



Tecnológico de Monterrey

Campus:
Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos (Gpo 102)

Curso:
TC3006C.102

Módulo 2: Portafolio 2

Daniel Sánchez Villarreal

A01197699

Lugar y Fecha:
Monterrey, Nuevo León
7 de septiembre de 2024

En este Portafolio 2, decidí usar el dataset de breast_cancer y entrenar 6 modelos, los cuales fueron regresión logística, árboles de decisión, random forest, KNN, SVC, y Gaussian Naive Bayes. Esto con el fin de evaluar cuál de todos tiene el mejor desempeño y ese sería el que escogería como mi mejor modelo para después usar GridSearch para buscar e implementarle los mejores hiperparámetros y evaluar su rendimiento para finalmente hacer algunas predicciones con ese modelo mejorado.

Para esto, lo primero que hice fue la preparación de los datos. En este caso, definí la Y como la variable target, y la X para el resto de features en el dataset. Usé target_names para ver la clase de datos para la variable target, los cuales con este comando pude observar que cada instancia (sample) tenía un valor ya sea de benigno o maligno en esa variable target. Entonces lo que hice a continuación fue transformar esos datos para establecer benigno como 0 y maligno como 1 en todas las instancias de dicha variable target en el dataset.

Posteriormente, verifiqué si había o no datos nulos en el dataset y tras ver que no había ningún dato nulo en ninguna de las features, procedí con el entrenamiento de mis 6 modelos para después evaluarlos todos para poder saber cuál de todos tiene el mejor rendimiento y ese será el que voy a elegir para usar GridSearch para implementarle los mejores hiperparámetros y evaluar su rendimiento para después hacer predicciones con dicho modelo.

Comencé con el modelo de regresión logística, el cual tras evaluar sus métricas obtuve los siguientes resultados:

Portafolio2_Daniel_Sanchez.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Evaluación del modelo de regresión logística

```
[13] # Métricas
from sklearn.metrics import precision_score, recall_score, f1_score, r2_score, mean_squared_error as mse
acc_log = accuracy_score(y_test, y_pred)
print('Exactitud:', accuracy_score(y_test, y_pred))
pre_log = precision_score(y_test, y_pred)
print('Precisión:', precision_score(y_test, y_pred))
rec_log = recall_score(y_test, y_pred)
print('Recall:', recall_score(y_test, y_pred))
f1_log = f1_score(y_test, y_pred)
print('Score de F1:', f1_score(y_test, y_pred))
r2_log = r2_score(y_test, y_pred)
print('Score de r2:', r2_score(y_test, y_pred))
mse_log = mse(y_test, y_pred)
print('Error cuadrado medio (MSE):', mse(y_test, y_pred))
```

Exactitud: 0.9473684210526315
Precisión: 0.9692307692307692
Recall: 0.9402985074626866
Score de F1: 0.9545454545454546
Score de r2: 0.7827881867259447
Error cuadrado medio (MSE): 0.05263157894736842

```
# Matriz de confusión
mat = confusion_matrix(y_test, y_pred)
print(mat)
```

[[45 2]
 [4 63]]

Después, en el de árboles de decisión obtuve lo siguiente:

Evaluación del modelo de árboles de decisión

```
[17] # Métricas
acc_arb = accuracy_score(y_test, y_pred)
print('Exactitud:', acc_arb)
pre_arb = precision_score(y_test, y_pred)
print('Precisión:', pre_arb)
rec_arb = recall_score(y_test, y_pred)
print('Recall:', rec_arb)
f1_arb = f1_score(y_test, y_pred)
print('Score de F1:', f1_arb)
r2_arb = r2_score(y_test, y_pred)
print('Score de r2:', r2_arb)
mse_arb = mse(y_test, y_pred)
print('Error cuadrado medio (MSE):', mse_arb)
```

Exactitud: 0.9122807017543859
Precisión: 0.9523809523809523
Recall: 0.8955223880597015
Score de F1: 0.9230769230769231
Score de r2: 0.6379803112099078
Error cuadrado medio (MSE): 0.08771929824561403

```
# Matriz de confusión
mat_arb = confusion_matrix(y_test, y_pred)
print(mat_arb)
```

[[44 3]
 [7 60]]

