--

Logistic Regression from Scratch (NumPy)

--

```
import numpy as np from sklearn.datasets import load_breast_cancer from
sklearn.model_selection import train_test_split from sklearn.preprocessing import
StandardScaler from sklearn.metrics import accuracy_score, precision_score,
recall_score
```

----

1. Load Dataset

```
data = load_breast_cancer() X = data.data y = data.target
```

Train-test split

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

Scale features

```
scaler = StandardScaler() X_train = scaler.fit_transform(X_train) X_test =
scaler.transform(X_test)
```

----

2. Custom Logistic Regression Class

----

class CustomLogisticRegression:

```
def __init__(self, lr=0.01, epochs=1000):
    self.lr = lr
    self.epochs = epochs

def sigmoid(self, z):
    return 1 / (1 + np.exp(-z))

def fit(self, X, y):
    # Initialize weights
    self.m, self.n = X.shape
    self.W = np.zeros(self.n)
    self.b = 0

    for _ in range(self.epochs):
        # Predictions
        z = np.dot(X, self.W) + self.b
        A = self.sigmoid(z)

        # Gradients
        dw = (1 / self.m) * np.dot(X.T, (A - y))
        db = (1 / self.m) * np.sum(A - y)

        # Update weights
        self.W -= self.lr * dw
        self.b -= self.lr * db

def predict(self, X):
    z = np.dot(X, self.W) + self.b
    A = self.sigmoid(z)
    return (A >= 0.5).astype(int)
```

----

3. Train Custom Model

----

model = CustomLogisticRegression(lr=0.01, epochs=5000) model.fit(X_train, y_train)

y_pred_custom = model.predict(X_test)

accuracy_custom = accuracy_score(y_test, y_pred_custom) precision_custom = precision_score(y_test, y_pred_custom) recall_custom = recall_score(y_test, y_pred_custom)

----

4. Scikit-learn Logistic Regression

----

from sklearn.linear_model import LogisticRegression

sk_model = LogisticRegression(max_iter=5000) sk_model.fit(X_train, y_train)

y_pred_sk = sk_model.predict(X_test)

accuracy_sk = accuracy_score(y_test, y_pred_sk) precision_sk = precision_score(y_test, y_pred_sk) recall_sk = recall_score(y_test, y_pred_sk)

----

5. Print Results

```
print("\n---- Custom Logistic Regression ----") print("Accuracy:", accuracy_custom)
print("Precision:", precision_custom) print("Recall:", recall_custom)
```

```
print("\n---- Scikit-learn Logistic Regression ----") print("Accuracy:", accuracy_sk)
print("Precision:", precision_sk) print("Recall:", recall_sk)
```

```
print("\nLearned Coefficients (Custom Model):") print(model.W) print("Intercept:",
model.b)
```