

Klasifikacija tipova pirinča pomoću tehnika dubokog učenja

Tehnički izveštaj

Bojana Svilenković i Đorđe Nikolić | Duboko Učenje | Septembar 2022

Sadržaj

[Uvod 2](#_Toc114433579)

[Opis projekta 2](#_Toc114433580)

[Korišćene tehnologije 2](#_Toc114433581)

[Način validacije rezultata 2](#_Toc114433582)

[Rad nad podacima izdvojenih osobina 3](#_Toc114433583)

[Analiza i preprocesiranje 3](#_Toc114433584)

[Skaliranje 5](#_Toc114433585)

[Optimizacija modela neuronske mreže 7](#_Toc114433586)

[Kontrola pojave „overfitting“ 7](#_Toc114433587)

[Aktivacione funkcije 9](#_Toc114433588)

[Konačni rezultati 10](#_Toc114433589)

# Uvod

Ovaj dokument predstavlja tehnički izveštaj i pregled projekta pod imenom „Klasifikacija tipova pirinča pomoću tehnika dubokog učenja“ i rađen je u okviru predmeta „Duboko učenje“ na Elektronskom fakultetu u Nišu.

## Opis projekta

Projekat obuhvata primenu algoritama dubokog učenja na dva povezana setova podataka koji se tiču 5 različite vrste pirinča. Oba seta podataka obuhvataju 75000 uzoraka, po 15000 za svaku vrstu pirinča. Jedan set podataka za uzorke ima slike pirinča, dok drugi sadrži po 106 izdvojenih osobina za svaku sliku. Tehnike pomoću kojih su ove osobine izdvojene nisu predmet ovog projekta, ali su opisane u već postojećem radu koji se tiče kreiranja ovih setova podataka [1]. Obzirom na različite tipove podataka u ova dva seta, biće primenjene adekvatne tehnike treniranja modela sa ciljem postizanja približnih rezultata u poređenju sa rezultatima koji su postignuti nad ovim podacima u okviru postojećih radova [2].

## Korišćene tehnologije

Projekat je razvijan pomoću programskog jezika Python, verzije 3.10. Korišćene su sledeće biblioteke:

* Numpy (koristi se u ostalim bibliotekama kao matematička potpora)
* Pandas i Openpyxl (za učitavanje i upravljanje podacima)
* Seaborn i Matplotlib (za vizualizaciju podataka i rezultata)
* Sklearn (za preprocesiranje podataka i testiranje performansi modela)
* Keras i TensorFlow (za implementacije algoritama dubokog učenja)
* OpenCV (analizu i upravljanje slikama)
* Jupyter Notebook (za predstavljanje projekta u stilu dokumenta)

## Način validacije rezultata

Obzirom na izbalansirane klase podataka, tačnost se nudi kao adekvatna metrika procene kvaliteta modela. Naravno, biće korišćene i druge metrike, naime preciznost, odziv, i F1 ocena, ali će se tačnost gledati kao glavna metrika ocenjivanja.

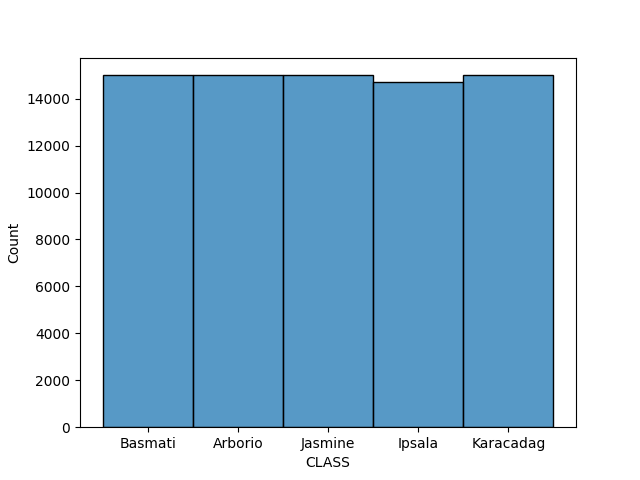
Gde je to moguće, tj. nije ometano predugim treniranjem modela, biće korišćena tehnika „Stratified K-Fold Cross Validation“ kako bi se dobile stabilne ocene kvaliteta modela.

# Rad nad izdvojenih osobina

Kako bi stekli uvid u stanje podataka koji su nam na raspolaganju, biće urađene pojedine analize podataka, a zatim i čišćenje istih. Zatim, biće proverene performanse dobijene različitim metodama skaliranja podataka, korišćenjem prostih neuronskih mreža, ali i pomoću referentnih modela stabla odluka. Nakon što je najbolji način skaliranja utvrđen, prelazimo na optimizaciju modela neuronskih mreža. Na kraju, imaćemo uvid u to koji model daje najbolje performanse.

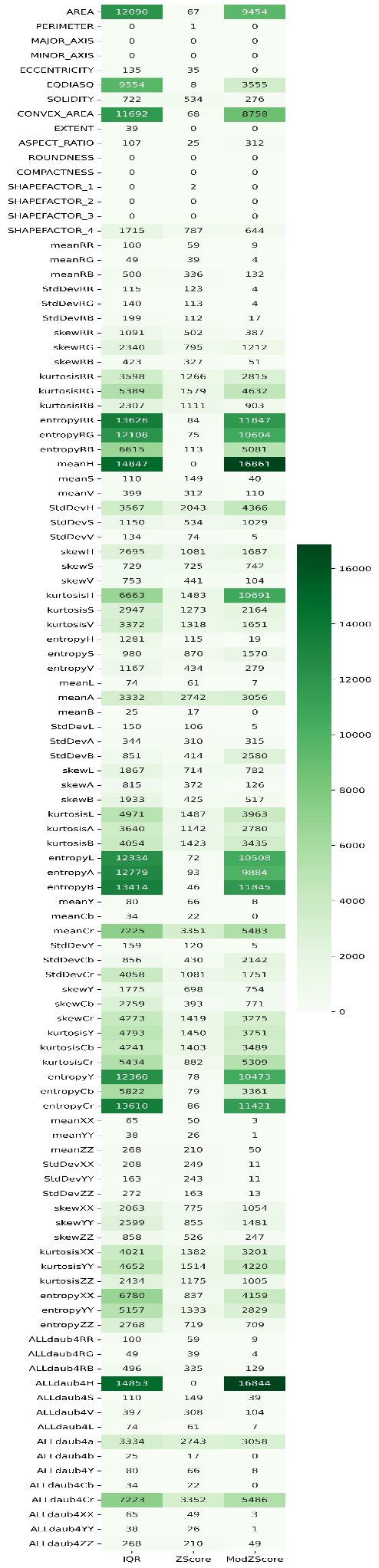
## Analiza i preprocesiranje

Kako je već pomenuto u uvodu, podaci obuhvataju 5 različitih klasa pirinča, i za svaku klasu postoje po 15000 uzoraka, dakle, klase su izbalansirane. Pronađeno je 8 uzoraka sa nepostojećim vrednostima pojedinih kolona, te su oni izbačeni. Zatim, utvrđeno je da u podacima postoje 296 duplikata uzoraka, te su i oni odbačeni. Na kraju ostaje set podataka sa 74696 uzoraka, čija je izbalansiranost klasa i dalje na veoma visokom nivou.



Slika 1. Broj uzoraka po klasama nakon čišćenja podataka

Nakon što su podaci očišćeni, dobro je proceniti količinu uzoraka koji su van ustaljenih granica, tj. „outliers“. To ćemo proceniti putem 3 različite metode: IQR, Z-Score i modifikovani Z-Score.



Slika 2. Broj uzoraka van prosečnih vrednosti

U zavisnosti od korišćenog algoritma, dobijene su dosta različite količine *outlier-a*, gde su IQR i modifikovani Z-Score dali dosta visok broj *outlier-a*, a običan Z-Score daje dosta manji broj. Svakako, ustanovljeno je da broj *outlier-a* u podacima nije zanemarljiv.

## Skaliranje

Obzirom na to da raspon vrednosti kolona nije standardizovan u datim podacima, potrebno je skalirati ih na neki način, kako bi neuronska mreža brže konvergirala. Isprobano je više metoda skaliranja, naime:

* Min Max normalization
* Standardization
* Robust Scaling
* Quantile (Uniform) transform
* Quantile (Normal) transform
* Power Transform

Svaka od ovih metoda skaliranja je isprobana sa referentnim modelom stabla odlučivanja, kao i prostim modelom neuronske mreže. Karakteristike datog modela su sledeće:

* 1 skriven sloj
* 80 neurona po sloju
* „RELU“ aktivaciona funkcija (sem izlaznog sloja gde je aktivaciona funkcija „softmax“
* Optimizator ADAM
* Računanje gubitka putem *categorical crossentropy*
* Veličina serije jednaka 100
* 3 epohe

Za treniranje i testiranje referentnih modela stabla odlučivanja korišćena je metoda „Stratified 10-Fold Cross Validation“, proces je ponovljen 10 puta i konačni rezultati predstavljaju prosek. Za testiranje modela neuronske mreže, nije korišćena prethodno navedena metoda, ali je proces takođe ponovljen 10 puta.

Na slici 3, moguće je videti rezultate testiranje, gde gornji red rezultata predstavlja ocene kvaliteta referentnog modela stabla odlučivanja, dok donji red predstavlja ocene kvaliteta modela neuronske mreže.

Rezultati su pokazali da najbolje performanse daju podaci skalirani putem standardizacije i *power transform-a*. Obzirom da će nam zbog testiranja određenih optimizacija biti potreban standardizovan set podataka, on će biti korišćen na dalje, ali na kraju rada, kada utvrdimo optimalne parametre, biće isproban i drugi set podataka dobijen putem *power transform* metode.

Graphical user interface

Description automatically generated

Slika 3. Rezultati testiranja različitih metoda skaliranja

Chart, line chart

Description automatically generated

Slika 4. Prikaz F1 ocena modela neuronske mreže po različitim metodama skaliranja

## Optimizacija modela neuronske mreže

Kako bi postigli što bolje performanse model biće izvršeno više metoda optimizacija, i na kraju ćemo kombinovati one koje daju najbolje performanse.

### Kontrola pojave „overfitting“

Sa željom smanjenja pojave *overfitting-a* (situacije kod koje model daje dobre rezultate za trening podatke ali ne i za realne podatke), biće isprobane nekoliko metoda, uključujući:

* Regularizacija težina (L2, faktor 0.01)
* Rano zaustavljanje prema gubitku
* „Dropout“ slojevi

Osobine modela koji se treniraju i testiraju ostaju iste, sem toga što će u ovom slučaju veličina serije biti 1000 i broj epoha biti 20. Podaci se jednom dele po pravilu da 30% podataka ostaje za testiranje, ali se proces ne ponavlja više puta.

Chart

Description automatically generated

Slika 5. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela bez optimizacije

Chart, histogram

Description automatically generated

Slika 6. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa regularizacijom težina

Chart, line chart

Description automatically generated

Slika 7. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa tehnikom ranog zaustavljanja

Chart

Description automatically generated

Slika 7. Rast i odnos tačnosti i gubitka kroz epohe treniranja prostog modela sa tehnikom *dropout* slojeva

Chart

Description automatically generated with low confidence

Slika 8. Rezultati testiranja performansi modela dobijenih kroz različite metode optimizacije

Gledajući rezultate, može se zaključiti da korišćenjem regularizacije težina i *dropout* slojeva modeli gube na tačnosti, dok tehnika ranog zaustavljanja ne utiče na tačnost u velikoj meri. Što je i logično, obzirom na to da ta tehnika samo zaustavlja treniranje kada u optimalnom momentu. Doduše, gledajući krivu gubitka kod modela koji ne koriste regularizaciju težina, čini se da model prebrzo konvergira, i to posle samo 1 epohe, dok je kod modela sa regularizacijom težina to dosta sporije, tek oko desete epohe. Tokom dalje optimizacije, biće korišćena tehnika ranog zaustavljanja, a na kraju će biti isprobana i kombinacija sa regularizacijom težina.

### Aktivacione funkcije

Kako bi se utvrdilo koje aktivacione funkcije daju najbolje rezultate, biće isprobane čak 11 različite funkcije za unutrašnje slojeve, i dve različite funkcije za izlazni sloj. Razlike u odnosu na osnovni model koji je ranije opisan su male:

* Koristi se tehnika ranog zaustavljanja
* Veličina serija je 1000
* 20 epoha

Graphical user interface, table, Excel

Description automatically generated

Slika 9. Rezultati testiranja različitih aktivacionih funkcija

Kako je ustanovljeno da većina aktivacionih funkcija daje slične rezultate, kako bi se dobili stabilni rezultati korišćena je tehnika „Stratified 10-Fold Cross Validation“. Rezultati su pokazali da *ReLU* i *PReLU* daju identicne rezultate, kada se primene na unutrašnje slojeve, a *softmax* funkcija daje najbolje rezultate kada se koristi na izlazni sloj.

## Konačni rezultati

Sa rezultatima koji su do sada dobijeni, mogu se doneti određeni zaključci koji nalažu kako je najbolje pripremiti model za date podatke. Ostalo je još nekoliko kombinacija koje se mogu proveriti, a to su:

* Sa ili bez regularizacije težina
* Standardizovani podaci, ili podaci skalirani putem *power transform* metode

Izgled konačne neuronske mreže pomoću koje će gorepomenute kombinacije biti testirane:

* 1 skriven sloj
* 80 neurona po sloju
* „PReLU“ aktivaciona funkcija (sem izlaznog sloja gde je aktivaciona funkcija „softmax“
* Optimizator ADAM
* Računanje gubitka putem *categorical crossentropy*
* Veličina serije jednaka 1000
* 20 epohe
* Koristi se tehnika ranog zaustavljanja

Chart, treemap chart

Description automatically generated

Slika 10. Konačni rezultati

Kao referentni rezultati uzimaju se vrednosti koje su dobijene u radu koji se pominje na početku ovog izveštaja [2], a ti rezultati su tačnost od 99.87%, koristeći običnu veštačku neuronsku mrežu, a 99.95% koristeći duboku veštačku neuronsku mrežu sa čak 19 skrivenih slojeva. Obzirom na to da su u okviru ovog projekta korišćene samo obične neuronske mreže sa samo jednim skrivenim slojem, dobijeni rezultati su i više nego adekvatni. Još jedan faktor koji nije modifikovan u okviru ovog projekta je sam optimizator neuronske mreže. Korišćen je samo ADAM algoritam, a SGD algoritam (engl. stochastic gradient descent) sa stepenom učenja kao parametrom nije isprobavan jer je sam ADAM davao dobre rezultate. Sigurno je da bi korišćenjem dubljih neuronskih mreža kao i optimizovanjem stepena učenja, u potencijalnom nastavku projekta moguće dostići, a možda i nadmašiti referentne performanse modela.

Kod pomoću kog je izvršeno prethodno istraživanje se nalazi u fajlu *FeatureAnalysis.ipynb*. U folderu *images/feature\_analysis* se mogu naći sve slike i generisane figure.

# Rad na slikama

# Bibliografija

|  |  |
| --- | --- |
| [1] | I. Cinar i M. Koklu, „Determination of Effective and Specific Physical Features of Rice Varieties by Computer Vision In Exterior Quality Inspection,“ *Selcuk Journal of Agriculture and Food Sciences,* t. 36, br. 2, 2022. |
| [2] | M. Koklu, I. Cinar i Y. Selim Taspinar, „Classification of rice varieties with deep learning methods,“ *Computers and Electronics in Agriculture,* t. 187, 2021. |