Универзитет у Крагујевцу
Факултет инжењерских наука

Семинарски рад из предмета:
ВЕШТАЧКА ИНТЕЛИГЕНЦИЈА

Тема:
Абалоне

Студент:
Милош Николић 554/2015

Професор:
др. Весна Ранковић, ред. проф.

Крагујевац 2023.

Садржај

1. Поставка задатка

Дата је поставка проблема и на:
https://archive.ics.uci.edu/ml/datasets/Abalone

Представљени проблем треба решити применом неуронских мрежа, генетским алгоритмом или неком од комбинација алгоритама вештачке интелигенције. Као програмски језик се препоручују Јава, C++ и сл. Матлаб се може искључиво користити за брзу проверу изабраног алгоритма и као верификација добијених резултата.

У самом семинарском раду обавено је:
- Објаснити задати проблем, улазне и излазне податке
- Визуализовати податке
- Објаснити коришћене алгоритме и оправдати њихово коришћење
- Дискутовати резултате и извести закључке
- Објаснити предности и мане коришћене методе

Семинарски рад је потребно послати на tijanas@kg.ac.rs (изворни код програма, извршна верзија, помоћне датотеке итд.) и предати у писаном облику најкасније до **28.05.2023.**

Писани део семинарског рада повезан у спиралу треба да садржи следеће делове:
- насловну страна са јасно написаним основним подацима
- поставку задатка
- опис делова програма са илустрацијама и самим изворним кодом
- списак коришћене литературе

Кандидат је у обавези да приликом одбране практичном демонстрацијом рада програма детаљно објасни рад појединих делова програма.

У Крагујевцу 28.03.2023. године

2. Увод

Абалоне је велики морски пуж са изузетно богатим и укусним месом. Због своје цењености и велике потражње доведен је до ивице изумирања на Западној обали у Сједињеним Америчким Државама.

Слика 1. *Абалоне (морски пуж)*

Једна од важнијих карактеристика која утиче на искористивост и цену абалонеа јесте његова старост. Да би се она одредила обично је потребно пребројавати број прстена под микроскопом, а затим на тај број додати 1,5. То је веома заморан посао који одузима доста времена. Међутим постоје мерења која су једноставнија за извршење, а могу помоћи да се старост абалонеа одреди са високом прецизношћу.

У овом задатку ће бити покушана обука неуронске мреже на тим подацима тако да буде способна да за нови, непознати тест примерак, одреди старост са што већом прецизношћу. Подаци су прикупљани мерењем више хиљада примерака.

За израду семинарског рада биће коришћен програмски језик *Python 3*, а због боље прегледности кода биће коришћен *Jupyter Notebook*.

3. Припрема података

Скуп улазних података се може преузети са сајта https://archive.ics.uci.edu/ml/datasets/Abalone при врху странице кликом на линк под називом *Data Folder*. Назив фајла јесте *Abalone.dat*. Овај фајл преводимо у датотеку са екстензијом *.csv*. У њему се не налазе називи атрибута, па их је потребно преузети кликом на линк *Data Set Description*. Када смо то урадили потребно је проучити скуп података (слика 2).

Data Set Characteristics:	Multivariate	**Number of Instances:**	4177	**Area:**	Life
Attribute Characteristics:	Categorical, Integer, Real	**Number of Attributes:**	8	**Date Donated**	1995-12-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1444140

Слика 2. *Генералне информације везане за скуп података*

Са приложене слике примећује се да постоји 8 атрибута и да су мешовитог типа (целобројни, реални, категорички), а укупан број инстанци је 4177. У овом скупу не постоје недостајуће вредности.

Атрибути су: пол, дужина, пречник, висина, укупна тежина, тежина меса, тежина утробе и тежина љуске. А вредност коју треба предвидети јесте број прстена.

3. 1. Учитавање података и елиминисање очигледних грешака при мерењу

Првобитно се увози библиотека *pandas* из које ће се позвати функција за учитавање *csv* датотеке, а затим се учитава сама датотека (слика 3). Вршимо приказ првих 5 редова како бисмо имали бољи увид у податке.

```
# Ucitavanje biblioteka
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Data frame - iz pandas-a
imena = ["Pol", "Dužina", "Prečnik", "Visina", "Ukupna težina", "Težina mesa", "Težina utrobe", "Težina ljuske", "Broj prstena"]
podaci_df=pd.read_csv('abalone.csv', names = imena) #ucitavanje podataka
podaci_df.head() #prikaz prvih 5 redova
```

Слика 3. *Учитавање података*

Када су подаци учитани потребно је проверити да ли смо успешно превели *.dat* датотеку и да ли подаци одговарају опису који је дат на сајту: https://archive.ics.uci.edu/ml/datasets/Abalone

Из излаза овог дела примећујемо да је атрибут Пол словног типа (слика 4).

	Pol	Dužina	Prečnik	Visina	Ukupna težina	Težina mesa	Težina utrobe	Težina ljuske	Broj prstena
0	M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	I	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Слика 4. *Излаз функције head()*

Кодираћемо вредности овог атрибута (код је приказан на слици 5) тако што ћемо заменити слово „M“ бројчаном вредношћу 1.0, слово „F“ бројем 1.5, а слово „I“ са 0.0. Ове бројеве смо одабрали јер примећујемо да сви остали подаци узимају вредности од 0 до 1.

```
# Kodiranje pola
podaci_df['Pol'] = podaci_df['Pol'].replace({'M': 1.0, 'F': 0.5, 'I': 0.0}) #normalizovane vrednosti
podaci_df.head()
```

	Pol	Dužina	Prečnik	Visina	Ukupna težina	Težina mesa	Težina utrobe	Težina ljuske	Broj prstena
0	1.0	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
1	1.0	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
2	0.5	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
3	1.0	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10
4	0.0	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7

Слика 5. *Кодирање атрибута Пол*

Да бисмо проверили остале информације користимо команду за опис података (слика 6).

```
# Opis podataka
podaci_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4177 entries, 0 to 4176
Data columns (total 9 columns):
Pol              4177 non-null float64
Dužina           4177 non-null float64
Prečnik          4177 non-null float64
Visina           4177 non-null float64
Ukupna težina    4177 non-null float64
Težina mesa      4177 non-null float64
Težina utrobe    4177 non-null float64
Težina ljuske    4177 non-null float64
Broj prstena     4177 non-null int64
dtypes: float64(8), int64(1)
memory usage: 293.8 KB
```

Слика 6. *Опис података*

Из приложеног се види да постоји 9 колона, од којих ће једна колона бити за класификацију. Сви осим тог атрибута су реалног типа, док је тај атрибут целобројан. Број инстанци је 4177, а обележени су индексима од 0 до 4176. Такође видимо да нема недостајућих вредности.

За још детаљнији приказ користимо код приказан на слици испод (слика 7).

```
# Opis podataka - nastavak
podaci_df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Pol	4177.0	0.522265	0.413907	0.0000	0.0000	0.5000	1.000	1.0000
Dužina	4177.0	0.523992	0.120093	0.0750	0.4500	0.5450	0.615	0.8150
Prečnik	4177.0	0.407881	0.099240	0.0550	0.3500	0.4250	0.480	0.6500
Visina	4177.0	0.139516	0.041827	0.0000	0.1150	0.1400	0.165	1.1300
Ukupna težina	4177.0	0.828742	0.490389	0.0020	0.4415	0.7995	1.153	2.8255
Težina mesa	4177.0	0.359367	0.221963	0.0010	0.1860	0.3360	0.502	1.4880
Težina utrobe	4177.0	0.180594	0.109614	0.0005	0.0935	0.1710	0.253	0.7600
Težina ljuske	4177.0	0.238831	0.139203	0.0015	0.1300	0.2340	0.329	1.0050
Broj prstena	4177.0	9.933684	3.224169	1.0000	8.0000	9.0000	11.000	29.0000

Слика 7. *Детаљнији опис података*

Оно што је важно приметити јесте да постоје елементи чија је вредност висине једнака нули. Како је то немогуће испитаћемо који су то елементи и искључити их из даљег разматрања (слика 8). Након тога потребно је ресетовати индексирање.

```
# Izbacivanje abalonea sa visinom = 0
print("Broj abalonea pre izbacivanja =", podaci_df.shape[0])
podaci_df = podaci_df.drop(podaci_df[podaci_df['Visina'] == 0].index).reset_index(drop=True)
print("Broj abalonea nakon izbacivanja =", podaci_df.shape[0])
```

```
Broj abalonea pre izbacivanja = 4177
Broj abalonea nakon izbacivanja = 4175
```

Слика 8. *Избацивање абалонеа са погрешном висином*

У наставку ћемо проверити да ли постоји грешка у евиденцији тежина. Укупна тежина абалонеа мора бити већа од тежина меса, утробе и љуске. Разлог зашто није једнака јесте губитак течности, односно крви при отварању и чишћењу абалонеа.

На слици испод (слика 9) видимо да се грешка десила у великом броју случајева (154).

```
# Izbacivanje abalonea sa greskom u evidenciji tezine.
# Ukupna tezina mora biti veca od pojedinacnih (desava se gubitak tecnosti pri otvaranju i ciscenju)
upit = podaci_df['Ukupna težina'] < podaci_df['Težina mesa'] + podaci_df['Težina utrobe'] + podaci_df['Težina ljuske']
podaci_df = podaci_df.drop(podaci_df[upit].index).reset_index(drop=True)
print("Broj abalonea nakon novog izbacivanja =", podaci_df.shape[0])
```

```
Broj abalonea nakon novog izbacivanja = 4021
```

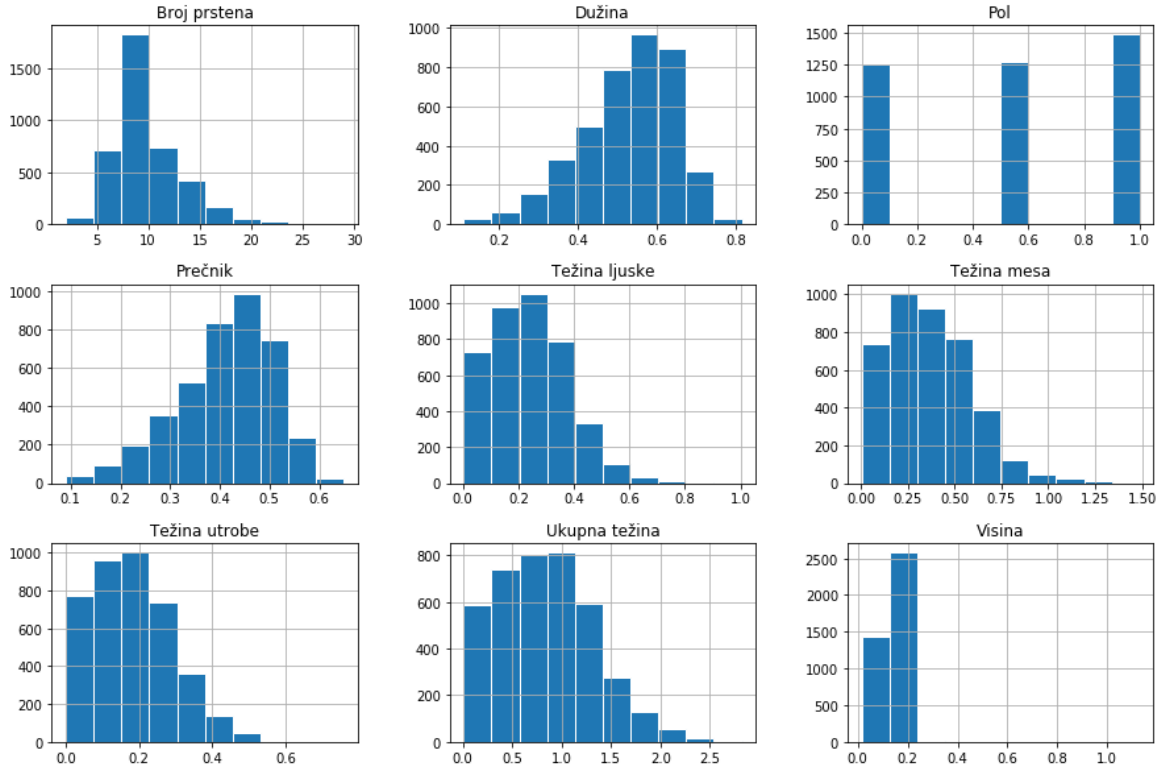
Слика 9. *Избацивање абалонеа са погрешном евиденцијом тежина*

3. 2. Визуелизација и обрада података

Из хистограма испод (салика 10) видимо да је расподела података углавном нормална и да немамо вредности које драстично одскачу. Према томе у наставку ћемо сматрати да су подаци сада исправни и наставити са даљим током рада.

```
# Histogrami za sve atribute
podaci_df.hist(figsize=(15, 10), edgecolor='white')
```

```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x00000000152A0788>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000000013CB80C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x0000000001498B3C8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x0000000015304DC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000000001533E7C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000000153781C8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000000001539FC88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000000153E78C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x00000000153F3488>]],
      dtype=object)
```



Слика 10. *Хистограми за све атрибуте*

Проверићемо колико имамо јединствених класа (слика 11).

```
# Provera broja klasa - broj prstena
broj_klasa = podaci_df['Broj prstena'].nunique()
print("Broj klasa (broj jedinstvenih prstenova) =", broj_klasa)
```

```
Broj klasa (broj jedinstvenih prstenova) = 27
```

Слика 11. *Број класа*

Видимо да је број класа велики, а бројчана разлика између њих мала, односно једнака јединици. У таквом случају било би веома тешко исправно класификовати те податке. Због тога је битно да смањимо број класа на логички исправан начин који неће утицати на примену у пракси.

Како се абалоне користи у исхрани идеја за поделу на класе која се намеће јесте подела на младе, зреле и старе. Да бисмо утврдили да ли је ова подела могућа и исправна потребно је извршити пар испитивања.

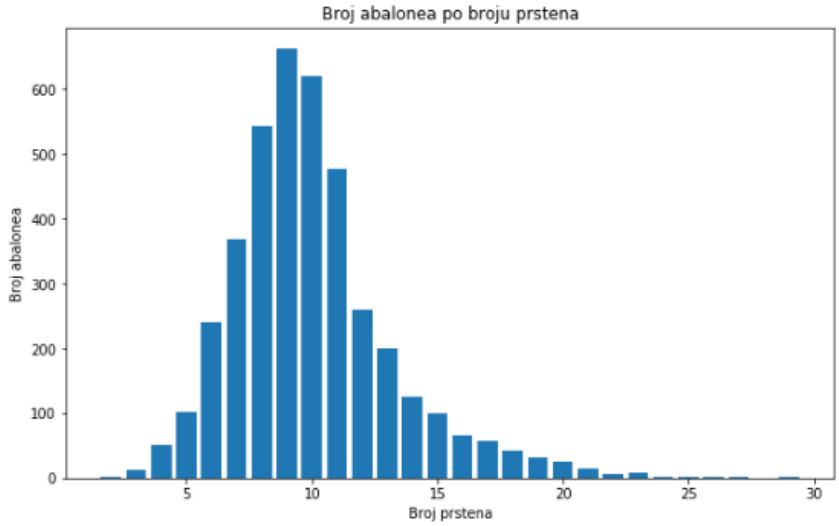
Са следеће две слике (слика 12 и слика 13) погледаћемо број абалонеа по броју прстена.

```
# Broj abalonea po broju prstena
broj_prstena = podaci_df['Broj prstena'].value_counts().sort_index()
print("Broj abalonea po broju prstena:\n")
print(broj_prstena)
```

```
Broj abalonea po broju prstena:

2       1
3      12
4      50
5     102
6     240
7     369
8     543
9     662
10    620
11    478
12    260
13    199
14    126
15    100
16     66
17     58
18     42
19     31
20     26
21     14
22      6
23      9
24      2
25      1
26      1
27      2
29      1
Name: Broj prstena, dtype: int64
```

Слика 12. *Број абалонеа по броју прстена*

```
# Graficka predstava broja abalonea po broju prstena
plt.figure(figsize=(10, 6))
plt.bar(broj_prstena.index, broj_prstena.values)
plt.xlabel('Broj prstena')
plt.ylabel('Broj abalonea')
plt.title('Broj abalonea po broju prstena')
plt.show()
```

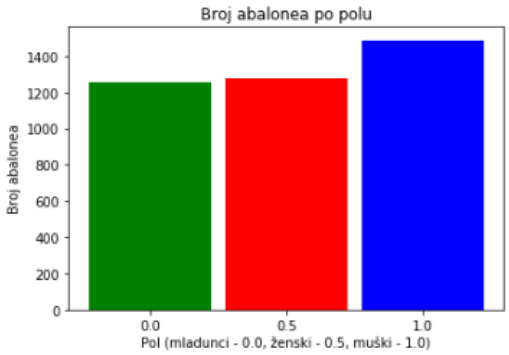


Слика 13. *График броја абалонеа по броју прстена*

На основу двеју слика видимо да велика већина абалонеа има од 3 до 23 прстена. Остали број прстена има једна или две јединке. Према томе највише пажње при подели на категорије ћемо обраћати управо на овај део.

У наставку ћемо погледати број абалонеа по полу (слика 14).

```
# Broj abalonea po polu
clanovi_po_polu = podaci_df['Pol'].value_counts()
boje = ['blue', 'red', 'green']

# Grafik
plt.bar(clanovi_po_polu.index, clanovi_po_polu.values, width = 0.45, color=boje)
plt.xlabel('Pol (mladunci - 0.0, ženski - 0.5, muški - 1.0)')
plt.ylabel('Broj abalonea')
plt.title('Broj abalonea po polu')

# Podeoci na x osi
plt.xticks((0.0, 0.5, 1.0))
plt.show()
```



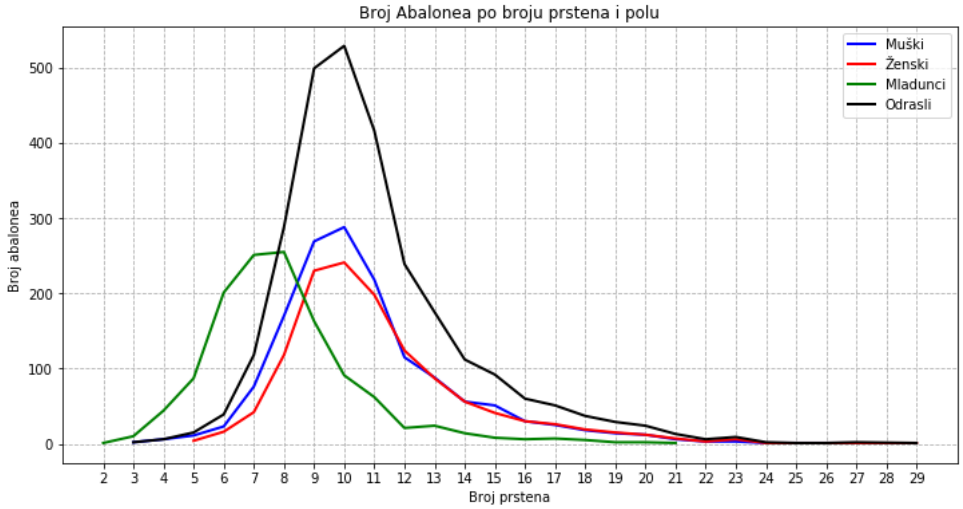
Слика 14. *График броја абалонеа по полу*

Видимо да је број јединки по полу сличан узимајући у обзир количину података. Како се при самом мерењу наглашава важност младунаца унутар пола, било би пожељно да пик њихових чланова припада категорији младих. Остатак мушких и женских јединки ћемо привремено спојити у још једну групу под називом одрасли.

Сада ћемо приказати број абалонеа по броју прстена и полу (заједно са одраслима). Код (слика 15) и график (слика 16) су дати у наставку.

```
# Podaci po polu
m_podaci_df = podaci_df[podaci_df['Pol'] == 1.0]
z_podaci_df = podaci_df[podaci_df['Pol'] == 0.5]
ml_podaci_df = podaci_df[podaci_df['Pol'] == 0.0]

# Podaci za odrasle
o_podaci_df = podaci_df[podaci_df['Pol'] > 0.0]

# Broj prstena po polu
m_prsteni = m_podaci_df['Broj prstena'].value_counts().sort_index()
z_prsteni = z_podaci_df['Broj prstena'].value_counts().sort_index()
ml_prsteni = ml_podaci_df['Broj prstena'].value_counts().sort_index()
o_prsteni = o_podaci_df['Broj prstena'].value_counts().sort_index()

plt.figure(figsize=(12, 6))

# Grafici
plt.plot(m_prsteni.index, m_prsteni.values, color='blue', linewidth=2, label='Muški')
plt.plot(z_prsteni.index, z_prsteni.values, color='red', linewidth=2, label='Ženski')
plt.plot(ml_prsteni.index, ml_prsteni.values, color='green', linewidth=2, label='Mladunci')
plt.plot(o_prsteni.index, o_prsteni.values, color='black', linewidth=2, label='Odrasli')

plt.xlabel('Broj prstena')
plt.ylabel('Broj abalonea')
plt.title('Broj Abalonea po broju prstena i polu')

plt.legend()
plt.grid(True, linestyle='--')

# Podeoci na x osi
plt.xticks(range(min(podaci_df['Broj prstena']), max(podaci_df['Broj prstena'])+1))

plt.show()
```

Слика 15. *Код за приказ графика: број абалонеа по броју прстена и полу*



Слика 16. *График: број абалонеа по броју прстена и полу*

Видимо да је облик графика за абалоне мушког и женског пола сличан, па можемо да кажемо да група „одрасли“ добро репрезентује ове групе.

Како бисмо укључили пикове младунаца у категорију младих, категорија младих ће обухватати све абалоне са 8 и мање прстена, а чланове одраслих и (делове младунаца) ћемо поделити на зреле и старе. У поставци задатка наведено је да је број прстена довољан податак да бисмо одредили старост, па категорије можемо поделити на једнаке делове без обзира на укупну расподелу броја прстена. Како смо претходно утврдили да најзначајнији број абалонеа има до 23 прстена и следећа категорија „зрели“ ће обухватити 8 прстена. Категорија „стари“ ће обухватити осталих 7 прстена као и остатак који ће на неки начин представљати допуну до осмог прстена. Старосна категорија биће додата у новој колони (слика 17), а колона са бројем прстена биће избачена из *data frame*-а (слика 18).

```
# Podela na starosne kategorije - smanjenje klasifikacije
uslovi = [
    podaci_df['Broj prstena'] <= 8, #mladi
    (podaci_df['Broj prstena'] > 8) & (podaci_df['Broj prstena'] <= 16), #zreli
    podaci_df['Broj prstena'] > 16 #stari
]
vrednosti = [0, 1, 2] #mladi, zreli, stari

# Dodavanje nove kolone - starost
podaci_df['Starost'] = pd.Series(pd.Categorical(np.select(uslovi, vrednosti), ordered=True))

podaci_df.head(1)
```

	Pol	Dužina	Prečnik	Visina	Ukupna težina	Težina mesa	Težina utrobe	Težina ljuske	Broj prstena	Starost
0	1.0	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15	1

Слика 17. *Подела на старосне категорије*

```
# Izbacivanje prstena
podaci_df = podaci_df.drop(['Broj prstena'], axis=1)
podaci_df.head(1)
```

	Pol	Dužina	Prečnik	Visina	Ukupna težina	Težina mesa	Težina utrobe	Težina ljuske	Starost
0	1.0	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	1

Слика 18. *Избацивање прстена*

У кораку 3.1. већ смо обавили одређену обраду података када смо избацили очигледне грешке и кодирали називе полова.

4. Неуронска мрежа

4. 1. Подела скупа података

Пре него што направимо неуронску мрежу потребно је поделити скуп података. Дефинисаћемо нове скупове и то:

- Тренинг скуп – који ће садржати 80% података
- Тест скуп – који ће садржати 20% података

За чување оба скупа биће потребне по две променљиве. Прва променљива ***X*** чуваће улазне, односно независне податке, док ће друга променљива ***y*** чувати излазне, односно зависне променљиве (слика 19).

```
from sklearn.model_selection import train_test_split

# Podela podataka na nezavisne i zavisne podatke
X = podaci_df.drop('Starost', axis=1)
y = podaci_df['Starost']

# Podela podataka na trening i test skup
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Слика 19. *Подела података на тренинг и тест скуп*

4. 2. Избор најбољих параметара

За решавање задатка коришћен вишеслојни перцептрон класификатор (енг. *Multi-layer Perceptron classifier*) - ***MLPClassifier***.

Овај класификатор примењује алгоритам вишеслојног перцептрона који тренира мрежу користећи пропагацију грешке уназад (енг. *Backpropagation*). Предложени су следећи параметри (слика 20):

```
#Trening skup - pronalazak najboljih parametara
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import GridSearchCV

parametri = {'solver': ['lbfgs', ], 'max_iter': [500, 1000, 1500], 'alpha': 10.0 ** -np.arange(1, 10),
             'hidden_layer_sizes':np.arange(4, 6), 'random_state':[10]}
```

Слика 20. *Тренирање мреже различитим комбинацијама параметара*

Функција ***GridSearchCV*** пролази кроз све могуће комбинације задатих параметара, а када се позове функција ***fit*** (слика 21) враћа најбољу могућу комбинацију.

```
mplc = GridSearchCV(MLPClassifier(), parametri, n_jobs=-1)
mplc = mplc.fit(X_train, y_train)
print("Tacnost na trening skupu: ", mplc.score(X_train, y_train))
print("Najbolji parametri:\n", mplc.best_params_)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\model_selection\_split.py:1978: FutureWarning: The default value of cv will
change from 3 to 5 in version 0.22. Specify it explicitly to silence this warning.
  warnings.warn(CV_WARNING, FutureWarning)

Tacnost na trening skupu:  0.7944651741293532
Najbolji parametri:
 {'alpha': 0.1, 'hidden_layer_sizes': 5, 'max_iter': 500, 'random_state': 10, 'solver': 'lbfgs'}
```

Слика 21. *Враћање најбоље комбинације и штампање тачности на тренинг скупу*

4. 3. Тестирање мреже

Тачност мреже на тест скупу дата је испод (слика 22).

```
#Test skup - odredjivanje tacnosti
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

y_test_pred = mplc.predict(X_test)
test_tacnost = accuracy_score(y_test, y_test_pred)

print('Tacnost: ', test_tacnost)
```

```
Tacnost:  0.8049689440993789
```

Слика 22. *Тачност на тест скупу*

На слици испод приказана је конфузиона матрица и извештај о класификацији (слика 23).

```
# Konfuziona matrica
konfuziona_matrica = confusion_matrix(y_test, y_test_pred)
print(konfuziona_matrica)
```

```
[[203  89   0]
 [ 28 436   4]
 [  0  36   9]]
```

```
# Izvestaj o klasifikaciji
print(classification_report(y_test, y_test_pred))
```

```
              precision    recall  f1-score   support

           0       0.88      0.70      0.78       292
           1       0.78      0.93      0.85       468
           2       0.69      0.20      0.31        45

    accuracy                           0.80       805
   macro avg       0.78      0.61      0.64       805
weighted avg       0.81      0.80      0.79       805
```

Слика 23. *Конфузиона матрица и извештај о класификацији*

5. Закључак

На основу добијених резултата прецизности на тест скупу, конфузионе матрице и извештаја о класификацији може се рећи да је неуронска мрежа успешна у класификацији задатог проблема.

Најмањи проценат погодака има на класи број 2, односно на старим абалонима. Овој класи између осталих припадају јединке које су у поставци имале више од 23 прстена. Како смо раније видели за сваки прстен већи од 23 постојала је само једна или две јединке, па је вероватно постојала грешка при прикупљању ових података.

6. Литература

1. Курс: Вештачка интелигенција, доступно на: http://moodle.fink.rs/course/view.php?id=989, приступљено 28.05.2023.]
2. Сајт: Anaconda Documentation, доступно на: https://docs.anaconda.com, приступљено 28.5.2023.]
3. Сајт: Jupyter Project Documentation, доступно на: https://docs.jupyter.org, приступљено 28.5.2023.]
4. Сајт: 3.11.3 Documentation, доступно на: https://docs.python.com, приступљено 28.5.2023.]
5. MLPClassifier, доступно на: https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html , приступљено 28.05.2023.]