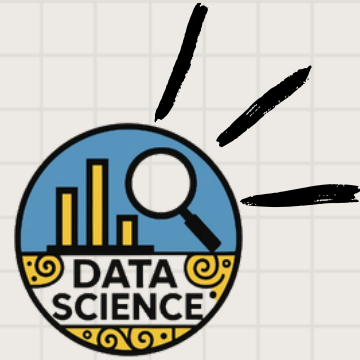


სერტიფიცირებადი PYTHON + შესავალი მონაცემთა მეცნიერებაში



შესავალი პითონისა და პროგრამირების საფუძვლებში, მონაცემთა ტიპები და ოპერატორები

- პითონის საფუძვლები: კომპილაცია **vs.** ინტერპრეტაცია, საკვანძო სიტყვები, ინდენტაცია, კომენტარები
- ლიტერალები და საბაზისო I/O (`print()`, `input()`).
- ცვლადები, დასახელების კონვენციები და მონაცემთა ტიპები (მთელი რიცხვები, **float**, სტრიქონები, ლოგიკური მნიშვნელობები).
- რიცხვითი ტიპები (**int**, **float**) და ოპერაციები (არიტმეტიკული, შედარების, ლოგიკური).
- სტრიქონის ოპერაციები (კონკატენაცია, მეთოდები).
- ტიპების გარდაქმნა (`int()`, `str()`, `float()`).

1

პირობის ოპერატორები

პროგრეს ტესტი წინა ლექციიდან

- **if, elif, else, match - case** ოპერატორები.
- **Logic gate tables (and, or, not);**
- შედარების ოპერატორები (`==`, `!=`, `<`, `>`, `<=`, `>=`);

პროექტი: მარტივი კალკულატორი

2

პირობის ოპერატორები და ციკლები

პროგრეს ტესტი წინა ლექციიდან

- **for** და **while** ციკლები.
- უპირატესობები და განსხვავებები
- **break** და **continue** ოპერატორები.
- ჩაშენებული ციკლები.

პროექტი 1: რიცხვის გამოცნობის თამაში
პროექტი 2: ASCII ხელოვნება

3

4

მონაცემთა სტრუქტურები (კოლექციები)

პროგრეს ტესტი წინა ლექციიდან

- **mutable** და **immutable** მონაცემთა ტიპები.
- სიები (**list**): შექმნა, ინდექსები, მეთოდები (`append()`, `remove()`, `sort()`).
- კორტეჟი (**tuple**).
- **List** - ისა და **tuple**-ის ერთმანეთთან შედარება. ძლიერი და სუსტი მხარეები

პროექტი: საყიდლების სიის მენეჯერი

5

მონაცემთა სტრუქტურები (კოლექციები)

პროგრეს ტესტი წინა ლექციიდან

- ლექსიკონები (**dictionary**): გასაღები-მნიშვნელობა, მეთოდები (**get()**, **keys()**, **values()**).
- სიმრავლეები (**sets**). მათემატიკური ოპერაციები
- **Dictionary**-სა და **set**-ის შედარება.

პროექტი 1: განმარტებითი ლექსიკონის შექმნა

პროექტი 2: უნიკალური ასოების მთვლელობა

6

ფუნქციები

პროგრეს ტესტი წინა ლექციიდან

- ჩაშენებული ფუნქციები (**len()**, **split()**, **join()** და ა.შ.).
- ფუნქციების განსაზღვრა, გამოძახება, პარამეტრები და არგუმენტები (***args**, ****kwargs**)
- **Return** -ისა და **yield**-ის შესახებ
- ლოკალური **vs.** გლობალური ცვლადები
- ანონიმური ფუნქციები.
- კასკადური ფუნქციები
- რეკურსიის შესავალი.

პროექტი: ფაქტორიალის, რიცხვების ჯამისა და საშუალოს გამოთვლის ფუნქციები

7

8

ფაილებთან მუშაობა

პროგრეს ტესტი წინა ლექციიდან

- ფაილის წაკითხვა და ჩაწერა (**open()**, **read()**, **write()**, **close()**).
- **with** კონტექსტის მენეჯერი.
- **OS** (ოპერაციული სისტემის) მოდულთან მუშაობა

9

პროგრამის შეცდომებთან მუშაობა და გენერატორები

პროგრეს ტესტი წინა ლექციიდან

- **Try... Except ...** კონსტრუქცია (**try**, **except**, **else**, **finally**)
- პროგრამის შეცდომების ტიპები (**ValueError**, **TypeError**, **IndexError**).
- შეცდომის აღძვრა (**raise**).
- გენერატორები
- კლასიკური მიდგომა **vs** გენერატორები (სისწრაფის შედარება)

პროექტი: ფაილის წაკითხვა, მასში არსებული სიტყვების, სტრიქონებისა და სიმბოლოების დათვლა შეცდომების გარეშე

10

ობიექტზე ორიენტირებული პროგრამირება (OOP)

პროგრეს ტესტი წინა ლექციიდან

- ობიექტზე ორიენტირებული პროგრამირების არსი
- კლასები, ობიექტები, ატრიბუტები, მეთოდები, ინსტანცია

11

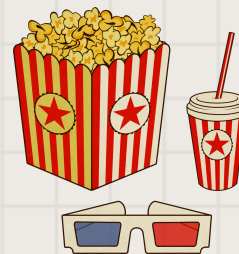
ობიექტზე ორიენტირებული პროგრამირება (OOP)

პროგრეს ტესტი წინა ლექციიდან

- მემკვიდრეობა,
- ინკაფსულაცია,
- პოლიმორფიზმი.
- ინტერფეისები
- მეთოდების გადატვირთვა

პროექტი: ბანკის ანგარიშის მართვა

12



13

შუალედური გამოცდა

Pandas ინსტრუმენტი

პროგრეს ტესტი წინა ლექციიდან

- მონაცემთა ჩატვირთვის და ამოკითხვის მექანიზმები
- **Pandas** მოდულის შესახებ
- სვეტების შერჩევა, დამატება და ამოშლა
- ფილტრაცია და შეტყობინებები
- მონაცემებში გამოტოვებების დამუშავება
- ცვლადების ტიპები
- ფილტრაცია ქვანტილების მიხედვით
- **Pivot table** და **groupby** ინსტრუმენტების გამოყენება
- **Heatmap**-ის მნიშვნელობა
- კორელაცია და **"phik"** ინსტრუმენტი
- კავშირების შემოწმების მექანიზმი
- "სათამაშო" მონაცემები
- აგრეგაციის ფუნქციონალი და გამოტოვებები
- მონაცემების ვიზუალიზაცია
- **Scatter plot** - განაწილების დიაგრამა
- **Join**-ების შესახებ
- **NumPy** მოდულის მიმოხილვა

დავალება

14



15

გამოყენებითი "ადამიანური განსწავლა" რეალურ მონაცემებზე

პროგრეს ტესტი წინა ლექციიდან

- სამუშაო მონაცემთა ჩატვირთვა
- "ადამიანური განსწავლის" შესახებ
- სასწავლო და სატესტო მონაცემები (**Train and Test**)
- მონაცემთა ანალიზი
- **Baseline**-ს ცნება
- საშუალო წლიური მაჩვენებლის გამთვლელი ალგორითმი
- წლიური მედიანის გამომთვლელი ალგორითმი
- მოდელში მონაცემების საჭირო პარამეტრების დამატება
- მოდელის თანდათანობით გაუმჯობესება

დავალება

გამოყენებითი “მანქანური განსწავლა” რეალურ მონაცემებზე

პროგრეს ტესტი წინა ლექციიდან

- **Catboost** პაკეტის შესახებ
- სამუშაო მონაცემების ჩატვირთვა
- სავალიდაციო მონაცემები
- “**Feature**”-ს ნუსხა
- კატეგორიზირებული “**Feature**”-ბის გარდაქმნის პრინციპები
- **Catboost** ინსტრუმენტის პირველი გაშვება
- განსწავლა ყველა “**Feature**”-ის ბაზაზე
- **Random seed** ინსტრუმენტის შესახებ
- **Learning rate** პარამეტრის მნიშვნელობა
- განსწავლა მთელი მონაცემების ბაზაზე
- რეგრესიის მეტრიკების შესახებ
- დანაკარგის ფუნქციის ცნება

დავალება

მოდელის ანალიზის ეტაპი

პროგრეს ტესტი წინა ლექციიდან

- შეცდომების ანალიზი
- “აბსოლუტური შეცდომის” ანალიზი
- “აბსოლუტური შეცდომის” ანალიზი სხვადასხვა პარამეტრის მიმართებით
- “**Feature**”-ს მნიშვნელობა
- მოდელების “ახსნის” მეთოდი - **SHAP**
- ტოპ **10** ხშირი შეცდომის გაანალიზება
- რამდენიმე ობიექტის ერთიანად ვიზუალიზაცია
- **Dependance** და **Summary plot** ინსტრუმენტები

დავალება

კლასიფიკაციის მეტრიკები

პროგრეს ტესტი წინა ლექციიდან

- **Score** და საზღვრები
- შეცდომების მატრიცა
- **Recall**, **Precision** მეტრიკები
- **Recall** და **Precision** მეტრიკების ინტუიციური აღქმა
- **F1** მაჩვენებელი
- **Accuracy** მეტრიკა
- კლასიფიკაციის **report**-ი
- განსხვავებული საზღვრის მქონე **predict**-ი
- მაქსიმალური **F1**
- **ROC** მრუდის შესახებ, გრაფიკის აგება
- **ROC AUC** მეტრიკა
- **ROC** დაუბალანსებელ მონაცემებზე
- **Precision recall** მრუდი
- ადამიანური და მანქანური “განსწავლის” დადარება
- კლასების დისბალანსი
- მრავალკლასიანი კლასიფიკაცია

დავალება

კლასიფიკაციის ამოცანა:

პროგრეს ტესტი წინა ლექციიდან

- სამუშაო მონაცემების ჩატვირთვა და ანალიზი
- მონაცემების სტრატეფიკაცია: სავარჯიშო (**train**), სატესტო (**test**) და სავალიდაციო (**validation**) ჯგუფებად
- **Phik**-ის ნახვა
- ადამიანური “განსწავლის” ეტაპი
- **Baseline**-ს განსაზღვრა
- **Uplift**-ით გარკვეული ინფორმაციის მიღება
- მონაცემებში არსებული “ხმაურის” შესახებ
- ფუნქციის დაორგანიზება **uplift**-ისთვის
- მოდელში სხვადასხვა საჭირო პარამეტრის ჩამატება
- მანქანური განსწავლის ეტაპი
- მოდელის განსწავლა ყველა მახასიათებლის (**Features**) ბაზაზე
- მახასიათებლების (**Features**) კრიტიკული მნიშვნელობა მოდელისთვის
- **SHAP** მეთოდის გამოყენება
- კლასიფიკაციის და რეგრესიის გააზრება

დავალება

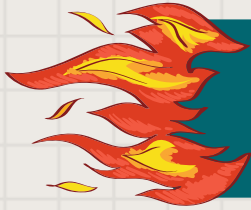
დამატებითი გამოყენებითი ილეთები

პროგრეს ტესტი წინა ლექციიდან

- “კროს-ვალიდაცია”
- **Grid Search** მეთოდი
- **Sklearn** პრინციპები
- წინადამუშავება და
მახასიათებლების ინჟინერია
- **feature engineering**
- კალიბრაცია
- ტექსტური მახასიათებლები
- ანალიტიკოსების და მონაცემთა
მეცნიერების შესახებ

20

21



ფინალური პროექტი