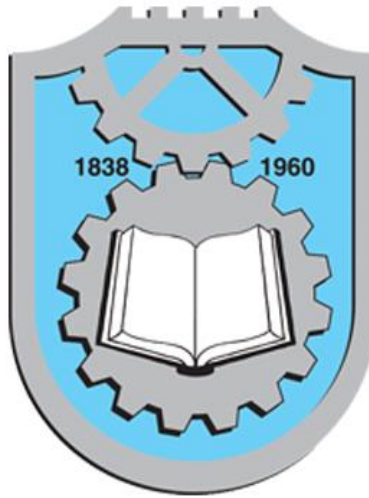


Факултет инжењерских наука

Универзитет у Крагујевцу



## Вештачка интелигенција

Семинарски рад:

*Коришћење неуронских мрежа за предвиђање  
старости морских пужева*

Студент:  
Лука Момчиловић 616/2018

Предметни наставници:  
Весна Ранковић  
Тијана Шуштершич

## Садржај

1. Објашњење проблема .....	3
2. Подаци .....	3
3. Визуализација података .....	4
4. Обрада података .....	5
5. Подела на тренинг и тест скуп .....	6
6. Различити приступи решавању проблема .....	6
7. Креирање различитих модела неуронске мреже .....	7
8. Перформансе класификационог модела .....	8
9. Перформансе регресионог модела .....	9
10. Поређење перформанси .....	10
11. Литература .....	11

## 1. Објашњење проблема

Потребно је истренирати неуронску мрежу која за дате физичке карактеристике морских пужева може да предвиди број прстена на попречном пресеку шкољке пужа. Научници користе ове прстенове да би приближно одредили старост морског пужа (број прстенова + 1,5 = број година морског пужа).

## 2. Подаци

Скуп података садржи 4177 тренинг примера и нема недостајућих вредности. Сваки пример садржи 9 атрибута (8 улазних и један излазни).

Улазни подаци:

- Пол – може имати 3 вредности: М (мужјак), F (женка) или I (младунче)
- Дужина, ширина и висина (изражене у милиметрима)
- Укупна маса, маса меса, маса шкољке и маса изнутрица (изражене у грамима)

Излазни подаци:

- Број прстенова

	Sex	Length	Diameter	Height	Whole weight	Shucked weight	Viscera weight	Shell weight	Rings
0	M	0.455	0.365	0.095	0.514	0.2245	0.101	0.15	15
1	M	0.35	0.265	0.09	0.2255	0.0995	0.0485	0.07	7
2	F	0.53	0.42	0.135	0.677	0.2565	0.1415	0.21	9
3	M	0.44	0.365	0.125	0.516	0.2155	0.114	0.155	10
4	I	0.33	0.255	0.08	0.205	0.0895	0.0395	0.055	7
5	I	0.425	0.3	0.095	0.3515	0.141	0.0775	0.12	8
6	F	0.53	0.415	0.15	0.7775	0.237	0.1415	0.33	20
7	F	0.545	0.425	0.125	0.768	0.294	0.1495	0.26	16
8	M	0.475	0.37	0.125	0.5095	0.2165	0.1125	0.165	9
9	F	0.55	0.44	0.15	0.8945	0.3145	0.151	0.32	19
10	F	0.525	0.38	0.14	0.6065	0.194	0.1475	0.21	14
11	M	0.43	0.35	0.11	0.406	0.1675	0.081	0.135	10
12	M	0.49	0.38	0.135	0.5415	0.2175	0.095	0.19	11
13	F	0.535	0.405	0.145	0.6845	0.2725	0.171	0.205	10
14	F	0.47	0.355	0.1	0.4755	0.1675	0.0805	0.185	10
15	M	0.5	0.4	0.13	0.6645	0.258	0.133	0.24	12

Слика 1 – табеларни приказ података

### 3. Визуализација података

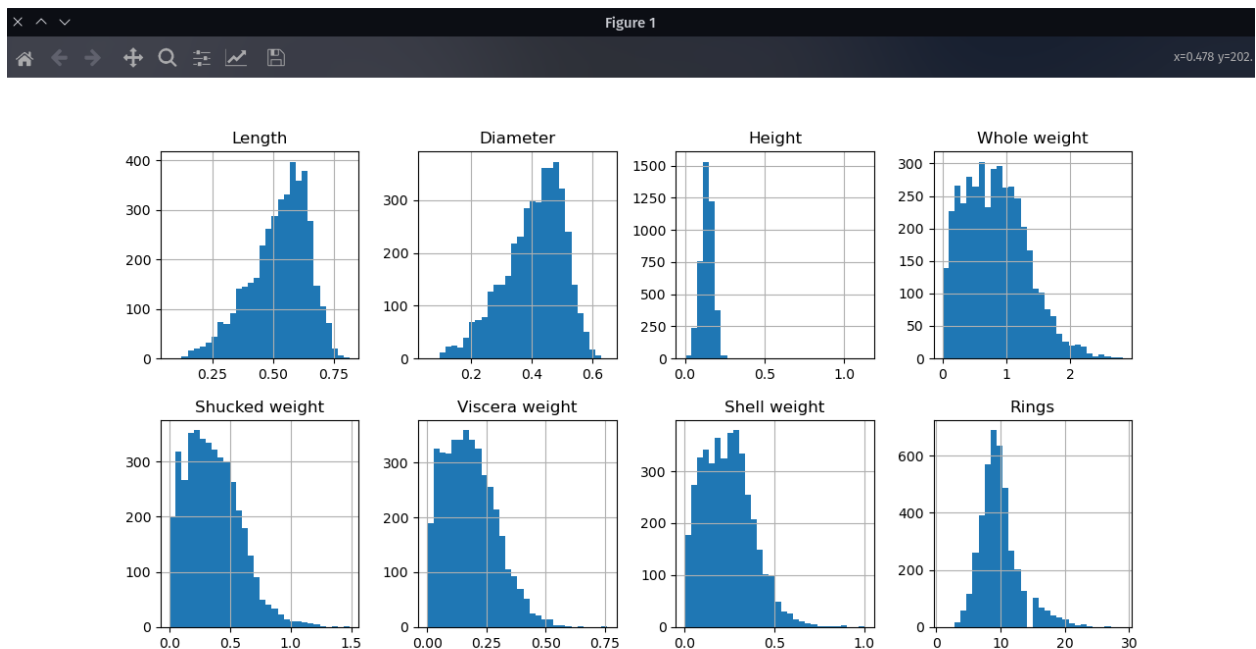
Визуализација података одрађена је у функцији `data_visualization`, као аргумент прима променљиву класе `DataFrame` и нема повратну вредност. Функцији се прослеђују необрађени подаци уčitани из датотеке.

```
def data_visualization(data):
    data_table_file = open("data/abalone.table", "w")
    data_table_file.write(tabulate(data, headers='keys', tablefmt='psql')) #tabelarni prikaz podataka

    data.hist(figsize=(20, 20), grid=True, layout=(2, 4), bins=30) #histogram
    plt.show()

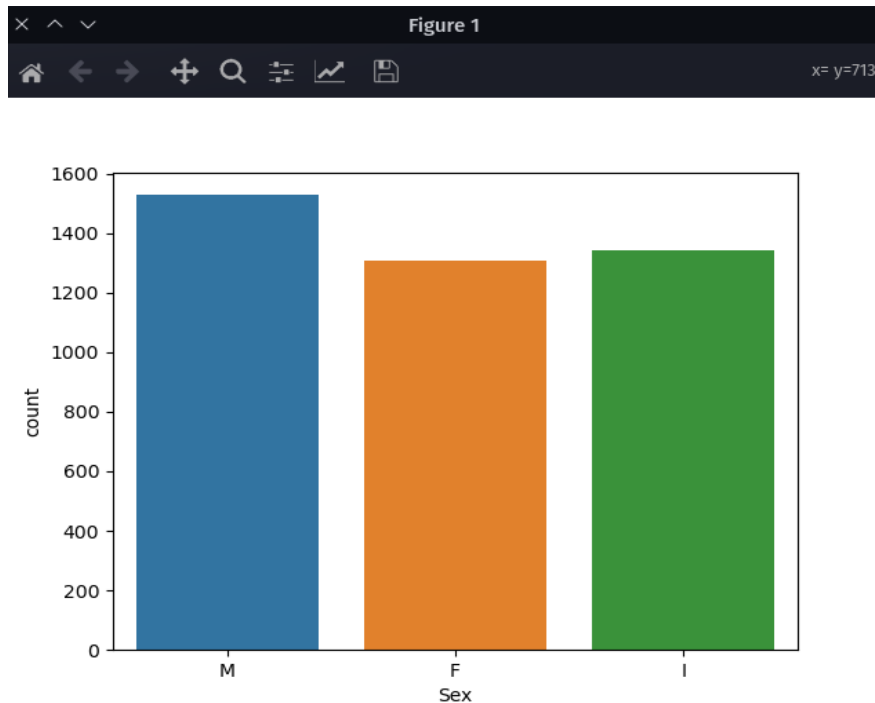
    sns.countplot(x='Sex', data=data) #distribucija polova
    plt.show()
```

Слика 2 – функција за визуализацију података



Слика 3 – дистрибуција података

Са претходног хистограма треба обратити посебну пажњу на дистрибуцију прстенова. Са хистограма се види да је највише пужева старости између 5 и 15 година.



Слика 4 – дистрибуција полова

Полови су готово једнако заступљени. Сматра се да јединка достиже полну зрелост и престаје да буде младунче негде између пете и седме године.

#### 4. Обрада података

Обрада података извршена је у функцији `load_and_preprocess_data` која не прима ни један параметар, а враћа обрађене податке класе `ndarray`, подељене на улазне и излазне (X и Y).

Поред обраде, у функцији се и учитавају подаци из датотеке и позива се функција `data_visualization` која је описана у претходном поглављу.

```
def load_and_preprocess_data():
    data = pd.read_csv("data/abalone.data") #ucitavanje podataka iz datoteke
    data_visualization(data) #vizualizacija podataka

    one_hot_gender = pd.get_dummies(data['Sex'], prefix='Sex')
    data = data.drop('Sex', axis=1)
    data = data.join(one_hot_gender) #menjanje kolone 'Sex' one hot vrednostima

    X = data.drop('Rings', axis=1).values.astype(float) #podela podataka na ulazne
    Y = data['Rings'].values.astype(float) # i izlazne

    scale = StandardScaler() #skaliranje podataka
    X[:, 0:7] = scale.fit_transform(X[:, 0:7]) #one hot vrednosti se ne skaliraju

    return X, Y
```

Слика 5 – функција `load_and_preprocess_data`

Пошто пол није представљен нумеричким вредностима потребно је извршити „one hot“ кодирање ове колоне, где се за сваку могућу вредност ове колоне (M, F или I) додаје по једна нова колона чија ће вредност бити 0 или 1, у зависности од првобитне вредности.

Након тога, подаци се деле на излазне (број прстенова, Y) и улазне (сви остали, X).

На крају је потребно извршити скалирање свих улазних података да би се убрзао процес учења. One hot кодиране колоне се не скалирају.

## 5. Подела на тренинг и тест скуп

Подаци се деле на тренинг скуп (подаци који служе за тренирање параметара неуронске мреже) и тест скуп (подаци који служе за тестирање перформанси неуронске мреже након њеног тренирања). Након поделе тренинг скуп ће обухватати 80% првобитног скупа података, а тест скуп 20%.

```
X, Y = load_and_preprocess_data()  
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=1)
```

Слика 6 – учитавање података из датотеке и подела на тренинг и тест скуп

## 6. Различити приступи решавању проблема

Проблем предвиђања старости морских пужева може се посматрати као регресиони или као класификациони проблем, од чега зависи и архитектура саме неуронске мреже.

У првом случају ће излазни слој чинити само један неурон чији ће излаз бити реалан број, функција губитака ће бити средња квадратна грешка, а оптимизујући параметар ће бити средња апсолутна грешка. Код овог приступа није циљ да се погоди тачан број прстенова, већ да се направи добра процена старости морског пужа.

У другом случају излазни слој ће чинити 30 неурона, где сваки представља вероватноћу да морски пуж има одређен број прстена, након чега softmax активациона функција враћа редни број оног са највећом вероватноћом. Функција губитака ће, у овом случају, бити sparse categorical cross entropy, а оптимизујући параметар прецизност (да ли је мрежа тачно предвидела број прстена или не). Ово значи да ће се мрежа максимално трудити да погоди тачан број прстенова, али и да ће мале и велике грешке у процени посматрати исто.

## 7. Креирање различитих модела неуронске мреже

Функција `create_model` има задатак да креира модел неуронске мреже који ће се касније користити за предвиђање броја прстенова. Од параметара прима листу чија дужина представља број скривених слојева, а вредности у листи број неурона у сваком скривеном слоју, као и тип излазног слоја, што одређује да ли ће мрежа овај проблем посматрати као регресиони или као класификациони. Повратна вредност функције је структура неуронске мреже.

```
def create_model(hidden_layers, output_layer='linear'):  
    model = tf.keras.Sequential()  
  
    for layer_size in hidden_layers:  
        model.add(tf.keras.layers.Dense(  
            layer_size,  
            activation='leaky_relu',  
            kernel_initializer='glorot_uniform',  
            bias_initializer='glorot_uniform'  
        ))  
  
    if output_layer == 'softmax':  
        model.add(tf.keras.layers.Dense(  
            30,  
            activation='leaky_relu',  
            kernel_initializer='glorot_uniform',  
            bias_initializer='glorot_uniform'  
        ))  
        model.add(tf.keras.layers.Softmax(-1))  
        loss = 'sparse_categorical_crossentropy'  
        metrics = ['accuracy']  
    else:  
        model.add(tf.keras.layers.Dense(  
            1,  
            activation='linear',  
            kernel_initializer='glorot_uniform',  
            bias_initializer='glorot_uniform'  
        ))  
        loss = "mean_squared_error"  
        metrics = ['mean_absolute_error']  
  
    model.compile(  
        optimizer='adam',  
        loss=loss,  
        metrics=metrics  
    )  
    return model
```

Слика 7 – функција `create_model`

## 8. Перформансе класификационог модела

Класификациони модел је на тренинг скупу тачно предвидео број прстенова за 30.20% морских пужева, док је на тест скупу имао прецизност од 26.79%.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
dense (Dense)	(3341, 16)	176
dense_1 (Dense)	(3341, 8)	136
dense_2 (Dense)	(3341, 6)	54
dense_3 (Dense)	(3341, 4)	28
dense_4 (Dense)	(3341, 2)	10
dense_5 (Dense)	(3341, 30)	90
softmax (Softmax)	(3341, 30)	0

```
=====  
Total params: 494  
Trainable params: 494  
Non-trainable params: 0
```

Слика 8 – структура класификационог модела



## 9. Перформансе регресионог модела

Средња апсолутна грешка регресионог модела је 1.4168 на тренинг скупу, односно 1.4297 на тест скупу.

```
Model: "sequential"
```

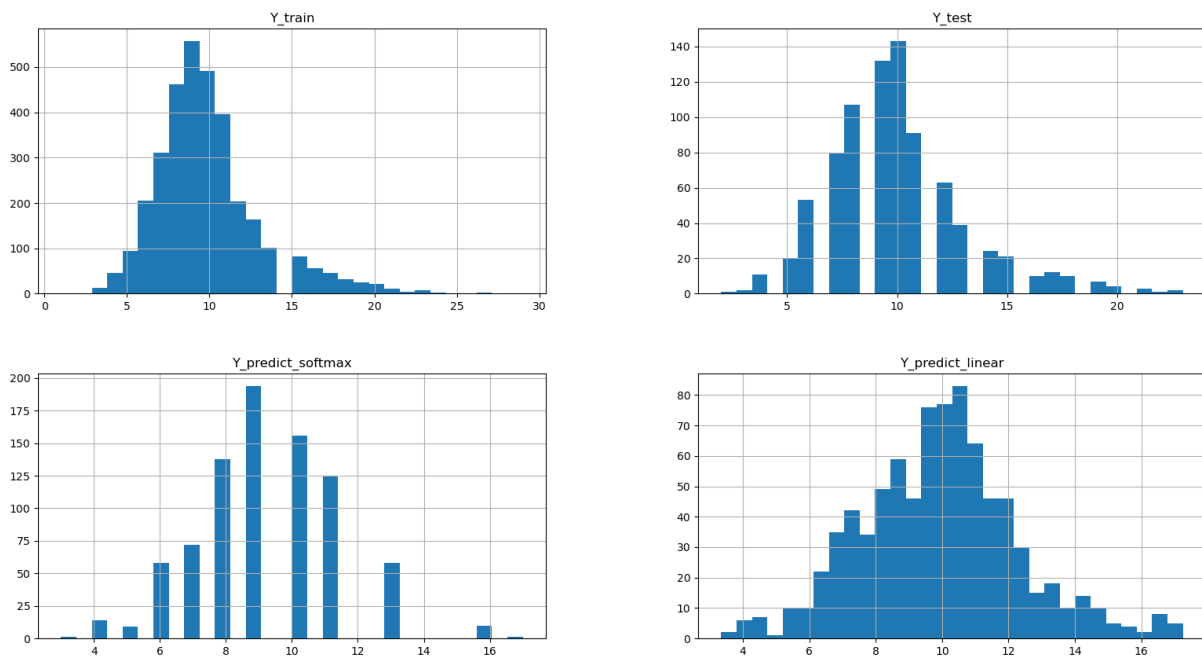
Layer (type)	Output Shape	Param #
dense (Dense)	(3341, 32)	352
dense_1 (Dense)	(3341, 16)	528
dense_2 (Dense)	(3341, 16)	272
dense_3 (Dense)	(3341, 8)	136
dense_4 (Dense)	(3341, 8)	72
dense_5 (Dense)	(3341, 4)	36
dense_6 (Dense)	(3341, 1)	5

```
Total params: 1,401  
Trainable params: 1,401  
Non-trainable params: 0
```

Слика 9 – структура регресионог модела

## 10. Поређење перформанси

На хистограму можемо да уочимо да ни регресиони ни класификациони модел нису превише успешни у предвиђању екстремних вредности (број прстенова $<4$  или број прстенова $>20$ ). Ипак, главни недостатак класификационог модела је што све грешке посматра исто, док регресиони модел настоји да минимизује грешке.



Слика 10 – хистограм дистрибуција тренинг и тест скупова и предвиђања модела

## 11. Литература

- <https://www.coursera.org/specializations/deep-learning> - принцип рада и примена неуронских мрежа
- [https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras) - библиотека за једноставно и брзо креирање неуронских мрежа
- <https://matplotlib.org/> - библиотека за визуализацију података
- <https://pandas.pydata.org/> - библиотека за учитавање и обраду података
- <https://towardsdatascience.com/predicting-the-survival-of-titanic-passengers-30870ccc7e8> - пример python апликације за машинско учење