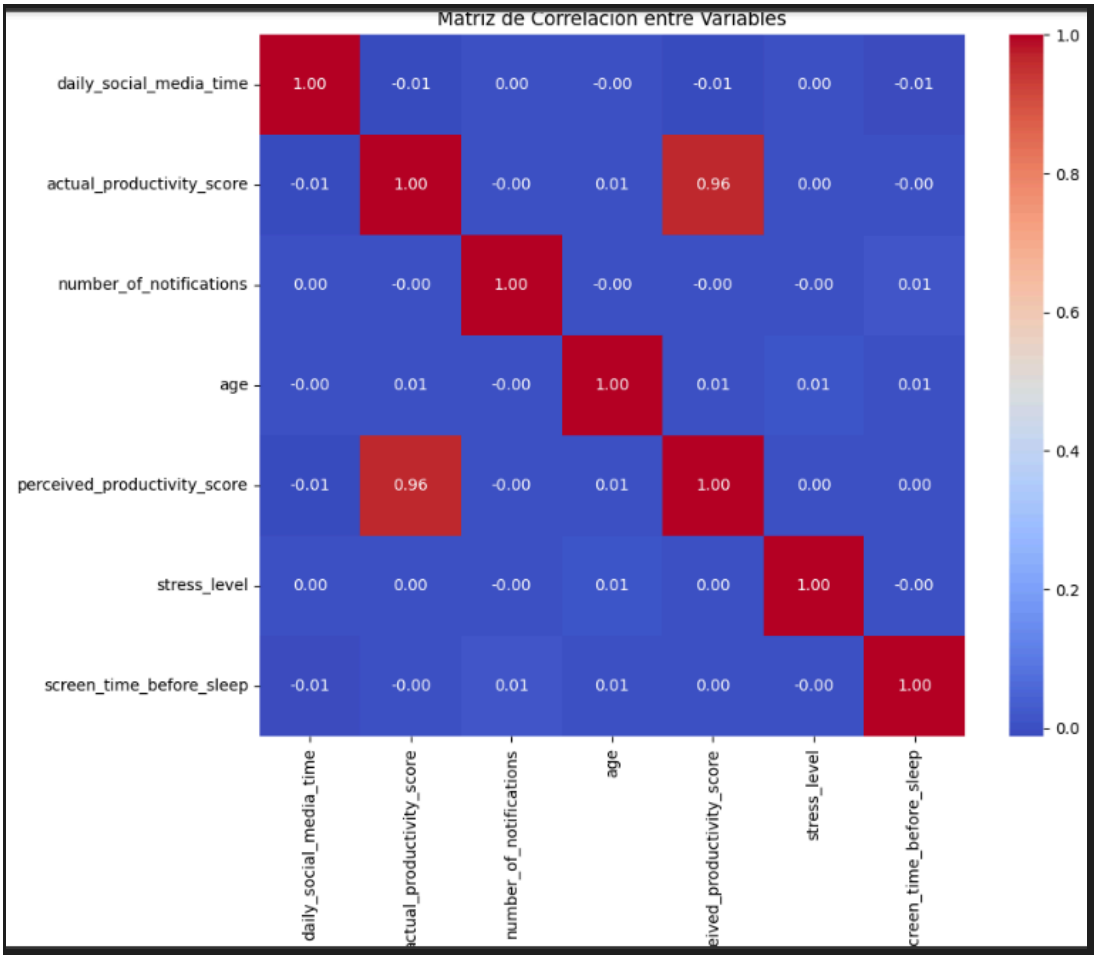
Informe:

Se escoge un set de datos que estudia las horas en las que las personas están conectadas a las redes sociales más importantes, su género, ocupación, estrés, nivel de productividad, etc.

```
age                                1493
gender                             1450
job_type                           1539
daily_social_media_time            4131
social_platform_preference         1504
number_of_notifications            1481
work_hours_per_day                 1521
perceived_productivity_score       3028
actual_productivity_score          3727
stress_level                       3362
sleep_hours                        3916
screen_time_before_sleep           3577
breaks_during_work                 1489
uses_focus_apps                    1535
has_digital_wellbeing_enabled      1536
coffee_consumption_per_day         1507
days_feeling_burnout_per_month     1528
weekly_offline_hours               1513
job_satisfaction_score             4072
dtype: int64
```

Se coloca data válida donde hay datos inválidos
Se aplica inyección de datos nulos al número faltante
Se hace un análisis previo de la información, con Medias, Mediana, Desviación estándar, counts,

Se agrupan los datos de daily_social_media_time en categorías, donde 3 es la más mala ya que son más de 5 horas al día, se crea la columna category_social_m_time.
Se calcula la matriz de correlación, donde se visualiza una alta correlación entre 'perceived_productivity_score' y 'actual_productivity_score'



Se generan gráficos de distribución con respecto a category_social_m_time creada, y histogramas

Se llenan los datos nulos con ffill para que los datos sean más coherente

#----------Escoger modelo----------

Se deja inputs que el modelo va a usar para aprender menos el que quiere predecir, en este caso actual_productivity_score

Se divide el dataset en conjunto de entrenamiento (70%) y prueba (30%).
Dónde 70 ayuda al modelo a aprender y 30 para testear
Se aplican los modelos de regresión, Linear, Ridge, Lasso, Decision Tree, Random Forest
Se excluye 'SVR' ya que lleva mucho tiempo de ejecución por lo que no se pudo ejecutar

Se calcula rmse y r2 que son los valores que nos ayudan a escoger el modelo:

	RMSE	*R^2*
Linear	*1.136219*	*0.631986*

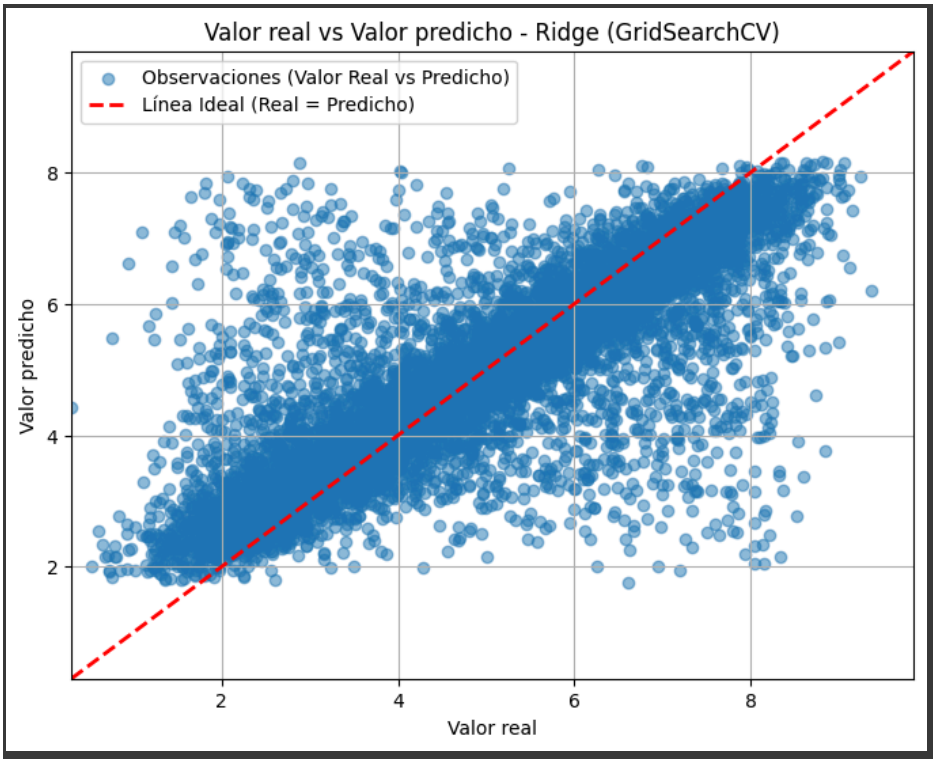
Ridge 1.136219 0.631986
Lasso 1.135520 0.632438
Decision Tree 1.219124 0.576322
Random Forest 1.140082 0.629480
Se escoge el modelo con un RMSE bajo y un R^2 alto: Ridge y Linear

Modelo 1----Ridge----------
#--Escoger modelo con ajuste de hyperparámetros con GridSearchCV y RandomizedSearchCV

Se define el rango de hiperparámetros para GridSearchCV
Para Ridge, el hiperparámetro principal es 'alpha'
Se van a tomar el alpha de 20 valores aleatorios entre 0.001 y 100

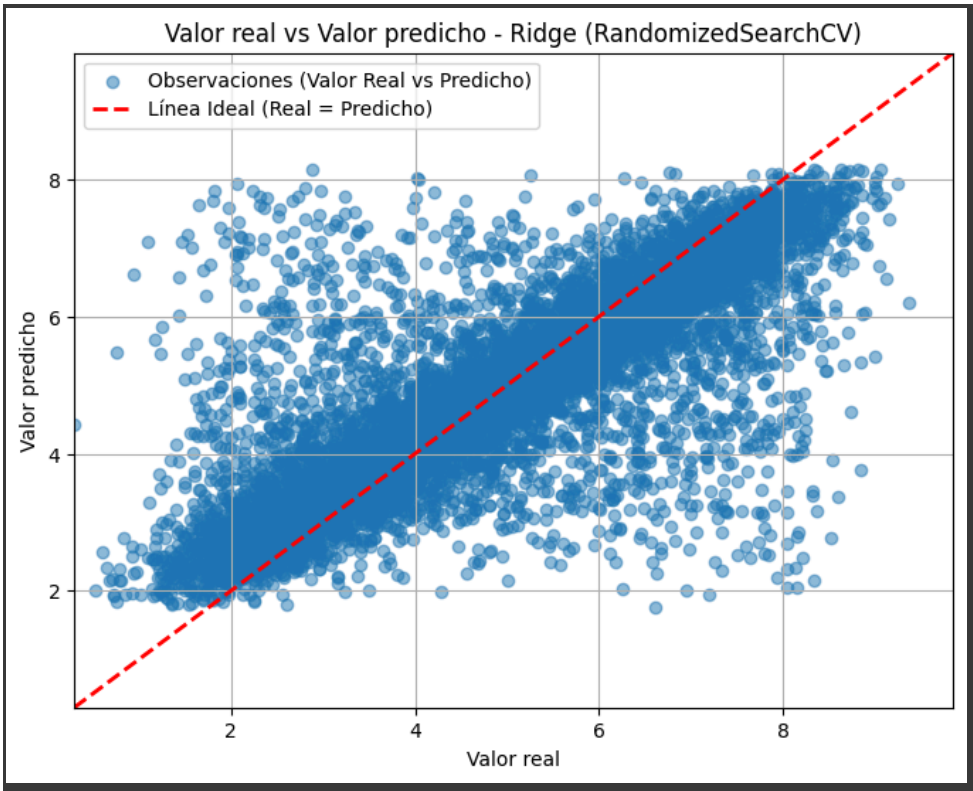
Se entrenar GridSearchCV en los datos de entrenamiento, esto realizará la validación cruzada para cada combinación de hiperparámetros

Se obtienen los mejores hiperparámetros encontrados, se grafica
la comparación para el mejor modelo Ridge (GridSearchCV)
Mejores hiperparámetros encontrados con GridSearchCV para Ridge: ridge__alpha: 10
Resultados del mejor modelo Ridge (GridSearchCV) en el conjunto de prueba:
RMSE: 1.1362
R^2: 0.6320
Se grafica la comparación para el mejor modelo Ridge (GridSearchCV)



Se aplica RandomizedSearchCV para Ridge
Se probará 20 combinaciones aleatorias con validación cruzada de 5 pliegues
Se entrena RandomizedSearchCV en los datos de entrenamiento
y obtener los mejores hiperparámetros encontrados

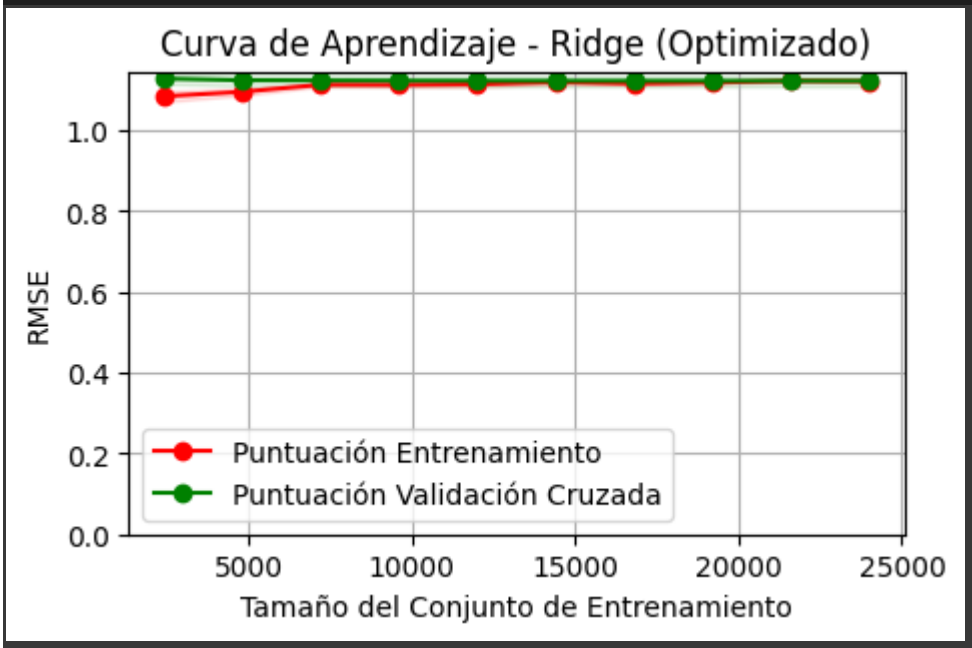
Mejores hiperparámetros encontrados con RandomizedSearchCV para Ridge:
{'ridge__alpha': np.float64(26.537209598828213)}
Resultados del mejor modelo Ridge (RandomizedSearchCV) en el conjunto de prueba:
RMSE: 1.1362
R^2: 0.6320



#----------Curvas de Aprendizaje Ridge----------

Se calcula los puntos para la curva de aprendizaje, se grafica

Se grafica la curva de entrenamiento con banda de error



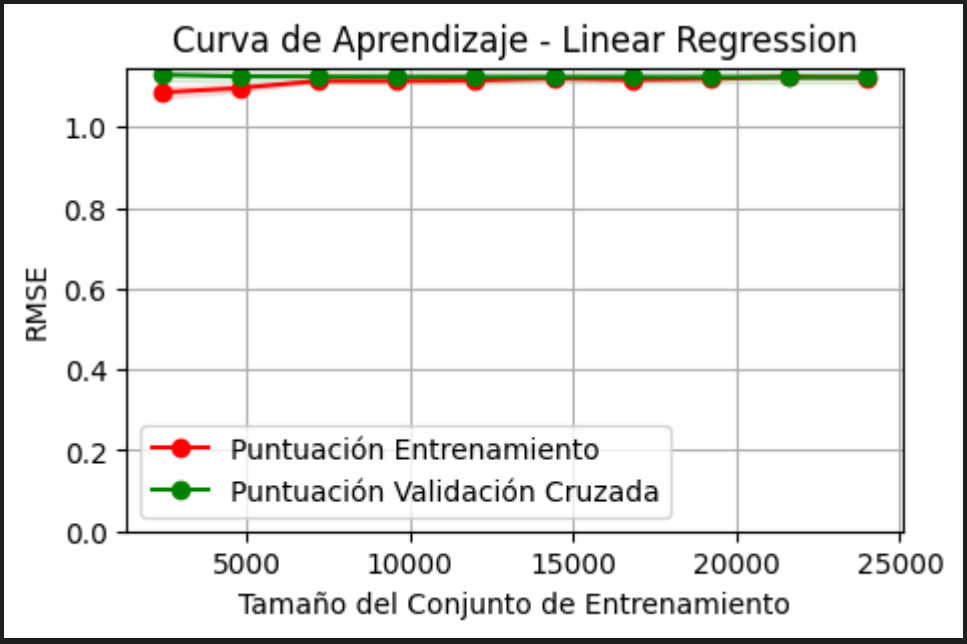
Curva de aprendizaje implícita

El flujo sugiere una curva estable para Ridge:
Buen rendimiento en entrenamiento y prueba

Baja varianza entre distintos modelos
Conclusión: El modelo no está ni subajustado ni sobreajustado (overfitting ni underfitting) y se encuentra en una zona óptima de la curva de aprendizaje. Esto indica un aprendizaje efectivo, donde agregar más datos no necesariamente mejorará mucho el modelo sin cambiar su estructura.

Modelo 2--------Linear---------
No se entrena directamente sin GridSearchCV ni RandomizedSearchCV ya que no aplica
Se calcula RMSE y R^2
RMSE (Linear): 1.136218923931588
R^2 (Linear): 0.6319861109098237

#----------Curvas de Aprendizaje Lineal----------
Se genera curva de aprendizaje para y se grafica



#------Compara el desempeño de los dos modelos----------
Se hace predicciones de los dos modelos definitivos
Se calcula métricas de desempeño
Correlación Pearson entre predicciones

Coeficiente de correlación intraclase (ICC)

Predicciones como evaluadores

Ridge - RMSE: 1.1362, R²: 0.6320, MAE: 0.7370
Linear - RMSE: 1.1362, R²: 0.6320, MAE: 0.7368
Correlación Pearson entre predicciones: 1.0000 (p=0.0000)

	Type	ICC	CI95%
0	ICC1	1.0	[1.0, 1.0]
1	ICC2	1.0	[1.0, 1.0]
2	ICC3	1.0	[1.0, 1.0]
3	ICC1k	1.0	[1.0, 1.0]
4	ICC2k	1.0	[1.0, 1.0]
5	ICC3k	1.0	[1.0, 1.0]

RMSE (Raíz del Error Cuadrático Medio):
Ambos modelos tienen el mismo nivel de error.
R² (Coeficiente de determinación) razonable.
MAE (Error Absoluto Medio): Ambos son casi idénticos, con una mínima diferencia no significativa.
Valor: 1.0000
p-valor: 0.0000
Las predicciones de ambos modelos predicen lo mismo
ICC (Coeficiente de Correlación Intraclase): están en 1.0,
Perfecta consistencia y acuerdo entre las predicciones de Ridge y Regresión Lineal.
CI95% (Intervalo de confianza): [1.0, 1.0] → Confianza en esa igualdad.
Ridge no mejora el desempeño frente a la regresión lineal estándar.
Probablemente: no hay multicolinealidad significativa en las variables predictoras