
VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMACINIŲ SISTEMŲ INŽINERIJOS STUDIJŲ PROGRAMA

# Antra užduotis (Vieno neurono mokymas sprendžiant klasifikavimo uždavinį)

Laboratorinio darbo ataskaita

Atliko: Klaidas Kubilius

VU el. p.: klaidas.kubilius@mif.stud.vu.lt

Vilnius
2025

# 1. DARBO EIGA

## 1.1. Duomenys

Buvo naudojama krūties vėžio duomenų aibė, kurią sudarė 699 įrašai ir 11 požymių (įskaitant ID ir klasę). Pašalinus trūkstamas reikšmes, įrašų skaičius sumažėjo iki 683. Taip pat klasių žymės buvo pakeistos: 2 (nepiktybinis navikas) ir 4 (piktybinis navikas) konvertuotos į 0 ir 1.

Požymiai:

- Clump Thickness: 1 - 10
- Uniformity of Cell Size: 1 - 10
- Uniformity of Cell Shape: 1 - 10
- Marginal Adhesion: 1 - 10
- Single Epithelial Cell Size: 1 - 10
- Bare Nuclei: 1 - 10
- Bland Chromatin: 1 - 10
- Normal Nucleoli: 1 - 10
- Mitoses: 1 - 10

## 1.2. Duomenų padalijimas

Paruoša duomenų aibė buvo padalinta į tris dalis: mokymo (80 %), validacijos (10 %) ir testavimo (10 %) rinkinius. Rinkinys padalijamas į 80 % ir 20 % dalis, kur 80 % duomenų atsitiktinai priskiriama mokymo rinkiniui. Likę 20 % duomenų toliau padalijami į dvi lygias dalis – 10 % validacijos rinkiniui ir 10 % testavimo rinkiniui, taip pat atsitiktinai parenkant įrašus.

- Mokymo rinkinys – naudojamas modelio mokymui, svorių ir parametrų atnaujinimui.
- Validacijos rinkinys – padeda įvertinti modelio našumą mokymo metu ir yra naudojamas hiperparametrų derinimui. Po kiekvienos mokymo iteracijos modelis tikrinamas su validacijos duomenimis, siekiant nustatyti, ar jis nėra per daug pritaikytas mokymo duomenims (pernelyg išmokęs) ir ar geba generalizuoti žinias naujiems duomenims.
- Testavimo rinkinys – skirtas galutiniam modelio vertinimui po mokymo ir validavimo.

## 1.3. Svoriai ir poslinkis

Generuojami atsitiktiniai skaičiai pagal vienodą paskirstymą tarp 0 ir 1. Kiekvienas svoris yra paskirtas atsitiktinai intervale. Pradinis poslinkis yra 0.

## 1.4. Gradientiniai nusileidimai

- **Mokymo epocha** – tai neuronų mokymo proceso dalis, kurios metu apdorojamas visas mokymo duomenų rinkinys vieną kartą.

- **Paketinis gradientinis nusileidimas** – taikant paketinį gradientinio nusileidimo algoritmą (batch gradient descent) svoriai keičiami pagal formulę:

  $w_k := w_k - \eta \nabla E(W) := w_k - \eta \frac{1}{m} \sum_{i=1}^{m} \nabla E_i(W)$

  Taikant šį metodą, vienos iteracijos metu panaudojami visi mokymo duomenys. Pirmiausia į neuroną perduodame visus mokymo duomenis ir apskaičiuojame kiekvieno duomenų įrašo paklaidos funkcijos gradientą. Tada imame gradientų vidurkį ir atnaujiname svorius (ir poslinkį) naudodami apskaičiuotą vidurkį. Paketinio gradientinio nusileidimo atveju viena epocha atitinka vieną iteraciją.

- **Stochastinis gradientinis nusileidimas** – algoritme tikras funkcijos $E\ W$ gradientas aproksimuojamas gradientu, gautu pagal vieną mokymo duomenų įrašą:

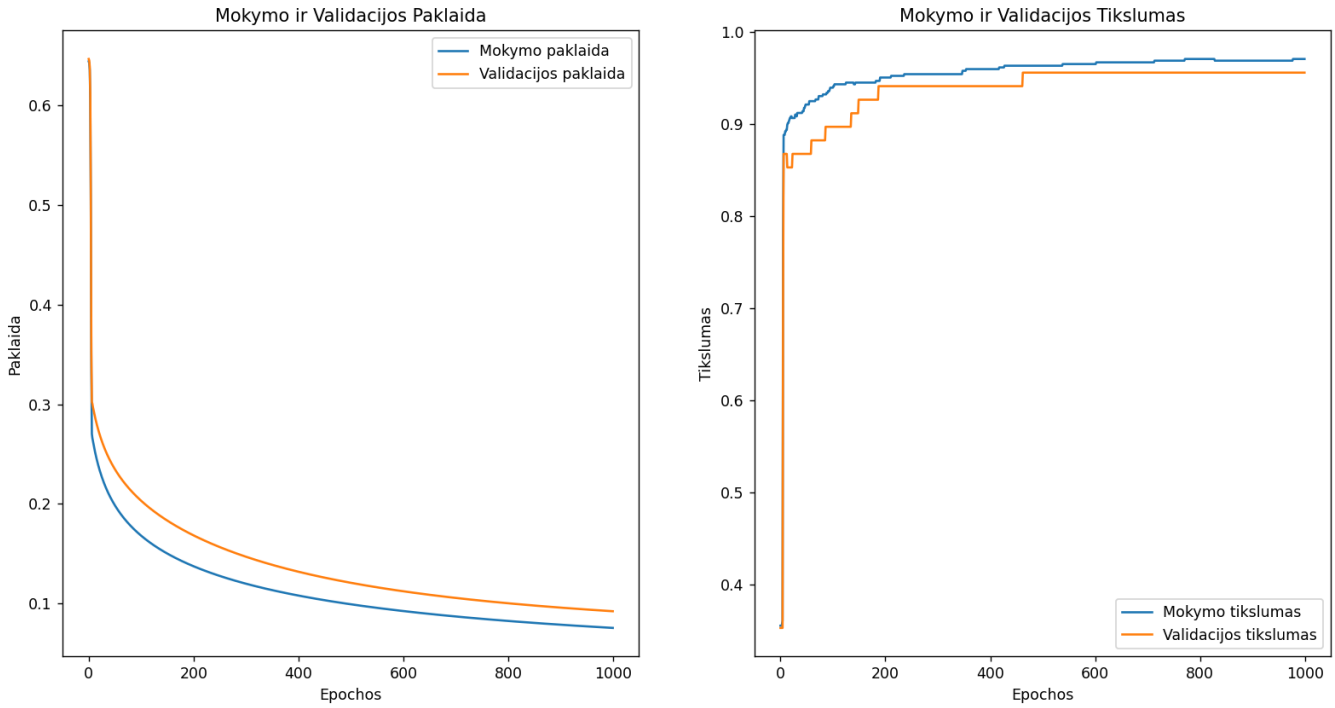
  $w_k := w_k - \eta \nabla E(W)$

  Siekiama, kad paklaida būtų minimali i-tajam mokymo duomenų įrašui. Taikant stochastinį gradientinį nusileidimą, vienos iteracijos metu panaudojamas tik vienas mokymo duomenų įrašas. Kiekvienam įrašui skaičiuojamas gradientas ir atnaujinami svoriai (ir poslinkis). Stochastinio gradientinio nusileidimo atveju viena epocha atitinka $m$ iteracijų, čia $m$ yra mokymo duomenų kiekis.

## 1.5. Tyrimo rezultatai

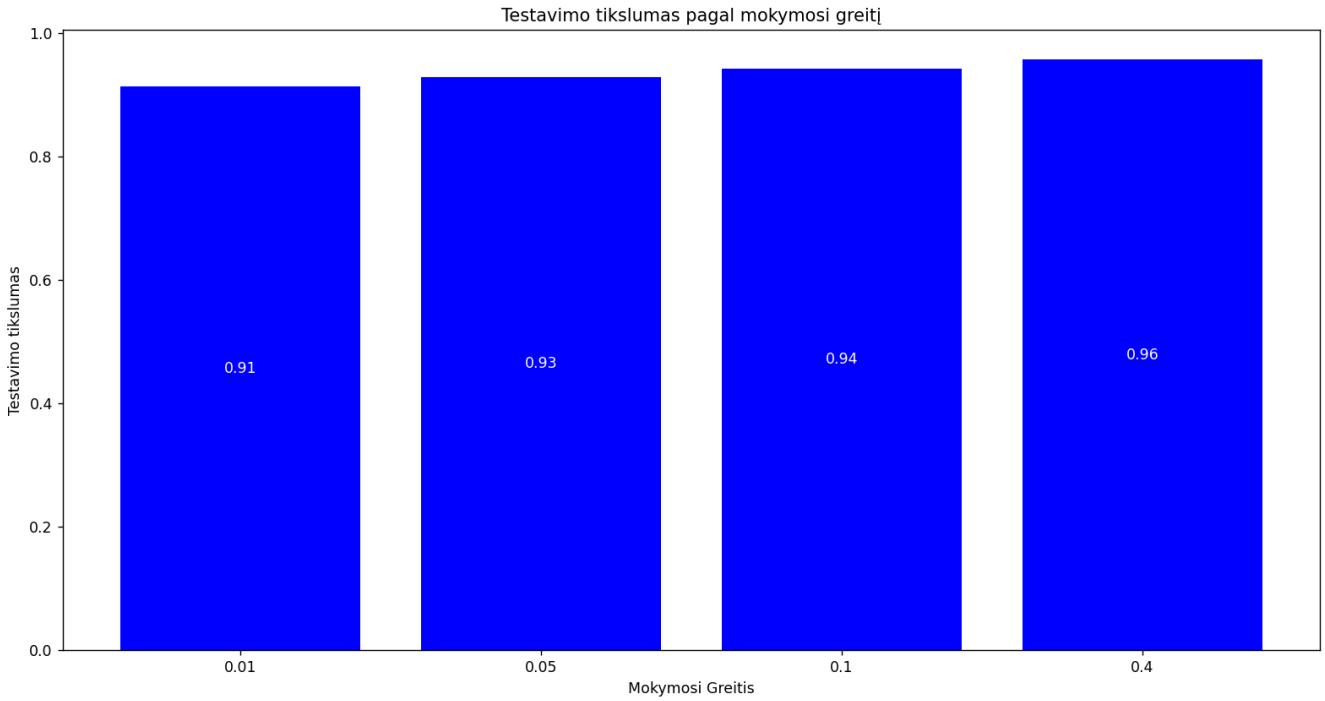
Atlikti tyrimai su paketiniu gradientiniu nusileidimu ir stochastiniu gradientiniu nusileidimu ir skirtingais mokymo greičiais.

- **Metodas:** Paketinis gradientinis nusileidimas.
- **Mokymosi greitis:** 0.1.
- **Epochų skaičius:** 1000.
- **Svoriai**: 0,064;  0,472; 0,329;  0,055; -0,332; 0,376; 0,0404;  0,209;  0,109.
- **Poslinkis**: -4,453.
- **Testavimo paklaida**: 0,092.
- **Testavimo tikslumas**: 94,20%.
- **Mokymo laikas**: 0,13 sekundės.




1.1 pav. Mokymo ir validavimo paklaida ir tikslumas epochų metu (Paketinis)

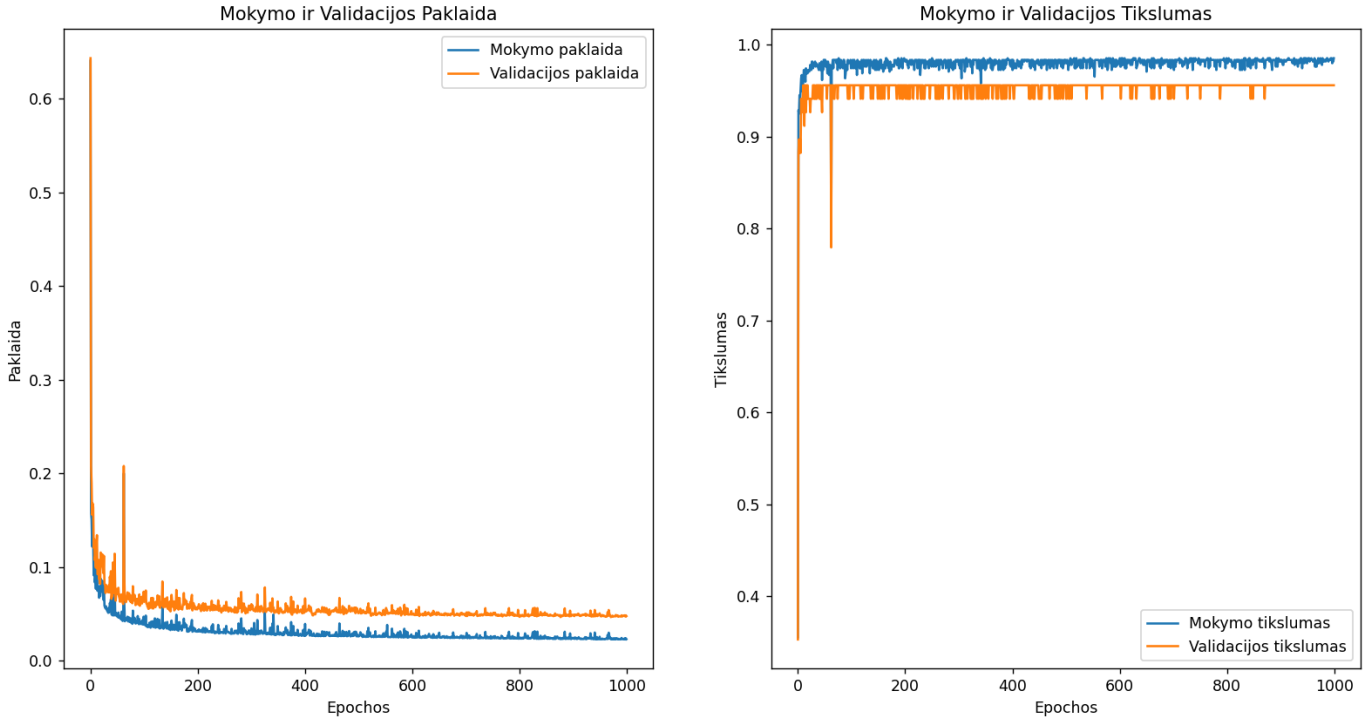
Pastebimas staigus paklaidos sumažėjimas ir tikslumo augimas pirmosiose epochose. Vėliau tiek mokymo, tiek validacijos paklaidos toliau pamažu mažėja, tačiau vis lėčiau, o tikslumo augimas taip pat sulėtėja.




1.2 pav. Testavimo tikslumas keičiant mokymo greitį (Paketinis)

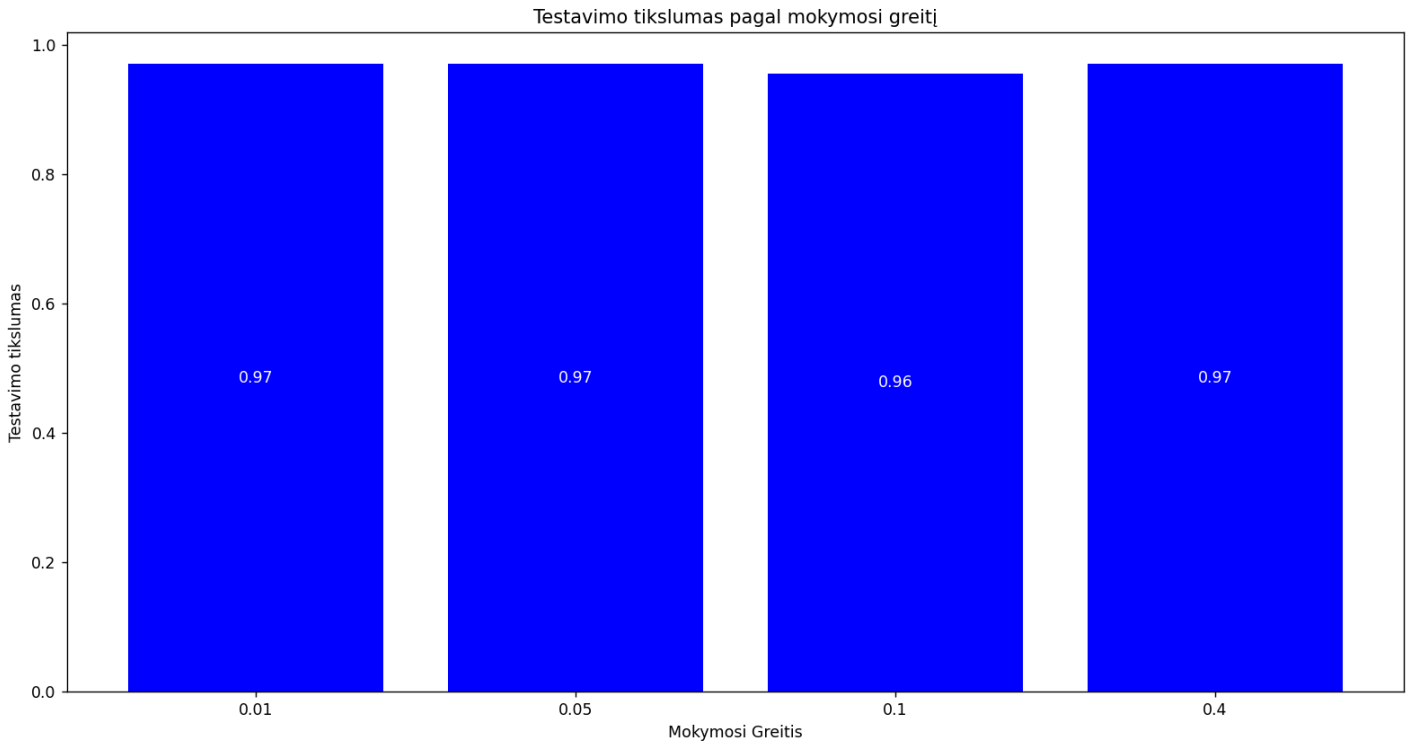
Pastebimas nedidelis testavimo tikslumo sumažėjimas, didinant mokymo greitį, kelių procentų skirtumu.

- **Metodas:** Stochastinis gradientinis nusileidimas.
- **Mokymosi greitis:** 0.1.
- **Epochų skaičius:** 1000.
- **Svoriai**: 0,712;  0,639; 1,069; 0,168; 0,117; 0,801; 0,411; 0,63; 0,802.
- **Poslinkis**: -15.804.
- **Testavimo paklaida**: 0,0402.
- **Testavimo tikslumas**: 95,65%.
- **Mokymo laikas**: 12,28 sekundės. (Apie 94 kartus didesnis už paketinį).




1.3 pav. Mokymo ir validavimo paklaida ir tikslumas epochų metu (Stochastinis)

Kaip ir paketinio gradientinio nusileidimo metode pastebimas staigus paklaidos sumažėjimas ir tikslumo augimas pirmosiose epochose. Vėliau tiek mokymo, tiek validacijos paklaidos toliau pamažu mažėja, tačiau vis lėčiau, o tikslumo augimas taip pat sulėtėja. Esminis skirtumas, kad stochastiniame metode, dėl didesnio duomenų šurmulio ir dažnesnio svorių atnaujinimo, linijos turi bangas, kurios atspindi didesnį svyravimą tarp epochų.




1.4 pav. Testavimo tikslumas keičiant mokymo greitį (Stochastinis)

Taikant stochastinį gradientinį nusileidimo metodą ir keičiant mokymo greitį testavimo tisklumas beveik nesikeičia.

## 1.6. Geriausi rezultatai

Buvo surastas paketinio gradientinio nusileidimo geriausias variantas tarp šių mokymo greičių: 0,001, 0,01, 0,05, 0,1, 0,4, 0,8.

- Geriausias mokymosi greitis: 0,05.
- Epochu skaičius: 1000.
- Mokymo paklaida paskutinėje epochoje: 0,0984.
- Validavimo paklaida paskutinėje epochoje: 0,1196.
- Mokymo tikslumas paskutinėje epochoje: 0,9634.
- Validavimo tikslumas paskutinėje epochoje: 0,9559.
- Testavimo paklaida: 0,0290.
- Testavimo tikslumas: 0,9710.

1.1 lentelė. Testavimo duomenų įrašų palyginimas

| | | |
|---|---|---|
| Testavimo įrašas 1 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 2 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 3 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 4 | Nustatyta klase: 0 | Tikroji klase: 0 |

| | | |
|---|---|---|
| Testavimo įrašas 5 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 6 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 7 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 8 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 9 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 10 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 11 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 12 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 13 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 14 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 15 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 16 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 17 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 18 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 19 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 20 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 21 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 22 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 23 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 24 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 25 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 26 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 27 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 28 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 29 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 30 | Nustatyta klase: 1 | Tikroji klase: 0 |
| Testavimo įrašas 31 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 32 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 33 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 34 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 35 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 36 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 37 | Nustatyta klase: 0 | Tikroji klase: 1 |
| Testavimo įrašas 38 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 39 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 40 | Nustatyta klase: 0 | Tikroji klase: 0 |

| | | |
|---|---|---|
| Testavimo įrašas 41 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 42 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 43 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 44 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 45 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 46 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 47 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 48 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 49 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 50 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 51 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 52 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 53 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 54 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 55 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 56 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 57 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 58 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 59 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 60 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 61 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 62 | Nustatyta klase: 1 | Tikroji klase: 1 |
| Testavimo įrašas 63 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 64 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 65 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 66 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 67 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 68 | Nustatyta klase: 0 | Tikroji klase: 0 |
| Testavimo įrašas 69 | Nustatyta klase: 0 | Tikroji klase: 0 |

Buvo surastas stochastinio gradientinio nusileidimo geriausias variantas tarp šių mokymo greičių: 0,001, 0,01, 0,05, 0,1, 0,4, 0,8.

- Geriausias mokymosi greitis: 0,001.
- Epochu skaicius: 1000.
- Mokymo paklaida paskutineje epochoje: 0,0964.
- Validavimo paklaida paskutineje epochoje: 0,1154.
- Mokymo tikslumas paskutineje epochoje: 0,9689.
- Validavimo tikslumas paskutineje epochoje: 0,9559.
- Testavimo paklaida: 0,0435.
- Testavimo tikslumas: 0,9565.

1.2 lentelė. Testavimo duomenų įrašų palyginimas

| | | |
|---|---|---|
| Testavimo įrašas 1 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 2 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 3 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 4 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 5 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 6 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 7 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 8 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 9 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 10 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 11 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 12 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 13 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 14 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 15 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 16 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 17 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 18 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 19 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 20 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 21 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 22 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 23 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 24 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 25 | Nustatyta klasė: 1 | Tikroji klasė: 1 |

| Testavimo įrašas 26 | Nustatyta klasė: 1 | Tikroji klasė: 0 |
|---|---|---|
| Testavimo įrašas 27 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 28 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 29 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 30 | Nustatyta klasė: 1 | Tikroji klasė: 0 |
| Testavimo įrašas 31 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 32 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 33 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 34 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 35 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 36 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 37 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 38 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 39 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 40 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 41 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 42 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 43 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 44 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 45 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 46 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 47 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 48 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 49 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 50 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 51 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 52 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 53 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 54 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 55 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 56 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 57 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 58 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 59 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 60 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 61 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 62 | Nustatyta klasė: 1 | Tikroji klasė: 1 |
| Testavimo įrašas 63 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 64 | Nustatyta klasė: 0 | Tikroji klasė: 0 |

| | | |
|---|---|---|
| Testavimo įrašas 65 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 66 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 67 | Nustatyta klasė: 1 | Tikroji klasė: 0 |
| Testavimo įrašas 68 | Nustatyta klasė: 0 | Tikroji klasė: 0 |
| Testavimo įrašas 69 | Nustatyta klasė: 0 | Tikroji klasė: 0 |

Didesnį tiksluma turėjo paketinis gradientis nusileidimas, bet tik keliais procentais. Geriausi mokymo greičiai tarp metodų skiriasi žymiai.

# 2. IŠVADOS

- Mokymosi greitis stochastiniam metodui turėjo mažesnę įtaka palyginus su paketiniu metodu.
- Skirtumas tarp metodų tikslumo ir paklaidos yra mažas, bet stochastinio metodo mokymo laikas yra žymiai ilgesnis už paketinio. Atsižvelgus į laiko ribojimą geriau būtų paketinis.
- Paketinio metodo atveju didėjant mokymo greičiu tikslumas taip pat didėja, o stochastinio metodu atveju skirtumai tarp mokymo greičio labai maži.
- Po kelių šimtų epochų mokymo tikslumo didėjimas tampa nereikšmingas, nes modelis pasiekia konvergenciją.

# 3. PAPILDOMA

Generuota ChatGPT:

- Gramatinėm klaidom taisyti ataskaitoje
- Stochastinio metodo kodui

# 4. PRIEDAI

## 4.1. Kodas

```
 import numpy as np
import pandas as pd
import time
import matplotlib.pyplot as plt
from sklearn.utils import shuffle
from sklearn.model_selection import train_test_split

# Sigmoidinė aktyvavimo funkcija
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Sigmoidinės funkcijos išvestinė
def sigmoid_derivative(x):
    return x * (1 - x)

# Funkcija, skirta modelio mokymui su batch ar stochastic metodu
def train_model(X_train, y_train, X_val, y_val, learning_rate, epochs, n_features,
method='batch'):
    w = np.random.uniform(0, 1, (n_features, 1))
    b = 0.0

    # Laiko matavimas
    start_time = time.time()

    # Saugojimui
    train_losses, val_losses = [], []
    train_accuracies, val_accuracies = [], []

    m = len(X_train)  # Mokymo duomenų dydis
    epoch = 0

    while epoch < epochs:
        gradientSum = np.zeros(n_features) # Masyvas, kuriame saugomi gradientų sumos

        if method == 'batch':
            # Batch mokymo metodas
            z = np.dot(X_train, w) + b # Funkcija Xw + b
            y_pred = sigmoid(z) # Priskiria arčiausiai 0 arba 1
            error = y_pred - y_train # Netikslumas tarp prognozės ir tikros reikšmės
            gradient_w = np.dot(X_train.T, error) / len(X_train) # Svorių gradientas
            gradient_b = np.mean(error)    # Poslinkio gradientas

            w -= learning_rate * gradient_w # Atnaujina svorius
            b -= learning_rate * gradient_b # Atnaujina poslinkį

        elif method == 'stochastic':
            # Stochastic mokymo metodas
            X_train, y_train = shuffle(X_train, y_train) # Duomenų maišymas
            for i in range(len(X_train)):
                xi = X_train[i].reshape(1, -1) # Paimama i-oji treniravimo duomenų
eilutė ir paverčiama 2D eilutės vektoriumi
                yi = y_train[i].reshape(1, -1) # Paimama i-oji tikroji reikšmė ir
paverčiama 2D eilutės vektoriumi
                linear_output = np.dot(xi, w) + b # Funkcija Xw + b
                y_pred = sigmoid(linear_output) # Priskiria arčiausiai 0 arba 1
                error = yi - y_pred # Netikslumas tarp prognozės ir tikros reikšmės
                w_gradient = np.dot(xi.T, error * sigmoid_derivative(y_pred))  #
Svorių gradientas
                b_gradient = np.sum(error * sigmoid_derivative(y_pred)) # Poslinkio
gradientas
```

```
                # Svorių ir poslinkio atnaujinimas
                w += learning_rate * w_gradient
                b += learning_rate * b_gradient

        # Klasifikavimo tikslumo ir paklaidos skaičiavimas
        train_output = sigmoid(np.dot(X_train, w) + b)  # Apskaičiuojama prognozė
mokymo duomenims
        train_predictions = (train_output >= 0.5).astype(int)
        train_accuracy = np.mean(train_predictions == y_train)
        train_loss = np.mean(np.abs(y_train - train_output))

        val_output = sigmoid(np.dot(X_val, w) + b)  # Apskaičiuojama prognozė
validacijos duomenims
        val_predictions = (val_output >= 0.5).astype(int)
        val_accuracy = np.mean(val_predictions == y_val)
        val_loss = np.mean(np.abs(y_val - val_output))

        # Išsaugoti paklaidas ir tikslumus
        train_losses.append(train_loss)
        val_losses.append(val_loss)
        train_accuracies.append(train_accuracy)
        val_accuracies.append(val_accuracy)

        epoch += 1

        # Kiekvienos 100 epochų metu išvedami rezultatai
        if epoch % 100 == 0:
            print(f"Epochos {epoch}: Mokymo paklaida {train_loss:.4f}, Mokymo
tikslumas {train_accuracy * 100:.2f}%")
            print(f"Epochos {epoch}: Validacijos paklaida {val_loss:.4f}, Validacijos
tikslumas {val_accuracy * 100:.2f}%")

    # Laiko skaičiavimas
    end_time = time.time()
    training_time = end_time - start_time

    return w, b, train_losses, val_losses, train_accuracies, val_accuracies,
training_time

# Duomenų įkėlimas ir paruošimas
file_path = "cleaned_breast_cancer.csv"
df = pd.read_csv(file_path)
X = df.drop(columns=["Class"]).values
y = df["Class"].values.reshape(-1, 1)

# Duomenų padalijimas į mokymo, validacijos ir testavimo rinkinius
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2,
random_state=100)
X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=100)

# Modelio parametrai
learning_rate = 0.1
epochs = 1000
n_features = X_train.shape[1] #požymiai
method = 'stochastic'  # Pasirinkti mokymo metodą: batch arba stochastic

# Mokymas
w, b, train_losses, val_losses, train_accuracies, val_accuracies, training_time =
train_model(
    X_train, y_train, X_val, y_val, learning_rate, epochs, n_features, method)

# Testavimas
test_output = sigmoid(np.dot(X_test, w) + b)
test_predictions = (test_output >= 0.5).astype(int)
```

```
test_accuracy = np.mean(test_predictions == y_test)
test_loss = np.mean(np.abs(y_test - test_output))

# Rezultatų išvedimas
print("Galutiniai svoriai:", w.flatten())
print("Galutinis poslinkis:", b)
print(f"Testavimo paklaida: {test_loss:.4f}")
print(f"Testavimo tikslumas: {test_accuracy * 100:.2f}%")
print(f"Mokymo laikas: {training_time:.2f} sekundes")

# Grafikai
plt.figure(figsize=(12, 5))

# Paklaidos grafikas
plt.subplot(1, 2, 1)
plt.plot(range(epochs), train_losses, label="Mokymo paklaida")
plt.plot(range(epochs), val_losses, label="Validacijos paklaida")
plt.xlabel("Epochos")
plt.ylabel("Paklaida")
plt.title("Mokymo ir Validacijos Paklaida")
plt.legend()

# Tikslumo grafikas
plt.subplot(1, 2, 2)
plt.plot(range(epochs), train_accuracies, label="Mokymo tikslumas")
plt.plot(range(epochs), val_accuracies, label="Validacijos tikslumas")
plt.xlabel("Epochos")
plt.ylabel("Tikslumas")
plt.title("Mokymo ir Validacijos Tikslumas")
plt.legend()

plt.show()

# Įvairūs mokymosi greičiai
learning_rates = [0.001, 0.01, 0.05, 0.1, 0.4, 0.8]  # Skirtingi mokymosi greičiai
test_accur_lr = []

# Saugo validavimo tikslumus ir kitus reikalingus duomenis
val_accuracies = [] # Saugo validavimo tikslumus
train_losses_all = []  # Saugo visus mokymo praradimus
val_losses_all = []    # Saugo visus validavimo praradimus
train_accuracies_all = []  # Saugo visus mokymo tikslumus
val_accuracies_all = []    # Saugo visus validavimo tikslumus

for lr in learning_rates:
    # skirtingi mokymosi greičiai
    w, b, train_losses, val_losses, train_accuracies, val_accuracies_temp,
training_time = train_model(
        X_train, y_train, X_val, y_val, lr, epochs, n_features, method)

    # Išsaugome praradimus ir tikslumus visoms epochoms
    train_losses_all.append(train_losses)
    val_losses_all.append(val_losses)
    train_accuracies_all.append(train_accuracies)
    val_accuracies_all.append(val_accuracies_temp)

    # Išsaugome paskutinės epochos validavimo tikslumą
    val_accuracies.append(val_accuracies_temp[-1] if val_accuracies_temp else 0)

    # Testavimas
    test_output = sigmoid(np.dot(X_test, w) + b)
    test_predictions = (test_output >= 0.5).astype(int)
    test_accuracy = np.mean(test_predictions == y_test)
    test_accur_lr.append(test_accuracy)

# Surasti geriausią rezultatą pagal validavimo duomenis
```

```
best_idx = np.argmax(val_accuracies)  # Indeksas, kurio validavimo tikslumas
didžiausias
best_lr = learning_rates[best_idx]  # Geriausias mokymosi greitis

# Gautos reikšmės
best_train_loss = train_losses_all[best_idx][-1]  # Paskutinės epochos mokymo paklaida
best_val_loss = val_losses_all[best_idx][-1]  # Paskutinės epochos validavimo paklaida
best_train_acc = train_accuracies_all[best_idx][-1]  # Paskutinės epochos mokymo
tikslumas
best_val_acc = val_accuracies_all[best_idx][-1]  # Paskutinės epochos validavimo
tikslumas

# Paklaida testavimo duomenims
test_output = sigmoid(np.dot(X_test, w) + b)
test_predictions = (test_output >= 0.5).astype(int)
test_loss = np.mean((test_predictions - y_test) ** 2) # Testavimo paklaida
test_accuracy = np.mean(test_predictions == y_test)  # Testavimo tikslumas

# Geriausi rezultatai
print(f"Geriausias mokymosi greitis: {best_lr}")
print(f"Epochu skaicius: {epochs}")
print(f"Mokymo paklaida paskutineje epochoje: {best_train_loss:.4f}")
print(f"Validavimo paklaida paskutineje epochoje: {best_val_loss:.4f}")
print(f"Mokymo tikslumas paskutineje epochoje: {best_train_acc:.4f}")
print(f"Validavimo tikslumas paskutineje epochoje: {best_val_acc:.4f}")
print(f"Testavimo paklaida: {test_loss:.4f}")
print(f"Testavimo tikslumas: {test_accuracy:.4f}")

# Kiekvieno testavimo įrašo prognozės
for i, (pred, actual) in enumerate(zip(test_predictions, y_test)):
    print(f"Testavimo irasas {i + 1}: Nustatyta klase: {pred}, Tikroji klase:
{actual}")

#Stulpeline diagrama mokymosi greičiui
plt.figure(figsize=(8, 6))
bars = plt.bar([str(lr) for lr in learning_rates], test_accur_lr, color='blue')
plt.xlabel('Mokymosi Greitis')
plt.ylabel('Testavimo tikslumas')
plt.title('Testavimo tikslumas pagal mokymosi greiti')

for bar, accuracy in zip(bars, test_accur_lr):
    yval = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, yval / 2, f'{accuracy:.2f}',
ha='center', va='center', color='white')

plt.show()
```