Simple Linear Regression Example

Problem: Predict salary based on years of experience.

```
# Import libraries
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

# Dataset (Years of Experience vs Salary in $1000s)
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9]).reshape(-1, 1)
y = np.array([35, 40, 50, 55, 65, 70, 80, 85, 95])

# Create model
model = LinearRegression()
model.fit(X, y)

# Prediction
y_pred = model.predict(X)

# Print slope and intercept
print("Intercept (b0):", model.intercept_)
print("Coefficient (b1):", model.coef_)

# Visualization
plt.scatter(X, y, color='blue', label="Actual Data")
plt.plot(X, y_pred, color='red', linewidth=2, label="Regression Line")
plt.xlabel("Years of Experience")
plt.ylabel("Salary (in $1000s)")
plt.legend()

plt.show()
```

2. Multiple Linear Regression Example

Problem: Predict student performance based on study hours and attendance.

```
import pandas as pd

from sklearn.linear_model import LinearRegression


# Dataset

data = {

    'Study_Hours': [2, 3, 4, 5, 6, 8, 9],

    'Attendance': [70, 75, 80, 85, 88, 92, 95],

    'Score': [50, 55, 65, 70, 75, 85, 90]

}

df = pd.DataFrame(data)


# Features and Target

X = df[['Study_Hours', 'Attendance']]

y = df['Score']


# Model training

model = LinearRegression()

model.fit(X, y)


# Predictions

pred = model.predict([[7, 90]])  # Example: 7 study hours & 90% attendance

print("Predicted Score:", pred[0])
```

3. Polynomial Regression Example

Problem: Predict growth trend (non-linear data).

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

# Dataset
X = np.array([1, 2, 3, 4, 5, 6]).reshape(-1, 1)
y = np.array([3, 6, 19, 45, 100, 180])  # Non-linear growth

# Create Polynomial Regression model (degree=2)
poly_model = make_pipeline(PolynomialFeatures(2), LinearRegression())
poly_model.fit(X, y)

# Prediction
y_pred = poly_model.predict(X)

# Visualization
plt.scatter(X, y, color='blue', label="Actual Data")
plt.plot(X, y_pred, color='green', label="Polynomial Fit (deg=2)")
plt.xlabel("X")
plt.ylabel("Y")
plt.legend()
plt.show()
```

4. Logistic Regression Example

Problem: Predict whether a student passes (1) or fails (0) based on study hours.

```
from sklearn.linear_model import LogisticRegression
```

```
# Dataset
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9]).reshape(-1, 1)
y = np.array([0, 0, 0, 0, 1, 1, 1, 1, 1])  # 0 = Fail, 1 = Pass

# Model training
log_reg = LogisticRegression()
log_reg.fit(X, y)

# Predictions
print("Prediction for 4 study hours:", log_reg.predict([[4]])[0])
print("Prediction for 7 study hours:", log_reg.predict([[7]])[0])

# Probability values
print("Pass probability at 4 hrs:", log_reg.predict_proba([[4]])[0][1])
print("Pass probability at 7 hrs:", log_reg.predict_proba([[7]])[0][1])

# Visualization
plt.scatter(X, y, color='blue', label="Actual Data")
plt.plot(X, log_reg.predict_proba(X)[:,1], color='red', label="Sigmoid Curve")
plt.xlabel("Study Hours")
plt.ylabel("Pass Probability")
plt.legend()
plt.show()
```