Icon

Description automatically generatedINSTITUT ZA MATEMATIKU I INFORMATIKU

PRIRODNO-MATEMATIČKI FAKULTET

UNIVERZITET U KRAGUJEVCU

UPUTSTVO ZA PYTHON I MACHINE LEARNING

**TIM: NEURON HORIZON**

|  |  |
| --- | --- |
| **Dokumentaciju kreirali:** | **Mentori:** |
| Jefimija Stamenović 81/2019 | dr Boban Stojanović, predmetni profesor |
| Milutin Aleksić 31/2018 | Lazar Krstić, asistent |
|  | Andreja Živić, asistent |
|  | Filip Bojović, asistent |
|  |  |
|  |  |

Kragujevac, 2022. Godina

Verzije dokumenta:

|  |  |  |  |
| --- | --- | --- | --- |
| **Broj verzije:** | **Datum:** | **Autor:** | **Opis:** |
| 1.0 | 26.02.2022. | Jefimija Stamenović i Milutin Aleksić | Kreiran dokument |
| 1.1. | 28.02.2022. | Jefimija Stamenović | Izmene |
| 1.2. | 01.03.2022. | Jefimija Stamenović | Izmene |

Sadržaj

[1. Uvod 4](#_Toc97123579)

[2. Radno okruženje i kreiranje Python skripte 4](#_Toc97123580)

[3. Neuronske mreže u Pythonu 6](#_Toc97123581)

# Uvod

U ovom dokumentu je dato objašnjenje instalacije neophodnih programa i ekstenzija i kako se kreira jedna neuronska mreža.

# Radno okruženje i kreiranje Python skripte

Kao pogodno radno okruženje za mašinsko učenje, koristićemo *Visual Studio Code (VSC).* Programski jezik koji ćemo koristiti je Python. Prvo ćemo instalirati poslednju verziju *VSC-a* koju možete preuzeti na sledećem [linku](https://code.visualstudio.com/" \l "alt-downloads). Kada instalirate *VSC-a* potrebno je da instaliramo *Python ekstenziju* kako bismo mogli da kucamo kod u Python-u (slika ispod).

A screenshot of a computer

Description automatically generated with medium confidence

Nakon što smo instalirali ekstenziju, možemo da nastavimo dalje i da kreiramo jednu *Python skriptu*.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Za rad sa neuronskim mrežama, potrebno je da instaliramo biblioteke. Postupak za instalaciju biblioteka je univerzalan. Potrebno je da otvorimo terminal i u okviru njega da napišemo komandu ***pip install naziv\_biblioteke.***

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

# Kreiranje neuronske mreže

Kreiranje neuronskih mreža u Pythonu možemo da podelimo u nekoliko koraka:

1. ***Uključivanje biblioteka*** – ovo se odnosi na biblioteke *numpy, pandas, sklearn, mathplotlib, tensorflow* itd. Ukoliko ih nemate instalirane pogledajte poglavlje br. 2.
2. ***Definisanje ulaznih i izlaznih podataka*** – da bismo kreirali mrežu, potrebni su nam ulazi i izlazi, odnosno data frame koji će da sadrži te podatke. Rad sa data frame-om nam omogućava biblioteka *numpy.*
3. ***Definisanje težina i bias-a -* t**ežine se odnose na težinu ulaza koji utiču na izlaz mreže, a bias se odnosi na dodatnu težinu praga koja se dodaje izlazu
4. ***Treniranje mreže* –** određujemo trenažni skup podataka koji će nam služiti za treniranje modela I vršimo treniranje mreže
5. ***Testiranje mreže –*** određujemo testni skup podataka koji će nam služiti za validaciju modela, a potom testiramo mrežu
6. ***Predviđanje modela sa dodatnim parametrima čije se vrednosti koriste za kontrolu procesa učenja*** – na osnovu ovog koraka možemo utvrditi tačnost modela tako što ćemo vrednosti modela uporediti sa vrednostima testnog skupa

# Primer

Graphical user interface, text

Description automatically generatedDemonstriraćemo kreiranje jedne neuronske mreže. Da bismo uopšte mogli da radimo sa neuronskim mrežama i da se bavimo mašinskim učenjem, potrebno je da uvezemo sledeće biblioteke : ***pandas, numpy*** i ***tensorflow***.

Manipulaciju podacima nam obezbeđuje biblioteka *pandas, numpy* obezbeđuje funkcije za pripremu podataka za neuronsku mrežu, dok nam biblioteka *tensorflow* obezbeđuje metode za rad sa neuronskim mrežama.

Za svaku neuronsku mrežu su nam potrebni podaci od kojih ćemo formirati *data frame*. Kako bismo videli prvih nekoliko redova u data frame-u, koristićemo funkciju *head().*

Table

Description automatically generated with medium confidence

Uvek je dobro da pogledamo statističke podatke za data frame poput broja podataka, srednje vrednosti, medijane, kvantila itd. Ovakav pregled statistike nam omogućava funkcija *describe()*.

Za kreiranje neuronske mreže, potrebno je da odredimo ulazne i izlazne podatke. U konkretno ovom primeru, mi smo se odlučili da u ulazne podatke uđu svi redovi i sve kolone počev od 3. kolone, a da nam izlazni podaci budu poslednja kolona. *Iloc* nam omogućava da se pozicioniramo na određeni deo data frame-a.



Pošto računari razumeju samo brojeve, potrebno je da sve što nisu brojevi na neki način konvertujemo u brojeve i tako stižemo do *LabelEncodera*. Za rad sa *LabelEncoderom* moramo da koristimo biblioteku *sklearn* i njen deo *preprocessing* gde se nalazi klasa *LabelEncoder*. U našem primeru, želimo da enkodiramo kolonu *Geography* pomoću *LabelEncodera.*

Text

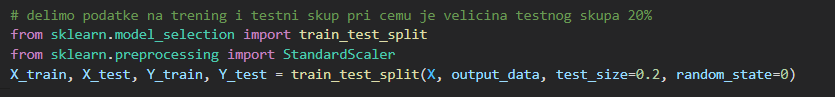
Description automatically generated

Za dalji rad će nam biti potreban *OneHotEncoder*. Uključićemo još neke biblioteke pored već uključenih biblioteka, a to su *sklearn.compose* i *sklearn.prepocessing* iz kojih ćemo pozvati klase *ColumnTransformer* i *OneHotEncoder.*

Text

Description automatically generated

Dalje je potrebno da podatke podelimo na trening i test podatke pri čemu ćemo setovati veličinu testnog skupa.



Sada je potrebno skaliranje trening podataka i za to ćemo koristiti standardizaciju koja će sve vrednosti skalirati u opsegu od -3 do 3.



Kada smo podesili sve što se tiče ulaznih podataka, možemo da krenemo na deo za kreiranje neuronske mreže. Prvo je potrebno da je inicijalizujemo.



Nakon inicijalizacije, potrebno je da dodamo slojeve u naš model.

Text

Description automatically generated

Kada smo završili sa podešavanjem mreže, možemo da krenemo na treniranje modela.

A picture containing text

Description automatically generated

Pošto je završeno treniranje mreže, možemo da kreiramo predviđanja. Kao što možemo da vidimo, model ne daje najbolja rešenja. Za bolje rezultate, potrebno je podesiti mrežu sa drugačijim parametrima. Na slici ispod je dat poređenje između stvarnog rezultata i rezultata modela.

Text

Description automatically generated

# Zaključak

Kao što ste videli na primeru, Python nudi ugodan rad sa neuronskim mrežama jer poseduje biblioteke koje nam omogućavaju laku pripremu podataka za neuronsku mrežu. Naravno, ove blblioteke nude još puno mogućnosti čime bi mogap da se poboljša rad neuronskih mreža što možete samostalno istražiti.