



```

# Оновлення ваг
self.w -= self.lr * dw
self.b -= self.lr * db

def predict(self, X):
    y_pred = self.sigmoid(np.dot(X, self.w) + self.b)
    return [1 if i > 0.5 else 0 for i in y_pred]

model = LogisticRegression(learning_rate=0.1, n_iters=1000)
model.fit(X_train, y_train)

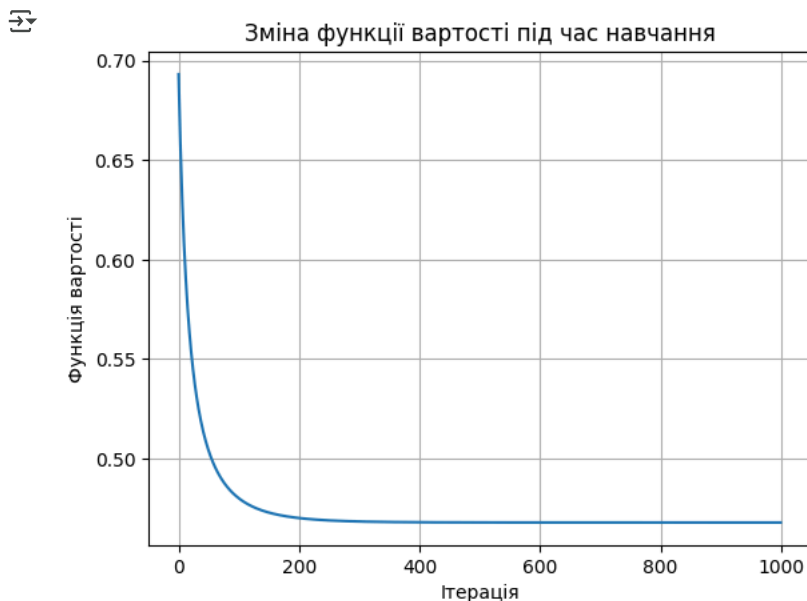
y_pred = model.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Precision:", precision_score(y_test, y_pred))
print("Recall:", recall_score(y_test, y_pred))
print("F1 Score:", f1_score(y_test, y_pred))

Accuracy: 0.7532467532467533
Precision: 0.6491228070175439
Recall: 0.6727272727272727
F1 Score: 0.6607142857142857

plt.plot(range(len(model.costs)), model.costs)
plt.xlabel("Ітерація")
plt.ylabel("Функція вартості")
plt.title("Зміна функції вартості під час навчання")
plt.grid(True)
plt.show()

```



Чим відрізняється логістична регресія від лінійної регресії? Лінійна регресія прогнозує неперервні значення (реальні числа).

Логістична регресія прогнозує ймовірність приналежності до класу (класифікація, вихід — від 0 до 1).

2. У чому полягає роль сигмоїдної функції в логістичній регресії? Сигмоїдна функція  $g(z) = 1 / (1 + e^{(-z)})$  перетворює будь-яке число на діапазон (0, 1), що дозволяє інтерпретувати вихід як ймовірність належності до класу 1.
3. Чому для логістичної регресії не використовується функція вартості, як у лінійній регресії? У логістичній регресії помилка не є квадратичною, тому замість MSE використовується логістична функція вартості (крос-ентропія), яка краще підходить для оцінки ймовірностей.
4. Які є способи запобігання перенаванчання (overfitting) в логістичній регресії? Регуляризація (L1 або L2)

Зменшення розмірності даних

Більший обсяг тренувальної вибірки

Раннє зупинення навчання

Dropout (у більш складних моделях, наприклад нейронних мережах)

5. Як впливає параметр регуляризації  $\lambda$  на модель? Чим більше  $\lambda$ , тим сильніше штрафуються великі ваги, що може зменшити перенавчання, але погіршити точність.

Чим менше  $\lambda$ , тим менше обмеження на ваги, що може призвести до перенавчання.

6. Як обчислити точність, precision, recall та F1-score для моделі класифікації? Формули:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

$$\text{Precision} = TP / (TP + FP)$$

$$\text{Recall} = TP / (TP + FN)$$

$$\text{F1-score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

де:

TP — істинно позитивні

TN — істинно негативні

FP — хибно позитивні

FN — хибно негативні

7. Що таке матриця помилок (confusion matrix) і як її інтерпретувати? Матриця помилок — це таблиця 2x2 для задачі бінарної класифікації, що показує кількість:

Правильних і неправильних прогнозів для кожного класу.

Структура:

Прогноз: 0 Прогноз: 1 Факт: 0 TN FP Факт: 1 FN TP

8. Які переваги та недоліки логістичної регресії порівняно з іншими методами класифікації? Переваги:

Простота реалізації та інтерпретації

Швидке тренування

Працює добре для лінійно роздільних даних

Недоліки:

Не працює добре з нелінійними залежностями

Чутлива до мультиколінеарності між ознаками

9. У яких випадках доцільно використовувати логістичну регресію, а в яких інші методи класифікації? Використовуйте логістичну регресію, коли:

Є лінійна межа розділення між класами

Потрібна інтерпретація коефіцієнтів

Використовуйте інші методи (наприклад, дерева рішень, SVM, нейронні мережі), коли:

Є складні нелінійні залежності

Дані великі і потребують більш гнучкої моделі

10. Як можна інтерпретувати коефіцієнти моделі логістичної регресії? Коефіцієнт показує, наскільки зміниться логарифм odds (співвідношення ймовірностей) при збільшенні ознаки на 1 одиницю.

Якщо коефіцієнт  $> 0$  — ознака збільшує ймовірність класу 1.

Якщо коефіцієнт  $< 0$  — ознака зменшує ймовірність класу 1.