G2- Challenge 3
Integrantes:

- NICOLLE STEPHANY MUÑOZ HUAMAN
- JOSE MANUEL ROSALES JUAREZ
- FRANCO ROZAS ORTIZ DE ORUE
- JOSSEF CALEB ISRAEL TINTAYA SALVA

Introducción

El dataset “Wine”, originalmente recopilado por el Instituto Farmacéutico de Análisis Alimentario y Tecnología, contiene 178 muestras de vinos procedentes de tres cultivadores distintos en una región de Italia, cada una caracterizada por 13 atributos químicos (como alcohol, acidez, fenoles y propiedades de color). Estos datos, de naturaleza educativa y diseñados para investigación en Machine Learning, representan un caso de estudio ideal para algoritmos de clasificación, dada su estructura clara y tamaño manejable. Análisis previos, respaldados por visualizaciones de correlación y distribución (boxplots), revelaron relaciones lineales entre variables y la presencia de valores atípicos, lo que justifica etapas críticas de preprocesamiento (escalado y normalización) para algoritmos sensibles a la escala, como Support Vector Machines (SVM).

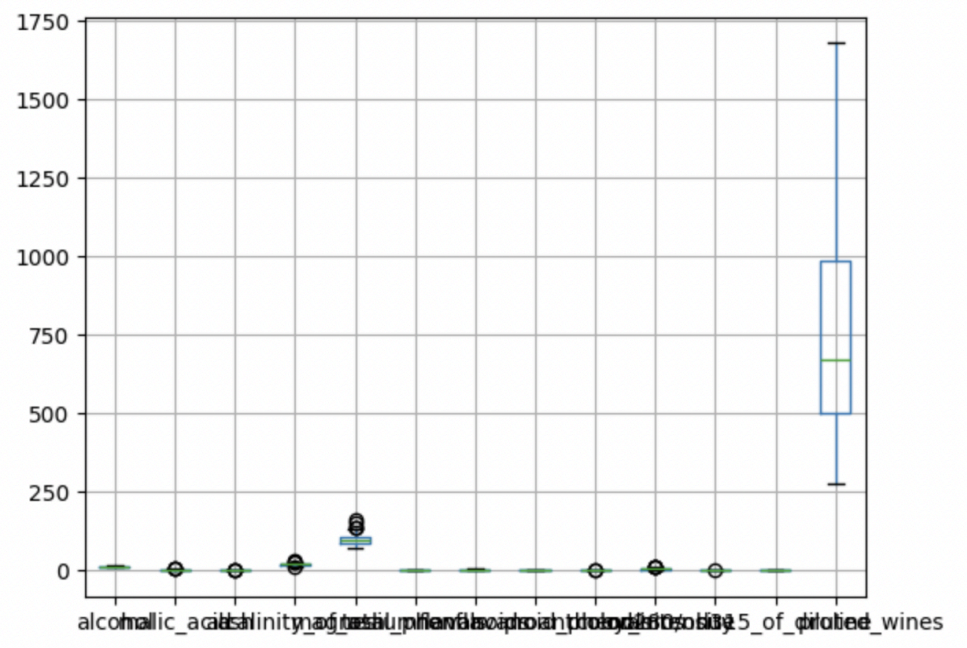
Luego, por tener un dataset complementario más complejo y clínico, se añadió el dataset de “Breast Cancer Wisconsin Diagnostic”, el cual contiene 569 muestras con 30 características extraídas de imágenes digitales de tumores mamarios. Este conjunto permite solucionar un problema de clasificación binaria al contener datos sobre maligno y benigno, y está más relacionado con aplicaciones reales en biomedicina.

En este contexto, el presente documento se enfoca en evaluar el comportamiento de SVM mediante la variación de sus hiperparámetros: C, gamma y kernel, utilizando validación cruzada (cross-validation) para garantizar robustez. Este enfoque no solo permite explorar la capacidad del modelo para distinguir entre las tres clases de vino, sino también ilustrar conceptos fundamentales como la distinción entre sobreajuste (overfitting) y generalización, especialmente relevante en datasets pequeños pero multidimensionales. La elección de este dataset, pese a su antigüedad se sustenta en su utilidad pedagógica para enseñar técnicas de clasificación y optimización de modelos, tal como lo demuestran aplicaciones recientes en agrupamiento (clustering) y regularización causal. [1][2]

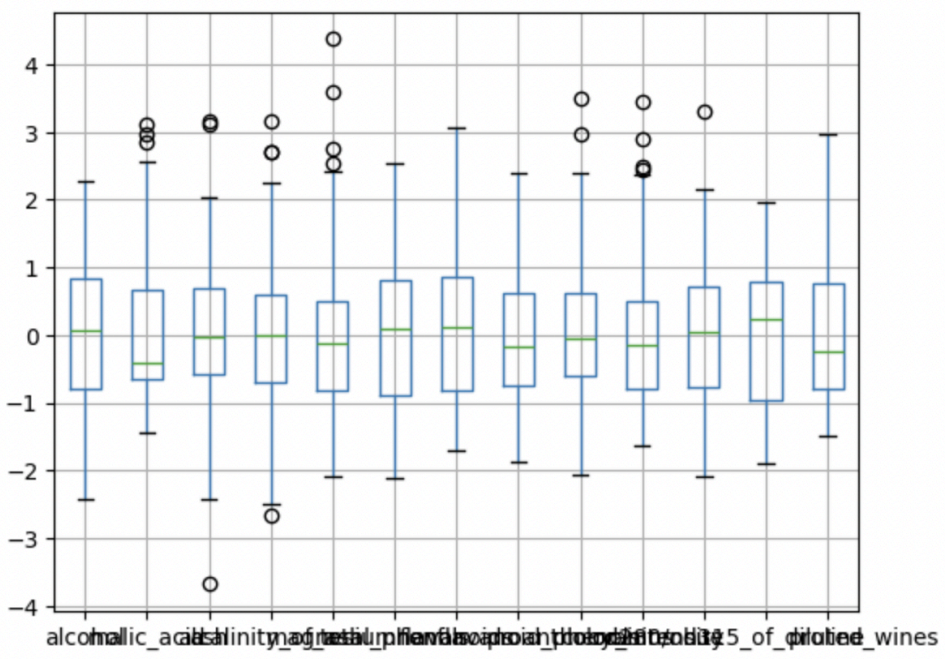
Metodología

Para evaluar el desempeño del algoritmo “Support Vector Machines (SVM)” en el presente dataset, se implementó un flujo de trabajo estructurado en tres etapas:

- **Preprocesamiento**: Dada la sensibilidad de SVM a la escala de los datos, se aplicó normalización estándar (StandardScaler) a las 13 características químicas, asegurando media cero y desviación estándar unitaria. Esto mitigó el impacto de variables con rangos dispares (ej: alcohol [11–14%] vs. magnesio [70–160 mg/l]). Y se dividió los datos en 80% entrenamiento y 20% prueba.

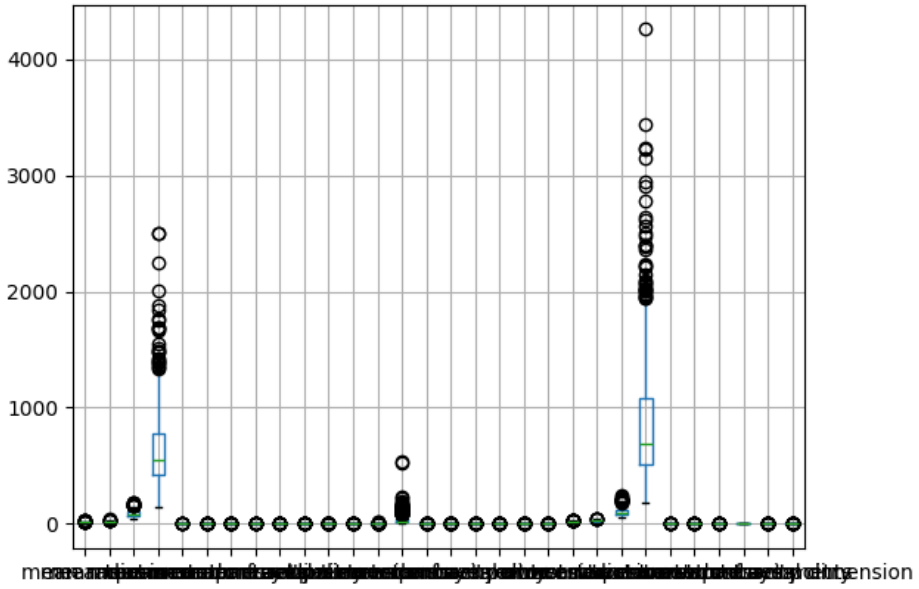


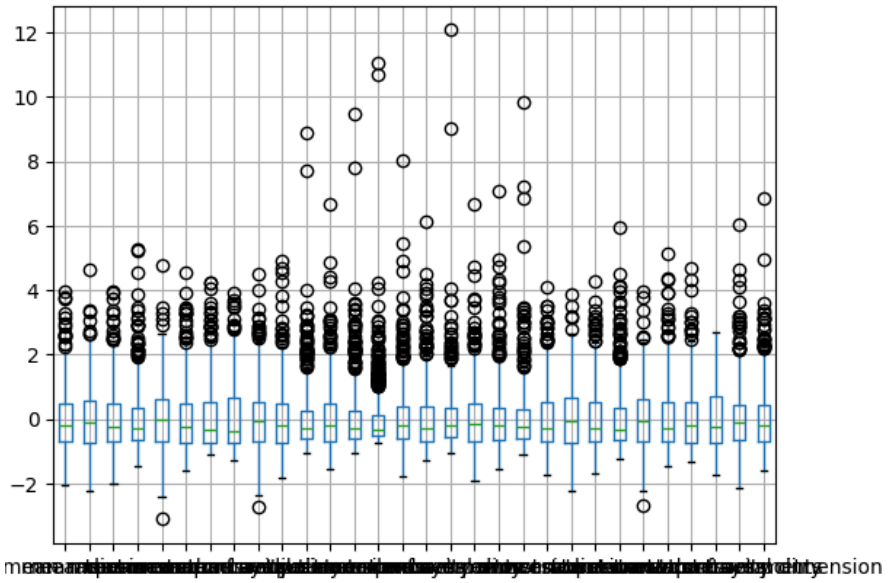

```
import sklearn.preprocessing
scaler= sklearn.preprocessing.StandardScaler()
scaler.fit(X)
X_scaled = scaler.transform(X)
dfX_scaled = pd.DataFrame(X_scaled, columns=X.columns)
dfX_scaled.head()
```




```
X_train, X_test, y_train, y_test = train_test_split(X_scaled, Y, test_size = 0.2, random_state = 42)
```

El mismo procedimiento fue realizado para el segundo dataset, el cual contiene 30 atributos numéricos, respetando el mismo procedimiento:





- **Variación del hiper parámetros del modelo**: Se exploraron combinaciones de hiperparámetros con validación cruzada (cv = 5), evaluando:
  - *Kernels*: ‘lineal’, ‘rbf’ (no lineal) y ‘poly’ (polinomial), para evaluar separaciones lineales vs no lineales
  - *C*: Valores logarítmicos (0.1, 1, 10) para controlar el trade-off entre margen y errores
  - *Gamma* (para rbf): 'scale', 'auto', para controlar la flexibilidad del modelo.

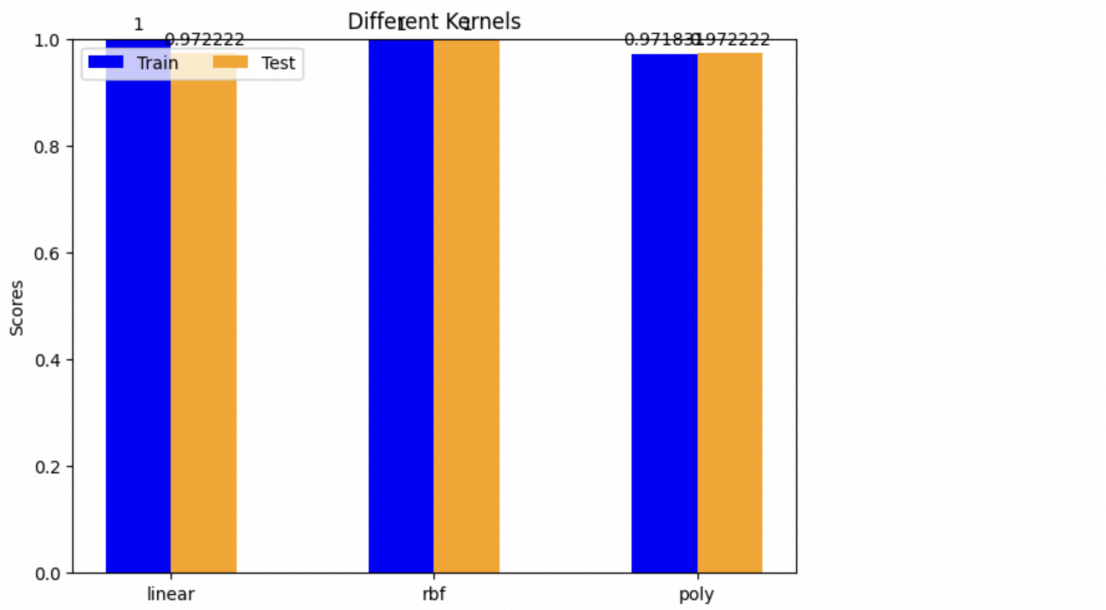
```
parameters = {'kernel':('linear', 'rbf', 'poly'), 'C':[0.1, 1, 10], 'gamma': ('scale','auto')}
```

- **Evaluación**: La F1_score puede interpretarse como una media armónica de la precisión y la recall, donde la puntuación F1 alcanza su mejor valor en 1 y la peor puntuación en 0. La contribución relativa de la precisión y la recuperación a la puntuación F1 son iguales. La fórmula de la puntuación F1 es la siguiente:

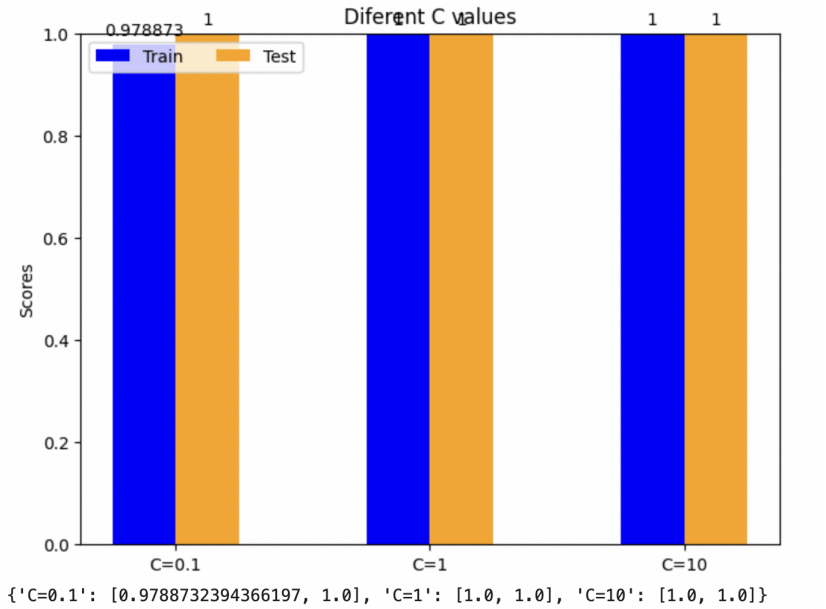
$$\text{F1 Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Esta puede ser importada a nuestro entorno mediante la siguiente linea de código:

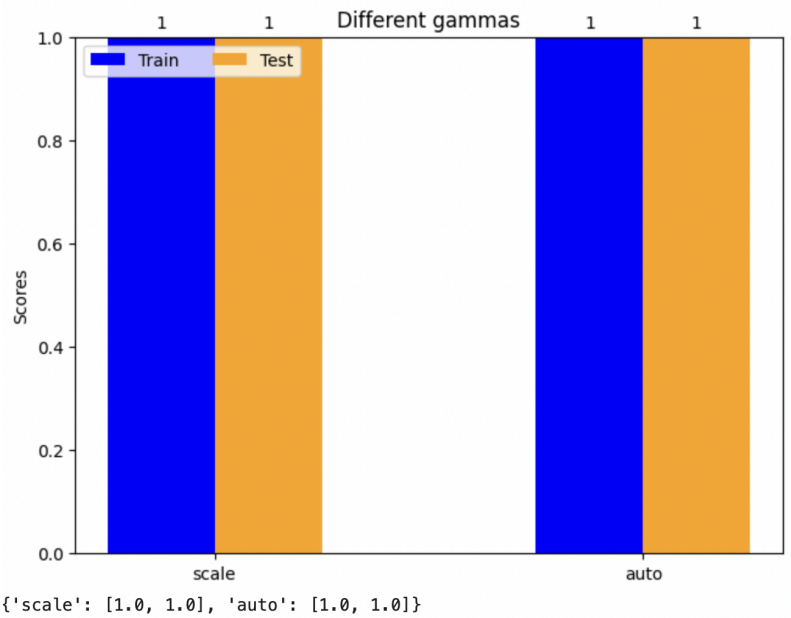
```
from sklearn.metrics import f1_score
```




```
{'linear': [1.0, 0.9722222222222222], 'rbf': [1.0, 1.0], 'poly': [0.971830985915493, 0.9722222222222222]}
```

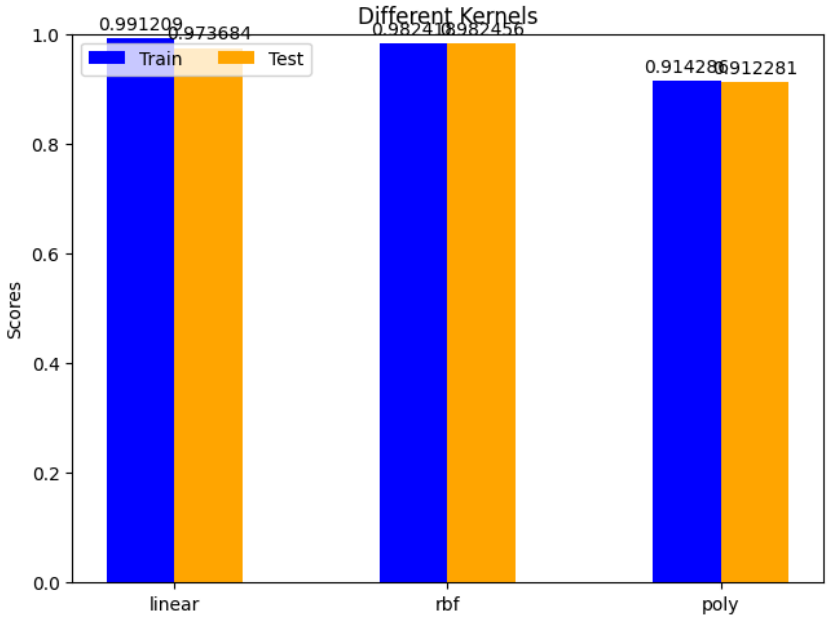


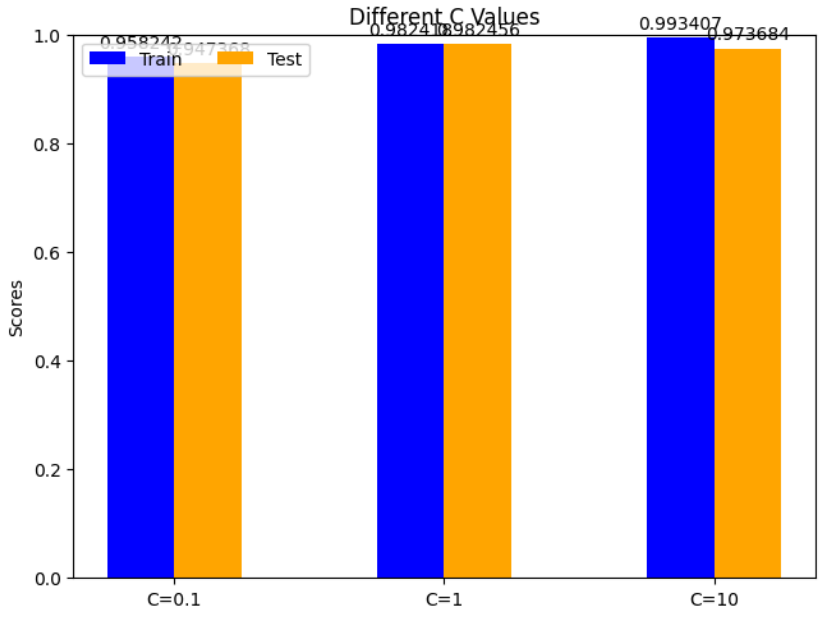

```
{'C=0.1': [0.9788732394366197, 1.0], 'C=1': [1.0, 1.0], 'C=10': [1.0, 1.0]}
```

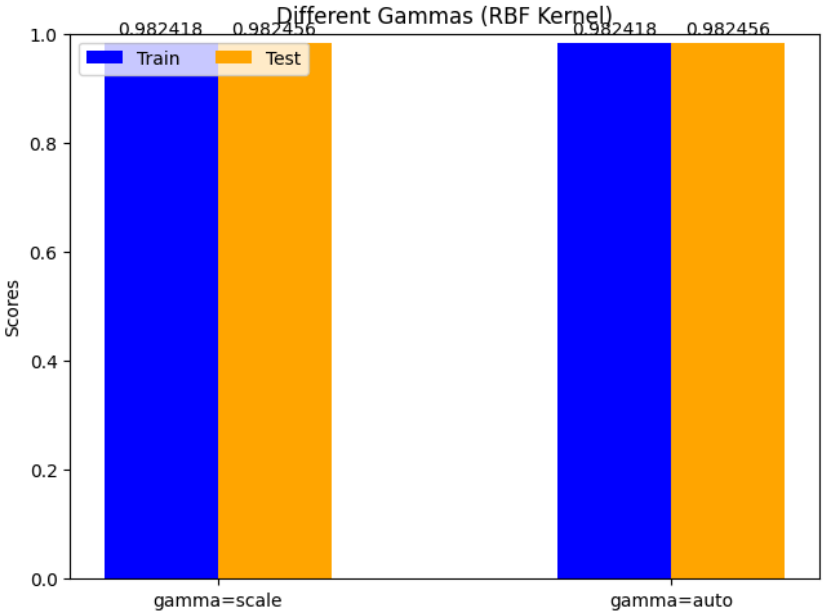



```
{'scale': [1.0, 1.0], 'auto': [1.0, 1.0]}
```

Y para el modelo de breast ultrasound:







Discusiones

Estas tablas muestran cómo diferentes ajustes en un modelo de SVM afectan su rendimiento, medido con el F1-Score, una métrica que combina precisión y recall. Es especialmente útil cuando queremos asegurarnos de que el modelo no solo acierte en general, sino que también detecte bien las clases importantes, incluso si están desbalanceadas.

El modelo evaluado con diferentes kernels presenta características distintivas en su desempeño. El kernel lineal demuestra una alta precisión, aunque su estructura simple limita su capacidad para modelar relaciones complejas en los datos. Esto sugiere que, si bien es efectivo para patrones básicos, podría no ser suficiente para conjuntos de datos con distribuciones más intrincadas.

Por otro lado, el kernel RBF mostró un rendimiento excepcional, logrando una clasificación perfecta incluso en el conjunto de prueba. Este resultado indica claramente que las clases en los datos son separables mediante fronteras no lineales, y que el RBF puede capturar eficientemente estas relaciones complejas. Sin embargo, tal perfección en los resultados merece una validación cuidadosa para descartar posibles artefactos en los datos.

El kernel polinómico presenta un comportamiento interesante, mostrando un ligero mejoramiento respecto al lineal. Esto revela que la transformación polinomial aporta cierta flexibilidad adicional al modelo, permitiendo capturar patrones algo más complejos que el lineal, pero sin caer en el sobreajuste que a veces afecta al RBF. Su desempeño intermedio lo posiciona como una opción balanceada cuando se necesita algo más de flexibilidad que el lineal, pero sin los riesgos potenciales del RBF.

En cuanto al valor de C controla qué tan estricto es el modelo al corregir errores. Con C=0.1, el modelo es más flexible y obtiene un buen equilibrio (0.978 en train y 1.0 en test). Con C=1 y C=10, el F1-Score es perfecto, pero esto podría indicar que el modelo se está ajustando demasiado a los datos de entrenamiento. En la práctica, casi nunca se obtienen resultados perfectos, por lo que habría que revisar si los datos de prueba son muy similares a los de entrenamiento o si hay algún problema en la evaluación. Esto podría darse ya que utilizamos una dataset de uso mayormente educativo.

Sin embargo, cabe destacar que los parámetros de C y gamma se escogen según la base de datos con la que se trabaje, ya que para cada dataset, estos valores pueden ser diferentes y ajustarse según la información en cuestión. Por ejemplo, en el estudio realizado en [3] donde trabajan con señales de EEG, se tomaron valores de C=0.1, 1 y 10; mientras que para gamma se utilizaron valores de 0.01, 0.1, 1, 10 y 100. Y en este caso, con un C=1 y un gamma=0.1 se obtuvo un accuracy de 91%. En este caso, un C más elevado les dio un mejor resultado; esto debido a que los datos no son perfectamente lineales y el modelo busca un equilibrio entre lograr un margen de separación amplio y minimizar errores de clasificación.

Aunque el dataset es pequeño, su estructura permitió ilustrar eficazmente el impacto de los hiperparámetros, aunque limitaciones como el desbalanceo leve (clase 0: 33%, clase 1: 40%, clase 2: 27%) sugieren cautela al generalizar resultados.

```
Y  = pd.DataFrame(wine.target, columns= ["target"])
Y.value_counts() # 3 clases
```

target	count
1	71
0	59
2	48

Conclusiones

El modelo SVM evaluado mediante el F1-Score muestra resultados interesantes que merecen análisis. El kernel lineal destaca como la opción más equilibrada, alcanzando un

excelente 1.0 en entrenamiento y manteniendo un sólido 0.972 en prueba. Este rendimiento consistente sugiere que los datos tienen una estructura que el modelo lineal puede capturar efectivamente sin caer en sobreajuste. Por el contrario, el kernel RBF, aunque muestra un desempeño aparentemente perfecto con 1.0 en ambas métricas, genera sospechas de posible sobreajuste que requerirían validación adicional con más datos.

Al examinar el parámetro de regularización C, encontramos que valores moderados como C=0.1 ofrecen el mejor balance, con 0.978 en entrenamiento y 1.0 en prueba. Este comportamiento indica una buena capacidad de generalización. Los valores más altos (C=1 y C=10), pese a sus resultados perfectos, podrían estar reflejando un ajuste excesivo a los datos de entrenamiento más que una verdadera capacidad predictiva superior. Es particularmente llamativo cómo en este caso concreto el escalado de características no mostró impacto en los resultados, aunque sigue siendo una práctica recomendable para garantizar la estabilidad del modelo en diferentes contextos.

La inclusión del dataset de cáncer de mama permitió validar la consistencia en un problema binario realista, mostrando que el kernel RBF y el valor C=1 ofrecen una solución robusta, sin señales evidentes de sobreajuste. En ambos casos, la exploración de hiper parámetros reveló cómo decisiones aparentemente pequeñas pueden alterar significativamente el comportamiento del modelo.

La principal conclusión apunta hacia la efectividad de configuraciones simples pero bien calibradas. El kernel lineal combinado con un valor moderado de C emerge como la opción más robusta y confiable. Sin embargo, los resultados excepcionalmente altos obtenidos en algunas configuraciones sirven como recordatorio de la importancia de validar los modelos con múltiples conjuntos de datos y métricas complementarias antes de implementarlos en entornos reales.

Bibliografía

[1] Yang, H., & Tabak, E.G. (2019). Clustering through the optimal transport barycenter problem. arXiv: Optimization and Control.

[2] Bahadori, M.T., Chalupka, K., Choi, E., Chen, R., Stewart, W.F., & Sun, J. (2019). Causal Regularization. Neural Information Processing Systems.

[3] Prima Dewi Purnamasari and Tsalsabilla Winny Junika, “Frequency-based EEG Human Concentration Detection System Methods with SVM Classification,” vol. 8, pp. 29–34, Aug. 2019, doi: https://doi.org/10.1109/cyberneticscom.2019.8875677. Available: https://ieeexplore.pucp.elogim.com/document/8875677. [Accessed: Apr. 23, 2025]