Sprawozdanie ćwiczenia nr 4 – Łukasz Szydlik

Cel i opis eksperymentów

Celem ćwiczenia jest zaimplementowanie algorytmu regresji logistycznej oraz ocena jego działania w klasyfikacji na zbiorze danych ”Breast Cancer Wisconsin (Diagnostic)”. Należy policzyć wynik dla przynajmniej 3 różnych sposobów przygotowania danych, na przykład usuwając niektóre kolumny, dodając normalizację wartości.

Dane dostępne pod adresem:

https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

Instrukcja programu

Instalacja

1. Należy pobrać repozytorium

2. Wejść w folder lab4

3. Upewnić się, że posiadamy wymagane biblioteki:

‘pip install -r requirements.txt’

4. Zainstalować rozszerzenie Jupyter (VS Code)

Uruchomienie programu

Odpowiednio sformatowane dane możemy utworzyć za pomocą komendy:

`python .\createDataFile.py`

W celu uruchomienia LogisticRegression.ipynb wymagane jest korzystanie z notatnika Jupyter

W Jupyter ustawiamy kernel na nasze środowisko python. Następnie po naciśnięciu ”Run All” wyniki eksperymentów oraz ich wykresy zostaną wyświetlane w notatniku.

Możemy sami przygotować własne dane według wzoru ”Implement data and parameters”, a następnie wprowadzić je do funkcji show_results().

Wyniki ćwiczenia

W ramach eksperymentów przeprowadzono testy dla czterech różnych konfiguracji przetwarzania danych wejściowych:

1. Użycie wszystkich cech bez normalizacji.
2. Użycie wszystkich cech z normalizacją.
3. Wybranie podzbioru cech (kolumny 15-21) bez zastosowania normalizacji.
4. Wybranie podzbioru cech (kolumny 15-21) z zastosowaniem normalizacji.

Podział zbioru danych na zbiory uczący i testowy wykonano w stosunku 75:25. Dla każdej konfiguracji mierzono jakość działania algorytmu za pomocą metryk: accuracy, F1 oraz AUROC.

Startowe wagi i bias wybrano losowo według rozkładu normalnego.

Eksperyment1				
Normalizacja:	Nie		Liczba cech:	30
Parametry			Parametry	
Learning rate	0,001		Learning rate	0,001
Iterations	100		Iterations	1000
Wyniki Train			Wyniki Train	
Accuracy	0,73		Accuracy	0,91
F1	0,42		F1	0,87
AUROC	0,70		AUROC	0,94
Wyniki Test			Wyniki Test	
Accuracy	0,73		Accuracy	0,92
F1	0,45		F1	0,88
AUROC	0,64		AUROC	0,94

Eksperyment3				
Normalizacja:	Nie		Liczba cech:	6
Parametry			Parametry	
Learning rate	0,01		Learning rate	0,01
Iterations	100		Iterations	1000
Wyniki Train			Wyniki Train	
Accuracy	0,16		Accuracy	0,68
F1	0,12		F1	0,29
AUROC	0,07		AUROC	0,93
Wyniki Test			Wyniki Test	
Accuracy	0,13		Accuracy	0,69
F1	0,09		F1	0,35
AUROC	0,04		AUROC	0,90

Eksperyment2				
Normalizacja:	Tak		Liczba cech:	30
Parametry			Parametry	
Learning rate	0,1		Learning rate	0,1
Iterations	100		Iterations	1000
Wyniki Train			Wyniki Train	
Accuracy	0,96		Accuracy	0,99
F1	0,94		F1	0,99
AUROC	0,99		AUROC	1,00
Wyniki Test			Wyniki Test	
Accuracy	0,95		Accuracy	0,95
F1	0,93		F1	0,93
AUROC	0,97		AUROC	0,99

Eksperyment4				
Normalizacja:	Tak		Liczba cech:	6
Parametry			Parametry	
Learning rate	0,1		Learning rate	0,1
Iterations	100		Iterations	1000
Wyniki Train			Wyniki Train	
Accuracy	0,89		Accuracy	0,92
F1	0,85		F1	0,89
AUROC	0,96		AUROC	0,98
Wyniki Test			Wyniki Test	
Accuracy	0,86		Accuracy	0,90
F1	0,80		F1	0,87
AUROC	0,95		AUROC	0,97

Eksperyment 1: Wszystkie cechy bez normalizacji

Algorytm osiągnął wysoką skuteczność klasyfikacji, jednak brak normalizacji wpływa na stabilność uczenia.

Eksperyment 2: Wszystkie cechy z normalizacją

Normalizacja poprawiła wyniki klasyfikacji na zbiorze testowym. Po 100 iteracjach każda kolejna nie poprawia znacząco modelu.

Eksperyment 3: Podzbór cech (15-21) bez normalizacji

Algorytm osiągnął słabe wyniki. Brak normalizacji oraz ograniczenie cech wpłynęły negatywnie na jakość klasyfikacji w porównaniu do pełnych danych.

Eksperyment 4: Podzbór cech (15-21) z normalizacją

Redukcja liczby cech pogorszyła jakość klasyfikacji. Jednak algorytm nadal osiąga dobre wyniki, co wskazuje, że nie wszystkie cechy są kluczowe dla zadania.

Wykresy

Wykresy funkcji kosztu, ROC Curve oraz przykładowe predykcje modelu dostępne po uruchomieniu pliku LogisticRegression.ipynb w notatniku Jupyter

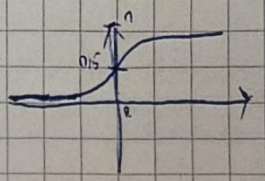
Wnioski

1) Normalizacja danych poprawia skuteczność algorytmu.
2) Redukcja liczby cech może być korzystna w celu zmniejszenia złożoności obliczeniowej, ale należy uważać, aby nie usuwać istotnych atrybutów.
3) Zastosowanie mniejszych podzbiorów cech pozwala na redukcję czasu treningu, ale może wymagać starannej selekcji cech kluczowych dla problemu.
4) Algorytm regresji logistycznej dobrze radzi sobie z klasyfikacją w zastosowanym zbiorze danych z normalizacją, osiągając AUROC powyżej 0.9 we wszystkich testach.
5) Wyniki wskazują, że normalizacja jest szczególnie ważna, gdy dane wejściowe zawierają cechy o różnych zakresach wartości.
6) Zwiększenie liczby iterations ponad 100 lub 1000 (dla małego kroku) nie przynosi już konkretnej efektywności.
7) Ważne jest dobranie parametru learning_rate. Dla zbyt dużego parametru algorytm nie zadziała poprawnie, a dla zbyt małego liczba iterations potrzebna do osiągnięcia satysfakcjonujących wyników będzie musiała być ogromna.

Notatki na następnych dwóch stronach:

Funkcja sigmoid

$$\sigma(z) = \frac{1}{1+e^{(-z)}}$$



- Obserwacje (dane)
  - cechy
    $X \in R^{N \times D}$
    N - liczba wymiarów
    D - liczba cech
  - kategorie
    $y \in \{0,1\}^N$

- Model / Reklykcja

$$h(x; w, b) = \sigma(x_1 w_1 + x_2 w_2 + \dots + x_D w_D + b)$$
$$= \hat{y} = \sigma(Xw + b) \in (0,1)^N$$

- parametry modelu
  $w \in R^D$, $b \in R$

- Funkcja straty - entropia krzyżowa (cross entropy, CE)

$$CE(\hat{y}, y) = -\frac{1}{N} \sum_{i=1}^{N} \left(y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \right) \in R$$

$$J(w, b) = CE(\hat{y}, y)$$

- Pochodne funkcji straty

$$\nabla J(w) = \frac{1}{N} X^T \delta \qquad \delta = \hat{y} - y \in R^N$$
$$\nabla J(b) = \frac{1}{N} (\delta_1 + \delta_2 + \delta_3 + \dots + \delta_N)$$

MIARY JAKOŚCI

	Sklasyfikowany jako chory „M"	Sklasyfikowany jako zdrowy „B"
pacjent chory „M"	TP (true positive)	FN (false negative)
pacjent zdrowy „B"	FP (false positive)	TN (true negative)

* Trafność (accuracy)

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

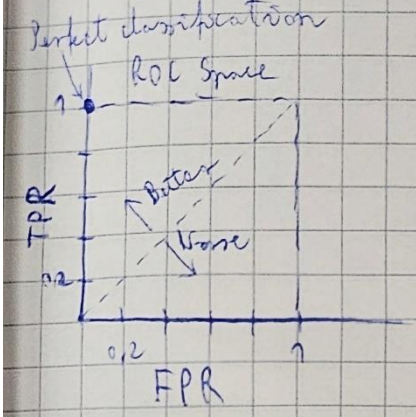
- Sensytywność (recall (prawdopodobieństwo wykrycia choroby)

$$\text{sensitivity} = \text{recall} = TPR = \frac{TP}{TP + FN}$$

- Specyficzność (prawdopodobieństwo negatywnego dla zdrowych pacjentów)

$$\text{specyficzność} = TNR = \frac{TN}{TN + FP}$$

- Krzywa ROC



TPR = sensitivity

FPR = 1 - specificity

*AUROC - area under ROC (miara jakości klasyfikatora)

- Pole pod krzywą ROC

- Przyjmuje on wartości:

  1, dla najlepszego klasyfikatora

  0,5, dla losowego

  0, dla najgorszego

*F1 (miara jakości klasyfikatora)

- zależna od precision i recall (sensitivity)

- Precision (w dla ponizszych klasyfikator zwrócił dobry wynik)

$$precision = \frac{TP}{TP + FP}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$