Carolina Figeroa

Julian David Quijano

Sebastian Perez

Nicolas Sarmiento

Diego Peñuela

Teniendo en cuenta el resumen dado en el código para los 3 modelos entrenados:

|  |  |  |  |
| --- | --- | --- | --- |
|  | LINEAL | KNN | MLP |
| TIEMPO | 0.0045 | 0.0145 | 0.4360 |
| PRECISION | 0.7727 | 0.7402 | 0.7792 |

Para el caso del modelo de KNN se eligió un hiperparámetro de 16 vecinos ya que fue el que mejores resultados obtuvo con respecto a la precisión (se realizó una iteración utilizando de 1 a 20 vecinos)

Para el caso del modelo de MLP se eligieron los hiperparámetros de hidden\_layer\_sizes=(10,), activation='logistic' ya que fueron los mejores al realizar una iteración de los hiperparametros en colaboración con chatgtp xd

El código usado para esta iteración fue el siguiente

# Definir el espacio de hiperparámetros para probar

param\_grid = {

'hidden\_layer\_sizes': [(5,), (10,), (5, 5), (10, 10), (5, 6, 7)],

'activation': ['relu', 'tanh', 'logistic'],

'solver': ['adam', 'sgd'],

'alpha': [0.0001, 0.001, 0.01],

'learning\_rate': ['constant', 'adaptive'],

'max\_iter': [500, 1000, 1500]

}

# Crear instancia de la técnica

mlp = MLPClassifier(random\_state=42)

# Configurar GridSearchCV

grid\_search = GridSearchCV(mlp, param\_grid, cv=5, n\_jobs=-1, verbose=2)

# Entrenar el modelo con búsqueda de hiperparámetros

grid\_search.fit(X\_train, y\_train)

# Mejor modelo encontrado

best\_mlp = grid\_search.best\_estimator\_