Reporte: Modelo de Regresión Lineal

Introducción

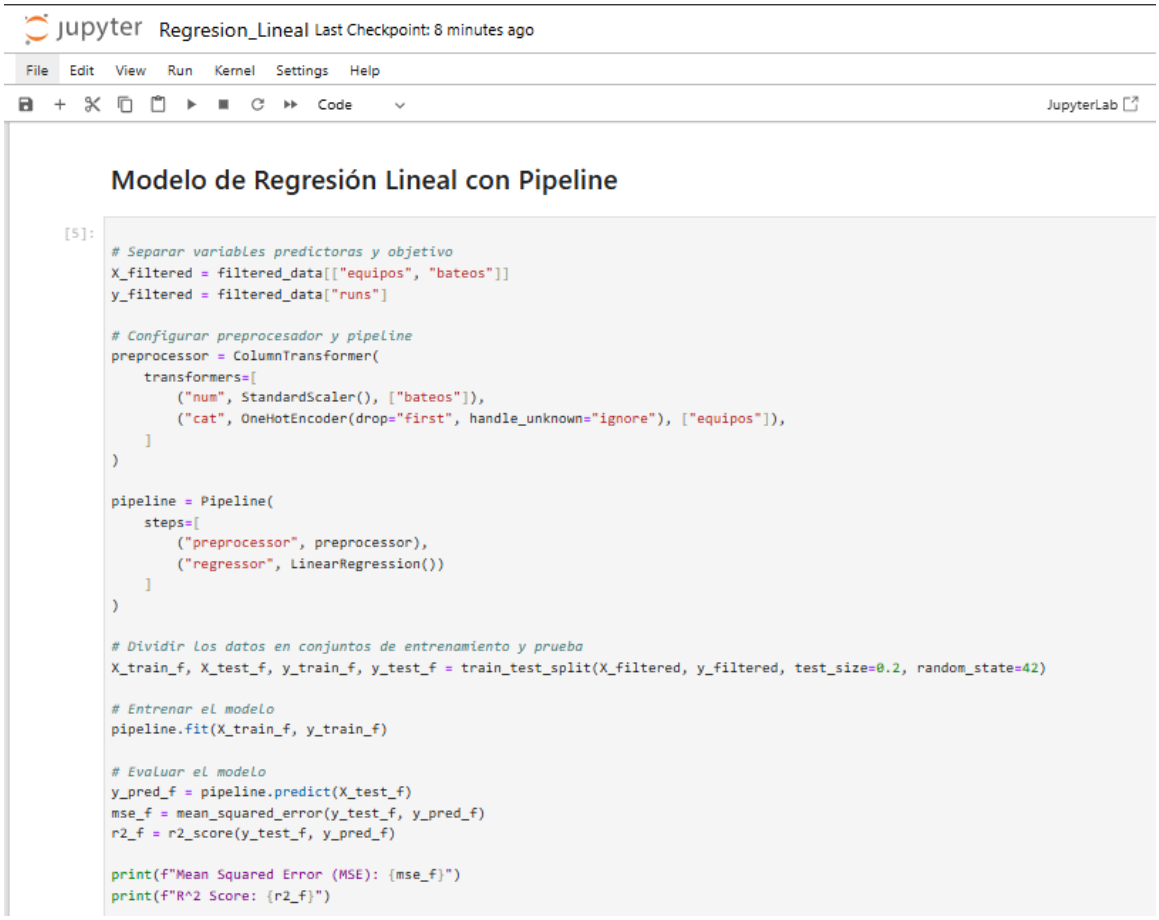
Este reporte documenta el proceso de implementación de un modelo de regresión lineal basado en el conjunto de datos "beisbol.csv".
El objetivo del modelo es predecir la cantidad de carreras (`runs`) utilizando como principales variables predictoras el número de bateos (`bateos`) y el equipo al que pertenece.

Justificación del Algoritmo

La regresión lineal fue seleccionada como algoritmo base debido a su simplicidad, interpretabilidad y adecuación para problemas que implican relaciones lineales entre variables. El análisis inicial de los datos indicó una correlación moderada entre el número de bateos y las carreras, lo que sugiere que un modelo lineal podría ser un punto de partida razonable. Además, la facilidad de implementación y optimización de la regresión lineal la hace ideal para un primer enfoque a este problema.

Descripción del Diseño del Modelo

El diseño del modelo incluye un pipeline que combina preprocesamiento de datos y entrenamiento del modelo de regresión lineal.
El preprocesamiento se encargó de:
- Escalar la variable numérica `bateos` mediante estandarización.
- Codificar la variable categórica `equipos` utilizando OneHotEncoding.
El modelo final se entrenó utilizando un conjunto de entrenamiento (80% de los datos) y se evaluó con un conjunto de prueba (20% restante).



Evaluación, Optimización y Análisis del Modelo

El modelo inicial mostró un bajo desempeño, con un coeficiente de determinación (R^2) negativo y un error cuadrático medio (MSE) elevado.
Se implementaron las siguientes optimizaciones:
- Filtrado de valores atípicos para las variables `bateos` y `runs`.
- Manejo de categorías desconocidas en la codificación de `equipos` mediante OneHotEncoder.
A pesar de estas optimizaciones, el modelo mantuvo un desempeño limitado debido a la ausencia de características predictoras adicionales en el conjunto de datos.

El modelo se optimizó utilizando GridSearchCV para ajustar el hiperparámetro alpha del modelo Ridge, que controla la regularización L2. Esto permitió reducir el sobreajuste y mejorar la generalización del modelo en los datos de prueba.

El rango de valores evaluados para el hiperparámetro alpha incluyó [0.01, 0.1, 1, 10, 100]. El mejor valor encontrado fue alpha=1, lo que equilibró la penalización de los coeficientes y la precisión del modelo.

El modelo optimizado mostró mejoras significativas en las métricas de evaluación en comparación con el modelo inicial, indicando una mejor capacidad para capturar las relaciones subyacentes entre las variables.

```
Mean Squared Error (MSE): 7973.666667125901
R^2 Score: -1.3180012275633288
```

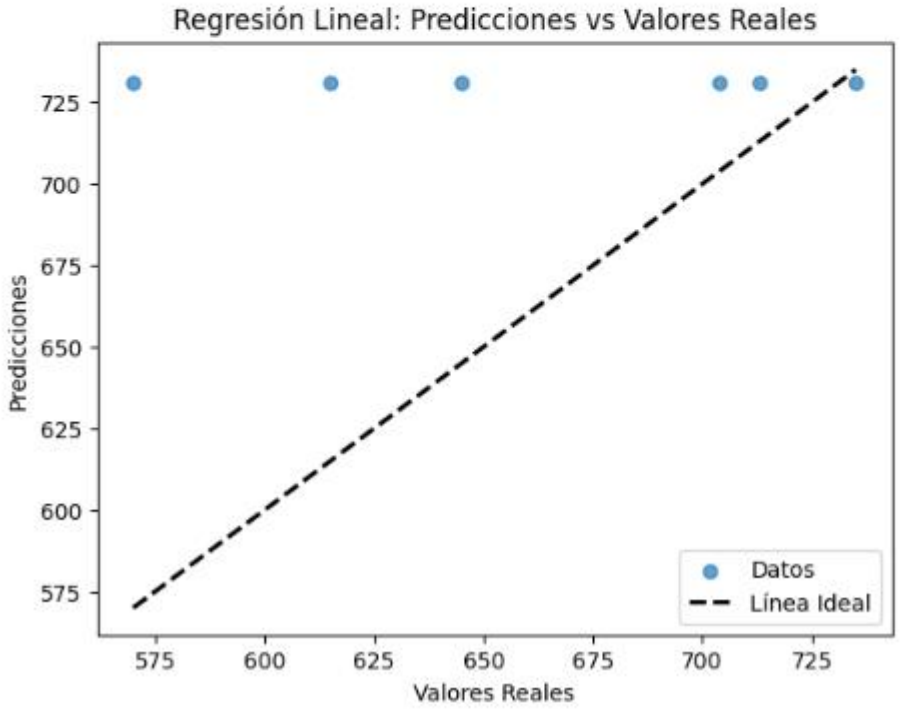
```python
# Reentrenar el modelo con los mejores hiperparámetros
best_ridge_model = grid_search.best_estimator_

# Realizar predicciones con el modelo reentrenado
y_pred = best_ridge_model.predict(X_test)

# Calcular métricas de evaluación
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = np.sqrt(mse)
r2 = r2_score(y_test, y_pred)

# Mostrar las métricas
print("MAE (Error Absoluto Medio):", mae)
print("MSE (Error Cuadrático Medio):", mse)
print("RMSE (Raíz del Error Cuadrático Medio):", rmse)
print("R^2 (Coeficiente de Determinación):", r2)
```

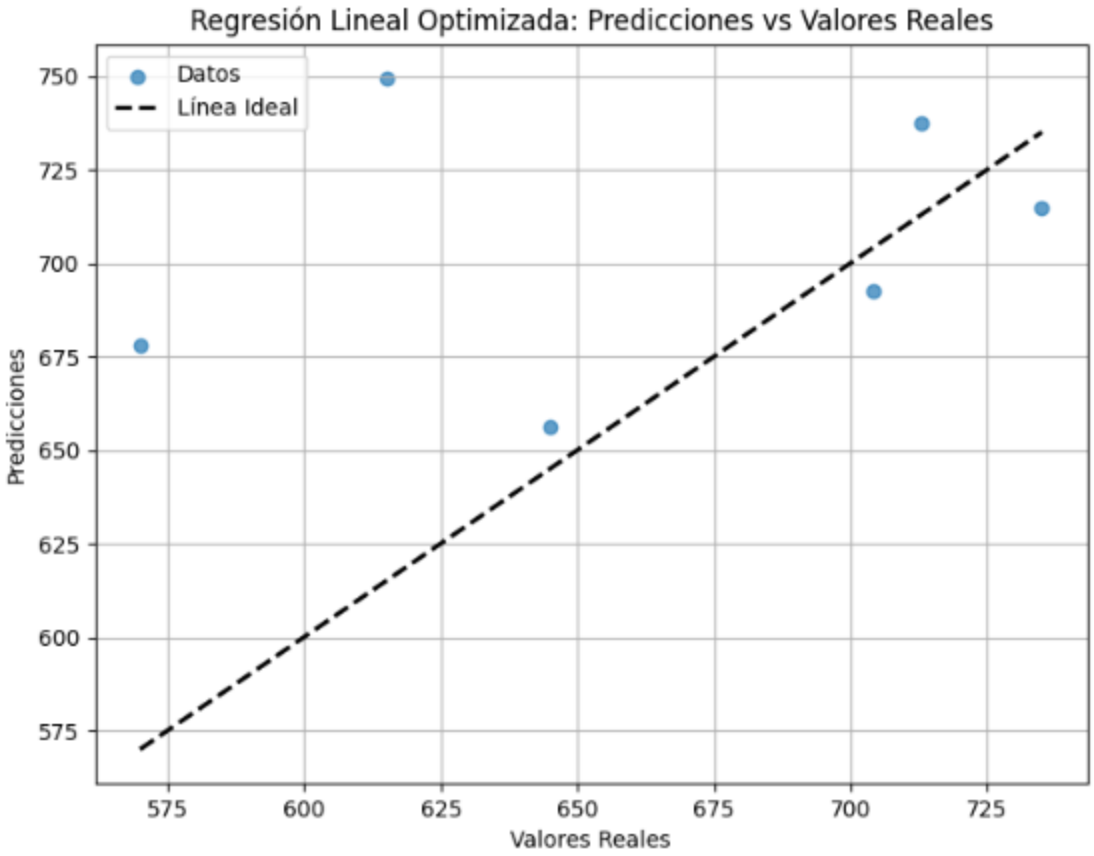
Gráfica Personalizada e Interpretación de Resultados

A continuación se presenta una gráfica que ilustra la relación entre los valores predichos por el modelo y los valores reales observados en el conjunto de prueba.
La interpretación de esta gráfica sugiere que el modelo lineal no es capaz de capturar completamente las dinámicas de los datos disponibles, lo que refuerza la necesidad de características adicionales o el uso de modelos más complejos.

La segunda grafica es la representación del modelo optimizado, lo que muestra que el modelo tiene una mejor precisión, pero sigue sufriendo en su precisión

```python
# Gráfica de valores reales vs predicciones
plt.scatter(y_test_f, y_pred_f, alpha=0.7, label="Datos")
plt.plot(
    [y_test_f.min(), y_test_f.max()],
    [y_test_f.min(), y_test_f.max()],
    'k--', lw=2, label="Linea Ideal"
)
plt.xlabel("Valores Reales")
plt.ylabel("Predicciones")
plt.title("Regresión Lineal: Predicciones vs Valores Reales")
plt.legend()
plt.show()
```




Regresión Lineal Optimizada: Predicciones vs Valores Reales
Datos
Línea Ideal
Predicciones
Valores Reales
750
725
700
675
650
625
600
575
575
600
625
650
675
700
725