

Actividad de Aprendizaje 6-Análisis de datos(Modelos)

Mónica Fernanda Rojas Celis

Yeimy Vanessa Ricardo Ballestas

Daniel Gómez Pérez

Servicio Nacional de Aprendizaje

Tecnología en Análisis y desarrollo de software – Ficha 2828523

Medellín, 10 de abril de 2025

Actividad 2 – Modelos

1. Instalación de bibliotecas necesarias

`pip install pandas numpy matplotlib seaborn scikit-learn`

```
danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/Coquito-web-1
$ pip install pandas numpy matplotlib seaborn scikit-learn
Requirement already satisfied: pandas in c:\users\danie\appdata\local\programs\python\python39\lib\site-packages (2.2.3)
Requirement already satisfied: numpy in c:\users\danie\appdata\local\programs\python\python39\lib\site-packages (2.0.2)
Requirement already satisfied: matplotlib in c:\users\danie\appdata\local\programs\python\python39\lib\site-packages (3.9.4)
Requirement already satisfied: seaborn in c:\users\danie\appdata\local\programs\python\python39\lib\site-packages (0.13.2)
Collecting scikit-learn
  Downloading scikit_learn-1.6.1-cp39-cp39-win_amd64.whl (11.2 MB)
    11.2/11.2 MB 2.6 MB/s eta 0:00:00
```

2. Importación de librerías y carga de datos

`import pandas as pd`

`import numpy as np`

`import matplotlib.pyplot as plt`

`import seaborn as sns`

`from sklearn.model_selection import train_test_split`

`from sklearn.linear_model import LinearRegression`

`from sklearn.metrics import mean_squared_error, r2_score`

```
1  import pandas as pd
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import seaborn as sns
5  from sklearn.model_selection import train_test_split
6  from sklearn.linear_model import LinearRegression
7  from sklearn.metrics import mean_squared_error, r2_score
8
```

Cargar el conjunto de datos Iris desde seaborn

`data = sns.load_dataset('iris')`

Ver las primeras filas del conjunto de datos

`print(data.head())`

```
9  data = sns.load_dataset('iris')
10
11  print(data.head())
```

Resultado

```
danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/Coquito-web-1
$ python modelos.py
   sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2   setosa
1           4.9           3.0           1.4           0.2   setosa
2           4.7           3.2           1.3           0.2   setosa
3           4.6           3.1           1.5           0.2   setosa
4           5.0           3.6           1.4           0.2   setosa
```

3. Preparación de los datos

Seleccionamos las características (features) y la variable objetivo (target)

X = data[['sepal_length', 'sepal_width', 'petal_width']] # Variables predictoras

y = data['petal_length'] # Variable objetivo

Dividir el conjunto de datos en entrenamiento y prueba (80% entrenamiento, 20% prueba)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Tamaño del conjunto de entrenamiento:", X_train.shape)

print("Tamaño del conjunto de prueba:", X_test.shape)

```
13 # Seleccionamos las características (features) y la variable objetivo (target)
14 X = data[['sepal_length', 'sepal_width', 'petal_width']] # Variables predictoras
15 y = data['petal_length'] # Variable objetivo
16 # Dividir el conjunto de datos en entrenamiento y prueba (80% entrenamiento, 20% prueba)
17 X_train, X_test, y_train, y_test = train_test_split(X, y,
18 test_size=0.2, random_state=42)
19 print("Tamaño del conjunto de entrenamiento:", X_train.shape)
20 print("Tamaño del conjunto de prueba:", X_test.shape)
```

4. Creación del modelo de regresión lineal

Crear el modelo de regresión lineal

model = LinearRegression()

Entrenar el modelo

model.fit(X_train, y_train)

Ver los coeficientes y la intersección (intercepto) del modelo

print(f"Coeficientes: {model.coef_}")

print(f"Intersección (intercepto): {model.intercept_}")

```
22 # Crear el modelo de regresión lineal
23 model = LinearRegression()
24 # Entrenar el modelo
25 model.fit(X_train, y_train)
26 # Ver los coeficientes y la intersección (intercepto) del modelo
27 print(f"Coeficientes: {model.coef_}")
28 print(f"Intersección (intercepto): {model.intercept_}")
29
```

5. Hacer predicciones

Realizar predicciones sobre el conjunto de prueba

y_pred = model.predict(X_test)

Mostrar las predicciones y los valores reales

```

predictions_df = pd.DataFrame({'Real': y_test, 'Predicción':
y_pred})
print(predictions_df.head())

```

```

30 # Realizar predicciones sobre el conjunto de prueba
31 y_pred = model.predict(X_test)
32 # Mostrar las predicciones y los valores reales
33 predictions_df = pd.DataFrame({'Real': y_test, 'Predicción':
34 y_pred})
35 print(predictions_df.head())
36

```

Resultado

```

danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/Coquito-web-1
● $ python modelos.py
  sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
Tamaño del conjunto de entrenamiento: (120, 3)
Tamaño del conjunto de prueba: (30, 3)
Coeficientes: [ 0.72281463 -0.63581649  1.46752403]
Intersección (intercepto): -0.2621959025887044
   Real  Predicción
73   4.7    4.127716
18   1.7    1.882002
118  6.9    7.025659
78   4.5    4.432110
76   4.8    4.927191

```

6. Evaluación del modelo

Calcular el error cuadrático medio (MSE) y R^2

```
mse = mean_squared_error(y_test, y_pred)
```

```
r2 = r2_score(y_test, y_pred)
```

```
print(f"Error cuadrático medio (MSE): {mse}")
```

```
print(f"Coeficiente de determinación  $R^2$ : {r2}")
```

```

37 # Calcular el error cuadrático medio (MSE) y R²
38 mse = mean_squared_error(y_test, y_pred)
39 r2 = r2_score(y_test, y_pred)
40 print(f"Error cuadrático medio (MSE): {mse}")
41 print(f"Coeficiente de determinación R²: {r2}")
42

```

Resultado

```

danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/Coquito-web-1
$ python modelos.py
  sepal_length  sepal_width  petal_length  petal_width  species
0           5.1           3.5           1.4           0.2  setosa
1           4.9           3.0           1.4           0.2  setosa
2           4.7           3.2           1.3           0.2  setosa
3           4.6           3.1           1.5           0.2  setosa
4           5.0           3.6           1.4           0.2  setosa
Tamaño del conjunto de entrenamiento: (120, 3)
Tamaño del conjunto de prueba: (30, 3)
Coeficientes: [ 0.72281463 -0.63581649  1.46752403]
Intersección (intercepto): -0.2621959025887044
   Real  Predicción
73   4.7   4.127716
18   1.7   1.882002
118  6.9   7.025659
78   4.5   4.432110
76   4.8   4.927191
Error cuadrático medio (MSE): 0.13001626031382688
Coeficiente de determinación R²: 0.9603293155857664

```

7. Visualización de resultados

Graficar los valores reales vs. las predicciones

```

plt.figure(figsize=(8, 6))
plt.scatter(y_test, y_pred, color='blue', label='Predicciones')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
color='red', linewidth=2, label="Línea de referencia")
plt.xlabel("Valores reales")
plt.ylabel("Predicciones")
plt.title("Valores reales vs Predicciones")
plt.legend()
plt.show()

```

```

43 # Graficar los valores reales vs. las predicciones
44 plt.figure(figsize=(8, 6))
45 plt.scatter(y_test, y_pred, color='blue', label='Predicciones')
46 plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()],
47          color='red', linewidth=2, label="Línea de referencia")
48 plt.xlabel("Valores reales")
49 plt.ylabel("Predicciones")
50 plt.title("Valores reales vs Predicciones")
51 plt.legend()
52 plt.show()
53

```

Resultados

```

danie@LAPTOP-0MT3MCT1 MINGW64 ~/Downloads/Coquito-web-1
● $ python modelos.py
    sepal_length  sepal_width  petal_length  petal_width  species
0             5.1           3.5           1.4           0.2  setosa
1             4.9           3.0           1.4           0.2  setosa
2             4.7           3.2           1.3           0.2  setosa
3             4.6           3.1           1.5           0.2  setosa
4             5.0           3.6           1.4           0.2  setosa
Tamaño del conjunto de entrenamiento: (120, 3)
Tamaño del conjunto de prueba: (30, 3)
Coeficientes: [ 0.72281463 -0.63581649  1.46752403]
Intersección (intercepto): -0.2621959025887044
    Real  Predicción
73    4.7    4.127716
18    1.7    1.882002
118   6.9    7.025659
78    4.5    4.432110
76    4.8    4.927191
Error cuadrático medio (MSE): 0.13001626031382688
Coeficiente de determinación R²: 0.9603293155857664

```

