

## კურსის სახელწოდება: AI ციფრული სახელოსნო: შექმენი შენი პირველი ჩატბოტი

აუდიტორია: X-XII კლასის მოსწავლეები (პერსონა: 17 წლის ლუკა)

ხანგრძლივობა: 14 შეხვედრა

ფორმატი: 14 პრაქტიკული დავალება (13 AnITa-ს Python-ის სიმულატორში და 1 ფინალური პროექტი Google Colab-ში).

### შეხვედრა 1: მოგზაურობა ხელოვნური ინტელექტის სამყაროში

- რა არის ხელოვნური ინტელექტი (AI)?
  - AI-ს მარტივი განმარტება: როგორ ვასწავლოთ კომპიუტერს "ფიქრი".
  - ცხოვრებისეული მაგალითები: ხელოვნური ინტელექტი შენს სმარტფონში, Google Map-ში და YouTube-ში.
  - კურსის მთავარი მიზანი: ჩვენ შევქმნით საკუთარ "მოაზროვნე" პროგრამას - ჩატბოტს.
- როგორ "ფიქრობს" კომპიუტერი?
- 2.1. განსხვავება ადამიანისა და მანქანის ინტელექტს შორის.
  - 2.2. ხელოვნური ინტელექტის ტიპები: ვინრო (ANI), ზოგადი (AGI) და სუპერინტელექტი (ASI).
- 2.3. ტიურინგის ტესტი: როდის გაუტოლდება კომპიუტერის ინტელექტი ადამიანის ინტელექტს?
4. ხელოვნური ინტელექტის გავლენა ჩვენს მომავალზე.
  - 3.1. მომავლის პროცესიები და ხელოვნური ინტელექტის უნარ-ჩვევების საჭიროება.
  - 3.2. ეთიკური დილემები: ვისი ბრალია, როცა ხელოვნური ინტელექტი შეცდომას უშვებს?
  - 3.3. AnITa-ს პლატფორმისა და Python-ის სიმულატორის გაცნობა.

### პრაქტიკული დავალება 1: ჩემი პირველი პროგრამა

- ალენერა: დაწერე Python კოდი, რომელიც ეკრანზე დაბეჭდავს მისალმებას შენი მომავალი ჩატბოტის სახელით. მაგალითად: "გამარჯობა, მე ვარ AnITa-ს ასისტენტი! მზად ვარ სწავლისთვის!".
- პლატფორმა: AnITa-ს Python-ის სიმულატორი.

### შეხვედრა 2: პირველი ნაბიჯები Python-ში – ჩვენი ჩატბოტის "მეხსიერება"

1. რატომ Python-ი?
  - 1.1. Python-ი, როგორც ადამიანურ ენასთან ყველაზე ახლოს მდგომი პროგრამირების ენა.
  - 1.2. რატომ ირჩევენ Python-ს ხელოვნურ ინტელექტთან სამუშაოდ.
2. ცვლადები და მონაცემთა ტიპები.
  - 2.1. ცვლადი, როგორც ინფორმაციის შესანახი "ყუთი" (სახელი, ასაკი, ქალაქი).
  - 2.2. ძირითადი მონაცემთა ტიპები: ტექსტი (str), მთელი რიცხვი (int), ათწილადი (float), ლოგიკური (bool).
  - 2.3. print() ფუნქცია: როგორ "ავალაპარაკოთ" ჩვენი პროგრამა.
3. მონაცემების დინამიურად გამოყენება.
  - 3.1. ფორმატირებული სტრიქონები (f-strings): როგორ ჩავსვათ ცვლადები ტექსტში.
  - 3.2. კომენტარების (#) როლი: შენიშვნები საკუთარი თავისთვის და სხვებისთვის.

## პრაქტიკული დავალება 2: პერსონალური მისალმება

- **აღნერა:** შექმენი ორი ცვლადი: `bot_name` და `user_name`. პროგრამამ უნდა გამოიყენოს ეს ცვლადები და დაბეჭდოს პერსონალური მისალმება, მაგალითად: გამარჯობა, [მომხმარებლის სახელი]! მე ვარ [ბოტის სახელი].
  - **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.
- 

## შეხვედრა 3: გადაწყვეტილებების მიღება კოდით – ვასწავლოთ ბოტს პასუხის გაცემა

1. მომხმარებლისგან ინფორმაციის მიღება.
  - 1.1. `input()` ფუნქცია: როგორ დავუსვათ მომხმარებელს კითხვა და მივიღოთ პასუხი.
  - 1.2. მიღებული ინფორმაციის შენახვა ცვლადში.
2. პირობითი ოპერატორები: `if-else`.
  - 2.1. ანალოგია: თუ ამინდი კარგია, წავალ გარეთ, სხვა შემთხვევაში - დავრჩები სახლში.
  - 2.2. შედარების ოპერატორები (`==`, `!=`, `<`, `>`): როგორ შევადაროთ მნიშვნელობები.
  - 2.3. `if`, `elif`, `else` კონსტრუქცია ლოგიკური განშტოებების შესაქმნელად.
3. მარტივი დიალოგის აგება.
  - 3.1. როგორ დავწეროთ პროგრამა, რომელიც მომხმარებლის პასუხის მიხედვით განსხვავებულად მოიქცევა.
  - 3.2. ფსევდოკოდი: ლოგიკის დაგეგმვა კოდის დაწერამდე.

## პრაქტიკული დავალება 3: "მარტივი დიალოგი"

- **აღნერა:** დაწერე პროგრამა, რომელიც მომხმარებელს ჰქონითხავს "როგორ ხარ?". თუ მომხმარებელი დაწერს "კარგად", ბოტმა უნდა უპასუხოს "მიხარია!". ყველა სხვა შემთხვევაში, ბოტმა უნდა უპასუხოს "იმედია, ყველაფერი კარგად იქნება!".
  - **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.
- 

## შეხვედრა 4: მონაცემთა სტრუქტურები (სიები) და ციკლები – გავამრავალფეროვნოთ ბოტის პასუხები

1. მონაცემთა სტრუქტურები: სიები (Lists).
  - 1.1. სია, როგორც ერთი ცვლადის ქვეშ გაერთიანებული ბევრი მნიშვნელობა.
  - 1.2. სის შექმნა, ელემენტების დამატება (append) და წაშლა (remove).
  - 1.3. ელემენტზე წვდომა ინდექსის ([0], [1]) საშუალებით.
2. მოქმედებების გამოყენება: for ციკლი.
  - 2.1. ანალოგია: როგორ დავათვალიეროთ სავაჭრო სის ყველა პროდუქტი სათითაოდ.
  - 2.2. for ციკლის გამოყენება სის ყველა ელემენტზე მოქმედების შესასრულებლად.
3. შემთხვევითობის ელემენტი.
  - 3.1. random ბიბლიოთეკის იმპორტი.
  - 3.2. random.choice() ფუნქცია სიიდან შემთხვევითი ელემენტის ამოსარჩევად.

### პრაქტიკული დავალება 4: "არაპროგნზირებადი ბოტი"

- **აღნერა:** შექმნი მისალმებების სია (მაგ. ["გამარჯობა!", "მოგესალმები!", "სალამი!"]). დაწერე კოდი, რომელიც random.choice()-ის გამოყენებით, პროგრამის ყოველი გაშვებისას შემთხვევით მისალმებას დაბეჭდავს.
  - **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.
- 

## შეხვედრა 5: მონაცემთა სტრუქტურები (ლექსიკონები) – ვემნით ბოტის ტვინს

1. მონაცემთა სტრუქტურები: ლექსიკონები (Dictionaries).
  - 1.1. ლექსიკონი, როგორც წყვილების საცავი: გასაღები-მნიშვნელობა.

- 1.2. ანალოგია: სატელეფონო წიგნაკი (სახელი - ნომერი) ან განმარტებითი ლექსიკონი (სიტყვა - განმარტება). 1.3. ლექსიკონის შექმნა, ახალი წყვილის დამატება და მნიშვნელობაზე წვდომა გასაღებით.
2. ჩატბოტის მოდელი: **მიზანი-პასუხი**
- 2.1. მიზანი (Intent): რა მიზანი აქვს მომხმარებელს, რისი თქმა სურს მომხმარებელს (მაგ. greeting, goodbye).
- 2.2. პასუხი (Response): რა პასუხი უნდა გასცეს ბოგმა მომხმარებლის შეკითხვას?
- 2.3. ლექსიკონის გამოყენება ამ მოდელის ასაგებად: {'greeting': 'გამარჯობა!', 'goodbye': 'წახვამდის!'}
3. ლექსიკონისა და if პირობის კომბინირება.
- 3.1. როგორ შევამოწმოთ, არსებობს თუ არა მომხმარებლის შეტყობინება ჩვენი ლექსიკონის გასაღებებში.
- 3.2. .get() მეთოდი უსაფრთხო პასუხის მისაღებად, თუ გასაღები არ მოიძებნა.

## პრაქტიკული დავალება 5: ლექსიკონის პრინციპით შექმნილი ბოტი

- აღნერა: შექმენი ლექსიკონი, სადაც გასაღები იქნება მომხმარებლის სავარაუდო შეტყობინება ("გამარჯობა", "როგორ ხარ", "წახვამდის"), ხოლო მნიშვნელობა - ბოტის პასუხი. დაწერე კოდი, რომელიც მომხმარებლის შეტყობინების მიხედვით შესაბამის პასუხს დაბეჭდავს ლექსიკონიდან.
- პლატფორმა: AniTa-ს Python-ის სიმულატორი.

## შეხვედრა 6: ფუნქციები: ვაქციოთ ჩვენი კოდი მრავალჯერადად გამოყენებად ინსტრუმენტად

- რა არის ფუნქცია?
  - ფუნქცია, როგორც კოდის ბლოკი, რომელიც კონკრეტულ ამოცანას ასრულებს.
  - ანალოგია: კალკულატორზე ღილაკი +, რომელიც ყოველთვის შეკრების ოპერაციას ასრულებს.
  - def საკვანძო სიტყვა ფუნქციის შესაქმნელად.
- ფუნქციის პარამეტრები და დაბრუნებული მნიშვნელობა.
  - პარამეტრი (არგუმენტი): ინფორმაცია, რომელსაც ფუნქციას გადავცემთ სამუშაოდ.
  - return საკვანძო სიტყვა: შედეგის დაბრუნება ფუნქციიდან.
- ჩატბოტის ლოგიკის ფუნქციაში მოქმედვა.
  - შევქმნათ ფუნქცია get\_response(user\_input), რომელიც მიიღებს მომხმარებლის ტექსტს.
  - ფუნქციის შიგნით განვათავსოთ ლექსიკონთან და if-else ლოგიკასთან

დაკავშირებული კოდი.

3.3. ფუნქციამ უნდა დააბრუნოს (return) ბოტის საბოლოო პასუხი.

### პრაქტიკული დავალება 6: ფუნქციონალური ბოტი

- **აღწერა:** წინა დავალების კოდი გადაკეთების შემდეგ გენერირების ლოგიკა მოექცეს ფუნქციაში `get_bot_response(message)`. პროგრამამ უნდა გამოიძახოს ეს ფუნქცია და დაბეჭდოს მის მიერ დაბრუნებული პასუხი.
- **პლატფორმა:** AnITA-ს Python-ის სიმულატორი.

---

## შეხვედრა 7: შესავალი ბუნებრივი ენის დამუშავებაში (NLP) – ვასწავლოთ ბოტს წაკითხვა

1. როგორ ესმით კომპიუტერებს ადამიანური ენა?
  - 1.1. NLP-ის (Natural Language Processing) განმარტება.
  - 1.2. ძირითადი გამოწვევა: ადამიანური ენა სავსეა ორაზროვნებითა და კონტექსტით.
2. ტექსტის მომზადება ანალიზისთვის.
  - 2.1. ტექსტის დაპატარავება (`.lower()`): "გამარჯობა" და "გაგიმარჯოთ" ერთი და იგივეა.
  - 2.2. ტოკენიზაცია: ტექსტის დაშლა სიტყვებად (ტოკენებად).
  - 2.3. არასაჭირო სიტყვების (Stop Words) და სასვენი ნიშნების მოშორება.
3. სიტყვების ფუძეების პოვნა.
  - 3.1. სტემინგი და ლემატიზაცია: რატომ არის "წავიდა", "მიდის", "წასვლა" ერთი და იგივე ძირის მქონე.
  - 3.2. როგორ გვეხმარება ეს ბოტისთვის მომხმარებლის განზრახვის უკეთ გაგებაში.

### პრაქტიკული დავალება 7: ტექსტის გამზადება

- **აღწერა:** დაწერე ფუნქცია `clean_text(text)`, რომელიც მიიღებს წინადადებას, მოაშორებს სასვენ ნიშნებს (,, ., !, ?) და დააბრუნებს გასუფთავებულ ტექსტს.
- **პლატფორმა:** AnITA-ს Python-ის სიმულატორი.

## შეხვედრა 8: შესავალი მანქანურ სწავლებაში (Machine Learning)

- რა არის მანქანური სწავლება?
  - განსხვავება ტრადიციულ პროგრამირებასა (წესების გაწერა) და ML-ს (მონაცემებიდან სწავლა) შორის.
  - ანალოგია: ბავშვი სწავლობს ცხოველების ცნობას სურათებიდან და არა მათი აღწერილობის წაკითხვით.
- მანქანური სწავლების ტიპები.
  - სწავლა მასწავლებლის დაზმარებით (Supervised Learning): სწავლა "მასწავლებელთან" ერთად (დასათაურებული მონაცემები).
  - დამოუკიდებლად სწავლა (Unsupervised Learning): ფარული კანონზომიერების (Patterns) პოვნა "მასწავლებლის" გარეშე.
  - ცდისა და შეცდომის მეთოდით სწავლა (Reinforcement Learning): სწავლა ცდისა და შეცდომის მეთოდით.
- მონაცემების როლი.
  - სავარჯიშო (Training) და სატესტო (Test) მონაცემების მნიშვნელობა.
  - "ნაგავი შედის, ნაგავი გამოდის" (Garbage In, Garbage Out) პრინციპი.
  - მონაცემთა მიკერძოების (Bias) პრობლემა და მისი საფრთხეები.

### პრაქტიკული დავალება 8: "პრობლემის იდენტიფიკაცია კოდით"

- აღნერა: შექმენი Python ლექსიკონი, სადაც გასაღები იქნება პრობლემის აღნერა (მაგ. "სახლის ფასის პროგნოზირება"), ხოლო მნიშვნელობა - მანქანური სწავლების შესაბამისი ტიპი. დაწერე კოდი, რომელიც for ციკლის გამოყენებით დაბეჭდავს თითოეულ პრობლემას და მის ტიპს.
- პლატფორმა: AniTa-ს Python-ის სიმულატორი.

## შეხვედრა 9: გედამხედველობითი სწავლება განზრახვის კლასიფიკაცია

- რა არის კლასიფიკაცია?
  - პრობლემის ფორმულირება: ობიექტის კატეგორიის (კლასის) განსაზღვრა.
  - მაგალითები: სპამის ფილტრი, სურათზე კატის ან ძაღლის ამოცნობა.
  - ჩვენი ამოცანა: მომხმარებლის შეტყობინების კლასიფიკაცია განზრახვის მიხედვით (greeting, question, goodbye).
- მონაცემების მომზადება კლასიფიკაციისთვის.
  - ნიშან-თვისებების (Features) და სამიზნე ცვლადის (Label/Target) ცნება.
  - როგორ ვაქციოთ ტექსტი რიცხვებად, რომლებიც კომპიუტერს ესმის (Bag-of-Words მოდელი).

3. პირველი მოდელის იდეა: გადაწყვეტილების ხე.
  - 3.1. როგორ მუშაობს ალგორითმი, რომელიც მონაცემების გასაყოფად ლოგიკურ "კი/არა" კითხვებს სვამს.
  - 3.2. ანალოგია: თამაში "გამოიცანი ვინ ვარ?".

### პრაქტიკული დავალება 9: "მონაცემთა ბაზის შექმნა"

- **ალენა:** შექმნი ორი სია. პირველ სიაში ჩანს ჩანს 3-4 წინადადება, ხოლო მეორეში - მათი შესაბამისი განზრახვები (მაგ. "მისალმება", "კითხვა"). დაწერე კოდი, რომელიც for ციკლის გამოყენებით ეკრანზე გამოიტანს თითოეულ წინადადებას და მის განზრახვას.
- **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.

---

### შეხვედრა 10: მონაცემთა ანალიზის ბიბლიოთეკები: Pandas

1. რა არის ბიბლიოთეკა პროგრამირებაში?
  - 1.1. მზა ინსტრუმენტების გამოყენების უპირატესობა.
  - 1.2. import pandas as pd - როგორ "გამოვიძახოთ" პროფესიონალი მონაცემთა ანალიტიკოსი.
2. Pandas ბიბლიოთეკა.
  - 2.1. DataFrame - მონაცემების წარმოდგენა ჭკვიანი ცხრილის სახით.
  - 2.2. მონაცემების ჩატვირთვა CSV ფაილიდან.
  - 2.3. მონაცემების დათვალიერება: head(), describe(), info().
3. მონაცემებთან მუშაობა.
  - 3.1. კონკრეტულ სვეტებსა და რიგებზე წვდომა.
  - 3.2. მონაცემების ფილტრაცია და საბაზისო ანალიზი.

### პრაქტიკული დავალება 10: მონაცემების დათვალიერება

- **ალენა:** გამოიყენე Pandas, რათა ჩატვირთო CSV ფაილი, რომელიც შეიცავს 2 სვეტს: text და intent. დაბეჭდე:
  1. მონაცემთა ბაზის პირველი 5 რიგი.
- 2. თითოეული განზრახვის (intent) რაოდენობა მონაცემთა ბაზაში.
- **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.

## შეხვედრა 11: პირველი ML მოდელის აგება: Scikit-learn

1. **Scikit-learn** ბიბლიოთეკა.
  - 1.1. მრავალფუქციური ინსტრუმენტი მანქანური სწავლებისთვის Python-ში.
  - 1.2. მოდელის შექმნის 4 ძირითადი ეტაპი: იმპორტი, ინიციალიზაცია, ვარჯიში, პროგნოზირება.
2. ტექსტის ვექტორიზაცია.
  - 2.1. **CountVectorizer**: ტექსტების გადაქცევა რიცხვების ვექტორებად.
  - 2.2. როგორ იქმნება "ყველა სიტყვის ლექსიკონი" და როგორ ხდება სიტყვების სიხშირის გამოთვლა.
3. მოდელის აგება და ვარჯიში.
  - 3.1. ლოგისტიკური რეგრესიის (Logistic Regression) მოდელის გამოყენება კლასიფიკაციისთვის.
  - 3.2. `model.fit(X_train, y_train)`: მოდელის დატრენინგების პროცესი.
  - 3.3. `model.predict(X_test)`: ახალი მონაცემებისთვის პროგნოზის გაკეთება.

## პრაქტიკული დავალება 11: "განზრახვის პროგნოზირება"

- **აღნერა:** წინა დავალების მონაცემებზე (`text, intent`) დაყრდნობით, `scikit-learn`-ის გამოყენებით, გაავარჯიშე `CountVectorizer` და `LogisticRegression` მოდელი. გამოიყენე განზრობილი მოდელი, რათა იწინასწარმეტყველო ახალი წინადადების ("დღეს კარგი ამინდია") განზრახვა.
- **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.=

## შეხვედრა 12: შესავალი ნეირონულ ქსელებსა და ღრმა სწავლებაში (Deep Learning)

1. რა არის ნეირონული ქსელი?
  - 1.1. ადამიანის ტვინის სტრუქტურით შეთავონებული მოდელი.
  - 1.2. ანალოგია: კომპანია, სადაც სხვადასხვა დეპარტამენტი (ფენა) და თანამშრომელი (ნეირონი) ერთად მუშაობენ რთული პრობლემის გადასაჭრელად.
  - 1.3. ფენების, ნეირონებისა და კავშირების კონცეფცია.
2. რა არის ღრმა სწავლება (Deep Learning)?
  - 2.1. მრავალფუქციური ღრმა ნეირონული ქსელები.
  - 2.2. რატომ არის Deep Learning განსაკუთრებით ძლიერი ენასთან და სურათებთან მუშაობისას.
3. დიდი ენობრივი მოდელები (LLMs).
  - 3.1. რა არის Transformer არქიტექტურა და მისი გარდამტები როლი

ხელოვნური ინტელექტის ეპოლუციაში.

3.2. როგორ მუშაობენ ChatGPT, Gemini და სხვა თანამედროვე მოდელები.

3.3. Fine-tuning კონცეფცია: როგორ ვასწავლოთ გენიოსს ახალი, კონკრეტული საგანი.

### პრაქტიკული დავალება 12: ნეირონული ქსელის აღწერა კოდით

- **აღწერა:** შექმენი დექსიკონი, რომელიც სქემატურად აღწერს ნეირონულ ქსელს (მაგ. `{'input_layer': 784, 'hidden_layer_1': 128, 'output_layer': 10}`). დაწერე კოდი, რომელიც ამ დექსიკონის მიხედვით დაბეჭდავს ქსელის სტრუქტურის აღწერას.
- **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.

---

### შეხვედრა 13: ფინალური პროექტისთვის მზადება: Google Colab

#### 1. რა არის Google Colaboratory (Colab)?

1.1. უფასო Jupyter Notebook გარემო ღრუბელში, Google-ის სერვერებზე.

1.2. რატომ არის ის იდეალური ხელსაწყო მანქანური სწავლების პროექტებისთვის.

1.3. Colab-ის ინტერფეისის გაცნობა: ტექსტური და კოდის უჯრები.

#### 2. Colab-ში მუშაობის საფუძვლები.

2.1. როგორ გავხსნათ, შევინახოთ და გავაზიაროთ პროექტი.

2.2. ბიბლიოთეკების ინსტალაცია `!pip install` ბრძანებით.

2.3. ფაილების ატვირთვა და მათან მუშაობა.

#### 3. ფინალური პროექტის მიმოხილვა.

3.1. პროექტის მიზანი: შევქმნათ მარტივი ჩატბოტი, რომელიც უპასუხებს კითხვებს AniTa-ს კურსის შესახებ.

3.2. მონაცემთა ბაზის მომზადება: კითხვა-პასუხის წყვილების შექმნა.

3.3. პროექტის შესრულების ეტაპების განხილვა.

### პრაქტიკული დავალება 13: გავარჯიშება სიმულატორში

- **აღწერა:** AniTa-ს სიმულატორში, `pandas` ბიბლიოთეკის გამოყენებით შექმენი მარტივი DataFrame-ი, რომელიც შეიცავს 2 სვეტს (კითხვა, პასუხი) და 3 რიგს. დაბეჭდე ეს DataFrame-ი. ეს დაგეხმარება ფინალური პროექტის მონაცემთა ბაზის სტრუქტურის გააზრებაში.
- **პლატფორმა:** AniTa-ს Python-ის სიმულატორი.

---

## შეხვედრა 14: მოდულის შეჯამება და ფინალური პროექტი

1. რა ვისწავლეთ ამ კურსში?
  - 1.1. გავიარეთ გზა `print("Hello World")`-დან მანქანური სწავლების მოდელის აგებამდე.
  - 1.2. Python-ის საფუძვლები, NLP-ის კონცეფციები და ML-ის პრაქტიკული გამოყენება.
  - 1.3. როგორ ვაქციოთ იდეა მოქმედ პროგრამად.
2. ეთიკა და ხელოვნური ინტელექტი (შეჯამება).
  - 2.1. მონაცემთა კონფიდენციალურობა და ალგორითმული მიკერძოება.
  - 2.2. AI-ს პასუხისმგებლიანი გამოყენების მნიშვნელობა.
3. ფინალურ პროექტზე მუშაობის დაწყება.
  - 3.1. კითხვა-პასუხის სესია და პროექტთან დაკავშირებული საკითხების გარკვევა.
  - 3.2. დამოუკიდებელი მუშაობის დაწყება მენტორის დახმარებით.

### პრაქტიკული დავალება 14 (ფინალური პროექტი): "AnITa-ს ასისტენტი"

- **აღნერა:** Google Colab-ში გამოიყენე კურსის განმავლობაში მიღებული ცოდნა და შექმნი მარტივი ჩატბოტი.
  1. **მონაცემები:** შექმნი მცირე ზომის კითხვა-პასუხის ბაზა AnITa-ს კურსის შესახებ (მაგ. "რამდენი შეხვედრაა კურსში?", "რას ნიშნავს NLP?").
  2. **მოდელი:** გამოიყენე `scikit-learn`-ი, რათა გაავარჯიშო კლასიფიკატორი, რომელიც მომხმარებლის კითხვის მიხედვით ამოიცნობს სწორ პასუხს (შეგიძლია გამოიყენო კოსინუსური მსგავსება ან სხვა მარტივი მეთოდი).
  3. **ინტერფეისი:** შექმნი ციკლი, რომელიც მომხმარებელს საშუალებას მისცემს, მუდმივად დაუსვას კითხვები ბოტს და მიიღოს მისგან პასუხები.
- **პლატფორმა:** Google Colab (პროექტის ბმული უნდა აიტვირთოს AnITa-ს პლატფორმაზე).