con una precisión del 98.66%. Es uno de los modelos más precisos y rápidos, lo que lo hace muy eficiente en problemas donde la clasificación lineal es suficiente.

Support Vector Classifier (SVC):

```
Entrenar el Clasificador SVC tomó  91 segundos
DEBUG::Las labels del Clasificador SVC son::
[0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 0 1 0 0 1 1 1 0 0 0 0 1 1 1 0 0 0
 0 1 0 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 0
 0 0 0 1 0 0 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 0 0 0 0 0 0 0 1 1 0
 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 1 1 0 1 0 1 0 1 0 1 0 0 0 1 1 0 0 1 0 1 0 0 0 1
 1 0 0 0 1 0 1 1 0 1 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0 0 0 0 0
 0 1 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 1 0 1 0 0 0 0 0 0 1 0 1 0 0
 0 1 0 1 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 0 1 0 1 0 0 1 1
 0 0 0 0 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1
 0 1 0 1 0 0 0 1 0 0 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 0 0
 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 0 1 0 1 1 1 0
 1 1 0 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0 1 1 1 1 0 0 0 1 0 1 1 0 0 0 0
 0 0 1 1 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1
 1 1 1 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0
 1 0 0 1 1 1 1 1 0 0 0 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 1 0 0 1 0 0 1 1 0 0
 1 1 1 0 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0
 0 0 0 1 0 0 1 0]
DEBUG::El accuracy score del Clasificador SVC es::
0.8016666666666666
```

- Precisión (Accuracy): 0.8016
- Tiempo de entrenamiento: 91 segundos
- Observaciones: A pesar de ser un modelo sólido, el SVC tuvo un rendimiento más bajo en comparación con otros modelos, con un accuracy score del 80.16%. Además, fue uno de los más lentos, con un tiempo de entrenamiento significativamente mayor.

Random Forest Classifier:

```
Entrenar el Random Forest Classifier tomó   3 segundos
DEBUG::Las etiquetas RF predecidas son::
[0 0 0 1 1 1 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0
 0 1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 1 0
 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 0 1 0 0 0 1 0 0 1 1 0
 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 1 0 1 0 1 0 0 0 1 1 1 0 1 0 1 0 1 0 1
 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 0
 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 1 1 1 0 0 1 1 1 0 1 1 1 0 0
 0 1 0 1 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1
 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 1 1 1 1 0 1 0 0 1 1
 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0 1 1 1 1 1 1
 0 0 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1
 0 1 1 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 1 1 1 0
 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1
 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 0 1
 1 1 1 0 0 0 1 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0
 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 1 0 0
 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0
 0 0 0 1 0 1 1 1]
DEBUG::El RF testing accuracy score es::
0.9783333333333334
```

- Precisión (Accuracy): 0.9783
- Tiempo de entrenamiento: 3 segundos
- Observaciones: Este modelo fue muy eficiente en términos de tiempo de entrenamiento (solo 3 segundos) y presentó una alta precisión del 97.83%. Esto lo hace una opción atractiva cuando se necesitan modelos rápidos y con buen rendimiento.

GridSearchCV con Random Forest:

```
Available hyper-parameters for systematic tuning available with RF:
{'bootstrap': True, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini',
'min_samples_leaf': 1, 'min_samples_split': 2, 'min_weight_fraction_leaf': 0.0, '
Fitting 3 folds for each of 27 candidates, totalling 81 fits
Los mejores parámetros encontrados:
{'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 100}
La accuracy estimada es:
0.9816666666666667
```

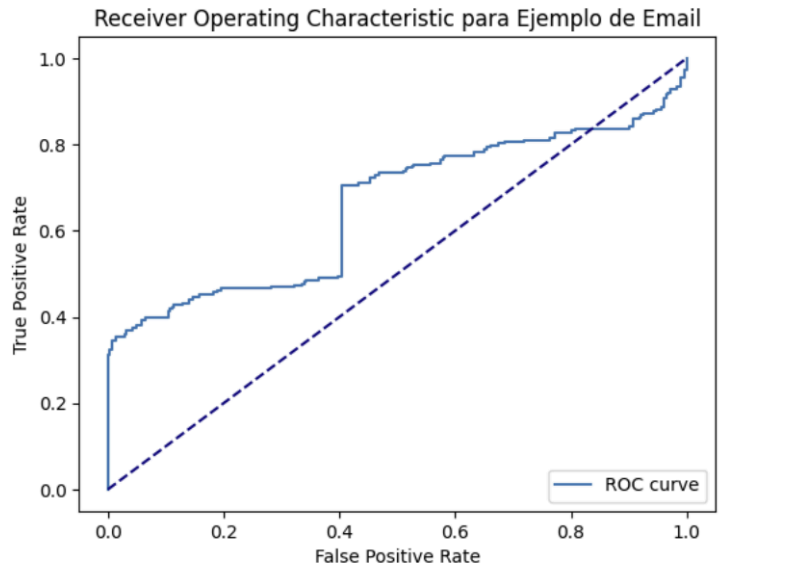
- Precisión estimada: 0.9816
- Observaciones: Después de ajustar los parámetros con GridSearchCV, el Random Forest mejoró levemente su precisión a un 98.16%. Este enfoque ayuda a obtener el mejor rendimiento posible ajustando los hiperparámetros del modelo.

Gradient Boosting Classifier:

```
Model Report
Accuracy : 0.9979
AUC Score (Train): 0.999586
CV Score : Mean - 0.99447 | Std - 0.002911059 | Min - 0.9908088 | Max - 0.9994894
El entrenamiento del Gradient Boosting Classifier tomó 262 segundos
DEBUG::Los labels predecidos de Gradient Boosting son::
[0 0 0 1 1 1 1 1 0 0 0 1 0 1 0 1 1 1 1 1 0 1 0 0 1 1 1 1 0 1 0 1 1 1 0 1 0
 0 1 1 0 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 1 0 1 1 1 0 0 1 1 0 0 1 0 1 0
 0 0 0 1 1 0 1 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 1 0 0 0 0 0 0 1 1 0
 1 1 0 0 1 1 1 0 1 1 0 0 0 0 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 0 1
 1 0 0 0 1 0 1 1 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 1 1 1 1 0 1 0 1 0 1 0 1 0
 1 1 1 1 1 1 0 1 1 1 1 0 0 1 0 0 0 1 1 0 1 0 1 1 1 1 0 0 0 1 1 0 1 1 1 0 0
 0 1 0 1 0 0 0 0 0 1 1 1 0 0 1 0 0 1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 0 1 0 1 0 1
 0 0 0 1 0 1 0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 1 0 0 0 1 1 1 0 1 0 1 0 0 1 1
 0 1 1 1 1 0 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 1 1 1 1
 0 0 0 1 1 1 1 1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 0 0 1 0 1 1 1 0 0 1 0 0 0 1 1
 0 1 1 1 0 0 0 0 0 0 0 0 0 1 0 1 0 0 1 1 0 1 0 0 0 0 1 1 0 1 1 1 0 1 1 1 0
 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 1 0 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 1 0 1 0 1
 0 0 0 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1 0 1 0 0 0 0 0 0 1 1 0 1 0 1 1 0 0 1 0 1
 1 1 1 0 0 0 1 0 1 1 1 1 0 0 1 1 0 1 1 0 1 0 1 1 1 0 1 1 0 0 1 0 0 0 1 1 0
 1 0 0 1 1 1 1 0 0 1 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 0 1 0 1 1 0 1 1 1 0 0
 1 1 1 0 0 0 1 0 1 0 0 1 0 0 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 1 0 1 1 0 0 1 1 0
 0 0 0 1 0 1 1 1]
DEBUG::El testing accuracy score de Gradient Boosting es::
0.975
```

- Precisión (Accuracy): 0.975
- Tiempo de entrenamiento: 262 segundos
- Observaciones: El Gradient Boosting fue uno de los modelos más precisos con un accuracy score de 97.5%, pero también fue el más lento, tardando 262 segundos en entrenarse. Esto sugiere que es un modelo robusto, pero podría no ser ideal para situaciones en las que el tiempo de entrenamiento sea crítico.

Curva ROC:



- El gráfico de la curva ROC mostró el desempeño del modelo en términos de tasa de verdaderos positivos frente a la tasa de falsos positivos. Este gráfico es útil para analizar el comportamiento de un modelo más allá de la precisión, mostrando cómo varía el rendimiento con diferentes umbrales de decisión.

Comparación de Modelos

Modelo	Accuracy	Tiempo de Entrenamiento (s)
Regresión Logística	0.9866	N/A
Support Vector Classifier	0.8016	91
Random Forest Classifier	0.9783	3
GridSearchCV (RF)	0.9816	N/A
Gradient Boosting	0.975	262

Tabla 1.1 Tabla comparativa obtenida con las métricas y tiempos del Notebook en cuanto a la evaluación de los modelos