

Унапређење тренинга и тестирања модела вештачке интелигенције кроз валидацију података помоћу Python библиотеке Pydantic

Improving training and testing of artificial intelligence models through data validation using the Python library Pydantic

Лука Илић¹, Александар Шијан², Дејан Видука³

¹ Универзитет у Нишу, Електронски факултет, Ниш, Србија, luka.ilic@mef.edu.rs

² Универзитет у Нишу, Електронски факултет, Ниш, Србија, aleksandar@mef.edu.rs

³ Универзитет Привредна академија у Новом Сад, Факултет за примењени менаџмент, економију и финансије, Београд, Србија, dejan.viduka@mef.edu.rs

Апстракт: Квалитетни подаци су веома битан фактор за постизање одређених циљева. Тако и код вештачке интелигенције, тј. машинског учења, квалитет улазних података игра битну улогу у поузданости и тачности података који ће бити употребљени за даљу обраду и анализу. Постоје разни начини за валидацију како улазних тако и излазних података, један од начина је да сами креирамо тестове улазних и излазних података, а други начин је да користимо готова решења. Овај рад има за задатак да покаже једно готово решење које је заправо Python библиотека Pydantic која служи за валидацију података како не би дошло до употребе података који нису тачни (пример, 18 и „18“ нису два иста податка).

Кључне речи: податак, тачност, вештачка интелигенција, машинско учење, Python, Pydantic

Abstract: Quality data is a very important factor for achieving certain goals. Likewise with artificial intelligence, i.e. machine learning, the quality of the input data plays an important role in the reliability and accuracy of the data that will be used for further processing and analysis. There are various ways to validate both input and output data, one way is to create input and output tests ourselves, and another way is to use ready-made solutions. This paper has the task of showing a ready-made solution, which is actually the Python library Pydantic, which serves to validate data so that incorrect data is not used (for example, 18 and "18" are not the same data).

Keywords: data, accuracy, artificial intelligence, machine learning, Python, Pydantic

Увод

Вештачка интелигенција постоји као идеја и концепт на коме се ради већ дужи низ година, али је тек недавно почела да добија на снази. Представља област рачунарских наука који се убрзано развија и примењује у различитим сферама људског живота, аутомобилска индустрија, медицина, економија, обрада језика итд. У основи се налазе модели машинског учења (eng. Machine learning - ML) који су претходно обучени на великим количинама података. Најважнији фактор за постизање успешног обучавања ових модела јесте висок квалитет улазних података. Квалитет података игра веома битну улогу у поузданости и тачности модела вештачке интелигенције. Лоши или невалидни подаци могу довести до нетачних закључака и неефикасних модела. С обзиром на све ово, поставља се питање како унапредити процес припреме, обуке и тестирања модела како би се осигурала тачност улазних података.

Овај научни рад се бави управо овим изазовима и истражује могућности за унапређење процеса обуке и тестирања модела у области вештачке интелигенције. У раду ћемо се фокусирати на употребу *Pydantic* библиотеке, који је популаран алат за валидацију података у *Python* програмском језику и његову примену у програмима написаним у *Python* програмском језику који служе за рад са вештачком интелигенцијом, тј. машинским учењем. Поред *Pydantic* библиотеке постоје и друге библиотеке (*mypy*, *typing*) али смо се определили за *Pydantic* због његовог једноставног коришћења што је заправо и предност *Python* програмског језика. *Pydantic* омогућава дефинисање структуре података и типова података, што може на одређени начин олакшати проверу и припрему улазних података за моделе машинског учења (Lathkar, 2023).

Задатак овог рада је приказ како *Pydantic* библиотека може да побољша квалитет података у машинском учењу, врши аутоматску проверу улазних података и на тај начин унапређује тачност модела. Такође, погледаћемо пример из праксе интегрисања *Pydantic* у процесе обуке и тестирања у вештачкој интелигенцији. На крају, анализирамо предности и ограничења ове интеграције.

Pydantic: Алат за валидацију података у Python програмском језику

Pydantic је *Python* библиотека за рашчлањивање и валидацију података. Користи механизам за наговештавање типова новијих верзија *Python* програмског језика (од верзије 3.6) и проверава типове током извршавања програма.

A screenshot of a code editor with a dark background and light-colored text. The code defines a Pydantic model named 'User' and demonstrates its use with a dictionary of input data. The code includes comments in Croatian. The code is as follows:

```
from pydantic import BaseModel

class User(BaseModel):
    username: str
    email: str

input_data = {
    "username": "john_doe",
    "email": "johndoe@example.com"
}

try:
    user = User(**input_data)
    print("Validni podaci:")
    print("Korisničko ime:", user.username)
    print("Email adresa:", user.email)
except Exception as e:
    print("Greška pri validaciji:", e)
```

Слика 1. Пример кода за *Pydantic*

Pydantic је врло користан алат за проверу података у *Python* програмском језику, пружајући програмерима моћан механизам за осигуравање тачности и исправности података у њиховим апликацијама. Основна функционалност *Pydantic* је дефинисање модела који прецизно описују структуру очекиваних података. Ови модели садрже информације о типовима података, обавезним или опционалним вредностима и правилима валидације, чиме олакшавају прецизну спецификацију очекивања. Ову библиотеку издваја способност аутоматског генерисања *Python* кода за валидацију на основу дефинисаних модела. Овај аутоматски приступ смањује потребу за ручним писањем програма који ће да проверавају типове података које програмери користе, чиме програмерима штеди време и смањује ризик од грешака јер су све библиотеке које су

доступне претходно добро тестиране. Поред основне провере података, *Pydantic* пружа подршку за парсирање и претварање података у одговарајуће *Python* објекте, олакшавајући манипулацију и обраду података након што су прошли кроз процес провера података, што чини поступак писања и тестирања кода бржим.

Важност квалитета података у AI

Квалитет података има незамењиву улогу у области вештачке интелигенције, дубоког учења (*deep learning*) и машинског учења (*machine learning*). Прецизност и поузданост података су кључни фактори у обезбеђивању успешног функционисања модела и алгоритама у AI. Лоши или неквалитетни подаци могу озбиљно нарушити перформансе модела, укључујући смањење тачности и способност генерализације. Овај проблем постаје нарочито видљив у дубоком учењу, где су модели често комплексни и врло осетљиви на квалитет улазних података. Квалитет података обухвата различите аспекте, укључујући тачност, потпуност, доследност, релевантност и актуелност података. Тачност података се односи на прецизност вредности које су заступљене, док потпуност података подразумева да сви потребни подаци постоје. Доследност се односи на то да подаци буду конзистентни и без конфликта, док релевантност значи да су подаци релевантни за одређени проблем или задатке AI (Elmore & Lee, 2021; Jain *et. al.*, 2020). Унапређење квалитета података постаје кључан задатак у контексту вештачке интелигенције, јер омогућава већу поузданост у процесима одлучивања и генерализацији на нове податке.

У свету свеprisутних података, квалитет података постаје кључни фактор у области вештачке интелигенције. Квалитетни подаци су као чврст темељ на којем почива изградња и функционисање различитих AI система, укључујући системе за машинско учење и дубоко учење. Шта тачно значи "квалитет података"? То се односи на тачност и поузданост података. Квалитетни подаци морају бити тачни и актуелни, без грешака и неслагања. Другим речима, морамо бити сигурни да подаци које користимо за тренирање и доношење одлука за AI системе су прецизни и поуздани. Лоши подаци могу довести до низа проблема (Chu *et. al.*, 2016). Модели вештачке интелигенције могу постати непоуздани, непрецизни и склони прерачунима са неквалитетним подацима. То значи да ће одлуке које доносе бити мање тачне и мање корисне. Унапређење квалитета података постаје кључно, јер то значи да можете осигурати да наши AI системи раде са што прецизнијим и поузданијим информацијама. Кроз процесе валидације, чишћења и обогаћивања података, можете постићи већу прецизност у моделима и боље резултате (Tae *et. al.*, 2019).

Интеграција *Pydantic* у процес обуке модела

У оквиру унапређења квалитета података у области вештачке интелигенције, интеграција *Pydantic* у процес обуке модела представља кључан корак. *Pydantic*, као моћна библиотека за валидацију података у *Python*, доноси значајне предности кроз строга правила за очекиване улазне податке. Примењујући *Pydantic*, програмери осигуравају да улазни подаци у модел буду тачни, доследни и релевантни за конкретни проблем. На пример, у контексту дубоког учења, *Pydantic* може бити употребљен за проверу карактеристика слика, као што су димензије и формат, пре него што се користе за обуку модела. Ова пракса помаже у спречавању неприкладних или неквалитетних података да утичу на перформансе модела. Интеграција *Pydantic* може се одвијати и током процеса валидације података током обуке модела. То значи да се подаци који улазе у модел прво проверавају и валидирају пре него што буду употребљени за тренирање. *Pydantic* може генерисати одговарајуће грешке или упозорења уколико се уоче неправилности или неквалитетни подаци, што омогућава програмерима да благовремено реагују и исправе потенцијалне проблеме. Кроз овакву интеграцију, програмери осигуравају да модели за вештачку интелигенцију буду обучени на квалитетним и поузданим подацима, што доноси повећану тачност и способност генерализације модела. Осим тога, *Pydantic* омогућава

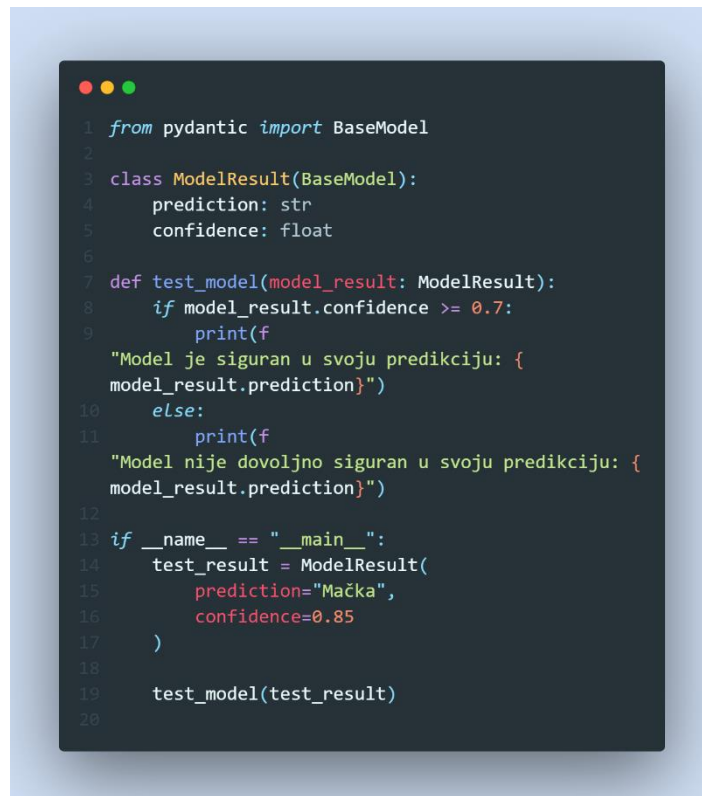
бољу документацију и разумевање улазних података, чиме се олакшава сарадња између различитих тимова и програмера у процесу обуке модела (Peralta, 2023).



Слика 2. Имплементација процеса обуке модела

Примена *Pydantic* у тестирању модела

Примена *Pydantic* у процесу тестирања модела пружа драгоцене алате за осигуравање тачности и поузданости AI система. Осим што се користи за валидацију улазних података током обуке модела, *Pydantic* може бити искоришћен и за тестирање модела након обуке. Приликом тестирања модела, *Pydantic* се може користити за детаљну проверу излазних резултата модела. На пример, ако је модел развијен за класификацију слика, *Pydantic* може аутоматски проверити тачност и доследност класификација, упоређујући их са очекиваним вредностима. Ово обезбеђује темељно тестирање и потврду да модел ради онако како се очекује. *Pydantic* такође омогућава проверу перформанси модела у „стварном свету“. Може се користити за валидацију података које модел прима у реалном времену током оперативног рада, идентификујући евентуалне проблеме у стварним, реалним, подацима. Овај приступ омогућава брзо реаговање на промене у околини и осигурава да модел остане поуздан и тачан.



Слика 3. Имплементација процеса тестирања модела

Предности и ограничења *Pydantic*

Пре коришћења алата *Pydantic* (и друге сличне алате), морамо знати за предности и ограничења алата. У наставку су наведене неке од предности и ограничења коришћења ове библиотеке:

Предности:

- Једноставна употреба
- Јака валидација података
- Аутоматска документација
- Одржавање кода

Ограничења:

- Перформансе
- Недостатак потпоре за нове типове података
- Мањак флексибилности

Без обзира на приказана ограничењима, *Pydantic* је и даље моћана библиотека (алат) за проверу података у *Python* програмском језику, често се користи за осигуравање тачности и поузданости података у пројектима који захтевају овакав вид рада. Програмери пре коришћења библиотека за проверу типова података (и њихову валидацију) требају пажљиво размотрити предности и ограничења које *Pydantic* библиотека има како би донели исправне одлуке о његовој примени у својим пројектима. Поред *Pydantic* постоје и друга решења (Viafore 2021).

Закључак

Када је реч о подацима, веома је битно да ти подаци буду тачни, прецизни и непромењени како би гарантовали сигурност, поузданост и тачност излазних вредности. Квалитетни подаци чине

основу за обуку модела који се користе за машинско и дубоко учење и утичу на понашање и перформансе алгоритама. *Python* библиотека за програмирање *Pydantic* гарантује тачну и поуздану валидацију података има бројне предности, као што су једноставност за корисника, поуздана валидација, аутоматска документација и лак надзор кода. Међутим, треба имати на уму да *Pydantic* има одређене недостатке, који укључују могући утицај на перформансе и нефлексибилност. Ипак, *Pydantic* је и даље ефикасан алат за валидацију података за програмере, не само у пројектима вештачке интелигенције већ иу другим подухватима. С обзиром на то, програмери морају пажљиво испитати како *Pydantic* одговара специфичним захтевима њиховог пројекта и израчунати да ли су његове предности вредне ограничења.

Референце

- Chu X., Ilyas I., Krishnan S., Wang J., (2016). Data Cleaning: Overview and Emerging Challenges, SIGMOD 16, 2201–2206.
- Elmore J., Lee C., (2021). Data Quality, Data Sharing, and Moving Artificial Intelligence Forward, JAMA Netw Open. 2021
- https://docs.guardrailsai.com/defining_guards/pydantic/ (13.11.2023)
- <https://docs.pydantic.dev/latest/> (7.11.2023)
- <https://dzone.com/articles/what-is-pydantic-1> (13.11.2023)
- <https://medium.com/@bobbymystery/pydantic-the-package-for-data-validation-and-modeling-like-a-pro-90169ae875ea> (13.11.2023)
- <https://medium.com/@jshreyansh606/pydantic-usage-and-its-applications-7c2b1dc63fe5> (14.11.2023)
- <https://medium.com/mllearning-ai/improve-your-data-models-with-pydantic-f9f10ca66f26> (13.11.2023)
- <https://wandb.ai/jxnlco/function-calls/reports/Better-Data-Extraction-Using-Pydantic-and-OpenAI-Function-Calls--Vm1ldzo0ODU4OTA3> (13.11.2023)
- https://www.tutorialspoint.com/fastapi/fastapi_pydantic.htm (7.11.2023)
- Jain A., Patel H., Nagalapatti L., Gupta N., Mehta S., Guttula S., Mujumdar S., Afzal S., Mittal R., Munigala V., (2020). Overview and Importance of Data Quality for Machine Learning Tasks, KDD 20, 3561–3562.
- Lathkar M., (2023). High-Performance Web Apps with FastAPI, Apress, 65-92.
- Peralta J. H., (2023). Microservice APIs: Using Python, Flask, FastAPI, OpenAPI and more, Manning publication Co, 34-40.
- Tae K., Roh Y., Oh Y., Kim H., Whang S., (2019). Data Cleaning for Accurate, Fair, and Robust Models: A Big Data - AI Integration Approach, DEEM 19, 1–4.
- Viafore P., (2021). Robust Python, O'Reilly Media, Inc., 199-210.